2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Improving Underwater Vehicle Navigation State Estimation Using Locally Weighted Projection Regression

Georgios Fagogenis[1], David Flynn[2] and David M. Lane[1]
{gf63, D.Flynn, D.M.Lane}@hw.ac.uk

***Abstract*— Navigation is instrumental in the successful deployment of Autonomous Underwater Vehicles (AUVs). Sensor hardware is installed on AUVs to support navigational accuracy. Sensors, however, may fail during deployment, thereby jeopardizing the mission. This work proposes a solution, based on an adaptive dynamic model, to accurately predict the navigation of the AUV. A hydrodynamic model, derived from simple laws of physics, is integrated with a powerful non-parametric regression method. The incremental regression method, namely the Locally Weighted Projection Regression (LWPR), is used to compensate for un-modeled dynamics, as well as for possible changes in the operating conditions of the vehicle. The augmented hydrodynamic model is used within an Extended Kalman Filter, to provide optimal estimations of the AUV's position and orientation. Experimental results demonstrate an overall improvement in the prediction of the vehicle's acceleration and velocity.**

***Index Terms*— underwater navigation, adaptive model, dead reckoning, sensor failure**

I. INTRODUCTION

Autonomous Underwater Vehicles (AUVs) are used ever more in commercial, military and research missions. Inspection of underwater infrastructure, mine detection and scientific data gathering are typical examples of AUV applications. The success of the latter relies on the accuracy of the navigation. The navigation system pertains the computation of the vehicle's pose (position and orientation with respect to a fixed coordinate frame), given the data from its sensors.

For this purpose, various sensors have been developed and deployed on AUVs. Inertial Measurement Units (IMUs) measure the acceleration along the three axes of rotation (roll, pitch, yaw). Most recently, IMUs integrate the measured accelerations on-line, thus providing estimations of the angular velocities and positions. Likewise, linear velocities are measured using Doppler-based sensors. Integration of the latter provides estimations of the vehicle position. Absolute measurements are also acquired during the mission. Magnetic compasses provide an absolute measurement of the vehicle's orientation, whereas Global Positioning System (GPS) sensors estimate the vehicle's absolute position. This is accomplished by the trilateration of satellite signals. Under water, however, the reception of such signals is limited. To this end, AUV's often surface amidst mission execution, to get a GPS fix. An alternative way, to exploit the benefits of a GPS system, is with the aid of a surface vehicle. The relative position of the two vehicles is computed using a Long-Baseline (LBL) Acoustic Positioning System. Given the relative position of the two vehicles and the GPS fix from the surface vehicle, the AUV corrects its position estimation. Hence, GPS technologies are still applicable, albeit they increase the complexity of the navigation system.

Alternatively, Simultaneous Localization and Mapping (SLAM) techniques [1] compute the position of the robot with respect to a map of the environment. The latter is either given a priori or it is created on line. The map is constructed by accumulating position information about observed landmarks in the environment. While moving, the robot predicts the expected position of the aforementioned landmarks. This is accomplished by combining its previous position estimation, its kinematic model and the map, as constructed until that specific time instant. Following that, the robot attempts to match current observations with the expected landmarks in sight. In this manner, the robot updates its belief about its location in the map. Localization relative to landmarks is very practical for robotic applications. SLAM, however, relies on the assumption, that the environment is rich in features, which can be perceived by the robot. This assumption is often violated in the underwater domain.

Since the direct measurement of the robot's position is difficult and SLAM methods are not always applicable, underwater navigation depends strongly on dead reckoning. Given the robot's initial position and orientation, the measurements from the navigation sensors (mainly from the IMUs and the Doppler Velocity Log (DVL)) are integrated to yield the robot's current pose. Nevertheless, in the case of a sensor failure, the robot has no means to navigate further. Consequently, the success of the mission is compromised. Moreover, without accurate navigation, it may be impossible to retrieve the robot. As a remedy, the dynamics are used to estimate the robot's position and help recover it from the water.

In this paper, a novel algorithm for underwater navigation is presented. The algorithm is based on the robot's hydrodynamic model, combined with a non-linear additive term. The latter compensates for unmodeled non-linearities, as well as for possible variance in the operating conditions. The corrective term is approximated by the Locally Weighted Projection Regression algorithm [2], an incremental method for non-linear function approximation with high dimensional input spaces. The augmented hydrodynamic model (i.e. the original hydrodynamic model together with the non-linear corrective term) is used within the prediction step of an Ex-

[1]Ocean Systems Laboratory, School of Physical Sciences, Heriot Watt University, Edinburgh EH14 1AS, United Kingdom
[2]Microsystems Engineering Centre (MiSEC) , School of Physical Sciences, Heriot Watt University, Edinburgh EH14 1AS, United Kingdom

tended Kalman Filter (EKF). The accuracy of the augmented model's predictions exceeds that of the plain hydrodynamic model.

This paper is organized as follows: In Section II, relevant work is summarized and contrasted with our approach. Next, the algorithm , as well as the tuning process of the LWPR, is described in section III. Section IV demonstrates the performance of the algorithm on experimental data. The paper concludes with a summary of the results and directions for future work.

II. RELEVANT WORK

AUV Navigation has long attracted the interest of the robotics community. Early research in this field was mostly focused on sensor development. As a result, several authors discussed the deployment of navigation sensors, such as Inertial Measurement Units (IMUs), Doppler-Based sensors etc. A good review on the early advances of underwater navigation research can be found in [3].

Following the advancement of navigation sensors, many authors combined measurements, as such, with mathematical models. Lohmiller et al [4] developed a velocity estimator for an underwater vehicle. The velocity estimator was based on an accurate non-linear model of the vehicle. Similarly, [5] used a detailed model of the vehicle to estimate its full state (i.e. position and velocity).

The use of mathematical models became more sophisticated, by taking into account the characteristics of the measurement noise. The Kalman Filter and its non-linear variant, namely the Extended Kalman Filter (EKF), is often found in modern navigation algorithms [6], [7], [8]. The aforementioned approaches employ a pure kinematic model for the process, together with all the information form the sensors. In this way, the Kalman Filter merely fuses the incoming sensor data sensor in an optimal fashion. More sophisticated algorithms leverage the dynamic equations of the vehicle as the process model [9].

Further to the use of dynamics within navigation, algorithms that adapt the dynamic models on-line gained popularity. Such algorithms tune the process and the measurement models in real time, as new data enter the system. Wang et al [10] describe a neural network aided Adaptive Kalman Filter (AKF). The filter attempts to match the theoretical with the actual covariances, as computed from the data. Song [11] formalized the tuning of covariances of an Uncented Kalman Filter as an optimization problem. The cost function of the optimization is the error between the innovation covariance and the predicted covariance.

Similarly to our work, Jwo in [12] used a neural network to compensate for unmodeled non-linearities in GPS navigation. Another example of the use of neural networks together with the system's dynamics comes from the diagnostics literature. Zhang et al [13] used neural networks to compute a corrective term for the dynamic model of an aircraft engine. Despite the capacity of neural networks in approximating non linear functions, the design decisions relative to the network's structure (e.g. number of hidden layers) influence its performance significantly. LWPR, on the other hand, achieves very good performance easily; without the need for elaborate tuning (see Section III-B). [14] used a Gaussian Process to approximate a non-linear corrective term for the dynamics of a blimp. The advantage of our method over that, originates from the adaptive traits of the model; LWPR provides a straightforward way of adaptation to changes in either the vehicle's dynamics or operating conditions.

III. ALGORITHM

The pose of the robot is predicted using the hydrodynamic model, together with a non-linear additive term (augmented hydrodynamic model). The hydrodynamic model was derived from first principles of physics. Given the forces applied by the thrusters, the first term in (1) estimates the vehicle's acceleration.The additive term is used to account for unmodelled non-linear phenomena, as well as to adapt the dynamic model in real time, if necessary. The corrective term was approximated using the LWPR algorithm. The navigation system is described by the following equations:

$$\begin{aligned}\dot{\mathbf{x}} &= f(x,u) \\ &= f_{hydrodynamics}(\mathbf{x},\mathbf{u}) + \hat{f}_{LWPR}(\mathbf{x},\mathbf{u})\end{aligned} \tag{1}$$

Where:

$$\begin{aligned}\mathbf{x} &= [\mathbf{x}_{translation}, \mathbf{x}_{rotation}] \\ \mathbf{x}_{translation} &= [\text{north, surge, east, sway, depth, depth\_rate}] \\ \mathbf{x}_{rotation} &= [\text{roll, roll\_rate, pitch, pitch\_rate, yaw, yaw\_rate}]\end{aligned}$$

$$\mathbf{u} = [F_1, F_2, F_3, F_4, F_5, F_6]$$

$$F_i = \text{Force from thruster } i$$

The first two thrusters accelerate the robot along the surge direction. Thrusters 3, 4 are mounted perpendicular to the robot's long axis and control the sway and the yaw of the robot. The last two are used to submerge/surface the robot.

A. Hydrodynamics

In this section, the derivation of the hydrodynamic model is described. Newton's second law of motion has been applied to yield the vehicle's dynamic model. The forces, which were considered for the model derivation, comprise the hydrodynamic drag, the forces from the thrusters, as well as the friction between the robot's skin and the water. The friction was modeled with a constant term, whose sign depends on the direction of motion, together with a first-order Coulomb friction model. In more detail:

$$\dot{\mathbf{x}} = \mathbf{M}^{-1} \cdot (\mathbf{u} - \mathbf{F}_{drag} - \mathbf{F}_{friction}) \tag{2}$$

Where:

$$\mathbf{M} = diag\{M, M, M, I_x, I_y, I_z\}$$
M : weight of the robot
I_x : moment of inertia along the roll axis
I_y : moment of inertia along the pitch axis
I_z : moment of inertia along the yaw axis

The moments of inertia I_x, I_y, I_z were computed analytically, assuming that the geometry of the robot is a perfect cylinder. The forces in the *i-th* direction are computed as follows:

$$F^{(i)}_{thrust} = gain_i \times (\%max\_thrust) \quad (3)$$
$$F^{(i)}_{drag} = 0.5 \cdot C_i \rho_{water} \cdot S_i \cdot |\dot{x}_i| \cdot \dot{x}_i \quad (4)$$
$$F^{(i)}_{friction} = F^{(i)}_{static} + F^{(i)}_{coulomb} = -k_{sv} \cdot \dot{x}_i - k_c \cdot sign(x_i) \quad (5)$$

TABLE I
HYDRODYNAMIC PARAMETERS

	C_i	k_s	k_c	S_i
surge	0.45	2.0	0.8	0.1414
sway	0.55	23	0.6	0.8105
heave	0.65	0.0	0.8	0.8105
roll	0.0	10.0	0.8	0.0
pitch	0.1	0.0	1.4	0.8105
yaw	0.1	0.0	1.4	0.8105

The values for the above parameters have been identified experimentally in a wave tank within Heriot-Watt University. During the identification process, a non-negligible delay of 1 sec was noted in the thrusters' step response. To account for this, the forces from the thrusters were passed from a first-order filter, with transfer function as shown in (6). The step response of filter is presented in Figure 1.

$$G(s) = \frac{3.25}{s + 3.25} \quad (6)$$

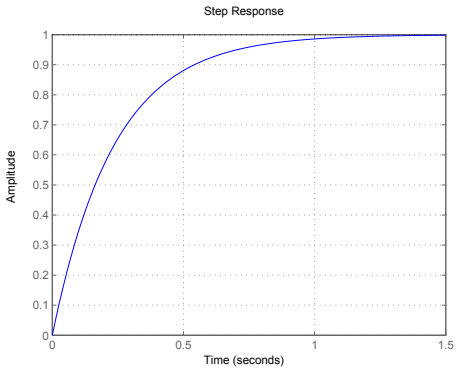

Fig. 1. Step response of the first-order filter that has been used to model the motors' delay

The acceleration in the body frame is integrated to yield the vehicle's velocity. Using the vehicle's kinematic equations, the velocities are transformed into the world coordinate frame. The latter are integrated further, to compute the displacement of the robot along the north and east directions, as well as the vehicle's depth and orientation (roll-pitch-yaw). To accomplish that, a 4-th order integration method from the Runge-Kutta family is employed. The main advantage of this method is the increased accuracy, even when large integration steps are used.

B. Linear Weighted Projection Regression

Locally Weighted Projection Regression (LWPR) [2] is a novel algorithm, which uses linear models to approximate locally, high dimensional non-linear functions. Similarly to the Partial Least Squares (PLS) [15], LWPR performs local dimensionality reduction, to identify the most important directions in the input space. This feature of LWPR makes it suitable for systems with redundant inputs. After the dimensionality reduction, the algorithm computes locally a hyperplane, to approximate the non-linear target function. The region, where each local model is activated, namely the *Receptive Field* of the local model, is defined using a kernel. Given a new query point, all the local models compute a prediction. The final prediction of the model is calculated as the weighted sum of all the predictions from the local models. The weight, which controls the contribution of each model in the final estimation, comes out from the kernel of the receptive field. The receptive fields described above adjust on-line, as more data become available. An extensive description of the algorithm is beyond the scope of this brief. The full details of the method can be found in [2]

1) Training: The LWPR method was used to learn a corrective term for the surge, the sway and the yaw dynamics. To this end, the robot was deployed in a wave tank, manually driven around using a joystick. While gathering the training data, the main objective was to cover the typical range of the robot's thrusters. Several parameters control the training phase of the algorithm. The most important are:

- *init_D*: initial size of a receptive field
- *update_D*: switches on/off the online adaptation of the receptive fields
- *init_alpha*: learning rate for the RF distance metric adaptation
- *penalty*: This parameter is used to prevent receptive fields from becoming indefinitely small

The parameter *init_D* is updated on-line, if *update_D* is set to true. Small values for *init_D* may fail to capture the non-linearities of the target function, whereas large values may lead to over-fitting. As mentioned in the algorithm's documentation [16], a typical tuning procedure commences with the estimation of *init_D*. In the first stage of tuning, the on-line adaptation of the receptive fields is switched off. Various models with different *init_D* values in the range $[1, 200]$ have been tested on a cross validation set. The model with the best performance on the validation tests was chosen for the second stage of the tuning process.

Following that, the incremental adaptation of the receptive fields was activated. One more grid search was performed for the *init_alpha* parameter. The final models for each direction are summarized in Table II.

TABLE II
PARAMETERS FOR THE LWPR ALGORITHM AFTER TUNING

DoF	init_D	init_alpha	penalty
surge	190	6	0.0001
sway	130	11	0.0001
yaw	32	22	0.0001

2) Validation of the model: To evaluate the generalization capacity of the LWPR, predictions were computed on cross-validations datasets. The ***normalized Mean Square Error*** was used as the metric for the comparison between models. The $nMSE$ is computed using (7), (8).

$$MSE = \sqrt{\frac{\sum_{i=1}^{n} (x_1^{(i)} - x_2^{(i)})^2}{n}} \tag{7}$$

$$nMSE = \frac{MSE}{x_{max} - x_{min}} \tag{8}$$

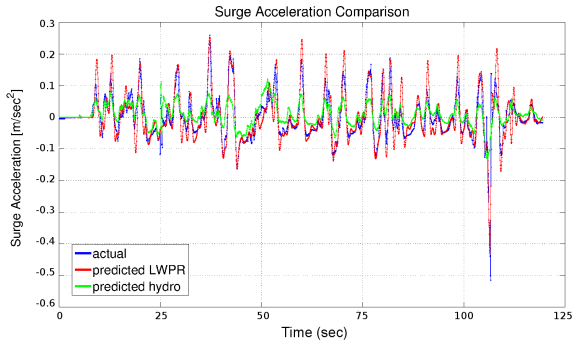

Fig. 2. Comparison between the actual surge velocity and the predictions from the hydrodynamics and from our method.

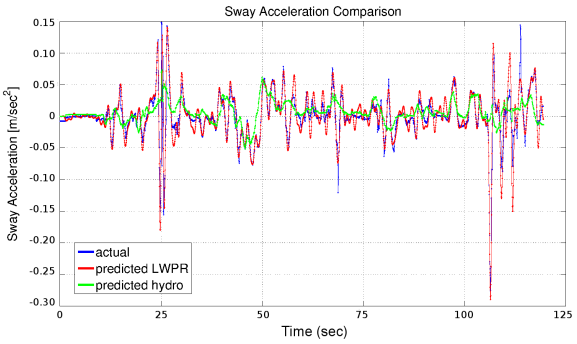

Fig. 3. Comparison between the actual sway velocity and the predictions from the hydrodynamics and from our method.

Figures 2, 3, 4 illustrate the predicted accelerations from the augmented hydrodynamical model, compared to the standard hydrodynamics along the surge, sway and yaw

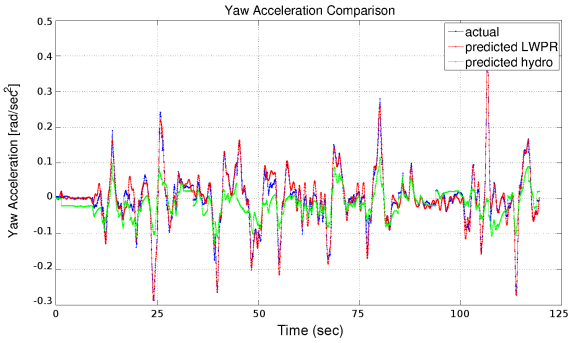

Fig. 4. Comparison between the actual yaw acceleration and the predictions from the hydrodynamics and from our method.

dimensions respectively. The augmented model outperforms the hydrodynamics in all dimensions. Table III summarizes the performance of the two models in terms of the normalized mean square error of the prediciton with respect to the ground truth, as measured by the sensors of the vehicle.

TABLE III
CROSS VALIDATION STATISTICS

DoF	Hydrodynamics	LWPR
surge	0.44	0.12
sway	0.95	0.19
yaw	0.59	0.05

C. Extended Kalman Filter

Kalman filtering [17] is a well established technique, which enables the fusion of noisy measurements. The measurements are combined with the aid of the noise statistics, as well as the system's dynamics. Briefly, the Kalman Filter operates in two steps: Firstly, the future state is computed (prediction step) by the system's dynamical model. Next, the estimation is corrected, leveraging the incoming sensor measurements. The accuracy of each sensor, as encoded in the respective measurement model, is exploited to weigh the sensor's influence on the final estimation of the state. Hence, the Kalman Filter fuses noisy measurements in a statistically optimal fashion. The standard Kalman Filter, however, only applies to systems with linear dynamics. To remedy that, the *Extended Kalman Filter* (EKF) has been developed.

The ***update*** step of the Kalman Filter commences with the computation of the ***innovation***. The innovation measures the difference between the predictions of the model and the observations (sensor measurements). Taking into account the sensor statistics, the Kalman gain is computed. The Kalman gain is, in fact, a weighting factor between the model's predictions and the measurements. Most often, the Kalman Filter utilizes an accurate dynamic model of the system, as the means to compensate for noisy sensor measurements. On the contrary, the proposed algorithm uses the innovation of the filter to improve the accuracy of the dynamic model (i.e. to learn the corrective term in (1)). At first, this may

Fig. 5. Nessie is the main research platform of the Oceans System Lab. It is a hover capable torpedo shaped AUV with a variety of sensors that are used for navigation (DVL,Gyro,Compass) as well as a Blueview forward looking sonar for perception.

seem contradictory to the typical usage of the filter; yet it is done intentionally. The main objective of this research is to construct a model, which resembles the behavior of the robot's sensors. As the difference between model prediction and observations, the Kalman Filter innovation is used to learn the corrective term.

IV. EXPERIMENTS

For the experiments, a torpedo shaped AUV has been used. Specifically, *Nessie* (Figure 5) is a research vehicle, developed within the Ocean Systems Lab. Table IV lists the navigation sensors which are available on Nessie.

TABLE IV
NESSIE'S NAVIGATION SENSORS

DVL	Teledyne Explorer PA
FOG	KVH DSP-3000
Compass	TCM 6
Depth Sensor	Keller Series 33X

In order to gather data from the navigation sensors, Nessie was driven manually on a closed-loop trajectory. The experiments took place in a wave tank within Heriot Watt University. The starting point was marked to make sure that the robot returned to its initial configuration. The measurement from the IMU, the DVL and the pressure sensure were recorded using ROS. Different trajectories were recorded for the training and the validation of the algorithm. The sensor measurements also served as the ground truth, against which the predictions of the augmented hydrodynamic model were compared. The estimation of the vehicle's velocity, along the surge dimension, is shown in Figure 6. In the same figure, the actual velocity of the vehicle is plotted, as measured by the DVL. Similarly, Figures 7, 8 present the results for the sway and yaw dimensions. The augmented hydrodynamic model reconstructs the velocity profiles correctly in all the directions of interest.

Next, the velocities were integrated, to yield the position of the robot. Figure 9 illustrates the trajectory, as estimated by

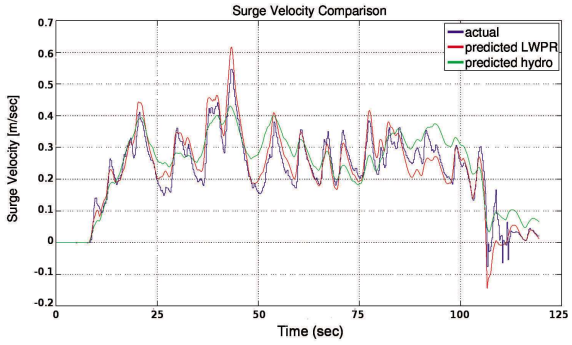

Fig. 6. Comparison of the estimated and the actual surge velocity

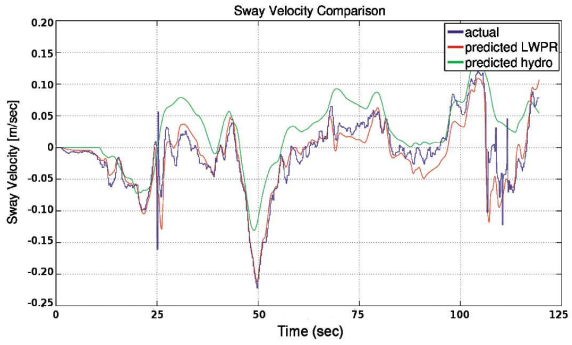

Fig. 7. Comparison of the estimated and the actual sway velocity

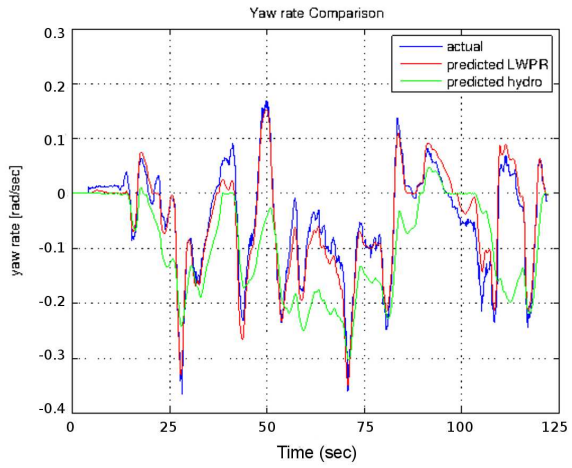

Fig. 8. Comparison of the estimated and the actual yaw rate

our method, compared against the estimations from the hydrodynamics. Additionally, the ground truth from the sensors is also plotted on the same graph. Despite the good match between the prediction and the acceleration measurements, the position estimation accumulates error as the experiment progresses. This is an effect of the two-step integration, which is required to compute the position of the robot from the acceleration.

Figure 10 provides a similar comparison for data, gathered during the SAUC_E-2013 underwater competition. This data were not used for the training of the augmented hydro-

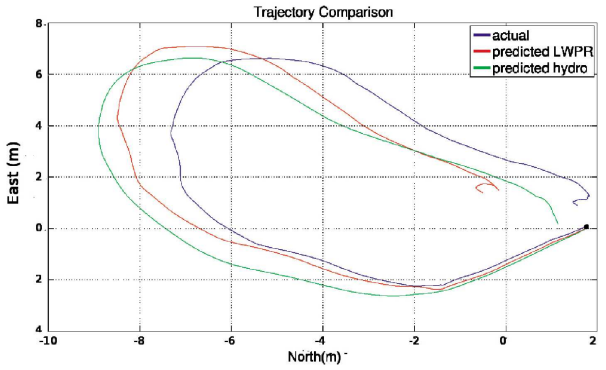

Fig. 9. In this figure the actual and the predicted trajectories are compared. Due to integration error accumulation we see that both the trajectory that was computed by the hydrodynamics as well as the one computed with our method suffer from non-negligible drift

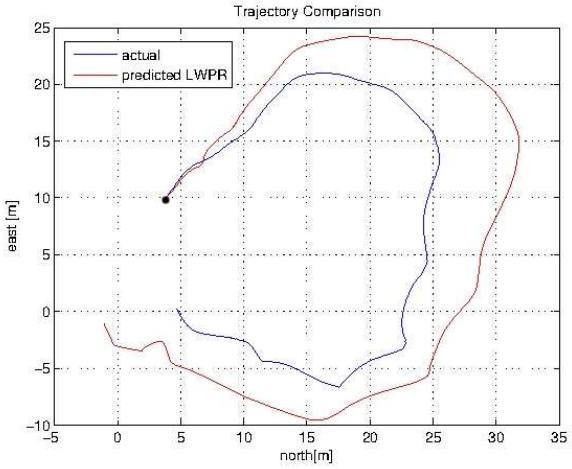

Fig. 10. Comparison of actual and predicted trajectory from data gathered during the SAUC_E-2013 underwater competition

dynamic model. Apart from the drift due to integration error, the predicted trajectory strongly resembles the actual trajectory of the vehicle. In this experiment the range of the model input was greater than during the experiments in the wave tank. Thus, the augmented hydrodynamic model doesn't fail in the case of unknown inputs. Nevertheless, the accuracy of the estimation deteriorated.

V. CONCLUSIONS

As shown previously, the proposed algorithm accurately estimates the vehicle's velocity. From our recent experience with Nessie, the operation of the DVL is often interrupted; especially when the pitch or roll of the robot is high or in the presence of strong acoustic reflections (e.g. close to the tank's wall). Similar behavior was observed for Nessie's one-axis gyro. As a remedy, the predicted values from the augmented hydrodynamic model may substitute corrupted measurements from a temporarily unavailable or failed sensor.

In the future, the adaptive traits of LWPR will be exploited, to compensate for the external disturbances (waves, currents). Moreover, the case of thruster failures will be investigated. All the above will be combined with a prognostic framework that estimates the health condition of the sensors and the actuators. When the prognostic system predicts a possible sensor or actuator failure, the respective model will be activated, to alleviate the impact on the navigation.

ACKNOWLEDGMENT

The authors would like to thank the Ocean Systems Lab staff for their input and support and in particular Dr. Tom Larkworthy for the valuable advice on the LWPR tuning and integration.

REFERENCES

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

[2] S. Vijayakumar, A. D'Souza, and S. Schaal, "Lwpr: A scalable method for incremental online learning in high dimensions," 2005.

[3] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *IFAC Conference of Manoeuvering and Control of Marine Craft*, 2006.

[4] W. Lohmiller and J.-J. Slotine, "Contraction analysis: A practical approach to nonlinear control applications," in *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1. IEEE, 1998, pp. 1–5.

[5] J. C. Kinsey, *Advances in precision navigation of oceanographic submersibles*, 2007, vol. 67, no. 11.

[6] M. Blain, S. Lemieux, and R. Houde, "Implementation of a rov navigation system using acoustic/doppler sensors and kalman filtering," in *OCEANS 2003. Proceedings*, vol. 3. IEEE, 2003, pp. 1255–1260.

[7] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 2417–2424.

[8] C. N. Roman, "Self consistent bathymetric mapping from robotic vehicles in the deep ocean," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[9] R. Van Der Merwe, "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, University of Stellenbosch, 2004.

[10] J. J. Wang, W. Ding, and J. Wang, "Improving adaptive kalman filter in GPSSDINS integration with neural network," *Proceedings of ION GNSS 2007*, 2007.

[11] Q. Song, "An adaptive ukf algorithm for the state parameter estimations of a mobile robot," *Acta Automatica Sinica*, vol. 34, no. 1, May 2008.

[12] D.-J. Jwo, C.-S. Chang, and C.-H. Lin, "Neural network aided adaptive kalman filtering for GPS applications," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, 2004, p. 36863691.

[13] X. Zhang, L. Tang, and J. Decastro, "Robust fault diagnosis of aircraft engines: A nonlinear adaptive estimation-based approach," *IEEE Transactions on Control Systems Technology*, 2012.

[14] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp," in *IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 2007, pp. 742–747.

[15] H. Wold, "Partial least squares," *Encyclopedia of statistical sciences*, 1985.

[16] "Lwpr software tutorial," [accessed 10-August-2013]. [Online]. Available: http://vcms.inf.ed.ac.uk/pab/slmc/research/software-lwpr

[17] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.