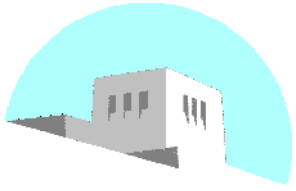
10-26-2010

A Comparison of a Curve Fitting Tracking Filter and Conventional Filters under Intermittent Information

Domagoj Tolic

Rafael Fierro

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING



SCHOOL OF ENGINEERING
UNIVERSITY OF NEW MEXICO

A Comparison of a Curve Fitting Tracking Filter and Conventional Filters under Intermittent Information

Domagoj Tolić and Rafael Fierro [1]
Department of Electrical and Computer Engineering
The University of New Mexico
Albuquerque, NM 87131
e-mail: dtolic@ece.unm.edu, rfierro@ece.unm.edu

UNM Technical Report: EECE-TR-10-0005

Report Date: October 24, 2010

[1] Supported by NSF grants ECCS CAREER #0811347, IIS #0812338, CNS #0709329.

Abstract

This technical report accompanies the tracking filter based on geometric properties of targets' maneuvers developed in [7]. Herein, we compare estimation quality, processing load, scalability and complexity of our tracking filter with Unscented Kalman Filter (UKF) and Particle Filter (PF). The filters have to estimate targets' positions given limited knowledge and intermittent detections of the targets, and measurement noise. It is shown that our filter requires substantially less information and processing power than UKF and Sampling Importance Resampling PF (SIR PF) while providing comparable tracking performance in the presence of limited information and noise. In addition, corroborating numerical simulations are provided.

Keywords

Curve Fitting, Adaptive Sampling, Unscented Kalman Filter, Sampling Importance Resampling Particle Filter

1 Introduction

Managing networks of heterogeneous robots for accomplishing a common goal, such as, detecting, tracking, and eventually capturing one or more targets, autonomously is very challenging because of the limited availability of information about the targets, processing power, and energy that characterize most surveillance scenarios. When the robots are equipped with embedded wireless systems, information about the targets and the workspace can be obtained directly, through onboard sensors and microprocessors, or indirectly, through wireless communication devices that allow the robots to exchange information with each other, or with a central station. Modern technologies and embedded systems allow sensor measurements to be processed and communicated in real time, allowing the robots to estimate environmental and target characteristics with accuracy and precision that can increase rapidly over time. Therefore, the performance of these networks can be significantly improved by decision making approaches that adapt the sensing and motion policies online, based on the latest sensor measurements. Despite the availability of onboard sensors and processors, these policies must also be applicable when little or no target measurements are available, and must minimize energy and processing power, without compromising the estimation performance, in an effort to prolong the surveillance mission.

The geometric filter presented in [7] is motivated by distributed pursuit-evasion games in which a network of heterogeneous (e.g., ground and aerial) sensors, referred to as *pursuers*, is deployed in an obstacle-populated area of interest, in order to detect, track, and, eventually, capture multiple ground vehicles, referred to as *evaders*. Pursuit should be done as quickly as possible, accurately, with high probability of capturing the *evaders* and with minimal possible consumption of energy.

Our tracking filter considers motions of mobile agents as higher order Markov chains. A kinematic model can give possible geometry of targets' maneuvers and every maneuver takes time for completion. Therefore, maneuvers can be represented as higher order Markov chains. Due to the lack of measurements, we are not able to capture every erratic maneuver, but try to capture tendencies in motions of mobile agents. Erratic maneuvers are not energy efficient, and therefore, represent unnatural behavior. In addition, erratic maneuvers are usually a consequence of poorly designed control systems, and are unwanted behaviors of agents. Hence, our approach does not require exact model of the noise and is broad enough to be applied to many different types of evaders. The only assumption on a measurement noise is to be white noise, zero-mean (non-zero mean noise can always be transformed into a zero-mean noise based on the sensor's characteristics) and with finite variance (fulfilled for all commercially available sensors). Notice that Kalman Filters (KFs) and Bayesian Filters (BFs) exploit (first order) Markov property of a process being estimated.

2 Problem Formulation and Approach

In this section, we deliberately leave the problem of interest general enough to be suitable for many pursuit-evasion scenarios. In other words, the tracking methodology developed in this section can be used in a wide range of applications. A concrete application of the methodology is presented in [7] and more concrete results are obtained. Note that this section does not deal with a problem of area surveillance. Our approach for surveillance of an area-of-interest is presented in [7]. Other applications using the methodology developed in this section are already successfully implemented and are to be published, but are out of scope of this technical report.

The reason why we investigate estimation and tracking approaches under intermittent information as a problem on its own right is because, to the best of our knowledge, nothing similar or very little has been done in the settings we are interested in. Also, in many real-time implementations and experiments, effects of intermittent information cannot be neglected as they impair performance of a control system. Therefore, before applying any estimation or tracking strategy, we investigate and compare methods that already exist with the method developed to suit our research interests.

2.1 Problem Statement and Assumptions

We consider a pursuit-evasion game where the set $\mathcal{P}$ of N heterogeneous robotic sensors have to detect, track and eventually capture M randomly moving heterogeneous targets members of the set $\mathcal{T}$. Elements of the sets $\mathcal{P}$ and $\mathcal{T}$ are denoted $\mathcal{P}_i$ and $\mathcal{T}_i$, respectively. With $I_{\mathcal{P}}$ we denote the index set of $\mathcal{P}$, and with $I_{\mathcal{T}}$ the index set of $\mathcal{T}$. The game takes place in a polygonal area-of-interest (AOI) $\mathcal{S} \subset \mathbb{R}^2$ with boundary $\partial\mathcal{S}$. The area $\mathcal{S}$ is populated by n fixed and convex obstacles $\{O_1, \ldots, O_n\} \subset \mathcal{S}$. The geometry of the i^{th} pursuer's platform is assumed to be a convex polygon denoted by $\mathcal{A}_i$, with a configuration $q_{\mathcal{P}_i}(t)$ that specifies its position and orientation at time t with respect to a fixed (or inertial) Cartesian frame $\mathcal{F}_{\mathcal{S}}$ related to $\mathcal{S}$. Similarly, a FOV of the i^{th} pursuer is assumed to be a convex polygon denoted by $\mathcal{B}_i$. For the sake of simplicity, we will use $\mathcal{A}_i(t)$ and $\mathcal{B}_i(t)$ instead of expressing them through $q_{\mathcal{P}_i}(t)$. When dealing with targets, position and orientation of the i^{th} target at time t are comprised in $q_{\mathcal{T}_i}(t)$.

The problem considered herein can be postulated as follows.

Problem 2.1. *Given a set $\mathcal{P}$ of N mobile sensors with dynamics $\dot{q}_{\mathcal{P}_i}(t) = f_i(t, q_{\mathcal{P}_i}, u_i), i = 1, \ldots, N$, and FOVs $\{\mathcal{B}_1(t), \ldots, \mathcal{B}_N(t)\}$ that are deployed to track, and eventually capture a set $\mathcal{T}$ of M targets with dynamics $\dot{q}_{\mathcal{T}_j}(t) = g_j(t, q_{\mathcal{T}_j}, u_j), j = 1, \ldots, M$, find a set of control policies $\{u_1(t), \ldots, u_N(t)\}$ and find sampling policies $\delta^i_j(t) \in \{0, 1\}$ for the mobile sensors resulting in minimal exposure of sensors and no-escaping targets that are being tracked. Next, find (or choose from existing ones) an estimation method that takes the best use of intermittent measurements and limited knowledge of targets (e.g., intelligent and unknown targets) with respect to estimation quality, complexity, scalability and processing load of the method.*

Note that the estimation method is independent of the sampling policy because intermittent measurements can be a consequence of environment (e.g., target occlusion) or characteristics of sensors and communication channels (e.g., collisions and channel throughput). A sampling policy can be just one of many reasons for intermittent measurements. The sampling policy considered herein is motivated by the Marco Polo game presented in [7].

In this work, the pursuers are holonomic vehicles (UGVs and UAVs) equipped with sensors with limited FOV and limited processing power, and are faster than the evaders (targets). Targets are nonholonomic unmanned ground vehicles (UGVs) and perform purely stochastic maneuvers within the AOI. Sensors should track targets using the minimum of resources at disposition. When pursuers are UAVs, a sensor's FOV represents an area in $\mathcal{S}$ monitored by the pursuer while the altitude of the pursuer is kept constant by an autopilot. In addition, we consider heterogeneous multi-agent groups where all agents may have different FOVs, dynamics, and goals (see [7]). When it comes to the simulations and experiments in this technical report, for the sake of convenience the sensors are omnidirectional (a circular FOV) so the orientations of the pursuers can be neglected. Hence, $q_{\mathcal{P}_i}(t)$ is a 2-dimensional vector $[X_{\mathcal{P}_i}(t)\ Y_{\mathcal{P}_i}(t)]^T$. An example of the sensor with a circular FOV is an omnidirectional camera [1]. Illustrations of the notation and terminology introduced in this section are provided in Figures 3(a) and 3(b).

2.2 Tracking Methodology

This section is omitted in this technical report. The reader is referred to [7].

3 Estimation Performance of the Filters

The proposed tracking strategy has been implemented and applied to numerous case studies, and a quality of an estimate of targets' positions has been compared to Unscented Kalman Filter (UKF) and BF implemented as Sampling Importance Resampling PF (SIR PF) or bootstrap PF. In order to make it concise, this technical report includes only two case studies to convey our observations. SIR is not the most progressive of all PFs, but is well studied and frequently implemented. The implemented PF is capable of dealing with non-periodic observations.

A basic study of PF dealing with intermittent observation as a consequence of having more targets than sensors could be found in [6]. Between two consecutive observations there are no information of the inputs. That lack of information is circumvented using "zero-order-hold" strategy, i.e., a previous input is used until the new one comes (for a comprehensive discussion regarding this issue see [4]).

We start analysis of filters' performance with two one-sensor-one-target case studies. The first case study is simple and the target makes a circular path having almost constant inputs. The second case study includes a zig-zag path and is very challenging for estimation. The reasons are the complicated geometry of the path and abrupt changes of inputs including negative linear and angular velocities and accelerations while turning[1]. Our filter is used in the version with periodic detections and in the version with adaptive sampling rate. In all simulations, minimal possible sampling time T_s is 0.07 s. This period is the minimal period needed to run simulations in a real-time manner. Several outcomes of these simulations could be found in Figures 1 and 2. The sensor's radius is 2.3 m and the AOI is 15×15 m^2 in both case studies. In addition, maximum linear velocity of the sensor is 2 m/s, while the target's linear velocity attains values between -0.7 and 0.7 m/s and angular velocity's values are between -1.5 and 1.5 rad/s.

For KF and BF, the first detection is provided through the initialization of those filters, i.e., initial state and prior distribution, respectively. Therefore, the first detection is not shown in the aforementioned figures, but is counted as a detection for cases with KF and BF. Our filter initializes itself by making detections in the first three consecutive steps. Since the initialization is performed only once, this does not present a big increase in number of detections. Hence, the first three detections are counted as one detection for the cases in which our filter is used.

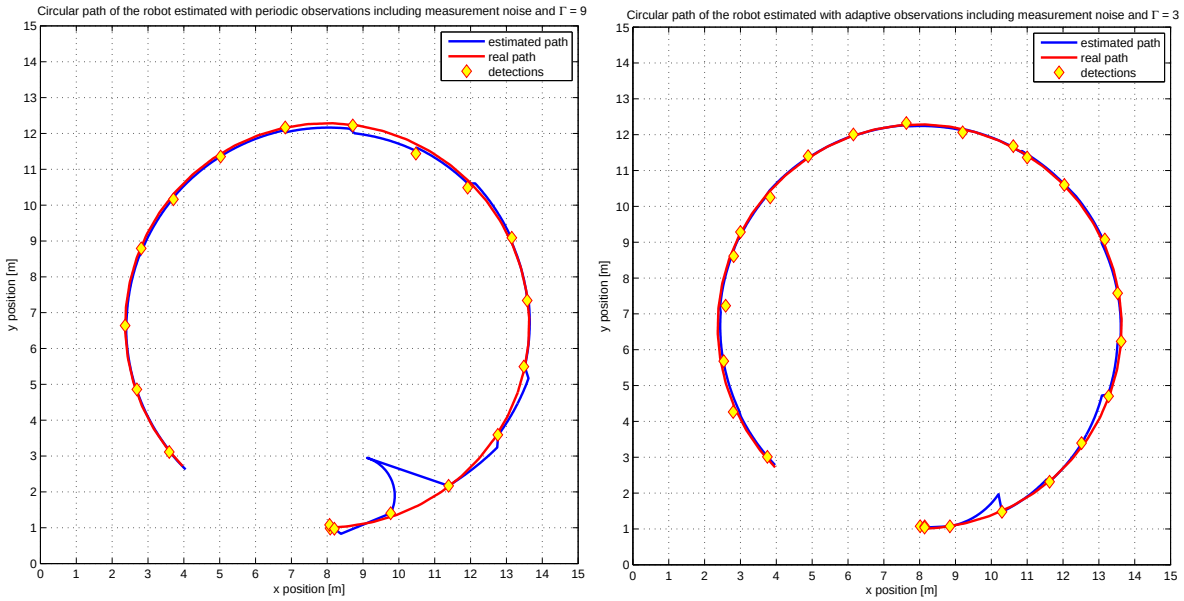
Mean of Root Mean Square Errors (RMSEs) along with corresponding variance of 100 Monte Carlo simulations using different filters are presented in Tables I, II, III and IV. Since the implemented noise is Gaussian, the performances of UKF and PF are quite similar. The performance of our filter is highly competitive with respect to PF. It should not be forgotten that our filter does not require availability of inputs and exact modeling, performs faster and skips estimation process when possible (for more regarding the latter two properties see Subsection 4.1). Therefore, in certain applications the proposed tracking approach is more suitable than UKF and PF, especially in settings where the modeling is difficult to make (e.g., due to lack of knowledge) and increased sampling period.

One of the reasons why our filter performs with such a good quality, despite being deprived of many information that UKF and SIR PF are provided with, is the fact that it uses Γ of samples creating a Γ^{th}-order Markov process model of targets' maneuvers. This makes our method more robust to overcome measurement noise and missing detections. The effect of the noise is averaged out through Γ observations. However, our method brings a certain inertia in the estimate that is sometimes undesirable. The inertia can be diminished by decreasing Γ.

Sensors measure the target position buried in a noise. The noise, for the sake of simplicity, is taken to be a white zero-mean Gaussian with diagonal covariance matrix $R = diag(r_1, r_2)$ for all sensors, where $r_1 = r_2 = 10^{-2}$. The use of lower noise levels (smaller r_1 and r_2) results in several times smaller RMSE of our filter comparing to UKF and PF for the circular path. For higher noise levels, our filter accurately estimates the geometry of the circular path, but the velocity estimate is significantly deteriorated. For the zig-zag path, a decrease in the noise level gives a smaller advantage to UKF and PF with respect to our filter taking RMSE as a criterion.

However, one thing should be pointed out when it comes to the actual implementation of UKF and PF and the modeled noise levels. As the sampling period increases, we have to introduce process noise with covariance matrix $Q = diag(q_1, q_2, q_3)$ in order for the filters to converge. The process noise needs to be increased in order to account for a higher uncertainty. For instance, in the highly sporadic information scenarios, we need to use even $Q = diag(10^{-1}, 10^{-1}, 10^{-1})$ for UKF and PF. By doing so, we can use a smaller number of particles for PF and their impoverishment is decreased (for more details about the impoverishment refer to [3]). Increasing the number of particles is computationally more demanding than increasing the noise. For example, more than 10000 particles have to be used to compensate for an increase in uncertainty without increasing the level of the modeled noise in order for PF to converge. More particles lead to increased running time per second of the PF.

[1] Videos of the simulations and experiments are available at http://marhes.ece.unm.edu/documents/Domagoj/



(a) (b)

Figure 1: Estimates and real paths using different filters: (a) An estimate of the circular path with the Periodic Filter (#det. = 17); and, (b) An estimate of the circular path with the Adaptive Filter (#det. = 23).

With increased modeled noise, we compensate for increased uncertainty and can use a smaller number of particles (1000 particles) to obtain convergence. The running time per step is 3.5ms in the case of 1000 particles, and in the case of 10 times more particles it becomes almost 140 times greater. On the other hand, overly augmented noise levels lead to greater RMSEs. Hence, intermittent information have to be handled carefully and with a lot of tuning of model parameters in the case of UKF and PF. The case of PF with 1000 particles is elaborated more thoroughly in Subsection 4.1. On the other hand, our filter does not have any convergence problems.

Since the simulations' duration is $T_{final} = 55$ s and $T_s = 0.07$ s, the maximal number of detections is 786. Also, the minimal number of observations (maximal sampling period), based on the sensors' radius, is also given in Tables I and III along with corresponding statistics. As it can be seen, the Adaptive Filter has almost minimal possible number of observations. The Adaptive Filter can approach the minimal possible number of observations even more, but the actual number depends on the moving-sensing trade-off determined by a user (and a realization of the noise, of course). The implemented size of the 'No Moving Area' gives a rather high cost to the movement. That is the reason for the above observation. In addition, the Periodic Filter does not give any cost to movement nor takes the noise into account. In the simulations, we use $\eta_j \in [1/4R_j, 3/4R_j]$ where R_j is a sensing radius of the j^{th} pursuer. In addition, the simulations show that $\zeta_j \in [3T_s * \max\{|v_{i,\min}|, |v_{i,\max}|\}, 9T_s * \max\{|v_{i,\min}|, |v_{i,\max}|\}]$ is a valid choice when tracking the i^{th} target.

4 Multi-agent Tracking and Filtering Algorithms' Analysis

After verifying the proposed filter on one-sensor-one-target scenarios, a progressive multi-agent scenario is implemented. In this scenario, four sensors cooperatively track two targets whose behavior is purely stochastic. Each target has to visit 10 uniformly distributed points in the AOI. All targets start within one of the sensors' FOVs. Targets are not tracked by the same sensor during the simulation. Since sensors communicate and collaborate, any sensor can look after any target depending on the game progression. However, one sensor is in charge of only one target at any time. Hence, the idle sensors are available to take care of upcoming targets. Simulations show that such approach does not allow targets to escape from the sensors' FOVs (a formal treatment of this

Table I: A comparison of standard filters with the Periodic Filter for the circular path

Circular Path	UKF	SIR PF	Periodic
# detections	786		
RMSE	0.0960	0.0954	0.1546
Variance	$4.1*10^{-6}$	$3.6*10^{-6}$	$2.6*10^{-4}$
# detections	100		
RMSE	0.1060	0.1407	0.1405
Variance	$4.8*10^{-5}$	$6*10^{-5}$	$2*10^{-4}$
# detections	31		
RMSE	0.1480	0.2084	0.2031
Variance	$2.8*10^{-4}$	$2.6*10^{-4}$	0.0011
# detections	22		
RMSE	0.2080	0.2571	0.2697
Variance	0.0022	$4.8*10^{-4}$	0.0059
# detections	17		
RMSE	0.4623	0.3352	0.3785
Variance	0.0334	$7.4*10^{-4}$	0.0137

Table II: Statistics of the Adaptive Filter for the circular path

Circular Path	# det. mean	# det. Var.	RMSE	Variance
Adaptive	22.14	0.9392	0.3262	0.0050

property of the tracking methodology is provided in [7]). Centralized access to all observations is assumed in the implementation. Illustrations of this scenario are given in Figures 3. Notice that both sensors and targets form heterogeneous groups [8]. For brevity, the properties of the sensors and targets are not included herein.

4.1 Complexity Analysis

When comparing algorithms, several criteria can be used. We decide to use *Running Time Per Step* (RTPS) and "big O" notation as the criteria. The former informs about the actual implementation and is prone to one's programming skills. Nevertheless, RTPS is useful when comparing execution times of algorithms applied on actual problems. The second criterion contains information regarding the potential of an algorithm, i.e., the

Table III: A comparison of the standard filters with the Periodic Filter for the zig-zag path

Zig-Zag Path	UKF	SIR PF	Periodic
# detections	786		
RMSE	0.0966	0.0959	0.1459
Variance	$4.6*10^{-6}$	$3.6*10^{-6}$	$2.1*10^{-5}$
# detections	100		
RMSE	0.1170	0.1425	0.2687
Variance	$6.6*10^{-5}$	$4.3*10^{-5}$	$4.8*10^{-4}$
# detections	25		
RMSE	1.0728	0.8357	0.9156
Variance	0.0133	0.0114	0.0065
# detections	20		
RMSE	1.5459	1.1704	1.2350
Variance	0.0509	0.0724	0.0080

Table IV: Statistics of the Adaptive Filter for the zig-zag path

Zig-Zag Path	# det. mean	# det. Var.	RMSE	Variance
Adaptive	25.380	3.0976	1.0044	0.0306

Table V: A complexity comparison of the standard filters with the filter we developed. RTPSs are obtained with $\#particles = 1000$ for PF, and $\Gamma = 17$ for our filter.

Filter	RTPS [s]		"Big O"
	Mean	Variance	
UKF	$4.9 * 10^{-4}$	$1.7 * 10^{-8}$	$O((\#states)^3)$
SIR PF	0.0098	$5.6 * 10^{-7}$	$O(\#particles)$
Our Filter	0.0035	$4.3 * 10^{-7}$	$O((\Gamma + ord) * ord^2)$

scalability of an algorithm. Results for both criteria are shown in Table V. The simulations are performed on a computer with a 2.4GHz processor and 2GB of memory. Our implementation of UKF and PF is in accordance with "big O" complexity of these filters in literature (extensive literature for UKF, and for PF see [9]) as it should be. According to [5], the number of operations for calculating a fitting curve to Γ points is $(a * \Gamma + b * ord) * ord^2$ where *ord* is an order of the curve and a and b are constants. The order of a curve is the number of generators constituting a basis for the ideal as a vector space. When the curve is a 2D circle, the expression becomes the one in Table V with $ord = 4$. Apparently, the scalability of the fitting circle algorithm can be determined with respect to Γ or *ord*. The algorithm is linear in Γ and cubic in *ord*. It should be noted that Γ in our simulations is a number up to 20 and that the difference in RTPS when $\Gamma = 3$ and $\Gamma = 20$ is 0.1 ms. In addition, *ord* does not grow rapidly as space, in which fitting takes place, grows (see [5]). For instance, "the curse of dimensionality" is a well-known problem for PFs ([2]) and a limiting factor in real-time implementations. Therefore, a possible generalization of our algorithm to 3D scenarios is less computationally demanding than the same generalization using a PF.

Moreover, our filter has the ability to decrease processing requirements when a new detection is anticipated with a certain ρ from previous detections. By choosing ρ, a user can accommodate the filter to a complexity of targets' maneuvers and decrease computational requirements while controlling estimation error. About 10 to 25 % of detections for adaptive type of the filter does not lead to a calculation of a new fitting circle in the case of the zig-zag path (Figures 4(c) and 4(d)). For the case of the circular path and periodic version of the filter, it is between about 10 and 35 % (Figures 4(a) and 4(b)). These percentages represent a significant saving of processing power needed for communication, signal processing, path planning, etc.

When it comes to the complexity of a multi-agent estimation, the complexity is $N * O(filter)$. Each sensor has to obtain readings and targets' positions are determined using one of the above filters for each target.

From Table V and the previous discussion, it can be concluded that our filter demands a smaller portion of the processing power than standard filters and its parameters do not grow as rapidly as for PFs. Furthermore, our filter requires memory for only Γ (about 10) detections while PF has to keep in memory all particles (1000 in our simulations) and an importance density.

5 Conclusion

In this technical report, we compare the performance of the *predictor-corrector* tracking filter developed in [7] with the state of the art filters (UKF and PF) under intermittent information. Precisely, quality of the estimation, scalability, processing load and complexity of the filters are compared. Our filter provides better estimate of targets' positions than UKF as the sampling period increases, and its performance is comparable to the estimation quality of PFs. Moreover, our filter performs faster and demands less memory resources than PF; hence, it is

more suitable for on-line estimation. In addition, the developed filter requires substantially less information then the existing filters (no inputs of evaders, no exact model of evaders and noise).

References

[1] Aveek K. Das, Rafael Fierro, Vijay Kumar, James P. Ostrowski, John Spletzer, and Camillo J. Taylor. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825, October 2002.

[2] F. Daum. Nonlinear filters: beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57 – 69, August 2008.

[3] A. Doucet and A.M. Johansen. *A tutorial on particle filtering and smoothing: fifteen years later*. Department of Statistics, University of British Columbia, December 2008. technical report.

[4] O.C. Imer. *Optimal Estimation and Control Under Communication Network Constraints*. PhD thesis, University of Illinois at Urbana-Champaign, Illinois, 2005.

[5] V. Pratt. Direct least-squares fitting of algebraic surfaces. *ACM SIGGRAPH Computer Graphics*, 21(4):145 – 152, July 1987.

[6] Z. Tang. *Information-Theoretic Management of Mobile Sensor Agents*. PhD thesis, The Ohio State University, Ohio, 2005.

[7] Domagoj Tolić, Silvia Ferrari, Brian Bernard, and Rafael Fierro. Adaptive decision-making in heterogeneous robotic networks for detecting and tracking maneuvering targets with limited information. *Search and Pursuit/Evasion with Mobile Robots*. in preparation.

[8] Domagoj Tolić, Rafael Fierro, and Silvia Ferrari. Cooperative multi-target tracking via hybrid modeling and geometric optimization. In *17th Mediterranean Conference on Control and Automation*, pages 440–445, Thessaloniki, Greece, June 24-26 2009.

[9] Hui Wang, A. Szabo, J. Bamberger, D. Brunn, and U.D. Hanebeck. Performance comparison of nonlinear filters for indoor WLAN positioning. In *11th International Conference on Information Fusion*, pages 1 – 7, Cologne, June-July 2008.

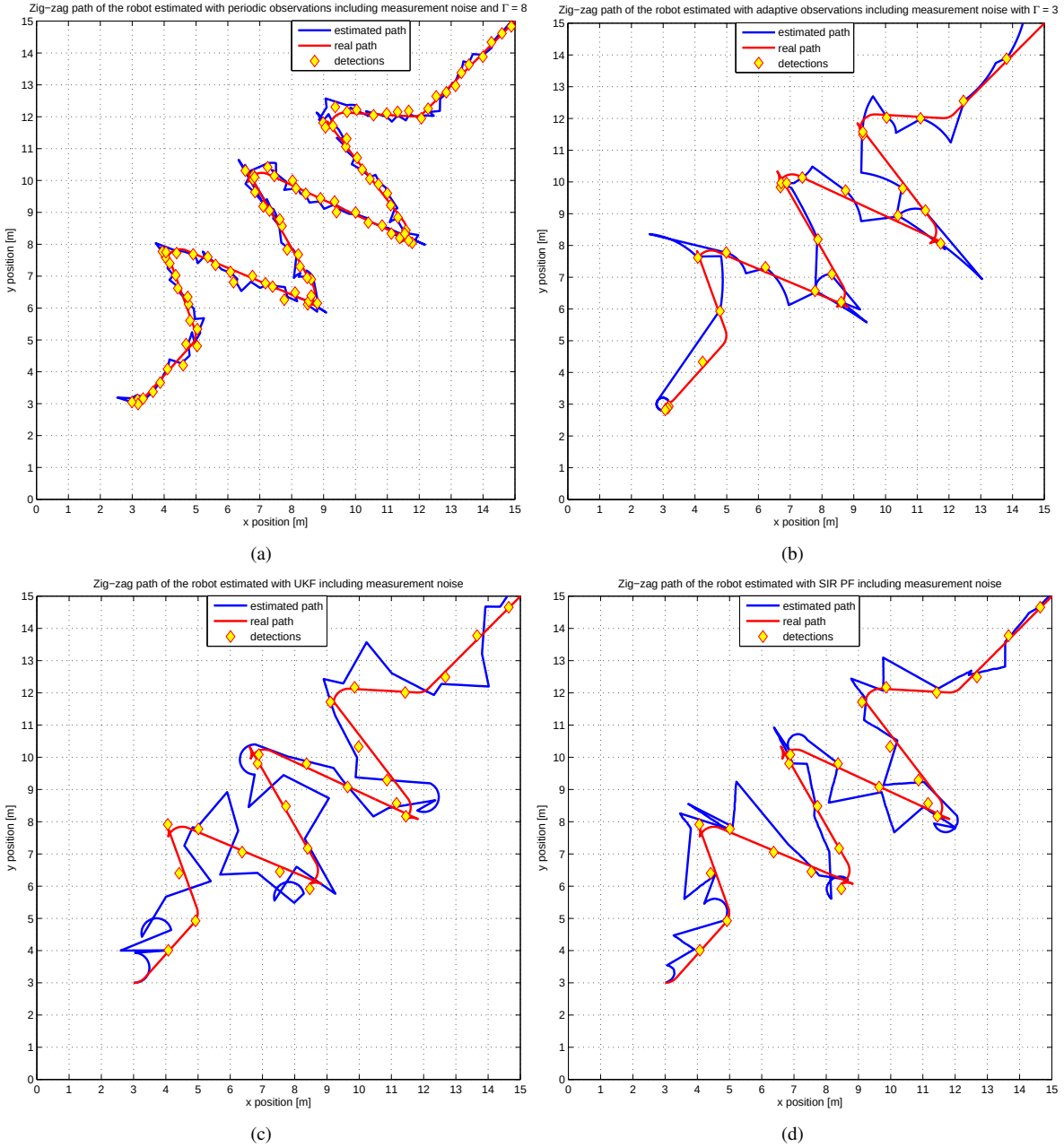
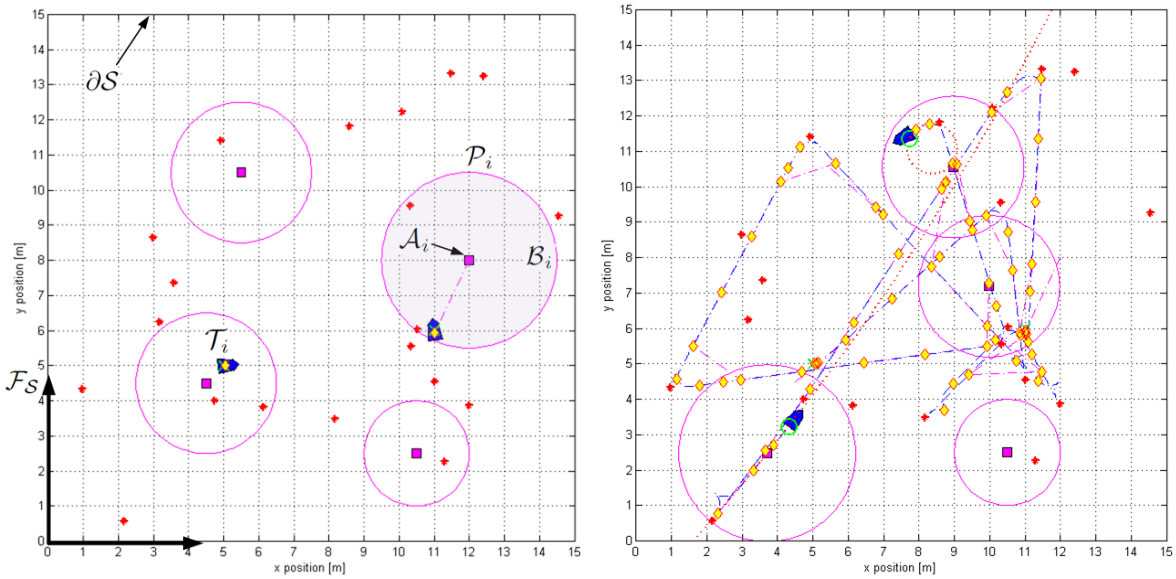


Figure 2: Estimates and real paths using different filters: (a) An estimate of the zig-zag path with the Periodic Filter (#det. = 100); (b) An estimate of the zig-zag path with the Adaptive Filter (#det. = 25); (c) An estimate of the zig-zag path with UKF filter (#det. = 25); and, (d) An estimate of the zig-zag path with PF filter (#det. = 25).



(a) (b)

Figure 3: Snapshots of the multi-agent scenario: (a) Initial state of a cooperative scenario including 4 sensors and 2 targets. The simulation snapshot is edited in order to visualize the notation and terminology introduced in Section 2; and, (b) Final state of a cooperative scenario including 4 sensors and 2 targets. Purple squares represent sensor platforms, purple circles represent the sensors' FOV boundaries, blue polygons represent targets, while yellow diamonds represent detections. Red dotted circles (i.e., the circle and the line) represent fitting circles, red stars are the targets' waypoints, blue hashed curves are the targets' paths, purple hashed curves are the sensors' paths, green crosses are initial positions of the targets and green circles are the estimates of targets' positions.

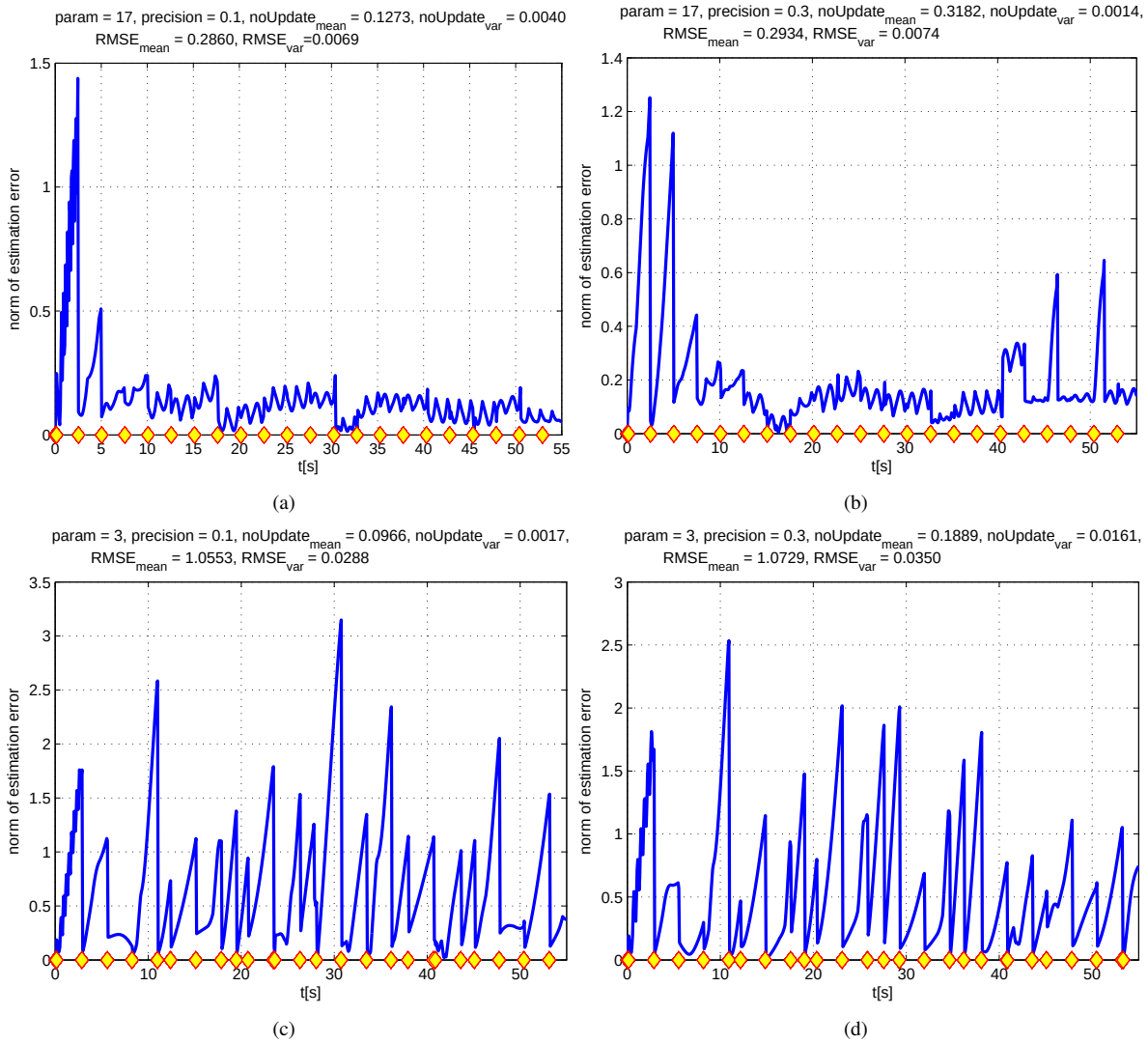


Figure 4: Estimation error in time: (a) A realization of estimation error in time for circular path using the Periodic Filter with $\rho = 0.1$ and $\Gamma = 17$. Title of the plot comprises statistics of 50 Monte Carlo simulations; (b) A realization of estimation error in time for circular path using the Periodic Filter with $\rho = 0.3$ and $\Gamma = 17$. Title of the plot comprises statistics of 50 Monte Carlo simulations; (c) A realization of estimation error in time for zig-zag path using the Adaptive Filter with $\rho = 0.1$ and $\Gamma = 3$. Title of the plot comprises statistics of 50 Monte Carlo simulations; and, (d) A realization of estimation error in time for zig-zag path using the Adaptive Filter with $\rho = 0.3$ and $\Gamma = 3$. Title of the plot comprises statistics of 50 Monte Carlo simulations.