(Preprint) AAS 21-522

A UNIQUE GROUND SIMULATOR FOR ON-ORBIT SERVICE MISSION EMULATION

Anirudh Chhabra[*] and Donghoon Kim[†]

Autonomous on-orbit servicing missions require high precision because any unwanted contact forces can cause damage to space systems. This makes the on-ground hardware-in-the-loop simulations necessary as operations must be thoroughly evaluated before such missions are carried out in space. A unique robotic platform is proposed where a 3 degrees-of-freedom (DOF) robotic arm is placed atop a 6 DOF Stewart platform. A mathematical model is developed for the resulting 9 DOF redundant manipulator, and the corresponding dynamics are modeled and verified. Simulation studies are conducted using a constrained and singularity-robust pseudo-inverse control scheme, and the system's capability to follow a scaled-down orbital trajectory is tested. The results are presented, and the future implications of this study are discussed.

INTRODUCTION

Space launches are tricky, and multiple problems which cannot be solved once satellites and/or spacecraft have been deployed can arise. With the huge number of public and private partnerships around the world planning regular space missions in the foreseeable future, astronauts may struggle in solving problems and thinking creatively during long space missions. The use of intelligent robotic systems to perform servicing tasks helps overcome such issues and adds flexibility in achieving mission objectives. Autonomous on-orbit servicing missions include operations like rendezvous & docking (RvD) with a non-cooperative satellite, refueling an old satellite, capturing a tumbling non-cooperative client satellite with a servicer spacecraft, etc. Such missions require high precision because any unwanted force generated upon contact can lead to unstable maneuvers and/or damage to the space systems.[1] A non-cooperative satellite is defined as one with which the servicer spacecraft is not compatible. Satellites in these categories were not originally designed for repair and refurbishment and lack modern docking mechanisms like target location markers or rendezvous sensors.[2] This makes the on-ground hardware-in-the-loop (HIL) simulations necessary as any contact operations must be thoroughly evaluated and verified before such missions are carried out in space. As physical testing of such systems in three-dimensional (3D), zero-gravity environments is extremely difficult, robotic manipulators are being developed and used to create a HIL simulation environment.

Serial manipulators are most commonly used for industrial applications such as auto manufacturing, painting, welding, assembly/disassembly, product testing, and inspection. This is primarily due to the wide range of motion of serial manipulators. Furthermore, the primary requirement of such applications is a synchronized operation. Serial manipulators have certain disadvantages such as slow response to control commands due to high stiffness and do not offer high precision. Because of such properties, they are more suited to industrial applications where the main requirement is the overall efficiency of the process cycle. However, to generate precise maneuvers and trajectories for the purpose of operations that require minimal contact forces, the response to control must be fast and the operational accuracy must be high. Therefore, such robots are not suitable by themselves for the purpose of a spacecraft simulator.

In contrast, parallel manipulators are known to perform with higher precision. Stewart platform SP is one of the widely used parallel manipulators.[3] It has 6 degrees-of-freedom (DOF) and is known for its

[*]Graduate Student, Department of Aerospace Engineering, University of Cincinnati, Cincinnati, Ohio 45221.
[†]Assistant Professor, Department of Aerospace Engineering, University of Cincinnati, Cincinnati, Ohio 45221.

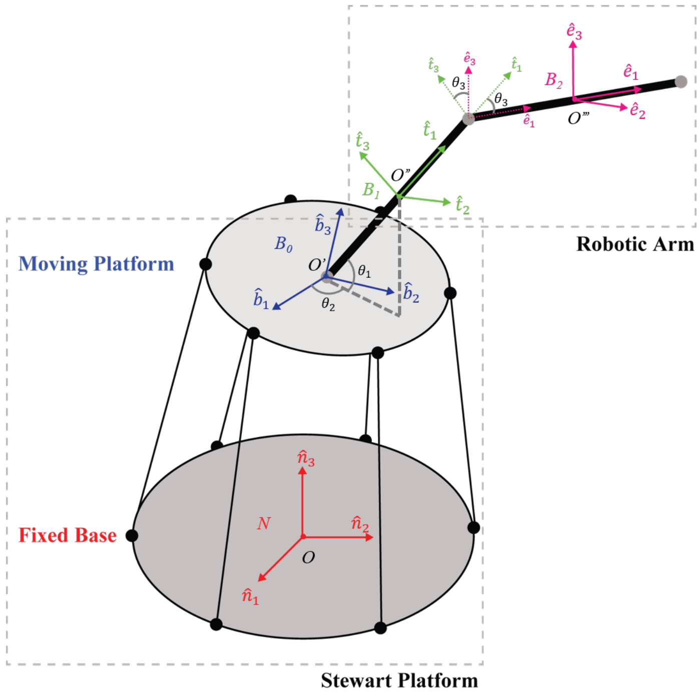


Figure 1: Proposed Integrated Platform

strong capability as an aircraft simulator. Other than that, this manipulator offers multiple advantages, such as precise positioning capability, high structural rigidity, and a large load-to-weight ratio. Furthermore, it has a high sampling rate which allows for a faster response to control commands. However, one major disadvantage of an SP is that it has a very limited range of motion due to a highly rigid structure. To overcome such a limitation, a unique robotic structure is proposed (See Figure 1). This structure aims to combine the advantages of serial and parallel manipulators offering high precision with a wide range of operations. In the proposed structure, a 3 DOF robotic arm (RA) is linked to the moving platform of the SP manipulator and acts as the end-effector of the manipulator. In the past, hybrid manipulators have been proposed because of their ability to combine the advantages of serial and parallel manipulators. Kumar et al.[4] discussed the modularity of hybrid robots by highlighting the vast number of combinations of serial and parallel robots that can be used for a number of applications. The proposed manipulator also offers the important advantage of redundancy which allows for a more stable control while taking into consideration the hardware constraints.

In some of the previous works, a number of models for spacecraft simulators have been developed. In 2010, Boge et al. developed a spacecraft simulator using 6 DOF KUKA robots with one on a stationary platform while the other on a linear track to simulate an RvD mission. Although a number of advantages, such as range of motion, were observed, there were some issues with the use of serial manipulators, such as low response time.[5] Nakka et al.[6] proposed the M-STAR spacecraft dynamics simulator in 2018 based on an air bearing system (ABS) controlled by thrusters providing 5 DOF and a vertical translational DOF. This is similar to some other work done on spacecraft simulator platforms based on the ABS, such as the ORION project at the University of Central Florida.[7] The major disadvantages of using the ABS for simulating the motion of spacecraft are the limited capacity of on-board tanks, battery endurance, and a huge investment in the floor/table to maintain a smooth motion. Certain studies have been done with hybrid manipulators based on an SP, such as the NASA 6 DOF dynamic test system which proposes the use of a stationary platform on top of an SP motion base.[8] A holonomic omnidirectional motion emulation robot, called HOMER, was proposed in 2007 and represents the spacecraft motion using an SP on top of a mobile platform. The primary issue with both of these platforms is the limited range of actuation of the SP as an end-effector.[9] The VES-II simulator developed at MIT proposed a serial manipulator (PUMA robot) on top of an SP. In this study, instead, the SP is used to represent the spacecraft, and the RA is used to represent the manipulator. Although a different approach to the one proposed in this study, Dubowsky et al.[10] highlighted the increased capability of using the SP as a motion base for a serial robotic manipulator. Neutral buoyant systems offer an efficient representation of 6 DOF motion but can face issues due to uncertainties in the dynamics model due to fluid damping and inertia disturbances and are therefore not desirable.[11] On the other hand, suspension systems-based spacecraft simulators are highly complex and present a great deal of difficulty in developing a reliable and accurate simulator.[12,13] The proposed robotic platform offers numerous advantages in the realm of robotic manipulation and capture in space operations by offering a highly capable HIL simulation platform.

The primary contributions of this study are the modeling of system kinematics and dynamics, verification of dynamics, scaling of an orbital trajectory based on constraints, and an inverse kinematics-based control law for redundancy resolution. The mathematical model of the proposed platform is developed, and the corresponding dynamics of the system are formulated using Lagrangian mechanics. A scaled-down orbital trajectory is generated using orbital dynamics as the reference trajectory for validating the integrated platform. A pseudo-inverse control scheme is applied to control the redundant manipulator. This allows the avoidance of singularities using null-space optimization as well as the avoidance of joint limits. Simulation studies are conducted for the integrated platform in order to track the scaled-down orbital trajectories. This allows us to validate the dynamics of the integrated platform and identify certain properties. At the end, the potential future work is discussed keeping in mind the results of this study.

PRELIMINARIES

Problem Considerations

For the purpose of this study, the limbs of the SP and the RAs are considered to be thin uniform rods. The moving platform of the SP is assumed to be a thin uniform disc. These considerations allow easier modeling of the system dynamics. Hardware constraints are only considered for the generalized coordinates but not for actuator velocities, forces, and torques. Furthermore, it is assumed that the motion of the robots is continuous without the consideration of minimum motion step size.

Stewart Platform

SP is a type of parallel manipulator with six limbs between two platforms (base and end-effector) offering 6 DOF motion for the end-effector. This makes it a highly suitable device in a number of industrial scenarios. Such a mechanism was firstly developed by Stewart for the purpose of being used as a flight simulator by simulating the conditions in space.[3] Then, Gough came up with a model with six parallel linear actuators which is now known as the SP. A number of developments started coming up soon in this area so as to properly define the highly nonlinear dynamics of this mechanism. Hunt[14] proposed the use of this mechanism as a robotic manipulator and called for a detailed analysis of this mechanism as parallel robots like this one offers high precision position capability as compared to serial robots. However, the control mechanism also gets complicated with the highly nonlinear nature of the parallel manipulator. Dasgupta and Mruthyunjaya[15] discuss the historical developments in the dynamics modeling and control of the SP.

The generalized coordinates for the SP can be defined in terms of the translational and rotational motion of the moving platform, which is the end-effector of the SP, as follows:

$$\boldsymbol{\chi} = \begin{bmatrix} x & y & z & \theta_x & \theta_y & \theta_z \end{bmatrix}^T \in \mathcal{R}^6 \tag{1}$$

where $\begin{bmatrix} x & y & z \end{bmatrix}^T$ denotes the Cartesian position vector of the moving platform with respect to the fixed base, and $\begin{bmatrix} \theta_x & \theta_y & \theta_z \end{bmatrix}^T$ represents the rotation vector from the fixed base to the moving platform. The complete dynamics of the SP under the aforementioned assumptions can then be given by[16]

$$M_{SP}(\boldsymbol{\chi})\ddot{\boldsymbol{\chi}} + C_{SP}(\boldsymbol{\chi}, \dot{\boldsymbol{\chi}})\dot{\boldsymbol{\chi}} + G_{SP}(\boldsymbol{\chi}) = F_{SP} \tag{2}$$

where the components M_{SP} and C_{SP} include the dynamics of the end-effector as well as the limbs. Also, $F_{SP} = J^T \cdot \boldsymbol{u}$, where $J \in \mathcal{R}^{6\times 6}$ is the Jacobian of the manipulator and $\boldsymbol{u} \in \mathcal{R}^6$ represents the forces generated in each of the six limbs.

MODELING OF THE INTEGRATED PLATFORM

The proposed robotic platform with a 3 DOF RA on top of a 6 DOF SP is shown in Figure 1. The first link of the RA has 2 DOF, and the second link has 1 DOF. Lagrange's method is used to determine the equations of motion, and the modeling procedure is described in this section.

Generalized Coordinates

The SP is defined by six generalized coordinates as mentioned in Eq. (1). Once the RA is attached on top of it, three more generalized coordinates are added to this set, which are $(\theta_1, \theta_2, \theta_3)$. Here, θ_1 and θ_2 are the angles made by the first link ($O' - O''$) around $\hat{\boldsymbol{b}}_1$ and $\hat{\boldsymbol{b}}_2$ axes in B_0 frame, whereas, θ_3 is the angle made by the second link ($O'' - O'''$) about $\hat{\boldsymbol{t}}_2$ axis in B_1 frame. This leads to a total of nine generalized coordinates that are defined as the vector $\boldsymbol{q}$ as follows:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\chi}^T & \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \in \mathcal{R}^9 \tag{3}$$

Frame Definitions

Figure 1 also shows the different frames that are used to define the multi-body robotic manipulator. The inertial frame is defined as $N : \{O, \hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_2, \hat{\boldsymbol{n}}_3\}$. This comprises of the fixed-platform or the base of the SP. The second frame is the moving platform of the SP or the first body frame, and it is defined as $B_0 : \{O', \hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2, \hat{\boldsymbol{b}}_3\}$. The remaining frames B_1 and B_2 define the two links of the RA and are expressed as $B_1 : \{O'', \hat{\boldsymbol{t}}_1, \hat{\boldsymbol{t}}_2, \hat{\boldsymbol{t}}_3\}$ and $B_2 : \{O''', \hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3\}$, respectively.

The rotation from N to B_0 is defined by the (3-2-1) Euler Angle $(\theta_z, \theta_y, \theta_x)$ sequence, and the direction cosine matrix (DCM) is given by

$$C_N^{B_0} = \begin{bmatrix} \mathrm{c}(\theta_y)\mathrm{c}(\theta_z) & \mathrm{c}(\theta_x)\mathrm{s}(\theta_z) + \mathrm{c}(\theta_z)\mathrm{s}(\theta_x)\mathrm{s}(\theta_y) & \mathrm{s}(\theta_x)\mathrm{s}(\theta_z) - \mathrm{c}(\theta_x)\mathrm{c}(\theta_z)\mathrm{s}(\theta_y) \\ -\mathrm{c}(\theta_y)\mathrm{s}(\theta_z) & \mathrm{c}(\theta_x)\mathrm{c}(\theta_z) - \mathrm{s}(\theta_x)\mathrm{s}(\theta_y)\mathrm{s}(\theta_z) & \mathrm{c}(\theta_z)\mathrm{s}(\theta_x) + \mathrm{c}(\theta_x)\mathrm{s}(\theta_y)\mathrm{s}(\theta_z) \\ \mathrm{s}(\theta_y) & -\mathrm{c}(\theta_y)\mathrm{s}(\theta_x) & \mathrm{c}(\theta_x)\mathrm{c}(\theta_y) \end{bmatrix} \tag{4}$$

where c and s denote the cosine and sine function, respectively. Similarly, the DCM from B_1 to B_0 defined by a (3-2-1) rotation of Euler angles $(\theta_2, \theta_1, 0)$ and the DCM from B_2 to B_1 defined by a (3-2-1) rotation of Euler angles $(0, \theta_3, 0)$ are given by

$$C_{B_0}^{B_1} = \begin{bmatrix} \mathrm{c}(\theta_1)\mathrm{c}(\theta_2) & -\mathrm{c}(\theta_1)\mathrm{s}(\theta_2) & \mathrm{s}(\theta_1) \\ \mathrm{s}(\theta_2) & \mathrm{c}(\theta_2) & 0 \\ -\mathrm{c}(\theta_2)\mathrm{s}(\theta_1) & \mathrm{s}(\theta_1)\mathrm{s}(\theta_2) & \mathrm{c}(\theta_1) \end{bmatrix} \tag{5}$$

$$C_{B_1}^{B_2} = \begin{bmatrix} \mathrm{c}(\theta_3) & 0 & \mathrm{s}(\theta_3) \\ 0 & 1 & 0 \\ -\mathrm{s}(\theta_3) & 0 & \mathrm{c}(\theta_3) \end{bmatrix} \tag{6}$$

Kinematics

The position vectors of the two links of the RA in N frame are denoted by ${}^N\boldsymbol{r}_{B_1/N}$ $(O - O'')$, and ${}^N\boldsymbol{r}_{B_2/N}$ $(O - O''')$, respectively and can be expressed as

$${}^N\boldsymbol{r}_{B_1/N} = {}^N\boldsymbol{r}_{B_0/N} + C_{B_0}^{N}\ {}^{B_0}\boldsymbol{r}_{B_1/B_0} \tag{7}$$

$${}^N\boldsymbol{r}_{B_2/N} = {}^N\boldsymbol{r}_{B_0/N} + C_{B_0}^{N}\ {}^{B_0}\boldsymbol{r}_{B_1/B_0} + C_{B_0}^{N} C_{B_1}^{B_0}\ {}^{B_1}\boldsymbol{r}_{B_2/B_1} \tag{8}$$

where the terms ${}^N\boldsymbol{r}_{B_0/N}$, ${}^{B_0}\boldsymbol{r}_{B_1/B_0}$, and ${}^{B_1}\boldsymbol{r}_{B_2/B_1}$ are denoted by the vectors $O - O'$, $O' - O''$, and $O'' - O'''$, respectively. This leads to the translational velocity expression as follows:

$${}^N\boldsymbol{v}_{B_1/N} = \frac{d}{dt}\,{}^N\boldsymbol{r}_{B_1/N} \tag{9}$$

$${}^N\boldsymbol{v}_{B_2/N} = \frac{d}{dt}\,{}^N\boldsymbol{r}_{B_2/N} \tag{10}$$

Similarly, the expressions for the angular velocities are written as follows:

$$
{}^{N}\boldsymbol{\omega}_{B_1/N} = {}^{N}\boldsymbol{\omega}_{B_0/N} + C^{N}_{B_0}\,{}^{B_0}\boldsymbol{\omega}_{B_1/B_0} \tag{11}
$$

$$
{}^{N}\boldsymbol{\omega}_{B_2/N} = {}^{N}\boldsymbol{\omega}_{B_0/N} + C^{N}_{B_0}\,{}^{B_0}\boldsymbol{\omega}_{B_1/B_0} + C^{N}_{B_0}C^{B_0}_{B_1}\,{}^{B_1}\boldsymbol{\omega}_{B_2/B_1} \tag{12}
$$

Dynamics

Using the translational and rotational velocities for both the links of the RA, the kinetic energy (T) and potential energy (U) for the RA are given by

$$
T = \frac{1}{2}\sum_{\gamma=1}^{2}\begin{bmatrix}{}^{N}\boldsymbol{v}_{B_\gamma/N}\\ {}^{N}\boldsymbol{\omega}_{B_\gamma/N}\end{bmatrix}^{T}\begin{bmatrix}m_\gamma I_{3\times3} & 0_{3\times3}\\ 0_{3\times3} & {}^{N}\mathcal{I}_\gamma\end{bmatrix}\begin{bmatrix}{}^{N}\boldsymbol{v}_{B_\gamma/N}\\ {}^{N}\boldsymbol{\omega}_{B_\gamma/N}\end{bmatrix} \tag{13}
$$

$$
U = \sum_{\gamma=1}^{2} m_\gamma \boldsymbol{g}^{T}\,{}^{N}\boldsymbol{r}_{B_\gamma/N} \tag{14}
$$

where the subscript $\gamma = 1, 2$ represents the first and second link of the RA, $\boldsymbol{g} \in \mathcal{R}^3$ denotes the gravity vector, m is the mass, I is the identity matrix, and the moment of inertia (MOI) is given as

$$
{}^{N}I_\gamma = C^{N}_{B_\gamma}\,{}^{B_\gamma}I_\gamma C^{\gamma}_{B_N} \tag{15}
$$

where

$$
{}^{B_\gamma}I_\gamma = \frac{1}{3}\begin{bmatrix}0 & 0 & 0\\ 0 & m_\gamma l_\gamma^2 & 0\\ 0 & 0 & m_\gamma l_\gamma^2\end{bmatrix} \tag{16}
$$

Here, l is the length of the link.

Now, one can form the Lagrangian for the RA as follows:

$$
\frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\boldsymbol{q}} \tag{17}
$$

$$
= \frac{d}{dt}\frac{\partial T}{\partial \dot{\boldsymbol{q}}} - \frac{\partial T}{\partial \boldsymbol{q}} + \frac{\partial U}{\boldsymbol{q}} = Q \tag{18}
$$

where $L = T - U$ and $Q = F_{RA} = \begin{bmatrix}\mathbf{0}_{1\times6} & \tau_1 & \tau_2 & \tau_3\end{bmatrix}^T$ defines the generalized forces applicable to the RA coordinate $\begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}^T$. The equation of motion for the RA can then be expressed in matrix form as follows:

$$
M_{RA}(\boldsymbol{q})\ddot{\boldsymbol{q}} + C_{RA}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G_{RA}(\boldsymbol{q}) = F_{RA} \tag{19}
$$

where the inertia matrix $M_{RA}(\boldsymbol{q})$ can be expressed by rewriting the kinetic energy expression in terms of the generalized coordinates using the following relations:

$$
\boldsymbol{v}_\gamma = {}_{\mathrm{L}}J_\gamma \cdot \dot{\boldsymbol{q}} \tag{20}
$$

$$
\boldsymbol{\omega}_\gamma = {}_{\mathrm{A}}J_\gamma \cdot \dot{\boldsymbol{q}} \tag{21}
$$

Here, ${}_{\mathrm{L}}J_i$ and ${}_{\mathrm{A}}J_i$ are the linear and angular Jacobian matrices that define the translational and rotational velocities of the two links of the RA in terms of $\boldsymbol{q}$. These Jacobians may be determined elementwise using any symbolic computation platform. The kinetic energy can then be stated as

$$
T = \frac{1}{2}\sum_{\gamma=1}^{2}\left(m_\gamma \dot{\boldsymbol{q}}^T({}_{\mathrm{L}}J_\gamma)^T({}_{\mathrm{L}}J_\gamma)\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T({}_{\mathrm{A}}J_i)^T\,{}^{N}\mathcal{I}_\gamma({}_{\mathrm{A}}J_\gamma)\dot{\boldsymbol{q}}\right) \tag{22}
$$

This leads to write the inertia matrix directly in terms of the generalized coordinates as follows:

$$M_{RA}(\boldsymbol{q}) = \sum_{\gamma=1}^{2} m_\gamma ({}_{\mathrm{L}}J_\gamma)^T ({}_{\mathrm{L}}J_\gamma) + ({}_{\mathrm{A}}J_\gamma)^T \, {}^{N}\mathcal{I}_\gamma ({}_{\mathrm{A}}J_\gamma) \tag{23}$$

Then, the first term in Eq. (18) becomes

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} = \frac{d}{dt}\left(\sum_{j=1}^{9} M_{ij}\dot{q}_j\right) \tag{24}$$

$$= \sum_{j=1}^{9} M_{ij}\ddot{q}_j + \sum_{j=1}^{9} \frac{dM_{ij}}{dt}\dot{q}_j \tag{25}$$

where $i = 1, 2, \ldots, 9$. The first term in Eq. (25) represents the dynamic interactions among the multiple joints due to accelerations. Basically, this represents the first term in Eq. (19).

The second term in Eq. (25) can be rewritten using chain rule as

$$\sum_{j=1}^{9} \frac{dM_{ij}}{dt}\dot{q}_j = \sum_{j=1}^{9}\sum_{k=1}^{9} \frac{\partial M_{ij}}{\partial q_k}\dot{q}_k\dot{q}_j \tag{26}$$

and the second term in Eq. (18) also yields partial derivatives from the inertia matrix and can be stated as

$$\frac{\partial T}{\partial q_i} = \frac{1}{2}\sum_{j=1}^{9}\sum_{k=1}^{9} \frac{\partial M_{jk}}{\partial q_i}\dot{q}_k\dot{q}_j \tag{27}$$

To obtain the Coriolis matrix $C_{RA}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, these terms must be combined, and one defines the Christoffel's symbol to make the computation of the Coriolis matrix direct and simple as follows:

$$Z_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i}\right) \tag{28}$$

Thus, the Coriolis matrix can now be expressed as

$$C_{RA}(\boldsymbol{q}, \dot{\boldsymbol{q}})_{ij} = \sum_{k=1}^{9} Z_{ijk}\dot{q}_k \tag{29}$$

The last term on the LHS in Eq. (19) represents the gravitational torques on the system and is written as

$$G_{RA}(\boldsymbol{q}) = \frac{\partial U}{\partial \boldsymbol{q}} \tag{30}$$

Combining Eq. (19) and Eq. (2) yields the complete system model as follows:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) = Q \tag{31}$$

where

$$M(\boldsymbol{q}) = \begin{bmatrix} M_{SP}(\boldsymbol{\chi}) & 0_{6\times3} \\ 0_{3\times6} & 0_{3\times3} \end{bmatrix} + M_{RA}(\boldsymbol{q}) \tag{32}$$

$$C(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} C_{SP}(\boldsymbol{\chi}, \dot{\boldsymbol{\chi}}) & 0_{6\times3} \\ 0_{3\times6} & 0_{3\times3} \end{bmatrix} + C_{RA}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{33}$$

$$G(\boldsymbol{q}) = \begin{bmatrix} G_{SP}(\boldsymbol{\chi}) \\ \boldsymbol{0}_{3\times1} \end{bmatrix} + G_{RA}(\boldsymbol{q}) \tag{34}$$

$$Q = \begin{bmatrix} F_{SP} \\ \boldsymbol{0}_{3\times1} \end{bmatrix} + F_{RA} \tag{35}$$

Verification of Dynamics

To verify the obtained dynamic system, the following characteristic properties of a robot equation must be satisfied.[17]

- $M(\boldsymbol{q})$ must be symmetric and positive definite.
- $\dot{M} - 2C \in R^{9\times 9}$ must be a skew-symmetric matrix.

The positive definiteness of $M(\boldsymbol{q})$ can be observed directly in its definition as given in Eq. (23) as the Jacobian matrices are multiplied by themselves, and the remaining inertial terms are positive definite. The second property can be proven analytically. The expressions for C can be obtained by using Eqs. (28) and (29) as follows:

$$
\begin{aligned}
(\dot{M} - 2C)_{ij} &= \dot{M}_{ij} - 2C_{ij} && (36)\\
&= \sum_{k=1}^{9} \left(\underbrace{\frac{\partial M_{ij}}{\partial q_k}\dot{q}_k}_{\text{Using chain rule}} - \frac{\partial M_{ij}}{\partial q_k}\dot{q}_k - \frac{\partial M_{ik}}{\partial q_j}\dot{q}_k + \frac{\partial M_{kj}}{\partial q_i}\dot{q}_k \right) && (37)\\
&= \sum_{k=1}^{9} \left(\frac{\partial M_{kj}}{\partial q_i}\dot{q}_k - \frac{\partial M_{ik}}{\partial q_j}\dot{q}_k \right) && (38)
\end{aligned}
$$

By replacing i and j in Eq. (38), one can obtain the transpose and prove the skew-symmetric property as

$$
(\dot{M} - 2C)_{ji} = \sum_{k=1}^{9} \left(\frac{\partial M_{ki}}{\partial q_j}\dot{q}_k - \frac{\partial M_{jk}}{\partial q_i}\dot{q}_k \right) \tag{39}
$$

This is known as the *passivity* property of the robot equations, which means that the total energy of the system is conserved under the influence of conservative forces.[18]

Another way of validating the dynamics is to use the conservation of energy given as follows:

$$
\Delta E = \Delta T + \Delta U = 0 \tag{40}
$$

TRAJECTORY PLANNING

Orbital Trajectory Generation

The orbital trajectory can be determined by converting the Keplerian coordinates into Cartesian coordinates. The Keplerian coordinates are defined using the following orbital elements: semi-major axis (a), eccentricity (e), inclination (i), Right ascension of ascending node (Ω), argument of perigee (ω), and true anomaly (ν). Then, the initial position and velocity of the orbital trajectory is found as[19]

$$
\begin{aligned}
x_0 &= r\left(\mathrm{c}(\Omega)\mathrm{c}(\omega+\nu) - \mathrm{s}(\Omega)\mathrm{s}(\omega+\nu)\mathrm{c}(i)\right) && (41)\\
y_0 &= r\left(\mathrm{s}(\Omega)\mathrm{c}(\omega+\nu) + \mathrm{c}(\Omega)\mathrm{s}(\omega+\nu)\mathrm{c}(i)\right) && (42)\\
z_0 &= r\left(\mathrm{s}(\omega+\nu)\mathrm{s}(i)\right) && (43)\\
\dot{x}_0 &= \frac{x_0 h e}{ra(1-e^2)}\mathrm{s}(\nu) - \frac{h}{r}\left(\mathrm{c}(\Omega)\mathrm{s}(\omega+\nu) + \mathrm{s}(\Omega)\mathrm{c}(\omega+\nu)\mathrm{c}(i)\right) && (44)\\
\dot{y}_0 &= \frac{y_0 h e}{ra(1-e^2)}\mathrm{s}(\nu) - \frac{h}{r}\left(\mathrm{s}(\Omega)\mathrm{s}(\omega+\nu) - \mathrm{c}(\Omega)\mathrm{c}(\omega+\nu)\mathrm{c}(i)\right) && (45)\\
\dot{z}_0 &= \frac{z_0 h e}{ra(1-e^2)}\mathrm{s}(\nu) + \frac{h}{r}\left(\mathrm{c}(\omega+\nu)\mathrm{s}i\right) && (46)
\end{aligned}
$$

where

$$r = \frac{a(1-e^2)}{1+e\,\mathrm{c}\nu} \tag{47}$$

$$h = \sqrt{\mu a(1-e^2)} \tag{48}$$

Now, the desired trajectory waypoints can be obtained using the initial conditions in Eqs. (41)-(46) by solving the following two body equation[20]

$$\ddot{\boldsymbol{r}} + \frac{\mu}{||\boldsymbol{r}||^3}\boldsymbol{r} = 0 \tag{49}$$

where μ is the gravitational parameter of a planet, and $\boldsymbol{r} = \begin{bmatrix} x_d & y_d & z_d \end{bmatrix}^T$ is the desired position vector. This work considers the Earth ($\mu = GM = 3.986 \times 10^{14}\mathrm{m}^3/\mathrm{s}^2$). The time period of the orbit (i.e. the time taken to complete one orbit) is given as

$$T_f = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{50}$$

Scaling Factor Determination

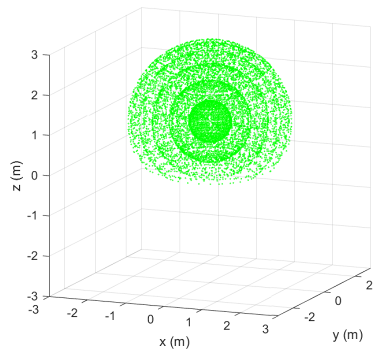
The semi-major axis of the orbital trajectory is used to define the size of an orbit, which is essentially a trajectory around the earth. For the proposed robotic platform to follow such a trajectory, it must be scaled down from its actual size to a size that fits within the platform workspace. In order to do this, a scaling factor is determined such that when multiplied with the semi-major axis and the earth constants, it provides a trajectory within the workspace of the platform. The first step of determining the scaling factor is to define the workspace of the platform end-effector. The workspace can be determined based on the hardware constraints of the platform (Table 1), and it is represented as a point cloud (Figure 2a). The smallest sphere which encloses this point cloud is then defined using Welzl's Algorithm[21] that determines the minimum (smallest) enclosing circles and spheres for a given point cloud. Welzl's algorithm is a recursive algorithm that runs in linear time $\mathcal{O}(2^{d+1}n)$, where d is the number of dimensions of the enclosing shape, and n is the number of points in the point cloud. This algorithm is initialized by randomly choosing three points from the point cloud and forming an initial sphere. Then, the next randomly chosen point is tested for inclusion. If inside, the algorithm moves ahead and tests the next random point. If not, the process is restarted with a constraint in which the outside point is now on the surface of the new enclosing sphere. This ensures that for every new point, a unique minimum sphere is defined that also includes all previous points.

Table 1: Constraints on the Generalized Coordinates

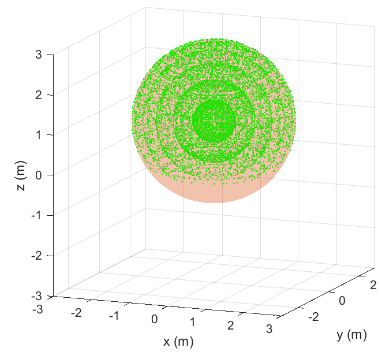
Coordinate ($\boldsymbol{q}$)	Minimum Value	Maximum Value	Unit
x	-0.1	0.1	m
y	-0.1	0.1	m
z	0.95	1.05	m
θ_x	-15	15	deg
θ_y	-15	15	deg
θ_z	-30	30	deg
θ_1	-180	0	deg
θ_2	-720	720	deg
θ_3	-180	180	deg

The first scaled-down trajectory is then defined by equating the major axis ($2a$) to the diameter of the enclosing sphere (d_{sphere}). This provides the following initial scaling factor

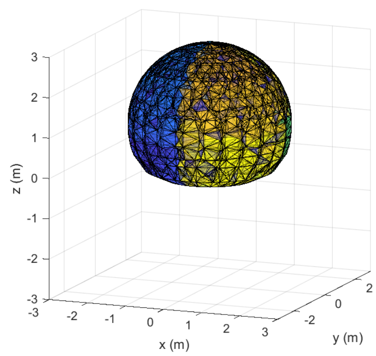
$$\eta_0 = \frac{d_{\text{sphere}}}{2a} \tag{51}$$



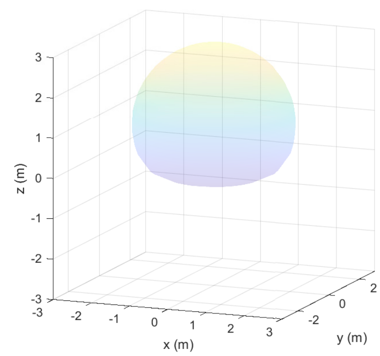
(a) Step 1: Point Cloud Generation



(b) Step 2: Minimum Enclosing Sphere



(c) Step 3: Delaunay triangulation

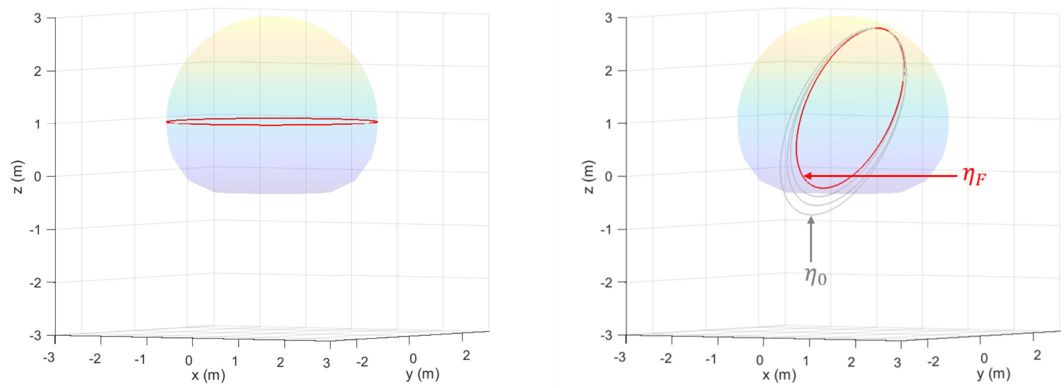


(d) Step 4: Final Enclosing Mesh

Figure 2: Workspace Determination

It is important to note that certain trajectories with high inclination do not fit inside the workspace because the point cloud distribution is not completely spherical. Mesh generation is a technique that is used to convert continuous geometric spaces, such as point cloud distribution, into a collection of discrete geometrical surfaces. This provides the capability to perform geometric operations such as checking whether a point lies within the mesh. Without mesh generation, it becomes a difficult task to perform such operations with just the point cloud data. A tetrahedral mesh is formed using the point cloud and is generated using Delaunay triangulation.[22] Tetrahedral meshes are the most basic and least computationally expensive meshes to represent 3D point cloud data. Delaunay triangulation is a special case of regular point cloud triangulation as it aims to maximize the minimum angles in all the tetrahedrons. In triangulation, circumscribing spheres are created to find the tetrahedrons with the points on their vertices. This provides a set of tetrahedrons $T(V, E)$, where V is the set of vertices, and E is the set of edges. In Delaunay triangulation, the circumscribing spheres of the tetrahedrons are defined such that it does not enclose any points. It offers certain advantages, such as consistency and a unique final result (independent of algorithm initial conditions). An incremental divide and conquer algorithm is used for this process which runs with an acceptable time-complexity of $\mathcal{O}(n \log(n))$.[23]

The outer surface of this triangulation is a convex polygon that is chosen as the final enclosing mesh as shown in Figures 2c and 2d. All the trajectory waypoints are then checked if they lie in this mesh. If not, it reduces the scaling factor by a user-defined percentage (λ). In this work, 5% is used. Then, the trajectory is shifted so that the new upper vertex coincides with the old upper vertex. This is because the point cloud is not spherical at the bottom due to hardware constraints (Table 1). The scaled trajectory is initially determined based on the enclosing sphere (Figure 2b) which can lead to certain trajectories with high inclination to go outside the bottom of the workspace (Figure 3). This process is repeated until the trajectory



(a) Low Inclined Trajectory

(b) High Inclined Trajectory

Figure 3: Reduction of scaling factor

is completely inside the workspace, and the final scaling factor is obtained (η_F) as shown in an example case in Figure 3b. Algorithm 1 describes the method to find the best-fit scaling factor. In this algorithm, the `generate_trajectory` function generates the trajectory by scaling the earth constants G, M, and the semi-major axis ($\tilde{a} = \eta a$) with the new scaling factor.

Algorithm 1: Scaling Factor Reduction

Result: η_F
η_0, waypoints;
while *lower vertex is outside mesh* **do**
    $\eta = (1 - \lambda)\eta$
    waypoints = `generate_trajectory`(η)
    shift = ||old upper vertex - new upper vertex||
    waypoints = waypoints + shift
end
$\eta_F = \eta$

CLOSED-LOOP INVERSE KINEMATICS (CLIK) CONTROL

The desired trajectory for a robotic manipulator can be planned in the task space or the joint space. The task space is used to define the position and/or orientation of the end-effector, whereas the joint space is used to define the exact configuration of all the joints. In a redundant manipulator, the number of joint space variables is more than the number of variables used to define the task space. In this study, only the 3D position of the end-effector ($\boldsymbol{x}_e$) can be defined because the end-effector is the tip of the RA, and the number of joint space variables for the proposed redundant hybrid manipulator is equal to the generalized coordinates. Henceforth, joints are used to refer to the generalized coordinates. The degree of redundancy (δ_r) for the proposed platform can therefore be defined as $\delta_r = 9 - 3 = 6$. This makes the inverse kinematics of the complete platform a difficult task as there can be a large number of solutions for the joint configuration that can lead to a particular task-space configuration. The translational velocity for the end-effector can be expressed in terms of the generalized coordinates using a Jacobian matrix $\left(J_e(\boldsymbol{q}) \in \mathcal{R}^{3\times 9}\right)$ as follows:

$$\dot{\boldsymbol{x}}_e = J_e(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{52}$$

This equation is known as the differential kinematics equation that linearly maps the joint-space to the task-space and is used to solve for the joint velocities for a desired end-effector position as follows:

$$\dot{\boldsymbol{q}} = J_e^\dagger(\boldsymbol{q})\dot{\boldsymbol{x}}_e \tag{53}$$

where

$$J_e^\dagger(\boldsymbol{q}) = J_e^T \left(J_e J_e^T\right)^{-1} \tag{54}$$

To generate the desired velocities for the generalized coordinates to achieve the desired task-space velocity is known as redundancy resolution at velocity level.[24] Equation (52) can then be differentiated to obtain the following differential relation for accelerations:

$$\ddot{\boldsymbol{x}}_e = \dot{J}_e\dot{\boldsymbol{q}} + J_e\ddot{\boldsymbol{q}} \tag{55}$$

Using this relationship, the joint accelerations can be expressed in terms of the end-effector acceleration as follows:

$$\ddot{\boldsymbol{q}} = J_e^\dagger\left(\ddot{\boldsymbol{x}}_e - \dot{J}_e\dot{\boldsymbol{q}}\right) \tag{56}$$

Equation (56) can be expressed as a control law in order to track the desired trajectory obtained from Eq. (49) as follows:

$$\ddot{\boldsymbol{q}} \equiv \ddot{\boldsymbol{q}}_P = J_e^\dagger\left(\ddot{\boldsymbol{r}} + K_D\dot{\boldsymbol{e}} + K_P\boldsymbol{e} - \dot{J}_e\dot{\boldsymbol{q}}\right) \tag{57}$$

where $\boldsymbol{e} = \boldsymbol{r} - \boldsymbol{x}_e$, $\boldsymbol{r}$ is the desired scaled-down orbital trajectory, and K_P and K_D are the proportional and derivative gains in order to drive $\boldsymbol{x}_e$ to $\boldsymbol{r}$. Now, since the desired orbital trajectory only contains the position and velocity information, it is assumed that $\ddot{\boldsymbol{x}}_d = \boldsymbol{0}$. This represents the pseudo-inverse control law. It allows the tracking of a task-space reference trajectory by using the pseudo-inverse of the Jacobian matrix. There are certain disadvantages of using this control law by itself as it does not avoid singularities. Furthermore, secondary tasks, such as collision avoidance and joint-limit avoidance, cannot be achieved using the redundancy of the manipulator.

CLIK control is a technique that offers to resolve the kinematic redundancy in order to follow a desired task-space trajectory as the primary task with the secondary task being the avoidance of the joint-limits using null-space optimization.[24,25] Taking into account the secondary task, the CLIK control law at the acceleration level is as written

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_P + \ddot{\boldsymbol{q}}_S \tag{58}$$

where $\ddot{\boldsymbol{q}}_P$ is the primary solution of the control obtained from Eq. (57) that is used to achieve the desired task-space trajectory. To solve the secondary task, in this case, the avoidance of joint limits, an additional joint acceleration vector ($\ddot{\boldsymbol{q}}_S$) is added to the primary solution. The additional joint accelerations are obtained from an arbitrary vector (Φ_N) that belongs to the null-space of J_e and is known as the null-space velocity. The null-space velocity error ($\dot{e}_N$) is determined by projecting the joint-velocities to the null-space using $(I_{9\times9} - J_e^\dagger J_e)$ as follows:

$$\dot{\boldsymbol{e}}_N = \left(I_{9\times9} - J_e^\dagger J_e\right)\left(\Phi_N - \dot{\boldsymbol{q}}\right) \tag{59}$$

To ensure that the joint velocities converge to the desired null-space velocity (i.e. $\dot{\boldsymbol{e}}_N \to \boldsymbol{0}$), the additional acceleration required to achieve the secondary task is defined as

$$\ddot{\boldsymbol{q}}_S = (I - J_e^\dagger J_e)(\dot{\Phi}_N + K_N\dot{\boldsymbol{e}}_N) - (J_e^\dagger\dot{J}_e J_e^\dagger + \dot{J}^\dagger)J_e(\Phi_N - \dot{\boldsymbol{q}}) \tag{60}$$

where K_N is the positive-definite null-space gain that drives the null-space velocity error to zero. The aforementioned control law ensures that $||\dot{\boldsymbol{e}}_N||$ goes to zero, monotonically, and this can be proven by defining the Lyapunov function as $V = \frac{1}{2}||\dot{\boldsymbol{e}}_N||$. The null-space velocity can now be defined in a manner to achieve the desired sub-task, that is the avoidance of joint limits. Here, Φ_N can be defined as the negative gradient of the

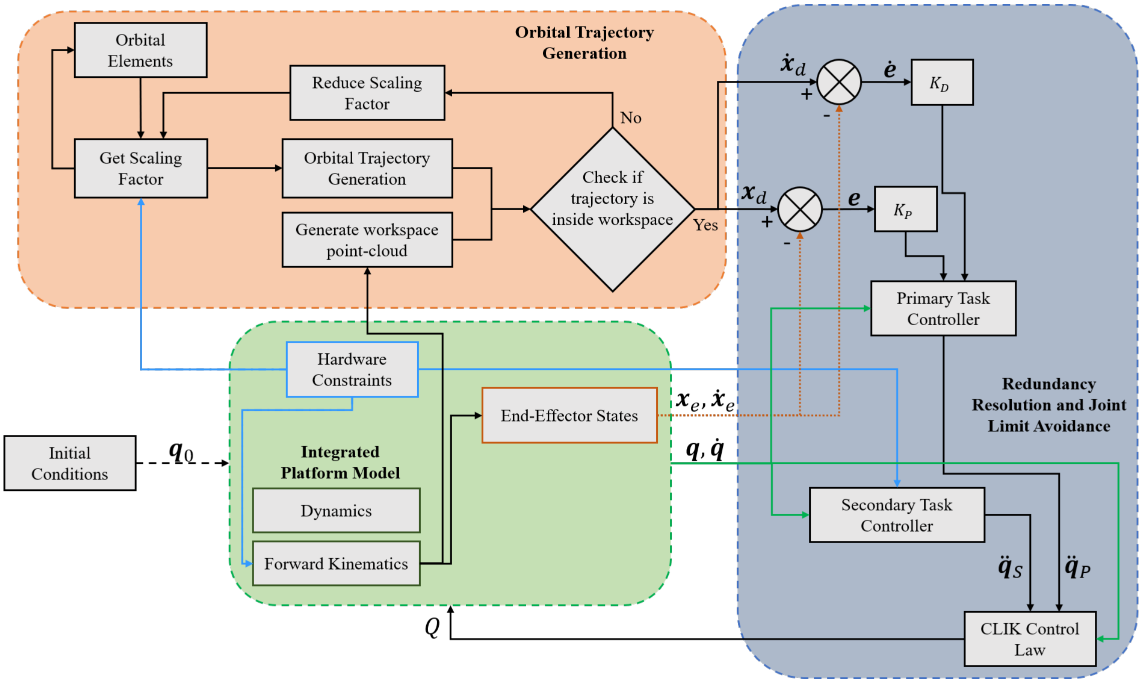


Figure 4: Proposed Approach

cost function (ϕ_N) that needs to be minimized to keep all the generalized coordinates around the middle of their specified mechanical limits. This allows for a realistic control based on the available hardware. Here, the cost function is given as

$$\phi_N = \frac{1}{9}\sum_{i=1}^{9}\left(\frac{q_i - \bar{q}_i}{\max(q_i) - \min(q_i)}\right)^2 \tag{61}$$

where $\bar{q}_i$ denotes the middle value of the generalized coordinates range. Therefore, Φ_N can now be expressed as

$$\Phi_N = -K\frac{\partial \phi_N}{\partial \boldsymbol{q}} \tag{62}$$

where $K \in \mathcal{R}^{9\times 9}$ denotes the gain matrix that can be understood as the freedom of movement for each generalized coordinate around $\bar{q}_i$.

Figure 4 depicts the simultaneous approach for the trajectory generation, scaling algorithm, and the CLIK control law. CLIK control using the pseudoinverse of the Jacobian is an asymptotically stable control and is proven using the Lyapunov stability theorem. The proof is not mentioned here due to the limitation of the space.

SIMULATION STUDY

The numerical simulation method can be summarized using Figure 4. The initial conditions are defined along with the simulation parameters for the integrated platform. Then, the scaling factor is determined based on the hardware constraints of the integrated platform. This is used to scale down the orbital trajectory and generate the reference waypoints for the end-effector of the integrated platform. Then, CLIK control is applied to track the reference trajectory.

Simulation Parameters

Three satellite orbits - SUPERBIRD 8*, TURKSAT 5A†, and Interstellar Boundary Explorer (IBEX)‡ - are considered, and the corresponding orbital elements and scaling factors are listed in Table 2. The orbital trajectories are chosen on the basis of their eccentricity - Low (LeO), Medium (MeO), and High (HeO) elliptical orbit. These must not be confused with the popular terminology LEO, MEO, and HEO, which is instead categorized on the basis of semi-major axes as Low, Medium, and High Earth orbits.

Table 2: Orbital Parameters

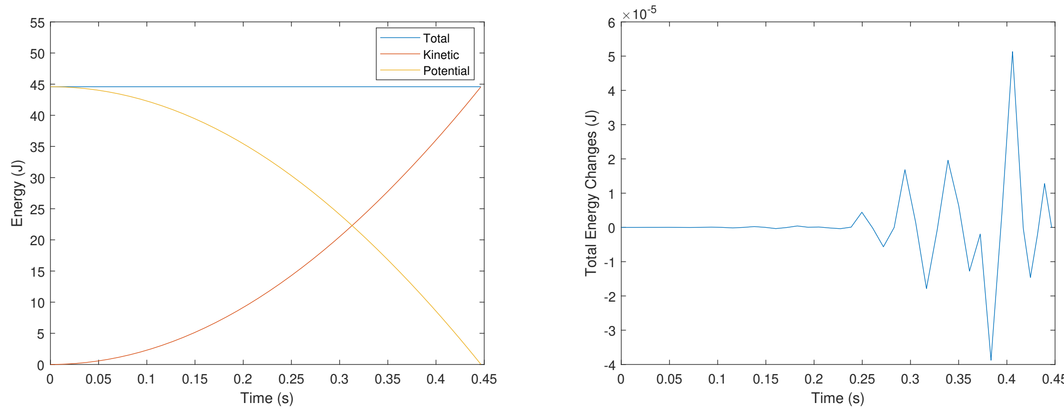
Description	Unit	LeO	MeO	HeO
Satellite in orbit	-	SUPERBIRD	TURKSAT 5A	IBEX
a	km	35793.7	42456	202811
$\tilde{a}$	km	0.002	0.002	0.02
e	-	7×10^{-5}	0.38	0.66
i	deg	0.038	2.824	26.0179
ω	deg	114.448	201.283	22.573
Ω	deg	260.67	315.87	93.95
ν_0	deg	146.125	21.992	0
η	-	5.5876×10^{-8}	4.7108×10^{-8}	9.8614×10^{-9}

Table 3: Parameters for the Integrated Platform

Description		Value	Unit
r_B	Fixed-base radius	1.5	m
r_P	Moving platform radius	0.75	m
m_P	Moving platform mass	1.43	kg
m_{cylinder}	Cylinder mass	0.39	kg
m_{piston}	Piston mass	0.39	kg
c_{cylinder}	Cylinder center of mass	0.5	m
c_{piston}	Piston center of mass	0.5	m
I_P	Moving platform MOI	diag(0.2,0.2,0.4)	kg.m^2
I_{cylinder}	Cylinder MOI	diag(0,0.1,0.1)	kg.m^2
I_{piston}	Piston MOI	diag(0,0.1,0.1)	kg.m^2
m_1	Mass of RA link-1	0.39	kg
l_1	Length of RA link-1	1	m
m_2	Mass of RA link-2	0.39	kg
l_2	Length of RA link-2	1	m
$\boldsymbol{g}$	Gravity vector	$\begin{bmatrix} 0 & 0 & -9.8 \end{bmatrix}^T$	m/s^2
K_P	Proportional gain - primary task	10	-
K_D	Derivative gain - primary task	10	-
K_N	Null-Space gain	10	-
	Cost function gain matrix: LeO	$\text{diag}(25, 35, 15, 50, 40, 70, 300, 5, 5)$	-
K	Cost function gain matrix: MeO	$\text{diag}(25, 25, 5, 25, 25, 5, 50, 5, 5)$	-
	Cost function gain matrix: HeO	$\text{diag}(25, 25, 5, 100, 100, 5, 100, 25, 25)$	-
T_s	Simulation time	150	s

The simulation parameters for the proposed platform are given in Table 3. The SP parameters are obtained based on some previous studies,[26] and the mass and length of the RA are assumed to be the same as that of the

*https://en.wikipedia.org/wiki/Superbird-B3
†https://en.wikipedia.org/wiki/T%C3%BCrksat_5A
‡https://www.nasa.gov/mission_pages/ibex/index.html



(a) Energy vs Time

(b) Total Energy Changes vs Time

Figure 5: Energy Plots

limbs of the SP. The scaled-down time periods (T_f) for the three trajectories are 15.9360 seconds for LeO, 18.9021 seconds for MeO, and 90.2950 seconds for HeO. These values are very small especially for LeO and MeO, which can lead to a large amount of forces and torques required to accurately track the trajectory. Therefore, the simulation final time is chosen as 150 seconds. The initial position and orientation of the SP is kept as the same for the three scenarios as follows:

$$\chi_0 = \begin{bmatrix} 0\text{ m}, & 0\text{ m}, & 1\text{ m}, & 0\text{ deg}, & 0\text{ deg}, & 0\text{ deg} \end{bmatrix}^T \tag{63}$$

and the initial RA joint angles are taken as $\begin{bmatrix} 0 & 180 & 0 \end{bmatrix}^T$ deg for LeO & MeO, and $\begin{bmatrix} 0 & 120 & 0 \end{bmatrix}^T$ deg for HeO. The hardware constraints that are to be used for the integrated platform are listed in Table 1. These are selected based on some basic analysis of some of the popular products available in the market.*†

Results and Discussion

Figure 5 shows the energy plot of the integrated platform and the changes in total energy over time under the influence of gravitational forces only without any control forces or torques applied to the system. It is evident from the figure that the total energy remains constant over time with minor changes caused by numerical integration errors. This can be used to conclude that the dynamic modeling is correct and consistent with the intrinsic properties of the robot equations of motion as aforementioned.

Figures 6-8 depict the tracking of the three orbital trajectories by the proposed platform. It is observed that the platform starts from its initial position and then adapts to the reference trajectory successfully.

Figure 9 shows the absolute position and velocity errors on a log scale. It is shown that the proposed platform quickly adjusts to the reference trajectory, and the error values are low after some initial errors. It must be noted that since the tolerance chosen for numerical simulation was 10^{-3}, an average error value of less than 10^{-3} is reasonable and is evidence of a good tracking performance.

Furthermore, the time histories of the generalized coordinates for the three scenarios are shown in Figures 10-12. It is evident from these results that the proposed platform can efficiently use the redundant DOF in order to easily track the three orbital trajectories. It can also be observed that the generalized coordinates stay within the specified limits according to the control law and specified constraints. In order to keep the RA joint angles within limits, appropriate SP motion is generated. This is especially evident in the case of LeO trajectory tracking as it can be seen in Figure 10 that the full capability of the SP is used in order to accurately

*https://www.pi-usa.us/en/products/6-axis-hexapods-parallel-positioners/
†https://www.trossenrobotics.com/robotic-arms.aspx

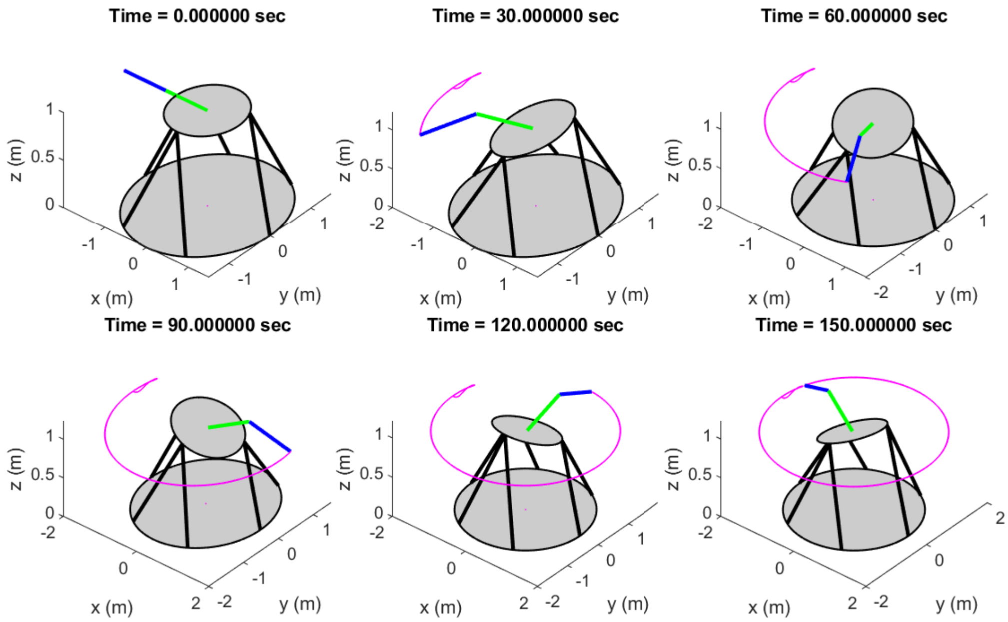


Figure 6: LeO Trajectory Tracking

track the trajectory while avoiding the joint limits. In the cases of MeO and HeO, it can be seen that the SP orientation and the RA joint angles change in a cooperative manner. This highlights the vast capability of the proposed platform in properly utilizing its redundant DOF.

Figure 13 shows the forces applied in the SP linear actuators and the torques applied at the RA joints. There are noticeable oscillations primarily in the SP actuator forces which can be understood as corrective forces generated in order to maintain accuracy. This is also evident from the behavior of system states (Figures 10-12) as no such oscillations are observed in the state response. This leads to the conclusion that the forces and torques show this corrective behavior due to a sub-optimal choice of control gains (K_P, K_D, and K_N). Furthermore, the primary task controller is based on a PD control technique which usually presents a steady state error leading to the observed response. It can also be observed that the forces and torques do not present large and sudden changes, which is an advantageous characteristic of the proposed platform. It can be concluded that a detailed modeling and tuning of PID controllers can lead to a better and more smooth control response.

CONCLUSION

In this study, a unique robotic platform developed using a 6 DOF Stewart platform and a 3 DOF robotic arm is proposed. The dynamic and kinematics modeling is presented using the Lagrangian method, and the derived dynamic model is verified using the characteristic properties of robot equations and the energy method. Furthermore, a Closed-Loop Inverse Kinematics (CLIK) control law is developed to track reference trajectories while modeling the constraints and the redundancies. The reference trajectories are chosen to be scaled-down orbital trajectories based on different eccentricity values. A method to scale down the trajectories is presented using computational algorithms. The results show that the proposed platform is able to track scaled-down orbital trajectories subject to the specified constraints on its translational and rotational movements. The results for this study will be used to develop a more holistic control law that takes into consideration a number of factors, such as constraints on maximum velocities of linear actuation, saturation of forces and torques, actuator resolution in addition to the constraints on generalized coordinates. Furthermore, in future studies, the end-effector of the platform will be used to represent 6 DOF spacecraft motion.

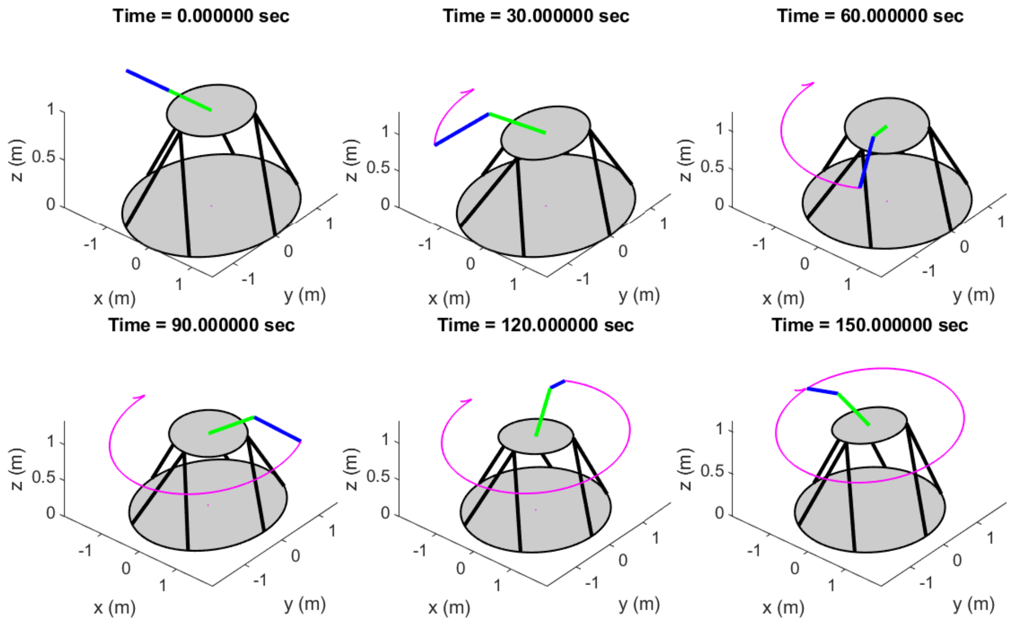


Figure 7: MeO Trajectory Tracking

REFERENCES

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

[2] Y. Liu, Z. Xie, B. Wang, H. Liu, Z. Jing, C. Huang, and S. Bao, "A practical detection of non-cooperative satellite based on ellipse fitting," *2016 IEEE International Conference on Mechatronics and Automation*, 2016, pp. 1541–1546, 10.1109/ICMA.2016.7558793.

[3] 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.

[4] 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.

[5] 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.

[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] M. Wilde, B. Kaplinger, T. Go, H. Gutierrez, and D. Kirk, "ORION: A simulation environment for spacecraft formation flight, capture, and orbital robotics," *2016 IEEE Aerospace Conference*, 2016, pp. 1–14, 10.1109/AERO.2016.7500575.

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

[9] J. Davis, J. Doebbler, K. Daugherty, J. Junkins, and J. Valasek, "Aerospace Vehicle Motion Emulation Using Omni-Directional Mobile Platform," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, 10.2514/6.2007-6325.

[10] S. Dubowsky, W. Durfee, T. Corrigan, A. Kuklinski, and U. Muller, "A laboratory test bed for space robotics: the VES II," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, Vol. 3, 1994, pp. 1562–1569 vol.3, 10.1109/IROS.1994.407647.

[11] 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.

[12] H. Fujii, K. Uchiyama, H. Yoneoka, and T. Maruyama, "Ground-based simulation of space manipulators using test bed with suspension system," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, 1996, pp. 985–991, 10.2514/3.21736.

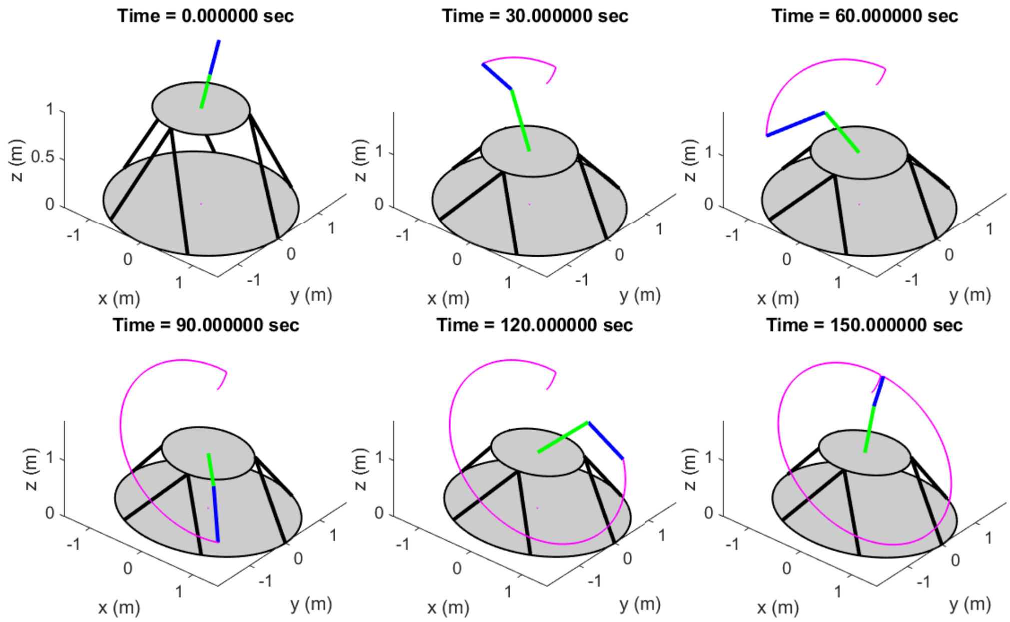


Figure 8: HeO Trajectory Tracking

[13] B. Hockman, A. Frick, I. A. D. Nesnas, and M. Pavone, *Design, Control, and Experimentation of Internally-Actuated Rovers for the Exploration of Low-Gravity Planetary Bodies*, p. 283–298. Springer Tracts in Advanced Robotics, 2016, 10.1007/978-3-319-27702-8_19.

[14] K. Hunt, *Kinematic Geometry of Mechanism*. Oxford University Press, 1978.

[15] B. Dasgupta and T. Mruthyunjaya, "The Stewart Platform Manipulator: A Review," *Mechanism and Machine Theory*, Vol. 35, Jan. 2000, pp. 15–40, 10.1016/S0094-114X(99)00006-3.

[16] R. Oftadeh, M. M. Aref, and H. D. Taghirad, "Explicit dynamics formulation of Stewart-Gough platform: A Newton-Euler approach," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2772–2777.

[17] P. From, J. Gravdahl, and K. Pettersen, *Properties of the Dynamic Equations in Matrix Form*, pp. 285–305. Advances in Industrial Control, 01 2014, 10.1007/978-1-4471-5463-1-9.

[18] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: A tutorial," *Automatica*, Vol. 25, No. 6, 1989, p. 877–888, 10.1016/0005-1098(89)90054-x.

[19] D. Koks, "Changing Coordinates in the Context of Orbital Mechanics," tech. rep., Cyber and Electronic Warfare Division, Defence Science and Technology Group, Department of Defence, Australian Government, January 2017.

[20] S. Oghim, H. Leeghim, and D. Kim, "Real-time spacecraft intercept strategy on J2-perturbed orbits," *Advances in Space Research*, Vol. 63, No. 2, 2019, p. 1007–1016, 10.1016/j.asr.2018.09.023.

[21] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," *New Results and New Trends in Computer Science*, 1991.

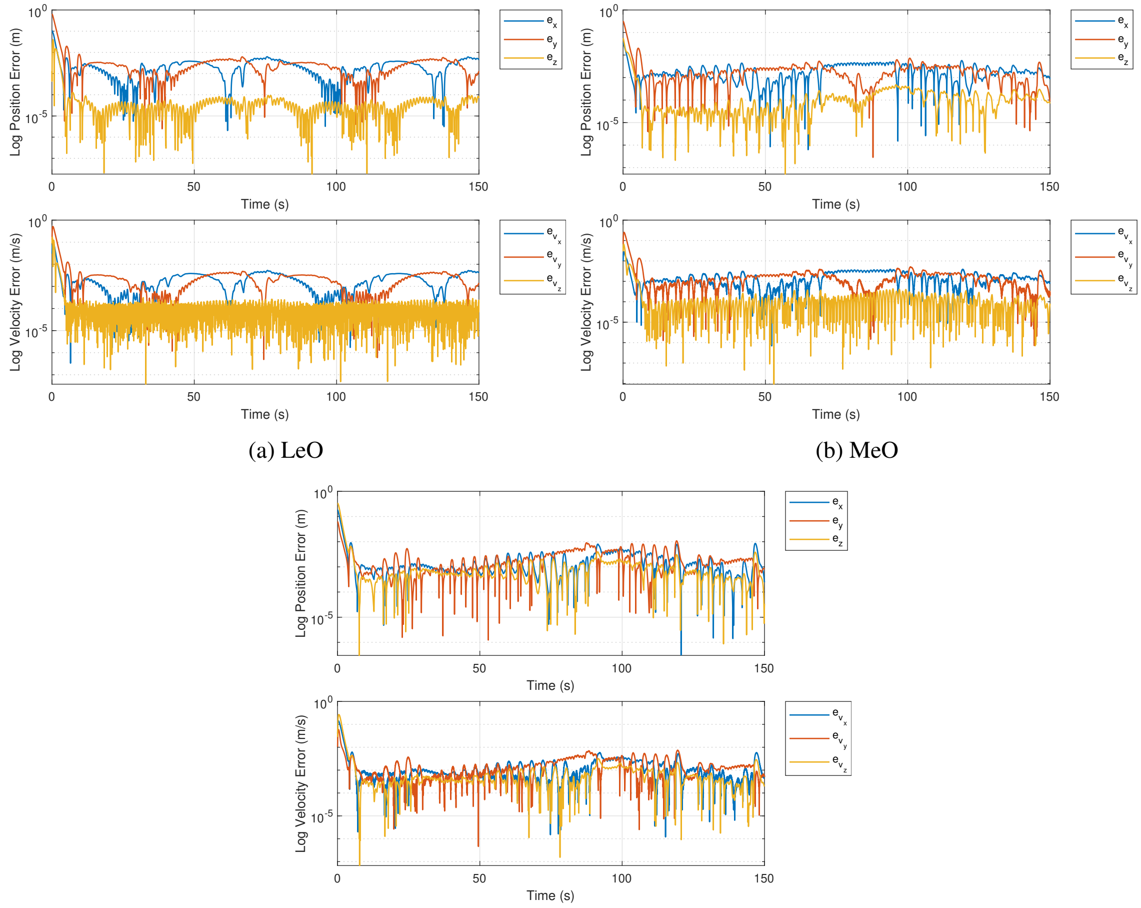
[22] P. L. George and F. Hermeline, "Delaunay's mesh of a convex polyhedron in dimensiond. application to arbitrary polyhedra," *International Journal for Numerical Methods in Engineering*, Vol. 33, No. 5, 1992, p. 975–995, 10.1002/nme.1620330507.

[23] R. A. Dwyer, "A faster divide-and-conquer algorithm for constructing delaunay triangulations," *Algorithmica*, Vol. 2, No. 1-4, 1987, p. 137–151, 10.1007/bf01840356.

[24] D. Subedi, "Kinematic Control of Redundant Manipulators," *Thesis*, July 2017.

[25] J. Wang, Y. Li, and X. Zhao, "Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm," *International Journal of Advanced Robotic Systems*, Vol. 7, No. 4, 2010, p. 37, 10.5772/10495.

[26] A. M. Lopes, "Dynamic modeling of a Stewart platform using the generalized momentum approach," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, No. 8, 2009, p. 3389–3401, 10.1016/j.cnsns.2009.01.001.



(a) LeO

(b) MeO

(c) HeO

Figure 9: Trajectory Errors

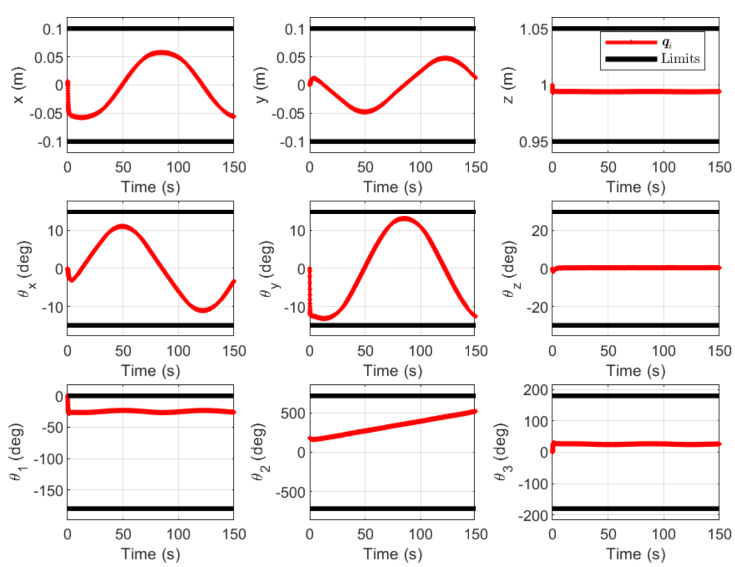


Figure 10: Time Histories of Generalized Coordinates: LeO

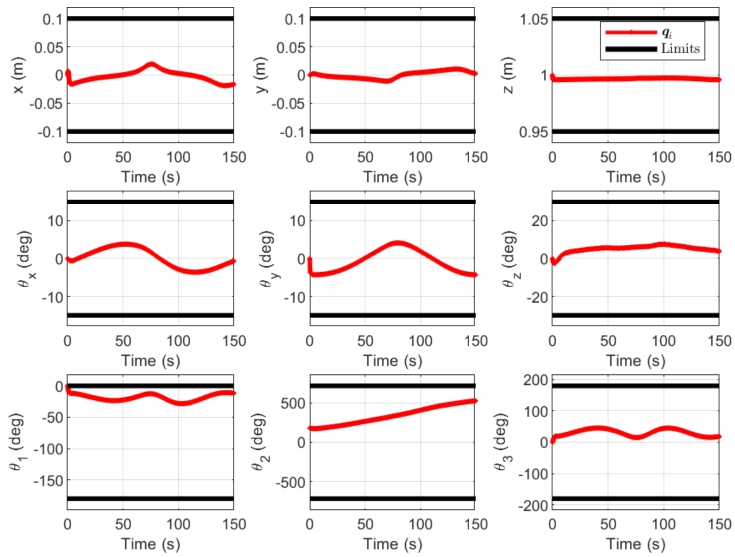


Figure 11: Time Histories of Generalized Coordinates: MeO

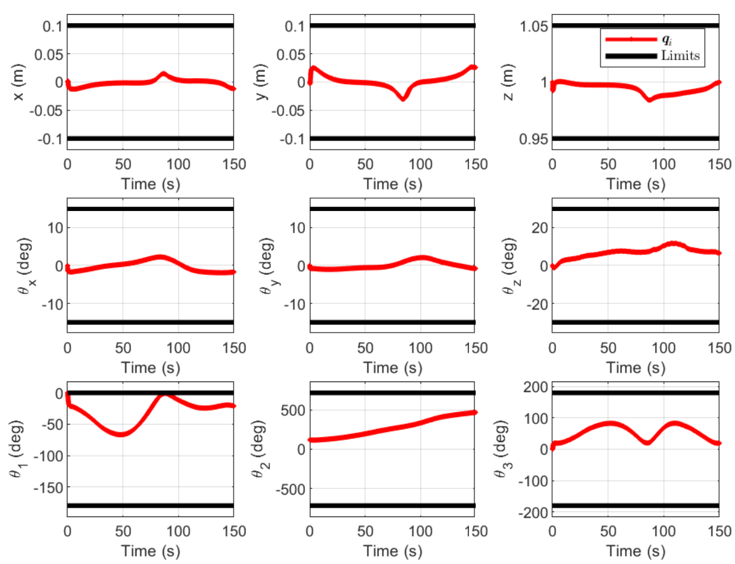
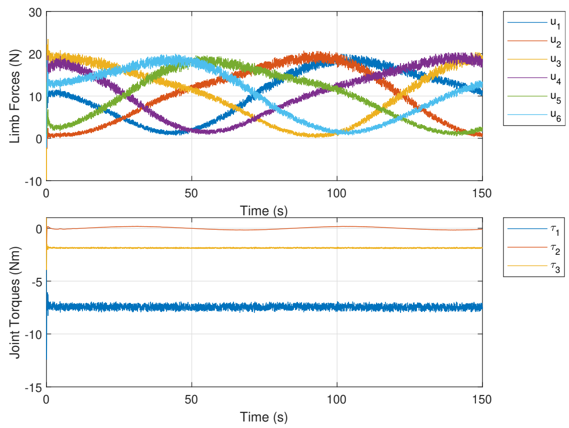
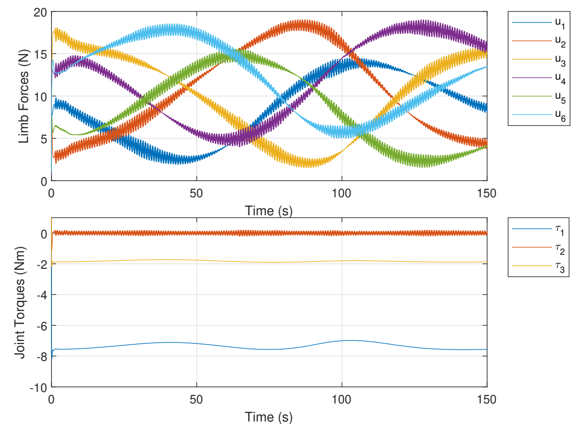


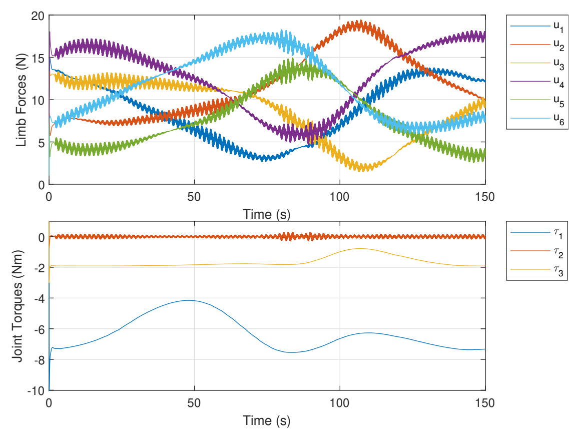
Figure 12: Time Histories of Generalized Coordinates: HeO



(a) LeO



(b) MeO



(c) HeO

Figure 13: Forces and Torques