Modeling Motor-Perceptual Behaviors to Enable Intuitive Paths in an Aquatic Robot

Carlos E. Romero-Martínez, L. Abril Torres-Méndez
Robotics and Advanced Manufacturing Group
CINVESTAV Campus Saltillo
Ramos Arizpe, Coahuila, MEXICO
Emails: carlos.romero;abril.torres@cinvestav.edu.mx

Edgar A. Martínez-García
Lab. de Robótica, Inst. of Eng. and Tech.
Universidad Autónoma de Cd. Juárez
Juarez, Chihuahua, Mexico
Email: edmartin@uacj.mx

***Abstract*—This ongoing research focuses on providing skills based on motor-perceptual behaviors to an underwater vehicle in order to execute collision-avoidance trajectories in a more natural and intuitive way. An intuitive action can be seen as a reflex in humans and some animals. Reflexs do not involve a conscious reasoning at the time of execution, this is because a motor-perceptual skill for a particular action has been already developed in the brain from a past execution. For the case of a robot, the analogy would be that, during a collision-avoidance action, no sensor feedback is needed to correct the robot's state, and thus there is no need of applying a control law to achieve the required locomotion. The direct consequence of this would be the reduction of robot cost while maintaining its performance. Our approach involves a training phase, in which a set of primitive curve trajectories, at different radius and velocities, parameterized in time and orientation, are performed by using a PD control. By applying a linear regression classifier, only those output control parameters that yield stable and accurate trajectories are fed to a knowledge database. These parameters are used to perform intuitive trajectories without using a control law. Preliminary results show the feasibility of our method.**

I. Introduction

The exploration of underwater environments, and particularly the importance of monitoring the fragile life of coral reef ecosystems, lead this project to develop a behavior-based motor-perceptive platform to assist an Autonomous Underwater Vehicle (AUV) during the collection of data. Coral reefs contains 25% of the sea life, presenting a great opportunity to develop robotic platforms that can function as tools in the research and monitoring activities of scientists. An specific task of the AUV is to avoid collision during the navigation task. To achieve this goal is important to consider the non linear dynamic effects that the environment presents, the uncertainties of the model and the absence of a position measurement.

Robust control techniques are usually applied to solve most of these problems, *e.g.*, sliding mode controls have been used on simulations to control maneuvering and tracking of an AUV [1], [2], although it is recommended to use a separation of variables such as linear speed, steering and diving control; it is also important to consider that the control needs the information of the state of the robot, meaning the position and orientation. Combinations of this control law with adaptive models have also been simulated [3] to control the vehicle on a dive plane, and when the state is not available, an observer of the robot model is used and the uncertainties are compensated by the sliding mode control, the results showed stabilization of the system working at slow rates of speed, but for the purpose of executing fast actions this may not be the best option.

Limitations on current technology make the accurate estimation of AUVs position during navigation difficult. However, this is also true for human divers, who use visual information to guide their navigation towards the site of interest and make the required maneuvers to avoid collisions. This can be seen as a perceptual-motor action, which depending on the situation may correspond to an intuitive behavior. Whereas the guiding towards a region of interest is a task that needs continuous sensor feedback to correct the robot's state by using a control's law, the collision avoidance task in a high dynamic environment indicates that a control feedback needs to be overlooked to provide a robot with *agile* human-like behaviors during navigation. In order to do this, the robot needs to previously learn the motor-perceptive movements that will lead the right execution of the avoidance trajectories. This involves knowledge of space, time and sensorial information.

Thus, our research efforts are focused on enabling an AUV to perform intuitive navigation behaviors when a fast but effective collision avoidance action is required. In previous work, we have used visual attention algorithms to guide the navigation of an AUV, similarly as a human would do it and using evasion strategies to avoid obstacles [4]. In the present work, we have now added intuitive behaviors to avoid collisions with the fragile coral reef habitat by providing to the AUV the ability to perform previous learned trajectories without the need of a control law. In our approach, the process of learning motor-perceptual skills involves two phases: field training and classification. First, in the training phase, by using a control law, the robot executes several times a set of trajectories with different radius and velocity in a real underwater environment. Also, in this phase, latency variables are adjusted according to what is expected. Second, in the classification phase, the output of each computed control is then stored and classified by using a nonlinear regression model so that the best control outputs are stored in the knowledge base. The data in the database is used to reproduce similar actions performed in the past.

The rest of this paper is organized as follows. First, Section II describes our proposed method in detail. In Section III, we give the experimental results obtained during our field trials. Finally the conclusion and future work is presented in Section IV.

978-0-933957-43-5 ©2015 MTS

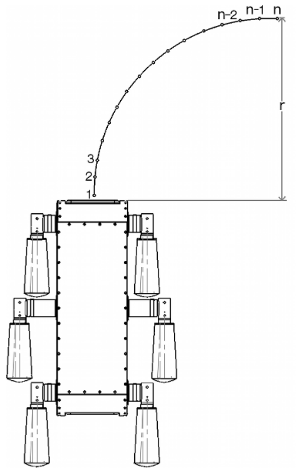

Fig. 1. A parameterized curve trajectory.

II. METHODOLOGY

Our methodology starts with the training phase, in which the robot performs a set of trajectories (curves of different radiuses r and velocities v) by using a controller. To achieve this, it is necessary to parameterize the curves (see Fig. 1) in terms of the variables that the robot is capable to measure with some degree of accuracy. These parameters are: *time* and *orientation*. By using the desired orientation ψ and assuming an environment with minimum of perturbations (*i.e.*, without strong currents), the program computes a linear speed v_x for each elapsed time, as follows:

$$\psi = -tan^{-1}\left[\frac{r * sin(ta)}{r * cos(ta)}\right]. \tag{1}$$

$$v_x = \frac{r \cdot \psi}{T}. \tag{2}$$

The robot performs the trajectories using a *PD* control, defined as:

$$\psi_d^R = K_p\left(\psi_d - \psi_c\right) + K_d\left(\dot{\psi}_d - \dot{\psi}_c\right), \tag{3}$$

where K_p and K_d are the proportional and derivative control gains.

Only the output control parameters that yield stable and accurate trajectories are fed to a knowledge database. A set of programming procedures (called nodes) were implemented for that purpose. The first node *parameterizes* the trajectory in terms of orientation and time. A second node *publishes* the desired orientation in a correct frequency, this node needs the information of the linear speed (Eq. 2), and the desired orientation (Eq. 1) and it also computes the estimated time to publish the next desired step of the trajectory. Another node *reads* the control outputs and stores in a *.txt* file every calculation of the control law and the computer time when this was published.

Next, the training data is analyzed by a linear regression classifier (LRC) to determine if the dataset (control output vector) belongs to a group or not. This step will define the set of curves trained to perform a particular curve of the same radius and speed. The classifier achieves this by making a decision based on the value of a linear combination of curve features, known as *feature values*, and the vector that contains the control output is called *feature vector*. The procedure that we are implementing is shown below:

$$Y_c = X \cdot X^T \cdot Y, \tag{4}$$

$$\epsilon = \mathcal{L}_1(Y_c - Y), \tag{5}$$

where $X \in \Re^{[m\times n]}$ is a matrix that contains the control outputs, also called knowledge base matrix, $Y \in \Re^{[m\times 1]}$ is the new vector to be classified, by projecting it onto matrix X; m is the number of feature vectors, n is the number of vectors in the knowledge base; ϵ is the norm $\mathcal{L}_1$ of the error between the projected vector Y_c and the features vector Y.

In order to achieve an effective intuitive behavior in the robot, *i.e.*, the execution of a previous learned and classified trajectory without using a controller while setting directly the desired velocity, it is necessary to obtain at each elapsed time, the desired orientation in the right frequency. Therefore, the linear speed, the desired orientation as well as the average latency times need to be taken into account. An *execution* node, that publishes control commands without processing any additional input from the robot's sensors is going to perform the task of a motor-perceptual behavior. The clear advantage of this is in the reduction of the execution time as the robot only receives the set of forces that need to apply on its motors to perform the trajectory.

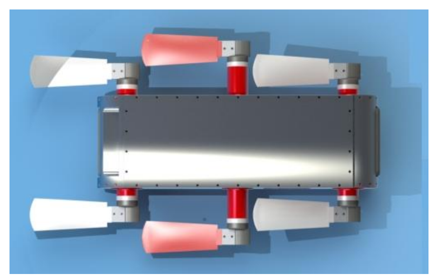

Fig. 2. Our robot platform, MexiBot.

The experimental protocol has two stages, the training and the execution of learned trajectories. Both were performed in pool and open sea water to validate the results.

Before presenting the procedures and considerations that we followed, we first describe the robotic platform we use.

A. Platform and technical information

The platform used for this project is a six paddle robot (Fig. II), from the AQUA family called MexiBot. Its architecture and sensors include an IMU, pressure sensor, three cameras, six paddles and two computers; one computer for the vision programs and one for the motors control. The control computer has its own operating system RHex developed specifically for this integration [5]. It contains the configuration of the actuators and the dynamic model of the robot. The vision computer runs under ROS and it is mainly used to program the vision algorithms and make an interface using the visual information to send commands to achieve goals by the orientation control.

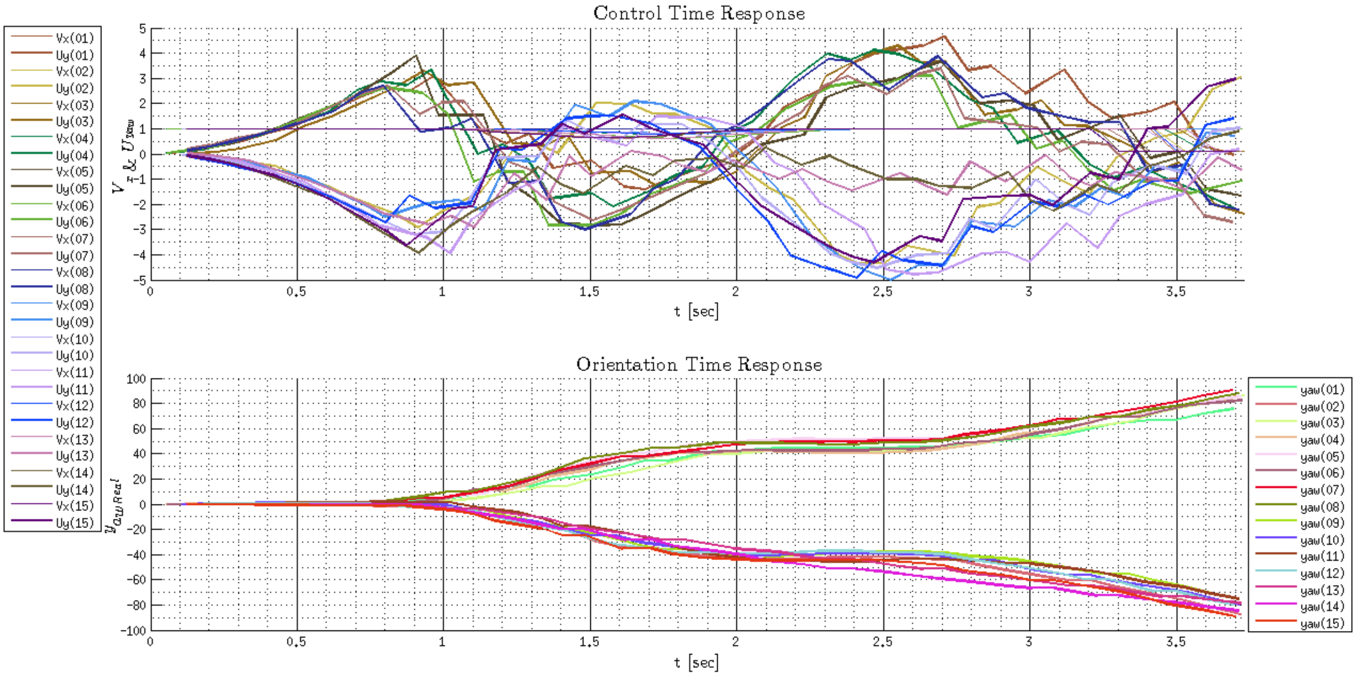

Fig. 3. Training curves or radius 0.25m.

III. Experimental protocol

In the training phase, suitable adjustment of the control gains was crucial to have good performance on the execution of the trajectories. Each trajectory, with its respective parameters of linear speed, radius and orientation, was executed several times in order to assure that the execution has minimal variations. The control values of each type of curve was stored. Two aspects need to be considered : 1) the rising time of the robots linear speed; this time exists due to the friction added by the fluid (water in this case) in which the robot is moving; and 2) the maneuverability of the AUV, the linear speed was gradually reduced and then increased in the middle section of the trajectory, this will allow the robot to perform a more stable curve trajectory (without chattering) and thus to arrive softly to the final desired orientation.

In the second phase, the control publisher node is going to perform the control actions based on the stored control commands, the result expected at this stage of the research is to have a similar execution when using a control law. The same control publisher node is going to perform the same trajectories, but based on a regression model obtained from the repetitions of each trajectory. Based on the regression models, a learning algorithm is going to help us to predict the new control commands to perform trajectories at radiuses and linear velocities, that have not been trained before. The only aspect we considered in the second phase was the correct monitoring of the computer time, if this time is well synchronized, then we can assure an execution of the trajectory with minimal variations.

IV. Field experimental results

A. Training group: 0.25 meter curves

The first type of curve is a quarter of circle shape with a radius of 0.25 m, the training conditions for the first set of vectors are: a terminal surge speed of 1.0 m/s with a slow down of 0.8 m/s; the gain values for the control are: $k_p = 3$, $k_d = 5$; the number of points for the parametrization is 44.

The set of curves used to build this knowledge base also contains the orientation and control error, which was obtained by removing each one of the vectors from the matrix of knowledge and then projecting it to X. Each error vector was normalized in order to perform a better analysis of the data.

The *mean* value obtained from the previous error vector was used as a threshold to determine the curves belong to the same group and discard the others. The mean value of ϵ_o was 0.4741, after normalizing the orientation vector, which was the vector used to perform the classification. Figure 3 shows the plot of the selected curves.

The average time to perform each of these curves using the control law was: 3.2155 sec. The average time to perform each of these curves without using a control law, was: 3.3399 sec. The average time to perform an 'U' turn by concatenating two of these curve segments was: 6.6209 sec.

B. Training group: 0.15 meter curves

The second type of curve is a quarter of a circle with radius of 0.15 m. We use the same training parameters as in the previous training curves.

The mean value of ϵ_o was 0.2672, after normalizing the orientation vector. Figure 5 shows the plot of the selected curves. The average time to perform each one of those curves

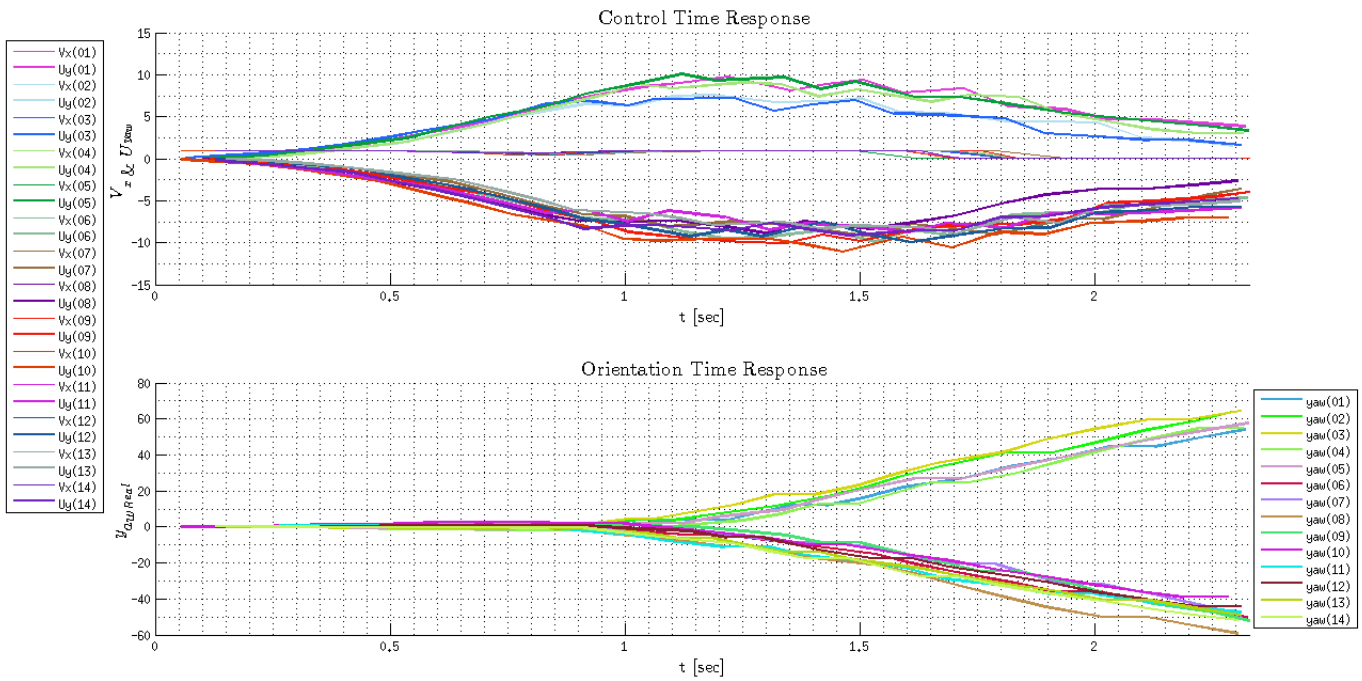

Fig. 4. Training curves of radius 0.15m.

using the control law, was: 1.5065 sec. The average time to perform a 'U' turn by concatenating two of these curve segments was: 3.2803 sec.

Field experimental results with and without using the control law are depicted in Fig. 5. The trajectory is a concatenated curve segments of radius $0.15m$ with sequence left-right-left-right. Figure 5a presents the control outputs when applying the PD control with variable surge speed at the middle. Figure 5b and c show the results with and without the controller, respectively. In both cases, it is plotted the orientation measured by the IMU. The results for this research consider a good reproduction of the trajectories without using the control law.

V. Conclusion and Future Work

We have presented an approach to learn motor-perceptual behaviors in trajectories executed by an AUV when using a controller, in order to be executed later without any kind of sensor feedback. Preliminary results of tests, carried out in pool and sea water, have shown that it is possible to obtain a good reproduction of the trajectories without using a control law, thus reducing the computation cost and therefore the processing time. The particular application on which this kind of behaviors, in the execution of trajectories, is needed is in collision-avoidance actions. In particular, for agile evasion actions, smaller radiuses at high speeds are desirable. The next step of this research is to apply a statistical learning method to obtain the control outputs for those input parameters (radius and velocity) of trajectories that are not in the knowledge base.

Acknowledgment

We would like to thank CONACyT for their support and project funding (CB-220540). We also thank Mar Adentro Diving, Mahahual, for their support during our sea trials.

References

[1] M. U. Cildoz, C. H. F. Dos Santos, and R. Reginatto, "Comparative study of chattering-free sliding mode controllers applied to an autonomous underwater vehicle (auv)," in *CONTROLO2014–Proc. of the 11th Portuguese Conference on Automatic Control*. Springer, 2014, pp. 587–596.

[2] A. J. Healey and D. Lienard, "Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles," *Oceanic Engineering, IEEE Journal of*, vol. 18, no. 3, pp. 327–339, 1993.

[3] R. Cristi, F. A. Papoulias, and A. J. Healey, "Adaptive Sliding Mode Control of Autonomous Underwater Vehicles in the Dive Plane," *15th Journal of Oceanic Engineering*, vol. IEEE, no. 3, 1990.

[4] A. A. Maldonado-Ramírez, L. A. Torres-Méndez, and F. Rodríguez-Télles, "Ethologically inspired reactive exploration of coral reefs with collision avoidance: Bridging the gap between human and robot spatial understanding of unstructured environments," in *Intelligent Robots and Systems, Proceedings of 2015 IEEE/RSJ International Conference on*. IEEE, 2015.

[5] C. Georgiades, "Simulation and Control of an Underwater Hexapod Robot," Department of Mechanical Engineering, McGill University, Montreal, February 2005.

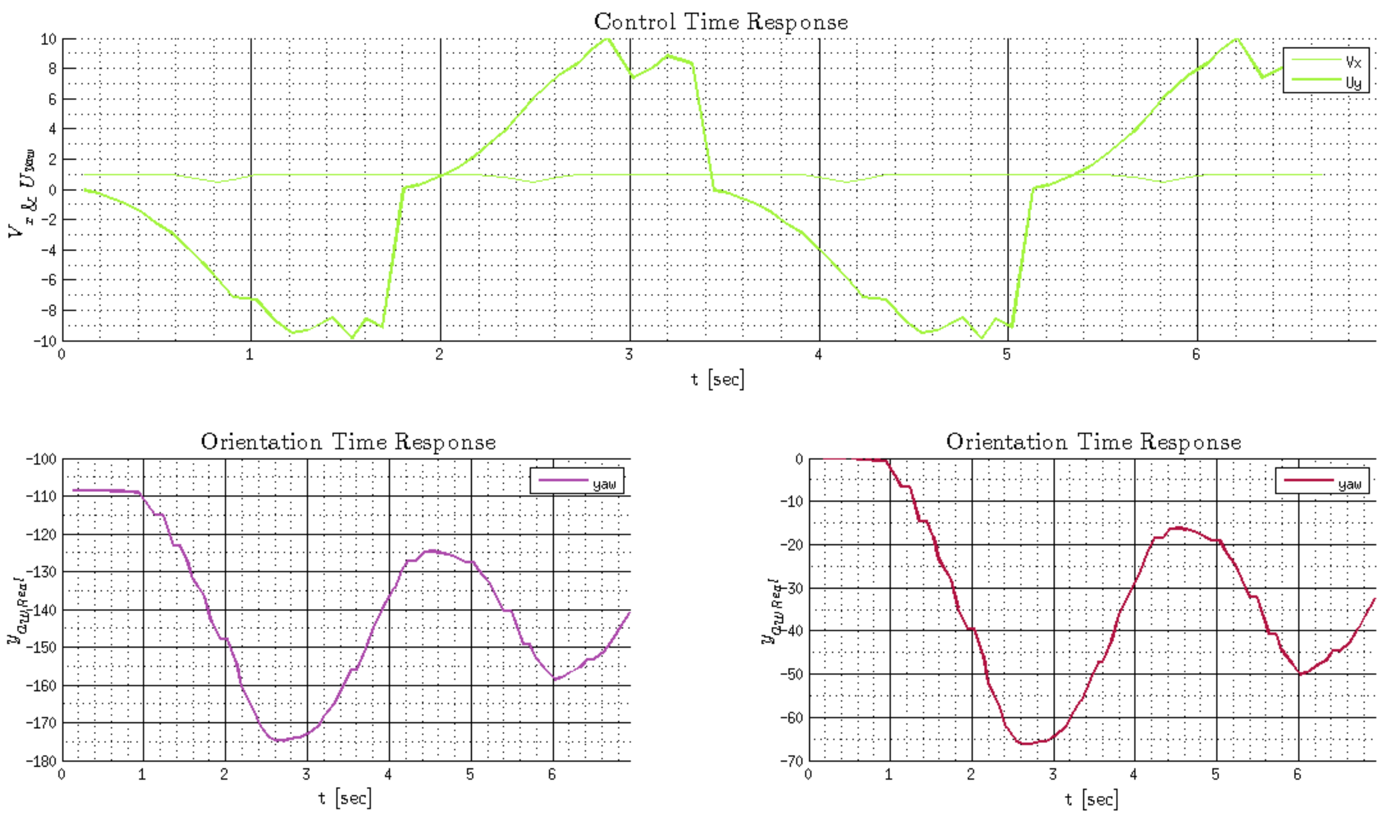

Fig. 5. Results on applying our approach *with* and *without* using a control law in a concatenated sequence of left-right-left-right curve segments of radius 0.15m.