Data-driven global linear modelling of satellite attitude dynamics via Koopman operator

Jianing Zhang

Faculty of Engineering and Information Sciences, University of Wollongong, Wollongong, Australia

jz701@uowmail.edu.au

Abstract. Satellite attitude dynamics are inherently nonlinear, and modelling becomes particularly challenging during large-angle manoeuvres or actuator saturation. Traditional first-principles modelling methods often require precise system parameters, yet even then, many complex behaviours may not be captured. To address this, this paper proposes a data-driven modelling method based on Koopman operator theory, which learns a global linear representation of satellite attitude dynamics from simulated trajectory data. Specifically, the model constructs a high-dimensional enhanced state space using a set of observable values, enabling the Koopman model to approximate nonlinear dynamics through linear systems. This approach achieves a relatively accurate prediction of angular velocity and attitude trajectories. Simulation results show that the Koopman-based model is effective and can be applied for real-time attitude prediction and satellite control.

Keywords: satellite attitude dynamics, Koopman operator, data-driven modelling, quaternion representation.

1. Introduction

Satellite attitude control plays a critical role in space missions, including Earth observation, interstellar exploration, space telescopes, and communication systems [1-2]. The attitude of a satellite refers to its direction relative to an inertial reference frame in three-dimensional space. This frame determines the direction of sensors, antennas, and propulsion systems; therefore, satellite attitude control requires extremely high precision. The attitude of a satellite will change over time due to internal torque control and external interference, and these changes all follow the control rotation dynamics equation.

Rotational dynamics are nonlinear, typically represented by quaternions or rotation matrices, and governed by nonlinear rigid body equations of motion. To avoid singularities or discontinuities, satellite attitudes are usually parameterised using quaternions or Euler angles, which are mathematical tools for describing 3D directions [3-4]. Rotation control is typically achieved through actuators such as reaction wheels, control torque gyroscopes, and thrusters, though these are subject to physical limitations such as torque saturation, rate constraints, and actuation delay [5-6]. In addition, accurate simulation is further complicated by external influences such as gravity gradient, drag, and solar pressure, which increase the complexity of control design [7-8].

Simplified linear time invariant (LTI) models are widely used in practice because of their mathematical tractability and suitability for controller design. These models are particularly effective for small-angle manoeuvres, where system dynamics can be reliably approximated by linearisation. Controllers such as proportional-derivative (PD) and linear quadratic regulator (LQR) have been widely adopted and have become standard in aerospace applications [9]. However, LTI-based methods neglect nonlinearities, coupling effects, and the true geometric shape of rotational motion present in large-angle rotations. These factors are particularly important for agile spacecraft, flexible-body satellites, or tasks operating at low actuation margins. In these cases, linear assumptions break down and controller performance suffers.

To overcome these limitations, recent studies have explored data-driven approaches such as neural networks (NNs), recurrent neural networks (RNNs), and Gaussian processes (GPs) trained on simulated or in-orbit telemetry data [10-11]. While these methods can achieve high predictive accuracy, their black-box nature makes physical interpretability, stability analysis, and integration with optimisation-based control frameworks (e.g., Model Predictive Control - MPC) difficult [12]. In contrast, Koopman operator theory offers a powerful alternative by enabling the modelling of nonlinear dynamical systems as linear systems in a lifted state-space [13-14]. The key idea is to transform the original system states into a higher-dimensional space of observables, where system evolution can be approximated by a linear operator (the Koopman operator), facilitating both analysis and control [15].

In this paper, we adopt a classical Koopman-based modelling approach. The lift function is constructed based on relevant domain knowledge, including quaternions, angular velocity, and their nonlinear interactions. This avoids training encoders or neural networks while still enabling global linear modelling in higher-dimensional spaces [16]. Quaternions and angular velocities are elevated to a higher-dimensional observable space composed of physically important features such as rotation matrices, interaction terms, and triangular expressions.

2. Problem formulation and data generation

This study aims to learn a predictive model of satellite attitude dynamics from purely simulated data without requiring prior knowledge of physical parameters. The satellite is modelled as a rigid body in three-dimensional space undergoing free rotation under external torques. Its rotational state is represented by a quaternion $q(t) = [q_0, q_1, q_2, q_3]^T \in \mathbb{R}^4$ and an angular velocity vector $\omega(t) = [\omega_x, \omega_y, \omega_z]^T \in \mathbb{R}^3$, forming the state vector $x(t) = [q(t)^T, \omega(t)^T]^T \in \mathbb{R}^7$.

Quaternions are preferred over Euler angles because they provide numerical stability, compactness, and freedom from singularities. Euler angles suffer from singularities such as gimbal lock, particularly when the pitch angle approaches $\pm 90°$, causing a loss of one rotational degree of freedom. In contrast, quaternions provide a globally non-singular representation of 3D rotations, ensuring smooth and continuous attitude tracking. Geometrically, a quaternion defines a rotation about an axis $u \in \mathbb{R}^3$ by an angle σ via $q = [\cos(\sigma/2), u^T \sin(\sigma/2)]^T$ (Figure 1), ensuring a one-to-one mapping to a rotation matrix on the special orthogonal group SO(3). Thus, quaternions allow continuous, globally valid tracking of satellite orientation.
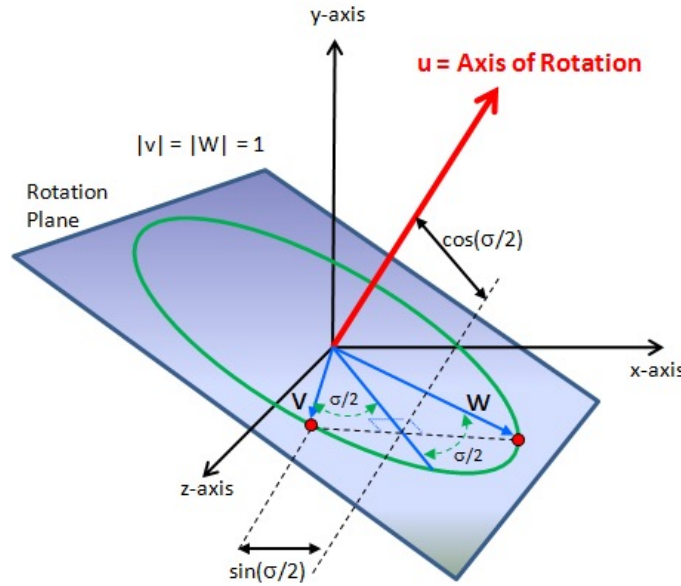

Figure 1. Geometric interpretation of a unit quaternion representing a 3D rotation

The rotation shown in Figure 1 is defined by an axis $u \in \mathbb{R}^3$ and an angle σ, where the quaternion is expressed as $q = [\cos(\sigma/2), u^T \sin(\sigma/2)]^T$. Here, σ denotes the total angle of rotation about the axis u, and the diagram illustrates the relation between the rotation plane and the quaternion components. The unit vector u specifies the direction of the rotation axis in three-dimensional space, and is typically expressed in the body-fixed or inertial reference frame depending on the application.

While quaternions offer significant advantages in representing three-dimensional rotations, particularly in avoiding singularities and ensuring smooth interpolation, Euler angles remain the standard representation in many real-world satellite sensors and control systems due to their intuitive physical interpretation. In practice, raw attitude measurements are often recorded as Euler angles and then converted to quaternions for use in dynamic modelling and prediction. After prediction, the resulting quaternion outputs are transformed back into Euler angles for analysis and visualisation.

The conversion from a unit quaternion $q(t) = [q_0, q_1, q_2, q_3]^T$ to Euler angles (ϕ, θ, ψ), corresponding to roll, pitch, and yaw, respectively, in a XYZ rotation sequence (Figure 2), is given by:

$$\phi = \text{arctan2}\left(2\left(q_0 q_1 + q_2 q_3\right), 1 - 2\left(q_2^2 + q_3^2\right)\right)$$

$$\theta = \arcsin2(2(q_0q_2 - q_3q_1))$$

$$\psi = \arctan2\left(2\left(q_0q_3 + q_1q_2\right), 1 - 2\left(q_2^2 + q_3^2\right)\right)$$
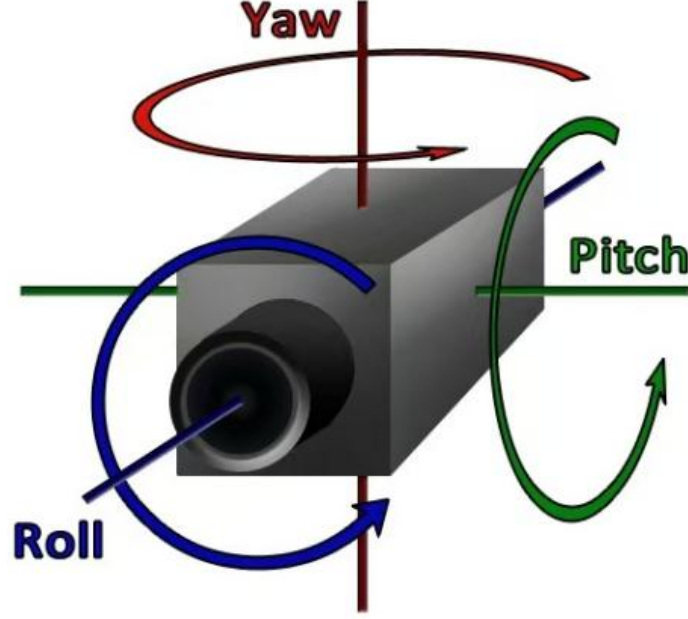

Figure 2. Three-axis rotation convention for satellite attitude: roll (rotation about the x-axis), pitch (rotation about the y-axis), and yaw (rotation about the z-axis)

The time evolution of the satellite's rotational dynamics is governed by the rigid-body equations of motion. The quaternion dynamics follow the differential equation:

$$\dot{q} = \frac{1}{2} Q(\omega)q$$

where $Q(\omega) \in \mathbb{R}^{4\times4}$ is a matrix-valued function of angular velocity given by

$$Q(\omega) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$

The angular velocity evolves according to Euler's rotational equation:

$$J\dot{\omega} = J\omega \times \omega + \tau$$

where $J \in \mathbb{R}^{3\times3}$ is the moment of inertia matrix and $\tau \in \mathbb{R}^3$ is the control torque input. For simulation purposes, we assume a symmetric rigid body with diagonal inertia $J = 1/6\, ma^2 I3$, where m = 7.0 kg is the mass and a = 0.1 m is the edge length of the cubic satellite body. Torques are applied in all three axes, randomly generated within a maximum bound of $\pm\, \tau_{max} = 0.001$ Nm.

To generate training data, we simulate N = 500 random attitude trajectories using MATLAB. Each trajectory spans T = 10 s with a fixed time step Δt = 0.1 s, resulting in 100 steps per trajectory and a total of 500×100=50,000 data samples. Each simulation begins with a randomly generated initial quaternion uniformly sampled on the unit 3-sphere S^3 and a randomly

perturbed angular velocity vector (Figure 3). At each step, a control input $\tau(t)$ is applied, and the state is integrated forward using explicit Euler integration. After integration, the updated quaternion is re-normalised to maintain unit norm. This guarantees valid orientation representation throughout the simulation. The resulting dataset includes triples (x_t, τ_t, x_{t+1}) capturing the full state transition under known torque input, and is stored in structured arrays for subsequent modelling.
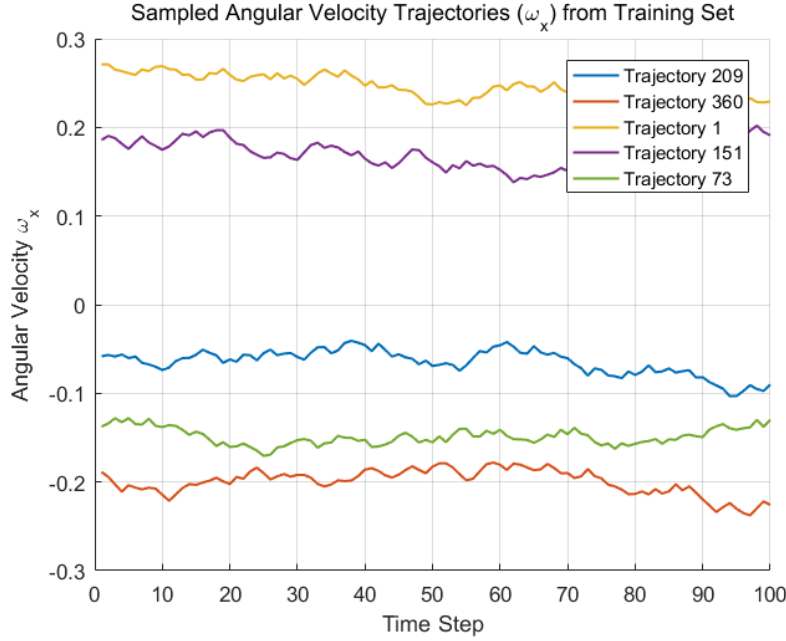

Figure 3. Five representative angular velocity trajectories (component ω_x) from the 500 training sequences, showing diverse dynamics used to train the Koopman-based model

The simulation pipeline is fully implemented in MATLAB without external symbolic or physics solvers. The use of randomised inputs and initial conditions ensures the training data covers a wide range of dynamic behaviours, including small- and large-angle rotations, acceleration and deceleration phases, and non-trivial torque interactions. The resulting dataset provides a representative basis for learning generalisable satellite attitude dynamics in a data-driven manner.

It is important to emphasise that the goal of this data generation process is not to analytically solve or identify the governing equations of satellite attitude dynamics. Instead, the focus lies in constructing a representative dataset from a nonlinear system whose true model is assumed to be unknown. The satellite's rotational dynamics, inherently nonlinear due to the coupling between quaternion kinematics and angular velocity evolution, cannot be directly captured using conventional linear approaches. Therefore, the Koopman operator framework is employed as a data-driven method to lift the nonlinear system into a higher-dimensional space where its evolution can be approximated linearly. In this context, data generation is essential for learning a predictive Koopman-based model, rather than for model validation or physics-informed parameter estimation.

3. Angular velocity and trajectory prediction

In this section, we introduce a method for learning a control-oriented satellite attitude dynamics model based on the Koopman operator. The Koopman framework provides a powerful way to model nonlinear systems by representing them in an enhanced observable space, where their evolution can be approximated by linear dynamics. Unlike traditional methods that require knowledge of physical laws and system parameters, the Koopman approach uses observed trajectory data to discover potential representations of nonlinear dynamic linear evolution. This feature makes it particularly suitable for integration with linear control frameworks such as MPC while maintaining high prediction fidelity.

Formally, let the original system state be denoted by $x_t \in \mathbb{R}^n$, and control input $\tau_t \in \mathbb{R}^m$. The true system evolution is governed by a nonlinear dynamical system $x_{t+1} = f(x_t, \tau_t)$. In the Koopman framework, the state is lifted via a function $g(x_{t+1}) \in \mathbb{R}^p$, where $p > n$, such that the dynamics in the lifted space can be approximated by a linear system:

$$g\left(x_{t+1}\right) = Ag\left(x_t\right) + B\tau t$$

where $A \in \mathbb{R}^{p \times p}$ and $B \in \mathbb{R}^{p \times m}$ are the Koopman operator matrices to be identified from data.

To construct the lifting function g(x_t), we designed a set of observable features that map the original satellite state into a higher-dimensional latent space in which the nonlinear rotational dynamics evolve approximately linearly. The original state x(t) $\in \mathbb{R}^7$ comprises a unit quaternion q(t) = [q_0, q_1, q_2, q_3]^T, representing the satellite's orientation, and an angular velocity vector ω(t) = [ω_x, ω_y, ω_z]^T. The lifting function g(x_t) $\in \mathbb{R}^{41}$ is constructed to preserve physical interpretability while introducing sufficient nonlinear structure for accurate approximation of the dynamics using a linear Koopman operator.

The lifted state includes several categories of features. First, we retain the original state components q and ω, providing the model with a baseline representation of the orientation and rotational velocity. Second, we embed rotational geometry by converting the quaternion into a 3×3 direction cosine matrix R(q), derived from the quaternion q(t) = [q_0, q_1, q_2, q_3]^T, which is used to embed orientation into the lifted state. Each entry of R(q) represents the cosine of the angle between body-fixed and inertial axes. To incorporate it into the Koopman framework, R(q) is reshaped into a 9-dimensional vector, providing a compact and globally valid representation of orientation that captures nonlinear rotational coupling for better approximation in the lifted linear space. The direction cosine matrix is derived from the quaternion as follows:

$$R\left(q\right) = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}$$

This allows the model to access the global rotational alignment between the body and inertial frames without suffering from singularities common to Euler angles.

To further enhance the model's capacity to capture nonlinearities, we introduce higher-order features including the squared components of both quaternions and angular velocities: q_0^2, q_1^2, q_2^2, q_3^2 and ω_x^2, ω_y^2, ω_z^2. These quadratic terms allow the linear model to internally represent second-order nonlinear interactions that commonly arise in rotational dynamics, such as energy-related quantities. Additionally, to capture the nonlinear interactions between orientation and rotational velocity, we explicitly construct 12 cross-product terms between quaternion components q(t) = [q_0, q_1, q_2, q_3]^T and angular velocity components ω(t) = [ω_x, ω_y, ω_z]^T, namely: $q_0\omega_x$, $q_0\omega_y$, $q_0\omega_z$, $q_1\omega_x$, $q_1\omega_y$, $q_1\omega_z$, $q_2\omega_x$, $q_2\omega_y$, $q_2\omega_z$, $q_3\omega_x$, $q_3\omega_y$, $q_3\omega_z$. These terms are included in the lifted state to enhance the expressive capacity of the model and help the linear predictor capture rotational coupling effects more accurately.

Lastly, to capture non-polynomial and periodic nonlinearities inherent in rotational dynamics, we include trigonometric terms sin(ω_x), sin(ω_y), sin(ω_y), cos(ω_x), cos(ω_y), cos(ω_z) in the lifted state. These features enable the model to approximate periodic effects more effectively and reflect the cyclical nature of angular motion, which are otherwise poorly represented by polynomial observables.

Combined, the lifting function yields a 41-dimensional observable vector:

$$g\left(x\right) = \left[q; \omega; \vec{v}ec\left(R\left(q\right)\right); q^2; \omega^2; q\omega; \sin\left(\omega\right); \cos\left(\omega\right)\right] \in \mathbb{R}^{41}$$

This effectively embeds the state into a space rich enough to capture the essential nonlinear dynamics while preserving linear evolution under the Koopman operator. This construction enables the model to learn globally consistent, interpretable dynamics from purely data-driven observations, facilitating long-term prediction and control design without explicit knowledge of the underlying physics.

With this lifting scheme, we construct a dataset of transition tuples (x_t, τ_t, x_{t+1}), where each state x_t is mapped into a higher-dimensional lifted state g(x_t) $\in \mathbb{R}^p$. We define: G = [g(x_1), g(x_2), ..., g(x_N)] $\in \mathbb{R}^{p \times N}$: the matrix of lifted current states, U = [τ_1, τ_2, ...,τ_N] $\in \mathbb{R}^{m \times N}$: the matrix of corresponding control inputs, G' = [g(x_1), g(x_2), ..., g(x_{N+1})] $\in \mathbb{R}^{p \times N}$: the matrix of lifted next states.

These matrices are constructed by collecting all simulation data across the trajectories, resulting in a total of N = 50000 transitions.

The Koopman matrices A $\in \mathbb{R}^{p \times p}$ and B $\in \mathbb{R}^{p \times m}$ are then learned by solving the following least-squares regression problem:

$$\min_{A,B} ||G' - AG - BU||_F^2$$

Notably, this Koopman model is trained entirely on data generated by the physics-based simulator, without relying on the explicit knowledge of the inertia matrix J or environmental disturbances. The learned Koopman model can then be used for long-term prediction of satellite attitude evolution, which we evaluate in the next section.

To evaluate the performance of the Koopman-based model in predicting satellite rotational dynamics, we conducted tests on three unseen trajectories, each consisting of 100 time steps. The results are presented in both graphical and quantitative forms.

For each trajectory, we compared the predicted angular velocity $\omega(t) = [\omega_x, \omega_y, \omega_z]^T$ and the corresponding orientation, represented in terms of Euler angles, against ground truth data.

In all three trajectories, the predicted angular velocities closely track the ground truth, as illustrated in the upper panels of Table 1. Despite some small fluctuations, the trends and magnitudes of ω_x, ω_y and ω_z are well preserved, indicating that the Koopman model successfully captures the underlying nonlinear rotational dynamics. The corresponding mean squared errors (MSEs) are extremely small, on the order of 10^{-21}, further confirming the precision of the model in short-horizon prediction tasks.

The lower panels of Figure 4 display the predicted versus true Euler angles for each trajectory. The agreement is visually compelling, with the predicted trajectories almost indistinguishable from the ground truth in many cases. For example, in Trajectory 1, the predicted yaw, pitch, and roll curves remain tightly aligned throughout the time horizon, with MSE values of 0.0048, 0.0012, and 0.0011 radians squared, respectively. Similar performance is observed in Trajectories 2 and 3, with slight variations depending on the complexity and rotational energy of the test motion. In Trajectory 2, which involves more significant pitch and roll changes, the MSEs are correspondingly higher (e.g., 0.0158 for Pitch), yet the predicted orientation still closely mirrors the true trajectory.

Overall, the results demonstrate that the Koopman-based modelling framework, trained on 50,000 randomly generated samples via MATLAB simulations, is capable of generalising to previously unseen rotational behaviours with high fidelity. The successful transformation from quaternion states to Euler angles not only facilitates interpretability but also highlights the model's ability to preserve both rate and orientation accuracy across different test cases. This consistency across multiple degrees of freedom and motion profiles indicates the robustness and effectiveness of the Koopman operator framework in learning satellite attitude dynamics.

Table 1. Mean Squared Error (MSE) of Koopman-based model predictions for angular velocity components (ω_x, ω_y, ω_z) and Euler angles (yaw, pitch, roll) across three representative test trajectories

	Trajectory 1			Trajectory 2			Trajectory 3		
MSE for each angular velocity	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
	0.1e-22	0.3e-22	2.9e-22	0.5e-22	1.5e-22	3.3e-22	0.7e-22	1.9-22	3.0e-22
MSE for Euler Angles	Yaw	Pitch	Roll	Yaw	Pitch	Roll	Yaw	Pitch	Roll
	0.0048	0.0012	0.0011	0.0018	0.0158	0.0024	0.0025	0.0028	0.0085

The results demonstrate extremely low prediction error for angular velocity (on the order of 10^{-21}), and consistently small errors for Euler angles, indicating accurate reconstruction of satellite attitude dynamics.
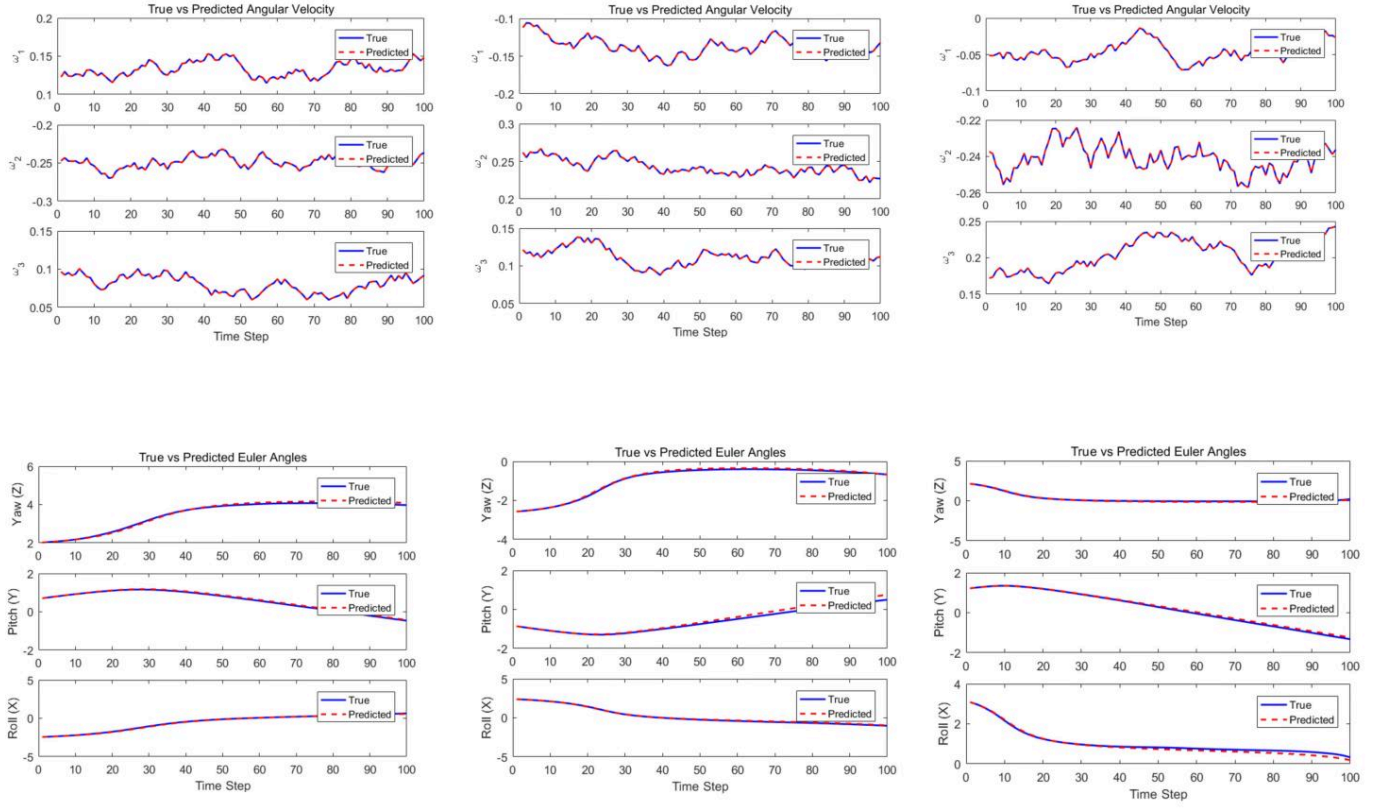
Figure 4. Prediction results of angular velocity and Euler angles for three representative test trajectories using the Koopman-based model

In Figure 4, the plots are arranged from left to right corresponding to Trajectory 1, Trajectory 2, and Trajectory 3. For each trajectory, the top three plots show the time evolution of angular velocity components (ω_x, ω_y, ω_z), while the bottom three plots depict the predicted Euler angles (yaw, pitch, roll). Solid blue lines represent the true values, and dashed red lines represent the predicted values.

4. MPC framework for angular velocity

This section will integrate the learned models into the MPC framework. The control objective is to asymptotically drive the angular velocity of the satellite to zero from a given initial state, thereby achieving attitude stability and angle maintenance. The controller is constructed using the previously identified Koopman linear model, characterised by the system matrices A and B, and adopts a standard quadratic cost function over a finite prediction horizon T:

$$\min_{u_0:T-1} \sum k=0^{T-1}\left(||gk\text{-}gref||_Q^2+||\boldsymbol{\tau}_k||_R^2\right)$$

$$\text{s.t. } g_{k+1} = Ag\left(x_t\right) + B\tau_t$$

where $g_k = g(x_k)$ denotes the lifted state at time step k, and gref is the reference lifted state corresponding to zero angular velocity in the original system space. The weighting matrices Q and R penalise state deviation and control effort, respectively. The optimisation problem is solved using MATLAB's quadprog solver in a receding horizon manner, applying only the first optimal control input at each step and updating the state accordingly for the next iteration.

Under the proposed Koopman-MPC framework, the satellite's angular velocity can be effectively stabilised. As shown in Figure 5, all three angular velocity components converge toward zero within a short time horizon, reflecting the controller's ability to suppress rotational motion and maintain attitude stability. This result highlights the feasibility of embedding Koopman-based models into predictive control frameworks to achieve real-time regulation of satellite dynamics, without resorting to the full nonlinear model. It also validates the linear predictive structure obtained via Koopman lifting as a viable tool for practical control applications.
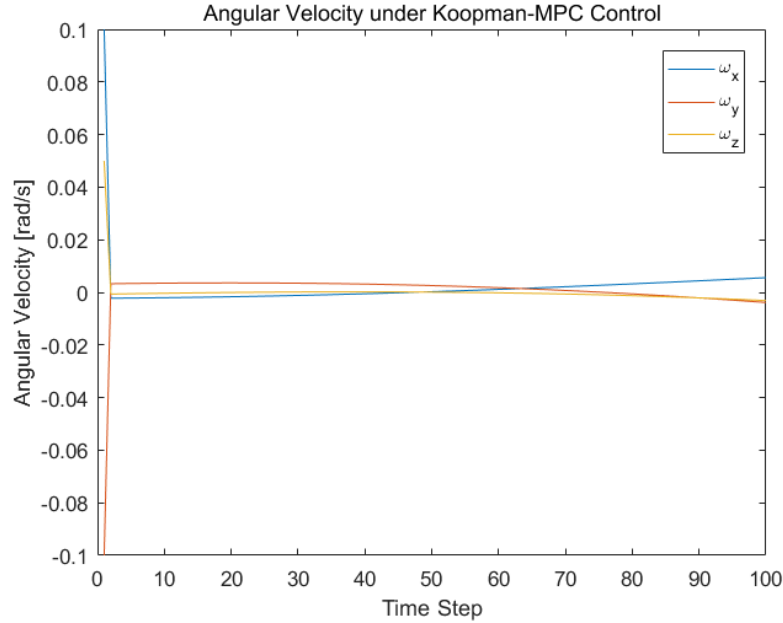

Figure 5. Angular velocity response of the Koopman-MPC-controlled satellite system over 100 time steps

The three curves shown in Figure 5 represent the angular velocities around the x-, y-, and z-axes (ω_x, ω_y, and ω_z). The results show that the controller effectively drives the angular velocity toward zero, indicating successful stabilisation.

5. Comparison of model performance

Prediction performance of different modelling approaches on spacecraft attitude dynamics was compared across three models: the Koopman operator-based model, a feedforward neural network (NN), and a linear time-invariant (LTI) system. Each model was tested on the same trajectory data to predict Euler angles (yaw, pitch, roll), and their accuracy was quantitatively assessed using the mean squared error (MSE).

As shown in Figure 6, the horizontal axis represents the three Euler angle components, while the vertical axis depicts the MSE on a logarithmic scale. It is evident that the Koopman-based model achieves the best overall performance, maintaining errors in the range of 10^{-3} to 10^{-2} across all directions. The prediction accuracy for the pitch angle is particularly high, suggesting relatively stable system behaviour along that axis.

The neural network model also performs well on the pitch direction (MSE $\approx 7.8 \times 10^{-3}$), but its performance degrades for yaw and roll, especially in the roll direction, where the MSE rises to approximately 0.47. This discrepancy may result from insufficient generalisation to sharp attitude changes or imbalanced data distribution across axes during training. Overall, the NN model captures some nonlinear dynamics but exhibits greater variance in performance.

In contrast, the LTI model produced significantly larger prediction errors at all Euler angles, with an MSE value of approximately 10^0. Especially in roll prediction, it showed the lowest accuracy (MSE ≈ 2.04), indicating that the linear model structure failed to consider the inherent nonlinearity of the system. This result is consistent with the visual comparison between the predicted trajectory and the actual trajectory.
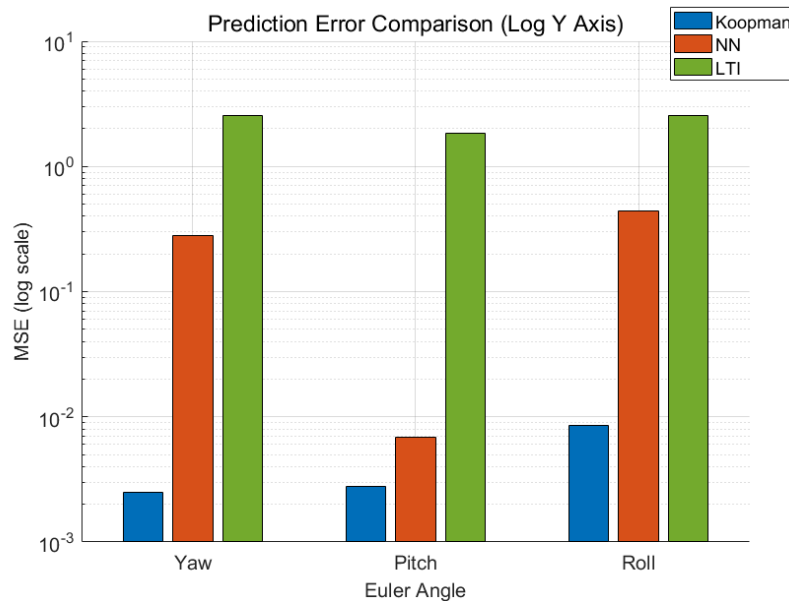
Figure 6. Comparison of Euler angle prediction errors (MSE) of Koopman, neural network (NN), and linear time-invariant (LTI) model, with errors shown on a logarithmic scale

6. Conclusion

This paper proposes a data-driven modelling method for satellite attitude dynamics using the Koopman operator. By using specially designed observable functions, the original nonlinear system state composed of quaternions and angular velocities is elevated to a higher-dimensional space. This framework is suitable to some extent for scenarios where it is difficult to accurately identify the system, such as early task stages or reconfigurable satellite structures. The Koopman model learned has high prediction accuracy for angular velocity and attitude. In addition, this method provides physical interpretability by using a quaternion-based lift. These results indicate the potential of the Koopman operator as a satellite dynamics modelling tool in situations where classical models are difficult to construct or maintain.

References

[1]   Wie, B. (2008). Space vehicle dynamics and control (2nd ed.). AIAA Education Series.
[2]   Wertz, J. R. (1978). Spacecraft attitude determination and control. Springer.
[3]   Kuipers, J. B. (1999). Quaternions and rotation sequences. Princeton University Press.
[4]   Markley, F. L., & Crassidis, J. L. (2014). Fundamentals of spacecraft attitude determination and control. Springer.
[5]   Hughes, P. C. (1986). Spacecraft attitude dynamics. Dover Publications.
[6]   Schaub, H., & Junkins, J. L. (2009). Analytical mechanics of space systems (2nd ed.). AIAA.
[7]   McInnes, C. R. (1999). Solar sailing: Technology, dynamics and mission applications. Springer.
[8]   Bate, B. L., Mueller, D. D., & White, J. E. (1971). Fundamentals of astrodynamics. Dover Publications.
[9]   Psiaki, M. (2001). Attitude-determination filtering via extended Kalman filtering. *Journal of Guidance, Control, and Dynamics*, 24(6), 1187–1195.
[10] Diaz, S. K., & Linares, R. (2021). Data-driven learning of satellite attitude dynamics using recurrent neural networks. *Aerospace Science and Technology*, 108, 106377.
[11] Huang, H., Li, S., Wang, X., & Zhang, Y. (2022). Gaussian process regression for attitude dynamics modeling in CubeSats. *Acta Astronautica*, 194, 209–218.
[12] Korda, M., & Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93, 149–160.
[13] Bruder, D., Kaiser, E., Goldschmidt, S., Brunton, S. L., & Duraisamy, K. (2020). Data-driven identification of Koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2(1), 015003.
[14] Mauroy, A., Susuki, Y., & Mezic, I. (2020). The Koopman operator in systems and control: Concepts, methodologies, and applications. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 67–100.
[15] Pan, S., & Duraisamy, K. (2024). A data-driven discovery of Koopman eigenfunctions for nonlinear systems. Nonlinear Dynamics. Advance online publication.
[16] Shi, L., & Karydis, K. (2021). Analytical construction of dictionaries for Koopman operator based representations of robotic systems. *IEEE Robotics and Automation Letters*, 7(2), 906–913.