2022 IEEE 17th International Conference on Control & Automation (ICCA)
June 27-30, 2022 (Hybrid) Naples, Italy

Model-Assisted Reinforcement Learning for Online Diagnostics in Stochastic Controlled Systems

Erfaun Noorani, Christoforos Somarakis, Raman Goyal, Alexander Feldman, and Shantanu Rane

***Abstract*— A mechanism to protect a controlled system in the event of *a priori* unknown abnormalities (e.g. faults, attacks) is the key to designing resilient and robust control systems. We explore bi-level control design architectures in which a supervisory Reinforcement Learning (RL) agent augments an over-observed controlled system. The RL agent monitors sensor signals, detects and takes action to mitigate unknown sensor faults. We use the system dynamics to extract features and develop a design method for the cost function of the RL module. We theoretically show that the designed cost function has a unique optimal policy that enables the diagnosis of arbitrary constant sensor faults. To conceptualize our architecture, we consider a linear version of an over-observed chemical process, controlled by a Linear Quadratic Gaussian (LQG) Servo-Controller with Integral Action. Our experimental results, coupled with our theoretical analysis, show that the RL-agent is successful in identifying and mitigating the faults in one or more sensors in an online fashion.**

I. Introduction

Cyber-Physical system abnormalities such as faults or malicious attacks are inevitable in today's increasingly interconnected world. The security of these systems has been the main focus of several communities with significant research activity that continuous into its third decade [1], [2], [3]. In particular, in the control community, the efforts have focused on fast and efficient methods of detection and tracking of misbehaviors [4], [5]. The complexity of the problem has led to approaches based on a taxonomy of the types of faults and attacks on the control system (see [6] and references therein). The drawback is that the applicability of these security solutions is restricted to particular attacks and systems. They cannot be easily generalized across CPS systems or faults. In many cases, they also suffer from the curse of dimensionality, especially in discrete event systems [7]. A resilient system design allows for the continuation of mission performance under unknown abnormalities as well as tracking and characterization of these abnormalities. Model-based diagnostics is a well-established research field that is closely related to resilience constraints in system design. This field has yielded a plethora of approaches that require strong assumptions on *a priori* intrinsic knowledge of model dynamics [1], [8], [9]. Fault Tolerant Control (FTC) [10], [11] concerns the use of model-based approaches in the design of control systems. The traditional FTC methods rely on prior knowledge of the abnormalities and their effects on the system, i.e., known system dynamics and faults. The performance of such model-based approaches is limited by model inaccuracies and is sensitive to model miss-specification. The traditional design is typically composed of a fault detection module which – upon detection of the fault – switches to a low-level controller appropriate for that specific fault from a set of predefined fault controllers. The sequential nature of such design introduces a time-delay into the system that may degrade performance, even destabilize the system.

To mitigate the delay introduced by the traditional FTC design, Blended Control (BC) proposes a hierarchical design with a weighted combination of low-level controllers. The weights with which the controllers get combined constitute a blending weight vector which is set by a high-level control module. A deep-learning BC design [12] uses an RL algorithm to set the blending weights, to provide a data-driven approach to BC, which in turn eliminates the need for prior knowledge of the system dynamics. In such architecture, the high-level control effectively implements the low-level controller and does not directly interact with the controlled system. However, BC designs are intrusive in the sense that they aim to synthesize a fault-tolerant controller.

Our contribution is twofold: (I) We explore a bi-level supervisory control design for non-intrusive tracking and mitigation of faults in control systems. We explore the basic design steps so that the AI-enabled supervisory module will be able to operate in conjunction with the controlled system. Furthermore, we validate our design in simulation for a chemical process in the presence of additive sensor faults with unknown magnitude. (II) We offer a method to design a cost function for the RL agent's successful training, following a model-based Inverse Reinforcement Learning approach. The context within which we develop our theory focuses on either exact or over-observed systems, that is systems the state of which is delivered through either a single observer or multiple independent observes, and through some sensor fusion scheme. Systems with redundancy in observations lie at the core of modern intelligent control in industrial systems [13] and also provide us with a relevant fault/attack scenario under which system state is observed by many but not reliable sensing units.

The rest of the paper is organized as follows: Section II describes the overall architecture. Section III describes our closed-loop control system. In Section IV, we present the

E. Noorani, C. Somarakis, R. Goyal, A. Feldman, and S. Rane are with Palo Alto Research Center - A Xerox Company, Palo Alto, CA, USA. E. Noorani is with the Department of Electrical and Computer Engineering, the Institute for System Research (ISR) at the University of Maryland College Park, College Park, MD, USA. E. Noorani is a Clark doctoral Fellow at A. James Clark School of Engineering. {enoorani}@umd.edu, {enoorani,somarakis,rgoyal}@parc.com, {afeldman,srane}@parc.com

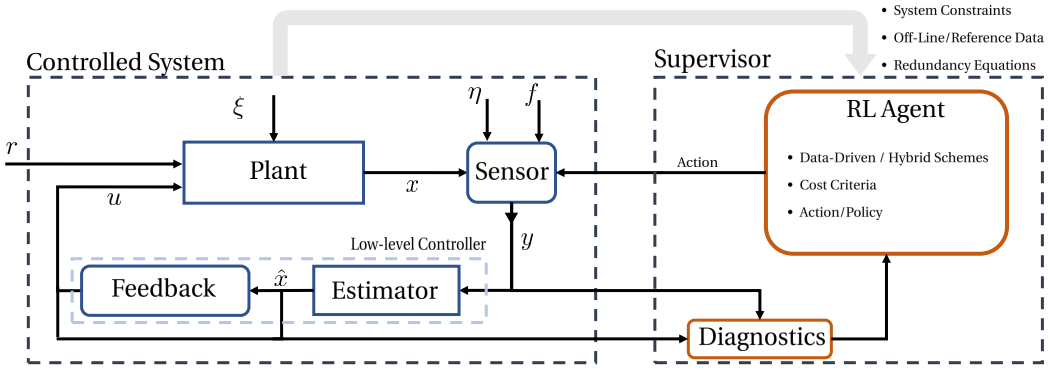

Fig. 1: The bi-level architecture with supervisory RL controller that takes action to defend the closed-loop plant against sensor faults with minimal intervention. The plant is prone to acceptable actuation and sensor noises. These are assumed to be zero-mean Gaussian and with covariance parameters known to the low-level controller. The input information arrives from the sensor fusion process out of a collection of heterogeneous sensors into some output signal y that is fed into the low-level controller. The RL agent is exposed to the output of the sensor and acts on its module.

high-level RL controller and present our approach to cost design in Section V. The training procedure and experimental results are discussed in Section VI. Finally, we make some concluding remarks in Section VII. Proofs of technical results are placed in the Appendix.

II. Bi-level Fault-Aware Control Architecture

Figure 1 illustrates the main components of a bi-level architecture considered in this work, that is comprised of:

- A controlled system which consists of: (I) A plant, (II) sensor module, (III) a closed-loop low-level controller. This controller can be a pre-designed traditional controller, e.g. PID, LQG, MPC, or a trained RL agent to drive output towards a reference signal.
- A diagnostic module which monitors the system for faults. The output of the diagnostic module provides the input (observations and cost signal) to the high-level RL-based controller.
- A high-level RL based controller which computes a mitigative action based on the input from the diagnostic module. The RL action is an input to the controlled system, e.g. an additive term to the sensor readings.

One advantage of this design idea is that it aims to guarantee sensor-based fault diagnostics for the controlled system without the need to re-synthesize the low-level controller. The challenge is to explore the terms under which the RL will be trained, collect data and take action to secure the system from abnormalities in cooperation with the low level closed-loop system. We will see in the sections to follow that this is a non-trivial task.

III. Controlled System

In this section, we discuss the elements of the controlled system as illustrated in Figure 1, that is the system dynamics, the low-level controller specs and the sensing component we will consider in this work.

A. State Equation

We consider linear time invariant

$$x_{t+1} = A\,x_t + B\,u_t + \Gamma\,\xi_t, \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $A \in \mathbb{R}^{n_x \times n_x}$ is the state matrix, $u \in \mathbb{R}^{n_u}$ is the control signal, $B \in \mathbb{R}^{n_x \times n_u}$ is the input matrix, $\xi_t \in \mathbb{R}^{n_\xi}$ is a noise vector of Gaussian white noise, and $\Gamma \in \mathbb{R}^{n_x \times n_\xi}$ is the diffusion matrix associated with the process noise source.

B. Sensor Module & Fault Models

The sensor module is instrumental in our framework. It consists of sensing units prone to standard noise but also vulnerable to faults or attacks. Also, it will be the gate where the RL agent will apply its actions. In this work we will consider two types of sensors:

1) A single sensing unit that delivers output

$$y_t = C\,x_t + N\,\eta_t + \phi_t \tag{2}$$

where $C \in \mathbb{R}^{n_y \times n_x}$ the output matrix, $\eta_t \in \mathbb{R}^{n_\eta}$ a vector of white noise with $N \in \mathbb{R}^{n_y \times n_\eta}$ diffusion matrix, and vector $\phi_t \in \mathbb{R}^{n_y}$ models disturbance due to possible faults.

2) A multi-sensing unit, that comprises of $s > 1$ sensors that observe system states independent of each other. Like the single-unit case, every sensor has its own fundamental inaccuracy modeled via sensor noise sources that are Gaussian in nature. Furthermore, sensors are vulnerable to additive faults modeled as constant, but uncertain, bias. Bias is the simplest and most common type of fault or attack. It can occur due to incorrect calibration, physical changes in the sensor system or it can be the result of types of jamming attacks [14], [15]. With these in mind, the k^{th} sensor outputs the following signal:

$$y_t^{(k)} = C_k\,x_t + N_k\,\eta_t^{(k)} + \phi_t^{(k)}, \qquad k = 1, \ldots, s. \tag{3}$$

Here $C_k \in \mathbb{R}^{n_y \times n_x}$ is the output matrix of the k^{th} sensor, $\eta_t^{(k)} \in \mathbb{R}^{n_\eta}$ the sensor noise and $N_k \in \mathbb{R}^{n_y \times n_\eta}$ is the associated diffusion matrix. Vector $\phi_t^k \in \mathbb{R}^{n_y}$ is a disturbance vector that models possible faults on the sensor. For simplicity, the sensor fusion module that feeds the low level controller is the average of sensors' outputs, i.e.:

$$\begin{aligned} y_t &= \frac{1}{s}\sum_{k=1}^{s}\left[C_k\, x_t + N_k\, n_t^{(k)} + \phi_t^{(k)}\right] \\ &= Cx_t + \frac{1}{s}\sum_{k=1}^{s}\left[N_k\, n_t^{(k)} + \phi_t^{(k)}\right], \end{aligned} \tag{4}$$

where $C \in \mathbb{R}^{n_y \times n_x}$ is the cumulative output matrix of x_t. Provided (A, C) is observable, the rationale behind selecting (4) as the input signal is that through this observation redundancy, the effect of sensor noise is averaged out.

For $s = 1$ Eqns. (3) and (4) boil down to (2) simplifying to single sensor measurements. However we will deliberately distinguish the $s = 1$ and $s > 1$ cases, both because of the way we will design RL agent's actions and interpret its behavior towards identifying and mitigating faults as we will explain in the sections to follow, and because of the significance of multi-sensing/over-observed systems and sensor fusion methods in industrial control systems.

Assuming controllability and observability of (1)-(2) , (1)-(4), and that we have information about the process and sensor noise statistics, we can design a linear LQG Servo Controller with integral action [16]. The controller includes a state estimator and implements an estimator feedback law and with gain that optimizes a quadratic cost in the mean value sense. The result is that output y_t asymptotically tracks reference signal r_t also in the sense of mean value. We will use the standard Kalman estimator, with dynamics

$$x_{t+1|t} = A\, x_{t|t-1} + B\, u_t + L\,(y_t - C\, x_{t|t-1}), \tag{5}$$

where $x_{t+1|t} =: \hat{x}_{t+1} \in \mathbb{R}^{n_x}$ is the state prediction before update, at time $t+1$, and the Kalman output

$$x_{t|t} = (I_{n_x} - MC)\, x_{t|t-1} + M\, y_t, \tag{6}$$

where I_{n_x} is the $n_x \times n_x$ identity matrix. In (5) and (6), L and M are the Kalman and innovation gains, respectively. In this paper we consider linear quadratic servo controllers with integral action (LQG-i) scheme for our low level controller. To this end, the integrator dynamics are

$$x_{t+1}^{(i)} = x_t^{(i)} + (r_t - y_t) \tag{7}$$

where $r_t \in \mathbb{R}^{n_y}$ is the given reference signal. The control law reads

$$u_t = -\begin{bmatrix} K_{\hat{x}} & K_{x^{(i)}} \end{bmatrix} \begin{bmatrix} x_{t|t} \\ x_t^{(i)} \end{bmatrix} \tag{8}$$

for matrices $K_{\hat{x}}$ and $K_{x^{(i)}}$, designed to minimize some quadratic cumulative objective functional[1].

[1]Details of Kalman filter gains and linear quadratic servo controllers with integral action are beyond the scopes of this work, and thus omitted. We refer the interested reader to [16] for detailed analysis.

C. Closed-loop dynamics

If we combine (1) - (8), we can express the coupled system dynamics, that acts on the augmented state vector

$$\mathbb{X} = \begin{bmatrix} x \\ \hat{x} \\ x^{(i)} \end{bmatrix} \in \mathbb{R}^{2n_x + n_y},$$

and updates according to

$$\mathbb{X}_{t+1} = \mathbf{A}\,\mathbb{X}_t + \mathbf{B}\,\mathbb{N}_t + \mathbf{\Delta}\,\mathbb{D}_t, \tag{9}$$

where

$$\mathbb{N}_t = \begin{bmatrix} \xi_t \\ \eta_t^{(1)} \\ \vdots \\ \eta_t^{(s)} \end{bmatrix} \in \mathbb{R}^{n_x + s n_\eta}, \quad \mathbb{D}_t = \begin{bmatrix} \phi_t^{(1)} \\ \vdots \\ \phi_t^{(s)} \\ r_t \end{bmatrix} \in \mathbb{R}^{(1+s)n_y},$$

is the lumped stochastic vector that includes all sources of noise, and the disturbance vector that includes (possible) sensor biases together with the reference signal, respectively. Matrices $\mathbf{A}, \mathbf{B}, \mathbf{\Delta}$ are obtained from straightforward algebra and in closed forms that we omit to state due to space limitations.

D. Preliminary Results

We will conclude this section with the first and second moment expressions of (9), which we will use in section V to design a cost function for the RL-agent. Under zero-mean, Gaussian assumption and statistical independence of elements $\mathbb{N}_t$, we have

$$\mathbb{E}[\mathbb{X}_t] = \mathbf{A}^t \mathbb{E}[\mathbb{X}_0] + \sum_{l=0}^{t-1} \mathbf{A}^{t-1-l} \mathbf{\Delta}\, \mathbb{D}_l, \quad t > 0. \tag{10}$$

If we further assume that $r_t \equiv r$ and that disturbances are constant, we can write Eq. (10) in a more concise form:

$$\mathbb{E}[\mathbb{X}_t] = \mathbf{A}^t \mathbb{E}[\mathbb{X}_0] + \mathbf{W}_{t-1}\mathbb{D}, \tag{11}$$

where $\mathbf{W}_{t-1} := \sum_{l=0}^{t-1} \mathbf{A}^{t-1-l}\mathbf{\Delta}$, and $\mathbb{D}$ is independent of time having assumed that both faults and reference are constant. Furthermore,

$$\mathbb{E}[x_t] = \tilde{\mathbf{C}}\,\mathbb{E}[\mathbb{X}_t] = \tilde{\mathbf{C}}\mathbf{A}^t\mathbb{E}[\mathbb{X}_0] + \tilde{\mathbf{C}}\mathbf{W}_{t-1}\mathbb{D}, \tag{12}$$

for $\tilde{\mathbf{C}} := [I_{n_x} \,:\, O_{n_x} \,:\, O_{n_y}] \in \mathbb{R}^{n_x \times (2n_x + n_y)}$, where O_{n_x} is the $n_x \times n_x$ zero matrix. The asymptotic behavior of x_t in expectation, yields the following result.

Lemma 1: Assume system dynamics (1) is controlled to follow a constant reference signal with an LQG-i controller using output (4) that is prone to constant faults. Then for $\mathbf{A}, \mathbf{\Delta}$ matrices from the augmented system (9) and $\mathbf{W}_{t-1}$ from (12), the following limit holds true:

$$\lim_{t \to +\infty} \mathbf{C}\mathbf{W}_{t-1} = \left[-\tfrac{1}{s} I_{n_y} : \cdots : -\tfrac{1}{s} I_{n_y} : I_{n_y}\right],$$

where $\mathbf{C} := \left[C : O_{n_x} : O_{n_y}\right]$, $C = \frac{1}{s}\sum_{k=1}^{s} C_k$ and the right hand-side matrix is of order $n_y \times (s+1)n_y$.
Another useful expression comes from second moment dynamics.

Lemma 2: Let $\mathbf{M}$ by any $(2n_x+n_y)\times(2n_x+n_y)$ matrix. Then $\mathbb{E}[\mathbb{X}_t^T\,\mathbf{M}\,\mathbb{X}_t]$ attains an $\mathbf{M}$-parametrized expression that outlines its dependency on vector $\mathbb{D}$.

$$\mathbb{E}[\mathbb{X}_t^T\,\mathbf{M}\,\mathbb{X}_t] = \vartheta_t + \theta_t^T\mathbb{D} + \mathbb{D}^T\,\Theta_t\,\mathbb{D}, \tag{13}$$

where

$$\begin{aligned}
\vartheta_t &:= \mathbb{E}^T[\mathbb{X}_0](\mathbf{A}^T)^t\mathbf{M}\mathbf{A}^t\mathbb{E}[\mathbb{X}_0]+\\
&\qquad \mathbb{E}\big[\sum_{l=0}^{t-1}\mathbb{N}_l^T\mathbf{B}^T(\mathbf{A}^T)^{t-1-l}\mathbf{M}\mathbf{A}^{t-1-l}\mathbf{B}\mathbb{N}_l\big],\\
\theta_t^T &:= 2\mathbb{E}^T[\mathbb{X}_0](\mathbf{A}^T)^t\mathbf{M}\mathbf{W}_{t-1},\\
\Theta_t &:= \mathbf{W}_{t-1}^T\mathbf{M}\mathbf{W}_{t-1}.
\end{aligned}$$

Both Lemma's outline the basic statistical behavior of quantities of interest and will play a vital role in establishing the main analytic result of this work, in the section to follow.

IV. Supervisory Diagnostic Module

In this section, we elaborate on how the RL-agent perceives the situation (observation-space), acts on the controlled system (action-space) and the task that it aims to accomplish.

At the beginning of each learning episode, the agent starts in an initial state, that is a function of $\mathbb{X}$, the controlled-systems augmented state[2]. At each time-step, the RL-agent receives an observation (e.g. from the diagnostic module) and executes an action (e.g. by adding a term to the sensor measurement) according to its policy, a mapping from the observation to action, (possibly a smooth parametrized function such as a Neural Network). Upon the execution of the action, the system transitions to a successor state, and the agent receives an instantaneous cost c_t. The cost defines the task at hand.

The agent's policy, the stochasticity in the environment, and the observation model induce a probability distribution over the sequence of states and actions taken by the agent, what constitutes RL agent's trajectory, τ. The probability distribution of τ following the agent's policy π in a given environment and sensing model is denoted by ρ_π. An RL agent aims to find a policy, to minimize the system's cost measure over a given horizon, denoted by T. In classical RL [17], the desired cost measure of the system is usually an expectation of some long-run objective. A common example of such objective in RL literature is expected (undiscounted) cumulative cost, i.e.,

$$J(\pi) := \mathbb{E}_{\tau\sim\rho_\pi}\Big[\mathcal{C}(\tau)\Big], \tag{14}$$

where $\mathcal{C}=\sum_{t=0}^{T-1} c_t$ is the cumulative cost over an episode, and the expectation is taken over the policy's trajectory distribution ρ_π. Once the agent is trained, it can be deployed to run simultaneously in parallel with the controlled system.

[2]Therefore RL state should not be confused with the state of the closed-loop system.

Feature Extraction: In our case-study, we use three types of input signals from the low-level system at each time-step as the set of observations to the RL algorithm. The observation space of the RL algorithm consists of the low-level control signal u, the state estimate $\hat{x}$, and sensor residuals, i.e., $y^{(k)} - C_k\,\hat{x}$.

Action Space: The way the RL agent applies its actions on the low-level controller is through additive intervention on the sensing module as it is illustrated in Figure 1. The formal expression on RL's integration into system dynamics is the following modification of (3):

$$y_t = C_k\,x_t + \eta_t^{(k)} + \phi_t^{(k)} + a_t^{(k)}, \qquad k = 1,\dots,s, \tag{15}$$

with the restriction of $a_t^{(k)} \in \mathcal{A} \subset \mathbb{R}$, $\forall k,t$. Here $\mathcal{A}$ is a compact subset of $\mathbb{R}$ modeling the range of feasible faults that can appear on the sensors. In other words, the RL agent, by design, intervenes in the sensor module, with a set of actions spanning the set of sensors. Beyond that, and following the minimal intervention constraint to the closed-loop plant dynamics, the agent has no information about the location and magnitude of the fault, nor the low-level controller action on output and state. The RL agent's objective is to apply an action at each time step that blocks the injection of faults into the system, i.e. $\alpha_t \equiv -\phi_t$ where $\alpha = [a^{(1)},\dots,a^{(k)}]^T$ and $\phi = [\phi^{(1)},\dots,\phi^{(k)}]^T$ are the stacked $\mathbb{R}^{n_y}$-valued vectors of actions and faults, respectively.

RL algorithm: We used a Deep Deterministic Policy Gradient (DDPG) RL algorithm [18] which can be applied to continuous action-spaces, e.g. our case-study, and yields a deterministic policy. We show that the proposed architecture and methodology can be applied with satisfactory results without the need to extensively search in the space of available algorithms or possible hyper-parameters.

V. Cost Design Principles: Faults Mitigation

The design of the cost function described in (14) for online security is considered as the main contribution of this work. Given low-level system dynamics, sensor module and diagnostic tasks, we will elaborate on the necessary design principles that (14) must follow to achieve cooperation between the control hierarchies. By taking the tracking error at each time-step as the instantaneous cost, the RL agent will try to learn a mapping from the observations to actions, a policy, that minimizes (14) for the choice of $c_t{=}\|Cx_t{-}r_t\|^2$. Our objective is to deploy an RL policy such that $C\,x_t$ will track r_t. This may turn out to be a challenging task due to the action of low-level servomechanism. The latter module always imposes tracking of r_t from the output y_t. In the presence of faults, this means that servo-mechanism will displace Cx_t if necessary, essentially steering the system state to the undesirable range of values. One way for RL to mitigate this effect is to learn a policy that will minimize:

$$\mathcal{C}_M(\tau) = \sum_{t=0}^{T-1}\|C\,x_t - r_t\|^2,$$

in the sense of (14). We define $c_t := \|C\,x_t - r_t\|^2$ as *mitigation* cost function term because the policy that RL agent is

set to learn achieves cancellation of fault without necessarily identifying, where the error comes from, i.e., which sensor is at fault. The following result outlines this property by examining the policy that minimizes long-term cost, i.e., $\lim_{t\to+\infty}\mathbb{E}[c_t]$. To highlight the effectiveness of $\mathcal{C}_M$ and facilitate further analysis the following result assumes that reference signal, faults and actions are time-invariant.

Theorem 1: Consider the closed-loop system dynamics (9) with RL agent acting on output observables according to (15). Assume that reference signal is $r_t \equiv r \in \mathbb{R}^{n_y}$ i.e. constant, and that faults $\phi_t^{(k)} \equiv \phi^{(k)} \in \mathbb{R}^{n_y}$ are also constant, yet arbitrary. Then on the space of time-invariant actions implemented by the RL agent, it holds that

$$\lim_{t\to+\infty}\mathbb{E}\big[c_t\big] = \theta_\infty + (\phi+\alpha)^T\big(\frac{1}{s^2}\mathbb{J}_s \otimes I_{n_y}\big)(\phi+\alpha),$$

where $\mathbb{J}_s$, $s \geq 1$ is the $s \times s$ matrix of ones and $\phi, \alpha \in \mathbb{R}^{sn_y}$ are the group fault and action vectors, respectively. Moreover, the steady-state minimization vector of $\lim_t \mathbb{E}[c_t]$ lies in

$$\mathcal{M} = \left\{\alpha \in \mathbb{R}^{sn_y} \;:\; \sum_{k=1}^{s} a^{(k)} = -\sum_{k=1}^{s}\phi^{(k)}\right\}.$$

A moment of reflection on Theorem 1, may reveal that the constant fault and action hypotheses are only to simplify analysis in view of linear underlying dynamics. The big picture here is that RL agent trained towards policies that minimize $\mathcal{C}_M(\tau)$ may be successful in handling a wide class of faults. Our agent is greatly assisted by the fact that the system is governed by exponentially stable linear dynamics. Here a constant fault will shift the state vector to some other set point and convergence will occur exponentially fast. If fault duration is on a higher time scale than system rate of convergence, Theorem 1 is a good approximation of the policy the agent should seek while the fault lasts. It is reasonable to conclude that $\mathcal{C}_M$ is a good candidate reward function for piece-wise constant or slowly time varying faults, as we will document in §VI. The design principles encapsulated in the form of $\mathcal{C}_M$ are, from the designer's perspective both necessary and sufficient conditions for optimality in the mean square sense. However, from the RL agent's perspective, they are only necessary conditions. We approached the problem from the former point of view explaining the steps to provide the RL agent with the best possible reward functions within the existing plant-sensing configuration. The conditions under which the RL agent will converge to the desired policy regard the implemented RL algorithms and fall beyond the scope of this work. We believe however that majority of modern gradient-based algorithms have high chances of converging to desired minima in the space of $\mathcal{M}$. Our belief is supported by extensive simulation results.

The non-trivial step in the context we study was to identify the sources of noise, that were not simply coming from additive signals. The RL-agent is agnostic to low level dynamics, and its actions can easily get competitive to the low-level controller. From the mitigation reward function $\mathcal{C}_M$ we see that agent intervenes in sensor signal trying to stabilize state x_t (so that $C\,x_t$ will track r) while LQG-i controller aims to force y toward r. This is the reason that several intuitively obvious choices of cost functions, such as $\|y-r\|$ would not work because they would fail to decouple the sources of the fault: the system induced as a result of existing disturbance and the disturbance itself.

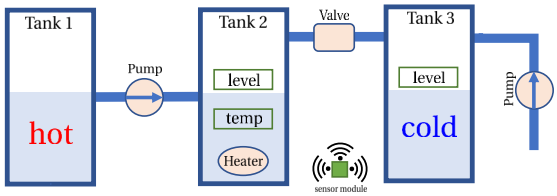

Fig. 2: The chemical process with four actuators (hot, cold pumps, valve and heater) that control the level and temperature of tank 2 and level of tank 3. The state is over-observed by three similar sensors.

VI. Numerical Simulation

We conceptualize our design and illustrate the effectiveness of our approach by means of simulation. We consider a chemical process shown in Figure 2 linearized along the lines of [19]. The three states $n_x = 3$ describe the level of water in tanks 2 and 3 and the temperature of water in tank 2. The control inputs is two flow pumps, one valve and one heater (i.e. $n_u = 4$), as illustrated in Figure 2. The actuator noise is assumed zero-mean Gaussian with covariance $\Gamma^T\Gamma{=}0.05I_3$. The low-controller objective is to regulate state vector around a reference value that for simplicity was taken equal to $r = [1,1,1]^T$. The linearized system outlined in [19] is given by

$$A{=}\begin{bmatrix} 0.96 & 0 & 0 \\ 0.04 & 0.97 & 0 \\ -0.04 & 0 & 0.90 \end{bmatrix}, B{=}\begin{bmatrix} 8.8 & -2.3 & 0 & 0 \\ 0.2 & 2.2 & 4.9 & 0 \\ -0.21 & -2.2 & 1.9 & 21 \end{bmatrix}.$$

The three-tank system is controlled by a LQG controller with integral action. The scenarios we consider are: exact (single sensor) observation, and redundant sensing with averaging sensor fusion scheme. In either scenario, sensors are prone to arbitrary random but constant faults and our goal is to train the supervisory agent embedded as in Figure 1, so that the controlled system will continue to regulate its output around r. We assume admissible faults within $[-18, 18]$ and therefore constrain the action space of the agent to $\mathcal{A}{=}[-20, 20]$. We trained the agent over 2000 episodes and 500 time-steps per episode. The Neural Networks structure and the hyper-parameters of the DDPG algorithm are left as the default values in the simulation platform (here `MATLAB` - Simulink).

A. Single-Sensor Measurements

Our explorations begin with one sensor that provides system measurements. The RL agent intervenes with an action signal a_t on the sensor that is prone to constant but arbitrary faults. Figure 3 consists of three plots that illustrate

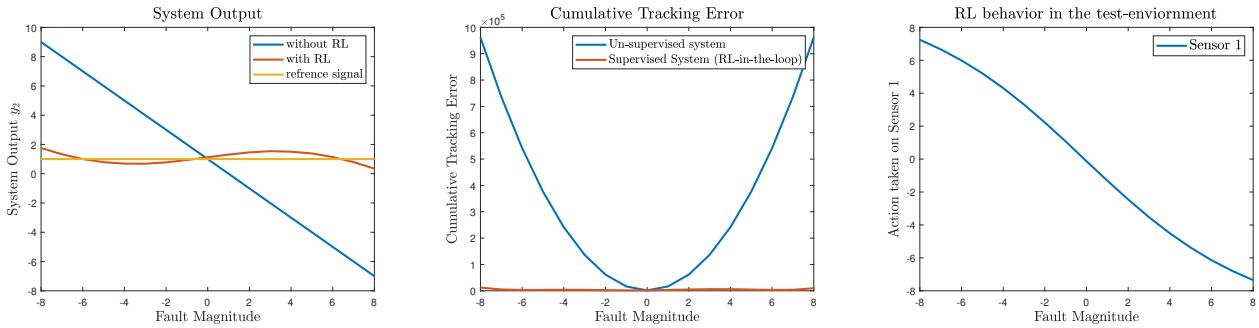

Fig. 3: Figure(a) (left) plots the average system output over an episode averaged over multiple runs against the fault magnitude at the test time; that is, the closer to the horizontal line $y = 1$, the better performance of the system. Figure(b) (middle) shows the system cumulative tracking error in an episode averaged over multiple runs against magnitude of the fault at the test time. Figure(c) (right) shows the mean action outputed by the RL agent over an episode averaged over multiple runs for different fault magnitudes. Note that the fault enters the system at the start of the episode.

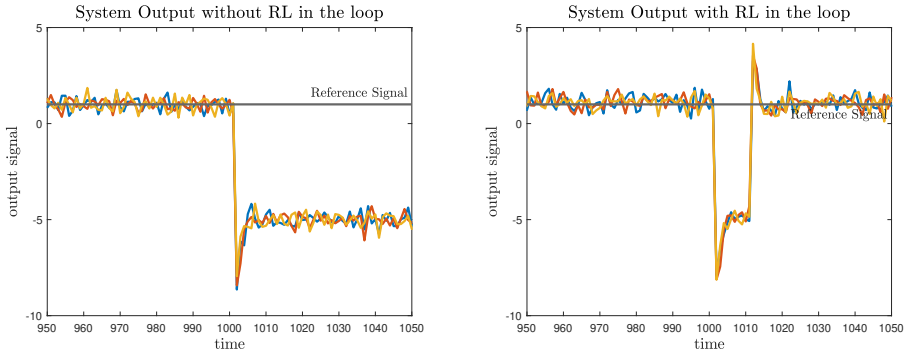

Fig. 4: Figures(a) (left) shows the timed behavior the controlled system (without RL module) when a fault of magnitude 6 enters the system at time $t = 1000$. Figure(b) (right) shows the behavior of the RL augmented system when a fault of magnitude 6 enters the system at time $t = 1000$.

statistical performance over a broad range of fault realization and their effect on the system with and without RL. We see that RL action successfully mitigates the effect of fault on the system state, allowing the low-level controller to stabilize system dynamics around the reference signal. To highlight the effectiveness of agent action, Figure 4 illustrates the effect of a constant fault on the system without and with the supervision of RL, on system dynamics. Here one single fault occurs at the 1000^{th} time-step. Next, we demonstrate the effectiveness of RL agent's training with exclusively constant faults. Following the remarks below Theorem 1, we illustrate that the agent is capable of handling cases of dynamics faults. Figure 5 illustrates the results of dynamic piece-wise constant faults that occur every 500 time-steps. We conclude the single sensor measurement case by testing the ability of the agent to respond to slowly varying periodic faults. Figure 6 illustrates the results. Interestingly enough, the agent is able to mitigate, to a large extent, the effect of a non piece-wise constant time-varying disturbance.

B. Multi-Sensor Measurements

Here we consider an over-observed system with $s = 3$ sensors with $C_k \equiv I_3$ and measurement noise $N_k^T N_k \equiv 0.01 I_3$ for $k = 1, 2, 3$. Due to space constraints, we limit this scenario to a mere illustration of Theorem 1. Figure 7 showcases the onset of fault on sensor 3. Figure 7(a) explains that the RL agent reaction to fault is the some of actions applied in all three sensors, exactly along the lines of space $\mathcal{M}$, defined in Theorem 1.

VII. Conclusion and Future Works

We explored a bi-level non-intrusive design to add the fault-tolerance capability to an existing controlled system, using an intelligent supervisory entity to take appropriate action to detect, identify, track and mitigate possible sensor faults while the controlled system is in operation. We approached the problem from the engineering perspective and presented model-based principles to design effective cost functions for the RL algorithm. Given a diagnostic task, controlled system dynamics and sensor fusion schemes can help us design cost functions with desired properties. We demonstrated the efficacy of such a design through a case-study on a chemical process with an underlying servo mechanism in the presence of sensor faults in single or multi-sensor configurations. Our contribution highlights the effectiveness of hybrid methods where Reinforcement Learning and data-driven methods are combined with model-based classic control theoretic approaches to yield improved adaptivity, robustness, and performance. We conjecture that our method will generalize to non-linear dynamics. We leave

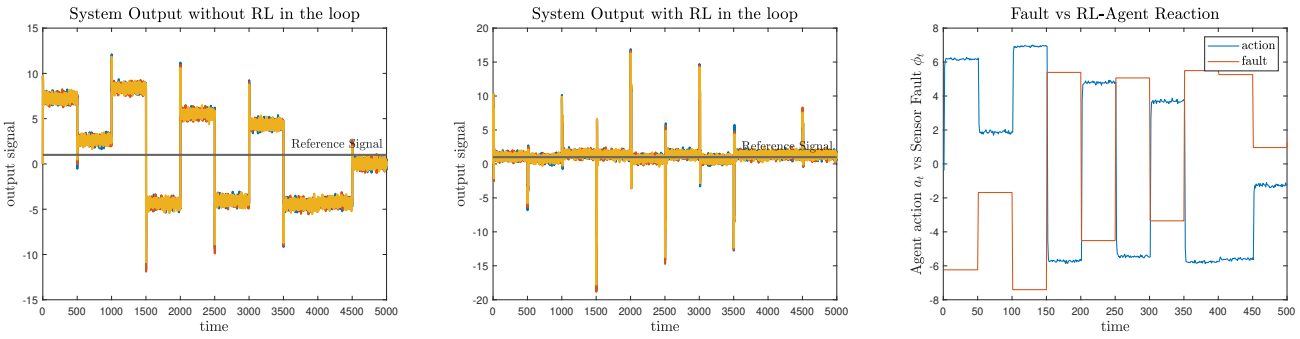

Fig. 5: Figures(a) (left) shows the timed behavior the controlled system (without RL module) when a piecewise fault with random magnitudes and jumps at every 50 time steps enters the system. Figure(b) (middle) shows the behavior of the RL augmented system when such fault is present. Figure(c) (right) shows the behaviour of the RL agent under such fault.

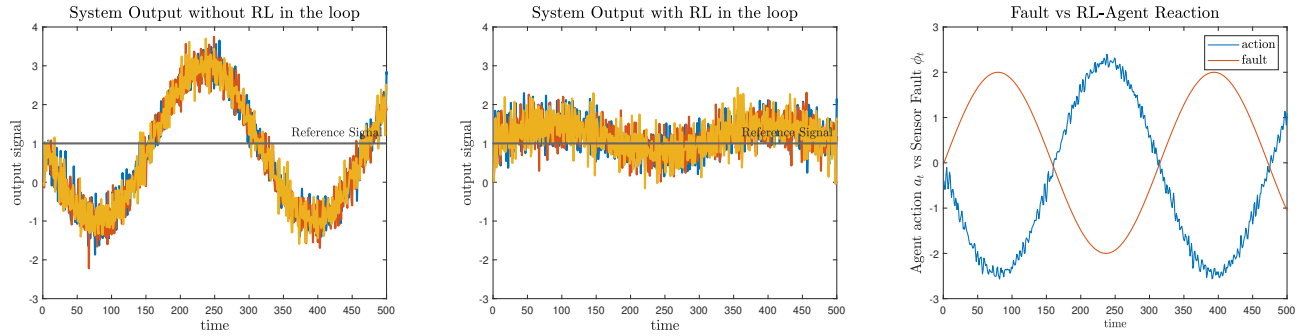

Fig. 6: Figures(a) (left) shows the timed behavior the controlled system (without RL module) when a sinusoidal fault with frequency of 0.01 rad and an amplitude of 2 enters the system. Figure(b) (middle) shows the behavior of the RL augmented system when such fault is present. Figure(c) (right) shows the behaviour of the RL agent under such fault. It is still not clear why agent is not able to completely cancel the oscillating effect of fault. It is speculated that remaining oscillation is due to slight lag in the action of behalf of agent.

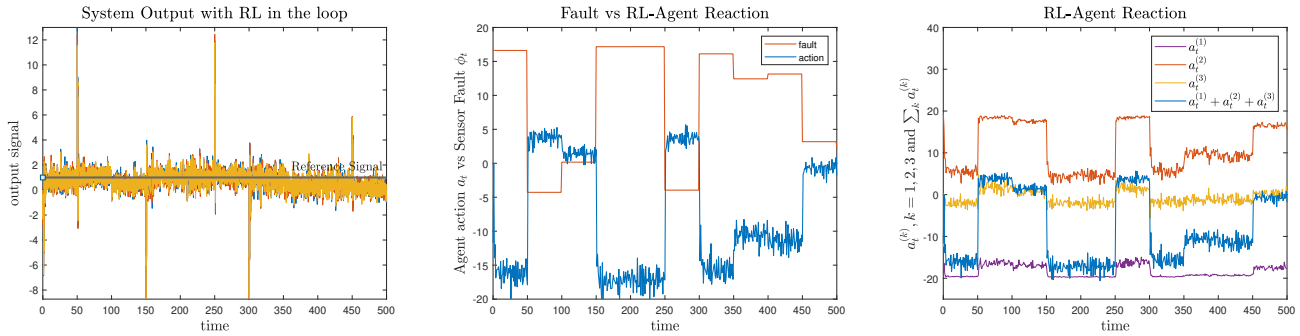

Fig. 7: Dynamic, piecewise-constant faults entering sensor 3, in an over-observed system. The actions of the RL-agent remove the fault in the average of the sensor measurements that is fed back to the controller. There are infinitely many sets of actions that would sum up to cancel the fault and the RL-agent would end up in a minima depending on the initial condition and the learning dynamics of the algorithm.

the validation of our design for such non-linear systems for future work. Lastly, we will also explore the space of possible unknown faults such as actuator or system disturbances of either additive or multiplicative nature.

APPENDIX

Proof: [of Lemma 1] A successful synthesis of an LQG-i controller for the low-level system, implies that matrix $\mathbf{A}$ is Hurwitz. Then, the asymptotic behavior of $\mathbb{E}[y_t]$ satisfies

$$\lim_{t\to+\infty} \mathbb{E}[y_t] = \lim_{t\to+\infty} \mathbf{C}\mathbf{W}_{t-1}\mathbb{D},$$

where we note that the limit of $\mathbf{W}_t$ exists. On the other hand, the asymptotic tracking of constant reference r implies

$$\lim_{t\to+\infty} \mathbb{E}[y_t] = C \lim_{t\to+\infty} \mathbb{E}[x_t] + \frac{1}{s}\sum_{k=1}^{s} \phi^{(k)}$$

$$= \mathbf{C} \lim_{t\to+\infty} \mathbf{W}_{t-1}\mathbb{D} + \frac{1}{s}\sum_{k=1}^{s} \phi^{(k)} = r$$

If we solve for $\mathbf{C}\lim_{t\to+\infty}\mathbf{W}_{t-1}\mathbb{D}$ we obtain

$$\begin{aligned}\mathbf{C}\lim_{t\to+\infty}\mathbf{W}_{t-1}\mathbb{D} &= r-\frac{1}{s}\sum_{k=1}^{s}\phi^{(k)}\\ &= \left[-\tfrac{1}{s}I:-\tfrac{1}{s}I:\cdots:-\tfrac{1}{s}I:I\right]\mathbb{D}.\end{aligned}$$

In view of $\mathbb{D}$ being constant but arbitrary vector, the proof is concluded. ■

Proof: [of Lemma 2] This is a result of straightforward algebra. The steps are omitted due to space limitations. ■

Proof: [of Theorem 1] We work as follows:

$$\begin{aligned}\mathbb{E}\left[\|C\,x_t-r\|^2\right] &= \mathbb{E}\left[\|\mathbf{C}\,\mathbb{X}_t-r\|^2\right]\\ &= \mathbb{E}\left[\|\mathbf{C}\mathbb{X}_t\|^2\right]-2r^T\mathbf{C}\,\mathbb{E}[\mathbb{X}_t]+\|r\|^2\end{aligned}$$

of matrix $\mathbf{C}=[C:O_{n_x}:O_{n_y}]$ that has the project property : $\mathbf{C}\mathbb{X}_t=x_t,\ \forall\ t$. Then

$$\mathbb{E}\left[\|Cx_t-r\|^2\right]=\mathbb{E}\left[\mathbb{X}_t^T\,\mathbf{C}^T\mathbf{C}\,\mathbb{X}_t\right]-2r^T\mathbf{C}\,\mathbb{E}[\mathbb{X}_t]+\|r\|^2$$

The term $\mathbb{E}\left[\mathbb{X}_t^T\mathbf{C}^T\mathbf{C}\mathbb{X}_t\right]$ is (13) with $\mathbf{M}=\mathbf{C}^T\mathbf{C}$. From Lemma 2 we have that

$$\mathbb{E}\left[\|C\,x_t-r\|^2\right]=\vartheta_t+\theta_t^T\mathbb{D}+\mathbb{D}^T\,\Theta_t\,\mathbb{D}-2r^T\mathbf{C}\,\mathbb{E}[\mathbb{X}_t]+\|r\|^2,$$

for ϑ_t,θ_t and Θ_t evaluated at $\mathbf{M}=\mathbf{C}^T\mathbf{C}$. Expanding $\mathbb{E}[\mathbb{X}_t]$ according to (11)

$$\begin{aligned}\mathbb{E}\left[\|C\,x_t-r\|^2\right] &= \vartheta_t-2r^T\mathbf{C}\mathbf{A}^t\mathbb{E}[\mathbb{X}_0]+\|r\|^2+\cdots\\ &\quad\cdots+\left(\theta_t^T-2r^T\mathbf{C}\mathbf{W}_{t-1}\right)\mathbb{D}+\mathbb{D}^T\Theta_t\mathbb{D}\end{aligned}$$

Since $\lim_{t\to+\infty}c_t=\lim_{t\to+\infty}\mathbb{E}\left[\|C\,x_t-r\|^2\right]$, we have

$$\begin{aligned}\lim_{t\to+\infty}\mathbb{E}\left[\|C\,x_t-r\|^2\right] &= \vartheta_\infty+\|r\|^2-2r^T\mathbf{C}\mathbf{W}_\infty\mathbb{D}+\ldots\\ &\quad\cdots+\mathbb{D}^T\,\Theta_\infty\,\mathbb{D}\end{aligned}$$

where $\mathbf{C}\mathbf{W}_\infty:=\mathbf{C}\lim_{t\to+\infty}\mathbf{W}_t$ and from Hurwitz property of $\mathbf{A}$, $\vartheta_\infty:=\lim_{t\to+\infty}\vartheta_t$ and $\Theta_\infty:=\lim_{t\to+\infty}\Theta_t$ exist while $\lim_{t\to+\infty}\theta_t=\mathbf{0}$. Furthermore, from Lemma 1

$$\begin{aligned}\mathbf{C}\mathbf{W}_\infty &= \left[-\tfrac{1}{s}I_{n_y}:\cdots:-\tfrac{1}{s}I_{n_y}:I_{n_y}\right]=:[W_{\varphi r}:I_{n_y}]\\ \Theta_\infty &= \left[\begin{array}{c:c}\frac{1}{s^2}\mathbb{J}_s\otimes I_{n_y} & -\frac{1}{s}\mathbf{1}_s\otimes I_{n_y}\\ \hdashline -\frac{1}{s}\mathbf{1}_s^T\otimes I_{n_y} & I_{n_y}\end{array}\right]\\ &=: \left[\begin{array}{c:c}W_{\varphi\varphi} & W_{\varphi r}\\ \hdashline W_{\varphi r}^T & I_{n_y}\end{array}\right]\end{aligned}$$

The sub-matrices $W_{\cdot,\cdot}$ act on sub-vectors of $\mathbb{D}=\begin{bmatrix}\varphi\\ r\end{bmatrix}$, $\varphi:=\phi+\alpha$, are subscripted in a self-explanatory manner. Expanding on φ and gathering all terms together we get

$$\begin{aligned}\lim_{t\to+\infty}\mathbb{E}\left[\|Cx_t-r\|^2\right] =&\vartheta_\infty+\|r\|^2-2r^TW_{\varphi r}\varphi-\cdots\\ &\cdots-2\|r\|^2\varphi^TW_{\varphi\varphi}\varphi+\cdots\\ &\cdots+2r^TW_{\varphi r}\phi+\|r\|^2\\ =&\vartheta_\infty+\varphi^T\,W_{\varphi\varphi}\,\varphi.\end{aligned}\tag{16}$$

Matrix $W_{\varphi\varphi}$ is positive semi-definite $sn_y\times sn_y$ matrix of rank 3. The 3 nonzero eigenvalues are identical and equal to $2/s^2$. Combine (16) with the form of $W_{\varphi\varphi}$ to conclude for the form of $\mathcal{M}$. ■

REFERENCES

[1] R. Patton, P. Frank, and R. Clark, *Issues of Fault Diagnosis for Dynamic Systems*. Springer London, 2000.

[2] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell, "A survey of physics-based attack detection in cyber-physical systems," *ACM Comput. Surv.*, vol. 51, no. 4, Jul. 2018.

[3] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[4] C. Li, H. Li, Y. Chen, H. Dong, X. Zhao, and L. Xiao, "Model-based sensor fault detection and isolation method for a vehicle dynamics control system," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 231, no. 2, pp. 147–160, 2017.

[5] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 3, pp. 1–39, 2010.

[6] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A Systems and Control Perspective of CPS Security," *Annual Reviews in Control*, vol. 47, pp. 394–411, 2019.

[7] D. Nicol, W. Sanders, and K. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 48–65, 2004.

[8] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 313–326, 2003.

[9] G. J. Vachtsevanos and G. J. Vachtsevanos, *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley Online Library, 2006, vol. 456.

[10] J. Selkäinaho and A. Halme, "An intelligent fault tolerant control system," *IFAC Proceedings Volumes*, vol. 20, no. 5, Part 3, pp. 69–73, 1987, 10th Triennial IFAC Congress on Automatic Control - 1987 Volume III, Munich, Germany, 27-31 July.

[11] Y. Diao and K. M. Passino, "Intelligent fault-tolerant control using adaptive and learning methods," *Control engineering practice*, vol. 10, no. 8, pp. 801–817, 2002.

[12] Y. Sohège, G. Provan, M. Quinones-Grueiro, and G. Biswas, "Deep reinforcement learning and randomized blending for control under novel disturbances," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8175–8180, 2020.

[13] A. Meystel, "Intelligent control," in *Encyclopedia of Physical Science and Technology (Third Edition)*, third edition ed., R. A., Ed. New York: Academic Press, 2003, pp. 1–24.

[14] E. Balaban, A. Saxena, P. Bansal, K. F. Goebel, and S. Curran, "Modeling, detection, and disambiguation of sensor faults for aerospace applications," *IEEE Sensors Journal*, vol. 9, no. 12, pp. 1907–1917, 2009.

[15] D. Adamy, *EW 102: A Second Course in Electronic Warfare*, ser. Artech House radar library. Artech House, 2004.

[16] P. C. Young and J. Willems, "An approach to the linear multivariable servomechanism problem," *International journal of control*, vol. 15, no. 5, pp. 961–979, 1972.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[18] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509.02971, 2016.

[19] J. Milošević, H. Sandberg, and K. H. Johansson, "Estimating the impact of cyber-attack strategies for stochastic networked control systems," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 747–757, 2019.