International Journal of Aeronautical and Space Sciences
https://doi.org/10.1007/s42405-024-00818-1

ORIGINAL PAPER

Relative Motion Transformation for Ground Robotics Test Bed in Rendezvous/Docking Simulation

Jaewook Lim[1] · Sejun Park[2] · Kyung-Mi Park[1] · Donghoon Son[2] · Dong-Hyun Cho[1]

Received: 16 April 2024 / Revised: 11 August 2024 / Accepted: 26 August 2024

Abstract

As missions utilizing multiple satellites become more prevalent, the significance of rendezvous and docking (RVD) technology has markedly increased compared to the previous focus on single-satellite missions. Developed RVD technology can be validated beforehand using robot-based ground test beds, effectively reducing risks and costs. However, predicting the movements of robots to simulate satellite motion and replicating orbital maneuvers in confined spaces can be challenging. Therefore, this paper aims to address these issues and enhance ground-based testing methodologies, including techniques such as transforming relative motion, minimizing satellite's rotational motion, and trajectory planning for precise replication of satellite kinematics. These approaches seem to strengthen the technological foundations essential for successful space missions and are anticipated to drive innovative progress in future space exploration and satellite operations.

Keywords Relative motion transformation · Satellite trajectory planning · Robot testbed · Rendezvous/docking simulation

List of Symbols

$\boldsymbol{p}$	Position vector
t	Discrete time equivalent to the robot's update rate
t_s	Discrete time equivalent to the satellite's update rate
T	Time interval corresponding to the satellite's update rate
$\boldsymbol{I_n}$	$n \times n$ Identity matrix
$\boldsymbol{q}_d$	Desired joint configuration of manipulator
q_i	Robot's $i\text{-}th$ joint angle or position
$\boldsymbol{q}$	Quaternion (scalar first)
$\boldsymbol{\omega}$	Angular velocity of the satellite
$\dot{\boldsymbol{\omega}}$	Angular acceleration of the satellite
$\widehat{\boldsymbol{\omega}}$	Rotation axis
$\boldsymbol{\varphi}$	Rotation vector ($\boldsymbol{\varphi} = \varphi\widehat{\boldsymbol{\omega}}$)
$\boldsymbol{x}$	Position in section 3
$\boldsymbol{v}$	Velocity
$\boldsymbol{a}$	Acceleration
$\Delta\boldsymbol{u}$	Algebraic difference of arbitrary vector $\boldsymbol{u}$
$\widehat{\boldsymbol{u}}$	Normalized arbitrary vector $\boldsymbol{u}$
$(\boldsymbol{u})_t$	Arbitrary vector $\boldsymbol{u}$ defined at discrete time t
$\boldsymbol{u}^{\times}$	Skew symmetric matrix of arbitrary vector $\boldsymbol{u}$
${}^{a}\boldsymbol{u}$	Arbitrary vector $\boldsymbol{u}$ expressed in $\{a\}$ frame
${}^{a}\boldsymbol{R}_b$	Rotation matrix from $\{b\}$ to $\{a\}$ (${}^{b}\boldsymbol{u}$ to ${}^{a}\boldsymbol{u}$)
${}^{a}\boldsymbol{T}_b$	Homogeneous transform matrix form $\{b\}$ to $\{a\}$

1 Introduction

Rendezvous/docking is a critical technology in performing missions between space vehicles [1]. This technology, which involves aligning the relative motion of two spacecraft in orbit and structurally connecting them, was first demonstrated during NASA's Gemini project in the 1960s, aimed at verifying technology development for lunar exploration missions [2, 3]. Since the successful docking missions between the US Apollo and the Soviet Soyuz spacecraft, the development of rendezvous/docking technology has been actively pursued [4]. Unlike the past, where humans manually controlled the process, there is currently active development of automated rendezvous/docking technologies [5]. Many studies are being conducted to utilize this technology in on-orbit servicing (OOS) applications, emphasizing the necessity of

Communicated by Hyuk Park.

✉ Dong-Hyun Cho
donghyun.cho@pusan.ac.kr

[1] Department of Aerospace Engineering, Pusan National University, Busan, Republic of Korea

[2] Department of Mechanical Engineering, Pusan National University, Busan, Republic of Korea

Published online: 16 September 2024

precise and accurate rendezvous/docking technology development for such purposes [6–8].

The development of rendezvous/docking technology requires performance evaluation through testing before actual operation. Conducting tests in the actual space environment entails significant costs and risks. Therefore, it is necessary to simulate appropriate space environments on the ground. Various testing methods have been used to simulate microgravity environments [9]. The use of air bearings involves suspending objects in mid-air to create a frictionless surface environment for testing [10–12]. While this method allows for relatively easy setup of test environments and long-term testing, it is limited to a 3 degree of freedom (DOF) environment. Limiting the analysis to only 2D and rotational motion makes it challenging to assess performance beyond this scope. The neutral buoyancy method, which utilizes buoyancy in a water environment, allows for simulations of three-dimensional motion. However, it is affected by fluid viscosity, and it is not possible to conduct tests using actual satellites [13]. Another method involves attaching cables to satellite models to simulate docking in three-dimensional space. However, controlling tension and dealing with friction pose challenges in this method [14, 15].

In contrast, utilizing robots for test offers significant advantages in implementing 6 DOF and simulating microgravity conditions. It enables relatively high-precision performance verification [16–19]. A prominent example of a test facility using robots for rendezvous/docking simulation is the EPOS at the German Aerospace Center (DLR) [20, 21]. This facility aims to test the final approach operation phase by attaching satellite bodies to the ends of two robotic arms. It allows dynamic motion in 6 DOF while implementing actual space environment conditions through lighting configurations. The motion of the two satellites is converted into the movement of the robotic arms to verify the satellite's motion, and this facility has been continuously developed over time. Additionally, the US Naval Research Laboratory has constructed a test facility using two robotic arms with 6 DOF to evaluate sensor performance for rendezvous/docking [22]. Furthermore, the Japan Aerospace Exploration Agency (JAXA) is developing the SATDyn (Satellite dynamics test platform for on-orbit servicing technology) facility, utilizing a 6 DOF robotic arm to simulate rendezvous and capturing dynamics [23].

The literature previously mentioned on the robot testbed focuses on topics such as robot arm control, and considerations for facility operation. In contrast, the main purpose of this paper is to transform a wider range of relative motions of satellites into ground-based test facilities with limited reach. To achieve this, as shown in Fig. 1, the "Initial Configuration" phase determines the initial poses of the satellites, and the robot based on the initial pose of the two satellites. In the "Relative Motion Simulation" phase, the relative motion of the satellite is converted into a form that can be accommodated by the ground testbed. Also, trajectory planning is performed to interpolate the satellite's state, which is updated slower than the robot's control frequency. The primary purpose of this phase is to validate that the transformed and planned relative motion can be replicated before operating the actual robotics testbed, which is directly related to the subject of this paper. Finally, the validated relative motion simulation will be emulated on an actual testbed in real time, with the satellite's state being updated based on control inputs received from the satellite's GNC (Guidance, Navigation, and Control) PC, functioning as a feedback controller. However, the content of this phase is part of our future work and is not covered in this paper.

In relation to these points, the primary contribution of this paper is presenting a method to simulate relative motion along a predefined straight line, rather than expanding the testbed to accommodate extensive relative motion of satellites. The main strategy involves aligning the Line-of-Sight (LOS) between two satellites with a reference direction during the "Initial Configuration" phase. This strategy is maintained during the "Relative Motion Simulation" phase, which is based on discrete position and attitude data calculated using the satellite's propagator.

However, even with the transformed relative motion of satellites, issues such as joint range limitations, kinematic singularities, and collisions can arise. Considerable research has already been conducted to address these problems. For example, by utilizing the null space of a redundant robot, it is possible to transfer the payload's movement to a rail or find a posture within the joint range [24]. In [25], the RRT-connect method is used to find a configuration that can avoid collisions in a static state, while [26] introduces a method for dynamically adjusting the robot's path to avoid obstacles in real-time. In [27], a method for avoiding singularities in dynamic situations using a task-priority approach is introduced, which can be applied to ensure stability in systems simulating the relative motion of large-scale satellites. Although these aspects are reflected in "Adjusting for collision-free configuration" and "Joint motion optimization" in Fig. 1, they are not the focus of this paper. Instead, as a second contribution, this study proposes a method to reduce the attitude changes during simulation compared to the initial attitude of the satellites, thereby passively mitigating the movement of the robot's end effector.

Even when considering the transformed relative motion and its rotation redistribution, the propagation period of a satellite is much slower than the typical control frequency of a robot. Therefore, the trajectory must be planned so that the robot can appropriately follow the given satellite's state. While a simple point-to-point path planning approach may allow for the simulation of the satellite's position and attitude at specific times, it results in initial and final velocities and

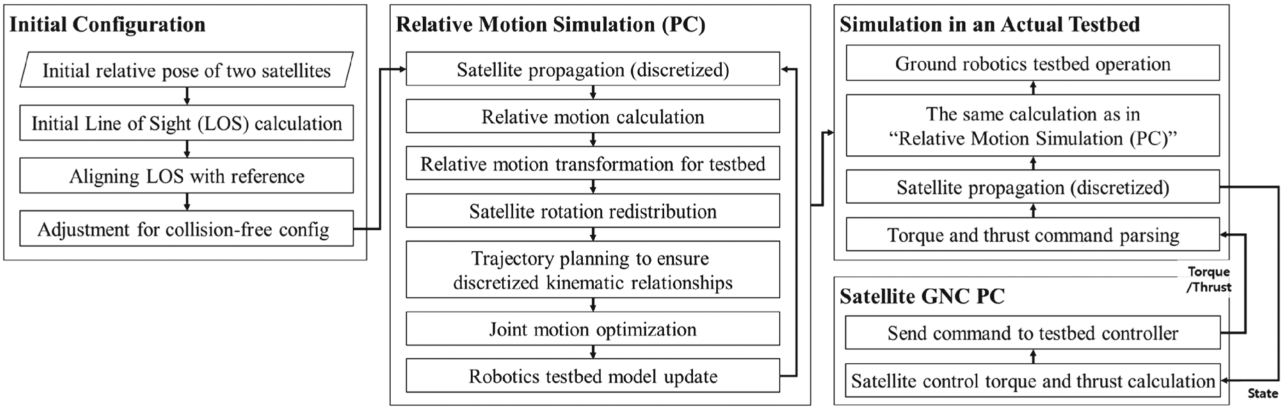

Initial Configuration
Initial relative pose of two satellites
Initial Line of Sight (LOS) calculation
Aligning LOS with reference
Adjustment for collision-free config
Relative Motion Simulation (PC)
Satellite propagation (discretized)
Relative motion calculation
Relative motion transformation for testbed
Satellite rotation redistribution
Trajectory planning to ensure discretized kinematic relationships
Joint motion optimization
Robotics testbed model update
Simulation in an Actual Testbed
Ground robotics testbed operation
The same calculation as in "Relative Motion Simulation (PC)"
Satellite propagation (discretized)
Torque and thrust command parsing
Torque /Thrust
Satellite GNC PC
Send command to testbed controller
Satellite control torque and thrust calculation
State

Fig. 1 Diagram of relative motion transformation

accelerations being zero, failing to ensure kinematical relationships. Estimating relative velocity from sensors on the satellite could lead to issues with the reliability of test results. Therefore, the third contribution of this paper is the trajectory planning, which considers the robot's velocity and acceleration, and also kinematical relationships by using Hermite interpolation employed in STK (System Tool Kit) [28].

Hence, this paper is organized as follows: Sect. 2.1 introduces an overview of the testbed, which is the primary focus of this paper, and the reference frame used for simulations on this testbed. Section 2.2 discusses the transformation that aligns the Line of Sight (LOS) of the two satellites in a predefined direction to simulate the relative motion of satellites, which requires a large workspace, on a smaller ground-based testbed. Section 2.3 introduces rotational redistribution to minimize the rotational motion of individual satellites while maintaining relative motion. Since the transformed and modified relative motion is given as discrete positions, velocities, and orientations, Sect. 3 discusses the path planning that ensures the kinematical relationships of the discretized satellite states while imposing constraints on the robot's velocity and acceleration. Finally, Sect. 4 presents the conclusion.

2 Relative Motion Transformation for Ground Testbed

2.1 Reference Frame

Typically, the position or orientation of a satellite is defined in either the Earth Centered Inertia (ECI) frame or Earth Centered Earth Fixed (ECEF) frame. Alternatively, for rendezvous/docking, the relative navigation solutions used to be calculated in the coordinate system of the Target or Chaser satellite, the former being serviced and the latter providing the service. However, transforming the coordinate systems defined on Earth or satellites to a robotics test bed introduces limitations on simulations due to the limited workspace of robots or collisions caused by facilities. Therefore, there is a need to establish a new coordinate system to facilitate the transformation of the relative motion between the two satellites.

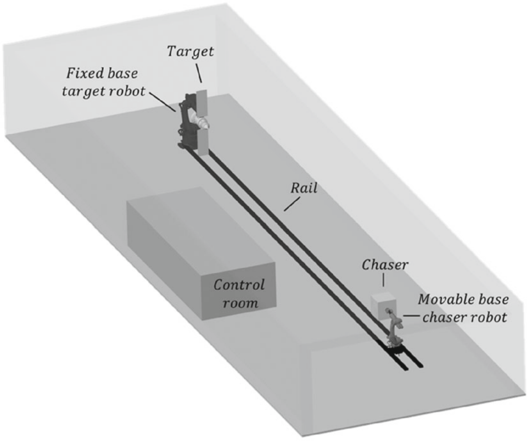


Fig. 2 Robotics ground test bed for rendezvous and docking simulations

Based on the test bed trends outlined in [16–21], the proposed facility comprises a configuration similar to Fig. 2. In this paper, a robot moving on rails is referred to as the Chaser robot, which delivers the Chaser. The Target robot, located on the opposite side, has a fixed base and controls the motion of the Target. The Chaser robot in this research is KR210 R2700, with a payload of 210 kg and a reach of about 2.7 m. The Target robot is KR510 R3100 with a payload of 510 kg and a reach of about 3.1 m. The inertia frame to be used in the test bed can be defined anywhere, but for computational convenience, it is considered to be identical to the base frame $\{TB\}$ of the Target robot, which remains stationary in position and orientation. The initial position and orientation of the Target satellite are defined in $\{TB\}$, while those of the Chaser satellite are derived from the relative navigation solution with respect to the Target satellite.

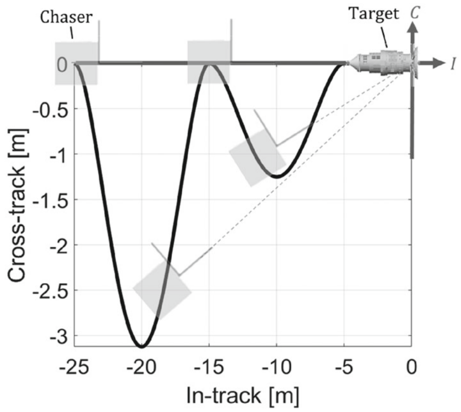


Fig. 3 Relative motion with respect to the RIC frame, expressed in the plane including the in-track and cross-track axes of the Target

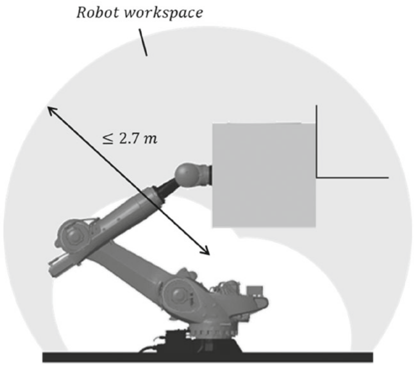


Fig. 4 Example of the workspace of a 6 DOF manipulator

2.2 Projecting Line of Sight in a Specified Direction

The relative motion between two satellites in space is often extensive. For example, if the Chaser's position and orientation profile, in Fig. 3, is specified in the Radial In-track Cross-track (RIC) frame of the Target, aligning the C-axis of the Target with the rail direction of the test bed may not entirely replicate the relative motion using a robot shown in Fig. 4. Here, the robot's maximum reach is less than 2.7 m. Even if the I-axis of the Target is parallel to the rail, it would still not adequately follow the motion in cross-track direction.

In this context, it is evident that the maximum distance between Chaser and Target at each snapshot depicted in Fig. 3 aligns with the LOS direction from Chaser to Target. This indicates that the rail direction capable of achieving the longest distance must align with the LOS. Referring to Fig. 5, this can be achieved by aligning $^{T}\widehat{\boldsymbol{p}}_{C2T}$, the LOS unit vector expressed in the Target's body frame $\{T\}$, with $^{TB}\widehat{\boldsymbol{x}}$, the x-axis of $\{TB\}$. Additionally, to determine the orientation of Chaser and Target in 3D space, at least two axes must be defined. One is $^{T}\widehat{\boldsymbol{p}}_{C2T}$ and the other axis $^{T}\widehat{\boldsymbol{p}}_{r}'$ can be obtained by projecting an arbitrary unit vector $^{T}\widehat{\boldsymbol{p}}_{r}$ expressed in $\{T\}$ onto the plane normal to $^{T}\widehat{\boldsymbol{p}}_{C2T}$. In other words, the attitude of Target with respect to $\{TB\}$, $^{TB}\boldsymbol{R}_{T}$, can be defined as shown in Eq. (1). Once the position of the Target with respect to the base frame $\{TB\}$, denoted as $^{TB}\boldsymbol{p}_{T}$, is determined, the homogeneous transform matrix $^{TB}\boldsymbol{T}_{T}$ defining the initial configuration of the Target can be expressed as shown in Eq. (2). Then, the pose of the Chaser, derived from the relative navigation solution with the Target, represented as $^{T}\boldsymbol{T}_{C}$, can be expressed in $\{TB\}$ as demonstrated in Eq. (3) (see Fig. 6).

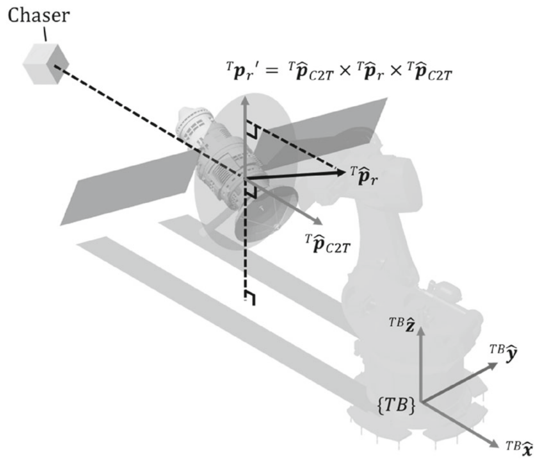

Chaser
$^{T}\boldsymbol{p}_{r}' = {}^{T}\widehat{\boldsymbol{p}}_{C2T} \times {}^{T}\widehat{\boldsymbol{p}}_{r} \times {}^{T}\widehat{\boldsymbol{p}}_{C2T}$
$^{T}\widehat{\boldsymbol{p}}_{r}$
$^{T}\widehat{\boldsymbol{p}}_{C2T}$
$^{TB}\widehat{\boldsymbol{z}}$
$^{TB}\widehat{\boldsymbol{y}}$
$\{TB\}$
$^{TB}\widehat{\boldsymbol{x}}$

Fig. 5 Initial configuration of the Target with respect to $\{\boldsymbol{TB}\}$

$$^{TB}\boldsymbol{R}_{T} = \begin{bmatrix} {}^{T}\widehat{\boldsymbol{p}}_{C2T} & \frac{{}^{T}\boldsymbol{p}_{r}{}' \times {}^{T}\widehat{\boldsymbol{p}}_{C2T}}{\left|{}^{T}\boldsymbol{p}_{r}{}' \times {}^{T}\widehat{\boldsymbol{p}}_{C2T}\right|} & \frac{{}^{T}\boldsymbol{p}_{r}{}'}{\left|{}^{T}\boldsymbol{p}_{r}'\right|} \end{bmatrix}^{T}, \tag{1}$$

$$^{TB}\boldsymbol{T}_{T} = \begin{bmatrix} {}^{TB}\boldsymbol{R}_{T} & {}^{TB}\boldsymbol{p}_{T} \\ 0\;0\;0 & 1 \end{bmatrix}, \tag{2}$$

$$^{TB}\boldsymbol{T}_{C} = {}^{TB}\boldsymbol{T}_{T} \times {}^{T}\boldsymbol{T}_{C}. \tag{3}$$

After the satellites arrangement was determined in the test bed, an offset was simply added to $^{TB}\boldsymbol{p}_{T}$. when changing the position of the satellite pair. If it is necessary to change the orientation of the satellite pair due to issues such as collisions, the two satellites can be rotated by an angle θ_{los} around the $^{T}\widehat{\boldsymbol{p}}_{C2T}$ axis to ensure that the LOS direction remains parallel to the x-axis of $\{TB\}$. This kind of rotation can be converted into a rotation matrix of Eq. (4) and applied to update the initial orientation of the Target by post-multiplying it with Eq. (1), where $\boldsymbol{I}_{3}$ denotes 3 × 3 identity matrix and superscript × represents skew-symmetric matrix.

$$\boldsymbol{R}\left({}^{T}\hat{\boldsymbol{p}}_{C2T},\;\; \theta_{los}\right) = \boldsymbol{I}_{3} + \sin(\theta_{los})\,{}^{T}\hat{\boldsymbol{p}}_{C2T}^{\times} + \{1 - \cos(\theta_{los})\}\left({}^{T}\hat{\boldsymbol{p}}_{C2T}^{\times}\right)^{2} \tag{4}$$

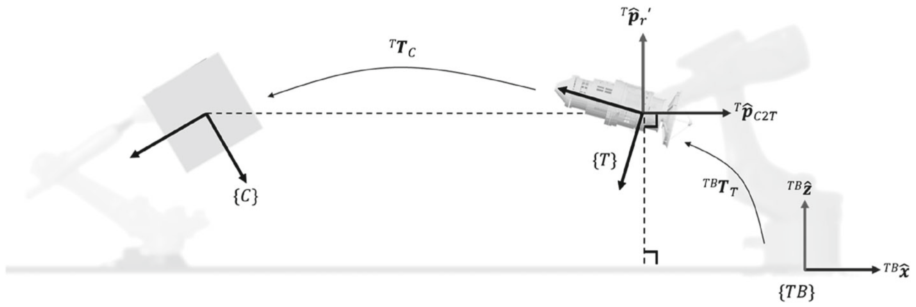
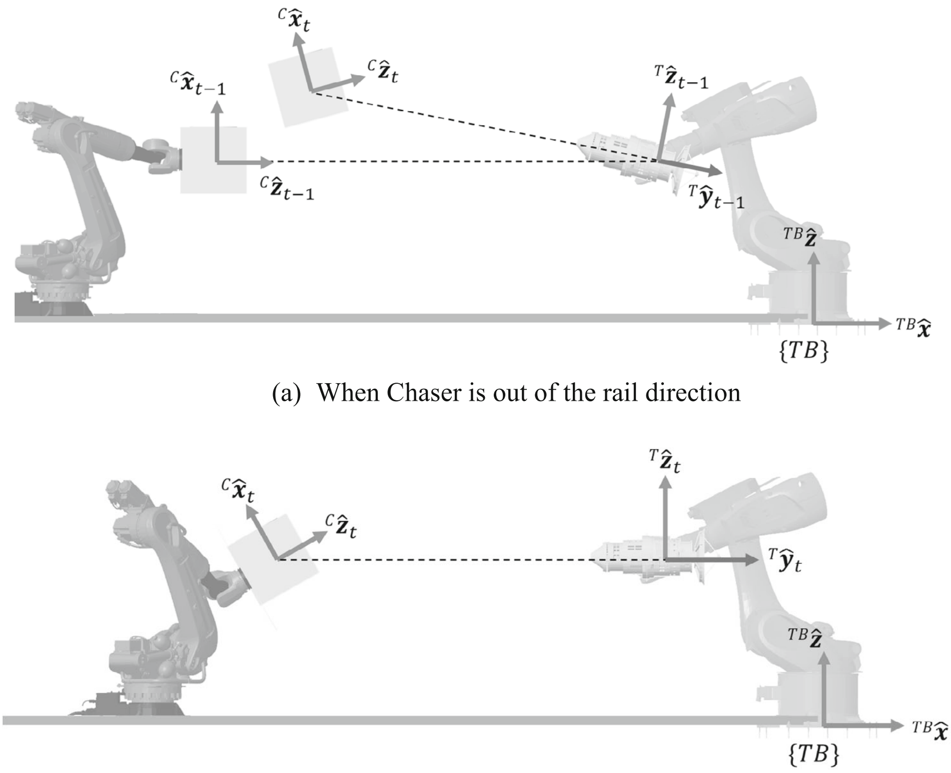

${}^{T}\hat{\boldsymbol{p}}_{r}'$
${}^{T}\boldsymbol{T}_{C}$
${}^{T}\hat{\boldsymbol{p}}_{C2T}$
$\{T\}$
$\{C\}$
${}^{TB}\boldsymbol{T}_{T}$
${}^{TB}\hat{\boldsymbol{z}}$
${}^{TB}\hat{\boldsymbol{x}}$
$\{TB\}$

Fig. 6 The side view of the Chaser and Target mounted on robotics test bed


${}^{C}\hat{\boldsymbol{x}}_{t}$
${}^{C}\hat{\boldsymbol{z}}_{t}$
${}^{C}\hat{\boldsymbol{x}}_{t-1}$
${}^{T}\hat{\boldsymbol{z}}_{t-1}$
${}^{C}\hat{\boldsymbol{z}}_{t-1}$
${}^{T}\hat{\boldsymbol{y}}_{t-1}$
${}^{TB}\hat{\boldsymbol{z}}$
${}^{TB}\hat{\boldsymbol{x}}$
$\{TB\}$
${}^{T}\hat{\boldsymbol{z}}_{t}$
${}^{C}\hat{\boldsymbol{x}}_{t}$
${}^{C}\hat{\boldsymbol{z}}_{t}$
${}^{T}\hat{\boldsymbol{y}}_{t}$
${}^{TB}\hat{\boldsymbol{z}}$
${}^{TB}\hat{\boldsymbol{x}}$
$\{TB\}$

(a) When Chaser is out of the rail direction

(b) When Chaser is aligned with the rail direction

Fig. 7 Iterative Chaser motion parallel to the rail direction (b) or not (a)

After initial configuration, collisions and robot joint limit issues should not occur during the rendezvous simulation. Thus, we adhere to the strategy of aligning the LOS unit vector with the x axis of $\{TB\}$, while using the relative navigation solution to maintain the relative position and attitude. In Fig. 7, the subscript t denotes the axis or vector at discrete time t. As assuming the $\{T\}$ at $t-1$ is the inertia frame, then the LOS unit vector $\left({}^{T}\hat{\boldsymbol{p}}_{C2T}\right)_{t}$ due to the relative motion of the Chaser at t will not be parallel to ${}^{TB}\hat{\boldsymbol{x}}$, as shown in Fig. 7a. To specify the transformation equation for relative motion from the relationship between the two vectors, we align the $\{T\}$ with $\{TB\}$ while preserving the relative pose of the Target and Chaser. This ensures that the relationship between $\left({}^{T}\hat{\boldsymbol{p}}_{C2T}\right)_{t}$ and ${}^{TB}\hat{\boldsymbol{x}}$ becomes identical to the relationship between $\left({}^{T}\hat{\boldsymbol{p}}_{C2T}\right)_{t}$ and ${}^{T}\hat{\boldsymbol{x}}_{t}$. Then, the rotation matrix R to align $\left({}^{T}\hat{\boldsymbol{p}}_{C2T}\right)_{t}$ with ${}^{T}\hat{\boldsymbol{x}}$ can be derived using Rodrigues' rotation formula. When defining the angle between the two vectors as ψ, and the cross product of them as $\hat{\boldsymbol{\omega}}$, $\boldsymbol{R}$ is derived according to Eqs. (5)–(7). Since $\boldsymbol{R}$ and ${}^{TB}\boldsymbol{R}_{T}$ are equivalent, Eq. (7) becomes the rotation matrix between $\{TB\}$ and $\{T\}$ described in Fig. 7b.

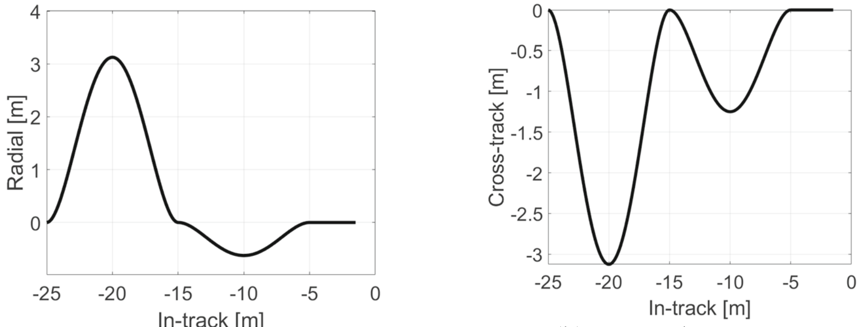
$$\hat{\boldsymbol{\omega}} = {}^{T}\hat{\boldsymbol{p}}_{C2T} \times [1\ 0\ 0]^{T}, \tag{5}$$

$$\psi = \operatorname{acos}\left({}^{T}\hat{\boldsymbol{p}}_{C2T} \cdot [1\ 0\ 0]^{T}\right), \tag{6}$$

$$\boldsymbol{R} = \boldsymbol{R}(\hat{\boldsymbol{\omega}}, \psi), \tag{7}$$

To validate this, trajectories of the Chaser's position and attitude relative to the Target's body frame were generated, and the error from the original relative motion were compared. For clarity, assuming that $\{T\}$ is same with the Target's

Fig. 8 Chaser's in-plane (a) and out-of-plane (b) motion with respect to Target's body frame



(a) In-track vs. Radial

(b) In-track vs. Cross-track

Table 1 Simulation condition of robotics testbed in Simulink

Parameter	Value	Parameter	Value
Chaser robot model	KR210 R2700	Time step	0.004 s
Target robot model	KR510 R3800	All simscape parameters are set to default	
Satellite motion update rate	10 Hz	All measurements are obtained from the Transform Sensor block by SMT	
Robot motion update rate	250 Hz		

RIC frame, the position profile of the Chaser can be represented as Eq. (8), with in-plane and out-of-plane paths illustrated in Fig. 8. The relative attitude is expressed in the form of a rotation matrix as shown in Eq. (9), ensuring that the Chaser's z-axis is constantly directed towards the Target.

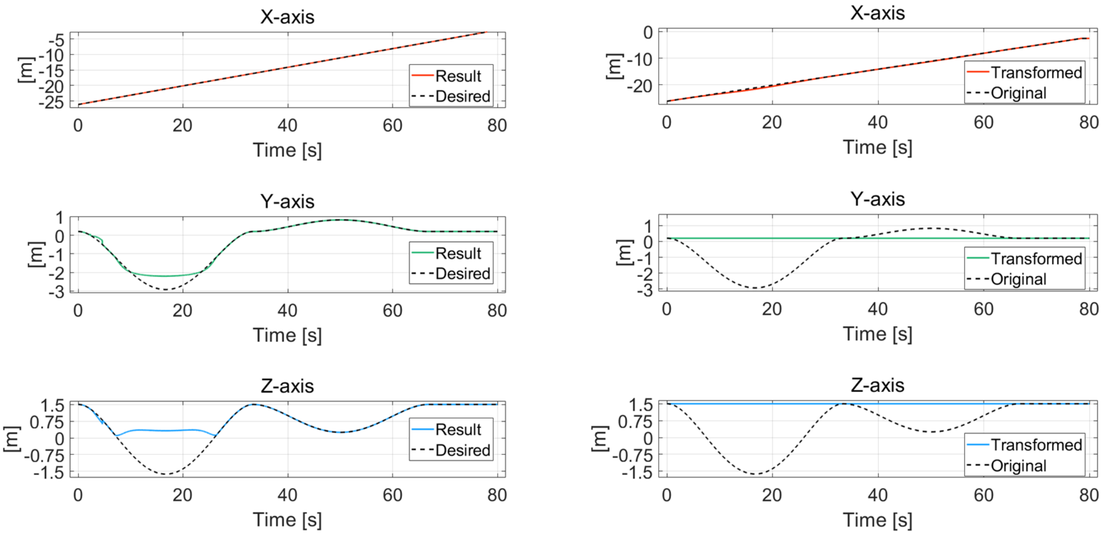
$$
\begin{aligned}
({}^{T}y_{T2C})_t &= y_t = -25 + 0.03t \\
({}^{T}x_{T2C})_t &= \begin{cases} 0.005(y_t + 25)^2(y_t + 15)^2 \ (y_t < -15) \\ -0.001(y_t + 15)^2(y_t + 5)^2 \ (-15 \le y_t < -5) \\ 0 \ (-5 \le y_t\,) \end{cases} \\
({}^{T}z_{T2C})_t &= \begin{cases} -0.005(y_t + 25)^2(y_t + 15)^2 \ (y_t < -15) \\ -0.002(y_t + 15)^2(y_t + 5)^2 \ (-15 \le y_t < -5) \\ 0 \ (-5 \le y_{\,t}) \end{cases},
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
{}^{T}\boldsymbol{R}_C &= \left[{}^{T}\hat{\boldsymbol{x}}\ {}^{T}\hat{\boldsymbol{y}}\ {}^{T}\hat{\boldsymbol{z}} \right] \\
&= \left[{}^{T}\hat{\boldsymbol{y}} \times {}^{T}\hat{\boldsymbol{z}}\ \ {}^{T}\hat{\boldsymbol{z}} \times [1\ 0\ 0]^T \times {}^{T}\hat{\boldsymbol{z}}\ \ -{}^{T}\hat{\boldsymbol{p}}_{T2C} \right].
\end{aligned}
\tag{9}
$$

The testbed in Fig. 2, was created using Simulink's Simscape Multibody Toolbox and Robotics System Toolbox with a rail length of 30 m and the simulation conditions presented in Table 1. As shown in Fig. 9a, KR210 R2700, with a maximum reach of approximately 2.7 m, cannot perfectly simulate the relative motion due to its reach and joint limitations. This figure includes Chaser's trajectory with respect to $\{TB\}$. The legend 'Desired' refers to the generated satellite trajectory and 'Result' represents measurements from Transform Sensor of Simscape Toolbox. In contrast, the relative motion proposed in this Section has the benefit that the only motion is in the direction of x-axis of $\{TB\}$, as shown in Fig. 9b, where 'Original' represents the generated trajectory and 'Transformed' denotes the proposed one. The error of Chaser's trajectory with respect to $\{T\}$ is presented in Fig. 10. The position error and attitude error are both negligible, at the levels of 1e-14 m and 1e-6 rad, respectively. These results indicate that relative motions exceeding the robot's workspace can be adequately simulated along a straight line without displacement along the y and z axes of $\{TB\}$. For a more intuitive comparison, the motion of the robot is visualized in Fig. 11. The first row of Fig. 11 represents the case where $\{T\}$ is the inertial coordinate system with 'Original' trajectory, while the second row represents the case with 'Transformed'.

2.3 Reducing Attitude Change through Rotation Redistribution

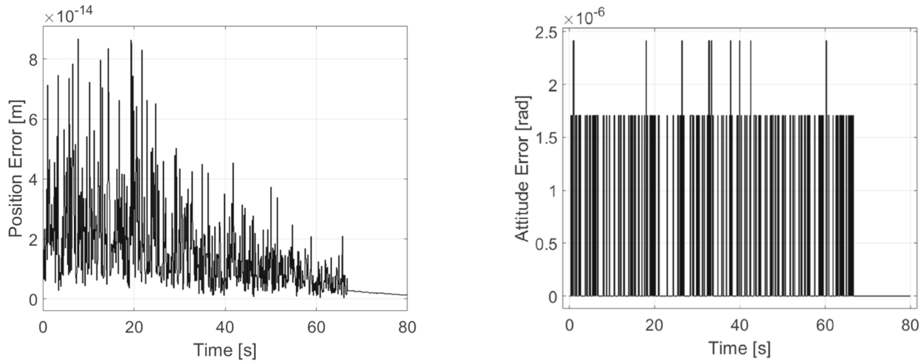
The method of relative motion transformation introduced in Sect. 2.2 aligns the LOS axis with the x-axis of $\{TB\}$. In other words, if the alignment of only one axis is ensured in three dimensions, unpredicted rotation with respect to the LOS can occur. Unnecessary rotations can impose stress on the robot's joints or lead to collisions with facilities. Thus, this section explains the method of reducing or distributing the rotation of satellite by applying a common rotational angle about the LOS to the satellite pair. At discrete time t, if a satellite has current attitude matrix $\boldsymbol{R}\left({}^{TB}\hat{\boldsymbol{\omega}}_t,\ \psi_t\right)$ where ${}^{TB}\hat{\boldsymbol{\omega}}_t$ is an arbitrary vector expressed in $\{TB\}$ and ψ_t is an angle, this can be represented by the satellite's initial attitude $\boldsymbol{R}\left({}^{TB}\hat{\boldsymbol{\omega}}_0,\ \psi_0\right)$ and the attitude change $\Delta\boldsymbol{R}$ between the initial and current attitude. The $\Delta\boldsymbol{R}$ again can be separated into the rotational matrix $\boldsymbol{R}\left({}^{0}\hat{\boldsymbol{l}},\ \theta_{los}\right)$ representing a rotation of θ_{los} about the



(a) Original trajectory and tracking result (b) Original and transformed trajectory

Fig. 9 Chaser's trjaectory expressed in $\{T\,B\}$

Fig. 10 Position (**a**) and attitude (**b**) error converted relative motion compared to its origin



(b) Position error (b) Attitude principal angle error

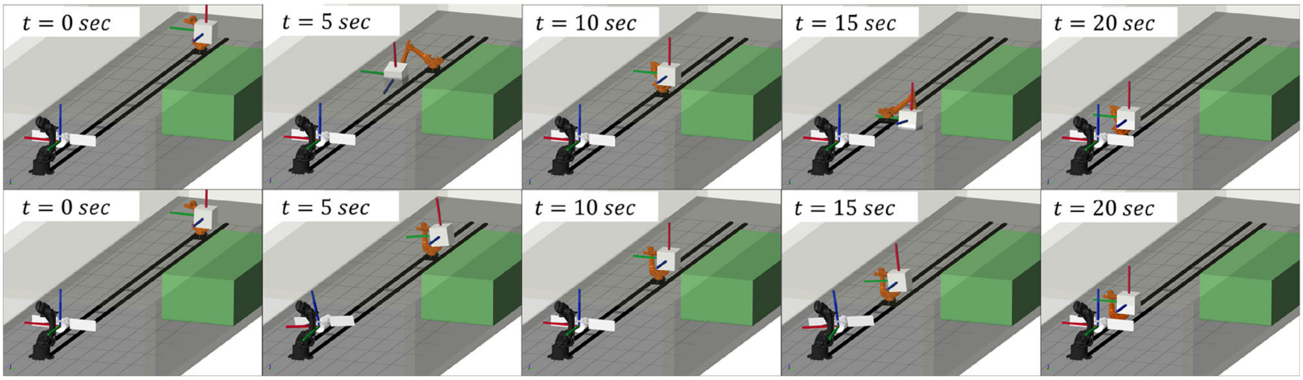


Fig. 11 Relative motion using 'Original' (first row) and 'Transformed' (second row) trajectory

LOS with respect to the initial attitude, and $\boldsymbol{R}({}^{0}\widehat{\boldsymbol{k}}, \phi)$ excluding the LOS based rotation, as shown in Eq. (10).

$$\boldsymbol{R}\left({}^{TB}\widehat{\boldsymbol{\omega}}_t,\ \psi_t\right) = \boldsymbol{R}\left({}^{TB}\widehat{\boldsymbol{\omega}}_0,\ \psi_0\right)\Delta\boldsymbol{R},$$

$$\Delta\boldsymbol{R} = \boldsymbol{R}\left({}^{0}\widehat{\boldsymbol{\omega}}_t,\ \delta\psi\right) = \boldsymbol{R}\left({}^{0}\widehat{\boldsymbol{l}},\ \theta_{los}\right)\boldsymbol{R}({}^{0}\widehat{\boldsymbol{k}},\ \phi). \tag{10}$$

Since Eq. (4) is equivalent to Eq. (11) and the slopes of the cosine functions in the diagonal elements are less than or equal to zero, ψ is minimized when the trace is maximized [30] by Eq. (12) where $tr(\boldsymbol{R})$ is trace of matrix $\boldsymbol{R}$. In the same manner, minimum $\delta\psi$ can be found by maximizing the trace of $\Delta\boldsymbol{R}, tr(\Delta\boldsymbol{R})$. It can be calculated using Eq. (13) where the dot product of ${}^{0}\widehat{\boldsymbol{k}}$ and ${}^{0}\widehat{\boldsymbol{l}}$ is defined as G, the coefficients of $\cos(\theta_{los})$ and $\sin(\theta_{los})$ are denoted as A and B, respectively. Then, $tr(\Delta\boldsymbol{R})$ is maximized if $\theta_{los} - \tan^{-1}(B/A) = 0$, where θ_{los} is given by Eq. (14).

$$\boldsymbol{R}(\widehat{\boldsymbol{\omega}},\ \psi) = \begin{bmatrix} \cos\psi + \widehat{\omega}_1^2(1-\cos\psi) & \widehat{\omega}_1\widehat{\omega}_2(1-\cos\psi) - \widehat{\omega}_3\sin\psi & \widehat{\omega}_1\widehat{\omega}_3(1-\cos\psi) + \widehat{\omega}_2\sin\psi \\ \widehat{\omega}_1\widehat{\omega}_2(1-\cos\psi) + \widehat{\omega}_3\sin\psi & \cos\psi + \widehat{\omega}_2^2(1-\cos\psi) & \widehat{\omega}_2\widehat{\omega}_3(1-\cos\psi) - \widehat{\omega}_1\sin\psi \\ \omega_1\widehat{\omega}_3(1-\cos\psi) - \widehat{\omega}_2\sin\psi & \widehat{\omega}_2\widehat{\omega}_3(1-\cos\psi) + \widehat{\omega}_1\sin\psi & \cos\theta + \widehat{\omega}_3^2(1-\cos\psi) \end{bmatrix}, \tag{11}$$

$$\psi = cos^{-1}\left(\frac{tr(\boldsymbol{R}) - 1}{2}\right)(0 \le \psi \le \pi), \tag{12}$$

$$G = {}^{0}\widehat{\boldsymbol{k}} \bullet {}^{0}\widehat{\boldsymbol{l}},$$

$$A = \left(\left(1 + G^2\right)cos\phi + \left(1 - G^2\right)\right),$$

$$B = -2Gsin\phi,$$

$$C = \left(1 - G^2\right)cos\phi + G^2,$$

$$tr(\Delta\boldsymbol{R}) = \sqrt{A^2 + B^2}cos\left(\theta_{los} - \tan^{-1}(B/A)\right) + C, \tag{13}$$

$$\theta_{los} = \begin{cases} \tan^{-1}(B/A)\ (A > 0) \\ \pi - \tan^{-1}(B/A)\ (A < 0) \\ \pi/2\ (A = 0,\ B > 0) \\ 0\ (A = 0,\ B < 0) \end{cases}, . \tag{14}$$

To verify the results of the rotation offset presented in this section, the example from Sect. 2.2 was reused. The method proposed in this section can minimize attitude change only when considering individual satellites. However, if the θ_{los} of Target and Chaser differ, the relative attitude between two satellites cannot be maintained. Therefore, the averaged or interpolated θ_{los} should be substituted into Eqs. (10), and this can be described as a reduction or sharing of rotation rather than a complete minimization. In Fig. 12, the rotation angle $\delta\psi$ between initial and current attitude of both Chaser and Target were depicted. The original one labeled as "Original" while the redistributed one after offset labeled as "Proposed". The attitude changes of the Chaser decreased from a maximum of 8.95° to 0.35°, and the Target decreased from a maximum of 13.57° to 8.90°.

3 Trajectory Planning for Discrete Relative Motion

The control frequency of industrial robots is typically over 250 Hz, while satellite control operates at around 10–20 Hz. Therefore, to simulate the motion of a satellite with an update frequency of approximately 50–100 ms using a robot controlled at 4 ms, interpolation is necessary. Interpolation refers to the method of generating a trajectory between given waypoints. The trajectory planning in task space involves solving inverse kinematics at each iteration, leading to a higher computational load and increased potential for singularity issues. Nevertheless, it allows the end effector to track the discrete positions, velocities, and accelerations of the given satellite. In Sect. 3.1, we discuss a trajectory planning method that is applicable for both translational and rotational motion while maintaining kinematical relationships. Additionally, Sect. 3.2 discusses methods for incorporating the robot's maximum speed and acceleration. Although not addressed in this paper, since the robotics system covered in this paper aims to maximize the Chaser robot's base movement through null space optimization when the end effector moves in a direction parallel to the rail, the base's speed and acceleration serving as constraints on the trajectory.

3.1 Hermite Polynomial Trajectory

If the motion of the satellite and its desired robot configuration command are computed at a discrete time t_s, the robot will replictate the motion after a certain period time. Although the time difference is relatively short, there may be discrepancies between the relative state in simulation data and the sensor measurement attached to the satellite. Therefore, in this section, it is assumed that the satellite state and the desired configuration of the robot are computed in advance up to $t_s + k$ $(k \ge 1)$ steps, and these are propagated to the manipulator.

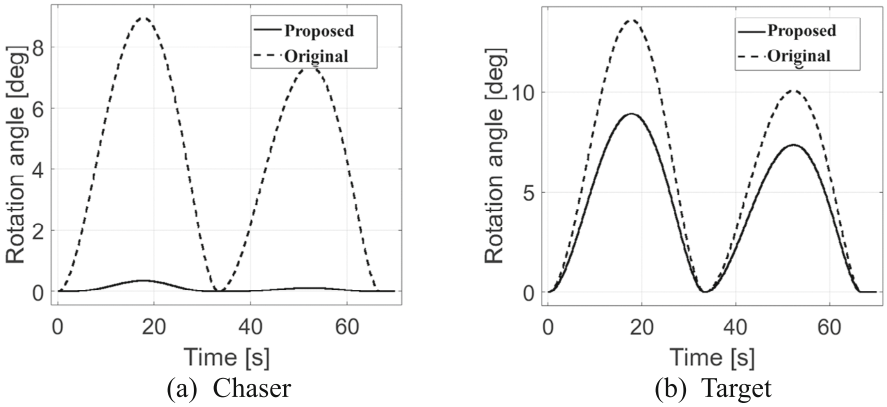


Fig. 12 Satellite rotation angle

In general, industrial robots plan continuous profiles for position ($\boldsymbol{x}$), velocity ($\boldsymbol{v}$), acceleration ($\boldsymbol{a}$), and jerk ($\boldsymbol{j}$) to achieve high precision and minimize vibration or abrasion. Individual Lagrange interpolation for each element carries the risk of overlooking the kinematic relationships. Therefore, it is crucial to consider the boundary conditions for each element in all interpolations. Septic Hermite interpolation could be one option. This is because it can cover the first and higher-order derivatives when there are more than three waypoints, and it can generate a continuous jerk profile. Additionally, it is also a method used in the commercial orbit software System Tool Kit (STK) for interpolating satellite attitude to ensure kinematical relationships. This is represented in Eq. (15), where a continuous trajectory connecting two arbitrarily selected waypoints is referred to as a "segment". The initial and final conditions of the segment are distinguished by subscripts 0 and 1, respectively. The superscript (n) denotes the n th derivative, and T represents the duration of each segment. The elapsed time of the segment is denoted by t, and such an expression ensures the continuity of each segment.

$$\begin{aligned}\boldsymbol{x}^{(n)} &= \boldsymbol{x}_0 p_0^{(n)} + \boldsymbol{x}_1 p_1^{(n)} + \boldsymbol{v}_0 q_0^{(n)} + \boldsymbol{v}_1 q_1^{(n)} + \boldsymbol{a}_0 r_0^{(n)} \\ &\quad + \boldsymbol{a}_1 r_1^{(n)} + \boldsymbol{j}_0 s_0^{(n)} + \boldsymbol{j}_1 s_1^{(n)} \quad (n = 0,\, 1,\, 2,\, 3)\,,\end{aligned}$$

$$\left\{\begin{array}{c} p_0 = \frac{(\mathrm{T}-t)^4\left(\mathrm{T}^3+4\,\mathrm{T}^2\,t+10\,\mathrm{T}\,t^2+20\,t^3\right)}{\mathrm{T}^7},\ p_1 = 1 - p_0 \\ q_0 = \frac{t\,(\mathrm{T}-t)^4\left(\mathrm{T}^2+4\,\mathrm{T}\,t+10\,t^2\right)}{\mathrm{T}^6},\ q_1 = -\frac{t^4\left(15\,\mathrm{T}^3-39\,\mathrm{T}^2\,t+34\,\mathrm{T}t^2-10\,t^3\right)}{\mathrm{T}^6} \\ r_0 = \frac{t^2\,(\mathrm{T}-t)^4\,(\mathrm{T}+4\,t)}{2\,T^5},\ r_1 = \frac{t^4\,(\mathrm{T}-t)^2\,(5\,\mathrm{T}-4\,t)}{2\,\mathrm{T}^5} \\ s_0 = \frac{t^3\,(\mathrm{T}-t)^4}{6\,\mathrm{T}^4},\ s_1 = -\frac{t^4\,(\mathrm{T}-t)^3}{6\,\mathrm{T}^4} \end{array}\right. . \tag{15}$$

The trajectory planning described above can also be applied to the attitude, angular velocity, and angular acceleration of the satellite. When the difference between the initial and final orientations of each segment is represented by a rotation vector $\boldsymbol{\varphi}$, the first and second derivatives of the rotation vector are denoted as $\dot{\boldsymbol{\varphi}}$ and $\ddot{\boldsymbol{\varphi}}$, respectively. If the magnitude of the rotation vector $|\boldsymbol{\varphi}|$ is sufficiently small, the following relationships hold when representing the satellite's orientation as a quaternion ($\boldsymbol{q}$) and its angular velocity and acceleration as $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$, respectively. [28, 29]

$$\boldsymbol{q} = \begin{bmatrix} 0 \\ \varphi/2 \end{bmatrix},$$

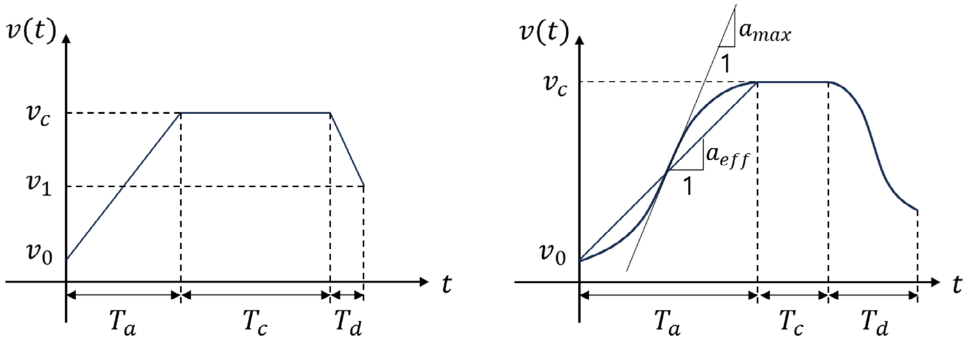
$$\boldsymbol{\omega} \simeq \dot{\boldsymbol{\varphi}},$$

$$\dot{\boldsymbol{\omega}} \simeq \ddot{\boldsymbol{\varphi}}. \tag{16}$$

When the relative distance between satellites is under 100 m, the relative angular velocity of the satellite is typically a few deg/s, and when T is about 0.1 s, substituting $\boldsymbol{x} = \boldsymbol{\varphi}$, $\boldsymbol{v} = \boldsymbol{\omega}$ and $\boldsymbol{a} = \dot{\boldsymbol{\omega}}$ into Eqs. (15) allows for planning the rotational motion. If $|\boldsymbol{\varphi}|$ or T is not sufficiently small, the error between the state of shorter period propagation and the planned trajectory may increase. However, there is no error at the boundaries of the segment. Therefore, the path planning discussed in this section focuses on discretely simulating the state of the given satellite at $t = T$ for each segment. As a result, the trajectory of the segment may differ from the actual one, and sometimes a trajectory that cannot be simulated by the robot may be required. To prevent this, a method for limiting the maximum speed and acceleration of the trajectory is discussed in Sect. 3.2.

3.2 S-curve Considering Maximum Velocity and Acceleration

In fact, in final approach phase, the relative velocity of satellites is typically on the order of a few cm/s or a few deg/s, so a polynomial trajectory may suffice. Nevertheless, considering the maximum speed and acceleration of the robot, the following approach can be considered: when the displacement h of the path is positive and it is desired to limit the maximum speed and acceleration of the robot, a trapezoidal



(a) Trapezoidal

(b) S-curve

Fig. 13 Trapezoidal and s-curve velocity profile

velocity profile like that shown in Fig. 13a could be one of solutions.

Since the trapezoidal profile consists of linear functions for the rising segment ($t \leq T_a$), the falling segment ($t \geq T - T_d$), and the cruising segment ($T_a < t < T - T_d$), both acceleration and jerk are discontinuous. Therefore, industrial robots prefer an s-curve trajectory like that shown in Fig. 13 (b). There are various methods for planning an s-curve, but in this paper, we interpolated the rising and falling segments using 7th-order polynomial equations as described in Sect. 3.1. To limit the maximum speed and acceleration of the robot, the s-curve was replaced with a trapezoidal profile and its characteristics were utilized.

In Fig. 13 (b), a_{max} represents the point where the slope of $v(t)$ is maximum. For a 7th-order polynomial, it occurs at $t = 0.5T_a$ where Eq. (17) yields the maximum acceleration. In this equation, v_c and $a_c = 0$ represent the velocity and acceleration in the cruising segment, respectively. When replacing the rising segment of the S-curve with the rising segment of the trapezoidal profile, the effective acceleration a_{eff} and $a_{\max}$ of the trapezoidal profile are related by Eq. (18).

$$a_{\max} = \frac{30(v_c - v_0) - 7(a_c + a_0)T_a}{T_a}, \tag{17}$$

$$a_{\text{eff}} = \frac{(v_c - v_0)}{T_a} = \frac{16a_{max} + 7(a_c + a_0)}{30}. \tag{18}$$

By substituting T_d, a_1, and v_1 for T_a, a_0, and v_0 in the descending segment, the minimum acceleration a_{min} can be found. Moreover, if the durations of the rising and falling segments exceed half of the segment duration T, neither the trapezoidal nor the s-curve trajectory can be implemented. Therefore, a_{eff} must satisfy Eq. (19), and if the equation holds true, $T_a = T_d = \frac{1}{2}T$ with no cruising segment T_c, corresponding to a triangular profile [31].

$$a_{eff} \geq \frac{2h - T(v_0 + v_1) + \sqrt{4h^2 - 4h(v_0 + v_1)T + 2\left(v_0^2 + v_1^2\right)T^2}}{T^2}. \tag{19}$$

The cruising velocity of the S-curve can be expressed as a function of a_{eff} from the characteristics of the trapezoidal trajectory, as shown in Eq. (20). Additionally, if the required cruising speed is too high, path planning may not be feasible, and therefore, it should not exceed the maximum allowable cruising speed $v_{\max}$, satisfying Eq. (21).

$$v_c = \frac{1}{2}\left(v_0 + v_1 + a_{\text{eff}}T - \sqrt{a_{\text{eff}}^2 T^2 + 2a_{eff}\left(v_0 + v_1\right)T - 4a_{\text{eff}}h - \left(v_0 - v_1\right)^2}\right), \tag{20}$$

$$v_{\max} \leq \sqrt{a_{eff}h + \frac{v_0^2 + v_1^2}{2}}. \tag{21}$$

The summary of this section for the ascending segment is as follows: Given the robot's maximum velocity and acceleration as v_u and a_u, respectively, simultaneous satisfaction of Eqs. (17) and (21) allows for the generation of a 7th-order interpolated s-curve trajectory. Examples of trajectory planning using the boundary conditions for the satellite's x position in Table 2 are presented in Fig. 14. Assuming the maximum speed and acceleration as $|v_u| = 5\text{cm/s}$ and $|a_u| = 180\text{cm/s}^2$, Fig. 14 represents a nearly straight movement trajectory. When the equal condition of Eq (19) is satisfied, the trajectory looks like a triangular one (min a_{eff} in Fig. 14), while the equal condition of Eq. (22) results in the trajectory closer to a linear motion (max a_{eff} in Fig. 14).

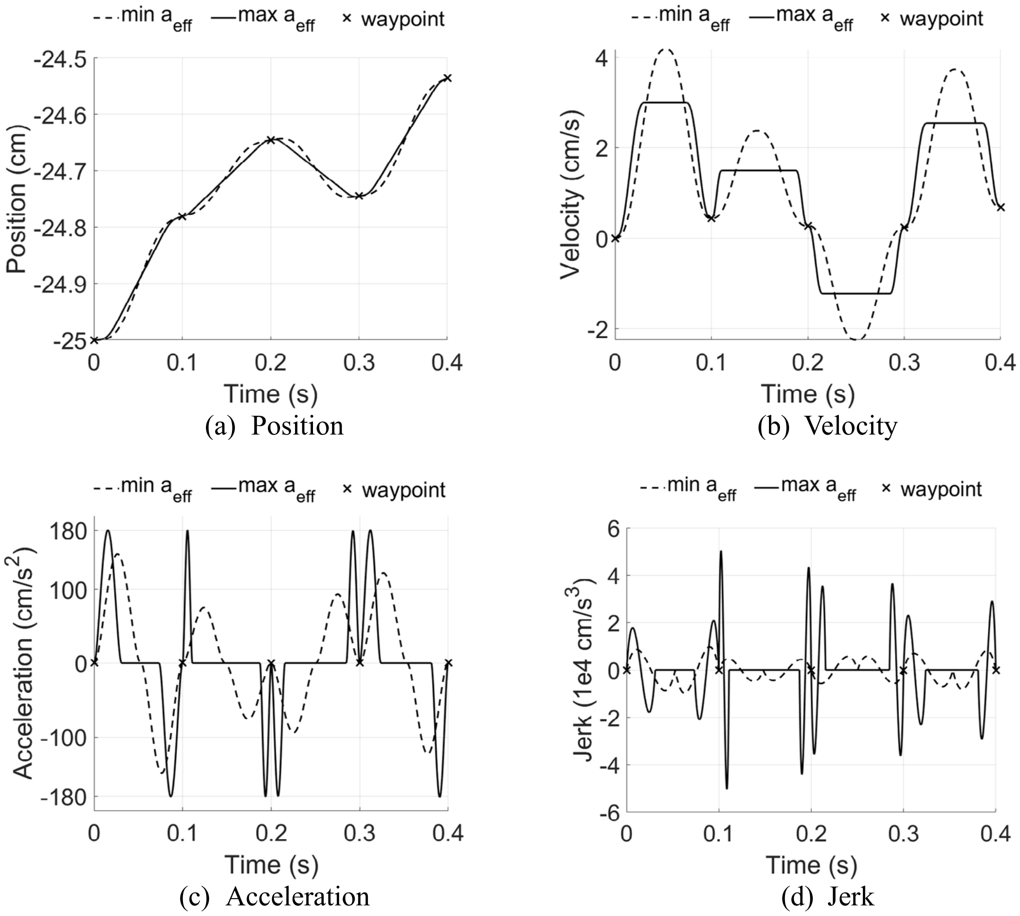
$$a_{\text{eff}} \leq \min\left(\frac{16a_u + 7a_0}{30}, \frac{1}{h}\left(v_u^2 - \frac{v_0^2 + v_1^2}{2}\right)\right). \tag{22}$$

4 Conclusion

This paper presents a method to simulate rendezvous and docking operations within the constraints of a ground-based testbed. By aligning the line-of-sight between satellites with a fixed reference direction, the approach transforms complex satellite motions into a manageable, straight-line simulation. This method effectively addresses the limitations of testbed

Table 2 Boundary conditions for k-th segment ($T = 0.1$s)

k	x_k (cm)	v_k (cm/s)	a_k (cm/s^2)	j_k (cm/s^3)
0	– 25.0000	0	0.3985	0
1	– 24.7810	0.4380	0.4775	0
2	– 24.6450	0.2720	0.0666	0
3	– 24.7446	0.2388	0.4110	0
4	– 24.5356	0.6901	0.9691	0



(a) Position

(b) Velocity

(c) Acceleration

(d) Jerk

Fig. 14 S-curve where the effective acceleration is minimum (min a_{eff}) or maximum (max a_{eff})

workspaces, allowing for accurate performance evaluations without requiring extensive testbed expansions. The study also introduces techniques to minimize attitude changes of satellites during simulations, which reduces the movement and stress on the robot's end effector. By leveraging advanced trajectory planning methods, including Hermite interpolation, the research ensures that kinematic relationships are maintained, and realistic motion profiles are achieved.

Overall, the proposed approach offers a practical solution for evaluating rendezvous and docking technologies by adapting simulations to fit within physical constraints. This advancement enhances the fidelity of ground-based tests such as relative navigation research using actual sensors, thereby improving the reliability of automated rendezvous and docking systems crucial for future space missions. However, this paper has the limitation of focusing solely on the method for simulating relative motion. Therefore, we believe that including the design of a real testbed, the process of setting up the testbed, and validation results obtained through actual hardware implementation would greatly enhance the reliability of the study.

Acknowledgements This paper is research conducted under the support of Agency for Defense Development with funding from the Defense Acquisition Program Administration in 2024, as part of the "SDRS Satellite Ground Simulation Test Facility" project (Contract No. UG2230613D).

Data availability The data used in this study contains sensitive information and is therefore not publicly available for security reasons. Researchers who wish to access the data may be granted limited access following a security review and approval. Access requests should be directed to the corresponding author.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Xie Y, Chen C, Liu T, Wang M (2021) Guidance, Navigation, and control for spacecraft rendezvous and docking: theory and methods. Springer Nature
2. Goodman JL (2006) History of space shuttle rendezvous and proximity operations. J Spacecr Rocket 43(5):944–959. https://doi.org/10.2514/1.19653
3. John U (2021) 55 Years Ago: Gemini VIII, the First Docking in Space, NASA. https://www.nasa.gov/history/55-years-ago-gemini-viii-the-first-docking-in-space/. Accessed 1 Mar 2024
4. Ezell EC (1978) The partnership: a history of the Apollo-Soyuz test project. NASA, Nat. Aeronautics and Space Administration
5. Fehse W (2003) Automated rendezvous and docking of spacecraft, 16th. Cambridge University Press
6. Li WJ, Cheng DY, Liu XG, Wang YB, Shi WH, Tang ZX et al (2019) On-orbit service (OOS) of spacecraft: a review of engineering developments. Prog Aerosp Sci 108:32–120. https://doi.org/10.1016/j.paerosci.2019.01.004
7. Kim HD, Lim SM (2012) Technologies of space robot and satellite services. Curr Ind Technol Trends Aerosp 10(1):171–178
8. Luo Y, Zhang J, Tang G (2014) Survey of orbital dynamics and control of space rendezvous. Chin J Aeronaut 27(1):1–11. https://doi.org/10.1016/j.cja.2013.07.042
9. Wilde M, Clark C, Romano M (2019) Historical survey of kinematic and dynamic spacecraft simulators for laboratory experimentation of on-orbit proximity maneuvers. Prog Aerosp Sci 110:100552. https://doi.org/10.1016/j.paerosci.2019.100552
10. Santaguida L, Zhu ZH (2023) Development of air-bearing microgravity testbed for autonomous spacecraft rendezvous and robotic capture control of a free-floating target. Acta Astronaut 203:319–328. https://doi.org/10.1016/j.actaastro.2022.11.056
11. Rybus T, Seweryn K (2016) Planar air-bearing microgravity simulators: Review of applications, existing solutions and design parameters. Acta Astronaut 120:239–259. https://doi.org/10.1016/j.actaastro.2015.07.026
12. Sabatini M, Palmerini GB, Gasbarri P (2015) A testbed for visual based navigation and control during space rendezvous operations. Acta Astronaut 117:184–196. https://doi.org/10.1016/j.actaastro.2015.07.026
13. A design methodology for Neutral Buoyancy simulation of space operations
14. Langley Research Center (2017) Langley Rendezvous and Docking Simulator. NASA https://www.nasa.gov/image-article/langley-rendezvous-docking-simulator/. Accessed 1 Mar 2024
15. Yoshida K, Hirose S, Ogawa T (1994) Development of the dynamic motion simulator of 3D micro-gravity with a combined passive/active suspension system. In: Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space
16. Boge T, Benninghoff H, Zebenay M, Rems F (2012) Using robots for advanced rendezvous and docking simulation
17. Boge T, Benninghoff H (2013) Rendezvous simulation for on-orbit servicing missions using advanced robotic technology. IFAC Proc Vol 46(19):155–160. https://doi.org/10.3182/20130902-5-DE-2040.00093
18. Zebenay M (2014). Development of a robotics-based satellites docking simulator. https://doi.org/10.4233/uuid:d349ac8f-b91a-43ea-a33b-2b86d3235293
19. Trube MJ, Hyslop AM, Carignan CR, Easley JW (2012) Ground simulation of an autonomous satellite rendezvous and tracking system using dual robotic systems. In: International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)
20. Benninghoff H, Rems F, Risse EA, Mietner C (2017) European proximity operations simulator 20 (EPOS)-a robotic-based rendezvous and docking simulator. J Large-Scale Res Fac JLSRF 15:10. https://doi.org/10.17815/jlsrf-3-155
21. Rems F, Frei H, Risse EA, Burri M (2021) 10-year anniversary of the European proximity operations simulator 2.0—looking back at test campaigns, rendezvous research and facility improvements. Aerospace 8(9):235. https://doi.org/10.3390/aerospace8090235
22. Bell R, Collins J, Wertz J, Hansen LJ (2003) Hardware-in-the loop tests of an autonomous GN&C System for on-orbit servicing. In: AIAA Space 2003 Conference Exposition. https://doi.org/10.2514/6.2003-6372
23. Okamoto H, Kato H (2022) The Development of the Hybrid Dynamics Simulation System for Rendezvous and Docking: SAT-Dyn. 2022 IEEE Aerospace Conference (AERO) pp 1–9. https://doi.org/10.1109/AERO53065.2022.9843677
24. Liegeois A (1977) Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans Syst Man Cybern 7(12):868–871. https://doi.org/10.1109/TSMC.1977.4309644
25. Kuffner JJ, and Steven ML (2000) RRT-connect: an efficient approach to single-query path planning. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). Vol. 2. https://doi.org/10.1109/ROBOT.2000.844730
26. Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. Int J Robot Res 5(1):90–98. https://doi.org/10.1177/027836498600500010
27. Chiaverini S (1997) Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Trans Robot Autom 13(3):398–410. https://doi.org/10.1109/70.585902
28. Sergei T (2003) Attitude Interpolation. In: AAS/AIAA Space Flight Mechanics Meeting, Ponce, Puerto-Rico
29. Sergei T (2006) Parametric optimization of closed-loop slew control using interpolation polynomials. In: AAS/AIAA Space Flight Mechanics Meeting
30. Lynch KM, Park FC (2017) Modern robotics mechanics, planning, and control. Cambridge University Press (**ISBN 9781107156302**)
31. Biagiotti L, Melchiorri C (2008) Trajectory planning for automatic machines and robots. Springer Science & Business Media

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.