

Adaptive Finite Horizon Degradation-Aware Regulator

Amirhossein Hosseinzadeh Dadash and Niclas Björzell

Abstract Predicting the failure and estimating the machine’s state of health is information that supports the production planning and maintenance management systems to increase productivity and reduce maintenance and downtime costs. However, controlling the degradation in the machines will improve the system’s reliability and resilience and make high-level decisions more accurate and reliable. To control the degradation in the machines, time should be included in the cost function as a variable, which alters the markovian properties of the system dynamic. In this article, we include the degradation cost in the quadratic cost function of the infinite horizon controller and calculate the optimal feedback according to the dynamics of the degradation using dynamic programming. It will be shown that the infinite horizon control will convert to the finite horizon, and the controller will be able to control the degradation according to the desired degradation at the desired time. In the end, with the help of simulation, we show that the degradation controller can control the degradation in the MIMO systems.

1 Introduction

In the era of Industry 4.0, the horizons of the definition of ”optimal control” can include more variables than its classical definition. Keeping the system output close to the desired output, which was the optimal control’s primary goal, is now among many other goals that must be achieved simultaneously. For instance, the Linear Quadratic Regulator (LQR) is considered an optimal controller when the only parameters to

Amirhossein Hosseinzadeh Dadash

Department of Electronics, Mathematics and Natural Sciences, University of Gävle, 801 76 Gävle, Sweden e-mail: amirhossein.hosseinzadeh.dadash@hig.se

Niclas Björzell

Department of Electronics, Mathematics and Natural Sciences, University of Gävle, 801 76 Gävle, Sweden e-mail: niclas.bjorsell@hig.se

control are the input and output, but being optimal in controlling the degradation simultaneously needs some new information from the system and modifications to the method [1, 2].

Fault-tolerant controllers have been the subject of research [3]. However, degradation tolerant controllers, which are more helpful in long-term production planning and maintenance management systems [4, 5], have not been discussed much. For instance, one of the methods for reducing the maintenance cost is to synchronize the machines' degradation so they can reach the maintenance time simultaneously. This simultaneity is achieved in [6] using advanced high-level controllers and has the limitation that it is only applicable to systems with similar machines working on the same process.

To reduce the calculation and data storage cost and make the degradation control applicable to machines that are not identical and do not work on the same process, the degradation controller should be integrated with the controller or work on its side without a need to exchange much information with high-level controller (production management) [7, 8]. For doing this integration, the degradation model should be identified. This identification should be in a way that the model can be related to the system dynamics. This identification can be made using process-aware neural networks [9, 10] or sparse regressions [11, 12]. In both cases, the identification of the degradation might not be as accurate as model identification without the limitation of considering the process parameters, but it can be used for control purposes which have their own benefits.

Assuming that the machine and degradation models are known, it is possible to control the degradation in two ways: including the degradation as the system's state or including it in the feedback loop. The traditional way of including degradation inside the model has been tested successfully [13, 14], but having a closed form of including the degradation in the feedback loop will be helpful in the future and can change the controller design process. Also, an adaptive feedback loop can be configured faster with the degradation model change than the adaptive controller.

This paper will introduce the adaptive feedback loop for controlling the degradation at the same time as the output. In the first section, the method for calculating different parts of this feedback loop will be shown, and in the second section, the method will be validated using a simulation model.

2 Method

The method for calculating the adaptive degradation-aware feedback comprises two parts, the formulation of the feedback, which is explained first, and mapping from the recorded degradation into the system's states.

2.1 Formulation

The state-space model of the system can be written as follows:

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + Mv_1(t) \\ y(t) = Cx(t) + v_2(t) \end{cases} \quad \Omega = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (1)$$

where $x(t)$ includes system state(s) at time t , $u(t)$ is system input(s) at time t . A and B represent the physical system parameters, and C defines the relationship between the output(s) and state(s) of the system. v_1 and v_2 are the input and sensor noise, respectively.

The quadratic cost that the LQ controller minimizes for the infinite horizon is [15]

$$J = \|e\|_{Q_1}^2 + \|u\|_{Q_2}^2, \quad (2)$$

where $e = x - r$ and r is the reference signal, and e is the error; and Q_1 and Q_2 are penalty matrices for the error and input signal, respectively. However, adding the third term for controlling the degradation converts (2) to

$$J(t = 1 : T) = E \left\{ \sum_{k=0}^N \|e_k\|_{Q_1}^2 + \sum_{k=0}^{N-1} \|u_k\|_{Q_2}^2 + \sum_{k=0}^N \|G_{x^f, u^f}(t)\|_{Q_3}^2 \right\}, \quad (3)$$

$$G_{x^f, u^f}(t) = D_{x^f, u^f} - D_d(t), \quad (4)$$

where x^f, u^f are feature of x and feature of u respectively, D_{x^f, u^f} is the degradation at state x^f with the input u^f , and $D_d(t)$ is the desired degradation at time t . To better understand x^f and u^f , some explanation is needed. First, the working cycle should be defined as the interval where the machine starts from its standby state and settles on the desired output (usually the step response of the machine till it fully settles). For example, a working cycle of a wood-cutting saw starts when it starts rotating from the stop position until it reaches the desired speed (full load working). Second, in reality, the degradation rate is lower than the rate of change in the system state (the degradation happens over tens or hundreds, or even thousands of cycles of operation while the controller works on the scale of a fraction of a second). So, in order to be able to detect the degradation from the recorded signal, a fixed point (in time) or a feature from one machine's working cycle (for each state and input) should be recorded for different cycles. This way, the degradation trend can be detected and mapped to the system state. These recorded features are defined to be x^f and u^f . Fig.1 shows this process more clearly. Finally, as the features are the variables that are going to be used in the last part of (3), t will be defined as a cycle; this means

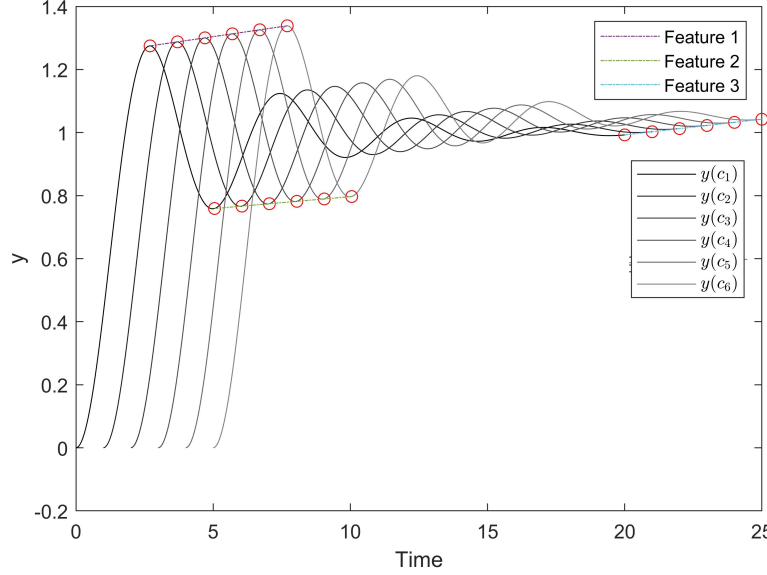


Fig. 1 Relation of the features per cycle and recorded signal (red circles are the features e.g., x^f or u^f that are recorded at each cycle, and their trend is used to map the degradation to the system states)

that t is not an instance and instead is a time duration. However, as only one feature from each recorded signal during each cycle is recorded in (3), it can be considered a time instance.

After these adaptations of the cost function with the limitation of the function estimation methods, the whole optimization becomes a finite horizon optimization that tries to control the degradation of the machine. Hence, it reaches the maintenance condition (x_N) at the desired time (T). Note that the degradation of the machine is not time-dependent (Markovian). However, the desired degradation is only the function of the time (because the optimization goal is to make the machine reach the desired maintenance time while keeping the output within acceptable limits).

Applying the dynamic programming for this optimization, we have

$$J_N(T) = e_N(T)' Q_1 e_N(T) \quad (5)$$

$$J_{k=1:N-1}(t = 1 : T) = \min_{u_k} E \left\{ e' Q_1 e + u' Q_2 u + G' Q_3 G + J_{k+1}(Ax_k + Bu_k + v_{1k}) \right\}, \quad (6)$$

where J_k is the cost-to-go from state x_k at time t to x_N at time T .

Now for calculating the best trajectory, the cost function should be calculated backward from x_N at time T to x_1 at time 0 [16]. For now, we assume that the degradation of the machine can be identified as a function of the machine's states and inputs. In this case, we have

$$D_{x,u} = \begin{bmatrix} W_x & W_u \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}. \quad (7)$$

Now expanding (6) gives

$$J_k(t) = e' Q_1 e + \min_{u_k} E \left\{ u' Q_2 u + G' Q_3 G + (Ax_k + Bu_k + w_k)' Q_1 (Ax_k + Bu_k + w_k) \right\}. \quad (8)$$

Expanding this statement and differentiating it in order to find the u_k^* (optimal input) and assuming that $E\{v_1\} = 0$ will result in

$$u_x(t)^* = -(Q_2 + B' Q_1 B)^{-1} (B' Q_1 Ax_k + (W'_x x_k(t-1) + W'_u u_k(t-1) - D_d(t-1)) W_u Q_3). \quad (9)$$

This equation is composed of two parts, the first part

$$u_x^* = -(Q_2 + B' Q_1 B)^{-1} (B' Q_1 Ax_k), \quad (10)$$

which is the infinite horizon optimal feedback, and only depends on x_k , and second part

$$u(t)^* = -(Q_2 + B' Q_1 B)^{-1} (W'_x x_k(t-1) + W'_u u_k(t-1) - D_d(t)) W_u Q_3, \quad (11)$$

which depends on t, x and u and is the adaptive degradation compensation feedback.

2.2 Relevance Vector Machine

In (7), it was assumed that the degradation of the system could be calculated as a function of the system's state and input. There are many methods to do this mapping. One of these methods used for this research is the Relevance Vector Machine (RVM). The relevance vector machine (RVM) was first introduced in [17]. The RVM structure is very similar to the support vector machine, which is given as follows:

$$\hat{y}(x) = \sum_{\ell=1}^L w_{\ell} k(\mathbf{x}, \mathbf{x}_{\ell}) + c, \quad (12)$$

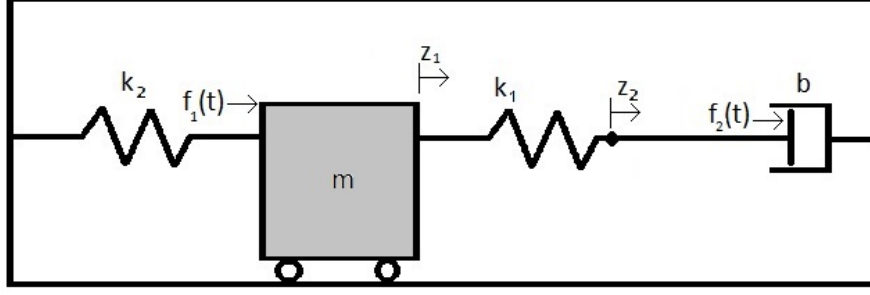


Fig. 2 Mass and Spring Model

where w_ℓ is a coefficient in \mathbf{w} , which is the vector of coefficients, $k(\cdot, \cdot)$ is the kernel function, and c is the bias parameter. RVM defines the conditional distribution for the target value \mathbf{y} given the vector of covariates \mathbf{x} , prediction outcome $\hat{\mathbf{y}}$, regression coefficients \mathbf{w} , and a precision parameter called ψ as

$$p(y_\ell | \mathbf{x}_\ell, \mathbf{w}, \psi) = \mathcal{N}(y_\ell | \hat{y}(\mathbf{x}_\ell), \psi^{-1}), \quad (13)$$

$$\hat{\mathbf{y}} = \Phi(\mathbf{X})\mathbf{w}, \quad (14)$$

$$\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_L)]^T, \quad (15)$$

$$\phi(\mathbf{x}_\ell) = [1, k(\mathbf{x}_\ell, \mathbf{x}_1), k(\mathbf{x}_\ell, \mathbf{x}_2), \dots, k(\mathbf{x}_\ell, \mathbf{x}_L)]. \quad (16)$$

The likelihood function for \mathbf{y} can be written as

$$p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \psi) = \prod_{\ell=1}^L p(y_\ell | \mathbf{x}_\ell, \mathbf{w}, \psi^{-1}). \quad (17)$$

RVM introduces a prior distribution for each w in \mathbf{w} as a hyperparameter α

$$p(\mathbf{w} | \alpha) = \prod_{i=1}^M \mathcal{N}(w_i | 0, \alpha_i^{-1}), \quad (18)$$

where M is the number of covariates (bias included). The hyperparameter α measures the precision of each w_i . Following Bayesian inference, the distribution of the weights becomes Gaussian and takes the following form:

$$p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \alpha, \psi) = \mathcal{L}(\mathbf{w} | \mathbf{m}, \Sigma), \quad (19)$$

in which

$$\mathbf{m} = \psi \Sigma \Phi^T \mathbf{y}, \quad (20)$$

$$\Sigma = (\Delta_\alpha + \psi \Phi^T \Phi)^{-1}, \quad (21)$$

and $\Delta_\alpha = \text{diag}(\alpha_i)$.

2.2.1 Results

For this proof of idea, the model used is the simple mass and spring model with two inputs. The reason for considering two inputs is that, for SISO systems, with a correct configuration of Q s, the optimal input of the degradation controller would be the same as the optimal input calculated by solving the Ricatti equation for LQR because the only parameter available to control both output and degradation would be a single input which prioritizing the output quality will make the system a regular controller, not a degradation controller. However, having more than one input will make the controller capable of controlling two outputs (the system's output and system degradation).

The model used for this simulation is shown in Fig.2, and its respective state-space is shown in (22) and (23) [18].

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{k_2-k_1}{m} & 0 & -\frac{k_1}{m} \\ \frac{k_1}{b} & 0 & -\frac{k_1}{b} \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ \dot{z}_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{b} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad (22)$$

$$y = Ix, \quad (23)$$

where z_1 and z_2 are displacements shown in Fig.2, m is the mass, k_1 and k_2 are respective spring constants, b is the damping ratio and f_1 and f_2 are respective input forces. The parameter to control or the desired output for this simulation is z_1 , which is the position of the mass. Although, according to different conditions, the degradation might differ in real situations, to reduce the complexity of the result and be able to focus on the research idea, the degradation parameter chosen to be k_1 . The degradation model for k_1 considered to be exponential [19]:

$$k_1(t+1) = k_1(t) + 2 \times 10^{-5} \times \exp(t * 5 * 10^{-5}). \quad (24)$$

Two failure thresholds were defined for this simulation:

$$|\hat{z}_1(t) - z_1(t)| > 0.01, \quad (25)$$

$$k_1 > 0.02, \quad (26)$$

where $\hat{z}_1(t)$ is defined as the desired output at time t , and $z_1(t)$ is the system output at time t , and the system is considered as failed (from the maintenance point of view) when the difference between actual output and desired output passes this threshold. The closed-loop step response of the system is shown in Fig.3. It can be seen that the controller is controlling the system according to the desired output. Fig.4 shows the same controller under degradation explained in (24) and failure threshold defined in (25) and (26). The left part of the figure shows the normal controller and on the right is the response of the degradation controller. Note that the x-axis of the Fig.4 is in cycles, which means that only features of the output (in this case, maximum) are recorded from each cycle and plotted in the figure, so each point in the figure is a feature recorded from a whole cycle as shown in Fig.1). The degradation coefficients

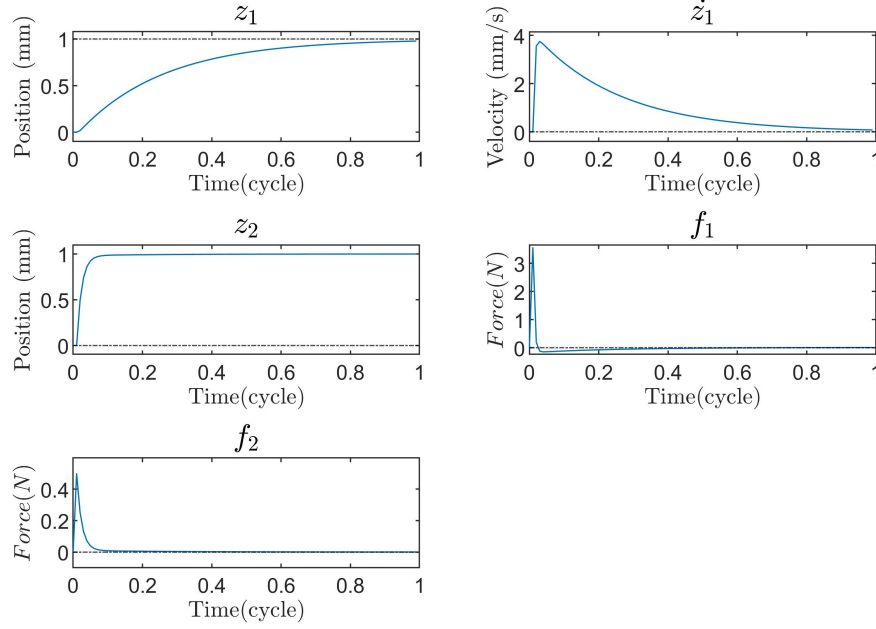


Fig. 3 Closed-loop step response of the infinite horizon controller

W_x and W_u are calculated according to recorded data shown in the left part of Fig.4.

It can be seen that the regular controller loses control, the system output deviates from the desired output by an amount more than the failure threshold, and the system is considered as failed after around 90 cycles. However, the degradation controller keeps the output inside the acceptable threshold for around 500 cycles. In the end, the simulation finishes not because of the output deviation but because of the failure threshold mentioned in (26)..

3 Discussion

Controlling the degradation at the same time as the output will impact the industry and reduce the maintenance cost. Unlike most existing control methods, the degradation control will depend upon machine learning methods because identifying the physical laws governing the degradation is costly and environment-dependent. Although machine learning methods are considered the best solutions for solving modern control problems, adapting them to existing methods will be challenging. This article proved that with the correct choice of machine learning method, it is possible to adapt the traditional control method with machine learning. The stability of the closed loop is among the most critical questions that answering it is outside the scope of this

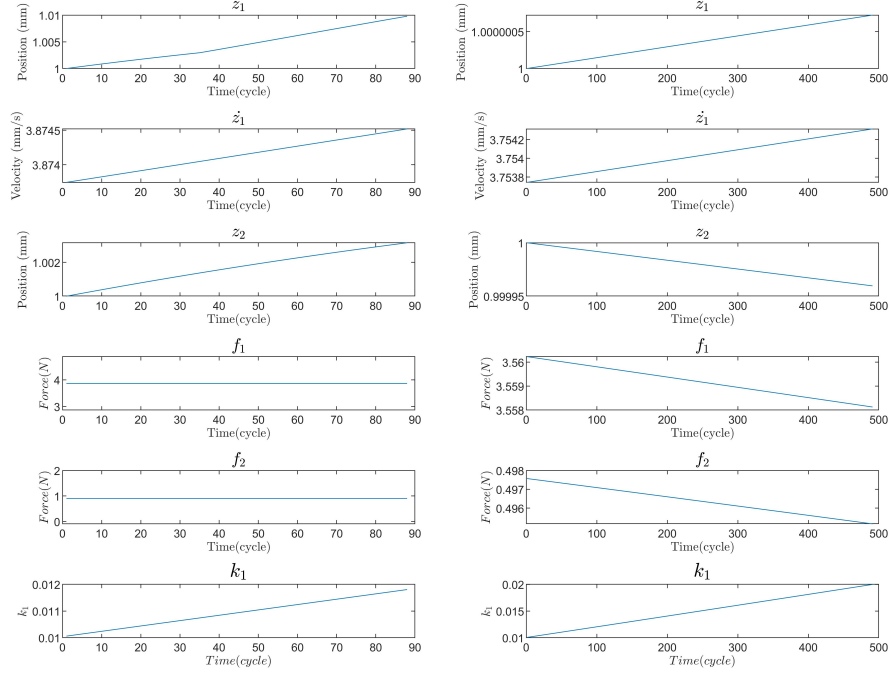


Fig. 4 Controller without degradation awareness (left) vs. degradation controller (right)

article. However, the short answer is that another cost function can be used to find the optimal point if the feedback makes the system unstable.

4 Conclusion

In this article, the degradation-aware adaptive feedback loop was introduced. First, the formulation of the quadratic cost function of the infinite horizon controller was updated, and the third term for penalizing the controller was introduced. Then using dynamic programming, the optimal feedback was calculated, which is a combination of the infinite horizon optimal feedback and time, state, and output dependent function. Then the limitation of identifying the degradation was considered in the formulation and adapted to the limitations, and RVM was used as the machine learning method compatible with the formulation and limitations. Finally, with the help of simulation, the functionality of the degradation-aware feedback loop was proved.

Acknowledgements The research project is financed by the European Commission within the European Regional Development Fund, Swedish Agency for Economic and Regional Growth, Region Gävleborg, and the University of Gävle.

References

1. C. Hua, L. Li and S. X. Ding, "Reinforcement Learning-Aided Performance-Driven Fault-Tolerant Control of Feedback Control Systems," in *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3013-3020, June 2022, doi: 10.1109/TAC.2021.3088397.
2. E. Chemali et al., "Minimizing battery wear in a hybrid energy storage system using a linear quadratic regulator," *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 003265-003270, doi: 10.1109/IECON.2015.7392603.
3. Abbaspour, Alireza, et al. "A survey on active fault-tolerant control systems." *Electronics* 9.9 (2020): 1513.
4. J. Hu, Q. Sun, Z.-S. Ye, Condition-based maintenance planning for systems subject to dependent soft and hard failures, *IEEE Transactions on Reliability* 70 (2020) 1468–1480.
5. E. M. Omshi, A. Grall, S. Shemehsavar, A dynamic auto-adaptive predictive maintenance policy for degradation with unknown parameters, *European Journal of Operational Research* 282 (2020) 81–92
6. N. Björzell, A. H. Dadash, Finite horizon degradation control of complex interconnected systems, *IFAC-PapersOnLine* 54 (2021) 319–324.
7. Salazar, Jean Carlo, et al. "Health-aware and fault-tolerant control of an octorotor UAV system based on actuator reliability." *International journal of applied mathematics and computer science* 30.1 (2020): 47-59.
8. Ahmed, Ibrahim, Marcos Quiñones-Grueiro, and Gautam Biswas. "Fault-Tolerant Control of Degrading Systems with On-Policy Reinforcement Learning." *IFAC-PapersOnLine* 53.2 (2020): 13733-13738.
9. J. S. Read, X. Jia, J. Willard, A. P. Appling, J. A. Zwart, S. K. Oliver, A. Karpatne, G. J. Hansen, P. C. Hanson, W. Watkins, et al., Processguided deep learning predictions of lake water temperature, *Water Resources Research* 55 (2019) 9173–9190
10. A. Daw, A. Karpatne, W. Watkins, J. Read, V. Kumar, Physics-guided neural networks (pgnn): An application in lake temperature modeling, *arXiv preprint arXiv:1710.11431*
11. S. H. Rudy, S. L. Brunton, J. L. Proctor, J. N. Kutz, Data-driven discovery of partial differential equations, *Science advances* 3 (2017).
12. S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the national academy of sciences* 113 (2016) 3932–3937.
13. J. Sun, H. Zuo, W. Wang, M. G. Pecht, Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance, *Mechanical systems and signal processing* 28 (2012) 585–596.
14. G. Storvik, Particle filters for state-space models with the presence of unknown static parameters, *IEEE Transactions on signal Processing* 50 (2002) 281–289.
15. T. Glad, L. Ljung, *Control theory*, CRC press, 2018
16. Bertsekas, Dimitri. *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific, 2012.
17. M. E. Tipping, Sparse bayesian learning and the relevance vector machine, *Journal of machine learning research* 1 (2001) 211–244.
18. Close, C.M., Frederick, D.K. and Newell, J.C., 2001. *Modeling and analysis of dynamic systems*. John Wiley & Sons.
19. Gebraeel, N., 2006. Sensory-updated residual life distributions for components with exponential degradation patterns. *IEEE Transactions on Automation Science and Engineering*, 3(4), pp.382-393.