

# Augmented State Kalman Filtering for AUV Navigation

R. Garcia, J. Puig, P. Ridao and X. Cufi

Computer Vision and Robotics Group

Institute of Informatics and Applications

University of Girona, E.P.S.

17071 Girona, Spain

e-mail: {rafa,jpuig,pere,xcufi}@eia.udg.es

## Abstract

*This paper addresses the problem of estimating the motion of an Autonomous Underwater Vehicle (AUV), while it constructs a visual map (“mosaic” image) of the ocean floor. The vehicle is equipped with a down-looking camera which is used to compute its motion with respect to the seafloor. As the mosaic increases in size, a systematic bias is introduced in the alignment of the images which form the mosaic. Therefore, this accumulative error produces a drift in the estimation of the position of the vehicle. When the arbitrary trajectory of the AUV crosses over itself, it is possible to reduce this propagation of image alignment errors within the mosaic. A Kalman filter with augmented state is proposed to optimally estimate both the visual map and the vehicle position.*

## 1 Introduction

Creating visual maps of the ocean floor is an important tool for underwater navigation. Consider an AUV equipped with a down-looking camera, which provides images of the seabed as the vehicle moves. The alignment of these images provides the necessary information to estimate the position and orientation of the vehicle [1,2]. At the same time, warping the aligned images creates a visual map, known as *mosaic*, which can be used for planning future missions [3]. While the mosaic is constructed, the vehicle can localize itself in this map, following the *Concurrent Mapping and Localization* methodology [4,5].

Unfortunately, as the vehicle moves, cumulative error is introduced in the alignment of consecutive images within the mosaic, and therefore in the estimation of the position of the vehicle. This drift can be corrected by taking the submersible periodically to the surface to perform a GPS reading. However, this is a very inefficient solution when the submersible carries out a mission in deep sea, and new errors can be accumulated while the robot goes up and down through the water column. An alternative solution consists in placing various acoustic emitter/receiver devices covering the area of interest. Then, this *acoustic transponder network* (ATN) can be used to obtain absolute 3D position measures of the vehicle when navigating

within the volume covered by the network. This option, though accurate, has the disadvantages of its high operating cost, and limited navigation area.

For the case in which it is not possible to obtain absolute positioning measures (GPS or ATN), Stanford/MBARI researchers proposed to exploit the additional information gained when the vehicle path crosses itself [6]. When performing a mission, the submersible follows an arbitrary path, while the mosaicking system constructs a visual map of the surveyed area. As the mosaic increases in size, image local alignment errors increase the error margin associated to the position of the vehicle. Occasionally, this path may cross over itself. In this situation new information is available, and the system can readjust the position estimates. Then, the uncertainty concerning vehicle position can be reduced, since the images that have been previously added to the mosaic have a smaller error variance. The technique proposed by Stanford/MBARI was based on an iterative smoother-follower filter [6,7]. In this paper, we propose the use of a Kalman Filter to both (a) forward *filter* the position of the vehicle, and (b) backward *smooth* the vehicle trajectory at crossover points.

## 2 Overview of the System

Our testing platform is the GARBI underwater vehicle [8]. Among other sensors, GARBI is equipped with a down-looking camera and a sonar altimeter. When the robot navigates close to the ocean floor, the altimeter provides the distance from the robot to the ocean floor ( $z$ ), while a mosaicking algorithm provides an estimation of its 2D motion and yaw orientation [1,9]. This estimation requires previous knowledge of the intrinsic parameters of the camera and periodic altimeter readings. The coordinate system of GARBI is located at the focal point of the camera. Due to the distribution of weight, the vehicle is stable in pitch and roll. The differences in elevation of the floor relief are assumed to be negligible with respect to the navigation altitude of the vehicle. Every time an image is processed by the mosaicking module, an incremental motion estimation is obtained for  $x$ ,  $y$  and yaw orientation ( $\Delta_x$ ,  $\Delta_y$  and  $\Delta_\psi$ ). These measures are incremental with respect to the previous measurement. Therefore, they provide a measure of the motion of the camera (and the vehicle) between the acquisition of two consecutive images.

### 3 Positioning from KF smoothing

#### 3.1 Introduction

As described before, the perception system accumulates the positioning errors associated to every incremental measurement provided by the mosaicking algorithm, producing an unbounded error in absolute positioning. Therefore, camera position variances will increase monotonically along the sequence, and at the same time the quality of the mosaic will decrease. In order to detect a crossing path, we use the technique proposed by Fleischer *et al.* [6,7]. It consists of exploiting the information provided by the error variance window associated to every image, which can be used to detect when the vehicle path loops back upon itself. Since this window represents the bounded area in which the image is located, it is necessary to check if the area covered by the error variance of the current image intersects the mosaic image in an already surveyed zone. In this case, both images are correlated and a new estimation of the vehicle location is obtained.

Once the crossover has been detected, the next step consists in re-aligning the sequence of images that form the mosaic, taking into account: (a) the incremental measures computed by the mosaicking; and (b) the crossover data. This implies to estimate, at time  $k$ , the position and orientation of the vehicle prior to time  $k$ . The estimation of the state of a system  $\hat{\mathbf{x}}_i$  at time instant  $i$ , based on the measures up to time  $k$ , with  $i < k$ , is known as *smoothing* in the literature [10]. An optimal estimation technique is needed to minimize the variances on the image positions. We propose to perform this optimization by means of classical filtering techniques, estimating the vehicle motion and, at the same time, use these estimations to improve mosaic alignment when crossovers occur.

The proposed strategy consists of developing a Kalman filter [11,12] capable of dynamically estimating both the current vehicle position and past trajectory. The state vector  $\mathbf{x}(k)$  of our filter includes position and yaw orientation of the submersible. Therefore, the vehicle is assumed to be passively stable in pitch and roll, since its center of mass is below its center of buoyancy.

At every time step, the mosaicking system measures local displacements, which are converted into global position estimates build upon consolidating every image into the global mosaic frame. The measured local displacements are always referenced to a node of the image chain that form the mosaic. The nodes are defined by the location of the central point of an image. Normally, this node is the previous image of the sequence, but eventually it could be another node of the image chain as a consequence of crossover detection. A general block diagram of the overall smoothing process is illustrated in Figure 1. The incremental information provided by the mosaicking system (dashed box at the left of the Figure)

feeds the smoother module, which provides a new state estimation. This estimation is used to update the mosaic and vehicle positioning, as well as providing information for detection of future crossovers.

Therefore, the state vector of the system has to keep the information regarding the position of the center of all the images of the sequence. Moreover, since measurements are incremental, it is necessary to keep track of all measures. This is not possible with a standard state vector  $\mathbf{x}(k)$ , since  $\mathbf{x}(k)$  contains only information about the current state of the system (at time step  $k$ ). This problem can be solved by augmenting the state vector every time a new measure has been added to the system. Therefore, the state vector of our filter has the following form:

$$\mathbf{x}_{aug}(k) = [\mathbf{x}_v^T(k) \quad \mathbf{x}_{k-1}^T(k) \quad \mathbf{x}_{k-2}^T(k) \quad \dots \quad \mathbf{x}_0^T(k)]^T \quad (1)$$

where  $\mathbf{x}_v(k)$  is the state of the vehicle and  $\{\mathbf{x}_i(k), i = 0, \dots, k-1\}$  are the locations of the central point of the first  $k$  images which form the mosaic. When augmenting the state vector, the current estimate error covariance has also to be augmented, to reflect the new state, giving rise to:

$$\mathbf{P}_{aug}(k) = \begin{bmatrix} \mathbf{P}_{v,v}(k) & \mathbf{P}_{v,k-1}(k) & \mathbf{P}_{v,k-2}(k) & \dots & \mathbf{P}_{v,0}(k) \\ \mathbf{P}_{k-1,v}(k) & \mathbf{P}_{k-1,k-1}(k) & \mathbf{P}_{k-1,k-2}(k) & \dots & \mathbf{P}_{k-1,0}(k) \\ \mathbf{P}_{k-2,v}(k) & \mathbf{P}_{k-2,k-1}(k) & \mathbf{P}_{k-2,k-2}(k) & \dots & \mathbf{P}_{k-2,0}(k) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{0,v}(k) & \mathbf{P}_{0,k-1}(k) & \mathbf{P}_{0,k-2}(k) & \dots & \mathbf{P}_{0,0}(k) \end{bmatrix} \quad (2)$$

which is obviously symmetric and where submatrix  $\mathbf{P}_{v,v}(k)$  represents the covariance of the vehicle (uncertainty in the vehicle's position at time  $k$ ); and submatrices  $\mathbf{P}_{v,j}(k)$  and  $\mathbf{P}_{i,j}(k)$  are the covariance between the vehicle and the  $j^{\text{th}}$  image and the covariance between the  $i^{\text{th}}$  image and the  $j^{\text{th}}$  image, respectively.

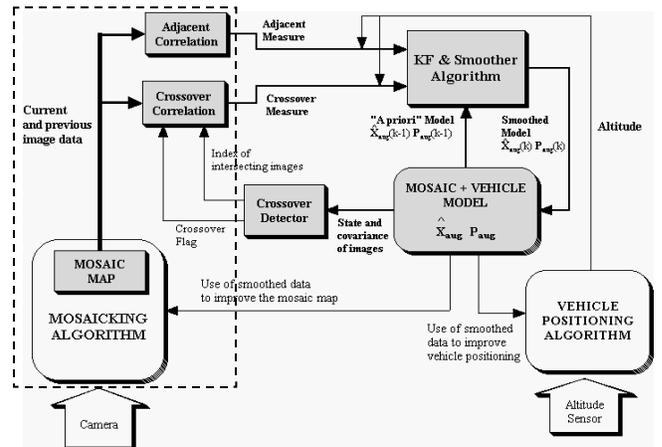


Figure 1. Block diagram of the KF smoother for global state estimation.

This leads to the implementation of an *augmented state Kalman filter* (ASKF) [13,14], which integrates either the filtering of the vehicle motion and the smoothing of the mosaic based on intersecting trajectories. This approach has several advantages:

- It is able to integrate all the available information: vehicle's dynamic model, correlation of consecutive (adjacent) images, crossover correlation and other sensor measurements (e.g. sonar-based altimeter).
- It continuously updates, with a simple procedure, the state of the vehicle and that of the images which form the mosaic. At the same time, it updates their associated covariances: vehicle-to-vehicle, vehicle-to-image and image-to-image.
- It permits dealing with trajectories of any complexity (i.e. multiple loops) in a simple manner. If multiple crossovers occur, the state estimation error covariance  $\mathbf{P}_{aug}(k)$  would evolve accordingly to the complexity of the trajectory, due to smoothing in intermediate loops.

Therefore, ASKF is a good framework to keep track of the state of the vehicle and those of every image of the mosaic images; containing all this information in a single state vector. However, it has the drawback of the increasing size of the matrices involved in the filter.

### 3.2 Theoretical Principles

From the description presented above, the state estimation  $\hat{\mathbf{x}}_{aug}(k)$  and its associated covariance  $\mathbf{P}_{aug}(k)$  are propagated according to KF *time update* equations:

$$\hat{\mathbf{x}}_{aug}^-(k+1) = \mathbf{A}_{aug}(k) \hat{\mathbf{x}}_{aug}(k) + \mathbf{B}_{aug}(k) \mathbf{u}_{aug}(k) \quad (3)$$

$$\mathbf{P}_{aug}^-(k+1) = \mathbf{A}_{aug}(k) \mathbf{P}_{aug}(k) \mathbf{A}_{aug}^T(k) + \mathbf{B}_{aug}(k) \mathbf{Q}_{aug}(k) \mathbf{B}_{aug}^T(k) \quad (4)$$

where, as the position of images does not vary as a function of time, the system dynamics  $\mathbf{A}_{aug}(k)$  and the system noise covariance  $\mathbf{Q}_{aug}(k)$  are:

$$\mathbf{A}_{aug}(k) = \begin{bmatrix} \mathbf{A}_v(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}; \quad \mathbf{Q}_{aug}(k) = \begin{bmatrix} \mathbf{Q}_v(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (5)$$

where the identity matrix  $\mathbf{I}$  has a size  $k \cdot \dim(\mathbf{x}_i)$ . Since the system does not have any input,  $\mathbf{u}(k) = \mathbf{0}$  and  $\mathbf{B}(k) = \mathbf{I}$ .

At every time step  $k$ , the mosaicking system finds the registration parameters between two consecutive (adjacent) images. Therefore, a new measure  $\mathbf{z}_{adj}(k)$  is obtained at every time step. However, when a crossover is detected, an additional measure  $\mathbf{z}_{cross}(k)$  is obtained.

In the case of registration of consecutive images (adjacent case), the measure  $\mathbf{z}(k)$  measures the position of the  $k^{\text{th}}$  image (which corresponds to the position of the vehicle) with respect to the  $(k-1)^{\text{th}}$  image, so that:

$$\mathbf{z}(k) = \mathbf{z}_{adj}(k) \quad (6)$$

$$\mathbf{H}_{aug}(k) = [\mathbf{H}_v(k) \quad -\mathbf{H}_{k-1}(k) \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \quad (7)$$

However, when a crossover is detected, the current image  $k^{\text{th}}$  also intersects with the mosaic image. Then, the measurement vector  $\mathbf{z}(k)$  becomes:

$$\mathbf{z}(k) = [\mathbf{z}_{adj}^T(k) \quad \mathbf{z}_{cross}^T(k)]^T \quad (8)$$

which means that we have two measures, a measure with respect to the previous image  $\mathbf{z}_{adj}(k)$ , and a second one with respect to the area where the crossover has been detected  $\mathbf{z}_{cross}(k)$ . If the crossover corresponds to an image  $j$ , the measurement matrix  $\mathbf{H}_{aug}(k)$  incorporates a measurement in column  $j$ , becoming:

$$\mathbf{H}_{aug}(k) = \begin{bmatrix} \mathbf{H}_v(k) & -\mathbf{H}_{k-1}(k) & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{H}_v(k) & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & -\mathbf{H}_j(k) & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \quad (9)$$

The form of the measurement submatrix of the vehicle  $\mathbf{H}_v(k)$ , image measurement submatrices  $\mathbf{H}_{k-1}(k)$  and  $\mathbf{H}_j(k)$ , and measuring subvectors  $\mathbf{z}_{\{adj,cross\}}$  are detailed in section 3.3.

Then, difference between the measurement  $\mathbf{z}(k)$  and the previous ‘‘a priori’’ estimation is given by:

$$\mathbf{r}(k) = \mathbf{z}_{aug}(k) - \mathbf{H}_{aug}(k) \hat{\mathbf{x}}_{aug}^-(k) \quad (10)$$

This difference is called *innovation*, and its covariance  $\mathbf{S}(k)$  is:

$$\mathbf{S}(k) = \mathbf{H}_{aug}(k) \mathbf{P}_{aug}^-(k) \mathbf{H}_{aug}^T(k) + \mathbf{R}(k) \quad (11)$$

where  $\mathbf{z}_{aug}(k)$  is the measuring vector,  $\mathbf{R}(k)$  the measurement error covariance and  $\mathbf{H}_{aug}(k)$  is the measurement matrix.

The measures allow the correction of the estimated state and its associated covariance are corrected according to the KF *measurement update* equations. So, the filter gain can be expressed as:

$$\mathbf{K}(k) = \mathbf{P}_{aug}^-(k) \mathbf{H}_{aug}^T(k) \mathbf{S}(k)^{-1} \quad (12)$$

the corrected state estimate corresponds to:

$$\hat{\mathbf{x}}_{aug}(k) = \hat{\mathbf{x}}_{aug}^-(k) + \mathbf{K}(k)(\mathbf{z}(k) - \mathbf{H}_{aug}(k) \hat{\mathbf{x}}_{aug}^-(k)) \quad (13)$$

and its error covariance becomes:

$$\mathbf{P}_{aug}(k) = (\mathbf{I} - \mathbf{K}(k) \mathbf{H}_{aug}(k)) \mathbf{P}_{aug}^-(k) (\mathbf{I} - \mathbf{K}(k) \mathbf{H}_{aug}(k))^T + \mathbf{K}(k) \mathbf{R}(k) \mathbf{K}(k)^T \quad (14)$$

It has been proved in [10] that equation (14), which updates the error covariance, is better conditioned for numerical computation than the equivalent usual form shown in equation (15).

$$\mathbf{P}_{aug}(k) = (\mathbf{I} - \mathbf{K}(k) \mathbf{H}_{aug}(k)) \mathbf{P}_{aug}^-(k) \quad (15)$$

Once the phases of **propagation** and **correction** have been completed, the state and covariance are augmented to add the positioning of the new  $k^{\text{th}}$  image.

### 3.3 Implementation

The augmented Kalman filter, as described in the previous section, allows the estimation of the state of both the vehi-

cle and the images of the mosaic. This section introduces the assumed dynamic model of the vehicle, the state vector of the images, and some characteristics of the measurement model and the addition of new images to the filter.

### Vehicle Model

Although an accurate dynamic model of the vehicle which has been used in simulation is available [15], our KF approach assumes a mathematical description based on a linear model. This assumption is made to obtain a more generic and simple filter, which can be more easily adapted to other submersibles. The vehicle state is described by its position and velocity in the following way:

$$\mathbf{x}_v(k) = [x \ y \ z \ \Psi \ \dot{x} \ \dot{y} \ \dot{z} \ \dot{\Psi}]^T \quad (16)$$

where  $(x,y)$  are relative to a mosaic-fixed coordinate system,  $z$  is relative to an earth fixed coordinate system and  $\Psi$  (*yaw*) is the heading of the vehicle with respect to a fixed coordinate system.

The considered dynamic model of the vehicle  $\mathbf{A}_v(k)$  is

$$\mathbf{A}_v(k) = \begin{bmatrix} \mathbf{I}_{4 \times 4} & dt \cdot \mathbf{I}_{4 \times 4} \\ \mathbf{0}_{4 \times 4} & \mathbf{I}_{4 \times 4} \end{bmatrix} \quad (17)$$

where  $\mathbf{I}$  is the 4-dimension identity, and  $dt$  is the sampling period. Finally, the process noise  $\mathbf{Q}_v(k)$  is given by [5,14]:

$$\mathbf{Q}_v(k) = \begin{bmatrix} \frac{1}{4} dt^4 \cdot \sigma_v^2 & \frac{1}{2} dt^3 \cdot \sigma_v^2 \\ \frac{1}{2} dt^3 \cdot \sigma_v^2 & dt^2 \cdot \sigma_v^2 \end{bmatrix} \quad (18)$$

where  $\sigma_v^2$  is a diagonal 4-dimensional matrix of process noise variances in  $(x,y,z,\Psi)$  coordinates, used as tuning parameters.

### Image model

Every image has an associated state vector which contains the information required to position the corresponding image in the mosaic, so that:

$$\mathbf{x}_i(k) = [x_i \ y_i \ z_i \ \Psi_i]^T \quad (i=0, \dots, k-1) \quad (19)$$

The vectors measuring the displacement with respect to the previous image  $\mathbf{z}_{adj}(k)$ , and the mosaic area where the crossover has been detected  $\mathbf{z}_{cross}(k)$ , are described by

$$\mathbf{z}_{\{adj, cross\}}(k) = [\Delta_x \ \Delta_y \ z \ \Delta_\Psi]^T \quad (20)$$

where  $\Delta_x$ ,  $\Delta_y$  and  $\Delta_\Psi$  are the coordinates of the position of the present image with respect to the previous image (“*adj*” subindex) or with respect to the closer node of the mosaic image (“*cross*” subindex). On the other hand,  $z$  represents the altitude of the vehicle at the time the present image has been taken. This absolute measurement can be obtained from a sonar altimeter.

### Measurement matrix

Therefore, the vehicle measurement submatrix  $\mathbf{H}_v(k)$  should

be defined as

$$\mathbf{H}_v(k) = [\mathbf{I}_{4 \times 4} \ \mathbf{0}_{4 \times 4}] \quad (21)$$

and image measurement submatrices  $\mathbf{H}_{k-1}(k)$  and  $\mathbf{H}_j(k)$

$$\mathbf{H}_{k-1}(k) = \mathbf{H}_j(k) = \text{diag}\{1,1,0,1\} \quad (22)$$

Note that the component corresponding to measure  $z$  in equation (27) is not updated from the image, but directly provided by the altimeter sensor.

The measurement covariance matrix in the case of adjacent measure is  $\mathbf{R}(k) = \sigma_{adj}^2(k)$ . However, if there is a crossover measurement in addition to the adjacent one,  $\mathbf{R}(k)$  becomes:

$$\mathbf{R}(k) = \begin{bmatrix} \sigma_{adj}^2(k) & \mathbf{0}_{4 \times 4} \\ \mathbf{0}_{4 \times 4} & \sigma_{cross}^2(k) \end{bmatrix} \quad (23)$$

where  $\sigma_{\{adj, cross\}}^2(k) = \text{diag}\{\sigma_x^2(k), \sigma_y^2(k), \sigma_z^2(k), \sigma_\Psi^2(k)\}$ .  $\sigma_x^2$ ,  $\sigma_y^2$  and  $\sigma_\Psi^2$  are the measurement variances of image correlation in the mosaic, and  $\sigma_z^2$  the variance of the sonar altimeter.

### State augmentation

Finally, the information of the present image which is necessary to augment the state vector and its covariance matrix can be obtained from the terms relative to the vehicle. Therefore:

$$\hat{\mathbf{x}}_v(k+1) = [\mathbf{I}_{4 \times 4} \ \mathbf{0}_{4 \times 4}] \hat{\mathbf{x}}_v(k+1) \quad (24)$$

$$\mathbf{P}_{k_i(v,k,k-1,\dots,0)}(k+1) = [\mathbf{I}_{4 \times 4} \ \mathbf{0}_{4 \times 4}] \mathbf{P}_{v_i(v,v,k-1,\dots,0)}(k+1) \quad (25)$$

where equation (25) selects the information from the row and column related to the vehicle position.

### 3.4 Optimizations

In order to take advantage of the sequential character of the filter, we should pay attention to its computational cost. Obviously, as the ASKF incorporates new measures at every iteration, its size will increase. Given that it essentially involves matrix multiplications, the cost will be approximately  $O(n^3)$ , where  $n$  is the number of images added to the mosaic. However, in this implementation, the cost can be significantly reduced considering the trivial submatrices (zeros and identities) in  $\mathbf{A}_{aug}(k)$ ,  $\mathbf{Q}_{aug}(k)$  and  $\mathbf{H}_{aug}(k)$ . Then, only the products which involve non-trivial submatrices have to be computed, and then the trivial and non-trivial parts of the matrices can be linked together to form the final matrix. In this way the computational cost can be reduced from  $O(n^3)$  to  $O(n)$ .

Although this improvement is quite significant, as state augments it becomes more and more difficult to obtain real-time performance. Therefore, the number of images which are added to the state should be kept to a minimum. Then, although new images are processed at constant time intervals, incremental position estimations can be injected into the filter at a given sampling rate, instead of using all

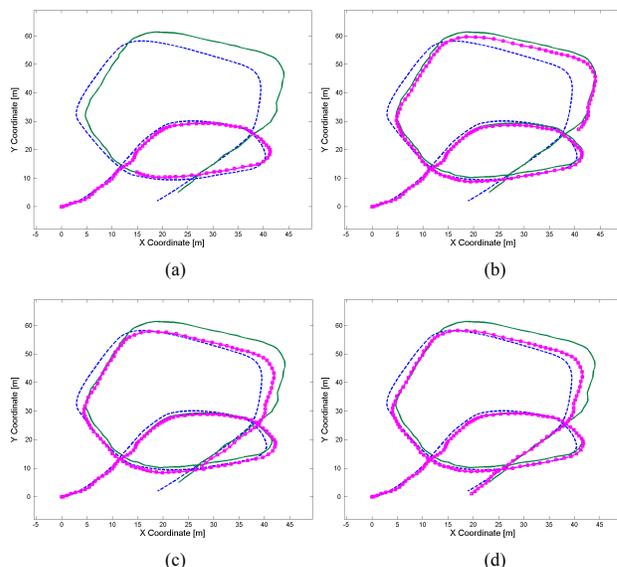
the images to update the filter. In this way, the matrices involved in the computations of the ASKF do not increase so rapidly.

## 4 Results

In order to create a set of test trajectories, the GARBI Autonomous Underwater Vehicle Simulator (AUVS) for virtual and/or real applications [16] has been used. This simulator incorporates the identified dynamic model of GARBI, and it is able to create the sort of trajectories that the vehicle would follow in a real mission. Moreover, the mosaic-based perception system has been modeled in the GARBI simulator. Every time a new image is added to the mosaic, an incremental displacement measure referred to the previous image is obtained. As derived from our experimentation [17], every new measure is affected by a zero-mean Gaussian error. In this way, accumulated drift error can be simulated.

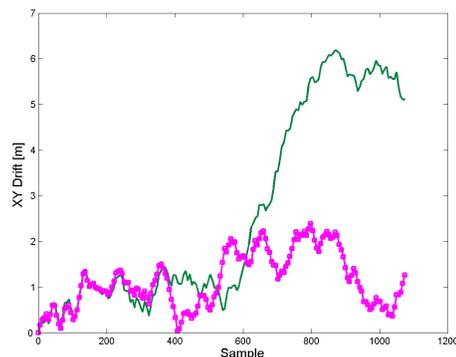
The simulator has been used to generate a trajectory with 3 crossing paths, with the vehicle navigating in an area of 40×50 meters. In addition to the real trajectory, the simulator provides the estimated trajectory provided by the perception system. Figure 2 shows the sample trajectory including 3 crossover paths. The dashed blue line represents the real trajectory and the trajectory estimated by the mosaicking system is shown in solid green. The path of the vehicle starts at the bottom left of the image. The evolution of the smoothed trajectory can be followed in the different sub-figures (illustrated in magenta with a marker at every sample point). Figure 2(a) shows the trajectory filtered by the ASKF before the first crossover is detected. It basically follows the trajectory computed by the mosaicking system (solid-green). It can be observed that the smoothed trajectory stops before intersecting its path. This means that the *crossover detector* tells the robot that it has already arrived to a crossover point, although the robot thought that it was further from that point. With this information, the ASKF smooths back the positions of the previous images, and then it goes on filtering the vehicle trajectory (Figure 2(b)). Again, an intersection is detected. It can be observed in Figure 2(c) that the smoothed trajectory in the top of the map goes down, after the second crossover, approaching the real trajectory. Finally, Figure 2(d) shows the smoothed trajectory after the third path intersection. Here we have shown the evolution of the smoothed trajectory superimposed on the final real and measured paths; however, it should be noted that the smoothed trajectory is updated sequentially, as the mosaic provides every new measurement.

The accumulated drift is illustrated in Figure 3. Initially, both the trajectory computed by the mosaic and the ASKF smoother present a similar drift. When the first crossover is detected, drift of the smoothed path can be reduced to a

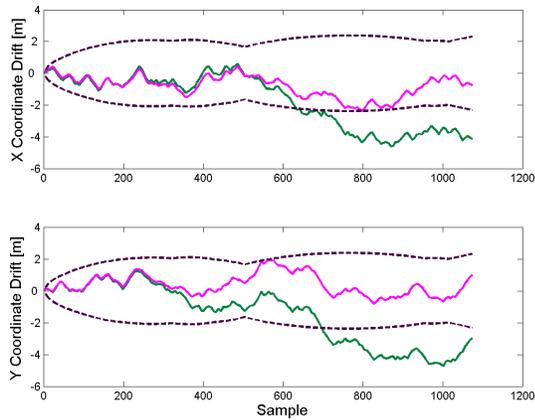


**Figure 2. Sample trajectory with 3 intersections. The sequence shows the real trajectory (dashed-blue), non-smoothed estimated trajectory (solid-green) and the evolution of the smoothed trajectory (magenta with markers) as new crossovers are detected.**

very low value. Then it increases again, but the second loop keeps the drift in a considerably smaller range than the measured trajectory. Figure 4 shows the drift independently plotted for the  $x$  and  $y$  coordinates. The diagonal components of the final state error covariance  $\mathbf{P}_{aug}$  are used to compute the uncertainty bounds of the smoothed trajectory, drawn as 3 times the estimated standard deviation at every point.



**Figure 3. Drift evolution of the vehicle trajectory in the mosaic plane. Drift of the smoothed (magenta with markers) and non-smoothed (solid-green) trajectories.**



**Figure 4. Drift of the smoothed (magenta) and non-smoothed (solid-green) trajectories for the X and Y coordinates. The uncertainty bounds of the smoothed trajectory (dashed) is drawn as 3 times the estimated std. deviation (obtained from the final  $P_{aug}$ ).**

## 5 Conclusions

In this paper we have presented a positioning system for AUV navigation. A method for the optimal estimation of the trajectory of the vehicle after a successful crossover path has been described.

Kalman filtering with augmented state (ASKF) has proved to be the adequate framework for the development of the optimal estimator. Although the idea of taking profit of additional information when the vehicle path crosses itself is not new, our approach presents several advantages with respect to previous systems. First, the system is able to cope with several loops. Second, it is a sequential algorithm. Therefore it can optimize dynamically as new data gets into the system, instead of having to wait for all the data to process it afterwards, like batch filters. Finally, the filter performs forward iterations which allow the system to estimate the trajectory from the noisy data.

## References

- [1] R. Garcia, J. Batlle, X. Cufi, and J. Amat, "Positioning an Underwater Vehicle through Image Mosaicking," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2779–2784, Seoul, Rep. of Korea, 2001.
- [2] S. Negahdaripour, X. Xu, A. Khamene and Z. Awan, "3D motion and depth estimation from sea-floor images for mosaic-based station-keeping and navigation of ROVs/AUVs and high-resolution sea-floor mapping," in *Proc. IEEE W. on Autonomous Underwater Robots*, pp. 191–200, 1998.
- [3] N. Gracias and J. Santos-Victor, "Underwater Video Mosaics as Visual Navigation Maps," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 66–91, 2000.
- [4] C. Smith, J. Leonard, A. Bennett and C. Shaw, "Feature-based concurrent mapping and localization for AUVs," in *Proc. of the MTS/IEEE OCEANS Conf.*, pp. 896–901 1997.
- [5] I. Tena Ruiz, Y. Petillot, D.M. Lane and C. Salson, "Feature Extraction and Data Association for AUV Concurrent Mapping and Localisation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, pp. 2785–2790, 2001.
- [6] S.D. Fleischer, H.H. Wang, S.M. Rock and M.J. Lee, "Video Mosaicking Along Arbitrary Vehicle Paths," in *Proc. of the OES/IEEE Symp. on Autonomous Underwater Vehicle Technology*, pp. 293–299, Monterey, CA, 1996.
- [7] S.D. Fleischer, S.M. Rock and R.L. Burton, "Global Position Determination and Vehicle Path Estimation from a Vision Sensor for Real-Time Video Mosaicking and Navigation," in *Proc. of the MTS/IEEE OCEANS Conference*, Halifax, Nova Scotia, vol. 1, pp. 641–647, 1997.
- [8] J. Amat, J. Batlle, A. Montferrer, J. Salvi and P. Ridao, "Capabilities of GARBI-A Low cost underwater Vehicle," Workshop on Recent trends in Intelligent Underwater Robotics, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, Canada, 1998.
- [9] R. Garcia, X. Cufi and J. Batlle, "Detection of Matchings in a Sequence of Underwater Images through Texture Analysis," *IEEE Int. Conf. on Image Processing*, vol. 1, pp. 361–364, Thessaloniki, Greece, 2001.
- [10] A.H. Jazwinski, "Stochastic Processes and Filtering Theory," Academic Press, New York, 1970.
- [11] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME Journal of Basic Engineering*, pp. 35–45, 1960.
- [12] R.E. Kalman and R.S. Bucy, "New results in linear filtering and prediction theory," *Transactions of the ASME Journal of Basic Engineering*, pp. 95–108, 1961.
- [13] R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in I.J. Cox, and G.T. Wilfon (Eds.): *Autonomous Robot Vehicles*, Springer-Verlag, 1990.
- [14] R.H. Deaves, "Covariance bounds for augmented state Kalman filter application," *IEEE Electronics Letters*, vol. 35, no. 23, pp. 2062–2063, 1999.
- [15] P. Ridao, J. Batlle and M. Carreras, "Model identification of a low-speed UUV with on-board sensors," in *Proc. of the IEEE Control Applications in Marine Systems Conf.*, Scotland, U.K., 2001.
- [16] P. Ridao, J. Batlle and M. Carreras, "An underwater autonomous agent: from simulation to experimentation," in *Proc. of the IEEE Mediterranean Conf.*, Croatia, 2001.
- [17] R. Garcia, J. Batlle and X. Cufi, "A System to Evaluate the Accuracy of a Visual Mosaicking Methodology," in *Proc. of the MTS/IEEE OCEANS Conf.*, vol. 4, pp. 2570–2576 Hawaii, 2001.