

## **dSPACE DS-1104 based Real-Time Verification of 3D Collision Avoidance for Autonomous Underwater Vehicles**

**Saravanakumar Subramanian**

Research Scholar, Department of Engineering Design, IIT Madras, India

**Thomas George**

B.Tech Student, Department of Mechanical Engineering, IIT Hyderabad, India

**Asokan Thondiyath**

Associate Professor, Department of Engineering Design, IIT Madras, India

### **ABSTRACT**

Hardware-in-the-Loop (HIL) simulations play a major role in the testing and verification of underwater robots and control strategies. Not many literatures are found for the real-time HIL verification of the control algorithms for Autonomous Underwater Vehicles (AUV) in 3D space. Most of the reported real-time simulators are virtual simulators and are mainly used for visualizing the AUV motion only. Hence a HIL test set up is required well before the actual vehicle is developed and deployed in the water. This paper discusses the development of HIL setup for collision avoidance of an underactuated flat-fish type AUV using dSPACE DS1104 R&D controller and ControlDesk experiment software. The results show that the dSPACE HIL setup is an effective tool for the verification of control algorithms and the developed collision avoidance algorithm can be used in real-time for the flat-fish type AUV. By using this developed dSPACE environment, the necessary improvements and modifications can be done before actually testing the vehicle and probability of test-bed vehicle collisions with obstacles can be greatly reduced. In this way, dSPACE products and solutions provide safe and reliable evaluation.

### **INTRODUCTION**

Autonomous Underwater Vehicle (AUVs) is a robotic device that is driven through the water by a propulsion system, controlled and piloted by an onboard computer, and maneuverable in three dimensions. In recent years, AUVs have become more popular due to their autonomous nature of operation, long-duration exploration of the ocean, ability to gather information and great potential in both military and civilian applications. Real-time testing and verification of the control architecture and control algorithm of such vehicles is not an easy task. It involves not only the development and deployment cost of the vehicle but also the construction time, real-time testing and monitoring of the vehicle [1], [2]. The data acquisition and analysis is also a costly process as the real-time testing of actual vehicle in remote environments is often tedious due to inaccessibility and hazardous nature. Safety is a primary issue that one is facing in

conducting actual vehicle tests [3]. Virtual simulators are often employed for testing either in off-line or on-line method. In offline simulators, the validation of control algorithm is not possible as the temporal aspect of the simulation is not taken into account and the control algorithm may fail when it is implemented on the real system. In on-line simulations such as virtual reality simulator, though the temporal aspects are considered, the temporal behavior of the computer used for the simulation can be different from the one onboard the real system. Hence a HIL test set up is required well before the actual vehicle is developed and deployed in the water to detect and prevent unnecessary malfunctions of hardware, software and other control systems [4].

HIL tests have been widely used in automobile industry, aviation, aerospace, robotics and large complex control systems in which investment and risks are high. In the field of underwater robotics, the real-time validation of control algorithm that uses HIL simulation tools is slowly emerging and recently surfaced in literature [5]. The dSPACE is one of the HIL simulation tools and is very powerful for real-time validation of the control algorithms. It uses its own real-time interface implementation software to generate and then download the real-time code to specific controller board. Some of the major features of dSPACE are: 1) Automatic generation of controller code for hardware implementation 2) Direct interface of Simulink models 3) On-line tuning of controller gains which eliminates rebuild and download a new Simulink model 4) Simple graphical user interface for ease of operation[6]. In this paper we discussed the development of HIL setup for collision avoidance of an underactuated flat-fish type AUV using dSPACE DS1104 R&D controller and ControlDesk experiment software. It has been observed from the results that the dSPACE HIL setup is an effective tool for the verification of control algorithms and the developed collision avoidance algorithm can be used in real-time for the flat-fish type AUV.

## **HIL SIMULATION SYSTEM**

The detailed structure of HIL simulation used for verifying the proposed collision avoidance algorithm for AUV is shown in Fig. 1. The dSPACE DS1104 R&D controller board is used here as it is a complete real-time control system based on 603 PowerPC floating point processor running at 250MHz. For advanced I/O purposes, the board includes a slave-DSP subsystem based on the TMS320F240 DSP microcontroller. The interfacing unit CLP1104 Connector/LED combi Panel connects the DS1104 R&D Controller Board with the devices to be communicated to the processor. It also provides an array of LEDs indicating the states of the digital signals. The implementation software of dSPACE, "RTI" is used to convert the Simulink simulation model into real-time interface model. The control algorithm is developed in Simulink and executed using dSPACE DS1104 controller board. ControlDesk version 3.5 provides all the functions to control, monitor and automate experiments. These features make the development of control algorithms more effective. It allows the user to access any I/O signals that are connected to the hardware components.

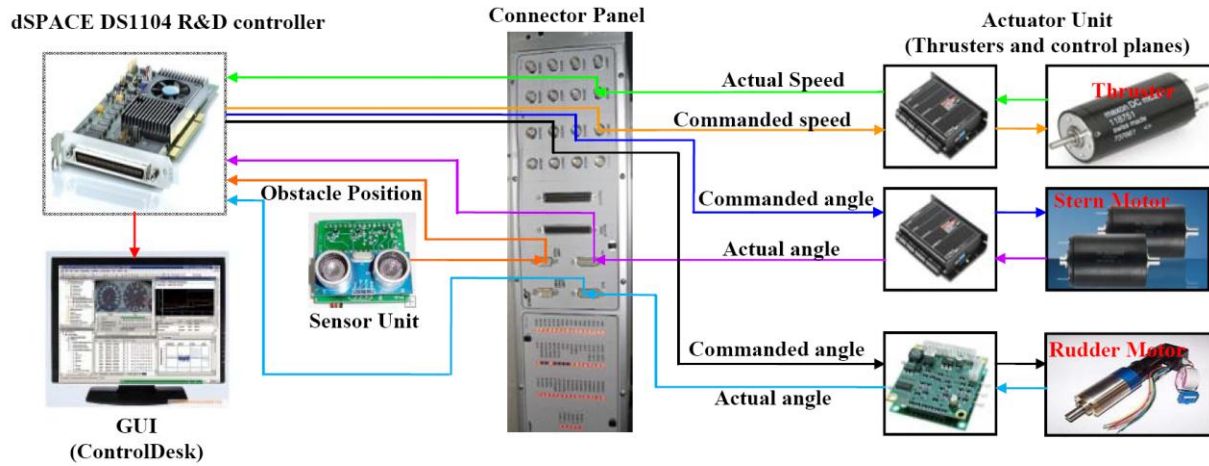


Fig. 1. Schematic diagram of HIL Simulation system

## DEVELOPMENT AND IMPLEMENTATION COLLISION AVOIDANCE TO AUV

The objective of the collision avoidance algorithm is to avoid collision with static and dynamic obstacles in the environment surrounding the vehicle. An improved potential field method is used as a basic approach for developing our algorithm [9]. The main idea of the potential field method is to generate attraction and repulsion potentials for the target and the obstacles. The target has an attraction potential and the obstacles have repulsion potential. In the improved method, the total potentials are generated at multiple points. By determining the point at which the minimum potential exists among the total potentials, the vehicle can be commanded to that point. The developed collision avoidance algorithm is interfaced with the AUV dynamic model and other necessary systems such as a sensor signal processing module, a trajectory planner, a controller and actuators (thrusters and control planes). Figure 2 shows the structure of the collision avoidance algorithm for an AUV. The collision avoidance algorithm is executed at a time interval ' $T$ ' that is proportional to the velocity of the vehicle. Once the user inputs are given, this block gets the sensor data for every time,  $T$  and then it defines the next one-step position as path elements to the trajectory planner. The present positions are delayed and given as past input to the trajectory planner in order to calculate the desired yaw angle. Upon receiving the data, the desired trajectory is generated. As the forward speed of the vehicle is too slow and is taken as constant, a straight line trajectory in Cartesian space is implemented. The desired trajectory information is given to the PID controller. Finally, the manipulated variable (total force) is obtained from the actuator and is given to the vehicle dynamics. The blocks in the inner loop are executed with a time interval ' $t$ ' of the step size specified in the solver. The current states of the vehicle are given to the controller and the collision avoidance algorithm till the vehicle reaches the target.

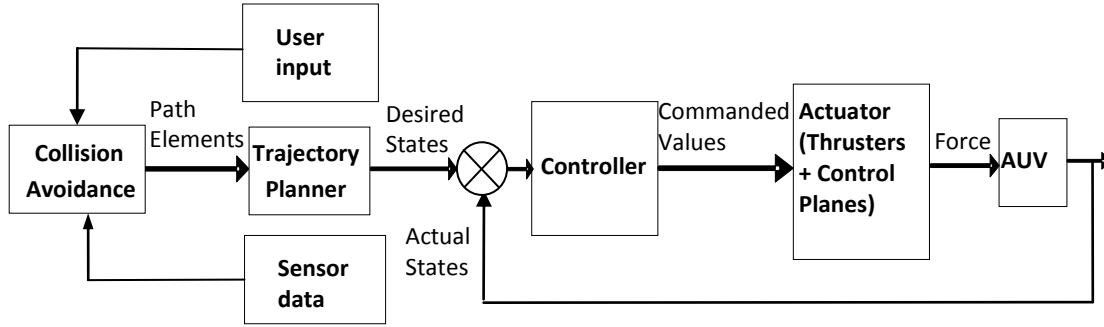


Fig. 2. Implementation of collision avoidance algorithm.

An ultrasonic sensor is placed at the front portion of the AUV to detect the obstacles. The sensor not only detects the presence of the obstacles but also measures distance between the vehicle and obstacle. The simulation model receives the starting and goal positions as user input and the obstacles positions from the sensor data. The collision avoidance algorithm generates the next one-step positions as path elements and the trajectory planner generates the desired trajectory. The coordinate values of the desired trajectory are compared with the actual values obtained from the AUV simulation model. The tracking errors are calculated and the corresponding control commands (desired speed and angle) are generated by the controller. These compensating signals are converted into equivalent analog voltages and these voltages are given to the corresponding driver boards of the actuator unit. The actuator unit consists of three motors which are used to control the motion in surge axis (thruster), pitch axis (stern plane) and yaw axis (rudder). The encoder associated with the surge control motor unit measures the analog voltages corresponding to the actual forward speed. Similarly the encoders of the pitch and yaw control motor units measure the corresponding angle information. The measured signals are converted into equivalent digital signal and they are given to actuator dynamics in which the required forces are generated for the vehicle model. These values are taken as the inputs of the simulation model and the actual states are calculated.

The real-time interface to the Simulink model is shown in Fig. 3. Here the standard I/O channels blocks such as ADC and DAC and more complex I/O devices blocks such as incremental encoder and RS-232 are picked up from the I/O library and attached to the Simulink model. The serial receiver is set to receive the obstacle sensor data at 9600 baud rate. The input channel ADC 5 is used to read the analog voltage corresponding to the actual speed of the thruster motor. The output channels DAC 1, 2 and 3 are used for sending out the desired analog signals to the thruster, pitch (stern) and yaw (rudder) motors respectively. The angular position of pitch and yaw motors are read by the encoder channels 2 and 1. After connecting the necessary I/O blocks, the Simulink model is then transferred into real-time code using the RTI implementation software. Once the simulation model is converted into RTI model, the real hardware can be connected. The various parameters to be displayed and plotted are selected from the instrument and data acquisition panels of the ControlDesk software. Their corresponding data

are captured for further analysis of the results. In order to get clear idea of the developed hardware-software integrated environment, the HIL test bench setup is shown in Fig. 4.

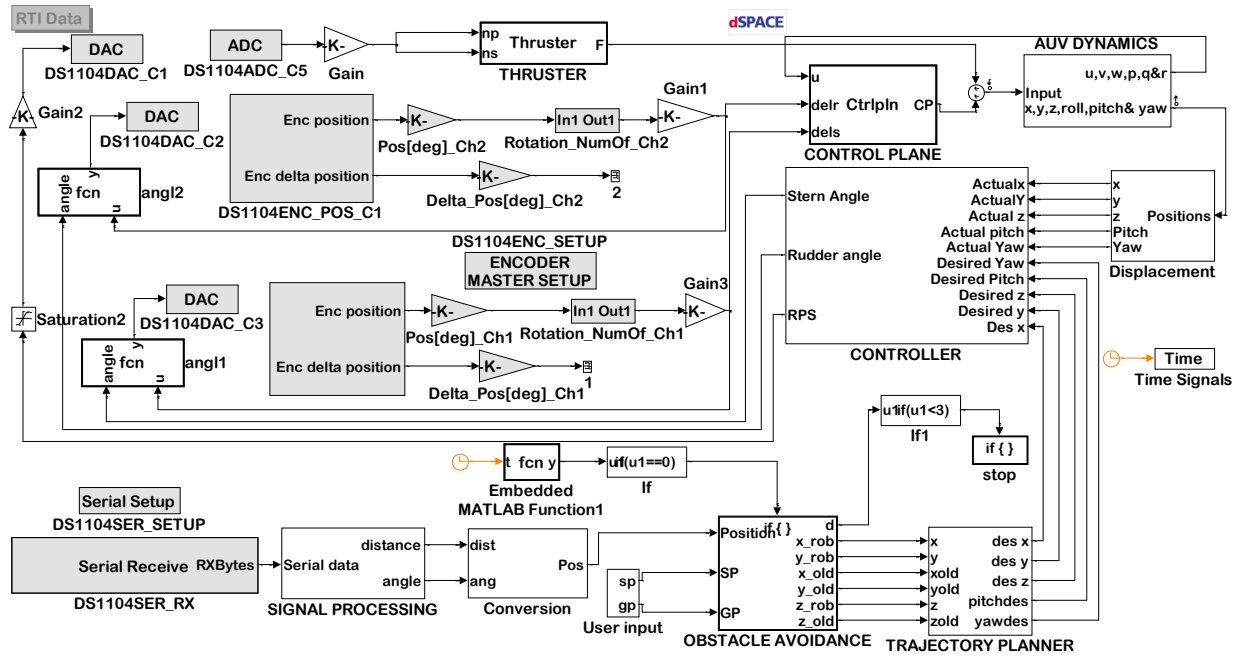


Fig.3. Simulink model of HIL simulation system

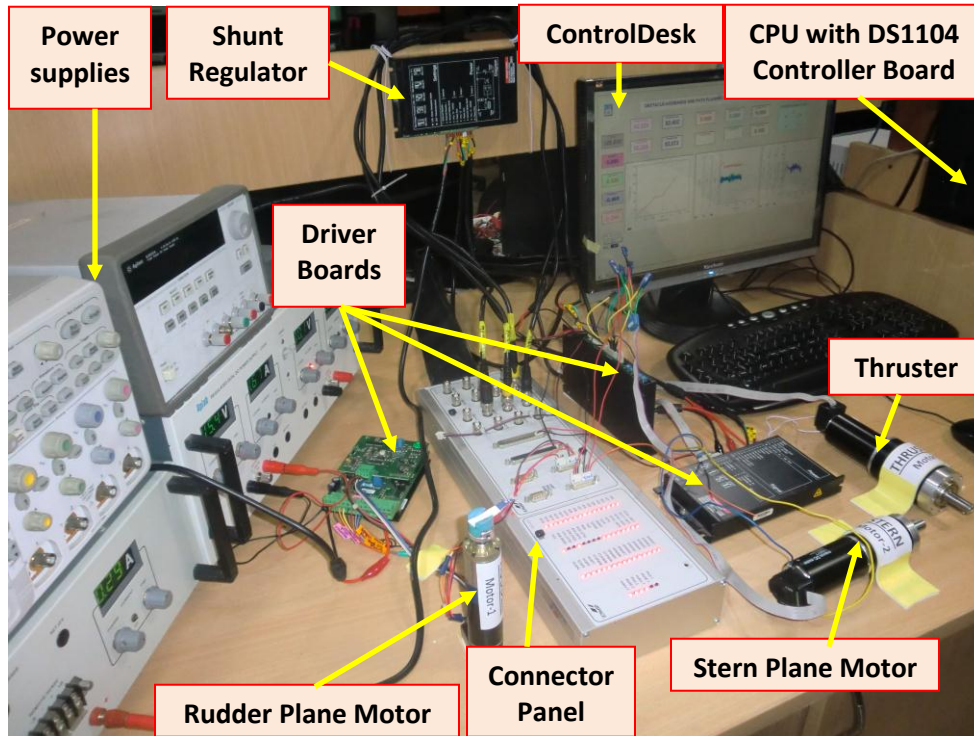


Fig. 4. HIL test bench setup.

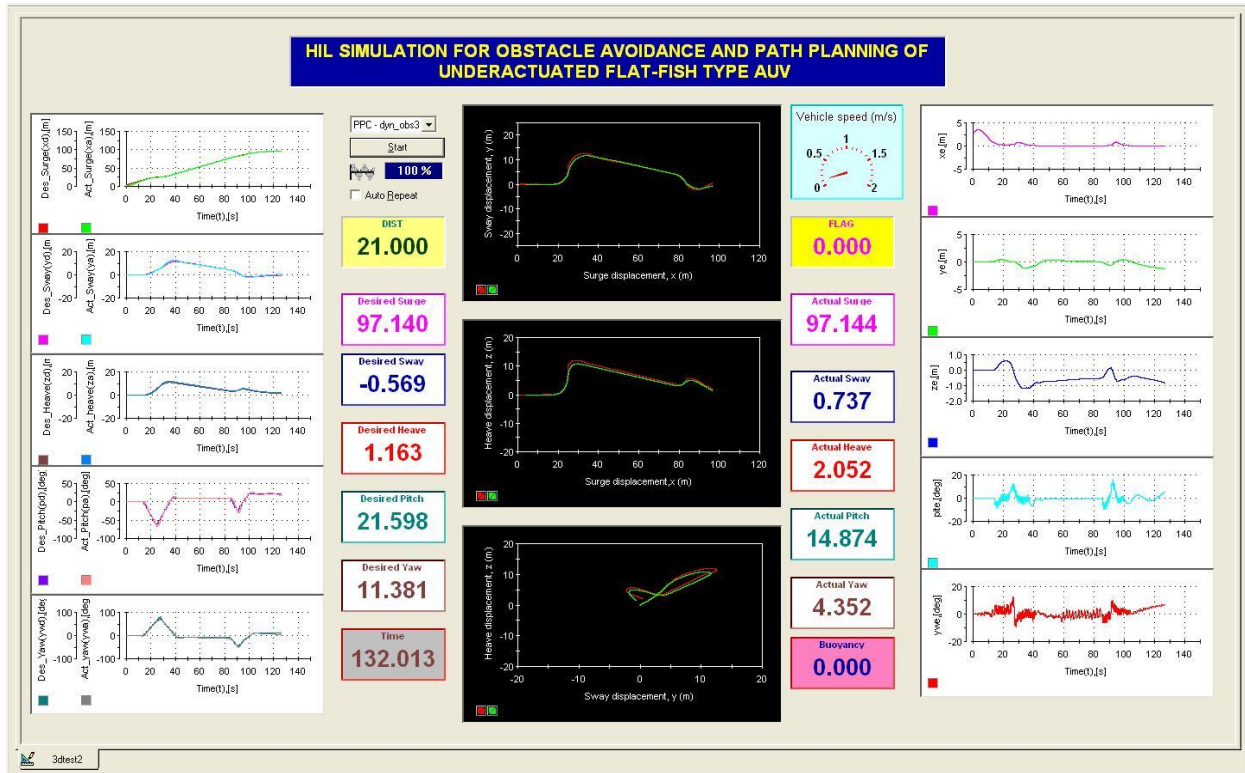


Fig. 5. GUI based ControlDesk layout.

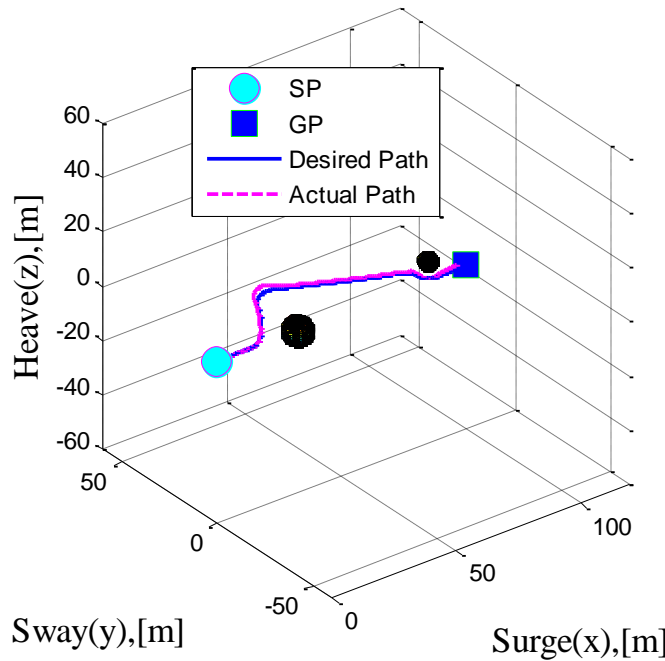


Fig.6. Path generated using dSPACE HIL setup in 3D space.

There are two obstacles introduced along the path of the vehicle at random times. The control desk layout for this case is shown in Fig. 5. If the obstacle is detected then the flag is set to one else it is reset to zero. The distance and flag data are displayed in the layout. It has been found from the HIL simulation results that the developed collision avoidance with AUV dynamic model avoids the collision effectively and reaches the goal position. The position and orientation are changed when the vehicle approaches the obstacle. It has also been observed that the position and orientation errors are within the acceptable limits. The 3D path generated by the on-line collision avoidance by multipoint potential field method and the path tracked by the vehicle is shown in Fig. 6.

## CONCLUSION

We have presented the HIL simulation for verifying the collision avoidance algorithm of AUV in 3D space using dSPACE DS 1104 real-time controller. The results show that HIL setup is very much useful and effective for verifying the control algorithms of real time systems. With this developed dSPACE environment necessary improvements and changes can be done before testing the actual vehicle in underwater and the probability of test-bed vehicle collisions with obstacles can be greatly reduced. It also reduces the time required for validating the developed control algorithm for real-time implementation. In this way, dSPACE products and solutions provide safety and reliability evaluation. The HIL setup will be extended for various control algorithms and the results will be presented in near future.

## REFERENCES

- [1] S. Saravanakumar, K. Divya, B. Keerthana and T. Asokan, "3D modeling and virtual simulator for a flat-fish shaped autonomous underwater vehicle", in *Proc. 2nd International Conference on Modeling, Simulation and Analysis*, India, 2011, pp.385-390.
- [2] T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics AS, Trondheim, Norway, 2002.
- [3] "HIL Testing", <http://www.marinecybernetics.com/hil-testing>.
- [4] A. Martin and M. R Emami, "An architecture for robotic hardware-in-the-loop simulation", in *Proc. IEEE International Conference on Mechatronics and Automation*, 2006, pp.2162-2167.
- [5] G. Cai, B. M. Chen, T. H. Lee and M. Dong, "Design and implementation of a hardware-in-the-loop simulation system for small scale UAV helicopters", in *Proc. IEEE International Conference on Automation and Logistics (ICAL 2008)*, 2008, pp.29-34.
- [6] "dSPACE", Catalog, dSPACE, S. Jaswandi, and G.Divyesh, "dSPACE DSP DS-1104 based state observer design for position control of DC Serve motor", in *dSPACE User Conference*, India, 2010.

[7] <http://www.dSPACE.com>

[8] <http://www.dynafusiontech.com>

[9] S. Saravanakumar, T. Asokan, T, “Waypoint guidance based planar path following and obstacle avoidance of autonomous underwater vehicle”, in *8th International Conference on Informatics in Control, Automation and Robotics*, Noordwijkerhout, The Netherlands, 2011 pp.191—198.

#### CONTACT:

The work has been carried out in the Robotics Laboratory of Engineering Design Department, IIT Madras.

#### **Saravanakumar Subramanian**

Research scholar,

Robotics Laboratory,

Department of Engineering Design,

IIT Madras, Chennai-600036, India

E-mail: [srsaravanakumar@gmail.com](mailto:srsaravanakumar@gmail.com)

#### **Dr. Asokan Thondiyath**

Associate Professor,

Department of Engineering Design,

IIT Madras, Chennai-600036, India.

E-mail: [asok@iitm.ac.in](mailto:asok@iitm.ac.in)

Home page: <http://ed.iitm.ac.in/~asokan>