A Case for Low-level Jamming Attacks on Mobile CPS in Target Tracking Applications

EMAD GUIRGUIS
Department of Computer Science
Texas State University-San Marcos
eg1290@txstate.edu

MINA GUIRGUIS
Department of Computer Science
Texas State University-San Marcos
msg@txstate.edu

NIKHIL HALKUDE
Department of Computer Science
Texas State University-San Marcos
nh1129@txstate.edu

***Abstract*— Jamming wireless signals has been one of the most effective attack techniques against any system that relies on wireless communications. Such attacks have been shown to cause degradation of quality, severe inefficiencies, and may potentially lead to instabilities. Surprisingly, this paper uncovers some of the benefits realized when systems are subjected to some level of jamming. In particular, we show cases in which low-level jamming attacks can improve the convergence time for systems when compared to systems that are not subjected to attacks. We illustrate such cases using traditional controllers in target tracking scenarios for Mobile Cyber-Physical Systems. We drive our results through analysis, simulation and real experimental implementation using the Surveyor SRV-1 blackfin robots.**

I. INTRODUCTION

Cyber-Physical Systems (CPSs) are systems composed of physical and computational components that interact over communication mediums. The behavior of the physical system (common referred to as the "plant") is typically controlled by actuators and monitored by sensors to meet specific operational and performance goals. Although the integration between the physical and computational components is not new, the advances in wireless technologies and embedded systems have opened up new venues that would shape our future for years to come [1].

Mobile CPS is a subclass of CPS systems in which a subset of the components are mobile. Recently, there has been a lot of research efforts invested in Mobile CPS applications, specially in the area of autonomous vehicles for exploration [2], [3], border control [4]–[6], and search and rescue operations in land [7], [8], water [9], and air [10], [11], among others. Such need for mobility in CPS applications is vital in filling the gap in many scenarios in which it is inefficient/hazardous/impossible to involve humans directly (e.g., harsh weather conditions, chemical leaks, malicious territories, and disaster impacted sites [12]–[14]).

The advances in wireless technologies have offered convenient communication channels for mobile CPS. Their shared mediums, however, present serious challenges due to intentional jamming by adversaries [15], [16]. It has been shown that a determined constant jammer can easily bring down the whole system. This has prompted research in the area of intelligent and reactive jamming techniques that aim to minimize the jamming intensity to avoid detection [17], [18].

This paper studies the behavior of a number of common controllers subjected to low level jamming attacks. We focus on the case of jamming packets that are used in control (as opposed to data packets). Control packets carry important signals to the mobile CPS and jamming them may appeal to determined attackers. Our study is limited to controllers that compute an error signal between a target set value and a current measurement (e.g, Proportional, Proportional Integral, and Proportional Integral Derivative). The error signal is then used to adjust the control signal. Surprisingly, we show cases in which jamming a small number of control packets can lead to faster convergence. This is the case because when controllers miss a new control signal, the error signal remains unchanged, causing the system to react more aggressively.

Contributions and Implications: This paper demonstrates the presence of a fine margin in which jamming improves the performance of the system. We illustrate this through common controllers that are widely applied. The following points describe the implications of our findings:

- In a scenario where jamming may naturally occur (e.g., known levels of interference), systems can embrace this interference and may even introduce it for stability.
- From an adversarial prospective, an attacker blindly mounting a low-rate jamming attack may actually be helping the system rather than causing damage.
- For closed and legacy systems in which it is not possible to update the software of the controllers or set its paraments, one can selectively choose the control signals to interfere with to improve performance and change the behavior of the controller.

Target tracking as a case study: To study the impact of low level jamming attacks, we use case studies in target tracking scenarios for Mobile CPS applications. We consider an environment in which a target is to be tracked by mobile devices (e.g., agent robots). The environment is considered to be populated by adversaries who can interfere with the communication signals by launching low-level jamming attacks. We assume that the target moves independently with a known speed. A base station detects the target and sends control signal to an agent to locate the target. Such cases arise in emergency response, search and rescue and disaster recovery scenarios.

Paper Organization: In Section II, we present the system and the adversarial models, making the case for low-level jamming

1087-4089/12 $26.00 © 2012 IEEE
DOI 10.1109/I-SPAN.2012.36

attacks on a number of controllers. In Section III, we show implementation results based on target tracking scenarios with the Surveyor SRV-1 Blackfin robots. We present related work in Section IV and we conclude the paper in Section V.

II. A General Model

In this section, we outline a general Linear Time Invariant system for a networked control system. We then illustrate the impact of low-level jamming attacks on the convergence properties of the system.

A. The System Model

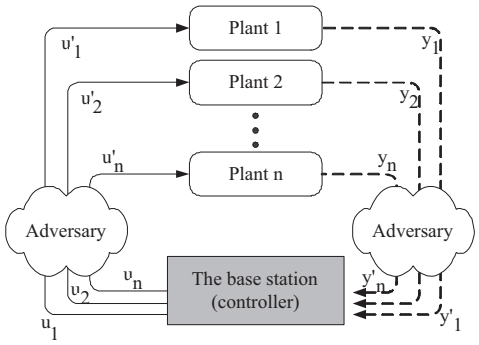

Fig. 1. A CPS with multiple plants and a controller.

Figure 1 illustrates a general block diagram for a CPS composed of a number of plants (e.g., mobile devices) and a controller (e.g., base station). Each plant receives a stream of control signals (denoted by u_i for plant i) from the controller to adjust its operation. A stream of measurement signals (denoted by y_i from plant i) is fed-back from each plant to the controller to update its control rules. The goal is to adjust the operation of the plants to meet a particular function (e.g, reaching equilibrium, tracking targets, controlling the ratio of a certain chemical in an environment, etc). We assume that the measurement and control signals traverse network segments that may be jammed by adversaries.

We consider a Linear Time-Invariant (LTI) system that can be described by Equations (1) and (2), where x represents the state of the system (plants and controller), y is the output vector (measurements from the plants to the controllers) and u is the control vector to all plants. We omit the plant's number and refer to u and y as the general control and measurement signals, respectively.

$$\dot{x} = Ax + Bu + w \quad (1)$$

$$y = Cx + z \quad (2)$$

Matrices A, B, and C represent the plants coefficient matrix, the control matrix, and the output matrix, respectively. w represents a Gaussian random variable with a 0 mean and a covariance matrix Q. We refer to Q as the process noise covariance matrix and it is independent from x. Similarly, z represents a Gaussian random variable with a 0 mean and a covariance matrix R. We refer to R as the measurement noise covariance matrix and it is independent from x. Since the control/measurment signals are typically continuous but they are transmitted in packets over the network, we assume the presence of a sampler and a holder. We assume that the measurement signal $y(t)$ is sampled at times t_k, so we have $y_k = y(t_k)$. Similarly, we assume that the control signal from the controller can be held by the plant for a duration τ, so we have $u_\tau = u(t_\tau)$.[1]

B. The Adversary Model

Based on the above model, we consider an adversary that can jam a subset of the measurement and control signals, y and u, respectively. We assume that jammed packets are dropped and not retransmitted. This assumption is realistic since these packets either carry important control signals or will be used in applying the control rules. If we allow them to be retransmitted by the sender, then the system would always be lagging behind in its control (due to the additional delay in packet loss detection, timeouts and retransmissions). We consider a stealthy adversary, that tries to drop a small number of packets to avoid being detected.

C. A Case for Low-level Jamming Attacks

Consider an instantiation from the model above in which a base station controls the speed of the robot through sending control signals that carry the voltage value to be applied at the robot's circuits that control the motors. For simplicity, we assume that the speed of the robot is a linear function of the voltage signal applied. The robot reports its measured speed back to the base station. Due to different slopes and various types of surfaces, the voltage needs to be dynamically adjusted for the robot to maintain a specific speed. Based on the model above, the control signal u is given by:

$$u = Ky - r \quad (3)$$

where K is the control matrix, r is the reference point (robot target speed) and y is the measured speed.

Consider a Proportional Integral (PI) controller employed by the base station that adjusts the voltage based on the deviation between the current speed and the target speed. This deviation arises due to factors, such as different slopes/surface-textures in the environment, manufacturing imperfections, etc. The discrete equation for the PI controller is given by:

$$u(i) = u(i-1) + k_P e(i) + k_I(e(i) - e(i-1)) \quad (4)$$

where $e(i)$ is the deviation (error signal) between the measurement $y(i)$ and the reference point r at time i. k_P and k_I are the controllers' constants.

Figure 2 shows two sets of 4 different scenarios with a PI controller, in which the measurement signal y is prone to

[1]We ignore the effect of quantization when representing continuous values by their discrete ones.

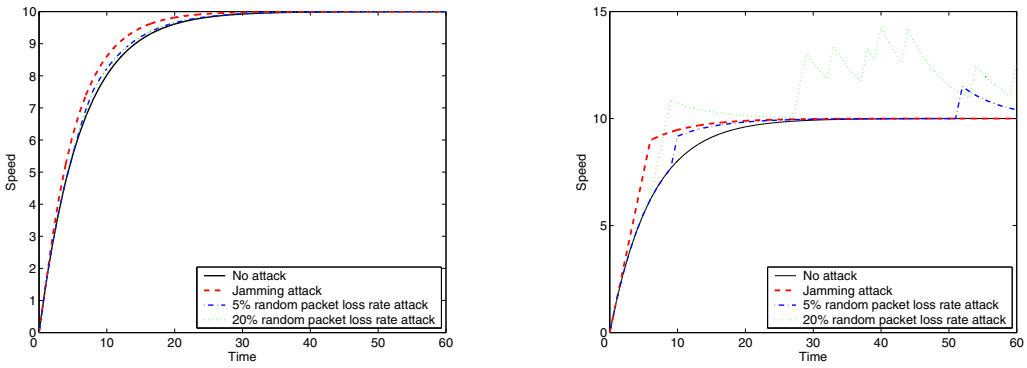

Fig. 2. Convergence under different scenarios for a PI controller.

loss. The Left plot shows results for a system that reapplies the last known measurement value in the absence of new measurements. The Right plot shows results for a system that applies a zero value for lost measurement signals. The base station aims to control the speed of the robot at 10 meters per time unit. We assume convergence when the speed of the robot enters and remains within 1% of the target speed. The PI controller parameters, k_P and k_I are chosen to be 1.5 and 1, respectively. These values are chosen to ensure convergence in a reasonable time (we will show below that our observations hold true for a reasonable choice of control parameters). Every second, the robot sends its measured speed to the base station and the base station sends its control signal. Figure 2 (Left) shows that without attacks, it took the system 28.4 seconds to converge. With a jamming attack on the measurement packets, the system actually converged at time 23, around 19% faster! We also plot the cases of random losses on the measurement packets of 5% and 20%. Figure 2 (Right) shows that with a jamming attack, the system converged at time 20.3, around 28% improvement. This occurs with only 5 packets being jammed. If these jammed packets are the result of an intelligent jamming attack, then this attack would have actually helped the system converges faster to its steady state.

Is this behavior due to the chosen control parameters? For the PI controller, it can be shown that this behavior is independent from the choice of parameters for an over-damped system. Equation 4 can be rewritten as:

$$u(n) \quad = \quad u(1) + k_P \sum_{j=1}^{n} e(j) + k_I(e(n) - e(1)) \qquad (5)$$

Let's compare between two systems, the first one experiences no loss in the measurement signals while the other experiences a lost measurement signal at time $n+1$. For the first system, the control signal $u^1(n+1)$ is given by:

$$u(1) + k_P \sum_{j=1}^{n+1} e(j) + k_I(e(n+1) - e(1)) \qquad (6)$$

and for the second system, the control signal $u^2(n+1)$ is given by:

$$u(1) + k_P \sum_{j=1}^{n} e(j) + k_P e(n) + k_I(e(n) - e(1)) \qquad (7)$$

Thus the difference between $u^2(n+1)$ and $u^1(n+1)$ is given by:

$$k_P(e(n) - e(n+1) + k_I(e(n) - e(n+1)) \qquad (8)$$

which is positive for any converging series of error signals and for any positive value of k_P and k_I. Given our assumption that the speed is a linear function of the voltage applied, the second system would actually converge faster due to the more aggressive control signal applied.

What about the behavior of other controllers? The question is whether other controllers (e.g, PID) are susceptible to this exact behavior during convergence. It can be shown that for systems with controllers that strictly reduce the error signal (difference between the measured signal and the reference point) between two consecutive iterations, would experience the same behavior. Let e_i denote the error signal at time instant i. Consider a family of controllers that ensure convergence in which e_{i+1} is strictly less than e_i, $\forall\ i$. Consider a stream of consecutive measurement signals of length n that are jammed starting at time $i+1$. In the absence of fresh signals and in the case of reapplying the last known measurement, the error signal at each of these n iterations, remains e_i. Thus, the reaction of the controller is more aggressive over these n iterations in comparison to the case when no measurement signals are lost (in which the controller reacts less aggressively over each iteration since $e_{i+1} < e_i$), leading to faster convergence. Notice that for the case in which a zero value is applied for lost measurement signals, the system converges at a rate that is directly proportional to the reference point (and potentially much faster as in Figure 2 (Right).

Are there any tradeoffs to this behavior? So far, we have shown that a small number of lost measurement signals can reduce the convergence time by ensuring a more aggressive behavior by the controller. Figure 3 shows the convergence time (on a log y-axis) versus the aggressiveness of the con-

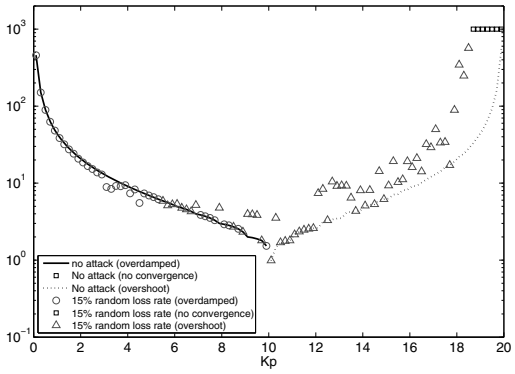

Fig. 3. Convergence under different scenarios for a PI controller.

troller reflected by the parameter k_P on the x-axis.[2] We vary k_P from 0.1 till 20. We plot the convergence time under no loss (indicated by two straight lines) and under a 15% random packet losses (indicated by circles and triangles). Under no packet loss, any k_P value below 10 ensures an over-damped convergence, while any value above 10 ensures an under-damped convergence. In this case, the system always converged. Under 15% random packet loss rate, the range of k_P that ensures over-damped convergence have shrunk (shown by the circles), but reduced the convergence time. The range of k_P that converged with an overshoot has grown (shown by the triangles). Notice that with larger values of k_P, the system has not converged at all (indicated by squares) which we set to 10^3 to indicate non-convergence. Thus, to summarize, a small rate of packet loss improves convergence time but may cause the system to overshoot when controllers are aggressive.

Figure 4 shows sample results under K_P value set to 5. Figure 4 (Left) shows the average convergence time under different packet loss rates over 1000 independent runs, along with the standard deviation. One can see that the convergence time decreases until the packet loss rate is very high. Figure 4 (Right) complements the story. It shows the percentage of experiments in which the system overshot its target set point. One can see that with low level of jamming attacks (e.g, 0.5% and 1%), the system did not overshoot. Once the intensity of the attacks increases, the system overshoots quite often.

III. Experimental Results

A. The Experimental Setup

Our experimental setup is composed of the Surveyor SRV-1 Blackfin robots [19] and a base station. The SRV-1 is an open source robot and uses 802.11b/g for communication. Each robot has two laser pointers for detecting distances to obstacles as well as an onboard camera. Since the robots can be configured to be remotely controlled by the base station, we were able to develop two target-tracking related experiments (by having the robots act as plants and a PI controller controls their actions at the base station as shown in Figure 1. The measurement signal is composed of the distance reported from the robot to the first object in front of it (this is collected through specific commands sent to the robots). The control signal tells the robot to go backward or forward with specific speed for specific number of milliseconds (which translates into distance). Although, we have noticed inaccuracies in the actions taken by the robot (mainly due to physical characteristics of the surface, the power level of the battery and the resolution of the measured distance, etc...), our experiments reflect common difficulties that an agent would experience in real-world scenarios.

[2] k_I has a little effect on the aggressiveness of the controller.

B. Target Tracking Scenarios

In this experiment, we have a target tracking scenario in which a target moves according to a specific pattern and an agent tries to follow the target keeping it at a specific distance. We define an iteration to be a single step taken by the target and the agent. We have fixed the path of the target (which is generated once based on random variables obtained from a normal distribution with mean 5 and variance 4) across all experiments in order to compare between different cases. Thus, the target can be viewed as a plant that does not receive any control signals from the base station (but reports its measurement to the base station) and the agent represents a plant that receives control signals from the base station and also reports its measurements back to it. Since we are not using the camera to detect objects, we assume a virtual target (the path is programmed into the base station). We only focus on the case where an attacker jams the measurements from the agent. The base station employs a PI controller with a specific value of K_p to adjust the speed and direction of the agent based on the error distance between the target and the agent. The goal is to have the agent at a distance of 35cm from the target.

Figure 5 shows our implementation results (averaged over a number of independent experiments). We have experimented with three different cases; the first case is without any jamming attacks, the second one is under jamming 5% of the signals uniformly at random, and the third one is under jamming 20% of the signals uniformly at random. Figure 5 (Left) shows the total distance covered under no attack for different values of K_P. One can see that starting from K_P value of 6, the PI overshoots as indicated by longer distances travelled. In some instances with even larger values of K_P (8, 9 and 10) the system did not converge.

Figure 5 (Right) shows the effect of packet jamming on the number of iterations performed for different K_P values. One can see that with K_P below 5, jamming leads to faster convergence. Starting from K_P equals 6, the inefficient effect of jamming kicks in, leading to longer convergence times. We have to highlight that this graph *only plots the converged experiments* which explains why the convergence time keeps decreasing as value of K_P increases. Experiments that do not converge, potentially have an infinite number of iterations. In our experiments, an experiment is considered to be non-convergent if the distance grows over 70 cm. For example,

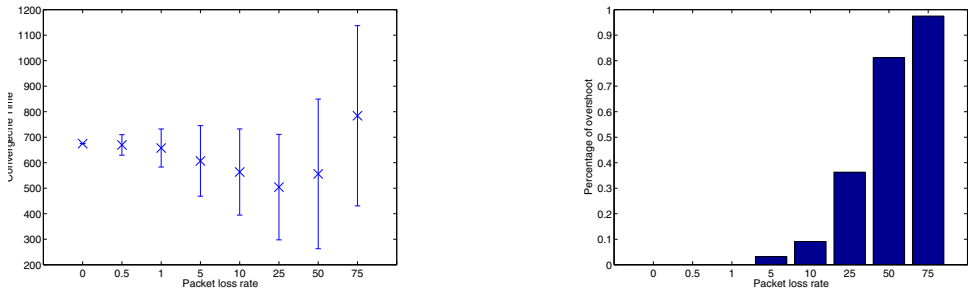

Fig. 4. Convergence time versus packet loss rate (Left) and overshoot percentage versus packet loss rate (Right) under K_P set to 5.

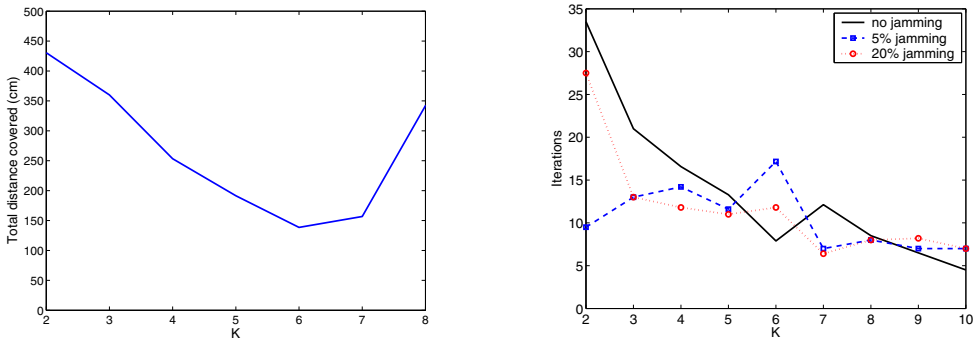

Fig. 5. Implementation results in target tracking. Left: Total distance covered under no jamming. Right: number of iterations for different cases of jamming. K_P is shown on the x-axis.

convergence probability, with K_P set to 9, was 80% and with K_P set to 10 was 72%.

C. Positioning Scenarios

In this experiment, the goal is to adjust the robot at a specific distance from an obstacle. In this experiment the robot measures the distance from the fixed object and reports it to the base station. Then the base station calculates the error between location of the robot and the desired location, applies the PI controller on the measurements, and sends command to the robot to move forward or backward till it reaches the target distance. In this experiment we focus on the jamming policy of a single packet that impacts convergence.

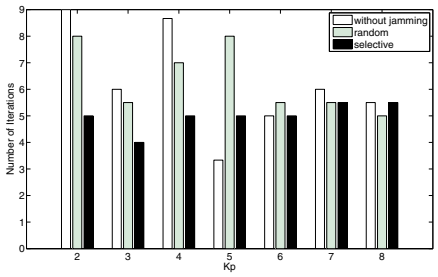

Fig. 6. Implementation results with positioning under 3 different jamming cases.

In Figure 6, we present the number of iterations it took the robot to get into position (30 cm from the obstacle) with different values of K_P and under three different jamming cases; the first case is without jamming any packets, the second is with jamming a single packet at random, and the third is with jamming a single *specific* packet (this packet is chosen to illustrate in the case in which interference can lead to much faster convergence). Each value is the average of several runs from the same initial distance. We can see how low-level jamming attacks can lead to better performance as indicated by faster convergence.

We can observe three main points: the first one is that jamming one *specific* packet usually results in faster convergence than jamming packets at random (since the randomly selected one can be the last one needed for convergence, which adds unnecessary iterations). This is the scenario when the robot reaches its target location but the measurement packet is dropped and the base station is not aware of this. The second point is that when the controller acts in an aggressive manner (e.g, K_P is set to 5) any jamming attack would lead to a slower convergence. Jamming a single packet implies a higher jamming rate when it only takes the system few iterations to converge. The third point is that with larger K_P values, some experiments did not converge (and we do not average those in the graph as we did in the target tracking experiments). Also, while executing random packets jamming experiments, we observed that when the random selection jammed more than one packet, it resulted in worse convergence. This can be explained again by the significant effect of the percentage of attacked packets. Only jamming a fine margin of packets makes the case for low-level jamming attacks.

IV. Related Work

Existing studies from control theory have investigated the impact of loss and delay of measurement and control signals on the overall stability of the system. Those studies vary in their assumptions about the process involved in delaying and/or dropping packets. The authors in [20] consider the problem of optimal estimation under a Bernoulli packet dropping process through a time-varying Kalman filter. They show the existence of a particular drop rate, beyond which the estimation error covariance becomes unbounded. In [21], the authors consider the packet arrival process to follow a Markov model (rather than the traditional Bernoulli model). In [22], the authors assess the impact of deterministic dropping rates on an optimal controller. The authors in [23] study the problem of control and estimation under the effect of common networking protocols (e.g., TCP and UDP). Different studies advocate different actions for missing measurement and control packets. In [24], the authors consider the case where the controller would output zeros for missing measurement packets. Alternatively, in [25], the author considers the use of timers on the receipt of measurement packets. If a timeout occurs, a new control signal is predicted based on the last value(s) of control signals applied. The above studies, however, have always regarded jamming/noise as a negative factor. This work shows that a small margin in noise/attacks can improve the performance of the system under some conditions.

V. Conclusions

In this paper, we have studied the impact of low-level jamming attacks on a number of systems employing traditional controllers. We have shown that low-level jamming attacks may surprisingly improve the performance of the system – in terms of faster convergence time – specially for systems that are not aggressive enough. We have illustrated this through analysis, simulation and real implementation experiments with the SRV-1 Blackfin robots. Our findings suggest the following: (1) the behavior of a given system can be adjusted and tuned without having to modify the controllers themselves nor their parameters, but by simply preventing a small percentage of control signals to arrive, and (2) intelligent and reactive attacks (that aim to minimize the number of jammed packets to avoid detection) may be counter effective from an adversary's standpoint if mounted blindly. In future work, we plan to examine the behavior of other controllers under low-level jamming attacks for more complex applications.

Acknowledgment

This work is supported in part by the NSF CNS grant #1149397.

References

[1] J. Sztipanovitz and J. Stankovic, "Cyber-Physical Systems: A National Priority for Federal Investment in Infrastructure and Competitiveness," http://www.cra.org/ccc/docs/init/Cyber-Physical_Systems.pdf.

[2] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative Multi-Robot Exploration," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2000.

[3] W. Haynes M. Zapata, N. Kannen, M. Sullivan, and J. Conrad, "An Autonomous Vehicle for Space Exploration," in *Proceedings of IEEE Southeastcon*, Huntsville, AL, 2008.

[4] T. Fong, C. Thorpe, and C. Baur, "Multi-robot Remote Driving with Collaborative Control," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 699–704, 2003.

[5] A. Marino, F. Caccavale, L. Parker, and G. Antonelli, "Fuzzy Behavioral Control for Multi-Robot Border Patrol," in *Proceedings of the 17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, June 2009.

[6] A. Marino, L. Parker, G. Antonelli, F. Caccavale, and S. Chiaverini, "A Modular and Fault-Tolerant Approach to Multi-Robot Perimeter Patrol," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, December 2009.

[7] R. Murphy, "Rescue Robotics for Homeland Security," *Communications of the ACM*, vol. 47, no. 3, 2004.

[8] D. Stormont, "Autonomous Rescue Robot Swarms for First Responders," in *Proceedings of the Computational Intelligence for Homeland Security and Personal Safety*, Orlando, FL, April 2005.

[9] H. Okada, T. Iwamoto, and K. Shibuya, "Water-Rescue Robot Vehicle with Variably Configured Segmented Wheels," *Journal of Robotics and mechatronics*, vol. 18, no. 3, pp. 278, 2006.

[10] U. Zengin and A. Dogan, "Real-Time Target Tracking for Autonomous UAVs in Adversarial Environments: A Gradient Search Algorithm," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, December 2004.

[11] J. Kim and Y. Kim, "Moving Ground Target Tracking in Dense Obstacle Areas Using UAVs," in *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea, 2008.

[12] D. Stormont, A. Arora, M. Chandrasekharan M. Datar, U. Dave, C. Gharpure, J. Jorge, A. Kutiyanawala, V. Patel, B. Ramaswamy, L. Sackett, and Z. Song, "Blue Swarm 3: Integrating Capabilities for an Autonomous Rescue Robot Swarm," in *Proceedings of the AAAI Wokshop*, San Jose, CA, 2004, vol. 28.

[13] K. Chuengsatiansup, K. Sajjapongse, P. Kruapraditsiri, C. Chanma, N. Termthanasombat, Y. Suttasupa, K. Sattaratnamai, E. Pongkaew, P. Udsatid, B. Hatthaand P. Wibulpolprasert, P. Usaphapanus, N. Tulyanon, M. Wongsaisuwan, W. Wannasuphoprasit, and P. Chongstitvatana, "Plasma-RX: Autonomous Rescue Robots," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Bangkok, Thailand, Feb 2009.

[14] A. Davids, "Urban Search and Rescue Robots: from Tragedy to Technology," *Communications of the ACM*, vol. 47, no. 3, 2004.

[15] W. Xu, K. Ma, T. Wade, and Y. Zhange, "Jamming Sensor Networks: Attack and Defense Strategies," *IEEE Network*, 2006.

[16] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A Survey on Jamming Attacks and Countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[17] D. Thuente and M. Acharya, "Intelligent Jamming in Wireless Networks with Applications to 802.11 b and Other Networks," in *Proceedings of IEEE MILCOM*, 2006.

[18] M. Acharya, T. Sharma, D. Thuente, and D. Sizemore, "Intelligent Jamming in 802.11 b Wireless Networks," in *Proceedings of OPNETWORK*, Washington, D.C, 2004.

[19] Surveyor, "Surveyor SRV-1 Open Source Mobile Robot," http://www.surveyor.com/SRV_info.html.

[20] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman Filtering with Intermittent Observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[21] S. Smith and P. Seiler, "Estimation with Lossy Measurements: Jump Estimators for Jump Systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, 2003.

[22] M. Yu, L. Wang, T. Chu, and G. Xie, "Stabilization of Networked Control Systems with Data Packet Dropout and Network Delays via Switching System Approach," in *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[23] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of Control and Estimation Over Lossy Networks," *IEEE*, vol. 95, no. 1, pp. 163, 2007.

[24] C. Hadjicostis and R. Touri, "Feedback Control Utilizing Packet Dropping Network Links," in *Proceedings of IEEE Conference on Decision and Control*, 2002.

[25] J. Nillson, *Real-Time Control Systems with Delays*, Ph.d. thesis, Lund Institute of Technology, Lund, Sweden, 1998.