5th Iranian Conference on Fuzzy Systems
Sept. 7-9, 2004, Tehran

Developing a Tracking Algorithm for Underwater ROV Using Fuzzy Logic Controller

***M.H. Saghafi*[1], *H. Kashani*[2], *N. Mozayani*[3], *G. R. Vossoughi*[4]**
mh_saghafi@yahoo.com

ABSTRACT
One of the major aspects of Underwater ROVs control is tracking a well-defined path with predetermined speed. In this study a fuzzy decision making algorithm has been developed based on the distance of ROV from an ideal path and the angle error. Its outputs are the desired speed and the angle in local coordinate; used in close loop control system. Implementing rules, ROV reaches to the predetermined path with the desired speed in an almost optimum manner. Finally, abilities and robustness of the algorithm is shown by testing it to reach a line and a circle from outside and inside. Also the tracking of a spline in low and high velocity is successfully done.

Keywords: Fuzzy decision making, Path Tracking, ROV, Underwater

1. Introduction

With the advent of autonomous underwater vehicles (AUVs) and remotely operated vehicles (ROVs) their application has vastly developed in hardware zone underwater area. Also these kinds of vehicles are potentially very appropriate for carrying various types of instruments like manipulators, vision instruments, diverse sensors, etc [1].
Due to nonlinearities and the existence of uncertainties in dynamic of ROV's, they have become common useful tools for testing versatile and sophisticated control laws [2]. Generally, the final goal is to control the system in a special environment but in this paper we intend not only to control the system but also to follow a predetermined path. Therefore the control of ROVs has tow main aspects: the first one is the close loop control of ROV which is a law between output feedback and desired values of control system and the second is producing congenial values of close loop system to follow a well-defined path.

In this study we are seeking an algorithm for reaching to a specified path as best as possible. This means that applying this algorithm, in each initial state, the ROV will reach the desirable path in an optimized manner and starts following the path with predetermined speed. To achieve this goal we might use a classical solution in crisp mathematics to find an ideal trajectory using initial and final states of the system. Such a solution can be very complicated

1- M.S. in applied mechanics, Mechanical Engineering, Sharif University of Technology
2- M.S. Student in Artificial Intelligence, Computer Engineering department, Iran University of Science and Technology, hamedkashani@yahoo.com
3- Assistant professor, Computer Engineering department , Iran University of Science and Technology, Mozayani@iust.ac.ir
4- Associate professor, Center of Excellence in Design of Robotics and Automation, Mechanical Eng. department , Sharif University of Technology

[3]. However the target optimization function could be defined in a lot of different functions of time and location terms. According to these different target functions we may reach different solutions. In such cases a fuzzy decision making could be a powerful and proper alternative of crisp solutions [4]. The fuzzy system will be implemented according to a set of rules based on experiences and dynamical equations. Fuzzy algorithm will find desired values of ROVs' parameters according to the path specifications and dynamics of problem. These desired values are inputs to the close loop control system and finally results in reaching to the path and following it with desired speed.
Our proposed method has to be simulated on an underwater ROV, which is designed and manufactured at Sharif University of Technology, called SROV (fig 1).

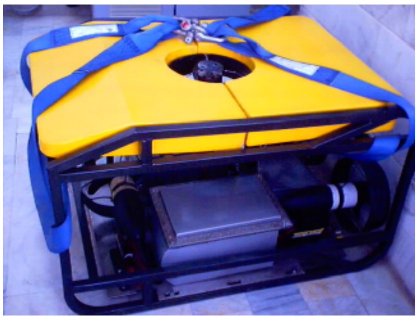

Figure 1: A vision of SROV

The SROV has passed its open-loop tests and is preparing to implement a control algorithm on it. Because of actuators of SROV (fig 2), it has four control direction (sway, surge, heave and yaw) of six possible direction of any rigid body and the remainders are passively controlled. We also use a back stepping controller for close-loop control law [2].

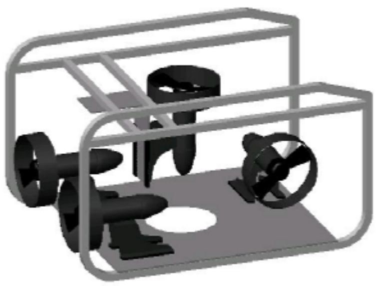

Figure 2: Combination of thrusters mounted on the ROV

In this case the user of ROV can define velocities of vehicle in x and y direction, and yaw angle which is its angle in x-y plane. Also user defines work depth of ROV. In next sections we explain the definition of problem. After that, the conceptual design of fuzzy path following algorithm and the implementation of fuzzy solution to the problem is introduced. Finally simulation results are presented.

2. Definition of path following problem

As mentioned previously, the user determines the specific path to be followed by desired speed. In practical applications instead of defining a complete path, the user inputs some points through which the ROV should pass. These points have a depth from the surface of water which is controlled by a separate algorithm. Meanwhile angles in x-z and y-z planes are passively controlled. Thus, we determine an ideal path in x-y plane by splines of 3 degree which passes through all predetermined points. Our final target is to reach the path as soon as possible and then to follow it by a desired speed.

Now we consider the problem of how to reach to a path from outside. We begin with an initial speed an orientation for ROV and would like to obtain their desired values toward the path. For this purpose we should determine exact inputs and outputs of the algorithm. It is to be noted that all input output parameters should be fuzzy terms in order to be used in our fuzzy decision making system. The input to the algorithm is the present distance of ROV to ideal path .It is the distance of ROV to nearest point of ideal path which could be far, medium, near or very near (fig3). Very near zone is the area in which we want to stable the place of ROV on ideal path.

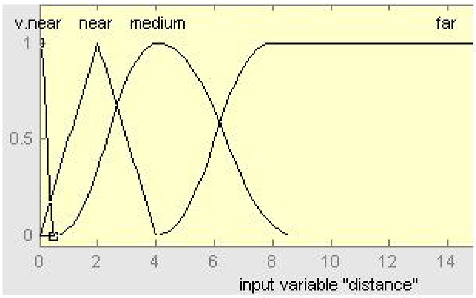

Fig 3: Fuzzification of distant

Another input to the algorithm is the angle error between the present and the desired angle. The angle is measured between tangential angle to the desired path at the nearest point and the direction of ROV. This angle can vary from -180 to +180 (fig 4).

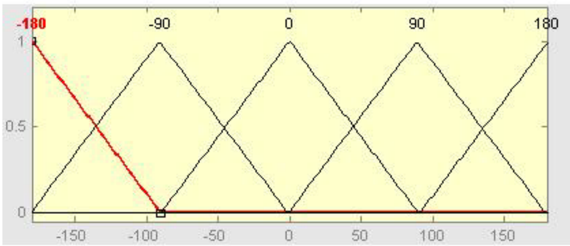

Fig4: fuzzification of ROV angle

Outputs of the system are desired speeds (V_X, V_Y) and the angle in local coordinate. These outputs are inputs to the control equation which is mentioned in [2] as a back stepping control method.

Due to the saturation of ROV actuators, there are limitations on values of these speeds. V_X Can be defined from -0.3 to +0.5 m/s. We express this value with five fuzzy terms. One of these terms is called *desired* speed that user determines (fig 5).

Because of lower significance of ROV's speed in Y direction (V_Y), it has a minor role for our control system. The range of this parameter is from -0.1 to 0.2 m/s. The final value of this speed is zero. The value of V_Y is divided to 3 fuzzy terms, consisting -fast, slow, +fast (fig5).

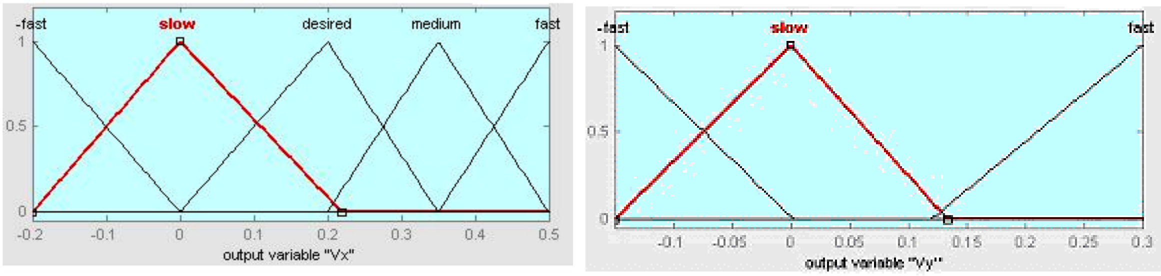

Fig 5: Fuzzification of output desired values of V_X, V_Y

The output angle can vary from -90 to 90 degree, otherwise ROV will move in opposite direction (fig6).

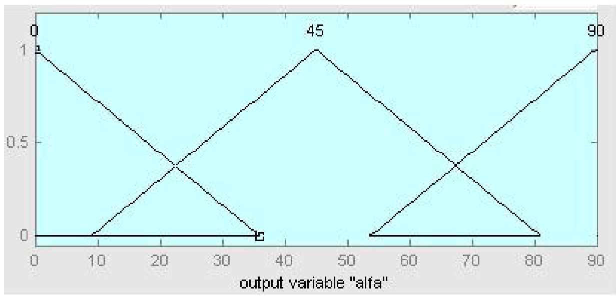

Fig 6: Angle output of fuzzy algorithm

2.1 Basis of fuzzy rules

Having simply defined inputs and outputs of the fuzzy algorithm, we are seeking now for some rules to implement the fuzzy decisions. To reach to a solution, suppose the ROV has to reach to a straight path with an identified velocity. What decision should we make for it? If the ROV goes to the nearest point of the track, which is perpendicular to the path at that point, it is necessary for ROV to halt on the track. After that it must change its direction in such a way to be tangential to the path, then speed up to the desired value of velocity. Obviously, it is a simple possible solution but not the optimum one [5]. Better say, this approach will waste energy and time.

It would be better that ROV change its direction from perpendicular to tangential angle to the ideal path before settling on it. Therefore we are apt to reach to the path by perpendicular way as fast as possible to nearest point of the path when the ROV is far away. In that situation, once the ROV comes near to the path, it is preferred to incline ROV to track's tangential angle at the nearest point. Meanwhile we try to bring ROV's velocity gradually near to its desired value. Also it should be mentioned that nearest point of the track to ROV updates at each moment due to characteristic of track and ROV's position (fig 7).

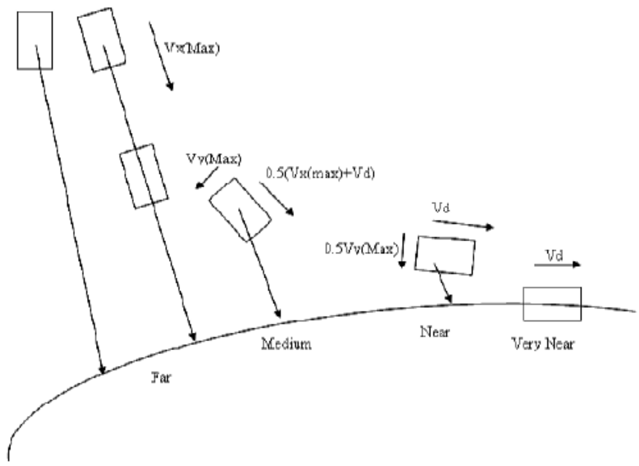

Fig 7: Schematic of ROV different states

By recourse to foretell logic, we set about the development of rules according to ROV's distance from nearest point and its angle. Furthermore, we have to take into account other parameters like the velocity in Y direction to improve the approach to the path.

2.2 Fuzzy rules

To define rules, we start from a situation that ROV is far from the path. Analysis of this situation falls into 5 angle states, including -180,-90, 0, 90, 180. This angle is the difference between ROV's direction and the tangential angle to the path at nearest point. In these states the dominant law for -180 and 180 are the same. Suppose that ROV is above of the path, it means that:

$$y_{ROV} - f(x_{ROV}) > 0 \qquad (1)$$

In which $f(x) = 0$ is equilibrium of desired path and (x_{ROV}, y_{ROV}) are ROV's coordinate in XY plane. In this case we want to get nearer to the path by -90°. One approach is that, if the ROV's angle is not -90°, all of linear velocities should be zero until reaching to -90° and then it should move by maximum velocity. But a better solution is setting velocities in such a way that they help to draw ROV near to the path (table 1).

Table 1: Fuzzy logic in far case

IF ROV is	And its angle is	Then	V_X desired is	V_y desired is	Desired angle is
Far	-180	$\Rightarrow$	Slow	fast	-90
Far	-90	$\Rightarrow$	Fast	slow	-90
Far	0	$\Rightarrow$	Slow	-fast	-90
Far	90	$\Rightarrow$	-fast	slow	-90
Far	180	$\Rightarrow$	Slow	fast	-90

It should be mentioned that if ROV is under the path:

$$y_{ROV} - f(x_{ROV}) < 0 \qquad (2)$$

We only need negate the input and desired output angle of fuzzy logic controller. Also if desired velocity is negative, the output angle of fuzzy logic will be converted to its complementary. These propositions hold true for other situations of farness. Therefore, from now on we only discuss about cases that ROV is above the track and that the desired value of velocity is positive.

Now, when it is in medium distant from the path it will try to come to a velocity between maximum and the desired value. In this case, the desired output angle will be -45 (having a linear velocity, ROV will approach better and nearer to the path) (table 2).

Table 2: Fuzzy logic in medium case

IF ROV is	And its angle is	Then	V_X desired is	V_y desired is	Desired angle is
Medium	-180	⇒	Slow	Fast	-45
Medium	-90	⇒	Medium	Slow	-45
Medium	0	⇒	Medium	Slow	-45
Medium	90	⇒	-fast	Slow	-45
Medium	180	⇒	Slow	Fast	-45

When the ROV comes near to the path its desired value of angle is zero and only the velocity in Y direction helps it to lie in the path. V_X gets also its desired value (table 3).

Table 3: Fuzzy logic in far case

IF ROV is	And its angle is	Then	V_X desired is	V_y desired is	Desired angle is
Near	-180	⇒	Slow	Medium	0
Near	-90	⇒	Slow	Slow	0
Near	0	⇒	Desired	-fast	0
Near	90	⇒	Slow	Slow	0
Near	180	⇒	Slow	medium	0

In the final case, when it is very near to the path, the value of desired angle of ROV is zero and the velocity reaches to its desired value. The only difference with the previous case lies in steady situation, which V_Y set to zero. The submitted rules when ROV is very near to the path are described in table 4.

Table 4: Fuzzy logic in very near case

IF ROV is	And its angle is	Then	V_X desired is	V_y desired is	Desired angle is
Very near	-180	⇒	Slow	slow	0
Very near	-90	⇒	Slow	slow	0
Very near	0	⇒	Desired	slow	0
Very near	90	⇒	Slow	slow	0
Very near	180	⇒	Slow	slow	0

3. Simulation results

For testing the proposed fuzzy method for generating desired values of control system for our path follower, a simulation by Matlab6.5/Simulink [6] software has been developed (fig 8). In this simulation the ROV is considered as the follower for which characteristics and dynamical equations are described in [7] and its multipliers are obtained in [8].

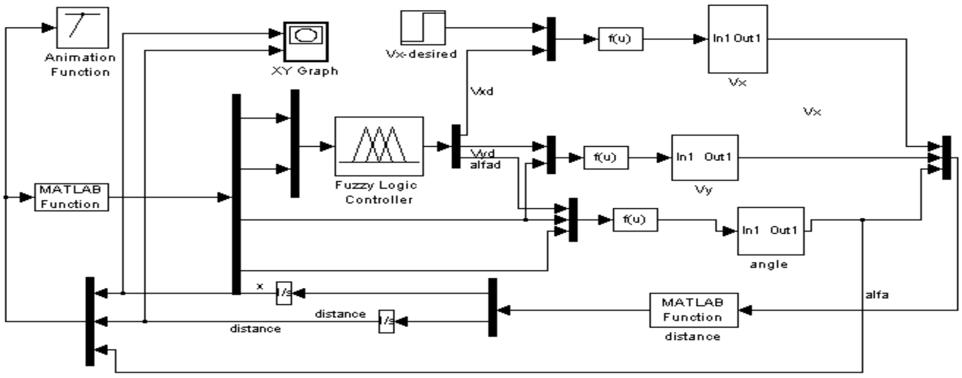

Fig8: ROV path following simulation, created by Matlab6.5/Simulink software

In order to verify the performance of our fuzzy path following algorithm, we firstly tested it for following a line with the velocity of 0.4 m/s (fig 9). We obtained promising results; the ROV approached rapidly toward the path. More ROV got near, better it inclines its angle and velocity to its final values (the sample time of vehicle is one second).

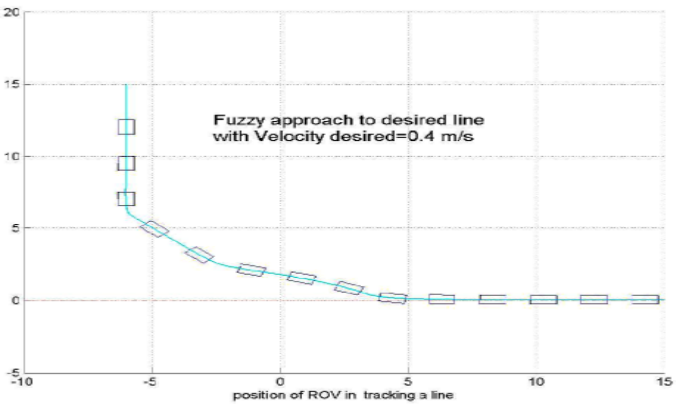

Fig 9: tracking a line by 0.4 m/s as desired velocity

In the next step the simulation has been done for tracking a circle from outside and inside (Fig 10, 11). These experimentations show the sensitivity of algorithm to the convexity of track which will be studied in future works. In this test we have supposed that the track contains no sharp convexity. We have also obtained good results for this step (Fig 10, 11).

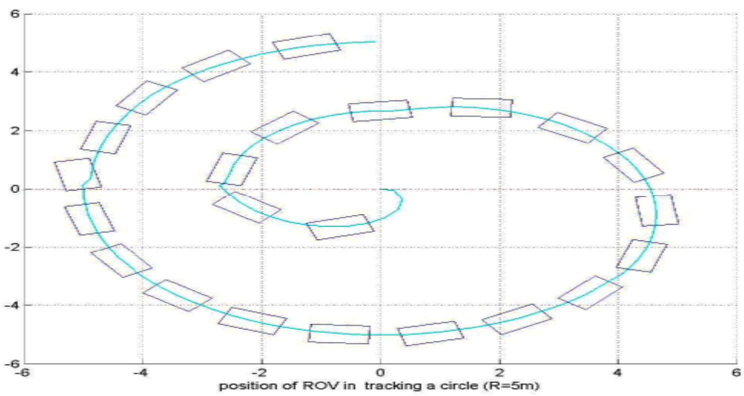

Fig 10: Tracking a line by 0.4 m/s as desired velocity from inside

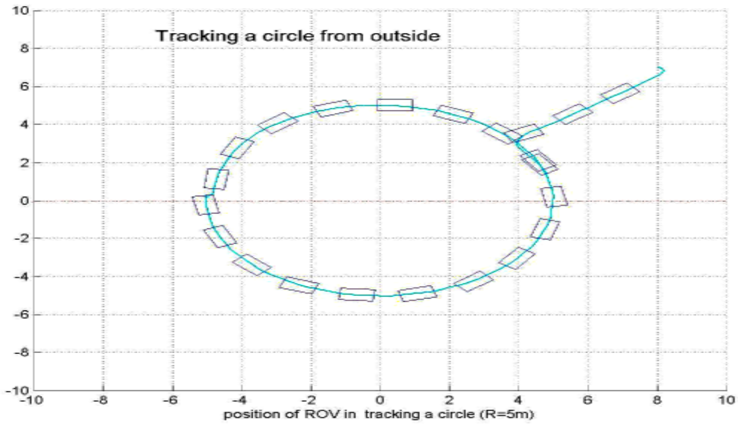

Fig 11: Tracking a line by 0.4 m/s as desired velocity from outside

At a final experience, we turned to tracking a spline path. The path was defined as a third degree spline curve (interpolated to the points). We tested the algorithm with two different velocities, a low value (0.2m/s) and a higher one (0.4m/s). In this experimentation, the ROV approached reasonably to the path with a well-shaped trajectory (Fig 12, 13).

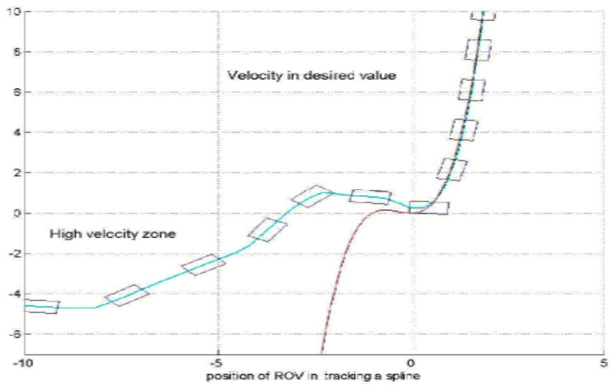

Fig 12: Tracking of a spline with 0.4 m/s

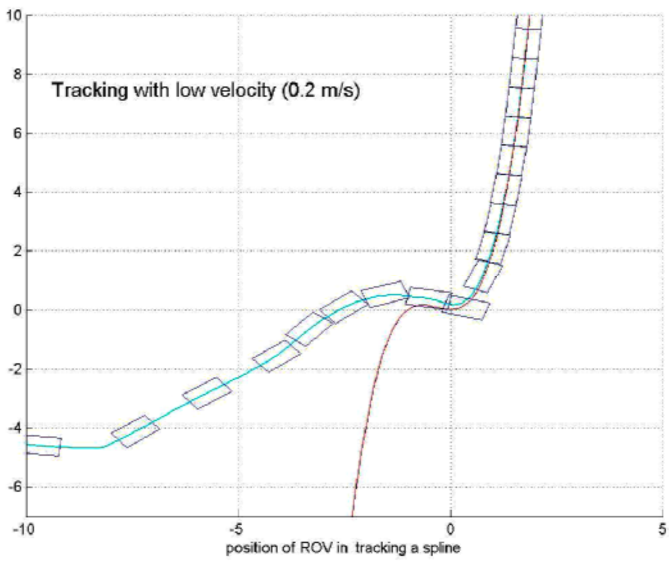

Fig 13: Tracking of a spline with 0.2 m/s

4. Conclusion

As it has been shown in simulations, by applying this fuzzy decision making algorithm, our ROV becomes able to follow several kinds of paths (such as linear, circular and a more general one) with positive and negative convexities. We have also observed that the tracking is done in a smooth manner with no abrupt changes in direction and speed. In this system, fuzzy rules have been designed according to the dynamic laws and realistic findings and experimentations. Consequently, we have minimized the time to reach the expected path. It is finally shown that the algorithm is capable of accepting isolated points as input data. The set of discrete points are covered by a spline curve and the ROV successfully approaches to the curve with different values of velocity. We believe that this algorithm can be used for variety of real applications, especially military underwater ones.

References

[1] Schjolberg, I. and T. I. Fossen, (1994), "Modeling and Control of Underwater Vehicle-Manipulator", 3, Marine Craft Maneuvering Control, pp.45-57, Southampton.

[2] M. H. Saghafi, (2004), "Control of an Underwater ROV, equipped with a Robotic Manipulator, using Adaptive Back Stepping", M.S. Thesis, Sharif University of Technology.

[3] R. Murray and S. Sastry, (1993), "Nonholonomic motion planning: steering using sinusoids", IEEE Transactions on Automatic Control, 38 (5), pp. 700–716.

[4] H.J. Zimmerman, (1991), "Fuzzy Set Theory and its Application", Kluwer Academic publishers, Boston.

[5] R. Braunstingl, (1995), "A wall following Robot with a fuzzy logic controller optimized by a genetic algorithm", IEEE /IFES '95 fuzzy robot competition, Yokohama, P.77-8.

[6] Zadeh, (2001), "Fuzzy logic toolbox, for use with Matlab", Math Works Inc.

[7] G. R. Vossoughi, A. Meghdari, H. Borhan, (2003) "Dynamic modeling and design of sliding mode control law for an underwater ROV", ISME, Mashhad, Iran.

[8] H. Borhan, G. R. Vossoughi, M. H. Saghafi,(2004) "Identification of Dynamical multipliers and thrusters of an underwater ROV", ISME, Tehran, Iran.