Evolution in Electrical and Electronic Engineering Vol. 2 No. 2 (2021) 709-718

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Simulation of Adaptive Control System for Multiagent Autonomous Underwater Cleaning Robots

Nurul Syafiqah Mohd Razali[1], Khalid Isa[1]*

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2021.02.02.085
Received 20 July 2021; Accepted 13 September 2021; Available online 30 October 2021

Abstract: Autonomous Underwater Vehicles (AUVs) have been extensively used in numerous applications and have gained worldwide recognition. These platforms can also be optimized and reconfigured according to the conditions in the aquatic world and should be autonomous. However, the AUVs may have problems in solving some tasks alone due to the large area of the underwater environment, and the underwater environment also had led to many problems in the kind of its cleanliness, which mostly caused by the bio-fouling activity problem. Using such a method may affect the system's stability and reliability. Therefore, this project is purposely to design an adaptive control system for multiagent autonomous underwater cleaning robots that can be applied to solve the problems of the underwater environment cleanliness, single robot problems, and avoid insignificant techniques that may cause faulty to the system. The MATLAB and Simulink software has been used in this project to design the adaptive control system of the multiagent autonomous underwater cleaning robots and simulate the designed controllers with multiple uncertain parameters together with a disturbance to be analyzed. The collected result consists of the designed 3D model using SolidWorks, a mathematical modelling structure of the robot, the analysis of open-loop control system, closed-loop control system of LQR, and the adaptive controller of MPC and lastly, the design of the multiagent system model structure for this work. The results show the adaptiveness, adjustment, and stability for the adaptive control system of multiagent underwater cleaning robots in order to cope with the situations of the underwater environment by using simulation.

Keywords: Autonomous Underwater Vehicle, Cleaning Robots, Adaptive Control System, Multiagent

1. Introduction

Malaysia is a maritime nation, and a source of revenue for 134,000 fishermen is its fishing industry [1]. Not just that, many underwater structures, such as storage tanks for petroleum and coal, electric power plants, ships, bridges, and oil rigs, are invaluable tools for our everyday lives and are the base

Corresponding author: halid@uthm.edu.my

for industry and the economy. Aquaculture is currently being marketed as a key engine of growth in Malaysia, with the goal of making it the country's economic foundation. So, it is important that Malaysia must be sustainable in maintaining the underwater environment.

However, biofouling build-up material is a big concern concerning underwater structures, and the solution is cleaning. This problem may reduce fish production because of the low oxygen concentration in the aquaculture cage. In order to maintain the aquaculture cleanliness, Malaysia had undergone a traditional hull cleaning method using divers with brushes, but this can put the delicate eco-system at risk and damage costly anti-fouling hull coatings. To overcome this, multiagent underwater cleaning robots must be introduced.

By introducing a system for a multiagent system architecture and controller algorithm approach, it aids in the control of multiagent robots to perform collective tasks while also adapting to dynamic situations such as those seen in biological systems [2]-[6]. Thus, the proposed controller algorithm aims to control the task of biofouling cleaning in a stable, and reliable manner. Therefore, this research's theoretical and algorithmic contributions contribute to a coherent system that possibly enables distributed agents to accomplish the task of self-adaptive cleaning locally and globally.

1.1 Aim of the Project

This project aims to design an adaptive control system for multiagent autonomous underwater cleaning robots that can be applied for bio-fouling cleaning tasks in the aquaculture industry.

1.2 Objectives

The objectives for the project are to design an algorithm for an adaptive control system for multiagent autonomous underwater cleaning robots, to simulate the adaptive controller with multiple uncertain parameters and disturbance of the underwater environment, and to analyze the performance of the proposed adaptive control system.

2. Methodology

The section will present the methodology used to develop the proposed adaptive control system for multiagent autonomous underwater cleaning robots. The first subsection presents the overall block diagram of the system, followed by structure design used to create the mathematical model of the robot for the simulation and finally the general notation used for the motion of the autonomous underwater cleaning robot.

2.1 Block Diagram

The first phase is to design the multiagent system architecture. The multiagent system is a computerized system composed of various intelligent agents that interact. So, in this project, there will be three robots that will cooperate and communicate with each other to do their task. The intelligence in this project may involve the control system algorithm. As a group of autonomous decision-making entities called the agents, a system will be modelled. Each agent assesses their situation independently and, on the basis of a set of rules, makes the decisions.

Then, in order to work on the multiagent system, the second phase is needed. MATLAB and Simulink will be used for the second phase to build the mathematical model and control algorithm. The mathematical model consists of two types of models, which is first the kinematic model and second is the dynamic model. This mathematical model is basically used for multiagent robots, underwater environments, and external disturbances. Other than that, the control algorithm is basically based on using a model to estimate the performance of the process on a future horizon and computing the control sequence that reduces a performance criterion involving an expected output sequence. The commonly used algorithm for this is the decentralized control algorithm based on self-adaptive concepts in biology.

The planned and collaborative activities can be achieved by multiagent structures. Each agent can use its distributed sensors, actuators, and communication devices to solve the cleaning task autonomously and cope with changes in the environment.

Using MATLAB and Simulink, the adjustment, stability, and adaptability to their environment will be analyzed and tested in order to simulate the performance of the control algorithm after the control algorithm has been developed. Lastly, the performance of the simulation of the system will be validated in terms of its adjustment, stability, and adaptively to its environment. The framework methodology diagram that intends to follow in order to achieve the concept of the project has been shown in Figure 1.

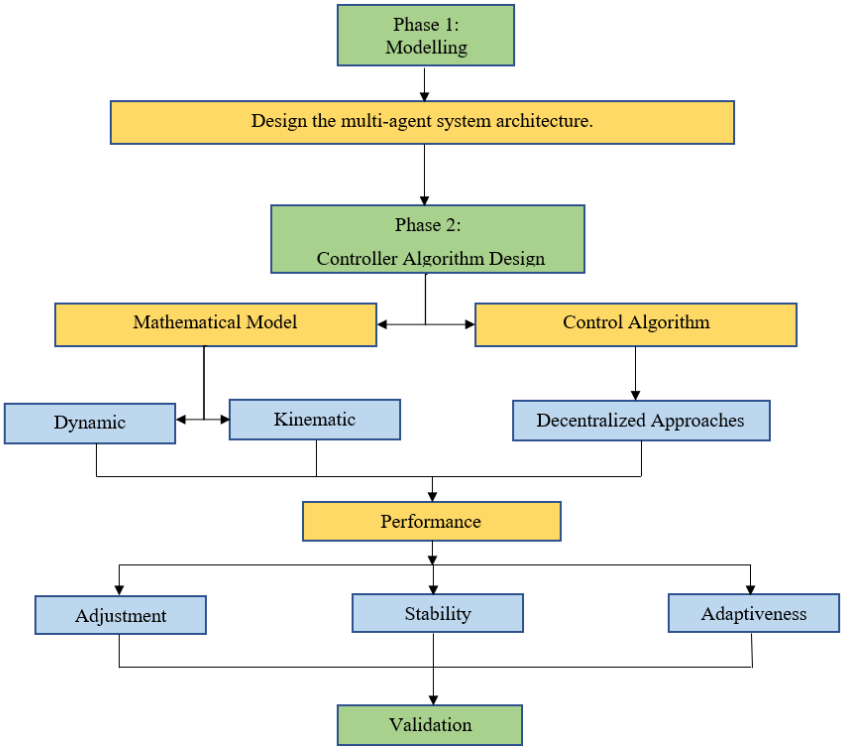

Figure 1: The phases of the framework methodology

2.2 Structure Design

The multiagent 3D model was created by using Computer-Aided Design (CAD) platform called SolidWorks. This structure was designed to obtain the mass properties of the robot created from the rigid computer model, SolidWorks. The structure design for the underwater robot in this project is triangular and equipped with two thrusters for the propulsion system. This underwater robot design is inspired by the Stingray fish, as shown in Figure 2.

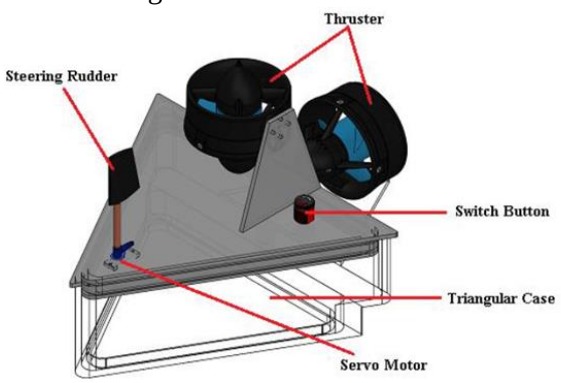

Figure 2: The conceptual designs of Underwater Cleaning Robot inspired by StingRay fish

2.3 General Notation for the Motion of the Autonomous Underwater Cleaning Robot

Six degrees of freedom (DOF) can be used to characterize underwater vehicle movement since six separate coordinates are required to determine the direction and orientation of the rigid body. As shown in Table 1, the six different motion components are described as 'surge',' sway',' heave',' roll',' pitch' and 'yaw'.

Table 1: Notation used for the underwater robots

DOF		Forces and moments	Linear and Angular velocity	Position and Euler angles
1	Motions in the x-direction (surge)	X	u	x
2	Motions in the y-direction (sway)	Y	u	y
3	Motions in the z-direction (heave)	Z	w	z
4	Rotation about x-axis (roll)	K	p	ϕ
5	Rotation about y-axis (pitch)	M	q	θ
6	Rotation about z-axis (yaw)	N	r	ψ

In principle, an AUV would be to travels in 6 Degree of Freedoms (DOF) that is linear and rotational motion in three directions, respectively, as illustrated in Figure 3. The motion of the fixed body frame is described relative to an inertial reference frame.

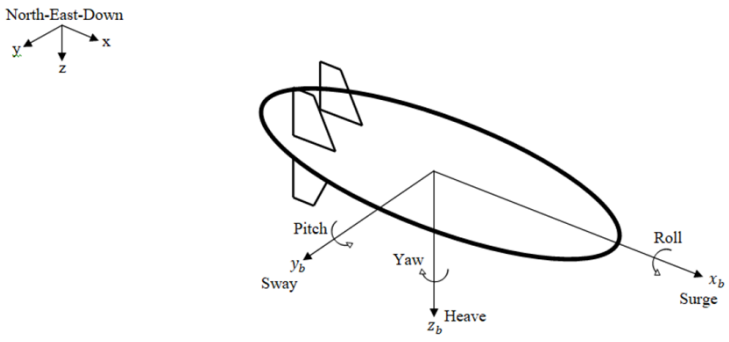

Figure 3: The inertial coordination systems and the 6 DOF illustration [2]

3. Results and Discussion

The results and discussion section presents the analysis of four different system design for the underwater cleaning robot.

3.1 Open-loop Control System

An open-loop is a system without feedback that directly generates the output in response to the input signal. The open-loop in this project is implemented just to show the nonlinear motion control scheme of the Underwater Robot motions without any feedback yet. By referring Figure 4, the obtained simulation results for this system show that the robot's nonlinear movement without a proper direction, or it can be said that the robot will moves freely in the simulation result.

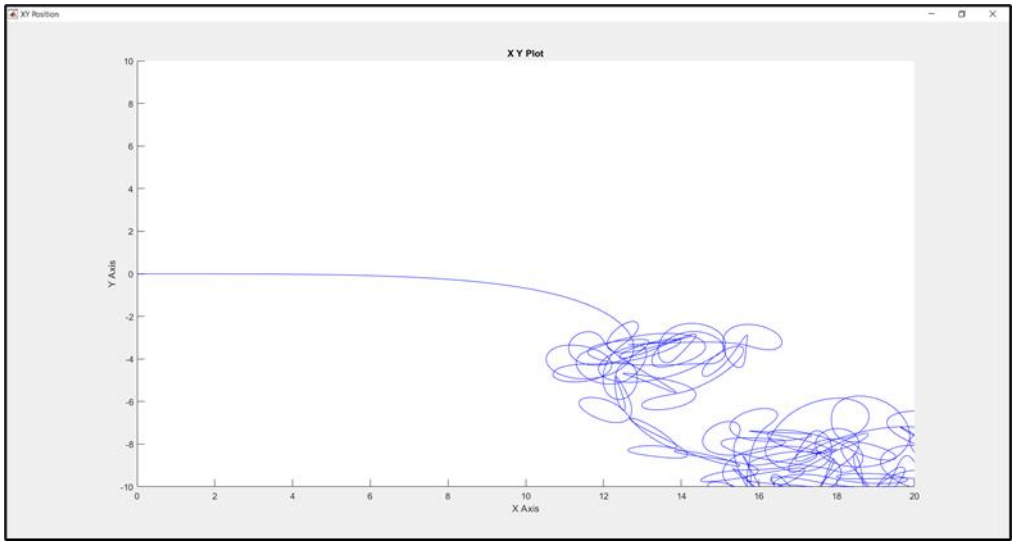

Figure 4: The final simulation result of the nonlinear motion of the robots in Simulink

3.2 Closed-loop Control System

Linear Quadratic Regulator (LQR) has been used in the closed-loop system for this project. The Linear Quadratic Regulator is one of the easiest controllers to construct, and it's also one of the finest in terms of performance and durability, so it is a great choice for evaluating this nonlinear model's closed-loop behavior.

The nonlinear model begins with the same input and initial condition as were used in the linearization. A slow and steady water current is used as a disturbance, and noise is added to each available measurement.

The attitude resulting from the closed-loop simulation is shown in Figure 5. In the presence of noise and disturbances, the LQR setpoint control is capable of stabilizing the nonlinear system and achieving satisfactory tracking performance. This can be seen in Figure 5 that the graph result was not constant at the first sample time. Then, the controller seems to interact with the LQR, the result became constant and stabilized at one point until the end of the running simulation.

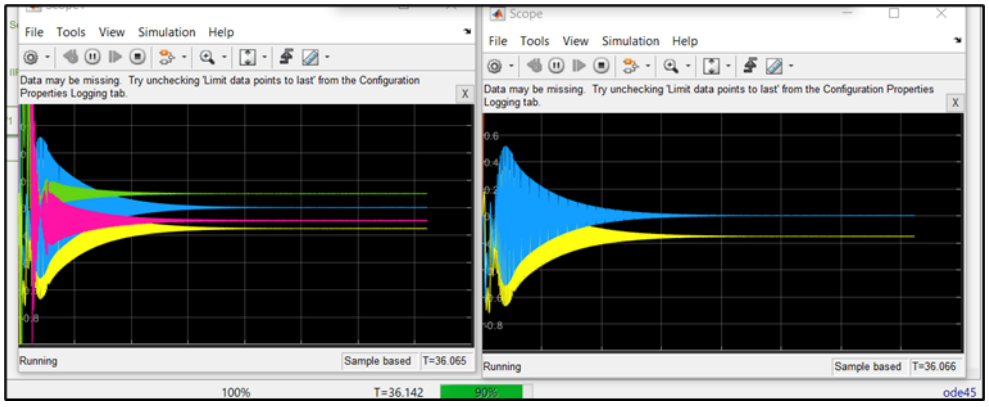

Figure 5: Closed-loop Simulation results.

3.3 Adaptive Controller of Model Predictive Control

In this work, the MPC circuit has been designed using the Simulink, as shown in Figure 6. The circuit is the continuous work to the closed-loop LQR circuit. Model predictive control (MPC) is an advanced form of process control that uses a set of constraints to regulate a process. Model predictive controllers are based on dynamic process models. MPC makes predictions about the system's future

behavior using a model of the system. MPC is capable of handling multi-input multi-output systems with interconnected inputs and outputs. It may also deal with input and output restrictions. The major benefit of MPC is that it allows optimizing the present timeslot while keeping future timeslots in mind.

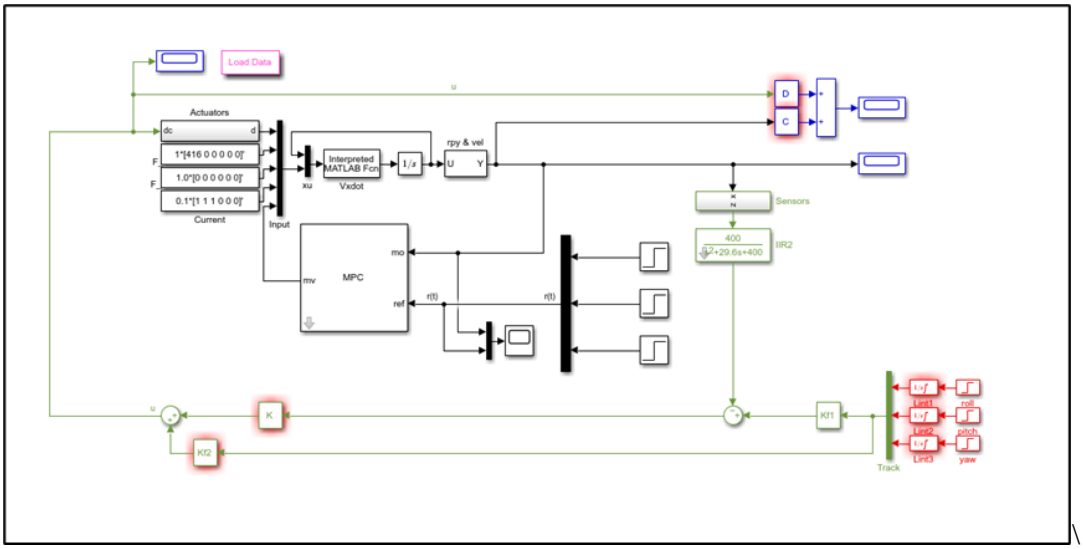

Figure 6: MPC Controller Simulink Circuit

In MPC, there are some design parameters that need to be concerned. They are sample time (Ts), prediction horizon, control horizon, constraints, and weights. Choosing proper values for this parameter is important as it affects not only the controller performance but also the computational complexity of the MPC algorithm that solves the optimization problems at each time step.

The first step to design the parameters for the MPC is to set the sampling time. In this work, the sampling time, Ts, that has been set in the MPC design is 0.1 seconds. Hence, after the MPC is being defined and linearize using the sample time that has been set, the app imports and linearizes the plant from the Simulink model and uses it as an internal model as shown in the blue box in the slide. In this work, the prediction horizon has been set as 10. If the prediction horizon in the MPC design setting increases to 15 or 20, the graph responses look more sluggish. Therefore, the prediction horizon for the MPC design in the Simulink was just set as 10.

The MPC controller makes predictions about the future plan output, and the optimizer finds the optimal sequence of control inputs that drives the predicted plan output as close to the setpoint as possible. The number of controls in the time step is called the control horizon. The control horizon can be taught as a free variable that needs to be computed by the optimizer. So, the smaller the control horizon, the fewer the computation moves. Therefore, choosing the really large control horizon only increases the computational complexity. The best choice in choosing the control horizon is setting it to a 10 to 20 prediction horizon and having a minimum of two to three steps of the control horizon.

For the constraints, the input constraints are dictated by the physical limitation of the vehicle. The attack of angle between the wind frame and fin fixed were assumed at 45 degrees. So, the minimum and maximum were set as pi/4 radian for the input constraints. Then, for the limit rate of change minimum and maximum, the attack angle was set to 22.5 degrees per second, so it becomes pi/8 radian.

Lastly, noted that MPC has multiple goals. The output needs to track as close as possible to their setpoint, but at the same time, we want to have smooth control moves to avoid aggressive control manuals. The default input weight at 0 was kept that way as it does not need to track a target. Then, the default weight value also has been kept. However, it can also be increased if an even smaller input increment is needed. For the output, the weight for y(1) position is set to 2.8, and the y(2) angle weight

is the divided value of 10 from the y(1) position weight value, which will be 0.28, as the position tracking is the primary objection.

Therefore, the result for the input and output responses for the plant model of the vehicle for this project after designing all the MPC parameters is shown in Figure 7.

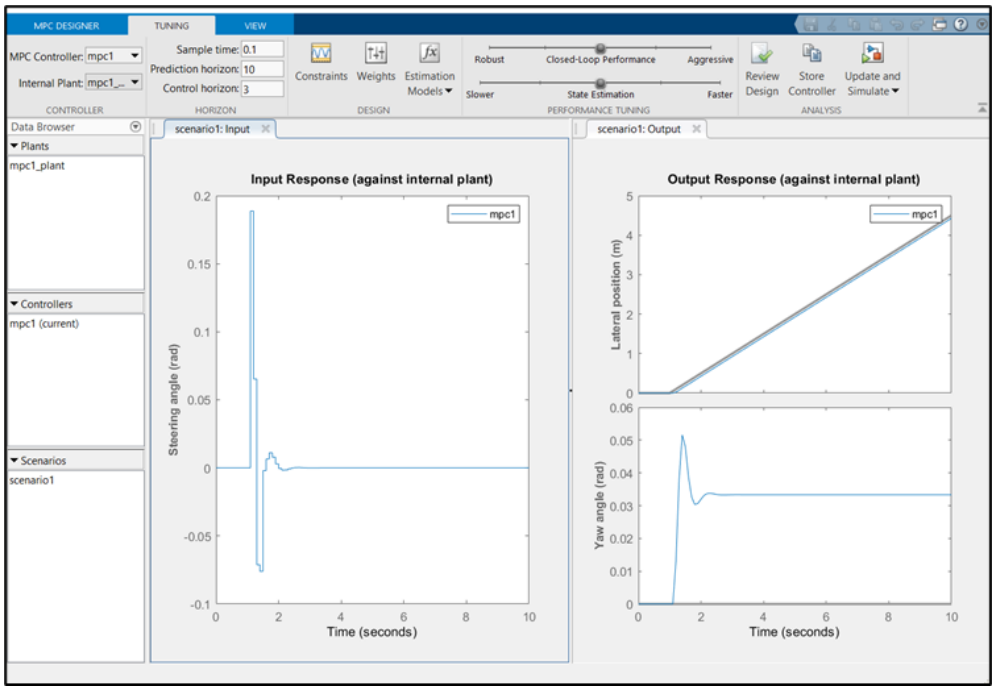

Figure 7: The final result of MPC input output responses after design the MPC parameters

3.4 Multiagent System Model

As shown in Figure 8, from the existing adaptive control structure of a single robot, it has been designed by adding two more of the controller circuit and connected all together. From the connection of the three-control system circuit, it shows that there are three agents for the multiagent system model.

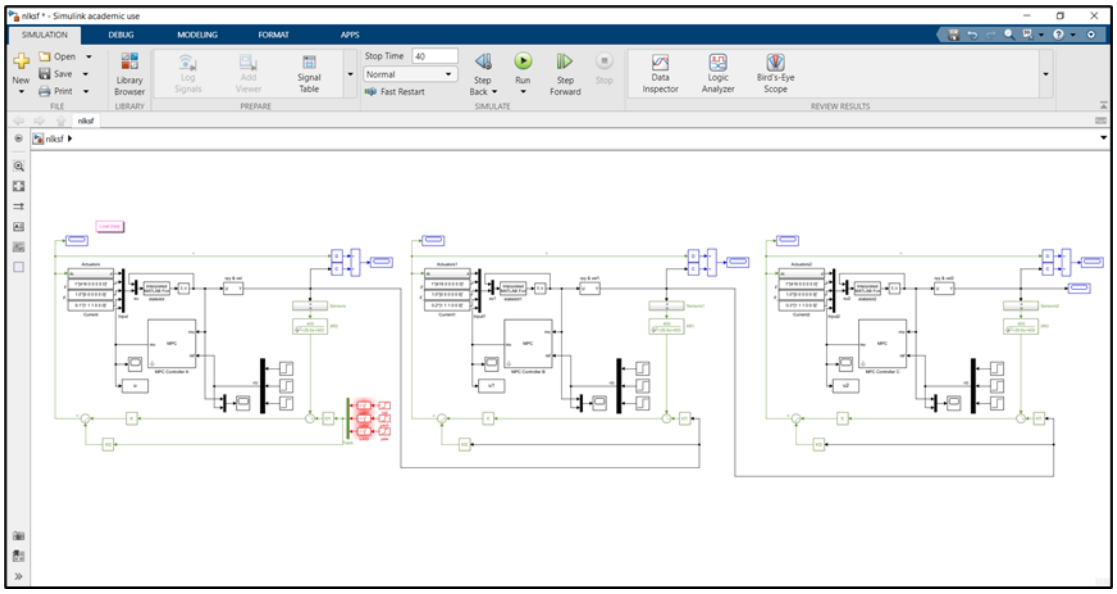

Figure 8: Multiagent system model in MATLAB Simulink

The connection among the three agents represents the interactions among the agents by sharing the control inputs from one agent to another so that their motions are coordinated together as a multiagent system. Therefore, by creating the multiagent system model, it can be described that a single robot cannot be utilized successfully for larger (more effort-intensive) tasks in which it is unable to complete the entire objective. A huge area that cannot be cleaned by a single cleaning robot due to time and resource constraints is an example of such a task. Using many robots simultaneously for a particular job is one such approach.

Lastly, the responses obtained for all the three agents are all the same and there is no different in the responses all the agents, as shown in Figure 9, even though the current applied to each agent are different, which the agent A current was 0.1, agent B was 0.2, and agent C was 0.3. This is because, even though there is different on disturbances, the motion of the agents is still shown that they are actually following the leader. The agent B and C are following the responses of agent A.

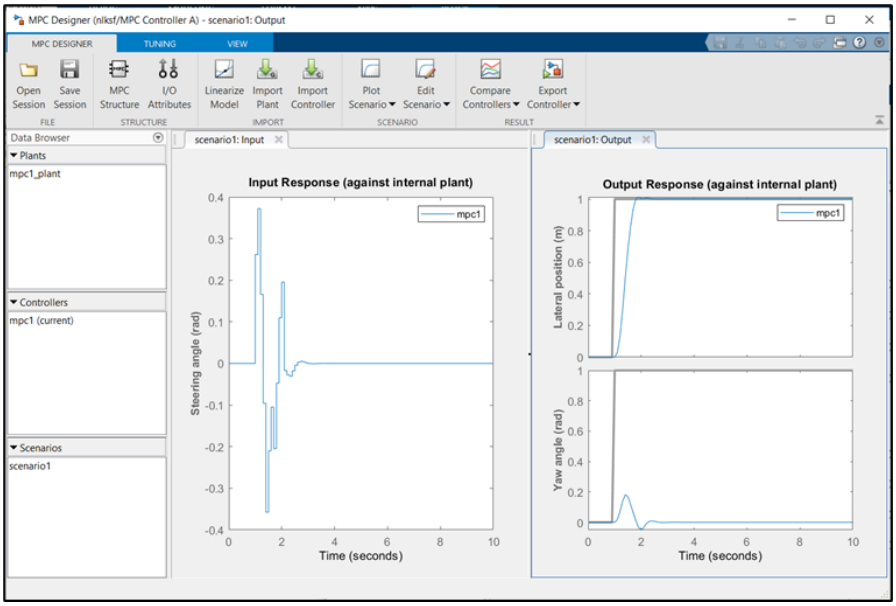

Figure 9: Output responses of the agent A, B and C adaptive controller are the same.

4. Conclusion

In conclusion, the objective of this project is accomplished, which is to design an algorithm for an adaptive control system for multiagent autonomous underwater cleaning robots, to simulate the adaptive controller with multiple uncertain parameters and disturbance of underwater environment, and to analyze the performance of the proposed adaptive control system. The adaptive control system design consists of the closed-loop LQR and MPC analysis. The algorithms of the project have been developed using the mathematical model of kinematic and dynamic. The kinematic mathematical model consists of the reference frames, the kinematics, which focuses on the orientation and position, the water currents, and also the current frames for the existence of the disturbance. Hence, the dynamic mathematical model consists of the robot equations of motion, which mainly discussed the forces that cause the motion. Other than that, the simulation being analyzed with the occurrence of the designated disturbances such as water currents, and each accessible measurement has some noise added to it. Hence, it was also being analyzed with some uncertain parameters' adjustment to the controllers for the analysis. Lastly, the designed control system was analyzed in terms of the performance of the controller, the differences between two types of controllers, which are LQR and MPC, and also the designed parameters used.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Yusoff, Aishah, Status of resource management and aquaculture in Malaysia, Proceedings of the International Workshop on Resource Enhancement and Sustainable Aquaculture Practices in Southeast Asia 2014 (RESA) (pp. 53-65).

[2] M. Andersson, "Automatic Tuning of Motion Control System for an Autonomous Underwater Vehicle," 2019.

[3] Tewolde, G.S.; Changhua, W.; Yu, W.; Weihua, S. Distributed multi-robot work load partition in manufacturing automation. In Proceedings of the 2008 IEEE International Conference on Automation Science and Engineering, Arlington, VA, USA, 23–26 August 2008; pp. 504–509.

[4] T. Fossen and S. I. Sagatun, "Adaptive control of nonlinear underwater robotic systems," in Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference, pp. 1687–1694 vol.2, Apr 1991.

[5] S. Zhao, J. Yuh, and H. T. Choi, "Adaptive dob control of underwater robotic vehicles," in OCEANS, 2001. MTS/IEEE Conference and Exhibition, vol. 1, pp. 397–402 vol.1, 2001.

[6] Corina Barb˘alat˘a, Valerio De Carolis, Matthew W. Dunnigan, Yvan Pétillot, David Lane, " An Adaptive Controller for Autonomous Underwater Vehicles," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Congress Center Hamburg, Sept 28 - Oct 2, 2015.

Appendix A

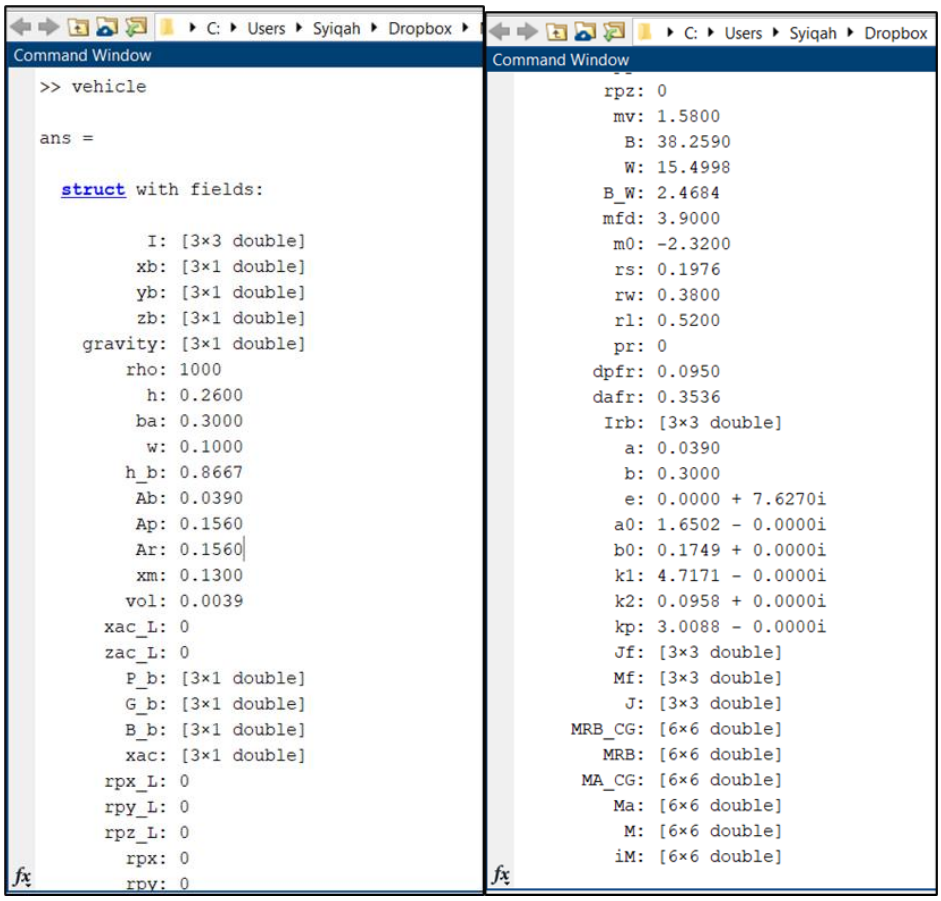

Figure A: The Result of Structure Modelling of the vehicle's structure model in MATLAB Command Window