Dynamic probabilistic graphical model applied to the system health diagnosis, prognosis, and the remains useful life estimation

Kamrul Islam Shahin

Ecole doctorale IAEM Lorraine

Modèle graphique probabiliste appliqué au diagnostic de l'état de santé des systèmes, au pronostic et à l'estimation de la durée de vie résiduelle

THÈSE

Présentée et soutenue publiquement le 10 novembre 2020

Pour l'obtention du

Doctorat de l'Université de Lorraine
(Automatique, Traitement du signal et des images, Génie informatique)

par

Kamrul SHAHIN

Composition du jury

Rapporteurs:	Christophe Bérenguer	Professeur, Grenoble-INP, GIPSALAB
	Bruno Castanier	Professeur, University of Angers, LARIS
Examinateurs :	Zohra Cherfi Boulanger	Professeur, University of Technology of Compiègne, MS2T
	Zineb Simeu-Abazi	Professeur, Polytech Grenoble, G-SCOP
	Dewan Md-Farid	Associate Professeur, United International University, Bangladesh
	Benoit Iung	Professeur, University of Lorraine, CRAN
	Christophe Simon	Maître de conférences, University of Lorraine, CRAN (Thesis co-director)
	Philippe Weber	Professeur, University of Lorraine, CRAN (Thesis director)

CRAN UMR 7039, Université de Lorraine, CNRS

To mom, dad, and my little brother,

You have helped me to see this dream through a lot of sacrifice and patience, and in return, I am able to fulfill your wish today. This is not only my accomplishment but also yours in equal.

To Philippe and Christophe,

I will continue making you proud by digging into the scope of contribution to science. I will not waste your advice and guidance. -- I promise!

Acknowledgements

Thanks to the almighty who gave me the opportunity to defend my Ph.D. in time as intended.

First of all, I would like to thank Christophe Bérenguer and Bruno Castanier for agreeing to report this thesis as well as Zohra Cherfi Boulanger, Zineb Simeu-Abazi, Dewan Md-Farid, and Benoit Iung for agreeing to examine it. I really enjoyed discussing the thesis with them during the book review and my presentation.

I would like to thank especially Dr. Christophe Simon who co-directed this thesis throughout the three years. Thank you for the many discussions I had with him, and for the time he agreed to devote to me despite the crucial pandemic time through online advising. His perspective on my work has always been very enriching. I am also very thankful to prof. Philippe Weber who directed this thesis. He was very kind, friendly, and very welcoming from the first day of this journey to the very end. All of his remarks and ideas contributed particularly to the quality of the work done. Both of my director and co-director, who have monitored this thesis throughout these 3 years, for all the advice, ideas, and corrections they have provided, but also for the patience they have always shown. I also thank both of them for the confidence they have placed in my work, and it is with pleasure that I will continue to collaborate with them after this thesis.

Many thanks to all the great colleagues who have helped me throughout these three years (Didier Theilliol, Fara Gabsi, Mahdi, Didi, Shoaib, Nahin, Sara). They have ensured a convivial and enriching atmosphere, which has greatly contributed to the very good development of this theme. Thanks to the master's students whom I had the chance to review their thesis projects which were helpful to develop my literature review.

Thanks to all the office staff and technical staff for their help and support. Special thanks to my friends (Abedil, Tehreem, Abir, Tonu) and all the well-wishers for their suggestions and cheers.

Finally, thanks to my mother, father, and my little brother who supported me and encouraged me from the beginning.

Contributions (3 Journals and 8 Conference papers)

Vous trouverez ci-dessous la liste des documents constituant ma production scientifique avec le résumé en anglais et en français du contenu de l'article. 3 articles de journaux sont prêts à être soumis et nous avons identifié les journaux cibles. 8 articles ont été acceptés dans des conférences nationales ou internationales. 3 posters ont également été produits, mais n'ont pas été détaillés dans le récapitulatif de production.

01.

Journal 01: Engineering Applications of Artificial Intelligence (To be submitted)

IOHMM for Estimating the Remaining Useful Life of an Aircraft Engine Under Multiple Operating Conditions

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

***Keywords*:** Input Output Hidden Markov Model, Degradation, Diagnostic, Prognostic, RUL.

Abstract: This paper proposes an Input-Output Hidden Markov Model (IOHMM) to describe how the remaining useful life (RUL) of aircraft gas turbine engines can be estimated under multiple operating conditions. The PHM data challenge 2008 is used to design the system with the IOHMM. In this paper, multiple inputs and multiple outputs are considered through the proposed model. The thermodynamic simulation model generated the data of all sensors as a function of variations of flow and the efficiency of the modules concerned. The exponential rate of flow variation and efficiency loss was established in each data set, starting at a randomly selected initial deterioration setpoint. The data set is in two parts: the training set and the testing set. The training set is used to estimate the model parameters. Well-known algorithms dedicated to Hidden Markov Model (HMM) are adapted to train IOHMM. Finally, the learned model is applied to the testing set for predicting the RUL of the system. Prognostic RUL considering both the offline and online operations is presented with a numerical example (Shahin, 2020).

Résumé : Ce document propose un modèle de Markov caché à entrées-sorties (IOHMM) pour décrire comment la durée de vie résiduelle (RUL) des moteurs à turbine à gaz d'avion peut être estimée avec de multiples conditions de fonctionnement. Les données du PHM data challenge 2008 sont utilisées pour concevoir le modèle d'estimation par IOHMM. Dans ce document, de multiples entrées et sorties sont prises en compte par le modèle proposé. Dans le PHM data challenge 2008, un modèle de simulation thermodynamique a généré les données de tous les capteurs en fonction des variations de débit, des conditions de vol et de l'efficacité des modules concernés. Le taux exponentiel de variation du débit et de perte d'efficacité a été établi dans chaque ensemble de données, en partant d'un point de consigne de détérioration initiale choisi au hasard. L'ensemble de données se compose de deux parties : l'ensemble d'apprentissages et l'ensemble de tests. L'ensemble d'apprentissages est utilisé pour estimer les paramètres du modèle. Notre contribution porte sur l'adaptation d'algorithmes bien connus dédiés au modèle de Markov caché (HMM) pour leur application au PHM data challenge. Enfin, le modèle appris est appliqué à l'ensemble de tests pour prédire le RUL du système. Un exemple numérique de pronostic RUL prenant en compte à la fois le fonctionnement en ligne et hors ligne est présenté.

02.

Journal 02: Reliability Engineering & System Safety, 2020 (To be submitted)

Bootstrap-IOHMM to Manage Remaining Useful Life Considering Multiple Operating Conditions

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: System health, PHM, Input Output Hidden Markov Model, Degradation design, RUL, Condition monitoring.

Abstract: Online remaining useful life (RUL) assessment is a significant asset in prognostic and health management systems (PHM) in many industrial domains where safety, reliability and cost reduction are of high importance. To reduce the cost, one solution is to repair/replace the system before an unexpected failure which usually handled by setting a maintenance window according to the estimated RUL. However, it is not easy to predict the breakdown state of a system if it has multiple operating conditions, because system degradation varies with the dynamics of the operations. So, the current estimated RUL could be different than estimated RUL in the future due to the different operating conditions. That is why online RUL assessment is a very important and much more effective approach on condition-based maintenance which could be done using the new measurement that comes from the system each time. This paper presents an Input-Output Hidden Markov Model (IOHMM) that estimates the online RUL based on available measurements. The model then learns the impact of the operating condition on the RUL and offers to manage it by changing the corresponding operating conditions. A reference managing algorithm is presented to match the estimated RUL to a given target RUL. The well-known algorithms are used in the model training and diagnostic application which are adapted from HMM to IOHMM. A numerical application is given where a bootstrap method applied to show the importance of good prediction from a limited number of data sequences. Since degradation is a slow process, it's difficult to have enough data sequences having full information that covers till the system failure. Therefore, the bootstrap is an interesting method to train the IOHMM model by resampling with replacement.

Résumé : L'évaluation en ligne de la durée de vie résiduelle (RUL) est un enjeu majeur pour les systèmes de pronostic et de gestion de la santé (PHM) dans de nombreux domaines industriels où la sécurité, la fiabilité et la réduction des coûts sont de grande importance. Pour réduire les coûts, une solution consiste à réparer/remplacer le système avant une panne imprévue, ce qui est généralement fait en fixant une fenêtre de maintenance en fonction d'une estimation de la durée de vie restante. Cependant, il est difficile de prévoir l'état de santé d'un système si celui-ci est soumis à plusieurs conditions de fonctionnement, car la dégradation du système varie en fonction de la dynamique de commande. Ainsi, la valeur estimée de la RUL pourrait être différente en raison des différences de conditions de fonctionnement. C'est pourquoi l'évaluation en ligne de la RUL est une approche très efficace pour la maintenance basée sur l'état de santé estimée en utilisant chaque nouvelle mesure sur le système. Cet article présente un modèle de Markov caché (Input-Output Hidden Markov Model, IOHMM) qui estime en continu la RUL en ligne sur la base des mesures disponibles. Dans un premier temps, le modèle apprend l'impact de la condition de fonctionnement sur l'évolution de la RUL et propose de la gérer en changeant les conditions de fonctionnement correspondantes. Un algorithme de gestion de référence est présenté pour faire correspondre la RUL estimée à une RUL cible donnée. Des versions adaptées d'algorithmes bien connus sont utilisées dans la phase d'apprentissage et de diagnostic du modèle. Une application numérique est donnée où une méthode Bootstrap est appliquée pour montrer l'importance d'une bonne prédiction à partir d'une quantité limitée de séquences de données. En effet, comme la dégradation est un processus lent, il est difficile d'avoir suffisamment de séquences de données ayant des séquences d'information complètes c'est-à-dire jusqu'à la défaillance du système. Ainsi, le Bootstrap est une méthode intéressante pour entraîner le modèle IOHMM par rééchantillonnage avec remplacement.

03.

Journal 03: Engineering Applications of Artificial Intelligence, 2020 (To be submitted)

Estimating Remaining Useful Life of Flow Distribution Systems Under Missing Data Challenges

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: FDS, System health, IOHMM, Input Output Hidden Markov Model, Missing data, RUL.

Abstract: This paper presents an IOHMM based method that monitors health states of structured systems under missing data challenges. Since the structured systems have multiple subsystems/components in which any of them could have produced data sequences that contain missing elements. The missing data is important to handle in order to have a good estimation of the component's health states. Otherwise, it would lead us to have the wrong idea/prediction of the system's health conditions. The proposed model diagnosed each of the components by overcoming this challenge individually. After that, the health states over the system are estimated based on the estimated health condition of all components. Then, an algorithm is presented which constructs the individual models (for the components) to a single model that represents the entire system. Finally, the constructed model is used to prognostic the remaining useful life of the system based on the estimated health states of the system. The model considers operating conditions in the diagnostic and prognostic algorithms. A numerical application is presented which is simulated following subpart of a water distribution network in Barcelona City.

Résumé : Ce document présente une méthode basée sur l'IOHMM qui permet de surveiller l'état de santé de systèmes structurés en présence de données manquantes. En effet, les systèmes structurés comportent de multiples sous-systèmes/composantes dans lesquels n'importe lequel d'entre eux peut produire des séquences de données contenant des éléments manquants. Il est important de traiter les données manquantes afin d'avoir une bonne estimation des états de santé des composants. L'absence de données dans certaines séquences conduit généralement à une moins bonne estimation des états de santé du système monitoré. Le modèle proposé permet de diagnostiquer l'état de chacun des composants pour l'étendre à celui du système. Le passage des modèles individuels d'évolution de l'état de santé à celui du système est automatisé, ce qui nous permet l'estimation en ligne de la RUL du système structuré. Le modèle tient compte des conditions de fonctionnement dans les algorithmes de diagnostic et de pronostic. Une application numérique basée sur une portion du réseau de distribution d'eau de la ville de Barcelone est présentée.

04.

Conference paper 01: ESREL, 2019

Estimating IOHMM Parameters to Compute Remaining Useful Life of System

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: Health assessment**,** Input Output Hidden Markov Model, PHM, RUL, Degradation, Operating conditions.

Abstract: This paper is about Input-Output Hidden Markov Model (IOHMM) to compute the remaining useful life (RUL) of a system with different operating conditions. Well-known algorithms dedicated to Hidden Markov Model (HMM) are extended to IOHMM. The processing to compute the RUL considering further operating conditions are proposed through an application example. In this paper, a single input, and multiple outputs IOHMM is considered, but can be generalized to multiple inputs easily.

Résumé : Cet article porte sur le modèle de Markov caché à entrées-sorties (Input-Output Hidden Markov Model - IOHMM) pour calculer la durée de vie résiduelle (RUL) d'un système avec différentes conditions de fonctionnement. Des algorithmes bien connus dédiés au modèle de Markov caché (HMM) sont étendus au modèle IOHMM. Le traitement permettant de calculer la RUL en tenant compte des conditions d'exploitation ultérieures est proposé à l'aide d'un exemple d'application. Dans ce document, une seule entrée et plusieurs sorties IOHMM sont considérées, mais cela peut être facilement généralisé à plusieurs entrées indépendantes.

05.

Conference paper 02: CIGI QUALITA, 2019

Input-output hidden Markov model for diagnosis of complex systems

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: Complex system, degradation, health assessment, operating conditions, IOHMM, parameter learning.

Abstract: Prognosis system state of degradation and estimating its remaining useful life requires the system health assessment. For a correct prognostic, a good diagnostic as health assessment is required. Complex systems are difficult to manage for modeling reasons considering complexity, environmental and operational conditions. This paper deals with a stochastic model for generic modeling purposes and considers operating conditions in order to determine the system health. The proposed model is an Input-Output Hidden Markov Model that is able to model a degradation process of complex systems given operational conditions and allows assessing the system health. Well-known algorithms dedicated to HMM are adapted to IOHMM for multiple observation sequences and inputs.

Résumé : Le pronostic de l'état de dégradation d’un système et l'estimation de sa durée de vie résiduelle nécessitent une évaluation de son état de santé. Pour un pronostic correct, un bon diagnostic de l’état de santé du système est nécessaire. Les systèmes complexes sont difficiles à gérer pour des raisons de modélisation tenant compte de la complexité, des conditions environnementales et opérationnelles. Ce document traite d'un modèle stochastique à des fins de modélisation générique et prend en compte les conditions d'exploitation afin de déterminer l'état de santé du système. Le modèle proposé est un modèle de Markov caché à entrée-sortie capable de modéliser un processus de dégradation de systèmes complexes dans des conditions opérationnelles données et permet d'évaluer l'état de santé du système. Des algorithmes bien connus dédiés au HMM sont adaptés à l'IOHMM pour des séquences d'observation avec des entrées multiples.

06.

Conference paper 03: IFAC, 2020

Bootstrap Confidence Interval on IOHMM Parameters for System Health Diagnostic Under Multiple Operating Conditions

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: System health, PHM, Input Output Hidden Markov Model, Condition based diagnostics, Degradation design, Condition monitoring.

Abstract: The operating conditions have an important impact on system degradation. This paper uses the Input-Output Hidden Markov Model to represent the system degradation having multiple operating conditions. In this paper the bootstrap method is applied to estimate the model parameters and applied to diagnostic system health. Parameters of the model are computed with 95% confidence intervals. The uncertainty about multiple data sequences and degradation speed is handled in the proposed model. A numerical application is given to explain the methodologies used to estimate the model parameters and the system health diagnostic.

Résumé : Les conditions de fonctionnement ont un impact important sur la dégradation d'un système. Cet article utilise le modèle de Markov caché à entrée-sortie pour représenter la dégradation d'un système ayant des conditions de fonctionnement multiples. Dans cet article, la méthode Bootstrap est appliquée pour estimer les paramètres du modèle et appliquée pour diagnostiquer la santé du système. Les paramètres du modèle sont calculés avec des intervalles de confiance de 95%. L'incertitude concernant les séquences de données multiples et la vitesse de dégradation est traitée dans le modèle proposé. Une application numérique est donnée pour expliquer les méthodologies utilisées pour estimer les paramètres du modèle et diagnostiquer l'état du système.

07.

Conference paper 04: MED, 2020

Input-Output Hidden Markov Model for System Health Diagnosis Under Missing Data

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3], Didier Theilliol[4]

Keywords: degradation, parameter learning, sensor saturation, health assessment, incomplete information.

Abstract: Sensor data can be used to diagnose the system's health. A challenge comes when the data contain missing or invalid data. It is common that sensors misread for various reasons. So, data contain missing measurements and sensor saturation. The main contribution in this paper is to implement a method based on the Input-Output Hidden Markov Model that trains the model using the missing measurements and sensor saturation, then diagnoses the system health at given operating conditions. Usually, if a data set contains some sequences with missing elements then they can be excluded from the analysis. It cleans the data set but reduces its size. This strategy knows as list-wise or case-wise deletion is less suitable for real application cases. The proposed method includes the sequences with missing data into the analysis by generating the missing elements to complete the sequence. The maximum likelihood is applied to estimate IOHMM parameters that offer substantial improvements over list-wise deletion. A numerical application with simulated data sets illustrates the method.

Résumé : Les données des capteurs peuvent être utilisées pour diagnostiquer l'état de santé d'un système. Un défi se pose lorsque les données contiennent des données manquantes ou censurées. En effet, il est fréquent que des capteurs fassent une mauvaise lecture pour diverses raisons. Ainsi, les données contiennent des mesures manquantes ou une saturation des capteurs à censurer. La principale contribution de cet article est de mettre en œuvre une méthode basée sur le modèle de Markov caché à entrée-sortie qui estime les paramètres du modèle en utilisant les mesures manquantes et la saturation des capteurs, puis diagnostique la santé du système dans des conditions de fonctionnement données. La censure des données a le désavantage de réduire le volume de données utiles, mais agit sur la qualité d'estimation. Cette stratégie, connue sous le nom de suppression par liste ou par cas, est moins adaptée aux cas d'application réels. La méthode proposée consiste à inclure les séquences comportant des données manquantes dans l'analyse en générant les éléments manquants pour compléter la séquence. La probabilité maximale est appliquée pour estimer les paramètres IOHMM qui offrent des améliorations substantielles par rapport à la suppression par liste. Une application numérique avec des ensembles de données simulées illustre la méthode.

08.

Conference paper 05: ESREL, 2020

Input-Output Hidden Markov Model to Manage the Remaining Useful Life of System Under Missing Data

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: Degradation, Diagnostic, Prognostic, IOHMM, Operating Condition, Managing RUL.

Abstract: This paper proposes a statistical model for diagnostic and prognostic system health by using the sensor data. Sometimes sensor misreads the observation for various reasons which contains one or more holes in the measured data and sensor saturation. The main contribution in this paper is to estimate and manage the remaining useful life (RUL) of the system considering multiple operating conditions under missing data. A recursive technique based on Input-Output Hidden Markov Model is proposed in this article for identifying the missing measurements or sensor saturation (ROSS) then predict the system failure at given operating conditions. An optimization performs based on the production speed that controls the operating conditions to manage the RUL. An example is given where the model parameters are estimated from the data set that has about 13% of missing data. The well-known algorithms are adapted in the model training and application.

Résumé : Cet article propose un modèle statistique pour le diagnostic et le pronostic de l'état de santé d'un système en utilisant les données des plusieurs capteurs. Parfois, le capteur interprète mal l'observation pour diverses raisons, ce qui entraîne l'absence de données. La principale contribution de cet article est d'estimer et de gérer la durée de vie résiduelle (RUL) d'un système en tenant compte des multiples conditions de fonctionnement malgré des données manquantes. Une technique récursive basée sur le modèle de Markov caché à entrées-sorties est proposée dans cet article pour identifier les mesures manquantes ou la saturation du capteur (ROSS) puis prédire la défaillance du système dans des conditions de fonctionnement données. Une optimisation s'effectue sur la base d'entrées qui contrôle les conditions de fonctionnement afin de gérer la RUL. Un exemple est donné où les paramètres du modèle sont estimés à partir de l'ensemble de données qui comporte environ 13% de données manquantes. Les algorithmes connus sont adaptés dans la phase d'apprentissage et de test du modèle.

09.

Conference paper 06: ESREL, 2020

Estimating the Remaining Useful Life of a Flow Distribution System

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: IOHMM, degradation, flow distribution system, remaining useful life, health assessment.

Abstract: Flow distribution systems (FDS) are widely used in many industrial processes when it is necessary to distribute products in several ways and then to collect them into one or several discharge destinations, such as water supply, heat supply, electricity supply network, etc. The maintenance decisions on a flow distribution system are challenging because the degradation of individual components is independent and not fully detectable. It can lead to inaccurate diagnostic and prognostic results. Industrial systems are often equipped with multiple sensors on each component to collect the efficient information that helps computing the remaining life of the system (RUL). As multiple data are captured, it is a multiple output system. This paper proposes an Input-Output Hidden Markov Model (IOHMM) for RUL assessment to maintenance aid decision-making of a multi-component flow distribution system. The components are considered independently for the diagnostic and prognostic of their health conditions and thus of the system. The goal of this paper is to find the best path to know the health conditions of the components for supplying the demand to the destination. The study offers an optimal solution in two steps to keep the system alive longer while it can fulfil customer needs. The first step is the independent monitoring of all components to determine the most appropriate supply planning strategy. The second step is to identify all the possible routes where the flow gets through different components for discharging to the destinations. This aims to benefit such as the alternative flow paths or the target maintenance activities at the system level. The operating conditions are considered as inputs for each of the components independently.

Résumé : Les systèmes de distribution de flux (FDS) sont largement utilisés dans de nombreux processus industriels lorsqu'il est nécessaire de distribuer des produits selon plusieurs chemins puis de les collecter vers une ou plusieurs destinations de rejet, comme l'approvisionnement en eau, l'approvisionnement en chaleur, le réseau d'alimentation électrique, etc. Les décisions de maintenance d'un système de distribution de flux sont difficiles à prendre, car la dégradation des différents composants est indépendante et n'est pas entièrement détectable. Elle peut conduire à des résultats de diagnostic et de pronostic inexacts. Les systèmes industriels sont souvent équipés de plusieurs capteurs sur chaque composant afin de collecter les informations efficaces qui permettent de calculer la durée de vie résiduelle d'un système (RUL). Comme de multiples données sont saisies, il s'agit d'un système à sorties multiples. Ce document propose un modèle de Markov caché à entrées-sorties (IOHMM) pour l'évaluation de la RUL afin d'aider à la prise de décision en matière de maintenance d'un système de distribution de flux à composants multiples. Les composants sont considérés indépendamment pour le diagnostic et le pronostic de leur état de santé et donc du système. L'objectif de ce document est de trouver le meilleur moyen de connaître l'état de santé des composants pour fournir la demande à la destination. L'étude propose une solution optimale en deux étapes pour maintenir le système en vie aussi longtemps que nécessaire pour répondre à la demande client. La première étape est le contrôle indépendant de tous les composants afin de déterminer la stratégie de planification de l'approvisionnement la plus appropriée. La deuxième étape consiste à identifier tous les itinéraires possibles où le flux passe par différents composants pour être déchargé vers la destination. Cela vise à obtenir des avantages tels que les voies de flux alternatives ou le ciblage des activités de maintenance au niveau du système. Les conditions d'exploitation sont considérées comme des entrées indépendantes pour chacun des composants.

10.

Conference paper 07: ESREL, 2020

Online Remaining Useful Life Management Considering Operating Conditions to Match the Given Maintenance Date

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: RUL, Input Output Hidden Markov Model, IOHMM, Online assessment, PHM, Condition monitoring.

Abstract: Online remaining useful life (RUL) is a major challenge of prognostic and health management systems (PHM) in many industrial domains where safety, reliability and cost reduction are of high importance. To reduce the cost, one solution is to match the maintenance date with the estimated remaining life of the system. This prediction of the RUL allows fixing time in the future to organize a maintenance action which can be called maintenance time-window. Nevertheless, the RUL can change due to the different dynamics of the operating conditions over the operation time. It may shift the maintenance window into another time. System health should be updated when new measurements come in analysis and the prognostic shows an updated RUL. Thus, the online RUL prediction is a much more effective approach on condition-based maintenance. This paper presents an Input-Output Hidden Markov Model (IOHMM) that estimates the online prognostic based on passed to current measured data from the system which is used to manage the RUL that corresponds to a target remaining time. A reference manager is designed to figure out the next input condition according to the new measurements in order to reschedule the maintenance time window. An example is shown in which well-known algorithms dedicated to HMM are adapted to IOHMM for online prognostic when the system emits a new observation.

Résumé : La durée de vie résiduelle en ligne (RUL) est un défi majeur des systèmes de pronostic et de gestion de la santé (PHM) dans de nombreux domaines industriels où la sécurité, la fiabilité et la réduction des coûts sont d'une grande importance. Pour réduire les coûts, une solution consiste à faire correspondre la durée de vie résiduelle estimée avec la date de maintenance du système. Cette prédiction de la RUL permet de fixer un délai dans le futur pour organiser une action de maintenance que l'on peut appeler fenêtre temporelle de maintenance. Néanmoins, la RUL peut changer en raison de la dynamique différente liée aux conditions de fonctionnement au cours de la période d'exploitation. En contrôlant ces conditions, il est possible de faire correspondre les temps pour opérer une maintenance. L'état de santé du système doit être mis à jour à chaque nouvelle mesure ainsi que le pronostic de la RUL. Ainsi, le pronostic en ligne permet d'élaborer une approche plus efficace dans la maintenance basée sur l'état de santé du système. Cet article présente un modèle de Markov caché (Input-Output Hidden Markov Model, IOHMM) qui estime le pronostic en ligne sur la base des données mesurées du système qui sont utilisées pour gérer la RUL et la faire correspond à un temps restant cible. Un gestionnaire de référence est conçu pour déterminer la prochaine condition d'entrée en fonction des nouvelles mesures afin de reprogrammer la fenêtre de temps de maintenance. Un exemple est montré dans lequel des algorithmes bien connus dédiés au HMM sont adaptés au IOHMM pour le pronostic en ligne lorsque le système émet une nouvelle observation**.**

11.

Conference paper_08: Congrès Lambda Mu, 2020

Modèle IOHMM pour le diagnostic et le pronostic de Systèmes

Kamrul Islam Shahin[1], Christophe Simon[2], Philippe Weber[3]

Keywords: dégradation, état de santé, condition opérationnelle.

Abstract: In this paper, we propose to exploit an advanced form of Hidden Markov Model (HMM) to represent the process of evolution of the health status of a system. This type of model is of several interests. It corresponds well to the principle of evolution of a complex element (health status) which is not directly observable, but through observation variables. Its stochastic nature allows uncertainty to be taken into account: observation noise (data uncertainty), evolution processes that are more complex than imagined (model uncertainty). In this context, dynamic Bayesian networks and in particular the 2 Time slices Bayesian Networks are particularly well suited to represent an HMM conditioned by states of operational conditions and thus form an advanced class of HMM, the Input-Output HMM (IOHMM).

Résumé : Dans cet article, nous proposons d'exploiter une forme évoluée de modèle de Markov caché (HMM) pour représenter le processus d'évolution de l'état de santé d'un système. Ce type de modèle présente plusieurs intérêts. Il correspond bien au principe d'évolution d'un élément (état de santé) complexe et non observable directement, mais au travers de variables d'observation. Sa nature stochastique permet la prise en compte de l'incertitude : bruit d'observation (incertitude de données), processus d'évolution plus complexe qu'imaginé (incertitude de modèle). Dans ce contexte, les réseaux bayésiens dynamiques et notamment les 2 Time slices Bayésien Networks sont particulièrement bien appropriés pour représenter une HMM conditionnée par des états de conditions opérationnelles et ainsi formé une classe évoluée de HMM, les Input-Output HMM (IOHMM).

Table of Contents

List of Figure

List of Table

Notations and acronyms

$S = \{s_1, s_2, s_3, \cdots, s_N\}$ are hidden states.
N is the number of states in the model.
$X = \{X^{(1)}, X^{(2)}, \dots, X^{(L)}\}$ are state sequences.
L is the number of sequences.
$A = (a_{ij})_{ij}$ is state transition matrix.
$\hat{A}$ is the estimated transition matrix.
$V = \{v_1, v_2, v_3, \cdots, v_M\}$ are observation symbols.
C_{v_m} is the count of v_m observation.
M is the number of observation symbols for an output.
$B = (b_{jl})_{jl}$ is the state emission matrix.
π is the initial state distribution.
Λ is the HMM.
$\hat{\Lambda}$ is the IOHMM
$Y = \{Y^{(1)}, .., Y^{(L)}\}$ are the set of data observation sequence Y
$\bar{Y} = \{\bar{Y}^{(1)}, ..., \bar{Y}^{(L)}\}$ are the set of observation sequence with missing elements
$\mathcal{Y} = \{Y^1, Y^2, ..., Y^Q\}$ is the set of outputs.
$q, (1 \le q \le Q)$ is the number of outputs.
K_l is the length of observation sequence $Y^{(l)}$.
$U = \{U^{(1)}, U^{(2)}, ..., U^{(L)}\}$ are input sequences.
$\bar{U} = \{\bar{U}^{(1)}, \bar{U}^{(2)}, ..., \bar{U}^{(L)}\}$ are input sequences with missing elements.
u = input id.
$d, (1 \le d \le D)$ block of missing elements in data sequences
p is the number of operating conditions.
$R\hat{A}^p$ matrix ration in the input sequence
C_p is the count of p^{th} matrix.
CI is the confidence interval.
$\mathbb{N}$ is the set of strict positive integers.
$\alpha(X_k)$ is the forward auxiliary variable.
$\beta(X_k)$ is the backward auxiliary variable.
$\omega_k(j)$, $\varepsilon_k(i,j)$ are the Baum Welch auxiliary variables.
$\gamma(X_k)$ is the forward Viterbi auxiliary variable.
$\delta(X_k)$ is the backward Viterbi auxiliary variable.
C is a threshold value to stop prognosis computation
Sc_i is the penalty score of model performance

Modèle graphique probabiliste appliqué au diagnostic de l'état de santé des systèmes, au pronostic et à l'estimation de la durée de vie résiduelle

Mots-clés : Évaluation de la santé, Diagnostic, Pronostic, RUL, Gestion des RUL, PHM, Conception de la dégradation, Condition de fonctionnement, Système complexe, Système structuré, Évaluation en ligne, Modèle de Markov caché entrée-sortie, Apprentissage des paramètres, Données manquantes

Résumé de thèse

Cette thèse contribue au développement des recherches dans le domaine du Pronostic et Health Management : gestion de l'état de santé des systèmes complexes. Dans un contexte de management opérationnel et de sûreté de fonctionnement des systèmes, nous proposons d'étudier comment la modélisation par un Modèle Graphique Probabiliste Dynamique (MGPD) permet le diagnostic de l'état de santé courant d'un système, le pronostic de cet état et de l'évolution des dégradations, ainsi que l'estimation de sa durée de vie résiduelle en fonction de ses conditions opérationnelles.

La dégradation des composants est en général inconnue et nécessite un arrêt du système pour être observée. Cependant, cela est difficile, voire impossible, durant l'exploitation du système. Néanmoins, un ensemble de grandeurs observables sur le système ou le composant peut caractériser le niveau de dégradation et faciliter l'estimation de la durée de vie résiduelle du composant et du système.

Les MGPD offrent une approche adaptée à la modélisation de l'évolution de l'état de santé des systèmes et des composants. Récemment, l'utilisation de HMM (Hidden Markov Model) ou de HSMM (Hidden Semi-Markov Model) pour modéliser un processus non observable de dégradation et le relier à des observations de leurs conséquences a déjà été exploitée avec des résultats intéressants. Toutefois, la non-prise en compte des conditions opérationnelles, influant sur les processus de dégradation, limite la performance de ces outils. Les algorithmes d'apprentissage et d'inférence rendent exploitables ces modèles complexes pour une exploitation dans une problématique de pronostic.

Il s'agit dans cette thèse de transposer et de capitaliser l'expérience de ces travaux antérieurs dans un contexte de pronostic sur la base d'un MGPD plus efficace compte tenu des connaissances disponibles sur le système. Nous étendons la modélisation classique des modèles de la famille des HMM vers les IOHMM pour permettre une propagation temporelle de l'incertitude afin de résoudre le problème de pronostic de l'état de santé et de l'estimation de la durée de vie résiduelle. Cette recherche comprend l'extension des algorithmes d'apprentissage et d'inférence appliqués aussi bien dans le cas d'un composant que pour un système structuré. Les variantes du modèle HMM sont proposées pour intégrer le contexte opérationnel dans le pronostic.

Cette thèse a pour but de contribuer à lever les verrous scientifiques suivants :

- Considérer l'état de santé quelle que soit la complexité du système par un modèle stochastique et apprendre les paramètres du modèle à partir des mesures disponibles sur le système.
- Établir un diagnostic de l'état de santé du système et le pronostic de son évolution en intégrant plusieurs conditions opérationnelles.
- Estimer la durée de vie résiduelle des composants et des systèmes structurés (série, parallèle) à partir de ses composants.

L'enjeu est majeur, car le pronostic de la dégradation des composants du système permet de définir des stratégies soit de pilotage soit de maintenance par rapport à la durée de vie résiduelle du système. Cela permet la réduction de la probabilité d'occurrence d'un arrêt pour cause de dysfonctionnement du système, soit en ajustant la vitesse de dégradation pour s'accorder à un plan de maintenance préventif, soit en planifiant les interventions de maintenance de manière proactive.

Dynamic Probabilistic Graphical Model applied to the system health diagnosis, prognosis, and the remains useful life estimation

Keywords: Health assessment, Diagnostic, Prognostic, RUL, RUL Management, PHM, Degradation design, Operating Condition, Complex system, Structured System, Online assessment, Input Output Hidden Markov Model, Parameter learning, Missing data

Thesis abstract

This thesis contributes to prognosis and health management for assessing health condition of complex systems. In the context of operational management and operational safety of systems, we propose to investigate how Dynamic Probabilistic Graphical Modelling (DPGM) can be used to diagnose the current health state of systems, prognostic the future health state, and the evolution of degradation, as well as estimate its remaining useful life based on its operating conditions.

System degradation is generally unknown and requires shutting down the system to be observed. However, this is difficult or even impossible during system operation. Though, a set of observable quantities on a system or component can characterize the level of degradation and help to estimate the remaining useful life of components and systems.

The DPGM provides an approach suitable for modelling the evolution of the health state of systems and components. Recently, interesting results have been obtained by using HMM (Hidden Markov Model) or HSMM (Hidden Semi-Markov Model) to model unobservable degradation processes and to relate them to observations of their consequences. However, the performance of these models is limited because they are not able to consider the operational conditions that affect degradation processes. Learning and inference algorithms allow these complex models to be used for prediction problems.

The aim of this thesis is to transpose and capitalize on the experience of these previous works in a prognostic context on the basis of a more efficient DPGM taking into account the available knowledge on the system. We extend the classical HMM family models to the IOHMM to allow the time propagation of uncertainty to address prognostic problems. This research includes the extension of learning and inference algorithms. Variants of the HMM model are proposed to incorporate the operating environment into the prognosis.

The aim of this thesis is to contribute to solving the following scientific locks:

- Considering the state of health whatever the complexity of the system by a stochastic model and learning the model parameters from the available measurements on the system.

- Establish a diagnosis of the state of health of the system and the prognosis of its evolution by integrating several operational conditions.

- Estimate the remaining useful life of components and structured systems with series and parallel components.

This is a major challenge because the prognosis of the degradation of system components makes it possible to define strategies for either control or maintenance in relation to the residual life of the system. This allows the reduction of the probability of occurrence of a shutdown due to a system malfunction either by adjusting the degradation speed to fit in with a preventive maintenance plan or by proactively planning maintenance interventions.

Résumé étendu en français

Dans le contexte de l'évolution industrielle actuelle, l'un des plus grands défis est de maintenir les systèmes avec un niveau important de sécurité, de fiabilité et de disponibilité. Cette évolution se place dans le cadre de ce qui est désigné aujourd’hui par plusieurs termes : Industrie du futur (IdF), Industrie 4.0, la quatrième évolution de l'industrie, Partenariat de fabrication avancée, Made in China 2025, Usine du Futur … Ce concept d'évolution industrielle est basé sur l'innovation et la technologie numérique avec l'excellence opérationnelle révélant le potentiel de transformation et d'amélioration de la performance industrielle (Iung, 2018). Pour des raisons de clarté, nous utiliserons le vocable Usine du Futur dans le reste du document.

De nos lectures, nous avons isolé quatre grands principes qui aident l'industrie à identifier et à mettre en œuvre différents scénarios pour cette évolution – « révolution » (Hermann et Otto, 2016) :

- Le premier principe est l'interconnexion entre les différents composants d'un système complexe via l'Internet des Objets (Bonner, 2018). Un système complexe est défini comme un système ou une machine qui comporte plusieurs composants ou sous-systèmes qui fonctionnent dans plusieurs conditions de fonctionnement.

- Le deuxième principe est la transparence de l'information, qui fait référence à la capacité de fournir aux utilisateurs du système une grande quantité d'informations qui l’aide à prendre des décisions liées au fonctionnement du système (Bonner, 2018).

- Le troisième principe est celui de l'assistant technologique qui collecte et observe les informations sur le système pour aider les utilisateurs/opérateurs à prendre des décisions et à résoudre des problèmes urgents de fonctionnement (Gronau, 2016). Pour cela, il est important que les utilisateurs du système soient constamment conscients de la détérioration de l’état de santé du système et de l’impact de cette dégradation sur différents indicateurs comme les taux de production, les activités de maintenance, les pannes du système ou les arrêts imprévus. En effet, la dégradation de l’état de santé peut causer des dommages aux opérateurs et augmenter les coûts de maintenance.

- Le quatrième principe est la prise de décision décentralisée, c'est-à-dire la capacité d’un système à prendre ses propres décisions et à les exécuter seul (Gronau, 2016). Le système fonctionne en autonomie en tenant compte des différentes anomalies, telles que les interférences, les objectifs contradictoires ou une planification automatique qui fixe le temps pour différentes actions à entreprendre pendant le fonctionnement du système.

Comme on le voit dans ces 4 principes fondamentaux de l’Usine du Futur, l’essentiel repose sur la disponibilité d’informations instantanées, notamment les conditions opérationnelles des sous-systèmes interconnectés, les états de fonctionnement, les technologies de l’information utilisées dans le système …

Dans ce contexte et de notre point de vue, il est important de maintenir les systèmes industriels sous surveillance constante pour aider les utilisateurs à prendre des décisions éclairées 24 heures sur 24, 7 jours sur 7. L'utilisateur a besoin de toutes les informations pertinentes sur le fonctionnement du système, sur son état de santé, et sur sa fiabilité. Il y a là d’ailleurs un paradigme à résoudre entre la masse d’information et la pertinence de ces informations.

Si nous focalisons notre propos sur notre domaine de recherche et vis-à-vis des 4 grands principes que nous avons relevés, l’état de santé d’un système est donc une information essentielle pour la prise de décision. Toutefois, c’est un concept assez difficile à appréhender et impossible à mesurer. En revanche, il peut être déterminé, estimé, en observant le comportement du système (symptômes) et en appliquant

des techniques de diagnostic sur la base des observations issues des capteurs et autres éléments technologiques, implémentés sur le système, susceptibles de fournir des informations d'intérêt.

Dans le cadre de l'Usine du Futur, c'est plus que l'état de santé du système qui nous intéresse, mais son évolution et la projection de cette évolution dans le futur. Sur la base de cette information estimée, il devient envisageable de pronostiquer la durée de vie utile restante ou résiduelle que l'on connait sous le vocable de RUL. Ces informations de durée peuvent aider à planifier proactivement la maintenance préventive pour réduire les coûts de maintenance, résoudre les problèmes de conformité, définir les activités attribuées automatiquement, augmenter la vitesse de production et à terme les bénéfices.

La 4e révolution 'Usine du Futur' propose donc de moderniser l'outil de production et d'augmenter la diffusion des technologies numériques au sein de l'entreprise, pour accompagner une évolution de la gamme et une personnalisation toujours plus grande des produits. Partout dans le monde, les industriels mettent en œuvre le concept de l'Usine du Futur dans leurs entreprises et la tendance se développe progressivement. L'Usine du Futur devient la principale préoccupation et le principal concept de profit manufacturier et de politique commerciale pour la prochaine génération d'industries.

En conséquence, les systèmes se complexifient progressivement avec les interconnexions entre leurs sous-systèmes intégrants des conditions de fonctionnement compliquées voir complexes. Différents sous-systèmes et composants du système deviennent dépendants les uns des autres pour la fabrication des produits. C'est le cas des systèmes de fabrication, où la disponibilité de l'infrastructure, l'évolution des objectifs et la maintenance du système deviennent chaque jour un défi pour répondre aux besoins de nos sociétés modernes.

Par conséquent, l'étude de la disponibilité et de la maintenance des systèmes est désormais un axe de recherche de plus en plus fort. De plus, au niveau mondial, un grand nombre de systèmes encore en service depuis de nombreuses années approchent désormais de leur fin de vie et nécessitent donc un entretien régulier voir de plus en plus fréquent à défaut d'une rénovation complète. Pour répondre à ces exigences, les stratégies de maintenance doivent être améliorées et notamment les techniques de maintenance prévisionnelle/conditionnelle.

Étant donné que des actions de maintenance sont prises avant qu'une défaillance du système ne survienne, les décisions de maintenance ont un impact sur la sécurité du système et le travail des opérateurs. Cela peut conduire à des coûts importants. Par conséquent, afin de définir les plans de maintenance et optimiser la disponibilité du système, nous devons mettre en place un système de gestion et de pronostic de l'état de santé des systèmes industriels (PHM). Ce type d'étude vise à réduire les coûts de maintenance et d'assurer une haute fiabilité/disponibilité du système. D'un point de vue économique, l'étude des méthodes de PHM est une question extrêmement importante pour les industriels afin de rester compétitif sur le marché.

Dans les applications PHM, le pronostic a besoin du diagnostic puisqu'il faut définir l'état courant afin de projeter son évolution dans le futur. Le diagnostic doit donc étudier la dégradation du système. Cette dégradation du système est un processus dynamique qui décrit les dommages sur le service fourni par le système pendant sa durée de vie. Ce processus dynamique est complexe, d'une part parce qu'il est caché, est non mesurable (il n'y a pas de capteur d'état de santé) et d'autre part parce que sa dynamique dépend de facteurs internes et externes qui ont un impact sur l'état de santé du système. La complexité vient également du fait que plusieurs modes de dégradation opèrent de manière concourante, mais que nous ne pouvons en observer que les conséquences. Les causes majeures de la dégradation de l'état de santé sont l'usure de pièces ou de composants du système, les accidents, le manque d'entretien … Les impacts de ces raisons diffèrent en fonction des différentes conditions telles que les conditions de fonctionnement ou condition opérationnelles, les conditions environnementales, etc.

Dans ce travail de thèse, nous étudions la dégradation d'un système complexe compte tenu des conditions de fonctionnement. Si l'on ne considère aucune exception (accident, erreurs système / pièce, etc.), et aucune variation des conditions de fonctionnement alors le processus de dégradation peut être raisonnablement compliqué à comprendre et à modéliser en faisant l'hypothèse qu'il n'y a qu'une seule dynamique. Sinon, la dégradation aura plusieurs dynamiques ou plusieurs comportements lorsque différentes variétés de conditions de fonctionnement sont appliquées au système. La complexité de modélisation du processus de dégradation est alors plus importante.

Dans cette thèse, nous considérons un processus complexe de dégradation. Aussi, pour proposer une solution plus globale, il faut tenir compte de l'existence des différentes conditions de fonctionnement ayant un impact sur la dégradation des systèmes (Shahin, 2019a). Ces conditions de fonctionnement que nous regroupons sous le vocable de conditions opérationnelles sont, soit de conditions environnementales (non contrôlé), soit des conditions opératives (contrôlé). L'estimation du niveau de dégradation courant, le résultat de la phase de diagnostic doit donc tenir compte non seulement de l'état passé, mais aussi des conditions opérationnelles passées et courantes. Cette phase de diagnostic n'en est alors que plus difficile. De fait, la phase de pronostic doit certes tenir compte de l'état de santé estimé courant, mais aussi des conditions opérationnelles futures. La RUL estimée à chaque instant est donc dépendante des conditions opérationnelles, ce qui constant un challenge difficile (Shahin, 2019b).

Dans cet objectif, il existe historiquement deux grandes approches pratiquées pour la gestion de l'état de santé des systèmes (PHM) : l'approche basée sur les modèles et l'approche basée sur les données.

- L'approche basée sur les modèles, également connue sous le nom d'approche physique, utilise un modèle mathématique dynamique du système qui exploite directement les processus physiques qui affectent la santé du composant ou du système (Skormin, 1994).
- Alors qu'une approche basée sur les données permet la construction du modèle de dégradation en utilisant des données d'observation collectées à partir de capteurs installés sur le système. Cette approche est généralement préférée lorsque les modèles de système ne sont pas disponibles ou pas assez robustes (Namburu, 2007).

Les deux approches ont leurs avantages et leurs inconvénients (effort de modélisation, précision, connaissances, etc.). De nombreux chercheurs les ont utilisés ensemble pour surmonter leurs inconvénients et tirer parti de leurs avantages (Liao et Köttig, 2014). Ils ont défini cette combinaison comme l'approche hybride. Cependant, nous avons décidé d'utiliser l'approche basée sur les données pour sa diversité, sa flexibilité et ses avantages qui facilitent la réalisation de l'objectif de notre recherche. Alors que l'approche basée sur un modèle nécessite les informations physiques du système, cette thèse propose des méthodes de diagnostic et de pronostic à partir de l'approche basée sur les données lorsque les informations physiques ne sont pas suffisantes ou ne sont pas disponibles. L'approche basée sur les données convient parfaitement à un système complexe qui comporte plusieurs composants avec un comportement non linéaire. Cette approche nous permet de traiter la grande dimension des données pour prédire la dégradation de plusieurs composants.

Différents types de modèles peuvent être adaptés dans le cadre des approches basées sur les données. Les modèles les plus populaires et les plus fréquemment utilisés sont soit déterministes, soit stochastiques. Le modèle déterministe fournit les sorties qui sont entièrement déterminées par les paramètres et les conditions initiales sans tenir compte du caractère aléatoire. En revanche, un modèle stochastique est un outil d'estimation dont l'analyse se concentre sur une séquence aléatoire d'observations. Il traite les propriétés stochastiques des variables aléatoires et gère les conditions de fonctionnement. Cette approche gère également les incertitudes des données en raison de ses propriétés aléatoires et d'approximation, qu'il est difficile de gérer pour le modèle déterministe. Par conséquent, le modèle stochastique nous semble intéressant pour atteindre l'objectif de cette thèse.

Il existe plusieurs modèles stochastiques que l'on peut trouver dans la littérature et qui peuvent être construits rapidement à faible coût. Les modèles les plus utilisés sont les modèles de réseau bayésien, les modèles de logique floue, les modèles de réseau neuronal et les modèles de Markov. Le réseau bayésien est un modèle graphique probabiliste qui utilise l'inférence bayésienne pour le calcul des probabilités. Le réseau bayésien offre une base mathématique solide et est présenté graphiquement de sorte que chaque variable peut être directement connectée entre elles. C'est l'une des méthodes de modélisation les plus populaires dans le domaine de recherche actuel. Cependant, l'une des principales limites des réseaux bayésiens est le traitement des variables dynamique. Les réseaux bayésiens dynamiques ne peuvent traiter les variables continues que de manière limitée (Friedman et Goldszmidt, 1996; Jensen, 2001 ; Weber et Simon 2016). Un autre modèle populaire est la logique floue qui fonctionne selon le raisonnement humain. Il est populaire pour sa flexibilité dans l'utilisation de mathématiques simples pour des systèmes non linéaires, intégrés et complexes. Le développement de règles floues et de fonctions d'appartenance est fastidieux, et la sortie floue peut être interprétée de diverses manières, ce qui rend l'analyse difficile. De plus, le développement d'un système flou nécessite une grande quantité de données et d'expertise. Dans ce cas, le réseau de neurones offre une excellente solution alternative. La base mathématique du réseau de neurones nous permet de gérer une grande quantité de données et d'entrées non linéaires. Ce modèle est flexible et bien adapté à une utilisation pour les problèmes de régression et de classification. Pourtant, comme ce modèle repose sur une grande quantité de données d'entraînement, il peut conduire à des problèmes d’apprentissage en cas de données insuffisantes (Yi, 2018). De plus, c’est un modèle de type boîte noire, il ne donne aucune information sur la mesure dans laquelle une variable affecte les autres variables ni sur la façon dont la couche cachée représente l'évolution de la probabilité ou de la dégradation. Ainsi, pour les cas où le concept de boîte noire ne suffit pas ou ne constitue pas une solution efficace, les modèles markoviens peuvent être une excellente alternative. Le modèle de Markov ou modèle de Markov caché (HMM) permet non seulement d'observer les états cachés et leur vraisemblance, mais permet également d'accéder et de modifier les valeurs pendant l'apprentissage et à tout moment. Un HMM peut être vu comme un observateur de l’état caché du système.

Le HMM a été introduit par Baum au début des années 1970 (Baum et Petrie, 1966) et (Rabiner, 1989) a proposer une modélisation HMM pour la première fois dans une application de reconnaissance de la parole. Il a ensuite été utilisé dans les défis PHM. Les deux problèmes : la reconnaissance de la parole et l’estimation de la dégradation des systèmes industriels sont en fait très similaires, c'est pourquoi le HMM est étudié dans le domaine PHM. Le HMM est un modèle populaire pour la modélisation de données de séries chronologiques. Nous nous intéressons à un système stochastique, dans lequel l'évolution des états est aléatoire et cachée ou inconnue. Un HMM est bien adapté à l'objectif de notre thèse, car il permet le calcul d’une distribution jointe prenant en compte une série temporelle de distributions conditionnelles.

Cependant, le HMM ne permet aucune entrée dans le modèle alors que notre objectif est de considérer les conditions de fonctionnement comme une entrée. Par conséquent, cette thèse propose d’étudier une version avancée de HMM appelée modèle de Markov caché d'entrée-sortie (IOHMM). Un IOHMM surmonte une partie de la limitation d’un HMM et permet de commuter les modèles en fonction des conditions de fonctionnement répertoriées dans les données. L'IOHMM est introduit en 1995 (Bengio et Frasconi, 1995). Il a été utilisé dans diverses applications (Hu, 2015; Just, 2004), mais à notre connaissance, il n'est pas encore utilisé dans des problématiques de PHM. Une discussion sur les bases du modèle HMM et IOHMM est donnée dans cette thèse. Nous développerons les équations et les algorithmes permettant l’estimation des modèles IOHMM, qui sont présentés et illustrés par des exemples appliqués au PHM.

Les problèmes de diagnostic et de pronostic de systèmes complexes intégrant de multiples conditions de fonctionnement peuvent être classés en trois problèmes majeurs :

1. Considérer l'état de santé quelques soit la complexité du système par un modèle stochastique et apprendre les paramètres du modèle à partir des mesures du système.

> Des informations sur l'état de santé se trouvent dans l'ensemble de données. La question qui se pose est la qualité de cet ensemble de données, est-il bon ou mauvais ? Existe-t-il suffisamment

d'informations dans les données pour estimer les paramètres du modèle ? Comment l'apprentissage gère-t-il les séries de données incomplètes ou les données manquantes ? Ce sont des questions critiques qui constituent l'incertitude des données. Il existe plusieurs autres incertitudes qui doivent être prises en compte dans la formalisation des modèles. Par exemple, le nombre de paramètres du modèle. Il n'existe pas de méthode pour fixer le nombre approprié de paramètres des modèles que nous allons utiliser. La quantité limitée de données est également un problème important dans le cadre PHM, à relier au nombre de paramètres à estimer dans le modèle. Habituellement, une petite quantité de données limite le nombre de paramètres. Alors, comment le modèle gère-t-il une petite quantité de données pour analyser un système complexe ? Plusieurs modèles peuvent-ils être utilisés pour représenter la complexité d'un système, et comment faire confiance à ces modèles ? Pouvons-nous prouver que le modèle est suffisamment bon pour le système ? Le modèle nous permet-il de gérer plusieurs conditions de fonctionnement ? Nous présentons des réponses à ces questions dans cette thèse, où l'approche proposée prend en compte les différentes incertitudes des données, des paramètres du modèle et des conditions de fonctionnement. Dans notre proposition, nous appliquons la méthode Bootstrap-IOHMM pour analyser la confiance sur les paramètres estimés lorsque le modèle prend en compte plusieurs conditions de fonctionnement et des sorties mesurables des systèmes. Une technique pour décider du nombre de paramètres est donnée. Enfin, plusieurs méthodes de validation croisée sont appliquées pour valider le modèle pour une petite quantité de données.

2. Diagnostic et pronostic de la santé du système dans plusieurs conditions de fonctionnement.

Le pronostic nécessite dans un premier temps de diagnostiquer l'état de santé du système. Il existe une corrélation entre les deux estimations (pronostic et diagnostic) qui seront effectuées dans notre proposition via le même modèle. Un autre défi pour le pronostic de l'état de santé du système concerne les conditions d'exploitation futures. Même si nous diagnostiquons l'état de santé actuel, nous devons tenir compte des futures conditions d'exploitation pour déterminer les futurs états de santé. La question que nous nous sommes posée est : comment le modèle peut-il gérer les conditions de fonctionnement si elles sont inconnues ? Étant donné que le fonctionnement futur des systèmes dépend de nombreux enjeux (vitesse de production, délais, date de maintenance, etc.), les conditions de fonctionnement peuvent changer plusieurs fois dans le futur. C'est une question importante qui est prise en compte dans notre proposition. Cela implique également de vérifier si le modèle proposé est capable de gérer les conditions de fonctionnement pour calculer les RUL du système. Ces problèmes sont pris en compte dans la solution proposée, et le modèle proposé nous permet également de faire des diagnostics / prédictions hors ligne et en ligne.

3. Pronostic le RUL pour les systèmes structurés à partir de ses composants pour étudier la dégradation du système dans son ensemble.

Habituellement, les chercheurs se concentrent sur un composant. Le composant est lui-même un système complexe de dégradation, car plusieurs phénomènes sont en jeu (électriques, mécaniques, chimiques, etc.). Il peut aussi être complexe, car le concepteur du modèle a un point de vue global (de nombreux composants pour un sous-système), mais le défi est de s'approcher d'un niveau de complexité acceptable et de combiner des modèles pour gérer des systèmes plus grands en connaissant leur structure fonctionnelle. Alors, la question est de savoir comment construire le modèle d'un système à partir des modèles de dégradation des composants. Nous proposons, à partir du diagnostic des composants, de les combiner pour diagnostiquer l'état de santé global du système. Nous avons supposé que les composants n'interagissent pas les uns avec les autres, mais comme ils font partie du système, leur état de santé représente l'intégrité du système. Nous proposons de définir l'état de santé du système en fonction de l'état de santé des composants et calculer la RUL du système.

Le manuscrit est organisé en 6 chapitres :

Après avoir présenté le concept général de PHM, le chapitre 1 passe en revue les approches de PHM et les modèles correspondants de la littérature existante. Ce chapitre étudie les avantages et les inconvénients des différentes approches et des principales méthodes existantes. Ensuite, une technique de modélisation est choisie en la comparant avec des modèles trouvés dans la littérature. Le modèle est sélectionné en tenant compte de sa capacité, de sa flexibilité et de son adaptabilité. Toutes ces considérations, ainsi que les caractéristiques du système étudié pour permettre de justifier les méthodes retenues et utilisées dans cette thèse.

Le chapitre 2 décrit le contexte du modèle stochastique. Il décrit la chaîne de Markov, la modélisation par HMM, puis IOHMM avec leurs composants, propriétés et fondements mathématiques. La différence entre la chaîne de Markov, HMM et IOHMM est définie dans ce chapitre. Les jalons de HMM et trois problèmes de base sont également expliqués. Différents algorithmes tels que les algorithmes Baum-Welch, Forward-Backward et Viterbi qui sont dédiés à HMM sont décrits dans ce chapitre.

Le chapitre 3 illustre la première contribution de cette thèse. L'algorithme de Baum-Welch et l'algorithme Forward-Backward sont adaptés à la modélisation par IOHMM pour considérer plusieurs entrées et sorties dans le modèle. L'incertitude des données (c'est-à-dire les données manquantes, etc.) et l'incertitude sur les paramètres du modèle (c'est-à-dire le nombre de paramètres, etc.) sont gérées dans l'apprentissage du modèle. La méthode Bootstrap est mise en œuvre pour évaluer la confiance dans l'estimation des paramètres. Les paramètres sont estimés avec des intervalles de confiance à 95%, une valeur moyenne et des erreurs standard.

Le chapitre 4 présente la deuxième contribution de cette thèse. L'algorithme de Viterbi est adapté au modèle IOHMM pour diagnostiquer la santé du système à partir des données d'observation et des conditions opérationnelles. Puis, les algorithmes pour le pronostic sont appliqués pour prédire l'état de santé du système et la RUL. Les pronostics en ligne et hors ligne sont réalisés dans de multiples conditions de fonctionnement. Pour gérer l'incertitude sur les conditions opérationnelles futures inconnues, la simulation de Monte-Carlo est appliquée pour simuler les conditions d'exploitation. Trois applications simulées (numériques) sont présentées pour valider les algorithmes proposés, la méthode Bootstrap et le traitement des données manquantes. Une méthode est mise en œuvre pour gérer les problèmes numériques dans le modèle.

Le chapitre 5 présente la troisième contribution de cette thèse qui est la mise en œuvre des algorithmes d'apprentissage et de pronostic sur les données du défi PHM en 2008. La méthode proposée permet de pronostiquer un moteur d'avion dans de multiples conditions de fonctionnement. L'analyse en composantes principales (ACP) est appliquée à la préparation des données. Différentes incertitudes (données, conditions d'exploitation, etc.) et défis mentionnés au chapitre 4 sont traités dans la démonstration. Une expérimentation est réalisée pour gérer la taille du modèle en fonction des conditions de fonctionnement données. La fonction de distribution de probabilité a été mise en œuvre pour gérer l'incertitude de l'estimation de la RUL. Enfin, trois techniques de validation croisée (leave-p-out, leave-one-out, and k-fold) ont été appliquées pour valider les performances du modèle avec des scores prometteurs d'erreur quadratique (RSE) et d'erreur quadratique moyenne (MSE).

Le chapitre 6 décrit la quatrième contribution de cette thèse qui est la proposition d'une méthodologie pour pronostiquer l'état de santé d'un système structuré qui a plusieurs sous-systèmes ou composants. Chacun des composants est représenté par un HMM qui est estimé et diagnostiqué séparément. Un algorithme est développé pour diagnostiquer l'ensemble du système en utilisant le diagnostic de tous les composants. Après cela, un HMM est construit à partir des paramètres estimés de tous les HMM pour estimer le RUL pour l'ensemble du système. Un exemple numérique est donné pour démontrer une application réelle : le réseau d'eau potable (DWN) dans la ville de Barcelone.

Enfin, tous les travaux de la thèse sont résumés dans la conclusion, et les perspectives sont présentées dans le dernier chapitre.

Introduction

In the context of the current industrial evolution, one of the biggest challenges is maintaining systems with safety, reliability, and availability. This evolution has been called by several terms: Industry of the Future (IoF), Industry 4.0, the fourth evolution of industry, the Advanced Manufacturing Partnership, Made in China 2025, the Factory of Future (FoF). This concept of evolution is based on innovation and digital technology with operational excellence revealing the potential for industry performance transformation and enhancement (Iung, 2018). This book defines it as the Factory of the Future or FoF. There are four main principles that support the industry in identifying and implementing different scenarios for this advancement (Hermann and Otto, 2016):

The first principle is the *interconnection* between the various components of a complicated system via the Internet of Things (Bonner, 2018). A complicated system defines such a system or machine that has multiple components or subsystems within it that operate through multiple operating conditions.

The second principle is *information transparency*, which refers to the ability to provide the operator with a vast amount of information that helps the operator to make decisions related to the operation (Bonner, 2018).

The third principle is the *technological assistant*, which collects and observes information about the system to support humans in making decisions and solving urgent problems (Gronau, 2016). It is important for human workers to be constantly aware of deteriorating system health during operations to support human workers' monitoring of several activities such as operating conditions, production rates, maintenance activities, system failures/unplanned shutdowns, which can cause harm to the human workers and increase maintenance costs.

The fourth principle is decentralized decision-making, i.e. the system is able to make its own decisions and execute them on its own (Gronau, 2016). The system operates itself by considering different anomalies, such as interference, conflicting goals, or automatic scheduling which fixes the time for different actions to be taken during system operations.

The principles of FoF rely on instant information such as operational information of interconnected subsystems, information of technologies that used in the system, information of various components, etc. One of the importance is keeping the industrial systems under continues monitoring to help the user to various operational decisions around the clock. The user needs all information about the system's function and corresponding health status. System health state can be found out by observing system behaviour and apply diagnostics and prognostics techniques on the observations. Moreover, prognostic remaining useful life can help auto-scheduling in potential maintenance for reducing maintenance costs, satisfying legal compliance issues, setting automatically allocated activities, increasing production speed, and profits.

FoF offers modernizing the production tool and increasing the dissemination of digital technologies within enterprise, to support a range evolution and ever greater customization of products. In all over the world, the industrialists are implementing the FoF concept in their industries and the tendency is growing progressively. FoF becoming the major concern and leading concept of manufacturing profit and business policy for the next generation of industry.

In consequences, the systems are becoming complicated gradually with the interconnections between their subsystems with complicated operating conditions. Different subsystems and components of the system are becoming reliant to each other for manufacturing the cumulative productions. This is the case of complex operating systems, where the availability of infrastructure, changing aspect, and system maintenance are becoming challenging every day to meet the needs of our modern societies. Therefore, the study of the system's availability and system's maintenance are now an increasingly high demand in research area. Furthermore, a large number of systems that have been in service worldwide for many years is now nearing the end of their life and therefore require regular maintenance and therefore require more and more frequent servicing. To meet these requirements, maintenance strategies should be improved. Since maintenance actions are taken before a system failure occurs, significant economic losses and decisions on maintenance strategies appear to have an impact on system safety and human co-working. Therefore, in order to define the maintenance plans and optimize system availability, we need to study the Predictive and Health Management System (PHM). This study will help to reduce maintenance costs and ensure high reliability of the system. From an economic point of view, PHM study is an extremely important issue for manufacturers that affect the quality of industry's image in the market.

In PHM applications, prognostic needs the diagnostic; and diagnostic needs to study the degradation of the system. Degradation is a damaging process of the system over the service it gives in its lifetime. The degradation process itself complex because it is hidden, and many constraints impact on the system's health to get affected. The impacts of these reasons differ based on different conditions such as operating conditions, environmental conditions, etc. This thesis studies the degradation of a complex/complicated system considering the operating conditions. If we consider no exception (accident, system/part errors, etc.), and no variation of operating condition on the degradation, then it is reasonably less complicated to understand and to model the degradation because it will have only one dynamic. Otherwise, degradation will have more dynamics or behaviours when there are different varieties of operating conditions applied to the system. The more the operating conditions applied, the more the degradation gets difficult to study. This thesis studies the degradation under multiple operating conditions to propose a global solution of health assessment for those system that have different dynamics in their operations (Shahin, 2019a). The estimated degradation allows to diagnostic of the current health states of the system at the given observation data. The prognostic method then uses this information to predict the future health state of the system. The proposed prognostic method estimates the remaining useful life (RUL) or Remaining Lifetime (RL) of the system from any time to the end of the useful life of the system (Shahin, 2019b). The considered time stands between the starting time of the system until it fails.

Historically, there are two main approaches practiced in PHM society: model-based approach and data-driven approach. The model-based approach, which is also known as the physical approach uses a dynamic mathematical model of the system that directly exploits the physical processes that affect the health of the component (Skormin, 1994). Instead, the data-driven approach allows the degradation model construction by using observation data collected from installed sensors on the system. This approach is usually preferred when system models are not available or not robust enough (Namburu, 2007). Both approaches have their advantages and disadvantages (modelling effort, accuracy, knowledge, etc.). Many researchers used them together to overcome their disadvantages and get all their benefits (Liao and Köttig, 2014). They defined this combination as the hybrid approach. However, we decided to use the data-driven approach for its diversity, flexibility, and advantages which make it easier to accomplish the goal of this research. While the model-based approach requires the physical information of the system, this thesis offers diagnostic and prognostic methods based on the data-driven approach when the physical information is not necessary or available. The data-driven approach is a good fit for such a complex system that has multiple components with nonlinear behaviour. This approach allows us to deal with the high dimension of data for predicting the degradation of multiple components.

Different types of models can be adapted under the data-driven approaches. The most popular and frequently used models are stochastic. A stochastic model is an estimation tool whose analysis focuses on a random sequence of observations. It deals with the stochastic properties of random variables and manages operating loads according to the operating conditions. Thus, the uncertainty of the operating conditions can be handled by the stochastic models. This approach also manages the data uncertainties because of its randomness and approximation properties. On the other hand, it is difficult for the deterministic model to manage the randomness of variables. Therefore, the stochastic model seems to be fit for achieving the objective of this thesis. Additionally, the algorithms behind stochastic models are comparatively less complicated to adapt and use. It allows us to include variables into the mathematical formulas as well.

There are several stochastic models that can be found in the literature which can be built quickly at low cost. Mostly used models are the Bayesian network models, Fuzzy Logic models, Neural Network models and Markov models. The Bayesian network is a probabilistic graphical model that uses Bayesian inference for probability computation. The Bayesian network offers a strong mathematical foundation and is presented graphically so that each variable can be directly connected to each other. It is one of the popular model structures or methods in the current research field. However, one of the main limitations of Bayesian networks is the discrete treatment of continuous variables. Though, it can only deal with continuous variables in a limited way (Friedman and Goldszmidt, 1996; Jensen, 2001; Weber and Simon, 2016). Another popular model is the Fuzzy logic that works following human reasoning. It is popular for its flexibility in using simple mathematics for nonlinear, integrated, and complex systems. The development of fuzzy rules and membership functions is cumbersome, and the fuzzy output can be interpreted in various ways, which makes analysis difficult. In addition, developing a fuzzy system requires a large amount of data and expertise. In this case, the Neural network offers a great alternative solution. The mathematical foundation of the Neural network allows us to handle a large amount of nonlinear data and inputs. This model is flexible to use for both regression and classification problems. Yet, since this model relies on a large amount of training data, it can lead to overfitting and generalization problems (Yi, 2018). Furthermore, it works like a black box, so it does not give any information about how much the independent variable affects the dependent variable, nor how the entire hidden layer of likelihood evolution is developed. Thus, for those cases in which the black box concept is not enough or an effective solution, Markovian models can be a great alternative. The Markov model or Hidden Markov Model (HMM) not only allows to observe the hidden states and their likelihood but also gives access and change the values during training at any time.

HMM was introduced by Baum in the early 1970s (Baum and Petrie, 1966), and (Rabiner, 1989) used it for the first time in an application for recognition of speech. It was later used in PHM challenges. Two problems are similar and close to recognizing speech and degradation that is why the HMM is investigated in the PHM domain. HMM is a popular model for time series data modelling. We are interested in a stochastic system, in which state evolution is random and hidden or unknown. HMM is well fit for the objective of this thesis which allows the joint distribution to be factored into a series of conditional distributions.

However, the HMM does not allow any input to the model while our goal is to consider the operating conditions as input. Therefore, this thesis offers the advanced version of HMM called the Input-Output Hidden Markov Model (IOHMM). IOHMM overcomes some the limitation of HMM and allows to switch the models according to the given input operating conditions. IOHMM is being introduced in 1995 (Bengio and Frasconi, 1995). Since it has been used in various applications (Hu, 2015; Just, 2004) but in our knowledge, it is not yet being used in PHM. A brief discussion about the basics of the model is given in this book in which the difference between several versions of HMMs along with the benefit of IOHMM is presented with examples and application to PHM.

Research questions

The issues in diagnostic and prognostic of complex systems under multiple operating conditions can be categorized by three major problems:

1. Considering health state with whatever the system complexity is by a stochastic model and learn model parameters from system measurements.

Information on health status can be found in the data set. But now, the question comes, what to say about this data set, is it good or bad? Is there enough information in the data to train the model? How does the model handle incomplete data series or missing data? These are sensitive issues need to be studied even under the data uncertainty. There are several other uncertainties that need to be taken into account in model training. For example, fixing the number of model parameters. It is a difficult task to fix the appropriate number of model parameters. Limited data is also an important issue in which the number of parameters is relevant. Usually, a small set of data limits the number of parameters. Several models can be used to represent the complexity of a system, how can we compute the confidence in these models? Can we prove that the recommended model is good enough for the system? Does the model allow us to manage multiple operating conditions at a given input sequence? We present a solution for these issues in this book, where the proposed approach takes into account the different uncertainties of the data, model parameters, and operating conditions. In our proposal, we apply the Bootstrap-IOHMM method to analyse the confidence of the estimated parameters when the model takes into account multiple operating conditions and measurable outputs of systems. A technique to decide the number of parameters is given in the following chapter. Lastly, several cross-validation methods are applied to validate the model for a small amount of dataset.

2. Diagnostic and prognostic of the system health under multiple operating conditions.

Prognostics the health states require diagnosing the current health status of the system. There is a correlation between the two estimations that need to be done through the same model structure because we cannot use the model to predict future health states if we do not learn the model from previous health states. Another challenge for the prognostic system's health is the future operating conditions. Even if we diagnose the current state of health, we need to consider the future operating conditions to determine future health states. The question is, how does the model manage the operating condition if it is unknown? Since the future operation of systems depends on many issues (production speed, deadlines, reaching maintenance dates, etc.), the operating conditions may change several times in the future. It is an important issue that is taken into account in our solution. However, this implies that there is a relationship between operating conditions and predictive degradation which leads to another important question of whether the proposed model capable of managing operating conditions to manage RUL of the system to reach a given date (maintenance date, etc.) before a failure occurs. These problems are considered in the proposed solution, and the model also allows us to make offline and online diagnoses/predictions.

3. Prognostic the RUL for structured systems from its components to study the entire system reliability.

Usually, researchers focus on a component. The component itself has a complex degradation process because it can degrade in materials, electrical, mechanical, chemical, etc. It can also be complex because the model designer has a global point of view (many components for a sub-system) but the challenge is to go close to an acceptable level of complexity and to combine models to handle bigger systems knowing their functional structure. So, the question is how to do that? Well, one solution can be diagnosed the components separately then combine them together to diagnosis the global health state of the system. We assumed the components do not interact with each other, but since they are part of the system, so their health states represent the system’s health. We present a solution for these issues in this book to define the health state of the system given the health state of the components.

The manuscript is organized in 6 chapters:

After introducing the general concept of PHM, Chapter 1 reviews PHM approaches and corresponding models from the existing literature. This chapter studies the advantages and disadvantages of different approaches and the main existing methods. Then a modelling tool is defined by comparing with comparable models found in the literature. The model is selected under consideration of its capability, flexibility, and adaptability. All these considerations together with the characteristics of the system studied to make it possible to justify the selected methods used in this book.

Chapter 2 describes the background of the stochastic model. It describes Markov chain, HMM, and then IOHMM with their components, properties, and mathematical foundations. The difference between Markov chain, HMM, and IOHMM are drawn in this chapter. The milestones of HMM and three basic problems are explained as well. Different algorithms such as the Baum-Welch, the Forward-Backward, and the Viterbi algorithms that are dedicated to HMM are described in this chapter.

Chapter 3 illustrates the first contribution of this thesis. Baum welch algorithm and the forward-backward algorithm are adapted to IOHMM for considering multiple inputs and outputs in the model. The data uncertainty (*i.e*. missing data, etc.) and the uncertainty about model parameters (*i.e.* number of parameters, etc.) are handled in model training. The Bootstrap method is implemented for developing confidence in parameter estimation. The parameters are estimated with 95% confidence intervals, mean value, and standard errors. Then, the mean values are used to construct the models to use in the diagnostic and prognostic health state of the system.

Chapter 4 presents the second contribution of this thesis. The Viterbi algorithm is adapted to the model to diagnostic system's health at given observation data. Later, the prognostic algorithms are applied to predict the RUL. Online and offline prognostic are made under multiple operating conditions. To prevent the uncertainty of future operations at unknown operating conditions, the Monte Carlo simulation is applied to simulate the operating conditions. Three simulated (numerical) applications are presented to validates the adapted algorithms, bootstrap method, and missing data handling, and a method is studied to manage the numerical problems in the model.

Chapter 5 demonstrates the third contribution of this thesis which is an application given by the PHM challenge in 2008. The proposed method is used to prognostic an aircraft engine under multiple operating conditions. Principle component analysis (PCA) is applied to data preparation. Different uncertainties (data, operating conditions, etc.) and challenges mentioned in chapter 4 are handled in the demonstration. An experiment is done to manage the model size according to the given operating conditions. The probability distribution function (PDF) was implemented to deal with the uncertainty of RUL estimation. Lastly, three cross-validation techniques (leave-p-out, leave-one-out, and k-fold) were applied to validate the model performance with promising scores of root square error (RSE) and mean square error (MSE).

Chapter 6 describes the fourth contribution of this thesis in which the proposed methodologies to prognostic the structured system that has multiple subsystems or components. Each of the components is represented by an IOHMM which is learned and diagnosed separately. Then, an algorithm is developed to diagnose the entire system by using the diagnostics of all the components. After that, the model is built from the estimated parameters of all the IOHMMs to estimates the RUL for the whole system. A numerical example is given to demonstrate a real application: drinking water network (DWN) in Barcelona city.

Finally, all the work in the thesis is summarized in the conclusion, and the perspectives are given which will be considered for future works.

Chapter 1

State of the Art: The generalities of maintenance, diagnostic, prognostic and PHM

Table of Contents

1 State of the Art

Prognosis and Health Management (PHM) is a multidisciplinary field with evolving capabilities and needs. It uses the real-time information of health states of subsystems and components to provide actionable knowledge that enables intelligent decisions to improve performance, safety, reliability, mission criticality, and economic viability among others (Saxena, 2010). The goal of PHM is to provide decision support, *i.e.* actionable information to support decision making (Kalgren, 2006).

PHM of a component or system involves diagnostics and prognostics activities. The diagnostic is the process of detecting the health states at the considering time, while prognostic is the process of predicting future states mainly through the remaining useful life (RUL) based on the diagnostic (Ly, 2009). Prognostic performs by understanding that the system will fail after a period of degradation or will not satisfy its mission, and if it is measured, it can be used to prevent system failure and minimize operating costs (Tian, 2012). Essentially, PHM is a method of assessing system reliability to predict the probable failures (Sun, 2010) which reduces the time for planning maintenance (Banks and Merenich, 2007).

1.1 Maintenance

Maintenance can be defined as "all technical, administrative and management actions during the life cycle of an asset, intended to maintain it or restore it to a state in which it can perform the required function” (Chebel-Morello, 2010). The goal of maintenance is to preserve or restore the functionality of systems and products throughout their life cycle. Most maintenance is either completely reactive or blind preventive maintenance (Djurdjanovic, 2003). The oldest maintenance strategy is fixing after the breakdown which leads to the problems of unplanned downtime, potentially serious safety violations, and potentially significant damage to the manufacturing equipment. The next natural step is to monitor and maintain the system at predetermined intervals (preventive maintenance), which is often costly (Kothamasu, 2006). The development of reliability engineering in the 1950s led to the introduction of time-based maintenance (TBM) based on increasing failures over time (Takata, 2004). Then, in the 1970s, the development of machine diagnostics led to the concept of condition-based maintenance (CBM) (Ahmad and Kamaruddin, 2012; Jardine, 2006; Latrous, 2018; etc.), where preventive actions are based on detected failure symptoms. Nowadays, to minimize the probability of failure, downtime, and maintenance costs, we use of diagnostics and prognostics activities to predict the RUL of the system (Kothamasu, 2006).

Figure 1 is a general flowchart of the PHM system development which is adapted from (Vogl, 2019). This figure is divided into two processes, (a) general PHM system development process and (b) essential PHM system process. The general process begins with a cost and reliability analysis to identify the components to be monitored. The data management system is then initialized so that maintenance data can be collected, processed, visualized, and archived. Once measurement techniques are established, diagnostic and prognostic methods are developed and tested to ensure that the desired objectives are achieved. Finally, the system gets validated and verified through an iterative process of training and testing modules prior to the final deployment of the system. After that, the general PHM process emerges with the essential steps of the PHM process. The next section discusses the essential aspects of the PHM process where different steps of diagnostics and prognostics act and reach the conclusion to plan the maintenance.

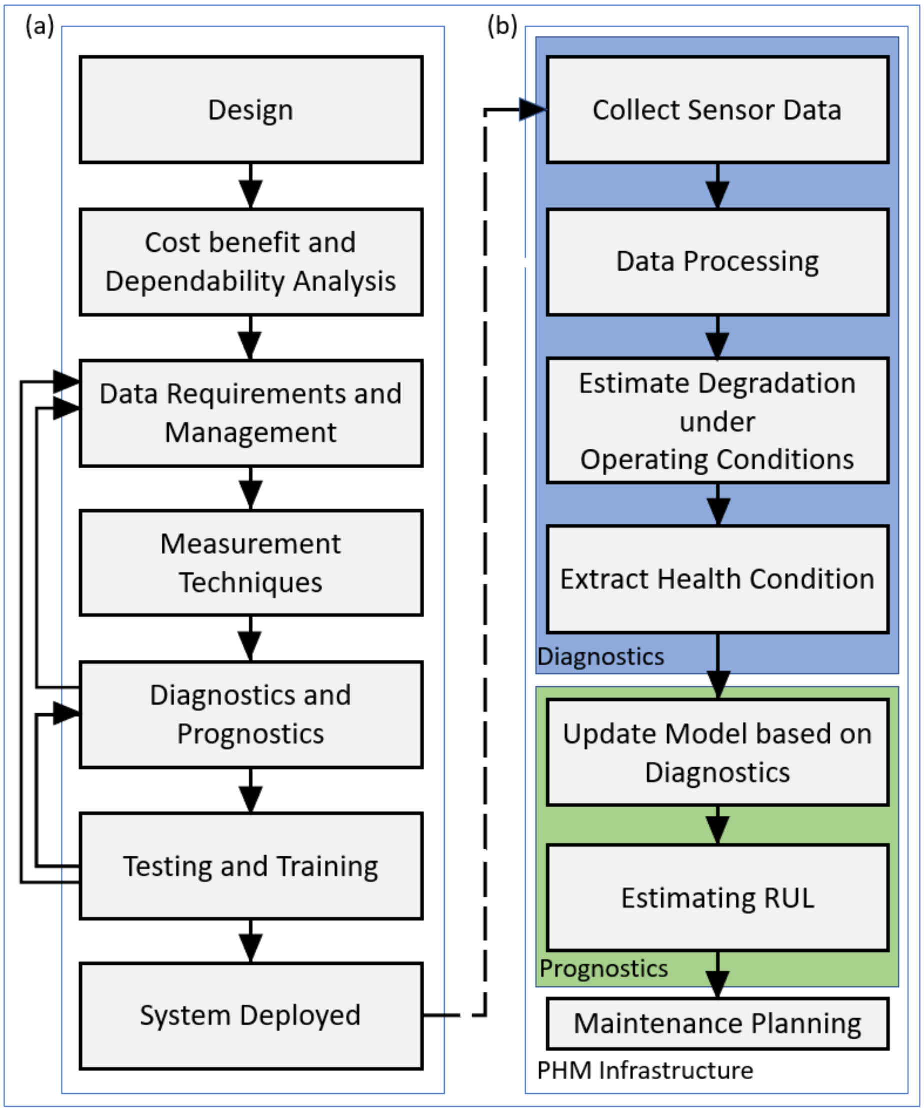


Fig. 1: General PHM system development process (a) and essential PHM system process (b)

Different challenges and needs of the essential PHM process are shown in Fig. 1 (b), as discussed in this section.

1.2 Degradation

Degradation of a system is a process of lowering the health condition of the system to a less respective state. It is also known as a downcast state. In the case of a stochastic process that characterizes the physical deterioration (if observable), the prognostic models will obviously affect the prediction of remaining useful life, and therefore influence the decision of the maintenance strategy and its economic performance. It can be classified based on whether the degradation states are discrete or continuous. There are several methods applied to model degradation (*e.g.* Chen, 2019; El Hajj, 2016; Hao, 2019; Oumouni, 2019; Saxena, 2019; etc.). However, in the scenario with discrete states, Markov chain models (Bloch-Mercier, 2002; Chen, 2003; Xiang, 2012; Yeh, 1997) are often adopted. After specifying the probability transition matrix among all states, these models can be used to compute the time-to-failure distribution from any state. The Markov chain models are particularly useful when the degradation states of the system cannot be precisely measured. That is why, a roughly presumed category (*e.g.* good, moderate, bad, etc.) is usually assumed as different health states of the system (Fig. 2). After this discretization, the remaining useful life may depend only on this granularity.

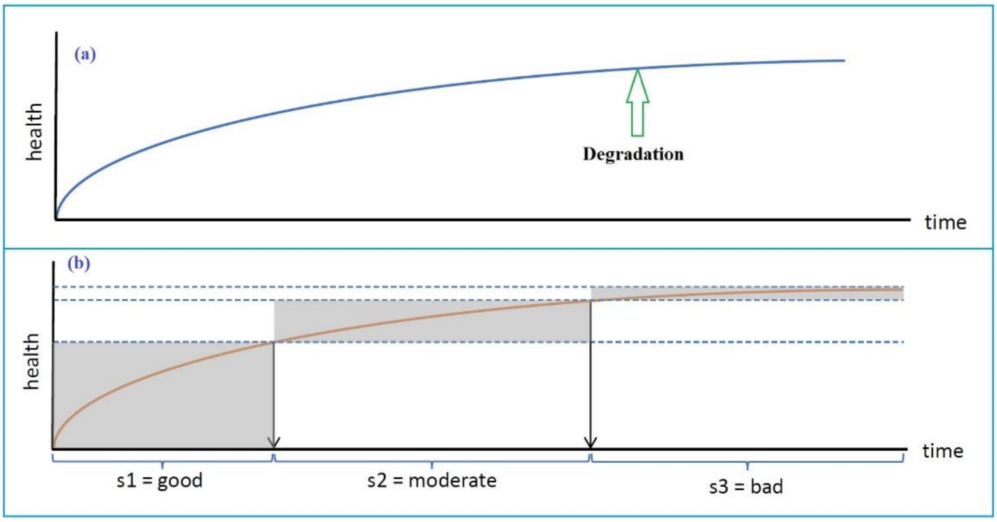


Fig. 2: Health degradation of systems

Great reviews on degradation process have been done in (Knights, 2004; Schmittinger and Vahidi, 2008; Wu, 2008; Yousfi-Steiner, 2009, 2008). They are relative to various fields: component degradation, water management, contamination (CO poisoning, presence of impurities initiating chemical attacks), reactant gas starvation or thermal management (influence of freezing or elevated temperature). If the degradation is not properly monitored and estimated, then it can lead to inaccurate diagnostics and prognostics information so thus the inefficient maintenance planning.

1.3 Diagnostic

The collected data represents deferent point of view of systems. In this book, we consider the data that is collected from sensors where the data provides information about the damage level of the health conditions or the degradation. The health condition is defined as an observed variable that changes from good state to bad state over the lifecycle of the system. The data represents the symptoms of abnormality of the system compared to the performance over time. The goal is to estimate the current condition of health according to the symptoms. Diagnosis of damage level for the system’s health requires several features that apply to the system (or component) usage information and a diagnostic search strategy that can match the observed symptoms and the known set of possible states. Furthermore, the results of diagnostic are needed for developing probable failure detection (prognostics) so that system breakdowns are avoided. The ability to right diagnosis is challenging due to the data uncertainties, lack of information (Patrick, 2009), dependence on environmental and operating conditions, and uncertainties in maintenance schedules.

In this section, a description of some requirements that the diagnostic process has to be considered in order to be used in prognostic applications. For a machine learning model to be useful in solving system health diagnostic, the following features are required to be considered: good performance, dealing with missing data, dealing with data uncertainty, the ability to deal with multiple operating conditions, the transparency knowledge, and the ability of the diagnostics algorithm to reduce the tests necessary to obtain reliable information.

a. **Good performance**: The algorithm must be able to extract meaningful information from the available data. The diagnostic accuracy for new measurements should be as high as possible. In most learning problems, the different approaches (leave-p-out, leave-one-out, k-fold, etc.) used by the selected algorithm may be a good solution, as one approach may be significantly better than the other (Michie, 1994). Therefore, in terms of performance criteria, these approaches can hardly be excluded. Instead, several learning methods should be tested on the available data and the use of the few methods with the best-estimated performance should be considered in application development.

b. **Dealing with missing data**: In health diagnosis very often the sequences of system's records lack certain data. The algorithms must be able to appropriately deal with such incomplete sequences of data elements.
c. **Dealing with data uncertainty**: Sensor data typically suffer from uncertainty and errors. Therefore, the selected algorithm must have effective means for handling the uncertainties (*e.g.* noise or error in data).
d. **Operating conditions:** Operating conditions define the operational context of the system or components based on various circumstances or requirements. A system may have multiple operating conditions, for which the system undergoes various dynamic degradations.
e. **Transparency of diagnostic knowledge**: The generated knowledge and the explanation of decisions should be transparent to the agents (*e.g.* operator, other methods, etc.). The knowledge should be able to analyzed and understandable. Ideally, the generated knowledge provides a novel point of view on the given problem and may reveal new interrelations and regularities by the operator to the next operations, as well as the prognostic method to the next predictions, etc.
f. **Reduction of the number of tests**: In PHM practice, the collection of data is often expensive and time-consuming. Therefore, it is desirable to have a classifier that can reliably diagnose with a small amount of dataset. However, the process of prediction can be done by using the subsets of data several times which may be an effective solution. Some methods are themselves able to select an appropriate subset of data, *e.g.* the bootstrap method where the data selection is done randomly during the learning process.

Diagnostics are vital to successful prognostics, as acceptable prognostic approaches begin with robust diagnostic since the uncertainty in estimating the health state of the system affects any future predictions (Hess, 2006; Patrick, 2009). Challenges to diagnostics exist in the problems of validation and confirmation for the false alarms. The cause of such false alarms is one of the main challenges of PHM systems. Furthermore, systems can be complex and have multiple operating conditions. Therefore, diagnostic and prognostic methods must be able to handle different uncertainties (*e.g.* operating conditions, data, etc.). Otherwise, these uncertainties can lead to high false alarm rates, inaccurate predictions, and consequently, wrong decisions (Hess, 2006).

1.4 Prognostic

Global performance requirements have led industries to enhance their ability to predict degradation phenomena and failures. This is largely due to the realization of prognostic capabilities, which appear to be a key process in the move from "failure to repair" to "predictive to prevention" strategies (Fig. 3) which improves system reliability, availability, and safety while reducing costs and downtime. However, prognostic is even more challenging than diagnosis, which mainly explains why prognosis is an underdeveloped component of PHM systems (Belkacem, 2017; Patrick, 2009). Some failures are occasional and therefore difficult to predict (Sun, 2010). Hence, there is still no universally accepted method of prognostics (Lee, 2011). Despite being a very challenging part of PHM, it is also one of the most beneficial aspects of industry 4.0's perspectives (Hess, 2006).

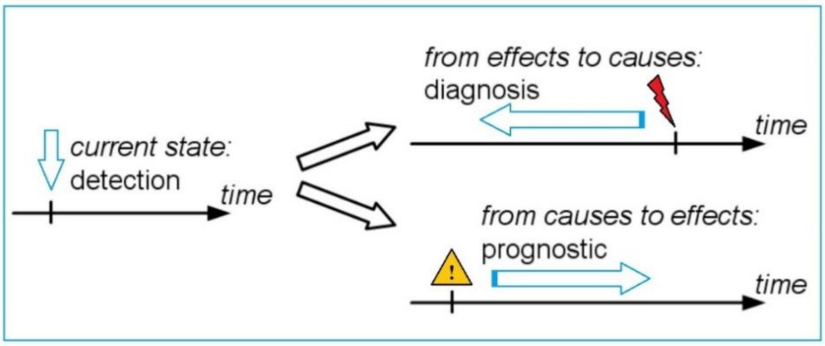


Fig. 3: Anticipating instead of repairing (Jouin, 2013)

Although, there are some different points of view in the literature. The prognostics can be defined as “prognostics” by the International Organization for Standardization (ISO) which is the estimation of time to failure currently or in future (ISO13381-1, 2004). In this view, prognostics is also referred to as "system life prediction" because it is a process that aims to predict the Remaining Useful Life (RUL) of systems under current health states and future operating conditions (Fig. 4).

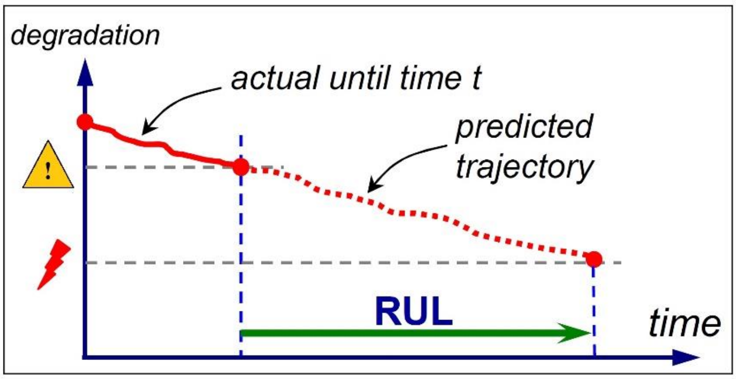


Fig. 4: RUL illustration (Jouin, 2013)

RUL prediction considered for nearly half of all PHM data challenge applications due to their importance in assisting with maintenance planning and health management (Kononenko, 2001). In practical applications, it is common to view health assessment as a critical step in support of RUL prediction, which aims to estimate the remaining operating time of a system before its health deteriorates to a threshold that indicates system failure or unacceptable production quality.

RUL prediction is essentially a conditional-based estimation to make better-informed maintenance decisions. Without a corresponding measure of uncertainty, the RUL has little value (Engel, 2000; Sandborn and Wilkinson, 2007). Thus, RUL estimation is a "recognized challenge" in the PHM system (Ly, 2009). Indeed, few PHM methods produce continuous real-time RUL estimations (Patrick, 2009) and improve the prediction based on new measurements. Furthermore, systems can be complex with perhaps thousands of subsystems and various operating conditions which make it more difficult to predict the reliability and performance of the system (Lee, 2011). Therefore, the prognostic algorithms must be able to adapt these characteristics to provide high-quality performance. Since the prediction of unknown future is inherently uncertain, it should be viewed as a probabilistic process, where the predicted RUL is represented by a probability density function (PDF). The PDF will then be used to inform the maintenance engineer about the turnaround time based on the required maintenance process.

The limitation of prognostic is that there will always be a constraint to the accuracy and precision of condition-based RUL estimation due to the inherent uncertainty in predicting the future. Failure mechanisms have a certain amount of physical randomness and environmental conditions that increase the inherent error of the prediction process due to imperfections in sensor data, pre-processing, feature extraction, and failure prediction methods (Engel, 2000). In fact, prognostic may not be feasible due to the highly unpredictable nature of failure modes (Roemer, 2001). Multiple failures can also complicate RUL prediction (Engel, 2000).

In the following sections of this chapter, the most popular and best-performing methodologies in the field of prognostics and health management are identified. Section 2.1 describes the modelling approaches used in this field or relative research for over 50 years. Section 2.2 presents the different types of models developed using the methods identified in the previous section and their performance, this section presents why stochastic models are chosen to achieve the objectives of this thesis. Section 2.3 reviews several stochastic models, which are listed based on optimal performance. Finally, the proposed modelling tool is selected to design such a system that provides the opportunity to practice the proposed methodology.

1.5 PHM approaches

Machine learning methods and algorithms have been intensively studied at PHM over many years. Several researchers are studying several strategies to implement these algorithms through PHM (Kumar, 2008; Lamoureux, 2015; Lee, 2014; Saxena, 2010; Uckun, 2008). These algorithms can be divided into three main categories: (1) physics or model-based methods, (2) data-based methods, and (3) hybrid methods. Another approach built on expert knowledge can be found in the literature called experience-based approach. However, only the most common methods are discussed in this book. Finally, after understanding the advantages and disadvantages of each approach, a statistic method is selected for accomplishing the objective of the study.

1.5.1 Model-based prognostic approaches

Model-based approaches, also known as physical approaches, are one of the commonly used approaches for failure analysis and identification applications. The approaches utilize a dynamic mathematical model of the system that directly exploits the physical processes that affect the health of the components. The results are used in intelligent monitoring systems that work well under any load, including steady-state and transient conditions and unanticipated operating conditions and regimes (Skormin, 1994). These approaches provide good results for early fault detection by monitoring physically important parameters. The behaviour of the system degradation process leading to prognostics that are described by mathematical models in which the equation is derived from the physical system.

Mathematical models are combined with parameter identification and prediction of future health evolution. Model-based prediction methods commonly used in the literature are the Paris-Erdogan law (Irwin and Paris, 1971; Paris and Erdogan, 1963) and the Forman law (Forman, 1972). Some recent applications of model-based bearings and batteries are given in (Hu, 2019; Sierra and Goebel, 2020; Wang, 2020). Hu proposed methods for near degradation and RUL estimation, and Wang and Sierra proposed methods for state estimation and prediction applied to battery life. Many other applications, such as fault prediction for electro-hydraulic drives (Dalla Vedova, 2020) and prediction algorithms for electromechanical flight control drives (Di Rito, 2018), are proposed through model-based approaches.

A model-based approach for lithium-ion cells is presented in (Daowd, 2010), where the degradation model of the equivalent battery circuit is adapted. (Tagade, 2016) have developed an improved Particle Filtering (PF) algorithm to display battery status estimation for different driving cycle protocols. In (Eker, 2015), a physics-based model for filter clogging phenomena was presented. An overview of forecasts based on dynamic system models is presented in (Ekanayake, 2019), where different opinions on advantages and disadvantages are described.

Advantages and disadvantages

From a lot of papers on the subject, several pros and cons of using a model-based approach can be drawn. For instance, less interest in historical knowledge (data), and false alarm rates can be adjusted. Although model-based approaches can provide relatively accurate predictive results. It is very challenging to drive models from real physical systems due to their complexity and the random degradation behaviour of components which is a complex process with concurrent form. In some cases, it may not be a viable solution in industrial applications due to the uniqueness of the failure mode or the dynamic evolution of the damage, and the specific theoretical knowledge of the physical system. The main model-based prediction tools studied in the literature are presented in Table 1 below with their areas of application, advantages, and disadvantages.

Table 1: Model-based prognostics tools and their advantages and disadvantages.

Prognostic tools	Advantages	Disadvantages	Application	References
Paris' law (PL)	-Model parameters are adaptable for conditional changes	-Linear correlation with defect size and vibration RMS level -Empirical determination of material constants are needed	-Bearings, gearbox, fatigue crack propagation	(Ding, 2018; Pugno, 2006; Xu, 2012)
Forman law (FL)	-Links both monitored data and crack growth physics to life models	- Poor results for complex systems	-Fatigue crack propagation	(Nelson, 1982)
Fatigue spall initiation/progression model (FSI/P)	-Calculates time up to initiation and from initiation up to failure -Damage is cumulative	-Many parameters to be determined	- Bearings, fatigue crack propagation	(Kotzalas and Harris, 2001; Sadeghi, 2009)
Kalman Filter (KF)	-Estimates current/future states -Estimation error is corrected with the latest measurement	-System/measurement model need to be defined -Sensitive to noise -Applicable to linear systems with Gaussian noise	- Gearbox bearings, PEMFC, batteries	(Aidala, 1979; Chen, 2019; Sun, 2011)
Particle Filter (PF)	-Applicable to non-linear systems with non-Gaussian noise -Better accuracy -Avoids degeneracy problem by resampling	-System/measurement model need to be defined -High dimensional data increases computational cost	-PEMFC, batteries, filter clogging, fatigue crack propagation	(Cheng, 2017; Schwunk, 2013; Sreenuch, 2015)

The degradation processes of subsystems and components in systems may differ from those of other systems, which limits the applicability of physical models to system-wide predictions. Thus, another approach is widely used in PHM, the data-based approach. Due to its scalability, flexibility, and quick implementation, data based PHM has clearly become a necessity for the next industrial revolution.

1.5.2 Data driven prognostic approaches

The data-driven approaches are other popular approaches that allows the use of observational data collected from sensors to build degradation models. These approaches are often preferred when the system model is not available or robust enough. For example, the underlying physics is too complex to model, but the observational data can be used for systematic monitoring (Namburu, 2007). Different parameters such as vibration, temperature, velocity, and pressure are continuously collected to detect any changes in the normal operating conditions of any system. These parameters can be analysed by observing and collecting data from the daily operation of the system as soon as a new data set arrives at the monitoring centre. The mathematical foundation of modelling the system is designed from this data, that notifies the monitoring and diagnostic-based decision-making activities. Different PHM activities such as prognostics, RUL, diagnostics, and maintenance related literature are given below.

Several prognostics applications have recently been proposed (Balali, 2019; Elattar, 2019), where distributed electrical systems (Balali, 2019) and safety-critical systems (Elattar, 2019) are monitored through data-driven approaches. In (Khumprom and Yodo, 2019), a data-driven prediction strategy for aero-engine degradation prediction is proposed, and (Wang, 2019) combines this strategy with another class of deep learning approach for predicting the prognosis of the system battery. Different data-driven remaining life estimators for cutting tools (Liu, 2019) and bearings (Zhu, 2020) have recently been proposed, in which Zhu considers remaining life estimators for various operating conditions in his algorithm. Many authors used this approach in combination with several other methods to have different perspectives. In (Ellefsen, 2019), the authors combined a data-driven labelling approach with a novel deep network structure for residual lifetime prediction. Liu proposed an approach based on this method covering several detections and data recovery technique (Liu, 2019). Mosallam combined this approach with a Bayesian approach in (Mosallam, 2016) for direct RUL prediction.

Several degradation models based on data-driven approaches have recently been proposed. Severson proposed a method for predicting cell lifecycle before capacity degradation (Severson, 2019). (Wu, 2008) proposed a degradation model as a time-fault prediction data-driven method for rolling body bearings in motors. In (Ma, 2018), the authors provided another method for fuel cell degradation prediction. They combine a data-driven approach with a deep learning approach. A data-driven approach is presented in (Esmaeili, 2019) for predicting the effect of temperature on the relative permeability of oil and water.

Furthermore, predictions of degradation data for miniature light bulbs were studied by modelling the trend and seasonal components of the time series analysis technique (Huang, 2010). Other interesting applications can be mentioned, where the authors used the data-driven approach to obtain a better result. For example, in (Hu, 2012), a dataset-based prognostic approach was proposed, which combines several algorithms using different sum-weighted functions for accurate predictions. (Liu, 2007) developed a new prediction algorithm, called Correspondence Matrix (MM), with a long prediction horizon. The non-parametric statistical method of Gaussian process regression (GPR) has been adopted for the prediction of nuclear power plants (Baraldi, 2015). In (Yu, 2011), Gaussian Mixture (GM) has been integrated into the feature extraction methodology for diagnosing bearing damage. An off-line methodology for structural health prediction was proposed in (Hu, 2012).

A review on degradation estimation based on the data-driven predictive model approach is given by (Balali, 2020). The author reviews the general data-driven approaches for the most important degradation-based reliability estimation models proposed by several researchers during the last few decades. Many other scientists have given their discussions and reviews on diagnostics and prognostics applications based on these approaches. For example (Heidary, 2018) reviews various pitting corrosion degradation models for PHM analysis of oil and gas pipelines. In (Wu, 2016; Li et al. 2019), the authors reviewed different data-driven methodologies for estimating the lifetime prediction of lithium-ion batteries. (Zhang, 2019) review this approach for predictive maintenance of industrial equipment. Another important review on system hardware diagnostics by different approaches where some important work based on the data-driven approaches is discussed. The necessity of these approaches is highlighted, and some applications are given as examples. The key features along with its advantages and disadvantages are presented hereafter.

Advantages and disadvantages

The data-driven approaches show the importance of research experience. It may not always be the best in all cases, but it allows business leaders to evaluate success in some cases. Data can be an important resource for improving decision-making, communication, and productivity. We see some of the advantages and disadvantages of these approaches, categorized by the forecasting tools and applications in Table 2.

Data uncertainty is one of the major challenges in data driven approaches. Data required to be good qualitative and should be degradation trend for designing an effective model. The data driven algorithms are difficult to adjust for false alarms to misdetection rate. However, by understanding its limitations, this approach can provide more practical and available solutions for diagnosing and predicting complex systems rather than building physical models. Various algorithms can be easily combined to enhance the results such as multicomponent degradation, or overall system’s degradation.

Table 2: Data-driven prognostics tools and their advantages and disadvantages.

Prognostics Tools	**Advantages**	**Disadvantages**	**Application**	**References**
ARIMA Models	-Applicable to linear systems with stationary behavior -Uses less amount of data	-Short term prediction -Not useful for non-stationary processes	Rotating machinery	(Wen-ping and Peng, 2004; Wu, 2007; Zhou, 2020)
Match Matrix (MM)	-Deals with high dimensional data	-Needs sufficient historical data	Rotating machinery	(Li, 2018)

	-Provides long term prediction -Suitable for non-stationary processes	-Data should have degradation trend		
Gaussian Mixture (GM)	-Many Gaussian functions can be used to approximate an arbitrary distribution and accuracy	-Initialization methods are important in parameter optimization -Determining number of mixtures is difficult	Bearings, CNC machines	(Kong, 2017; Nelwamondo, 2006; Yu, 2011)
Gaussian Process Regression (GPR)	-Adaptable to environment and can learn from experience	-Needs covariance function determination -Suitable for Gaussian likely hood	Nuclear power plants, batteries	(Baraldi, 2015; Richardson, 2017; Yang, 2018)
Artificial Neural Networks (ANN)	-Applicable for complex systems and which have non-linear behavior -Adaptable to the system	-Network structure is not determinable -Needs resources for computation	Bearings, batteries, Turnout point machines	(Michalak, 2019; Sharma, 2019; Wu, 2018; Zhang, 2019)
Fuzzy Logic (FL)	-Inputs can be imprecise noisy/incomplete -Appropriate for complex systems	-Needs rule development based on expert knowledge	Bearings	(Du, 2020; Rehman, 2019)
Bayesian Networks (BN)	-The number of structure parameters are reduced by conditional probability distribution -Visualizes variable pair dependency links	-Has complex and costly learning -Prior knowledge is needed	Bearings	(Lu, 2020)

1.5.3 Hybrid approaches

As discussed in the previous sections, both model-based and data-based prognostic approaches have their advantages and limitations. The hybrid approaches are such approaches that aim to integrate the advantages of different approaches and minimize their limitations to better estimate health status and predict RULs at the system and component level. None of the approaches proposed in the literature is superior to others but is suitable for a practicable problem. It is therefore important to note that the advantages of the forecasting approach can only be determined on a case-by-case basis. In (Liao and Köttig, 2014), the author presented a comprehensive literature review that aims to develop a hybrid prediction method by combining the advantages of different prediction methods. They used the hybrid predictive approach as a case study to validate and develop its potential benefits in degradation applications. He divided the prediction models based on experience, data and physics, and proposed different combinations of mixed prediction methods.

1. experience based plus data driven
2. experience based plus physical based or model based
3. data driven plus physical based
4. more than one different data driven models
5. experience based plus data driven plus physical based

(Satish and Sarma, 2005) proposed an economic method of predicting bearing failure using a combination of ANN (artificial neural networks) and fuzzy logic, and (Swanson, 2001), proposed a hybrid approach using Kalman Filter (KF) and fuzzy logic algorithms to solve the crack propagation problem. In (Gebraeel, 2004), an RNA (recurrent neural networks) based hybrid bearing damage prediction method was proposed. The authors collected vibration signals from 25 accelerated bearing tests and trained 25 ANNs to predict bearing failure times. Then, the remaining life is predicted by weighting the output of ANN. Peng, in (Peng and Dong, 2011) proposed a hybrid approach of HMM and grey model to predict pump wear. (Gu, 2010) also studied the gray prediction model for electronic prediction and integrated the gray prediction model with the HMM (Hidden Markov Model) RUL prediction algorithm and integrated it with the aging factor for asset prediction. (Kumar, 2008) proposed a hybrid approach based on data and model for electronic prediction. (Hong-feng, 2012), proposed a fusion framework to prognostic and health management of systems by using data and model-based approaches.

1.5.4 Conclusion

The systems to be monitored are mainly complex systems with multiple components where each of the components has multiple operating conditions which are one of the main reasons for their health degradation in different dynamics. System monitoring is defined as monitoring the health states of the system in this book which is a difficult challenge due to the uncertainties of diagnostics and prognostics. This section has presented different PHM approaches which are classified into three main groups: model-based, data-driven, and hybrid approaches.

- The model-based approaches are used based on the system's physical degradation phenomena. These methods are generally good, efficient, and give the best results. However, they are complicated to implement and are mainly applied only to simple systems. In order to model complex systems, this approach does not perform reasonably while considering uncertainties.
- The data-driven approaches consist of analyzing data to estimate the state of degradation and then predict the remaining useful life span of the system. The performance of these methods depends on the quality and quantity of the data. These methods are applicable to complex systems and give comparatively good results than the model-based approaches. They are able to adapt the environmental conditions and learn from experience. The uncertainty can be characterized in terms of the instants of a probability density function through this approach. In this book, the data-driven approaches are chosen to practice the PHM activities of system health. We want to propose a diagnostic and prognostic method without having a deep knowledge of the physical condition of the system. The scope of the approach allows us to model systems with multiple components and nonlinear behavior. The proposed method should deal with the high dimension but less amount of data for predicting the degradation of multiple components. These are the major reasons why the data-driven approaches are suitable and appropriate approaches.
- The last group is the hybrid approaches which basically combine both the approaches for keeping their benefits.

In the next section, this chapter presents different types of PHM models in details which are developed based on the data-driven approach. For example, deterministic models, stochastic or probabilistic models, etc. A brief review of these models along with their advantages and disadvantages are given with examples.

1.6 Model types

In the PHM application, several models under the data-driven and stochastic approaches are used to derive a posterior distribution of the hidden and random variables from the observations to calculate the expectations associated with this distribution. However, since this is often difficult to compute, an approximation scheme must be used. Deterministic approximation models and statistical approximation models are alternative techniques to methods based on numerical sampling of time series data. This section will present the advantages and disadvantages of both to help in selecting models according to the objectives of the thesis.

1.6.1 Deterministic models

The output produced by the deterministic model is determined entirely by the parameter values and initial conditions, without considering any approximation or randomness. It uses posterior distributions that are analytically approximate. The approximated distributions are factual in convenient expressions such as Gaussian, almost never leads to accurate results (Bishop, 2006).

Recently, several researchers used deterministic models in diagnostics and prognostics-based applications such as a deterministic based prognostic method for the pyramidal tract (Rosenstock, 2017), predicting remaining useful life of bearing applying deterministic extended Kalman filter (Shen, 2019), computer-aided diagnostic systems or fault tolerance control with diagnostic results (Schuh and Lunze, 2016; Zafar and Khan, 2019), and failure or first fault diagnosis of various systems (Chen, 2019; Shao, 2019; Wang, 2019). Some authors mixed deterministic model with other techniques to improve results. (Zheng, 2015) proposes a prognostic method for lithium-ion batteries by using Bayesian approach based deterministic model. In (Bayraktaroglu and Orailoglu, 2002), the authors offer a cost-effective diagnostic method where they used a scan based deterministic model. Ma, (2010) reviewed the deterministic machine availability and Garcia gives a survey (Garcia and Frank, 1997) about deterministic nonlinear observer-based approaches. Another review (Sun, 2013) is interesting where the authors described deterministic approximate Bayesian learning. Propagation expectations (Minka, 2001) extend the assumed density filtering in the batch case, including iterative improvements to the approximate posterior. In some probabilistic models, their performance is significantly higher than that of the assumed density filtering method and some other approximate methods (Kuss, 2006; Minka, 2001). Propagation of beliefs (Pearl, 1988) provides an effective framework for the precise derivation of posterior boundary distributions in a tree structure probability graph model. It comes in various algorithmic formulations and the most advanced treatment of which is the sum-product algorithm for graphical representation of the factors (Barber, 2012; Bishop, 2006).

Advantages and disadvantages

In this section, we described the deterministic approximation models using a selection of literature, considering that it is not possible or necessary to mention all deterministic approximation techniques to date. The review gives an overall idea of the pros and cons of the deterministic model based on different applications. Deterministic models have the advantage of helping to avoid the arbitrary selection of performance and provide the necessary theoretical basis for studying the relative importance of various factors that affect the outcome results. It has done a relatively good job of identifying the necessary and sufficient conditions. The deterministic model tends to rely on a categorical dependent variable, which depends on Boolean logic to classify all cases into a single cell in a table. It is generally quick and easy to apply but it gets unwieldy in a large dataset. It does not allow for a greater variety of variables. Since deterministic models do not consider the randomness of variables, it is difficult to cover different uncertainties. Then again, if the data set happens to be stochastic then it tends toward stochastic modelling, not deterministic. When we are using data series, it is for following the evolution and predict the future valuations. Stochastic models can be a good option to overcome these limitations by making predictions from probability distributions using statistical methods. The next section details the stochastic model and its advantages over deterministic models.

1.6.2 Stochastic models

The stochastic model is such a model whose analysis focuses on a process involving a random sequence of observations, each of which is considered a sample of an element in the probability distribution. It is also known as a probability model and generally consists of three elements: deterministic parameters, variables including latent and stochastic parameters, and observable variables that jointly specify the probability distribution (Sun, 2013). However, there is an essential difference between probabilistic and stochastic models. A probabilistic model is a relatively broad concept that incorporates random variables and probability distributions into a model of an event or phenomenon. The most probable results are independent in this model where the past results do not affect current probabilities. For example, winning lottery numbers are designed to be completely independent of each other. Today’s numbers are determined by the same probability distributions as yesterday, but with no memory of past results. On the other hand, the stochastic model calculates the likelihood of the occurrence of certain events during the system execution which changes over time are described by its past plus the probability of future changes (Kwiatkowska, 2007). Statistics play an important role in this process where the frequency of past events is studied. For example, tomorrow's stock price is its current price plus an unknown change. This unknown change is usually small enough to make tomorrow's situation reasonably predictable.

Stochastic model has been used for a long time in various applications based on time series data. For example, Hikaru, (1973) applied the model to predict sulfuric oxides based on a time series data set. Finzi, (1980) used another time series model with pollutants and meteorological variables for a single point multivariate study, and Murray, (1982) performed statistical modelling of the visibility sulfate history database using time series analysis for Salt Lake City. There are also authors who have applied Markov, fuzzy, and neural network methods to model different statistical applications. For example, North, (1984) developed a Markov model based on the rising and falling phases of the daily threshold of the carbon monoxide concentration series. Raimondi, (1997) proposed an air pollution model that takes into account model uncertainty and describes the daily dynamics of the variable Dose Area Product (DAP). Drozdowicz, (1997) proposed a neural network-based model for predicting carbon monoxide concentrations in urban areas of the city of Rosario. The theory is concerned with the decision-making process regarding human health assessment.

In the last 15 years, stochastic models have been applied for different diagnostics and prognostics applications in industrial context. For example, Gebraeel, (2008) developed a predictive degradation model for calculating and updating residual lifetime distributions in time-varying environments. He, (2009) provided stochastic modelling of damage physics using state indicators for prediction of mechanical components. (Bian, 2013) described a stochastic approach for prediction in continuously changing environments. Bian also studied the stochastic modelling of real-time prognostic predictions for multi-component systems with degradation rate interactions. In (Le Son, 2013), the author provides a detailed review of the remaining lifetime estimates based on stochastic deterioration models. Mainly based on this review and the above literature, the advantages and disadvantages of the stochastic model are listed below.

Advantages and disadvantages

Advantages:

- Stochastic can take into account stochastic properties of random disturbance variables; thus, it adjusts control actions properly.
- It allows to include variables into the formula of optimizing problems.
- It can be formulated in a distributed manner and thus the computational results can be split among several solvers.
- It does not require to have expert knowledge about the system (observed data and survey information can be useful).
- It requires comparatively less data and low costly than deterministic models.

Disadvantages:

- Sometimes it relies on historical consumption data.
- Maybe render incorrect result due to the false alarms in data.
- Prevent measures that may have no impact.

1.6.3 Hybrid models

Hybrid models use both deterministic and stochastic assets to maintain their merits and avoid their limitations. It attempts to model the system through one structure with both elements in a given situation. Some works based on hybrid models in different fields are given below.

Pakniyat et al (2016) described optimal control of deterministic and stochastic hybrid systems in theory and applications. In (Alwan, 2018), the authors provide a detailed overview of the mixed systems theory of deterministic and stochastic concepts. Yang et al (2017) propose a combination of these two models for vibration analysis of uncertainty structures. Pierro et al (2017) propose a model based on both the model structures for solar power prediction. In (Popescu, 2016), Popescu et al (2016) provides a hybrid deterministic along with stochastic X-ray transmission simulation for advanced detector noise models for transmission computed tomography. A Hybrid Monte Carlo method (HMC) is presented by Shen (Shen, 2018) where the authors investigate the HMC based statistical inversion approach and suggests that it raises more efficiency in dealing with the increased complexity and uncertainty faced by the geo-steering problems.

The hybrid models can be built by combining determinism with randomness and/or some other technique depending on the problem. Although the hybrid model provides a dual nature that guarantees improved performance, it is not easy to implement. Engineers cannot rely on data alone, but also need to have a good understanding of physical systems. So, it is a complex, costly development project.

1.6.4 Conclusion

In this section, different model structures are discussed. These structures have some unique and useful features that address specific problem issues. They also have some limitations. One model cannot cover another's usability. Although hybrid models are used to maximize the effectiveness of deterministic and stochastic models, this is only for specific cases. In this thesis, we decided to practise a stochastic model structure to provide a general solution to a similar problem that follows the objectives of the thesis. It is easier to use stochastic models to deal with the uncertainties of modelling the degradation of systems (data, operating conditions, etc.) Stochastic models are an interesting fitting method for dealing with probabilities and random numbers where different methods (*e.g.* Monte Carlo, etc.) can be used to constrain the number of states in a data set. There are several stochastic models that can be found in the literature. In this book, only the popular and commonly used models are reviewed.

1.7 Stochastic models

Over the past few decades, several stochastic models have been applied to predictive and health management systems. All models offer different benefits of uses along with some disadvantages. Some alternative models with recent works are described in this section.

1.7.1 Fuzzy Logic models

Fuzzy logic is a generalization of standard logic, where the truth of a concept can be anywhere between 0.0 and 1.0. It is the fuzzy set theory proposed by Lotfi Zadeh in (Lotfi Zadeh, 1965). However, the study of fuzzy logic began in the 1920s, and in the 1960s, Dr. Lotfi Zadeh of the University of California, Berkeley, first introduced the concept of fuzzy logic as infinite value logic which is now largely developed in many fields (Pelletier, 2000). It is a popular model structure for its simplicity and flexibility. It can handle problems with imprecise and incomplete data. It uses simple mathematics for nonlinear, integrated and complex systems.

Recently, several researchers used this logic to diagnostics and prognostics applications systems. Cosme et al (2018) proposed a prognostic approach based on interacting multiple model filters and fuzzy systems. In (Jiang, 2019), the author described a novel ensemble fuzzy model for degradation prognostics of rolling element bearings. Kang researched on Remaining Useful Life Prognostics based on Fuzzy Evaluation-Gaussian Process Regression Method in (Kang, 2020). Škrjanc et al (2019) given a detailed overview in his survey which is evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification and classification.

The Fuzzy logic sometimes works with Neural Networks as it mimics how a person would make decisions, only much faster. A brief overview of Neural Networking models is given below.

1.7.2 Neural Networking models

Neural networking is another popular detection technique that can be used for similar perspectives (of using Fuzzy logic) in which it works by simulating a huge number of interconnected processing units that resemble abstract versions of neurons.

The Neural network is a model that has specialized algorithms to identify the underlying relationships in a set of data by mimicking the processes of the human brain. An artificial neural network is consisting of neurons or nodes in the modern sense of solving artificial intelligence problems. A brief overview of recent forecasting efforts to date using NN and ANN is provided below.

The preliminary theoretical base for modern neural networks was proposed by Alexander Bain et al (1873) and William James et al (1890). After that, it has been used in many applications in different fields. Recently, several PHM applications are found in the literature that has been proposed using this model structure. Li et al (2018) offered a prognostic technique by using deep convolution neural networks. The author used neural networking based deep learning method on the popular C-MAPSS dataset (Saxena, 2008) for predicting the RUL of aero-engine units accurately. Palau et al (2018) proposed a recurrent neural networking model for real-time distributed collaborative prognostics. The author demonstrates the basic implementation of real-time distributed collaborative learning, where collaboration limited to sharing trajectories to failure in real-time among clusters of similar assets. In (Khera, 2018), the author offers the ANN for prognostics of aluminum electrolytic. The training is done off-line with experimental data using the back-propagation learning algorithm. Further, the weighted ANN is used to estimate the equivalent series resistance of the system. Guo et al (2017) developed a recurrent neural network-based health indicator for remaining useful life prediction of bearings. He used a feature extraction method to map the classical time and frequency domain features with diversity ranges to some target features ranging from 0 to 1.

A couple of survey papers (Yi, 2018; Marugán, 2018) presented a detailed overview of neural network applications in the PHM domain. Marugán et al (2018) present an exhaustive review of artificial neural networks used in wind energy systems. He identified the methods most employed for different applications and demonstrates that Artificial Neural Networks can be an alternative to conventional methods in many cases. Yi, (2018) provide a brief review of the PHM for special vehicles where he highlighted the neural networking technologies behind the prognostic applications with their benefits. Recently, bidirectional Long Short-Term Memory (BiLSTM) approach for Remaining Useful Life (RUL) estimation is proposed in (Wang, 2018) which benefits of taking sequence data in bidirectional.

The neural network model is flexible in both regression and classification problems. A well-trained neural network model is quite fast at prediction. The mathematical basis behind the model allows for the processing of non-linear data along with any number of inputs and layers. However, since this model structure relies on a large amount of training data, it can lead to overfitting and generalization problems. Another important limitation is that it is a black box process. It is impossible to know how much the independent variable affects the dependent variable, or how the entire hidden layer of likelihood evolution proceeds. Therefore, in cases where the black box concept is inefficient and ineffective, the Markov model can be an alternative good choice.

1.7.3 Markov Models

The Markov Model (MM) is a stochastic tool used to model a system with random variations. It assumes that future state depends only on the current state and not on previous events (Gagniuc, 2017). Among Markov models that can be used to represent the states of an autonomous systems, Markov chains and Hidden Markov Models are well used in the PHM domain.

In Markov Model, the states are fully observable. On the other hand, in the Hidden Markov models, the states are hidden and partially observable. The HMM is another generation of Markov model which was proposed by Baum in the early 1970s (Baum, 1966) and was first used in speech recognition applications by Rabiner et al (1989). Later HMM-based applications, using actual data collected from complex systems became a very common practice in PHM. For example, (Kumar, 2018; Dong, 2007; Basia, 2019; etc.) proposed the application of HMM for the diagnostics system's health. In (Chinnam, 2003), the authors describe a technique for autonomous diagnosis and prognosis through a competitive learning-driven HMM-based clustering technique.

Several other forms of HMM have been proposed in the literature. For example, (Dong, 2007) proposed Hidden Semi-Markov models (HSMM) for a diagnostic and prognostic framework that monitors the condition of hydraulic pumps. HSMM has the same structure as HMM except that the hidden part is semi-Markov rather than Markov. The author modified the forward-backward algorithm to estimate the HMM parameters. Another application based on the Hierarchical Hidden Markov Model (HHMM) for predicting the health state of drilling rigs was proposed in (Camci, 2006). HHMM is derived from the HMM in which, each state is considered to be a self-contained probabilistic model. More precisely, each state of the HHMM is itself an HMM. Fernández et al (2018) proposed a prediction technique based on the Multilayer Hidden Markov Model (MLHMM) for diagnosing the bearing failure. MLHMM is a generalization of HMM that is tailored to accommodate longitudinal data from multiple individuals simultaneously. Some researchers have combined HMM with different tactics such as Mixture of Gaussian HMM used in (Tobon-Mejia, 2012) for bearing degradation modeling and RUL estimation. The Gaussian hybrid model and parallel calculations are combined in (Wang, 2018) for health estimation and prognosis prediction of turbofan engines. A Hierarchical Dirichlet Process-Hidden Markov Model (HDP-HMM) is described for prognostic mechanical equipment (Wang, 2019), etc. These advanced forms of HMMs, and their combinations, make better RUL predictions than traditional HMMs.

All these HMM-based applications and studies are very interesting and proven methods in PHM applications. Nevertheless, none has integrated operating conditions in their models whereas it seems clear that operating conditions influence the state dynamics. They tried different forms of HMM and mixed it with other techniques to produce better results, but these models cannot be used to integrate operating conditions because these models do not allow any input. However, in (Le, 2016), a Multi-Branch HMM (MBHMM) is proposed to consider the operating conditions for estimating RUL of systems. This is an innovative proposal, but the author did not use the operating conditions as inputs. The authors classify the observations according to the operating conditions and train different HMMs consequently then fuse them into one model. The operating conditions can be switched at any time during the operation, but no switching control is established in (Le, 2016).

However, despite the input condition, the HMM proves acceptability and applicability in PHM applications for long times. The Hidden Markov model not only allows us to observe hidden states and their likelihoods, but we can also change the values during the process, as necessary. This model structure also has a strong mathematical basis and can handle different uncertainties (data, models, etc.) well.

HMMs can be considered the simplest dynamic Bayesian network (DBN), which is an advanced class of BN. HMMs have been shown to produce solutions equivalent to DBN. In this book, MM, HMM, and other versions of HMM are usually represented in a DBN form.

1.7.4 Conclusion

As part of predictive maintenance, information on the current health state of systems and its projection into the future are the main support for orienting the service of maintenance. This information can allow be used to make maintenance decisions through diagnostic and prognostic processes. Therefore, we have chosen to focus our work on the design of an approach that characterizes diagnostic-prognostic coupling degradation. This chapter has detailed the issues related to PHM approaches and models that highlight the question of optimizing diagnostics and prognostics activities.

By detailing the existing PHM approaches, the reason for selecting a data-driven approach is given which leads to the development of predictive maintenance policies and presented the advantages that these policies can bring to industrialists. We have precisely defined diagnostics, prognostics, RUL, and the degradation complexity related to the operating conditions and data uncertainties. After reviewing the advantages and disadvantages of different approaches the data-driven is chosen because of its ability and scope of handling complex systems under different uncertainties.

After that, different types of model are reviewed to select the delicate model type since the data-driven offers several *i.e.* deterministic, stochastic, etc. These types have their own benefits and limitations due to the specific problem issues. Since the goal of this thesis is to propose a solution to the PHM society that concerns randomness and explicit assumptions, a stochastic model type is being preferred instead of a deterministic model. A stochastic model allows the assumptions to be tested by a variety of techniques.

There are many stochastic models (*e.g*. neural network, Markov models, etc.) that are practiced in the PHM applications. Sometimes, fuzzy logic works with neural networks, but they rely on a large amount of training data which can be caused for overfitting problems. Observing internal factors is also hard to follow in these models. Especially the neural network, which is a black-box process that does not allow us to see how the hidden layer evaluates in time. However, our case study includes not only observe the inside likelihood evaluation but also modify the value at any time instants during the process. HMM is fairly fitting to our case study which can be trained by comparably less amount of data than the other models. HMM has a solid statistical foundation and efficient learning algorithms. It allows for consistent handling of insertion and deletion penalties in the form of locally teachable methods.

We have exposed several versions of HMM in this chapter (HMM, HSMM, MBHMM, HHMM, etc.). All versions are dedicated to specific cases of problems. Yet, none of them has taken into account the operating conditions as input. However, because of the property of HMMs, it is also impossible to consider inputs into the model. The state of the HMM explains the level of health and the dynamic of its evolution. As the HMM is unique, the dynamic is also unique and is not influenced by any condition. Hence, considering operating conditions as inputs is not possible by HMM. Therefore, we proposed the Input-Output Hidden Markov Model (IOHMM) which is more general version of HMM. The specificity of IOHMM is that it allows an input. So, the operating conditions can be introduced into the model. This model is defined in (Bengio, 1995) where the author explains the scope and the ability of this model and its strong relation with ANN. Since then, it has been used in several fields (Just, 2004; Hu, 2015) but, as far as we know, it has not been applied to the PHM field before our proposal. Moreover, the learning of model parameters has not been completely solved for each inputs and outputs structure.

IOHMM takes the operating conditions as inputs and switches the model at the given input sequence at each time instant. Therefore, IOHMM deals better with the time series problem of long-term dependencies than standard HMMs (Bengio, 1995). It has a faster training process that uses the entire dataset along with the operating conditions to learn the models in one go. No data classification is required because the proposed method switches the operating conditions corresponding to the given dataset during the training session. This is a time-consuming approach and more realistic compared to MBHMM. That is why IOHMM assesses more practical degradation and prognostic that close to reality. The background of IOHMM is described in the next chapter.

Chapter 2

Background of the Model from MC to IOHMM

Table of Content

2 Background of the Model from MC to IOHMM

As we conclude in the previous chapter, data-driven models are chosen for the purpose of diagnostic-prognostic systems, mainly because the expert knowledge required to build this model is less important than for physical models. Moreover, this model allows computing online diagnostics from system behaviour by consuming all data available.

In this context of data-driven models, stochastic models are specifically of interest because they can handle our inability to define a complex problem such as system health evolution or system state degradation. Nevertheless, by using these models, it relies on an efficient algorithm to estimate the model parameters. Hidden Markov Models use some well-known algorithm for training a model from a sequence of data but does not take into consideration operating conditions. As each probabilistic structure e.g. Markov Chain, HMM ... requires an adapted algorithm, then modifying the probabilistic structure needs to adapt the algorithms to be employed in learning, diagnosing, and prognostic the health state of the modelled system. These algorithms enter the scope of Machine Learning.

2.1 Markov Chain notations

Markov Chain (MC) gives the probability of sequences of random states, each of which can take values from a given set. It assumes future states based on the current state of matters. The states before the current one has no influence on the future, except through the present state (Keselj, 2009). Let's assume a system being assumed as in one of the states, $\{s_1, s_1, \ldots, s_N\}$, N is the number of states. We denote the time instants associated with the state transitions as $(X_1, X_2, \ldots, X_K)$, where X_1 holds a state at the first time-instant and X_K holds a state at the last time instant. If the current time instant defined as k where $1 \leq k \leq K$ then the current transition probability would be:

$$P(X_k = s_j | X_{k-1} = s_i),\ 1 \leq i, j \leq N$$

with the state transition properties of $\sum_j^N a_{ij} = 1$, a_{ij} represents the transition probabilities from state s_i to s_j. The initial state distribution is defined as $\pi = P(X_1 = s_i)$.

An MC with two states and the transitions are shown in Fig. 5.

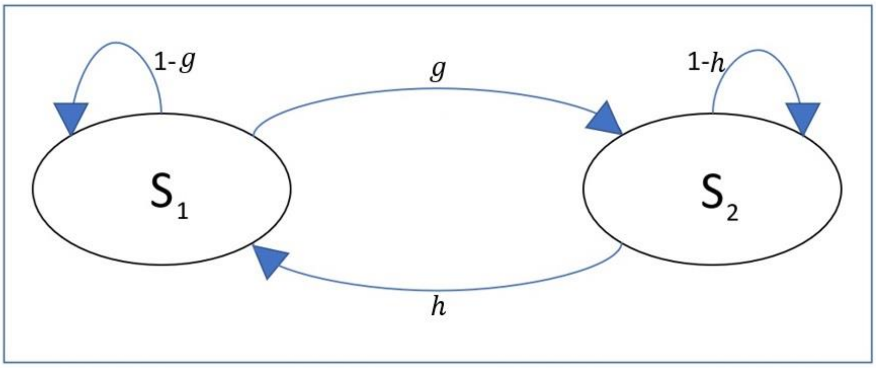


Fig. 5: Two-state Markov chain $\{s_1, s_2\}$

The MC assigns a probability to a sequence of health states of systems. The healthy state is defined as "s_1", and degraded state is defined as "s_2" which is the final state of this model. The states are represented as nodes and the probability transitions as edges. The transition probabilities of a state must sum to 1 as it represents the transition matrix. A transition matrix, also called stochastic matrix, probability matrix, substitution matrix, or Markov matrix, is a square matrix used to characterize transitions for a finite Markov chain. The elements of the matrix must be real numbers in the closed range [0, 1]. Each of the rows represents the transitions from a state to other states along with itself. That is why the sum of each row is 1.

According to the Fig.5, the transition matrix A of the model is:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 1-g & g \\ h & 1-h \end{pmatrix}$$

In summary, the basic Markov model is a state diagram with transition probabilities. At each time step, the model undergoes a transition that changes its state so that the modelling system follows a state evolution pattern.

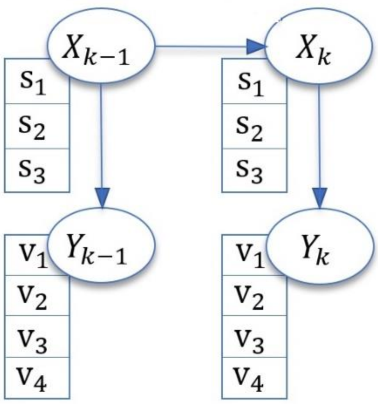
A Markov model is specified by the following components:

$X = (X_1, X_2, \ldots, X_K)$	The state sequence, each one drawn from the variable $S = s_1, s_2, \ldots, s_N$; N is number of hidden states
$A = (a_{11} a_{12} \ldots a_{n1} \ldots a_{nn})$	The transition probability matrix, each a_{ij} representing the probability of transiting from state *i* to state *j*, s.t. $\sum_j^N a_{ij} = 1 \;\; \forall i$
$\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$	The initial probability distribution over states. π is the probability that the Markov chain will start in state *i*. Some states *j* may have $\pi_i = 0$, meaning that they cannot be initial states. Also $\sum_i^N \pi_i = 1$

2.2 Hidden Markov Model

The systems generally produce observable emissions that can be characterized by signals (temperature, vibrations, sound signals, etc.). In the last decades, research in artificial intelligence has focused on how to characterize such signals. Among the many methods for modelling such real phenomena, HMMs have proven to be particularly effective. It is a Markov chain in which the states are no longer directly observable. That is why it called the hidden states which can be observed by the observations. The hidden states and the observations are linked to each other in a probabilistic way. The Hidden Markov Model considers observation data where the probability distribution of the observed symbol depends on the underlying state.

The HMM is showed in Fig. 6 is the simplest two times dynamic Bayesian network.



Variable X is state sequence, each one drawn from the variable $S = \{s_1, s_2, s_3\}$

Variable Y is observation sequence, each one drawn from the emitted symbol $V = \{v_1, v_2, v_3, v_4\}$

Variable k is the time instant.

Fig. 6: Three state HMM with four observation symbols

2.2.1 HMM Structure

2.2.1.1 Definitions

Initial probabilities

It is the probability of being in a state at the beginning ($k = 1$) is given by $\pi = P(X_1 = s_i)$ with $1 \leq i \leq N$.

Transition probabilities

It is the probability of transiting from one state to the other states.

The states are hidden where each one of them can be drawn from the variable $S = \{s_1, s_2, s_3, \cdots, s_N\}$. HMM evolves in a sequence of states $X = (X_1, X_2, \ldots, X_K)$ where each takes value from S. $A = (a_{ij})$ denotes the state transition probability matrix where $a_{ij} = P(X_k = s_j | X_{k-1} = s_i)$ is the transition probability from state $X_{k-1} = s_i$ to state $X_k = s_j$, $1 \leq i, j \leq N$ and $k \in \mathbb{N}$ is a strict positive integer and represents a discrete time instant. a_{ij} represents the probability of all the transitions from state i to state j, so the summation of a_{ij} for each state j is 1.

The dimension of the transition matrix is N by N.

Emission probabilities

It is the probability of observed emission Y_k given the state X_k.

Let us assume the hidden states emit a total of M possible symbols as $V = \{v_1, v_2, v_3, \cdots, v_M\}$. The observation sequence $Y = (Y_1, Y_2, \ldots, Y_K)$ with the same length as the state sequence where each time instant the variable contains one of the symbols from V. The variable $B = (b_{jm})$ denotes the state emission probability matrix, where $b_{jm} = P(Y_k = v_m | X_k = s_j)$ is the emission probabilities of state $X_k = s_j$ with $1 \leq m \leq M$. b_{jm} represents the probability of all possible emissions of output state s_j, so the summation of b_{jm} for each state j is 1.

The dimension of the emission matrix is N by M.

According to the Fig. 6, the transition matrix B of the HMM is:

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix}$$

Now, if we denote the HMM model by Λ, then the triplet $\Lambda = (A, B, \pi)$ completely defines the model.

2.2.1.2 Absorbent state

It is such a state which does not have any transition paths to other states but itself. Once the model reaches this state, it cannot come out of that, it stays in the state forever. An HMM can have more than one absorbent state but, in this book, the model considers only one absorbent state which is called the final state or the breakdown state. In Fig. 7, the node s_3 represents the final state of an HMM.

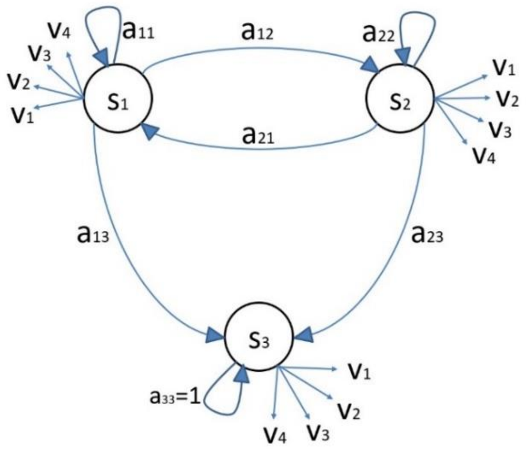


Fig. 7: HMM with one final state

The transition matrix of the HMM showed in Fig. 7 is:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

2.2.1.3 Left-right model

The left-right model is a specific type of HMM where there are no transitions from a higher indexed state to a lower indexed state. That means there is no back transitions. It also called the Bakis model (Yuan, 2018). The degradation process of a system always evolves towards bad states. By means of which, if a system goes from any state s_i to another state s_j where $i <= j$, then it cannot go back to the previous state s_i. The transition will only happen when from left to right graphically.

The corresponding HMM can be presented as Fig. 8.

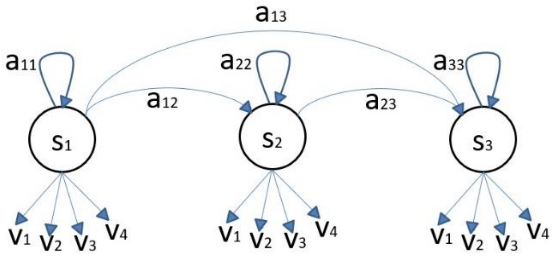


Fig. 8: Left-right model HMM model

In this model, the state transit to the next states and itself. For example, the transitions from state s_2. There are two transitions that happened from this state, s_2 to s_3 which is defined as a_{23}, and to itself which is defined as a_{22}, but there is no transition from s_2 to s_1 neither from s_3 to s_2 or s_1.

The transition matrix for this HMM is:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix}$$

2.2.1.4 HMM components

The HMM is specified by the following components:

K	The length of the sequence
$X = (X_1, X_2, \ldots, X_K)$	The state sequence
$S = \{s_1, s_2, \ldots, s_N\}$	The set of hidden states
N	The number of hidden states
$A = (a_{11}a_{12} \ldots a_{n1} \ldots a_{nn})$	The transition probability matrix, a_{ij} represents the probability of transiting from state i to state j, s.t. $\sum_j^N a_{ij} = 1 \ \forall i$
$Y = (Y_1, Y_2, \ldots, Y_K)$	The observation sequence
$V = \{v_1, v_2, \ldots, v_M\}$	The set of observation symbols
M	The number of observation symbols
$B = b_{jm}$	The sequence of observation likelihoods which is also called emission probabilities, each expressing the probability of an observation Y_k being generated from a state j, $\sum_m^M b_{jm} = 1 \ \forall m$

$\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$	The initial probability distribution over states. π is the probability that the Markov chain will start in state *i*. Some states j may have $\pi_i = 0$, meaning that they cannot be initial states. Also $\sum_i^N \pi_i = 1$

2.2.1.5 *Three basic problems of HMM*

Given such a hidden Markov model $\Lambda = (A, B, \pi)$ where the observation sequence is Y and state sequence is X, HMM can be used to solve three types of problems:

1) **The learning problem**: Learn the parameters of the model $\Lambda = (A, B, \pi)$ from the observation sequences. The problem is how to adjust the HMM parameters, so the given observation set is represented by the model in the best way. The Baum Welch algorithm which is a class of *Expectation Maximization* (EM) algorithm can be used to solve the learning problem.

2) **The evaluation problem**: It is also called the likelihood problem. The probability to emitting an observation sequence Y given the model $\Lambda = (A, B, \pi)$ i.e. $P(Y|\Lambda)$. A sim0ple probabilistic argument can be used as a solution, but the computation complexity, in this case, is big (orders KN^K). That is why the *forward-backward* (FB) algorithm is being used in this book to reduce the complexity as KN^2.

3) **The decoding problem**: The most likely sequence of hidden states $P(X)$ which generated the observation sequence Y. This solution depends on the way of how the "most likely state sequence" is defined. One approach can be to find the most likely state X_k at time k and to concatenate all such 'X_k's, but sometimes it does not provide a physically meaningful state sequence. Therefore, the *Viterbi* (Vt) algorithm is an alternative option to using which overcomes such a problem and finds the whole state sequence with maximum likelihood.

The mathematical foundation of the algorithms (EM, FB, Vt) is given below.

2.2.2 The Forward-backward (FB) algorithm

Forward and backward algorithms are widely used in HMM problems. They can efficiently compute the probability of a sequence being generated by an HMM. Therefore, they assume that the model $\Lambda = (A, B, \pi)$ is known. If the observed sequence of variables Y is given, then the algorithm can calculate $P(X|Y)$ according to the following recursion using the forward-backward algorithm:

Given the transition probabilities $A = P(X_k|X_{k-1})$, the emission probabilities $B = P(Y_k|X_k)$, and the initial distribution $\pi = P(X_1 = s_i)$, the forward algorithm can be derived as a function of X_k where $P(X_k|Y)$ is proportional to the joint distribution of $P(X, Y)$.

$$P(X_k|Y) \propto P(X, Y)$$
$$P(X_k|Y) = P(X_k, Y_{1:k})P(Y_{k+1:K}|X_k, Y_{1:k})$$

If $Y_{k+1:K}$ is conditionally independent on $Y_{1:k}$:

$$P(X_k|Y) = P(X_k, Y_{1:k})P(Y_{k+1:K}|X_k)$$

Finally, a recursion has formalized for both forward and backward processes to reduce the computational complexity of the algorithm:

2.2.2.1 *Forward auxiliary variable*

$$\alpha(X_k) = P(X_k, Y_{1:k})$$

where $P(X_k, Y_{1:k})$ is a joint probability of observation Y is from time instant 1 to k and hidden state X is at time instant k, given the model $\Lambda = (A, B, \pi)$.

Computational structure:

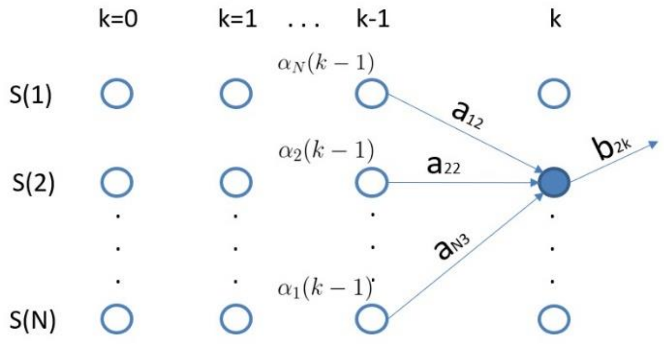


Fig. 9: Forward computation

According to the forward structure shown in Fig. 9, the variable X_{k-1} can be introduced into the equation as:

$$P(X_k, Y_{1:k}) = \sum_{X_{k-1}=s_1}^{s_N} P(X_k, X_{k-1}, Y_{1:k})$$

After factorizing the equation, it becomes:

$$P(X_k, Y_{1:k}) = \sum_{X_{k-1}=s_1}^{s_N} P(Y_k|X_k, X_{k-1}, Y_{1:k-1})P(X_k|X_{k-1}, Y_{1:k-1})P(X_{k-1}, Y_{1:k-1})$$

Applying Markovian properties as if Y_k is conditionally independent of both X_{k-1} and $Y_{1:k-1}$, then given X_k the equation would be:

$$P(X_k, Y_{1:k}) = \sum_{X_{k-1}=s_1}^{s_N} P(Y_k|X_k)P(X_k|X_{k-1}, Y_{1:k-1})P(X_{k-1}, Y_{1:k-1})$$

Similarly, if X_k is conditionally independent on $Y_{1:k-1}$, given X_{k-1}, then:

$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1})P(Y_k|X_k)P(X_k|X_{k-1})$, followed: $\alpha(X_k) = P(X_k, Y_{1:k})$

- Basis: at time step $k = 1$

$$\alpha(X_1) = P(Y_1|X_1)P(X_1)$$

- Recursion: when $2 \leq k \leq K$

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1})P(X_k|X_{k-1})P(Y_k|X_k) \quad (1)$$

2.2.2.2 Backward auxiliary variable

The backward algorithm is similar to the forward algorithm except it starts from the last time instant and calculates in reverse.

In backward computation, the probability variables use a different indexing following the transition probabilities as $A = P(X_{k+1}|X_k)$, the emission probabilities $B = P(Y_{k+1}|X_{k+1})$, and the initial distribution $\pi = P(X_1 = s_i = 1), \forall i$ the backward algorithm can be derived as a function of X_k where $P(X_k|Y)$ is proportional to the joint distribution of $P(X, Y)$.

$$\beta(X_k) = P(Y_{k+1:K}|X_k)$$

where $P(Y_{k+1:K}|X_k)$ is the probability of observation $Y_{k+1:K}$ from time $k + 1$ to K given the hidden state X at time k and the model $\Lambda = (A, B, \pi)$.

Computational structure:

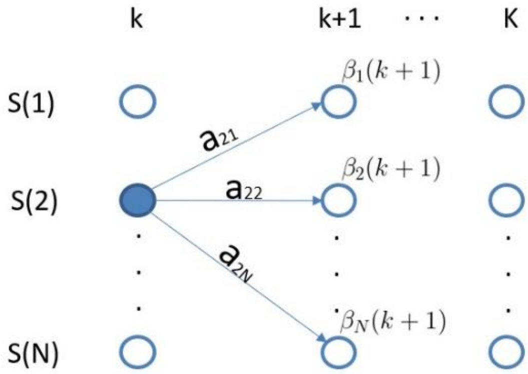


Fig. 10: Backward computation

According to the forward structure shown in Fig. 10, the variable X_{k+1} can be introduced into the equation as:

$$P(Y_{k+1:K}|X_k) = \sum_{X_{k+1}=s_1}^{s_N} P(Y_{k+1:K}, X_{k+1}|X_k)$$

After factorizing the equation:

$$P(Y_{k+1:K}|X_k) = \sum_{X_{k+1}=s_1}^{s_N} P(Y_{k+2:K}|Y_{k+1}, X_{k+1}, X_k) P(Y_{k+1}|X_{k+1}, X_k) P(X_{k+1}|X_k)$$

By applying the Markovian properties, it comes:

If Y_{k+2} is conditionally independent on X_k and Y_{k+1}, given X_{k+1}, then:

$$P(Y_{k+1:K}|X_k) = \sum_{X_{k+1}=s_1}^{s_N} P(Y_{k+2:K}|X_{k+1}) P(Y_{k+1}|X_{k+1}, X_k) P(X_{k+1}|X_k)$$

Similarly, if Y_{k+1} is conditionally independent on X_k, given X_{k+1}, then:

$$\beta(X_k) = \sum_{X_{k+1}=s_1}^{s_N} \beta(X_{k+1}) P(Y_{k+1}|X_{k+1}) P(X_{k+1}|X_k)$$

followed: $\beta(X_k) = P(Y_{k+1:K}|X_k)$.

- Basis: at time step K

$$\beta(X_K = s_i) = \begin{pmatrix} 1 \\ 1 \\ .. \\ 1 \end{pmatrix}$$

where $\beta(X_K = s_i)$ is a column vector having all hidden state distribution as 1 for $1 \le i \le N$.

- Recursion: when $(K-1) \le k \le 1$

$$\beta(X_k) = \sum_{X_{k+1}=s_1}^{s_N} \beta(X_{k+1}) P(Y_{k+1}|X_{k+1}) P(X_{k+1}|X_k) \quad (2)$$

On the basis of this forward-backward algorithm, we can use Eq. 3 to calculate the probability that the HMM will generate a sequence of observations.

$$P(Y_k|\Lambda) = \sum_{i=1}^{N} \alpha_i(k)\, \beta_i(k)\,; \forall k \quad (3)$$

2.2.3 The Baum Welch algorithm

The Baum Welch algorithm first described in the late 1960s by Lloyd R. Welch and Leonard E. Baum [Baum 1960]. However, it is used in the 1980s for the first time in speech recognition. One of the problems of HMM is to determine that $\Lambda = (A, B, \pi)$ knows the sequence of observations *Y*. It is a search for which parameters to maximize P(*Y*| Λ). In this case, the Baum-Welch algorithm is used. It is a dynamic programming type of expectation-maximization algorithm. The expectation step computes the expected state occupancy count and the expected state transition count based on current probabilities of *A* and *B*. The maximization step uses the expected counts from E-step and update the probabilities of *A* and *B*. It can eventually converge to a local minimum.

The EM algorithm uses the FB algorithm to solve this problem in an iterative way. It starts with an initial probability of the parameters and adjusts the parameters iteratively.

The maximization problem is algorithmically complex. Using the previous algorithm, we use Eq. 4 to calculate the probability of HMM *Λ* generating all *Y* sequences.

$$P(Y_k|\Lambda) = \sum_{i=1}^{N} \alpha_i(X_k)\, \beta_i(X_k)\,; \forall k \tag{4}$$

The probability of being in state i at time k and in j at time $k+1$ given the observed sequence Y and the parameters of HMM *Λ*:

$$\varepsilon_k(i,j) = \frac{\alpha_i(X_k).\, a_{ij}.\, b_{jk}.\, \beta_j(X_{k+1})}{P(Y|\Lambda)} \tag{5}$$

The auxiliary variable $\varepsilon_k(i,j)$ is defined by Eq. 5, where i, j represents the hidden state and emission symbols, respectively.

Finally, $\varepsilon_k(i,j)$ is being used to update the parameters according to the following algorithm.

- Initial state probability:

$$\pi_i = \varepsilon_1(i,j), \text{ where } 1 \le i \le N$$

- Transition probability:

$$a_{ij} = \frac{\sum_{k=1}^{K-1} \varepsilon_k(i,j)}{\sum_{k=1}^{K-1} \sum_{j=1}^{N} \varepsilon_k(i,j)} \tag{6}$$

- Emission probability:

$$b_{jk} = \frac{\sum_{k=1}^{K} \varepsilon_k(i,j)}{\sum_{k=1}^{K} \sum_{j=1}^{N} \varepsilon_k(i,j)} \tag{7}$$

Repeat these steps until the changes between the two results converge.

2.2.4 The Viterbi algorithm

The Viterbi algorithm is also a well-known dynamic programming algorithm named after Andrew Viterbi, who proposed it in 1967 as a decoding algorithm for the first time. The algorithm computes $P\ (X_{1:k}|Y_{1:k})$, the maximum likelihood state sequence $X_{1:k}$ from the given observations $Y_{1:k}$ which is useful for determining the most probable system state. It is given by the most probable value of $P(X_k|Y_{1:k})$ at the current time k. Nevertheless, the Viterbi algorithm computes the maximum likelihood path of $P(X_{1:k}|Y_{1:k})$, which contains the probability of $P(X_k|Y_{1:k})$ as well. That is why this algorithm is chosen for diagnosing. There is a strong relationship between these two probabilities. This algorithm computes the value of $P(X_{1:k}, Y_{1:k})$ from the maximum values of the previous state distribution, the state transition, and the maximum emission of the current state. Since the computation is done by all the maximum values, it gives the maximum distribution at the last time instant of the path as well which represents the current health states of the system (level of degradation). The maximum probability in the distribution represents the most probable health condition at the current time.

The mathematical foundation is given below:

We know that $P(X|Y) \propto P(X,Y)$

$$max_{(X_{1:k})}\, P\,(X|Y) = max_{(X_{1:k})}\, P\,(X,Y)$$

- basis: $\omega(X_1) = P(X_1, Y_1)$

maximizations of a single number: $\omega(X_2) = max_{(X_1)}\, P(X_{1:2}, Y_{1:2})$

maximizations of the recursive: $\omega(X_k) = max_{(X_{1:k-1})}\, P(X_{1:k}, Y_{1:k})$

based on the conditional independency:

$$max_{(X_{1:k-1})} P(X_{1:k}, Y_{1:k}) = max_{(X_{1:k-1})}\, P(Y_k|X_k)P(X_k|X_{k-1})P(X_{1:k-1}, Y_{1:k-1})$$
$$= max_{(X_{k-1})}\, P(Y_k|X_k)P(X_k|X_{k-1}) max_{(X_{1:k-2})}\omega(X_{k-1})$$

- So, the recursion:

$$\omega(X_k) = max_{(X_{k-1})}\, P(Y_k|X_k)P(X_k|X_{k-1}) max_{(X_{1:k-2})}\omega(X_{k-1}) \tag{8}$$

This algorithm finds N paths starting from each of the initial states. Finally, it gives the max path that holds the maximum distribution of the last state.

As can be seen that the Viterbi algorithm along with the Baum Welch and the forward-backward algorithms are dedicated to HMM where the input is not considered. These algorithms provide solutions in terms of the HMM, not the IOHMM. Only one variable ($P(X_k|X_{k-1})$ or $P(X_{k+1}|X_k)$) is considered where there is no conditioning on the input variables, which requires ad hoc modifications for application to the IOHMM.

2.3 Input-Output Hidden Markov Model

A system can have several input conditions which cannot be modeled by the classic HMM. That is why the IOHMM is being selected as the modeling tool in this book because it allows the model to consider the inputs. The IOHMM is an advanced version of HMM which allows the model to consider inputs.

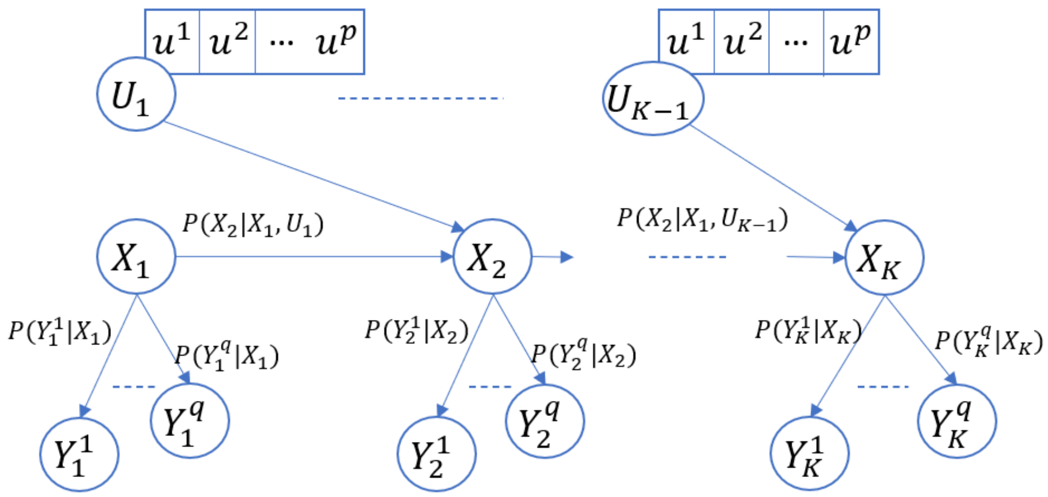


Fig. 11: Input output Hidden Markov Model

The variables X are hidden states sequence where each one drawn from the states as $S = \{s_1, s_2, \ldots, s_N\}$;. The variables Y is the sequence of observations where each one drawn from the emitted symbol as $V = \{v_{1_Y}, v_{2_Y}, \ldots, v_{M_Y}\}$. The variable U is the input sequence that contains the ids as $U = \{u^1 u^2 \ldots u^p\}$ of the input conditions.

The IOHMM provides multiple transitions matrices corresponding to the number of operating conditions presented as p by the Fig. 11. In our hypothesis, the inputs are currently considered independent to each other. Therefore, multiple inputs with several modes can be manage only one variable U which holds the index of the operating conditions and selects only one among them at each time instant for the transition from state i to j. So, the transition probability becomes conditioned by U as $(P(X_k|X_{k-1}, U_{k-1})$.

Multiple outputs can also be considered by this model in which it provides multiple emission matrices corresponding to the number of outputs presented as q. The outputs are also considered to be independent in this thesis. So, the model computes the emission probability as $P(Y_k^q|X_k)$ for $1 \leq q \leq Q$.

An input-output hidden Markov model is specified by the following components:

K	The length of the sequence
$U = \{u^1 u^2 \ldots u^p\}$	The input sequences.
$X = (X_1, X_2, \ldots, X_K)$	The state sequence
$S = \{s_1, s_2, \ldots, s_N\}$	The set of hidden states
N	The number of hidden states
$A^p = (a_{11}^p a_{12}^p \ldots a_{n1}^p \ldots a_{nn}^p)$	The transition probability matrix, a_{ij}^p representing the probability of moving from state i to state j, s.t. $\sum_j^N a_{ij}^p = 1 \ \forall i$ and p fixed; p is the index of transition matrices
P	The number of transition matrices
$Y^q = (Y_1^q, Y_2^q, \ldots, Y_K^q)$	The observation sequence, each one drawn from the emitted symbols $V = v_1, v_2, \ldots, v_M$; The symbol set is dedicated to the output so each output has its one symbol set whose size can be different?
Q	The number of emitted outputs
$V = \{v_{1_Y}, v_{2_Y}, \ldots, v_{M_Y}\}$	The set of observation symbols
M_Y	The number of observation symbols of output Y.
$B^q = b_{jk}^q$	The sequence of observation likelihoods which is also called emission probabilities, each expressing the probability of an observation Y_k being generated from a state j;
$\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$	The initial probability distribution states. π is the probability that the Markov chain will start in state i. Some states j may have $\pi_i = 0$, meaning that they cannot be initial states. Also $\sum_i^N \pi_i = 1$

2.4 Conclusion

This chapter discusses the basic background of the model. It explains the evolution of the model from MC to HMM and then to IOHMM with several examples. Three algorithms (forward-backward, Baum Welch, Viterbi) are derived to solve three problems of HMM. However, the goal is to solve these problems through IOHMM by considering the input conditions into the model. So, in the next chapter, these algorithms are adapted from the IOHMM perspective and applied to prognostic applications.

Chapter 3

The First Contribution: Learning Model Parameters

Table of Contents

3 The First Contribution: Learning Model Parameters

Three major contributions of the thesis are addressed in this book. The first contribution concerns the **IOHMM parameter learning** which covers the system designing along with algorithms adaptations and different training constraints.

The IOHMM represents a system degradation process that degrades considering multiple operating conditions. The model parameters should be learned from multiple outputs observations on which the degradation process has some effect. As the IOHMM is an extended version of HMM model, the proposed learning method is based on the well-known Baum-Welch and forward-backward algorithms. There are several important issues of model training explained in this chapter. A numerical application is made at the end to show and discuss the performance of the proposed method. Based on the key issues given in the introduction chapter, I subdivided them into several questions:

- IOHMM learning algorithms adaptation concerning:
  - Multiple operating conditions as multiple inputs
  - Multiple outputs
  - Multiple sequences
- Handling uncertainties in model training:
  - Numerical problems
  - The number of hidden states
  - Missing data
  - Small dataset
- Numerical applications
  - Modeling under multiple operating conditions
  - Modeling under missing data
  - Modeling by using the bootstrap method which is useful to provide confidence over the parameter estimation and give a reasonable result for small datasets.

One of the major contributions is adapting the learning algorithms from HMM to IOHMM by introducing some new variables into the formula (Eq.1 to Eq.8) to consider the operating conditions as multiple inputs and the multiple emitted outputs.

3.1 The learning algorithms adaptation

As the Baum-Welch algorithm uses the Forward-Backward as a product, it is necessary to first adapted this product to consider the different issues.

3.1.1 Multiple input conditions

Let us consider an IOHMM with one input with several conditions (modes) and only one output and let us develop the improvement of the forward algorithm first then on the backward algorithm.

Forward algorithm with inputs

Let us recall the general formula of forward algorithm which is dedicated to HMM is represented as the auxiliary variable $\alpha(X_k)$.

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k|X_{k-1}) P(Y_k|X_k)$$

In this equation, $P(X_k|X_{k-1})$ is the transition probability from $the\ state\ X_{k-1}$ to $state\ X_k$ at time instant k. $P(Y_k|X_k)$ stands as the probability of emitting the observation Y_k given the state X_k at the same time k. This formula does not consider the input, so a new variable needs to be added here representing the input. This adaptation is proposed by Bengio in (Bengio et Frasconi, 1995) where he explained the corresponding forward and backward algorithm considering the input. There are two limitations of this version of adaptation: (1) the author did not give any explanation about considering multiple observation sequences for the outputs, and (2) this algorithm cannot consider multiple outputs which is one of the major issues of our hypothesis. However, inspired by Bengio's work, our contribution and hypothesis are given in the following explanations.

If a system has multiple operating conditions, then it degrades in different dynamics. That means the state transition happens according to the operating conditions. A new variable U is introduced into the equation to represent multiple inputs as given by Fig. 12.

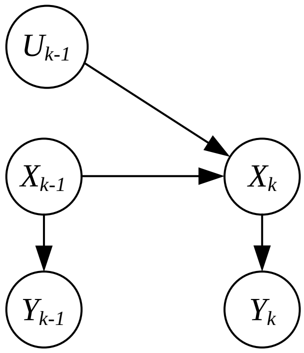


Fig. 12: 2TBN representation of 1 input-1 output IOHMM

The transition from X_{k-1} to X_k jointly depends on both X_{k-1} and U_{k-1}, written as $P(X_k|X_{k-1}, U_{k-1})$. The variable U representing the state of the input condition, selects an identity number of operating conditions following the given input sequence at each time instant.

Forward algorithm with inputs

So, the new version of the forward formula considers the input which is represented by Eq.9:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k|X_{k-1}, U_{k-1}) P(Y_k|X_k) \tag{9}$$

Backward algorithm
Similarly to the forward algorithm, the backward algorithm mainly based on an auxiliary variable $\beta(X_k)$ should also consider the transition probability with a joint probability of X_k and U_k, written as $P(X_{k+1}|X_k, U_k)$.
So, based on Eq. 2, the new version of backward formula presented by Eq.10:

$$\beta(X_k) = \sum_{X_{k+1}=s_1}^{s_N} \beta(X_{k+1}) P(X_{k+1}|X_k, U_k) P(Y_{k+1}|X_{k+1}) \quad (10)$$

The input allows to switch the operating conditions between one-another following the given input sequence. For example, if a system has 2 operating conditions, then U_k indicates the respective model for the given operating condition at any time instant k for the next transition. The variable U_k contains the ids (*i.e.* u_1, u_2) of the models (Models 01 and 02) see in Fig. 13.

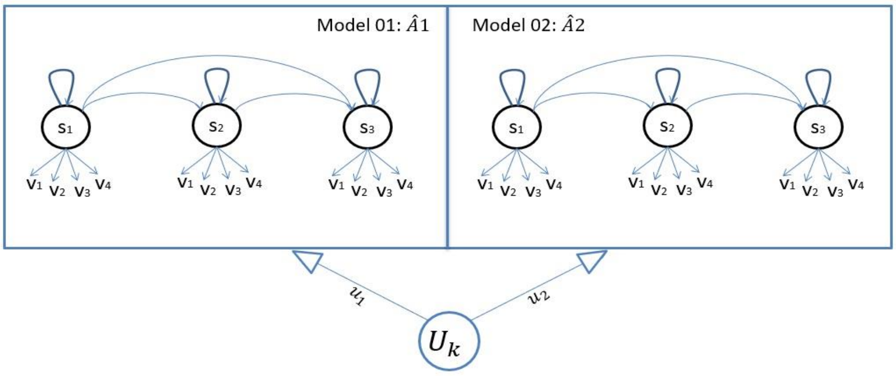


Fig. 13: Switching between two operating conditions
[u_1 and u_2 are the identity of operating conditions 1 and 2]

3.1.2 Multiple inputs case

The multiple inputs case means that several inputs influence the transition between hidden states. These input conditions model the operating and operational conditions. The operating conditions refer to the control inputs whereas the operational conditions refer to the environmental conditions.

Considering these input conditions, the case of independence between conditions can be considered as shown in Fig. 14:

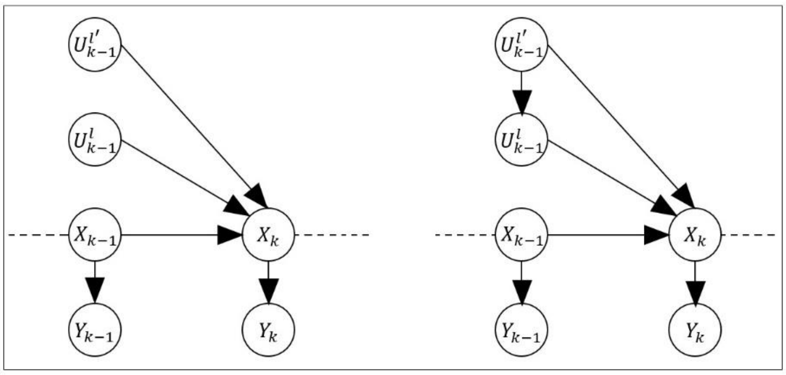


Fig. 14: DBN representing multiple inputs IOHMM with independent inputs (left) and dependent inputs (right)

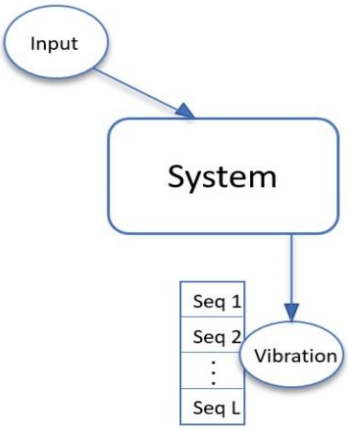
In many industrial cases, the control inputs are considered as independent and already known. So, this assumption is made to develop the consideration of inputs into Eq. 2 and 3. If the input conditions U^1 to U^l are considered independent with their own number of states then the DBN representation of Fig. 14 (left) can be reduced to Fig. 12 by considering only one input conditions whose states is the cartesian products of the states of all inputs. Therefore, the number of states is increased exponentially which induces the increase of the number of transition matrices accordingly.

In this thesis, the inputs are assumed independent. However, as future work the dependence between input conditions can be considered.

In conclusion, the solution of the multiple inputs case is given by Eq. 9 and 10 by considering that a fictive input replaces all the inputs considering the cartesian product of all states. In all the rest of the book, only one input is considered.

3.1.3 Multiple sequences case

Systems can be monitored by observing the emitted outputs of the system. A sensor can be installed on a system to observe an output that provides a single data sequence which is just one representation of the system's degradation. Likewise, multiple sensors can be used to monitor similar systems for collecting multiple sequences. Multiple sensors provide multiple sequences of observations that make a confident statement of the system's degradation. Finally, all the sequences are used to model the system or similar kind of systems (Fig. 15).



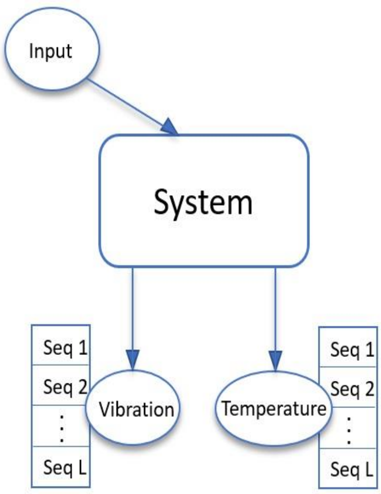
here L is the number of sensors that produce L number of observation sequences of vibration reading.

The length can be different for each sequence.

Fig. 15: Multiple sequence of vibration readings

3.1.4 Multiple outputs cases

Alternatively, the better option is to observe multiple outputs (*i.e.* vibration, temperature, speed, etc.) simultaneously for a better system-modelling. In this case, each of the outputs produces multiple observation sequences. Let us assume, this time the system is monitored by observing its vibration and the temperature both (Fig. 16). So, two sets of observation sequences can be used to model the same system instead of one output show in Fig. 15. In this book, the outputs are considered as independents. However, as future work the dependence between the outputs, and with inputs can be considered.



here L number of sensors are used to observe both the outputs which have the same number of sequences.

The length of the conjugative sequences for the outputs should be the same. For example, Seq_i for both outputs have the same length for $1 \leq i \leq L$.

Fig. 16: Multiple outputs case

In the case of multiple outputs, the emission probability $P(Y^q|X)$ can be used to represent the distribution for observing the output q. So, the forward algorithm $\alpha(X_k)$ can be written as following:

Let's consider the case of two independent outputs (Y1 and Y2). So, the Eq. 2 becomes:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k|X_{k-1}, U_{k-1})\, P(Y_k^1|X_k) P(Y_k^2|X_k)$$

If we consider Q number outputs, then it becomes:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k|X_{k-1}, U_{k-1})\, P(Y_k^1|X_k) P(Y_k^2|X_k) \dots P(Y_k^Q|X_k)$$

Forward algorithm with multiple outputs

It can be generalized as Eq.11:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k|X_{k-1}, U_{k-1}) \prod_{q=1}^{Q} P(\mathcal{Y}_k|X_k) \tag{11}$$

here $\mathcal{Y}$ is the set of outputs. $\mathcal{Y} = \{Y^1, Y^2, \dots, Y^Q\}$

This is a complete version of forward algorithm considering multiple inputs and multiple outputs.

Backward algorithm with multiple outputs

Similarly, to the forward algorithm, the backward algorithm is also adapted as Eq.12:

$$\beta(X_k) = \sum_{X_{k+1}=s_1}^{s_N} \beta(X_{k+1}) P(X_{k+1}|X_k, U_k) \prod_{q=1}^{Q} P(\mathcal{Y}_{k+1}|X_{k+1}) \tag{12}$$

here $\mathcal{Y}$ is the set of outputs. $\mathcal{Y} = \{Y^1, Y^2, \dots, Y^Q\}$

This is a complete version of backward algorithm considering multiple inputs and multiple outputs.

3.1.5 Normalization

The values in the α and β tables can come very close to zero. Multiplying them together can cause the risk of exceeding the precision of the floating-point number and getting the results in a number which is smaller than what the proposed model is capable of the store. In this case, a normalization is used in the equations by scaling to guarantee good numerical properties.

Equation 13 represents the scaled forward algorithm:

$$\alpha(X_k) = \frac{\sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k|X_{k-1}, U_{k-1}) \prod_{q=1}^{Q} P(\mathcal{Y}_k|X_k)}{\sum_{X_k=s_1}^{s_N} \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k|X_t, U_{k-1}) \prod_{q=1}^{Q} P(\mathcal{Y}_k|X_k)} \tag{13}$$

Equation 14 represents the scaled backward algorithm:

$$\beta(X_k) = \frac{\sum_{X_{k+1}=s_1}^{s_N} \beta(X_{k+1}) P(X_{k+1}|X_k, U_k) \prod_{q=1}^{Q} P(\mathcal{Y}_{k+1}|X_{k+1})}{\sum_{X_k=s_1}^{s_N} \sum_{X_{k+1}=s_1}^{s_N} \beta(X_{k+1}) P(X_{k+1}|X_k, U_k) \prod_{q=1}^{Q} P(\mathcal{Y}_{k+1}|X_{k+1})} \tag{14}$$

There is another popular approach which is the log space. It turns the multiplications into additions and thus avoids too small values.

Forward algorithm for IOHMM

$$log\ \alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \{log\ \alpha(X_{k-1}) + log\ P(X_k|X_{k-1}, U_{k-1}) + log\ \prod_{q=1}^{Q} P(\mathcal{Y}_k|X_k)\} \tag{15}$$

Backward algorithm for IOHMM

$$log\ \beta(X_k) = \sum_{X_{k+1}=s_1}^{s_N} \{log\ \beta(X_{k+1}) + log\ P(X_{k+1}|X_k, U_k) + log \prod_{q=1}^{Q} P(\mathcal{Y}_{k+1}|X_{k+1})\} \tag{16}$$

Equation (15) and (16) are the complete adaptation of forward-backward algorithm for IOHMM which are used in the Baum Welch algorithm to learn the model parameters.

3.1.6 The Baum Welch adaptation

The adapted Baum-Welch algorithm uses the adapted forward-backward algorithm (Eq. 15 and 16) through the expectation and maximization steps for learning the IOHMM parameters $\Lambda = (A^p, B^q, \pi)$ (cf. chapter 2). This algorithm also requires some initial values for variables π, A^p, and B^q to run the learning process. This initialization would be better if it follows the system nature, otherwise, random values could also be chosen.

Baum Welch algorithm for IOHMM

Following the classical Baum Welch variables given by Eq.4 and 5, the probability of being in state j at time k given multiple observed sequences $(Y^1, \dots, Y^q)$ and the parameters of Λ is given below:

$$\omega_k(i) = \frac{\alpha_i(X_k)\beta_i(X_k)}{P(Y^1, \dots, Y^q|\Lambda)}$$

The probability of being in state i and j at time k and $k+1$ given the observed sequences of $(Y^1, \dots, Y^q)$, the input operating conditions U and the parameters of Λ is given in the equation below:

$$\varepsilon_k(i,j) = \frac{\alpha_i(X_k).\, a^p(U_{k-1})_{ij}.\, b^q{}_j.\, \beta_j(X_{k+1})}{P(Y^1, \dots, Y^q|\Lambda)}$$

Parameters updating:

Initial state probability:

$$\hat{\pi}_i = \varepsilon_1(i,j), \text{ where } 1 \le i \le N \tag{17}$$

Transition probabilities:

$$\hat{a}^p{}_{ij} = \frac{\sum_{k=1}^{K-1} \varepsilon_k(i,j).\, 1_{U_{k-1}=p}}{\sum_{k=1}^{K-1} \ \omega_k(j).\, 1_{U_{k-1}=p}} \tag{18}$$

where $1_{U_{k-1}=p} = \begin{cases} 1 \ if \ U_{k-1} = p \\ 0 \quad otherwise \end{cases}$

Emission probabilities:

$$\hat{b}^q{}_{jk} = \frac{\sum_{k=1}^{K} \ \omega_k(j).\, 1_{Y_k^q=v_m}}{\sum_{k=1}^{K} \ \omega_k(j)} \tag{19}$$

where $1_{Y_k^q=v_m} = \begin{cases} 1 \ if \ Y_k^q = v_m \\ 0 \quad otherwise \end{cases}$

Repeat these steps until the changes between the two results converge.

In this section, the classical forward-backward algorithms are adapted in several versions to integrate multiple inputs, outputs, normalization, and numerical solution. The Baum welch algorithm is also adapted from the classical HMM to the IOHMM version where the parameters get updated according to the given inputs. Finally, the parameters of the model are estimated as $\Lambda = (\hat{A}^p, \hat{B}^q, \hat{\pi})$ which completely represents the IOHMM.

3.2 Numerical Illustration (IOHMM learning)

To show the proposed methodology a numerical application is simulated. The application is assumed to have such complexity that covers several challenges to explore the importance of the proposed methods. Different uncertainties are handled in the model training (e.g. data uncertainty, small dataset, missing data, model size, operating conditions, etc.). The numerical problems are handled by scaling the small values and applied the logarithm method. The training is also done by using the bootstrap method which is useful to provide confidence over the parameter estimation and give a reasonable result for small datasets.

This application assumed to have two observation outputs and one operating condition with two operating modes. For example, if the speed of a system considered an operating condition then two operating modes can be the high and the low speed. Two operating modes provide two stochastic matrices to describe two different transition probabilities for the system's degradation. The degradation of the system assumed to have three hidden states (good, moderate, bad) in simulations for easy and simple computation. Each of the states emits two outputs with two probabilities which are represented by two emission matrices. There are three discrete variables considered as the emitted symbols.

The goal is to use a simulated dataset and training the model to estimate the parameters of the model considering different issues of uncertainties and constraints. The training is done in three different phases to solve different issues.

- **Modeling under multiple operating conditions and output observations**. It is the classical problem in which the dataset assumed as a complete dataset that does not have any incomplete or missing data sequences. The adapted algorithms (Eq. 1 to Eq. 5) are used in this training phase.
- **Modeling under missing data**. The missing data is a typical challenge in a data-driven approach. In this phase, a solution is proposed to handle the dataset with missing elements. The adapted algorithms are modified again in this phase for managing the missing data.
- **Use the bootstrap method for having the confidence over the estimated model**. Bootstrap method can provide a scale of confidence for the estimated parameters even from a small amount of data. Usually, the data amount is small for diagnostic and prognostic applications. In this phase, the bootstrap method is implemented to train the model from a small data amount.

3.2.1 Modeling under multiple operating conditions

This is the classical model training considering multiple operating conditions and multiple outputs where the dataset is complete. The model provides two transition matrices for two operating conditions and two emission matrices for two observation outputs.

3.2.1.1 Data preparation

A simulator is developed based on the IOHMM concept to simulate the data sequences using a given model structure. For the sake of illustration, a structure is given below in which the attributes can be different based on different applications. Later, the estimated parameters are compared with the given model structure.

Given model architecture:

- Data unit: discrete
- Model type: left-right model
- The number of input states: two
- The number of hidden states: three (assumed as "good", "moderate", and "bad")

- The number of observation symbols: three

Transition matrices: the parameters are chosen randomly by conditioning the diagonal values bigger compared to other parameters because the matrix represents the left-right type respect to the system degradation nature. Usually, systems temp to stay on a health state during a long-time step compared to transition in the final state.

Transition matrices:

$$A^1 = \begin{pmatrix} 0.9788 & 0.0212 & 0 \\ 0 & 0.9516 & 0.0484 \\ 0 & 0 & 1 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 0.8443 & 0.1557 & 0 \\ 0 & 0.7899 & 0.2101 \\ 0 & 0 & 1 \end{pmatrix}$$

Emission matrices: (selected randomly)

$$B^1 = \begin{pmatrix} 0.8980 & 0.0513 & 0.0507 \\ 0.0534 & 0.8980 & 0.0486 \\ 0.0499 & 0.0505 & 0.8996 \end{pmatrix}, B^2 = \begin{pmatrix} 0.7981 & 0.1520 & 0.0499 \\ 0.2017 & 0.7488 & 0.0495 \\ 0.1007 & 0.0495 & 0.8498 \end{pmatrix}$$

Initial state distribution: (assumed as in good health)

$$\pi = (1 \quad 0 \quad 0)$$

The training sequence is defined as it has the failure measurements at the end of the sequence. Two sets of observation sequences (Y^1, Y^2) and one set of input sequences are simulated which correspond to the state sequences. The state sequences are also simulated for result comparison purpose.

The simulator initializes the distribution from the first state according to π. The method takes the initial state as $X_1 = s_1$ and simulates the emission distributions $P(Y_1^1|X_1 = s_1)$ and $P({Y^2}_1|X_1 = s_1)$ for the two outputs Y^1 and Y^2. After that, it takes the transition probability as $P(X_2|X_1 = s_1,\, U_1)$. The observations at time $k+1$ are simulated following the same process. These steps are repeated to simulate a state sequence as X and a single sequence of observations for each output Y^1 and Y^2. The length is chosen randomly between 180 to 250 for having a different length for the sequences of corresponding X, Y^1, and Y^2. A different range can be chosen for the length. The important issue is to have at least one failure information per sequence.

3.2.1.2 Application properties

Issue of resolution: The smaller is the time window covered by the sequence the better the resolution is (the resolution for detecting the state of health improves during implementation). However, this is only possible to the extent that there is sufficient information in the order of possible state detection. Observation sequences that are too short can lead to false alarms or misclassifications (Baruah, 2005). In this study, it is assumed that a resolution of one failure state (the first) is acceptable.

The number of health states: The degradation mechanism of a system has several distinct health states prior to reaching the failure state. Any number of hidden states that more than one can be set for the simulation. The important issue is to justify the selection of more or less hidden states. In our thesis, three health states are defined along with the failure state. The initial state, labelled as ‘good’, is a period during which the system assumed has no degradation. The second state, labelled as ‘moderate, is a period that indicates a slight degradation but still intolerance. The third state labelled as ‘bad’, is defined as the final state which about to shut down the system. It is also defined as the breakdown state or failure state.

Preparing training and testing data sets: About 100 data sequences are simulated where each one of them assumes to end up when the system gets a failure. This is the training set that is used in IOHMM training to estimates the parameters. Another data set is simulated having 100 data sequences where each one of them stopped a prior time earlier than the failure. This is the test set used in diagnostic/prognostic application and model validations.

Choice of IOHMM type: An ergodic model is most appropriate for imposing the enforcement of a strict left-right state transition constraint that may impair its ability to model the time series data. It is called the Bakis model (Yuan, 2018). If a system physically degrades as a left-right process, then modelling the observation through the left-right HMM (LR-HMM) will lead to better generalizations (Baruah, 2005).

Iteration: The Baum Welch algorithm rotates the training process several times starting from different initial states to ensure a good result of convergence towards a locally optimal solution. The given application sets the iteration by a given number. The number could be different corresponding to the complexity of the model.

3.2.1.3 Result

Since this is a simulated application from a given model structure, so the original number of hidden states is known. However, in real case, it is unknown. So, the experiment starts the model training with the minimum two states and gradually does the next training with a different number of states. The model is trained several times with different numbers of hidden states by assuming the original model structure is unknown. The same dataset is being used in all cases.

Selects the number of hidden states:

The degradation requires at least two hidden states (good and bad) to represent itself from good health state to bad health state. That is why this experiment starts with two hidden states. Nevertheless, there is no limitation in choosing the number of hidden states. The more the states the more accurate the degradation speed would be. However, noted that, a greater number of states makes the model more complex to estimate.

We decided to consider 4 models for the same system and compare them to decide which one is enough and suitable for the system. Every model learns one initial distribution, two transition matrices, and two emission matrices because the system has two inputs operating conditions and two observation outputs. However, in Table 3, only the first transition matrix and the first emission matrix of each model are presented to explain the solution. Other matrices can also be used to come up with a similar conclusion.

Table 3: Learning parameters of different matrices

Model	Transition matrices	Emission matrices
2 states-Model	$A^1 2 = \begin{pmatrix} 0.9867 & 0.0133 \\ 0 & 1.0000 \end{pmatrix}$	$B^1 2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.75 & 0.25 \end{pmatrix}$
3 states-Model	$A^1 3 = \begin{pmatrix} 0.9807 & 0.0193 & 0 \\ 0 & 0.9699 & 0.0301 \\ 0 & 0 & 1.0000 \end{pmatrix}$	$B^1 3 = \begin{pmatrix} 0.86 & 0.14 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
4 states-Model	$A^1 4 = \begin{pmatrix} 0.9801 & 0.0199 & 0 & 0 \\ 0 & 0.9856 & 0.0144 & 0 \\ 0 & 0 & \mathbf{1.0000} & 0 \\ 0 & 0 & 0 & \mathbf{1.0000} \end{pmatrix}$	$B^1 4 = \begin{pmatrix} 0.86 & 0.14 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

Here $A^1 2$ and $B^1 2$ are the estimated parameters of the first matrices of 2 states IOHMM.

Similarly, $A^1 3$ and $B^1 3$ are from 3 states IOHMM and so on.

5 states-Model	$A^15 = \begin{pmatrix} 0.9858 & 0.0142 & 0 & 0 & 0 \\ 0 & 0.8547 & 0.1453 & 0 & 0 \\ 0 & 0 & 0.9948 & 0.0052 & 0 \\ 0 & 0 & 0 & 0.9227 & 0.0773 \\ 0 & 0 & 0 & 0 & 1.0000 \end{pmatrix}$	$B^15 = \begin{pmatrix} 1 & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$

This experiment is done in the following three procedures:

- Look-out the transition parameters
- Look-out the emission parameters
- Compare by model-performance

1) Look-out the transition parameters: The proposed method is performing as a left-right model where no back transition is considered. We choose to model a system according to the left-right model which tends to stay in a healthy state with a higher probability then goes into the next state. In this case, the model follows a diagonal pattern having the maximum probability than the other parameters of the matrix because usually the health stays at one state long time compared to transfer into the next state. For example, the transition matrix A^13 which is learned from a 3-state model where all the diagonal parameters (1,1), (2,2), and (3,3) have the max-probabilities about 97%, 98%, and 100%. Another important issue is, if any state carries a 100% probability, then it is considered as breakdown state. It is also called an absorbent state from which there is no transition to any other state but itself. We considered only one absorbent state in the model.

Now, if we look at another transition matrix A^14, we find two parameters (3,3) and (4,4) holding 100% probabilities. By means of which, it has two absorbing states which are not acceptable as we defined to consider only one per model. So, the first absorbent state (3,3) can be chosen as the breakdown state because the model gets into this state first then there is no way of transiting to the state (4,4). So, agreeing to these insignificance parameters of the matrix, A^14 is not the elegant model to represent the system.

2) Look-out the emission parameters: Sometimes a transition matrix carries all the parameters with a significant probability. So, it is not possible to define the model is suitable or not such as matrix A^15. This matrix has all the parameters according to the left-right model and with the diagonal pattern holding the maximum probabilities. In this case, we can investigate the corresponding emission matrix B^15. There is a strong relationship between the transition matrix and the emission matrix of the same model structure. According to the emission matrices (B^12, B^13*)* from the other two models, the parameters in diagonal are holding the maximum probabilities in the matrix. In the matrix B^12, the parameter (1,1) carries 1, which means the probability of 1st state emits the 1st symbol is 100%. Another parameter (2,2) carries 0.75 means the 2nd state emits the 2nd symbol with a probability of 75%. B^13 is similarly holding the maximum probability in the diagonal positions.

The experience from different emission matrices and their nature says that there is a tendency of having the maximum probability of 1st state emits the 1st symbol, 2nd state emits the 2nd symbol, and so on. That means the probability of n^{th} state emits the m^{th} symbol (when $n = m$) is usually higher than emitting the other symbols. However, if we see the emission matrix B^15, then we can notice the parameters are quite unusual then other emission matrices. Here, the 3rd state mostly emits 2nd symbol and the 5th state mostly emits 3rd symbol where $n \neq m$, which is fairly different than what we learned earlier.

There is another way of explaining this issue which is the repeated parameters in the consecutive rows in the emission matrix. In matrix B5, there is a strong confusion between hidden states 2 and 3 and hidden states 4 and 5. It is not satisfying because finally we cannot know which hidden state is active.

The 2nd row and the 3rd row are having a 100% probability of emitting the same (2nd) symbol, which is usually meant to be in 2nd row. Similarly, the 4th and the 5th rows are having a 100% probability of emitting the 3rd symbol, which is usually meant to show by the 3rd row. So, it works as the 3 states model.

So, according to this explanation, the 2nd and 3rd rows can be aggregated as state-2, and the 4th and 5th rows as state-3. Therefore, the modified matrix becomes a 3 by 3 matrix which presents the highest probability of n^{th} state emits the m^{th}, where $n = m$. That means, corresponding to some insignificance parameters of emission matrix $B^1 5$, this system can be represented by a 3-state model rather than by 5-state model.

3) Compare by model-performance: This technique performs based on the performance of the models. It compares the model performances to select the better one among them all. The most probability of generating observation sequence $P(Y|\Lambda)$ is selected as a suitable model for the system and the dimension of the transition matrix as the number of hidden states. A test sequence is selected to test the results. The model is considered the best which generates the sequence with maximum likelihood.

The IOHMM with three states provides better performance compared to other models. Noted that, the first two techniques of this experiment are also indicated that a 3-state model is suitable to represent the system.

The estimated transition parameters are:

$$\hat{A}^1 = \begin{pmatrix} 0.9807 & 0.0193 & 0 \\ 0 & 0.9699 & 0.0301 \\ 0 & 0 & 1.0000 \end{pmatrix}$$

$$\hat{A}^2 = \begin{pmatrix} 0.8514 & 0.1486 & 0 \\ 0 & 0.7901 & 0.2099 \\ 0 & 0 & 1 \end{pmatrix}$$

This technique can be applied to those models which cannot be decided from the first two techniques.

The estimated emission parameters are:

$$\hat{B}^1 = \begin{pmatrix} 0.8977 & 0.0512 & 0.0511 \\ 0.0557 & 0.8964 & 0.0479 \\ 0.0500 & 0.0505 & 0.8995 \end{pmatrix}$$

$$\hat{B}^2 = \begin{pmatrix} 0.7977 & 0.1523 & 0.0500 \\ 0.2038 & 0.7470 & 0.0492 \\ 0.1006 & 0.0494 & 0.8500 \end{pmatrix}$$

The estimated initial state distributions are:

$$\hat{\pi} = (1 \quad 0 \quad 0)$$

Remark: a higher number of states for a small margin of performance is better not to consider for avoiding the computational complexity problem, which is called “Occam’s Razor” (Thorburn, William M et al 1918).

3.2.1.4 Limitation

1) The Baum Welch algorithm uses the forward algorithm repeatedly, which can be a time-consuming process for a large data amount.

2) The data set is simulated as complete sequences where no data element is missing. For missing data sets the learning algorithms need to be adapted.

3.2.2 Modeling under missing data

Phase two covers the limitation of the first phase which is considering the missing data in model training. It is usual that the sensors sometimes misread the observation for different reasons. Misreading observation contains both the missing measurement and sensor saturation. The main contribution in this phase is to present a technique based on the IOHMM adapted algorithms that handles the missing data. Typically, if a dataset contains data sequences with some missing elements, the sequences can be excluded from the analysis. As a result, the data set becomes smaller which may lose some valuable information. This strategy is known as list-wise deletion or case-wise deletion (Allison, 2001), but it is less suitable for a small amount of dataset. The method followed in this section includes the missing data sequences into the analysis by simulating the missing portion of the sequence to produce a complete set of data. A technique such as the maximum likelihood is applied to estimate IOHMM parameters that offer substantial improvements over list-wise deletion.

3.2.2.1 Missing Data

A sensor may miss measurements in periodic segments such as all the sensors stopped at the same time for an accident or mechanical issues. Sometimes sensors could misread the measurements for a random time segment. One or more sensors could stop measuring data for several time periods.

Data units as zero:

Let's assume a system is observed by three sensors: one sensor collects the input measurements represented by U and the other two sensors collect two output measurements as Y^1 and Y^2. A clean data sequence means there is no missing data in the entire sequence. A missing data sequence means that at least one data unit in the sequence is missing or contains an unreadable data unit (see Fig. 17: The missing data replaced by zero).

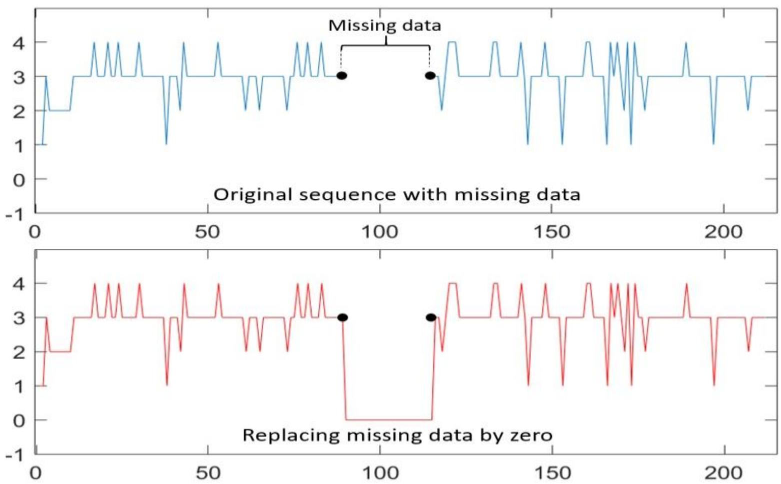


Fig. 17: The missing data replaced by zero
[X-axis: sequence length, Y-axis: discrete symbols]

An unreadable data unit could be some form of noisy or strange reading than the usual signal which does not give any information. For simple computation purposes in this work, a missing or abnormal data unit is replaced by "zero" which is out of the symbol list.

About 50% of complete sequences are transformed to missing data sequences through a converter. The amount of data elements and the index/indexes of removing the data are selected randomly. Some sequences have multiple missing windows/blocks. The converter takes each (complete) sequence and converts to missing data sequence according to two parameters: number of blocks, number of elements in each block.

Figure 18 gives an example of a complete sequence and a transformed sequence with sequence length of about 200 Data unit.

The number of blocks can be chosen randomly as $1 \leq d \leq D$

The number of elements in a block is also a random selection as $1 \leq Block\ size \leq K$, K is the sequence length.

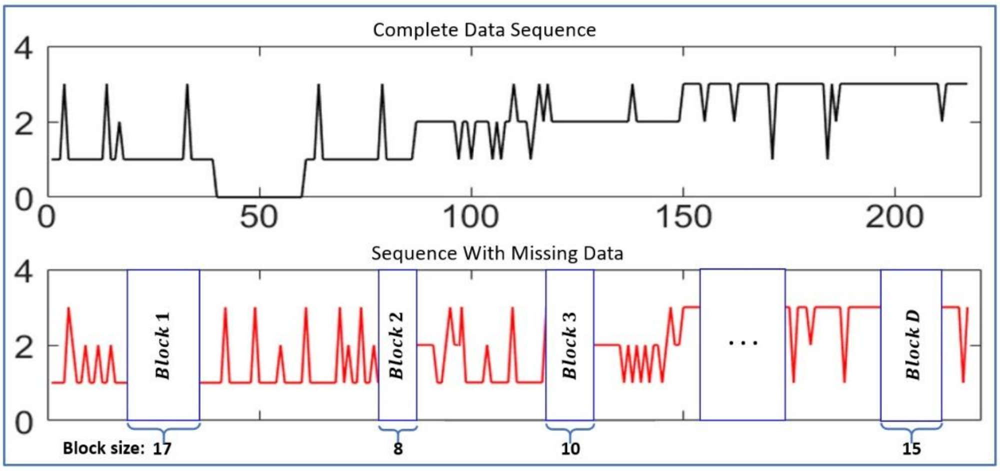


Fig. 18: The missing data conversion
[X-axis: sequence length, Y-axis: discrete symbols]

A data set could have one or more sequences with missing data in different cases in terms of different sensors failure. All possible cases for the three sensors are shown in Table 4.

Table 4: Different Cases of Data Unit

Case	Missing data combination	Description
1	$\bar{U}_k, \bar{Y}_k^1, \bar{Y}_k^2$	All three sequences have missing data
2	$\bar{U}_k, \bar{Y}_k^1, Y_k^2$	The input and the first output sequence has missing data, but the second output sequence has full length of clean data.
3	$\bar{U}_k, Y_k^1, \bar{Y}_k^2$	Only the first output sequence has full length of clean data
4	$\bar{U}_k, Y_k^1, Y_k^2$	Only the input sequence has some missing data
5	$U_k, \bar{Y}_k^1, \bar{Y}_k^2$	Only the input sequence has the full length of clean data
6	$U_k, \bar{Y}_k^1, Y_k^2$	Only the first output sequence has some missing data
7	$U_k, Y_k^1, \bar{Y}_k^2$	Only the second output sequence has some missing data
8	U_k, Y_k^1, Y_k^2	No sequence has any missing data

The data sequences having missing elements are defined as $\bar{U}$ (missing input), $\bar{Y}^1$ (missing first output) and $\bar{Y}^2$ (missing second output).

The IOHMM is trained in two parts. The first part only considers the case-8 where the sequences have no missing data (clean data set). In the second part, the model considers the rest of the cases (1 to 7) to get trained again. Case 4, 6, and 7 are less complex than cases 2, 3, and 5 because only one sensor gives the missing data in these cases. Case 1 is the most complex and less efficient because it has all the sequences with missing data.

3.2.2.2 Methodologies to handle the missing data

The model trains by using the missing data apply the Eq. 20 to Eq. 25 which are a slightly modified version of Eq. 11 and 12.

The forward algorithm for missing data:

The forward equation of IOHMM can be adapted to cover all the cases of Table 4. For example, *Case 4*: the input U_k is absent, but the outputs are available at time k:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \sum_{p=1}^{P} \alpha(X_{k-1})\, P(X_k|X_{k-1}, U_{k-1} = p\,)P(\mathcal{Y}_k|X_k)P(U = p) \qquad (20)$$

here $P(U = p)$ is the weight of using the p^{th} matrix over the inputs:

$$P(U = p) = \frac{C_p\,(count\ of\ p\text{th matrix})}{the\ number\ of\ elements\ of\ all\ the\ sequences\ in\ U}$$

In the Case 5: when two output's observations are missing, but the input is available. It can be handled in two approaches: replacing the emission probability by one or considering emitted symbol weight.

Approach one: Replacing the emission probability by one:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1})P(X_k|X_{k-1}, U_{k-1}) \qquad (21)$$

This method is used when any sequence shows zero elements (zero as missing data). The emission probabilities are missing because both the emitting outputs are missing so, the probability is assumed as $P(\mathcal{Y}_k|X_k)$=1, it is removed from the equation. However, if one of the emitted outputs has a non-zero element let's say the first output (Y_k^1) then the probability of $P\left(Y_k^1|X_k\right)$ is selected from the corresponding emission matrix but not the $P\left(Y_k^2|X_k\right)$ since the second output (Y_k^2) is zero, so $P\left(Y_k^2|X_k\right)$ is considered as 1.

When one output observation is absent while the other output observation and the input are available:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1})P(X_k|X_{k-1}, U_{k-1})\, P(\mathcal{Y}_k|X_k) \text{ for } \mathcal{Y}_k \neq 0. \qquad (22)$$

Similar approach is applicable when Y_k^1 is zero but Y_k^2 is nonzero, $P\left(Y_k^2|X_k\right)$ selects from the emission matrix and $P\left(Y_k^1|X_k\right)$ considered as 1.

Approach two: Considering emitted symbol weight

Another approach is considering the missing output observation to compute the emission probability by summing over all possible emitted symbols, weighted by their appearance probability in the training observation sequences. This approach simulates the probable emission distribution for the missing window according to existing data for computing the state transition:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \sum_{m=1}^{M} \alpha(X_{k-1})\, P(X_k|X_{k-1}, U_{k-1})\, P(\mathcal{Y}_k = v_m|X_k)P(\mathcal{Y} = v_m) \qquad (23)$$

here v_m is an emitted symbol could be 1 to M, and $P(Y^q = v_m)$ is the symbol weight by their probability in the training observations.

The variable probability $P(\mathcal{Y} = v_m)$ represents the weight of the emitted symbols as:

$$P(\mathcal{Y} = v_m) = \frac{C_{v_m}(count\ of\ the\ symbol\ v_m\ observation)}{the\ number\ of\ elements\ of\ all\ the\ sequences\ in\ \mathcal{Y}}$$

Equation (20) and (23) together cover all the missing cases of three sensors mentioned earlier. Likewise, the backward algorithm, the Baum Welch algorithm, and the Viterbi algorithm are also modified to deal with the missing data. The modification of these three algorithms is shown only for case-1 (most complex one).

The backward algorithm:

$$\beta(X_k) == \sum_{X_{k+1}=s_1}^{s_N} \sum_{m=1}^{M} \sum_{p=1}^{P} \beta(X_{k+1}) P(X_{k+1}|X_k, U_k = p) d_p\ P(\mathcal{Y}_k = v_m|X_k) P(\mathcal{Y} = v_m) \quad (24)$$

The Baum Welch algorithm

$$\varepsilon_k(i,j) == \frac{\sum_{m=1}^{M}\ S \times P(\mathcal{Y}_k = v_m|X_k) P(\mathcal{Y} = v_m)}{P(\mathcal{Y}_{1:k}|\Lambda)} \quad (25)$$

where, $S = \sum_{p=1}^{P} \alpha_i(X_k).\, a^p(U_k = p)_{ij}.\, b^q{}_{jk}.\, \beta_j(X_{k+1}) P(U = p)$

3.2.2.3 Flow chart of model training under missing data

To use the adapted methods considering the cases of missing data summarized in Table 4, the model flowchart is proposed in Fig. 19 where four phases are of main concern.

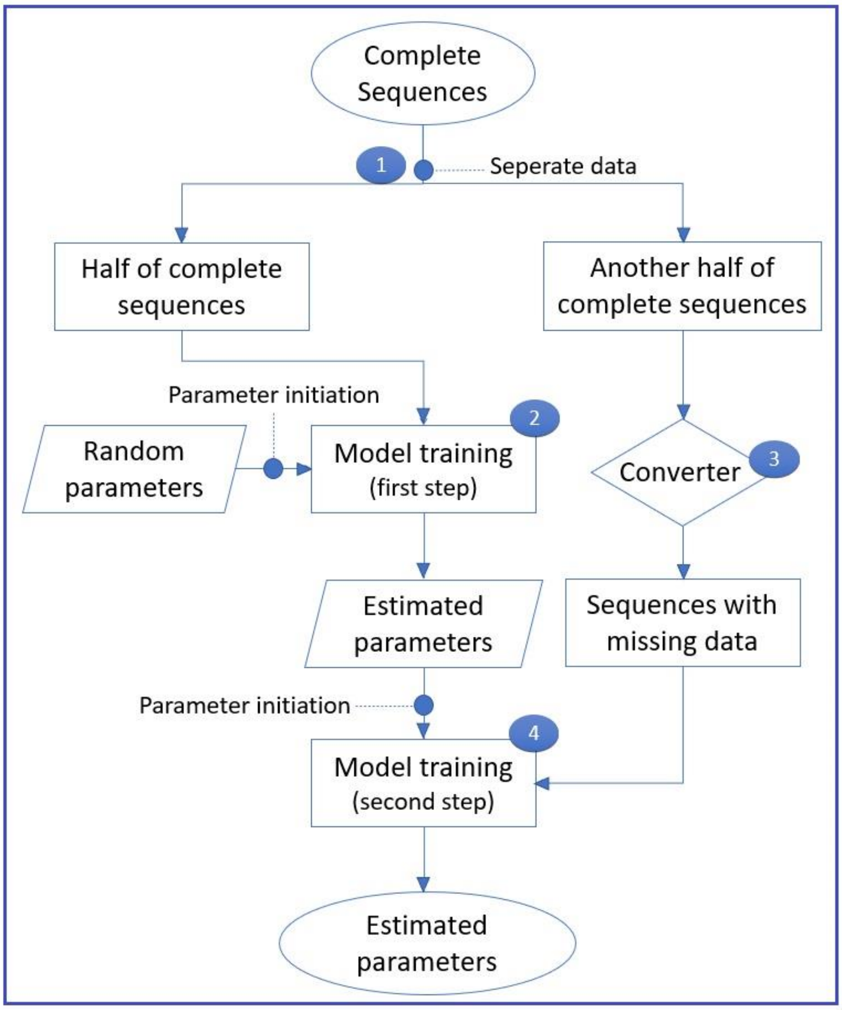


Fig. 19: The model flowchart

Phase one: all the data sequences are separated in half. The first half as the sequences to convert as missing data sequences and the second half as the sequences keep as complete and clean sequences.

Phase two: the model gets training in two steps. The first step is done in this phase. Only the complete sequences are used to training the model by applying the classical algorithms of IOHMM (Eq.15 to Eq.19). The initial parameters are taken as the random values corresponding to the left-right property.

Phase three: in this phase, the separated sequences are converted to missing data sequences.

Phase four: in the final phase, the second step of model training is done by using the missing data sequences. In this step of training, the initial parameters are not taken randomly but the estimated parameters of the first step. This is how the model ensures to use all the available sequences in model training.

The data preparation and the corresponding results are presented in the following sections.

3.2.2.4 Data generation

A random set of 60 clean and complete data sequences are simulated from the given model structure. After that, 30 random sequences are converted into missing data sequences by randomly removing some elements from several indexes. A total of about 12.81% of data elements are removed from the training data sets where each of the sequences can miss the data in a bulk of one or more times. The goal is to compare the model performance between the results produced by using the clean data set and missing data set.

The output of the examination is presented in two steps:

Step one: use only 30 clean data sequences and apply Eq. 15 to 19 for parameter estimation.

Step two: use estimated results of the first phase as the initial parameters in the second phase and apply two different approaches on the other 30 sequences having missing elements for the final parameter estimation.

- Approach one: considering emission probability as 1 (*i.e*. Eq. 22) based on different cases of missing output data.
- Approach two: computes the emission probability based on all the possible emitted symbols, weighted by their probability (*i.e*. Eq. 23).

3.2.2.5 Result

Table 5 presents the nonzero estimated parameters of the transition matrices. Three cases (1, 2, 6) are examined where one or more sensor data assumed to be missing some measurements and case 8 with all clean data sequences as a reference. The missing data set is formatted according to the cases represented in Table 4. Case 6 is selected from the simplest cases (4, 6, 7) where at least one sensor is set to produce the missing data. Case 2 is selected from a little bit more complex cases (2, 3, 5) where a minimum of two sensors produce the missing data. Case 1 is the most complex case where all three sensors may produce the missing data. Finally, the original model parameters are also given in the last column of the table for comparison purposes.

Table 5: Model parameters from the first approach

Parameter	Case 1	Case 2	Case 6	Case 8	Original	Case 2 (listwise)
Transition Matrix $\hat{\boldsymbol{A}}^{\mathbf{1}}$						
$\hat{\boldsymbol{A}}^{\mathbf{1}}_{\mathbf{11}}$	0.9801	0.9801	0.9798	0.9793	0.9788	0.9803
$\hat{\boldsymbol{A}}^{\mathbf{1}}_{\mathbf{12}}$	0.0199	0.0199	0.0202	0.0207	0.0212	0.0197
$\hat{\boldsymbol{A}}^{\mathbf{1}}_{\mathbf{22}}$	0.9687	0.9682	0.9670	0.9689	0.9516	0.9723

$\widehat{\boldsymbol{A}}^1_{23}$	0.0313	0.0318	0.0330	0.0311	0.0484	0.0377
$\widehat{\boldsymbol{A}}^1_{33}$	1	1	1	1	1	1
Transition Matrix $\widehat{\boldsymbol{A}}^2$						
$\widehat{\boldsymbol{A}}^2_{11}$	0.8643	0.8543	0.8613	0.8434	0.8443	0.8617
$\widehat{\boldsymbol{A}}^2_{12}$	0.1357	0.1457	0.1387	0.1566	0.1557	0.1383
$\widehat{\boldsymbol{A}}^2_{22}$	0.7969	0.7769	0.7929	0.7869	0.7899	0.7944
$\widehat{\boldsymbol{A}}^2_{23}$	0.2031	0.2231	0.2071	0.2131	0.2101	0.2056
$\widehat{\boldsymbol{A}}^2_{33}$	1	1	1	1	1	1
D_Error	0.0908	0.0818	0.0728	0.0434	-	0.0882

D_Error represents the error distance of the estimated parameters from the original parameters. According to the error score, Case-8 has the lowest error (0.0434) that means the better parameters compared to others because this case uses the maximum information in the training as it does not have any missing data. Similarly, case 6 gives a better result than case 2 and case 2 gives better result than case 1 corresponding to the amount of missing data consideration. Case 1 has the maximum missing data elements, so it has the maximum error score (0.0908).

All the parameters shown in table 5 are estimated by applying the first approach (eliminating emission probability). This is less complicated to implement compared to the second approach which considers the emitted symbols weighted by their probability. However, the second approach is a complex implementation but gives a better result such as $P(\mathcal{Y}|\Lambda_{c=2}) = 6.0e^{-128}$ for case 2, while eliminating the emission probability gives less probability as $3.5e^{-128}$. However, if an application requires only the max path but does not care about the distribution $P(\mathcal{Y}|\Lambda)$, then the first approach is suitable because in several experiments give the same max path and show that it is a time-consuming approach.

The second approach is examined for case 2 for giving a solution that covers all the challenges of other cases. The input ($\overline{U}$) and the first output ($\overline{Y}^1$) sequences are having some missing data but the second output (Y^2) has the clean data sequences. Two different results are produced based on this case. The first one is shown by column 3 that has the distance error of 0.0818 which is the result by considering both complete and incomplete sequences. On the other hand, the second result is produced by considering the list-wise (Allison, 2001) approach where the model is trained only by the complete sequences. It provides a distance error of 0.0882 (in the last column) which show an improvement of considering the incomplete sequence with missing data elements.

However, in this case 2, there are three discrete symbols (v_1, v_2, v_3) in the missing data sequences of $\overline{Y}^1$ and their prior distribution weights is $P(Y^q = v_1) = 0.19$, $P(Y^q = v_2) = 0.14$, and $P(Y^q = v_3) = 0.67$. Now, this information is used in the model training by using (12) and for estimating the model parameters.

The estimated transition parameters of IOHMM considering the $\Lambda_{c=2}$ (case 2):

The estimated transition parameters for case 2 are:

$$\hat{A}^1 = \begin{pmatrix} 0.9793 & 0.0207 & 0 \\ 0 & 0.9689 & 0.0311 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\hat{A}^2 = \begin{pmatrix} 0.8434 & 0.1566 & 0 \\ 0 & 0.7869 & 0.2131 \\ 0 & 0 & 1 \end{pmatrix}$$

$\hat{A}^1$ is the low-stress model where the mean transition probability from the first state to the last state is $(0.0207 + 0.0311)/2 = 0.0259$.

$\hat{A}^2$ is the high-stress (*i.e.* high speed) model transitions where the mean transition probability from the first state to the last state is $(0.1566 + 0.2131)/2 = 0.1849$. It goes to high degradation faster than matrix $\hat{A}^1$. Therefore, the mean time to reach the final state is lower than using the matrix $\hat{A}^1$.

3.2.2.6 Discussion

Missing data is a common problem in modern statistical research. It appears in analyzing sensor measurements where data get missing due to many reasons.

The size of the dataset is an important issue in statistical applications such as prognostic and health management of the system. Unfortunately, the sample size is not large in this domain because the degradation is a slow process and the observation sequences require to have at least one failure measurement. Therefore, a few amounts of missing data can reduce the effectiveness of the result.

Although, the missing can be random in size and index in the sequence, yet the sequence is not empty. There are still some data available inside the sequences which could be useful. Therefore, the proposed method can be a useful solution that does not compromise to lose any information from the available data elements in the sequences.

3.2.2.7 Limitations

The algorithm would be less efficient for too much missing data, because there is a possibility of losing a significant amount of information if missing data amount is huge.

3.2.3 Modeling by using the bootstrap method

The bootstrap method is a sampling technique used to estimate IOHMM parameters by sampling a dataset with replacement. This method used to estimate measures of accuracy, such as confidence intervals, the sample mean, standard deviation, variance, etc. Because of the replacing technique this method also provides a good result for a small dataset compared to the classical method.

Resampling with replacement selects a subset from the original sample randomly for training the model. After that, it returns to the subset into the sample again for another selection. Resampling size can be equal to the sampling size which may have some repeated dataset. This technique maintains data structure but reshuffles values, extrapolating to the data population. This repeated process uses the new sample to generate the sampling distribution of the mean. Bootstrapping is useful for estimating IOHMM parameters when the data amount is small, data pollution is unknown, data are non-normal, or have unknown statistic properties, etc. The method provides standard calculations such as 95% confidence intervals or the coefficient of variation, etc.

3.2.3.1 Bootstrap properties

Confidence interval (CI): Confidence interval estimated from observed statistical data, which may contain an unknown population parameter. The CI communicates the accuracy of a probabilistic estimate. It expresses a range in which it is fairly certain that the population parameter is present. The range-width depends on the variation within the population of interest and the sample size (Efron, 1986).

Population variation: If all values in a large data population are almost the same, then the sample also has a small variation. It gives a small confidence interval. On the other hand, more varied data will lead to more varied samples, which makes less sure that the sample average is close to the population mean. That means the CI is large in this case. The greater variation of the data leads to a wider CI.

Sample size: The sample size also affects the width of a confidence interval. Small samples differ more from each other and have less information. There is more variation due to a sampling error. The CI may be larger. On the other hand, larger samples will be more similar. The effect of the sampling error is less, and the information is more. The confidence interval may be smaller in this case (Efron, 1986).

Calculating confidence intervals: The confidence interval calculation (for a mean uses) the Eq. 26:

$$CI = \bar{X} \pm t\frac{s}{\sqrt{n}} \tag{26}$$

here $\bar{X}$ is the sample mean, t is the t-distribution which depends on the sample size and the chosen level of confidence, s is the sample standard deviation and n is the sample size.

Sample: A sample is a selection of observations from the population of interest. The selection criterion is random, convenient, systematic, clustered, layered, etc.

Sampling error: A sample is only a selection of objects from the population. It will never be a perfect representation of the population. Different samples of the same population will yield different results. This is called sampling error or sampling variation. There will always be a sampling error (Efron, 1986).

The sample means: Defined as the average of observations in the sample of the population. The sample mean is considered as the estimate of the population mean.

Sample standard deviation: It is the average distance of the sample data from the sample mean.

3.2.3.2 Data preparation

To demonstrate the bootstrap method, a set of training sequences are simulated from the same given model structure that has been used in the first two phases. About 1000 training sequences and another 1000 testing sequences are simulated to train the model and test the model performance. Because of the big data amount, we decided to fix a small resampling size (30 sequences) to have a fair analysis of the poor quantity of data. We randomly select these sequences from the main dataset and apply the bootstrap method on it which takes a total of 1000 iterations and store the measurements of confidence intervals, standard error, bounds, and mean.

3.2.3.3 Result

Figure 20 shows the confidence interval for each parameter (total of 36 parameters) of transition and emission matrices. The X-axis of each rectangle represents the probabilities, and the Y-axis represents the boot execution number.

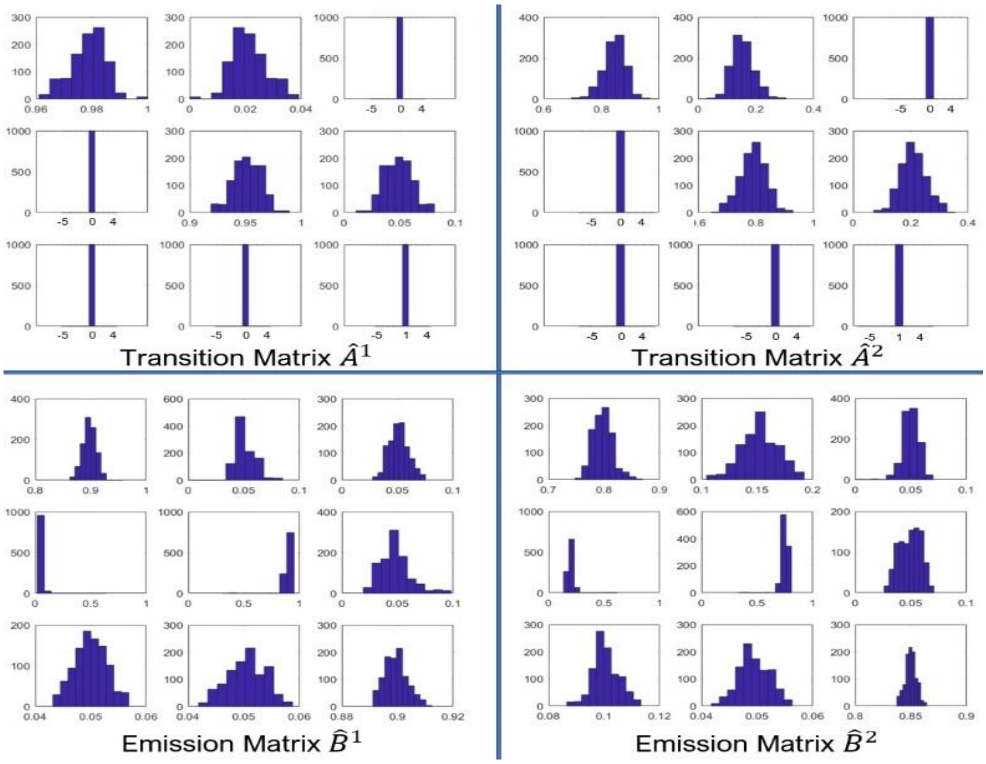


Fig. 20: Distribution of matrices parameters

Both the transition matrices are having some zeros on corresponding parameters. These parameters did not get any transition probability during the training following the nature of the system. The left-right model is used in data simulation as mentioned earlier. This is the reason why the transition matrices have zeros on (2,1), (3,1), (3,2) position (Fig.).

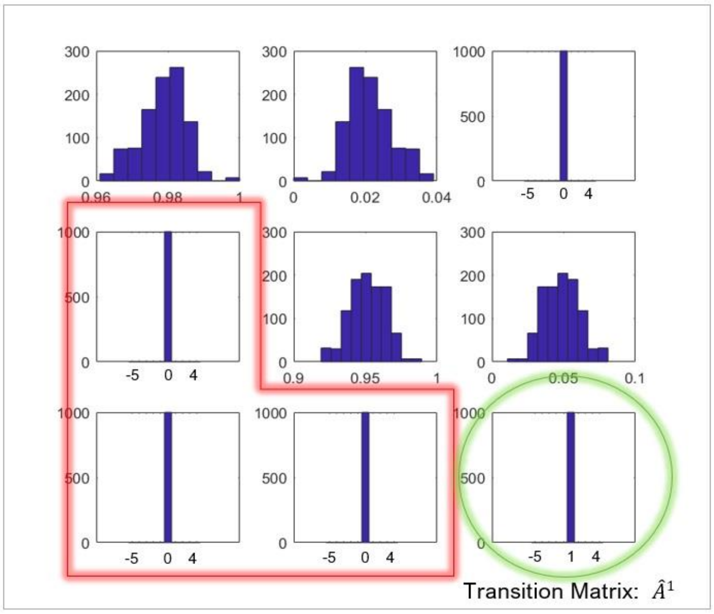


Fig. 21: Parameter distribution for the first transition matrix.

The green circle on the position (3,3) presents the absorbent state with a 100% probability. Besides these four, all the other parameters are estimated with a 95% confidence interval (see Table 6: Bootstrap parameters). This table shows all the parameters (transition parameters, emission parameters, and the initial state distributions) of the model. The row represents different information (lower bound, higher bound, mean, standard error) about a parameter. Parameters having zero value are ignored in the table.

Table 6: Bootstrap parameters

Parameter	**Lower bound**	**Higher bound**	CI Mean	**Standard Error**	**Original Distribution**
Transition Matrix $\hat{\boldsymbol{A}}^1$					
$\hat{\boldsymbol{A}}^1_{11}$	0.9783	0.9791	**0.9787**	1.96×10^{-4}	0.9788
$\hat{\boldsymbol{A}}^1_{12}$	0.0209	0.0217	**0.0213**	1.96×10^{-4}	0.0212
$\hat{\boldsymbol{A}}^1_{22}$	0.9508	0.9523	**0.9515**	3.91×10^{-4}	0.9516
$\hat{\boldsymbol{A}}^1_{23}$	0.0477	0.0492	**0.0485**	3.91×10^{-4}	0.0484
$\hat{\boldsymbol{A}}^1_{33}$	1	1	**1**	0	1
Transition Matrix $\hat{\boldsymbol{A}}^2$					
$\hat{\boldsymbol{A}}^2_{11}$	0.8428	0.8477	**0.8453**	0.0012	0.8443
$\hat{\boldsymbol{A}}^2_{12}$	0.1523	0.1572	**0.1547**	0.0012	0.1557
$\hat{\boldsymbol{A}}^2_{22}$	0.7861	0.7916	**0.7889**	0.0014	0.7899
$\hat{\boldsymbol{A}}^2_{23}$	0.2084	0.2139	**0.2111**	0.0014	0.2101
$\hat{\boldsymbol{A}}^2_{33}$	1	1	**1**	0	1
Emission Matrix $\hat{\boldsymbol{B}}^1$					
$\hat{\boldsymbol{B}}^1_{11}$	0.8970	0.8984	**0.8977**	3.70×10^{-4}	0.8980
$\hat{\boldsymbol{B}}^1_{12}$	0.0506	0.0517	**0.0512**	2.82×10^{-4}	0.0513
$\hat{\boldsymbol{B}}^1_{13}$	0.0506	0.0517	**0.0511**	2.80×10^{-4}	0.0507
$\hat{\boldsymbol{B}}^1_{21}$	0.0524	0.0591	**0.0557**	17×10^{-4}	0.0534

$\hat{\boldsymbol{B}}^1_{22}$	0.8929	0.8999	**0.8964**	18×10^{-4}	0.8980
$\hat{\boldsymbol{B}}^1_{23}$	0.0470	0.0487	**0.0479**	4.32×10^{-4}	0.0486
$\hat{\boldsymbol{B}}^1_{31}$	0.0498	0.0501	**0.0500**	0.89×10^{-4}	0.0499
$\hat{\boldsymbol{B}}^1_{32}$	0.0503	0.0507	**0.0505**	1.07×10^{-4}	0.0500
$\hat{\boldsymbol{B}}^1_{33}$	0.8993	0.8998	**0.8995**	1.29×10^{-4}	0.9000
Emission Matrix $\hat{\boldsymbol{B}}^2$					
$\hat{\boldsymbol{B}}^2_{11}$	0.7966	0.7988	**0.7977**	5.57×10^{-4}	0.8000
$\hat{\boldsymbol{B}}^2_{12}$	0.1513	0.1533	**0.1523**	5.25×10^{-4}	0.1500
$\hat{\boldsymbol{B}}^2_{13}$	0.0496	0.0505	**0.0500**	2.33×10^{-4}	0.0500
$\hat{\boldsymbol{B}}^2_{21}$	0.2011	0.2065	**0.2038**	14×10^{-4}	0.2000
$\hat{\boldsymbol{B}}^2_{22}$	0.7444	0.7497	**0.7470**	14×10^{-4}	0.7500
$\hat{\boldsymbol{B}}^2_{23}$	0.0486	0.0498	**0.0492**	3.05×10^{-4}	0.0500
$\hat{\boldsymbol{B}}^2_{31}$	0.1003	0.1009	**0.1006**	1.47×10^{-4}	0.1000
$\hat{\boldsymbol{B}}^2_{32}$	0.0493	0.0496	**0.0494**	0.87×10^{-4}	0.0500
$\hat{\boldsymbol{B}}^2_{33}$	0.8096	0.8502	**0.8499**	1.60×10^{-4}	0.8500
Initial state distribution					
π (1)	0.9761	0.9917	**0.9839**	0.0040	1
π (2)	0.0083	0.0239	**0.0161**	0.0040	0
π (3)	0	0	**0**	0	0

Total standard error in matrix $\hat{A}^1$ is $11.74\ \times 10^{-4}$, matrix $\hat{A}^2$ is 52×10^{-4}, matrix $\hat{B}^2$ is 51.894×10^{-4}, matrix $\hat{B}^2$ is 48.139×10^{-4}, and initial state distribution is 80×10^{-4}. Matrix $\hat{A}^2$ comparably has a larger standard error than the matrix $\hat{A}^1$ because of the amount of training data dedicated to each matrix. $\hat{A}^2$ is trained with about 20% data while 80% data are used to train matrix $\hat{A}^1$.

Now, if the matrices are organized with the mean values then we find the estimated parameters of IOHMM as following:

- Estimated transition parameters:

$$\hat{A}^1 = \begin{pmatrix} 0.9787 & 0.0213 & 0 \\ 0 & 0.9515 & 0.0485 \\ 0 & 0 & 1 \end{pmatrix}$$

$\hat{A}^1$ is the lowest stressed (*e.g.* low speed) model transitions where the mean transition probability from the first state to the last state is $(0.0213 + 0.0485)/2 = 0.0349$. The lowest stressed model is defined as the model which gives the maximum mean time to reach the final state compared to the other models.

$$\hat{A}^2 = \begin{pmatrix} 0.8453 & 0.1547 & 0 \\ 0 & 0.7889 & 0.2111 \\ 0 & 0 & 1 \end{pmatrix}$$

$\hat{A}^2$ is the highest stressed (*e.g.* high speed) model transitions where the mean transition probability from the first state to the last state is $(0.1547 + 0.2111)/2 = 0.1829$**.** The highest stressed model is defined as the model which gives the minimum mean time to reach the final state compared to the other models.

- Estimated emission parameters:

$\hat{B}^1$ presents emission probabilities for the first output sequences (*e.g.* temperature) and $\hat{B}^2$ is for second output (*e.g.* vibration).

$$\hat{B}^1 = \begin{pmatrix} 0.8977 & 0.0512 & 0.0511 \\ 0.0557 & 0.8964 & 0.0479 \\ 0.0500 & 0.0505 & 0.8995 \end{pmatrix}, \hat{B}^2 = \begin{pmatrix} 0.7977 & 0.1523 & 0.0500 \\ 0.2038 & 0.7470 & 0.0492 \\ 0.1006 & 0.0494 & 0.8499 \end{pmatrix}$$

- Initial state distribution: (estimated as in good health)

$$\pi\ = (0.9839 \quad 0.0161 \quad 0)$$

Comparison between the estimated and the original parameters

Figure 22 presents the distance between the estimated parameters and the original parameters used in data simulation. The parameters estimated twice, with bootstrap and without bootstrap method.

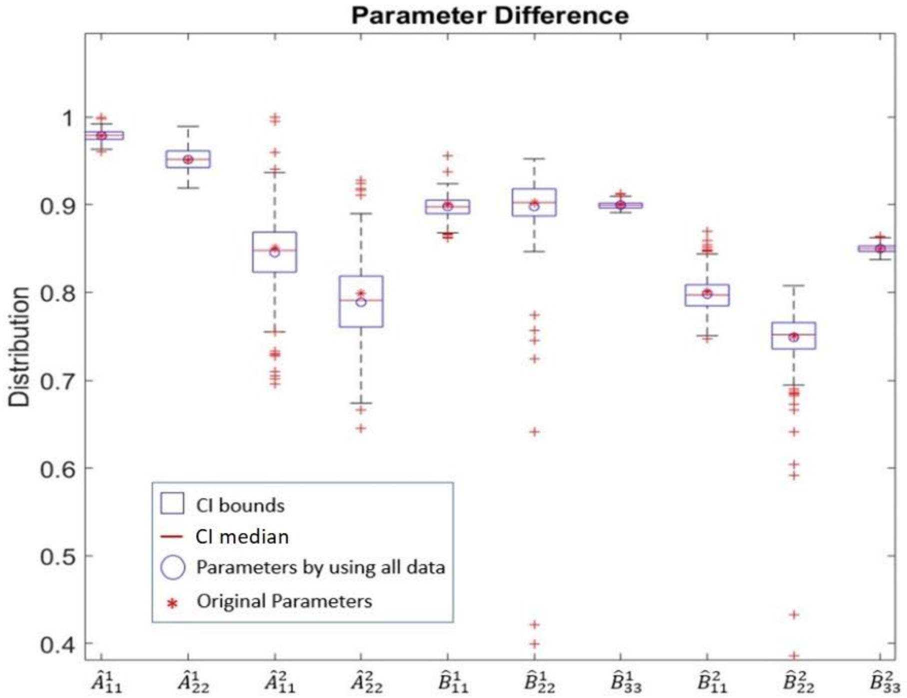


Fig. 22: Different parameters of learned and original models

Only the non-zero parameters are highlighted in the figure where the transition and the emission parameters are compared. The blue box represents the CI bounds where the red line inside the box represents the median and the star symbol represents the original parameters. The CI median is the estimated median using the bootstrap-IOHMM method. Another circle is inside each box which represents the second estimated parameters (without bootstrap) but with all the data.

The probability distributions of the parameters are given in table 7 where just one matrix ($\hat{A}^1$) is presented to give a benchmarking between the original parameters with the estimated parameters. The second column in the table is the CI mean written as the parameters with bootstrap. Both the estimated parameters are very close to the original parameters. However, the parameters with the bootstrap method are marginally better than the parameters that came from the training without bootstrap.

Table 7: Benchmarking parameters

Para-meters	Confidence Interval bound	Results With bootstrap	Results Without bootstrap (Same data size)	Original
$\hat{A}^1_{11}$	[0.9783, 0.9791]	0.9787	0.9847	0.98
$\hat{A}^1_{12}$	[0.0209, 0.0217]	0.0213	0.0153	0.02
$\hat{A}^1_{22}$	[0.9508, 0.9523]	0.9515	0.9472	0.95
$\hat{A}^1_{23}$	[0.0477, 0.0492]	0.0485	0.0528	0.05
$\hat{A}^1_{33}$	1	1	1	1
D_Error	-	0.0056	0. 0150	-

D_Error (Distance Error) = $\sum_{c=1}^{N}\sum_{d=1}^{N}\sqrt{(A^p_{cd}-\hat{A}^p_{cd})^2}$

3.2.3.4 Discussion

The bootstrap-IOHMM is being used because of its accuracy and control over the error rate. The bootstrap is a globally accepted method for its simplicity. It is a straightforward way to obtain standard error and confidence intervals which provide meaning over the distribution, coefficients, and abound of probability. Most of the accuracy and maintenance related problems use this method rather than the standard assumption to check and control the stability of the results. This is asymptotically more accurate than the standard ranges obtained using sample dispersion and normal assumptions. Bootstrapping is also a convenient method to avoid the cost of repeating the experiment to obtain other sample data sets. That is why the proposed method with bootstrapping is a smart choice for a problem with limited data sequences.

3.2.3.5 Limitation

The bootstrap does not provide a general finite sample guarantee. The result may depend on the representative data sample. It can be time-consuming depending on the sampling size.

3.3 Conclusion

This section described the modelling of systems by IOHMM. The Baum Welch and the forward-backward algorithms are adapted to learn the model parameters considering different data uncertainties and model uncertainties.

Three simulated applications are shown to explain three major issues of the training. The first application demonstrates the adaptations of the algorithms considering multiple operating conditions and their impact on health's degradation of systems. Multiple observation outputs are also integrated into this application.

Three techniques are proposed here to handle the uncertainty of model size such as fixing an appropriate number of hidden states of the model. The second application describes the data uncertainty of missing data. A different version of the adapted algorithms is used which are dedicated to handling the data sequences having missing elements into the model training to extract as much information as possible even from the incomplete sequences. The third application is about the bootstrap method implementation. This method incorporated with the proposed model to determine the parameter estimation with a confidence interval. A benchmarking is given where the results are compared between the original parameter, parameters with bootstrap, and parameters without bootstrap. The result of the bootstrap method is promising which is accepted by a number of researchers with their reviews and examinations through several conferences.

The next section is about the second contribution of the thesis where the proposed methodologies are used in diagnostics and prognostics applications.

Chapter 4

The Second Contribution: Diagnostic and Prognostic

Table of Contents

4 The Second Contribution: Diagnostic and Prognostic

The second contribution of the thesis is presented in this chapter which answers the second research question mentioned in the introduction. This contribution concerns about diagnostics and prognostics algorithms in order to estimate the remaining useful life (RUL) of systems under multiple operating conditions. RUL is a major challenge of prognostic and health management systems (PHM) in many industrial domains where safety, reliability, and cost reduction are of high importance. To reduce the cost, one solution is to match the maintenance date with the estimated remaining life of the system. The RUL prediction allows fixing time to organize a maintenance action which can be called maintenance time-window. Nevertheless, the RUL can change due to different dynamics of operating conditions over the system's run-time. That is why the distribution over the health state needs to be updated in a continuous process according to new measurements. Therefore, the online RUL prediction is a much more effective approach in condition-based maintenance. In this chapter, we described both the online and offline RUL estimation by using IOHMM. The diagnostic and the prognostic methodologies are described first, then the simulated application is given to demonstrate the methodologies.

Key issues:

- Diagnostic: predict the current health state of the system by applying the Viterbi algorithm which is adapted from HMM to IOHMM.
- Prognostic: predict the probable failure state. After that, compute the mean time between the current time to the failure time which defined as RUL. Two methods are demonstrated:
    - Numerical integration
    - Matrix computation
- Prediction types
    - Offline prediction: does not apply the new measurements into the analysis.
    - Online prediction: consider the new measurements to update the predictions.
- Handling uncertainties in RUL estimation
    - Future operating conditions: operating conditions that comes after the diagnostics which can be given or unknown.
    - RUL computation: the uncertainty about the RUL prediction is handled by applying the system RULs are predicted applying the probability distribution function (PDF) along with the Monte Carlo simulation.
- Numerical applications
    - Diagnostic and prognostic under multiple operating conditions
    - Managing RUL by managing the operating conditions to reach a given target

4.1 Diagnostic

In the scientific literature of control theory community, the diagnostic is the detection and isolation of system faults. In our context, it is the evaluation (computation or estimation) of the current health situation of a monitored system. This is a prerequisite dependence for future performance estimation and effective RUL computation of the system health. Diagnostic is challenging while the system degrades along with multiple operating conditions. There are several diagnostic applications in the literature, but they are less concerned about the operating conditions. Moreover, they are more particularly designed to control problems as health management. In addition, there is a strong relation between the model dedicated to diagnosing the system health and those to prognose the health evolution (Michel, 2018).

In this section, we proposed a solution to diagnose the health state of a system by considering multiple operating conditions and their effects on the degradation evolution and based on IOHMM. After that, we proposed to compute the mean time RUL to help organizing the maintenance schedule (which is out of the scope of the thesis).

The Viterbi algorithm adaptation

The Viterbi algorithm is dedicated to HMM and should be adapted to IOHMM. It computes the maximum likelihood sequence of hidden states according *i.e.* in determining the current health situation given the observations. The adaptation means integrating the input sequences into the algorithm.

The work of adaptation is done in three steps:

- Integrating multiple outputs
- Integrating multiple inputs
- Integrating a backward computation

Integrating multiple outputs

The algorithm (Eq.8) can be developed to integrate multiple outputs from classical formula: $P(X_k|\mathcal{Y}_k) \propto P(X_k, \mathcal{Y}_k)$, where X_k is the health state and $\mathcal{Y}_k$ is multiple observations vector at time k. This adaptation is explained in the forward-backward algorithm section in this chapter. Because of the similarities between these two algorithms a similar procedure is followed to integrate multiple outputs in the Viterbi algorithm.

Basis: $\gamma(X_1) = P(X_1, \mathcal{Y}_1)$

Maximization of the recursion: $\gamma(X_k) = max_{(X_{1:k-1})} P(X_{1:k}, \mathcal{Y}_{1:k})$

$$\gamma(X_k) = max_{(X_{k-1})} P(\mathcal{Y}_k|X_k)P(X_k|X_{k-1})\gamma(X_{k-1}) \quad (27)$$

Integrating multiple inputs

In this step, the variable U is introduced in the Eq.27 as $P(X_k|X_{k-1}, U_{k-1})$ for selecting each of the transitions based on operating conditions according to the given input.

No transition is considered before the initial state, therefore the input U is initiated at time $k = 2$ as U_{k-1}. Then, Eq. 27 becomes:

$$\gamma(X_k) = max_{(X_{k-1})} P(\mathcal{Y}_k|X_k)P(X_k|X_{k-1}, U_{k-1})\gamma(X_{k-1}) \quad (28)$$

Integrating a backward computation

The classical Viterbi algorithm computes the max path by the default formula $\omega(X_k)$ (Eq. 28) which computes the max path through a forward pass. In this section, we extend the algorithm for computing the max path through a backward pass along with the forward pass. The modification ensures to avoid the misleading computation of state distribution over the given sequence.

The modification is done in three phases:

1. A state distribution $P(X_k|\hat{\Lambda})$ is generated given the observations $\mathcal{Y}_{1:k}$ and the state distribution $P(X_{k-1}|\hat{\Lambda})$.
2. The method updates the previous state-distributions $P(X_{k-1:1}|\hat{\Lambda}, X_k)$ based on the generated $P(X_k|\hat{\Lambda})$ and the given observations $\mathcal{Y}_{1:k-1}$.
3. Finally, it updates the state distribution $P(X_k|\hat{\Lambda})$ again by using the updated state distribution $P(X_{k-1}|\hat{\Lambda})$ and the given observations $\mathcal{Y}_{1:k}$.

The state distributions are generated using the adapted Viterbi algorithm by following these three phases.

- Basis: $\delta\ (X_K = s_i) = (1; 1; \ldots ; 1)$

  K is final index (end of the sequence), s_i is hidden states.

Viterbi Algorithm for IOHMM

- Recursive:

$$\delta(X_k) \ = max_{(X_{k-1})}\ P(\mathcal{Y}_k|X_{k+1})P(X_{k+1}|X_k, U_k)\delta(X_{k+1}) \tag{29}$$

The evaluation of the Viterbi is computed by multiplying Eq. 28 and Eq. 29.

$$max\ P\big(\mathcal{Y}_{1:k}|\hat{\Lambda}\big) = \gamma(X_{1:k})\ \delta(X_{1:k}); \tag{30}$$

This is the final version (Eq. 30) of the adapted Viterbi algorithm that computes the maximum distribution for each of the hidden states under the consideration of multiple inputs and multiple outputs.

The Viterbi algorithm computes the max path with the state probability of $P\big(X_{1:k}|\mathcal{Y}_{1:k}\big)$ where the current health state probability $P\big(X_k|\mathcal{Y}_{1:k}\big) = \gamma(X_{1:k})\ \delta(X_{1:k})$ also exists, which can be extracted as the health diagnostic given the observation $\mathcal{Y}_{1:k}$ which is used to prognostic system health.

4.2 Prognostic: RUL prediction

The prognostic is an estimation of future health conditions based on the current health state given by the hidden states and the future operating conditions. Two techniques are used to predict the RUL of the system. The first one is the meantime RUL by using a cumulative summation formula (a numerical integration) with monte Carlo simulation and the second one is a direct computation.

- **The first technique:** numerical integration

The mean value of RUL is defined as the mean time between the current time and the first time reaching the final state (absorbent state). The RUL can be computed with future inputs given operating conditions (case 2) or without inputs when the operating condition is not given or unknown (case 1).

Case 1 : The expected RUL at time k when there is no input is given can be estimate by the following formula:

$$RUL_k = \sum_{t=k+1}^{+\infty} \{(1 - P(s_m)) * (t + 1) - (1 - P(s_m)) * (t)\} \tag{31}$$

here k is the final index of the given sequence assumed as the current time, $P(s_m)$ is the probability of being in the absorbing state for $k + 1 \leq t < +\infty$. The computation stops until the changes between the two results converge.

Case 2: The expected RUL with a given input sequence. In this case, the operating condition for the future operation is known. So according to a given input sequence $U_{k+1:+\infty}$, the formula would be:

$$RUL_k = \sum_{t=K+1}^{+\infty} \{(1 - P(s_m|U_k)) * (t + 1) - (1 - P(s_m|U_k)) * (t)\} \tag{32}$$

Since the given sequence comes with a fixed length there is possibility that the system does not fails but the sequence get finished. In that case, the model repeats the sequence and continues the operation until the two consecutive results does come into a given threshold.

Eq. 31 and Eq. 32 is a discrete probability distribution integral formula for computing the meantime from any time k towards infinity. The RUL is predicted applying the probability distribution function (PDF) in which the unknown operating conditions are simulated through the Monte-Carlo simulation along with the weight ($P(U = p)$) of the operating conditions. The PDF requires the current health state as an initial distribution, the HMM, final state to reach and the number of iterations to compute the RUL.

- **The second technique:** matrix computation

The prognostic by an HMM consists of characterizing the moment when the undesirable hidden state or defining an unacceptable level of performance is reached, knowing that the current state is defined by Eq.30. Several techniques can be used for this purpose. A formal calculation using Eq. 33 will give the meantime to reach the unacceptable state which is absorbent.

$$\boldsymbol{MTRUL} = \frac{\boldsymbol{det}\begin{vmatrix} \mathbf{0} & \boldsymbol{P(X_k)} \\ \mathbf{1} & \boldsymbol{A^*} \end{vmatrix}}{\boldsymbol{det}|\boldsymbol{A^*}|} = \frac{\boldsymbol{det}\begin{vmatrix} \mathbf{0} & \boldsymbol{P(X_k)} \\ \mathbf{1} & \boldsymbol{P(X_k|X_{k-1})} \end{vmatrix}}{\boldsymbol{det}|\boldsymbol{P(X_k|X_{k-1})}|} \tag{33}$$

This formula is adapted from the concept of computing the meantime to failure (MTTF) from the Markov chain by (Amiri, 2014) which uses the determinant of the transition matrix.

Here, $\boldsymbol{A^*}$ represents the transition matrix $\boldsymbol{A}$ but without the final state. The parameter for the final state is not considered because the objective of prognostic RUL is to determine the duration between the current time and the time instant when the model first time gets to the final state. The model does not require to find out how long the system stays on the final state for RUL prediction. The probability $\boldsymbol{P(X_k)}$ is the current health state distribution comes from the diagnostic.

4.3 Offline and Online Operation

There are two types of operation which can follow in PHM applications: offline and online operations. The offline operation uses the existing observations and gives the results. This operation does not update the prediction for any new measurements. On the other hand, the principle of online operations is updating the predictions based on new measurements that come from the system.

Offline operation: Offline operation means that data from $\mathcal{Y}_{1:K}$ is known and the model can define the states $X_{1:K}$ given the observations from $\mathcal{Y}_{1:K}$.

Online operation: Online operation is the online prognostic based on the online diagnostic at time instant K (the last information which addressed as the current time) then predict the RUL_k. Each new observation k helps revising the computations.

4.4 Application

For the sake of illustration, two applications are simulated to design a system with multiple operating conditions (A^p) and multiple emitted outputs ($\mathcal{Y}$). The goal is to demonstrate the diagnostic and prognostic methodologies under multiple operating conditions. Two applications are simulated which focus on two important issues of prognostic applications.

- The first application is about diagnostic and prognostic the health state of the system and predicts the RUL under multiple operating conditions.
- The second application demonstrates how the predicted RUL can be managed by controlling the estimated operating conditions.

4.4.1 The first application: Diagnostic and prognostic under multiple operating conditions

This application simulated following the same procedure that mentioned in the previous section (4.1.2). Moreover, the bootstrap method is being used here to learn the model parameters. Since the learning steps are explained earlier, this section skips the model learning and uses the estimated parameters directly to the diagnostic application and continues to the prognostic part.

4.4.1.1 Data simulation

The sampling data are generated for a system while the system is assumed as to have three operating conditions and two outputs. Let us assume the system degradation has three hidden states and the observation symbols are also three without loss of generality. The corresponding transition matrices according to the input modes are:

$$A^1 = \begin{pmatrix} 0.98 & 0.02 & 0 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{pmatrix}$$
$$A^2 = \begin{pmatrix} 0.90 & 0.10 & 0 \\ 0 & 0.96 & 0.04 \\ 0 & 0 & 1 \end{pmatrix}$$
$$A^3 = \begin{pmatrix} 0.95 & 0.05 & 0 \\ 0 & 0.98 & 0.02 \\ 0 & 0 & 1 \end{pmatrix}$$

The model type is chosen as a left-right model because the system degradation does not go back from one state to its previous state. We also put a zero on (1,3) because normally degradation speed goes from 1 to 2 to 3, but this is possible to have some transition from 1 to 3 as well. In that case, it would be a non-zero parameter.

The emission matrices are:

$$B^1 = \begin{pmatrix} 0.90 & 0.08 & 0.02 \\ 0.03 & 0.90 & 0.07 \\ 0.01 & 0.09 & 0.90 \end{pmatrix}$$
$$B^2 = \begin{pmatrix} 0.99 & 0.01 & 0.00 \\ 0.01 & 0.98 & 0.01 \\ 0.01 & 0.02 & 0.97 \end{pmatrix}$$

Initial state distribution: $\pi = (1 \quad 0 \quad 0)$, the system starts from a good health.

About 100 complete data sequences are generated as the training set, and another 100 sequences as the testing set. After that, the data set is used in IOHMM training through the bootstrap method for estimating the parameters with a 95% confidence interval.

4.4.1.2 Estimated parameters

The IOHMM learns three models based on three operating condition modes applying the bootstrap method. The model estimates the transition parameters as well as the emission and the initial parameters.

The estimated transition matrices are:

$$\hat{A}^1 = \begin{pmatrix} 0.9781 & 0.0219 & 0 \\ 0 & 0.9917 & 0.0083 \\ 0 & 0 & 1.0000 \end{pmatrix}$$
$$\hat{A}^2 = \begin{pmatrix} 0.9129 & 0.0871 & 0 \\ 0 & 0.9506 & 0.0494 \\ 0 & 0 & 1.0000 \end{pmatrix}$$
$$\hat{A}^3 = \begin{pmatrix} 0.9429 & 0.0571 & 0 \\ 0 & 0.9706 & 0.0294 \\ 0 & 0 & 1.0000 \end{pmatrix}$$

here, $\hat{A}^1$ represents the low stress model and $\hat{A}^2$ as high stress. These matrices are constructed from the confidence intervals mean values.

The estimated emission matrices are:

$$\hat{B}^1 = \begin{pmatrix} 0.9070 & 0.0930 & 0 \\ 0 & 0.9350 & 0.0650 \\ 0 & 0 & 1.0000 \end{pmatrix}$$

$$\hat{B}^2 = \begin{pmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{pmatrix}$$

The estimated initial distributions are:

$$\hat{\pi} = (1.0000 \quad 0.0000 \quad 0.0000)$$

4.4.1.3 Diagnostic

The testing data sequences for one input, and two outputs are randomly selected from the test set (shown in the first two graph of the Fig. 23) and used to demonstrate the offline and the online diagnostic performances (shown in the last two graph of the Fig. 23). We can see the difference between the two results. The online prediction is unusual at the beginning when the model has a few data, but the prediction becomes good when the model gets more data later.

After that, the same sequence is cut down at time instant $k = 130$ to predict the expected $RUL_{k=130}$ at assuming the current time $k = 130$.

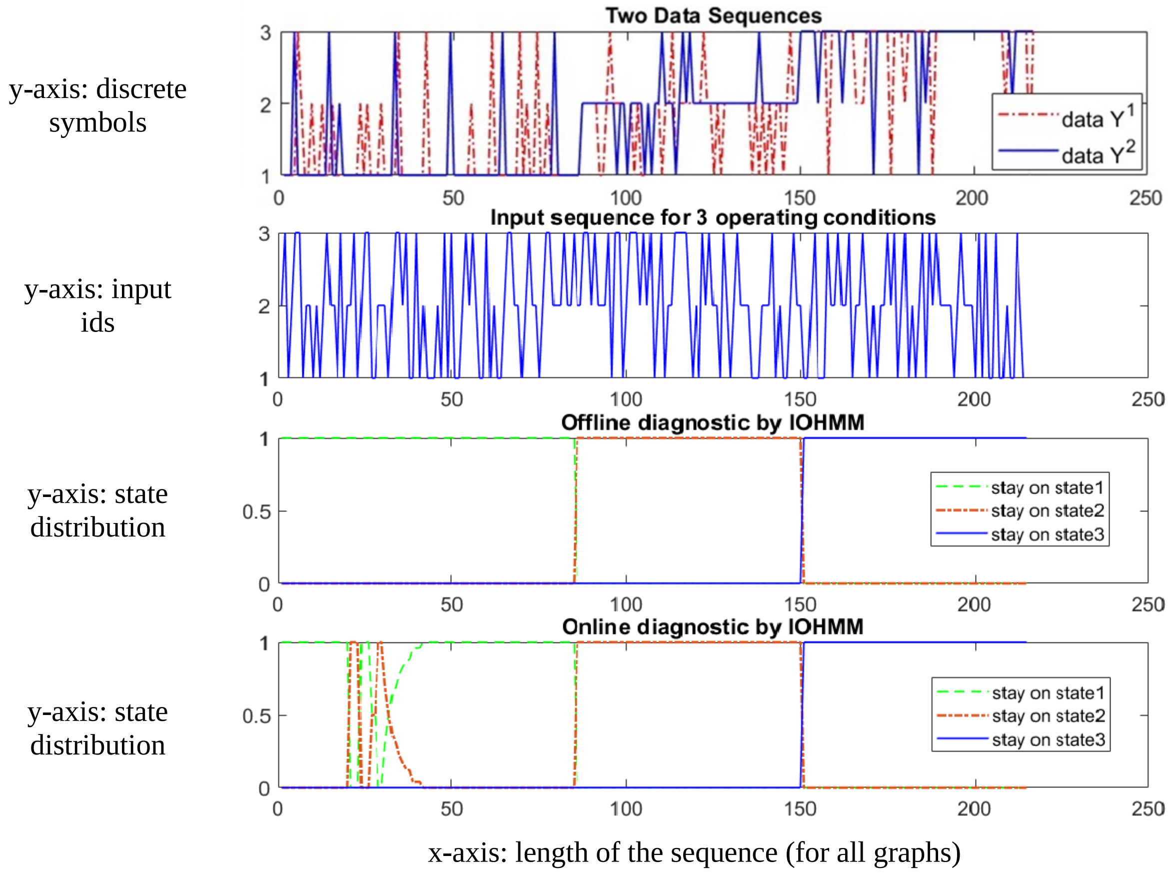


Fig. 23: Diagnostic over the time from starting point to breakdown

Degradation level of the system health can be obtained from the estimated max path. The current health state of the system at time $k = 130$ is estimated as the distribution of $P(X_{k=130}) = (8.7 \times 10^{-149}; 2.7 \times 10^{-25}; 0)$. These are the raw values from the Viterbi calculation. The diagnostic is defined by scaling the result to 1 such as $(0;\ 1;\ 0)$ following the maximum value in the distribution. The result denotes that the system is at state 2 (partially degraded). This information is required in the next step in the application: the prognostic.

4.4.1.4 Prognostic

The prognostic usually depends on the diagnostic, but in this section, a couple of examples are given to explain that the prognostic not only depends on the diagnostic but also on the future operating conditions. By means of which, it depends on how the system is going to be operated in the future. Systems can have multiple operating conditions with several modes. In this section, a discussion about one input with multiple modes is given. The aim is to predict the RUL of the system based on the current health state ($P(X_K)$) and future operating conditions.

There are two possibilities for the future operating conditions. Either it is given or unknown. However, this chapter gives two solutions for these two cases:

1. Prognostic for the unknown input sequence
2. Prognostic for a known input sequence

Prognostic for the unknown input sequence

In this case, the diagnostic is estimated according to the given observations and input sequences but the operating condition for future operation could be unknown. Therefore, the probable operating conditions are simulated by Monte-Carlo simulation by using the weight of the operating conditions $P(U = p)$ in the training set. The same formula was used to calculate the weights for the missing data is used here as well:

$$P(U = p) = \frac{C_p\ (count\ of\ p\text{th matrix})}{the\ number\ of\ elements\ of\ all\ the\ sequences\ in\ U}$$

here $P(U = p)$ is the weight of using the p^{th} matrix over the inputs:

In the illustration, the highest and the lowest stressed operating conditions are represented by two transition matrices $\hat{A}^2$ and $\hat{A}^1$. The weight of these operating conditions can be calculated from the input sequences used in model training. Since the future operating conditions are not certain, one possible way to solve this problem by assuming the system will be operated by the similar operating conditions as previously used. Therefore, the operating conditions are simulated according to their weights by applying the Monte Carlo simulation. The weight is calculated following the Eq. 34 and uses it into the future state evolution for each time instant following the Eq. 35.

$$R\widehat{A}^p = \frac{C\widehat{A}^p}{the\ length\ of\ inputs} \tag{34}$$

here $R\hat{A}^p$ is the weight ratio of operating condition $\hat{A}^p$, p is the id of the operating condition, and $C\hat{A}^p$ is the count of the operating condition triggered in the input sequences.

$$P(X_{k+1}) = \sum_{\boldsymbol{p=1}}^{P} P(X_k)P\left(X_{k+1}|X_k, \widehat{A}^p\right) * R\widehat{A}^p \tag{35}$$

Figure 24 presents the meantime RUL which is computed using all the estimated parameters and the current health state distribution. This method uses Eq. 35 to estimate the meantime RUL, which is about 79 days (time unit is considered as day). The upper and lower limits are obtained corresponding the fixed inputs as lower and higher stressed models.

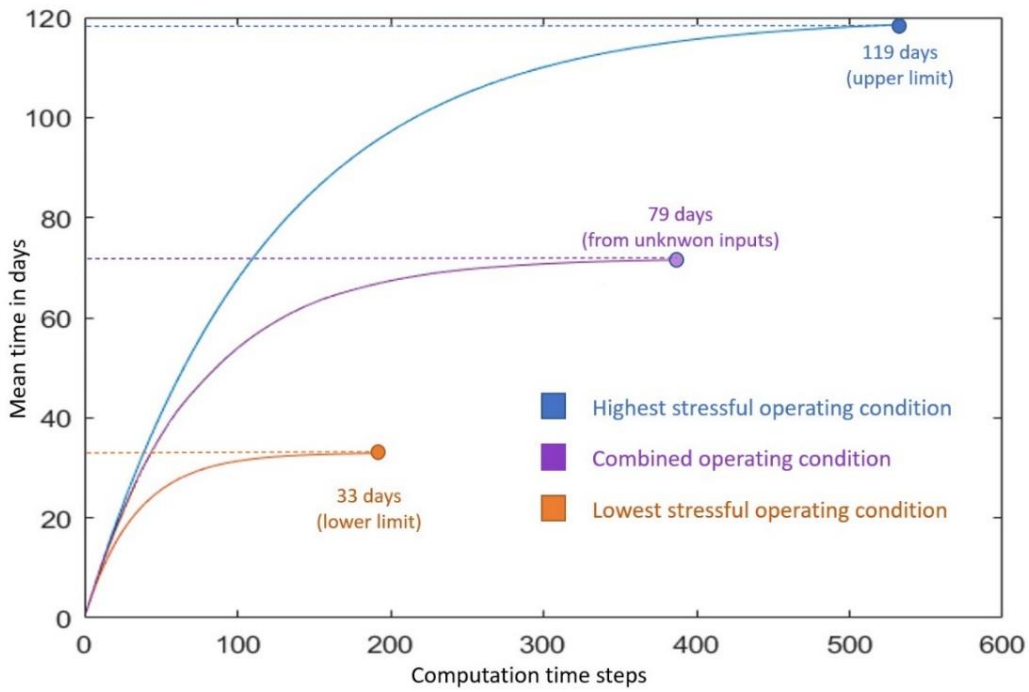


Fig. 24: Mean time RUL for unknown inputs

The low stressed model ($\hat{A}^1$) is defined to have the weight probability as $R\hat{A}^1 = 100\%$ and the high-stressed model ($\hat{A}^2$) as $R\hat{A}^2 = 0\%$. In this case, the estimated RUL is the highest (about 119 days). After that, the method defines the weight of the high-stressed model ($\hat{A}^2$) as $R\hat{A}^2 = 100\%$ and the low stressed model ($\hat{A}^1$) as $R\hat{A}^1 = 0\%$. In this case, the estimated RUL is the lowest (about 33 days). These two results (119, 33) represent the bound of the RUL. So, all possible RUL falls into this range (33-119 days) since the process (Eq. 19) is monotonous.

Remarks: the results are presented in a probabilistic point of view in which the RUL is predicted through a probability distribution function in order to handle RUL prediction uncertainty.

The bounds are illustrated to represent the nearest breakdown with high pressured operating conditions (minimum RUL) and low pressured operating conditions (lowest pressured) (maximum RUL). This information is useful for regulating future operations to delay the possible breakdown point. An example of the given input sequence is discussed in the next section.

Prognostic for a known input sequence

This a situation when the future operation conditions are known. An input sequence would be given to prognostic system health. The given input sequence has a fixed length which means a fixed time of (future) operation, but the principle of estimating the meantime RUL is quite different. As mentioned earlier, a system breakdown can happen anytime. It cannot be said that the system will get into the final state according to the given length. So, the RUL cannot be computed in a fixed time length. In this case, a similar solution can be proposed following the Eq. 32.

The only difference is assigning the weight-ratio of operating conditions $R\hat{A}^p$ which is computed from the given input sequence for future operations, not from that sequence used in model training. So, the ratio is now different which gives a different meantime RUL of 67 days (see Fig .25) while for unknown input, it was 79 days.

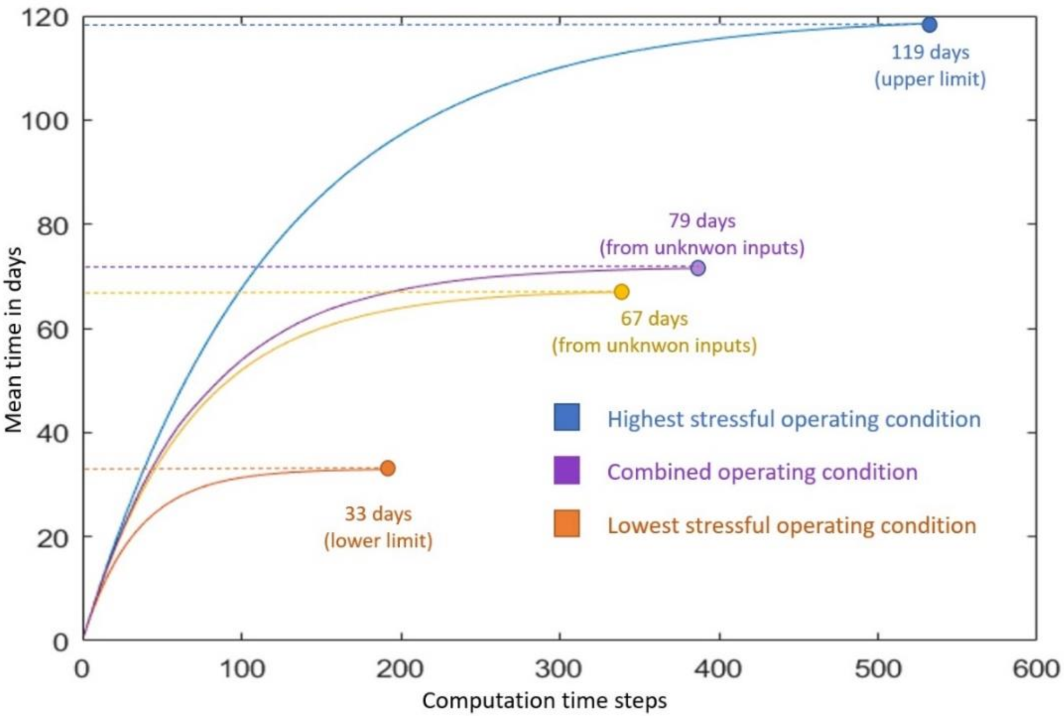


Fig. 25: Result for given input sequence

4.4.1.5 Conclusion

This application uses the IOHMM model to estimate the RUL under multiple operating conditions. The model training has done through the bootstrap method applying the adapted Baum Welch and forward-backward algorithms. Then, the estimated model is used in diagnostic and prognostic system health to demonstrate how the RUL can be estimated considering the uncertainties in the degradation process. A new concept of forward-backward Viterbi algorithm is proposed to diagnosis the system health. Prognosis estimation and the meantime RUL are computed by considering the unknow operating conditions.

4.4.2 The second application: Managing the RUL

RUL changes during the operation of a system because of several dynamics of operating conditions. As the high-stressed condition reduces RUL, the low stressed condition makes the system lasts long. If a system has multiple operating conditions with different varieties of operating stress then, the system degrades in different dynamics. Sometimes, the system degrades typically sometimes not. A high degradation can happen when the system increases the stress of the operating condition. There is a relation between the degradation speed and the operating conditions. So, by controlling the operating conditions, we can manage the production speed as well as the degradation speed. This application subsection illustrates the RUL management by online assessment considering multiple operating conditions.

The graphical representation of online RUL assessment is shown in Fig.26. The IOHMM takes the same input U_{k-1} of the system and the corresponding output $\mathcal{Y}_{1:k}$ to diagnostic the health state $\hat{X}_k$ at the current time k. After that, it estimates the $R\widehat{U}L_k$ and the reference manager (RM) compares it with the target RUL to decide the next input U_k to the system (cf. Fig. 26).

The RM applies the algorithm (Algo 3) to manage the input for matching the target RUL.

If the estimated RUL is less than the target RUL at time k then, the system should be operated with (comparatively) low-stressed condition at $k+1$. Noted that, if there are several low-stressed operating conditions available then the RM selects the one that produces maximum production. The algorithm is specially designed not only to match the target RUL but also to maximize production.

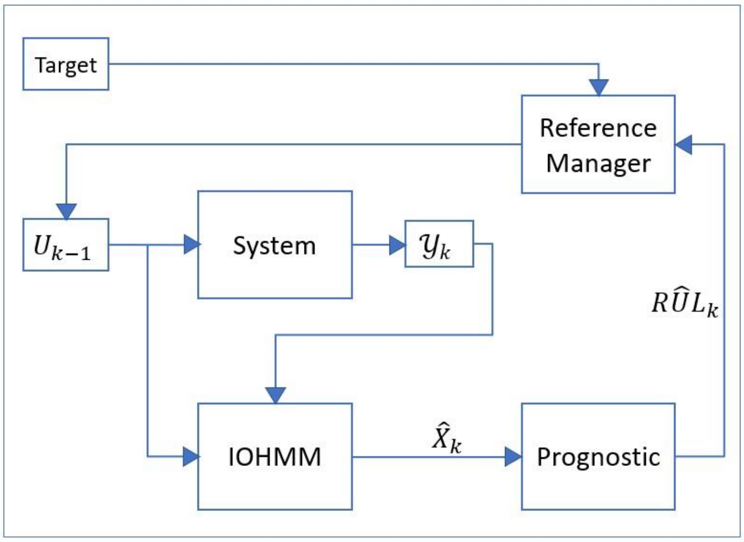


Fig. 26: Online RUL management

However, if the estimated RUL is greater than the target RUL then, the next operating condition selects comparatively a high stress model, otherwise it selects the low stress model. For example, if the target RUL stands between estimated RUL by using the model A^1 and A^2 then the RM selects the low stressed model (A^1) until the target RUL gets into the next part: estimated RUL from A^2 and A^3. In this case, the RM selects comparatively the low stress model (A^2) and continues the process to match the given RUL until the model gets to the breakdown state.

Algo 3: RUL Managing Algorithm: Reference manager RUL matching

```
function()   [RUL matching with a given time]
begin
   targetRUL ← intial value;          // given time
   inputTrack ← initial array;        // save the conditions
   k ← initial values;                // current time (in days)

   [first transition at time (k)]
   newTransition ← use (A1,A2,A3) with weight;
   [the first input]
   inputTrack(k) ← model-W //all model with weight;
   [RUL at current time]
   RUL(k) ← according to the newTransition;

   while(RUL>0)          // runs till the breakdown
         if (targetRUL<RUL(A1) && targetRUL>=RUL(A2) )
               newTransition ← select the model A1;
               inputTrack(k+1) ←  selected model;
         elseif (targetRUL<RUL(A2) && targetRUL>=RUL(A3) )
               newTransition ← select the model A2;
               inputTrack(k+1) ←  selected model;
         else
               newTransition ← select the model A3;
               inputTrack(k+1) ←  model-W ;
         end if
   k ← k + 1;       // represented in days
   RUL(k) ← according to the newTransition;
   targetRUL ← targetRUL - 1;    // run once in a day
end function
```

4.4.2.1 Data preparation

This application is simulated to represent a system that has three operating conditions (inputs) and one output. It allows us to manage the RUL by managing the operating conditions using by the estimated IOHMM. Each of the data sequences assumed to be started in a good health state and finished at a breakdown state.

The parameters used in the simulation are:

Transition matrices:

$$A^1 = \begin{pmatrix} 0.99 & 0.01 & 0 \\ 0 & 0.95 & 0.05 \\ 0 & 0 & 1 \end{pmatrix}, A^2 = \begin{pmatrix} 0.98 & 0.02 & 0 \\ 0 & 0.94 & 0.06 \\ 0 & 0 & 1 \end{pmatrix}, A^3 = \begin{pmatrix} 0.99 & 0.01 & 0 \\ 0 & 0.90 & 0.1 \\ 0 & 0 & 1 \end{pmatrix}$$

Two data sets (train set and tests set) are generated for estimating the model parameters and testing the model characteristics .

Emission matrices:

$$B^1 = \begin{pmatrix} 0.99 & 0.01 & 0 & 0 \\ 0.30 & 0.70 & 0 & 0 \\ 0.01 & 0.80 & 0.15 & 0.04 \end{pmatrix}$$

Initial state distribution: (assumed as in good health)

$$\pi = (1 \quad 0 \quad 0)$$

4.4.2.2 Results

Estimated Parameters

IOHMM learns three transition matrices according to three operating conditions.

The estimated matrices are:

$$\hat{A}^1 : \begin{pmatrix} 0.9923 & 0.0077 & 0 \\ 0 & 0.9478 & 0.0522 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\hat{A}^2 : \begin{pmatrix} 0.9904 & 0.0096 & 0 \\ 0 & 0.942 & 0.058 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\hat{A}^3 : \begin{pmatrix} 0.9901 & 0.0099 & 0 \\ 0 & 0.909 & 0.091 \\ 0 & 0 & 1 \end{pmatrix}$$

The model also learns the emission matrix:

$$\hat{B}^1 : \begin{pmatrix} 0.9998 & 0.0002 & 0 & 0 \\ 0.2951 & 0.7049 & 0 & 0 \\ 6.4\mathrm{e}^{-31} & 0.8161 & 0.1436 & 0.0402 \end{pmatrix}$$

The estimated initial state distribution:

$$\hat{\pi} = (1 \quad 0 \quad 0)$$

Diagnostic and prognostic

A random data sequence is selected from the test set and split a prior time earlier than the breakdown point (end of the sequence). The current health states of the system are estimated as $P(X_k)$ =$(7.08 \times 10^{-10} \quad 0.7871 \quad 6.23 \times 10^{-04})$, where k is the current time. The diagnostic is defined as

$(0 \quad 1 \quad 0)$ following the maximum value in the distribution (scaled to 1). It denotes the system health is at state 2 (partially degraded). The estimated diagnostic is used to estimate and manage the RUL.

There are two steps in this part of the application: the offline prognostic which is required to decide either reaching the target RUL is possible or not, and the online prognostic which executes the RUL managing algorithm.

Step 01: Offline Prognostic

The prognostic is performed offline and showed the result at the current time k from the estimated diagnostic $P(X_k)$.The challenge comes when the system does not have the operating conditions for future operations to manage the model switching. That means there is no information about the switching of operating conditions during the system runtime. However, at least four different RULs at the current time can be computed based on the available information. Three of them ($A^1 = 159\ days, A^2 = 131\ days,$ $A^3 = 122\ days$) come from the estimated models separately (see Table 8), where A^1 is the lowest stressed operation, A^2 is the medium stressed operation and A^3 is the highest stressed operation.

Table 8: Different RULs

No	Model Name	Estimated RUL
1	$\hat{A}^1$	159 days
2	$\hat{A}^2$	131 days
3	$\hat{A}^3$	122 days
4	Previous Conditions	147 days

The IOHMM also computes the RUL (147 days) using the existing operating condition which is used in the training dataset. This is applicable when the system does not require to change its operation but simply follows the same operating condition that is used from the beginning. This table shows several possibilities for the system being alive according to different operating conditions.

The bound can be defined as [122-159 days]. If the target RUL is inside these limits, then the managing algorithm proceeds to execute. The target RUL is the time that the system should reach before it goes into the breakdown state. In this application, the target RUL is set as 150 days which is inside the bound. So, RUL management can apply to match the date.

Step 02: Online Prognostic

There are two different techniques that can be followed to manage the operating conditions online to match the target RUL. One is simulating the future operating condition by Monte Carlo simulations considering the weight of operating conditions. Another one is to use the proposed RUL managing algorithm (Algo. 03). This algorithm manages operating conditions by switching them to match the predicted RUL with the target date. This algorithm intends to use the highest stressed operating conditions until it covers the target value instead of using the weight of the models. It is more realistic in the sense of following the operation on the real system.

Figure 27 represents online estimated RUL from the new measurements coming from the system. Three operating conditions and the previous operating condition are used separately to predict the RUL online. Whenever a new measurement comes IOHMM diagnostic the current health and uses the state distribution to estimate the RUL. The computation continuous until the system gets into the breakdown state. The first evolution in the figure comes from the lowest operating condition represented by a matrix A^1 which is the highest limit for the online RUL prediction.

Similarly, A^3 produces RUL with the minimum limits because it is the highest stress model. Essentially, all combinations of the operating condition should estimate the RUL that stays over this limit. For example, A^2 and the previous input condition provide the RULs that stand inside the limit. Noted that, both the horizontal and vertical axis are represented as days.

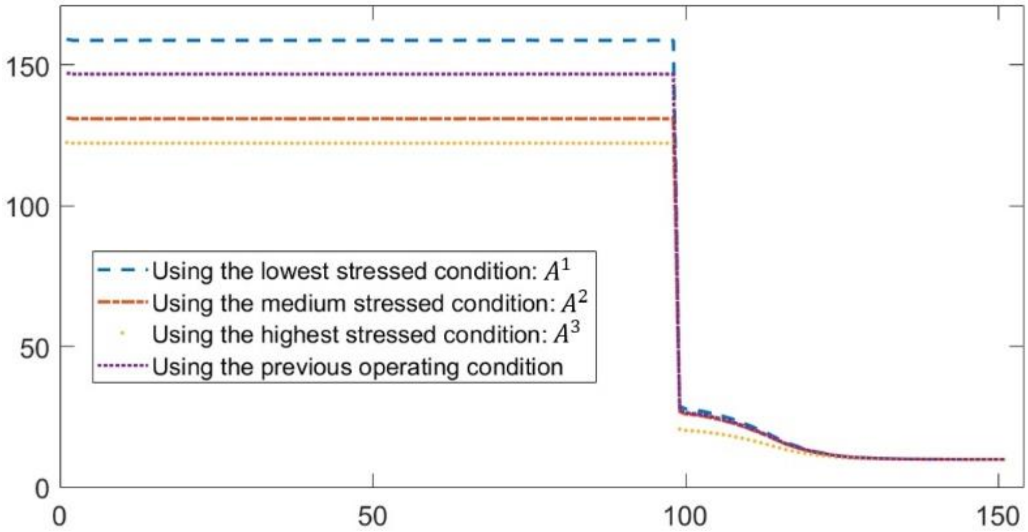


Fig. 27: Online RUL of several operating conditions [y-axis: days; x-axis: days]

This figure explains the evolution of the RUL given the time and the input mode. For instance, at time 50 if the input mode is 3 until the breakdown then the RUL is 122 days whereas if the input mode is 1 until the breakdown then the RUL is 159 days. At time 51, the diagnostic and the RUL are revised according to the new observations and so on until the breakdown.

As mentioned earlier, the simulated data sequences end up at breakdown state. That is why different operating conditions estimate a similar RUL at the end of the sequence where the system is really close to the breakdown time. Even though the estimated RULs are different at the beginning of the sequence, but they are intended to finish at breakdown state when the measurements indicate the probable breakdown state.

Managing RUL is an extended process of an online RUL estimator where the operating conditions change at each time instant to get one step closer to the target RUL. Figure 28 represents the result of applying the RUL management algorithm to test the model performance to match the target RUL (150 days). The model predicts the RUL at each time instant *k* and compares if the RUL reaches 150 days or not. If the estimated RUL does not reach the target then, the model applies the lowest stressed operating conditions to increase the probable RUL and cross over 150 days. Noted that, if the RUL shows more than 150 days then, the highest operating conditions can be chosen to increase the production speed. This is how the operating conditions can be switched between the lowest and highest stress to manage the RUL and match the target date. For example, whenever the target RUL gets closer to one of the three estimated RULs, the reference manager changes the operating condition to the corresponding once.

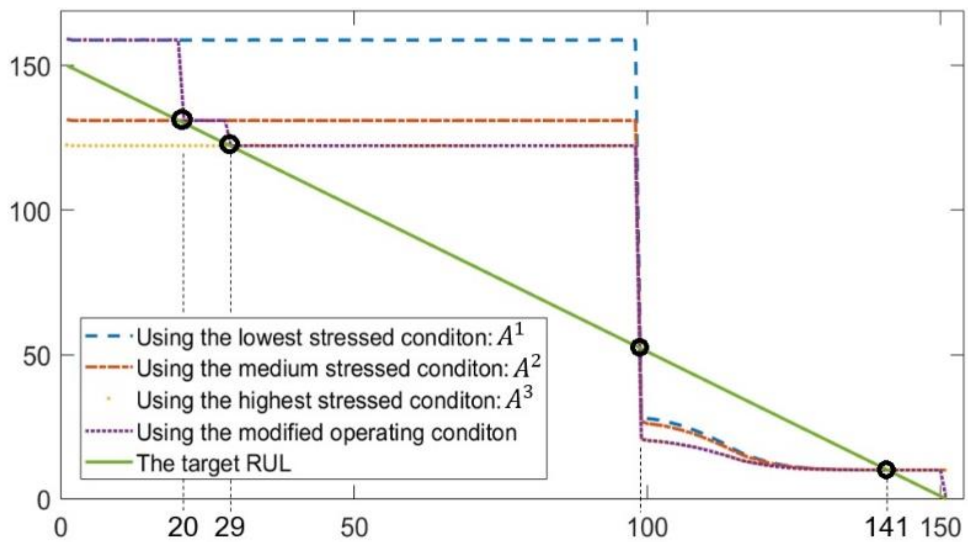


Fig. 28: Online RUL matching with the target RUL [y-axis: days; x-axis: days]

Figure 28 highlights four indexes (20, 29, 98, 141) between the starting point to the end where the target RUL indicates to change the operating conditions. However, the model changes the operating condition only twice: from the lowest condition to the medium condition at index 20 and from the medium condition to the highest condition at index 29. The model did not change the operating condition at index 98 and 141 because all three estimated RULs at these two indices are almost the same which get into the breakdown state at the same time.

This algorithm can be modified to use instead of low production for a high cost as well. Either way, the user would be benefited to match any target RUL in the limit. This could be such an important use for rescheduling the maintenance window of a system under multiple operating conditions.

4.4.2.3 Discussion

RUL assessment has been the subject of extensive studies to determine its performance reliability, production safety, system maintenance, etc. It becomes encouraging interest in condition-based maintenance (CBM) (Do, 2015, Hong, 2014) and prognostic and health management (PHM) (Lee, 2014, Esteves et a. 2015). Online RUL assessment gets in-depth research for a decade, now it is a growing interest in monitoring the online health condition and the production safety of the system (Niu, 2017). Many industrial domains are putting high importance on the recursive RUL assessment for reliability and cost reduction of system maintenance.

Matching the maintenance date with the estimated RUL would be a good solution to reduce the maintenance cost (Khelif, 2014). The proposed method allows us to predict the RUL considering the operating conditions separately (Fig. 27) which lets the model decide the next operation for reaching the target. The reference manager (see in Fig. 26) compares the prediction with the target at each time instant to decide the next operating condition for immediate time instant. The reference manager handles the uncertainty of changing the RUL which can be different in each time instant. The RUL can move in a different time (forward or backward) compared to the target for several reasons. That is why the proposed model continuously tries to get informed about the system's health by diagnosing online. RUL management mostly relies on the current health condition; therefore, the degradation assessment should be updated when a new measurement comes into the analysis (Zhou, 2018). Degradation is not reversible and not directly measurable online, so this model analysis the observation of the system performance online which is used to model the degradation.

To assess the non-measurable degradation of the system by using observations, the proposed model is a fitting match. This model can be useful to schedule the maintenance window according to any given date that stands in the RUL prediction bound.

4.5 Conclusion

Maintenance scheduling is a complex task due to the uncertainties of system degradation. System degradation itself is a complex process that includes multiple uncertainties (data uncertainty, model uncertainty, environmental conditions, etc.). It is even more difficult when the system degrades under multiple conditions. A model needs to be designed with such a capability that can handle these uncertainties and predict the degradation with good accuracy. Only good model of the degradation can provide good diagnosis and prediction, which is essential for maintenance planning.

The key issue in scheduling a decent and effective maintenance action is frequently monitoring the health states of the system. In order to monitor the health state of the system, the operating conditions need to be identified and then, the degradation of the system requires to be estimated considering the operating conditions in real time. Finally, if the estimated RUL does not cover the target, it should be adjusted to match the target by managing the operating conditions.

An IOHMM-based model is presented where the proposed methodology identifies the model parameters according to operating conditions. Well-known algorithms (*i.e*. BW, FB, Viterbi) are adapted to train the model and apply for diagnostics of the system in real time to compute the probabilities distribution over the health states of the system. The model updates the diagnostic results based on the observations from the beginning of the life of the system until the last new measurements observed on the system. Afterwards, we propose equations to predict the RUL of the system based on the updated degradation through an online process. A reference manager is demonstrated that compares the estimated RUL with the maintenance window. It manages the operating conditions by switching the operating conditions to keep the RUL at a level that meets the target.

Several applications are simulated in this chapter to demonstrate the adapted algorithms and the prognostic methodologies. The RUL is estimated based on the current health state of a system and the operating conditions. The probable evolution of degradation would follow a similar nature in the nearest past which is hidden information in the observation data. That is why it is intended to estimate the system damage over time from its observed data come from the sensor installed on the system. However, it is difficult to estimate the RUL due to the stochastic nature of deterioration phenomena. Existing solutions deal with high computational complexity, which increases the difficulty in real-time condition monitoring with high accuracy.

Nowadays users want to control a system life cycle for adjusting its manufacture and energy consumption by proactive strategies. Therefore, information about the RUL would be great to deal with it. The RUL estimation can be good if it includes accuracy and precision, which can be done by considering the uncertainties in which the degradation depends on. Even though the measurements are crucial information to know about the hidden degradation but only the observation cannot provide all those hidden issues in the data which are generated from different uncertain sources. It is meaningless to estimate RUL without considering these uncertainties such as the operating conditions. If the operation comes with several dynamics then, it needs to be tracked down for better understanding the system behavior. The number of conditions and the dynamics of usability could be used to diagnostic the system's health state. There is no information about operating conditions or the observations for prognostic, so, another uncertainty needed to be treated about the prognostic without any observations. The uncertainty about the future operating conditions is handled by providing two different solutions for known inputs and for unknown inputs. This model provides a possible limit for the future health transformation of the system from low degradation to high degradation. This method can be used to estimate the RUL of structured system multiple components. Each of the components can be modeled and diagnosed separately then combining their health state together to predict the RUL of the entire system.

Chapter 5

The Third Contribution: Estimating RUL of Aircraft

Table of Contents

5 The Third Contribution: Estimating RUL of Aircraft

In the previous chapter, we have developed two main contributions on PHM by considering inputs in HMM algorithms in order to provide a tool (IOHMM) able to estimate parameters from data, manage uncertainties and diagnose and prognose the RUL of a system modelled by an IOHMM. Some illustrations are provided on a toy system to show the behaviour of our algorithms and the benefits but also the limitations.

This chapter is dedicated to a real application based on an aircraft engine through the dataset from the PHM data challenge 2008. We first describe the dataset then we define the health parameter modelling by IOHMM and finally we discuss the performance evaluation. The methods described here were applied to the 2008 PHM Challenge, an IEEE-sponsored competition to evaluate prognostic models (Le, 2016; Le Son, 2012). The dataset is suitable for tracking and predicting the progression of damage in the system because the data set contains the measurement which starts from a different initial health conditions to system failure. It has 3 input parameters and provides measurements from 21 output sensors (Saxena, 2008). It contains the data corresponding to 218 turbines from the initial moment to the time of failure, which can be used in the learning of the model leads to the construction as $\Lambda = (A, B, \pi, U)$. Another set of data was dedicated to test and evaluate the model performance. It also contains the data from the same turbines but is randomly truncated before failure. These data were created from a simulation of NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) model (Saxena, 2008).

5.1 C-MAPSS

C-MAPSS is a simulating tool used to simulate a realistic large commercial aircraft engine model of the 90,000 lb thrust class and the package includes an atmospheric model capable of simulating operations with (i) altitudes ranging from sea level to 40,000 ft, (ii) Mach numbers from 0 to 0.90, and (iii) sea-level temperatures from –60 to 103 °F (Saxena, 2008). The kit also includes a power management system that allows the engine to be operated over a wide range of thrusts throughout flight conditions. The engine has three high-limit regulators for managing the speed, High-Pressure Turbine (HPT), and the High-Pressure Compressor (HPC). Figure 29 represents the engine with the main elements and Fig. 30 shows the flowchart of how various modules are assembled in the simulation.

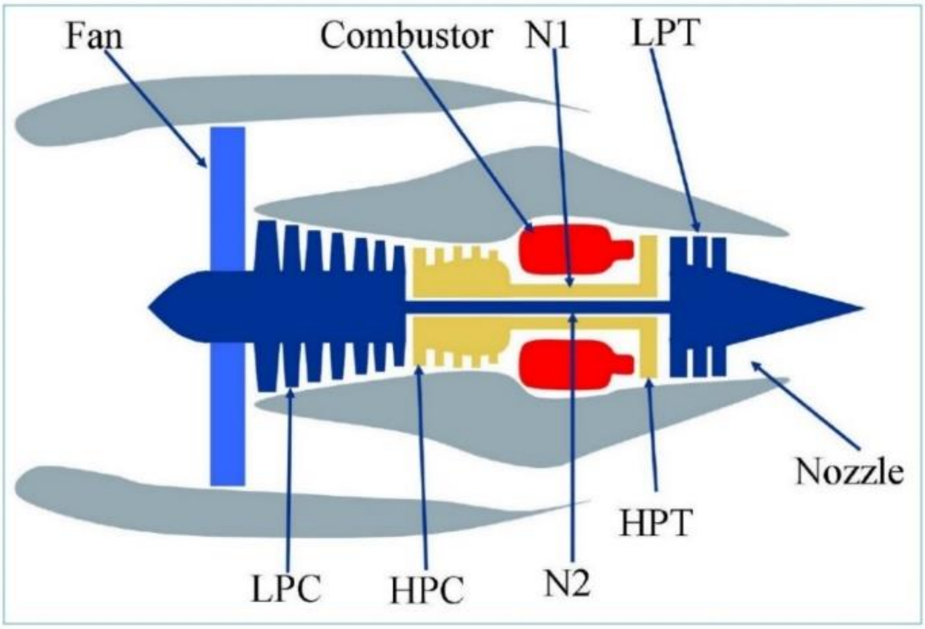


Fig. 29: Aircraft engine simulated for the PHM challenge by 2008

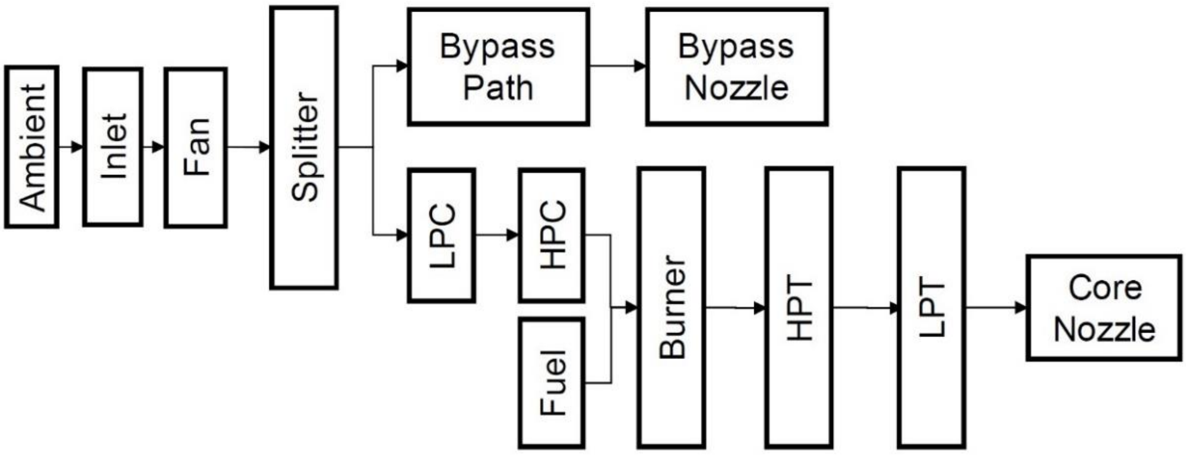


Fig. 30: The different modules and their connections such as modelled in the simulation

This simulator has 14 inputs (Table 9) that allow the user to simulate the effects of component failure and deterioration of the five rotating engine components (e.g., fan, LPC, HPC, HPT, and LPT). The outputs include a variety of sensor responses and operability margins.

Table 9: C-MAPSS inputs to simulate various degradation of the five rotating components

Name	Symbol
Fuel flow	Wf
Fan efficiency modifier	fan_eff_mod
Fan flow modifier	fan_flow_mod
Fan pressure-ratio modifier	fan_PR_mod
LPC efficiency modifier	LPC_eff_mod
LPC flow modifier	LPC_flow_mod
LPC pressure-ratio modifier	LPC PR_mod
HPC efficiency modifier	HPC_eff_mod
HPC flow modifier	HPC_flow_mod
HPC pressure-ratio modifier	HPC_PR_mod

HPT efficiency modifier	HPT_eff_mod
HPT flow modifier	HPT_flow_mod
LPT efficiency modifier	LPT_eff_mod
HPT flow modifier	LPT_flow_mod

Out of the 58 different outputs provided by the model, a total of 21 variables (Table 10) were provided to the participants of the competition. These variables are sensor measurements of temperature, pressure, velocity, etc. for 218 independently identical units (Saxena, 2008).

Table 10: C-MAPSS outputs to measure system response

Symbol	Description	Unit
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	--
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	--
farB	Burner fuel-air ratio	--
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

5.2 Model Structure

Model structure needs to be defined first before training the IOHMM with the dataset. In this section, we define the number of hidden states, the number of observation symbols, the number of operating conditions, etc.

5.2.1 The operating conditions

The important requirement for the degradation modelling process is the availability of a suitable system model that allows input variations of health-related parameters and recording of the resulting output sensor measurements. In the PHM challenge data, there are three input parameters (Altitude, Mach number, and Throttle Resolver Angle) used to set the operating conditions (Le, 2015). The operational conditions for all engines can be clustered into six different regimes (Fig. 31a). The six dots are six highly concentrated clusters that contain thousands of sample points each (Saxena, 2008). C-MAPSS simulated the data through these 6 different operating conditions at altitudes ranging from sea level to 42K, Mach numbers from 0 to 0.84, and Throttle Resolver Angle (TRA) from 20 to 100 (see Table 11).

As mentioned in chapter 4, different models can be estimated according to the operating conditions, which provide different dynamics of the degradation process. Even if IOHMM model can handle any input conditions, the more mode we have, the more parameters should be estimated. We also know that the estimation accuracy depends on the quality of data and on its amount.

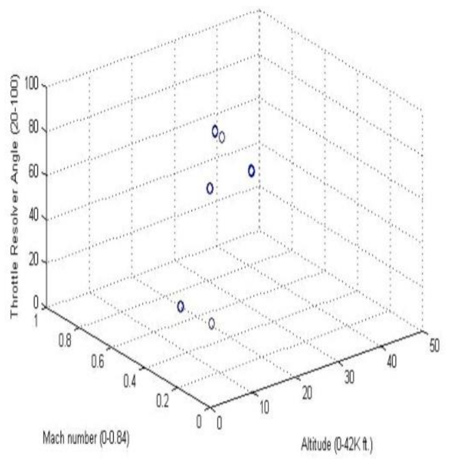


Fig. 31a: Six conditions (Saxena, 2008)

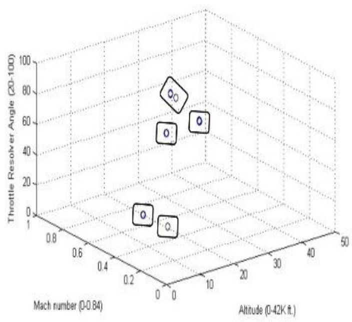


Fig. 31b: Five conditions

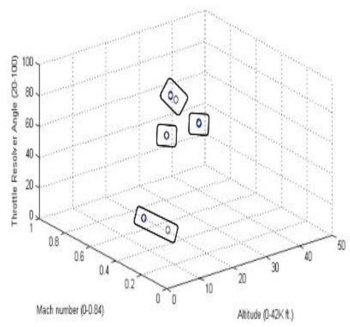


Fig. 31c: Four conditions

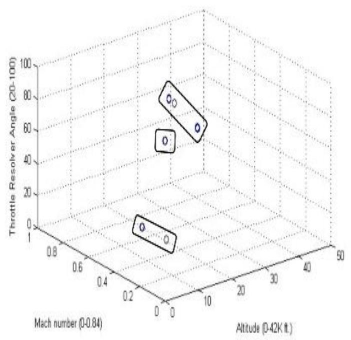


Fig. 31d: Three conditions

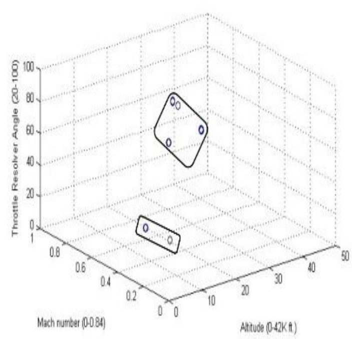


Fig. 31e: Two conditions

Fig. 31: Operating setting of all engines are clustered in different conditions

Here, we considered up to six operating conditions in five groups and observe the model performance in different combinations. Table 11 shows the values of input considerations according to the groups.

Table 11: Different operating conditions

<table>
<tr><th colspan="3">Input parameters</th><th>Group 1</th><th>Group 2</th><th>Group 3</th><th>Group 4</th><th>Group 5</th></tr>
<tr><td>Altitudes</td><td>Mach Numbers</td><td>TRA</td><td>Six Conditions</td><td>Five Conditions</td><td>Four Conditions</td><td>Three Conditions</td><td>Two Conditions</td></tr>
<tr><td>25K</td><td>0.62</td><td>80</td><td>1</td><td>1</td><td>1</td><td>1</td><td rowspan="4">1</td></tr>
<tr><td>20K</td><td>0.70</td><td>0</td><td>2</td><td rowspan="2">2</td><td rowspan="2">2</td><td rowspan="3">2</td></tr>
<tr><td>35K</td><td>0.84</td><td>60</td><td>3</td></tr>
<tr><td>42K</td><td>0.84</td><td>40</td><td>4</td><td>3</td><td>3</td></tr>
<tr><td>20K</td><td>0.25</td><td>20</td><td>5</td><td>4</td><td rowspan="2">4</td><td rowspan="2">3</td><td rowspan="2">2</td></tr>
<tr><td>0 K</td><td>0</td><td>100</td><td>6</td><td>5</td></tr>
</table>

5.2.2 Degradation indicator

To create an IOHMM describing the degradation of the engine, it is necessary to have the indicators of its degradation. Based on the available measurements (up to 21 measurements) nothing can define that each of them contains an indication of the degradation. To identify data that show a potential degradation indication and to reduce the size of data, we applied a Principal Component Analysis (PCA) on the dataset to find the indicators.

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of potentially correlated variables (entities each with different numerical values) into a set of linearly uncorrelated variable values called principal components. Each of the observation sequences gives an indicator of the degradation of the engine. Figure 32 shows the difference between the original data and the PCA results.

These are the scaled representation of the sequences as polynomial fitting from coefficients in a least-squares sense. This fitting and scaling transformation improves the numerical properties of the polynomial and the IOHMM algorithms.

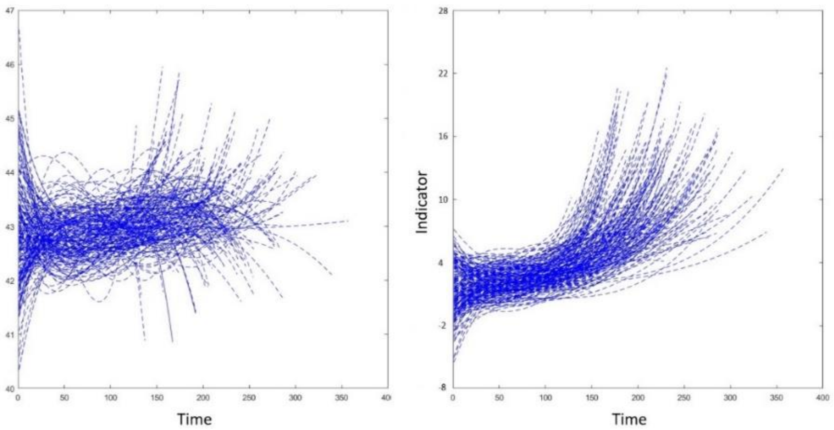


Fig. 32a. Original observations of sensor 11 Fig. 32b. A unit from the PCA results

Fig. 32: Difference between the original data and the PCA results

IOHMM performs well if the observation sequences are in the increasing form or if the observations are significant (different). The increasing form is expected because it means that something changes in a monotonous way (Fig. 32b). The original data (Fig. 32a) is difficult to model as it is not in exponential form.

Figure 33 represents the original data sequences of 21 sensors (/turbines).

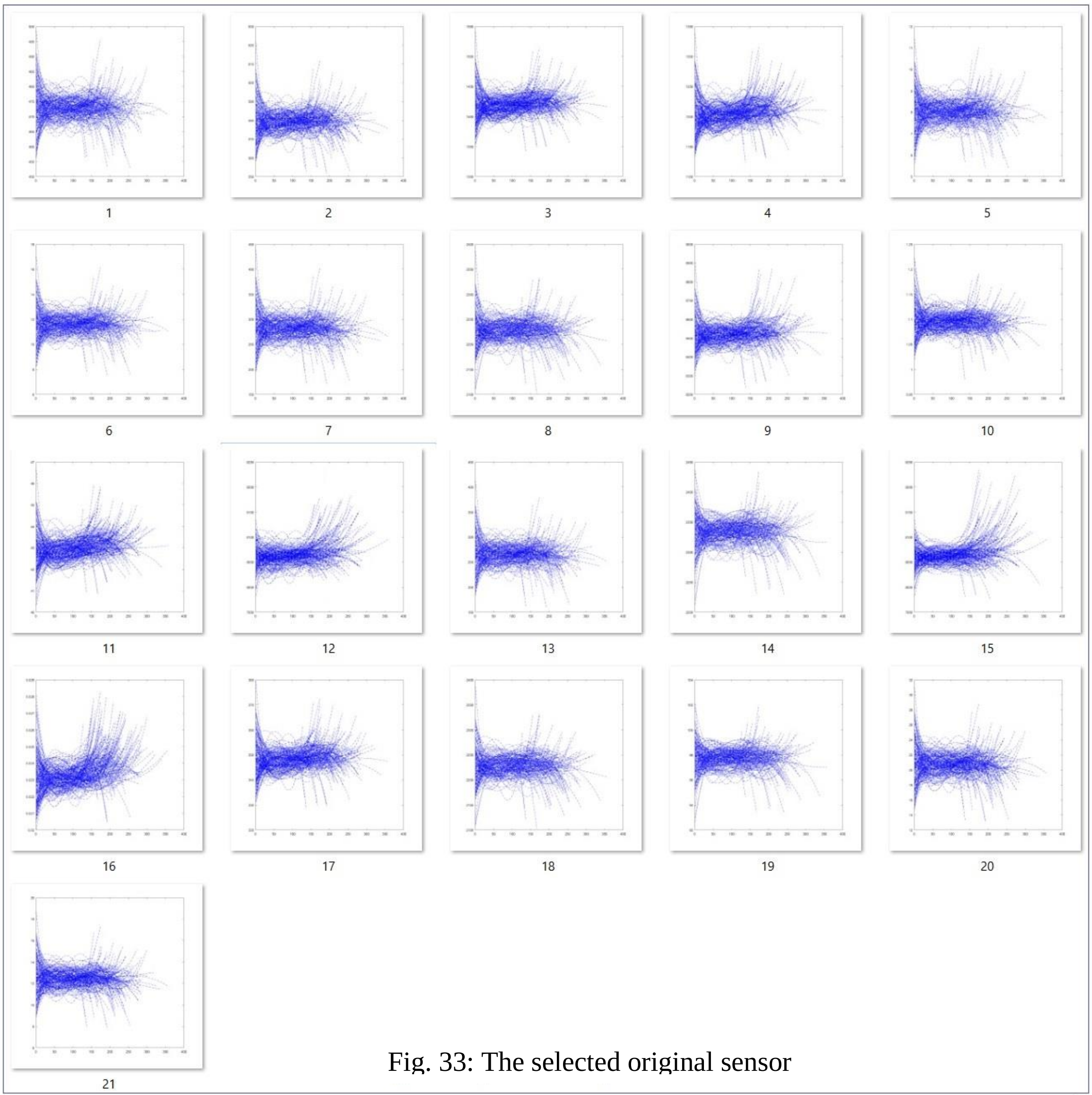


Fig. 33: The selected original sensor

These data are evaluated through the PCA method for having a set of indicators (Fig. 34). Working with all sensors means estimating a lot of parameters which means having a large amount of data. The behavior of some sensor outputs does not look like what we expect (i.e. global increasing/decreasing values). To increase the efficiency of data, we want to reduce the number of outputs that is why the PCA is computed to select significant indicators which are a combination of sensor data. To define an IOHMM for modeling the degradation of the aircraft engine, it is necessary to have the degradation indicators from the raw sensor data. Usually, the data should indicate the degradation of systems as increasing or decreasing graphical view. Otherwise, it is difficult to model the degradation with the IOHMM. For example, the given dataset (Fig. 33) shows the sequences are not monotonic. The behavior of these sequences does not look like what we expect (i.e. global increasing/decreasing values). That is why, to find out the monotonic indicator and to increase the efficiency of data, the PCA to this dataset.

Figure 34 represents the evaluated datasets with a coefficient response from the PCA results. The IOHMM is trained from these datasets and learn the parameters and applied to estimate the remaining useful life of the aircraft engine.

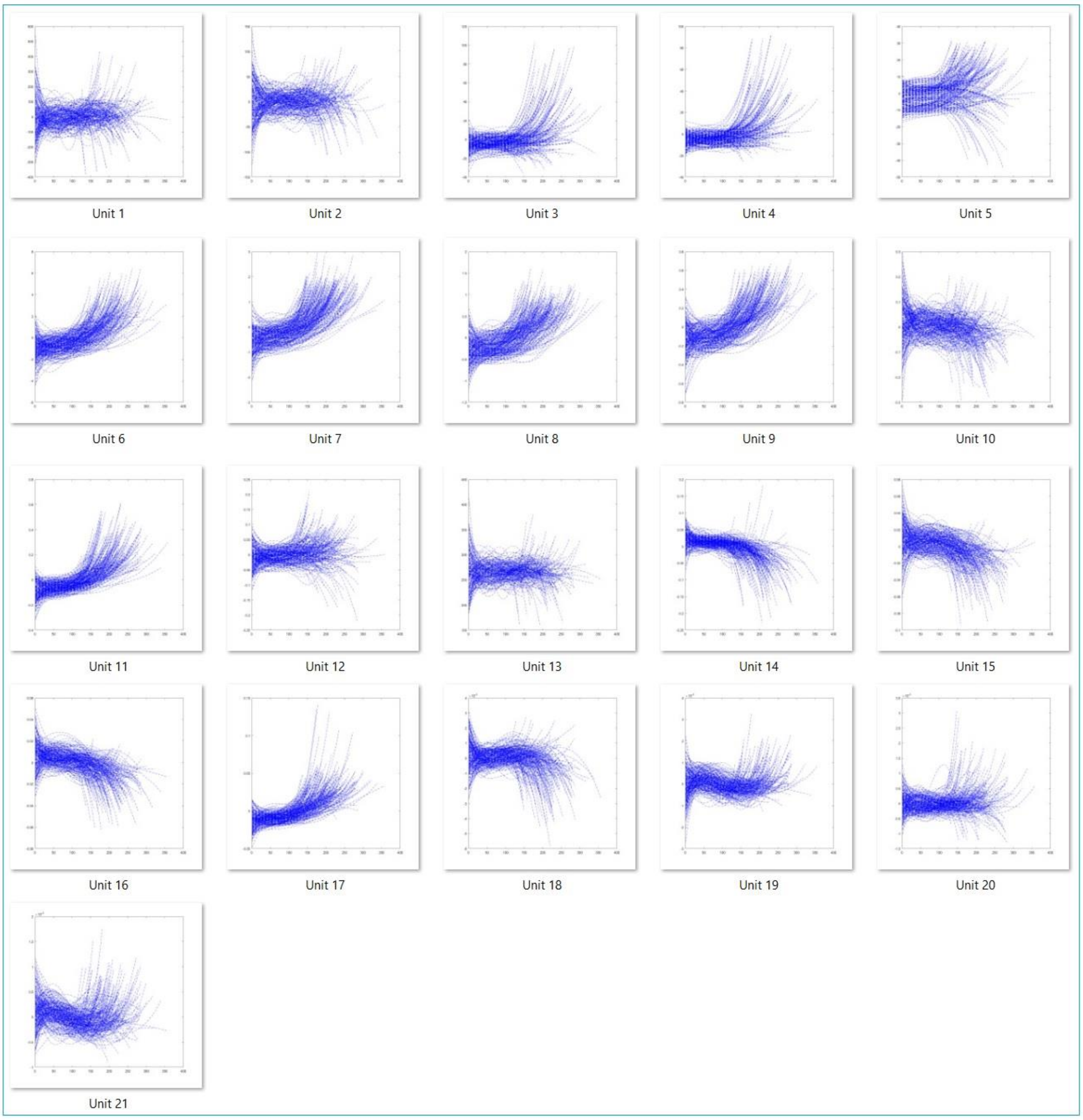


Fig. 34: Measurements after applying PCA

The following sections explain how the identified indicator from an original sequence used to define the hidden states and the emitted symbols. Noted that, the high order of units contains the most interesting coefficients of degradation. Usually, the first order (Unit 1) of PCA results contains the most coefficients that relates the inputs and the corresponding outputs. However, in our objective, we are looking for the relation between the inputs and the degradations that represented by the outputs. This is not exactly the same as usual which can be found in the higher order (i.e. Unit 6-9 or 11 e.t.c.).

5.2.3 Emitted symbols

We decided to use four discrete symbols to classify the indicators come from the observation sequences of turbine 11. A total of 218 indicators are obtained from 218 sequences of this turbine. The classification depends on the data amount and enough information in the data. The thresholds are defined based on the amount of data dedicated to each threshold, because the parameter estimation requires enough data and information. The threshold is adjusted by evolving the best response to the amount of data and the estimated parameters.

This classification is necessary because the IOHMM usually works with discrete data units. Therefore, continuous data sequences are converted to discrete data sequences (Fig. 35).

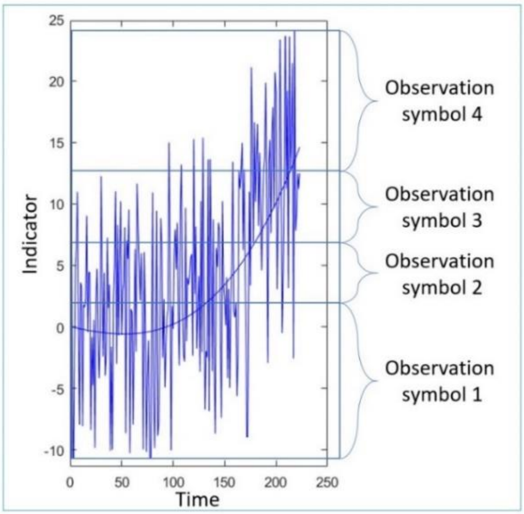


Fig. 35: Defining observation symbols on the indicators

Any one of the units (in Fig. 34) can be used to train the model. However, unit 11 is chosen because it gives the best RUL prediction performance compared to other units.

5.2.4 Defined IOHMM

The model structure is defined in this chapter by describing the inputs, outputs, and health states.

The defined structure of the proposed IOHMM has:

- Data unit: discrete. The continuous data are converted into discrete format in two steps: (1) finding the indicators from the data, and (2) classifying the indicators by different discrete symbols)
- The number of discrete symbols: four. The number of symbols can be increased based on the amount of data dedicated to the threshold of classification.
- The number of outputs: considered one output. This chapter focuses on the real situations to model the system by formatting the raw data, dealing with the operating conditions, and the number of hidden states of the system. One output has 218 data sequences which is enough to demonstrate each part of the model. However, multiple outputs produce better results, but it is a time-consuming process. Once the demonstration succeed multiple outputs can be adapted as well.
- Model type: left-right, also known as Bakis model (Yuan, 2018)
- The number of hidden states: three (good, moderate, bad). This is an initial setup which changed during the training session to compare with different numbers of hidden states.
- The number of transition matrices: six (according to group 1 in table 11)
- The initial state: state 1 (assumed as good)
- The results: all the results are presented as a statistical point of view

This model is designed to represent the aircraft engine with several operating condition modes. The model has been trained several times by assigning the operating conditions from different groups (mentioned earlier) to investigate if the operating conditions can be reduced without compromising the model performance. The performance of the model is evaluated based on several error rates defined below.

5.3 Model evaluation

The IOHMM is being trained under each of the operating conditions by the groups shown in Table 11. Estimated parameters from different training are compared and the model performance is evaluated. A benchmarking between HMM and different versions of IOHMM is presented in the result section. The best model is selected to perform the RUL prediction for the aircraft engine.

The model performance is evaluated by the score following Eq. 36. It is the score given in (K Le Son, 2013) to benchmark the methods. The lower is the score the better is the method. The score Sc is asymmetric that penalized the late predictions more than early prediction, and is defined as follows:

$$Sc = \sum_{i=1}^{218} Sc_i \tag{36}$$

here Sc_i is the penalty score for unit i, computed as follows:

$$Sc_i = \begin{cases} e^{-d_i/13} - 1, d_i \leq 0 \\ e^{d_i/10} - 1, d_i > 0 \end{cases}$$

here $d_i = R\widehat{U}L(i) - RUL(i)$ is the estimation error; $R\widehat{U}L(i)$ and $RUL(i)$ are the estimated and the real RUL values respectively of unit i. The acceptable window of estimation is presented in Fig. 36. An estimate of late failure (distance 10 units from original RUL) is more dangerous where the error d_i positive and the early failure estimation (distance 13 units to the original RUL) is considered as the negative d_i.

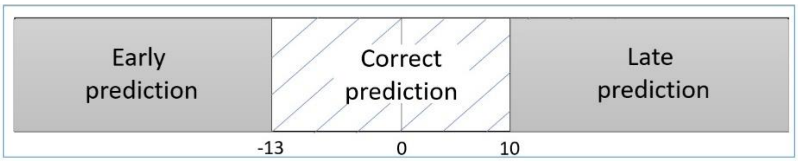


Fig. 36: Metrics of performance assessment

In this book, an interval [-10, +13] set by (Ramasso, 2013) is considered to assess the model performance (Fig 36). The prediction errors fall within this interval considered as the correct predictions. The errors which are less than the lower limit of the interval (-10) is considered as late prediction and greater than the higher limit (+13) are considered as early predictions. This interval is considered a serious condition compared to the literature (Goebel, 2005), but it can be differed according to the system complexity and prediction sensitivity. The penalty score function according to the Eq. 36 is given by the Fig. 37.

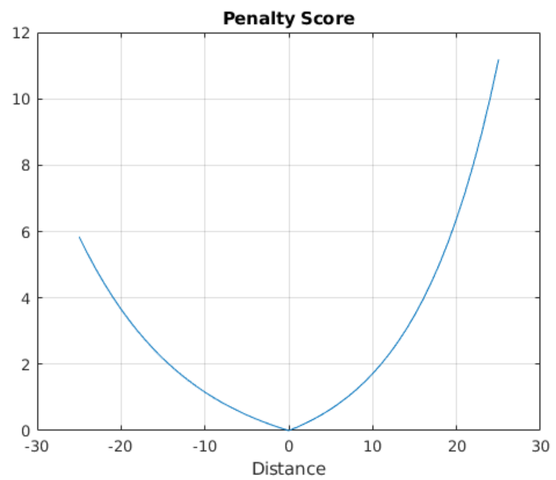


Fig. 37: Penalty Score function following Eq. 36

Some papers use two other criteria to compare the precision of methods, the root squared error (RSE) and the mean squared error (MSE), as defined below.

The root squared error (RSE):

$$RSE = \sqrt{\sum_{i=1}^{218} d_i^2} \tag{37}$$

The mean square error (MSE):

$$MSE = \sum_{i=1}^{218} \frac{d_i^2}{218} \tag{38}$$

These two accuracies are also like S, the lower the value is, the more accurate the performance is. In this chapter, we have used all three criteria to assess the model performances and create a benchmark between them. Moreover, the score S is used in the cross-validations to validate the estimated IOHMM by using the given data sets where the real failure time assumes as the length of the sequence. The cross-validation process is described hereafter.

5.4 Cross Validation

Model validation is a task which confirms the results of a model are sufficiently accurate to the results of the original data-generating process or not. The Cross-validation is experimented to analyse whether the predictive performance of the model deteriorates significantly when applied to new relevant data. This is also called rotation estimation (Geisser, 2017) which is a procedure to evaluate how the results of a statistical analysis will be generalized to an independent data set. There are three popular cross-validation methods are used in several papers. In this chapter, we applied all three methods to show the belief over the model performance:

1. Leave-p-out (LPO): this validation use *p* observation sequences as the validation data set and the rest as the training data set (Celisse, 2014). It is a one-time training and testing performance evaluation.
2. Leave-one-out (LOO): this is a particular case of leave-*p*-out cross-validation with $p = 1$. A random data sample is set aside for testing, and the model is trained with the remaining data. This method can be performed several times to produce a mean-performance RUL prediction.
3. *k*-fold cross-validation: in *k*-fold cross-validation, the training sample is divided into *k* equal size of groups. This technique repeated *k* times with *k* observation sequences as the validation data set and remaining observations as the training data set. It is similar to the leave-p-out validation, where the only difference is it performs *k* times. One group is selected as a validation data set and the rest as training data set, then a different group gets selected as a validation data set until all the groups get selected as a validation data set. k=10 is a commonly used case (McLachlan, 2005), but in general, *k* remains unfixed.

The first two are exhaustive cross-validation methods and the third validation is a non-exhaustive method which can be addressed as an approximation of leave-p-out cross-validation.

5.5 Results

The numerical results are given by using the 218 sequences from the unit 11 from the PCA output.

5.5.1 Parameter Learning

This section covers the model training considering the uncertainties about the model size and operating conditions. At first, the model is being trained with six operating conditions to fix the appropriate number of parameters to represent the system. Once the model size is fixed, we train the model several times by reducing the number of operating conditions and evaluate if the performance is good enough or not.

5.5.1.1 Number of Hidden States

The number of hidden states is not easy to decide while the states are unknown. Nevertheless, as mentioned earlier (in chapter 4) that, at least two states required to model the degradation of systems. The more the parameters are, the accurate the representation is. However, we are bounded to make the decision because too many parameters make the model complex which is difficult to learn.

We have experimented with an analysis to fix the suitable number of hidden states from multiple selections of choices. The procedures are already explained with a simulated application in chapter 4. Nevertheless, this time the method is applied to a real application to justify its impact.

Only one pair of transition and emission matrices of each model for each version of the models is highlighted in Table 12.

Table 12: Learning parameters of different Matrices

Model	Transition matrices	Emission matrices
2 states-Model	$\hat{A}^1 2 = \begin{pmatrix} 0.9901 & 0.0099 \\ 0 & 1.0000 \end{pmatrix}$	$\hat{B}^1 2 = \begin{pmatrix} 0.95 & 0.05 & 0 \\ 0 & 0.0.98 & 0.02 \end{pmatrix}$
3 states-Model	$\hat{A}^1 3 = \begin{pmatrix} 0.9711 & 0.0289 & 0 \\ 0 & 0.9556 & 0.0444 \\ 0 & 0 & 1.0000 \end{pmatrix}$	$\hat{B}^1 3 = \begin{pmatrix} 0.94 & 0.06 & 0 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{pmatrix}$
4 states-Model	$\hat{A}^1 4 = \begin{pmatrix} 0.9923 & 0.0067 & 0 & 0 \\ 0 & 0.9728 & 0.0272 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1.0000} & \mathbf{0} \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	$\hat{B}^1 4 = \begin{pmatrix} 0.95 & 0.05 & 0 \\ 0 & 0.99 & 0.01 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$
5 states-Model	$\hat{A}^1 5 = \begin{pmatrix} 0.9934 & 0.0066 & 0 & 0 & 0 \\ 0 & 0.9661 & 0.0339 & 0 & 0 \\ 0 & 0 & 0.9805 & 0.0195 & 0 \\ 0 & 0 & 0 & 0.9748 & 0.0252 \\ 0 & 0 & 0 & 0 & 1.0000 \end{pmatrix}$	$\hat{B}^1 5 = \begin{pmatrix} 0.99 & 0.01 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$

here $\hat{A}^1 2$ and $\hat{B}^1 2$ are the estimated parameters of the first matrices of 2 states IOHMM. Similarly, $\hat{A}^1 3$ and $\hat{B}^1 3$ are from 3 states IOHMM and so on.

The selection technique is done in three steps. The idea was to design a method to compare the performances of the models and selects the best performer for prognostics. However, it could be a time-consuming process in the sense of the number of models and their run times. That is why some quick investigations are applied to reduce some models based on the parameter's nature and inconvenience. These investigations are already discussed in chapter 4.

Look-out the transition parameters

It is an investigation on the estimated transition matrices to identify if there is any insignificant parameter exists or not. For example, there is no transition from state three to other states in the matrix A4. The third row represents all the possible transitions from state three to others. The parameter $\hat{A}^1 4$ (3,3) is holding a 100% transition probability, so it is an absorbent state. However, there is another parameter $\hat{A}^1 4$ (4,4) is also an absorbent state. The proposed model does not consider two absorbent states in the same transition matrix as it is explained before. Therefore, parameter $\hat{A}^1 4$ (4,4) and the corresponding row and column are removed from the matrix. This problem can be identified in another way which is studying the corresponding emission parameters. Noted that, the emission matrix $\hat{B}^1 4$ is repeating the same parameters (3rd and 4th rows). It is an indication that corresponding transition parameters are needed to be adjusted. In this case, for example, removing the fourth row, which makes the matrix as a 3-by-3 dimension.

Look-out the emission parameters

It is an investigation on the emission matrices based on the relation between the state and the emitted symbols. The parameters are modified/removed if there is more than one state holding similar properties as we mentioned in the previous section. For example, the emission matrix $\hat{B}^1 5$ contains two repeating cases with the parameters (row-3 and 4) and (row 5 and 6). These (transition) parameters are adjusted by merging the repeated rows together. Therefore, the size of the matrix became from a 5-by-5 to a 3-by-3 dimension.

These two steps provide a common indication about the model size which is 3-by-3. Even though matrix $\hat{A}^1 4$ and $\hat{A}^1 5$ were estimated as four and five dimensions of sizes but after reconsidering the significance of the parameters with the transition and emission properties both of them suggest that the engine can be represented by state models. However, there are two more models ($\hat{A}^1 2$ and $\hat{A}^1 3$) are available that contain all the transition and emission parameters as an acceptable format. So, these two are checked through the final step.

Compare by model-performance

The remaining models after applying the first two steps are treated through this step which is based on the model performances. A performance evaluating method is developed which evaluates the model performance by using the Eq. 36. The once gives the lowest score is the best model.

The matrix $\hat{A}^1 3$ has been selected for further experiments based on the performance evaluation.

5.5.1.2 Estimated Parameters

Once the number of hidden states is fixed, now the first step is to fix the number of operating conditions for the model. Saxena already suggested that the possible number of operating conditions for the engine is six (Saxena, 2008). This experiment to demonstrate if the number of operating conditions can be reduced without compromising the model performance. The grouping between the operating conditions is mentioned in Table 11. A different number of operating conditions (from zero to six) are applied to the model training and the estimated models are evaluated through the performance evaluator by the group. The parameters are shown by the group below.

HMM (IOHMM with no conditions):

The IOHMM with zero operating condition is equivalent to an HMM. So, the model learned with one transition matrix and one emission matrix. Initial distribution is also learned from the training, but only the transition matrix is presented below:

$$\hat{A} = \begin{pmatrix} 0.9906 & 0.0094 & 0 \\ 0 & 0.9330 & 0.0670 \\ 0 & 0 & 1.0000 \end{pmatrix}$$

IOHMM (with conditions): Only the transition matrices from each group are presented below.

- *IOHMM (six operating conditions represented by six models):*

$\hat{A}^1 = \begin{pmatrix} 0.9923 & 0.0077 & 0 \\ 0 & 0.9482 & 0.0518 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model one	$\hat{A}^2 = \begin{pmatrix} 0.9888 & 0.0112 & 0 \\ 0 & 0.9368 & 0.0632 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model two	$\hat{A}^3 = \begin{pmatrix} 0.9896 & 0.0140 & 0 \\ 0 & 0.9398 & 0.0602 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model three
$\hat{A}^4 = \begin{pmatrix} 0.9918 & 0.0082 & 0 \\ 0 & 0.9454 & 0.0546 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model four	$\hat{A}^5 = \begin{pmatrix} 0.9910 & 0.0090 & 0 \\ 0 & 0.9124 & 0.0876 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model five	$\hat{A}^6 = \begin{pmatrix} 0.9893 & 0.0107 & 0 \\ 0 & 0.9072 & 0.0928 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model six

- *IOHMM (five operating conditions represented by five models):*

$\hat{A}^1 = \begin{pmatrix} 0.9923 & 0.0077 & 0 \\ 0 & 0.9482 & 0.0518 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model one	$\hat{A}^2 = \begin{pmatrix} 0.9892 & 0.0108 & 0 \\ 0 & 0.9384 & 0.0616 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model two	$\hat{A}^3 = \begin{pmatrix} 0.9918 & 0.0082 & 0 \\ 0 & 0.9454 & 0.0546 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model three
$\hat{A}^4 = \begin{pmatrix} 0.9910 & 0.0090 & 0 \\ 0 & 0.9124 & 0.0876 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model four	$\hat{A}^5 = \begin{pmatrix} 0.9892 & 0.0108 & 0 \\ 0 & 0.9072 & 0.0928 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model five	

- *IOHMM (four operating conditions represented by four models):*

$\hat{A}^1 = \begin{pmatrix} 0.9922 & 0.0078 & 0 \\ 0 & 0.9483 & 0.0517 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model one	$\hat{A}^2 = \begin{pmatrix} 0.9923 & 0.0077 & 0 \\ 0 & 0.9482 & 0.0518 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model two
$\hat{A}^3 = \begin{pmatrix} 0.9918 & 0.0082 & 0 \\ 0 & 0.9458 & 0.0542 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model three	$\hat{A}^4 = \begin{pmatrix} 0.9902 & 0.0098 & 0 \\ 0 & 0.9094 & 0.0906 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model four

- *IOHMM (three operating conditions represented by three models):*

$\hat{A}^1 = \begin{pmatrix} 0.9922 & 0.0078 & 0 \\ 0 & 0.9483 & 0.0517 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model one	$\hat{A}^2 = \begin{pmatrix} 0.9904 & 0.0096 & 0 \\ 0 & 0.9420 & 0.0580 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model two	$\hat{A}^3 = \begin{pmatrix} 0.9901 & 0.0099 & 0 \\ 0 & 0.9090 & 0.0910 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model three

- *IOHMM (two operating conditions represented by two models):*

$\hat{A}^1 = \begin{pmatrix} 0.9923 & 0.0077 & 0 \\ 0 & 0.9477 & 0.0523 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model one	$\hat{A}^2 = \begin{pmatrix} 0.9903 & 0.0097 & 0 \\ 0 & 0.9304 & 0.0696 \\ 0 & 0 & 1.0000 \end{pmatrix}$ Model two

The IOHMM performed separately by using all these five groups with different operating conditions. An HMM is also used to prognostic system health, where the operating conditions are ignored, to compare with the other results. Next section explains the diagnostic and the prognostic results where the simplest operating conditions (group 5) are applied. Later, all the groups are compared by their performances based on the estimated prognostic.

5.5.2 Diagnostic: current health state estimation

The given sequence is used to demonstrate the diagnostic and prognostic performance. The cross-validation methods are applied where the estimated model is performed several times on randomly selected sequences. One example is given in Fig. 38 where the diagnostic result is given which is estimated from a given sequence:

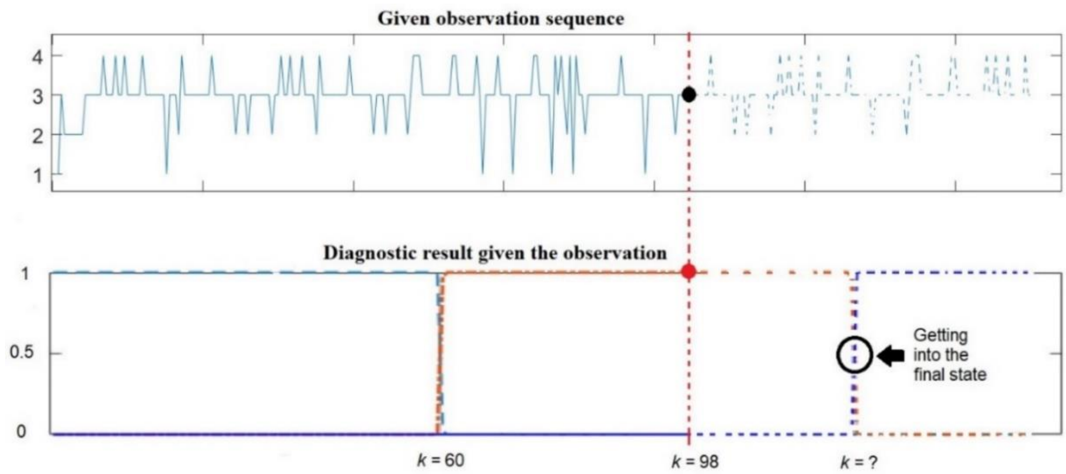


Fig. 38: Estimated diagnostic given a test sequence

The sequence is cut downed at the time instant k = 98. The diagnostic result shows that the system was in the first state at the beginning. Then, at k = 60 it transits to the second state and stays until the time instant k = 98 with the distribution as $P(X_{k=98}) = (7.2 \times 10^{-132}\ \ 5.6 \times 10^{-15}\ \ 0)$ based on the given sequence. After scaling by 1 it can be written as (0 1 0) which implies that the system partially degraded. The goal is to identify the time to go into the final state which is defined as the RUL that predicted in the next section.

5.5.3 Prognostic: the meantime RUL estimation

The meantime RUL is predicted according to the current health state and the operating condition. The future operating conditions for the engine are assumed as in two possibilities. The first possibility is that the operating conditions are unknown for future operations. Equation (15) is used to solve this problem by following the old operating conditions that have been applied so far. The second possibility is predicting the RUL at the given operating conditions. Both the cases are presented where the predicted RUL for unknown operating conditions is 96 days (Fig. 39), and the predicted RUL for a given operating condition is 82 days (Fig. 40).

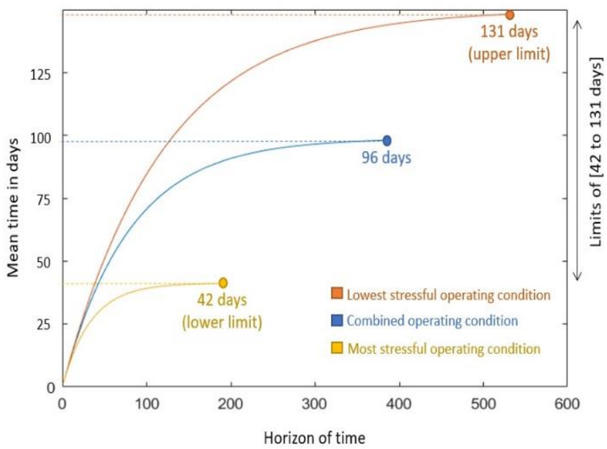


Fig. 39: Mean time RUL for unknown inputs

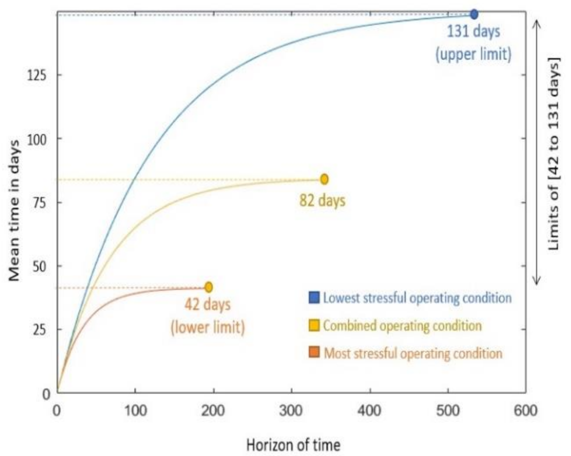


Fig. 40: Mean time RUL for known inputs

This figure represents the computing process for explain the accuracy. IOHMM also provides different RUL by using the operating conditions separately. The predicted RUL using the most stressful model is addressed as the lowest limit (as 42 days) and the lowest stressful operating condition as the highest RUL prediction limit (as 131 days).

The same process is followed by each group of operating conditions and the model training. Finally, a benchmarking between the model performances is given in the next section.

5.5.4 Benchmarking Between Different Models

A benchmark between HMM and different versions of IOHMM is presented concerning the score, RSE, and MSE by using the Eq.36, 37, and 38. The scores decrease while the number of conditions increases (shown in Table 13).

Table 13: Different model performances

ID	Model (Condition)	Score	RSE	MSE	Number of parameters
1	HMM (condition = 0)	18.01	31.88	32.79	24
2	IOHMM (conditions = 3)	17.32	31.15	31.30	42
3	IOHMM (conditions = 4)	**17.30**	**31.11**	**31.23**	**51**
4	IOHMM (conditions = 5)	17.31	31.12	31.23	60
5	IOHMM (conditions = 6)	**17.29**	**31.11**	**31.22**	**69**

The performance for an HMM is shown in this table as Score = 18.01, RSE = 31.88, MSE = 32.79. This model is equivalent to the IOHMM with no operating condition which provides the largest scores among the other IOHMMs except the model with condition 2. The IOHMM with 2 operating conditions is ignored because the classification shows insignificant results.

The classification is done based on the nearest neighbor's property but (particularly) the group number 5 does not strictly follow that. As a result, the learning algorithms get to learn a model by considering data that represents different dynamics of system behaviors. This is the fact when the unfamiliar data are given to test the classification was not accurate nor the performance is. It shows an unusual score which is worse than the HMM model.

IOHMM with six operating conditions estimates the RUL with the best scores (Score = 17.29, RSE = 31.11, MSE = 31.22). This experiment indicates that the more conditions give better results. However, it is also noted that the model complexity is proportional to the number of parameters of the model. Moreover, the amount of data is also an important fact since more parameters required a larger amount of data. That is why the model is chosen depending on the performance as well as the number of parameters following the Occam's razor principle (A Baker, 2007). For example, IOHMM with four operating conditions could be a good choice. It has about 26% less parameters and good accuracy (Score = 17.30, RSE = 31.11, MSE = 31.23 vs Score = 17.29, RSE = 31.11, MSE = 31.22) which is very close to the model with six conditions.

This experiment shows that IOHMM models allow different regimes to consider as different operating conditions of the system. IOHMM allows learning all the parameters of the model through a single training session. Several modes of IOHMM give promising RUL estimating performances than a standard HMM. Compared to the performance and the number of hidden states, the IOHMM with four operating conditions seems to be the best fit for representing the engine. This model can be used to further analysis of the dataset by compare the test set results performance with existing results.

5.5.5 Cross Validations

This section validates the selected IOHMM by the cross-validation methods. This experiment justifies the selected model by assessing its performance of estimated RUL compared to the known RUL by Eq.36. 218 sequences of the selected unit are used in these validations. Three different training techniques are applied to these methods.

Leave P Out (LPO): in this method, the training has done 10 times. In each training, a random set of (P = 5) sequences are selected for testing and the rest of 213 are used to train the model. The number of training and the size of P can be chosen differently.

Leave One Out (LOO): in this method, the training has done 50 times. In each training, a random sequence is selected for testing and the rest of 217 are used to train the model. The number of training can be chosen differently.

k-fold: in this method, the training has done 5 times. In each training, a random set of (one fifth of 218) sequences are selected for testing and the rest of the sequences are used to train the model. The number of training can be chosen differently.

The results from these methods are stored as early, on-time, and late predictions.

Table 14 shows the model performance by applying three cross-validations: LPO, LOO, *k*-fold (cf. section 5.4). The validation shows very few late predictions (6%) compared to the summation of on-time and early predictions. The late prediction contains the biggest penalty then the early predictions. If the early prediction considered as acceptable then, the proposed model enhances the RUL prediction performance up to 95% (LPO: 41+54).

Table 14: Cross validation results

Method ID	Method	RUL Prediction		
		Early	On-time	Late
1	LPO	41%	54%	**5%**
2	LOO	55%	39%	**6%**
3	*k*-fold	54%	40%	**6%**

5.6 Conclusion

This chapter describes how to model the health degradation of aircraft engines by IOHMM under multiple operating conditions. The reason for selecting this application is that it covers most of the objectives that we have proposed. This is an application where the given dataset represents the degradation of the aircraft under multiple combinations of operating conditions. To consider the degradation with different uncertainties under multiple operating conditions, the IOHMM is one of the ideal modeling tools to design the aircraft and apply the diagnostic and prognostic algorithms. An open data challenge is taken into account in this chapter for estimating RUL of the system from several settings of operating conditions.

The difference between the application of this chapter and the previous chapter (chapter 4) is the data set. In chapter 4, all the applications were simulated where the data were ready to train the model (IOHMM) for degradation representation. Nevertheless, it is not that easy in real cases where the data comes as raw elements from the sensor readings. Therefore, it requires to prepare for use in training and testing purposes. Usually, the original dataset does not have the indicator of degradation. In this chapter, a detailed explanation of data preparation for system modeling is given step by step with examples. The PCA method is applied to identify the indicators from each of the sequences. Then a set of thresholds is defined to classify the indicators by assigning the discrete symbols.

Another difference is that multiple outputs were not used in this application. The reason for that, the PCA method uses all the outputs to provide results where the coefficient of the outputs already exists. However, more than one output can be considered but the main concern of this chapter is to demonstrate the open challenge application simulating degradation under multiple (inputs) operating conditions.

Several versions of IOHMM are designed considering a different number of health states and operating conditions for the best fitting model to the engine. The model is validated by three cross-validations: LPO, LOO, and k-fold methods with a maximum of 6% late predictions. Three similar learning techniques are applied to these three validations procedures where the training set is used to train the model and test the results together. That is why no test set is used in this chapter. The model is trained by using the training set to have separated one/more randomly selected sequence/s for testing purposes. This repeatedly applied to provide confidence over the model performance.

The adapted Baum Welch and forward-backward algorithms are used to learn the IOHMM. Then the learned IOHMM is used in online and offline health prediction. A benchmarking of performance assessment between the HMM and several versions of IOHMM is presented with the error rate (the root squared error and the mean squared error). This comparison helps to decide the suitable number of operating conditions for modeling the degradation of the system.

Chapter 6

The Fourth Contribution: Estimating RUL of Structured System

Table of Contents

6 The Fourth Contribution: Estimating RUL of Structured Systems

In most of the works in the literature, the prognostic is dedicated to an entity (component, subsystem, system) considered as a whole. Nevertheless, systems are more a combination of components following a particular structure for a functional purpose. So, the systems are structured, and their health evolution modelling can be handled following this structure as it is usually done in reliability analysis.

Such systems are widely used in many industrial processes when it is necessary to distribute products in several ways and then to collect them into one or several discharge destinations. For example, flow distribution systems (FDS) like water supply, heat supply, electricity supply, etc. The maintenance decisions for the FDSs are challenging because the degradation of individual components is independent and not fully detectable. In this chapter, we propose a perspective methodology to prognostic the degradation of a structured system by diagnosis each of the components individually and constructing a model that represents the entire system health evolution. This is a perception of answering the third question of the thesis: *Prognostic the RUL for structured systems from their components to study the entire system health evolution.*

This chapter uses the previous IOHMM for RUL assessment to maintenance aid decision-making of a multi-component flow distribution system. As the main structures are series or parallel, the methodology is built on these two structures of connected components. Nevertheless, industrial systems are often equipped with multiple sensors to monitor outputs of components to collect the efficient information that helps to prognostic the RUL. These sensors can be real or virtual (Albertos, 2002). As multiple data are captured, it is a multiple output system. Moreover, the components or subsystems under study are driven by several inputs in several modes, it is then a multiple inputs system. Thus, IOHMM are considered better to estimate the components RUL than HMM. If an IOHMM focuses on a system sub-element, then the question of combination is the key point of our proposition. To demonstrate the proposed methodologies, a real system [Esrel 2011 Barcelona process] with simulated data is used. All the paths from sources to a destination are considered alternative options to supply the demands.

The proposal offers a solution in two steps. The first step is the independent path monitoring to determine the most appropriate supply planning strategy. The second step is to identify all the possible routes where the flow gets through different components for discharging to the destinations. It allows the system to select alternative paths to supply the flow. The operating conditions are considered as common inputs for all the components. Once all the model trainings are done, a big model is constructed from these model parameters to represent the entire system.

6.1 Model construction for prognosing the system RUL

As IOHMM are a particular combination of HMM, let us handle the problem by starting with HMM only. The main goal is to build a prognostic model of the whole system from the model of its components following the functional structure of the system. Nevertheless, let us recall some important notions from previous chapters. The learning and the diagnostic steps are based on a complete IOHMM model, but the prognostic part uses only the hidden Markov part. It means that a complete IOHMM can be built for each component or sub-system, but a combination of their hidden part is enough to estimate the system RUL.

Back from the previous chapters, the health state evolution of each system component C_i can be modelled by an IOHMM $\Lambda_i = (A_i, B_i, \pi_i)$ where A_i is the transition matrices defined given the input modes and B_i is a set of emission matrices according to the number of outputs. So, the health state evolution of the system should be modelized accordingly *i.e.* finding the function f for multiple components:

$$\Lambda = f(\Lambda_1, \Lambda_2, \dots, \Lambda_i)$$

Let us recall the notation of $\Lambda_i = (A_i, B_i, \pi_i)$:

$$A_i = \{A_i^1, \dots, A_i^{P_i}\}$$

where A_i^1 is the transition matrix of the hidden part given that the input U_1 is in mode 1: $U_1 = 1$ and $A_i^{P_i}$ is the transition matrix given that the input U_i is in mode P_i: $U_1 = P_i$

So, if we consider two components then we should consider two IOHMM to define the function f. Function f depends on the structure of the system with the two components *i.e.* series or parallel. It is possible to cover multiple components of a single path by applying the construction policy of two components. The goal is to find a model that represents the health states of the path. An iterative construction process can be applied to the components where each step considers first two components and then construct them into one model. After that, the same process can be applied to the constructed model with the model for the next component. This process continuous until the final model is built.

6.1.1 Series structure of two components with HMM models

For the sake of clarity, let us start with two HMM *i.e.* without considering inputs. The functional structure of the system is given by Fig. 41 which represents the Reliability Bloc Diagram (RBD).

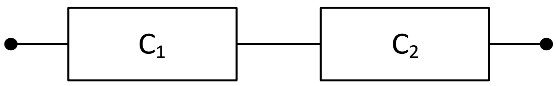


Fig. 41: RBD of a two components series system

Let us consider that C_1 has hidden states $S_1 = \{\mathrm{S}_1^1, \dots, \mathrm{S}_1^{\mathrm{N}}\}$. The evolution of the health state of the component C_1 is given by the BAKIS (Yuan, 2018) model $\lambda_1 = (A_1, B_1, \pi_1)$ with the transition matrix:

$$A_1 = \begin{bmatrix} 1 - \sum_{j=2}^{N} \alpha^{1j} & \alpha^{12} & \cdots & \alpha^{1N} \\ 0 & 1 - \sum_{j=3}^{N} \alpha^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \alpha^{(N-1)N} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{39}$$

The main goal is to estimate the RUL which only needs the computation of the diagnostic and the transition matrix of the system. The diagnostic is computed from the health states condition of components, so there is not a necessity to construct the emission matrices for the RUL estimation. Therefore, just transition matrices are constructed.

Respectively, the health state evolution of component C_2 described by the hidden heath state $\boldsymbol{S}_2 = \{\mathrm{S}_2^1, \dots, \mathrm{S}_2^{\mathrm{M}}\}$ is given by:

$$A_2 = \begin{bmatrix} 1 - \sum_{j=2}^{M} \beta^{1j} & \beta^{12} & \cdots & \beta^{1M} \\ 0 & 1 - \sum_{j=3}^{M} \beta^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \beta^{(N-1)M} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{40}$$

Based on the state of each component, the state of the series system can be given based on the cardinal product of component states. So, if C_1 has N states and C_2 has M states then the series systems S has $N \times M$ possible states:

$$\boldsymbol{S}_S = \boldsymbol{S}_1 \times \boldsymbol{S}_2 = \left\{\{S_1^1 S_2^1\}, \{S_1^1 S_2^2\} \dots, \{S_1^1 S_2^M\}, \{S_1^2 S_2^1\}, \dots, \{S_1^2 S_2^M\}, \{S_1^3 S_2^1\}, \cdots, \{S_1^N S_2^M\}\right\}$$

To define the hidden Markov model of the two components series system, the transition from one state to the other should follow the functional structure. In a series system, some states are not accessible since at least one of the components is in its absorbent state. The transition matrix of the whole system is defined by Eq. 41:

$$A_S = \begin{bmatrix} - & \beta^{12} & \dots & \beta^{1M} & \alpha^{12} & \dots & 0 & \alpha^{13} & \dots & 0 \\ 0 & - & \dots & \beta^{2M} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & - & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & - & 0 & \dots & \alpha^{12} & 0 & \dots & \alpha^{1N} \\ 0 & 0 & 0 & 0 & - & \dots & \beta^{2M} & \alpha^{23} & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & - & \beta^{xM} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & \dots & \alpha^{2N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (41)$$

This is a stochastic matrix so, the diagonal is: (1- summation of all distribution on the corresponding the same row), as $\sum_j^N a_{ij} = 1 \ \forall i$.

For the sake of illustration, we provide an example with two components having 3 hidden states each and following the BAKIS model (Yuan, 2018) structure. The transition matrices for two components are defined as:

Component C_1: $\begin{pmatrix} 1 - \sum_{j=2}^{3} \alpha^{1j} & \alpha^{12} & \alpha^{13} \\ 0 & 1 - \alpha^{13} & \alpha^{13} \\ 0 & 0 & 1 \end{pmatrix}$; Component C_2: $\begin{pmatrix} 1 - \sum_{j=2}^{3} \beta^{1j} & \beta^{12} & \beta^{13} \\ 0 & 1 - \beta^{13} & \beta^{13} \\ 0 & 0 & 1 \end{pmatrix}$

Based on these two transitions matrices, the Markov chain of the (two components) series system is given by Fig. 42.

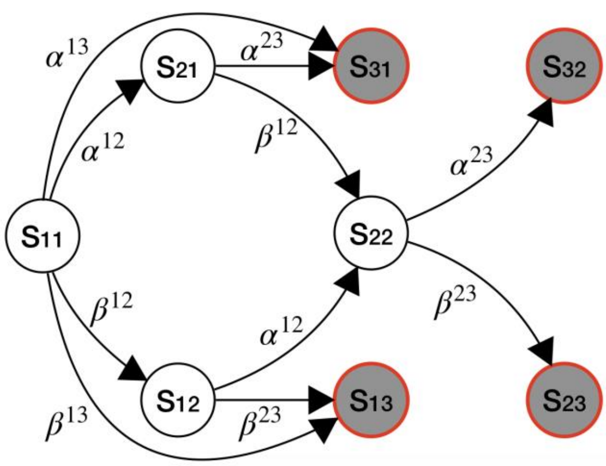


Fig. 42: Transition graph of two series components.

As we can see on Fig. 42, the states of the whole system are reduced from the cartesian product of states. S33 is not defined because there is no possibility to jump from any state to S33. The first component that joins his state S3 stops the whole system.

So, following Eq. 41 and the previous remark, the transition matrix is:

$$A_S = \begin{pmatrix} - & \beta^{12} & \alpha^{12} & 0 & \beta^{13} & \alpha^{13} & 0 & 0 \\ 0 & - & 0 & \alpha^{12} & \beta^{23} & 0 & 0 & 0 \\ 0 & 0 & - & \beta^{12} & 0 & \alpha^{23} & 0 & 0 \\ 0 & 0 & 0 & - & 0 & 0 & \beta^{23} & \alpha^{23} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

By merging the absorbing states:

$$\tilde{A}^1 = \begin{pmatrix} \times & \beta^{12} & \alpha^{12} & 0 & \beta^{13} + \alpha^{13} \\ 0 & \times & 0 & \alpha^{12} & \beta^{23} \\ 0 & 0 & \times & \beta^{12} & \alpha^{23} \\ 0 & 0 & 0 & \times & \beta^{23} + \alpha^{23} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{42}$$

Noted that the constructed matrix shows that the number of health states for the whole system is five ($s_{11}, s_{12}, s_{21,} s_{22}, s_f$) even though the components are assumed to have three hidden states.

6.1.2 Series structure of two components with IOHMM models

If we consider that two components degrade with regards to inputs, the HMM models are replaced by IOHMM. The evolution of the health state of component C_1 is given by the following IOHMM model $\Lambda_1 = (A_1, B_1, \pi_1)$ with $A_1 = \{A_1^1, \ldots, A_1^P\}$ considering the P modes of the input:

$$A_1^p = \begin{bmatrix} 1 - \sum_{j=2}^{N} \alpha_p^{1j} & \alpha_p^{12} & \cdots & \alpha_p^{1N} \\ 0 & 1 - \sum_{j=3}^{N} \alpha_p^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \alpha_p^{(N-1)N} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{43a}$$

The hidden states of C_1 are $S_1 = \{\mathrm{S}_1^1, \ldots, \mathrm{S}_1^{\mathrm{N}}\}$.

Respectively, the health state evolution of component C_2 is defined by the IOHMM $\Lambda_2 = (A_1, B_1, \pi_1)$ with $A_2 = \{A_2^1, \ldots, A_2^Q\}$ considering the Q modes of the input follows:

$$A_2^q = \begin{bmatrix} 1 - \sum_{j=2}^{M} \beta_q^{1j} & \beta_q^{12} & \cdots & \beta_q^{1M} \\ 0 & 1 - \sum_{j=3}^{M} \beta_q^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \beta_q^{(N-1)M} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{43b}$$

The hidden states of C_2 are $\boldsymbol{S}_2 = \{\mathrm{S}_2^1, \ldots, \mathrm{S}_2^{\mathrm{M}}\}$

To build the model of a series system of two components, the methodology follows exactly the same procedure but given that each transition matrix is selected among the set A_1^p (resp. A_2^q) by the input mode p (resp. q) then the transition matrix becomes as presented by Eq.44:

$$A_S^{pq} = \begin{bmatrix} - & \beta_q^{12} & \dots & \beta_q^{1M} & \alpha_p^{12} & \dots & 0 & \alpha_p^{13} & \dots & 0 \\ 0 & - & \dots & \beta_q^{2M} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & - & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & - & 0 & \dots & \alpha_p^{12} & 0 & \dots & \alpha_p^{1N} \\ 0 & 0 & 0 & 0 & - & \dots & \beta_q^{2M} & \alpha_p^{23} & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & - & \beta_q^{xM} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & \dots & \alpha_p^{2N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{44}$$

For the sake of illustration, an example is given to explain the series structure of two components with IOHMM models. The system is assumed to have 3 hidden states. Figure 43 represents the system which assumed to have the inputs U_{C1} and U_{C2} with two operating modes (1 & 2) for each. The variable U_{C1} is the input for component C_1, and U_{C2} is for component C_2.

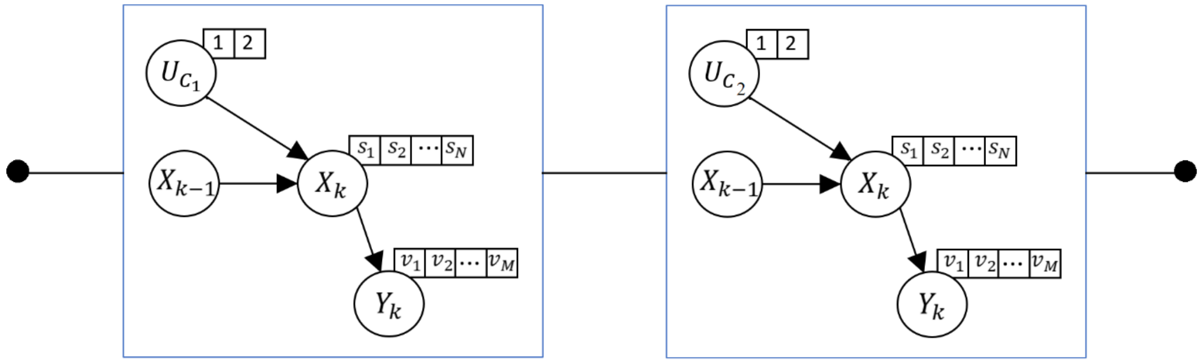


Fig. 43: Two components in a series system with separated inputs

The transition matrices for the components are defined according to the operating conditions as follows.

The IOHMM model of component C_1 provides two transition matrices (A_1^1, A_1^2) for two modes according to $U1$: (Here, matrix A_j^p represents the first component transition probabilities where p is the id of the operating condition and j is the id of the component.)

$$A_1^1: \begin{pmatrix} 1 - \sum_{j=2}^{3} \alpha_1^{1j} & \alpha_1^{12} & \alpha_1^{13} \\ 0 & 1 - \alpha_1^{13} & \alpha_1^{13} \\ 0 & 0 & 1 \end{pmatrix} \text{represented by input mode } p = 1.$$

$$A_1^2: \begin{pmatrix} 1 - \sum_{j=2}^{3} \alpha_2^{1j} & \alpha_2^{12} & \alpha_2^{13} \\ 0 & 1 - \alpha_2^{13} & \alpha_2^{13} \\ 0 & 0 & 1 \end{pmatrix} \text{represented by input mode } p = 2.$$

Component C_2 provides two transition matrices (A_2^1, A_2^2) for two modes:

Matrix number one A_2^1: $\begin{pmatrix} 1 - \sum_{j=2}^{3} \beta_1^{1j} & \beta_1^{12} & \beta_1^{13} \\ 0 & 1 - \beta_1^{13} & \beta_1^{13} \\ 0 & 0 & 1 \end{pmatrix}$ represented by input mode $q = 1$.

Matrix number two A_2^2: $\begin{pmatrix} 1 - \sum_{j=2}^{3} \beta_2^{1j} & \beta_2^{12} & \beta_2^{13} \\ 0 & 1 - \beta_2^{13} & \beta_2^{13} \\ 0 & 0 & 1 \end{pmatrix}$ represented by input mode $q = 2$.

If the system applies separate operating conditions to the components then, the operating modes between two components follow a set of combinations. For example, the operating modes for two components (p, q) provide 4 combinations:

- $(p = 1, q = 1)$, both components are on the first mode of their dedicated operating conditions.
- $(p = 1, q = 2)$, the first component is on the first mode and the second component is on the second mode of the dedicated operating conditions.
- $(p = 2, q = 1)$, the first component is on the second mode and the second component is on the first mode of the dedicated operating conditions.
- $(p = 2, q = 2)$, both components are on the second mode of their dedicated operating conditions.

The IOHMM (transition matrices) can be constructed according to the modes of the operating conditions which led us to apply the HMM construction methods. So, corresponding to the system presented in Fig 43, it gives two constructed transition matrices for two operating modes of this system.

- Constructed matrix for mode one A_S^{11}: (construction of matrices A_1^1 and A_2^1**)**

$$
\begin{array}{c}
\\ s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_{13} \\ s_{31} \\ s_{23} \\ s_{32}
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} s_{11} & s_{12} & s_{21} & s_{22} & s_{13} & s_{31} & s_{23} & s_{32} \end{array} \\
\begin{pmatrix}
1-(Vs_{11}) & \beta_1^{12} & \alpha_1^{12} & 0 & \beta_1^{13} & \alpha_1^{13} & 0 & 0 \\
0 & 1-(\alpha_1^{12}+\beta_1^{23}) & 0 & \alpha_1^{12} & \beta_1^{23} & 0 & 0 & 0 \\
0 & 0 & 1-(\beta_1^{12}+\alpha_1^{23}) & \beta_1^{12} & 0 & \alpha_1^{23} & 0 & 0 \\
0 & 0 & 0 & 1-(\beta_1^{23}+\alpha_1^{23}) & 0 & 0 & \beta_1^{23} & \alpha_1^{23} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

[here, $Vs_{11} = \beta_1^{12} + \alpha_1^{12} + \beta_1^{13} + \alpha_1^{13}$].

The system gets out of order if any one of these two components fails. Therefore, it does not get into the state s_{33}. So, the matrix is constructed as 8-by-8 dimension where four states $(s_{11}, s_{12}, s_{21}, s_{22})$ are considered as working states, and other four states $(s_{13}, s_{31}, s_{23}, s_{32})$ as the breakdown states. The matrix can be represented as a 5-by-5 matrix instead of 8-by-8 by replacing all the breakdown states as one state:

$$
A_S^{11} = \begin{pmatrix}
1-(\beta_1^{12}+\alpha_1^{12}+\beta_1^{13}+\alpha_1^{13}) & \beta_1^{12} & \alpha_1^{12} & 0 & \beta_1^{13}+\alpha_1^{13} \\
0 & 1-(\alpha_1^{12}+\beta_1^{23}) & 0 & \alpha_1^{12} & \beta_1^{23} \\
0 & 0 & 1-(\beta_1^{12}+\alpha_1^{23}) & \beta_1^{12} & \alpha_1^{23} \\
0 & 0 & 0 & 1-(\beta_1^{23}+\alpha_1^{23}) & \beta_1^{23}+\alpha_1^{23} \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

This is (A_S^{11}) the transition matrix that represents the entire system for the given operating condition with operating mode one.

- Constructed matrix for mode two A_S^{22}: (construction of matrices A_1^2 and A_2^2**)**

$$
A_S^{22} = \begin{pmatrix}
1-(\beta_2^{12}+\alpha_2^{12}+\beta_2^{13}+\alpha_2^{13}) & \beta_2^{12} & \alpha_2^{12} & 0 & \beta_2^{13}+\alpha_2^{13} \\
0 & 1-(\alpha_2^{12}+\beta_2^{23}) & 0 & \alpha_2^{12} & \beta_2^{23} \\
0 & 0 & 1-(\beta_2^{12}+\alpha_2^{23}) & \beta_2^{12} & \alpha_2^{23} \\
0 & 0 & 0 & 1-(\beta_2^{23}+\alpha_2^{23}) & \beta_2^{23}+\alpha_2^{23} \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

This is (A_S^{22}) the transition matrix that represents the entire system for the given operating condition with operating mode two.

These two transition matrices (A_S^{11}, A_S^{22}) can be used to predict the RUL of the series system with two components. The combination of model construction depends on the number of input modes. Similar approach can be extendable for handling multiple components with series connection. The next section explains model construction of the parallel components system.

6.1.3 Parallel structure of two components with HMM models

A parallel structured system with two components is presented in this section to explain how two HMMs can be constructed for parallel connection. Like the series system, here the explanation is also given starting by two HMM *i.e.* without considering inputs, then with input as IOHMM.

A system is presented in Fig. 44 as the RBD where two components (C_1, C_2) are parallel connected. Since there are no inputs, HMM is sufficient to represent the evolution of their health state.

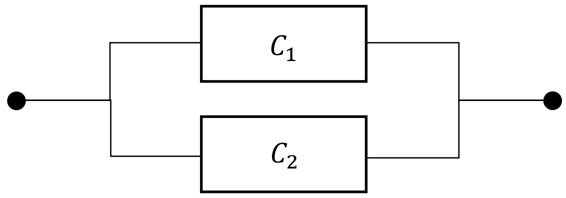


Fig. 44: RBD of a two components parallel system

The evolution of the health state of these two components are the same as it described in the series system (Eq. 39 and 40).

$$A_1 = \begin{bmatrix} 1-\sum_{j=2}^{N}\alpha^{1j} & \alpha^{12} & \cdots & \alpha^{1N} \\ 0 & 1-\sum_{j=3}^{N}\alpha^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \alpha^{(N-1)N} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1-\sum_{j=2}^{M}\beta^{1j} & \beta^{12} & \cdots & \beta^{1M} \\ 0 & 1-\sum_{j=3}^{M}\beta^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \beta^{(N-1)M} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C_1, given by $\lambda_1 = (A_1, B_1, \pi_1)$,
The hidden states are $S_1 = \{S_1^1, \dots, S_1^N\}$.

C_2, given by $\lambda_1 = (A_1, B_1, \pi_1)$,
The hidden states are $S_2 = \{S_2^1, \dots, S_2^M\}$

Based on the state of each component, the state of the parallel system can also be given by the cardinal product of component states. So, if C_1 has N states and C_2 has M states then the series systems S has NxM possible states:

$$\boldsymbol{S}_S = \boldsymbol{S}_1 \times \boldsymbol{S}_2 = \left\{\{S_1^1S_2^1\}, \{S_1^1S_2^2\} \dots, \{S_1^1S_2^M\}, \{S_1^2S_2^1\}, \dots, \{S_1^2S_2^M\}, \{S_1^3S_2^1\}, \cdots, \{S_1^NS_2^M\}\right\}$$

In a parallel system, there is only one state which is absorbent state. Therefore, the constructed transition matrix of the whole system would be:

$$A_S = \begin{bmatrix} - & \beta^{12} & \dots & \beta^{1M} & \alpha^{12} & \dots & 0 & \alpha^{13} & \dots & 0 \\ 0 & - & \dots & \beta^{2M} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & - & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & - & 0 & \dots & \alpha^{12} & 0 & \dots & \alpha^{1N} \\ 0 & 0 & 0 & 0 & - & \dots & \dots & \alpha^{23} & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & - & \dots & 0 & \dots & \beta^{1M} \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & \dots & \alpha^{2N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \beta^{2M} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{45}$$

An example is given to explain the construction and the evolution of the states of the parallel system. For the sake of illustration, a parallel system with 2 components having 3 hidden states each, following the BAKIS model structure. The transition matrices for the two components are defined as:

For the component C_1: $\begin{pmatrix} 1-\sum_{j=2}^{3}\alpha^{1j} & \alpha^{12} & \alpha^{13} \\ 0 & 1-\alpha^{13} & \alpha^{13} \\ 0 & 0 & 1 \end{pmatrix}$

For the component C_2: $\begin{pmatrix} 1-\sum_{j=2}^{3}\beta^{1j} & \beta^{12} & \beta^{13} \\ 0 & 1-\beta^{13} & \beta^{13} \\ 0 & 0 & 1 \end{pmatrix}$

Based on these matrices, the Markov chain of the system is given by Fig. 45.

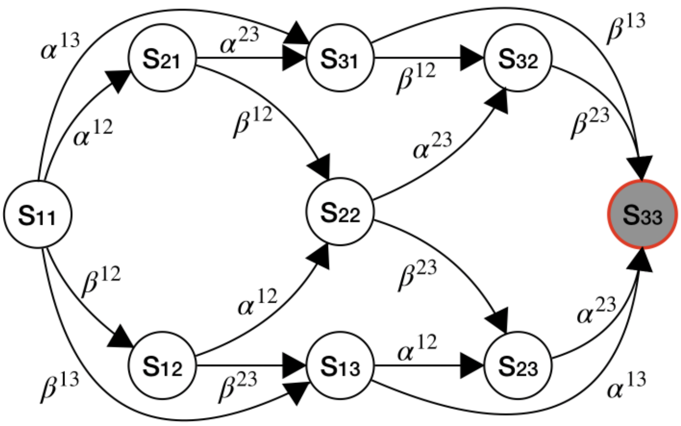


Fig. 45. Transition graph of two parallel components

Figure 45 represents the state transitions of the whole system based on the cartesian product of component states. The system stops only when both components join their state S3ie when the Markov chain reaches state S33. So, following Eq. 45 and the previous remark, the transition matrix is:

$$A_S = \begin{pmatrix} \times & \beta^{12} & \alpha^{12} & 0 & \beta^{13} & \alpha^{13} & 0 & 0 & 0 \\ 0 & \times & 0 & \alpha^{12} & \beta^{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & \times & \beta^{12} & 0 & \alpha^{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & 0 & 0 & \beta^{23} & \alpha^{23} & 0 \\ 0 & 0 & 0 & 0 & \times & 0 & \alpha^{12} & 0 & \alpha^{13} \\ 0 & 0 & 0 & 0 & 0 & \times & 0 & \beta^{12} & \beta^{13} \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & 0 & \alpha^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \times & \beta^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

6.1.4 Parallel structure of two components with IOHMM models

Since the degradation of a component considering the input suggests using IOHMM instead of HMM, the evolution of the health state of the components are unchanged as it described in a series system (Eq. 43a and 43b) as:

$$A_1^p = \begin{bmatrix} 1-\sum_{j=2}^{N}\alpha_p^{1j} & \alpha_p^{12} & \cdots & \alpha_p^{1N} \\ 0 & 1-\sum_{j=3}^{N}\alpha_p^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \alpha_p^{(N-1)N} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C_1, given by $\lambda_1 = (A_1, B_1, \pi_1)$,
The hidden states are $S_1 = \{S_1^1, \dots, S_1^N\}$.

$$A_2^q = \begin{bmatrix} 1-\sum_{j=2}^{M}\beta_q^{1j} & \beta_q^{12} & \cdots & \beta_q^{1M} \\ 0 & 1-\sum_{j=3}^{M}\beta_q^{2j} & \cdots & \vdots \\ 0 & 0 & \cdots & \beta_q^{(N-1)M} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C_2, given by $\lambda_2 = (A_2, B_2, \pi_2)$,
The hidden states are $S_2 = \{S_2^1, \dots, S_2^M\}$

To build the model of a parallel system of two components, the methodology follows the similar procedure but given that each transition matrix is selected among the set A1 (resp. A2) by the input mode p (resp. q) then Eq. 45 becomes:

$$A_S^{pq} = \begin{bmatrix} - & \beta_q^{12} & \dots & \beta_q^{1M} & \alpha_p^{12} & \dots & 0 & \alpha_p^{13} & \dots & 0 \\ 0 & - & \dots & \beta_q^{2M} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & - & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & - & 0 & \dots & \alpha_p^{12} & 0 & \dots & \alpha_p^{1N} \\ 0 & 0 & 0 & 0 & - & \dots & \dots & \alpha_p^{23} & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & - & \dots & 0 & \dots & \beta_q^{1M} \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & \dots & \alpha_p^{2N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \beta_q^{2M} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

To explain the parallel structured system, the same example of two components is taken into account but with parallel connection.

Constructed matrix A_S^{11} for mode one: (construction of matrices A_1^1 and A_2^1**)**

$$A_S^{11} = \begin{array}{c} \\ s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_{13} \\ s_{31} \\ s_{23} \\ s_{32} \\ s_{33} \end{array} \begin{array}{c} \begin{array}{ccccccccc} s_{11} & s_{12} & s_{21} & s_{22} & s_{13} & s_{31} & s_{23} & s_{32} & s_{33} \end{array} \\ \begin{pmatrix} - & \beta_1^{12} & \alpha_1^{12} & 0 & \beta_1^{13} & \alpha_1^{13} & 0 & 0 & 0 \\ 0 & - & 0 & \alpha_1^{12} & \beta_1^{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & - & \beta_1^{12} & 0 & \alpha_1^{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & - & 0 & 0 & \beta_1^{23} & \alpha_1^{23} & 0 \\ 0 & 0 & 0 & 0 & - & 0 & \alpha_1^{12} & 0 & \alpha_1^{13} \\ 0 & 0 & 0 & 0 & 0 & - & 0 & \beta_1^{12} & \beta_1^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & \alpha_1^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \beta_1^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

There is only one state (s_{33}) is the breakdown states when both components are failed. This is (A_S^{11}) the transition matrix that represents the parallel system (shown Fig. 44) for the given operating condition with the first mode.

- Constructed matrix A_S^{22} for mode two: (construction of matrices A_1^2 and A_2^2**)**

$$A_S^{11} = \begin{array}{c} \\ s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_{13} \\ s_{31} \\ s_{23} \\ s_{32} \\ s_{33} \end{array} \begin{array}{c} \begin{array}{ccccccccc} s_{11} & s_{12} & s_{21} & s_{22} & s_{13} & s_{31} & s_{23} & s_{32} & s_{33} \end{array} \\ \begin{pmatrix} - & \beta_2^{12} & \alpha_2^{12} & 0 & \beta_2^{13} & \alpha_2^{13} & 0 & 0 & 0 \\ 0 & - & 0 & \alpha_2^{12} & \beta_2^{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & - & \beta_2^{12} & 0 & \alpha_2^{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & - & 0 & 0 & \beta_2^{23} & \alpha_2^{23} & 0 \\ 0 & 0 & 0 & 0 & - & 0 & \alpha_2^{12} & 0 & \alpha_2^{13} \\ 0 & 0 & 0 & 0 & 0 & - & 0 & \beta_2^{12} & \beta_2^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & \alpha_2^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & \beta_2^{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

(A_S^{22}) is the transition matrix that represents the health evolution of the parallel system (Fig. 44) for the second mode of the input U.

These two matrices (A_S^{11}, A_S^{22}) are the final version of the transition matrices that defines the hidden process of the IOHMM which models a two components parallel system health evolution. It can be used by the prognostic algorithm (Eq. 33) to estimate the RUL.

6.1.5 A drinking water network illustration

The proposed method can also be used for complex structures that contain both the series and parallel components in the same path. For example, a flow distribution system (FDS) can be described which has several components in both connections (series and parallel). For this purpose, we present a subpart of the drinking water network (DWN) of Barcelona city which given by Fig. 46.

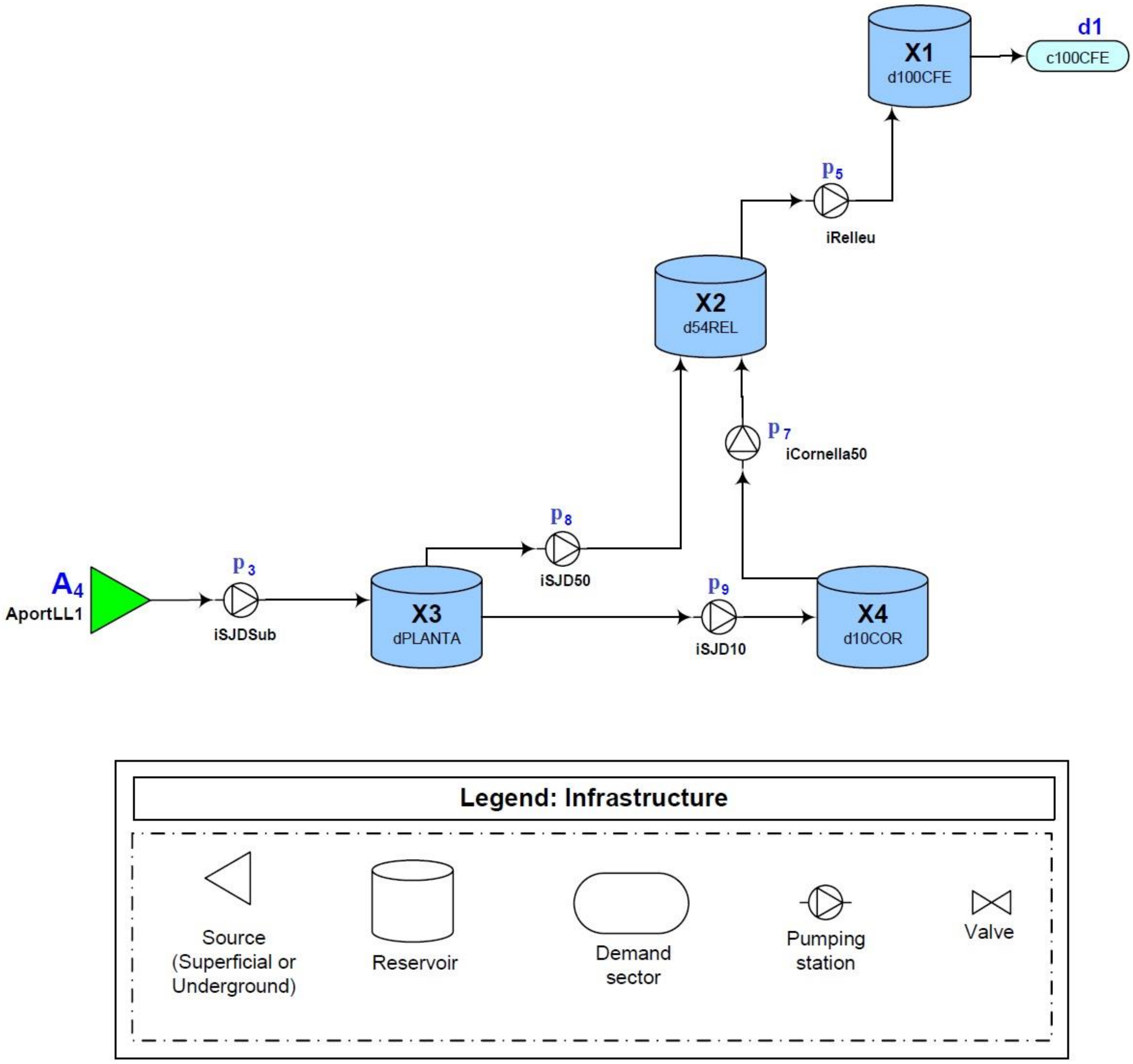


Fig. 46: The considered part of the Barcelona DWN

A DWN is a network that considers sources (supplying water), sinks (water demand points), and pipelines that link sources to sinks. It also contains active elements like pumps and valves. The network covers a territorial extension of 425km2, with a total pipe length of 4,470 km. Every year, it supplies 237.7hm3 of drinking water to a population of over 2.8 million inhabitants. The network has a centralized tele-control system, organized in a two-level architecture. At the upper level, a supervisory control system installed in the control centre of AGBAR is in charge of controlling the whole network by taking into account operational constraints and consumer demands.

The components (sources, sinks, tanks, and pipelines) that presented in Fig. 46 are considered perfectly reliable and without degradation. Only the active elements are subjected to degradation according to time and inputs. One source (*AportLL1*) of water and one sink (*C100CFE*) is considered (see Fig. 46). To supply the sink *C100CFE*, the source is needed through the DWN. So, from the structural point of view, the system should be considered as a series-parallel system because it is a parallel structure of 2 series paths where the paths are the active components from source to the sink.

Starting from the source and following the pipelines to the sink, two paths should be enumerated as follows:

Path 1: {*AportLL1*, *iSJDSub*, *iSJD50*, *iRelieu*} or {*AportLL1*, P3, P8, P5}

Path 2: {*AportLL1*, *iSJDSub*, *iSJD10*, *iCornella50*, *iRelieu*} or {*AportLL1*, P3, P9, P7, P5}

If we ignore the source and the reservoir from Fig. 46, then it highlights only the components (pumps) which given by Fig. 47a. So, the first path contains three components and the second path contains four components.

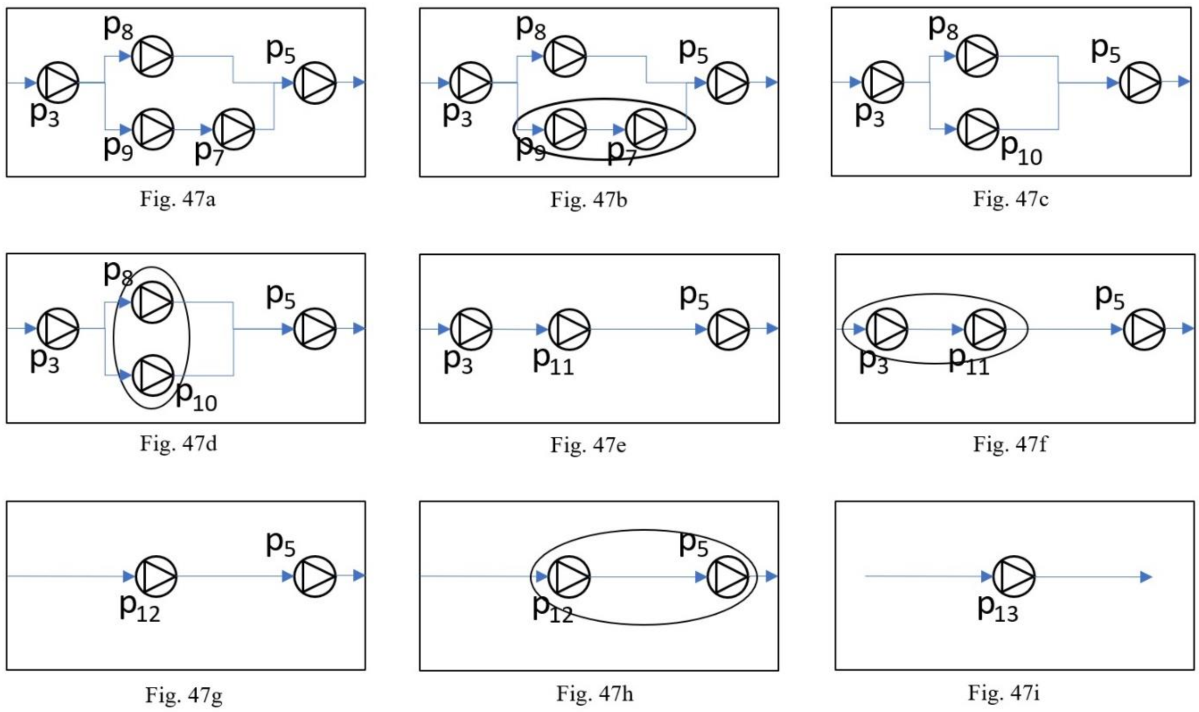


Fig. 47: The steps of model constructions

Figure 47 represents the steps of how more than two components of a system can be constructed through the techniques of two components constructions. For an easy explanation, let us assume the components are modelled by HMM with three hidden states for each.

- In the first step (Fig.47b), two models: p9, p7 are (series) constructed as a single model. Assume a new component as p10 came out of this construction (Fig. 47c). The number of hidden states of the new model is $3 \times 3 = 9$.
- In the second step (Fig.47d), two models: p8, p10 are (parallel) constructed as another single model. Assume another new component as p11 came out of this construction (Fig. 47e) which makes a series system with three components. The number of hidden states of this model is $3 \times 9 = 27$.

- In the third step (Fig.47f), two models: p3, p11 are (series) constructed as another single model. Assume the new component as p12 came out of this construction (Fig. 47g). The number of hidden states of this model is $3 \times 27 = 81$.

- In the final step (Fig.47h), two models: p5, p12 are (series) constructed as the final model. This is the entire system presented as a single component p13 (Fig. 47g) which has a total number of (3×81) 243 hidden states.

Now, this is the final version of the model that represents the entire system with 243 hidden states. The constructed matrix for the model is given below:

	S_1	1					1	2					2	3					3
S_1	S_2	1	2	..	k	..	81	1	2	..	k	..	81	1	2	..	k	..	81
1	1	$\times$	β_{12}	..	β_{1k}	..	β_{1_81}	α_{12}	0	..	0	..	0	α_{13}	0	..	0	..	0
	2	0	$\times$	β_{2-}	β_{2k}	..	β_{2_81}	0	α_{12}	..	0	..	0	0	α_{13}	..	0	..	0
	..	..	..	..	β_{-k}	..	β_{-81}	..	..	α_{12}	..	..	..	..	..	α_{13}	..	..	..
	k	0	0	..	$\times$	β_{k-}	β_{k81}	0	0	..	α_{12}	..	0	0	0	..	α_{13}	..	0
	..	..	..	..	..	..	β_{-81}	..	..	..	..	α_{12}	..	..	..	..	..	α_{13}	..
1	81	0	0	..	0	..	$\times$	0	0	..	0	..	α_{12}	0	0	..	0	..	α_{13}
2	1	0	0	..	0	..	0	$\times$	β_{12}	..	β_{1k}	..	β_{1_81}	α_{23}	0	..	0	..	0
	2	0	0	..	0	..	0	0	$\times$	β_{2-}	β_{2k}	..	β_{2_81}	0	α_{23}	..	0	..	0
	..	..	..	..	..	..	..	..	..	..	β_{-k}	..	β_{-81}	..	..	α_{23}	..	..	..
	k	0	0	..	0	..	0	0	0	..	$\times$	β_{k-}	β_{k81}	0	0	..	α_{23}	..	0
	..	..	..	..	..	..	..	..	..	..	..	..	β_{-81}	..	..	..	..	α_{23}	..
2	81	0	0	..	0	..	0	0	0	..	0	..	$\times$	0	0	..	0	..	α_{23}
3	1	0	0	..	0	..	0	0	0	..	0	..	0	$\times$	β_{12}	..	β_{1k}	..	β_{1_81}
	2	0	0	..	0	..	0	0	0	..	0	..	0	0	$\times$	β_{2-}	β_{2k}	..	β_{2_81}
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	β_{-k}	..	β_{-81}
	k	0	0	..	0	..	0	0	0	..	0	..	0	0	0	..	$\times$	β_{k-}	β_{k81}
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0	..	β_{-81}
3	81	0	0	..	0	..	0	0	0	..	0	..	0	0	0	..	0	..	$\times$

Here, S_1 represents the 3 hidden states of components p5, S_2 represents the 81 hidden states of the component p12. However, the number of hidden states can be reduced by the absorbent states and merging technique described by the Eq.42. However, this was an example of five components system which can be adapted to model systems with any number of components. An algorithm can be developed following the construction methods presented in this chapter. The major challenge of this method is the number of hidden states which increases by the multiplication of the number of hidden states of each components.

6.1.6 Diagnostic

Once the construction is done, the IOHMM uses the constructed matrices to predict the future health state of the system. However, the model needs the diagnostic of the system as well for prognostic the future health states. The diagnostic of the system is computed from the estimated diagnostics of all components applying the Eq. 46.

$$P(S = S_{ij}) = P(C1 = S_i) \times P(C2 = S_j), \text{ for } \forall\, i, j \tag{46}$$

Though, the health states of the system are estimated by path-wise, the diagnostic can be predicted considering each of the paths separately.

The prediction is made in two steps.

Step 01: All the components of a path are diagnosed separately by the Viterbi algorithm.

Step 02: Diagnostic the system health by following the path using Eq. 46.

Once the diagnostic is done, the prognostic algorithm uses it to estimate the RUL based on the operating conditions. Finally, one path among all is getting selected which gives the maximum RUL compared to other paths.

A simulated application is given in the next section to demonstrate the proposed methodology of estimating the RUL of a structured system.

6.2 Application

This application is dedicated to illustrating our proposed methodology. For this purpose, we focus on the first construction part which is represented by the Fig 47.b. Following that, a series structured system (Fig.48) is simulated that has two components with a serial connection.

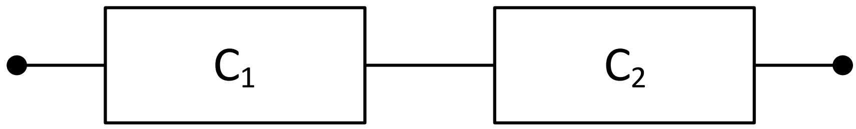


Fig. 48: Series component system

Despite keeping the same number of hidden states for each component as three, some other elements need to be assumed in order to simulate the application. Each component is assumed to have two outputs and one input (operating condition) with two modes. The operating condition provides two transition matrices, and the outputs provide two emission matrices for each component. There are four discrete symbols considered in the emitted observation sequences. However, the outputs could have a different number of discrete symbols as well. This assumption can vary depending on different application scenarios.

6.2.1 Data simulation

Two sets of training sequences are simulated for two components. All the transition matrices that are used in data simulations are 3-by-3 matrices corresponding to three hidden states. The transition matrices are sup-triangular matrices because of the BAKIS property.

The initial state distributions are assumed as $\pi = (1 \quad 0 \quad 0)$, which represents the components are in good health state at the beginning. The sequences are generated as complete form with no missing data consideration. The length of the sequence is selected randomly between 250 to 300 for each. The length can be longer.

About 1000 training data sequences are generated where each one of them assumed to have information about system failure. Another dataset (1000 sequences) is generated for testing purposes.

6.2.2 Model Learning

The IOHMM learns two models for each of the components according to the number of operating conditions. The training is done following the methodologies (Eq. 15 to Eq. 19) described in chapter 4. So, the number of learned models is two for each component.

For the 1st operating condition

Two transition matrices $(\hat{A}_1^1, \hat{A}_2^1)$ learned from two components.

- Transition parameters for component C_1:

$$\hat{A}_1^1 = \begin{pmatrix} 0.9585 & 0.0415 & 0 \\ 0 & 0.9499 & 0.0501 \\ 0 & 0 & 1 \end{pmatrix}$$

- Transition parameters for component C_2:

$$\hat{A}_2^1 = \begin{pmatrix} 0.9818 & 0.0182 & 0 \\ 0 & 0.9738 & 0.0262 \\ 0 & 0 & 1 \end{pmatrix}$$

The constructed transition matrix for the system is:

$$\tilde{A}^1 = \begin{array}{c} \\ s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_f \end{array} \begin{array}{c} \begin{array}{ccccc} s_{11} & s_{12} & s_{21} & s_{22} & s_f \end{array} \\ \begin{pmatrix} \times & 0.0182 & 0.0415 & 0 & 0 \\ 0 & \times & 0 & 0.0415 & 0.0262 \\ 0 & 0 & \times & 0.0182 & 0.0501 \\ 0 & 0 & 0 & \times & 0.0763 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

The diagonal of the matrix ($\tilde{A}^1$) is calculated as [1 - (*sum of all probabilities on the same row*)] because of the stochastic property. For example, (s_{11}, s_{11}) = 1- (0.0182 + 00415) = 0.9403. The probabilities of getting into the last state (s_f) are ignored because the method for the RUL estimation stops when the model reaches the final states for the first time. Therefore, the matrix is formatted stochastic without the last state.

The final version of the matrix $\tilde{A}^1$ with diagonal distributions:

$$\begin{array}{c} \\ s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_f \end{array} \begin{array}{c} \begin{array}{ccccc} s_{11} & s_{12} & s_{21} & s_{22} & s_f \end{array} \\ \begin{pmatrix} 0.9403 & 0.0182 & 0.0415 & 0 & 0 \\ 0 & 0.9323 & 0 & 0.0415 & 0.0262 \\ 0 & 0 & 0.9317 & 0.0182 & 0.0501 \\ 0 & 0 & 0 & 0.9237 & 0.0763 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

For the 2nd operating condition

- Transition parameters for component C_1:

$$\hat{A}_1^2 = \begin{pmatrix} 0.9151 & 0.0849 & 0 \\ 0 & 0.9048 & 0.0952 \\ 0 & 0 & 1 \end{pmatrix}$$

- Transition parameters for component C_2:

$$\hat{A}_2^2 = \begin{pmatrix} 0.9088 & 0.0912 & 0 \\ 0 & 0.9018 & 0.0982 \\ 0 & 0 & 1 \end{pmatrix}$$

The constructed transition matrix for the system is:

$$\tilde{A}^2 = \begin{array}{c} \\ s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_f \end{array} \begin{array}{c} \begin{array}{ccccc} s_{11} & s_{12} & s_{21} & s_{22} & s_f \end{array} \\ \begin{pmatrix} \times & 0.0912 & 0.0849 & 0 & 0 \\ 0 & \times & 0 & 0.0849 & 0.0982 \\ 0 & 0 & \times & 0.0912 & 0.0952 \\ 0 & 0 & 0 & \times & 0.1934 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

The diagonal of the matrix ($\tilde{A}^2$) is also computed following the same formula: [1 - (*sum of all probabilities on the same row*)].

The final version of the matrix $\tilde{A}^2$ with diagonal distributions:

$$\begin{array}{c c} & \begin{array}{ccccc} s_{11} & s_{12} & s_{21} & s_{22} & s_f \end{array} \\ \begin{array}{c} s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_f \end{array} & \begin{pmatrix} 0.8239 & 0.0912 & 0.0849 & 0 & 0 \\ 0 & 0.8169 & 0 & 0.0849 & 0.0982 \\ 0 & 0 & 0.8136 & 0.0912 & 0.0952 \\ 0 & 0 & 0 & 0.8066 & 0.1934 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Now the constructed matrices ($\tilde{A}^1$and $\tilde{A}^2$) are ready for use in the prognostic algorithm. However, the algorithm requires the current health state of the system which can be estimated from the diagnostic of the components by applying the proposed algorithm (Algo. 02). The next section presents the diagnostic results.

6.2.3 Diagnostic

The components C_1 and C_2 are diagnosed separately under the operating conditions at the given test sequences. One sequence for each component is selected randomly from the test set. The estimated diagnostics applied the Viterbi algorithm are given below:

The current health states distribution of C_1: (s_1= 0.05; s_2= 0.57; s_3= 0.00), which is mainly partially degraded into second state.

The current health states distribution of C_2: (s_1= 0.00; s_2= 0.42; s_3= 0.03), which is partially degraded into second state. These are the raw values that are not scaled to one. The scaled values are:

For C_1: (s_1= 0.08; s_2= 0.92; s_3= 0.00), and for C_2: (s_1= 0.00; s_2= 0.93; s_3= 0.07)

Following the maximum value in the distribution, both the components are at state 2 (moderate state). A cumulative degradation is presented in Table 15 for the entire system applying Eq. 46.

Table 15: Computed cumulative degradation

Component 1 state	Component 2 state	System state	Diagnostic: state by state
s_1= 0.08	s_1= 0.00	s_{11}	$s_1 \times s_1$= 0.0000
s_1= 0.08	s_2= 0.93	s_{12}	$s_1 \times s_2$= 0.0744
s_2= 0.92	s_1= 0.00	s_{21}	$s_2 \times s_1$= 0.0000
s_2= 0.92	s_2= 0.93	s_{22}	$s_2 \times s_2$= 0.8556
s_1= 0.08	s_3= 0.07	s_{13}	$s_1 \times s_3$= 0.0056
s_3= 0.00	s_1= 0.00	s_{31}	$s_3 \times s_1$= 0.0000
s_2= 0.92	s_3= 0.07	s_{23}	$s_2 \times s_3$= 0.0644
s_3= 0.00	s_2= 0.93	s_{32}	$s_3 \times s_2$= 0.0000

So, the diagnostic of the system is (s_{11}= 0.0000; s_{12}= 0.0744; s_{21}= 0.0000; s_{22}= 0.8556; s_{13}= 0.0056; s_{31}= 0.0000; s_{23}= 0.0644; s_{32}= 0.0000), where the summation of the distribution is one. However, since "$s_{13}, s_{31}, s_{23}, s_{32}$" states merged in s_f and we are only considering the working states, the diagnostic vector for the entire system would be (s_{11}= 0.00; s_{12}= 0.0744; s_{21}= 0.00; s_{22}= 0.8556). This is the distributions of the current health state of the system which indicates that the system is in state s_{22} that is not the final state. That means the system is still has some useful life remain.

6.2.4 Prognostic

The prognostic algorithm performs only if the system is still on working state. Since the current health state of the system is estimated as a working state, the prognostic algorithm (Eq. 17) is used to predict RUL predictions.

6.2.4.1 Offline prognostic

Since there is no input sequence for switching the future operating conditions, the vector of future state transition can be estimated twice following the two operating modes. This is the case when the IOHMM acts like the classical HMM because no input sequence is given to switch the models for further computation.

The difference between the two operating conditions is the stress level. The low-stress condition (*operating condition 1*) gives about 73 days of RUL (at the current time) while the high-stress condition (*operating condition 2*) gives about 28 days for the same system. This is an offline prognostic in which the prediction is done once at the current time to define the RUL bound. So, [28, 73] can be defined as the approximate bounds for the future RUL prediction of the system according to the current health states. That means, all the possible combinations of operating conditions would provide the meantime RUL between 28 and 73 days.

6.2.4.2 Online prognostic

The RUL is updated according to the change's aspects of health states over the run-time. This is defined as online prognostic.

For functioning the online prognostic, the diagnostic is also estimated online in which the current health states of the system are updated for a new observation. Two sequences are randomly selected from the simulated datasets for online diagnostic.

The estimated RULs are stored in a vector to use in the online prognostics. The vector is being used to predict the RUL that presented in Fig. 49.

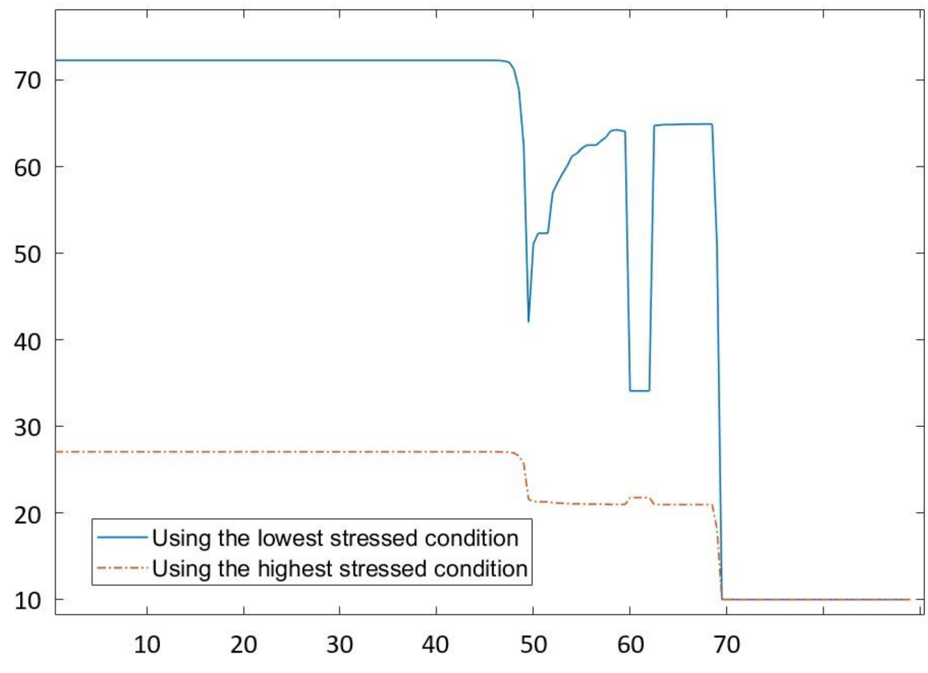


Fig. 49: Online RUL estimation of the FDS

The lowest stressed operating condition shows the RUL which is always higher (the strait blue line) than the RUL coming from the highest stressed operating condition (the dotted red line). This graph can be used as the bound of online RUL estimation.

This is an example where the experiment showed that, once the model construction is done it can be used to offline and online prognostic as like the general application scenario given in chapter 4. Similarly, this concept can be applied to such complicated systems considering multiple components in series and parallel connections with the uncertainties, *e.g.* the data uncertainties (small amount, missing data, etc.), the operating conditions uncertainties (given operating conditions, unknown operating conditions, etc.), model uncertainties (model size or the number of hidden states), etc. All these uncertainties considered while the individual components are diagnosed. Once the diagnostic is done, the model construction then performs according to the operating conditions and the connections between the components. And finally, the RUL estimation can be done by using the proposed methodology.

6.3 Conclusion

RUL assessment of a structured system provides information of different paths to supply the demand to the destination by considering the less degradation corresponding to the other paths. It also allows us to take maintenance decisions to prevent the unexpected failure of any component. Some industries often want to observe all the components independently by installing multiple sensors on each component (Xi, 2019). In this contribution, all the system components are observed individually to prognostic the overall system's health which lets us monitor the components individually as well as the entire system.

However, it is not an easy task to prognostic health states of structured systems with multiple components. Multiple components work together as a system in which each of them has its own degradation. Therefore, gaining knowledge about the health condition of the system is difficult especially under multiple operating conditions. In this chapter, a method based on IOHMM is proposed to model such a structured system with input conditions. The main challenges for predicting RUL of the structured system has been explained with examples.

All the components are modelled by separate IOHMMs and diagnosed following the adapted algorithms proposed in chapter 4. The components are assumed as independent of each other. That is why the current health states of the components are predicted separately and used them to compute the health state of the entire system. An algorithm is proposed to diagnostic structured system based on the diagnostic of its components.

The models for each component are stored and constructed together which represents the system and used to prognostic system health. The construction is done by (supplying) path-wise because this kind of system has multiple paths to supply the same demand. Each of the paths may contain multiple components with series or parallel connections or both. The model construction is different for series and parallels connected components. The construction of both connections is described with two systems (series system and parallel system).

Finally, the constructed model is being used to prognostic the RUL of the system given the operating condition. Prognostic health states of the structured system considering multiple components are useful to make a maintenance planning. This is one of the most appropriate approaches for maintenance planning where the components get monitored individually (Verbert, 2017).

To adapt for the real case of DWN, a numerical application is presented as FDS which has serially connected components. However, the proposed method can be applied to the parallel system components as well. As future work, this method can be extended to a larger network of structure system that has more components with both the serial and parallel connections.

Conclusion

The work presented in this thesis is a contribution to the system health state diagnostic, prognostic and Residual Useful Life estimation of components and structured systems by using a dynamic probabilistic graphical model. The thesis work is in the class of data-driven based model and more particularly in an advanced form of Hidden Markov Model to consider data series, i.e. Input-Output Hidden Markov Models.

More specifically, our contributions focus on the introduction of operating conditions into the model to estimate the system health state subjected to different dynamics of degradation. The basic algorithms are adapted to consider these operating conditions as inputs, therefore multiple outputs and multiple inputs sequences are used to estimate the parameters of IOHMM. The model is then used to compute a probability distribution according to the time that is considered in this work as the diagnostic and the prognostic of the system health state. The computation of the RUL from the model considering the input conditions is then proposed. At last, the work concerned the passage from the component level to the system level for the prognostic of the system RUL.

In the introduction of the thesis, we have defined our work in the context of Prognostics and Health Management which is part of the concept of Factory of Future. We briefly justified our choice of stochastic models and data-driven approaches then exhibited 3 research questions:

1. Considering health state of a system with whatever the system complexity is by a stochastic model and learn model parameters from system measurements.
2. Diagnostic and prognostic of the system health under multiple operating conditions.
3. Prognostic the RUL for structured systems from its components to study the entire system reliability.

The first chapter is concentrated on the state of the art. We recalled some key notions like maintenance, degradation, diagnostic and prognostic. The chapter showed a large panel of works concerning the prognostic methodologies. The main classes of PHM approaches and the model types are also reviewed to justify more deeply our choice of stochastic models. It gives arguments to our choice of a data-driven method and more specifically the Hidden Markov Model-based approaches. It defines the limitations of existing HMM-based RUL estimators and indicates a way out from that by proposing the IOHMM.

The second chapter was dedicated to providing the main mathematical background of Hidden Markov Model and the well-known algorithms to train (estimate the parameters of the model) and use HMM. This chapter defines the notation used in the following chapters.

The third chapter described our first contribution on algorithms adaptation for several cases. We considered first the case of multiple input conditions and the impact on the algorithms based on the mathematical formulations. Then we considered the case of multiple sequences of data. To complete the contribution, we adapted the previous algorithms to the case of multiple outputs. Numerical illustrations are provided considering a well-managed case of data that we produce. We discuss the definition of the model structure according to the data and the definition of the number of health states to be considered in the model. This section highlights that the model structure needs to be adapted to the information given by the data set and it cannot model behaviours of the system that are not enough represented in data. We also considered the case of missing data in order to be close to a real industrial situation. In addition, we introduced a bootstrap learning approach to provide a level of credibility of the estimated model.

Three numerical applications were presented to demonstrate the IOHMM learning algorithms to highlight three challenges in the training. The first application is shown to represent the complete data set where the observation sequences are clean if there is no missing data available. The second application was presented focusing on handling the missing data in the training sequences. The third application was presented to learn the model parameters by using the bootstrap method which estimates each of the parameters of the model with 95% confidence intervals. All three applications are simulated from given model structures which compared with the result of the model in the application section.

The fourth chapter was dedicated to our second contribution concerning the diagnostic and the prognostic with IOHMM models. We described how to diagnose the health state of the system or component from the observations with or without missing data. We also described different approaches to compute the health state evolution and the Residual Useful Life considering multiple operating conditions. Two applications were provided to illustrate first the diagnostic and prognostic under multiple conditions and secondly an illustration of input management process to simply control the RUL to match a maintenance time window.

The fifth chapter concerned a practical contribution based on the PHM Challenge 2008. As the data of the PHM challenge cannot be used directly by the stochastic model, a preparation step has been realized before training the model. Some issues about the model structure and parameter estimations were discussed. The diagnostic and Meantime RUL estimations were provided, scored and discussed.

The sixth chapter concerned our answer to the third research question: How to prognostic the health state of a system from the health state and model of its components. The chapter has described how the structure of a system defines the way is it possible to combine the components' IOHMM model. We focused on series and parallel structures but the IOHMM model can have any number of hidden states, observation states and input modes. The system model built serves to compute the system health state by diagnosing the components but « prognosing » the system. The combination of all the degraded states of the components result in a large number of states to define the global system. All these states are used to define the dimension of several matrixes according to the inputs (operating condition of the system) but these large matrixes are obtained by mathematical computations. Thereafter, the RUL computation is based on the combination of the component’s degradations with respect to the parallel and series structure of the global model.

Perspectives

The work accomplished can be consolidated in the future along with several appearances. The main perspectives are listed out upon the short term and long-term basis.

Short-term perspectives:

- Managing the operating conditions on real applications to control the production speed and cost. This perspective falls in the control theory area. Computing the RUL will serve to control the system through techniques like Model Predictive Control (Sun, 2015) or controlling systems by extremum seeking algorithms (Rotea, 2000).
- Practice alternative learning approaches and report the pro and cons for each of their strategy and performance. We currently adapted well-known historical algorithms, but other learning algorithms can give interesting results in the context of IOHMM. For instance, metaheuristics applied to HMM (Aupetit, 2005) can be extended to IOHMM.
- Control the prediction error obtained by several techniques to improve parameter learning algorithms. The focus would be on minimizing the prediction error (Dragomir, 2008). This control of the prediction error seems essential because it would allow us to validate our work on a real system and address the time implementation constraint.

Long-term perspectives:

- Since the prior knowledge is a key depending issue for maintenance scheduling/rescheduling, this approach can be applied to manage the system maintenance. This idea relies on some works of the lab managed by Pr. B. Iung, Pr. Levrat, or Dr. Do (Nguyen, 2018; Thomas, 2009). We showed that the RUL can be partially managed by tuning the inputs. This is perhaps a way of joining an opportune and predictive maintenance with input management.
- Develop approaches that can overcome the discrete data conversion from continuous data. In other words: improving learning models in the presence of the original signal data. The discretization of continuous physical data to match discrete state of our model can be discussed hardly. It has been really developed in the thesis accepted in the application to the PHM challenge. Some works as (Turin W., 2012; Jie Zhou and Xinyuan Song 2020) concern continuous HMM. The dependence of the hidden process to environmental conditions remains. A possible and interesting work concerns the development of continuous IOHMM.
- Actually, we have considered only the causal dependence between the hidden states and the observation states. This assumption in the model structure simplified all the equations used all along the thesis. Nevertheless, in order to be more general, we should consider a possible dependency between the inputs and the outputs. It will amplify the mathematical model complexity but will better represent real systems and fewer assumptions.
- Contrary to (Le https://hal.archives-ouvertes.fr/hal-01027509), we considered a Markovian model even if we introduce the input to condition the Hidden Process. Markovian property is a strong assumption and implies that state distribution follows exponential laws. This assumption takes us away a bit from reality. Nevertheless, it is possible to increase the number of hidden states (fictive states) to lower this. Unfortunately, more hidden states, means more data required to train models and more uncertainty in parameters estimation, diagnostic and prognostic. A perspective is to combine Input-Output principle with Hidden semi-Markov Models to build IOHsMM.

Reference

1. Ahmad, Rosmaini, and Shahrul Kamaruddin. "An overview of time-based and condition-based maintenance in industrial application." *Computers & industrial engineering* 63.1 (2012): 135-149.
2. Aidala, Vincent J. "Kalman filter behavior in bearings-only tracking applications." *IEEE Transactions on Aerospace and Electronic Systems* 1 (1979): 29-39.
3. Akaike, Hirotugu. "A new look at the statistical model identification." *IEEE transactions on automatic control* 19.6 (1974): 716-723.
4. Albertos, Pedro, and Graham C. Goodwin. "Virtual sensors for control applications." *Annual Reviews in Control* 26.1 (2002): 101-112.
5. Allison, P.D., 2001. Missing data (Vol. 136). Sage publications.
6. Amiri, Moslem, and Václav Přenosil. "General solutions for MTTF and steady-state availability of NMR systems." In *2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, pp. 1-4. IEEE, 2014
7. Aupetit, Sébastien. *Contributions aux modèles de Markov cachés: Métaheuristiques d'apprentissage, nouveaux modèles et visualisation de dissimilarité*. Diss. 2005.
8. Bain, Alexander. *Mind and body: The theories of their relation*. Vol. 4. Henry S. King, 1873.
9. Baker, Alan. "Occam's Razor in science: a case study from biogeography." *Biology & Philosophy* 22.2 (2007): 193-215.
10. Balali, Farhad, Hamid K. Seifoddini, and Adel Nasiri. "Essentials to develop data-driven predictive models of prognostics and health management for distributed electrical systems." *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2019.
11. Balali, Farhad, Hamid Seifoddini, and Adel Nasiri. "Data-driven predictive model of reliability estimation using degradation models: a review." *Life Cycle Reliability and Safety Engineering* (2020): 1-13.
12. Banks, Jeffrey, and John Merenich. "Cost benefit analysis for asset health management technology." *2007 annual reliability and maintainability symposium*. IEEE, 2007.
13. Baraldi, Piero, Francesca Mangili, and Enrico Zio. "A prognostics approach to nuclear component degradation modeling based on Gaussian Process Regression." *Progress in Nuclear Energy* 78 (2015): 141-154.
14. Barber, David. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.
15. Baruah, Pundarikaksha, and Ratna B. Chinnam*. "HMMs for diagnostics and prognostics in machining processes." *International Journal of Production Research* 43.6 (2005): 1275-1293.
16. Basia, Akash, et al. "Overcoming the Barriers in Diagnostics and Prognostics of the Circular Industrial System by Hidden Markov Model." *2019 International Conference on Control, Automation and Diagnosis (ICCAD)*. IEEE, 2019.
17. Baum, Leonard E., and Ted Petrie. "Statistical inference for probabilistic functions of finite state Markov chains." *The annals of mathematical statistics* 37.6 (1966): 1554-1563.
18. Bayes, Thomas. "LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S." *Philosophical transactions of the Royal Society of London* 53 (1763): 370-418.

19. Bayraktaroglu, Ismet, and Alex Orailoglu. "Cost-effective deterministic partitioning for rapid diagnosis in scan-based BIST." *IEEE design & test of computers* 19.1 (2002): 42-53.

20. Belkacem, Lobna, et al. "Diagnostic and prognostic of hybrid dynamic systems: Modeling and RUL evaluation for two maintenance policies." *Reliability Engineering & System Safety* 164 (2017): 98-109.

21. Bengio, Yoshua, and Paolo Frasconi. "An input output HMM architecture." *Advances in neural information processing systems*. 1995.

22. Birkel, Hendrik Sebastian, and Evi Hartmann. "Impact of IoT challenges and risks for SCM." *Supply Chain Management: An International Journal* (2019).

23. Bishop, Christopher M. "Pattern recognition and machine learning: springer New York." (2006).

24. Bloch-Mercier, Sophie. "A preventive maintenance policy with sequential checking procedure for a Markov deteriorating system." *European Journal of Operational Research* 142.3 (2002): 548-576.

25. Bonner, Mike. "What is industry 4.0 and what does it mean for my manufacturing?" *Saint Claire Systems. Available online: https://blog. viscosity. com/blog/what-is-industry-4.0-and-what-does-it-mean-for-mymanufacturing (accessed on 29 December 2018)* (2017).

26. Brandt, Jens, and Bernd Hitzmann. "Knowledge-based fault detection and diagnosis in flow-injection analysis." *Analytica chimica acta* 291.1-2 (1994): 29-40.

27. Bressel, Mathieu, et al. "Extended Kalman filter for prognostic of proton exchange membrane fuel cell." *Applied Energy* 164 (2016): 220-227.

28. Camci, Fatih, and Ratna Babu Chinnam. "Hierarchical HMMs for autonomous diagnostics and prognostics." *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006.

29. Chebel-Morello, Brigitte, Karim Haouchine, and Noureddine Zerhouni. "A methodology to conceive a case-based system of industrial diagnosis." *Engineering Asset Lifecycle Management*. Springer, London, 2010. 474-486.

30. Chen, Kui, Salah Laghrouche, and Abdesslem Djerdir. "Degradation model of proton exchange membrane fuel cell based on a novel hybrid method." *Applied Energy* 252 (2019): 113439.

31. Chen, Kui, Salah Laghrouche, and Abdesslem Djerdir. "Fuel cell health prognosis using Unscented Kalman Filter: Postal fuel cell electric vehicles case study." *International Journal of Hydrogen Energy* 44.3 (2019): 1930-1939.

32. Chen, Tianrui, David John Hill, and Cong Wang. "Distributed fast fault diagnosis for multimachine power systems via deterministic learning." *IEEE Transactions on Industrial Electronics* 67.5 (2019): 4152-4162.

33. Chen, Chin-Tai, Yi-Wen Chen, and John Yuan. "On a dynamic preventive maintenance policy for a system under inspection." *Reliability Engineering & System Safety* 80.1 (2003): 41-47.

34. Cheng, Yujie, Noureddine Zerhouni, and Chen Lu. "A prognostic framework for PEMFC based on least squares support vector regression-particle filter." *2017 IEEE Vehicle Power and Propulsion Conference (VPPC)*. IEEE, 2017.

35. Churbanov, Alexander, and Stephen Winters-Hilt. "Implementing EM and Viterbi algorithms for Hidden Markov Model in linear memory." *BMC bioinformatics* 9.1 (2008): 224.

36. Couallier, Vincent, Laurent Denis, and Franck Bayle. "Calcul de MTBF des systèmes aéronautiques inspectés périodiquement soumis à rupture aléatoire de systèmes de régulation." *Congrès Lambda Mu 20 de Maîtrise des Risques et de Sûreté de Fonctionnement, 11-13 Octobre 2016, Saint Malo, France* (2016).

37. Dalla Vedova, M. D. L., P. Maggiore, and F. Marino. "Electrohydraulic actuators affected by multiple failures: proposal of an alternative model-based prognostic paradigm." (2018).

38. Di Rito, G., F. Schettini, and R. Galatolo. "Model-based prognostic health-management algorithms for the freeplay identification in electromechanical flight control actuators." *2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE, 2018.

39. Ding, Fangfang, et al. "An integrated approach for wind turbine gearbox fatigue life prediction considering instantaneously varying load conditions." *Renewable energy* 129 (2018): 260-270.

40. Djurdjanovic, Dragan, Jay Lee, and Jun Ni. "Watchdog Agent—an infotronics-based prognostics approach for product performance degradation assessment and prediction." *Advanced Engineering Informatics* 17.3-4 (2003): 109-125.

41. Dong, Ming, and David He. "A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology." *Mechanical systems and signal processing* 21.5 (2007): 2248-2266.

42. Drozdowicz, B., et al. "A neural network-based model for the analysis of carbon monoxide contamination in the urban area of Rosario." *WIT Transactions on Ecology and the Environment* 21 (1997).

43. Du, Wenhua, et al. "A new fuzzy logic classifier based on multiscale permutation entropy and its application in bearing fault diagnosis." *Entropy* 22.1 (2020): 27.

44. Dunin-Barkowski, W. L., Yu A. Flerov, and L. L. Vyshinsky. "Prognosis of dynamical systems behavior based on cerebellar-type neural technologies." *Optical Memory and Neural Networks* 20.1 (2011): 43-58.

45. Ekanayake, Thushara, et al. "Model-based fault diagnosis and prognosis of dynamic systems: a review." *Procedia Manufacturing* 30 (2019): 435-442.

46. Eker, Omer F., Fatih Camci, and Ian K. Jennions. "Physics-based prognostic modelling of filter clogging phenomena." *Mechanical Systems and Signal Processing* 75 (2015): 395-412.

47. El Hajj, Boutros, et al. "A risk-oriented degradation model for maintenance of reinforced concrete structure subjected to cracking." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 230.5 (2016): 521-530.

48. Elattar, Hatem M., Hamdy K. Elminir, and Alaa Mohamed Riad. "Conception and implementation of a data-driven prognostics algorithm for safety–critical systems." *Soft Computing* 23.10 (2019): 3365-3382.

49. Ellefsen, André Listou, et al. "Validation of data-driven labeling approaches using a novel deep network structure for remaining useful life predictions." *IEEE Access* 7 (2019): 71563-71575.

50. Engel, Stephen J., et al. "Prognostics, the real issues involved with predicting life remaining." *2000 ieee aerospace conference. proceedings (cat. no. 00th8484)*. Vol. 6. IEEE, 2000.

51. Esmaeili, Sajjad, et al. "A data-driven model for predicting the effect of temperature on oil-water relative permeability." *Fuel* 236 (2019): 264-277.

52. Finz, G., G. Fronza, and A. Spirito. "Multivariate stochastic models of sulphur dioxide pollution in an urban area." *Journal of the Air Pollution Control Association* 30.11 (1980): 1212-1215.

53. Fisher, Bernard. "Fuzzy environmental decision-making: applications to air pollution." *Atmospheric Environment* 37.14 (2003): 1865-1877.

54. Forman, R. G. "Study of fatigue crack initiation from flaws using fracture mechanics theory." *Engineering Fracture Mechanics* 4.2 (1972): 333-345.

55. Friedman, Nir, and Moises Goldszmidt. "Discretizing continuous attributes while learning Bayesian networks." *ICML*. 1996.

56. Gagniuc, Paul A. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

57. Garcia, E. Alcorta, and Paul M. Frank. "Deterministic nonlinear observer-based approaches to fault diagnosis: a survey." *Control Engineering Practice* 5.5 (1997): 663-670.

58. Gebraeel, Nagi, et al. "Residual life predictions from vibration-based degradation signals: a neural network approach." *IEEE Transactions on industrial electronics* 51.3 (2004): 694-700.

59. Gershman, Samuel J., and David M. Blei. "A tutorial on Bayesian nonparametric models." *Journal of Mathematical Psychology* 56.1 (2012): 1-12.

60. Gillian, Nicholas, Benjamin Knapp, and Sile O'modhrain. "Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping." *Nime*. 2011.

61. Gronau, Norbert, Marcus Grum, and Benedict Bender. "Determining the optimal level of autonomy in cyber-physical production systems." *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*. IEEE, 2016.

62. Gu, Jie, et al. "Application of grey prediction model for failure prognostics of electronics." *International Journal of Performability Engineering* 6.5 (2010): 435-442.

63. Hao, Songhua, Jun Yang, and Christophe Berenguer. "Degradation analysis based on an extended inverse Gaussian process model with skew-normal random effects and measurement errors." *Reliability Engineering & System Safety* 189 (2019): 261-270.

64. Heckerman, David, Dan Geiger, and David M. Chickering. "Learning Bayesian networks: The combination of knowledge and statistical data." *Machine learning* 20.3 (1995): 197-243.

65. Heidary, Roohollah, et al. "A review of data-driven oil and gas pipeline pitting corrosion growth models applicable for prognostic and health management." *Int. J. Progn. Health Manag* 9.1 (2018).

66. Hermann, Mario, Tobias Pentek, and Boris Otto. "Design principles for industrie 4.0 scenarios." *2016 49th Hawaii international conference on system sciences (HICSS)*. IEEE, 2016.

67. Hess, Andrew, et al. "Challenges, issues, and lessons learned chasing the “Big P”: real predictive prognostics part 2." *2006 IEEE Aerospace Conference*. IEEE, 2006.

68. Shoji, Hikaru, and Tsuneo Tsukatani. "Statistical model of air pollutant concentration and its application to the air quality standards." *Atmospheric Environment (1967)* 7.5 (1973): 485-501.

69. Hong-feng, Wang. "Prognostics and Health Management for Complex system Based on Fusion of Model-based approach and Data-driven approach." *Physics Procedia* 24 (2012): 828-831.

70. Hu, Jiawei, Yanfeng Wang, and Ya Zhang. "IOHMM for location prediction with missing data." *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015.

71. Hu, Chang-Hua, et al. "A prognostic model based on DBN and diffusion process for degrading bearing." *IEEE Transactions on Industrial Electronics* (2019).

72. Hu, Chao, et al. "Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life." *Reliability Engineering & System Safety* 103 (2012): 120-135.

73. Huang, Rongqing, and Shiliang Sun. "Sequential training of semi-supervised classification based on sparse Gaussian process regression." *2012 International Conference on Machine Learning and Cybernetics*. Vol. 2. IEEE, 2012.

74. Huang, Tingting, Li Wang, and Tongmin Jiang. "Prognostics of products using time series analysis based on degradation data." *2010 Prognostics and System Health Management Conference*. IEEE, 2010.

75. Irwin, G. R., and P. C. Paris. "Fundamental aspects of crack growth and fracture." *Engineering Fundamentals and Environmental Effects*. Academic Press, 1971. 1-46.

76. ISO13381-1, Condition monitoring and diagnostics of machines prognostics - Part1: General guidelines, International Standard, ISO, 2004.

77. Iung, Benoît. "Factory of the future: Industrial and/or cultural revolution?." *60th Diamond Jubilee National Convention of IIIE and International Conference on “Role of Industrial Engineering in Industry 4.0 Paradigm, ICIEIND 2018*. 2018.

78. James, Jonathan. "MM algorithm for general mixed multinomial logit models." *Journal of Applied Econometrics* 32.4 (2017): 841-857.

79. James, W. "The Principles of Psychology. New York: H. Holt and Company." (1890).

80. Jardine, Andrew KS, Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance." *Mechanical systems and signal processing* 20.7 (2006): 1483-1510.

81. Jensen, F. V. "Bayesian Networks and Decision Graphs Springer-Verlag New York Inc." (2001).

82. Jianzhong, S. U. N., et al. "A data-driven health indicator extraction method for aircraft air conditioning system health monitoring." *Chinese Journal of Aeronautics* 32.2 (2019): 409-416.

83. Jordan, Michael I., et al. "An introduction to variational methods for graphical models." *Learning in graphical models*. Springer, Dordrecht, 1998. 105-161.

84. Jouin, Marine, et al. "Prognostics and Health Management of PEMFC–State of the art and remaining challenges." International Journal of Hydrogen Energy 38.35 (2013): 15307-15317.

85. Just, Agnes, Olivier Bernier, and Sébastien Marcel. *HMM and IOHMM for the recognition of mono-and bi-manual 3D hand gestures*. No. REP_WORK. IDIAP, 2004.

86. Kalgren, Patrick W., et al. "Defining PHM, a lexical evolution of maintenance and logistics." *2006 IEEE autotestcon*. IEEE, 2006.

87. Kenneth , Lange. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, 2016.

88. Keselj, Vlado. "Speech and Language Processing Daniel Jurafsky and James H. Martin (Stanford University and University of Colorado at Boulder) Pearson Prentice Hall, 2009, xxxi+ 988 pp; hardbound, ISBN 978-0-13-187321-6, $115.00." (2009): 463-466.

89. Khumprom, Phattara, and Nita Yodo. "A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm." *Energies* 12.4 (2019): 660.

90. Klingelschmidt, Timothée, et al. "Fault diagnosis and prognosis by using Input-Output Hidden Markov Models applied to a diesel generator." *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2017.

91. Knights, Shanna D., et al. "Aging mechanisms and lifetime of PEFC and DMFC." *Journal of power sources* 127.1-2 (2004): 127-134.

92. Koller, Daphne, and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

93. Kong, Dongdong, Yongjie Chen, and Ning Li. "Force-based tool wear estimation for milling process using Gaussian mixture hidden Markov models." *The International Journal of Advanced Manufacturing Technology* 92.5-8 (2017): 2853-2865.

94. Kononenko, Igor. "Machine learning for medical diagnosis: history, state of the art and perspective." *Artificial Intelligence in medicine* 23.1 (2001): 89-109.

95. Kothamasu, Ranganath, Samuel H. Huang, and William H. VerDuin. "System health monitoring and prognostics—a review of current paradigms and practices." *The International Journal of Advanced Manufacturing Technology* 28.9-10 (2006): 1012-1024.

96. Kotzalas, Michael N., and Tedric A. Harris. "Fatigue failure progression in ball bearings." *J. Trib.* 123.2 (2001): 238-242.

97. Kumar, Akhilesh, Ratna Babu Chinnam, and Finn Tseng. "An HMM and polynomial regression-based approach for remaining useful life and health state estimation of cutting tools." *Computers & Industrial Engineering* 128 (2019): 1008-1014.

98. Kumar, Sachin, et al. "A hybrid prognostics methodology for electronic products." *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008.

99. Kumar, Sachin, et al. "A hybrid prognostics methodology for electronic products." *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008.

100. Kuss, Malte, and Carl E. Rasmussen. "Assessing approximations for Gaussian process classification." *Advances in neural information processing systems*. 2006.

101. Kwiatkowska, Marta, Gethin Norman, and David Parker. "Stochastic model checking." *International School on Formal Methods for the Design of Computer, Communication and Software Systems*. Springer, Berlin, Heidelberg, 2007.

102. Lamoureux, Benjamin, Jean-Rémi Massé, and Nazih Mechbal. "Towards an integrated development of PHM systems for aircraft engines: In-design selection and validation of health indicators." *2015 IEEE Conference on Prognostics and Health Management (PHM)*. IEEE, 2015.

103. Latrous, Mohamed, et al. "Integration model for condition-based maintenance and a multivariate control chart: A comparative study univariate vs multivariate." 2018.

104. Lawrence, Neil David. *Variational inference in probabilistic models*. Diss. University of Cambridge, 2000.

105. Le Son, Khanh, et al. "Remaining useful life estimation based on stochastic deterioration models: A comparative study." *Reliability Engineering & System Safety* 112 (2013): 165-175.

106. Le, Thanh Trung, Christophe Bérenguer, and Florent Chatelain. "Multi-branch Hidden semi-Markov modeling for RUL prognosis." *2015 Annual Reliability and Maintainability Symposium (RAMS)*. IEEE, 2015.

107. Le, Thanh Trung, Florent Chatelain, and Christophe Bérenguer. "Multi-branch hidden Markov models for remaining useful life estimation of systems under multiple deterioration modes." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 230.5 (2016): 473-484.

108. Lee, Jay, et al. "Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications." *Mechanical systems and signal processing* 42.1-2 (2014): 314-334.

109. Lee, Jay, M. Ghaffari, and S. Elmeligy. "Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems." *Annual Reviews in Control* 35.1 (2011): 111-122.

110. Lee, Seungchul, et al. "Battery prognostics: SOC and SOH prediction." *International Manufacturing Science and Engineering Conference*. Vol. 54990. American Society of Mechanical Engineers, 2012.

111. Lee, Yang Ji, et al. "An Event-Driven Failure Analysis System for Real-Time Prognosis." *Korean Journal of Computational Design and Engineering* 18.4 (2013): 250-257.

112. Li, Qing, and Steven Y. Liang. "Multiple faults detection for rotating machinery based on bicomponent sparse low-rank matrix separation approach." *IEEE Access* 6 (2018): 20242-20254.

113. Li, Yi, et al. "Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review." *Renewable and Sustainable Energy Reviews* 113 (2019): 109254.

114. Liao, Linxia, and Felix Köttig. "Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction." *IEEE Transactions on Reliability* 63.1 (2014): 191-207.

115. Lim, Chi Keong Reuben, and David Mba. "Switching Kalman filter for failure prognostic." *Mechanical Systems and Signal Processing* 52 (2015): 426-435.

116. Liu, Jianbo, et al. "Similarity based method for manufacturing process performance prediction and diagnosis." *Computers in industry* 58.6 (2007): 558-566.

117. Liu, Liansheng, et al. "Data-driven remaining useful life prediction considering sensor anomaly detection and data recovery." *IEEE access* 7 (2019): 58336-58345.

118. Liu, Yen-Chun, et al. "Data-driven prognostics of remaining useful life for milling machine cutting tools." *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2019.

119. Lu, Yanfei, et al. "Bayesian optimized deep convolutional network for bearing diagnosis." *INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY* (2020).

120. Ly, Canh, et al. "Fault diagnosis and failure prognosis for engineering systems: A global perspective." *2009 IEEE International Conference on Automation Science and Engineering*. IEEE, 2009.

121. Ma, Rui, et al. "Data-driven proton exchange membrane fuel cell degradation predication through deep learning method." *Applied energy* 231 (2018): 102-115.

122. Ma, Ying, Chengbin Chu, and Chunrong Zuo. "A survey of scheduling with deterministic machine availability constraints." *Computers & Industrial Engineering* 58.2 (2010): 199-211.

123. Mechri, Walid, et al. "Fuzzy multiphase Markov chains to handle uncertainties in safety systems performance assessment." *Journal of Loss Prevention in the Process Industries* 26.4 (2013): 594-604.

124. Michalak, Piotr. "Application of artificial neural networks in predicting voter turnout based on the analysis of demographic data." *Polish Cartographical Review* 51.3 (2019): 109-116.

125. Michel Weinechter, Rémi Parouty, André Cabarbaye, François Farago, Emmanuel Remy, Vianney Bordeau, Naoëlle Matahri, Leila Marle, Joffrey Cardon, Ahmed Adjadj, Bruno Colin, Benoît Iung, Jean-Baptiste Leger, Jean-François Barbet, HUMS -Health & Usage Monitoring System -état de l'art et opportunités HUMS -Health & Usage Monitoring System -state of art and opportunities, Congrès Lambda Mu 21, Le Havre, 2018.

126. Michie, Donald, David J. Spiegelhalter, and C. C. Taylor. "Machine learning." *Neural and Statistical Classification* 13.1994 (1994): 1-298.

127. Miklós, István, and Irmtraud M. Meyer. "A linear memory algorithm for Baum-Welch training." *BMC bioinformatics* 6.1 (2005): 231.

128. Minka, Thomas P. "Expectation propagation for approximate Bayesian inference." *arXiv preprint arXiv:1301.2294* (2013).

129. Minka, Thomas Peter. *A family of algorithms for approximate Bayesian inference*. Diss. Massachusetts Institute of Technology, 2001.

130. Mohamed, Daowd, et al. "Battery models parameter estimation based on MATLAB/Simulink®." *Proceedings of the 25th Electric Vehicle Symposium (EVS-25), Shenzhen, China*. Vol. 59. 2010.

131. Mosallam, Ahmed, Kamal Medjaher, and Noureddine Zerhouni. "Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction." *Journal of Intelligent Manufacturing* 27.5 (2016): 1037-1048.

132. Murray, L. C., and R. J. Farber. "Time series analysis of an historical visibility data base." *Atmospheric Environment (1967)* 16.10 (1982): 2299-2308.

133. Namburu, Setu Madhavi, et al. "Data-driven modeling, fault diagnosis and optimal sensor selection for HVAC chillers." *IEEE transactions on automation science and engineering* 4.3 (2007): 469-473.

134. Neapolitan, Richard E. *Probabilistic reasoning in expert systems: theory and algorithms*. John Wiley & Sons, Inc., 1989.

135. Nelson, D. V. "Effects of residual stress on fatigue crack propagation." *Residual stress effects in fatigue*. ASTM International, 1982.

136. Nelwamondo, Fulufhelo V., Tshilidzi Marwala, and Unathi Mahola. "Early classifications of bearing faults using hidden Markov models, Gaussian mixture models, mel-frequency cepstral coefficients and fractals." *International Journal of Innovative Computing, Information and Control* 2.6 (2006): 1281-1299.

137. Nguyen, Ho Si Hung, et al. "Joint maintenance scheduling and routing optimization for geographically dis-persed production systems." *in Industrial Maintenance and Reliability Manchester, UK 12-15 June 2018* (2018): 109.

138. North, M., E. Hernandez, and R. Garcia. "Frequency analysis of high CO concentrations in Madrid by stochastic process modelling." *Atmospheric Environment (1967)* 18.10 (1984): 2049-2054.

139. Orsagh, Rolf F., Jeremy Sheldon, and Christopher J. Klenke. *Prognostics/diagnostics for gas turbine engine bearings*. Vol. 36843. 2003.

140. Oumouni, Mestapha, Franck Schoefs, and Bruno Castanier. "Modeling time and spatial variability of degradation through gamma processes for structural reliability assessment." *Structural Safety* 76 (2019): 162-173.

141. Paris, Pe, and Fazil Erdogan. "A critical analysis of crack propagation laws." (1963): 528-533.

142. Patrick, Romano, et al. "Diagnostic enhancements for air vehicle HUMS to increase prognostic system effectiveness." *2009 IEEE Aerospace conference*. IEEE, 2009.

143. Patrick, Romano, et al. "Diagnostic enhancements for air vehicle HUMS to increase prognostic system effectiveness." *2009 IEEE Aerospace conference*. IEEE, 2009.

144. Pearl, J. "1988, Probabilistic Reasoning in Intelligent Systems."

145. Pearl, J. "Probabilistic reasoning in intelligent systems: networks of plausible inference—Judea Pearl." (1988).

146. Pearl, Judea. "Bayesian netwcrks: A model cf self-activated memory for evidential reasoning." *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA, USA*. 1985.

147. Peng, Ying, and Ming Dong. "A hybrid approach of HMM and grey model for age-dependent health prediction of engineering assets." *Expert Systems with Applications* 38.10 (2011): 12946-12953.

148. Petrillo, Alberto, et al. "Model-based vehicular prognostics framework using Big Data architecture." *Computers in Industry* 115 (2020): 103177.

149. Petrović, Predrag B., and Dimitrije Rozgić. "Computational effective modified Newton–Raphson algorithm for power harmonics parameters estimation." *IET Signal Processing* 12.5 (2018): 590-598.

150. Piyush, Tagade, et al. "Recursive Bayesian filtering framework for lithium-ion cell state estimation." *Journal of Power Sources* 306 (2016): 274-288.

151. Poongodai, A., and S. Bhuvaneswari. "AI technique in diagnostics and prognostics." *International Journal of Computer Applications* 975 (2013): 8887.

152. Pugno, Nicola, et al. "A generalized Paris' law for fatigue crack growth." *Journal of the Mechanics and Physics of Solids* 54.7 (2006): 1333-1349.

153. Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE* 77.2 (1989): 257-286.

154. Raimondi, P. M., et al. "Short-time fuzzy DAP predictor for air pollution due to vehicular traffic." *WIT Transactions on Ecology and the Environment* 19 (1997).

155. Rehman, Waheed UR, et al. "Fuzzy logic–based intelligent control for hydrostatic journal bearing." *Measurement and Control* 52.3-4 (2019): 229-243.

156. Richardson, Robert R., Michael A. Osborne, and David A. Howey. "Gaussian process regression for forecasting battery state of health." *Journal of Power Sources* 357 (2017): 209-219.

157. Roemer, Michael J., E. O. Nwadiogbu, and G. Bloor. "Development of diagnostic and prognostic technologies for aerospace health management applications." *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*. Vol. 6. IEEE, 2001.

158. Room, Chat. "Hidden Markov Model.", updated in 2019

159. Rosenstock, Tizian, et al. "Specific DTI seeding and diffusivity-analysis improve the quality and prognostic value of TMS-based deterministic DTI of the pyramidal tract." *Neuroimage: Clinical* 16 (2017): 276-285.

160. Rotea, Mario A. "Analysis of multivariable extremum seeking algorithms." *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*. Vol. 1. No. 6. IEEE, 2000.

161. Sadeghi, Farshid, et al. "A review of rolling contact fatigue." *Journal of tribology* 131.4 (2009).

162. Salem, Abdeljabbar Ben, Alexandre Muller, and Philippe Weber. "Dynamic Bayesian Networks in system reliability analysis." *Fault Detection, Supervision and Safety of Technical Processes 2006*. Elsevier Science Ltd, 2007. 444-449.

163. Sallak, Mohamed, Christophe Simon, and Jean-Francois Aubry. "A fuzzy probabilistic approach for determining safety integrity level." *IEEE Transactions on Fuzzy Systems* 16.1 (2008): 239-248.

164. Sandborn, Peter A., and Chris Wilkinson. "A maintenance planning and business case development model for the application of prognostics and health management (PHM) to electronic systems." *Microelectronics Reliability* 47.12 (2007): 1889-1901.

165. Satish, B., and N. D. R. Sarma. "A fuzzy BP approach for diagnosis and prognosis of bearing faults in induction motors." *IEEE Power Engineering Society General Meeting, 2005*. IEEE, 2005.

166. Saxena, Abhinav, et al. "Metrics for offline evaluation of prognostic performance." *International Journal of Prognostics and health management* 1.1 (2010): 4-23.

167. Saxena, Abhinav, et al. "Requirements specification for prognostics performance-an overview." *Aiaa infotech@ aerospace 2010*. 2010. 3398.

168. Saxena, Saurabh, et al. "Accelerated degradation model for C-rate loading of lithium-ion batteries." *International journal of electrical power & energy systems* 107 (2019): 438-445.

169. Schmittinger, Wolfgang, and Ardalan Vahidi. "A review of the main parameters influencing long-term performance and durability of PEM fuel cells." *Journal of power sources* 180.1 (2008): 1-14.

170. Schuh, Melanie, and Jan Lunze. "Fault-tolerant control of deterministic I/O automata with ambiguous diagnostic result." *2016 13th International Workshop on Discrete Event Systems (WODES)*. IEEE, 2016.

171. Schwabacher, Mark, and Kai Goebel. "A Survey of Artificial Intelligence for Prognostics." *AAAI Fall Symposium: Artificial Intelligence for Prognostics*. 2007.

172. Schwunk, Simon, et al. "Particle filter for state of charge and state of health estimation for lithium–iron phosphate batteries." *Journal of Power Sources* 239 (2013): 705-710.

173. Seeger, Matthias, and Guillaume Bouchard. "Fast variational Bayesian inference for non-conjugate matrix factorization models." *Artificial Intelligence and Statistics*. 2012.

174. Severson, Kristen A., et al. "Data-driven prediction of battery cycle life before capacity degradation." *Nature Energy* 4.5 (2019): 383-391.

175. Shao, Junfeng, Tianrui Chen, and Cong Wang. "Fast Fault Diagnosis for Power Systems Using Deterministic Learning." *2019 Chinese Control Conference (CCC)*. IEEE, 2019.

176. Sharma, Amandeep, et al. "Bearing fault diagnosis using frequency domain features and artificial neural networks." *Information and Communication Technology for Intelligent Systems*. Springer, Singapore, 2019. 539-547.

177. Shen, Fei, et al. "Transfer between multiple working conditions: A new TCCHC-based exponential semi-deterministic extended Kalman filter for bearing remaining useful life prediction." *Measurement* 142 (2019): 148-162.

178. Si, Xiao-Sheng, et al. "Remaining useful life estimation–a review on the statistical data driven approaches." *European journal of operational research* 213.1 (2011): 1-14.

179. Sierra, Gina, and Kai Goebel. "A Novel Model-Based Framework for Power Consumption Prognosis in Multirotors Applied to the Battery End of Discharge Prognostics." *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems* 3.2 (2020).

180. Skliros, Christos, et al. "A review of model based, and data driven methods targeting hardware systems diagnostics." (2018).

181. Skormin, Victor A., Joseph Apone, and John J. Dunphy. "On-line diagnostics of a self-contained flight actuator." *IEEE Transactions on aerospace and electronic systems* 30.1 (1994): 186-196.

182. Slottner, Pontus, and Mathias Wa¨ rja. "Knowledge based prognostics models for gas turbine core components." *Turbo Expo: Power for Land, Sea, and Air*. Vol. 43116. 2008.

183. Sreenuch, Tarapong, Faisal Khan, and J. Li. "Particle filter with operational-scalable Takagi–Sugeno fuzzy degradation model for filter-clogging prognosis." *Journal of Aerospace Information Systems* 12.5 (2015): 398-412.

184. Sun, Bo, et al. "Benefits analysis of prognostics in systems." *2010 Prognostics and System Health Management Conference*. IEEE, 2010.

185. Sun, Congcong, et al. "Combining model predictive control with constraint-satisfaction formulation for the operative pumping control in water networks." *Procedia Engineering* 119 (2015): 963-972.

186. Sun, Fengchun, et al. "Adaptive unscented Kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles." *Energy* 36.5 (2011): 3531-3540.

187. Sun, Shiliang, and Xin Xu. "Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction." *IEEE Transactions on Intelligent Transportation Systems* 12.2 (2010): 466-475.

188. Sun, Shiliang. "A review of deterministic approximate inference techniques for Bayesian machine learning." *Neural Computing and Applications* 23.7-8 (2013): 2039-2050.

189. Swanson, David C. "A general prognostic tracking algorithm for predictive maintenance." *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*. Vol. 6. IEEE, 2001.

190. Takata, Shozo, et al. "Maintenance: changing role in life cycle management." *CIRP annals* 53.2 (2004): 643-655.

191. Teng, Wei, et al. "A Robust Model-Based Approach for Bearing Remaining Useful Life Prognosis in Wind Turbines." *IEEE Access* 8 (2020): 47133-47143.

192. Thomas, Édouard, et al. "Opportune maintenance and predictive maintenance decision support." *IFAC proceedings volumes* 42.4 (2009): 1603-1608.

193. Thorburn, William M. "The myth of Occam's razor." *Mind* 27.107 (1918): 345-353.

194. Tian, Zhigang, Daming Lin, and Bairong Wu. "Condition based maintenance optimization considering multiple objectives." *Journal of Intelligent Manufacturing* 23.2 (2012): 333-340.

195. Tobon-Mejia, Diego A., et al. "Hidden Markov models for failure diagnostic and prognostic." *2011 Prognostics and System Health Managment Confernece*. IEEE, 2011.

196. Tobon-Mejia, Diego Alejandro, et al. "A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models." *IEEE Transactions on reliability* 61.2 (2012): 491-503.

197. Turin, William, and Rittwik Jana. "Continuous state HMM modeling of flat fading channels." *2005 IEEE 61st Vehicular Technology Conference*. Vol. 1. IEEE, 2005.

198. Turin, William. *Performance Analysis and modeling of digital transmission systems*. Springer Science & Business Media, 2012.

199. Uckun, Serdar, Kai Goebel, and Peter JF Lucas. "Standardizing research methods for prognostics." *2008 International Conference on Prognostics and Health Management*. IEEE, 2008.

200. Uusitalo, Laura. "Advantages and challenges of Bayesian networks in environmental modelling." *Ecological modelling* 203.3-4 (2007): 312-318.

201. Vasile-Dragomir, Otilia Elena. *Contribution au pronostic de défaillances par réseau neuro-flou: maîtrise de l'erreur de prédiction*. Diss. 2008.

202. Vogl, Gregory W., Brian A. Weiss, and Moneer Helu. "A review of diagnostic and prognostic capabilities and best practices for manufacturing." *Journal of Intelligent Manufacturing* 30.1 (2019): 79-95.

203. Wainwright, Martin J., and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

204. Wang, Cunsong, et al. "A data-driven aero-engine degradation prognostic strategy." *IEEE Transactions on Cybernetics* (2019).

205. Wang, Dian, W. H. Tang, and Q. H. Wu. "Ontology-based fault diagnosis for power transformers." *IEEE PES General Meeting*. IEEE, 2010.

206. Wang, Jiujian, et al. "Remaining useful life estimation in prognostics using deep bidirectional lstm neural network." *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. IEEE, 2018.

207. Wang, Shaoyan, Tianrui Chen, and Cong Wang. "Fault Diagnosis for a Class of Sampled-data Systems via Deterministic Learning." *2019 Chinese Control and Decision Conference (CCDC)*. IEEE, 2019.

208. Wang, Yu, and Jack M. Winters. "An event-driven dynamic recurrent neuro-fuzzy system for adaptive prognosis in rehabilitation." *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*. Vol. 2. IEEE, 2003.

209. Wang, Yujie, et al. "A fractional-order model-based state estimation approach for lithium-ion battery and ultra-capacitor hybrid power source system considering load trajectory." *Journal of Power Sources* 449 (2020): 227543.

210. Weber, Philippe, and Christophe Simon. *Benefits of Bayesian network models*. John Wiley & Sons, 2016.

211. Weber, Philippe, Paul Munteanu, and Lionel Jouffe. "Dynamic Bayesian Networks modelling the dependability of systems with degradations and exogenous constraints." *IFAC Proceedings Volumes* 37.4 (2004): 207-212.

212. Wen-ping, L. E. I., and Z. H. E. N. G. Peng. "Forecast of the trend of vibration of rotating machinery by using ARIMA model." *Journal of Zhengzhou Institute of Light Industry* 2 (2004): 12.

213. Wu, Bin, et al. "Application of artificial neural networks in design of lithium-ion batteries." *Journal of Power Sources* 395 (2018): 128-136.

214. Wu, Jinfeng, et al. "A review of PEM fuel cell durability: Degradation mechanisms and mitigation strategies." *Journal of Power Sources* 184.1 (2008): 104-119.

215. Wu, Jun, et al. "Degradation data-driven time-to-failure prognostics approach for rolling element bearings in electrical machines." *IEEE Transactions on Industrial Electronics* 66.1 (2018): 529-539.

216. Wu, Lifeng, Xiaohui Fu, and Yong Guan. "Review of the remaining useful life prognostics of vehicle lithium-ion batteries using data-driven methodologies." *Applied Sciences* 6.6 (2016): 166.

217. Wu, Wei, Jingtao Hu, and Jilong Zhang. "Prognostics of machine health condition using an improved ARIMA-based prediction method." *2007 2nd IEEE Conference on Industrial Electronics and Applications*. Ieee, 2007.

218. Xiang, Yisha, C. Richard Cassady, and Edward A. Pohl. "Optimal maintenance policies for systems subject to a Markovian operating environment." *Computers & Industrial Engineering* 62.1 (2012): 190-197.

219. Xu, Dong, et al. "Residual fatigue life prediction of ball bearings based on Paris law and RMS." *Chinese Journal of Mechanical Engineering* 25.2 (2012): 320-327.

220. Yang, Duo, et al. "A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve." *Journal of Power Sources* 384 (2018): 387-395.

221. Yeh, Ruey Huei. "Optimal inspection and replacement policies for multi-state deteriorating systems." *European Journal of Operational Research* 96.2 (1997): 248-259.

222. Yousfi-Steiner, N., et al. "A review on polymer electrolyte membrane fuel cell catalyst degradation and starvation issues: Causes, consequences and diagnostic for mitigation." *Journal of Power Sources* 194.1 (2009): 130-145.

223. Yousfi-Steiner, Nadia, et al. "A review on PEM voltage degradation associated with water management: Impacts, influent factors and characterization." *Journal of power sources* 183.1 (2008): 260-274.

224. Yu, Jianbo. "Bearing performance degradation assessment using locality preserving projections and Gaussian mixture models." *Mechanical Systems and Signal Processing* 25.7 (2011): 2573-2588.

225. Yuan, Christopher. "Hidden Markov Models." (2018).

226. Zafar, Muhammad Rehman, and Naimul Mefraz Khan. "DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems." *arXiv preprint arXiv:1906.10263* (2019).

227. Zhang, Shuzhi, et al. "Synchronous estimation of state of health and remaining useful lifetime for lithium-ion battery using the incremental capacity and artificial neural networks." *Journal of Energy Storage* 26 (2019): 100951.

228. Zhang, Weiting, Dong Yang, and Hongchao Wang. "Data-driven methods for predictive maintenance of industrial equipment: A survey." *IEEE Systems Journal* 13.3 (2019): 2213-2227.

229. Zheng, Fangdan, et al. "Prognostics of lithium-ion batteries using a deterministic Bayesian approach." *2015 IEEE Conference on Prognostics and Health Management (PHM)*. IEEE, 2015.

230. Zhou, Jie, Xinyuan Song, and Liuquan Sun. "Continuous time hidden Markov model for longitudinal data." *Journal of Multivariate Analysis* (2020): 104646.

231. Zhou, Quan, et al. "A novel method based on nonlinear auto-regression neural network and convolutional neural network for imbalanced fault diagnosis of rotating machinery." *Measurement* (2020): 107880.

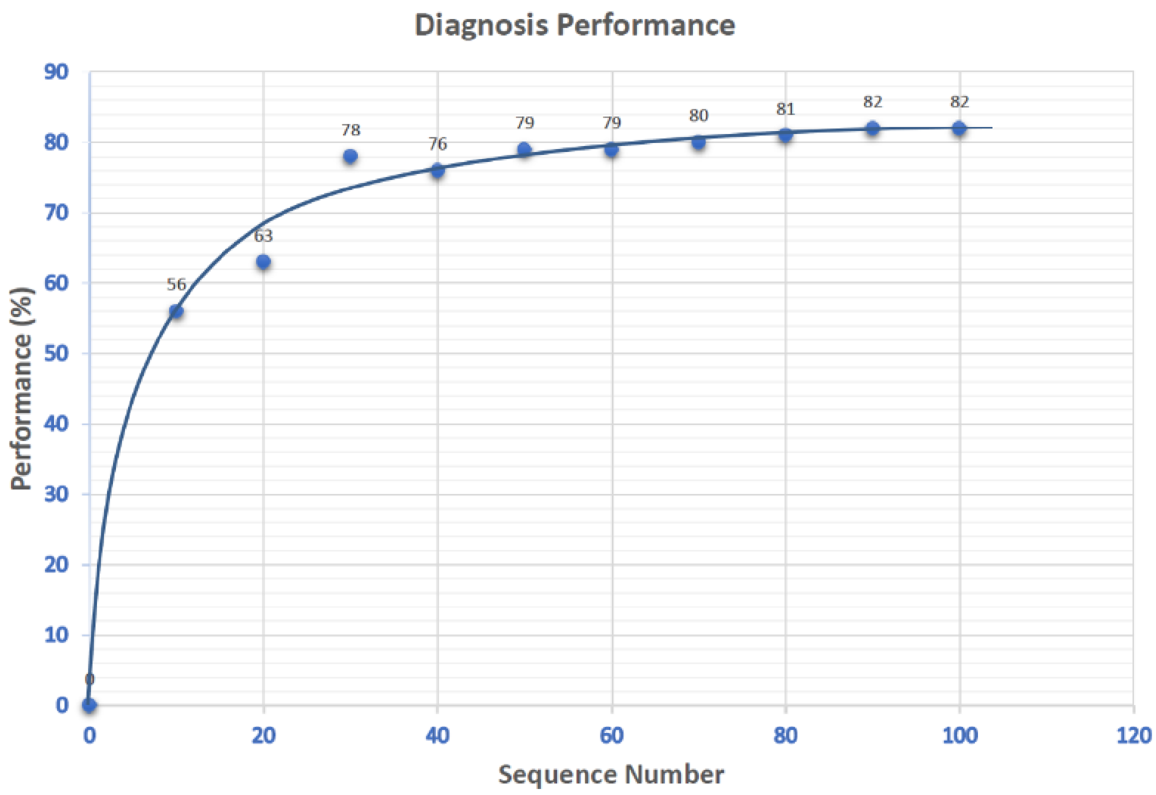
232. Zhou, Zhi-Jie, et al. "A model for real-time failure prognosis based on hidden Markov model and belief rule base." *European Journal of Operational Research* 207.1 (2010): 269-283.

233. Zhu, Jiang, and Shiliang Sun. "Single-task and multitask sparse Gaussian processes." *2013 International Conference on Machine Learning and Cybernetics*. Vol. 3. IEEE, 2013.

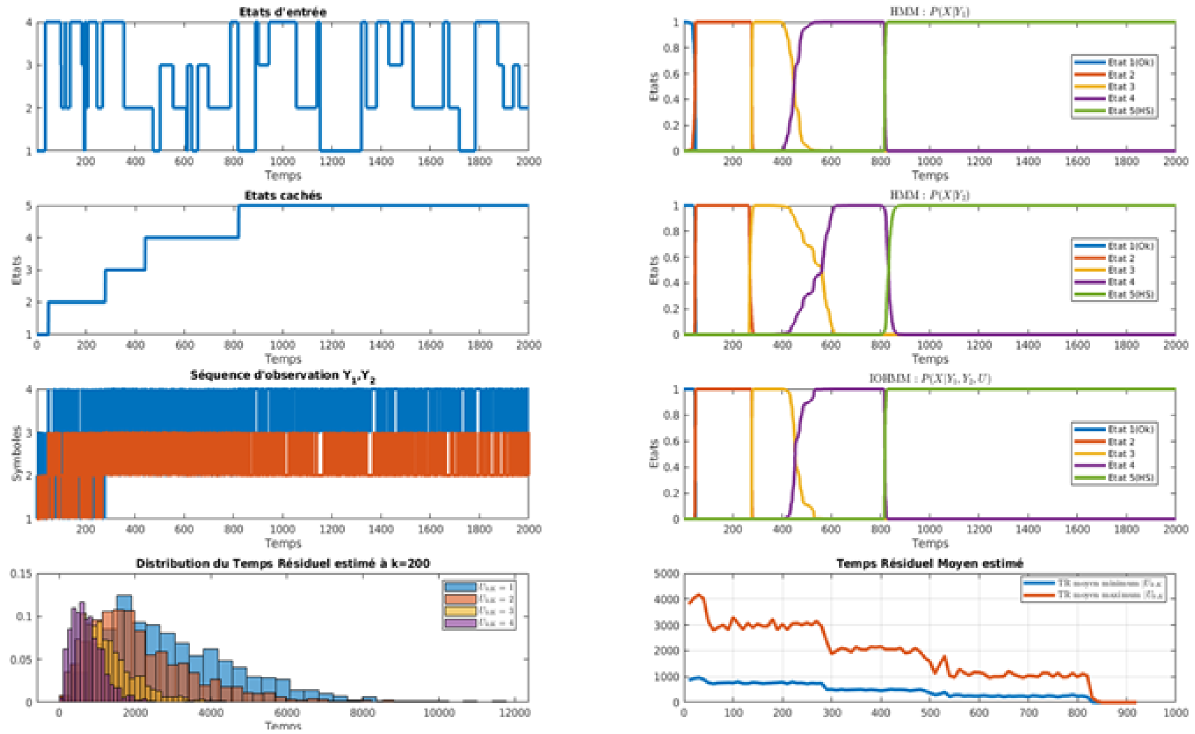
234. Zhu, Jun, Nan Chen, and Changqing Shen. "A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions." *Mechanical Systems and Signal* Processing 139 (2020): 106602.

Appendixes

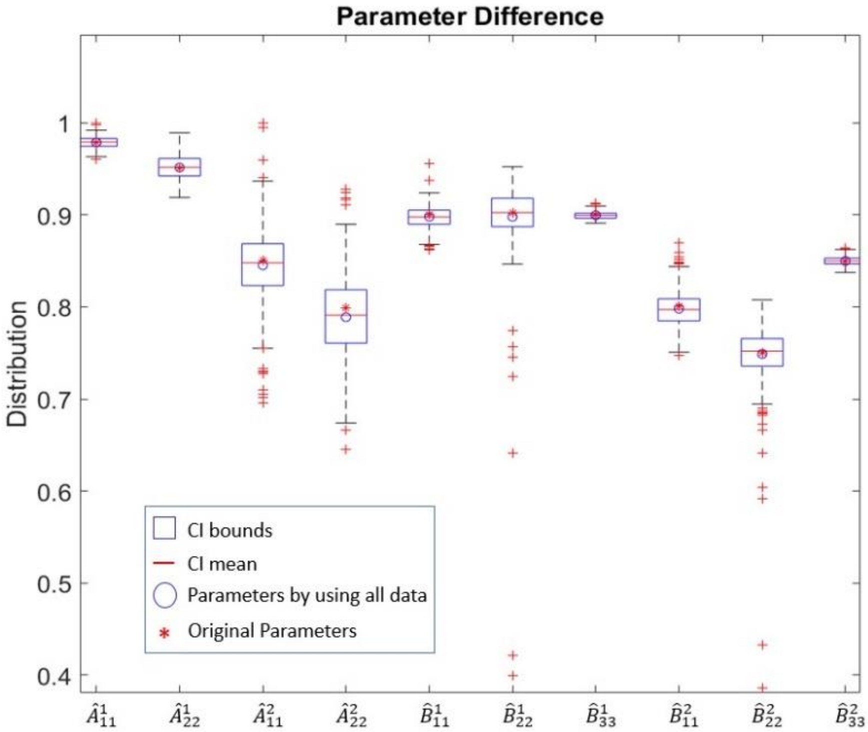
Diagnosis performance for IOHMM: [Shahin 2019a]



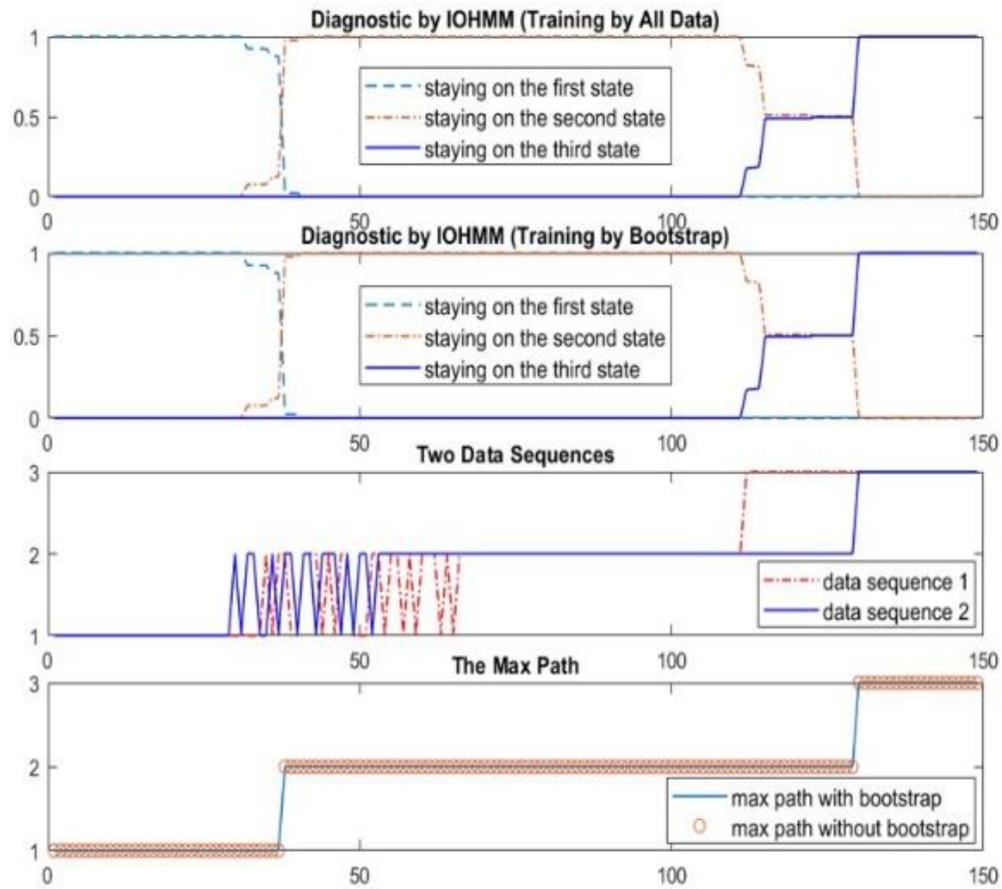
Diagnostic and prognostic by monte Carlo simulation: [Shahin 2020a]



Learning Parameters: [Shahin 2020b]



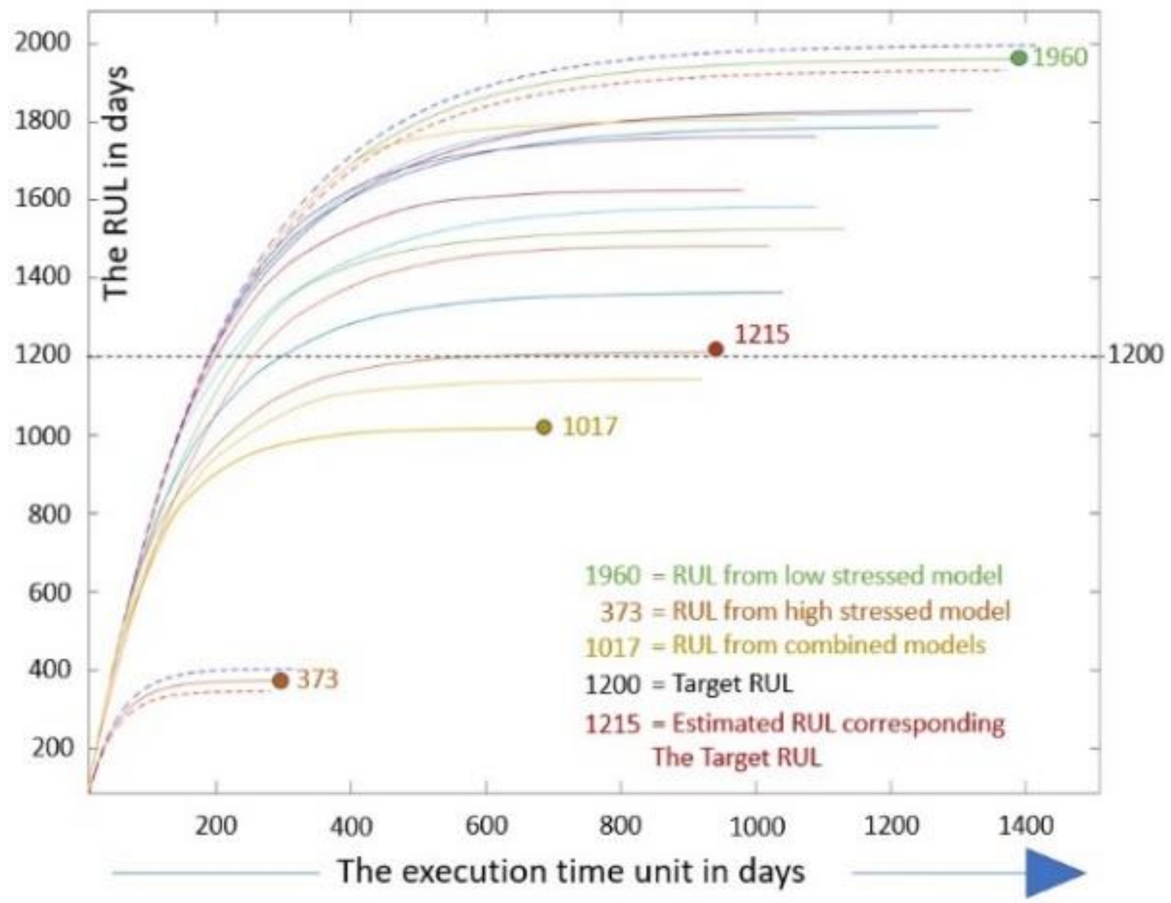
Diagnostic with bootstrap: [Shahin 2020b]



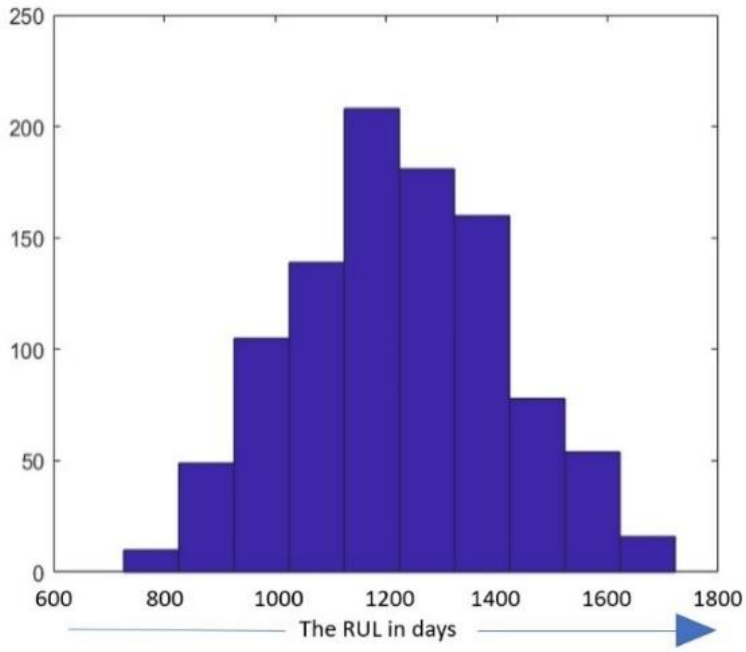
Estimated RUL Using Different Models: [Shahin 2020c]

No	Model Name	Estimated mean RUL
1	Matrix $A1$	1960 days
2	CI lower limit ($A1$)	1931 days
3	CI upper limit ($A1$)	1995 days
4	Matrix $A2$	373 days
5	CI lower limit ($A2$)	346 days
6	CI upper limit ($A2$)	403 days
7	Combined ($A1$ & $A2$)	1017 days

Different RULs using different operating conditions: monte Carlo simulation: [Shahin 2020c]



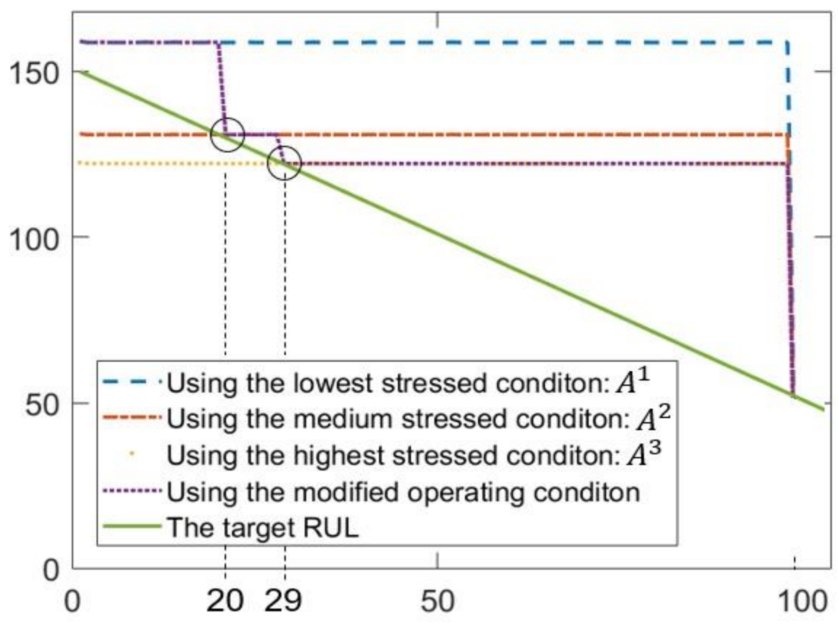
Distribution over the estimated RUL: [Shahin 2020c]



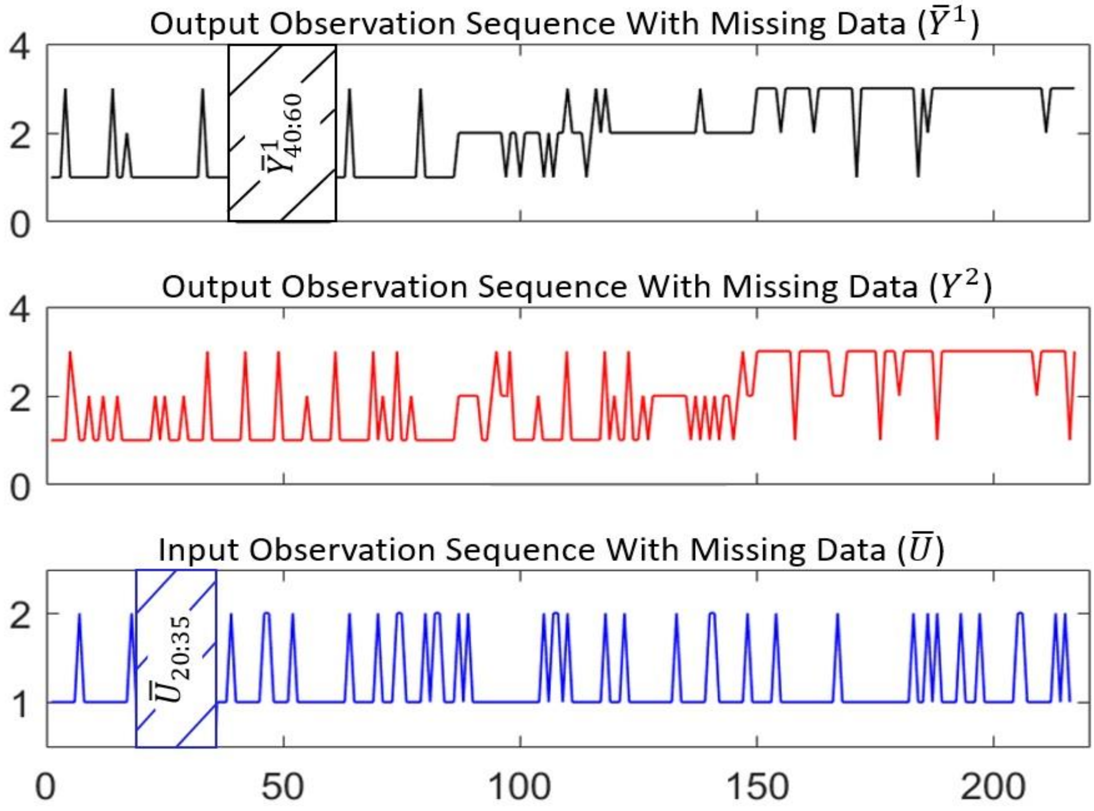
Different RULs at time *k* [Shahin 2020e]

No	Model Name	Estimated RUL
1	$\hat{A}^1$	159 days
2	$\hat{A}^2$	131 days
3	$\hat{A}^3$	122 days
4	Previous Conditions	147 days

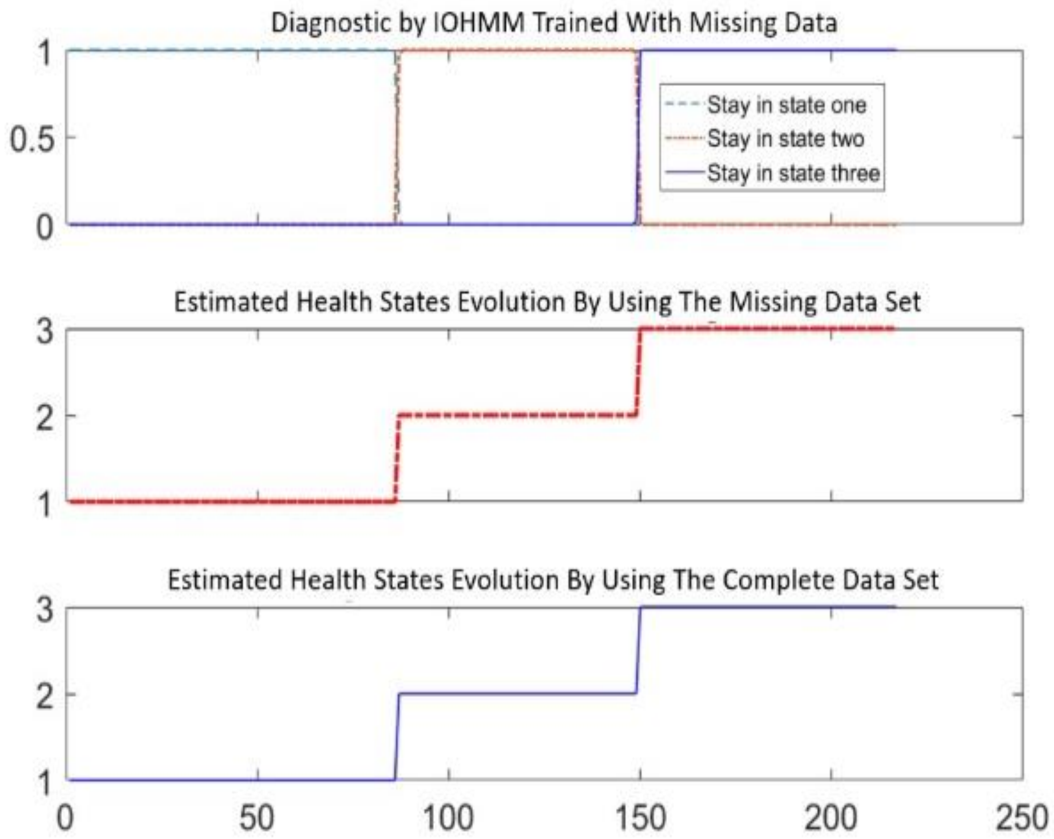
Online RUL matching with the target RUL [Shahin 2020e]



The sequence with missing data to diagnostic system health [Shahin 2020f]



Diagnostic under missing data: [Shahin 2020f]



Modèle graphique probabiliste appliqué au diagnostic de l'état de santé des systèmes, au pronostic et à l'estimation de la durée de vie résiduelle

Mots-clés : Évaluation de la santé, Diagnostic, Pronostic, RUL, Gestion des RUL, PHM, Conception de la dégradation, Condition de fonctionnement, Système complexe, Système structuré, Évaluation en ligne, Modèle de Markov caché entrée-sortie, Apprentissage des paramètres, Données manquantes

Résumé de thèse

Cette thèse contribue au développement des recherches dans le domaine du Pronostic et Health Management : gestion de l'état de santé des systèmes complexes. Dans un contexte de management opérationnel et de sûreté de fonctionnement des systèmes, nous proposons d'étudier comment la modélisation par un Modèle Graphique Probabiliste Dynamique (MGPD) permet le diagnostic de l'état de santé courant d'un système, le pronostic de cet état et de l'évolution des dégradations, ainsi que l'estimation de sa durée de vie résiduelle en fonction de ses conditions opérationnelles.

La dégradation des composants est en général inconnue et nécessite un arrêt du système pour être observée. Cependant, cela est difficile, voire impossible, durant l'exploitation du système. Néanmoins, un ensemble de grandeurs observables sur le système ou le composant peut caractériser le niveau de dégradation et faciliter l'estimation de la durée de vie résiduelle du composant et du système.

Les MGPD offrent une approche adaptée à la modélisation de l'évolution de l'état de santé des systèmes et des composants. Récemment, l'utilisation de HMM (Hidden Markov Model) ou de HSMM (Hidden Semi-Markov Model) pour modéliser un processus non observable de dégradation et le relier à des observations de leurs conséquences a déjà été exploitée avec des résultats intéressants. Toutefois, la non-prise en compte des conditions opérationnelles, influant sur les processus de dégradation, limite la performance de ces outils. Les algorithmes d'apprentissage et d'inférence rendent exploitables ces modèles complexes pour une exploitation dans une problématique de pronostic.

Il s'agit dans cette thèse de transposer et de capitaliser l'expérience de ces travaux antérieurs dans un contexte de pronostic sur la base d'un MGPD plus efficace compte tenu des connaissances disponibles sur le système. Nous étendons la modélisation classique des modèles de la famille des HMM vers les IOHMM pour permettre une propagation temporelle de l'incertitude afin de résoudre le problème de pronostic de l'état de santé et de l'estimation de la durée de vie résiduelle. Cette recherche comprend l'extension des algorithmes d'apprentissage et d'inférence appliqués aussi bien dans le cas d'un composant que pour un système structuré. Les variantes du modèle HMM sont proposées pour intégrer le contexte opérationnel dans le pronostic.

Cette thèse a pour but de contribuer à lever les verrous scientifiques suivants :

- Considérer l'état de santé quelle que soit la complexité du système par un modèle stochastique et apprendre les paramètres du modèle à partir des mesures disponibles sur le système.
- Établir un diagnostic de l'état de santé du système et le pronostic de son évolution en intégrant plusieurs conditions opérationnelles.
- Estimer la durée de vie résiduelle des composants et des systèmes structurés (série, parallèle) à partir de ses composants.

L'enjeu est majeur, car le pronostic de la dégradation des composants du système permet de définir des stratégies soit de pilotage soit de maintenance par rapport à la durée de vie résiduelle du système. Cela permet la réduction de la probabilité d'occurrence d'un arrêt pour cause de dysfonctionnement du système, soit en ajustant la vitesse de dégradation pour s'accorder à un plan de maintenance préventif, soit en planifiant les interventions de maintenance de manière proactive.

Dynamic Probabilistic Graphical Model applied to the system health diagnosis, prognosis, and the remains useful life estimation

Keywords: Health assessment, Diagnostic, Prognostic, RUL, RUL Management, PHM, Degradation design, Operating Condition, Complex system, Structured System, Online assessment, Input Output Hidden Markov Model, Parameter learning, Missing data

Thesis abstract

This thesis contributes to prognosis and health management for assessing health condition of complex systems. In the context of operational management and operational safety of systems, we propose to investigate how Dynamic Probabilistic Graphical Modelling (DPGM) can be used to diagnose the current health state of systems, prognostic the future health state, and the evolution of degradation, as well as estimate its remaining useful life based on its operating conditions.

System degradation is generally unknown and requires shutting down the system to be observed. However, this is difficult or even impossible during system operation. Though, a set of observable quantities on a system or component can characterize the level of degradation and help to estimate the remaining useful life of components and systems.

The DPGM provides an approach suitable for modelling the evolution of the health state of systems and components. Recently, interesting results have been obtained by using HMM (Hidden Markov Model) or HSMM (Hidden Semi-Markov Model) to model unobservable degradation processes and to relate them to observations of their consequences. However, the performance of these models is limited because they are not able to consider the operational conditions that affect degradation processes. Learning and inference algorithms allow these complex models to be used for prediction problems.

The aim of this thesis is to transpose and capitalize on the experience of these previous works in a prognostic context on the basis of a more efficient DPGM taking into account the available knowledge on the system. We extend the classical HMM family models to the IOHMM to allow the time propagation of uncertainty to address prognostic problems. This research includes the extension of learning and inference algorithms. Variants of the HMM model are proposed to incorporate the operating environment into the prognosis.

The aim of this thesis is to contribute to solving the following scientific locks:

- Considering the state of health whatever the complexity of the system by a stochastic model and learning the model parameters from the available measurements on the system.

- Establish a diagnosis of the state of health of the system and the prognosis of its evolution by integrating several operational conditions.

- Estimate the remaining useful life of components and structured systems with series and parallel components.

This is a major challenge because the prognosis of the degradation of system components makes it possible to define strategies for either control or maintenance in relation to the residual life of the system. This allows the reduction of the probability of occurrence of a shutdown due to a system malfunction either by adjusting the degradation speed to fit in with a preventive maintenance plan or by proactively planning maintenance interventions.