POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Structured Approach to Analyse Logistics Risks in the Blood Transfusion Process

Original
A Structured Approach to Analyse Logistics Risks in the Blood Transfusion Process / Cagliano, A.C., Grimaldi, S., Rafele, C.. - In: JOURNAL OF HEALTHCARE RISK MANAGEMENT. - ISSN 2040-0861. - ELETTRONICO. - (2021). [10.1002/jhrm.21458]

Availability:
This version is available at: 11583/2862374 since: 2021-01-18T12:20:57Z

Publisher:
Wiley Online

Published
DOI:10.1002/jhrm.21458

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Wiley postprint/Author's Accepted Manuscript

This is the peer reviewed version of the above quoted article, which has been published in final form at http://dx.doi.org/10.1002/jhrm.21458.This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.

(Article begins on next page)

27 November 2026

Hybrid Model Predictive Control with Physics-Informed Neural Network for Satellite Attitude Control

Carlo Cena†
Department of Electronics and Telecommunications, Politecnico di Torino
Torino, Italy
carlo.cena@polito.it

Mauro Martini
Department of Electronics and Telecommunications, Politecnico di Torino
Torino, Italy
mauro.martini@polito.it

Marcello Chiaberge
Department of Electronics and Telecommunications, Politecnico di Torino
Torino, Italy
marcello.chiaberge@polito.it

***Abstract*—Reliable spacecraft attitude control depends on accurate prediction of attitude dynamics, particularly when model-based strategies such as Model Predictive Control (MPC) are employed, where performance is limited by the quality of the internal system model. For spacecraft with complex dynamics, obtaining accurate physics-based models can be difficult, time-consuming, or computationally heavy. Learning-based system identification presents a compelling alternative; however, models trained exclusively on data frequently exhibit fragile stability properties and limited extrapolation capability. This work explores Physics-Informed Neural Networks (PINNs) for modeling spacecraft attitude dynamics and contrasts it with a conventional data-driven approach. A comprehensive dataset is generated using high-fidelity numerical simulations, and two learning methodologies are investigated: a purely data-driven pipeline and a physics-regularized approach that incorporates prior knowledge into the optimization process. The results indicate that embedding physical constraints during training leads to substantial improvements in predictive reliability, achieving a 68.17% decrease in mean relative error. When deployed within an MPC architecture, the physics-informed models yield superior closed-loop tracking performance and improved robustness to uncertainty. Furthermore, a hybrid control formulation that merges the learned nonlinear dynamics with a nominal linear model enables consistent steady-state convergence and significantly faster response, reducing settling times by 61.52%-76.42% under measurement noise and reaction wheel friction.**

***Index Terms*—Space robotics, Machine learning, Artificial Intelligence in Mechatronics**

I. Introduction

The effectiveness of a satellite attitude control system plays an important role in influencing pointing precision, operational reliability, and spacecraft longevity. Achieving these objectives is complicated by the nonlinear nature of spacecraft dynamics, persistent environmental perturbations, and actuator constraints. Reaction wheels (RWs), widely adopted for their smooth and accurate torque generation [1], introduce additional nonlinear effects, including friction-induced losses and saturation. Meanwhile, gravity-gradient forces, atmospheric drag, and geomagnetic disturbances continuously excite the rotational motion of the spacecraft, further complicating control design [2]. Historically, these challenges have been addressed through physics-based modeling and control methodologies [3]–[6], as well as, more recently, data-driven strategies [7]–[10]. Analytical approaches provide interpretability and theoretical grounding but often rely on simplifying assumptions that limit their effectiveness under uncertainty and operational variability [11]. Conversely, machine learning models can represent highly complex behaviors directly from data but typically lack robustness guarantees, hindering their adoption in safety-critical aerospace applications [12]. These limitations motivate the development of hybrid modeling paradigms that combine physical knowledge with data-driven learning. Physics-Informed Neural Networks (PINNs) [13]–[16] offer a promising solution by embedding governing equations into the optimization process, thereby promoting consistency with physical laws and improving extrapolation capabilities. This work targets attitude transition modeling conditioned on the system state and applied control torque, enabling seamless integration within predictive control architectures. We evaluate purely data-driven and physics-augmented learning strategies using high-fidelity datasets generated in the Basilisk simulation framework [17]. Using a Multilayer Perceptron (MLP) model, we show that the incorporation of physical constraints substantially improves both prediction accuracy and closed-loop behavior when embedded, alongside a linear model, in a hybrid Model Predictive Control (MPC) framework.

A. Contribution

The primary contributions of this work can be summarized as follows:

1) We develop a learning-based framework for modeling spacecraft attitude dynamics using a MLP architecture.

† Corresponding author
This work was supported by the project PNRR-NGEU which has received funding from the MUR – DM 117/2023.

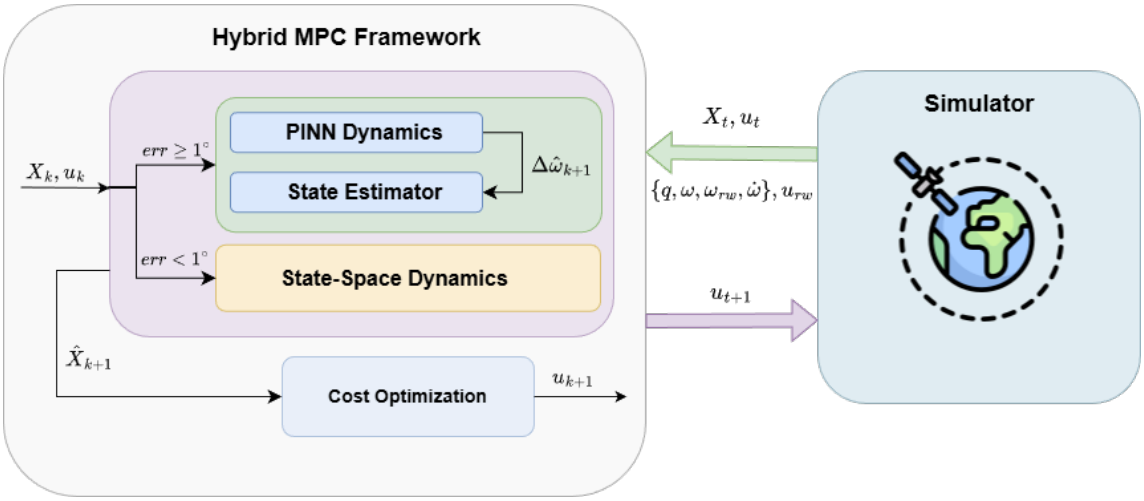


Fig. 1: Schematic of the proposed hybrid MPC with learned dynamics. When the attitude error is below 1 degree we switch from a nonlinear MPC with PINN state estimator (red) to a linear MPC (yellow).

2) We design a physics-informed loss formulation that enhances robustness and generalization, with an automatic trade-off between empirical and physical objectives.
3) We conduct a systematic comparison between purely data-driven and physics-constrained learning, demonstrating a 68.17% reduction in mean relative error over 10-step recursive prediction horizons.
4) We validate the control relevance of the learned models by embedding them into a hybrid PINN + linear MPC scheme, achieving improved closed-loop performance under realistic disturbances, including parameter uncertainties of up to 20%, observation noise of 3%, and reaction wheel friction.

B. Paper Organization

Section II formalizes the satellite attitude dynamics problem, outlines the proposed model and its associated loss functions, and describes the MPC scheme in which the learned dynamics it embedded. Section III provides a detailed description of the dataset, the neural network architectures, the baseline MPC implementations, the experimental setup, and the evaluation metrics used to assess system performance. Section IV reports and discusses the experimental findings. Finally, Section V concludes the paper and highlights the main implications of our framework.

II. Methodology

A. Satellite Attitude Dynamics

A rigid spacecraft in Earth orbit is influenced by both actuator control torques and environmental disturbance torques, which may arise from gravity-gradient effects, atmospheric drag, and geomagnetic interactions. Given the control torque in the body frame N_c, the spacecraft inertia matrix I_s, the reaction wheel inertia matrix I_{rw}, the satellite angular velocity ω, the reaction wheel angular velocity ω_{rw}, the total external torque N_e, and the skew-symmetric operator $S(\omega)$ defined in (1), the reaction-wheel-actuated attitude dynamics of the spacecraft is expressed in (2).

$$S(\omega) = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \tag{1}$$

$$\dot{\omega} = -I_s^{-1}S(\omega)I_s\omega - I_s^{-1}S(\omega)I_{rw}\omega_{rw} + I_s^{-1}N_c + I_s^{-1}N_e \tag{2}$$

The dynamics of the reaction wheels angular velocity is given in eq. (3).

$$\dot{\omega}_{rw} = I_{rw}^{-1}u_{rw} - \dot{\omega} \tag{3}$$

The spacecraft and reaction wheel angular velocities are obtained by time-integrating (2) and (3). These quantities then permit the computation of the spacecraft attitude, represented by the quaternion q, through integration of (4).

$$\dot{q} = \frac{1}{2}\Omega[\omega]q \tag{4}$$

Where $\Omega[\omega]$ is given by eq. (5).

$$\Omega[\omega] = \begin{bmatrix} 0 & -\omega_0 & -\omega_1 & -\omega_2 \\ \omega_0 & 0 & \omega_2 & -\omega_1 \\ \omega_1 & -\omega_2 & 0 & \omega_0 \\ \omega_2 & \omega_1 & -\omega_0 & 0 \end{bmatrix} \tag{5}$$

B. Problem Formulation

Developing reliable attitude control remains difficult due to the spacecraft's nonlinear dynamics and environmental uncertainties [2]. Accurate modeling is therefore crucial, especially for model-based controllers such as MPC and in the presence of state and parameter estimation errors. We propose a physics-informed neural network (PINN) to approximate the spacecraft dynamics. A physics-based loss term complements the data-driven objective, improving generalization and

robustness to noise. Fig. 1 summarizes the approach. The PINN takes the spacecraft state—satellite and reaction-wheel angular velocities—along with satellite angular acceleration and reaction-wheel control torque, and predicts the change in satellite angular velocity for integration within a nonlinear MPC scheme. When integrated into the proposed hybrid MPC, the attitude dynamics are computed either using the PINN or a linear state-space formulation, depending on the current attitude error. Specifically, when the attitude error, $err = 2arrcos(q_0)$ in Fig. 1, falls below 1 degree, the attitude dynamics estimator switches from the PINN model to the state-space model.

C. Neural Network Architecture

A MLP neural network is employed to learn the spacecraft attitude dynamics. The model input consists of the spacecraft inertia matrix concatenated with the current state, reducing dimensionality while retaining the quantities explicitly required in 2. Omitting these terms would force the network to approximate them, increasing numerical error and diverting model capacity. The network outputs a vector of size $(3S, 1)$, representing the predicted trajectory over S time steps with three variables per step. At time t, the models predict the angular-velocity change $\Delta\hat{\omega}_{t+1}$ from the input state comprising the satellite angular velocity ω_t, reaction-wheel velocity $\omega_{rw,t}$, commanded torque $u_{rw,t}$, and the spacecraft angular acceleration $\dot{\omega}_t$, obtained via first-order backward differencing.

D. Physics-Informed Training

As previously discussed, we employed a combination of two loss terms. Given a batch size *B*, the data-driven loss, L_{DD}, defined in eq. (6), corresponds to the Normalized Root Mean Squared Error (NRMSE) between the predicted change in angular velocity, $\Delta\hat{\omega}$, and the ground-truth values from the dataset. The normalization is performed using the standard deviation of the ground-truth change in angular velocity, $\sigma_{\Delta\omega}$, to scale the Root Mean Squared Error (RMSE).

$$L_{\text{DD}} = \frac{\sqrt{\frac{1}{B}\sum_i^B (\Delta\hat{\omega}_i - \Delta\omega_i)^2}}{\sigma_{\Delta\omega}} \tag{6}$$

The physics-informed penalty term L_{PI} (eq. (7)) regularizes the training of the dynamics model by embedding the system's physical laws directly into the loss function. Derived from the satellite attitude dynamics presented in Section II-A, this term consists of a weighted sum of two components: (I) the NRMSE of the predicted change in angular acceleration, $L_{\dot{\omega}}$, eq. (8), where $\dot{\omega}$ is computed using eq. (2); and (II) the MSE of the change in angular momentum, L_h, eq. (9).

$$L_{PI} = L_{\dot{\omega}} + pL_h \tag{7}$$

The MSE term on the change in angular momentum is scaled by a factor p set to $1e^{-2}$, selected through a grid search. To keep the physics-informed loss compact, the total external torque N_e in eq. (2) is set to zero, since this loss acts primarily as a regularizer enforcing general dynamical consistency. External disturbances are already reflected in the training data and are therefore captured by the data-driven loss. This design choice keeps the physics-informed loss robust and independent of environment-specific assumptions, focusing instead on intrinsic system dynamics.

$$L_{\dot{\omega}} = \frac{\sqrt{\frac{1}{B}\sum_i^B (\hat{\dot{\omega}} - \dot{\omega})^2}}{\sigma_{\dot{\omega}}} \tag{8}$$

$$\begin{aligned} L_h = &\frac{1}{B}\sum_i^B (||I_s \cdot (\omega + \Delta\hat{\omega}) + I_{rw} \cdot \hat{\omega}_{rw}|| \\ &- ||I_s \cdot (\omega + \Delta\omega) + I_{rw} \cdot \omega_{rw}||)^2 \end{aligned} \tag{9}$$

In equations (8) and (9), $\Delta\hat{\omega}$ and $\hat{\dot{\omega}}$ denote the predicted change in angular velocity and the predicted angular acceleration, respectively, where $\hat{\dot{\omega}} = \Delta\hat{\omega}/\Delta t$. $\Delta t = 0.1$ is the controller timestep. The term $\Delta\hat{\omega}_{rw}$ represents the change in RW angular velocity, computed using eq. (3) with $\hat{\dot{\omega}}$. Equation (10) summarizes the total loss, expressed as a weighted combination of the data-driven and physics-informed components controlled with the scalars α and β, where $\alpha = (1 - \beta)$, and β is constrained to the interval $[0, 1]$.

$$L = \alpha L_{DD} + \beta L_{PI} \tag{10}$$

A Lagrangian dual strategy [18], which adaptively adjusts their importance based on model performance, is used.

E. Non-linear MPC with Learned Dynamics

The objective of this work is to learn a satellite's attitude dynamics using a deep neural network, enabling more efficient and flexible spacecraft control. We evaluate the best learned models within a rest-to-rest maneuver scenario by integrating the learned dynamics into an MPC framework. This application also serves to assess how the learned model behaves when embedded in a model-based controller, particularly in terms of robustness to state-estimation errors and model uncertainties. The MPC state vector includes the quaternion q, the spacecraft angular velocity ω, the RWs angular velocity ω_{rw}, and the angular acceleration $\dot{\omega}$, such that $x = \{q, \omega, \omega_{rw}, \dot{\omega}\}$. The PINN predicts the spacecraft and RW dynamics from the current state x_t and RW torque input u_t, and these predictions are then integrated to compute the next state x_{t+1}, as expressed in equations 11–14.

$$\Delta\hat{\omega}_{t+1} = PINN_\theta(\omega_t, \omega_{rw,t}, \dot{\omega}_t, u_t, \theta) \tag{11}$$

$$\hat{\dot{\omega}}_{t+1} = \frac{\Delta\hat{\omega}_{t+1}}{\Delta t} \tag{12}$$

$$\hat{\dot{\omega}}_{rw,t+1} = I_{rw}^{-1} u_t - \hat{\dot{\omega}}_{t+1} \tag{13}$$

$$\hat{\dot{q}}_{t+1} = \frac{1}{2}\Omega[\omega_t]q \tag{14}$$

In particular, the PINN is used to estimate the angular-velocity increment $\Delta\hat{\omega}_{t+1}$, which is then used to compute the angular acceleration $\hat{\dot{\omega}}_{t+1}$ (eq. 12) and the RWs accelerations $\hat{\dot{\omega}}_{rw}$ by substituting the estimated

acceleration into eq. (3). During inference only the first predicted state is provided to the MPC as the estimated next state. The MPC's cost function is defined in eq. (15).

$$C = \sum_{k=0}^{n-1} \left(x_k^T Q x_k + u_k^T C u_k + \Delta u_k^T R \Delta u_k \right) + x_n^T Q x_n \quad (15)$$

Where u_k is an abbreviation of the commanded RWs torque u_{rw} at time step *k*, and Δu_k is the difference between the RWs torque required at the previous time step and the current one. *C* and *R* are diagonal cost matrices used to reduce the torques u_{rw} and to limit their variation, smoothing their trajectories. Their non-zero elements are equal to $1e-1$, while $Q^{13\times13}$ = diag(10000, 10000, 10000, 10000, $1e^{-2}, 1e^{-2}, 1e^{-2}, 1e^{-4}, 1e^{-4}, 1e^{-4}, 1e^{-2}, 1e^{-2}, 1e^{-2}$). Finally, *n* is the horizon of the MPC, which in our experiments is set to $n = 10$.

III. Experimental setups

A. Dataset

To train and evaluate the models, we generated a dataset using the Basilisk simulator [17]. The simulation considers a cubesat in Low Earth Orbit actuated solely by RWs and executing attitude maneuvers under an MRP feedback control module available in Basilisk. In addition to the commanded torques, the satellite is subjected to several environmental disturbances, including gravity–gradient effects, magnetic disturbances, and atmospheric drag. The relevant satellite parameters are summarized in Table I. We conducted 300 simulations by randomly

TABLE I: We show the satellite inertia matrix I_S, the RWs inertia matrix I_{rw}, the satellite mass, the maximum torque acting on the RWs u_{rw}, the maximum RWs speed ω_{rw}.

I_s	I_{rw}	Mass [Kg]	Max u_{rw} [Nm]	Max ω_{rw} [rpm]
5.700, 0.045, 0.002 0.045, 3.300, 0.012 0.002, 0.012, 6.100	0.001, 0., 0. 0., 0.001, 0. 0., 0., 0.001	58	0.05	6000

varying the initial attitude and orbital position to generate the main dataset, and an additional 50 simulations with modified satellite mass and inertia matrix to assess model robustness to approximately 10% parameter variations along all axes—an important consideration since these quantities are not known with perfect accuracy and may vary during operations. In each run, the satellite was initialized with a random reaction–wheel angular velocity uniformly sampled from the range [-300, 300] rpm. Each simulation lasted 3 minutes with a control and sampling period of 0.1 seconds, and an integration time step of 0.001 seconds using a fourth-order Runge–Kutta method. The dataset was constructed by forming input–output pairs as follows: each input x_i has shape (1, 12) and is defined as $x_i = \{\omega, \omega_{rw}, u_{rw}, \dot{\omega}\}$, while the target y_i corresponds to the change in angular velocity, i.e., $y_i = \Delta\omega_i$. The resulting dataset was randomly split into training and validation subsets using a 67–33% ratio.

B. Models

In our experiments, we evaluated multiple models with different inputs and architectures. Our goals were to achieve high performance in estimating the next attitude state from the current state and the commanded torque, while also ensuring a low inference time. Beyond varying the number of FC layers and the number of units per layer, we also assessed whether including the satellite and RWs inertia information could improve performance. To do so, we concatenated this information to the model input and again before the final layer. The most effective configuration was obtained by concatenating I_s and I_{rw} directly to the model input. After identifying the best-performing model structures, we conducted additional tests on the number of predicted steps, S. The model was trained to output all future S steps in a single forward pass, with the two losses computed by iteratively propagating the state while keeping the torques constant. The best performance was achieved with $S = 10$. During inference, only the first predicted state was used for evaluation or as the estimated state within the MPC. The best-performing model consisted of 4 FC layers with 16 units each. This configuration was then used to analyze the impact of concatenating the satellite and RWs inertia parameters from the training set, I_s and I_{rw}, to the input state. The results confirmed that including these inertia parameters yielded improved performance.

C. Traditional MPC

In this work, we compare our framework against two MPC baselines: one based on a state–space formulation (linear MPC) and another relying on a non-linear formulation. Using the MPC setup introduced in Section II-E, we apply the same input state and cost matrices together with the non-linear dynamics described in Section II-A, and modify them only when adopting the state–space model. Specifically, we define the input state as $x = \{q, w\}$, set all diagonal terms of the matrix $Q^{6\times6}$ to 1000, and leave all other parameters unchanged. The discrete state–space model was implemented following [5].

D. Evaluation Metrics & Experiments

To evaluate our models, we employed a set of metrics designed to assess their performance both as standalone regressors and as components within MPC controllers. These two perspectives correspond to the two evaluation stages carried out in our study. First, we trained several neural networks with different architectures and hyper-parameters to identify the most effective model for estimating the next attitude state from the current one and the commanded RWs torque, whose results are shown in Section IV-A. The metrics used were the Mean Relative Error (MRE) and the physics error, i.e., the physics-informed loss. Results were averaged across all time steps and simulations. For both metrics, we report the single-step prediction performance as well as the performance when predicting the next 10 steps under the assumption of constant command torque. Second, in Section IV-B, we used the best-performing neural networks to evaluate a range of non-linear

TABLE II: Results obtained when evaluating the MLP models as regressors. MLP-DD uses only the data-driven loss, while MLP-LD uses a Lagrangian dual approach.

Experiment	MRE	Physics Error	MRE self-loop	Physics Error self-loop
MLP-DD	19.67	24.38	22.63	137.34
MLP-LD	**6.26**	**17.18**	**6.96**	**95.86**

and linear MPCs. For each model, we conducted 300 Monte Carlo simulations to analyze robustness to noise. In these simulations, we added a Gaussian-distributed random error of up to 3% to each state variable, a continuously distributed random error of up to 10% to the inertia matrix, and up to 20% to the satellite mass. Additionally, RWs friction was modeled with linear dynamics up to $50\% \pm 12.5\%$ of the maximum wheel velocity. The initial attitude was randomly sampled within the interval $(\frac{\pi}{8}, \frac{\pi}{2})$. It is important to note that the neural networks were trained on noise-free simulations without RWs friction. To evaluate these experiments, we present the average steady-state error over the final minute of simulation, the time required for the attitude error to fall below 1 degree (Settling Time), and the distribution of the trajectories.

E. Implementation Details

The experiments were performed on a computer with an Intel i7-9700K 3.60GHz CPU, 64 GB of RAM, and a GeForce RTX 2080 Ti. The models and loss functions have been implemented in torch and we used the "do-mpc" library to implement the MPC with AI as dynamics estimator. The models have been trained on the GPU using a batch size of 16384, and evaluated in several edge devices: the maximum average time required was 7.91ms in a Raspberry PI 4B, while the minimum was 2.30ms in an AMD GX-412HC.

IV. Results

A. Attitude Dynamics Regressor

We compared the performance of model trained solely with the data-driven loss against the physics-informed neural network trained with both losses. Table II reports the outcomes of these experiments. Specifically, the table shows for each configuration the average MRE and physics error, both for single-step prediction and for the 10-step recursive predictions, where the model's output is recursively fed back as input. MLP-DD corresponds to the model trained only with the data-driven loss, while MLP-LD employed the Lagrangian dual method to automatically adjust the weight between data-driven and physics-informed loss. The experiments showed that the model trained with both losses achieved superior performance, improving both the physics error and the MRE. To further validate these findings and assess their statistical significance, we computed p-values using the Wilcoxon test. Following standard practice, results with p-values above 0.05 are considered not significant. All p-values were below this threshold, confirming that the observed improvements are statistically significant.

B. MPC Controllers

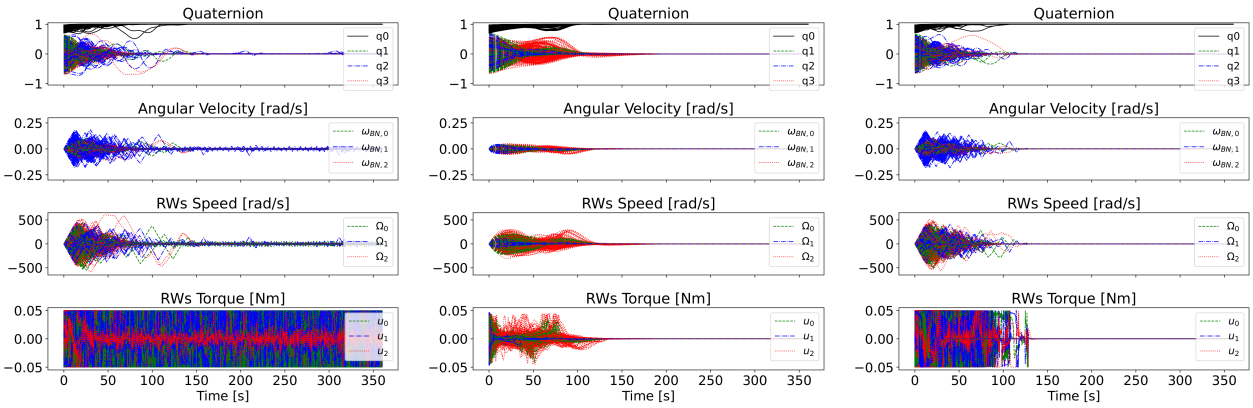
Given the results presented above, we selected the MLP-LD model as the dynamics predictor within the MPC controller to compare the performance of our best model against the two traditional MPC controllers introduced in Section III-C. In Fig. 2, we show the trajectories from 300 Monte Carlo simulations performing rest-to-rest maneuvers over 360 seconds using the PINN-based controller, the state-space MPC and the proposed hybrid MPC framework. Across these simulations, we randomly varied the orbital position and the initial satellite attitude, and introduced parameter estimation errors of up to 20% in the satellite mass and up to 10% in the inertia matrix. Additionally, state estimation was corrupted with up to 3% Gaussian noise, and the reaction wheels were subject to friction as described in Section III-D. The results indicate that all controllers can reach the desired attitude with a steady-state error below 1 degree, though the use of the MLP-LD neural network as dynamics estimator for the complete maneuver leads to a higher final error and continuous high RWs torques. The MPC using the hybrid MPC scheme (MLP-LD + Linear) instead achieves the fastest convergence despite the presence of noise, outperforming the other controllers in terms of settling time, and achieving comparable results to the linear MPC in terms of steady-state error. However, the RWs torque distributions show that this model leads to considerably higher torques than necessary. Table III reports the median settling time, defined as the time required to reach a pointing error below 1 degree, and the median steady-state error over the final minute of each trajectory, i.e., the angular difference in degrees between the current and target quaternion. The results show that the state-space MPC requires significantly more time, 114.1 seconds compared to the 43.9 seconds needed by MLP-LD + Linear model, to achieve convergence.

TABLE III: Median results of the 300 MC performed to study the robustness-to-noise of the PINN model when used in a MPC with several sources of noise.

Experiment	Steady-State Error [degrees]	Settling Time [s]
MLP-LD	0.2901	47.40
MLP-LD + Linear	**0.0023**	**43.90**
Non-Linear MPC	0.1215	186.2
Linear MPC	0.0024	114.1

V. CONCLUSIONS

In this study, we introduce a framework for learning spacecraft attitude dynamics using a multilayer perceptron (MLP) neural network trained with a loss function that combines data-driven supervision with a physics-informed penalty, where the relative weight is dynamically adjusted during training through the Lagrangian dual approach. Our analysis indicates that incorporating physical knowledge into the loss function significantly enhances model performance, resulting in notable reductions in the Mean Relative Error (MRE) for 10-step predictions. The robustness of our approach is demonstrated through 300 Monte



(a) MLP-LD (b) Linear MPC (c) MLP-LD + Linear

Fig. 2: 300 MC simulation with parameter estimation errors, state estimation noise and RWs friction for MLP-LD (left), traditional MPC with linear (center) and MLP-LD + Linear (right).

Carlo simulations with a high-fidelity simulator, which show resilient closed-loop behavior and resistance to observational noise and reaction wheel friction. We evaluated the average steady-state errors and settling times for both purely data-driven and physics-informed models, comparing them against traditional MPC controllers. Our results indicate improvements in settling time of up to 76.42%. We believe that the proposed approach exhibits strong generalization capabilities and could be extended to more complex non-linear scenarios, such as satellite berthing or docking with non-cooperative targets, or the control of under-actuated spacecraft. These situations involve greater modeling and control challenges, such as discontinuous dynamics, partial observability, and restricted actuation, which align well with the strengths of hybrid data-driven and physics-informed learning.

Acknowledgment

This work has been developed with the contribution of the Politecnico di Torino Interdepartmental Centre for Service Robotics (PIC4SeR) (https:// pic4ser.polito.it) and Argotec (https://argotecgroup.com). Computational resources were provided by HPC@POLITO (http://hpc.polito.it). The cover in Fig. 1 has been designed using resources from Flaticon.com.

References

[1] X. Xia, G. Sun, K. Zhang, S. Wu, T. Wang, L. Xia, and S. Liu, "Nanosats/cubesats adcs survey," in *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 5151–5158.

[2] J. K. Wayer, J.-F. Castet, and J. H. Saleh, "Spacecraft attitude control subsystem: Reliability, multi-state analyses, and comparative failure behavior in leo and geo," *Acta Astronautica*, vol. 85, pp. 83–92, 2013.

[3] P. Iannelli, F. Angeletti, and P. Gasbarri, "A model predictive control for attitude stabilization and spin control of a spacecraft with a flexible rotating payload," *Acta Astronautica*, vol. 199, pp. 401–411, 2022.

[4] J. Narkiewicz, M. Sochacki, and B. Zakrzewski, "Generic model of a satellite attitude control system," *International Journal of Aerospace Engineering*, vol. 2020, no. 1, p. 5352019, 2020.

[5] M. Pecorilla and F. Stesina, "Novel extended kalman filter + $\mathbf{H}_\infty$ optimal output feedback control configuration for small satellites with high pointing stability requirements," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 5, pp. 2991–3010, 2024.

[6] A. G. Romero and L. C. G. de Souza, "Satellite controller system based on reaction wheels using the state-dependent riccati equation (sdre) on java," in *Proceedings of the 10th International Conference on Rotor Dynamics – IFToMM*, K. L. Cavalca and H. I. Weber, Eds. Cham: Springer International Publishing, 2019, pp. 547–561.

[7] C. Cena, S. Bucci, and A. Balossino, "Deep reinforcement learning for under-actuated satellite attitude control and reaction wheel desaturation using solar radiation pressure," in *Proceedings of the International Astronautical Congress (IAC)*, 10 2023.

[8] W. Retagne, J. Dauer, and G. Waxenegger-Wilfing, "Adaptive satellite attitude control for varying masses using deep reinforcement learning," *Frontiers in Robotics and AI*, vol. Volume 11 - 2024, 2024.

[9] S. Wu, W. Chu, Z. Wu, W. Chen, and W. Wang, "Inertia matrix identification of combined spacecraft using a deep neural network with optimized network structure," *Advances in Space Research*, vol. 73, no. 3, pp. 1979–1991, 2024.

[10] Z. Zhang, H. Peng, and X. Bai, "Imitation learning for satellite attitude control under unknown perturbations." Pre-print techrxiv, Jul. 2025.

[11] M. Moldabekov, A. Sukhenko, Y. Orazaly, and A. Aden, "Dynamics analysis of a nonlinear satellite attitude control system using an exact linear model," *Mathematics*, vol. 11, no. 12, 2023.

[12] D. Celestini, D. Gammelli, T. Guffanti, S. D'Amico, E. Capello, and M. Pavone, "Transformer-based model predictive control: Trajectory optimization via sequence modeling," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9820–9827, 2024.

[13] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[14] S. Falas, M. Asprou, C. Konstntinou, and M. K. Michael, "Robust Power System State Estimation Using Physics-Informed Neural Networks," *IEEE Transactions on Industrial Informatics*, vol. pre-print, 2025.

[15] W. Gu, S. Primatesta, and A. Rizzo, "Physics-informed neural network for quadrotor dynamical modeling," *Robotics and Autonomous Systems*, vol. 171, p. 104569, 2024.

[16] J. Varey, J. D. Ruprecht, M. Tierney, and R. Sullenberger, "Physics-informed neural networks for satellite state estimation," in *2024 IEEE Aerospace Conference*, 2024, pp. 1–8.

[17] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A flexible, scalable and modular astrodynamics simulation framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, 2020.

[18] F. Fioretto, P. Van Hentenryck, T. W. K. Mak, C. Tran, F. Baldo, and M. Lombardi, "Lagrangian Duality for Constrained Deep Learning," in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track*, Y. Dong, G. Ifrim, D. Mladenić, C. Saunders, and S. Van Hoecke, Eds. Cham: Springer International Publishing, 2021, pp. 118–135.