Online Mission Planning for Cooperative Target Tracking of Marine Vehicles ⋆

M. Bayat * **F. Vanni** * **A. Pedro Aguiar** *

* *Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal. (e-mail: {mbayat,fvanni,pedro}@isr.ist.utl.pt)*

Abstract: In the cooperative target tracking maneuvering a group of vehicles is required to follow a target while maintaining a desired formation in space. For that effect, the trajectory of the target has to be estimated (for example by one of the vehicles in formation), translated into mission codes (to save communication resources) and transmitted to the other vehicles. This paper addresses the online mission planning task of converting the absolute or relative position data available from the positioning system installed on the target (e.g. INS, GPS) into a set of lines and constant curvature arcs. Two algorithms are proposed: one using a polynomial fitting and a more complex one using an Interactive Multiple-Model Kalman filter. Simulation results using experimental data and a set of simulated data corrupted with noise and data loss are presented and discussed.

Keywords: Target Tracking, Mission Planning, Interactive Multiple-Model, Polynomial Fitting

1. INTRODUCTION

The past few decades have witnessed considerable interest in the area of motion control of autonomous marine vehicles (AMV) (Aguiar and Pascoal (2007); Alonge et al. (2001); Encarnacao and Pascoal (2000); Fossen (1994); Jiang (2002); Lefeber et al. (2003); Leonard (1995)). In a great number of scenarios, which range from ocean exploration to military applications, AMVs have the capability to perform missions that would be unfeasible, or too dangerous, for a human crew. Current research however goes well beyond single vehicle control. Considerable effort is being placed on the deployment of groups of networked AMVs which can interact autonomously with the environment and other vehicles, resulting in a significant improvement in efficiency, performance, reconfigurability and robustness, and in the emergence of new capabilities beyond the ones of individual vehicles (Klein et al. (2008); Vanni et al. (2008); Aguiar et al. (2009)). One of the most challenging cooperative missions is cooperative target tracking, where a group of vehicles is required to follow a target while maintaining a certain formation in space. Amongst the many scenarios of this kind that can be envisioned, some may require the vehicles to move along the same trajectory that the target has traced (Blidberg et al. (1991)). For this case, the position of the target has to be known either to a ground station or to one of the vehicles in the formation (a "leader"), who could then communicate it to the other vehicles. However, this solution presents significant requirements in terms of communication resources, requirements that could be reduced if the target's trajectory was first approximated by a series of simple, parametrized curves, defined by a few variables that could be sent to the other vehicles at a lower rate. This would be particularly beneficial in the case of underwater applications, where vehicles exchange information over low bandwidth, short range communication channels that are plagued with intermittent failures, multi-path effects, and distance-dependent delays. Moreover, this strategy has the advantage of only requiring vehicles equipped with low cost off-the-shelf controllers, capable of steering them only along simple trajectories such as straight lines or constant curvature arcs.

This work proposes online 2D mission planning algorithms for cooperative target tracking. Fig. 1 shows the schematic block diagram of a generic cooperative target tracking scenario. The goal is to design a real-time algorithm that translates the absolute or relative position data available from the navigation system installed on the target (e.g. INS, GPS) into a set of mission codes. These mission codes are sent to the cooperative target tracking module (which is not addressed in this article) to be translated into identical missions for each of the follower AMVs, so that they can track the moving target in formation. It is important to stress that the translation of the original trajectory into mission codes allows the use of a low capacity communication channel with a low refresh rate. In this paper the translated mission code consists of a combination of lines and constant curvature arcs. This solution was adopted for the GREX Project (GREX (2006–2009)) and is currently being implemented.

We propose two methods of online mission planning. One uses polynomial fitting and the other Interactive Multiple-Model Kalman filter (IMM-KF). The next section describes the algorithms developed. To compare how these methods behave in a practical case, results based on simu-

⋆ This work was supported in part by projects GREX/CEC-IST (contract No. 035223), FREEsubNET (EU under contract number MRTN-CT-2006-036186), Co3-AUVs (EU FP7 under grant agreement No. 231378), DENO/FCT-PT (PTDC/EEA-ACR/67020/2006), and the CMU-Portugal program. The first author benefited from a PhD scholarship of FCT.

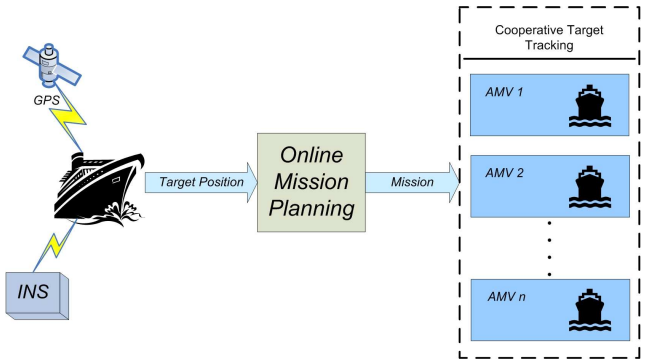


Fig. 1. Schematic block diagram of cooperative target tracking.

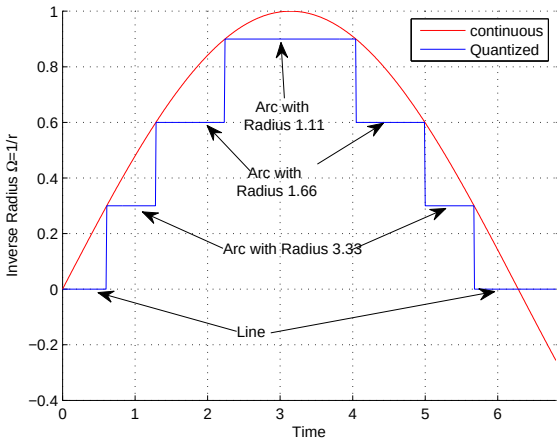


Fig. 2. Quantization: Detection of arcs and lines with quantization.

lated data and on a set of experimental data are compared in Section 3. Concluding remarks are given in Section 4.

2. ONLINE MISSION PLANNING ALGORITHMS

Consider a target moving in a 2D space with its associated magnitude of velocity vector V (m/s), and angular velocity ω (rad/s). Let $\Omega(rad/m)$

$$\Omega = \frac{\omega}{V} = \frac{1}{r} \tag{1}$$

be the curvature, where r denotes the radius. Note that for straight lines Ω is zero. Consider the set

$$\Phi = \{\Omega_0, \Omega_1, \Omega_2, ..., \Omega_n\}, \ \Omega_i \in R \tag{2}$$

composed by n specified curvatures and let $\Omega_0 = 0$. The online mission planning problem can be decomposed as follows:

- Computation of an estimative of Ω from the position (x, y) measurements of the target. In the following subsections we will describe two methods.
- Quantization of the estimated Ω according to Φ and formulation of the corresponding trajectory into a set of lines and arcs. Fig. 2 shows an example plot of Ω versus time quantized by the set $\Phi = \{0, 0.3, 0.6, 0.9\}$ with 7 arcs.
- Transmission of the quantized estimated arc only if it is different from the previous one.

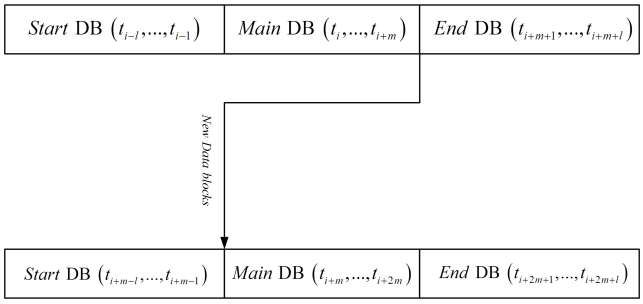


Fig. 3. Data blocks: Diagram of creating data blocks together with start and end data blocks.

2.1 Curvature estimation using polynomial fitting

The first solution proposed for curvature estimation consists in fitting the target position data through a low order polynomial. More precisely, we set a buffer of size m that is constantly filled as the measurements arrive with the last m position data from the target navigation system. Whenever the mission planning is called, the most recent unprocessed buffer is loaded to a data block (DB) inside the mission planning module. To compute Ω from the discrete position data (x, y) along time, a naive solution would be to compute V and ω and then use (1). However, this is not the best solution because the position data is corrupted with noise and computing its derivative will amplify it. Instead, we propose to first smooth the data by fitting it to a low order polynomial $p = (p_x, p_y)$ and only then compute the curvature as follows:

$$\Delta x(t_k) = p_x(t_k) - p_x(t_{k-1}) \tag{3}$$

$$\Delta y(t_k) = p_y(t_k) - p_y(t_{k-1}) \tag{4}$$

$$\theta(t_k) = \texttt{atan2}(\frac{\Delta y(t_k)}{\Delta x(t_k)}) \tag{5}$$

$$\Omega(t_k) = \frac{\theta(t_k) - \theta(t_{k-1})}{\sqrt{\Delta x(t_k)^2 + \Delta y(t_k)^2}} \tag{6}$$

Notice that the polynomial curve fitting may encounter considerable discontinuity in the junction of two consecutive fitted polynomials. To reduce this discontinuity, two more DBs of size l are attached to the beginning and end of the main DB (see Fig. 3).

2.2 Curvature estimation using IMM Kalman filter

In this case, an Interactive Multiple Model Kalman filter (IMM-KF) is developed to estimate the curvature Ω. The IMM-KF is a nonlinear filter that combines a bank of Kalman filters running in parallel, each one using a different model for target motion, with a dynamic system that computes the conditional probability of each KF. The output of the IMM-KF is the state estimate given by a weighted sum of the state estimations produced by each Kalman filter. Fig. 4 shows the block diagram of the IMM-KF. Further details on the IMM-KF can be found in the book by Bar-Shalom et al. (2002), and for a survey on IMM methods in target tracking the reader is referred to Mazor et al. (1998).

In the present work, each Kalman filter j is designed according to the following discrete process model with a constant sampling time T_s.

$$\begin{cases} x_{k+1}^j = x_k^j + T_s V_k^j \cos\theta_k^j \\ y_{k+1}^j = y_k^j + T_s V_k^j \sin\theta_k^j \\ \theta_{k+1}^j = \theta_k^j + T_s\omega^j + w_{\theta_k}^j\sqrt{T_s} \\ V_{k+1}^j = V_k^j + w_{V_k}^j\sqrt{T_s} \end{cases} \quad (7)$$

The sequences $w_{\theta_k}^j$ and $w_{V_k}^j$ are mutually independent, stationary zero mean white Gaussian sequences of random variables, with covariances Q_θ^j and Q_V^j respectively. Here the angular velocity ω^j is set to be a constant in each model, which is included in the set of possible ranges, $-\omega_{\max}$ to $\omega_{\max}$ including the origin (straight line). In (7), (x^j, y^j) denote the position in 2D Cartesian space, and θ^j is the angle of the speed vector $(\dot{x}^j, \dot{y}^j)$.

The IMM-KF algorithm is described as follows:

(1) Initialization:

The initial state vector $\hat{\boldsymbol{x}}^j(0)$ for each model j is assumed to be a Gaussian random variable with known mean

$$E\{\hat{\boldsymbol{x}}^j(0)\} = \bar{\boldsymbol{x}}^j(0) \quad (8)$$

and known covariance matrix

$$\boldsymbol{P}^j(0) = E\left\{\left(\hat{\boldsymbol{x}}^j(0) - \bar{\boldsymbol{x}}^j(0)\right)\left(\hat{\boldsymbol{x}}^j(0) - \bar{\boldsymbol{x}}^j(0)\right)^T\right\} \quad (9)$$

Also let $\mu_j(0)$ be the corresponding mixing probability initial condition.

(2) Calculation of the mixing probabilities:

$$\mu_{i|j}(k|k) = \frac{p_{ij}\mu_i(k)}{\bar{c}_j} \quad (10)$$
$$\bar{c}_j = \sum_{i=1}^{r} p_{ij}\mu_i(k)$$

where $P = [p_{ij}]_{n\times n}$ is the Markov chain transition matrix between models. Here p_{ij} represents the probability of switching from model i to model j.

(3) Mixed Initial condition for each j^{th} model:

$$\hat{\boldsymbol{x}}^{0j}(k|k) = \sum_{i=1}^{r} \hat{\boldsymbol{x}}^i(k|k)\mu_{i|j}(k|k) \quad (11)$$
$$\boldsymbol{P}^{0j}(k|k) = \sum_{i=1}^{r}(\mu_{i|j}(k|k)\{\boldsymbol{P}^i(k|k) + \sum_{j=1}^{r}[\hat{\boldsymbol{x}}^i(k|k) - \hat{\boldsymbol{x}}^{0j}(k|k)].[\hat{\boldsymbol{x}}^i(k|k) - \hat{\boldsymbol{x}}^{0j}(k|k)]^T\})$$

(4) Propagation of state estimate and covariance matrix: Each mixed initial condition $(\hat{\boldsymbol{x}}^{0j}(k|k), \boldsymbol{P}^{0j}(k|k))$ is used as initial condition for its corresponding model j to propagate the estimate $\hat{\boldsymbol{x}}^j(k+1|k+1)$ and the covariance matrix $\boldsymbol{P}^j(k+1|k+1)$ as follows:

$$\hat{\boldsymbol{x}}^j(k+1|k+1) = \hat{\boldsymbol{x}}^j(k+1|k) + K_{k+1}^j r_{k+1}^j$$
$$\boldsymbol{P}^j(k+1|k+1) = (I - K_{k+1}^j H_{k+1}^j)\boldsymbol{P}^j(k+1|k)$$

where

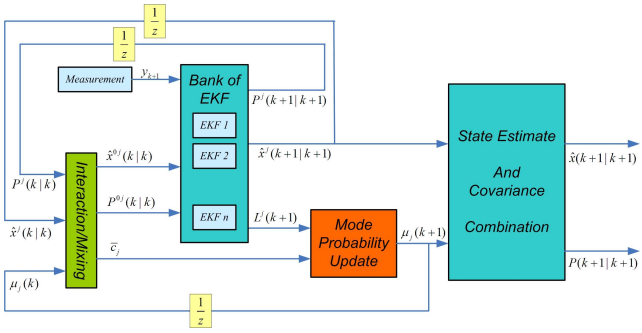


Fig. 4. Block diagram of the IMM-KF.

$$\hat{\boldsymbol{x}}^j(k+1|k) = f(\hat{\boldsymbol{x}}^{0j}(k|k), u_k) \quad (12)$$
$$\boldsymbol{P}^j(k+1|k) = A_{k+1}^j \boldsymbol{P}^{0j}(k|k) {A_{k+1}^j}^T + Q^j$$
$$S_{k+1}^j = H_{k+1}^j \boldsymbol{P}^j(k+1|k) {H_{k+1}^j}^T + R$$
$$r_{k+1}^j = y_{k+1} - h(\hat{\boldsymbol{x}}^j(k+1|k))$$
$$K_{k+1}^j = \boldsymbol{P}^j(k+1|k) {H_k^j}^T {S_{k+1}^j}^{-1}$$

(5) Mode-matched filtering: The Innovation or measurement residual r_{k+1}^j and its corresponding covariance matrix $S_{k+1|k}^j$ of the Kalman filter j are used to generate mode-matched filtering:

$$L_j(k+1) = \frac{\sqrt{\left|S_{k+1}^j\right|}}{(2\pi)^{\frac{n_s}{2}}} e^{-\frac{1}{2} {r_{k+1}^j}^T (S_{k+1}^j)^{-1} r_{k+1}^j} \quad (13)$$

where n_s represents the number of states, which is equal to 4 in this case.

(6) Mode probability update:

$$\mu_j(k+1) = \frac{1}{c} L_j(k+1)\bar{c}_j \quad (14)$$
$$c = \sum_{j=1}^{r} L_j(k+1)\bar{c}_j$$

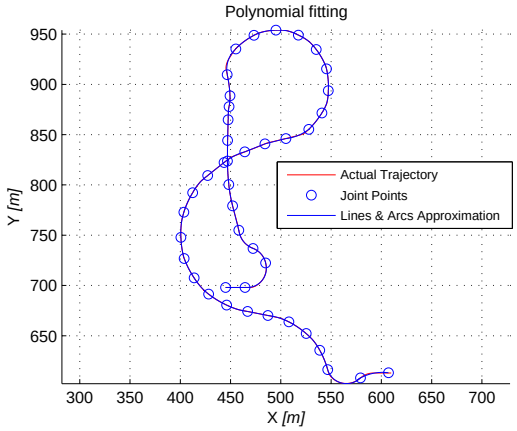
(7) Estimate and covariance combination:

$$\hat{\boldsymbol{x}}(k+1|k+1) = \sum_{j=1}^{r} \hat{\boldsymbol{x}}^j(k+1|k+1)\mu_j(k+1) \quad (15)$$
$$\boldsymbol{P}(k+1|k+1) = \sum_{j=1}^{r}(\mu_j(k+1)\{\boldsymbol{P}^j(k+1|k+1) + [\hat{\boldsymbol{x}}^j(k+1|k+1) - \hat{\boldsymbol{x}}(k+1|k+1)] .[\hat{\boldsymbol{x}}^j(k+1|k+1) - \hat{\boldsymbol{x}}(k+1|k+1)]^T\})$$

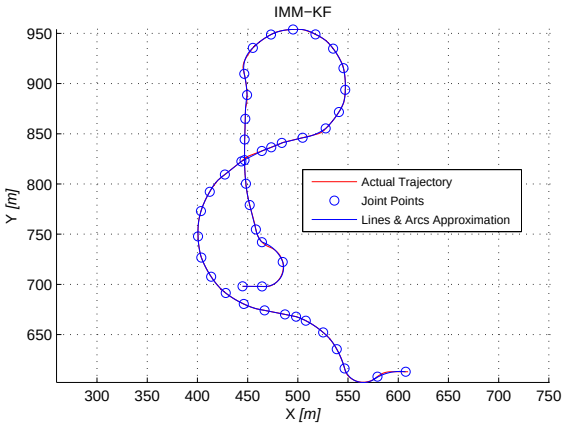
3. SIMULATION RESULTS

Figure 5 illustrates the online mission planning algorithms described in the paper to generate a mission composed only by lines and arcs from the path traced by the target, using polynomial fitting (Fig. 5(a)) and the IMM-KF filter (Fig 5(b)). In this simulation, the trajectory of the target is composed by a set of GPS real data acquired during the experimental tests done in scope of the GREX project (GREX (2006–2009)) in the summer of 2008 in Azores, Portugal.

Figure 6 shows the time evolution of the error between the target trajectory and the mission planning trajectory.



(a) polynomial fitting



(b) IMM-KF

Fig. 5. Simulation results of the online mission planning algorithms using real GPS data.

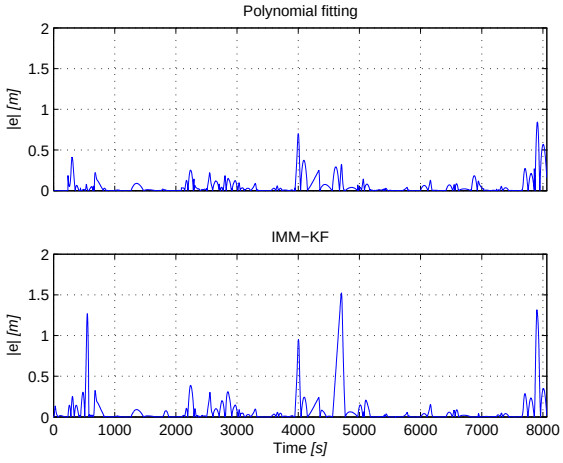
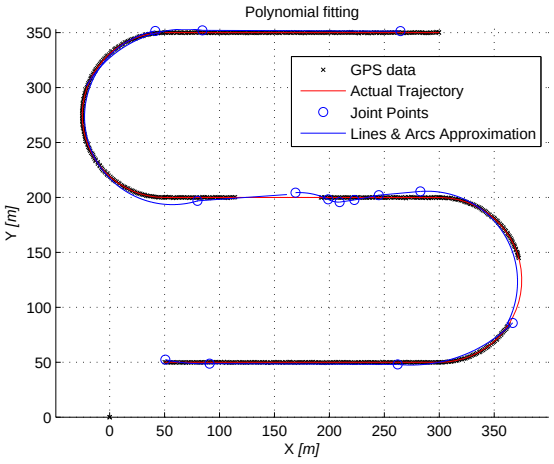
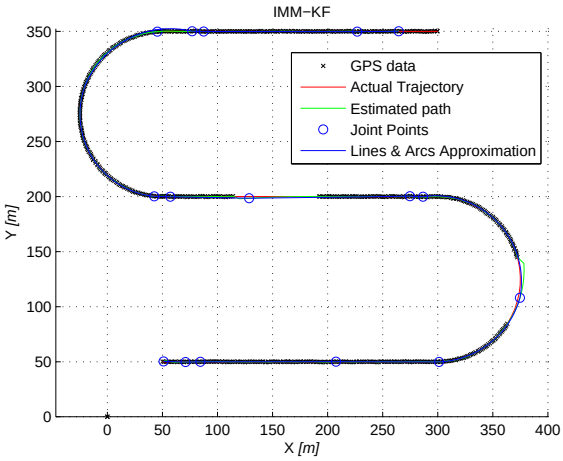


Fig. 6. Time evolution of the norm of the error between the target trajectory and the mission planning trajectory for the real GPS data experiment.



(a) polynomial fitting



(b) IMM-KF

Fig. 7. Simulation results of the online mission planning algorithms using simulated GPS data.

Table 1. Norm and RMS of the errors shown in Fig. 6.

	l_1	l_2	l_∞	RMS
Polynomial fitting	469	10.90	0.84	0.12
IMM-KF fitting	701	19.67	1.52	0.22

See also the computed l_1, l_2, l_∞ and RMS errors in Table 1. From this experiment we can conclude that both the polynomial fitting and IMM-KF perform very well. To test the robustness of the proposed algorithms with respect to GPS failures, a second simulation was performed.

Figures 7, 8 and Table 2 display the results for the case of a lawn mower trajectory corrupted with noise and data loss for periods of 40 and 50 seconds in two sections. As it was expected the IMM-KF approach predicts the target motion (using the kinematic model) when it does not receive data, and therefore the error norm is considerably smaller than the one achieved with the polynomial fitting strategy.

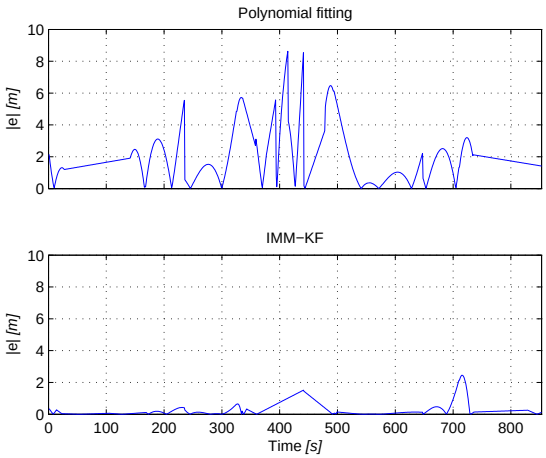


Fig. 8. Time evolution of the norm of the error between the target trajectory and the mission planning trajectory in the simulation for the simulated GPS data experiment

Table 2. Norm and RMS of the errors shown in Fig. 8.

	l_1	l_2	l_∞	RMS
Polynomial fitting	1671	72.8	8.63	2.49
IMM-KF fitting	239	14.8	2.45	0.51

4. CONCLUSION

We proposed two methods to generate missions in real time composed by lines and arcs. These algorithms play a key role on the target following maneuver where a group of vehicles are required to follow a target in formation. To convert the absolute or relative position data available from the positioning system installed on the target (e.g. INS, GPS) into a set of lines and constant curvature arcs one method resorts to a polynomial fitting algorithm and the second method utilizes an interactive multiple-model Kalman filter (IMM-KF). Both approaches produce good results: while the polynomial fitting method is simpler to implement, the IMM-KF approach benefits from a Kalman filter core, so it can predict the trajectory in case of data loss. A rigorous analysis of the impact of sensor noise and failures of communication on system performance is a topic for future research.

REFERENCES

Aguiar, A.P., Almeida, J., Bayat, M., Cardeira, B., Cunha, R., Hausler, A., P., M., Oliveira, A., Pascoal, A.M., Pereira, A., Rufino, M., Sebastiao, L., Silvestre, C., and Vanni, F. (2009). Cooperative autonomous marine vehicle motion control in the scope of the EU GREX project: Theory and Practice. In *Proc. IEEE Conference, Oceans'09*.

Aguiar, A.P. and Pascoal, A.M. (2007). Coordinated path-following control for nonlinear systems with logic-based communication. In *Proc. 46th IEEE Conference on Decision and Control*, 1473–1479.

Alonge, F., D'Ippolito, F., and Raimondi, F. (2001). Trajectory tracking of underactuated underwater vehicles. In *Proc. 40th IEEE Conference on Decision and Control*, volume 5, 4421–4426 vol.5.

Bar-Shalom, Y., Kirubarajan, T., and Li, X.R. (2002). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA.

Blidberg, D.R., Turner, R.M., and Chappell, S.G. (1991). Autonomous underwater vehicles: Current activities and research opportunities. *Robotics and Autonomous Systems*, 7(2-3), 139–150.

Encarnacao, P. and Pascoal, A. (2000). 3D path following for autonomous underwater vehicle. In *Proc. 39th IEEE Conference on Decision and Control*, volume 3, 2977–2982 vol.3.

Fossen, T.I. (1994). *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, Ltd., 2nd edition.

GREX (2006–2009). The GREX project: Coordination and control of cooperating heterogeneous unmanned systems in uncertain environments. URL `http://www.grex-project.eu`.

Jiang, Z.P. (2002). Global tracking control of underactuated ships by lyapunov's direct method. *Automatica*, 38(2), 301 – 309.

Klein, D.J., Bettale, P.K., Triplett, B.I., and Morgansen, K.A. (2008). Autonomous underwater multivehicle control with limited communication: Theory and experiment. In *Proc. NGCUV'08 - IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*.

Lefeber, E., Pettersen, K., and Nijmeijer, H. (2003). Tracking control of an underactuated ship. *IEEE Transactions on Control Systems Technology*, 11(1), 52–61.

Leonard, N. (1995). Control synthesis and adaptation for an underactuated autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 20(3), 211–220.

Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. (1998). Interacting multiple model methods in target tracking: a survey. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(1), 103–123.

Vanni, F., Aguiar, A.P., and Pascoal, A.M. (2008). Cooperative path-following of underactuated autonomous marine vehicles with logic-based communication. In *Proc. NGCUV'08 - IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*.