Strategies to Design Signals to Spoof Kalman Filter

Zhongshun Zhang, Lifeng Zhou and Pratap Tokekar

Abstract—We study the problem of designing spoofing signals to corrupt and mislead the output of a Kalman filter. Unlike existing works that focus on detection and filtering algorithms for the observer, we study the problem from the attacker’s point-of-view. In our model, the attacker can corrupt the measurements by adding spoofing signals. The attacker seeks to create a separation between the estimate of the Kalman filter with and without spoofing signals. We present a number of results on how to generate such spoofing signals, while minimizing the signal magnitude. The resulting algorithms are evaluated through simulations along with theoretical proofs.

I. INTRODUCTION

As autonomous systems proliferate, there are growing concerns about their security and safety [1], [2]. Of particular concern is their vulnerability to signal spoofing attacks [3]. As a result, many researchers are designing algorithms that enable an observer to detect and mitigate signal spoofing attacks (e.g., [4]–[8]). We study the problem from the opposite (i.e., the attacker’s) point-of-view. Our goal is to characterize the capabilities of the attacker that is generating the spoofing signals while assuming that the observer is using a Kalman filter for state estimation.

The problem of generating spoofing attacks has been studied specifically for GPS signals. Tippenhauer et al. [3] describe the requirements as well as present a methodology for generating spoofed GPS signals. Larcom and Liu [9] presented a taxonomy of GPS spoofing attacks.

The typical approach to mitigate sensor spoofing attacks is by designing robust state estimators [10]. Fawzi et al. presented the design of a state estimator for a linear dynamical system when some of the sensor measurements are corrupted by an adversarial attacker [11]. We focus on the scenario where the observer uses a Kalman Filter (KF) for estimating the state using measurements that are corrupted by additive spoofing signals by the attackers. We study the problem of generating spoofing signals of minimum energy that can achieve any desired separation between the KF estimate with spoofing and without spoofing. We show that for many practical cases, the spoofing signals can be generated using linear programming in polynomial time.

The work by Su et al. [12] is most closely related to ours. The authors show how to spoof the GPS signal without triggering a detector that uses the residual in the Kalman filter. They present a 1-step (greedy) online spoofing strategy that solves a linear relaxation of a Quadratically Constrained Quadratic Program (QCQP) at each timestep. We present a strategy that plans for $T$ future timesteps, instead of just the next timestep, while minimizing the spoofing signal energy. Furthermore, we characterize the scenarios under which our strategy finds the optimal solution in polynomial time.

Based on the motion model of the target and the evolution of the KF, three problems for spoofing design are formulated in Section II. Section III shows the approaches to solve these optimization problems. The simulations for verifying spoofing strategies are given in Section IV. Section V summarizes the conclusion and future work.

II. PROBLEM FORMULATION

Notation: We denote the set of positive real number by $\mathbb{R}^+$. The set of real vectors with dimension $n$ is denoted by $\mathbb{R}^n$, $n \in \mathbb{Z}^+$. The set of real matrices with $m$ rows and $n$ columns by $\mathbb{R}^{m \times n}$, $m, n \in \mathbb{Z}^+$. We write $\| \cdot \|_p$, $p \in \mathbb{Z}^+$ as the $p^{th}$ power of $L_p$ vector norm, $\mathbb{E}(\cdot)$ as the expectation of a random variable, $I_n$ as the identity matrix with size $n$, $n \in \mathbb{Z}^+$, and $N(\mu, \sigma^2)$ as the normal distribution with mean $\mu$ and variance $\sigma^2$.

We consider a scenario where an observer estimates the location of a target using a KF in 2D plane. The target misleads the observer by adding spoofing signals to the observer’s measurement. We define the target’s model as:

$$x_{t+1} = Fx_t + Gu_t + \omega_t,$$  \hspace{1cm} (1)

where $F, G \in \mathbb{R}^{2 \times 2}$, $x_t \in \mathbb{R}^2$ is the position of the target, $u_t \in \mathbb{R}^2$ is the control input and $w_t \sim N(0, R)$ is the Gaussian distribution, model noise of the motion model with $R \in \mathbb{R}^{2 \times 2}$.

The observer estimates the target’s position using linear measurement model:

$$z_t = Hx_t + v_t,$$  \hspace{1cm} (2)

where $H \in \mathbb{R}^{2 \times 2}$ and $v_t \sim N(0, Q)$ gives the measurement noise with $Q \in \mathbb{R}^{2 \times 2}$.

In order to mislead the observer, the target corrupts the observer’s measurement by adding spoofing signal to mislead the observer’s estimate. We assume the measurement received by the observer is $\hat{z}_t \in \mathbb{R}^2$ with spoofing signal (Equation (3)) instead of the true measurement $z_t \in \mathbb{R}^2$ without spoofing signal (Equation (2)). The spoofing signal $\epsilon_t := [\epsilon_{tx}, \epsilon_{ty}]^T \in \mathbb{R}^2$ adds additional measurement error:

$$\hat{z}_t = z_t + \epsilon_t.$$  \hspace{1cm} (3)

The observer uses a KF to estimate target’s position with initial distribution $N(m_0, \Sigma_0)$. Since it receives the spoofing measurement $\hat{z}_t$ for updating, we denote distributions generated by the evolution of its KF as $N(m_t, \Sigma_t)$ when step $t \geq$
where $\gamma_t \in \mathbb{R}^+$ is a weighing parameter and $T \in \mathbb{Z}^+$ is the optimization horizon.

B. Offline Spoofing Signal Design with Unknown $\mathcal{N}(m_0, \Sigma_0)$

Next we consider the case where the target does not know the initial condition in the KF. Instead we assume that the initial estimate $\hat{m}_0$, is not too far away from $m_0$ (in exception).

Problem 2 (Offline with Unknown $\mathcal{N}(m_0, \Sigma_0)$)
Consider a target with motion model (Equation (1)), measurement model (Equation (2)), and spoofing measurement model (Equation (3)). Assume the target starts spoofing with $\hat{m}_0$, where $\mathbb{E}(m_0 - \hat{m}_0) = M_0$ and $\Sigma_0 \neq \Sigma_0$. Find a sequence of spoofing signal inputs, $\{\epsilon_1, \epsilon_2, \cdots, \epsilon_T\}$ to achieve desired separation $d_t$ between $\hat{m}_t$ and $m_t$ (in expectation) at step $t$. Such that

$$\min \sum_{t=1}^{T} \gamma_t \cdot \|\epsilon_t\|_p^p$$

subject to,

$$\|m_t - \hat{m}_t\|_p^p \geq d_t^p, \ \forall t$$

where $\gamma_t \in \mathbb{R}^+$ is a weighing parameter and $T \in \mathbb{Z}^+$ is the optimization horizon.

III. Signal Spoofing Strategies

In this section, we show how to solve Problems 1 and 2 when $p = 1$ and $p = 2$. We first present the relationship between the separation $m_t - \hat{m}_t$ and the initial bias $m_0 - \hat{m}_0$.

Theorem 1 Consider a target with motion model (Equation (1)), measurement model (Equation (2)), and spoofing measurement model (Equation (3)). The evolutions of the KFs by applying $z_t$ and $\tilde{z}_t$ give the distributions $\mathcal{N}(m_t, \Sigma_t)$ and $\mathcal{N}(\tilde{m}_t, \tilde{\Sigma}_t)$, respectively. The difference, $m_t - \tilde{m}_t$ is,

$$m_t - \tilde{m}_t = \sum_{i=0}^{t-1} A_{t-i} \cdot (m_0 - \hat{m}_0) + \sum_{i=0}^{t-2} \left( \prod_{j=0}^{i} A_{t-j} \cdot (B_{t-1-i} + C_{t-1-i}) \right) + B_t + C_t,$$
Proof: From Equation 6, $E(m_t - \tilde{m}_t)$ follows,

$$E(m_t - \tilde{m}_t) = E \left( \sum_{i=0}^{t-2} \prod_{j=0}^{i} A_{t-j} \cdot B_{t-1-i} + \cdots \right)$$

The actual measurement is: $z_t = H(Fm_{t-1} + Gu_{t-1} + v_t) + \nu_t$, where $v_t$ and $\nu_t$ are Gaussian noises with zero mean. The expected measurement value is: $E(z_t) = H(Fm_{t-1} + Gu_{t-1})$ for all $i$, thus $E[z_t - H(Fm_{t-1} + Gu_{t-1})] = 0$. Since $E[B_t] = 0$, we have,

$$E(m_t - \tilde{m}_t) = \prod_{i=0}^{t-1} A_{t-i}E(m_0 - \tilde{m}_0) + \sum_{i=0}^{t-2} \prod_{j=0}^{i} A_{t-j} \tilde{K}_{t-1-i} \cdot \epsilon_{t-1-i} + \tilde{K}_t \epsilon_t.$$  

Since we assume $E(m_0 - \tilde{m}_0) = M_0$ in Problem 2, the claim is guaranteed.

Theorem 1 shows the difference between the two estimated means at step $t$ depends on the initial means, $m_0$ and $\tilde{m}_0$, and the initial covariance matrices $\Sigma_0$ and $\tilde{\Sigma}_0$. This is because the Kalman gain $K_t$ depends on the covariance matrix $\Sigma_t$. If target sets $m_0 = \tilde{m}_0$ and $\Sigma_0 = \tilde{\Sigma}_0$, it has $\Sigma_t = \tilde{\Sigma}_t$ for all $t$ since the covariance matrix is updated through the same Kalman prediction and update equation (see appendix). Thus, $B_t = 0_{2 \times 2}$ and then Equation (6) can be simplified as:

$$m_t - \tilde{m}_t = \sum_{i=0}^{t-2} \prod_{j=0}^{i} A_{t-j} \tilde{K}_{t-1-i} \cdot \epsilon_{t-1-i} + C_t.$$  

As a result, $m_t - \tilde{m}_t$ is independent of the measurements $\{z_1, z_2, \cdots, z_t\}$ when $m_0 = \tilde{m}_0$ and $\Sigma_0 = \tilde{\Sigma}_0$. Thus, the target can generate spoofing signal inputs by solving Problem 1 offline. Similarly following Corollary 1, Problem 2 can be solved offline as well.

Problems 1 and 2 are two nonlinear programming problems for arbitrary vector norms $L_p$. However, when $p = 1$, they can be formulated as linear programming problems. Linear programming can be solved in polynomial time [13]. When $p = 2$, they become QCQP (Quadratically Constrained Quadratic Program). The following shows the LP and QCQP formulations.

**Theorem 2** If $p = 1$ and the elements in $F$ and $I - K_tH$ are all positive, then Problems 1 and 2 can be solved optimally with linear programming. If $p = 1$ and the elements in $F$ and $I - K_tH$ are not all positive, then Problems 1 and 2 can be solved optimally with $4^k$ linear programming instances. If $p = 2$ and $\{H, F, Q, R\}$ are diagonal matrices, then Problems 1 and 2 can be solved optimally with linear programming.

**A. Linear Programming Formulation for $L_1$ Vector Norm**

Here, we show how to formulate Problem 1 using linear programming. A similar procedure can be applied to formulate Problem 2 as linear programming.

The constraint in Problem 1 (Equation 4) follows:

$$\|m_t - \tilde{m}_t\|_1 \geq \alpha_0 + a_1 \epsilon_{ix} + \cdots + b_{2t} \epsilon_{iy}$$

where $t = 1, 2, \cdots, T$ and $\prod_{j=0}^{i} A_{t-j} \cdot K_{t-1-i} \cdot \epsilon_{t-1-i} + \tilde{K}_t \epsilon_t.$

Thus, in the worst case, the optimal solution can be obtained by solving $4^k$ linear optimization problems. We run Algorithm 1 $4^k$ times by changing the sign of $\{H, F, Q, R\}$.
Algorithm 1: Linear Programming Formulation

1 Initial ← \{ (x_0, \Sigma_0, F, H, G, Q, R, u) \}
2 G ← 0_{k \times 2T}
3 Calculate Kalman gain \( \tilde{K}_1, \cdots, \tilde{K}_T \)

4 for \( q = 1 : k \)
5 for \( i = 1 \) to the \( q_{th} \) value in \( D_k \) /Equation (17)
6 \( g = \prod_{j=i}^{T-1} A_j + \tilde{K}_i \)
7 \( G_{q,i} = \text{sum of all rows in } g \)
8 \textbf{Return } \( G \)

of rows in \( g \) (Line 6) appropriately.

B. Quadratically Constrained Quadratic Program Formulation for \( L_2 \) Vector Norm

When \( p = 2 \), Problems 1 and 2 can be formulated as QCQPs [14]:

\[
\min \frac{1}{2} x^T P_0 x
\]
\[
\text{s.t. } -\frac{1}{2} x^T D_t^T D_t x + d_t^2 \leq 0, \quad \forall t \in \{1, \ldots, T\} \quad (11)
\]

where \( x = [\epsilon_1^2, \epsilon_2^2, \cdots, \epsilon_T^2] \), \( P_0 = I_{2T} \), and

\[
D_t = \begin{bmatrix}
\prod_{j=1}^{t-1} A_j + \tilde{K}_0 & 0 & 0 & 0 \\
\vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & \tilde{K}_t \\
0 & \cdots & 0 & 0 \\
0 & \cdots & 0 & 0 \\
\end{bmatrix},
\]

Unfortunately, the QCQP formulations for these three problems are NP-hard since the constraint in each problem is concave. If \( F, G, H, \Sigma_0 \) are diagonal matrices, it can be shown that \( D_t \) is also a diagonal matrix. We can transform the QCQP formulation to a linear programming problem by using change of variables \( \{\epsilon_1^2, \cdots, \epsilon_T^2\} \) and using a procedure similar to p.

If \( D_t \) is not a diagonal matrix, one solution is to apply the inequality \( \sqrt{2} \|x\|_2 \geq \|x\|_1 \) between \( L_1 \) vector norm and \( L_2 \) vector norm.

C. Receding Horizon: Spoofing with online measurement

Problems 1 and 2 describe the offline versions for spoofing. We also extend the offline problems to an online version. The following formulates an online spoofing scenario.

Consider a target with motion model (Equation (1)), measurement model (Equation (2)), and spoofing measurement model (Equation (3)). Assume the target does not know \( \mathcal{N}(x_0, \Sigma_0) \). It collects a series of measurements \( \{z^1_{\text{real}}, z^2_{\text{real}}, \cdots, z^t_{\text{real}}\} \) from step 1 to current step \( t^o \). Find a sequence of spoofing signal inputs, \( \{\epsilon_{t^o}, \epsilon_{t^o+1}, \cdots, \epsilon_{t^o+H}\} \)

\[
\text{to achieve desired separation } d_t \text{ between } \tilde{m}_t \text{ and } m_t \text{ (in expectation) within future } H \text{ steps. Such that}
\]

\[
\min \sum_{t=t^o}^{t^o+H} \gamma_t \cdot \|\epsilon_t\|_p^p
\]
\[
\text{s.t. } \| (m_t - \tilde{m}_t) \|_p^p \geq \gamma_t, \quad \forall t \in \{t^o, \cdots, t^o + H\} \quad (12)
\]

where \( \gamma_t \in \mathbb{R}^+ \) is a weighing parameter, \( t^o \) is the current time, and \( H \) is the predictive time horizon. The target applies \( \epsilon_t = \epsilon_{t^o} \) as spoofing signal input at each step \( t \).

IV. SIMULATIONS

In this section, we simulate the effectiveness of spoofing strategies for Problems 1, 2 and online case (Section III-C) where a target designs spoofing signals \( \epsilon_t \) to mislead an observer by achieving the desired separations between \( m_t \) and \( \tilde{m}_t \). Our code is available online.\(^1\)

We consider the \( L_1 \) vector norm and the following models,

\[
F = I_{2x2}, G = I_{2x2}, u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, R = 0.5I_{2x2}, Q = 0.5I_{2x2}.
\]

Set the weight \( \gamma_t = 1 \) for all \( t \).

For Problem 1, set the initial condition for the KF as,

\[
\Sigma_0 = I_{2x2}, \quad m_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\]

Since the target knows \( \mathcal{N}(x_0, \Sigma_0) \), it sets \( \tilde{m}_0 = m_0 \) and \( \tilde{\Sigma}_0 = \Sigma_0 \). We first consider a scenario where the target wants to achieve the desired separation at steps, \( t = 5, 10, 15 \), denoted as \( d_5 = 1.77, d_{10} = 3.54 \) and \( d_{15} = 5.30 \) with the optimization horizon \( T = 20 \). The target generates a sequence of spoofing signals \( \{\epsilon_1, \cdots, \epsilon_{20}\} \) offline by using a linear programming solver. The spoofing performance is shown in Figure 3-(a) where the true separations are the same as the desired separations. Same successful spoofing achieved when the desired separations are chosen as \( d_1 = 0.25\|u\|_2, t = \{3, \cdots, 15\} \), as shown in Figure 3-(b).

In Problem 2, the target knows \( \mathcal{N}(m_0, \tilde{\Sigma}_0) \) but does not know \( \Sigma_0 \). The spoofing result is no longer deterministic but holds in expectation \( \|E(m_t - \tilde{m}_t)\|_1 \geq d_t \), Figure 4-(a) shows spoofing signals for desired separations as \( d_1 = 2 \) with \( T = 6 \) and \( M_0 = 1 \). Set \( \mathcal{N}(\tilde{m}_0, \Sigma_0) \) as \( \mathcal{N}(0, 1.5I_2) \), \( m_0 \) as a random variable (\( m_0 \sim \mathcal{N}(1, 1) \)) and \( \Sigma_0 = I_2 \). In order to see the effectives of the spoofing signals \( \{\epsilon_1, \cdots, \epsilon_5\} \), we conduct 100 trials for each desired separation \( d_2 \in \{1, 2, 3, 4, 5\} \). Figure 4-(b) shows the \( \|m_1 - \tilde{m}_1\|_1 \) is no longer deterministic, but \( \|E(m_1 - \tilde{m}_1)\|_1 \) is close to the desired value \( d_1 = 2 \).

For online case, spoofing signals are continuously generated by using receding horizon optimization with new noisy measurements. We set the receding horizon as \( H = 15 \). Even though offline strategy performs comparatively as online strategy (Figure 5), online spoofing strategy achieves almost the same separation as the desired, while offline strategy has certain divergence (Figure 6). This is because online strategy can update the measurement at each step. Figure 7 shows the online strategy applies less total spoofing magnitude than offline strategy.

\(^1\)https://github.com/raaslab/signal_spoofing.git
We study the problem of spoofing signals to achieve any desired separation between a Kalman filter estimate without and with spoofing signals. Our main approach was to formulate the problems as nonlinear, constrained optimization problems in order to minimize the energy of the spoofing signal. We show that under some technical assumptions, the problems can be solved by linear programming optimally. We also present a more computationally expensive approach to solving the problem, without the aforementioned assumptions.

Our immediate future work is to study the game-theoretic aspects of the problem. In this work, we did not consider

![Fig. 3. Offline signal spoofing with known $(m_0, \Sigma_0)$.](image)

![Fig. 4. Offline signal spoofing with unknown $(m_0, \Sigma_0)$.](image)

![Fig. 5. Online spoofing and offline spoofing with unknown $(m_0, \Sigma_0)$.](image)

![Fig. 6. Divergence caused by online spoofing and offline spoofing. The blue line denotes $(\|m_t - \tilde{m}_t\|_1 - d_t)$ for online spoofing, and red line denotes $(\|\tilde{m}_t - m_t\|_1 - d_t)$ for offline spoofing.](image)

![Fig. 7. Total spoofing energy injected by online and offline strategies.](image)
any active strategy being employed by the observer to detect and/or mitigate the attack. In future works, we will consider the case of designing spoofing signals that explicitly take the attack detection and/or mitigation strategies into account. In all the problems considered in this paper, the desired separations are taken as inputs provided by the user. Instead, we can optimize over the desired separation trajectory in order to avoid detection by the observer.

APPENDIX

A. Proof of Theorem 1

Before we prove Theorem 1, we review the Kalman Filter update equations. Suppose the true measurement is \( z_t \), theKF estimation is:

\[
x_{t|t-1} = F x_{t-1|t-1} + G u_t, \tag{13}
\]

\[
x_{t|t} = F x_{t|t-1} + K_t (z_t - H x_{t|t-1}), \tag{14}
\]

where \( K_t \) is the Kalman gain and is given by:

\[
K_t = (F \Sigma_{t|t-1} F^T + R_t H^T (H \Sigma_{t|t-1} H^T + Q_t)^{-1}. \tag{15}
\]

According to the Kalman gain update equation (15), the evolution covariance matrix at step \( t \), \( \Sigma_t \), only depends on the state model parameters and the initial condition of the covariance matrix \( \Sigma_0 \). The Kalman gain at step \( t \), \( K_t \) depends on the covariance matrix \( \Sigma_t \). Both \( \Sigma_t \) and \( K_t \) do not depend on the control input series \( \{u_t\}_{t=1}^k \), measurement \( \{z_t\}_{t=1}^k \). Thus, the covariance matrix and the Kalman gain can be predicted from the Kalman covariance update steps.

\[
\Sigma_{t+1|t} = F \Sigma_{t|t} F^T + R_t,
\]

\[
\Sigma_{t+1|t+1} = (I - K_t H) \Sigma_{t+1|t}. \tag{16}
\]

From Equation (16), the Kalman gain can be predicted from the initial condition \( \Sigma_0 \).

We now prove our main result.

\textbf{Proof:} From the update of theKF, we have

\[
m_t = m_{t|t-1} + K_t (z_t - H m_{t|t-1})
\]

\[
= (I - K_t H) m_{t|t-1} + K_t z_t \tag{17}
\]

and

\[
\hat{m}_t = (I - K_t H)(F m_{t-1} + G u_{t-1}) + K_t (z_t + \epsilon_t). \tag{18}
\]

Recursively,

\[
m_t - \hat{m}_t = (I - K_t H)(F m_{t-1} + G u_{t-1}) + K_t z_t
\]

\[
- [(I - K_t H)(F \hat{m}_{t-1} + G u_{t-1}) + \hat{K}_t (z_t + \epsilon_t)]
\]

\[
= (F - K_t H F) m_{t-1} - (F - K_t H F) \hat{m}_{t-1}
\]

\[
- (K_t - \hat{K}_t) G u_{t-1} + [K_t z_t - \hat{K}_t (z_t + \epsilon_t)]
\]

\[
= (F - K_t H F) m_{t-1} - (F - K_t H F) \hat{m}_{t-1}
\]

\[
- (K_t - \hat{K}_t) G u_{t-1} + (K_t - \hat{K}_t) z_t - \hat{K}_t \epsilon_t
\]

\[
= (F - K_t H F) (m_{t-1} - \hat{m}_{t-1})
\]

\[
+ (K_t - \hat{K}_t) [z_t - H (F m_{t-1} + G u_{t-1})] - \hat{K}_t \epsilon_t.
\]

Define, \( A_t = F - K_t H F, \ B_t = (K_t - \hat{K}_t) [z_t -

\mathcal{H}(F m_{t-1} + G u_{t-1})] \) and \( C_t = -\hat{K}_t \epsilon_t \). Then, \( m_t - \hat{m}_t = A_t (m_{t-1} - \hat{m}_{t-1}) + B_t + C_t \)

\[
= A_t [A_{t-1} (m_{t-2} - \hat{m}_{t-2}) + B_{t-1} + C_{t-1}] + B_t + C_t
\]

\[
= \prod_{i=0}^{t-1} A_{t-i} \cdot (m_0 - \hat{m}_0) + (B_t + C_t)
\]

\[
+ A_t (B_{t-1} + C_{t-1}) + \cdots + A_t \cdots A_2 A_1 (B_1 + C_1)
\]

\[
= \prod_{i=0}^{t-1} A_{t-i} \cdot (m_0 - \hat{m}_0) + B_t + C_t
\]

\[
+ \sum_{i=0}^{t-2} \left( \prod_{j=0}^{i} A_{t-j} (B_{t-i-j} + C_{t-i-j}) \right).
\]

\[\Box\]

REFERENCES


