Deep Learning-accelerated Time Shift Governor for Spacecraft Proximity Operations

Taehyeun Kim*[,1], Robin Inho Kee†[,1], Ilya Kolmanovsky‡, and Anouck Girard§
Department of Aerospace Engineering, University of Michigan, Ann Arbor, 48109 MI, USA

This paper develops a time shift governor (TSG)-based control scheme, accelerated by a deep learning model, to enforce constraints during rendezvous and docking (RD) missions in the setting of the Two-Body problem. As an add-on scheme to the nominal closed-loop system, the TSG generates a time-shifted Chief spacecraft trajectory as a target reference for the Deputy spacecraft. The target reference gradually converges to the Chief spacecraft trajectory as the time shift decreases to zero. This modification of the commanded reference trajectory ensures that constraints are enforced while the time shift is reduced to zero to effect the rendezvous. The proposed Deep Learning model is trained using the imitation learning approach to compute the time shift parameter as a function of current and past Deputy and Chief spacecraft states. This time shift parameter value is verified through forward-in-time online simulations; if this verification step fails, the conventional TSG strategy based on bisections is triggered. We provide simulation results for RD missions based on Crew-3 docking mission data to demonstrate the effectiveness of the proposed control scheme. The proposed scheme reduces the time to compute the time shift parameter in most of the scenarios, and successfully completes rendezvous missions.

I. Introduction

Spacecraft proximity operations (SPO) involve two spacecraft maneuvering near each other in space, as in rendezvous and docking (RD). The primary spacecraft (referred to as the Chief) maintains a nominal orbit passively or actively, while the other spacecraft (referred to as the Deputy) uses an active controller to perform the RD mission [1]. The objective of the Deputy's controller is for the Deputy spacecraft to approach the Chief spacecraft within a thrust limit and bring the Deputy spacecraft into the vicinity of the Chief spacecraft, while maintaining the line of sight (LoS) constraint of the docking port and relative velocity constraint. The SPO missions are completed when the Deputy spacecraft achieves close proximity to the Chief spacecraft.

Many spacecraft RD missions were conducted near and beyond the Earth. The first spacecraft docking mission was done in the Gemini 8 mission, in which astronauts manually operated the docking with the target vehicle. The following RD missions also operated manually, as was done for the Apollo mission in 1969 and the Skylab mission in 1973. As docking techniques were cultivated through former missions, docking missions became complicated and required an autonomous system to ensure safety and reduce costs. The first autonomous spacecraft docking mission was performed by the Orbital Express mission in 2007. As examples in private sectors, Northrop Grumman then began to provide commercial resupply services to the ISS in 2014, and SpaceX launched spacecraft docking missions for supply service to NASA and for a commercial company

*Ph.D. candidate, Aerospace Engineering, 1320 Beal Ave, Ann Arbor, MI 48109
†Master student, Department of Mechanical Engineering, University of Michigan
‡Professor, Aerospace Engineering, 3038 FXB, 1320 Beal Ave, Ann Arbor, MI 48109, AIAA Associate Fellow.
§Professor, Robotics and Aerospace Engineering, 3264 FMCRB, 2505 Hayward Street, Ann Arbor, MI 48109, AIAA Associate Fellow.
[1]These authors contributed equally to this work

in 2020. Thus, the RD techniques played a crucial role in space missions while satisfying the growing interest of the space community in complicated missions.

Various control approaches have been studied for RD techniques. The artificial potential function (APF) methods have been applied to ensure safety in RD operations. In [2, 3], APF was combined with adaptive control and dual-quaternion formulations for translation and attitude constraints, respectively. While APF deals with managing motion constraints by employing potential functions that delineate avoidance zones and target proximity, its solutions are non-optimal, struggling with approach velocity constraints and thrust limitations. Meanwhile, model predictive control (MPC) has been developed to provide a solution to the constrained optimal control problem with a finite-horizon based on the Clohessy-Wiltshire-Hill (CWH) equations, which is a linearized relative motion model [4]. Despite its effectiveness, MPC's requirement for significant onboard computational resources remains a challenge. The integration of genetic algorithms (GA) with fuzzy logic controllers (FLC) has been studied for addressing the spacecraft RD challenge. It is useful in exploring global solutions in non-convex multi-dimensional optimization problems and represents an early use of AI concepts, demonstrating the potential of AI in space technology [5]. However, its reliance on subjective selection for fuzzy membership functions introduces a potential bias, undermining its consistency.

Recent research has demonstrated the potential of learning approaches in RD operations. Learning methods possess advantages in efficiency and performance over traditional technologies [6]. Deep reinforcement learning (DRL), in particular, has been implemented in autonomous guidance algorithms and proximity operation guidance problems [7–11]. These studies validated the role of DRL in advancing spacecraft autonomy and demonstrated efficacy in reducing resource usage. However, their DRLs, which are based on linearized dynamics, are limited in their applicability because the approaches are unable to guarantee stability when the dynamics are highly complicated. Moreover, the nonlinear functional operator, computed from reinforcement learning, needs further studies to address the convergence issues [12]. Thus, unresolved challenges in ensuring stability and convergence impede the space applications of the DRL for safety control [13].

The Time Shift Governor (TSG) approach, used for spacecraft formation and RD problems, enforces constraints by adjusting the time shift to ensure the stability of the controller in a spacecraft to the target [14–17]. The TSG, which is a variant of the parameter governor, is an add-on scheme used for managing constraints without redesigning the closed-loop system. It has been used effectively in scenarios involving circular Earth orbits in the Two-Body problem framework to more complex orbital configurations like halo orbits in the Circular Restricted Three-Body Problem (CR3BP) setting [14, 16]. Unlike general nonlinear model predictive controllers that solve high-dimensional optimization problems, TSG reduces the optimization problem to a single dimension, which involves a time shift. TSG is particularly adept at handling nonlinear and non-convex constraints and guarantees convergence to a solution that satisfies all constraints, provided that the initial conditions are feasible. This allows TSG to handle constraints in complex scenarios such as multi-body problems and missions in elliptical orbits. However, the computation demands of calculating the TSG parameter are still an issue in complicated scenarios, such as Three-Body problems or elliptic orbits.

This study enhances this TSG framework by integrating a Long Short-Term Memory (LSTM) network, known as one of the representative deep learning models for time series prediction [18, 19], a method to accurately and efficiently predict time shift parameters.

The original contributions of this paper are:

1) A novel framework that integrates a deep-learning model with the standard TSG, employing an LSTM cell and an artificial neural network to encode state inputs and compute optimal time shift parameters, thereby replacing iterative processes and accelerating TSG performance.

2) The Deep Learning-accelerated TSG (DL-TSG) utilizes a Phase Adaptive Sliding Window to improve performance.

3) A loss function guides the deep-learning model to compute optimal time shift parameter estimations that guarantee constraint satisfaction for a sufficiently long prediction horizon.

4) A hybrid model is developed to ensure the robustness of the algorithm.

This paper is organized as follows: In Section II, we summarize the coordinate system, spacecraft translational dynamics model, the nominal controller, as well as the constraints considered during the RD mission. Then, in Section III, we introduce the TSG to enforce the constraints. Section IV introduces our DL-TSG, which integrates an LSTM network with a phase-adaptive sliding window and a custom loss function, further enhanced by a hybrid prediction algorithm. Section V presents experimental results, demonstrating the efficacy of our approach. Finally, Section VI concludes the paper with a summary of contributions and potential directions for further research.

II. Problem Formulation

In our study, we consider a docking mission involving two spacecraft in an Earth orbit, which is governed by several mission-specific constraints. Throughout this mission, the secondary spacecraft, referred to as the Deputy, is positioned in proximity to the primary spacecraft, known as the Chief. Specifically, given a rendezvous and docking (RD) problem with two spacecraft on Low Earth Orbits (LEO), the goal is to find a trajectory of the optimal time shift parameters such that the Deputy spacecraft achieves the Chief spacecraft satisfying the mission constraints. This section will detail the coordinate systems, dynamics models, control strategies, and constraints that define the framework for our approach to the RD mission. We assign the subscripts c and d to denote the Chief and Deputy spacecraft respectively, while the subscript i is used to denote a spacecraft that could be either the Chief or the Deputy.

A. Coordinate systems

In this work, two right-handed coordinate systems are employed to describe the spacecraft equations of motion and relative motion, as shown in Fig . 1. The origin of the Earth-centered inertial (ECI) frame, $\mathcal{E} : \{\hat{x}_\mathcal{E}, \hat{y}_\mathcal{E}, \hat{z}_\mathcal{E}\}$, is at the center of the Earth $O_\mathcal{E}$, with the $\hat{x}$-axis pointing towards the vernal equinox—the direction from the Earth to the Sun at noon on the day of the spring equinox, the $\hat{z}$-axis pointing towards the Earth's rotational axis, and the $\hat{y}$-axis completing the right-handed system.

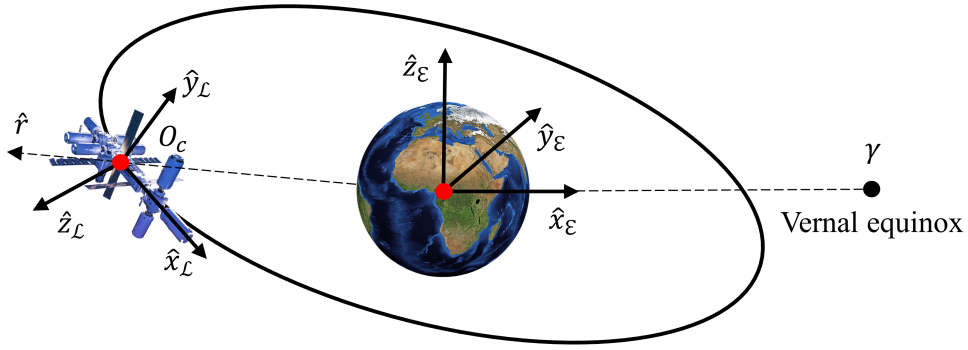

Fig. 1 ECI and VNB coordinate systems.

The local Velocity-Normal-Binormal (VNB) frame, $\mathcal{L} : \{O_c, \hat{x}_\mathcal{L}, \hat{y}_\mathcal{L}, \hat{z}_\mathcal{L}\}$, is also defined to describe the relative motion of objects with respect to the Chief spacecraft. Unlike the ECI frame, the VNB frame is a rotating and accelerating coordinate system and has its origin at the center of mass of the Chief spacecraft O_c with three orthonormal vectors which are defined as

$$\hat{x}_\mathcal{L} = \frac{v(X_c)}{\|v(X_c)\|}, \ \hat{y}_\mathcal{L} = \frac{p(X_c) \times \hat{x}_\mathcal{L}}{\|p(X_c) \times \hat{x}_\mathcal{L}\|}, \ \hat{z}_\mathcal{L} = \hat{x}_\mathcal{L} \times \hat{y}_\mathcal{L}, \tag{1}$$

where $p(\cdot), v(\cdot) : \mathbb{R}^6 \to \mathbb{R}^3$ are functions that map from a state to position and velocity, respectively, expressed in the ECI frame.

B. Dynamics

The equations of motion of spacecraft are given by

$$\dot{X}_i = f(t, X_i(t), u_i(t)), \tag{2}$$

where $X_i = [x_i, y_i, z_i, \dot{x}_i, \dot{y}_i, \dot{z}_i]^\mathsf{T}$, for $i \in \{c, d\}$, and $u_i = [u_{1,i}, u_{2,i}, u_{3,i}]^\mathsf{T}$ denote the spacecraft state and the control input to the spacecraft, expressed in the ECI frame. Note that the Chief spacecraft maintains its nominal orbit without using control input, i.e., $u_c = \vec{0}$, while the Deputy spacecraft uses control input to track the target. In this work, we omit the subscript d in the control input, i.e., $u = u_d$. The translational equations of motion for spacecraft are given as,

$$\ddot{\vec{r}}_i = -\frac{\mu}{r_i^3}\vec{r}_i + \vec{u}_i, \tag{3}$$

where μ stands for the gravitational parameter, $\vec{r}_i = p(X_i)$ is the spacecraft position vector and $r = \|\vec{r}\|$ is its 2-norm.

C. Discrete Time Linear Quadratic Controller

Since the TSG is an add-on method, we need a nominal controller that is (locally) stabilizing to a target along the reference orbit. In this work, the discrete-time linear quadratic (DTLQ) controller is implemented to provide an optimal control solution stabilizing the Deputy spacecraft to the target.

While the Chief spacecraft is assumed to follow a reference orbit, which is an unforced natural motion, the Deputy spacecraft has the feedback controller

$$u_d(t) = K(t_k)(X_d(t_k) - X_v(t_k)), \quad t_k \le t < t_{k+1}, \tag{4}$$

where $X_v(t)$ is the virtual target, which is determined by the TSG, and K is the periodic LQR gain which is calculated for the linearization of (2) along the state trajectory on the reference orbit. To save computational effort, the standard infinite-horizon LQ control gain K is pre-computed for an orbit as

$$K(t_k) = (B_d^\mathsf{T} S B_d + R)^{-1} B_d^\mathsf{T} S A_d, \tag{5}$$

where S is the infinite horizon solution of the Discrete Algebraic Riccati Equation (DARE),

$$S = Q - A_d^\mathsf{T}(t_k) S B_d (R + B_d^\mathsf{T}(t_k) S B_d(t_k))^{-1} B_d^\mathsf{T}(t_k) S A_d(t_k) + A_d^\mathsf{T}(t_k) S A_d(t_k), \tag{6}$$

where $Q \ge 0$ and $R > 0$ are the weighting matrices associated with the state and control, respectively, and S stands for the solution of DARE at the time instant t_k, e.g., $S = S(t_k)$. Here $A_d(t_k)$ and $B_d(t_k)$ represent a discrete-time linearized model at the virtual target state.

Note that we assume that the nominal controller, i.e., the DTLQ controller, is able to track a virtual target $X_v(t)$, which is a time-shifted state trajectory of the reference orbit. This assumption can be held by the solutions of the TSG that ensure its attractivity to the virtual target in a sufficiently long prediction horizon.

D. Constraints

In this paper, we consider the state and control input constraints for the Deputy spacecraft. During the RD mission, the Deputy spacecraft is maintained within the safe approach corridor from the docking port, assuming that the docking port is towards the opposite direction of the Chief spacecraft's velocity direction.

This ensures that the Deputy spacecraft is visible at all times from the camera on the docking port of the Chief spacecraft. The Line of Sight (LoS) constraint is written as

$$h_1 = -\frac{v(X_c)^{\mathsf{T}} p(X_d - X_c)}{\|v(X_c)\|\|p(X_d - X_c)\|} + \cos(\alpha) \leq 0, \tag{7}$$

where α is a LoS half-cone angle.

The physical limit of the thrust magnitude is expressed as

$$h_2 = \|u_d\| - u_{\max} \leq 0, \tag{8}$$

where $u_{\max}$ is the maximum total magnitude of the physical thrusters. Considering that the spacecraft typically uses a main thruster for maneuvers, we assume that the thruster nozzle direction accurately aligns with the desired thrust direction, which means that the attitude control problem is not incorporated. To leverage faster response, an alternative approach is implemented to enforce Eq. (8) using a saturation function rather than addressing Eq. (8) as a constraint by the TSG. The control input, limited by the saturation, is determined as

$$u_d(t) := \min(\|u_d(t)\|, u_{\max}) \cdot \hat{u}_d(t), \tag{9}$$

where $\hat{u}_d(t) = u_d(t)/\|u_d(t)\|$. Note that the prediction of the TSG accounts for the saturated control in Eq. (9) to enforce Eq. (8).

We limited the relative velocity magnitude of the Deputy spacecraft with respect to the Chief spacecraft to avoid the risk of high-speed collisions. Hence, the soft docking constraint is defined as

$$h_3 = \|v(X_d - X_c)\| - \gamma_2\|p(X_d - X_c)\| - \gamma_3 \leq 0, \text{ if } \|p(X_d - X_c)\| \leq \gamma_1, \tag{10}$$

where γ_1, γ_2 and γ_3 are predetermined coefficients considering mission specific requirements.

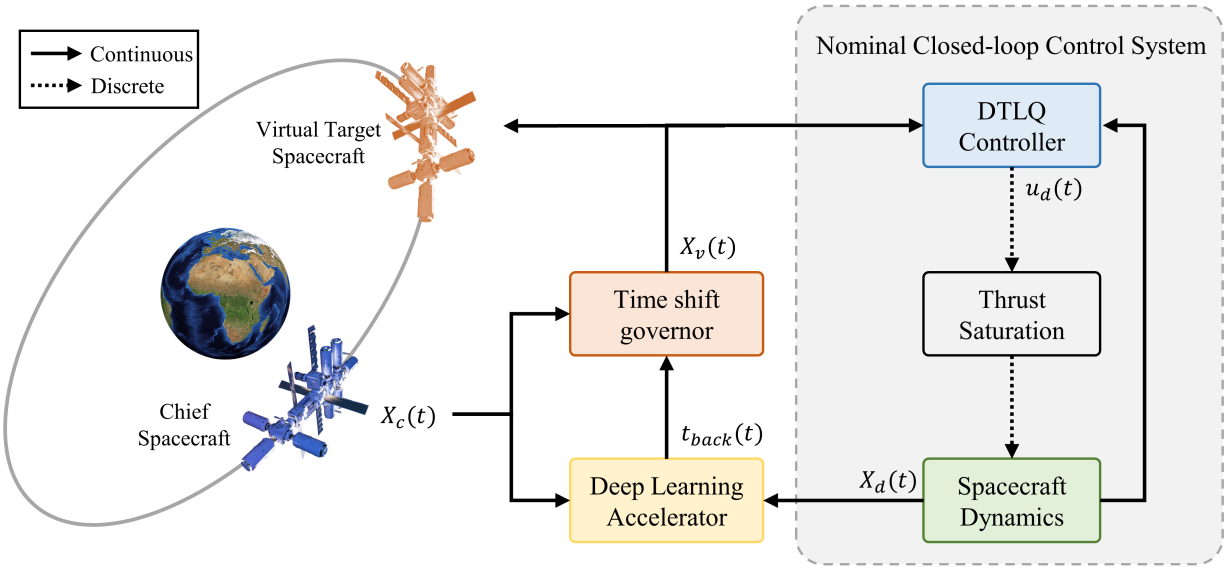

Fig. 2 Diagram of the control system architecture incorporating learning-based predictive model with TSG and DTLQ controller.

III. Time Shift Governor

As a variant of the parameter governor [20], the Time Shift Governor (TSG) is an add-on scheme that enforces state and control constraints, shown in Fig. 2. In this paper, we apply TSG to the spacecraft rendezvous and docking (RD) problem. TSG generates virtual targets, which are determined as time-shifted trajectory of the Chief spacecraft, by solving a lower dimensional optimization problem using a receding horizon trajectory subject to pointwise-in-time constraints.

To satisfy constraints, the TSG computes the smallest in magnitude non-positive time shift, which represents the minimum difference in time between a virtual target and the Chief spacecraft along the orbital track. The TSG then provides the virtual target corresponding to this time shift, i.e.,

$$X_v(t) = X_c(t + t_{\texttt{back}}(t)), \tag{11}$$

where $t_{\texttt{back}}(t) = t_{\texttt{back}}(t_k)$ for $t \in [t_k, t_{k+1})$ is a piecewise constant time shift. Additionally, the LQ control gain K is precomputed for one orbit period and implemented using linear interpolation. In this work, we assume that the Chief spacecraft is positioned forward along the orbital track relative to the Deputy spacecraft. Note that the upper and lower bounds of the time shift are set to be zero and the initial time shift, respectively.

The virtual target determined by the TSG needs the following properties to demonstrate its effectiveness. First, at the initial time instant, there exists a feasible time shift that guarantees a predicted trajectory satisfying constraints over a sufficiently long horizon. The nominal controller is then capable of (locally) stabilizing the Deputy spacecraft to a target. Lastly, with any time shift parameter $t_{\texttt{back}}$, the steady-state response strictly satisfies the constraints, which means that the Deputy spacecraft with TSG satisfies the constraints for all future time.

Considering the same relative dynamics repeated every orbit period, we use one orbit period of prediction horizon in a moving horizon manner. The initial time shift parameter is computed using the brute force algorithm. For computational efficiency, the search order begins at the point along the reference state trajectory closest to the Deputy spacecraft state. In other words, we adopt a warm start approach.

IV. Deep Learning-accelerated Time Shift Governor (DL-TSG)

In this section, we introduce a novel framework that accelerates the Time Shift Governor (TSG) computations through the use of a deep-learning model. This deep learning model utilizes a Long short-term memory (LSTM) cell layer and a fully connected layer to map sequential data input into the time shift parameter. The proposed DL-TSG computes the time shift parameter which results in enforcing constraints and accomplishing the rendezvous mission. We apply a phase-adaptive sliding window (PA-SW) approach to enhance the performance of the deep learning model during the RD mission. It is important to note that, in an ideal case, the deep learning prediction model can generate the time shift parameter corresponding to a state at a given time instant since it is proposed as a mapping function. However, from a practical point of view, the deep learning model can compute a more accurate time shift prediction by providing additional input data, such as a sequence of past states, compared to the time shift prediction using only the current state. In the training process, we implement a loss function based on a problem-specific heuristic. Thus the proposed DL model provides an imitation learning based or an explicit TSG implementation. This DL model is then integrated with the TSG that uses the bisection approach to handle cases where the output of the DL-based model does not enforce the constraints due to approximation errors.

A. Data Preparation

The dataset consists of 500 trajectories corresponding to different initial states of the Deputy spacecraft, $X_d(t_0)$, and with the conventional TSG based on bisections implemented to enforce the constraints. We allocate the dataset as follows: 60% for training, 20% for validation, and 20% for testing. To enhance the

training performance, we use Min-Max normalization to scale the time shift values to a range between zero and one, i.e., $t_{\texttt{back}}^{(\texttt{min/max})} \in [0, 1]$, maintaining the original distribution of the data without distortion of information. This causes the network to converge faster during training and improves the network's ability to learn from the data.

B. Phase-Adaptive Sliding Window

The DL model makes decisions based on a finite sequence of past deputy and chief spacecraft states, within a specified past window of data. We implement a PA-SW approach to enhance the efficiency and accuracy of our prediction model during the RD mission. This approach involves dynamically adjusting the window size based on the current phase of the mission, defined by the relative distance between the Deputy and Chief spacecraft. Sliding window approaches have been used in various studies to handle sequential data effectively [21, 22], and we modify this approach to implement the most effective prediction model corresponding to the specific phases of the RD mission. By adapting the window size according to the mission's phase, our approach ensures efficient data utilization and computational cost reduction throughout the mission.

For distances between 50km and 1km, the model utilizes small window sizes (1 and 2). As the Deputy spacecraft gets close to the Chief, from 1km to docking, the window size is increased to 3. This larger window increases model complexity and fidelity and enhances the prediction accuracy at critical close-range phases.

The window is adjusted as

$$W(t_k) = \{[X_c^{\mathsf{T}}(t_j), X_d^{\mathsf{T}}(t_j)]^{\mathsf{T}} \in \mathbb{R}^{12\times w_k} \mid j \in \mathcal{I}_w(t_k)\}, \tag{12}$$

where $\mathcal{I}_w(t_k) = \{j \in \mathbb{N} \mid j = \max(0, k - w_k + 1), \ldots, k\}\}$ is a set of indices in the dynamic sliding window at time step k and w_k is the window size, which is set dynamically based on the mission phase: $w_k = 1$ at the initial time, $w_k = 2$ during the first phase (50km to 1km), and $w_k = 3$ throughout the second phase (1km to 0km).

This phase-adaptive strategy ensures that immediate and appropriate predictions are made with the available data, allowing the model to adapt as more data becomes available and as the Chief spacecraft is approached where the constraints become more prominent. Sequential data provides context and temporal dependencies that are important in accurate prediction. The LSTM cell encodes the sequential spacecraft state information. Note that the encoded information computed by the LSTM cell encapsulates the dynamics information and imposed constraints. The fully connected neural network then decodes the encoded data as an output of the time shift parameter. Compared to the Multilayer Perceptron (MLP), which does not take into account the temporal dependencies in the sequential data, the LSTM cell shows better performance when we address the sequential data [23].

C. Network Architecture

In this study, we use the LSTM network for the time shift parameter prediction model. LSTMs are often used to capture temporal dependencies in sequential data [18, 19]. The LSTM-based model is a mapping from the past sequence of Chief and Deputy spacecraft states to the time shift parameter:

$$t_{\texttt{back}} = \Gamma_{\texttt{Pred}}(W(t_k)). \tag{13}$$

Fig. 3 illustrates the network architecture, implemented using PyTorch. To enhance stability and efficiency in training, we first apply a batch normalization layer to standardize inputs. An LSTM layer then processes the sequential data in a window. Rather than leveraging all input data, the dropout layer is used to randomly disable a portion of the inputs during training, addressing the overfitting issue. This method is used to develop a model which avoids excessively learning particular input data. At the end of the network, a fully connected

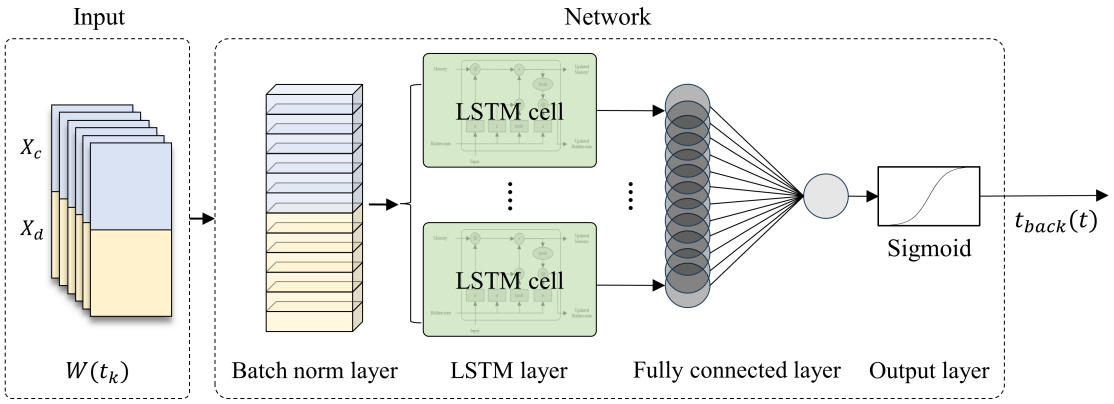

Fig. 3 Overview of LSTM-based time shift prediction model network architecture.

layer provides an output that translates the hidden states of the LSTMs using a sigmoid function. The sigmoid function constrains the output values between 0 and 1, aligning with the range established by the Min-Max scaling method applied during data preparation, ensuring consistency in the network's predictions. Moreover, the sigmoid function's smooth gradient prevents extreme changes in the output, which aids in stabilizing the learning process during training.

D. Custom Loss Function

We design a loss function that incorporates the mean squared error (MSE) and a heuristic term penalizing predicting small in magnitude time shifts than in the data which could be unsafe. MSE minimizes the average squared discrepancy of predictions with respect to the target values. Specifically, the loss function based on the MSE alone enables a prediction model to generate smaller time shift estimates than in the data which could be unsafe and lead to constraint violation. Thus, beside the MSE in the loss function, we augment a penalty term, which is the mean squared ReLU (MSReLU), with a penalty weight η, i.e.,

$$\mathcal{L}_{\texttt{Total}} = \mathcal{L}_{\texttt{MSE}} + \eta \cdot \mathcal{L}_{\texttt{MSReLU}}, \tag{14}$$

where

$$\mathcal{L}_{\texttt{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \quad \mathcal{L}_{\texttt{MSReLU}} = \frac{1}{N}\sum_{i=1}^{N}(\texttt{ReLU}(\hat{y}_i - y_i))^2, \tag{15}$$

and where $\hat{y}_i$ and y_i denote the predicted time shift and the target time shift, respectively, and N is the total number of samples. MSReLU applies penalties when prediction errors are positive, considering that the target time shift is the maximum time shift to satisfy the constraints. In the other setting with the Deputy spacecraft from the positive V-bar direction, the MSReLU function becomes $\mathcal{L}_{\texttt{MSReLU}} = \frac{1}{N}\sum_{i=1}^{N}(\texttt{ReLU}(y_i - \hat{y}_i))^2$. Note that the use of the square in MSReLU term enhances differentiability and improves unit matching with MSE term.

E. Hyperparameter Tuning

In optimizing hyperparameters for the LSTM-based prediction model, we use the grid search algorithm to find the optimal hyperparameter combination within the accessible hyperparameter space. The hyperparameters are defined as follows: A hidden state space $\{h \in \mathbb{N} \mid 2^s,\ s = 6, 7, 8\}$, dropout rate $\delta \in \{0.1, 0.15, \cdots, 0.3\}$, batch size space $\{N \in \mathbb{N} \mid 2^s,\ s = 7, 8, 9, 10, 11\}$, and the penalty weight $\eta \in \{0.5, 1.0, 10, 100\}$ in Eq. (14). We adapt the learning rate based on the phase-specific window size: a learning rate of 0.001 is used for the

first phase (window sizes 1 and 2), and a reduced rate of 0.0002 for the second phase (window size 3) to enhance precision and stability as the spacecraft approaches docking. We utilize a maximum of 300 epochs and apply an early stopping mechanism with a patience parameter of 15 epochs to save training time and prevent overfitting.

Moreover, to enhance the robustness and generalizability of our model, we incorporate a 10-fold cross-validation process. This method splits the dataset into 10 subsets, using each part once as the validation set while the remaining 9 parts are used as a training set. This approach ensures that the model is tested across a diverse range of scenarios. The optimal hyperparameters for each window size were those that minimized the mean validation loss of the cross-validation process, which was set to be the same custom loss function used during training. This consistency in loss functions across training and validation stages ensures that the performance metrics are directly comparable, enhancing the reliability of our model evaluation. These findings are summarized in Table 1 to outline the best-performing configurations.

Window Size	LSTM Hidden Size	Dropout	Batch Size	Penalty Weight
1	128	0.3	512	0.5
2	256	0.25	512	1.0
3	64	0.2	1024	0.5

Table 1 Tuned model hyperparameters for each window size.

F. Hybrid Prediction Algorithm

We combine a deep learning model with a time shift governor that uses the bisection algorithm. The deep learning model computes the time shift as a function of the states which is then verified through forward simulations; from state space to the time shift, and a numerical approach to guarantee the validity of if the verification fails, the algorithm reverts to the bisection between TSG. This approach reduces the computing time of the time shift parameter in most of the cases. More specifically, our proposed hybrid algorithm computes a time shift $t_{\texttt{back}}$ based on $W(t_k)$ as follows:

1) **Time Shift Estimation**: Every update period $P_{\texttt{TSG}}$, our deep-learning model estimates the time shift $t_{\texttt{back}}$ corresponding to $W(t_k)$, e.g., $t_{\texttt{back}} = \Gamma_{\texttt{Pred}}(W(t_k))$.
2) **Virtual Target State Computation**: The state of the virtual target, $X_v(X_c(t), t_{\texttt{back}})$, is updated to the state of the Chief spacecraft corresponding to the estimated time shift in Eq. (11).
3) **Estimated Time Shift Verification**: We check constraint satisfaction over the predicted trajectories of the Deputy and Chief spacecraft, for a sufficiently long prediction horizon $T_{\texttt{TSG}}$, with the virtual target corresponding to the estimated time shift from the deep-learning model, based on Eqs. (2) and (4). If the estimated time shift parameter is verified, the virtual target is updated corresponding to the time shift; otherwise, the previous virtual target holds.
4) **Verification Failure Handling**: Conventional TSG based on bisections is applied to determine the time shift.

Algorithm 1 describes the hybrid prediction algorithm.

V. Simulation Results

In this paper, we demonstrate the deep learning-accelerated TSG (DL-TSG) via the Crew-3 mission scenario. We design a scenario with an initial distance of about 50 km between two spacecraft: the Chief spacecraft orbits along the orbit of the International Space Station (ISS), and the Deputy spacecraft starts 50 km behind the Chief spacecraft.

Algorithm 1 Hybrid Prediction Algorithm

$k_1, k_2 = 0, 0$;
for k in $\{1, 2, 3, \ldots, N_{\texttt{sim}}\}$ **do**
  Estimate a time shift parameter using the deep learning model: $t_{\texttt{back}} = \Gamma_{\texttt{Pred}}(W(t_k))$;
  $1_{\texttt{safe}}$ = Validation$(t_k, X_c, X_d, t_{\texttt{back}}, T_{\texttt{TSG}})$;
  if $\mathbb{1}_{\text{safe}}$ **then**
    if $|t_{\texttt{back}}(t_k) - t_{\texttt{back}}(t_{k-1})| < \epsilon \ll 1$ **then**
      $k_1 = k_1 + 1$;
      if $k_1 < N_1$ **then**
        Use $t_{\texttt{back}}(t_k)$ computed from the deep learning model.
      else
        Compute $t_{\text{back}}(t_n)$ using the bisection algorithm.
        $k_1 = 0$;
      end if
    end if
  else if $1_{\text{safe}} = \texttt{False}$ **then**
    if $k_2 < N_2$ **then**
      $t_{\texttt{back}}(t_k) = t_{\texttt{back}}(t_{k-1})$;
      $k_2 = k_2 + 1$;
    else
      Compute $t_{\text{back}}(t_n)$ using the bisection algorithm.
      $k_2 = 0$;
    end if
  end if
end for

Algorithm 2 Validation$(t_k, X_c, X_d, t_{\texttt{back}}, T_{\texttt{TSG}})$

Validate the estimated time shift by checking constraint satisfaction over the predicted trajectory for the horizon $\tau_{\texttt{pred}} = [t_k, \cdots, t_k + T_{\texttt{TSG}}]$ using Eqs. (2), (4), and (11):
$\mathbb{1}_{\texttt{safe}} = \texttt{True}$
for $\tau \in \tau_{\texttt{pred}}$ **do**
  Compute inequality constraints $h_1(X_c, X_d)$, $h_2(X_c(\tau + t_{\texttt{back}}), X_d, u_d(\cdot))$;
  if $\|p(X_d - X_c)\| \leq \gamma_1$ **then** compute $h_3(X_c, X_d)$;
  if $\max\{h_i,\ i = 1, 2, 3\} > 0$ **then**
    $\mathbb{1}_{\texttt{safe}} = \texttt{False}$;
    break;
  end if
  Forward propagation of X_c, X_d, X_v for one prediction step, $\Delta\tau$.
end for
return $\mathbb{1}_{\text{safe}}$.

A. Simulation Specifications

In this simulation, we consider the application of DL-TSG to Crew-3 RD mission. We consider the Chief spacecraft following the reference orbit, where the orbital elements are specified in Table (2). The orbit has a period of 92.97 minutes. The one orbital period P_{ref} is chosen as the prediction horizon for the TSG at a time instant τ, i.e., the prediction is performed over the time interval, $[\tau, \tau + P_{\text{ref}}]$. The gravitational parameter is $\mu = 398600.4418\text{km}^3/\text{sec}^2$.

The nominal controller of the Deputy spacecraft is the DTLQ controller in Eq. (4) with an LQR gain $K(t_k)$ computed for the following state and control weighting matrices, Q and R:

$$Q = \text{diag}(10, 10, 10, 1, 1, 1), \quad R = \text{diag}(1, 1, 1). \tag{16}$$

The LQR gain $K(t_k)$ is computed from a linearized model at the virtual target, corresponding to $t_{\texttt{back}}$.

In our simulations, the learning-based TSG manages the constraints specified in Eqs. (7), (8), and (10) with parameters as follows: The half-cone angle α is set to 20°, the thrust magnitude limit $u_{\max}$ to 0.5 m$\cdot$ s^{-2}, and the approach velocity constraints are set with $\gamma_1 = 5$ km for distance, $\gamma_2 = 20$ rad $\cdot$ s^{-1} for angular rate, and $\gamma_3 = 0.001$ km$\cdot$ s^{-1} for relative velocity limit. The Crew Dragon spacecraft, utilized in the Crew-3 mission, has a launch mass of 12,519 km [24] and is equipped with 16 Draco engines, each capable of a maximum thrust of 400N [25], resulting in an actual thrust magnitude of 0.5112 m$\cdot$ s^{-2}. To ensure safety, we have conservatively set the thrust magnitude limit to 0.5m$\cdot$ s^{-2}.

Table 3 summarizes the estimated initial conditions of the Deputy spacecraft with respect to the Chief spacecraft in the ECI frame. At the beginning of this simulation, the Deputy is about 44.3752 km forward in the orbital track from the Chief. This simulation employs Monte Carlo methods to introduce variations, where 1000 different initial states of the Deputy spacecraft are tested to assess the robustness and reliability of the control systems.

To demonstrate the robustness of the DL-TSG, perturbations δX are added to the initial state of the Deputy spacecraft $\bar{X}_d(t_0)$, e.g., $X_d(t_0) = \bar{X}_d(t_0) + \delta X$. The perturbations on position and velocity are designed as, respectively,

$$p(\delta X) \sim \mathcal{N}(0, \sigma^2_{\texttt{pos}} \cdot [I]_3), \; v(\delta X) \sim \mathcal{N}(0, \sigma^2_{\texttt{vel}} \cdot [I]_3), \tag{17}$$

where standard deviations of position $\sigma_{\texttt{pos}}$ and of velocity $\sigma_{\texttt{vel}}$ are set to be a tenth of the initial relative distance of the Deputy spacecraft with respect to the Chief spacecraft and an hundredth of the initial relative velocity, respectively, e.g., $\sigma_{\texttt{pos}} = (0.1) \cdot \|p(\bar{X}_d(t_0) - X_c(t_0))\|$ and $\sigma_{\texttt{vel}} = (0.01) \cdot \|v(\bar{X}_d(t_0) - X_c(t_0))\|$. Note that we select the valid initial perturbed Deputy spacecraft states $X_d(t_0)$ based on two criteria:

- The states satisfy all the imposed constraints.
- The states have an initial time shift parameter that avoids constraint violation over the predicted trajectory for the prediction horizon $P_{\texttt{ref}}$.

B. Results

Figure 4 illustrates the Deputy spacecraft trajectories in the VNB frame for 100 different initial conditions, marked as purple circles, using the DL-TSG during RD missions. The DL-TSG provides time-shifted Chief spacecraft trajectories, marked as magenta asterisks, to enforce various constraints for the Deputy spacecraft. For all simulations, the Deputy spacecraft with DL-TSG is capable of completing the rendezvous with the Chief spacecraft, marked as a black circle, within the Line of Sight (LoS) cone, as shown in Fig. 4.

SMA, a	Eccentricity, e	Inclination, i	RAAN, Ω	Argp, ω
6798.281637 [km]	0.000551	0.900516 [rad]	5.909781 [rad]	1.872335 [rad]

Table 2 Classical orbital elements of the reference ISS orbit.

δx_0 [km]	δy_0 [km]	δz_0 [km]	$\delta \dot{x}_0$ [km/s]	$\delta \dot{y}_0$ [km/s]	$\delta \dot{z}_0$ [km/s]
-25.9809	27.8498	22.7715	-0.0350	-0.0066	-0.0234

Table 3 Expected initial state of the Deputy relative to the Chief, e.g., $\mathbb{E}\big[X_d(t_0) - X_c(t_0)\big]$.

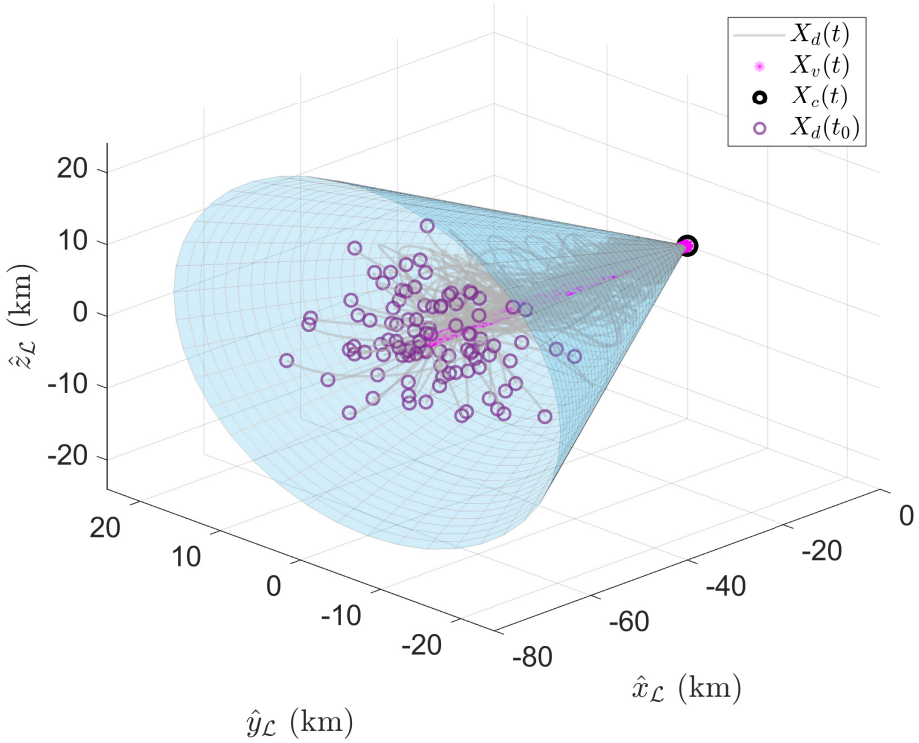

Fig. 4 Trajectories displayed in the local VNB frame: the deputy spacecraft's path (blue line) closely follows the virtual target's trajectory (magenta asterisk).

Figure 5 shows the mean of the relative position and velocity of the Deputy spacecraft with respect to the Chief spacecraft and the virtual target, respectively, for two orbit periods in the ECI frame. In Fig. 5a, the Deputy spacecraft starts 50 km away from the Chief spacecraft, behind the Chief spacecraft along the orbital track, and approaches the Chief spacecraft. During the one orbit period, the mean of the Deputy spacecraft trajectories reaches close to the trajectory of the Chief spacecraft in Figs. 5a, 5b. Then, the mean of the relative position and velocity of the Deputy spacecraft with respect to the Chief spacecraft maintains near zero after one orbit period, which means the Deputy spacecraft stays in the vicinity of the Chief spacecraft. Figures 5c and 5d provide the mean of the virtual target trajectories that are the closest target reference, along the reference orbital track, to the nominal closed-loop system to satisfy the constraints. Thus, the relative distance and velocity of the Deputy spacecraft with respect to the virtual target is smaller than that with respect to the Chief.

In Fig. 6, the time histories of three inequality constraints are presented from Monte Carlo simulations with perturbed initial conditions. Figure 6a provides the LoS constraint evolution during the RD mission, and the DL-TSG is capable of enforcing the LoS cone angle constraint h_1 for the perturbed initial conditions. Figure 6c shows the time histories of the relative velocity constraint h_3 given varying initial relative velocities, and DL-TSG is able to handle this constraint during the RD missions. Note that the thrust limit is applied

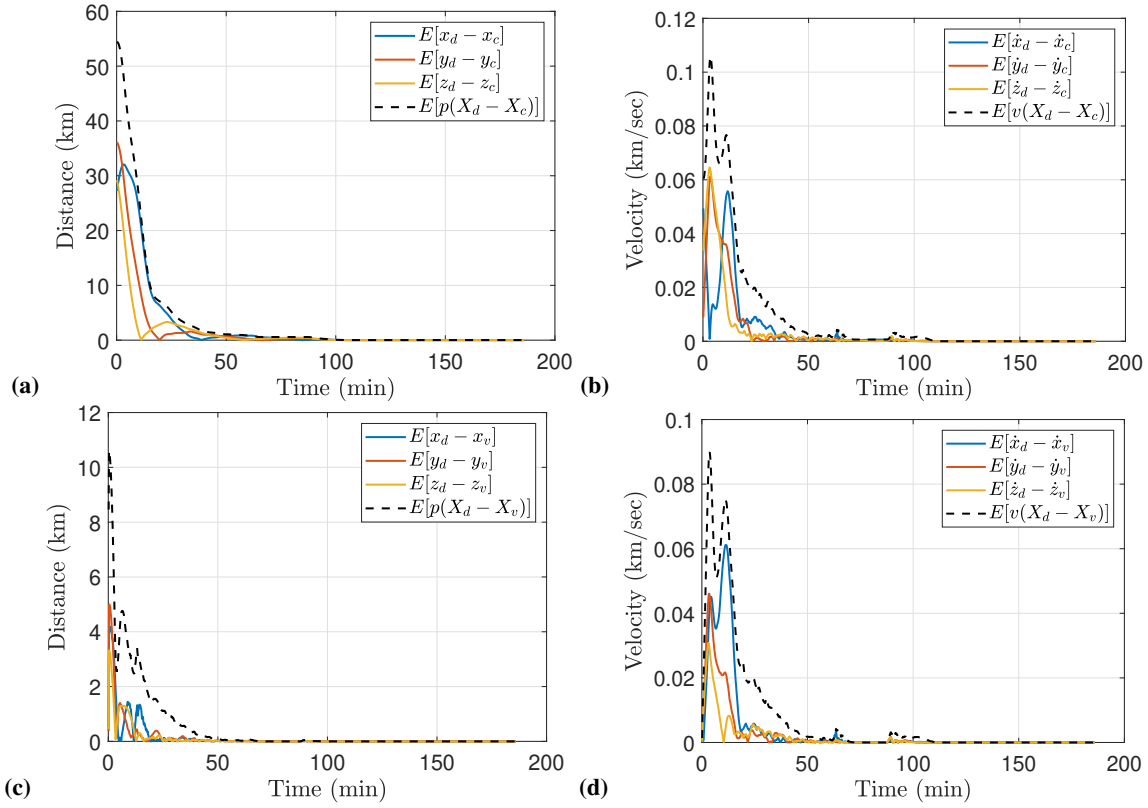

Fig. 5 Time histories of relative a) position; b) velocity to the Chief spacecraft; c) position; d) velocity to the virtual target.

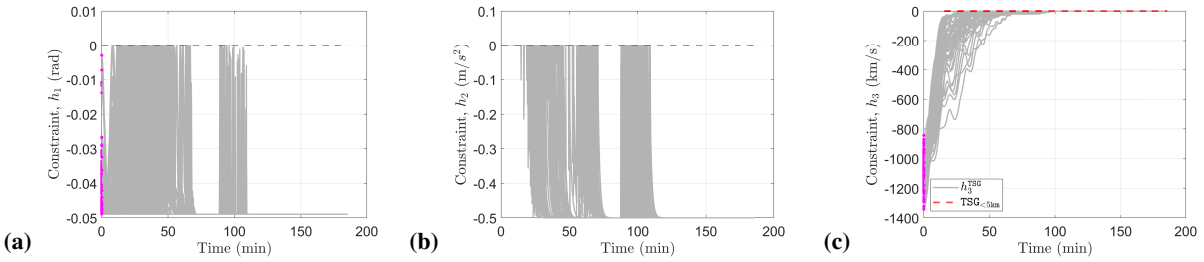

Fig. 6 Time histories of a) LoS cone angle constraint h_1; b) thrust constraint h_2; c) approach velocity constraint h_3.

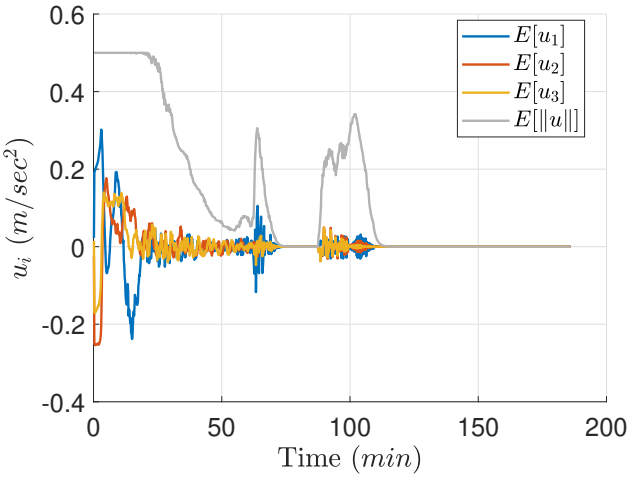

Fig. 7 Control input history of time shift throughout the mission.

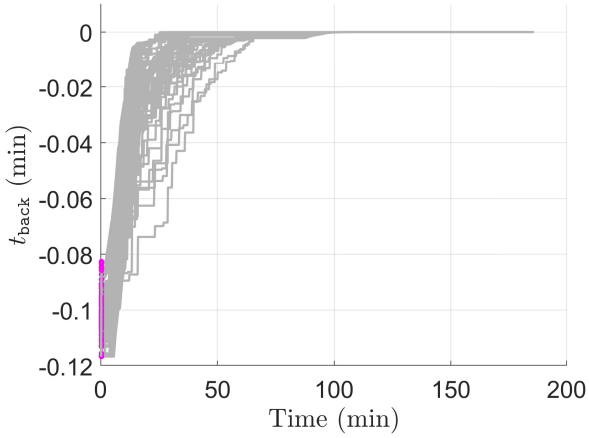

Fig. 8 Time history of time shift throughout the mission.

by the saturation function in Eq. (9), instead of DL-TSG, but it narrows the feasible set of the time shift by applying the limited thrust in the validation process. The results of the Monte Carlo simulations show that the DL-TSG is able to enforce these constraints because the LSTM cell in the DL-TSG addresses complex patterns in time series inputs to estimate a time shift parameter.

Figure 8 illustrates the trajectories of the time shift parameter t_{back} as a function of time in a Monte Carlo campaign. Starting with a valid initial time shift $t_{\text{back}}(0)$, the parameter gradually increases to zero using DL-TSG. In other words, the Deputy spacecraft is tracking the virtual target corresponding to the increasing time shift without constraint violation and stabilizes to the Chief spacecraft when the time shift reaches the zero value. During the training of the deep-learning model, the MSE term guides the time shift estimation to the optimal solution, and the MSReLU term guides the estimation to avoid solutions that result in constraint violations. Thus, the DL-TSG provides the resulting time shift trajectories, which are the maximum admissible time shifts to avoid constraint violations, considering the standard LQ controller as a nominal closed-loop system.

VI. Conclusion

This study provides and demonstrates a novel framework that integrates a Time Shift Governor (TSG) with a deep-learning model, called DL-TSG, to accelerate the performance of TSG during rendezvous and docking (RD) missions in the Two-Body problem setting. By adding the deep-learning model in our Monte Carlo campaign, DL-TSG can avoid 95.4% iterative processes in the bisection-based TSG. It showcases its capability to enforce state and control constraints during close proximity operations, including approach direction, relative velocity, and thrust limits. A target reference computed from DL-TSG guides the closed-loop system to avoid constraint violations via a time-shifted state trajectory of the Chief spacecraft. The Deputy spacecraft conducts the RD mission until achieving the target reference, which aligns with the Chief spacecraft. We demonstrate the robustness of the DL-TSG with respect to perturbations on initial states via Monte Carlo simulations.

References

[1] Hartley, E. N., "A tutorial on model predictive control for spacecraft rendezvous," 2015 European Control Conference (ECC), IEEE, 2015, pp. 1355–1361.

[2] Dong, H., Hu, Q., and Akella, M. R., "Safety control for spacecraft autonomous rendezvous and docking under motion constraints," Journal of Guidance, Control, and Dynamics, Vol. 40, No. 7, 2017, pp. 1680–1692.

[3] Dong, H., Hu, Q., and Akella, M. R., "Dual-quaternion-based spacecraft autonomous rendezvous and docking under six-degree-of-freedom motion constraints," Journal of Guidance, Control, and Dynamics, Vol. 41, No. 5, 2018, pp. 1150–1162.

[4] Weiss, A., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," IEEE Transactions on Control Systems Technology, Vol. 23, No. 4, 2015, pp. 1638–1647. https://doi.org/10.1109/TCST.2014.2379639.

[5] Karr, C. L., and Freeman, L. M., "Genetic-algorithm-based fuzzy control of spacecraft autonomous rendezvous," Engineering Applications of Artificial Intelligence, Vol. 10, No. 3, 1997, pp. 293–300.

[6] Huang, X., Li, S., Yang, B., Sun, P., Liu, X., and Liu, X., "Spacecraft guidance and control based on artificial intelligence: Review," Acta Aeronaut. Astronaut. Sin, Vol. 42, 2021, p. 524201.

[7] Wang, X., Wang, G., Chen, Y., and Xie, Y., "Autonomous rendezvous guidance via deep reinforcement learning," 2020 Chinese Control And Decision Conference (CCDC), IEEE, 2020, pp. 1848–1853.

[8] Hovell, K., and Ulrich, S., "Deep reinforcement learning for spacecraft proximity operations guidance," Journal of spacecraft and rockets, Vol. 58, No. 2, 2021, pp. 254–264.

[9] Federici, L., Benedikter, B., and Zavoli, A., "Deep learning techniques for autonomous spacecraft guidance during proximity operations," Journal of Spacecraft and Rockets, Vol. 58, No. 6, 2021, pp. 1774–1785.

[10] Qu, Q., Liu, K., Wang, W., and Lü, J., "Spacecraft proximity maneuvering and rendezvous with collision avoidance based on reinforcement learning," IEEE Transactions on Aerospace and Electronic Systems, Vol. 58, No. 6, 2022, pp. 5823–5834.

[11] Broida, J., and Linares, R., "Spacecraft rendezvous guidance in cluttered environments via reinforcement learning," 29th AAS/AIAA Space Flight Mechanics Meeting, American Astronautical Society, 2019, pp. 1–15.

[12] Xu, X., Zuo, L., and Huang, Z., "Reinforcement learning algorithms with function approximation: Recent advances and applications," Information sciences, Vol. 261, 2014, pp. 1–31.

[13] Tipaldi, M., Iervolino, R., and Massenio, P. R., "Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges," Annual Reviews in Control, Vol. 54, 2022, pp. 1–23.

[14] Frey, G. R., Petersen, C. D., Leve, F. A., Garone, E., Kolmanovsky, I. V., and Girard, A. R., "Time shift governor for coordinated control of two spacecraft formations," IFAC-PapersOnLine, Vol. 49, No. 18, 2016, pp. 296–301.

[15] Frey, G. R., Petersen, C. D., Leve, F. A., Garone, E., Kolmanovsky, I. V., and Girard, A. R., "Parameter governors for coordinated control of n-spacecraft formations," Journal of Guidance, Control, and Dynamics, Vol. 40, No. 11, 2017, pp. 3020–3025.

[16] Kim, T., Liu, K., Kolmanovsky, I., and Girard, A., "Time shift governor for constraint satisfaction during low-thrust spacecraft rendezvous in near rectilinear halo orbits," 2023 IEEE conference on control technology and applications (CCTA), IEEE, 2023, pp. 418–424.

[17] Kim, T., Kolmanovsky, I., and Girard, A., "Time Shift Governor for Spacecraft Proximity Operation in Elliptic Orbits," AIAA SCITECH 2024 Forum, 2024, p. 2452.

[18] Han, Z., Zhao, J., Leung, H., Ma, K. F., and Wang, W., "A review of deep learning models for time series prediction," IEEE Sensors Journal, Vol. 21, No. 6, 2019, pp. 7833–7848.

[19] Lara-Benítez, P., Carranza-García, M., and Riquelme, J. C., "An experimental review on deep learning architectures for time series forecasting," International journal of neural systems, Vol. 31, No. 03, 2021, p. 2130001.

[20] Kolmanovsky, I. V., and Sun, J., "Parameter governors for discrete-time nonlinear systems with pointwise-in-time state and control constraints," Automatica, Vol. 42, No. 5, 2006, pp. 841–848.

[21] Yu, Y., Zhu, Y., Li, S., Wan, D., et al., "Time series outlier detection based on sliding window prediction," Mathematical problems in Engineering, Vol. 2014, 2014.

[22] Hota, H., Handa, R., and Shrivas, A. K., "Time series data prediction using sliding window based RBF neural network," International Journal of Computational Intelligence Research, Vol. 13, No. 5, 2017, pp. 1145–1156.

[23] Ahmed, D. M., Hassan, M. M., Mstafa, R. J., et al., "A review on deep sequential models for forecasting time series data," Applied Computational Intelligence and Soft Computing, Vol. 2022, 2022.

[24] Heiney, A., "Top 10 Things to Know for NASA's SpaceX Demo-2 Return," https://www.nasa.gov/humans-in-space/top-10-things-to-know-for-nasas-spacex-demo-2-return/, July 2020. Retrieved 24 July 2020.

[25] SpaceX, "Dragon," https://www.spacex.com/vehicles/dragon, 2024. Accessed: 26 April 2024.