(Preprint) AAS 23-322

DEVELOPMENT OF A 12 DEGREES-OF-FREEDOM ROBOTIC TESTBED FOR EXPERIMENTAL ANALYSIS OF IN-SPACE MISSIONS

Anirudh Chhabra* **and Donghoon Kim†**

In-space servicing, assembly, and manufacturing missions have become increasingly important as an increasing number of space operations are conducted every year. These missions are becoming popular as they allow the repair and refueling of existing space platforms, autonomous space missions, etc. To successfully perform these missions, Hardware-In-the-Loop (HIL) simulations are necessary to experimentally validate planning methods, control algorithms, and maneuver design processes. For performing such HIL simulations, spacecraft simulators are required. This study proposes a new spacecraft simulator that offers several advantages over existing simulators. The paper focuses on the design and construction of the platform, followed by experimental validation of the platform by simulating a docking maneuver.

INTRODUCTION

In-space Servicing, Assembly, and Manufacturing (ISAM) missions are becoming increasingly important as the number of space operations increases every day.[1] These missions allow for increasing the lifespan of existing spacecraft/satellites, constructing new stations in space, and performing routine maintenance on important orbital systems like space stations. These missions can therefore fundamentally be categorized into rendezvous, docking, refueling, servicing, etc. Such operations require a servicer spacecraft to perform precise maneuvers in close proximity to a target spacecraft or satellite. Validating these missions and corresponding technologies in Hardware-In-the-Loop (HIL) simulations is necessary to study contact dynamics and ensure stable and accurate operation.

Several HIL spacecraft simulators have been designed and developed based on different underlying technologies in various facilities around the world. These can be broadly categorized into the following types: 1) Orbital micro-gravity systems, 2) Sub-orbital micro-gravity systems, 3) Neutral buoyancy facilities, 3) Air-bearing systems, 4) Suspension systems, and 5) Ground robotics-based simulators.[2] Orbital micro-gravity systems such as the SPHERES and ASTROBEE are very good since they are operated in space but are very expensive to operate and maintain.[3,4] Furthermore, the amount of time available to do several experiments is less. Neutral buoyancy facilities, although used for astronaut training, are not well-suited for robotic simulations since there are several uncertainties due to the dynamic behavior of the surrounding fluid.[5] Moreover, they only offer restricted rotational motion and have to be operated with a limited power supply. Air-bearing systems are

*Graduate Student, Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA, chhabrad@mail.uc.edu
†Assistant Professor, Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA, kim3dn@ucmail.uc.edu

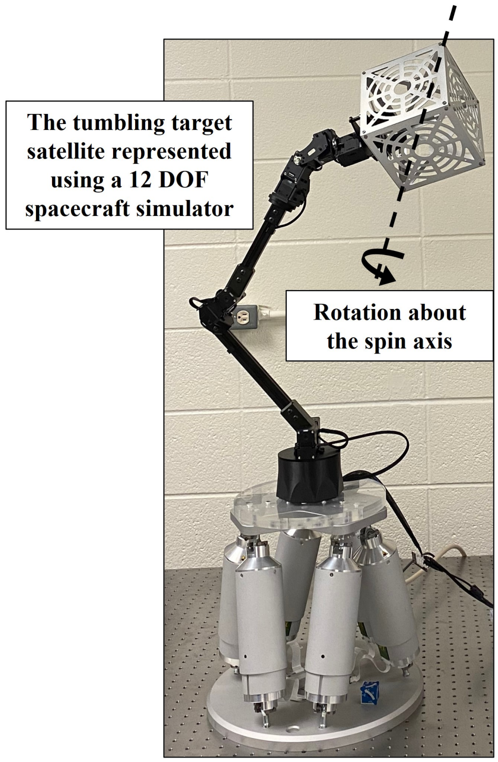


Figure 1: HIL Simulation of a Spacecraft using the Proposed Spacecraft Simulator.

one of the most common spacecraft simulation systems but are very expensive and only offer limited motion capability (translation in x and y axes and rotation about yaw axis) instead of 6 DOF motion capabilities since the platform hovers on a flat surface. Although such platforms can offer some motion about roll and pitch axes, it is always constrained due to the physical limitations of the design.[6] Additionally, they offer low endurance due to the dependency on the use of pressurized gas cylinders and/or limited power. Suspension systems have been used in several facilities such as the NASA Active Response Gravity Offload System (ARGOS) but are highly complex to use in lab environments and offer low accuracy since the behavior of suspension gantries is difficult to control on a big scale.[7] Finally, ground robotics-based simulators offer a lot of advantages such as a great range of motion and flexible control such as the European Proximity Operation Simulator (EPOS).[8] However, the dependence on serial manipulators has disadvantages such as low accuracy and slow response to control commands. To summarize, the systems discussed above do not offer a universal solution in terms of motion capability, accuracy, and cost-effectiveness. Furthermore, the current state-of-the-art doesn't consider a design based on the hybrid combination of a Stewart Platform (SP) and a robotic arm to represent the 6 Degrees-Of-Freedom (DOF) motion of a spacecraft. For this purpose, the authors designed a 9 DOF spacecraft simulator that incorporates a 3 DOF robotic arm mounted on a 6 DOF SP.[9] Numerical simulations were conducted by applying the Closed-Loop Inverse Kinematics (CLIK) control technique on the dynamic model of the platform for generating spacecraft motion on scaled-down orbital trajectories[9,10].

To extend the previous research, this paper focuses on the construction of the HIL simulator an optical table below the simulator). This is important to ensure that the proposed platform is safe and avoids maneuvers dangerous for the surrounding environment or the simulator itself.

To test and highlight the capabilities of the platform, a docking scenario is considered as shown in Figure 1. As shown, a satellite mock-up is attached to the end-effector acting as the simulated satellite/spacecraft. Although the simulator is designed to have a vision system installed on the mock-up for a realistic simulation scenario, this work considers a cooperative target satellite whose data is shared directly using a computer to the client satellite. This is because the primary aim of this study is to design, test, and showcase the capabilities of the proposed simulator in simulating space missions. In the future, an onboard system of vision and inertial sensors will be installed to simulate a real satellite/spacecraft. In the docking scenario, one simulator acts as the target satellite that is spinning with some slow angular velocity about the docking axis, and another simulator acts as the docking satellite and docks with the target satellite by moving translationally and aligning with it. To simulate the target and the docking satellites, individual inverse kinematic controllers are developed using the ROS. Since the target location is available, the docking satellite can maneuver towards it using a closed-loop trajectory controller. As a result, the experimental study highlights the HIL simulator's capabilities including the ability to successfully simulate ISAM missions and develop and test other control algorithms on the hardware platform.

SYSTEM DESIGN

Architecture

Figure 2 represents the overall system architecture that highlights the different stages of developing the proposed platform. The first stage is the problem formulation stage which covers the analytical study and dynamic modeling of the proposed platform. This is followed by the design of the proposed controller methodology and a detailed definition of the required sub-tasks and the corresponding secondary null-space accelerations. Following this, a numerical simulation model is generated to validate the dynamics of the proposed platform and provide an accurate representation of the physical platform's behavior. Then the designed controller is validated in simulations using the numerical simulation model. Then, the physical platform is constructed and the designed controller is validated on the constructed platform. Based on an analysis of the numerical simulations, the control environment is set up for hardware control. The results obtained from this hardware control setup are then used to improve the controller design, thus forming a closed loop. This closed loop allows iterative improvement of the control laws based on detailed analyses of experimental studies.

Design

The proposed spacecraft simulator is designed as a hybrid robotic platform comprising the SP (a parallel manipulator) as the base and the robotic arm (a serial manipulator) as the end-effector. Such a combination is chosen to enhance the capabilities of the simulator compared to using just a parallel or serial platform.[8,11,12] Generally, parallel platforms suffer from disadvantages such as a limited range of translational motion whereas the serial linkages often suffer from complex inverse kinematics leading to inaccurate control of the end-effector's position. However, by combining the two platforms, one can achieve highly improved motion-generating capabilities. Some of these capabilities include a wider range of motion, high accuracy, fast response to control commands, and kinematic redundancy. Kinematic redundancy is one of the most important aspects of this design which offers the capability for the platform to reach a certain position and orientation in 3D space in multiple ways.[9] This is specifically advantageous for the proposed application, i.e., simulation of

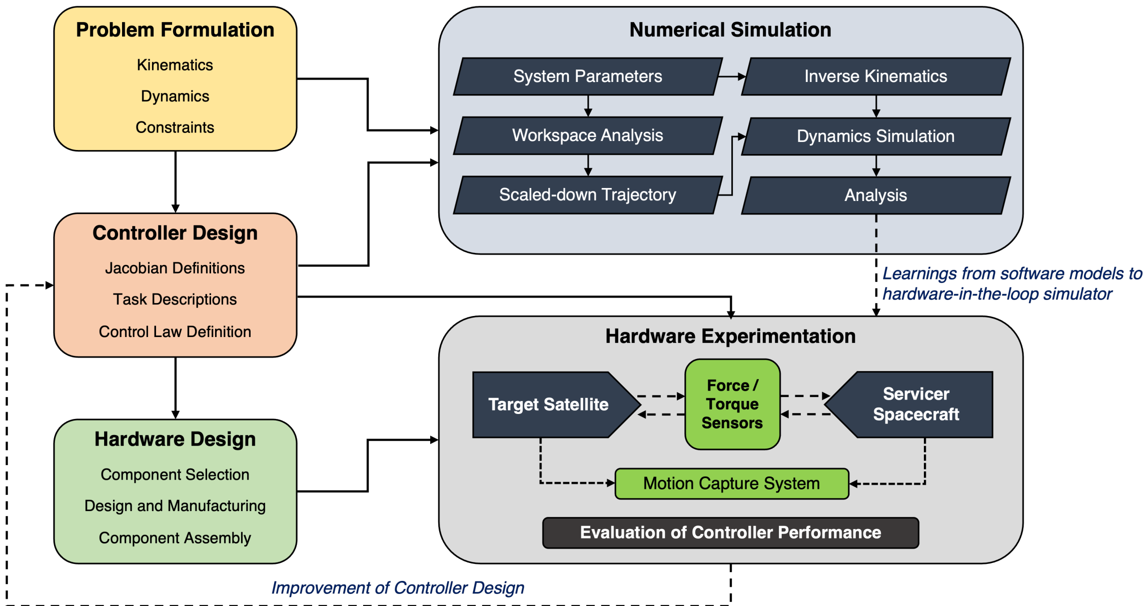


Figure 2: System Architecture.

space missions, since most of these missions require maneuvers that are difficult to simulate when the motion is limited or there exist only a few kinematic solutions to reach the desired manipulator pose.

The overall design framework of the proposed simulator is shown in Figure 3. The proposed simulator consists of a spacecraft as the end-effector of the robotic platform attached to sensors and markers. These sensors and markers allow for interaction with other spacecraft and measuring the states of the spacecraft itself and other spacecraft around it. Based on the desired simulation task (such as a docking mission or an orbital trajectory generation), the proper spacecraft control law provides the appropriate Cartesian trajectory for the spacecraft. The spacecraft Cartesian trajectory is then translated to the control and trajectory planning for individual robots based on the desired simulation task. This leads to a high-level control algorithm applied to the spacecraft (defined as the end-effector) and a low-level control algorithm applied to the spacecraft simulator. It must be noted that only the simulator construction (the SP and the robotic arm) and testing are included within the scope of this paper. The inclusion of sensors within the construction and development of spacecraft control mechanisms will be done as part of future studies.

Construction

To build the simulator, commercial off-the-shelf components are used as shown in Figure 4. The robotic arm selected for this research is the WidowX 250 Mobile 6 DOF robotic arm from Trossen Robotics.[13] The reason to select this arm is that it offers a convenient control environment development using Robot Operating System (ROS).[14] Similarly, the selected SP is the Physik Instrumente H.820 Hexapod[15] which also offers ROS-based control via common programming languages such as C++ and Python. Furthermore, the chosen SP offers high-accuracy motion that is needed for space mission emulation. This offers the capability to control the SP and the robotic arm simultaneously.

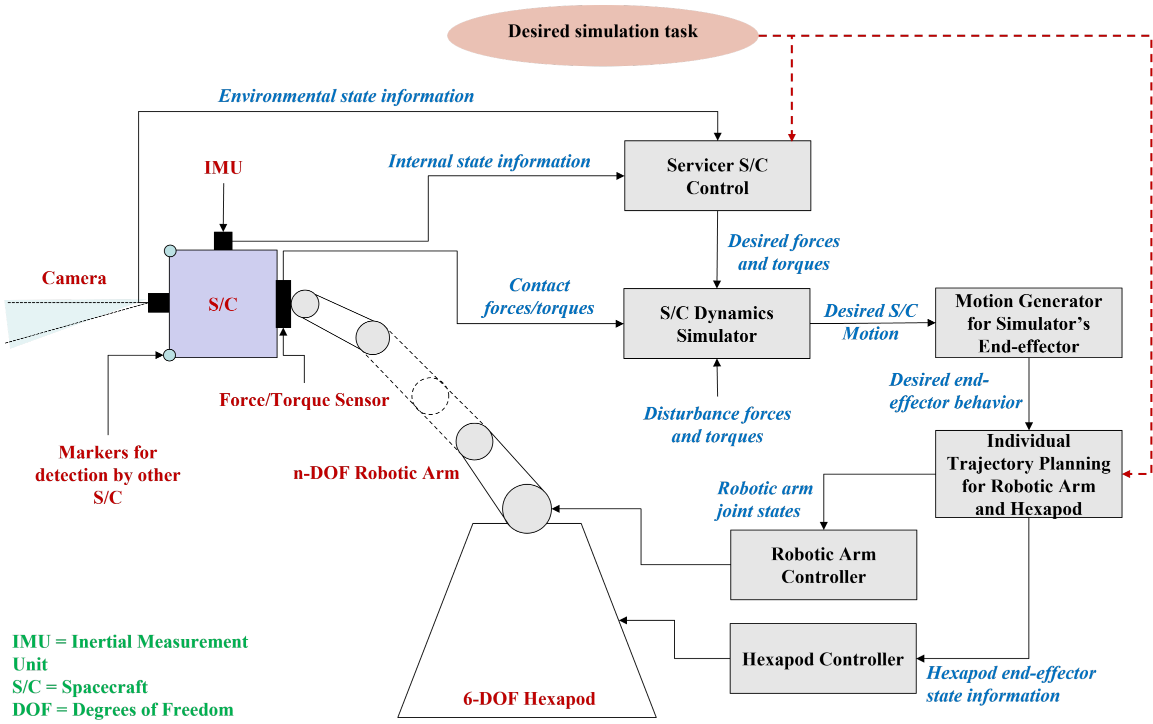


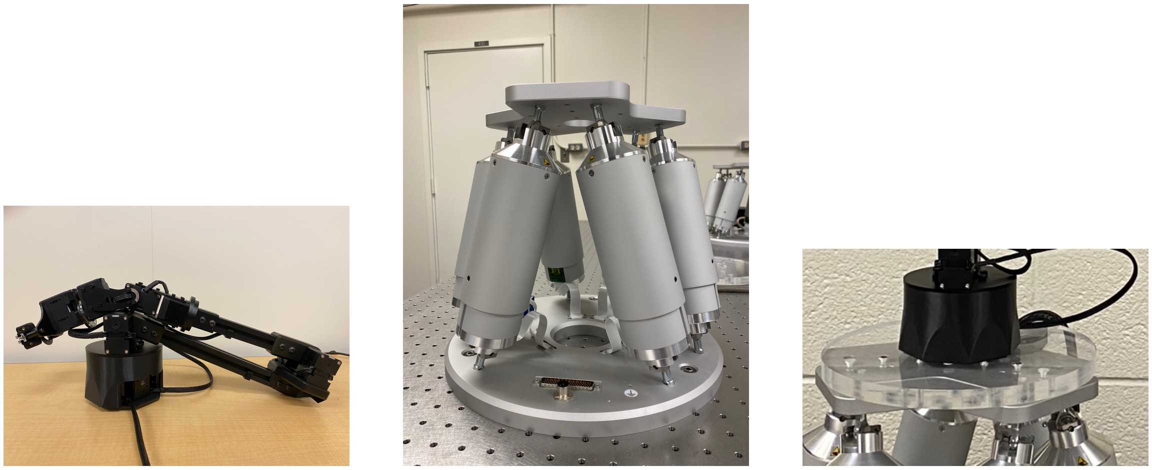
Figure 3: Design Framework.

To construct the platform, a connector plate is used to attach the robotic arm to the SP. The connector plate is made using a 0.7 in thick acrylic sheet. To make sure the robotic arm is attached in a safe and rigid manner, it contains counter-bored holes on each side of the plate for the SP and the robotic arm that allows for a seamless installation.

Based on the individual platform dimensions and specifications, the overall specifications of the constructed spacecraft simulator can be defined as shown in Table 2. Here, it must be noted that due to the redundancy of the constructed platform, the rotational travel ranges vary based on the location of the spacecraft in the simulator's workspace.

Table 1: Joint Ranges of the Proposed Simulator.

Robot	Joint	Unit	Value
SP	(x, y, z)	mm	$(\pm 50, \pm 50, \pm 25)$
	$(\theta_x, \theta_y, \theta_z)$	deg	$(\pm 15, \pm 15, \pm 30)$
Robotic Arm	Waist	deg	(-180, 180)
	Shoulder	deg	(-108, 114)
	Elbow	deg	(-123, 92)
	Forearm Roll	deg	(-100, 123)
	Wrist Angle	deg	(-180, 180)
	Wrist Rotate	deg	(-180, 180)



(a) Robotic Arm (b) Stewart Platform (c) Connector Plate

Figure 4: Components used for Platform Construction.

Table 2: Specifications of the Proposed Simulator.

Parameter	Unit	Value
Overall weight	kg	17
Travel range in X	mm	±700
Travel range in Y	mm	±700
Travel range in Z	mm	±700
Resolution	mm	0.1

EXPERIMENTAL STUDY

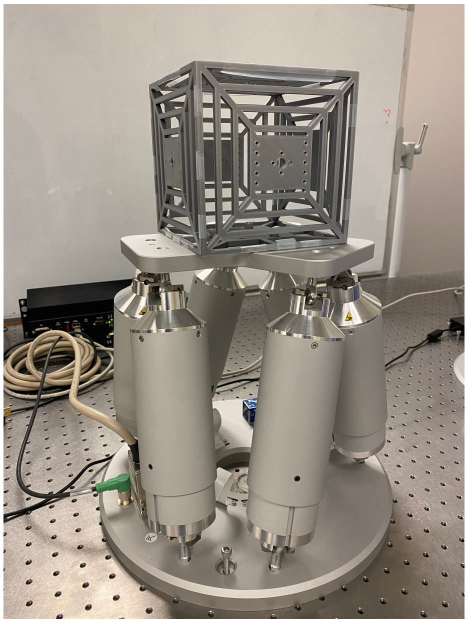
Scenario Description

To demonstrate the working platform and its capability to simulate an in-space mission, a rendezvous and docking scenario is chosen as an experimental scenario. A spinning target satellite is emulated using a SP due to its 6 DOF motion capability. On the other hand, to demonstrate the motion of the client satellite, the proposed 12 DOF spacecraft simulator platform is utilized.

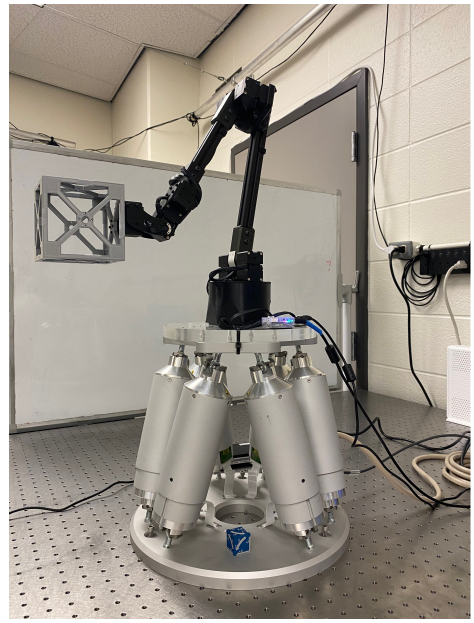
The two platforms and satellites can be seen in Figure 5. To represent realistic relative sizes, the target satellite is chosen to be 1.5U size (150 mm cube), and the client satellite is chosen to be 1U size, i.e. 100 mm cube. Furthermore, the two simulators are placed around 575 mm apart. Since this experiment aims to test the basic capabilities of the platform, the trajectory information of the target satellite is known to the client satellite. Based on this, the client satellite aims to dock with the spinning target satellite during motion by generating a Cartesian path from the starting position towards the target satellite.

Software Description

The software for the spacecraft simulators is developed in Python using the ROS programming interface. This involves the use of generating the ROS model of the combined platform using the individual ROS libraries available for the SP[16] and the robotic arm.[17] To operate the proposed



(a) Target satellite.



(b) Client satellite.

Figure 5: Spacecraft Simulators used in the Experiment Study.

simulator, the Unified Robotics Description Format (URDF) file is created for the robotic platform. Then, the ROS MoveIt[18] package is used to create a motion planning environment for the simulators.

The MoveIt package allows for using several open-source motion planning algorithms through the Open Motion Planning Library (OMPL) to control various robotic systems.[19] Using this package, an appropriate path-planning algorithm is selected to calculate the required manipulator poses for the platform based on the desired task. These poses are then converted to the individual joint state vectors using the inverse kinematics solver which are then sent to the corresponding trajectory controllers. The path planning algorithm used for this study is the Optimal Rapidly-exploring Random Tree (RRT) or RRT* as it allows for the optimal trajectory generation for the client satellite toward the target satellite based on the minimum path length criteria.[20,21] Furthermore, to solve the inverse kinematics, the Kinematics and Dynamics Library (KDL) is used as the primary inverse kinematics solver as it is able to resolve the 6 DOF task-space states into the 12 DOF joint-space state vector.[22]

Results

Once the software is developed in Python, the robots are connected to a PC which then runs the experiment. Figure 7 shows the experimental setup where the client satellite aims to dock with the spinning target satellite. Figure 8 shows the docking maneuver performed by the client satellite to dock with the target satellite. To validate the capability of the simulator to successfully perform the mission, the actual trajectory traversed by the client simulator is verified against the commanded trajectory generated by the planner. Figure 9 shows the comparison of the position trajectory in the three planes. It can be seen that the platform is able to successfully follow the planned trajectory toward the target satellite. Furthermore, it can be observed from Figure 10 that the orientation of the client satellite is also the same as the commanded orientation with small errors. These errors

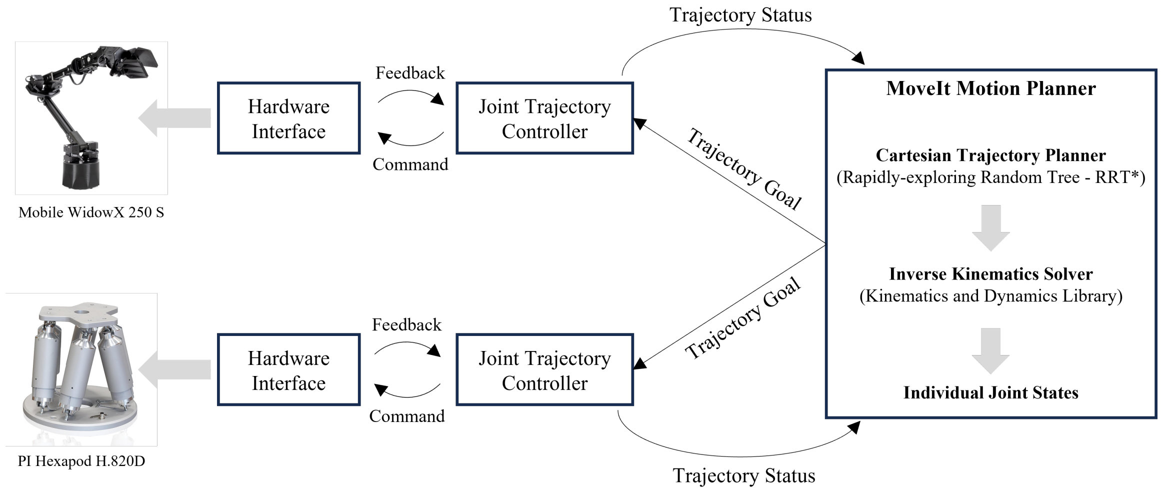


Figure 6: Software Framework.

can be attributed to the use of generic path planning and inverse kinematics algorithms that are not optimized as per hardware. Furthermore, the trajectory controllers used for the individual control of the robotic platforms only have target waypoint information.

Discussion and Future Work

The current experimental setup only focuses on highlighting the operability of the constructed spacecraft simulator to perform a basic docking maneuver. This achievement is accomplished using out-of-the-box open-source planners and controllers available in ROS. It can be seen that the simulator is able to track the planned trajectory and dock successfully with the target satellite. However, in the future, an adaptive controller will be designed that adaptively controls each actuator based on the situation. From a hardware perspective, a vision camera will be attached to the client satellite, enabling it to track the position and velocity of the target satellite in real-time. This availability of real-time data can allow for more robust tracking of the target satellite leading to efficient maneuvers. A force/torque sensor will also be installed at the docking surface to measure any contact forces. This information will be used to improve the controller design iteratively. Also, inertial sensors can be used to obtain the orientation information of the satellite more accurately. Overall, while the proposed design requires many modifications and improvements, the basic robotic platform design already offers the capability to perform space missions without any optimization.

CONCLUSION

In this study, a 12 DOF spacecraft simulator was designed and constructed to simulate in-space missions such as docking maneuvers. The 12 DOF simulator was constructed using a 6 DOF robotic arm placed on top of a Stewart platform (SP). The proposed design offers several advantages over conventional robots such as robotic arms. Furthermore, the kinematic redundancy offered by the simulator allows for greater flexibility and adaptability in simulating complex in-space missions. The study discusses the overall design framework of the platform followed by the construction process and lists the components used. Furthermore, the software design framework is described. To validate the constructed platform, a docking mission is experimentally validated using the simulator.

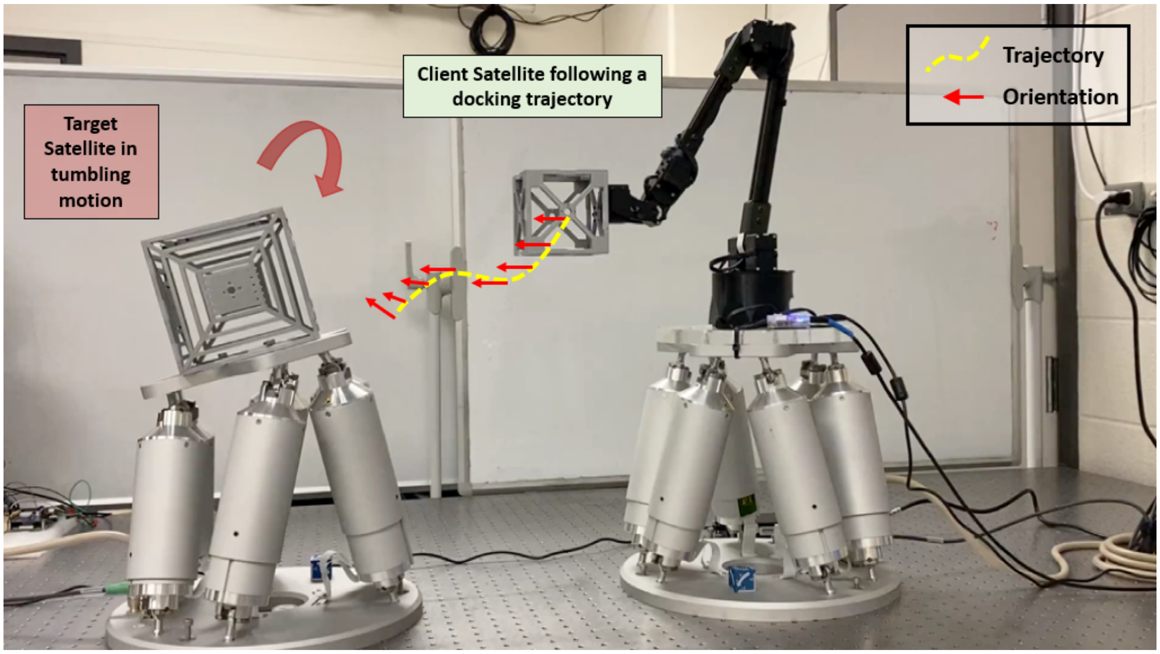


Figure 7: Experimental Setup.

From the results, it can be seen that the simulator is able to perform the required tasks using generic planning algorithms and controllers. In the future, the simulator will be further developed by adding additional components and thereby improving its capability to simulate complex space missions.

REFERENCES

[1] Goddard Space Flight Center, "On-Orbit Satellite Servicing Study," tech. rep., NASA, 2010.

[2] M. Wilde, C. Clark, and M. Romano, "Historical survey of kinematic and dynamic spacecraft simulators for laboratory experimentation of on-orbit proximity maneuvers," *Progress in Aerospace Sciences*, Vol. 110, 2019, p. 100552, https://doi.org/10.1016/j.paerosci.2019.100552.

[3] D. Miller, A. Saenz-Otero, J. Wertz, A. Chen, G. Berkowski, C. Brodel, S. Carlson, D. Carpenter, S. Chen, S. Cheng, D. Feller, S. Jackson, B. Pitts, F. Perez, J. Szuminski, and S. Sell, "SPHERES: a testbed for long duration satellite formation flying in micro-gravity conditions," *Advances in the Astronautical Sciences*, Vol. 105, Jan. 2000.

[4] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, E. Smith, and T. Team, "Astrobee: A New Platform for Free-Flying Robotics Research on the International Space Station," *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Jun. 2016.

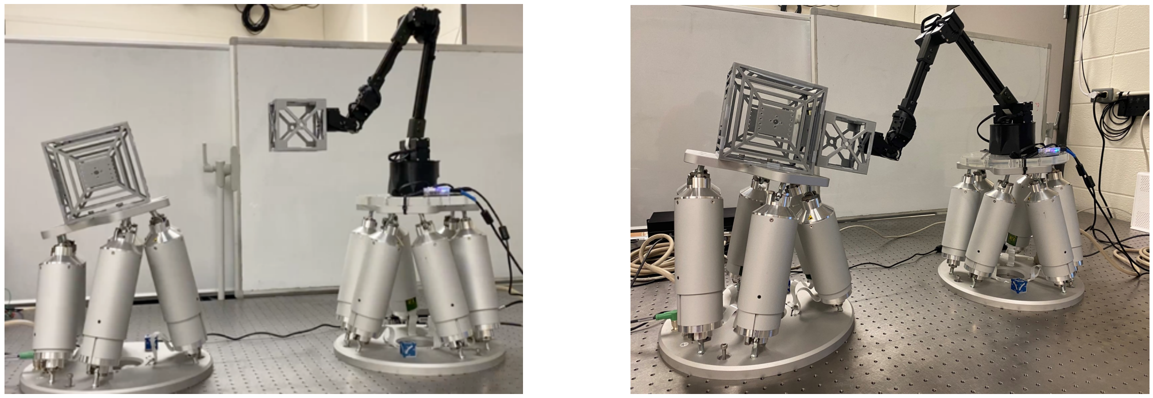
[5] C. R. Carignan, J. C. Lane, and P. J. Churchill, "Controlling robots on-orbit," *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No.01EX515)*, 2001, pp. 314–319, 10.1109/CIRA.2001.1013218.

[6] Y. Nakka, R. Foust, E. Lupu, D. Elliott, I. Crowell, S.-J. Chung, and F. Hadaegh, "Six Degree-of-Freedom Spacecraft Dynamics Simulator for Formation Control Research," *2018 AAS/AIAA Astrodynamics Specialist Conference*, 08 2018.

[7] B. Dunbar, "JSC Engineering - Active Response Gravity Offload System,"

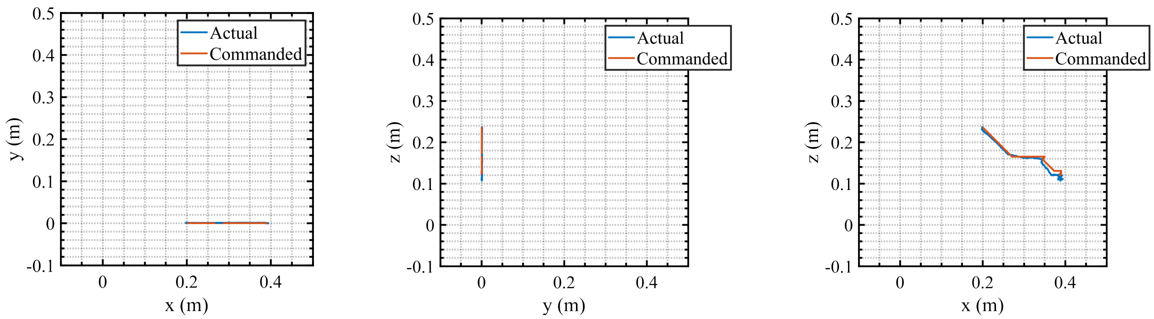
[8] T. Boge, T. Wimmer, O. Ma, and T. Tzschichholz, "EPOS - Using Robotics for RvD Simulation of On-Orbit Servicing Missions," *AIAA Modeling and Simulation Technologies Conference*, 2010, 10.2514/6.2010-7788.

[9] A. Chhabra and D. Kim, "Ground Robotic Platform for Simulation of On-Orbit Servicing Missions," *Journal of Aerospace Information Systems*, Vol. 19, No. 7, 2022, pp. 480–493, 10.2514/1.I011008.



(a) Simulation Start. (b) Simulation End.

Figure 8: Spacecraft Simulator performs the Docking Maneuver.



(a) X-Y. (b) Y-Z. (c) X-Z.

Figure 9: Commanded vs Actual Trajectory - Position.

[10] A. Chhabra and D. Kim, "A Unique Ground Simulator for On-Orbit Service Mission Emulation," *Advances in the Astronautical Sciences AAS/AIAA Astrodynamics 2022*, Vol. 177, August 2022, pp. 1–20. AAS 21-522.

[11] S. Kumar, H. Wöhrle, J. De Gea Fernández, A. Müller, and F. Kirchner, "A survey on modularity and distributivity in series-parallel hybrid robots," *Mechatronics*, Vol. 68, 2020, p. 102367, 10.1016/j.mechatronics.2020.102367.

[12] Johnson Space Center, "Engineering Test Facilities Guide," tech. rep., NASA, 2011.

[13] Trossen Robotics, "WidowX 250 Mobile Robot Arm 6DOF," Accessed: 2023-04-06.

[14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, may 2009.

[15] D. Stewart, "A Platform with Six Degrees of Freedom," *Proceedings of the Institution of Mechanical Engineers*, Vol. 180, No. 1, 1965, pp. 371–386, 10.1243/PIME_PROC_1965_180_029_02.

[16] PI-PhysikInstrumente, "Pi-physikinstrumente/pi_ros_driver: ROS driver for pi-hexapods,"

[17] Interbotix, "Interbotix/interbotix_ros_manipulators: Ros Packages for Interbotix Arms,"

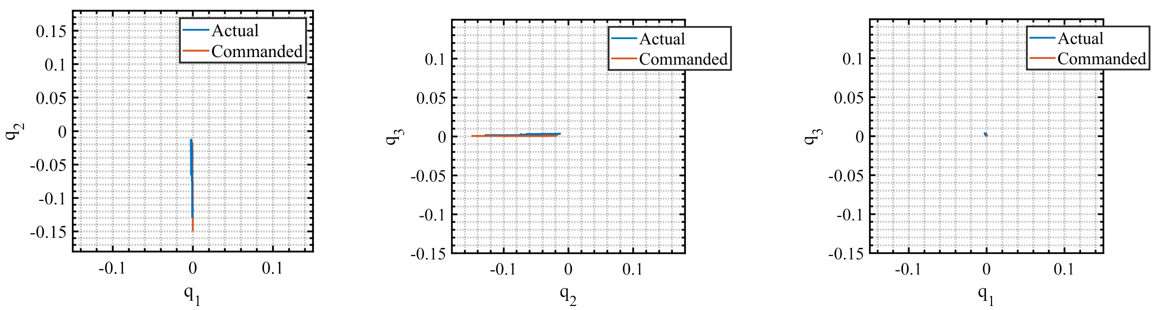
[18] D. T. Coleman, I. A. Sucan, S. Chitta, and N. Correll, "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study," 2014, 10.6092/JOSER_2014_05_01_P3.

[19] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, Vol. 19, December 2012, pp. 72–82, 10.1109/MRA.2012.2205651.

[20] M. M. Ioan A. Ṣucan, "Geometric::RRTstar class reference,"

[21] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," 2011.

[22] Open Robot Control Software, "Kinematics and Dynamics Library,"



(a) $q_1 - q_2$.

(b) $q_2 - q_3$.

(c) $q_1 - q_3$.

Figure 10: Commanded vs Actual Trajectory - Orientation.