BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(6), 2026, Article number: 3067
DOI: 10.24425/bpasts.2026.3067

Adaptive state estimation for LQG systems using hybrid Kalman-LSTM filtering under time-varying noise

Suleiman DIABAT, Leszek AMBROZIAK*, and Ewa PAWŁUSZEWICZ

Bialystok University of Technology, Faculty of Mechanical Engineering, Wiejska 45C, 15-351 Bialystok, Poland

Abstract. This work presents a simulation-based proof of concept for a hybrid estimation framework integrating a standard Kalman filter (KF) with a lightweight Long Short-Term Memory (LSTM) network for sequential online prediction of process and measurement noise covariances. The LSTM is trained offline using nonstationary datasets generated from an XArm6 ROS 2/Gazebo model and provides adaptive covariance estimates that are injected into the Kalman recursion during closed-loop execution. The covariance predictor uses simulator-provided joint states within its temporal input window, enabling controlled evaluation under known time-varying noise conditions. The proposed estimator improves state-estimation accuracy, reduces tracking RMSE, and improves NIS/NEES consistency relative to the static Kalman filter under highly nonstationary disturbances. The results provide clear evidence of the feasibility and effectiveness of the proposed hybrid Kalman–LSTM framework for adaptive covariance estimation in closed-loop robotic control. The conducted evaluation further confirms stable, bounded, and consistent estimator performance under controlled time-varying noise conditions.

Keywords: Kalman filter; LSTM; adaptive estimation; LQG control; robotics; time-varying noise; process noise; measurement noise.

1. INTRODUCTION

Accurate state estimation is one of the fundamental requirements for high-performance control of robotic manipulators operating under uncertain and time-varying conditions. Classical Kalman filtering assumes stationary and well-characterized noise statistics [1]. However, industrial manipulators such as the XArm6 exhibit nonstationary behavior caused by friction variation, actuator heating, payload changes, communication delays, and external disturbances [2]. Under these conditions, Kalman filters based on fixed noise covariances may become statistically inconsistent.

Adaptive Kalman filtering methods have been developed to address this limitation. Recent studies have investigated recursive and time-varying noise estimation within the Kalman filtering framework [3–6]. These methods generally provide greater robustness than fixed-covariance filters. Nevertheless, their adaptation mechanisms are often derived under specific modeling assumptions and may depend on manually selected tuning parameters.

Learning-based methods have also been investigated for adaptive covariance estimation. Convolutional and temporal neural networks have been applied to navigation and sensor-fusion problems [7–9]. Although these methods improve robustness under rapidly varying conditions, they are commonly evaluated as standalone estimators rather than as components of closed-loop optimal control systems.

Hybrid neural-Kalman architectures further integrate data-driven learning with Bayesian filtering [10]. Recent LSTM-based methods, for example, have demonstrated the ability to infer robot states from time-series sensor data under nonlinear and configuration-dependent dynamics. However, many existing hybrid methods estimate only one covariance component or are evaluated outside the feedback control loop.

In contrast, this work proposes a hybrid Kalman-LSTM estimator that predicts both process and measurement noise covariances online. The predicted covariances are incorporated directly into a Kalman filter operating within a Linear Qudratic Regulator framework (LQG). This integration enables the estimator to adapt to nonstationary disturbances while retaining the interpretable structure of classical stochastic estimation and optimal control.

The proposed framework is implemented as a modular ROS 2 pipeline. The LSTM is trained offline to identify temporal patterns associated with changes in the process and measurement noise levels. During online operation, its covariance predictions are injected into the Kalman recursion, and the resulting state estimate is supplied to the LQG controller. Simulation-based evaluations on Joint 2 of the XArm6 ROS 2/Gazebo model demonstrate improved estimation accuracy and NIS/NEES consistency relative to static and classical adaptive Kalman-filtering methods.

The main contributions of this work are: (i) a hybrid Kalman-LSTM estimator for online prediction of time-varying process and measurement noise covariances; (ii) integration of the adaptive estimator into an LQG control framework implemented in ROS 2; (iii) simulation-based evaluation on Joint 2 of the XArm6 model using RMSE, NIS, and NEES metrics; and

*e-mail: l.ambroziak@pb.edu.pl

Manuscript submitted 2026-03-30, revised 2026-07-22, initially accepted for publication 2026-08-03, published in November 2026.

© 2026 The Author(s). This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

(iv) closed-loop stability verification based on the eigenvalues of the LQR closed-loop system matrix.

2. SYSTEM MODELING

The XArm6 is a six-degree-of-freedom serial manipulator whose complete kinematic and dynamic behavior includes configuration-dependent inertia, Coriolis and centrifugal forces, gravity, friction, actuator effects, and inter-joint coupling [11–14]. The present estimator is formulated for Joint 2 as a reduced-order proof-of-concept model suitable for sequential KF-LQG implementation (as illustrated in Fig. 1). Nevertheless, the Joint 2 trajectory is generated by the complete six-DOF plant; therefore, the influence of the remaining joints and unmodeled nonlinear effects is retained in the observed joint response and represented as process uncertainty.

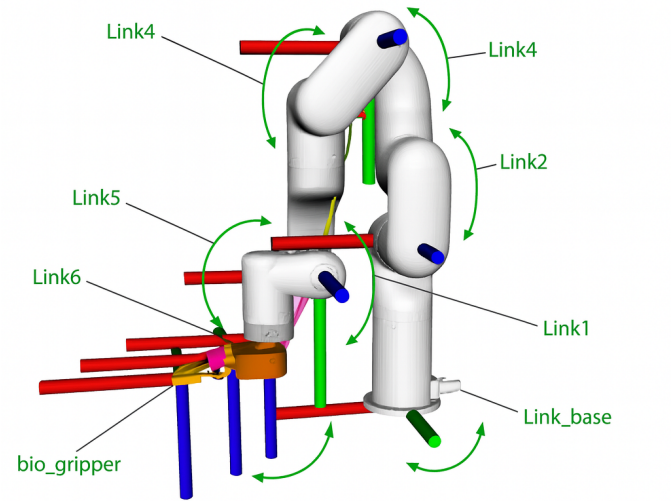

Fig. 1. Coordinate-frame visualization of the six-DOF XArm6 manipulator used in the experiments. The joint frames follow the standard ROS convention (X: red, Y: green, Z: blue), while the proposed KF–LSTM–LQG estimator is evaluated on Joint 2

It is important to distinguish the matrix used below from the geometric Jacobian of the manipulator. The complete XArm6 geometric Jacobian is of the dimension 6×6 and maps the six joint velocities to the end-effector velocity.

The projected dynamics of Joint 2 is represented by

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)+\tau_f(\dot{q})=\tau, \tag{1}$$

where $M(q)$ is the effective joint inertia, $C(q,\dot{q})\dot{q}$ contains the Coriolis and centrifugal effects, $g(q)$ is the gravity torque, $\tau_f(\dot{q})$ represents nonlinear friction, and τ is the applied motor torque. Defining $x=[\,q\;\;\dot{q}\,]^\top$, $u=\tau$ and taking into account external disturbances w, the nonlinear state model becomes

$$\dot{x}(t)=\begin{bmatrix}\dot{q}\\ f_q(q,\dot{q},u)\end{bmatrix}, \tag{2}$$

where $f_q(q,\dot{q},u)=\dfrac{1}{M(q)}\left[u-C(q,\dot{q})\dot{q}-g(q)-\tau_f(\dot{q})\right]$.

For LQG design, (2) is locally linearized about the quasi-static operating point (x_0,u_0):

$$\dot{\tilde{x}}(t)=A_c\tilde{x}(t)+B_c\tilde{u}(t)+w(t),$$

where $\tilde{x}=x-x_0$, $\tilde{u}=u-u_0$, and

$$A_c=\left.\frac{\partial f}{\partial x}\right|_{(x_0,u_0)}=\begin{bmatrix}0 & 1\\ \dfrac{\partial f_q}{\partial q} & \dfrac{\partial f_q}{\partial\dot{q}}\end{bmatrix}_{(x_0,u_0)},\qquad B_c=\left.\frac{\partial f}{\partial u}\right|_{(x_0,u_0)}.$$

Because Joint 2 operates in the considered quasi-static regime with $\dot{q}_0\approx0$ and $\ddot{q}_0\approx0$, the local stiffness, damping, friction, gravity, and residual coupling terms are not included explicitly in the nominal double-integrator model. Instead, their residual influence is lumped into $w(t)$. In contrast, the reduced estimator state is $x=[\,q_2\;\;\dot{q}_2\,]^\top$. Hence, $A_c=\partial f/\partial x$ is a 2×2 state Jacobian of the local Joint 2 dynamics. Accordingly, $\left.\dfrac{\partial f_q}{\partial q}\right|_{(x_0,u_0)}\approx0$, $\left.\dfrac{\partial f_q}{\partial\dot{q}}\right|_{(x_0,u_0)}\approx0$, and therefore

$$A_c=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}.$$

For the physical torque input,

$$\left.\frac{\partial f_q}{\partial u}\right|_{(x_0,u_0)}=\frac{1}{M(q_0)}.$$

Denoting the effective inertia at the operating point by $M_0=M(q_0)$ gives

$$B_c=\begin{bmatrix}0\\ 1/M_0\end{bmatrix}.$$

After dropping the deviation notation for simplicity, the continuous-time nominal model is

$$\dot{x}(t)=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}x(t)+\begin{bmatrix}0\\ 1/M_0\end{bmatrix}u(t)+w(t). \tag{3}$$

Thus, $u(t)$ remains the physical Joint 2 torque, whereas the corresponding input-induced acceleration is $u(t)/M_0$.

For controller design, the reduced-order model is discretized using the nominal sampling interval $\Delta t=0.01$ s, corresponding to the nominal 100 Hz joint-state stream in the ROS 2/Gazebo simulation. This does not imply that all ROS 2 nodes are independently clocked at 100 Hz; the estimator and controller nodes are activated by incoming messages. Applying forward-Euler discretization to (3) [15, 16] yields

$$x_{k+1}=\begin{bmatrix}1 & \Delta t\\ 0 & 1\end{bmatrix}x_k+\begin{bmatrix}0\\ \Delta t/M_0\end{bmatrix}u_k+w_k\,. \tag{4}$$

Indeed, since $\dot{q}=x_2$ and $\ddot{q}=u/M_0$, the component-wise updates are

$$q_{k+1}\approx q_k+\dot{q}_k\Delta t,\qquad \dot{q}_{k+1}\approx\dot{q}_k+\frac{\Delta t}{M_0}u_k,$$

which recover (4). During execution, the Kalman filters update their transition matrices using the elapsed time obtained from consecutive measurement timestamps, while (4) represents the nominal model used for LQR design. Consequently, position integrates velocity, whereas velocity integrates the acceleration produced by the applied torque. All remaining model mismatch is contained in w_k, whose covariance is adapted online by the LSTM predictor.

3. LSTM ARCHITECTURE AND TRAINING

The XArm6 joint encoders provide position measurements only, so the measurement model is

$$z_k = Hx_k + v_k,$$

with $H = [1 \;\; 0]$, where z_k denotes the encoder measurement and v_k represents the encoder noise. Velocity is not measured directly by the hardware; numerical differentiation of encoder data would amplify quantization and sampling noise. Instead, the Kalman filter exploits the process model to estimate the unmeasured velocity state. The pair (A, H) is observable, ensuring that both position and velocity can be reconstructed reliably.

Joint 2 exhibits nonstationary behavior due to friction variation, actuator heating, payload effects, and communication jitter. All unmodeled and time-varying dynamics are therefore represented as stochastic disturbances within the Kalman filtering framework. Accordingly, the process and measurement noises are modeled as $w_k \sim \mathcal{N}(0, Q_k)$, $v_k \sim \mathcal{N}(0, R_k)$, with time-varying covariances Q_k (process noise covariance) and R_k (measurement noise covariance). Classical adaptive approaches such as the Sage-Husa algorithm estimate these covariances from innovation statistics, which may result in slow adaptation or covariance drift under rapidly changing conditions [17, 18]. In contrast, the proposed approach predicts $\hat{Q}_k$ and $\hat{R}_k$ online using an LSTM trained on nonstationary trajectories, enabling faster and more stable adaptation.

The LSTM covariance predictor was trained offline using trajectories generated from Joint 2 of the XArm6 ROS 2/Gazebo model during cyclic operation. The trajectories contain alternating acceleration and deceleration phases and therefore provide temporal excitation across different motion regimes. Twenty piecewise-constant noise profiles, covering low-, medium-, and high-variance conditions, were introduced into the process and measurement channels.

Let q_k^{true} denotes the simulator ground-truth position, q_k^{proc} – the process-noise-corrupted position, q_k^{meas} – the noisy encoder measurement, and $r_k^{\text{sim}} = q_k^{\text{meas}} - q_k^{\text{true}}$ be a ground-truth-dependent simulator residual. Then, for each trajectory, the four training features were $\{q_k^{\text{true}}, q_k^{\text{proc}}, q_k^{\text{meas}}, r_k^{\text{sim}}\}$. Kalman innovation

$$\nu_k = z_k - H\hat{x}_{k|k-1}.$$

Sliding windows of $N = 50$ samples form $X_{k-49:k} \in \mathbb{R}^{50\times4}$, with regression targets $y_k = [\sigma_Q(k), \sigma_R(k)]$. The dataset structure is shown in Fig. 2.

Each input window is normalized using feature-wise z-score statistics [19]: $\tilde{X} = \frac{X-\mu_X}{\sigma_X}$, where $\mu_X, \sigma_X \in \mathbb{R}^4$ denote, respectively, the mean and standard deviation of each of the four input channels, computed once over the entire dataset. Normalizing each feature independently avoids scale differences between q^{true}, q^{proc}, q^{meas}, and r_k^{sim}, and was necessary for stable and consistent LSTM training.

The prediction model consists of two stacked LSTM layers with 128 hidden units each, followed by a fully connected regression head ($128 \to 64 \to 2$) with a ReLU activation between the linear layers. Preliminary ablation tests showed that smaller models (64 units) underfit the slow noise variations, whereas larger models (256 units) provided no measurable improvement and tended to overfit. The selected architecture therefore pro-

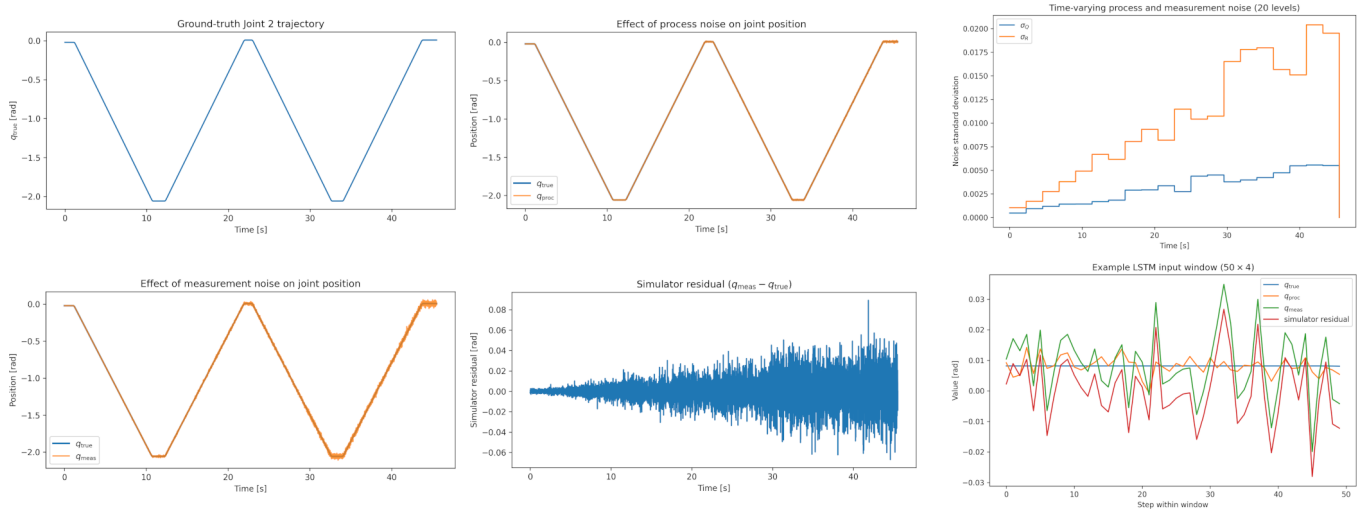

Fig. 2. Construction of the Joint 2 training dataset under time-varying noise: (a) simulator ground-truth joint trajectory; (b) trajectory after process-noise injection; (c) imposed process- and measurement-noise schedules; (d) noisy encoder measurement; (e) ground-truth-dependent simulator residual $r_k^{\text{sim}} = q_k^{\text{meas}} - q_k^{\text{true}}$; and (f) representative 50×4 LSTM training window comprising the ground-truth trajectory, process-noise-corrupted trajectory, encoder measurement, and simulator residual. The profiles cover low-, medium-, and high-variance operating intervals

vides sufficient capacity to capture nonlinear temporal patterns without unnecessary complexity. The final two outputs correspond to the predicted instantaneous process and measurement noise levels for time step k.

Training was performed in PyTorch using the Adam optimizer (learning rate 5×10^{-4}, batch size 64) for 120 epochs with an 80/20 train–validation split and a CosineAnnealing learning-rate scheduler. To enforce physically plausible temporal behavior in the predicted noise covariances, a smoothness regularizer was added to the training loss: $\mathscr{L} = \mathrm{MSE}(\hat{y}, y) + 0.05\big(\mathrm{TV}(\hat{\sigma}_Q) + \mathrm{TV}(\hat{\sigma}_R)\big)$, where the total-variation (TV) terms penalize abrupt temporal changes in the predicted noise levels. This regularization reflects the assumption that the dominant process and measurement noise characteristics vary gradually over short time intervals and suppresses high-frequency artifacts in the predicted covariance profiles, [20].

The selected LSTM configuration thus offers an efficient and robust trade-off for learning time-varying noise covariances under sequential ROS 2/Gazebo execution constraints.The offline validation results for the predicted process- and measurement-noise levels are presented in Fig. 3. The LSTM achieved validation coefficients of determination $R^2_Q = 0.542$ and $R^2_R = 0.735$. While estimating σ_Q is inherently more challenging due to the lack of direct process-level measurements, the network consistently captures the temporal trend and relative magnitude of both noise components. Such trend fidelity is sufficient for Kalman filtering, which primarily depends on accurately tracking the evolution of uncertainty rather than its exact absolute value. This accuracy level proved adequate to yield measurable improvements in adaptive KF closed-loop performance under nonstationary disturbances. During sequential ROS 2/Gazebo execution, the LSTM processes 50-sample windows of $[q_k^{\mathrm{true}}, q_k^{\mathrm{true}}, q_k^{\mathrm{meas}}, r_k^{\mathrm{sim}}]^\top$, where the second q_k^{true} approximates the unavailable q_k^{proc} and $r_k^{\mathrm{sim}} = q_k^{\mathrm{meas}} - q_k^{\mathrm{true}}$. Thus, the online tensor retains the 50×4 dimensionality used in training but relies on privileged simulator signals. The predicted covariances $\hat{Q}_k$ and $\hat{R}_k$ replace the fixed Kalman-filter covariances during sequential closed-loop execution. The Kalman filter recursion is given by:

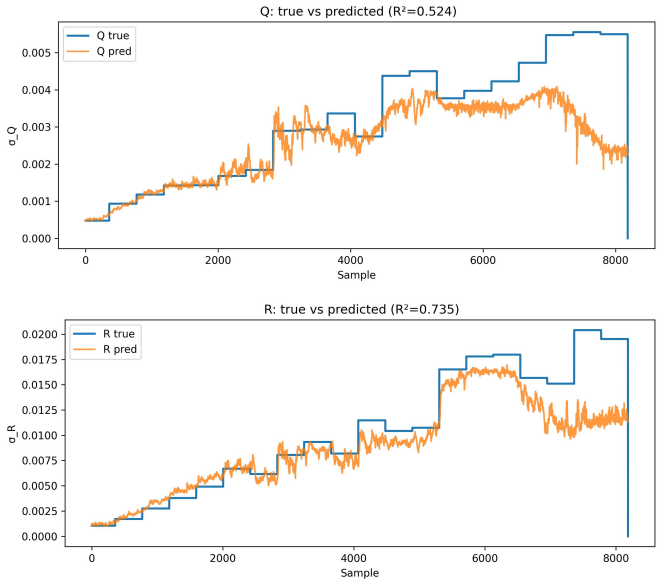

Fig. 3. Offline validation of the LSTM noise-level predictions. The upper panel compares the true and predicted process-noise level $\sigma_Q(k)$, while the lower panel compares the true and predicted measurement-noise level $\sigma_R(k)$ on the validation sequence

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}, \tag{5}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^\top + \hat{Q}_k, \tag{6}$$

$$K_k^{\mathrm{KF}} = P_{k|k-1}H^\top \left(HP_{k|k-1}H^\top + \hat{R}_k\right)^{-1}, \tag{7}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^{\mathrm{KF}}\left(z_k - H\hat{x}_{k|k-1}\right), \tag{8}$$

$$P_{k|k} = \left(I - K_k^{\mathrm{KF}}H\right)P_{k|k-1}. \tag{9}$$

By continuously updating the covariance terms, the estimator maintains statistical consistency under rapidly changing noise conditions caused by friction variation, encoder quantization, and actuator heating.

4. LQG CONTROL LAYER

The control layer employs a discrete-time Linear Quadratic Regulator (LQR) designed for the joint-level model introduced earlier. The controller minimizes the infinite-horizon quadratic cost

$$J = \sum_{k=0}^{\infty}\left(x_k^\top Q_c x_k + u_k^\top R_c u_k\right), \tag{10}$$

with weighting matrices selected as $Q_c = \mathrm{diag}(10,\ 1)$ and $R_c = 0.01$. This choice reflects standard joint-space LQR tuning practice, emphasizing accurate position tracking while maintaining moderate velocity damping and smooth control effort under quasi-static motion. The LQR feedback gain is computed by solving the algebraic Riccati equation, as commonly adopted in optimal control design [21]. For Joint 2, the resulting LQR feedback gain is $K_{\mathrm{LQR}} = \begin{bmatrix}13.2533 & 4.4192\end{bmatrix}$. The closed-loop system matrix $A_{\mathrm{cl}} = A - BK_{\mathrm{LQR}}$ has eigenvalues strictly inside the unit circle, confirming asymptotic stability of the discrete-time state-feedback system. The reference trajectory $x_{\mathrm{ref},k}$ is constructed from the commanded Joint 2 position reference. As observed in the simulation, the joint follows a slow triangular-like profile generated by the internal motion planner of the robot. Because this motion is quasi-static, the corresponding joint velocity remains very small throughout the experiment. Therefore, the reference state is defined as: $x_{\mathrm{ref},k} = \begin{bmatrix}q_{\mathrm{ref},k} & 0\end{bmatrix}^T$, where $q_{\mathrm{ref},k}$ is the desired position. Setting the velocity reference to zero is standard for slowly varying joint trajectories and avoids injecting artificial velocity commands that do not reflect the physical motion.

The torque applied to Joint 2 is computed using the LQG control law: $u_k = -K_{\mathrm{LQR}}(\hat{x}_{k|k} - x_{\mathrm{ref},k})$, where $\hat{x}_{k|k}$ is the state estimate provided by the hybrid adaptive KF-LSTM estimator.

The separation principle permits the Kalman filter and LQR to be designed separately, while the resulting closed-loop architecture retains the standard LQG structure when the estimator is stable. The adaptive covariance estimation plays a critical role: as noise statistics evolve during operation, the adaptive filter updates $(\hat{Q}_k, \hat{R}_k)$ online, enabling more accurate state estimates and smoother, more consistent torque commands.

To assess generalization beyond the training distribution, the estimators were evaluated on a highly non-smooth square-wave reference that differs from the smooth triangular trajectories used during training. The square wave introduces abrupt discontinuities and large simulator-residual excursions, providing a stringent test under twenty-level nonstationary Gaussian disturbances.

Figure 4 and Table 1 summarize the results. The static KF produces the largest RMSE, Sage-Husa improves the estimation accuracy, and the LSTM-adaptive KF achieves the lowest RMSE. However, the mean NIS and NEES values remain below their nominal expected values for all three methods; therefore, smaller NIS/NEES values are not interpreted as superior consistency. The results support improved estimation accuracy on the unseen trajectory while indicating conservative uncertainty estimates.

Table 1
Generalization test performance on square-wave trajectory

Method	RMSE [rad]	Mean NIS	Mean NEES
Static KF	0.1611	0.525	0.599
Sage-Husa	0.1498	0.574	0.577
LSTM-Adaptive KF	**0.1348**	0.427	0.457

5. SIMULATION SETUP

All evaluations used the XArm6 ROS 2/Gazebo model in ROS 2 Humble on Ubuntu 22.04. Joint 2 was selected because its shoulder-level location makes its response sensitive to downstream-link loading. Only Joint 2 was commanded; Joints 1 and 3–6 remained fixed at zero, contributing to the effective load, inertia, and gravity terms. Simultaneous multi-joint motion was outside the scope of this proof-of-concept study [22]. The modular pipeline is event-driven: estimation and control callbacks are activated by incoming messages, whereas the logger uses a 0.01 s timer [23–25]. Publisher and subscription integers denote QoS queue depths, not execution rates. Table 2 summarizes the activation mechanisms.

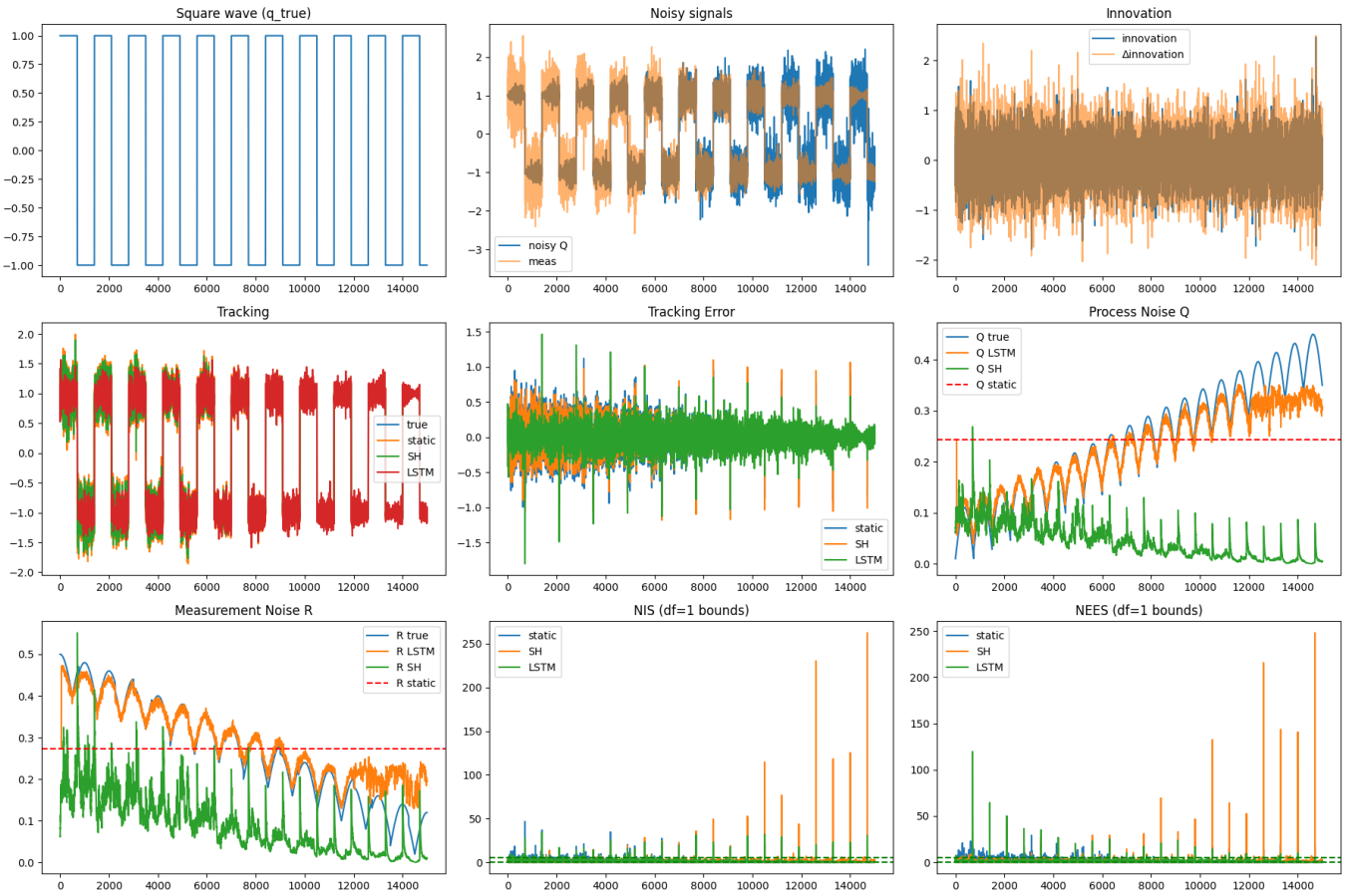

Fig. 4. Generalization performance on an unseen square-wave reference under time-varying Gaussian disturbances. The panels present the reference and noisy measurements, simulator-residual sequence, state estimates, adaptive covariance profiles, and NIS/NEES consistency measures for the static KF, Sage–Husa filter, and proposed LSTM-adaptive KF

Table 2
ROS 2 nodes and execution semantics

Node	Activation	Function
`q_true_joint2`	`/joint_states`	Publishes simulator joint state
`tv_noise_joint2`	`/joint_states`	Injects time-varying noise
`lstm_qr_joint2`	`/joint_states_noisy`	Predicts $\hat{Q}_k, \hat{R}_k$
`kf_static_joint2`	`/joint_states_noisy`	Static Kalman filter
`adaptive_kf_joint2`	`/joint_states_noisy`	LSTM-adaptive Kalman filter
`lqg_joint2`	`/kf_estimates/joint2`	Updates the control command
`logger_joint2`	0.01 s timer	Synchronous signal logging

The hybrid KF-LSTM-LQG framework is implemented as a modular ROS 2 pipeline with explicit topic-based data exchange between estimation, learning, and control components. Noisy encoder measurements are provided to both the static and adaptive Kalman filters, while the adaptive filter additionally receives time-varying noise covariance predictions $(\hat{Q}_k, \hat{R}_k)$ from the LSTM module. State estimates from both filters are made available to the LQG controller, and all relevant signals are logged synchronously for offline analysis and reproducibility. Table 3 summarizes the key ROS 2 topics used in the online pipeline, while Fig. 5 illustrates the complete topic-level connectivity. The architecture enforces deterministic data flow and strict modularity, allowing the learning-based covariance adaptation to be integrated without modifying the underlying estimation or control loops.

Table 3
ROS 2 topics used in the online pipeline

Topic	Description
`/joint_states`	Ground-truth joint state
`/joint_states_noisy`	Noisy encoder measurements
`/kf_estimates/joint2`	Adaptive KF estimate
`/kf_static/joint2`	Static KF estimate
`/qr_estimates/joint2/Q`	Predicted process noise covariance
`/qr_estimates/joint2/R`	Predicted measurement noise covariance
`/q_ref/joint2`	Reference trajectory
`/q_ref_corrected/joint2`	LQG-corrected reference

All experiments were designed to be fully reproducible at the code, data, and system level. Fixed random seeds were used for all stochastic components, including noise injection, dataset shuffling, and neural-network training, ensuring run-to-run repeatability of the reported results. The complete KF-LSTM-LQG pipeline – including dataset generation, LSTM training, offline evaluation, and ROS 2 deployment – is maintained within a single project repository. All datasets are stored in compressed NumPy archives and can be regenerated from the provided logged trajectories using the accompanying preprocessing scripts.

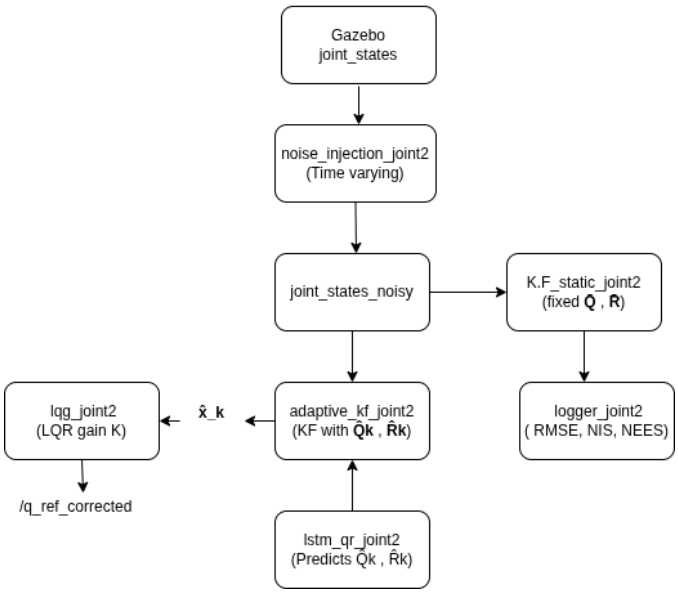

Fig. 5. ROS 2/Gazebo proof-of-concept implementation of the hybrid KF–LSTM–LQG pipeline. The LSTM uses temporal windows containing privileged simulator states to predict $(\hat{Q}_k, \hat{R}_k)$, which are supplied to the adaptive Kalman filter. The resulting state estimate drives the LQG controller, while the static KF is retained as a baseline. Simulator ground truth is used by the covariance predictor and for offline evaluation and would not be available in a physical implementation

ROS 2 node names, topic interfaces, and activation mechanisms are explicitly documented and a synchronized logger records all estimator and controller outputs. This design enables exact reconstruction of RMSE, NIS, NEES, and covariance-adaptation behavior from the saved logs. Table 4 summarizes the primary estimation metrics obtained during closed-loop operation. The adaptive Kalman filter achieves a 16.1% reduction in RMSE relative to the static filter, demonstrating substantially improved tracking accuracy under time-varying disturbances.

The normalized innovation squared (NIS) and normalized estimation error squared (NEES) [26, 27] are defined respectively as: $\mathrm{NIS}_k = \nu_k^\top S_k^{-1} \nu_k$ and $\mathrm{NEES}_k = e_k^\top P_k^{-1} e_k$. The central two-sided 95% confidence intervals are $\mathrm{NIS} \in [0.00098, 5.02389]$ for the scalar measurement and $\mathrm{NEES} \in [0.05064, 7.37776]$ for the two-dimensional state. The very small NIS and NEES values produced by the static KF indicate systematic *overestimation of*

Table 4
Adaptive KF improvement over static KF

Metric	Static	Adaptive	Imp.
RMSE [rad]	0.006705	0.005626	**16.1%** ↓
Mean NIS	0.000969	0.011581	closer to 1
Mean NEES	0.000450	0.195720	closer to 2

uncertainty, because its fixed $(\bar{Q}, \bar{R})$ do not match the true time-varying disturbance statistics. In contrast, the adaptive KF yields NIS and NEES values closer to their expected values of 1 and 2, respectively, indicating improved covariance modeling and statistical consistency. The corresponding closed-loop tracking performance is shown in Fig. 6. The LSTM-adaptive estimator follows the reference trajectory more accurately than the static KF, particularly during intervals of increased disturbance intensity. Figure 7 shows the empirical distributions of the predicted process and measurement noise covariances. KDE plots are used to summarize the marginal distributions; the corresponding temporal trajectories are shown later in the consistency analysis. The adaptive filter continuously adjusts $\hat{Q}_k$ and $\hat{R}_k$ to match the underlying disturbance level. In contrast, the static filter uses fixed values and therefore cannot represent rapid or slow variations in noise statistics.

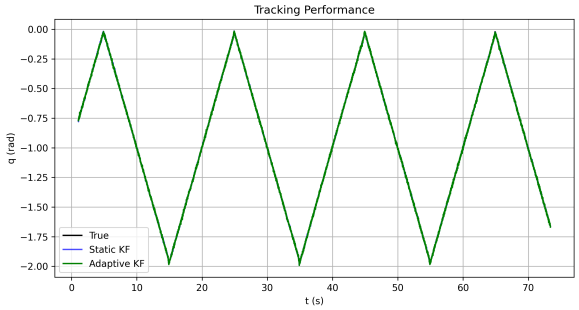

Fig. 6. Closed-loop Joint 2 tracking under the LQG controller and time-varying disturbances. The reference trajectory is compared with the static-KF and LSTM-adaptive-KF state estimates. The adaptive estimator reduces the tracking error particularly during intervals of increased noise variance

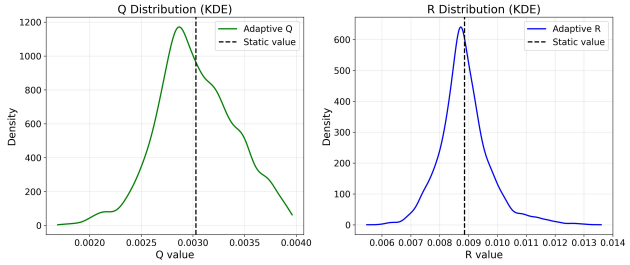

Fig. 7. Kernel-density estimates of the process and measurement covariances used by the estimators. The distributions of the LSTM-predicted covariances illustrate their variation across the experiment, whereas the dashed vertical lines represent the fixed covariance values used by the static KF

Figure 8 presents the time evolution of the NIS and NEES metrics. Compared with the static KF, the adaptive KF produces values closer to the expected NIS and NEES values, indicating improved statistical consistency under time-varying noise.

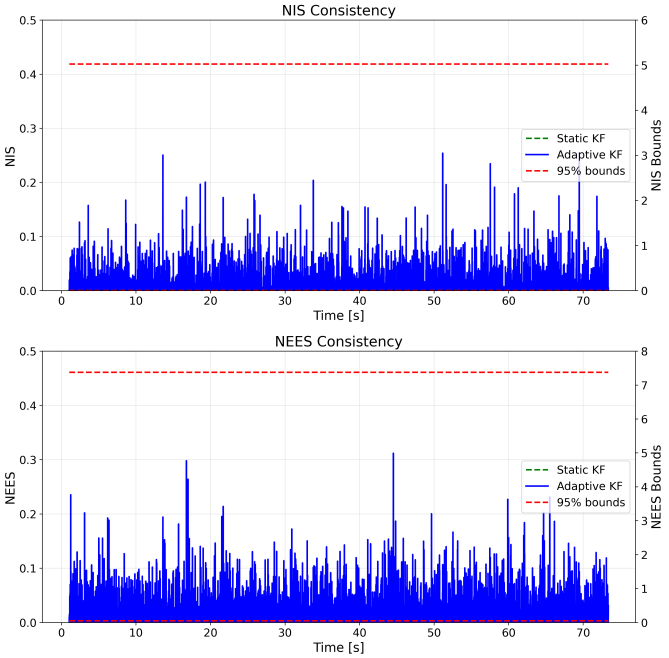

Fig. 8. Statistical-consistency analysis of the static and adaptive estimators: NIS evaluates scalar measurement-space consistency, whereas NEES evaluates the two-dimensional position–velocity state. The horizontal limits represent the corresponding two-sided 95% chi-square intervals

The proposed adaptive estimator preserves the stability properties of the Kalman filter because $\hat{Q}_k$ and $\hat{R}_k$ remain bounded and vary smoothly. For bounded, slowly time-varying covariances, the discrete-time Kalman filter remains mean-square stable under standard observability assumptions. This is reflected in the NIS and NEES trajectories, which remain close to their theoretical χ^2 intervals for the adaptive KF. The absence of divergence in either the state estimate or error covariance confirms Lyapunov-type practical stability.

Figure 9 shows the torque command produced by the LQG regulator when driven by the adaptive estimator. Adaptive noise estimation has a direct effect on the Kalman gain: $K_k^{\mathrm{KF}} \propto (HP_{k|k-1}H^\top + \hat{R}_k)^{-1}$, which modulates how aggressively the controller reacts. Larger $\hat{R}_k$ reduces estimator confidence in measurements, leading to smoother Larger $\hat{R}_k$ reduces estimator confidence in the measurements, leading to smoother control signals. In contrast, larger $\hat{Q}_k$ increases the predicted state uncertainty and generally strengthens the subsequent measurement correction through the Kalman gain. The resulting control effort remains bounded and free from high-gain spikes.

The source code, configuration files, and scripts required to reproduce the reported experiments are publicly available at https://github.com/suleiman994/Hybrid-KF-LSTM-LQG-XArm6-TimeVarying-Noise.

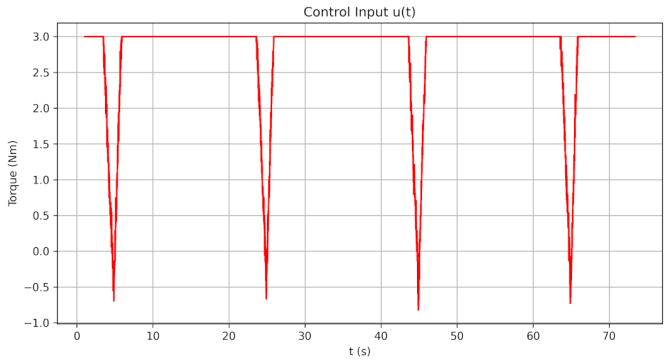

Fig. 9. Control input generated by the LQG controller using the LSTM-adaptive Kalman state estimate. The response remains bounded throughout the experiment and shows no high-frequency control spikes during covariance adaptation

6. CONCLUSIONS

This paper presented a simulation-based proof of concept for hybrid Kalman-LSTM covariance adaptation within an XArm6 ROS 2/Gazebo closed-loop framework. The LSTM predictor provides smooth and bounded covariance estimates that improve NIS/NEES consistency relative to the static KF and integrate directly into the Kalman recursion while retaining the standard LQG structure.

Offline and sequential ROS 2/Gazebo evaluations on Joint 2 demonstrated improved estimation accuracy, reliable tracking under nonstationary noise, and a 16.1% RMSE reduction relative to the static KF. The adaptive estimator also produced bounded closed-loop control behavior, demonstrating the potential of combining data-driven covariance prediction with model-based stochastic control.

A current limitation is the use of privileged simulator ground-truth features by the LSTM covariance predictor; therefore, the present results represent a simulation-based proof of concept. A separate follow-up study will replace these inputs with measurable signals, predicted states, and genuine Kalman-filter innovations and will provide physical-robot validation, including multi-joint operation under varying payload, friction, and communication-delay conditions. Overall, the results demonstrate the potential of learning-based covariance adaptation for robust, uncertainty-aware robotic control.

ACKNOWLEDGMENT

This work was partially funded under Project No. WZ/WM-IIM/4/2026, carried out at Bialystok University of Technology.

REFERENCES

[1] R.G. Brown and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 3rd ed. New York: John Wiley & Sons, 2012.

[2] L. Hao, R. Pagani, M. Beschi, and G. Legnani, "Dynamic and friction parameters of an industrial robot: Identification, comparison and repetitiveness analysis," *Robotics*, vol. 10, no. 1, p. 49, 2021.

[3] K.-V. Yuen, Y.-S. Liu, and W.-J. Yan, "Estimation of time-varying noise parameters for Unscented Kalman Filter," *Mech. Syst. Signal Process.*, vol. 180, p. 109439, 2022.

[4] N. Cohen and I. Klein, "Adaptive kalman-informed transformer," *Eng. Appl. Artif. Intell.*, vol. 146, p. 110221, 2025.

[5] I. Ullah, M. FayazA, N. Naveed, and D. Kim, "Ann based learning to Kalman filter algorithm for indoor environment prediction in smart greenhouse," *IEEE Access*, vol. 8, pp. 159 371–159 388, 2020.

[6] Y. Murato and A. Alkayal, "Novel temperature-effective modeling and state of- charge estimation based on sigma-point kalman filter for lithium titanate oxide battery," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 72, no. 5, p. e150809, 2024.

[7] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning gaussian noise models for state estimation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA),*, Brisbane, QLD, Australia, 2018, pp. 1436–1143.

[8] K. Chen, P. Zhang, L. You, and J. Sun, "Research on Kalman Filter fusion navigation algorithm assisted by cnn–lstm neural network," *Appl. Sci.*, vol. 14, no. 13, p. 5493, 2024.

[9] F. Song, Y. Li, W. Cheng, L. Dong, M. Li, and J. Li, "An improved kalman filter based on long short-memory recurrent neural network for nonlinear radar target tracking," *Wirel. Commun. Mob. Comput.*, vol. 2022, no. 1, pp. 1–10, 2022.

[10] G. Revach, N. Shlezinger, X. Ni, A.L. Escoriza, R. Ji, and Y.C. Eldar, "Kalmannet: Neural network aided Kalman filtering for partially known dynamics," *arXiv:2107.10043*, 2021.

[11] T. Rybus, "The obstacle vector field (ovf) method for collision-free trajectory planning of free-floating space manipulator," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 70, no. 2, p. e140691, 2022.

[12] L. L. Tien, A. Albu-Schaffer, A. de Luca, and G. Hirzinger, "Friction observer and compensation for control of robots with joint torque measurement," France, 2008, pp. 3789–3795.

[13] M. Czubenko and Z. Kowalczuk, "A simple neural network for collision detection of collaborative robots," *Sensors*, vol. 21, no. 12, p. 4235, 2021.

[14] M. Abayebas, F. Yumbla, J.-S. Yi, and W. Lee, "Passivity guaranteed dynamic friction model with temperature and load correction: Modeling and compensation for collaborative industrial robot," *IEEE Access*, vol. 9, pp. 1–14, 2021.

[15] K.J. Åström and R.M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2006.

[16] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Tech. Rep., 2006.

[17] W. Li, "Sage–husa algorithm based on adaptive double forgetting factors for real-time noise covariance updating," *Appl. Sci.*, vol. 15, no. 4, p. 1731, 2025.

[18] C. Jiang, Z. Wang, S. Tan, and H. Liang, "A novel adaptive noise covariance matrix estimation and filtering method: Application to multiobject tracking," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 626–641, 2024.

[19] Y. LeCun, L. Bottou, G.B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G.B. Orr and K.-R. Müller, Eds. Springer, 1998, vol. 1524, pp. 9–50.

[20] A.P. Wibawa, A.B.P. Utama, H. Elmunsyah, U. Pujianto, F.A. Dwiyanto, and L. Hernandez, "Time-series analysis with smoothed convolutional neural network," *J. Big Data*, vol. 9, p. 44, 2022.

[21] H. Górecki and M. Zaczyk, "Determination of optimal controllers. Comparison of two methods for electric network chain," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 66, no. 3, pp. 267–273, 2018, doi: 10.24425/123432.

[22] A. Ghorbanpour, "Cooperative robot manipulators dynamical modeling and control: An overview," *Dynamics*, vol. 3, no. 4, pp. 820–854, 2023.

[23] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, "Pronto: A multi-sensor state estimator for legged robots in real-world scenarios," *Front. Robot. AI*, vol. 7, pp. 1–18, 2020.

[24] J. Kang, Y. Wang, and X. Xiong, "Fast decentralized state estimation for legged robot locomotion via EKF and MHE," *IEEE Robot. Autom. Lett.*, vol. 9, no. 12, pp. 10 914–10 921, 2024.

[25] Y. Ye, Z. Nie, X. Liu, F. Xie, Z. Li, and P. Li, "ROS2 real-time performance optimization and evaluation," *Chin. J. Mech. Eng.*, vol. 36, p. 144, 2023.

[26] J.L. Crassidis and J.L. Junkins, *Optimal Estimation of Dynamic Systems*. CRC Press, 2011.

[27] O. Garcia Crespillo and A. Grosch, "Innovation vs residual kf based gnss/ins autonomous integrity monitoring in single fault scenario," in *Proceedings of the 30th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+)*, Portland, OR, USA, September 2017, pp. 2126–2136.