2017 IEEE Conference on Control Technology and Applications (CCTA)
August 27-30, 2017. Kohala Coast, Hawai'i, USA

Secure Reference-Tracking with Resource-Constrained UAVs

Vaibhav Katewa, Rajasekhar Anguluri, Akila Ganlath, and Fabio Pasqualetti

Abstract— **In this paper we study a security problem for resource-constrained autonomous systems. We consider a UAV tasked with tracking a reference trajectory, and an attacker capable of compromising the measurements taken at certain sensors. We consider a probabilistic attack model, where the attacker executes denial of service attacks against a subset of sensors based on a Bernoulli process. We assume that sensors have different accuracy, reliability, and require different computational times to be activated. Our approach is based on the formalism of Markov Jump Linear Systems. We develop and numerically validate an optimal security countermeasure that probabilistically selects which sensor to use at different time instants, so as to balance performance and security, and ultimately minimize the UAV's expected tracking error.**

I. Introduction

Cyber-physical systems in general, and autonomous robots in particular, are prone to a variety of failures and intentional attacks. Recent studies and incidents have demonstrated how GPS jamming, compromising message integrity, and denial of service attacks, among others, can deteriorate the performance of a cyber-physical system often impeding a controller to restore nominal operation [1], [2]. While attack detection and identification methods have been proposed for a variety of scenarios [3], the problem of ensuring satisfactory performance in an unsecured environment remains largely unsolved.

Ensuring security and satisfactory performance in real-time and resource-constrained cyber-physical systems is increasingly challenging because the platform can reserve only limited computational resources and time for security and control purposes [4]. In such scenarios, the control and security tasks usually compete with each other for limited resources. Therefore, besides being computationally efficient, viable security algorithms must coordinate with the control algorithm to schedule and divide the resources among them in an optimal manner. Through this study, we aim to characterize this optimal trade-off between security and control performance.

In this paper, we propose and evaluate a security method for a resource-constrained Unmanned Aerial Vehicle (UAV). UAVs are being increasingly used in military and civilian scenarios for a wide variety of applications such as surveillance, reconnaissance among others. For these applications, the UAV needs to estimate its position over time. These estimates are typically provided by an on-board Global Positioning System (GPS) unit and are then used to perform a desired control task, such as hovering or trajectory tracking. The measurements provided by the GPS are accurate and require limited computational resources [5]. Besides the GPS unit, the UAV is also usually equipped with a secondary sensor(s) that can provide position information. For instance, the UAV can have an on-board vision camera that takes ground images and processes them to detect certain features on the ground. It then compares these features with an environment map (we assume that UAV knows the map) to estimate its position [4], [6].

Due to ambient light conditions and map variations, the position estimates provided by the camera are usually less accurate when compared to the GPS unit [4], [6]. Further, obtaining the estimates from the camera involves the use of image processing and feature extraction algorithms, which require significant computational resources [7]. Usually, the UAV has limited computational capabilities due to hardware limitations and as a result, the secondary measurements are delayed or not available at every time instant. Due to these limitations, secondary sensors are preferred less and the measurements are obtained using the GPS.

However, a drawback of using the GPS estimates is that it is susceptible to attacks by an adversary. For example, an attacker can cause a Denial of Service (DoS) attack by transmitting a jamming signal which prevents the UAV from obtaining the GPS measurements. These attacks may significantly affect the control task of the UAV, and in extreme scenarios can lead to catastrophic events like UAV crash [8]. In such scenarios, secondary sensors can be leveraged to mitigate the adverse effects of the attacks. Although they have limitations described above, they are secure and can be useful in presence of such attacks. In this paper, we address the issue of how to use these two sensors in an optimal manner such that the adverse effects of the attack are minimized.

Related Work Analysis and detection of DoS attacks in cyber-physical systems is a rich field and it has been studied in deterministic [3] and stochastic settings [1], [2]. In [2], the authors present an optimal linear control in presence of random DoS attacks. In [1], an optimal jamming attack policy is derived that causes maximum system degradation. There have also been numerous studies on sensor selection/scheduling problems for estimation, both for stochastic [9] and deterministic [10] policies. The work closest to our paper is [9], wherein an optimal Markov stochastic policy is derived numerically that minimizes the estimation error. However, our problem deals with security against DoS attacks and studying the security vs performance trade-off. We use the theory of Markov Jump Linear Systems (MJLS),

This material is based upon work supported by ONR award N00014-14-1-0816. The authors are with the Department of Mechanical Engineering, University of California at Riverside, {vkatewa, rangu003, aganl001, fabiopas}@engr.ucr.edu.

which is an advanced field (see [11] and references therein).

The main contributions of the paper are as follows. We study the trade-off between security and control performance in a resource-constrained UAV in presence of DoS attacks. We formalize this trade-off as a stochastic sensor selection problem based on an underlying Markov chain which models the attacks and sensor computational times. We consider a LQG tracking problem and solve it to obtain the optimal control laws and characterize the tracking performance of the UAV. Finally, using numerical techniques, we obtain the optimal sensor selection policy that minimizes the expected tracking error.

Notations A positive definite (positive semi-definite) matrix is denoted by $Q > 0 (Q \geq 0)$. For $x \in \mathbb{R}^n$ and $Q \geq 0$, $||x||_Q^2 \triangleq x^{\mathsf{T}} Q x$. I_N denotes the $N \times N$ identity matrix. $0_{N\times M} \in \mathbb{R}^{N\times M}, 0_n \in \mathbb{R}^n (1_{N\times M}, 1_n)$ denote the all zero (one) matrix. y_i^j denotes the sequence $y_i, y_{i+1}, \cdots, y_j$ for $i \leq j$ and y_i^∞ denotes the infinite sequence $y_i, y_{i+1}, \cdots$.

II. System and Attack Model

Consider a UAV equipped with two sensors - a GPS receiver and a vision camera, denoted by S_1 and S_2, respectively. Although the dynamics of the UAV is non-linear and evolves in continuous time, for the purpose of this paper we consider the operation of the UAV in the vicinity of an equilibrium point via linearization. Further, since the control signal to the UAV is applied at discrete time instants, we discretize the continuous-time dynamics. The details of UAV model, linearization and discretization are presented in subsection V-A. The resulting discrete-time linear time-invariant (LTI) system is denoted as

$$x(k+1) = Ax(k) + Bu(k) + w(k), \tag{1}$$

where $x \in \mathbb{R}^n$, and $u \in \mathbb{R}^l$ are the state and control input, respectively and $w(k)$ is an i.i.d. Gaussian process noise with zero mean and variance $W > 0$. The initial condition $x(0) = x_0$ is Gaussian with zero mean and variance $X_0 \geq 0$.

We model the measurements provided by the GPS and camera sensor as a linear function of the state corrupted by additive Gaussian noise. Although more detailed/non-linear measurement models may exist, we focus on the linear model in this paper. The measurements obtained by S_1 (GPS) are thus given by

$$y_1(k) = Cx(k) + V_1 v(k), \tag{2}$$

where $y_1 \in \mathbb{R}^b$. $v(k) \in \mathbb{R}^b$ is the measurement noise, which is i.i.d. Gaussian with zero mean and variance I_b, and $V_1 > 0$. Thus, the effective noise variance for the GPS is V_1. We assume that the initial condition x_0, and the noises $w(k)$ and $v(k)$ are independent of each other. We make the following assumption on the linearized UAV model.

A1) (A, B) is controllable and (A, C) is observable [12].

As discussed in the introduction, obtaining measurements from S_2 (camera) requires significant computational resources. Due to computational limitations of the UAV, it results in high processing time and limited measurements. We model this fact by assuming that S_2 requires $D + 1$ time steps to provide a single measurement where $D \geq 0$ is the wait time in obtaining the measurement. Note that the wait time does not correspond to a delay in receiving the measurements. The measurements are not delayed, rather they are provided once every $D + 1$ time steps:

$$y_2(k) = \begin{cases} Cx(k) + V_2 v(k), & k = m(D+1), m = 1, 2, \ldots, \\ \phi_y, & \text{otherwise}, \end{cases} \tag{3}$$

where ϕ_y denotes that measurement is not received (which is equivalent to receiving pure noise). We assume that the two sensor clocks are synchronized. Further, we model the fact that the position estimates provided by the camera are less accurate than the GPS by assuming that

A2) $V_1 < V_2$, i.e. the camera measurements (when obtained) are more noisy than the GPS measurements.

Moreover, due to significant processing time and memory requirements in obtaining the measurement from S_2, we assume that:

A3) During the D-wait time period in S_2, sensor S_1 cannot be used to take a measurement.

This implies that at any given time instant, only one sensor can be used.

Control Task The control objective of the UAV is to track a pre-specified reference trajectory denoted by $d_0^\infty = d_0, d_1, \ldots$ There are a variety of control techniques to perform trajectory tracking [4], including Linear-Quadratic Gaussian (LQG) control, H_∞ control, and model predictive control. In this paper we focus on LQG-based tracking problem and present its detailed analysis and solution later in Section IV. Note that in determining the optimal control inputs, the controller needs to estimate the state of the UAV using the noisy measurements. We can readily observe from (2), (3) and Assumption **A2** that sensor S_1 will provide more accurate estimates when compared to S_2. Thus, if S_1 and S_2 are equally reliable, S_1 will be used at each time instant to receive the measurements. However, this is not true in presence of attacks, as described next.

Attack Model We consider Denial-of-service (DoS) attacks in which the attacker jams the GPS signal received by the UAV. In practice, the degradation due to DoS attacks is incremental w.r.t. the attack intensity, until a critical threshold beyond which the measurements are not received. We consider only this extreme case, assuming that when the attack is in progress no measurements are available from the GPS sensor. The attack process is denoted by $\{a_k\}$, where $a_k = 1$ (respectively $(a_k = 0)$) implies that the attack is in progress (respectively not in progress) at time k. The measurements from S_1 under the attack model are denoted by y_1^a, and can be described as

$$y_1^a(k) = \begin{cases} y_1(k), & \text{if } a_k = 0, \\ \phi_y, & \text{if } a_k = 1. \end{cases} \tag{4}$$

We consider a stochastic attack model where $\{a_k\}$ is an i.i.d. Bernoulli random process with parameter α. Thus,

$\mathbb{P}(a_k = 1) = \alpha$ is the probability of attack at time instant k. Many applications in control, communication and computer networks deal with Bernoulli packet drops. Therefore, although different/more sophisticated DoS attack models may be possible, we focus on this simplistic model for the paper. The attacker can choose the attack probability α based on the extent of desired system degradation and its willingness to get detected. Moreover, the UAV can estimate α empirically based on the number of measurements received in a given time window. Finally, we assume that the camera sensor S_2 is secure against the attacks.

Unlike in the absence of attacks, using S_1 at all times may result in considerable degradation of the tracking performance, particularly when the the attack probability is high. On the other hand, using S_2 may not be desirable if the wait time D in obtaining the measurements is large or $V_1 \ll V_2$. Thus, a security policy is needed to select among the two sensors in an optimal manner over time so as to maximize the UAV's tracking performance.

III. Sensor Selection Policy

In this section, we describe the sensor selection policy which determines which sensor to use and when. We then characterize the system performance for the given policy in terms of the expected LQG tracking cost.

Note that due to assumption **A3**, we cannot use both the sensors at the same time and this rules out any sensor fusion techniques. We consider a stochastic sensor selection policy in which, at each time instant, one of the two sensors is randomly selected according to a given probability distribution. This distribution is drawn from an underlying time-homogeneous Markov Chain (MC) as described in Fig. 1. States 1 and 2 of the MC correspond to sensor S_1 being in the state of no-attack and attack, respectively. States $3, 4, \ldots, D+3$ correspond to sensor S_2. Recall that sensor S_2 requires a processing time of $D+1$ to provide a measurement. This D-wait period is modelled by states $3, 4, \ldots, D+2$ in which no measurement is received. After the wait period, S_2 finally provides a measurement in state $D+3$. Assumption **A3** implies that switching from S_2 to S_1 is not allowed during the wait period (i.e. from states $3, 4, \ldots, D+2$). After S_2 provides a measurement in state $D+3$, we can either switch to S_1 at the next time instant or select S_2 again. If S_1 is selected, the MC enters state 1 or 2 depending on the absence/presence of attack. On the other hand, if S_2 is selected, it provides the next measurement after a wait time of D, and the policy continues. The overall transition probabilities from sensor S_1 to sensor S_2, and vice-versa are denoted by p and q, respectively.

Remark 1: ***(Alternative sensor selection policies)*** One may also consider alternative sensor selection policies, for example, a deterministic policy that periodically switches between the two sensors. However, because we focus on probabilistic attacks, stochastic policies are a natural choice. Further, stochastic policies are easy to implement and typically more tractable to optimize when compared to deterministic policies [9]. □

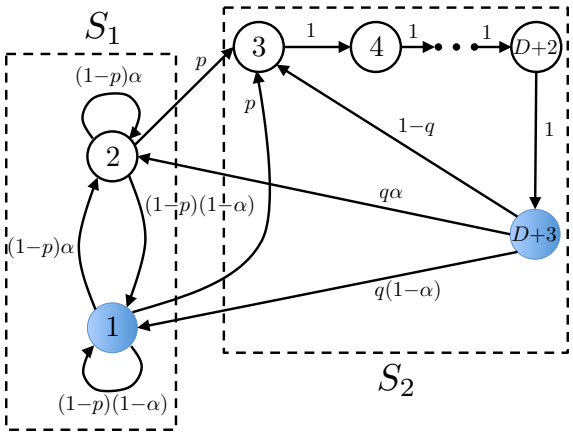


Fig. 1: Markov chain representation for the sensor selection policy. Measurements are received only in blue states.

Let $\mathbb{M}$ and r_k denote the Markov chain and its random state at time instant k, respectively. Let $\mathcal{M} \triangleq \{1, 2, \cdots, D+3\}$ be the state space of $\mathbb{M}$, and let $m = D+3$ denote its total number of states. The transition probability matrix of $\mathbb{M}$ is denoted by $P = [p_{ij}] \in \mathbb{R}^{m \times m}$, where $p_{ij} \triangleq \mathbb{P}(r_{k+1} = j / r_k = i)$ denotes the probability of transition from state i to j. From Fig. 1, we have

$$P = \left[\begin{array}{ccc:c} (1-p)(1-\alpha) & (1-p)\alpha & p & 0_{2\times D} \\ (1-p)(1-\alpha) & (1-p)\alpha & p & \\ \hdashline & 0_{D\times 3} & & I_D \\ \hdashline q(1-\alpha) & q\alpha & 1-q & 0_D^{\mathsf{T}} \end{array}\right]. \quad (5)$$

Note that the Markov chain is completely specified by the parameters $\{p, q, \alpha, D\}$.

Let $\pi_i(k) \triangleq \mathbb{P}(r_k = i)$ be the probability of the Markov chain being in state i at time k and let $\pi(k) = [\pi_1(k), \pi_2(k), \cdots, \pi_m(k)]^{\mathsf{T}}$. Then, $\pi(k) = (P^{\mathsf{T}})^k \pi_0$, where π_0 is the initial distribution of $\mathbb{M}$. It can be easily observed that $\mathbb{M}$ is irreducible and aperiodic for all values of $0 < p < 1$, $0 < q < 1$, and $0 \leq \alpha \leq 1$, and therefore it is ergodic. Thus, a stationary distribution exists and is denoted by $\pi_i \triangleq \lim_{k\to\infty} \pi_i(k)$.

Remark 2: ***(Generalization for multiple sensors)*** Our framework (UAV model and the MC) is not restrictive to two sensors and can be easily generalized to include scenarios with more than two sensors and with any combination of attacks/wait times affecting them. However, for ease of understanding, we present our results for two sensor case.□

Since the sensor switching policy is Markovian and the sensor attacks and wait times are incorporated into the Markov chain, our system belongs to the general class of Markovian Jump Linear Systems (MJLS) [11], and it can be described as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k), \\ y(k) &= C_{r_k} x(k) + R_{r_k} v(k), \end{aligned} \quad (6)$$

where C_{r_k}, R_{r_k} depend on the state r_k of $\mathbb{M}$. Recall that measurements are received only in states 1 (from sensor S_1) and m (from sensor S_2). Thus, $C_1 = C_m \triangleq C \in \mathbb{R}^{b \times n}$

and $R_1 = V_1$, $R_m = V_2$. Further, no measurements (ϕ_y) are received in states $2, 3, \cdots, m-1$. We model this by letting $C_i = 0_{b\times n}$, and $R_i = \sigma^2 I_b$ for $i = 2, 3, \cdots, m-1$, where $\sigma^2 >> 1$. In other words, absence of measurement is equivalent to receiving a measurement containing only noise. Next, we study the LQG problem for trajectory tracking and obtain the optimal control inputs and the tracking cost.

IV. Trajectory Tracking via LQG Control

In this section, we formally define the LQG tracking problem and solve it to obtain the optimal control inputs and the resulting tracking cost in terms of the transition probabilities p and q. The control task of the UAV is to track the reference trajectory d_0^∞ by applying appropriate inputs. We make the following assumption:

A4) The reference trajectory d_0^∞ is bounded.

To design the control inputs, we assume that the controller has access to past inputs, Markov chain states, measurements and the reference trajectory. Let this information be denoted by $\mathcal{I}_k \triangleq \{y_0^{k-1}, r_0^{k-1}, u_0^{k-1}, d_0^{k+1}\}$, and let $\mathcal{U}_k$ denote the class of control inputs at time k which use the information $\mathcal{I}_k$ in determining the control values. Then, the optimal trajectory tracking problem can be defined as the following infinite-horizon LQG minimization problem:

$$\begin{aligned}&\min_{u_0^\infty, p, q} \quad J(p, q, u_0^\infty)\\ &\triangleq \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{k=0}^{T} ||x(k) - d_k)||_Q^2 + ||u(k)||_R^2\right], \quad (7)\\ &\text{s.t.} \quad 0 \le p \le 1, \quad 0 \le q \le 1, \quad u(k) \in \mathcal{U}_k,\end{aligned}$$

where d_k is the reference at time step k, $Q > 0$ and $R > 0$ are the weighing matrices for the tracking error and input magnitude, respectively, and the expectation is taken with respect to the process noise w_0^T, the measurement noise v_0^T and Markov process r_0^T. Notice that the cost J in (7) depends on the parameters p, q, α, D. However, we highlight its dependence only on the transition probabilities p and q because these are our design parameters.

It can be easily observed that the minimization in (7) can be performed in two steps. First, we obtain the optimal control laws that minimize the cost for any given p and q, since the choice of optimal control law is independent of the transition probabilities. Next, we minimize the residual cost with respect to p and q to obtain the optimal selection policy.

Consider a Markov chain with given transition probabilities p and q, attack probability α and wait time D. To obtain the infinite-horizon optimal inputs, we first solve the following finite-horizon LQG tracking problem

$$\begin{aligned}\min_{u_0^N} J_N(p, q, u_0^N) \triangleq \mathbb{E}\Big[\sum_{k=0}^{N} \big[||x(k) - d_k||_Q^2 + ||u(k)||_R^2\big]\\ + ||x(N+1) - d_{N+1}||_{Q_{N+1}}^2\Big], \quad (8)\\ \text{s.t.} \quad u(k) \in \mathcal{U}_k,\end{aligned}$$

where $Q_{N+1} > 0$ denotes the terminal cost. Since the measurements are noisy, the controller implements a Kalman Filter (KF) to calculate the state estimates. Let the Minimum Mean Square Error (MMSE) estimate of state $x(k)$ computed by the KF based on the information $\mathcal{I}_k$ be denoted by $\hat{x}(k)$. Further, let $\Pi_k(p, q) \triangleq \mathbb{E}[||x(k) - \hat{x}(k)||^2]$ denote the estimation error covariance, where the expectation is taken with respect to to $w_0^{k-1}, v_0^{k-1}, r_0^{k-1}$. Note that the estimation error covariance also depends on the attack probability α and wait time D. However, we omit that dependence in the notation of Π_k.

Although the LQG tracking problem is well studied in literature [13], a consolidated result for MJLS is lacking. Thus, for completeness, we present the result and its proof using dynamic programming technique.

Lemma 4.1: ***(Finite-horizon optimal control)*** Consider the MJLS in (6). The optimal control inputs that minimize the cost in (8) are given by

$$u_N^*(k) = -L_{k+1}^{-1} B^\mathsf{T}(S_{k+1}(A\hat{x}(k) - d_{k+1}) + q_{k+1}),$$

where S_k evolves according to the following backward Riccati recursion

$$S_k = A^\mathsf{T} S_{k+1} Z_{k+1} A + Q, \quad \text{where} \quad (9)$$

$$Z_{k+1} \triangleq I_n - B L_{k+1}^{-1} B^\mathsf{T} S_{k+1}, L_{k+1} \triangleq B^\mathsf{T} S_{k+1} B + R, \quad (10)$$

$S_{N+1} = Q_{N+1}$, and q_k follows the backward recursion

$$\begin{aligned}q_k &= A^\mathsf{T} Z_{k+1}^\mathsf{T}(q_{k+1} + S_{k+1}\tilde{d}_{k+1}), \quad \text{where}\\ \tilde{d}_{k+1} &= A d_k - d_{k+1},\end{aligned} \quad (11)$$

with $q_{N+1} = 0_n$. Further, the resulting optimal cost is

$$\begin{aligned}J_N^*(p, q) &= tr(S_0 X_0) + d_0^\mathsf{T} S_0 d_0 - 2q_0^\mathsf{T} d_0 + c_0\\ &\quad + \sum_{k=1}^{N+1} tr(Y_k^\mathsf{T} L_k Y_k \Pi_{k-1}(p, q)) + tr(S_k W),\\ \text{where} \quad Y_k &\triangleq L_k^{-1} B^\mathsf{T} S_k A, \end{aligned} \quad (12)$$

and c_0 is given by the following backward recursion

$$c_{k-1} = c_k - q_k^\mathsf{T} B L_k^{-1} B^\mathsf{T} q_k + \tilde{d}_k^\mathsf{T} S_k Z_k \tilde{d}_k + 2q_k^\mathsf{T} Z_k \tilde{d}_k, \quad (13)$$

with $c_{N+1} = 0$.

Proof: With $\tilde{x}(k) \triangleq x(k) - d_k$, (6) can be rewritten as

$$\tilde{x}(k+1) = A\tilde{x}(k) + Bu(k) + \tilde{d}_{k+1} + w(k). \quad (14)$$

We prove the result by following the standard dynamic programming arguments. The cost-to-go at time t for the cost function in (8) is

$$\begin{aligned}J_N(t, u_t^N) \triangleq \mathbb{E}\Bigg[\sum_{k=t}^{N} \big[||\tilde{x}(k)||_Q^2 + ||u(k)||_R^2\big]\\ + ||(\tilde{x}(N+1)||_{Q_{N+1}}^2\Bigg].\end{aligned}$$

Let the optimal value of cost-to-go (by applying optimal inputs $u_N^*(t), \cdots, u_N^*(N)$) be of the following form

$$J_N^*(t) = \mathbb{E}[||\tilde{x}(t)||_{S_t} + 2q_t^\mathsf{T}\tilde{x}(t) + c_t] + \lambda_t,$$

for some $\lambda_t > 0$. Then, we have

$$J_N^*(t-1) = \min_{u(t-1)} ||\tilde{x}(t-1)||_Q^2 + ||u(t-1)||_R^2 + J_N^*(t).$$

Using (14) and after some tedious but straightforward manipulations, we get

$$\begin{aligned} J_N^*(t-1) = \min_{u(t-1)} \mathbb{E}\Big[&||u(t-1) \\ &+ L_t^{-1}B^\mathsf{T}\{S_t(A\tilde{x}(t-1)+\tilde{d}_t)+q_t)\}||_{L_t} + ||\tilde{x}(t-1)||_{S_{t-1}} \\ &+ 2q_{t-1}^\mathsf{T}\tilde{x}(t-1) + c_{t-1}\Big] + \mathbb{E}[||w(t-1)||_{S_t}] + \lambda_t, \end{aligned}$$

where $L_t, S_{t-1}, q_{t-1}, c_{t-1}$ are given by (10), (9),(11), (13). The above cost expression can be minimized by choosing

$$u_N^*(t-1) = -L_t^{-1}B^\mathsf{T}\{S_t(A\hat{x}(t-1)-d_t)+q_t),$$

and the resulting cost is given by

$$\begin{aligned} J_N^*(t-1) = \mathbb{E}[||\tilde{x}(t-1)||_{S_{t-1}} + 2q_{t-1}^\mathsf{T}\tilde{x}(t-1) + c_{t-1}] \\ + \underbrace{tr(Y_t^\mathsf{T}L_tY_t\Pi_{t-1}(p,q)) + tr(S_tW) + \lambda_t}_{\lambda_{t-1}}, \end{aligned}$$

where Y_t is given in (12). Following the recursions, the optimal cost is given by $J_N^*(0)$ and proof is complete. ■

Note that the Riccati recursion in (9) evolves independently of the reference trajectory. On the other hand, the recursions in (11) and (13) depend on the reference trajectory. As a result, c_0 and q_0 (and thus, the optimal cost) are dependent on the reference trajectory. Next, we generalize the finite-horizon result to obtain the optimal cost for the infinite-horizon tracking problem in (7).

Lemma 4.2: ***(Infinite-horizon optimal control)*** Consider the MJLS in (6) and the infinite-horizon optimization problem in (7). Assume that $\lim_{k\to\infty}\Pi_k(p,q) \triangleq \Pi(p,q)$ exists and is finite. Let $\beta_k \triangleq -q_k^\mathsf{T}BL_k^{-1}B^\mathsf{T}q_k + \tilde{d}_k^\mathsf{T}S_kZ_k\tilde{d}_k + 2q_k^\mathsf{T}Z_k\tilde{d}_k$ and $c = \sup\{||\beta_k|| : k \geq 0\}$. Then,

$$u^*(k) = -L^{-1}B^\mathsf{T}(S(A\hat{x}(k)-d_{k+1})+q_{k+1}), \tag{15}$$

$$J^*(p,q) \leq \bar{J}_\infty^*(p,q) \triangleq c + tr(Y^\mathsf{T}LY\Pi(p,q)) + tr(SW), \tag{16}$$

where $\{S, L, Z, Y\}$ denote the steady state values of $\{S_k, L_k, Z_k, Y_k\}$ in (9), (10), (12).

Proof: Since (A, B) is controllable, the Riccati recursion in (9) converges and $\{S_k, L_k, Z_k, Y_k\}$ are bounded for all $k \geq 0$. Further, due to assumption **A4**, $\tilde{d}_k$ is bounded for $k \geq 1$. Moreover, since the closed loop matrix AZ_k is stable for optimal LQG control [14], the recursion in (11) results in bounded q_k for $k \geq 0$. Thus, $\lim_{N\to\infty}\frac{1}{N}q_0 = 0_n$. Also, β_k is bounded for all $k \geq 0$ and therefore, c is finite and $c_0 \leq cN$. The result then follows from $J^*(p,q) = \limsup_{N\to\infty}\frac{1}{N}J_N^*(p,q)$. ■

Remark 3: ***(Optimal sensor selection policy is independent of reference trajectory)*** The upper bound in (16) is a result of the fact that $\beta_k's$ do not converge and thus, $\lim_{N\to\infty}\frac{1}{N}c_0$ does not exist for an arbitrary bounded reference trajectory. Furthermore, the cost $\bar{J}_\infty^*(p,q)$ depends on the sensor selection policy only through the steady state estimation error covariance $\Pi(p,q)$. Thus, we can state that the optimal sensor selection policy is independent of the reference trajectory. □

Next, we characterize the estimation error covariance of the MJLS. For clarity of presentation, we drop the notational dependence of Π_k on p, q. Let $\tilde{\Pi}_k$ denote the estimation error covariance for a given realization of Markov chain sequence r_0^{k-1}, i.e. $\Pi_k = \mathbb{E}[\tilde{\Pi}_k]$ where the expectation is with respect to the sequence r_0^{k-1}. Since the Kalman filter has access to the states of the Markov chain r_k, the estimates $\hat{x}(k)$ and the estimation error covariance $\tilde{\Pi}_k$ evolve according to the following time-varying Kalman Filter [11], [9]

$$\begin{aligned} \hat{x}(k+1) = A[\hat{x}(k-1) + K_k(y(k) - C_{r_k}\hat{x}(k-1))] \\ + Bu(k), \end{aligned} \tag{17}$$

$$\tilde{\Pi}_{k+1} = A\tilde{\Pi}_kA^\mathsf{T} + W - A\tilde{\Pi}_kC_{r_k}^\mathsf{T}\tilde{L}_k^{-1}C_{r_k}\tilde{\Pi}_kA^\mathsf{T}, \tag{18}$$

$$\text{where} \quad \tilde{L}_k \triangleq C_{r_k}\tilde{\Pi}_kC_{r_k}^\mathsf{T} + R_{r_k},\ K_k \triangleq \tilde{\Pi}_kC_{r_k}^\mathsf{T}\tilde{L}_k^{-1}, \tag{19}$$

with $\hat{x}(0) = 0$ and $\tilde{\Pi}_0 = P_0 = X_0$.

Since $\tilde{\Pi}_k$ depends on all the past realizations r_0^{k-1}, the exact characterization of $\Pi_k = \mathbb{E}[\tilde{\Pi}_k]$ is intractable. Therefore, we use an upper bound on Π_k presented in [9], and restate the result.

Lemma 4.3: ***(Upper bound on estimation error covariance [9])*** Let $\bar{\Pi}_i(k) \in \mathbb{R}^{n\times n}$ be positive-definite for $i = \{1, 2, \cdots, m\}$, $k \geq 0$ and evolve as

$$\pi_j(k)\bar{\Pi}_j(k+1) = \sum_{i=1}^m p_{ij}\pi_i(k-1)g_j(\bar{\Pi}_i(k)) \text{ with,} \tag{20}$$

$$\bar{\Pi}_j(1) \geq g_j(X_0) \quad \text{where,}$$

$$g_j(X) = AXA^\mathsf{T} + W - AXC_j^\mathsf{T}(C_jXC_j^\mathsf{T} + R_j)^{-1}C_jXA^\mathsf{T}, \tag{21}$$

Then, $\Pi_k \leq \bar{\Pi}_k \triangleq \sum_{i=1}^m \pi_i(k-1)\bar{\Pi}_i(k)$ for $k \geq 1$.

Proof: See [9], Theorem 5. ■

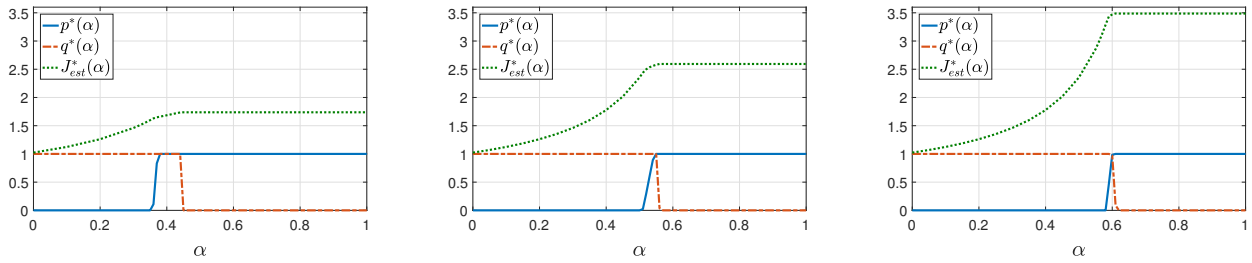
We assume that the recursion in (20) converges and let the steady state value of the bound be denoted by $\bar{\Pi}(p,q) \triangleq \sum_{i=1}^m \pi_i\bar{\Pi}_i$. The bound satisfies the steady state equations

$$\pi_j\bar{\Pi}_j = \sum_{i=1}^m p_{ij}\pi_i g_j(\bar{\Pi}_i) \qquad j = 1, 2, \cdots, m. \tag{22}$$

For a sufficient condition on convergence, see the discussion in [9], [15]. The bound on the estimation error covariance induces an upper bound on the infinite-horizon LQG cost. Using (16) and trace properties, the upper bound is given by

$$\begin{aligned} J^*(p,q) \leq \bar{J}^*(p,q) \triangleq c + tr(SW) \\ + \lambda_{max}(Y^\mathsf{T}LY)tr(\bar{\Pi}(p,q)). \end{aligned} \tag{23}$$

We have obtained an analytical upper bound for the LQG cost in (23) for a given Markov chain with transition probabilities $\{p, q\}$. Next, we aim to minimize this bound with respect to p and q to obtain the optimal sensor selection policy. It can be easily observed that minimizing the upper



(a) $D = 0$ (b) $D = 1$ (c) $D = 2$

Fig. 2: Optimal transition probabilities and scaled optimal cost.

bound in (23) is equivalent to minimizing $tr(\bar{\Pi}(p, q))$. Thus, the optimal sensor selection problem can be described as

$$\min_{p,q} \quad tr(\bar{\Pi}(p, q)) \tag{24}$$
$$\text{s.t.} \quad 0 < p < 1, \quad 0 < q < 1.$$

We solve this optimization problem in (24) using numerical techniques. The details are presented in Section V.

V. Simulations and Numerical Results

A. UAV Model and Linearization

To validate our theoretical results, we consider a quadrotor platform as the UAV model. Following the modelling approach of [16], a quadrotor is assumed to be a rigid-body with constant mass and distinct orientation. Let $\zeta \triangleq [x, y, z]^T$, $\eta \triangleq [\phi, \theta, \psi]^T$ and $\rho \triangleq [\zeta, \eta]^T$ denote the position, attitude, and configuration of UAV in the inertial frame of reference. Then, the translational dynamics is defined by

$$m\ddot{\zeta} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + RT_{\mathbf{B_f}} - \begin{bmatrix} D_x & 0 & 0 \\ 0 & D_y & 0 \\ 0 & 0 & D_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{25}$$

where m is mass of quadrotor, g is gravitational acceleration, $T_{\mathbf{B_f}} \triangleq [0, 0, T]^T$ is thrust vector in body frame of reference with T being the total thrust force, D_i's are the lumped first-order aerodynamic drag coefficients in the i direction, and R is the standard zyx rotation matrix. The rotational dynamics in the inertial frame of reference is given by

$$M(\eta)\ddot{\eta} = \mathcal{T}_{\mathbf{B_f}} - C(\eta, \dot{\eta})\dot{\eta}, \tag{26}$$

where $\mathcal{T}_{\mathbf{B_f}} \triangleq [\tau_\phi, \tau_\theta, \tau_\psi]^T$ is the vector representing the torques about the Euler angles, $M(\eta)$ is the moment of inertia and $C(\eta, \dot{\eta})$ is a matrix that captures the centrifugal and gyroscopic forces [16]. By defining state and input variables as $x_c \triangleq [\rho, \dot{\rho}]^T$ and $u_c \triangleq [T, \mathcal{T}^T_{\mathbf{B_f}}]^T$, we linearize the UAV dynamics (25)- (26) about the hovering operating point and then discretize the continuous-time linear model (with sampling time $0.1s$) to obtain the LTI form of (1). The parameters of our linearized UAV model are as follows: mass (m) is 0.468 kg, gravitational acceleration (g) is 9.81 m/s^2, and drag coefficients ($D_x = D_y = D_z$) are 1 kg/s. The roll and pitch moments of inertia in the UAV (non-inertial) frame of reference ($I_{xx} = I_{yy}$) are 4.85×10^{-3} kgm^2 and the yaw moment of inertia (I_{zz}) is 8.801×10^{-3} kgm^2, where all moments of inertia are contained within the matrix $M(\eta)$.

B. Effect of Attacks on Sensor Selection Policy

We implement the optimal sensor selection policy for the discretized linear system described in subsection V-A. Let $p^*(\alpha, D)$ and $q^*(\alpha, D)$ denote the optimal transition probabilities obtained via the minimization problem (24). We perform the minimization numerically using Matlab by performing an exhaustive search over the range of p and q. Since, the minimization of cost bound in 23 is equivalent to minimization of $tr(\bar{\Pi}(p, q))$, we present our results in this section in terms of the estimation cost $J_{est} \triangleq tr(\bar{\Pi}(p, q))$. Moreover, let the optimal cost be denoted as $J^*_{est}(\alpha, D) \triangleq tr(\bar{\Pi}(p^*(\alpha, D), q^*(\alpha, D)))$.

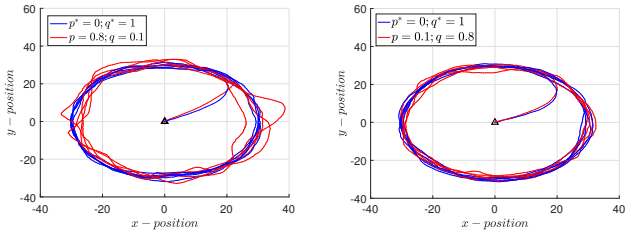
Fig. 2 depicts the optimal transition probabilities p^*,q^* and the scaled values (by a factor of 20 for visual clarity) of the optimal cost J^*_{est} as a function of the attack probability α, for three values of wait times $D = 0, 1, 2$ (the noise covariances are $W = V_1 = I_n, V_2 = 4I_n$). It can be observed that for small values of attack probability α , $p^* = 0$, $q^* = 1$, i.e., it is always optimal to use sensor S_1[1]. Instead, for large values of α, it is always optimal to use sensor S_2. Interestingly, a sharp switching from S_1 to S_2 occurs in a transition region for intermediate values of the α. Also, for some values of α in the transition region, $p^* = 1$, $q^* = 1$ which implies that the optimal policy is to periodically (deterministically) switch between S_1 and S_2. Moreover, it can be observed that the optimal cost is a non-decreasing function of α, since the system performance deteriorates on increasing the attack rate. Further, note that once we switch to S_2, the optimal cost does not change with α, since S_2 is secure to attacks.

Finally, one can easily observe from figs. 2b and 2c that as the wait time D increases, the transition region (from S_1 to S_2) shifts towards the right. This is obvious, since an increasing wait time D leads to fewer measurements by S_2, and thus S_1 is preferred more. Also, for each value of p^*, q^*, the optimal cost increases as D is increased.

C. Optimal Trajectory Tracking under Attacks

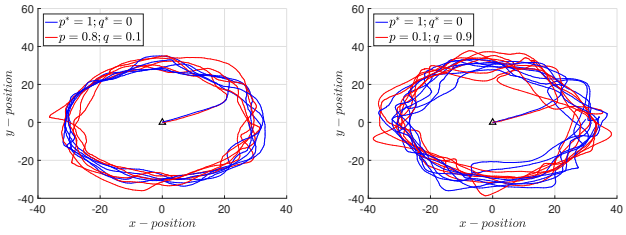
To visualize the tracking performance of our sensor selection policy, we simulate the trajectories of the UAV using

[1]Although we assume $0 < p < 1$, $0 < q < 1$ in (24), the limit of cost J_{est} exists when p and q approach these boundary values.



(a) $D = 0$ (b) $D = 3$

Fig. 3: Tracking under low rate DoS attack ($\alpha = 0.2$) for the optimal (blue) and sub-optimal (red) sensor selection policies.



(a) $D = 0$ (b) $D = 3$

Fig. 4: Tracking under high rate DoS attack ($\alpha = 0.8$) for the optimal (blue) and sub-optimal (red) sensor selection policies.

Matlab. The reference trajectory is a circle in the plane parallel to the $x-y$ plane, and is centered at $(0m, 0m, 30m)$ with radius $30m$. With the UAV's initial position at the origin, we consider two attack regimes:

(i) Low rate DoS attack ($0 < \alpha \leq 0.3$): As mentioned in subsection V-B, the optimum policy under low attack probabilities is to select S_1 all the time for obtaining measurements, i.e. $p^* = 0, q^* = 1$. Fig. 3 shows the UAV trajectories with the optimal policies and are compared with other sub-optimal policies. We can observe from fig. 3a (with wait time $D = 0$) that the optimal policy accurately tracks the circular trajectory. On the other hand, a sub-optimal policy that favours S_2 results in considerable tracking error, thereby demonstrating that choosing the right policy is crucial for tracking purposes. Further, fig. 3b (wait time $D = 3$) shows that a sub-optimal policy which favours S_1 has a performance similar to the optimal policy. Finally, note that the optimal policy performance under low attack rates is independent of the wait time, as evident from figs. 3a and 3b.

(iii) High rate DoS attack ($0.7 \leq \alpha \leq 1$): Fig. 4 shows the UAV trajectories when S_1 is subjected to attacks with a higher rate of $\alpha = 0.85$. Comparing with fig. 3, we can easily observe that the tracking performance degrades as the attack becomes more severe. The optimal policy in this high rate attack regime is to always select S_2 for measurements, i.e. $p^* = 1, q^* = 0$. When wait time $D = 0$ (fig. 4a), a sub-optimal policy of favouring S_2 yields similar performance to the optimal policy. Finally, in the case of larger wait time (fig. 4b with $D = 3$), both the optimal policy (selecting S_2) and the sub-optimal policy (favouring S_1) perform poorly since S_1 is affected by high rate attacks and S_2 by a large wait time.

VI. Conclusion and Future Work

In this paper we studied a security-performance trade-off in resource constrained autonomous UAVs that track a reference trajectory. We showed that under denial-of-service attacks on the GPS sensor, a camera sensor can be used to mitigate the effects of the attacks. We presented a stochastic Markovian sensor selection policy to balance security and tracking performance. We numerically obtained optimal switching probabilities between the two sensors so as to minimize the UAV's tracking error under DoS attack.

One direction of interest is to consider other types of attacks including incremental and non-critical jamming attacks, and integrity attacks such as GPS spoofing. Additionally, consideration of linear time-varying approximations of the non-linear UAV dynamics, exploring security-performance trade-off for different trajectory tracking techniques, and providing analytical solutions to the described optimization problems are directions of future research.

References

[1] H. Zhang, P. Cheng, L. Shi, and J. Chen. Optimal dos attack scheduling in wireless networked control system. *IEEE Transactions on Control Systems Technology*, 24(3):843–852, May 2016.

[2] Saurabh Amin, Alvaro A. Cárdenas, and S. Shankar Sastry. *Safe and Secure Networked Control Systems under Denial-of-Service Attacks*, pages 31–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[3] F. Pasqualetti, F. Drfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, Nov 2013.

[4] Farid Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012.

[5] Y Gao, Z Li, and JF McLellan. Carrier phase based regional area differential gps for decimeter-level positioning and navigation. In *Proceedings of the 10th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997)*, pages 1305–1313, 1997.

[6] Girish Chowdhary, Eric N. Johnson, Daniel Magree, Allen Wu, and Andy Shein. Gps-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft. *Journal of Field Robotics*, 30(3):415–438, 2013.

[7] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping:a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[8] Todd Humphreys. Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil gps spoofing. *University of Texas at Austin (July 18, 2012)*, 2012.

[9] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, 2006.

[10] Yilin Mo, Roberto Ambrosino, and Bruno Sinopoli. Sensor selection strategies for state estimation in energy constrained wireless sensor networks. *Automatica*, 47(7):1330 – 1338, 2011.

[11] Oswaldo Luiz Valle do Costa, Ricardo Paulino Marques, and Marcelo Dutra Fragoso. Markov jump linear systems. *Discrete-Time Markov Jump Linear Systems*, pages 1–14, 2005.

[12] T. Kailath. *Linear Systems*. Prentice-Hall, 1980.

[13] P Whittle. *Optimization over Time: Dynamic Programming and Stochastic Control*, volume 1. John Wiley & Sons, 1982.

[14] Brian D. O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.

[15] Smart Grid Investment Grant Program. Networked sensing estimation and control systems. Technical report, California Institute of Technology, Pasadena, 2009.

[16] Guilherme V. Raffo, Manuel G. Ortega, and Francisco R. Rubio. An integral predictive/nonlinear h-infinity control structure for a quadrotor helicopter. *Automatica*, 46(1):29–39, January 2010.