A Reinforcement Learning Approach to Health Aware Control Strategy

Mayank S Jha*, Philippe Weber, Didier Theilliol, Jean-Christophe Ponsart

Centre de Recherche en Automatique de Nancy
(CRAN) UMR 7039,
Faculté des Sciences et Technologies
Boulevard des Aiguillettes - BP 70239 - Bât. 1er cycle
54506 Vandoeuvre-lès-Nancy Cedex
France

Presented by Didier Maquin
Professor, Université de Lorraine, CRAN

** Corresponding Author: Mayank JHA , Email: mayank-shekhar.jha[at]univ-lorraine.fr*

Contents

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Introduction

This work:

- Novel framework for health aware control under component degradation.

- Reinforcement learning based approach:
  - Learn an optimal control policy integrating global system transition data and RUL prediction data.
  - The RUL prediction generated at each step, is tracked to a desired value of RUL.
  - Integrated within a cost function → maximized to learn the optimal control.

- Presents a novel way of integrating model based methods with data driven techniques.

- Also, presents method to integrate Artificial Intelligence for health aware control.

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Background

Health Aware control (HAC) → Control design based upon:

- Health of components/system,
- Remaining Useful Life (RUL) predictions → Prognostics of component/system.
- Change in System Loading / Operating condition → Assure mission completion, optimal performance, etc.

Challenges

- Health predictions generated by degradation model.
  - Degradation models → usually, unknown or partially known.
  - RUL model (transition model) not available: Predictions must be generated using *l*-step ahead predictions.

- Integration of RUL data base with system dynamics for control synthesis.

- How to obtain optimal control based upon l-step ahead RUL predictions?

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Background

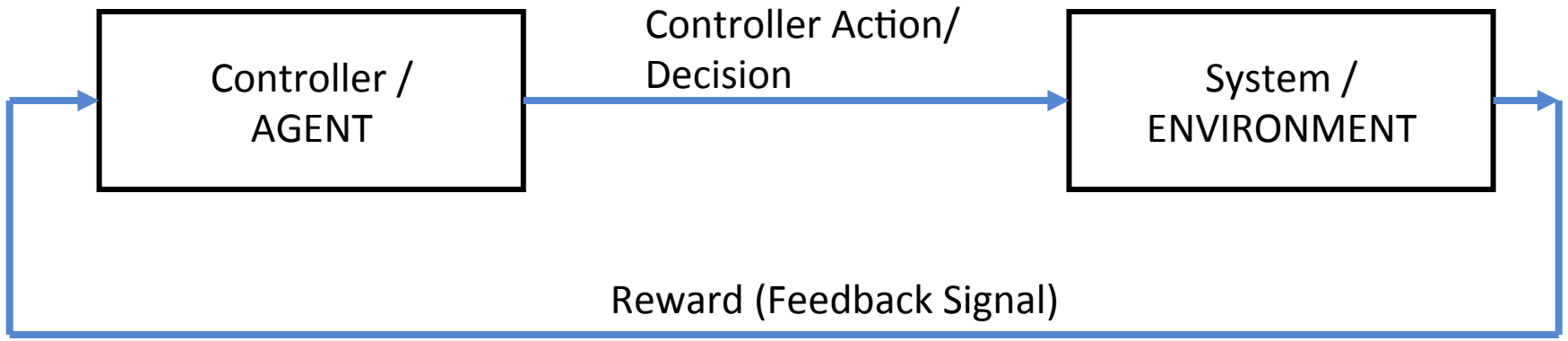
Reinforcement Learning: Interaction of controller (Agent) with system (Environment) .

System : Formalized using a Markov Decision process (MDP).

Based upon controller action (decision) → '***Reward***' (Feedback Signal) is generated by system output.

The cost (value) associated with a control policy (action) should be maximized.

Objective → maximize the reward received → learn optimal control policy that gives maximum reward *for all* system states.



Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Problem Formulation

- Nonlinear system affine in control in discrete time *k*

$$x_{k+1} = f_s(x_k) + g(x_k)u_k$$

- Control policy: $h(\cdot): X \rightarrow U$

$$u_k = h(x_k)$$

- Degradation Model : Critical component/subsystem (chosen a priori).

$$d_{k+1} = f_d(d_k, m, x_k)$$

- RUL_k is generated using a l-step ahead prediction:

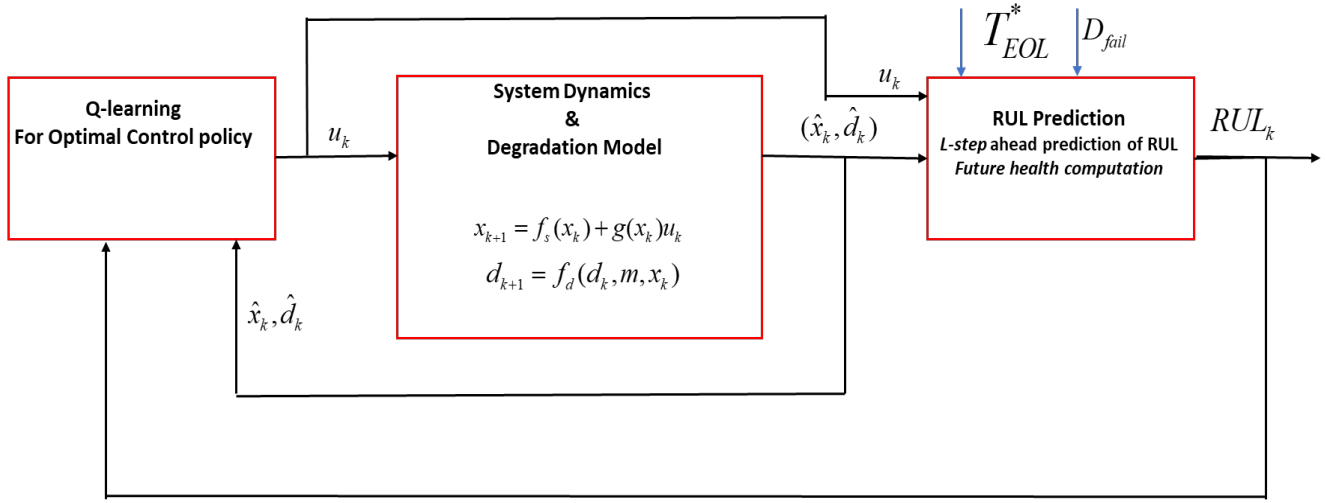
Projecting DM into future over an infinite horizon till failure value D_{fail} is reached, assuming control action same as u_k

D_{fail} : Given

$$d_{k+1} = f_d(d_k, m, x_k); \quad x_{k+1} = f_s(x_k) + g(x_k)u_k$$
$$d_{k+2} = f_d(d_{k+1}, m, x_{k+1}); \quad x_{k+2} = f_s(x_{k+1}) + g(x_{k+1})u_k$$
$$\vdots$$
$$d_{k+l} = f_d(d_{k+l-1}, m, x_{k+l-1}); x_{k+l} = f_s(x_{k+l-1}) + g(x_{k+l-1})u_k$$

Assumption 1: System states considered observable.

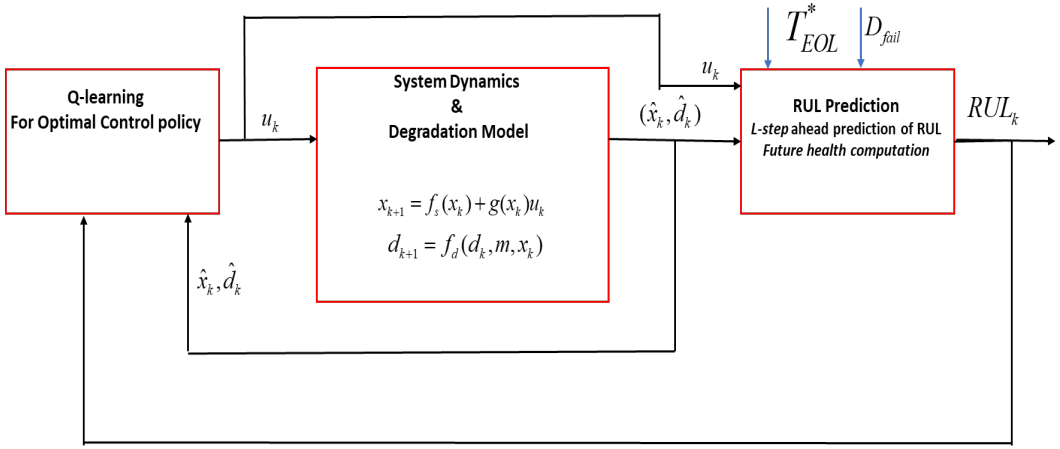
Assumption 2: There exists a admissible control policy so that closed loop is asymptotically stable.



Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Problem Formulation

- ***Desired end of life*** (EOL) T^*_{EOL} : Given
  - Mission/user dependent.
- ***Desired RUL*** at time *k* can be generated : $RUL^*_k = T^*_{EOL} - k\,T_s$
- The previous l-step ahead prediction produce the actual value of the RUL : RUL_k



The objective is now to compute an optimal RUL tracking in the framework of Q-learning algorithm

- Reward generation at *k* : $r_{k+1} = -\frac{1}{2}\left(x_k^T\,\mathbf{S}\,x_k + u_k^T\mathbf{R}\,u_k + (RUL^*_k - RUL_k)^T\,\mathbf{P}\,(RUL^*_k - RUL_k)\right) = \rho(x_k, RUL_k, u_k)$

  Minimize difference $(RUL^*_k - RUL_k)$ while assuring minimal energy consumption and system performance.

  Cumulative reward or return (from state and RUL): $R^h(x_k, RUL_k, u_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r_{i+1} = \sum_{i=k}^{\infty} \gamma^{i-k}\, \mathrm{r}_{k+1}(x_i, RUL_i, h(x_i))$

  (discounted cost over infinite horizon)

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Problem Formulation

Q-function (state-action pair) is defined as follows.

It generates return obtained by applying u_k at x_k

following policy $h(x_k)$ thereafter.

$$Q^h(x_k, RUL_k, u_k) = r_{k+1}(x_k, RUL_k, u_k) + \gamma R^h(x_{k+1}, RUL_{k+1}, h(x_{k+1}))$$

This approach is clearly linked to dynamic programing approach

Bellman's equation for Q-function :

$$Q^h(x_k, RUL_k, u_k) = \rho(x_k, RUL_k, u_k) + \gamma Q^h(x_{k+1}, RUL_{k+1}, h(x_{k+1}))$$

Bellman optimal equation for Q-functions:

$$Q^*(x,u) = \rho(x,u) + \gamma \max_{u'} Q^*(f_s(x,u),u')$$

Optimal control is the one that maximizes Q*

whatever the state considered

$$h^*(x) = \arg\max_{u \in U} Q^*(x, RUL, u)$$

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Q-Learning Algorithm

- Using only the observed state transitions and rewards, i.e., data tuples $(x_k, RUL_k, u_k, x_{k+1}, RUL_{k+1}, r_{k+1})$

$$Q_{k+1}(x_k, RUL_k, u_k) = Q_k(x_k, RUL_k, u_k) + \alpha_k \Big[\underbrace{r_{k+1} + \gamma \max_{u'} Q_k(x_{k+1}, RUL_{k+1}, u')}_{\text{updated estimate}} - \underbrace{Q_k(x_k, RUL_k, u_k)}_{\text{current estimate}} \Big]$$

temporal difference

- Control action selection : Exploration – Exploitation Routine (Epsilon-greedy strategy)

Exploration : Take a Random control action → Explore what kind of reward an action leads to.

Exploitation: Take only that action which gives maximum return → Greedy approach, exploit the learnt values.

$$u_k = \begin{cases} u \in \text{random (U)} \quad \text{with probability } \boldsymbol{\varepsilon} \quad (\textit{exploration}) \\ u \in \arg\max_{u \in U} Q(x, RUL, u) \text{ with probability } 1\text{-}\boldsymbol{\varepsilon} \ \ (\textit{exploitation}) \end{cases}$$

$\varepsilon \in (0,1)$ Exploration probability which can be indexed on the number of episodic runs.

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Simulation Study : DC motor with Shaft Wear

DC Motor Linear Model Discrete time (MDP)

(State Space → Markov Model, Control input → Decision variable)

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} 0.9 & -0.001 \\ 0.001 & 0.99 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} 0.01 \\ 0 \end{bmatrix} u_k$$

Simplified wear model (Archard Equation) :

Shaft wear as function of shaft speed.

$$Hw_{k+1} = Hw_k + T_s C_w \boldsymbol{\omega}_k$$

Open Loop Characteristics :

Step Input 10V (Maximum)

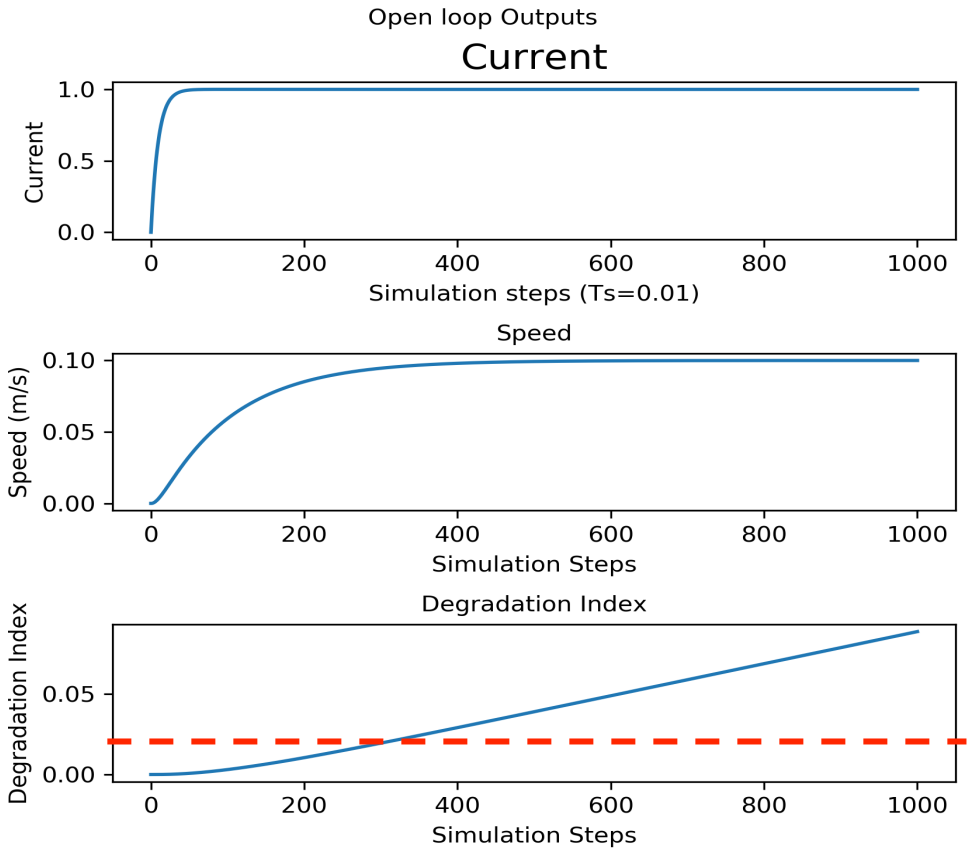
One ***Episodic Play*** **(10s of system functioning):**

1000 steps system simulation

(System data collection)

$$Hw_{fail} = 0.02 \ (\mathrm{m}^3/\mathrm{s})$$

$$\mathrm{T_{EOL}} = 3.4\mathrm{s}$$



Hw_{fail}

$$i(t) \in [0, 0.1]\,\mathrm{A}$$

$$\omega(t) \in [0, 10^{-1}]\,\mathrm{rad\ s^{-1}}$$

$$u(t) \in [0, 10]\,\mathrm{V}$$

$$T_s = 0.01s$$

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Simulation Study : DC motor with Shaft Wear

Objective: *Reach a desired T^*_{EOL}=9s by learning a suitable control law from system transition data.*

The wear rate *Hw* reach not more than 0.02 (m^3/s) at the end of 900 simulation steps

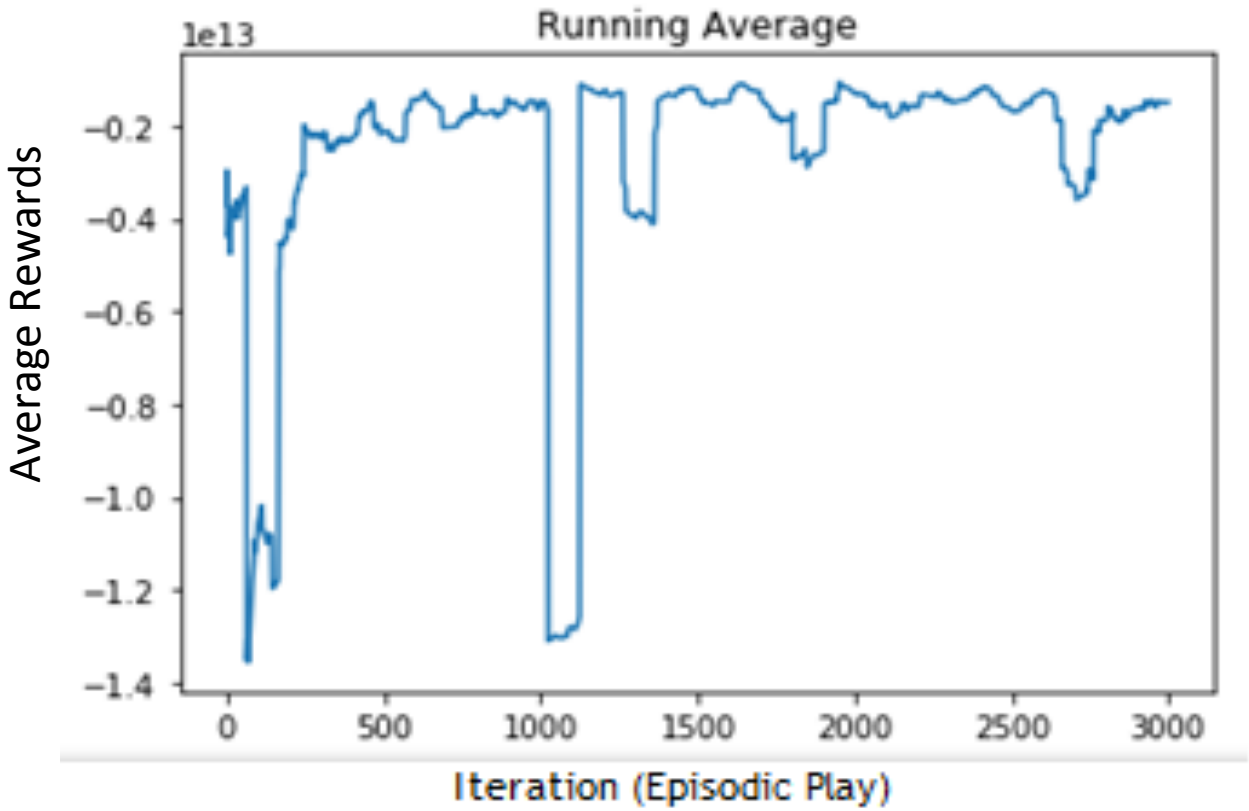
Reward: $$r_{k+1} = -\frac{1}{2}\left(\begin{bmatrix} i_k & \omega_k \end{bmatrix} \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + 10\, u_k^2 + 100\,(RUL_k^* - RUL_k)^2 \right)$$

- For Q-learning, a naïve tabular approach is adopted : State – Action pair table stores Q-value.
- *One episodic play*
  - Simulate DC motor model (generate system data) + Generate RUL prediction at each step k,
  - Generate reward
  - Update Q-value using Q-learning algorithm
- Repeat this process (for number of episodic plays) until saturation (optimality) of Q-values in Q-table is reached for all states.

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Results

- Total Episodic plays : 3000
- Convergence detected: approx. 1500 Episodic plays.



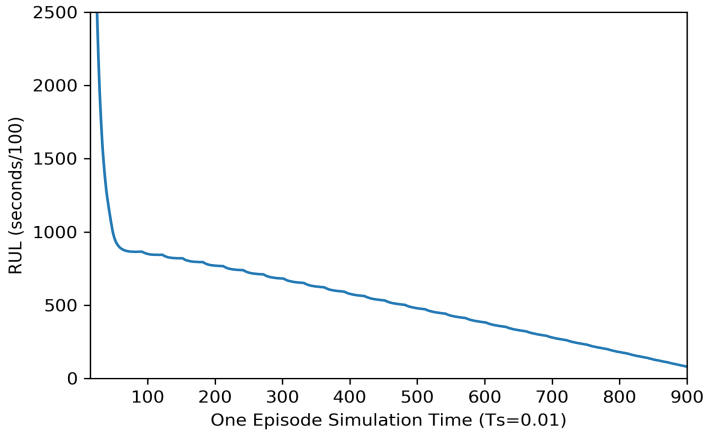
Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

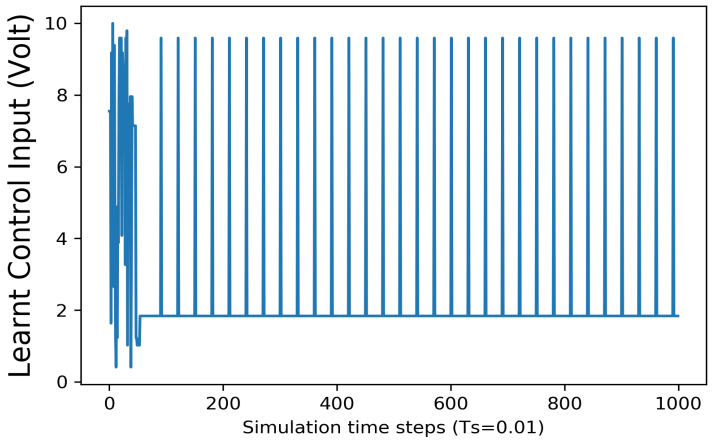
Results

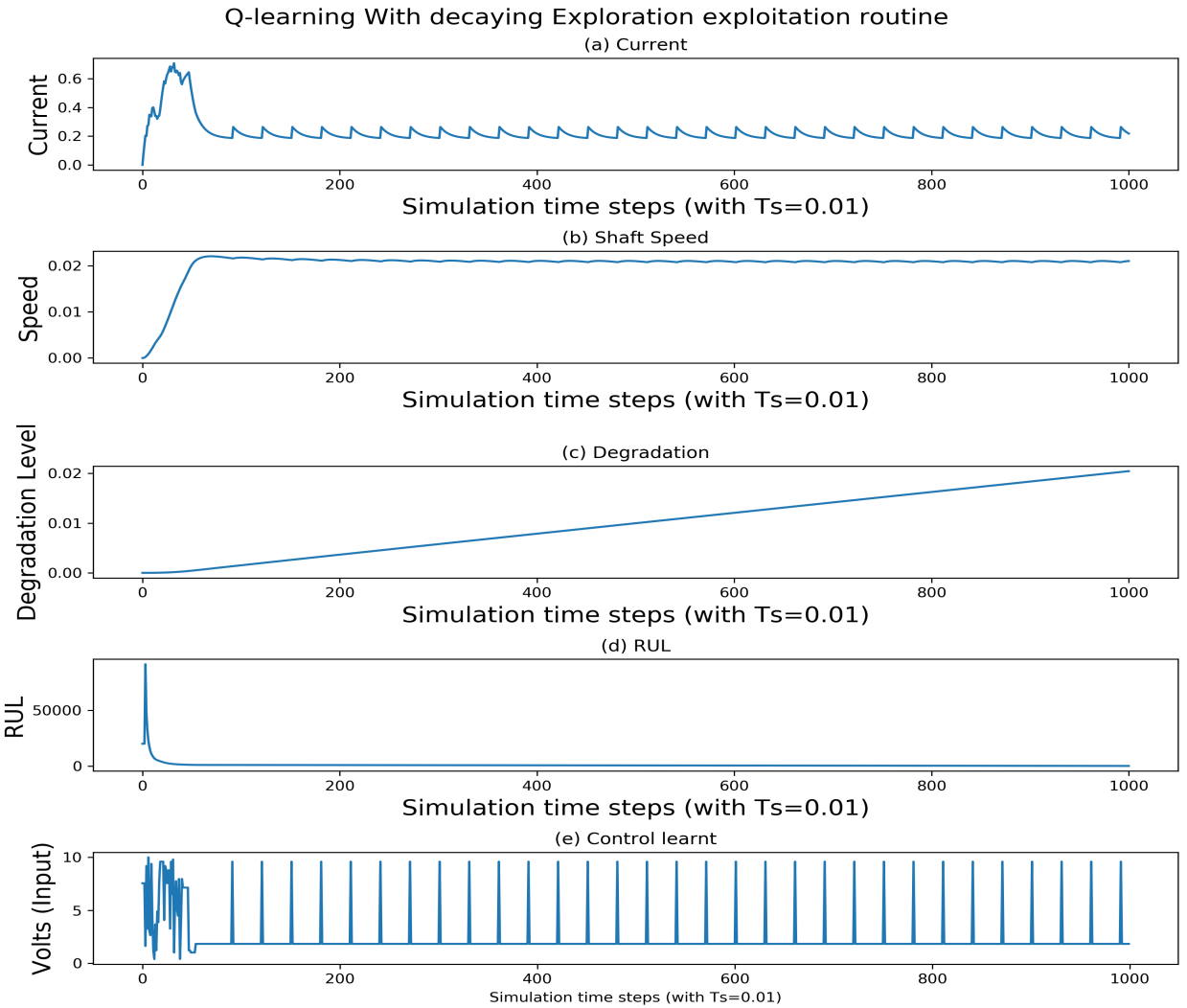
Behavior under learnt Control policy (law) :

Degradation levels is restricted to Hw_{fail} value at the end of 10s or 1000 simulation steps.

Desired RUL is reached at the end of 9s of system functioning (900 simulation steps).







Corresponding Author Email: ***mayank-shekhar.jha [at] univ-lorraine.fr***

Conclusions and Perspectives

- **Novel framework:** Reinforcement learning based optimal control law in face of component degradation.
- **Novel integration:** Global system transition data (generated by an analytical model that mimics the real system) and RUL prediction data to learn optimal control.
- Control policy that manages the speed of degradation in a way such that desired RUL is reached.
- *Q-functions* do not require model knowledge to learn optimal policy.

  In absence of accurate dynamics, *experience replay (*using episodic data iteratively) to attain optimality.
- An inevitable disadvantage to managing the RUL is that the control is learnt off-line.
- It would be interesting to handle non-linear degradation process and hidden system states.
- Efficient function approximation for large state spaces.

Corresponding Author Email: *mayank-shekhar.jha [at] univ-lorraine.fr*