IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Data-Driven Anomaly Detection in Robots using Matrix Chernoff Bounds

Richa Dubey, Niladri S. Tripathy, *Member, IEEE*, and Suril V. Shah, *Member, IEEE*

***Abstract*—This work proposes a novel data-driven anomaly detection framework for robotic systems, grounded in statistical concentration inequalities. The method leverages the Matrix Chernoff Inequality to establish probabilistic bounds on the eigenvalues of cumulative error covariance matrices computed over a sliding window of robot state deviations. An anomaly is flagged when the eigenvalues, computed in real time, violate these theoretical bounds. The proposed approach is model-independent, computationally efficient, and straightforward to implement, requiring only the numerical solution of two transcendental equations to determine the bounds. It further offers design flexibility via tunable parameters such as the confidence level and window size. The effectiveness of the detector is validated through both simulation and hardware experiments across distinct anomaly scenarios for different robots, including input delay, sensor corruption, and external perturbations. A comprehensive performance evaluation is also presented using standard metrics such as Detection Rate, False Positive Rate, Accuracy, and Receiver Operating Characteristics (ROC), along with a method for effective parameter selection and comparison with existing works.**

***Index Terms*—Anomaly detection, covariance matrix, matrix chernoff inequality.**

I. Introduction

ROBOTS operating in dynamic environments are susceptible to sensor failures, external disturbances, software bugs, and other irregularities collectively known as anomalies. These events can severely compromise performance, reliability, and safety [1], [2], [3]. As robots are readily being used across critical domains such as healthcare, manufacturing, and terrain exploration, robust anomaly detection mechanisms are essential to ensure dependable robot operation [4], [5]. Early and accurate detection of anomalies not only prevents system failures but also enables timely corrective actions, thus supporting safer and effective robot deployment in real-world settings [6].

Traditional anomaly detection methods often rely on empirically or heuristically selected thresholds [7], [8]. While convenient, these thresholds are typically scenario-specific and lack a rigorous statistical or theoretical foundation. Moreover, thresholds chosen by trial-and-error or intuition lead to non-reproducible outcomes, hindering scalability and standardization in real-world applications [9]. Some other methods for anomaly detection in robots rely on accurate system models for state estimation [2], [10]. However, an accurate system model can be challenging to obtain and maintain at all times, particularly in real-world environments where uncertainties and nonlinearities are prevalent. Alternatively, learning based strategies, such as DL models, have been explored to perform anomaly detection [11], [12]. However, they often introduce substantial computational overhead during offline training [2]. This makes it challenging to deploy them on resource-constrained platforms such as mobile robots, where accuracy and computational efficiency are critical considerations.

On the other hand, many statistical anomaly detection methods in robotics rely on classical multivariate techniques such as Mahalanobis distance based scoring [3], covariance-deviation tests [13], or PCA/SPC-based monitoring methods and residual statistics [14]. These approaches typically assume Gaussianity and a stationary covariance structure. As a result, their detection thresholds can become unreliable under distribution shift and changing operating conditions. Additionally, recent strategies incorporate model-based statistical filtering pipelines [15], [16]. Such methods, however, depend on accurate system models and tuned thresholds. These limitations motivate the need for a model-independent statistical alternative that does not warrant an assumption on the associated distributions and yields theoretically grounded thresholds.

In such a scenario, a novel data-driven method is proposed in this work, which employs statistical tools to derive reliable thresholds on the nominal behavior of robots. Specifically, the Matrix Chernoff Inequality [17] is used to derive probabilistic bounds on the eigenvalues of sum of covariance matrices, which characterize the error in robot states relative to the desired state. To the best of our knowledge, Matrix Chernoff Inequality has not yet been explored in the context of robotic systems. As opposed to complex learning based approaches, the proposed detector is simple to implement since it requires solving only two transcendental equations for obtaining the bounds, and operates on the well-known covariance matrices and their eigenvalues, without any dependency on system model. Additionally, since the bounds are derived from well-established statistical theory, the method is both theoretically grounded and scalable. Furthermore, its tunable parameters, such as confidence level and window size, offer design flexibility to adapt to diverse application requirements. The main contributions of this work are:

- Probabilistic bounds on nominal behavior are derived using the Matrix Chernoff Inequality and used to identify anomalies when deviations exceed these statistically grounded thresholds.

Received 16 September 2025; accepted 30 December 2025. Date of publication 22 January 2026; date of current version 2 February 2026. This article was recommended for publication by Associate Editor Yongbo Chen and Editor Lucia Pallottino upon evaluation of the reviewers' comments. *(Corresponding author: Richa Dubey.)*

Richa Dubey and Niladri S. Tripathy are with the Department of Electrical Engineering, Indian Institute of Technology, Jodhpur, Jodhpur 342037, India (e-mail: richa.1@iitj.ac.in; niladri@iitj.ac.in).

Suril V. Shah is with the Department of Mechanical Engineering, Indian Institute of Technology, Jodhpur, Jodhpur 342037, India (e-mail: surilshah@iitj.ac.in).

Digital Object Identifier 10.1109/LRA.2026.3656792

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

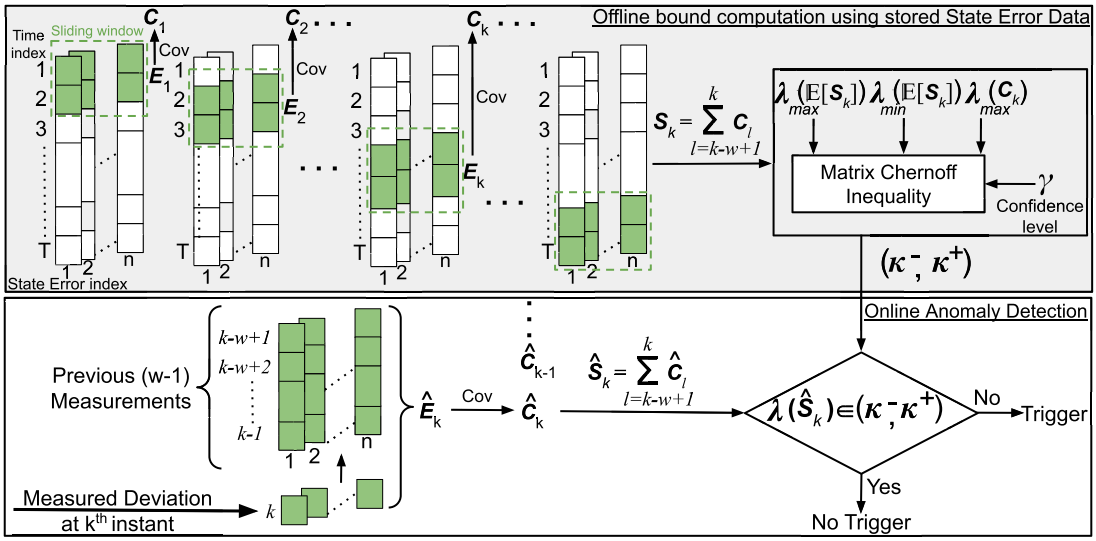


Fig. 1. Architecture of the proposed anomaly detector.

- The effectiveness of the proposed anomaly detector is demonstrated through simulations and hardware experiments, showcasing its ability to identify distinct anomalies in robotic systems.
- An in-depth performance analysis of the proposed detector is presented in terms of standard evaluation metrics, including comparison with existing works.

The rest of this work is organized as follows. Sec. II presents the problem statement and proposed solution. Details of the proposed solution are presented in Sec. III, followed by its exemplar applications in Sec. IV. Sec. V presents the results with an in-depth performance analysis of the proposed detector, and Sec. VI concludes this work.

II. Problem Description and Proposed Solution

Anomalies in robots typically appear as deviations from their desired states, termed as state errors. Covariance matrices can effectively capture these errors by representing both their magnitudes and inter-dependencies, thereby distinguishing between deviations caused by anomalies (appearing as specific correlations) and random noise (appearing as scattered, uncorrelated variations). They are symmetric, positive semi-definite and can be eigen-decomposed, making them useful in a wide range of applications [18]. In this work, we leverage them to develop a robust and versatile anomaly detector for robots. The problem addressed in this work is summarized below.

Problem Statement 1: Designing a Covariance matrix-based detector for robots that enables accurate detection of anomalies, maintains robustness to noise, is easy to implement, and eliminates the need for heuristically tuned parameters or system-specific models.

We next discuss the overview of the proposed solution as illustrated in Fig. 1. The framework operates in two phases, namely offline bound computation and online anomaly detection. During the offline phase (gray block in Fig. 1), stored state deviations from nominal runs are structured into error matrices $\boldsymbol{E}_k$ using a sliding window (green shaded blocks). These are used to compute covariance matrices $\boldsymbol{C}_k$, which are then summed over the past w steps to obtain a cumulative error covariance matrix $\boldsymbol{S}_k$ to track state error evolution over time. Since directly comparing matrices can be complex and may require heuristic metrics, we use the eigenvalues $\lambda(\boldsymbol{S}_k)$ instead, which are intuitive, compact, and invariant to orthogonal transformations. Precisely, Matrix Chernoff Inequality [17] is employed to derive bounds (κ^-, κ^+) such that $\lambda(\boldsymbol{S}_k) \in (\kappa^-, \kappa^+)$. During real-time operation (lower block of Fig. 1), the cumulative error covariance matrix $\hat{\boldsymbol{S}}_k$ is computed from live measurements, and its eigenvalues $\lambda(\hat{\boldsymbol{S}}_k)$ are evaluated against the precomputed bounds. If they lie outside (κ^-, κ^+), an anomaly is flagged. Further details of the proposed anomaly detector are presented in the next section.

III. Data-Driven Anomaly Detector

In this section, we present the construction of the cumulative error covariance matrix $\boldsymbol{S}_k$ via the error matrix $\boldsymbol{E}_k$ and the covariance matrix $\boldsymbol{C}_k$ using the stored state error data of the robot offline. Subsequently, probabilistic bounds are derived on $\lambda(\boldsymbol{S}_k)$ to detect anomalous behavior in real-time.

A. Constructing Cumulative Error Covariance Matrix $\boldsymbol{S}_k$

Let the error in i^{th} state of the robot be defined as

$$e_i[k] = x_i^d[k] - x_i[k], \tag{1}$$

where $i \in \{1, 2, \ldots, n\}$ and n is the total number of states, $x_i^d[k]$ is the desired state, and $x_i[k]$ is the measured state at k^{th} instant. Rather than using the entire past dataset, we construct a stacked error matrix over a sliding time window w as

$$\boldsymbol{E}_k = \begin{bmatrix} e_1[k] & e_2[k] & \cdots & e_n[k] \\ e_1[k-1] & e_2[k-1] & \cdots & e_n[k-1] \\ \vdots & \vdots & \ddots & \vdots \\ e_1[k-w+1] & e_2[k-w+1] & \cdots & e_n[k-w+1] \end{bmatrix}, \tag{2}$$

where each column captures the sequence of error over past w time-instants for the corresponding state of the robot, and each row captures the errors across all the states at a particular

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

time instant within the window, yielding a $w \times n$ matrix. During detection, this enables utilizing the most recent w data points for decision-making. Now, we construct $\boldsymbol{S}_k$ using $\boldsymbol{E}_k$.

Let the mean error vector over the window w be $\bar{\mathbf{e}}_k = \frac{1}{w}\mathbf{1}_w^\top \boldsymbol{E}_k$, where $\mathbf{1}_w$ is a $w \times 1$ vector of ones. Then, the $n \times n$ error covariance matrix $\boldsymbol{C}_k$ is given by

$$\boldsymbol{C}_k = \frac{1}{w-1}\left(\boldsymbol{E}_k - \mathbf{1}_w \bar{\mathbf{e}}_k\right)^\top \left(\boldsymbol{E}_k - \mathbf{1}_w \bar{\mathbf{e}}_k\right), \tag{3}$$

where $\boldsymbol{E}_k$ is substituted from (2). While $\boldsymbol{C}_k$ can be directly used as a metric for anomaly detection, it may not capture the effect of all anomalies. This is because certain anomalies in robots cause small, gradually emerging deviations in their states, which are difficult to capture using a single instance of $\boldsymbol{C}_k$ alone. To address this, it can be beneficial to aggregate covariances over a broader temporal range. Therefore, we define the cumulative error covariance matrix $\boldsymbol{S}_k$ as

$$\boldsymbol{S}_k = \sum_{l=k-w+1}^{k} \boldsymbol{C}_l. \tag{4}$$

Then, the bounds (κ^-, κ^+) are derived on the eigenvalues $\lambda(\boldsymbol{S}_k)$ using the stored nominal data. However, due to the finite size of data available and its inherent randomness caused by sensor noise and dynamic environment, it is impossible to avoid false triggers completely. We capture this behavior by defining a confidence level $\gamma \in (0, 1)$ as

$$\mathbb{P}\left[\lambda\left(\boldsymbol{S}_k\right) \notin \left(\kappa^-, \kappa^+\right)\right] \leq \gamma, \tag{5}$$

representing the tail probability of false triggers. Next, we present the computation of bounds on eigenvalues of $\boldsymbol{S}_k$.

B. Computing Bounds (κ^-, κ^+) on Eigenvalues of $\boldsymbol{S}_k$

Concentration inequalities can provide solid statistical foundations for bounding the probability of deviations from expected behavior. Since our framework relies on sums of sliding-window covariance matrices (4), we require concentration results that extend to matrix random variables, namely matrix concentration inequalities [17]. Among them, the Matrix Chernoff Inequality is chosen because it directly applies to sums of positive semidefinite matrices, yields sharp exponential tail bounds on eigenvalues, and has simple, verifiable conditions [17]. The following theorem states the proposed method of bound computation using the Matrix Chernoff Inequality.

Theorem 1: Let $\boldsymbol{S}_k$ (4) denote the cumulative error covariance matrix for the robot at k^{th} instant, constructed by summing a sequence of w consecutive $n \times n$ error covariance matrices $\boldsymbol{C}_k$ (3). Assuming that the maximum eigenvalue of each $\boldsymbol{C}_k$ is almost surely bounded as $\lambda_{\max}(\boldsymbol{C}_k) \leq R$, we define the minimum and maximum eigenvalues of expectation of $\boldsymbol{S}_k$ as

$$\mu_{\min} := \lambda_{\min}\left(\mathbb{E}[\boldsymbol{S}_k]\right) \text{ and } \mu_{\max} := \lambda_{\max}\left(\mathbb{E}[\boldsymbol{S}_k]\right). \tag{6}$$

Then, the lower and upper bounds on $\lambda(\boldsymbol{S}_k)$ are $\kappa^- := (1-\delta')\mu_{\min}$ and $\kappa^+ := (1+\delta'')\mu_{\max}$, where δ' and δ'' are the solutions of

$$\delta + (1-\delta)\ln(1-\delta) + \frac{R}{\mu_{\min}}\ln\left(\frac{\gamma}{2n}\right) = 0, \quad \text{for } \delta \in [0, 1], \tag{7}$$

$$\delta - (1+\delta)\ln(1+\delta) - \frac{R}{\mu_{\max}}\ln\left(\frac{\gamma}{2n}\right) = 0, \text{ for } \delta \geq 0, \tag{8}$$

respectively, for a confidence level γ (5).

proof 1: From the lower Matrix Chernoff Inequality [17], it can be deduced that for $\delta \in [0, 1]$,

$$\mathbb{P}\left\{\lambda_{\min}\left(\boldsymbol{S}_k\right) \leq (1-\delta)\mu_{\min}\right\} \leq n \cdot \left[\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right]^{\frac{\mu_{\min}}{R}}, \tag{9}$$

and the upper Matrix Chernoff Inequality [17] gives for $\delta \geq 0$

$$\mathbb{P}\left\{\lambda_{\max}(\boldsymbol{S}_k) \geq (1+\delta)\mu_{\max}\right\} \leq n \cdot \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{\frac{\mu_{\max}}{R}}. \tag{10}$$

Then, we define the bounds $\kappa^- := (1-\delta)\mu_{\min}$ and $\kappa^+ := (1+\delta)\mu_{\max}$ for (9) and (10), respectively. Now, assuming that γ (5) is distributed symmetrically on either side of the interval (κ^-, κ^+), we get

$$\mathbb{P}\left[\lambda_{\min}\left(\boldsymbol{S}_k\right) \leq \kappa^-\right] \leq \frac{\gamma}{2} \text{ and } \mathbb{P}\left[\lambda_{\max}\left(\boldsymbol{S}_k\right) \geq \kappa^+\right] \leq \frac{\gamma}{2}. \tag{11}$$

Next, we first derive κ^-, followed by κ^+.

Bound on Minimum Eigenvalue of $\boldsymbol{S}_k$ ($\lambda_{\min}(\boldsymbol{S}_k)$): Tight bounds can be derived by setting the LHS of (9), (11) equal to their respective RHS. Subsequently, substituting (11) in the LHS of (9) yields $\frac{\gamma}{2} = n \cdot \left[\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right]^{\frac{\mu_{\min}}{R}}$. Upon rearranging and taking the natural logarithm of both sides, the transcendental equation shown in (7) is obtained. Let δ' be the solution to (7). Then, $\kappa^- = (1-\delta')\mu_{\min}$.

Bound on Maximum Eigenvalue of $\boldsymbol{S}_k$ ($\lambda_{\max}(\boldsymbol{S}_k)$): Similarly, setting the LHS of (11) equal to RHS, and substituting it in the LHS of (10) yields $\frac{\gamma}{2} = n \cdot \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{\frac{\mu_{\max}}{R}}$. Upon rearranging and taking the natural logarithm of both sides, the transcendental equation shown in (8) is obtained. Let δ'' be the solution to (8). Then, $\kappa^+ = (1+\delta'')\mu_{\max}$.

We now discuss how the derived bounds enable a clear distinction between nominal and anomalous behavior. Since each eigenvalue of $\boldsymbol{S}_k$ quantifies the extent of variance of the robot's state errors along a particular principal direction, the interval (κ^-, κ^+) can be viewed as the statistically anticipated range of this variance during nominal operation under noisy sensors. If $\lambda_{\min}(\boldsymbol{S}_k)$ falls below the lower limit of nominal variability κ^-, it indicates an unusually small variance in the error dynamics, which may arise from an anomaly that suppresses the natural variations in measured states. Conversely, if $\lambda_{\max}(\boldsymbol{S}_k)$ exceeds the upper limit of nominal variability κ^+, it indicates an abnormally large growth in the cumulative deviation of state errors, and hence, the presence of an anomaly. This interpretation allows the bounds (κ^-, κ^+) to be viewed not only as abstract limits derived from the Matrix Chernoff Inequality, but also as probabilistic bounds distinguishing normal from abnormal robot behavior.

Remark 1: For using the Matrix Chernoff inequalities to derive bounds on $\lambda(\boldsymbol{S}_k)$ (4), $\boldsymbol{C}_l, l = k-w+1$ to $l = k$, should be a finite sequence of independent, random, self-adjoint matrices [17]. While $\boldsymbol{C}_l$ satisfies the self-adjoint property by definition (3), they are not strictly independent due to overlapping of sliding windows. However, the inequalities can still be employed because the temporal dependencies introduced by the overlapping windows are typically weak and do not significantly impact the concentration behavior of $\lambda(\boldsymbol{S}_k)$ (4), rendering the bounds informative and useful [19].

Remark 2: The confidence level γ denotes the likelihood of identifying an observation as an outlier, where a larger γ

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1: Offline Bound Computation.

Initialize: No. of states n, window w, confidence level γ
Input: Stored robot state data of length T
for $k = w$ to T **do**
  for $i = 1$ to n **do**
    $e_i[k] \leftarrow x_i^d[k] - x_i[k]$ using (1)
  end for
  for $j = 0$ to $w-1$ **do**
    for $i = 1$ to n **do**
      $\boldsymbol{E}_k[j+1, i] \leftarrow e_i[k-j]$ using (2)
    end for
  end for
  $\boldsymbol{C}_k \leftarrow \texttt{CovarianceMatrix}(\boldsymbol{E}_k)$ using (3)
end for
for $k = 2w-1$ to T **do**
  for $l = k-w+1$ to k **do**
    $\boldsymbol{S}_k \leftarrow \boldsymbol{S}_k + \boldsymbol{C}_l$
  end for
end for
$\mu_{\max} \leftarrow \lambda_{\max}(\mathbb{E}[\boldsymbol{S}_k])$
Solve (8) for δ to get $\kappa^+ \leftarrow (1+\delta)\mu_{\max}$
Compute κ^- analogously using (8)

Algorithm 2: Online Anomaly Detection:

Initialize: Bounds (κ^-, κ^+) from Algorithm 1
Input: Real-time measurements $\hat{x}_i[k]$
for $k = 1$ to **end do**
  for $i = 1$ to n **do**
    $\hat{e}_i[k] \leftarrow x_i^d[k] - \hat{x}_i[k]$ using (1) for measurement $\hat{x}_i$
  end for
  Follow steps 5-15 of Algorithm 1 to obtain $\hat{\boldsymbol{S}}_k$
  if $\lambda(\hat{\boldsymbol{S}}_k) \notin (\kappa^-, \kappa^+)$ **then**
    Anomaly detected and trigger generated
  else
    No anomaly
  end if
end for

increases the risk of false triggers. Its value, generally within the range $\gamma \in [0.01, 0.001]$ [20], is at the designer's discretion.

C. Real-Time Anomaly Detection Using the Bounds (κ^-, κ^+)

Algorithm 1 summarizes the proposed method of bound computation (Sec. III-B), and Algorithm 2 summarizes the steps for anomaly detection in real-time. Here, trigger value $\lambda(\hat{\boldsymbol{S}}_k)$ is computed using real-time measurements, and a practical guideline for choosing window length w and confidence level γ is presented in Sec. V-B3. Next section demonstrates exemplary applications of the proposed detector.

IV. Some Applications of the Proposed Detector

Robots deployed in real-world environments often encounter practical limitations such as communication latencies, hardware malfunctions, and dynamic obstacles. Among the most common anomalies introduced by these limitations are input delay, sensor data corruption, and external perturbations. This section demonstrates the application of the proposed detector in detecting these anomalies.

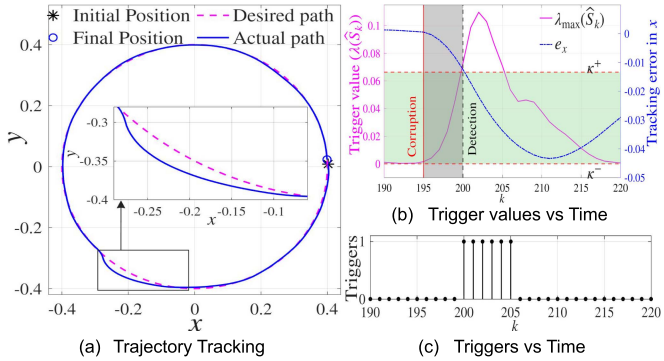


Fig. 2. Sensor anomaly detection in a robot.

A. Sensor Corruption Detection

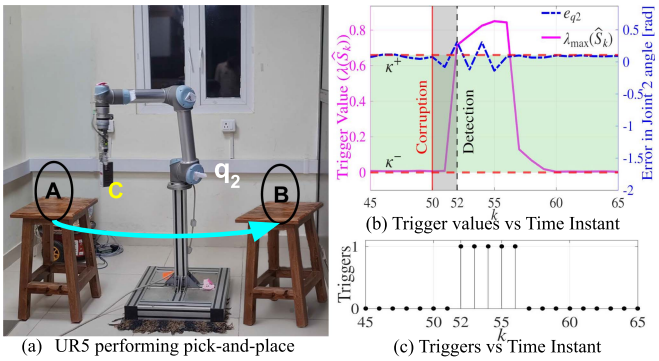
Sensor measurements are the primary source of state feedback in robotic systems, and their integrity is critical for accurate perception, decision-making, and control. However, various factors such as hardware faults, or environmental interference can corrupt sensor readings, causing controllers to act on misleading information. Timely and reliable detection of such corruption can enable deployment of corrective actions before the performance degrades. With this motivation, the following application focuses on detecting anomalous behavior caused by corrupted sensors in simulations and hardware experiment.

Simulation: We first consider a differential-drive robot with unicycle model and a feedback-linearization tracking controller [21], to generate nominal data in simulations. Linear and angular body velocities $[v, \omega]^\top$ are mapped to right/left wheel speeds via the wheel-body transform $\boldsymbol{M} = \begin{bmatrix} r/2 & r/2 \\ r/d & -r/d \end{bmatrix}$, where wheel radius $r = 0.02$m and wheelbase $d = 0.05$m. Given the sampling time $h = 0.1$s, the discrete-time model $\boldsymbol{x}[k+1] = \boldsymbol{A}_d\,\boldsymbol{x}[k] + \boldsymbol{B}_d\,\boldsymbol{u}[k]$ is obtained from a forward-Euler linearization of the unicycle dynamics about (θ, v). Here, state vector $\boldsymbol{x} = [x, y, \theta]^\intercal$ comprises of x, y coordinates and heading angle θ of the robot, input vector $\boldsymbol{u} = [v, \omega]^\intercal$, $\boldsymbol{A}_d = \begin{bmatrix} 1 & 0 & -h\,v\sin\theta \\ 0 & 1 & h\,v\cos\theta \\ 0 & 0 & 1 \end{bmatrix}$, and $\boldsymbol{B}_d = \begin{bmatrix} h\cos\theta & 0 \\ h\sin\theta & 0 \\ 0 & h \end{bmatrix}$. Wheel-speed saturation is enforced as $|\omega_{r,l}| \leq 10$ rad/s. To track the reference signals (x_d, y_d, θ_d), the PD controller gains are $k_p = 1$ and $k_d = 0.7$. Zero-mean Gaussian noise is injected into the sensor readings with variance 0.01 each, to simulate noisy sensors. This data is then used to compute the bounds for anomaly detection (κ^-, κ^+) offline via Algorithm 1, with window length $w = 5$ and confidence level $\gamma = 0.01$.

Next, real-time anomaly detection is validated via Algorithm 2 by introducing corruption in sensor measurements while the robot tracks a circular reference path, as shown in Fig. 2 a. Here, Fig. 2 b focuses on a specific interval of this trajectory tracking, where $k = 190 - 194$ depicts nominal behavior during anomaly-free operation with $\lambda(\hat{\boldsymbol{S}}_k) \in (\kappa^-, \kappa^+)$. At $k = 195$, an additive bias of magnitude 0.1 is manually injected into the x coordinate measurements to simulate sensor corruption.

This is illustrated in Fig. 2 b, where the blue dash-dotted line depicts tracking error in x (denoted by e_x in Fig. 2 b), and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) UR5 performing pick-and-place (b) Trigger values vs Time Instant (c) Triggers vs Time Instant

Fig. 3. Sensor anomaly detection on a UR5 manipulator arm during pick-and-place operation.

can also be observed from the zoomed-in plot within Fig. 2 a, where the solid blue line shows the actual path deviating from the desired dashed-magenta line after sensor corruption. Fig. 2 b and 2 c illustrate the corresponding anomaly detection process, where $\lambda_{\max}(\hat{\boldsymbol{S}}_k)$ exceeds κ^+ at $k = 200$ (dashed black vertical line), successfully triggering the presence of sensor corruption after a short latency of 5 time steps (gray-shaded region between corruption start and detection instants). Fig. 2 c shows the binary trigger signal generated by the proposed detector, which attains value 1 to indicate the presence of sensor corruption, and is 0 otherwise. Consequently, the proposed detector can be employed as a triggering mechanism that decides when to deploy mitigation strategies against sensor corruption, for e.g., mitigation methods based on state estimation [22], [23]. In the following, an experimental validation of the proposed detector is presented.

Experiment: In practical settings, robots such as 6-DoF manipulator arm are widely deployed for tasks including assembly, packaging, pick-and-place, and human-robot collaborative operations. In such scenarios, anomalies like joint-sensor corruption can introduce small joint-space deviations that propagate into significant end-effector errors, adversely affecting grasping accuracy and manipulation precision. To assess the detector under these realistic conditions, this subsection evaluates the proposed method through hardware experiments on UR5 manipulator, as described below.

The manipulator was operated via the RTDE interface using a sampling time of $h = 0.02$ s to perform a pick-and-place operation, carrying object C from point A $(0.475,\ 0.109,\ 0.609)$ to point B $(0.025,\ -0.486,\ 0.609)$, as shown in Fig. 3 a. Joint positions were recorded from the robot's internal controller and subsequently passed to the detector. Using nominal (anomaly-free) waypoint-tracking data collected offline, the bounds (κ^-, κ^+) were computed via Algorithm 1 with window length $w = 3$, confidence level $\gamma = 0.01$, and average and maximum computation times were 0.82 s and 0.91 s, respectively.

During online operation, sensor corruption was injected into Joint-2 (q_2), as shown in Fig. 3 b (blue dash-dotted line), in the form of a deviation having variable bias upper bounded by ± 0.2 rad. The corruption begins at $k = 50$ (red solid vertical line) and persists for 5 consecutive samples. Due to the serial kinematic structure of the manipulator, this joint-space corruption is amplified into task-space deviation according to the forward kinematics [24] and can result in an end-effector position deviation of the order 0.07 m. The presence of this anomaly is successfully detected after a short latency of 2 time steps (gray-shaded region between corruption start and detection), when $\lambda_{\max}(\hat{\boldsymbol{S}}_k)$ (solid magenta line) exits the nominal region (κ^-, κ^+) (green shaded region) at $k = 52$ (dashed black vertical line). Fig. 3 c illustrates the corresponding trigger signal, which is 1 when sensor corruption is detected, and is 0 otherwise.

B. Delay Detection

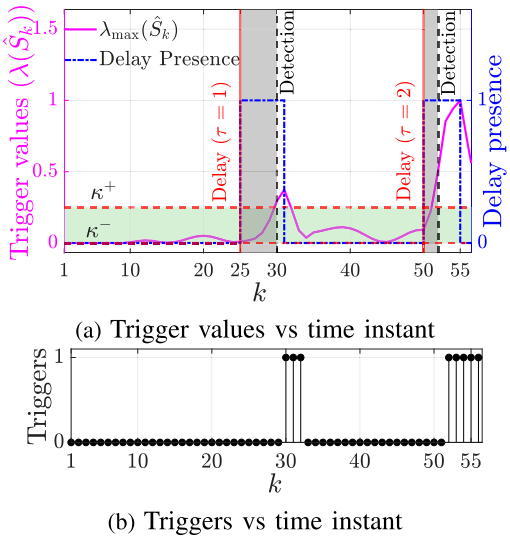
Input delays are prevalent in robots due to factors such as dynamics of low-level actuators and transmission delays when the controller and robot are not co-located [25]. Even small delays can degrade performance, amplify oscillations and trigger instability. Timely and reliable detection of delay can enable the controller to initiate appropriate mitigation and stabilization methods before performance deteriorates. With this motivation, we focus on delay detection in this application.

A one-link robot [24] is considered with moment of inertia $J = 1\,\mathrm{kg\,m^2}$, friction coefficient $b = 0.05\,\mathrm{N\,m\,s}$, link mass $m = 1\,\mathrm{kg}$, gravitational acceleration $g = 9.81\,\mathrm{m/s^2}$, and link length $l = 0.5\,\mathrm{m}$. Using a sampling time $h = 0.05\,s$, the linearized discrete model $\boldsymbol{x}[k+1] = \boldsymbol{A}_d\,\boldsymbol{x}[k] + \boldsymbol{B}_d\,\boldsymbol{u}[k]$ is obtained. Here, state vector $\boldsymbol{x} = [\theta, \dot{\theta}]^\top$ comprises of the rotation angle θ and angular velocity $\dot{\theta}$, $\boldsymbol{A}_d = \boldsymbol{I} + h\boldsymbol{A}$, and $\boldsymbol{B}_d = h\boldsymbol{B}$, where $\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -mgl/J & -b/J \end{bmatrix}$ and $\boldsymbol{B} = \begin{bmatrix} 0 \\ 1/J \end{bmatrix}$. The joint tracks a sinusoidal reference $\theta_d[k] = A_{\text{ref}} \sin(\omega_{\text{ref}} kh)$ with $A_{\text{ref}} = 0.25\,\text{rad}$ and $\omega_{\text{ref}} = 2\pi\ \text{rad/s}$. A discrete LQR controller regulates the tracking error using $\boldsymbol{Q} = \mathrm{diag}(40, 6)$ and $\boldsymbol{R} = 0.2$. To simulate noisy sensors, zero-mean Gaussian noise with standard deviation 0.01 is injected to the sensor readings. Using this tracking data, the bounds for anomaly detection (κ^-, κ^+) are computed offline via Algorithm 1, with window length $w = 5$ and confidence level $\gamma = 0.01$. Next, a τ-step delay is introduced in the input of the robot such that $\boldsymbol{x}[k+1] = \boldsymbol{A}_d\,\boldsymbol{x}[k] + \boldsymbol{B}_d\,\boldsymbol{u}[k-\tau]$. Real-time anomaly detection is then validated using Algorithm 2 when $\tau = 1$ for $k = 25 - 31$ and $\tau = 2$ for $k = 50 - 55$. This is illustrated via blue dash-dotted line in Fig. 4 a, which indicates the presence or absence of input delay. Fig. 4 b shows the trigger value (solid magenta line) along with the nominal region (κ^-, κ^+) (green region within dashed red lines). Fig. 4 b marks trigger instants. When 1-step input delay starts at $k = 25$ (solid red vertical line), $\lambda(\hat{\boldsymbol{S}}_k)$ exits the nominal region at $k = 30$ (dashed black vertical line), and the detector successfully triggers the presence of this delay. The vertical gray shaded region indicates the detection latency. Later, a 2-step delay is introduced similarly at $k = 50$, which is successfully detected after a short latency of 2 steps at $k = 52$ (dashed black vertical line). The shorter detection latency for 2-step delay arises because higher delays cause larger deviations of the robot from its desired state (1), pushing $\lambda(\hat{\boldsymbol{S}}_k)$ more rapidly beyond the nominal bounds (κ^-, κ^+). This behavior of the proposed detector is beneficial, as faster detection of higher delays is desirable. Once the presence of delay is established, its value τ can be estimated using existing approaches [26], [27], and the system can be stabilized through state-of-the-art techniques [28].

C. External Perturbation Detection

Robots often operate in dynamic and unpredictable environments, where they may experience external perturbations, such

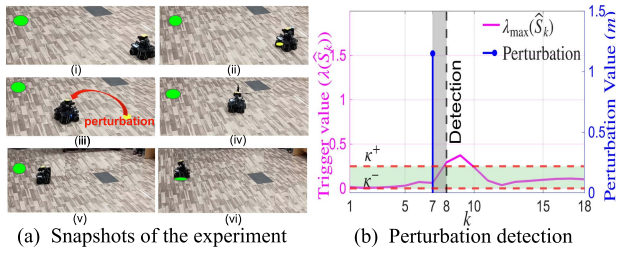
IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) Trigger values vs time instant

(b) Triggers vs time instant

Fig. 4. Delay anomaly detection in a robot. (a) Trigger values vs time instant. (b) Triggers vs time instant



(a) Snapshots of the experiment (b) Perturbation detection

Fig. 5. Perturbation anomaly detection in Turtlebot3.

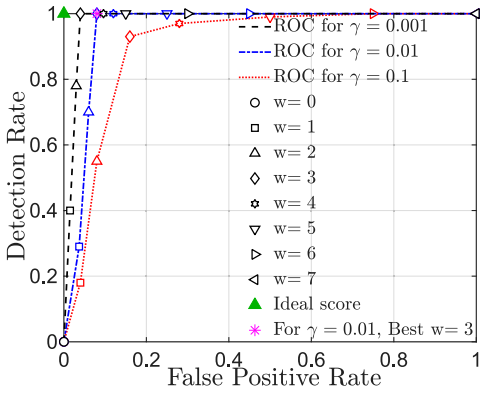


Fig. 6. ROC curves for w variation with γ.

as repositioning after encountering obstacles or interference from humans. These perturbations can deviate the robot from desired trajectory and compromise their performance. Timely and reliable detection of external perturbations enables the controller to deploy mitigation strategies such as adapting control inputs, or initiating recovery maneuvers, before adverse effects escalate. With this motivation, the following application focuses on detecting perturbations anomaly.

Firstly, we state the nature of perturbations considered in this application. Perturbations appear as deviations δ from the current state of the robot $\boldsymbol{x}[k]$, such that the perturbed state $\boldsymbol{x}_{prtrb}[k] = \boldsymbol{x}[k] + \delta$. These deviations may be instantaneous spikes or small, gradual shifts over time. Next, the real-time performance of the proposed detector in detecting perturbations is validated through hardware experiments.

Experiment: A set point tracking experiment is performed using a Turtlebot3 Waffle-Pi robot in an 3m $\times$ 3m experimental arena. The initial and final (desired) positions of the robot were (0.63,1.15) and $(1.70, -1.20)$, respectively. Control input $\boldsymbol{u} \in \mathbb{R}^2 : ||\boldsymbol{u}||_\infty \leq 0.22$ m/sec due to hardware constraints of the robot. An 8-camera Vicon Motion Capture system was used for tracking robot positions, which were fed back to a central computing unit to calculate the control inputs using the MPC framework described in [29]. The data obtained during this experiment is used to compute bounds (κ^-, κ^+) via Algorithm 1, with window length $w = 3$ and confidence level $\gamma = 0.01$. The corresponding average and maximum computation times were 0.59 s and 0.67 s, respectively.

Next, the experiment is repeated to validate real-time anomaly detection, as shown in Fig. 5. In the snapshots depicted in Fig. 5 a, solid green circles highlight the desired position of the robot. Using the proposed anomaly detector, trigger value computed from real-time sensor data $\lambda(\hat{\boldsymbol{S}}_k) \in (\kappa^-, \kappa^+)$ for $k = 1 - 6$, indicating nominal operation. A perturbation is then introduced in the robot, as shown by a red arrow in Fig. 5(a). This is also illustrated through the blue solid vertical line in Fig. 5(b), which indicates a perturbation $\delta \in \mathbb{R}^2 : ||\delta|| \approx 1.2\,m$ introduced at $k = 7$. At $k = 8$, $\lambda_{\max}(\hat{\boldsymbol{S}}_k)$ exceeds κ^+, indicating the presence of perturbation (black dashed vertical line in Fig. 5(b). Therefore, the proposed detector successfully detects perturbation anomaly during hardware experiments. Once perturbation is detected, suitable mitigation strategies [7], [30] can be deployed to enable the robot to reach the target smoothly. Additional simulations with gradual perturbations are conducted in the next section.

V. Results and Discussions

This section presents the performance evaluation of the proposed detector in terms of standard metrics, comparison with existing works, and related discussions.

Simulation Setup: Consider a robot having discrete-time LTI dynamics $\boldsymbol{x}[k+1] = \boldsymbol{A}_d\boldsymbol{x}[k] + \boldsymbol{B}_d\boldsymbol{u}[k]$, where $\boldsymbol{x}[k] \in \mathbb{R}^2$, $\boldsymbol{u}[k] \in \mathbb{R}^2$, $\boldsymbol{A}_d = \boldsymbol{I}$, $\boldsymbol{B}_d = h\boldsymbol{I}$, and $h = 0.01$s is the sampling time. Input is bounded by $||\boldsymbol{u}||_\infty \leq 2.5$. The state measurements are assumed to have additive Gaussian noise with mean $\mu = 0.01$ and variance $\sigma^2 = 0.01$. The robot tracks a reference trajectory using the MPC framework described in [29], with weight matrices $\boldsymbol{Q} = 15\boldsymbol{I}$, $\boldsymbol{R} = 8\boldsymbol{I}$, and using a system equipped with Intel Core i7 CPU and 16gb of RAM. Nominal (anomaly-free) tracking data of robot states is thus obtained, which is used to derive the bounds (κ^-, κ^+) via Algorithm 1 with a confidence level $\gamma = 0.01$ and window length $w = 3$. To validate detection performance, perturbations $\delta \in \mathbb{R}^2 : ||\delta||_\infty \leq 0.6$ (as defined in Sec. IV-C) are introduced in the robot at $k = 10, 25, 40$ for three consecutive steps each, during the trajectory tracking task. The following performance metrics are then computed.

Performance Metrics: True triggers (TT) are correct detections when both anomalies and triggers are present. True negatives (TN) indicate the absence of both. While false triggers

TABLE I
COMPARISON OF THE PROPOSED METHOD WITH EXISTING WORKS

Method	*TT*	*FT*	Accuracy	Detection Rate	FPR	F1-Score	Avg. I-T	Max. I-T
Heuristic bound [7]	5	24	0.73	0.33	0.21	0.22	1.98	8
Heuristic bound for Covariance Matrix [13]	9	16	0.82	0.67	0.14	0.47	2.56	10
Sliding window Mahalanobis distance [3]	14	10	0.90	0.95	0.08	0.71	3.85	15
Hybrid Model–Data FDD [16]	14	7	0.93	0.98	0.06	0.79	4.68	19
Multi-IMU Kalman Filter [15]	14	6	0.94	1.00	0.07	0.82	5.20	18
Proposed Detector	**15**	**5**	**0.96**	**1.00**	**0.04**	**0.85**	**6.29**	**21**

(FT) denote triggers without anomalies, false negatives (FN) are undetected anomalies. Then, the false positive rate (FPR) $= \frac{FT}{(FT+TN)}$ measures how often the detector triggers incorrectly. The detection rate (recall) $= \frac{TT}{(TT+FN)}$ quantifies the proportion of actual anomalies correctly detected. Accuracy$= \frac{(TT+TN)}{(TT+TN+FT+FN)}$ represents the overall correctness of the detector's decisions. Precision$= \frac{TT}{(TT+FT)}$, indicates the fraction of detected anomalies that are indeed true, while the F1-score is the harmonic mean of precision and recall. Further, the average inter-trigger time (Avg. I-T) reflects the mean interval between two consecutive triggers, whereas the maximum inter-trigger time (Max. I-T) denotes the longest observed interval between two triggers. Using the above simulation setup, the proposed detector is compared with existing methods as follows.

A. Results and Comparison

Table 1 presents a performance comparison of the proposed detector with heuristically chosen threshold [7], covariance matrix based approach [13], sliding-window Mahalanobis distance [3], hybrid model-data approach [16], and Kalman filter-based method [15], all evaluated under the identical simulation setup described in Sec. V. Notably, the proposed detector demonstrates a substantial improvement in minimizing FT, generating only 5 FT, compared to 24 in [7], 16 in [13], 10 in [3], 7 in [16], and 6 in [15]. This improvement arises because our method derives statistically grounded eigenvalue bounds using Matrix Chernoff Inequalities, which suppress noise-driven false triggers. Consequently, the F1-score reaches 0.857, compared to 0.225, 0.470, 0.715, 0.790, and 0.820 for [7], [13], [3], [16], and [15], respectively. This is because our method jointly achieves high precision and recall. Similarly, the accuracy of the proposed detector reaches 96%, significantly outperforming the rest, as probabilistic bounds balance sensitivity to anomalies with robustness against noise. While all methods demonstrate reasonable detection rates, the proposed detector achieves the detection rate of 1 by capturing both abrupt and gradual deviations. Additionally, the average and maximum inter-trigger times for the proposed detector are 6.29 and 21, respectively, which are significantly higher than those of the others (Avg. I-T ranging from 1.98 to 5.20, Max. I-T from 8 to 19, as shown in Table 1). This observation is supported by the reduction in total number of triggers observed in the proposed method. Extending this discussion to DL methods, it is observed that they often demand extensive offline training, typically needing several minutes to hours [31], [32]. In contrast, the proposed method computes the detection thresholds in under 1 s via simple matrix operations and evaluates each trigger value in only $1-3$ ms. This keeps the online detection delay minimal, enabling the method to offer both rapid offline setup and fast online response.

TABLE II
PERFORMANCE VARIATION WITH γ, w, μ, AND σ^2

γ	w	μ	σ^2	κ^+	FPR	Accu.	F1-Sc.	Det. dl.
0.1	2	0.05	0.05	1.69	0.11	0.90	0.65	0.03
0.1	2	0.07	0.05	1.46	0.13	0.89	0.62	0.04
0.1	2	0.05	0.07	1.55	0.12	0.89	0.63	0.04
0.1	4	0.05	0.05	0.92	0.10	0.91	0.67	0.02
0.1	4	0.07	0.05	0.82	0.11	0.90	0.64	0.03
0.1	4	0.05	0.07	0.86	0.10	0.90	0.66	0.03
0.01	2	0.05	0.05	2.33	0.07	0.93	0.73	0.02
0.01	2	0.07	0.05	1.70	0.10	0.90	0.65	0.03
0.01	2	0.05	0.07	1.91	0.10	0.91	0.67	0.03
0.01	4	0.05	0.05	1.22	0.06	0.93	0.74	0.01
0.01	4	0.07	0.05	1.05	0.10	0.90	0.66	0.02
0.01	4	0.05	0.07	1.10	0.09	0.91	0.68	0.02

Accu.:Accuracy; F1-Sc.:F1-Score; Det. dl.:Detection Delay (s)

B. Discussions

A deeper analysis of the proposed detector is now presented.

1) Effect of Confidence Level γ and Window Length w: The results shown in Table 2 are derived under different γ and w. It can be observed that for a fixed γ, the upper bound κ^+ decreases as w increases.

For e.g., at $\gamma = 0.1$, κ^+ drops from 1.69 (for $w = 2$) to 0.92 (for $w = 4$). This is because longer w overcomes the effect of random fluctuations in covariance matrices, yielding tighter bounds and a reduction in FT. Conversely, for a fixed w, κ^+ increases as γ decreases, i.e., more conservative bounds are obtained for lower γ. For e.g., at $w = 2$, κ^+ rises from 1.69 at $\gamma = 0.1$ to 2.33 at $\gamma = 0.01$. This increase in κ^+ is accompanied by lower FPR values, and improved Accuracy and F1-Score.

2) Effect of Noise Parameters: Noisy measurements also influence the detector performance, as shown in Table 2. Increasing the noise mean μ from 0.05 to 0.07 (with variance fixed) or increasing variance σ^2 from 0.05 to 0.07 (with mean fixed) reduces κ^+ and results in higher FPR and detection delay, and lower Accuracy and F1-Score. For e.g., at $\gamma = 0.01, w = 2$, raising μ from 0.05 to 0.07 lowers κ^+ from 2.33 to 1.70 and increases FPR from 0.070 to 0.104 and detection delay from 0.01 to 0.02, while Accuracy drops from 0.93 to 0.90. Nevertheless, the detector maintains reasonably good performance across all settings. In summary, decreasing γ strengthens the bounds and improves robustness, increasing w reduces noise-driven FT, while increasing noise μ or σ^2 degrades performance. By carefully tuning γ, w relative to the noise parameters, desirable performance can be achieved.

3) Confidence Level γ and Window w Selection: While both γ and w can be tuned to obtain suitable bounds, γ is typically at designer's discretion for choosing a fat- or thin-tail (Remark 2). To choose the best w for a given γ, one can utilize the ROC curve

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

obtained for a fixed γ, as shown in Fig. 6. Here, the performance of the proposed detector is observed by plotting Detection Rate against FPR for different values of w. Precisely, ROC is obtained for $\gamma = 0.001, 0.01, 0.1$ by varying $w = 0, 1, 2, \ldots$. The green triangle at (0,1) indicates the score of an ideal detector, denoting detection of all anomalies without FT. Consequently, the w value that yields FPR and Detection Rate closest to the ideal score is the best w for a given γ. For e.g., $w = 3$ is identified as the best w for $\gamma = 0.01$, and marked by a magenta star in Fig. 6, signifying the best achievable trade-off between high Detection Rate and low FPR for the chosen settings.

VI. Conclusion

A novel data-driven, covariance matrix-based anomaly detector is proposed for robots by leveraging the Matrix Chernoff Inequality. An in-depth performance analysis of the proposed detector is presented by introducing anomalies in robot states during simulations and hardware experiments. Its versatility was validated across three different scenarios, namely detecting sensor corruption, delay and perturbation anomalies, in various robot setups, each governed by a different controller. Additionally, the proposed detector exhibits effective performance in terms of standard metrics and comparison with existing works. Future work will focus on identification of malicious attacks, and coupling anomaly-specific mitigation with the proposed detector for appropriate control input selection.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

[2] P. Guo, H. Kim, N. Virani, J. Xu, M. Zhu, and P. Liu, "RoboADS: Anomaly detection against sensor and actuator misbehaviors in mobile robots," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2018, pp. 574–585.

[3] E. Khalastchi, M. Kalech, G. A. Kaminka, and R. Lin, "Online data-driven anomaly detection in autonomous robots," *Knowl. Inf. Syst.*, vol. 43, pp. 657–688, 2015.

[4] H. Wang, Y. Sun, and M. Liu, "Self-Supervised drivable area and road anomaly segmentation using RGB-D data for robotic wheelchairs," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4386–4393, Oct. 2019.

[5] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multi-modal anomaly detection," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1326–1333, Apr. 2020.

[6] T. Ji, A. N. Sivakumar, G. Chowdhary, and K. Driggs-Campbell, "Proactive anomaly detection for robot navigation with multi-sensor fusion," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4975–4982, Apr. 2022.

[7] R. Dubey, S. Gupta, S. Chaudhary, N. S. Tripathy, and S. V. Shah, "Finite-Time convergence of multi-robot segregation using mpc with aperiodic motion smoothing," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2024, pp. 2209–2214.

[8] Z. Xu, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "Onboard dynamic-object detection and tracking for autonomous robot navigation with rgb-d camera," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 651–658, Jan. 2023.

[9] M. H. Romanycia and F. J. Pelletier, "What is a heuristic," *Comput. Intell.*, vol. 1, no. 1, pp. 47–58, 1985.

[10] E. Khalastchi and M. Kalech, "On fault detection and diagnosis in robotic systems," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–24, 2018.

[11] K. M. Park, Y. Park, S. Yoon, and F. C. Park, "Collision detection for robot manipulators using unsupervised anomaly detection algorithms," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 2841–2851, Oct. 2021.

[12] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.

[13] S. Jin and D. S. Yeung, "A covariance analysis model for DDoS attack detection," in *Proc. IEEE Int. Conf. Commun.*, 2004, vol. 4, pp. 1882–1886.

[14] S. Joe Qin, "Statistical process monitoring: Basics and beyond," *J. Chemo.*, vol. 17, no. 8/9 pp. 480–502, 2003.

[15] E. Mounier, K. Malek, M. Korenberg,, and A. Noureldin, "Multi-IMU system for robust inertial navigation: Kalman filters and differential evolution-based fault detection and isolation," *IEEE Sensors J.*, vol. 25, no. 6, pp. 9998–10014, Mar. 2025.

[16] M. Zabihi, R. Mehrizi Valiollahi, A. Kasaiezadeh, M. Pirani, and A. Khajepour, "A hybrid model-data vehicle sensor and actuator fault detection and diagnosis system," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 8121–8133, Jul. 2024.

[17] J. A. Tropp, "User-Friendly tail bounds for sums of random matrices," *Found. Comput. Math.*, vol. 12, pp. 389–434, 2012.

[18] J. R. Schott, *Matrix Analysis for Statistics*. Hoboken, NJ, USA: Wiley, 2016.

[19] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*, vol. 48. Cambridge, UK: Cambridge Univ. Press, 2019.

[20] M. G. Natrella, *Experimental Statistics*. vol. 91. North Chelmsford, MA, USA: Courier Corporation, 2005.

[21] A. D. Luca and G. Oriolo, "Modelling and control of nonholonomic mechanical systems," in *Kinematics and Dynamics of Multi-Body Systems*. London, UK: Springer, 1995, pp. 277–342.

[22] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1454–1467, Jun. 2014.

[23] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O'Reilly, "A nonlinear disturbance observer for robotic manipulators," *IEEE Trans. Indus. Electron.*, vol. 47, no. 4, pp. 932–938, Aug. 2000.

[24] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. vol. 3, New York City, NY, USA: Wiley, 2006.

[25] E. Fridman, *Introduction to Time-Delay Systems*. vol. 75, Cham, Switzerland: Springer, 2014.

[26] W. Robinson and A. Soudack, "A method for the identification of time delays in linear systems," *IEEE Trans. Autom. Control*, vol. 15, no. 1, pp. 97–101, Feb. 1970.

[27] Y. Orlov, L. Belkoura, J.-P. Richard, and M. Dambrine, "On identifiability of linear time-delay systems," *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1319–1324, 2002.

[28] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of Time-Delay Systems*. New York, NY, USA: Springer, 2003.

[29] A. Anderson, A. H. González, A. Ferramosca, and E. Kofman, "Finite-Time convergence results in model predictive control," in *Proc. Eur. Control Conf.*, 2018, pp. 3203–3208.

[30] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 604–611, Apr. 2020.

[31] D. Azzalini, L. Bonali, and F. Amigoni, "A minimally supervised approach based on variational autoencoders for anomaly detection in autonomous robots," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2985–2992, Apr. 2021.

[32] S. Rajendran, V. Lenders, W. Meert, and S. Pollin, "Crowdsourced wireless spectrum anomaly detection," *IEEE Trans. Cogn. Comm. Netw.*, vol. 6, no. 2, pp. 694–703, Jun. 2020.