

A ROS-Based Control Framework for Simulating Locomotion of a Multi-Arm Space Assembly Robot

Sairaj R Dillikar¹[0000–0002–8245–3243], Cameron Leslie²[0009–0009–0152–2664], Saurabh Upadhyay²[0000–0002–2030–1594], Leonard Felicetti²[0000–0001–6830–4308], and Gilbert Tang²[0000–0002–5497–7269]

¹ MSc Robotics (2022-23), Cranfield University, Cranfield, United Kingdom,
sairaj.dillikar.102@gmail.com

² School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Cranfield, United Kingdom, {Cameron.F.Leslie, Saurabh.Upadhyay, Leonard.Felicetti, g.tang}@cranfield.ac.uk

Abstract. This paper proposes a ROS-based control framework for simulating the locomotion of a multi-arm space robot on a planar space structure. This framework has applications in technology demonstrations of space structure assembly, construction, and maintenance where a multi-arm robot has to traverse on a space structure building blocks known as Spatial Reticular Structures (SRS) to perform the space operation. The framework sets the desired environment and SRS structures in the first step and devises two movement primitives to achieve locomotion on the structure. A ROS-Gazebo-based simulation architecture is presented to execute the proposed control framework. The viability and limitations of the proposed control framework are discussed with simulation results.

Keywords: On-orbit operations · Manipulation in space · Space robotic locomotion · Multi-arm robot · Multi-robot cooperation/collaboration · Robot Operating System (ROS)

1 Introduction

Recent space mission studies have considered multi-arm robots to carry out the assembly tasks in orbit, e.g., the RAMST [1], MOSAR [2], PULSAR [3]. In these missions, a multi-arm robot walks or climbs on top of a spacecraft carrying a modular structure, such as a telescope mirror tile, which will be assembled onto the main structure while in orbit. Here, the space robot's locomotion is assisted by a simple mechanical coupling between the robot's end-effector and the Standard Interface (SI) on the mirror tiles. The locomotion of multi-arm space robots on the built structure is a crucial challenge to overcome [4], and these space robots must rely on a reliable control and simulation framework that can perform the coordinated multi-arm control and simulate the locomotion.

A typical coordinated control strategy for multi-arm utilises the primary arm for the specific defined aim and the secondary arm providing assistance. One such scenario is capturing a moving object using coordinated control [5], where the method

involves two cases: a) capturing the target when the base is free-floating and b) single-arm capturing while keeping the base's inertial pose using another arm. A free-flying dual-arm space robot utilises a coordinated motion control strategy for assembling modular parts in space, demonstrated using an air-bearing simulator [6]. They introduced two phases to accomplish the assembly task, in which coordinated motion planning is performed based on an optimisation method. The first phase (pre-assembly phase) establishes the dual-arm robots' General Relative Jacobian Matrix (GRJM, [7]) to manipulate and reach relative positions and attitudes cooperatively. For the final phase, one arm is kept at rest with respect to its base, and the other arm moves in a specified trajectory to achieve the final assembly.

The aforementioned approaches inspired the basis for the space robot to perform coordinated locomotion in this work, as the primary manipulator advances to take the next step while the secondary manipulator provides assistance or forces in the opposite direction, displacing the entire robot. Furthermore, two locomotion strategies are introduced based on bio-inspired gait mechanisms. The lateral movement is similar to the inchworm mechanism [8], while the vertical movement is adapted from the biped mechanism due to their resemblance to humans [9].

The viability of the introduced locomotion strategies will be demonstrated using the ROS-Gazebo-based simulation. In terms of ideation & developing the simulation, some insights from recently developed ROS-based projects were obtained for the locomotion strategies of lateral movement (inchworm mechanism) [10] and vertical movement (biped mechanism) ([11], [12]). Moreover, a typical coordinated control demonstration utilising dual-arm configured Universal Robots (UR) manipulators can be seen from a ROS-based simulation developed by [13], which provides insights for the development of the controller setup & dual-arm configuration for multi-arm control.

Strong underlying fundamentals in multi-arm configured space robots and simulation architecture to achieve the on-orbit operations can be understood from [14], also a backbone of this research. Furthermore, the current work provides simulated validation, mainly focusing on addressing the critical challenges associated with achieving mobility and multi-arm coordination of the aforementioned project.

The paper's main contribution is to showcase the simulation of a multi-arm space robot performing locomotion on a planar space structure. Two locomotion strategies are introduced and devised for the space robot to traverse a planar space structure. Finally, a simulation based on a Robot Operating System (ROS) and Gazebo was developed to validate the locomotion strategies.

The remainder of the paper is structured as follows: Sec. 2 sets the background for this paper by discussing the configuration of the robot, the space structure, and the stages of implementation through the technical platforms. Sec. 3 explains the control architecture, discusses the basis of derived approaches for the required locomotion through multi-arm coordination, and shows the implementation carried out through the ROS-Gazebo-based simulation. Results of the developed project are discussed briefly in Sec. 4, addressing all the approaches. The paper is concluded by providing insights for the future scope in Sec. 5.

2 Problem Overview

The configuration of Multi-Arm Robot for In-Orbit Operations (MARIO) comprising of three manipulators positioned 120 degrees from each other's base (Torso, see Fig. 1a) (for more insights on robot's configuration refer [14]). Here, two robotic arms are necessary to locomote the MARIO across the structure, whilst the third arm carries a single Spatial Reticular Structures (SRS) module. The Standard Interfaces (SI) (see Fig. 1b) acts as a coupling mechanism between the robotic arm's end-effectors and the SRS modules.

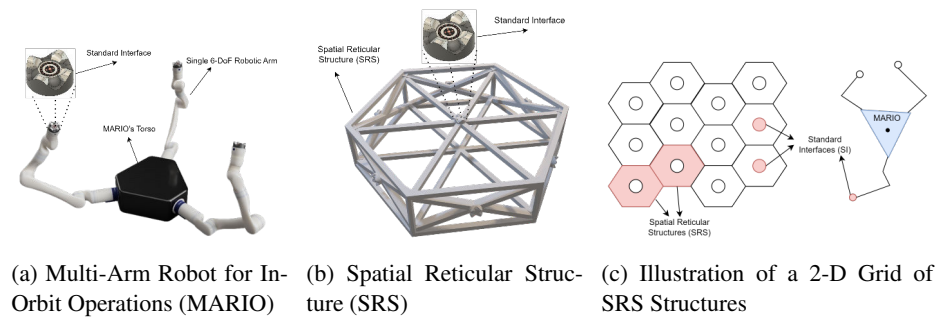


Fig. 1: Graphical Representation of MARIO and SRS Structures

With this, consider SRS structures in a 2-D grid plane as shown in Fig. 1c; each SRS and robotic arm has an SI unit attached, allowing the robot arm to be docked on an SRS. The coordinates of every SRS and the robot torso are available at all times from the simulator, along with considering the zero gravity constant for the simulation. The development of the simulation framework considers the following two sub-problems:

1. Determine and devise appropriate locomotion strategies for the multi-arm robot.
2. Develop a ROS-Gazebo-based simulation to validate the control framework to achieve locomotion of the space robot.

3 Implementation of Control Architecture Through the ROS

This section discusses the control architecture for locomotion and its execution in ROS. The pipeline for control architecture is illustrated in Fig. 2, and it comprises three major tasks:

1. Spawning of the SRS models
2. Obtaining the coordinates of SRS models
3. Executing the locomotion sequence

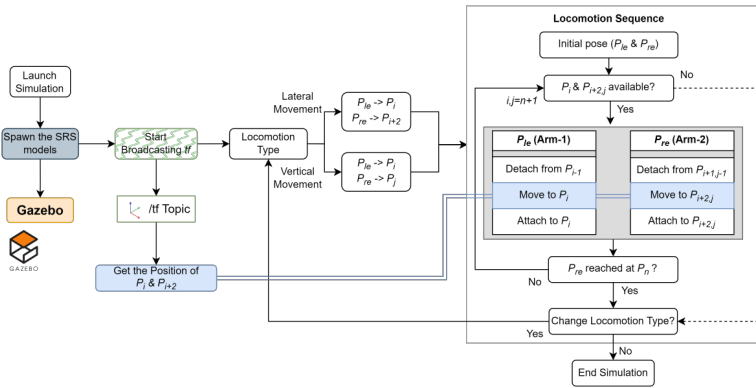
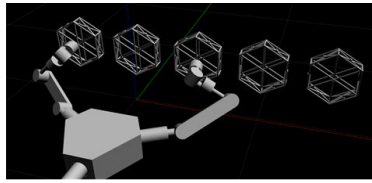


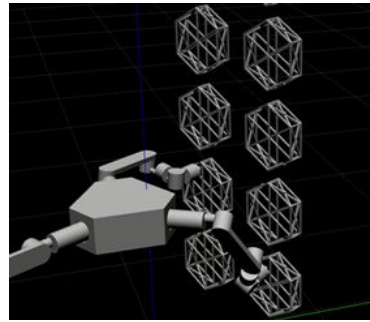
Fig. 2: Control Architecture

3.1 Spawning of the SRS Models

The first step during the simulation is to spawn the SRS models as SDF³ type, and these models are spawned within the gazebo using `/gazebo/spawn_sdf_model` service from the ROS services. The models are spawned for both lateral and vertical configurations, wherein for lateral locomotion configuration, the SRS is placed in a series manner along a single axis equidistantly as shown in the Fig. 3a. The vertical locomotion configuration from Fig. 3b requires the SRS to be placed on either side linearly to lie within the workspace of two active arms.



(a) Spawned SRS Models in Lateral Configuration



(b) Spawned SRS Models in Vertical Configuration

Fig. 3: SRS Models Spawned in Two Configurations

³ SDF - Separate XML-based Description Format for gazebo

3.2 Obtaining the Coordinates of SRS Models

The second-generation transformation frames (tf2) library⁴ keeps track of multiple coordinate frames within the simulation over time and their relationship in a tree structure. Every model has its link frames⁵ and the exact location of a particular model can be determined by looking up at the transformation frames tree structure. Since MARIO and all the SRS models are within the world frame, it is required such that only the locomotion is performed in terms of MARIO's torso frame (local frame of the multi-arm robot (see Fig. 4a)). The relative pose and orientation of SRS coordinate frames only with respect to the MARIO's torso frame are being broadcast to the `/tf` topic (see Fig. 4b & 4c for all link frames in both locomotion configuration).

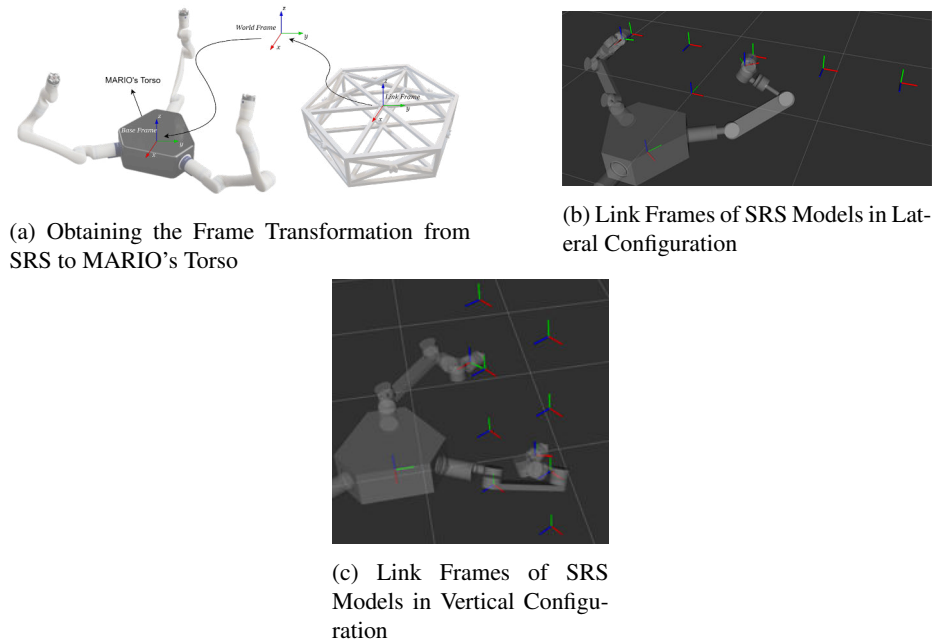


Fig. 4: Frame Transformation and Localised Link Frames

Additionally, to find a relationship between any two link frames, it is possible to look at the tree structure at any given time, which is maintained by the `/tf` topic. The pose and orientation of a particular SRS model relative to the torso's frame of reference are obtained through the frame transformation from the world frame. Once the coordinates (way-point) of the SRS model are transformed, the respective manipulator will be directed towards the SRS only if it is located within the workspace.

⁴ tf2-ros (Transformation Frames) - <http://wiki.ros.org/tf2>

⁵ A frame is the representation of Cartesian coordinate axes.

3.3 Locomotion Strategies & Sequence Execution

To perform locomotion on a planar structure composed of an SRS grid, we propose two locomotion primitives, namely lateral and vertical movements. A robot performing these two primitives can reach any point on the 2-D grid plane. These two primitives are discussed as follows:

Lateral Movement: The lateral movement scenario considers n SRS positioned linearly with their SI positions P_i , where $i = \{2, 3, \dots, n\}$ and the robot is moving from one SRS (left) to another SRS (right) in sequence shows a lateral movement as illustrated in Fig. 5. Two of the manipulators of MARIO perform this lateral movement, displacing its torso along the direction of the SRS sequence using an inchworm mechanism.

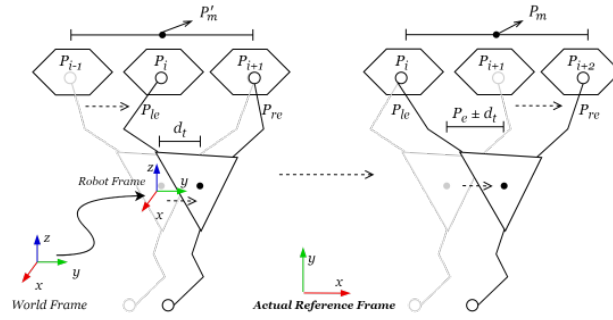


Fig. 5: Lateral Movement

For explanation, consider that the robot's left and right arms are docked on the positions P_{i-1} and P_{i+1} , respectively. The steps to achieve this movement are described as follows:

Step-1 $\rightarrow$ The left arm of the robot is moved from P_{i-1} to P_i . The left (P_{le}) and right (P_{re}) end-effector positions relative to the torso's reference frame can be calculated as,

$$P_{le} = P_i + d_s \quad (1)$$

$$P_{re} = P'_{re} - d_t \quad (2)$$

where d_s is the distance between SI position & the end-effector (due to empirical knowledge, it is a practical offset adapted within the simulation environment), d_t is the offset required to displace the torso applied to the non-moving arm, and P'_{re} is the previous position values of right end-effector.

Step-2 $\rightarrow$ The second step aligns the torso between both arms. The midpoint is calculated during each iteration, and this step is achieved initially by subtracting

the mid-point of both end-effectors during the previous (P'_m) & current pose (P_m) as follows,

$$P_m = \frac{P_{le} + P_{re}}{2} \quad (3)$$

$$P_e = P_m - P'_m \quad (4)$$

During the same instance, the midpoint offset (P_e) is added to both the end-effectors because this results in the coordinated motion of both the arms and, ultimately, obtaining the net displacement required for the torso, as stated below,

$$P_{le} = P_{le} - P_e \pm d_t \quad (5)$$

$$P_{re} = P_{re} - P_e \pm d_s \quad (6)$$

However, during each locomotion cycle the P_{i+2} could be out of reachable workspace for P_{re} , and it would be only possible if P_{le} coordinates such that the link frame of the P_{i+2} lies within the workspace of the P_{re} , provided that the end-effector's pose and orientation remains the same.

Vertical Movement: The vertical movement scenario considers n SRS positioned on either side linearly with their SI positions P_i & P_j for P_{le} & P_{re} respectively, where $i, j = \{1, 2, 3, \dots, n\}$. Both arms move sequentially with their respective SRS, and the resultant displacement of the torso resembles a biped mechanism, as shown in Fig 6.

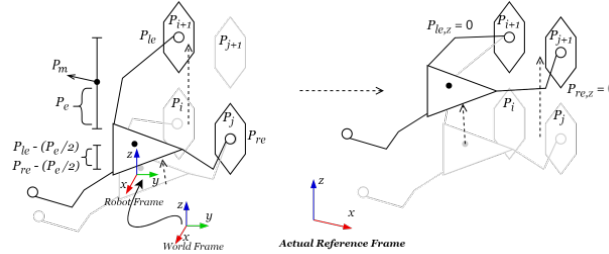


Fig. 6: Vertical Movement

Initially, consider the robot's arms P_{le} & P_{re} are docked on the SI positions $i, j = 1$. The steps to achieve this movement involve:

Step-1 $\rightarrow$ P_{le} moves to P_{i+1} , by considering left end-effector's previous (P'_{le}) & current positions (P_{le}), the mid-point (P_m) and the mid-point offset (P_e) is obtained from (8) & (9), respectively.

$$P_{le} = P_{i+1} \quad (7)$$

$$P_m = \frac{P_{le} + P'_{le}}{2} \quad (8)$$

$$P_e = P_m - P'_{le} \quad (9)$$

Now, P_e is added to both the end-effectors to achieve simultaneous motion, and can be stated with the (10) & (11) as follows,

$$P_{le} = P_{le} - \frac{P_e}{2} \quad (10)$$

$$P_{re} = P_{re} - \frac{P_e}{2} \quad (11)$$

Step-2 → During the second step, where P_{re} moves to P_{j+1} , aligning the torso by bringing both arms end-effectors to the same plane as torso (this motion can be achieved by simply equating the z-axis⁶ ($P_{le,z}$ & $P_{re,z}$) of both end-effectors as zero, shown in (12) & (13)).

$$P_{le,z} = 0 \quad (12)$$

$$P_{re,z} = 0 \quad (13)$$

Note that to perform these steps in simulation, the equations from (1) to (13) for both end-effectors P_{le} and P_{re} are implemented within the scope of the programming. By sequencing these two locomotion primitives, MARIO can reach any point on a planar surface grid; as a result, combined locomotion generalises the proposed strategy to complex locomotion scenarios on a plane. A use case for this approach is discussed in Sec. 4.2.

To execute the locomotion, the sequence of standard tasks to be performed for both the arms within a single locomotion sequence cycle are mentioned as follows (refer Fig. 2 for locomotion sequence),

Step-1 → Detach from $P_{i-1,j-1}$

Step-2 → Moving the Active arm to $P_{i,j}$

Step-3 → Attach to $P_{i,j}$

At the beginning of the simulation, the active arms must move to the initial pose so that the arm's joint configuration is well suited to perform the locomotion as planned. The notation for SRS models is different for both lateral and vertical scenarios, where P_i is only considered in lateral movement because of the single linear arrangement of the SRS models, but P_j along with P_i in vertical movement for the double linear arrangement of the SRS models.

During the locomotion of MARIO across the SRS structures, it is required that the end-effectors are docked to their respective SRS, forming a tight grasp so that if the arms are manipulated, the resultant motion is affected on the MARIO's torso. PAL Robotics⁷ has developed a plugin *gazebo_ros_link_attacher* [15] that attaches/detaches multiple objects as stated in Step-1 & 3 by creating/deleting an additional joint between any two links, by utilising this plugin, the coupling mechanism in simulation can be achieved.

⁶ Axis depends on the progress of motion and orientation of the end-effectors w.r.t the torso.

⁷ PAL Robotics - <https://pal-robotics.com/>

As mentioned in Step-2, the manipulators must be configured with appropriate controllers to direct the arms towards the SRS within the ROS simulation. This requirement can be accomplished using *position_controller* under the *JointTrajectoryController* because this controller receives the command in the desired position as input (as the requirement is feeding the way-point of a particular SRS model within the robot's workspace) and sends the position output.

4 Simulation Architecture & Results

This section explains the simulation architecture, followed by simulation results.

4.1 Simulation Architecture

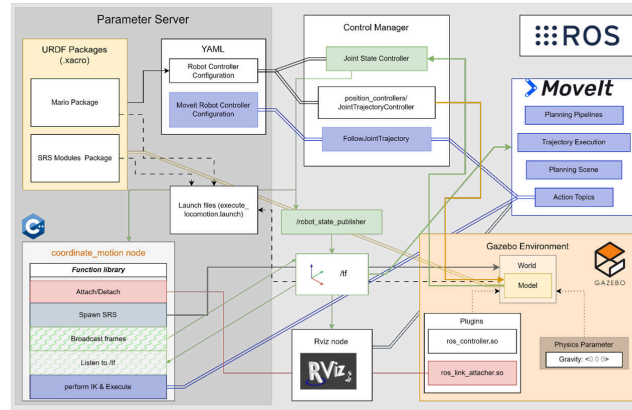


Fig. 7: ROS-Gazebo-Based Simulation Architecture

The validation of devised locomotion strategies is achieved with ROS-Gazebo-based simulation. The initial stage carried out to develop the simulation involving multi-arm-based robotic models requires the generation of necessary & working URDF packages, and the working arm groups within its semantic description file that are responsible for locomotion should be described as subgroups within a parent group so that multi-arm coordination is in effect. Furthermore, integrating the necessary plugins within the workspace ensures the appropriate working of the docking/coupling mechanisms.

The control framework was built through programming (C++) while inheriting the required functionality through essential libraries and packages such as MoveIt! and transformation frames. Since the motive for MARIO is to perform on-orbit operations, it is desired to simulate zero gravity conditions that can be established by simply zeroing the components of gravity function in the Gazebo physics simulator. Finally, Fig. 7 briefly illustrates the association of the aforementioned ROS development and connects it with the simulation environment.

The simulation was performed on a computer with Intel(R) Core™ i7-9750H CPU @2.60GHz processor paired with 16 GB RAM and an NVIDIA GeForce GTX 1050 graphics card on Ubuntu 2020.04 platform. Furthermore, the project was developed based on C++, and the simulation was validated through ROS Noetic Ninjemys 8 with Gazebo v11.11.

4.2 Simulation Results of the locomotion types

The data from the ROS simulation was logged with the help of rosbag⁸ package into a bag file, which is recorded by subscribing to the topics: */joint_states* and */gazebo/link_states* because the joint values and the end-effector's position in terms of CCS are published in the respective topics. Additionally, the plots were generated through PlotJuggler⁹. The simulation result considers three main scenarios for performing the locomotion, namely lateral movement, vertical movement, and combined locomotion, discussed as follows:

Lateral Movement: Fig. 8a, 8b, & 8c is the simulation representation of the approach for lateral movement similar to the inchworm mechanism as discussed in section 3.3. To begin with, Fig. 8a shows Arm-1 reaching the SRS (P_2), and Fig. 8b & 8c is the transition of the second step. Furthermore, Arm-1 (P_{le}) assists Arm-2 (P_{re}) in reaching the SRS (P_4). To further understand the locomotion performed within the simulation, plots Fig. 11a have been charted where markers show the exact point of occurrence in the timeline for these 2-steps discussed above. Furthermore, these graphs are generated in terms of CCS representing the central axis of the torso and end-effectors relative to the world frame and position-velocity relation of end-effectors.

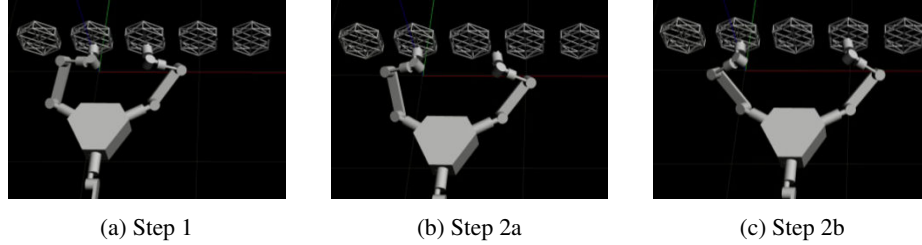


Fig. 8: Lateral Movement

Vertical Movement: The vertical movement is illustrated through Fig. 9a, 9b, & 9c, which is similar to the biped mechanism as derived in Sec. 3.3. The gradual increase in z-axis value from Fig. 11b indicates that the robot is moving along the z-axis.

⁸ rosbag - <http://wiki.ros.org/rosbag>

⁹ PlotJuggler - <https://plotjuggler.io/>

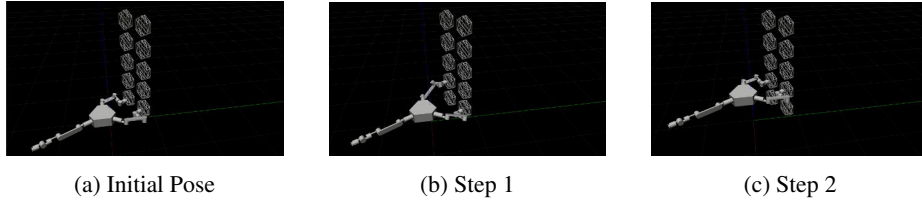


Fig. 9: Vertical Movement

Combined Locomotion A combination of both locomotion types was tested and demonstrated to validate the flexibility of the locomotion control framework as mentioned in Sec. 3.3, where Fig. 10a shows the completion of the vertical movement, Fig. 10b & 10c shows the transition to lateral movement. Finally, Fig. 10d indicates that Arm-1 aims for SRS (P_2), and the lateral locomotion cycle continues until Arm-2 is reached at the last SRS model. The markers in Fig. 11c represent the switching event taking place in the timeline. The ROS-based simulation videos from a playlist can be accessed here¹⁰, demonstrating the validation of locomotion approaches for a multi-arm robot.

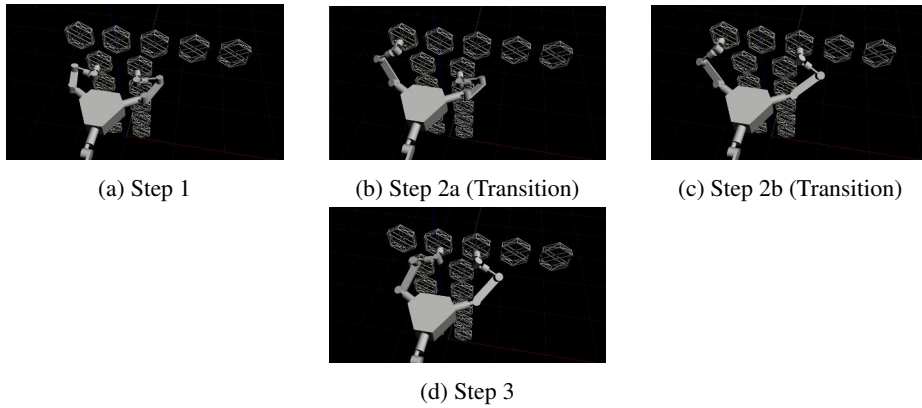


Fig. 10: Combined Locomotion

5 Conclusions

A simulation framework is developed for a multi-arm space robot performing locomotion on a planar space structure. The locomotion strategies are devised for lateral and vertical movements that can be combined to reach any point on a planar space structure. The locomotion is executed in three steps within a simulated environment, including the

¹⁰ Simulation Playlist - <http://tinyurl.com/MotionPlanning-MARIO>

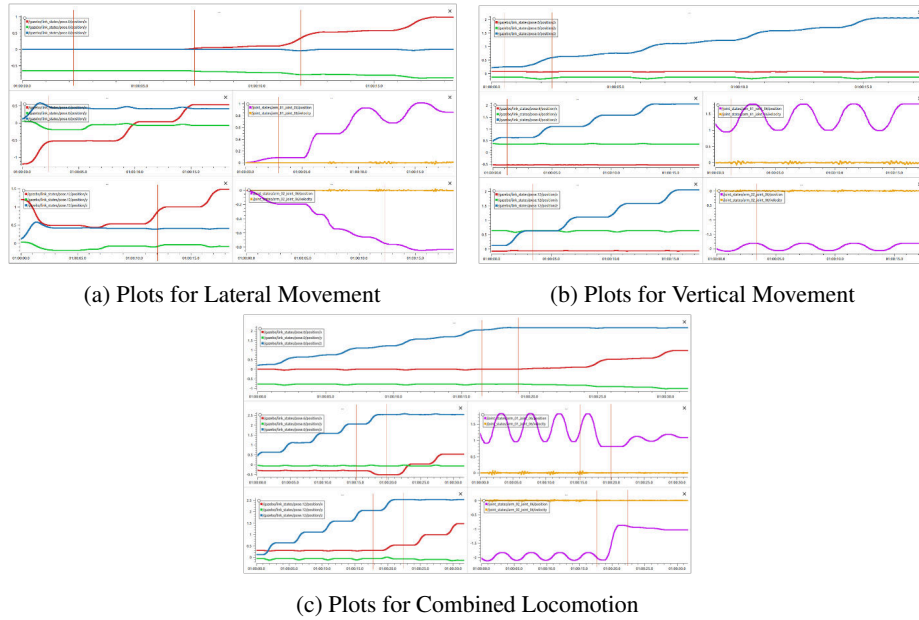


Fig. 11: Plots of the Locomotion Types

coordinated control of multiple arms. Simulation results show that the discussed ROS-based simulation architecture can generate the desired space scenario with assembly components and execute the proposed locomotion approach. Collision avoidance and improved combined locomotion approaches are immediate future works. The developed ROS-Gazebo-based simulation is available on the GitHub repository [16].

References

1. Nicolas Lee, Paul Backes, Joel Burdick, Sergio Pellegrino, Christine Fuller, Kristina Hogstrom, Brett Kennedy, Junggon Kim, Rudranarayan Mukherjee, Carl Seubert, and Yen-Hung Wu. Architecture for in-space robotic assembly of a modular space telescope. <https://doi.org/10.1117/1.JATIS.2.4.041207>, 2016.
2. Mathieu Deremetz, Pierre Letier, Gerhard Grunwald, Máximo Alejandro Roa Garzon, Bernhard Brunner, Benoit Lietier, and Michel Ilzkovitz. Mosarwm: A relocatable robotic arm demonstrator for future on-orbit applications. <https://iafastro.directory/iaac/paper/id/60119/summary/>, 2020.
3. Máximo A Roa, Korbinian Nottensteiner, Gerhard Grunwald, Pablo Lopez Negro, Aurelian Cuffolo, Sabine Andiappane, Mathieu Rognant, Antoine Verhaeghe, and Vincent Bissonnette. In-space robotic assembly of large telescopes. <https://www.eso.org/sci/facilities/eelt/>, 2019.
4. Hrishik Mishra, Marco De Stefano, and Christian Ott. Dynamics and control of a reconfigurable multi-arm robot for in-orbit assembly. <https://repositum.tuwien.at/handle/20.500.12708/101891>, 55:235–240, 7 2022.

5. Wenfu Xu, Yu Liu, and Yangsheng Xu. The coordinated motion planning of a dual-arm space robot for target capturing. <https://doi.org/10.1017/S0263574711001007>, 30:755–771, 2012.
6. Shuji Yang, Hao Wen, Yunhao Hu, and Dongping Jin. Coordinated motion control of a dual-arm space robot for assembling modular parts. <https://doi.org/10.1016/j.actaastro.2020.08.006>, 177:627–638, 12 2020.
7. Kazuya Yoshida, Ryo Kurazume, and Yoji Umetani. Dual arm coordination in space free-flying robot. <https://doi.org/10.1109/ROBOT.1991.132004>, 3:2516–2521, 1991.
8. Raymond H. Plaut. Mathematical model of inchworm locomotion. <https://doi.org/10.1016/j.ijnonlinmec.2015.05.007>, 76:56–63, 2015.
9. Ye Xie, Bin Lou, Anhuan Xie, and Dan Zhang. A review: Robust locomotion for biped humanoid robots. <https://dx.doi.org/10.1088/1742-6596/1487/1/012048>, 1487(1):012048, mar 2020.
10. Shuai Chang, Yansheng Liu, Haibo Du, Feng Xie, and Jinfeng Zhang. Modeling and gait planning simulation of tower-climbing robot based on ros. <https://doi.org/10.1109/ICNC52316.2021.9609063>, pages 194–200, 2021.
11. Eduardo Henrique Maciel, Renato Ventura Bayan Henriques, and Walter Fetter Lages. Control of a biped robot using the robot operating system. <https://doi.org/10.1109/SBR.LARS.Robocontrol.2014.46>, pages 247–252, 2014.
12. Antonio Manuele. Simulation-of-walking-biped. <https://github.com/AntoManuele/Simulation-of-Walking-Biped>, 7 2021.
13. Johannes Biermann. dual_arm_demo. https://github.com/bi3ri/dual_arm_demo, 3 2022.
14. Saksham Bhadani, Sairaj R Dillikar, Omkar N Pradhan, Irene Cotrina de los Mozos, Leonard Felicetti, Saurabh Upadhyay, and Gilbert Tang. A ros-based simulation and control framework for in-orbit multi-arm robot assembly operations. <https://dspace.lib.cranfield.ac.uk/handle/1826/20622>, 10 2023.
15. PAL Robotics. Gazebo ros link attacher. https://github.com/pal-robotics/gazebo_ros_link_attacher, 4 2016.
16. Sairaj Dillikar, Saurabh Upadhyay, Leonard Felicetti, and Gilbert Tang. Motion planning and control of multi-arm robot for in-orbit operations. <https://github.com/sairajdillikar/Motion-Planning-and-Control-of-Multi-Arm-Robot-for-In-Orbit-Operations>, 5 2023.



DSPACE

<https://dspace.org/>

A ROS-based control framework for simulating locomotion of a multi-arm space assembly robot

Dillikar, Sairaj R.; Leslie, Cameron; Upadhyay, Saurabh; Felicetti, Leonard; Tang, Gilbert

2025

Springer

https://doi.org/10.1007/978-3-031-72062-8_9

Dillikar SR, Leslie C, Upadhyay S, et al., (2025) A ROS-based control framework for simulating locomotion of a multi-arm space assembly robot. In: Towards Autonomous Robotic Systems, 25th Annual Conference, (TAROS 2024), 21-23 August 2024, London, UK, Part II. Lecture Notes in Computer Science, Volume 15052, pp. 93-105