



COBEM
2021 Florianópolis - Brasil



26th ABCM International Congress of Mechanical Engineering
November 22-26, 2021. Florianópolis, SC, Brazil

COB-2021-0620

DEVELOPMENT OF A HIGH FIDELITY SPACE ROBOT GAZEBO-BASED SIMULATOR FOR CLOSE-PROXIMITY OPERATIONS EXPERIMENTS

Luiz Manoel Santos Santana

Space Robotics Laboratory (LRE), Aerospace Engineering Division - Aeronautics Institute of Technology, Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, 12228-900, São José dos Campos - SP, Brazil
e-mail: luizmss@ita.br

Marcos R. O. A. Maximo

Autonomous Computational Systems Laboratory (LAB-SCA), Computer Science Division - Aeronautics Institute of Technology, Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, 12228-900, São José dos Campos - SP, Brazil
e-mail: mmaximo@ita.br

Luiz Carlos Sandoval Góes

Aeronautics Systems Laboratory (LSA), Mechanical Engineering Division - Aeronautics Institute of Technology, Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, 12228-900, São José dos Campos - SP, Brazil
e-mail: goes@ita.br

Ijar Milagre da Fonseca

Space Robotics Laboratory (LRE), Aerospace Engineering Division - Aeronautics Institute of Technology, Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, 12228-900, São José dos Campos - SP, Brazil
e-mail: ijar@ita.br

Abstract. *There is an increasing interest in robotic manipulators for orbital applications. This kind of unmanned robotic spacecraft can be used for maintenance and removal of satellites in orbit, for docking, for berthing, etc. Researchers often do experiments of space proximal operations to validate their control and navigation algorithms. In this context, a modern development process dictates that, before doing their experiments, they need to create a high fidelity simulator of the robotic system. A high-fidelity simulator reduces the gap between the real and virtual world. Researchers may execute different tests with the virtual robot without running the risk of damaging equipment. A fidelity simulator allows a good control system gains tuning. The robot belonging to the Space Robotics Laboratory (LRE) of Aeronautics Institute of Technology (ITA) consists of a manipulator's arm with five degrees of freedom coupled to a free-float base. The movement of the robotic arm is provided by servomotors, whereas the movement of the base is provided by a compressed air propulsion system using six nozzles. The Space Robots Laboratory has the equipment to do experiments with space robot manipulators but has not yet a reliable space robot simulation model. This work contributes by developing a space robot simulation model using the Gazebo simulator. The model's interface uses the Robot Operating System (ROS) framework, which is largely used by the robotics community. Robot parts were modeled in an CAD software using the real dimensions. Mass properties were previously obtained through a balancing table. The physical parameters of the servomotors were obtained by system identification techniques. After implementation, this simulator can be used for virtual twin, training reinforcement learning algorithms or testing control laws and navigation algorithms applied in space robotics experiments.*

Keywords: *space robotics, simulator, gazebo, robot operating system, robotics.*

1. INTRODUCTION

Space robots with manipulators for orbital applications are a growing field of research due to their importance in the space environment. There are many applications for this kind of spacecraft, from satellite maintenance (Beyer *et al.*, 2018) to space debris removal (Reiner *et al.*, 2017; Nishida and Kawamoto, 2011), passing through berthing (Fonseca *et al.*, 2017), dispose malfunctioned satellites (Reintsema *et al.*, 2011), on-orbit refueling operations (Liu *et al.*, 2020), and removal of spent launchers' stages (Felicetti *et al.*, 2016).

Researchers usually use free-floating base robots to do their ground-based testbeds of space systems (Wilde *et al.*,

2016; Zappulla *et al.*, 2017; Bevilacqua *et al.*, 2011; Tsiotras, 2014; Zappulla *et al.*, 2016). But, before starting an experiment of guidance, navigation, or control with a complex system such as a free-floating base with a robotic manipulator, it is a requirement to develop a simulator that faithfully represents the system under analysis. Using a simulator, researchers prevent accidents and equipment damage. Gazebo (Koenig and Howard, 2004) is a heavily used simulator for robotic systems.

Gazebo is a physics simulation engine for complex multi-robot simulations (Joseph and Cacace, 2018). It is designed to accurately reproduce the dynamic environment around the robot. All objects in the environment have mass, velocity, friction, and other physical parameters that make the robot's behavior realistic. A robot in Gazebo is made by rigid bodies connected via joints. Forces and torques can be applied to joints and surfaces to generate motion and interact with others elements of the simulation (Koenig and Howard, 2004). Gazebo also has a good interface with the Robot Operating System (ROS) (Quigley *et al.*, 2009). ROS is a robotic middleware that helps developers to program robotic applications, providing various tools and libraries to write robotic software (Joseph and Cacace, 2018). Gazebo and ROS are already used to program and simulate complex space robots, such as NASA's robots Robonaut 2 (Badger *et al.*, 2016), Astrobeer (Smith *et al.*, 2016; Bualat *et al.*, 2018), R5 (Hambuchen *et al.*, 2017), Valkyrie (Wonsick *et al.*, 2019), and the new lunar rover VIPER (Stukes *et al.*, 2021).

Our contribution in this paper is to develop a simulator for the free-floating space robot of the Space Robotics Laboratory (LRE), using Gazebo and programming the robot with ROS. We will implement all necessary parameters for a high fidelity simulation, as servomotor dynamics, sensors, and mass properties.

The remainder of this paper is organized as follows. Section 2. describes the main components of the robotic system, Section 3. explains the implementation of the space robot in Gazebo, Section 4. presents and discusses the results, finally, Section 5. concludes the paper.

2. SPACE ROBOT OVERVIEW

This section has the objective of introducing the main elements of the robot used in this work. Fig. 1 describes the main components of the space robot. The robotic arm (I) is a five degrees of freedom robotic manipulator that uses six MG995 servomotors to provides its motion. The arm can be used for many tasks, such as grasping objects, aid in docking, and berthing. The air bearings (II) reduce the friction between the robot and the surface where the robot will actuate, creating an effect of a free-floating base robot. In the case of LRE, we use a glass table to do the experiments. The robot has six nozzles (III), three on the left and three on the right. The nozzles ejects compressed air to exert forces on the robot's base. The two-cylinder of compressed air (IV) provides air for the nozzles and the air bearings. The solenoid valves (V) control the movement of the robot's base from the escape of the compressed air through the nozzles. The pressure regulator valves (VI) controls the pressure of the air through the nozzles and air bearings. The electronic control board (VII) is the device that controls the electromechanical components of the robot.

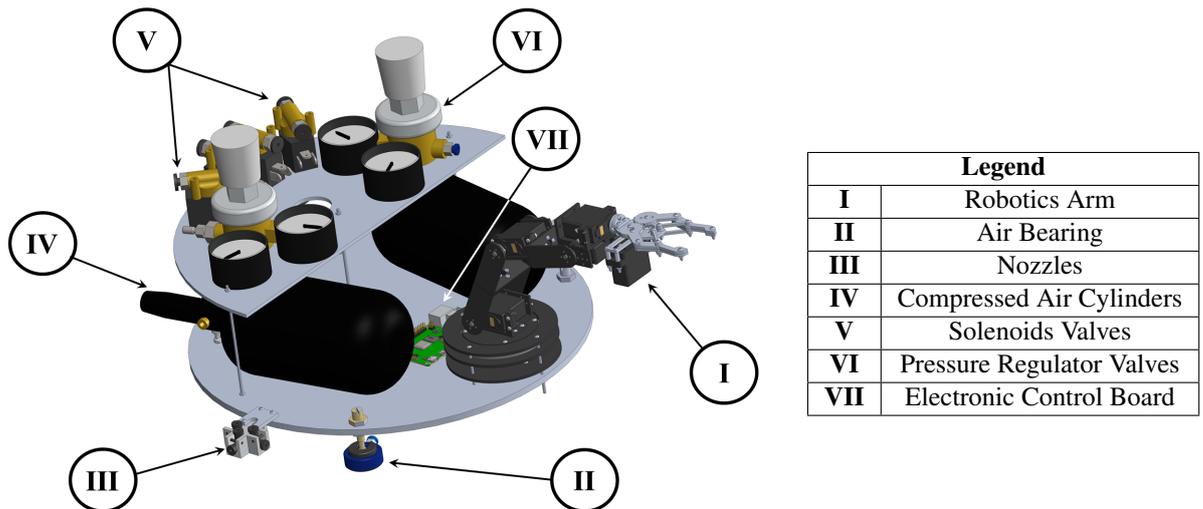


Figure 1. Robot Description.

3. SIMULATION

This section aims to detail the implementation of the space robot in the Gazebo environment. We divide it into the following: Subsection 3.1 presents the main elements of the simulation, Subsection 3.2 shows how to implement a servomotor's dynamics in Gazebo, Subsection 3.3 detail the modeling and configuration of the mechanical elements,

Subsection 3.4 describes the operation of the reaction propulsion system, and Subsection 3.5 explains the computer vision algorithm based on ArUco markers for pose detection.

3.1 Simulation Overview

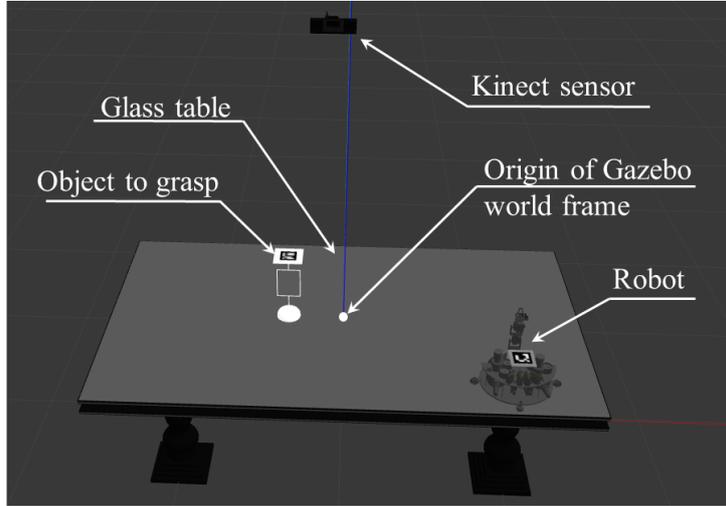


Figure 2. Gazebo simulation.

The simulation aims to get as close as possible to the LRE environment. The main elements of the experiment are: the space robot, a Kinect V2 sensor (Fankhauser *et al.*, 2015; Santana *et al.*, 2016), a glass table, and an object that serves as a target.

3.2 Servomotor

The Gazebo simulator does not have a way to implement an electric servomotor model. So, this impedes us from modeling a back electromotive force. But, Gazebo provides a revolute joint with viscous damping to simulate motors. In terms of dynamical system behavior, the back electromotive effect can be equivalently included by configuring the joint with the following damping (Maximo, 2017) :

$$b = b_m + \frac{\eta N^2 K_t^2}{R}, \quad (1)$$

where b_m is the motor's viscous friction, η is the gearbox efficiency, N is the gear ratio, K_t is the motor's torque constant, and R is the motor's resistance.

Another important behavior characteristic of an electric servomotor is the voltage saturation V_s . How Gazebo permit to configure the motor's torque, the voltage saturation can be modeled as an equivalent torque saturation τ_s , which cannot exceed the stall torque τ_{stall} of the servomotor (Maximo, 2017)

$$-\tau_{stall} \leq \tau_s \leq \tau_{stall} = V_s \frac{\eta N K_t}{R}. \quad (2)$$

The MG995 parameters were estimated using system identifications methods, based on ideas of Okuyama *et al.* (2015); Maximo *et al.* (2017), but adapting the procedures to account for the particularities of this servomotor, especially due to the fact that the datasheet of the MG995 lacks many important parameters. In this simulation we use $b = 0.8545 \text{ Nms/rad}$ and $\tau_s = 2.63 \text{ Nm}$.

Servomotors with large reduction gear ratio changes, in a considerable way, the inertia by the term $J_m \eta N^2$, and the motor inertia effect can not be neglected (Sciavicco *et al.*, 1995; Chen, 1989). The MG995 servomotor has this problem because has a large gear ratio. Unfortunately, to the best of our knowledge, there is no way to simulate this effect directly in Gazebo.

An approximation to model the motor inertia effect is to add an inertia of $J_m \eta N^2$ in the corresponding rotating axis to the body part the rotor (i.e. the rotating part of the motor) was attached to (Kajita *et al.*, 2014; Maximo, 2017).

For more details about modeling and servomotor implementation in Gazebo, we recommend reading Maximo (2017).

3.3 Mechanical modeling

We use CAD software to model all parts and equipment of the robotic system using real measurements. Gazebo is not compatible with part files generated by Solidworks, so we use Blender to convert the Solidworks files to Collada files.

All configurations of the robot were written in Xacro files. In these Xacro files, we configure the joint positions, sensors, Gazebo plugins, and ROS controllers (Chitta *et al.*, 2017).

We add odometry sensors to the base and gripper, as also collision sensors to the arm links. The physical device does not have these sensors, but this data is a requirement for training reinforcement learning algorithms (Sutton and Barto, 2018; Ibarz *et al.*, 2021). These sensors can also be important for collision avoidance simulations.

For an accurate simulation, the Gazebo simulator requires the implementation of mass, inertia matrix, and the position of the center of mass for each link, *i. e.*, robot's base and each link of the arm. These mass data were obtained previously by Unfried (2017). In our case, we change the inertia tensor of each link by adding the term $J_m \eta N^2$ to simulate the motor inertia effect.

3.4 Propulsion System

As mentioned in the previous section, the solenoid valves control the robot's base motion. If a nozzle is on, a force perpendicular to the exit of the nozzle is applied.

In this simulation, we set the force of each nozzle to 2 N, but we can change this value because the system has a pressure regulator. The cylinders used in this robot provide a maximum pressure of 3000 psi.

Table 1 describes the logic to turn on the nozzles for the desired action. Fig. 3 shows the nomenclature for each actuator.

Table 1. Nozzles logic.

Action	Nozzles ON
Go Forward	N4 and N3
Go Backward	N1 and N6
Rotate Left	N3 and N6
Rotate Right	N4 and N1
Go Left	N2
Go Right	N5

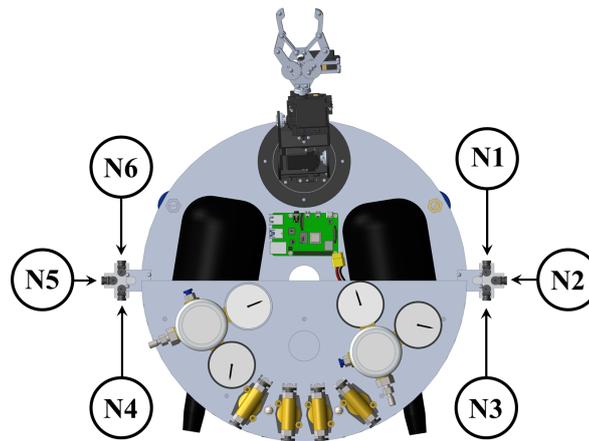


Figure 3. Reaction propulsion system.

3.5 Pose Detection using ArUco Markers

To identify the current robot position in the image provided by the RGB camera of the Kinect sensor, we attached an ArUco (Garrido-Jurado *et al.*, 2015; Romero-Ramirez *et al.*, 2018) fiducial marker on top of the robot. ArUco is a computer vision library for pose detection of fiducial markers (Fig. 4). This library is widely used for pose detection of mobile robots by its characteristics as robustness in marker detection, high speed, and accuracy (Romero-Ramirez *et al.*, 2018).

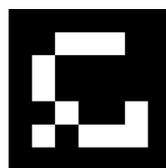


Figure 4. Example of a ArUco fiducial marker.

Gazebo simulator publishes a ROS node with the Kinect's image data. Before using the ArUco, we need to convert

the data generated by the Gazebo simulator to a format compatible with OpenCV. This can be done using the ROS library `cv_bridge`.

ArUco processing yields the position and the orientation (in Rodrigues vector) of the marker with respect to the camera frame. ROS and Gazebo work with quaternion. Therefore, we convert the Rodrigues vector to quaternion orientation. We can convert a Rodrigues vector \mathbf{b} to a quaternion \mathbf{q} using the following procedure:

Let the Rodrigues vector be given by

$$\mathbf{b} = [b_x \ b_y \ b_z] . \quad (3)$$

Then, the rotation angle defined by \mathbf{b} is

$$\alpha = \sqrt{b_x^2 + b_y^2 + b_z^2} , \quad (4)$$

so, the quaternion \mathbf{q} can be written as

$$\mathbf{q} = \begin{bmatrix} \frac{b_x}{\alpha} \sin\left(\frac{\alpha}{2}\right) \\ \frac{b_y}{\alpha} \sin\left(\frac{\alpha}{2}\right) \\ \frac{b_z}{\alpha} \sin\left(\frac{\alpha}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right) \end{bmatrix} . \quad (5)$$

Now, we have the position and orientation of the marker in the camera frame, to convert this pose to other reference frames, as the Gazebo world reference frame, we use the ROS library `tf`. These final data are published using a ROS topic.

4. RESULTS AND DISCUSSIONS

This section aims to demonstrate the robot's performance using a simulation in an environment that resembles with one found in the Space Robot's Laboratory. The objective of the space robot in this simulation is to use the nozzles to navigate through waypoints from an initial position to a desired position, move its arm joints to arrive at the end-effector's desired position, and grasp a target. All the results are expressed in the Gazebo world coordinate system. The simulation can be seen here: [Simulation](#).

The Navigation starts in the base's position $x = 1.00$ m and $y = -0.5$ m, so the robot goes to a waypoint located at the position $x = 1.00$ m and $y = 0.10$ m. After arriving at the waypoint, the robot starts to rotate the base from the current angle (approx. 90°) to 180° and go to the objective base's position (0.0, 0.0). The desired linear and angular velocity configured to the simulation are 0.05 m/s and 0.25 deg/s, respectively. Table 2 explicits the trajectory's order.

Table 2. Waypoints navigation.

Platform's Base Trajectory Plan				
Order	Description	Position x (m)	Position y (m)	Heading (deg)
1°	Start	1.00	-0.50	90.0
2°	Waypoint	1.00	0.10	90.0
3°	Rotation	-	-	180.0
4°	Objective	0.00	0.00	180.0

After arriving at the desired base's position, the robot's arm starts its movement to carry the gripper to the location where it will grasp the target. We can divide the arm movement into three distinct parts: open the gripper, move the end-effector to the objective and close the gripper. We use Moveit!(Coleman *et al.*, 2014) for the arm trajectory planning. Configuring the desired joint positions, Moveit! generates the code for the trajectory control, forward, and inverse kinematics. We also configure the velocity of the actuator to make the movement slowly, we use 10 percent of the servo's maximum speed.

Figure 5 shows the detected ArUco. The lines displayed on the detected markers represent their orientations. We can see that the computer vision algorithm is working as expected.

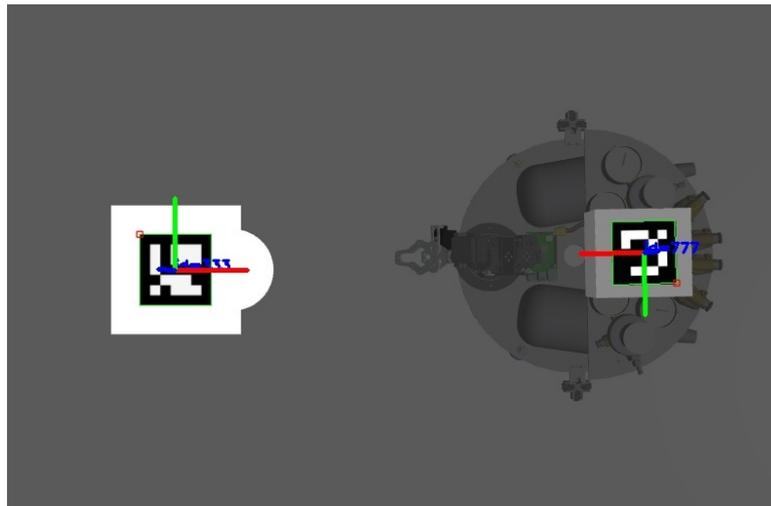


Figure 5. ArUco markers detected from the Kinect's image.

Analyzing the plots of Fig. 6, we can see that the robot passed by all the pre-determined positions. The orientation angle also has the expected behavior. There is considerable oscillation in orientation angle when the robot arrives at the base objective. This oscillation correlates with the beginning of the arm's movement, where the torques generated by the arm's servomotors disturb the base. However, this can easily be solved using a control law algorithm.

The base's velocities showed in Fig. 7 remained within the previously defined range.

Figure 8 illustrates the trajectory realized by the robot. It shows the positions of the waypoint, start, and objective. The robot goes through all points with no large displacement between the reference and actual positions. It is also easy to see the effect of the 90 degrees rotation motion. Even with the oscillations generated by the arm, the base remains in its position to grasp the target.

The plots of Fig. 9 show the movement of the arm's joints to arrive at the grasp position. We can see on the last, the time where the gripper was open and closed.

Finally, Fig. 10 illustrates the positions of the end-effector. Notice that the end-effector arrives at the location to grasp the target, *i.e.*, around $x = -0.2 \text{ m}$ and $y = 0.0 \text{ m}$.

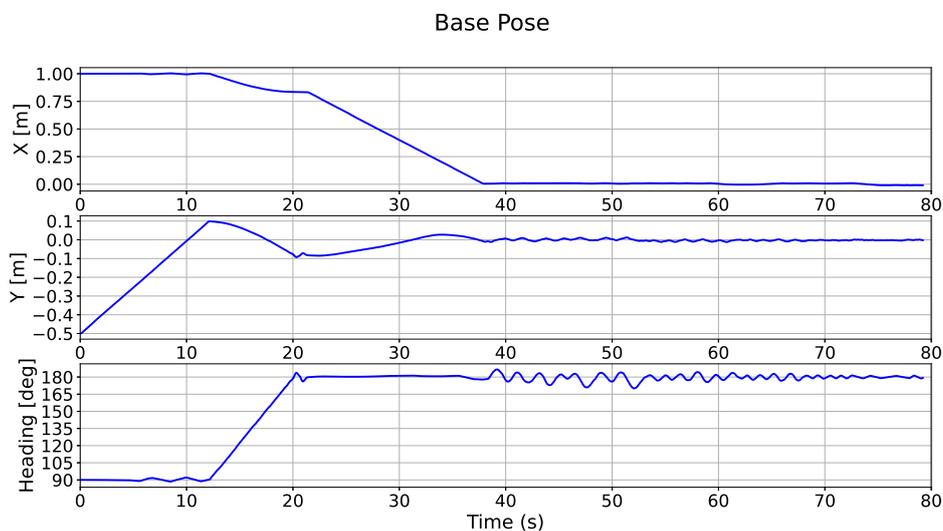


Figure 6. Space robot's base pose.

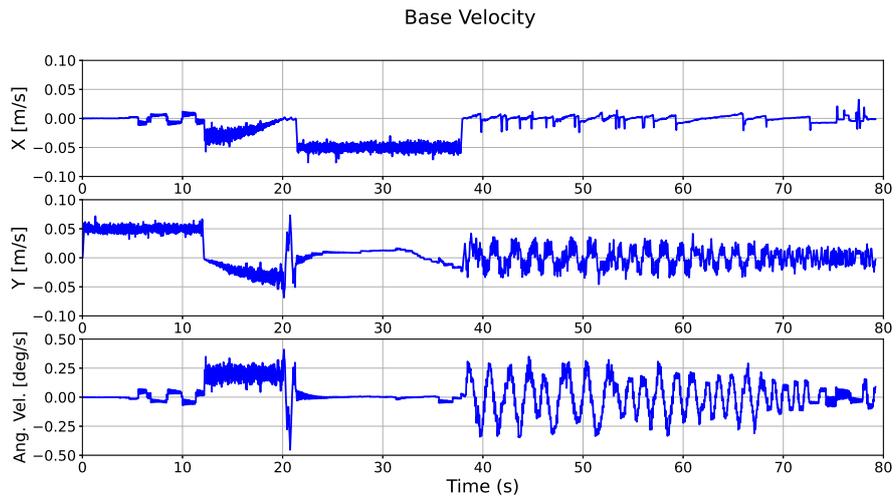


Figure 7. Space robot's base velocities.

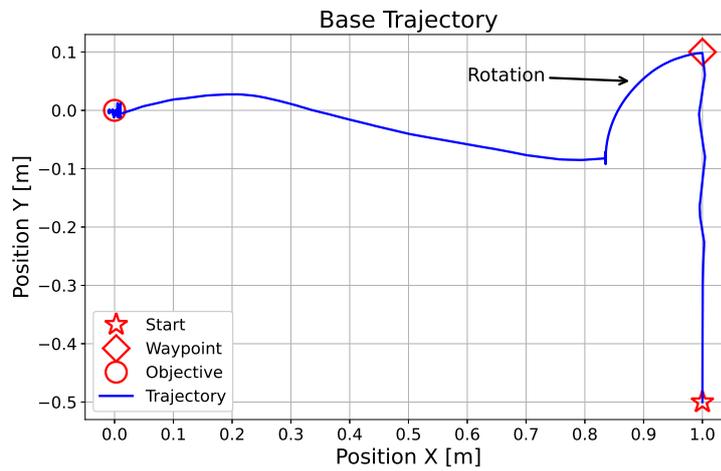


Figure 8. Robot's base trajectory .

Arm Joints Angle

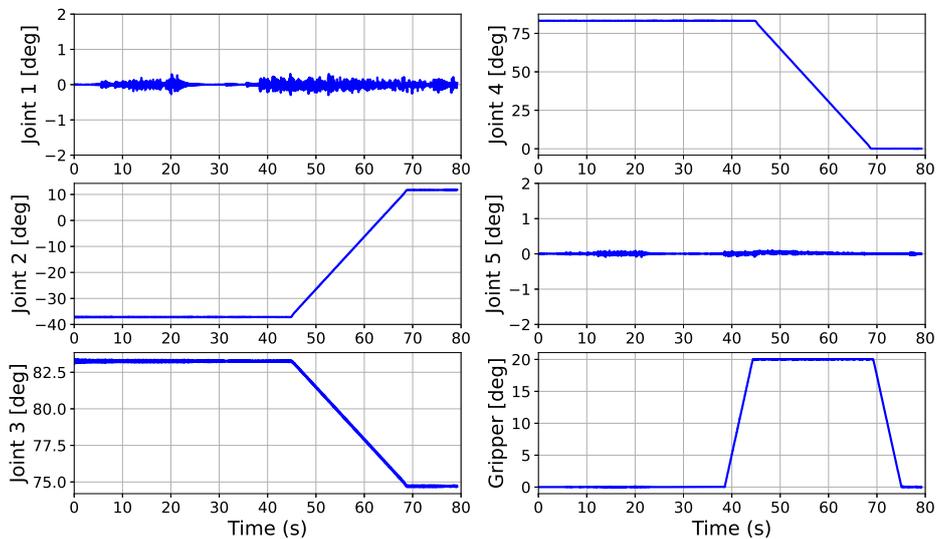


Figure 9. Movement of the arm's joints.

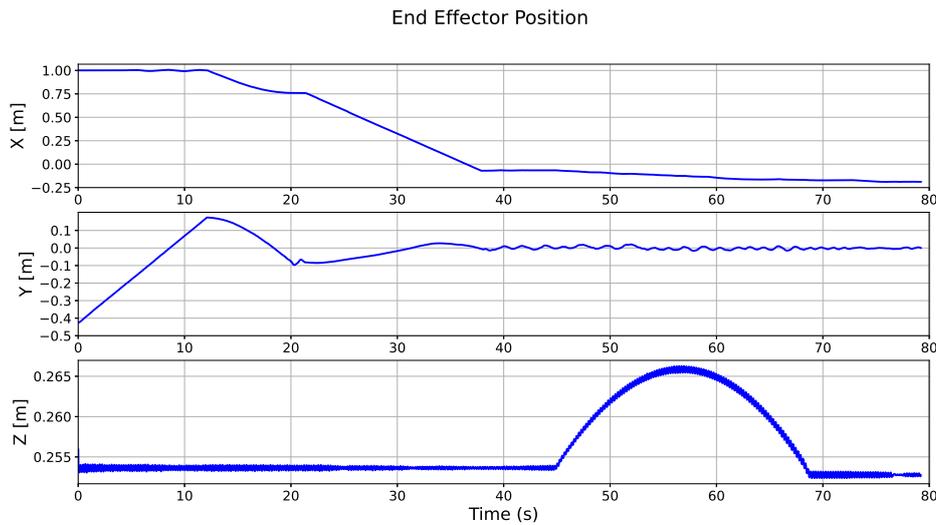


Figure 10. End-effector position.

5. CONCLUSION

Space robots with manipulators for orbital applications is a growing field of research. Many researchers do ground-based testbeds with free-floating base robots to validate their guidance, navigation and control algorithms. However, before test their algorithms on a physical device, it is a good practice to validate these new algorithms in a fidelity simulator of the system. In this paper, we developed a high-fidelity simulator for the free-floating base space robot of the Space Robotics Laboratory. We chose the Gazebo physics engine to be our simulator. Gazebo is largely used by roboticists to simulate complex robots and sensors. Gazebo also has a good interface with ROS, facilitating robot programming and simulation. For a good fidelity of the system, we implement an approximated dynamics of the servo motor MG995. All mass properties are included in the Gazebo model. We also implement a Kinect sensor to estimate the robot and target poses. From the results, the simulator proved to be a powerful tool for testing guidance, navigation and control algorithms. The robot managed to follow a predetermined trajectory and grabbed onto the target. For future works, we recommend trying other trajectories with more focus on docking or berthing simulations. We also intent to test control algorithms for the base and the arm, and use the simulator for training the robot as an intelligent agent with reinforcement learning algorithms.

6. ACKNOWLEDGEMENTS

The authors are grateful to the Brazilian Space Agency (AEB) for the physical device. The author Luiz M. S. Santana acknowledges the scholarship financial support of Coordination of Superior Level Staff Improvement (CAPES).

7. REFERENCES

- Badger, J., Gooding, D., Ensley, K., Hambuchen, K. and Thackston, A., 2016. "Ros in space: A case study on robonaut 2". In *Robot Operating System (ROS)*, Springer, pp. 343–373.
- Bevilacqua, R., Romano, M., Curti, F., Caprari, A.P. and Pellegrini, V., 2011. "Guidance navigation and control for autonomous multiple spacecraft assembly: analysis and experimentation". *International Journal of Aerospace Engineering*, Vol. 2011.
- Beyer, A., Grunwald, G., Heumos, M., Schedl, M., Bayer, R., Bertleff, W., Brunner, B., Burger, R., Butterfaß, J., Gruber, R. et al., 2018. "Caesar: Space robotics technology for assembly, maintenance, and repair". In *Proceedings of the International Astronautical Congress, IAC*.
- Bualat, M.G., Smith, T., Smith, E.E., Fong, T. and Wheeler, D., 2018. "Astrobee: A new tool for iss operations". In *2018 SpaceOps Conference*. p. 2517.
- Chen, J., 1989. "The effects of gear reduction on robot dynamics". In *Proc. NASA Conf. on Space Telerobotics*. Vol. 4, pp. 297–307.
- Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Rodríguez Tsouroukdissian, A., Bohren, J., Coleman, D., Magyar, B., Raiola, G., Lüdtke, M. and Fernández Perdomo, E., 2017. "ros_control: A generic and simple control framework for ros". *The Journal of Open Source Software*. doi:10.21105/joss.00456. URL <http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf>.
- Coleman, D., Sucan, I., Chitta, S. and Correll, N., 2014. "Reducing the barrier to entry of complex robotic software: A

- moveit”. *Case Study*. Available online: <https://arxiv.org/abs/1404.3785> (accessed on 25 June 2021).
- Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M. and Siegwart, R., 2015. “Kinect v2 for mobile robot navigation: Evaluation and modeling”. In *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, pp. 388–394.
- Felicetti, L., Gasbarri, P., Pisculli, A., Sabatini, M. and Palmerini, G.B., 2016. “Design of robotic manipulators for orbit removal of spent launchers’ stages”. *Acta Astronautica*, Vol. 119, pp. 118–130. ISSN 0094-5765. doi:<https://doi.org/10.1016/j.actaastro.2015.11.012>.
- Fonseca, I.M., Goes, L.C., Seito, N., da Silva Duarte, M.K. and de Oliveira, É.J., 2017. “Attitude dynamics and control of a spacecraft like a robotic manipulator when implementing on-orbit servicing”. *Acta Astronautica*, Vol. 137, pp. 490–497.
- Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. and Medina-Carnicer, R., 2015. “Generation of fiducial marker dictionaries using mixed integer linear programming”. *Pattern Recognition*, Vol. 51. doi:10.1016/j.patcog.2015.09.023.
- Hambuchen, K.A., Roman, M.C., Sivak, A., Herblet, A., Koenig, N., Newmyer, D. and Ambrose, R., 2017. “Nasa’s space robotics challenge: Advancing robotics for future exploration missions”. In *AIAA SPACE and Astronautics Forum and Exposition*. p. 5120.
- Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P. and Levine, S., 2021. “How to train your robot with deep reinforcement learning: lessons we have learned”. *The International Journal of Robotics Research*, Vol. 40, No. 4-5, pp. 698–721.
- Joseph, L. and Cacace, J., 2018. *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd.
- Kajita, S., Hirukawa, H., Harada, K. and Yokoi, K., 2014. *Introduction to humanoid robotics*, Vol. 101. Springer.
- Koenig, N. and Howard, A., 2004. “Design and use paradigms for gazebo, an open-source multi-robot simulator”. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3, pp. 2149–2154 vol.3. doi:10.1109/IROS.2004.1389727.
- Liu, J., Tong, Y., Liu, Y. and Liu, Y., 2020. “Development of a novel end-effector for an on-orbit robotic refueling mission”. *IEEE Access*, Vol. 8, pp. 17762–17778. doi:10.1109/ACCESS.2020.2964641.
- Maximo, M.R.O.A., 2017. *Automatic walking step duration through model predictive control*. thesis, Instituto Tecnológico de Aeronáutica.
- Maximo, M., Ribeiro, C.H. and Afonso, R.J., 2017. “Modeling of a position servo used in robotics applications”. *Proceedings of the 2017 Simpósio Brasileiro de Automação Inteligente (SBAI), Porto Alegre, SC, Brazil*.
- Nishida, S.I. and Kawamoto, S., 2011. “Strategy for capturing of a tumbling space debris”. *Acta Astronautica*, Vol. 68, No. 1-2, pp. 113–120.
- Okuyama, I.F., Maximo, M. and Pinto, S.C., 2015. “System identification of a hobby dc motor using a low cost acquisition setup”. *Simpósio Brasileiro de Automação Inteligente (SBAI)*.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y. et al., 2009. “Ros: an open-source robot operating system”. In *ICRA workshop on open source software*. Kobe, Japan, Vol. 3, p. 5.
- Reiner, M., Fernandez, J.G. and Ortega, G., 2017. “Combined control for active debris removal using a satellite equipped with a robot arm”. *Proceedings for the GNC 2017, 10th International ESA Conferece on Guidance, Navigation and Control Systems*.
- Reintsema, D., Sommer, B., Wolf, T., Theater, J., Radthke, A., Naumann, W., Rank, P. and Sommer, J., 2011. “Deos-the in-flight technology demonstration of german’s robotics approach to dispose malfunctioned satellites”. In *ESA 11th Symposium on Advanced Space Technologies in Robotics and Automation, ESTEC, Netherlands*.
- Romero-Ramirez, F., Muñoz-Salinas, R. and Medina-Carnicer, R., 2018. “Speeded up detection of squared fiducial markers”. *Image and Vision Computing*, Vol. 76. doi:10.1016/j.imavis.2018.05.004.
- Santana, L.M.S., dos Santos, D.A., Viana, Í.B. and Ballet, R., 2016. “A low-cost position and velocity estimation system for multicopter aerial vehicles using a static kinect sensor”. *Proceedings of the 2016 Congresso Brasileiro de Automática (CBA), Vitória, ES, Brazil*.
- Sciavicco, L., Siciliano, B. and Villani, L., 1995. “Lagrange and newton-euler dynamic modeling of a gear-driven robot manipulator with inclusion of motor inertia effects”. *Advanced robotics*, Vol. 10, No. 3, pp. 317–334.
- Smith, T., Barlow, J., Bualat, M., Fong, T., Provencher, C., Sanchez, H., Smith, E. et al., 2016. “Astrobee: A new platform for free-flying robotics on the international space station”. In *Int. Symp. on Artificial Intelligence, Robotics and Automation in Space*.
- Stukes, S., Allan, M., Bajjalieh, G., Deans, M., Fong, T., Hihn, J. and Utz, H., 2021. “An innovative approach to modeling viper rover software life cycle cost”. In *2021 IEEE Aerospace Conference (50100)*. pp. 1–16. doi:10.1109/AERO50100.2021.9438347.
- Sutton, R.S. and Barto, A.G., 2018. *Reinforcement learning: An introduction*. MIT press.
- Tsiotras, P., 2014. “Astros: A 5dof experimental facility for research in space proximity operations”. In *AAS Guidance*

and Control Conference. 2014-114.

- Unfried, L.M., 2017. *Dynamics Modeling and Attitude Control of a Robotic Manipulator Spacecraft*. dissertation, Instituto Tecnológico de Aeronautica.
- Wilde, M., Ciarcià, M., Grompone, A. and Romano, M., 2016. “Experimental characterization of inverse dynamics guidance in docking with a rotating target”. *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 6, pp. 1173–1187.
- Wonsick, M., Dimitrov, V. and Padir, T., 2019. “Analysis of the space robotics challenge tasks: From simulation to hardware implementation”. In *2019 IEEE Aerospace Conference*. IEEE, pp. 1–8.
- Zappulla, R., Virgili Llop, J., Park, H., Zagaris, C. and Romano, M., 2016. “Floating spacecraft simulator test bed for the experimental testing of autonomous guidance, navigation, & control of spacecraft proximity maneuvers and operations”. In *AIAA/AAS Astrodynamics Specialist Conference*. p. 5268.
- Zappulla, R., Virgili-Llop, J., Zagaris, C., Park, H. and Romano, M., 2017. “Dynamic air-bearing hardware-in-the-loop testbed to experimentally evaluate autonomous spacecraft proximity maneuvers”. *Journal of Spacecraft and Rockets*, Vol. 54, No. 4, pp. 825–839.

8. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.