

# Linearized Kalman Filter (LKF) with NEES and NIS Performance Evaluation

## Introduction

This document outlines the implementation of a Linearized Kalman Filter (LKF) for a UAV-UGV cooperative localization system. The LKF uses a linearized model of the system's dynamics and measurement functions to estimate the state of the system in the presence of noise. The performance of the LKF is evaluated using two statistical tests:

- **NEES (Normalized Estimation Error Squared):** Evaluates the accuracy of the state estimate.
- **NIS (Normalized Innovation Squared):** Evaluates the consistency of the filter by analyzing the measurement residuals.

## System Description

The dynamics of the system are nonlinear and are defined by the following state transition function:

$$f(\mathbf{x}) = \begin{bmatrix} v_g \cos(\theta_g) \\ v_g \sin(\theta_g) \\ \frac{v_g}{L} \tan(\phi_g) \\ v_a \cos(\theta_a) \\ v_a \sin(\theta_a) \\ \omega_a \end{bmatrix},$$

where  $v_g$  and  $\phi_g$  are the velocity and steering angle of the UGV, and  $v_a$  and  $\omega_a$  are the velocity and angular rate of the UAV.

The measurement function is given by:

$$h(\mathbf{x}) = \begin{bmatrix} \text{wrapToPi} \left( \tan^{-1} \left( \frac{\eta_a - \eta_g}{\xi_a - \xi_g} \right) - \theta_g \right) \\ \text{wrapToPi} \left( \tan^{-1} \left( \frac{\eta_g - \eta_a}{\xi_g - \xi_a} \right) - \theta_a \right) \\ \sqrt{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} \\ \xi_a \\ \eta_a \end{bmatrix}.$$

## Measurement Model

### Nonlinear Measurement Function ( $h(\mathbf{x})$ )

The nonlinear measurement function is given by:

$$h(\mathbf{x}) = \begin{bmatrix} \text{wrapToPi} \left( \tan^{-1} \left( \frac{\eta_a - \eta_g}{\xi_a - \xi_g} \right) - \theta_g \right) \\ \text{wrapToPi} \left( \tan^{-1} \left( \frac{\eta_g - \eta_a}{\xi_g - \xi_a} \right) - \theta_a \right) \\ \sqrt{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} \\ \xi_a \\ \eta_a \end{bmatrix},$$

where:

- $(\xi_g, \eta_g)$ : Position of the UGV in 2D space.
- $(\xi_a, \eta_a)$ : Position of the UAV in 2D space.
- $\theta_g, \theta_a$ : Heading angles of the UGV and UAV, respectively.
- $\text{wrapToPi}$ : A function that ensures the angles remain within the range  $[-\pi, \pi]$ .

### Jacobian of the Measurement Function ( $H(\mathbf{x})$ )

The Jacobian of the measurement function is computed as:

$$H(\mathbf{x}) = \begin{bmatrix} -\frac{\eta_a - \eta_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & \frac{\xi_a - \xi_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & -1 & \frac{\eta_a - \eta_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & -\frac{\xi_a - \xi_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & 0 \\ \frac{\eta_a - \eta_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & \frac{\xi_a - \xi_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & 0 & -\frac{\eta_a - \eta_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & -\frac{\xi_a - \xi_g}{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2} & -1 \\ \frac{\xi_a - \xi_g}{\sqrt{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2}} & \frac{\eta_a - \eta_g}{\sqrt{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2}} & 0 & -\frac{\xi_a - \xi_g}{\sqrt{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2}} & -\frac{\eta_a - \eta_g}{\sqrt{(\xi_a - \xi_g)^2 + (\eta_a - \eta_g)^2}} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

## Linearized Kalman Filter (LKF)

The LKF approximates the nonlinear system by linearizing it about the current state estimate. The filter proceeds as follows:

### Prediction Step

$$\delta \hat{\mathbf{x}}_{k+1}^- = \mathbf{F}_k \delta \hat{\mathbf{x}}_k^+ + \mathbf{G}_k \delta \mathbf{u}_k, \quad (1)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k^+ \mathbf{F}_k^\top + \mathbf{Q}_k, \quad (2)$$

where  $\mathbf{F}_k$  is the Jacobian of the dynamics  $f(\mathbf{x})$ ,  $\mathbf{G}_k$  maps the control inputs, and  $\mathbf{Q}_k$  is the process noise covariance.

## Measurement Update Step

$$\delta \hat{\mathbf{x}}_{k+1}^+ = \delta \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\delta \mathbf{y}_{k+1} - \mathbf{H}_{k+1} \delta \hat{\mathbf{x}}_{k+1}^-), \quad (3)$$

$$\mathbf{P}_{k+1}^+ = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^-, \quad (4)$$

where  $\mathbf{K}_{k+1}$  is the Kalman gain:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^\top (\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^\top + \mathbf{R}_k)^{-1}.$$

## NEES and NIS Calculations

- The NEES (Normalized Estimation Error Squared) is given by:

$$\epsilon_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^\top \mathbf{P}_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k),$$

where  $\mathbf{x}_k$  is the true state and  $\hat{\mathbf{x}}_k$  is the estimated state.

- The NIS (Normalized Innovation Squared) is given by:

$$\nu_k = \mathbf{v}_k^\top \mathbf{S}_k^{-1} \mathbf{v}_k,$$

where  $\mathbf{v}_k$  is the innovation (measurement residual), and  $\mathbf{S}_k$  is the innovation covariance.

## Simulation and Results

The system trajectory is simulated using the nonlinear dynamics and the true noise covariances. Noisy measurements are generated at each time step, and the LKF is used to estimate the states.

### NEES and NIS Tests

The filter's performance is evaluated using 400 Monte Carlo simulations. Confidence bounds for NEES and NIS are calculated using the chi-squared distribution:

$$\text{NEES bounds: } \left[ \chi_{\alpha/2, n}^2, \chi_{1-\alpha/2, n}^2 \right] / N, \quad (5)$$

$$\text{NIS bounds: } \left[ \chi_{\alpha/2, p}^2, \chi_{1-\alpha/2, p}^2 \right] / N. \quad (6)$$

## Results

- **NEES:** The NEES evaluates whether the state estimation error is consistent with the filter's covariance.
- **NIS:** The NIS evaluates whether the innovation (residual) is consistent with the measurement noise covariance.

Both metrics are plotted against their respective confidence bounds, showing the filter's consistency.

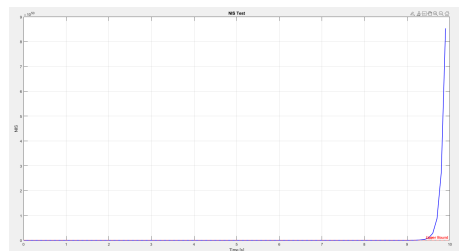


Figure 1: NIS mine

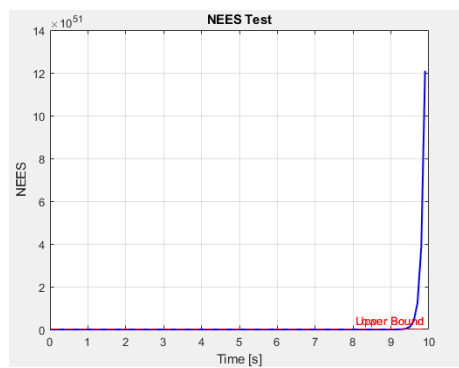


Figure 2: NEES mine