A Markovian Approach for Attack Resilient Control of Mobile Robotic Systems [*]

Nicola Bezzo Yanwei Du
Dept. of Computer & Information Science
University of Pennsylvania
{nicbezzo, duyanwei}@seas.upenn.edu

Oleg Sokolsky Insup Lee
Dept. of Computer & Information Science
University of Pennsylvania
{sokolsky, lee}@cis.upenn.edu

ABSTRACT

In this paper we propose an optimal planning strategy against malicious attacks on stochastic robotic cyber-physical systems (CPS). By injecting erroneous information and compromising sensor data, an attacker can hijack a system driving it to unsafe states. In this work we bear on the problem of choosing optimal actions while one or more sensors are not reliable. We assume that the system is fully observable and at least one measurement (however unknown) returns a correct estimate of a state. We build an algorithm that leverages the theory of Markov Decision Processes (MDPs) to determine the optimal policy to plan the motion of unmanned vehicles and avoid unsafe regions of a state space. We identify a new class of Markovian processes, which we call *Redundant Observable MDPs* (ROMDPs), that allows us to model the effects of redundant attacked measurements. A quadrotor case study is introduced and simulation and experimental results are presented to validate the proposed strategy.

1. INTRODUCTION

In recent years autonomous robotic systems are becoming more and more popular both in civilian and military operations. Thanks to the continue growing of sensing and computation capabilities, this class of cyber-physical systems can perform complex missions in unknown environments, with small or even inexistent user interactions. However this increase in functionality is also introducing security vulnerabilities which can compromise the integrity of the system. In fact not only a robot has to deal with noisy measurements from its sensors, uncertainties in the communication, and possible disturbances from the environments, but nowadays, also malicious attacks on the sensing and communication infrastructure of the system need to be taken into account to avoid reaching unsafe regions of a state space. Examples of these attacks have already been exploited and demonstrated with real vehicles like in [3] where a GPS was spoofed hijacking a yacht off route. Fig. 1 shows an example of a waypoint navigation scenario in which a quadrotor UAV needs to cross a grid, to reach a goal point (green areas) and avoid undesired regions (red areas). If the agent is confused about its current state, an improper action can result in driving the system inside the undesired regions of the workspace. The example displayed in Fig. 1 will be used as a reference for the experimental validations reported below. Thus, within this work we are interested in determining the optimal actions to take on an autonomous robotic system given uncertain estimations of its state, due to possible attacks on sensor measurements. Each vehicle is equipped with multiple sensors that, although they may perform

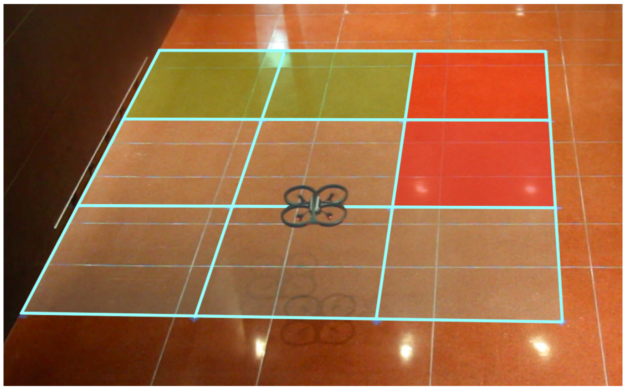

Figure 1: Example of mission envisioned in this work: the UAV needs to reach goal areas (green) and avoid the undesired areas (red) in the environment when one or more measurements are maliciously compromised by attacks.

[*] This material is based on research sponsored by DARPA under agreement number FA8750-12-2-0247. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

different physical measurements, can estimate one common state (e.g., from encoders, GPS, and IMU we can extract position estimates). Redundancy can also be achieved by considering communication with other vehicles that are estimating the states of the surrounding neighbors. Problems like the one described above, can be mapped into Markov decision processes (MDPs) and applied not only to robotic systems but to any cyber-physical system that consists of sensor measurements for state estimation, like power plants, medical systems, and transportation systems.

The contribution of this paper is threefold: i) we develop an algorithm that leverages MDPs to find the optimal planning policy when malicious attacks on one or more sensors are present, ii) the proposed technique can deal with up to $N-1$ sensors under attack, and iii) we run extensive simulations and hardware evaluations considering realistic quadrotor dynamics to validate the proposed technique.

The rest of the paper is organized as follows. In Section 2 we review some of the recent literature on the topics of attack detection and estimation. In Section 3 we formally define the problem under investigation followed by details about the ROMDP algorithm in Section 4. A quadrotor case study is then presented in Section 5 outlining simulation results under sensor attacks with noisy measurements and environment disturbances, followed by an indoor experiment with a self localizing UAV architecture. Conclusions and future work are finally drawn in Section 6.

2. RELATED WORK

The study of high assurance vehicular systems is a recent topic that is attracting several researchers in both the control and computer science communities. Malicious attacks are defined as adversarial actions conducted against a system or part of it and with the intent of compromising the performance, operability, integrity, and safety of the system. The main difference between failure and malicious attack is that the former is usually not coordinated while an attack is usually camouflaged or stealthy and behaves and produces results similar or expectable by the dynamics of the system and environment disturbances.

Even though this area of study is still at an early stage, some preliminary work on vehicular security was performed in [7] in which the authors showed through intensive experiments on common cars, that an attacker could take over the CAN bus of a vehicle and compromise the safety of the entire system. Standing from a control perspective, authors in [14] use plant models for attack detection and monitor in cyber-physical systems. For deterministic linear systems this problem of attack detection and estimation has been mapped into an l_0 optimization problem in [9]. In [13] leveraging the work in [9] we presented a state estimation method for linear systems in presence of attacks showing that an attacker cannot destabilize the system by exploiting the difference between the model used for the state estimation and the real physical dynamics of the system. In [5] we propose a recursive implementation based on the linear-quadratic estimator in which together with the update and predict phases a shield procedure is added to remove the malicious effects of attacks on noisy sensor measurements.

In this paper we move one step forward by considering sensor attacks on stochastic systems with input-output probabilistic models. To solve this problem we leverage the theory of MDP [16, 6] to obtain an optimal policy. In the literature we find several works that deal with the problem of partially observable MDPs [16, 6, 12], however the case analyzed in this work does not fit on this class of systems since in POMDPs it is assumed that the state is not fully observable, while in our case the state is fully observable by at least one sensor and corrupted in other measurements. Thus, POMDP theory is not necessary here to solve the problem presented in this paper.

3. PROBLEM FORMULATION

Within this work we are interested in finding an optimal strategy to maximize the probability that a robot under malicious attack can reach a desired state without being hijacked.

Given $\{s^{\text{goal}}, s^{\text{bad}}\} \in \mathcal{S}$ with $\mathcal{S}$ a finite set of states of the world, our problem for this work can be expressed as follows

Problem 1. **Optimal Planning against Attacked Sensors in Stochastic Systems** *Given a vehicle with N sensors measuring state $s \in \mathbf{x}$ while one or more sensor measurements $y_i \in \mathbf{y}$ are maliciously compromised by an adversarial attacker, find the optimal action policy π that maximizes the probability that the system can reach a desired state s^{goal} without being hijacked to s^{bad}.*

Differently from what assumed in [13, 5] in which an upper bound on the maximum tolerable number of attacked sensors was imposed equal to $N/2$, in this work we relax this constraint and consider scenarios where up to $N-1$ compromised sensors are possible when choosing an action. This scenario is considered as the extreme case, since to have all sensors attacked would imply a completely unobservable system from which it is impossible to estimate a state.

4. REDUNDANT OBSERVABLE MARKOV DECISION PROCESSES

The Redundant Observable MDP (ROMDP) framework that we propose in what follows attempts to solve this problem of computing the best policy to safely navigate an agent when its measurements are not consistent because under attack.

A ROMDP can be described as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \mathcal{C} \rangle$ where

- $\mathcal{S}$ is a finite set of states of the world;
- $\mathcal{A}$ is a finite set of actions;

- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \Gamma(\mathcal{S})$ is the transition probability: a function mapping elements of $\mathcal{S} \times \mathcal{A}$ into discrete probabilities distributions over $\mathcal{S}$. Specifically we will represent $\mathcal{P}_a(s, s') = Pr(s_{k+1} = s' | s_k = s, a_k = a)$ as the probability that action a in state s at time k will result into state s' at time $k + 1$. This mapping is crucial to take into account possible disturbance effects on the system;

- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function that specifies the instantaneous reward received by taking each action at each state. $\mathcal{R}_a(s)$ is the reward received by taking action a from state s;

- $\mathcal{O}$ is a finite set of weighted observations that follows a certain action a. For instance $\mathcal{O}_k = \{s_{i,N_i}, s_{j,N_j}\}$ indicates that two states, s_i and s_j were observed after taking N measurements at time k. s_i was recorded N_i times and s_j, N_j times with $N = N_i + N_j$;

- $\mathcal{C} : \mathcal{O} \rightarrow \Gamma(\mathcal{S})$ is a confidence function mapping elements of $\mathcal{O}$ into discrete probabilities distributions in $\mathcal{S}$. We will use $\mathcal{C}(s)$ to represent the probability that the agent is in state s given the observation $\mathcal{O}$.

Specifically here $\mathcal{O}$ and $\mathcal{C}$ are the two elements added from the conventional definition of MDP.

In this work we consider that attacking more sensors is more complicated and less probable, thus we calculate $\mathcal{C}$ by classifying $\mathcal{O}$ based on the number of different observations for each state. For instance, let us assume that a system performs three measurements and after sending an action a from state s, two of its sensors are compromised obtaining two classes of observations, $\mathcal{O} = \{o_1 = o_2 = s_1, o_3 = s_2\} = \{s_{1,2}, s_{2,1}\}$ both reasonable because within the noise and disturbance error model, then we could use $\mathcal{C}(s_1) = 2/3$ since two measurements out of three were on the same state while $\mathcal{C}(s_2) = 1/3$.

The transition probability for these scenarios, which we call *transition confidence* will have the following form:

$$\begin{aligned}\mathcal{P}_a(\mathcal{O}, \mathcal{C}, s') &= Pr(s_{k+1} = s' | s_k = s_i \in \mathcal{O}, \mathcal{C}(s_i), a_k = a) \\ &= \sum_{s_i \in \mathcal{O}} Pr(s' | s_i, \mathcal{C}(s_i), a) = \sum_{s_i \in \mathcal{O}} \mathcal{C}(s_i) \mathcal{P}_a(s_i, s')\end{aligned}$$

where $\mathcal{C}(s_i)$ is such that $\sum_{s_i \in \mathcal{O}} \mathcal{C}(s_i) = 1$. Given these premises, our ROMDP framework follows the MDP process in which a measure of the cumulative future received reward is maximized over a finite or infinite horizon K. More specifically if we consider a *infinite-horizon discounted* model, we have

$$E\left[\sum_{k=0}^{K} \gamma^k r_k\right] \tag{1}$$

with $K = \infty$. For a *finite-horizon* model $K \neq \infty$ and $\gamma = 1$. In (1) the rewards are summed over the lifetime of the agent, but discounted geometrically using the discount factor $0 < \gamma < 1$. Finally the agent should act as to maximize (1). The larger the discount factor, the more future rewards affect the current decision making. Since we are considering a Markov process, the current state and action are the only mean to predict the next state. In order to find the best action at each state, a *policy* π, mapping $\mathcal{S} \rightarrow \mathcal{A}$, is necessary to describe the behavior of the agent. Therefore π_k will be the policy to be used to choose the action a_k on state s_k, from the k^{th} time to K. Let $V_{\pi,k}(s)$ represent the expected sum of rewards gained by executing policy π_k from state s, then the k^{th} value associated with policy π_k from state s is given by

$$V_{\pi,k}(s) = R_{\pi_k(s)}(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{\pi_k(s)}(s, s') V_{\pi,k-1}(s') \tag{2}$$

which means that to evaluate the future, we need to consider all resulting states s', the probability of their occurrence $\mathcal{P}_{\pi_k(s)}(s, s')$ and their value $V_{\pi,k-1}(s')$ under a given policy π_k.

If the agent is not able to determine its state with complete reliability, we need to consider all possible outcomes when implementing a certain policy π, and select an optimal action a to implement. Based on the *Observation* and *Confidence* definitions introduced above, we can modify (2) to consider uncertainties in the observations due to malicious attacks. If one or more observations in $\mathcal{O}$ do not agree, then the system believes that either observations are attainable. Mathematically:

$$V_{\pi,k}(\mathcal{O}) = R_{\pi_k,\mathcal{O}} + \gamma \sum_{s' \in \mathcal{S}} \sum_{s_i \in \mathcal{O}} \mathcal{C}(s_i) \mathcal{P}_{\pi_k}(s_i, s') V_{\pi,k-1}(s') \tag{3}$$

with

$$R_{\pi_k,\mathcal{O}} = R_{\pi_k}(\mathcal{O}) = \sum_{s_i \in \mathcal{O}} \mathcal{C}(s_i) R_{\pi_k}(s_i) \tag{4}$$

Equation (4) represents the expected reward the agent receives using policy π_k given a certain confidence.

So far we have derived a formula to calculate a value function given a policy. If we go in the other direction, we can derive a policy based on the value function. The optimal policy at the k^{th} step, $\hat{\pi}_k$ is achieved from the $(k-1)$ step value function $\hat{V}_{k-1} = V_{\hat{\pi}_{k-1},k-1}(s')$

$$\hat{\pi}_k(\mathcal{O}) = \arg\max_a \left[R_{a,\mathcal{O}} + \gamma \sum_{s' \in \mathcal{S}} \sum_{s_i \in \mathcal{O}} \mathcal{P}_a(\mathcal{O}, \mathcal{C}, s') \hat{V}_{k-1}\right] \tag{5}$$

Finally by using Blackwell Optimality [15], since the state space and actions are finite, there exists an optimal stationary policy $\hat{\pi}(\mathcal{O})$ from which we can obtain the optimal value function $\hat{V}(\mathcal{O})$. As the horizon time increases $\hat{V}_k(\mathcal{O})$ approaches $\hat{V}(\mathcal{O})$. The optimal finite horizon $\hat{V}_k(\mathcal{O})$ is finally defined as

$$\hat{V}_k(\mathcal{O}) = \max_a \left[R_{a,\mathcal{O}} + \gamma \sum_{s' \in \mathcal{S}} \sum_{s_i \in \mathcal{O}} P_a(\mathcal{O}, \mathcal{C}, s') \hat{V}_{k-1}(s')\right] \tag{6}$$

It is important to note that if at any time, all observations are superimposing (i.e., only one state is observed at each iteration), then we are in the classical MDP scenario with $\mathcal{O} = \{s\}$ and $\mathcal{C}(s) = 1$.

Using this framework, an agent acts in order to maximize the expected sum of rewards that it gets on the next k steps. Selecting an improper reward function value may lead to different actions. If the objective is to avoid undesired states, the reward for these undesired states has to be large and negative (approaching -∞).

Solving (6) can be computationally expensive especially for large state and action spaces. Fortunately, there are several methods in the literature for finding optimal MDPs policies using finite horizon iterations [10, 16]. The most common way is to use the *value iteration* process that computes iteratively the sequence $\hat{V}_k$ of discounted finite-horizon optimal value functions in (6). The iterative algorithm presented below computes improved estimates of the optimal value function in ROMDPs.

Algorithm 1 ROMDP Value Iteration Algorithm

```
$k \leftarrow 1$
$V_k(s) \leftarrow 0 \ \forall s$
while $k \leq K$ or $|V_k(s) - V_{k-1}(s)| \geq \epsilon \ \forall s \in \mathcal{S}$ do
    $k \leftarrow k + 1$
    for all $s \in \mathcal{S}$ & $a \in \mathcal{A}$ do
        $\mathcal{C}_{a,k}(s) \leftarrow R_a(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_a(s_i, s') V_{k-1}(s')$
        $V_k(s) \leftarrow \max_a \mathcal{C}_{a,k}(s)$
    end for
    if $\mathcal{O} > 1$ then
        $\mathcal{C}_{a,k}(\mathcal{O}) \leftarrow R_{a,\mathcal{O}} + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_a(\mathcal{O}, \mathcal{C}, s') V_{k-1}(s')$
        $V_k(\mathcal{O}) \leftarrow \max_a \mathcal{C}_{a,k}(\mathcal{O})$
    end if
end while
```

5. QUADROTOR CASE STUDY

The case study investigated in this paper is a motion planning way-point navigation mission in which a quadrotor aerial vehicle needs to cross a workspace from a starting point to a desired goal avoiding undesired locations. Because of space constraints here we omit the quadrotor model which can be found in [11, 4].

5.1 Controller

Fig. 2 shows the diagram of the architecture used to control a quadrotor in ROMDP operations. The controller is derived by linearizing the equations of motion and motor models at an operating point that corresponds to the nominal hover state $\mathbf{x} = \{x, y, z\}$, $\theta = 0$, $\phi = 0$, $\psi = \psi_0$, $\dot{\mathbf{x}} = 0$ and $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0$ with ψ_0 the initial yaw angle and roll ϕ and pitch θ angles small, which leads to $\cos(\phi) = \cos(\theta) \approx 1$, $\sin(\phi) \approx \phi$, $\sin(\theta) \approx \theta$. The nominal values for the inputs at hover are $u_1 = mg$, $u_2 = u_3 = u_4 = 0$.

In order to control the quadrotor to follow a desired trajectory, we use a two levels decoupled control scheme: a low-level attitude control which usually runs at high frequency and a high-level position control running at lower rate.

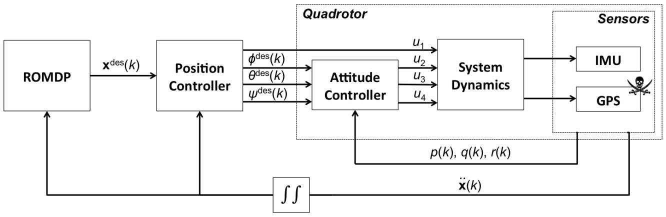

Figure 2: Diagram of the overall controller used on a quadrotor for ROMDP operations.

The position control is used to track a desired trajectory T characterized by $\mathbf{x}_T(t)$ and $\psi_T(t)$. Using a PID feedback controller on the error $e_k = (\mathbf{x}_{k,T} - \mathbf{x}_k)$ we can control the position and velocity of the quadrotor to maintain a desired trajectory. After linearizing the Newton's equation, we can obtain the relationship between desired roll and pitch angles and desired accelerations

$$\phi^{\text{des}} = \frac{1}{g}(\ddot{x}^{\text{des}} \sin(\psi_T) - \ddot{y}^{\text{des}} \cos(\psi_T)) \tag{7}$$

$$\theta^{\text{des}} = \frac{1}{g}(\ddot{x}^{\text{des}} \cos(\psi_T) + \ddot{y}^{\text{des}} \sin(\psi_T)) \tag{8}$$

and

$$u_1 = mg + m\ddot{z}^{\text{des}} \tag{9}$$

Finally, the attitude control is realized using a PD controller as follows

$$\begin{pmatrix} u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} k_{p,\phi}(\phi^{\text{des}} - \phi) + k_{d,\phi}(p^{\text{des}} - p) \\ k_{p,\theta}(\theta^{\text{des}} - \theta) + k_{d,\theta}(q^{\text{des}} - q) \\ k_{p,\psi}(\psi^{\text{des}} - \psi) + k_{d,\psi}(r^{\text{des}} - r) \end{pmatrix} \tag{10}$$

5.2 Environment Setup

The environment configuration plays an important role in ROMDPs. The state space is discretely represented as an occupancy grid map which maps the environment as an array of cells each representing a state and holding a probability value associated with the action taken by the robot. Fig. 3 shows an example of occupancy grid for the missions considered in this work. Green colored cells represent the goals to reach while red colored cells are areas to be avoided.

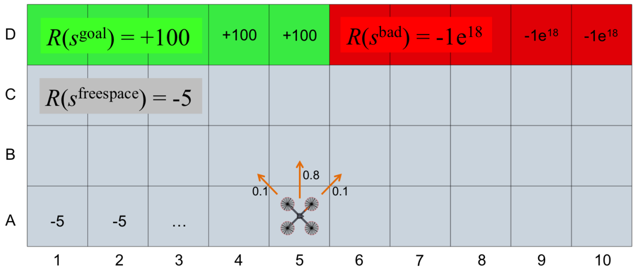
D
C
B
A
$R(s^{\text{goal}}) = +100$
+100
+100
$R(s^{\text{bad}}) = \text{-}1\text{e}^{18}$
-1e^{18}
-1e^{18}
$R(s^{\text{freespace}}) = \text{-}5$
0.8
0.1
0.1
-5
-5
...
1
2
3
4
5
6
7
8
9
10

Figure 3: Example of discretized environment. Each cell represents a different position state. Moving forward from state A5, the quadrotor has 0.8 probability of reaching B5 and 0.2 probability of reaching B4 and B6.

At each step the vehicle will make some observations and compute Algorithm 1 that will produce the optimal action to perform. We define a finite set of primitive actions as follows

- $\mathcal{A} = \{$ *move forward (F), move backward (B), move left (L), move right (R)* $\}$.

Each action will be mapped into the position and attitude control described in the previous section. Specifically here $\{F, B\}$ will be mapped into u_3 and $\{L, R\}$ into u_2 control inputs. Associated with each action there is a transition probability $\mathcal{P}_a$. UAVs often exhibit position drift because of noise and environmental disturbances. Here we assume that disturbances are bounded with a probability $\mathcal{P}_a(s, s^{\text{des}})$ that the agent will end up in the desired state s^{des} from s, and $1 - \mathcal{P}_a(s, s^{\text{des}})$ probability that will end up in either adjacent cells of s^{des} ($s^{\text{des}} - 1$ or $s^{\text{des}} + 1$), as shown in Fig. 3.

Assuming bounded sensor noise, the minimum radius of a cell has to be greater than the maximum displacement we can record due to noise.

5.3 Simulation Results

This section presents a series of Matlab and ROS simulation results on the ROMDP framework applied to a waypoint navigation case study for a quadrotor vehicle.

In the first simulation in Fig. 4, a 8×10 cells environment with 4 goal states s^{goal} and 4 undesired states s^{bad} is presented. Associated with each goal cell there is a reward $\mathcal{R}_a(s^{\text{goal}}) = 100$ while for the unwanted cells $\mathcal{R}_a(s^{\text{bad}}) = -1e^{18}$. For all other freespace locations of the environment $\mathcal{R}_a(s) = -5$. $\mathcal{P}_a(s, s^{\text{des}}) = 0.8$ and $P_a(s, s^{\text{des}} - 1) = P_a(s, s^{\text{des}} + 1) = 0.1$, as depicted in Fig. 3. Three position sensor measurements are available. A mismatch in the sensor measurements is displayed with gray colored cells. Fig. 4 shows a simulation result in which heavy disturbances (e.g., wind) were injected in the negative y direction drifting the quadrotor away from its path. Fig. 4(a) displays the cumulative trace of the attacked measurement (gray colored cells) in comparison with the actual path of the quadrotor. The attacker tries to leverage the disturbance effects, making the quadrotor believe that it is in a safe location, while it is actually in a cell where a lateral motion could send it inside the undesired red area. The ROMDP algorithm intervenes moving the robot forward and then right (Fig. 4(b)) where, although the disturbance, it reaches the desired goal [1].

The second simulation in Fig. 5 was created using ROS-RViz [2] which allows for a more dynamic 3D visualization as well as the same interface that will be used for the experimental validation. Specifically here we consider the navigation of a quadrotor in a obstacle populated environment. Based on the flight altitude of the quadrotor, the reward associated to the obstacle cells varies. In the case shown in Fig. 5 the quadrotor can fly over the obstacle and thus the reward has the same value as in the rest of the workspace. The quadrotor is under attack from the beginning, however, by using the ROMDP value iteration presented in this work, it is able to avoid the bad areas and to reach its desired goal. Finally, Table 1 summarizes some results run with different parameters and a different environment setup. The first row of the table shows the setup for the *goal* and *bad* areas while the remaining cells of the environment have the same uniform reward ($R(s^{\text{freespace}}) = -5$). Last column shows the outcome from the simulations: "goal" means that the quadrotor was able to reach the goal, "bad" means that it run inside one of the bad regions, and "lm" means that it got stuck in a local minimum. If $R(s^{\text{bad}})$ is sufficiently negative, the robot is always guaranteed to avoid s^{bad}, which means that sometime it gets trapped in a local minimum.

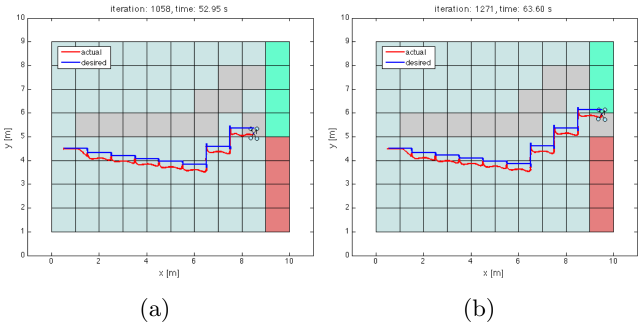

Figure 4: Simulation results with environmental disturbances in the -y direction.

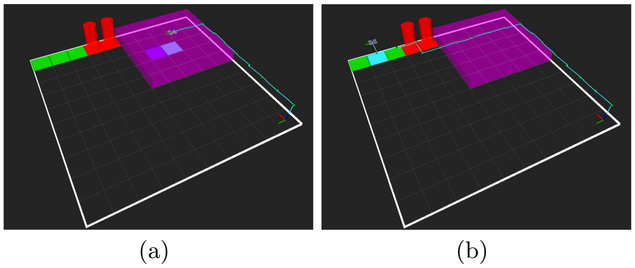

Figure 5: ROS-RViz 3D Simulation results in an obstacle polpulated environemnt.

Table 1: 5×5 area, 1 good cell , 4 bad cells, attack on the right

🟥🟥🟩🟥🟥

Case	γ	$R(s^{\text{goal}})$	$R(s^{\text{bad}})$	$R(s^{\text{freespace}})$	Result
1	0.1	100	-100	-5	goal
2	0.5	100	-1.0e18	-5	lm
3	0.5	1.0e6	-100	-5	bad

5.4 Hardware Implementation

Preliminary indoor experiments using an AR.Drone 2.0 Parrot quadrotor [1] were implemented with different environment setup. Here we show the results for one of these implementations on a 3 × 3, 3.6 m^2, cells environment, as depicted in Fig. 1. The Parrot is equipped with two cameras (front and underneath), a sonar facing downward, and an IMU. Onboard the vehicle has limited computational power allowing only low-level attitude control, leaving the position control and ROMDP implementation presented in this paper on a base station linux-based machine. The base station laptop is equipped with a quad core i7 running the Robot Operating System (ROS) [2]. The linux box communicates to the quadrotor using standard Wi-Fi protocol at an average rate of 200 Hz. The high level position estimator is implemented with an Extended Kalman Filter (EKF) which fuses together vision, inertial, and sonar measurements, as described in [8]. To implement our ROMDP strategy we need at least two position measurements. Because the Parrot has limited capabilities, we duplicated the measurement as if two measurements were propagated from the quadrotor to the base station. Then one of these measurements was attacked. The environment setup for this experiment follows Fig. 1 with 2 goal cells and 2 undesired cells. Fig. 6 shows the position estimates from both the good and the attacked sensors in comparison with the desired position command, all recorded during the hardware implementation. The quadrotor was under attack starting from its initial state for two consecutive steps in the y direction (middle plot in Fig. 6). The optimal sequence of actions chosen by the ROMDP solver was $\{L, F, F\}$. Once the quadrotor reaches one of the two goal locations, it is manually landed (bottom plot in Fig. 6) [1].

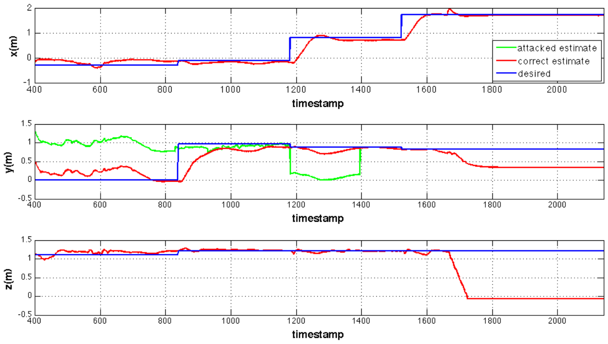

Figure 6: Position estimate for the ROMDP quadrotor hardware implementation.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a stochastic approach for optimal planning under malicious attack on sensors. Our ROMDP framework leverages the MDP theory to obtain an optimal policy (i.e., actions) to drive a vehicle that has not consistent observations of the world. We use redundancy in the sensor measurements considering up to $N-1$ compromised sensors at every time, hiding within the disturbance or the noise profile of the system. Resiliency against attacks is achieved by properly selecting the reward function thus avoiding actions that could hijack the vehicle to undesired regions of a state space. As demonstrated in the simulation and experimental results, this technique is promising and is especially advantageous for systems that are highly sensitive to environmental disturbances (e.g., UAVs). Based on the parameters used in the ROMDP value iteration algorithm, and environment configurations, the agent can get stuck in a local minimum. This behavior is correct since it respects the safety conditions imposed by the ROMDP formulation. The main drawback of this approach is that it is computationally expensive, growing with the number of actions and the square of the number of states.

Future work will be centered on studying the environment topology and performing reachability analysis for the proposed technique and analyzing different disturbance and attack models.

7. REFERENCES

[1] AR.Drone 2.0 Parrot. http://ardrone2.parrot.com.

[2] Robot Operating System. http://www.ros.org.

[3] Spoofers Use Fake GPS Signals to Knock a Yacht Off Course. http://www.technologyreview.com/news/517686/spoofers-use-fake-gps-signals-to-knock-a-yacht-off-course.

[4] N. Bezzo, B. Griffin, P. Cruz, J. Donahue, R. Fierro, and J. Wood. A cooperative heterogeneous mobile wireless mechatronic system. In *IEEE/ASME Transactions on Mechatronics*, pages 20–31. IEEE, 2012.

[5] N. Bezzo, J. Weimer, M. Pajic, O. Sokolsky, G. J. Pappas, and I. Lee. Attack resilient state estimation for autonomous robotic systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 3692–3698. IEEE, 2014.

[6] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, volume 2, pages 963–972. IEEE, 1996.

[7] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *Proc. of USENIX Security*, 2011.

[8] J. Engel, J. Sturm, and D. Cremers. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. *Robotics and Autonomous Systems*, 2014.

[9] H. Fawzi, P. Tabuada, and S. Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *arXiv preprint arXiv:1205.5073*, 2012.

[10] M. L. Littman. The witness algorithm: Solving partially observable markov decision processes. *Brown University, Providence, RI*, 1994.

[11] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The grasp multiple micro-uav testbed. *IEEE Robotics & Automation Magazine*, 17(3):56–65, 2010.

[12] T. M. Moldovan and P. Abbeel. Safe exploration in markov decision processes. *arXiv preprint arXiv:1205.4810*, 2012.

[13] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. Pappas. Robustness of attack-resilient state estimators. In *Proc. of the 5^{th} ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 163–174, Apr. 2014.

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

[15] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*, volume 414. John Wiley & Sons, 2009.

[16] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics.* MIT press, 2005.

[1] Videos about these and more ROMDP simulations and experiments are available at `http://www.seas.upenn.edu/~nicbezzo/ROMDP.html`.