20th International Congress on Modelling and Simulation, Adelaide, Australia, 1–6 December 2013
www.mssanz.org.au/modsim2013

Intelligent Submersible Manipulator-Robot, Design, Modeling, Simulation and Motion Optimization for Maritime Robotic Research

Peiwen Guo[a], Amir Anvar[a], Kuan Meng Tan[a]

[a] *School of Mechanical Engineering, Faculty of Engineering, Computer, and Mathematical Science, The University of Adelaide, South Australia 5005, AUSTRALIA*
Emails: peiwen.guo@student.adelaide.edu.au; amir.anvar@adelaide.edu.au; kuan.tan@adelaide.edu.au

Abstract: With the recent technological advancement in submersible systems, the research and development of underwater manipulator robot is highly desirable. This paper focuses on comparing simulation versus real-time manipulator dynamics control, and comparison of intelligent underwater manipulator-robot for effective Ocean-based research, industrial and defense applications. Software package such as MATLAB is used to simulate the results of underwater manipulator performances and compare it with the real-time trials.

A mathematical model for underwater manipulator which encompasses deriving modified Denavit-Hartenberg parameters, computing all transformation matrices, deriving the forward kinematic and generating trajectory. Base on cubic polynomials, the manipulator trajectory is generated by using the joint angles and translation of each trajectory point. The result of manipulator performances is simulated by using SIMULINK. The manipulator simulation, which is aimed to analyze the movement of each arm, through the parameters of locations, velocity and torques are obtained. This paper also includes the comparison of manipulator-robot performance simulation with real-time trials. In order to measure the results of real-time, the Nintendo Wii remote is attached to manipulator to record the performances. The comparison can reveal the differences between simulation and real-time test in order to optimize for underwater manipulator.

Keywords: *submersible, intelligent, robot, underwater, manipulator, maritime, robotic, model, Ocean, simulate, motion optimization, modeling, simulation*

1. INTRODUCTION

For the purposes of exploring the whole Ocean and its abundant resources, the research and development of underwater robotics have been recognized as high economic and meaningful values. However, due to the manipulator's nonlinear dynamics, highly uncertain environmental variables and complex hydrodynamics, underwater manipulators present difficult control and complex design problems as shown by Sabiha et. al. (2011). In order to solve these problems and build more adroit and accurate underwater manipulators, a large number of research studies have been performed mainly involving manipulator dynamics and control. Researchers such as Dunnigan (1998) used reduction method for the dynamic coupling, while Jun (2004) analyzes the manipulability of the underwater manipulator to simulate the results of underwater robotic arms. Other researchers have also look into enhancing the efficiency of underwater manipulator, including the force feedback control of underwater manipulators mounted on the Remotely Operated Vehicle (ROV) shown by Ryu et al. (2001), the computer-based control and real-time motion compensation of deep-sea manipulators show by Hildebrandt et al. (2008) and motion planning, and the control of autonomous underwater vehicle (AUV) manipulator systems shown by Sarkar et al. (2001).

The simulation for manipulator is an important method when coping with complex construction or environment such as underwater robot. It is effective to use simulation tests to obtain a better understanding of the underwater manipulators which can improve the robotic arm design and avoid the costs and risks of high pressure and wet tests in a preliminary environment as discussed by Antonelli (2003). This paper presents manipulator simulation using the pre-existing software packages; MATLAB and ANSYS to modify the manipulator's performance, such as velocity, position and pressure in the underwater environment. In addition, the comparison of manipulator-robot performance simulation with real-time trials will be discussed.

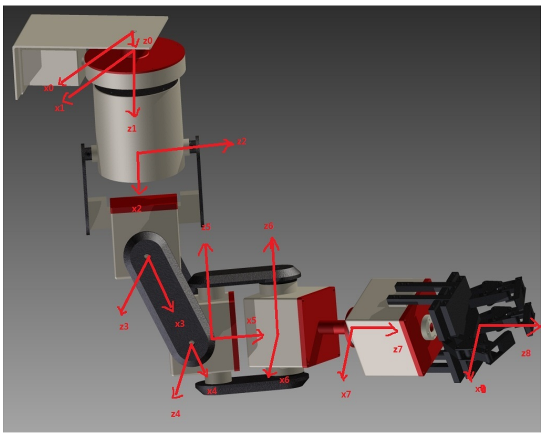

Figure 1. Manipulator structure and link frames

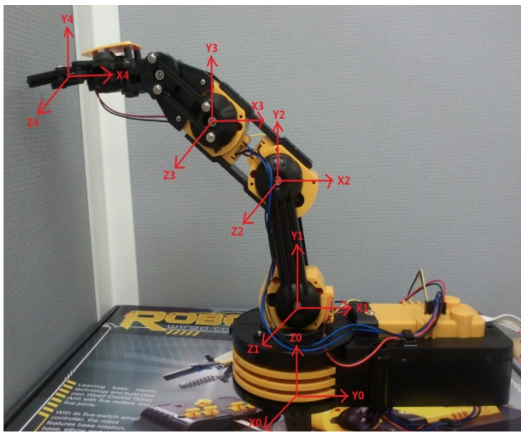

Figure 2. Model manipulator link frames

2. MANIPULATOR SIMULATION

It is to be noted that at the point of research, both manipulators (Figure 1 and Figure 2) are modeled within the simulator. However, for the purpose of comparing the accuracy of simulation model to the real-world manipulator interactions, off-the-shelf manipulator is purchased and used to describe the model (Figure 2). The purchased manipulator is modeled as a 4-degree of freedom joints with an end-effector, which includes

4 revolute joints with the order R1-R2-R3-R4. The manipulator can be indicated by the dimensions, which are L1, L3, L4, L5, θ1, θ2, θ3 and θ4. The symbols are used to represent the length and angle of the manipulator in the subsequent derivations. L represents the length of the arm in meters while θ specifies the rotation of the joints in radians.

2.1 Forwards kinematics

Table 1. D-H Parameter of the manipulator

i	α(i)	a(i)	d(i)	θ(i)
1	90	0	0	θ1
2	0	L2	0	θ2
3	0	L3	0	θ3
4	0	L4	0	θ4

The forward kinematics is used to determine the position and orientation of the end-effector, which is based on the information obtained from each local joint—frame using transformation matrix. This transformation matrix can be found by applying Denavit-Hartenberg parameters. Figure 2 shows the links of each frame for the manipulator. The manipulator is a robotic arm with four rotating joints and an end effector. The D-H parameters of the manipulator are illustrated in the Table 1. This table includes two parameters for the rotating angle and two for the link length. In this table, α is represented for the changing angle between adjacent Z axes called twist angle and d is the length between adjacent X axes called link offset. By providing a reference signal to the manipulator control system, the end effector can move to any desired position. Using the kinematics matrix and transformation matrix, the control system can express the values of the angles and orientation for each joint. The values of the angles, which generate control signals for the actuators, are used to move the manipulators. The transformation matrices can be obtained by using the D-H parameters of the manipulators.

$$ {}^0_1T = \begin{bmatrix} C1 & 0 & S1 & 0 \\ S1 & 0 & -C1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^1_2T = \begin{bmatrix} C2 & -S2 & 0 & L2C2 \\ S2 & C2 & 0 & L2S2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^2_3T = \begin{bmatrix} C3 & -S3 & 0 & L3C3 \\ S3 & C3 & 0 & L3S3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^3_4T = \begin{bmatrix} C4 & -S4 & 0 & L4C4 \\ S4 & C4 & 0 & L4S4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Then the forward kinematics matrix can be calculated by using the formula: ${}^0_4T = {}^0_1T \times {}^1_2T \times {}^2_3T \times {}^3_4T$

$$ {}^0_4T = \begin{bmatrix} n_x & O_x & S1 & d_x \\ n_y & O_y & -C1 & d_y \\ n_z & O_z & 0 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

Where:

$$ n_x = -C4(C1S2S3 - C1C2C3) - S4(C1C2S3 + C1C3S2) $$

$$ n_y = -C4(S1S2S3 - C2C3S1) - S4(C2S1S3 + C3S1S2) $$

$$ n_z = C4(C2S3 + C3S2) + S4(C2C3 - S2S3) $$

$$ O_x = S4(C1S2S3 - C1C2C3) - C4(C1C2S3 + C1C3S2) $$

$$O_y = S4(S1S2S3 - C2C3S1) - C4(C2S1S3 + C3S1S2)$$

$$O_z = C4(C2C3 - S2S3) - S4(C2S3 + C3S2)$$

$$d_x = L2C1C2 - L4S4(C1C2S3 + C1C3S2) - L4C4(C1S2S3 - C1C2C3) + L3C1C2C3 - L3C1S2S3$$

$$d_y = L2C2S1 - L4S4(C2S1S3 + C3S1S2) - L4C4(S1S2S3 - C2C3S1) + L3C2C3S1 - L3S1S2S3$$

$$d_z = L2S2 + L4C4(C2S3 + C3S2) + L4S4(C2C3 - S2S3) + L3C2S3 + l3C3S2$$

In (1), the symbol of C1 is represented by cosθ1. In the same way, C2 is cosθ2, C3 is cosθ3, C4is cosθ4, S1 is sinθ1, S2 is sinθ2, S3 is sinθ3 and S4 is sinθ4.

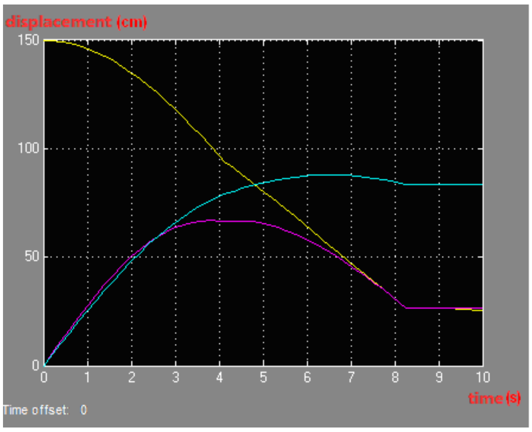

Figure 3. Displacement simulation of manipulator

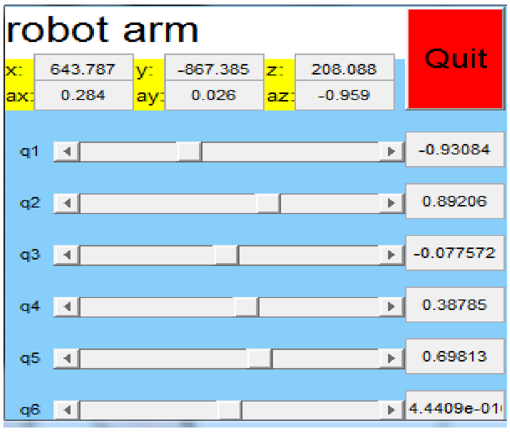

Figure 4. Manipulator Control Box GUI

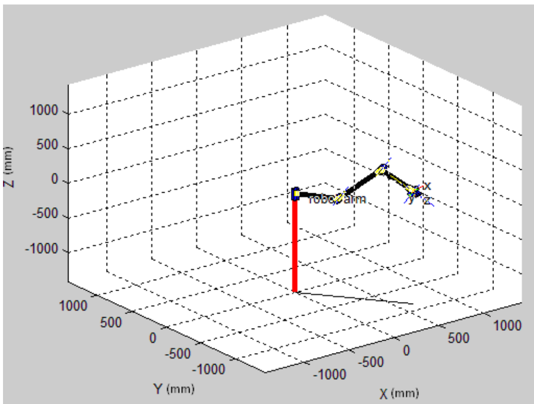

Figure 5. Manipulator representation in 3D space

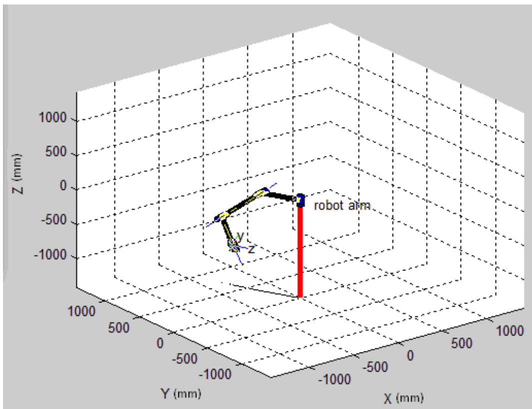

Figure 6. Changing orientations of manipulator

2.2 Manipulator simulation

Based on the transformation matrices and the D-H standard parameters, the graph which shows the displacements of three axes for end-effector of the manipulator can be calculated. A sample test case is shown (Figure 3) by maneuvering the manipulator from θ1 (-90º→0º), θ3 (0º→90º), and θ4 (0º→45º). In the Figure 3, the yellow line represents the X-displacement, the blue line represents Y-displacement, and purple line represents Z-displacement. While in the Figure 4, q1, q2 and q3 are the changing angles of the arms. The arms can move or rotate based on these three angles. q4, q5 and q6 are simulated for the end-effector of the manipulator. By varying these values, the end-effector final coordinates can be calculated; hence the mechanical grip is capable of reaching any target destinations (disregarding manipulator singularities). The following Figure 5 and 6 shows the graphical representation of the manipulator in 3D space within MATLAB environment.

2.3 Methodology and Results

Having low cost manipulator development in mind, one popular device which includes the required sensors for manipulator control, which is the Wii remote (Figure 7). The Wii remote also known as Wiimote, is the major controller for the Nintendo Wii video game console. It contains a three axis accelerometer, a vibration motor and a wireless Bluetooth communication. This provides a good base of sensor tracking and data logging functionalities. The software for computers, including Windows, Mac OS and Linux, is available for interacting with a Wii remote. Therefore, the Wii remote has become a popular device used in different new ways to track and control non Wii-related equipments. For the purpose of simple and quick tracking, the Wii remote is attached to the manipulator as a motion-tracking device to record the changing displacement.

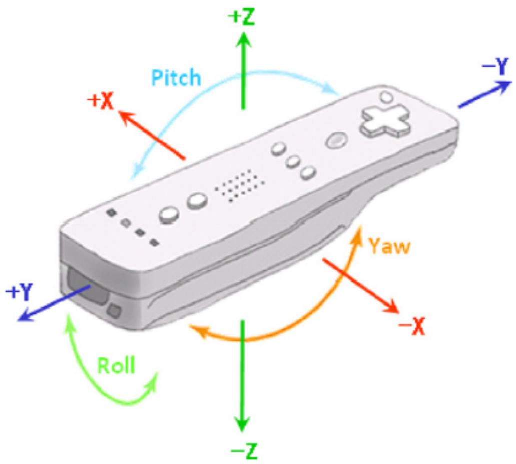

Figure 7. Wiimote reference axes

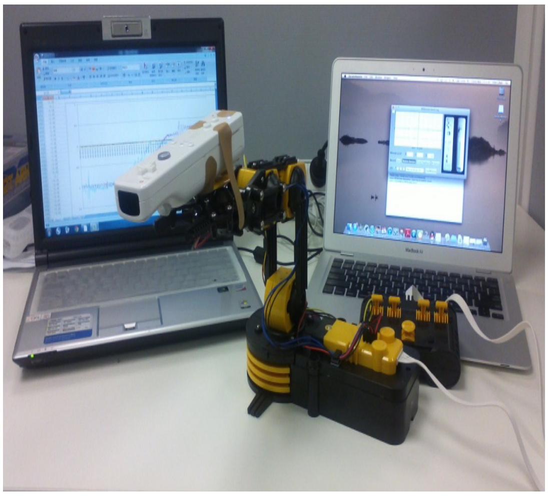

Figure 8. Manipulator arrangement

Possessing the accelerometer sensor and wireless Bluetooth connectivity, along with suitable assistive software, the Nintendo Wii remote controller can be used as a remarkable tracking device, and be applied to record the data for manipulator motion. The controller is used as a precise manipulator motion detector through software technique, to evaluate and record the displacement of manipulator which moves from one designated place to another designated place. Figure 10 shows the motion of the real-time manipulator which has the same trend as the simulated one. An approximation of experimental data using 4^{th}-order regression formula is applied to smooth out the accelerometer noise. It can be seen from the experiments that the z-axis acceleration data is not as accurate due to the nature of Wiimote's z-axis accelerometer and having the Wiimote strapped in parallel with the manipulator. The general trend of x, y, and z-displacement can be identified through this method. This shows the basic setup and fundamental tools required to conduct further experiments on this robotic manipulator. Further considerations such as proper filtering method (e.g. Kalman filter, band-pass filter) can also be applied to attain better reading; however, this is not covered within this paper.

2.4 Fuzzy Logic Control

The main purpose of applying fuzzy logic control is to be able to control the end-effector of the manipulator to suit required configurations. Mamdani fuzzy logic is being applied into the intelligent manipulator where three rotary torque sensors installed onto each rotational axis which allows for individual torques to be obtained. This can then be used as a membership function in the fuzzy logic control. Two linguistic variables are defined, the torques and the displacement between current manipulator configuration and the required destination of end-effector.

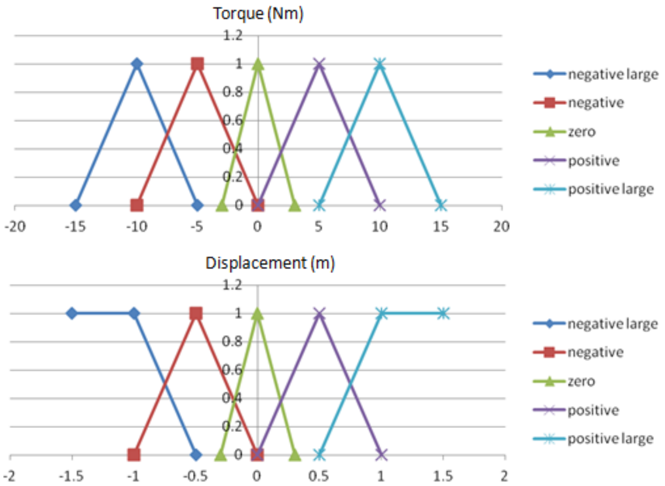

Figure 9. Fuzzy Logic Manipulator Control

3. DISCUSSION AND CONCLUSIONS

This project focuses on modeling and simulation of the submersible manipulator. In this study, the simulation model and real-time manipulator were compared. By discussing on one test case, it is easy to evaluate the differences between the simulation model and the real-time tests. This paper shows that the manipulator dynamics model can be easily produced with low-cost development prototyping. The comparison between the simulated scenario and real-world scenario can be obtained through the proposed method with ease (Figure 10).

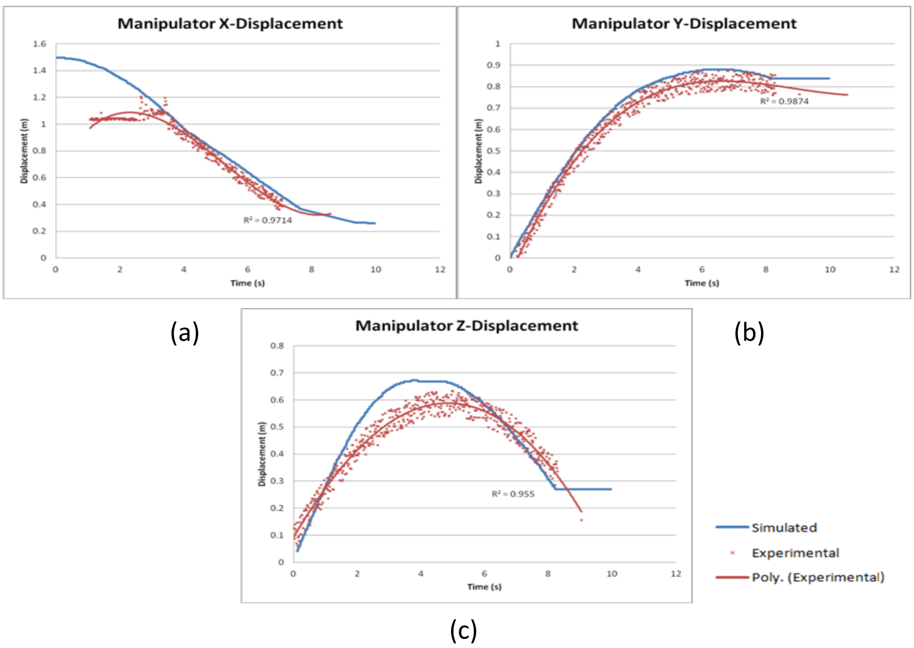

Figure 10. Comparison between simulation and experimental results using 4^{th} order regression

Future work includes, developing the underwater manipulator and applying the discussed method within the simulation and modeling. Detailed ANSYS analysis on drag of the manipulator within the underwater environment and its corresponding hydrodynamics parameters should be thoroughly studied. This will further improve the accuracy of the proposed model.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support from Defence Science and Technology Organisation (DSTO) during the course of the Project. A special thanks to the University of Adelaide and its Mechanical and Electronics Workshop staff for the support of equipment and software.

REFERENCES

Antonelli. (2003). A simulation package for underwater vehicle-manipulator systems. *Underwater Robots. STAR 2,* 159-163

Dunnigan, M.W. and Russell, G.T. (1998). Evaluation and reduction of the dynamic coupling between a manipulator and underwater vehicle. *IEEE Journal of Oceanic Engineering,* 23 (3), 260–273.

Hildebrandt, M., Albiez, J. and Kirchner, F. (2008). Computer-basedcontrolofdeep-sea manipulators. *OCEANS2008—MTS/IEEE KobeTechno-Ocean.*

Jun, B.H., Lee, P.M. and Lee, J. (2004). Manipulability analysis of underwater robotic arms on ROV. and application to task-oriented joint configuration. *MTS/IEEE/TECHNO-OCEAN.*

Ryu, J.H., Kwon, D.S. and Lee, P.M. (2001). Control of underwater manipulators mounted on an ROV using base force information. *2001 IEEE International Conference on Robotics and Automation.*

Sabiha, W. and Pushkin, K. (2011). Autonomous underwater vehicles modeling control design. *Taylor & Francis Group, LLC.*

Sarkar, N. and Podder, T.K. (2001). Coordinated motion planning and control of autonomous underwater vehicle manipulator systems subject to drag optimization. *IEEE Journal of Oceanic Engineering 26 (2)*, 228–239.