

Article

Not peer-reviewed version

Autonomous Real-Time Mass Center Location and Inertia Identification for Grappling Space Robotics

[Timothy Sands](#)*

Posted Date: 20 March 2025

doi: 10.20944/preprints202412.1297.v2

Keywords: Spacecraft; space robotics satellite servicing; autonomous grapple; adaption; learning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Autonomous Real-Time Mass Center Location and Inertia Identification for Grappling Space Robotics

Timothy Sands ^{1,2}

¹ Department of Mechanical Engineering (SCPD), Stanford University, Stanford, CA 94305, USA; dr.timsands@alumni.stanford.edu

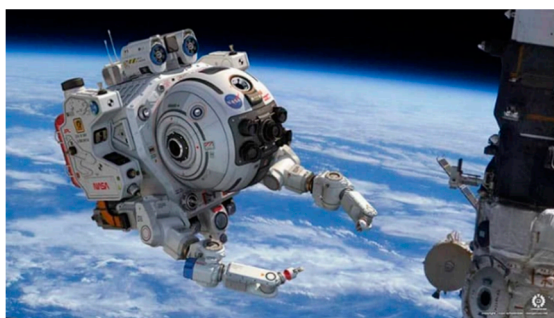
² Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

Abstract: Grappling actions by space robots for the purposes of stabilizing, refueling, repair, and equipment replacement necessitate autonomous abilities of a single grappling space robot to rapidly contend with large variations in total system inertia rapidly shifting system center of mass, as targets can be massive with possibly unknown or poorly known mass inertia properties. Grappling actions yield opportunities for novel online calculation of time varying location of the combined system's center of mass. Two norm optimal nonlinear, projection regression-based learning is implemented and juxtaposed to a comparative benchmark both qualitatively and quantitatively supported by a comparison of enhancements of Luenberger observers. Analysis precedes modeling and simulation to verify the design, and then spaceflight experiments are proposed for the sequel to validate the simulation results. Time varying mass locations are discerned, and the time varying location of the mass center is revealed to be 36–95 percent different than initially assumed, and 58–317 percent corrections to inertia identification is demonstrated. Combined three-dimensional maneuvers obscure identification compared to single-axis maneuvering.

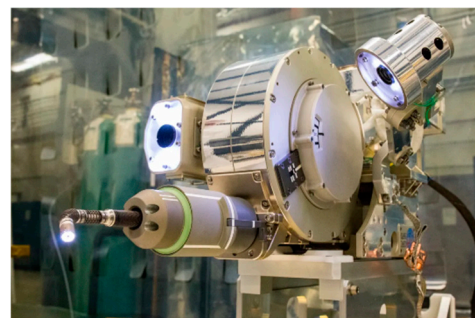
Keywords: Spacecraft; space robotics satellite servicing; autonomous grapple; adaption; learning

1. Introduction

Autonomous robotic grappling is the core requirement for on-orbit robotic servicing and assembly missions. [1] National Aeronautics and Space Administration (NASA) recently prototyped grappling arms for its Space Exploration Vehicle (SEV). [2]



(a)



(b)

Figure 1. (a) NASA PickNik Robotics in-orbit robot addresses a common gap in deploying robotic arms into environments where full autonomy is still too difficult permitting a human operator to supervise and quickly intervene when the space robot gets stuck. [3] (b) Robotic Refueling Mission 3 (RRM3) technologies were developed store and transfer super-cold cryogenic fluids. [4] Image credit: NASA. [5] See 'aside' note in Appendix A.

1.1. Review of the Literature

Long-term sustainability of space includes other robotics tasks like active debris removal, [6] necessitating arms for grappling and various grasping and anchoring mechanisms. [2] According to the Technical Committee for Space Robotics of the IEEE Robotics & Automation Society, priority areas for the technical committee include electromechanical design and control, and command & control (including teleoperated modes). [7] The study in reference [8] emphasizes center of mass changes due to deployment of payloads or instruments, solar panels, gravity booms, magnetometers, antennas, cameras, solar sails, etc. and used recursive least squares to estimate the mass center's position. The study in [9] misaligns a thruster from the anticipated center of mass and uses the governing dynamic equations of motion to estimation the mass center location using multiple Kalman filters.

Determination of moments of inertia (but not inertia products) about two axes was proposed in 1948 for airplane components prior to flight, where the component would be suspended in a lab and subjected to standard pendulum experiments. [10] Similarly, the moments for the entire airplane were similarly treated in [11]. Progress over the subsequent forty–six years added capabilities to use both angular acceleration methods and oscillation methods where the assumption is invoked that the suspension axis is assumed to be the axis of rotation. [12] Inertia cross–products were still not available. Twenty–nine years later light detection and ranging was utilized to estimate inertia parameters within twenty–percent, while location of the mass center was not performed. [13,14] Just this year, principal moments (not inertia cross products) of space tethers were estimated using unscented Kalman filters. [15]

Mass center estimation was recently proposed using a force platform with inertial sensors for balance evaluation in quiet standing and compared the results to traditional methods using optical motion capture system for truth values. [16] Three general classifications of methods were listed as follows, while this manuscript presents a novel alternative:

1. Integrate the horizontal acceleration of the center of mass [17],
2. Filter the center of pressure to estimate the center of mass [18],
3. Estimate the center of mass directly from the equations of motion of an inverted pendulum model

Reference [19] categorized the methods slightly differently as either sensors' networks or a single sensor while proposing using magneto-inertial measurement units. Calculation of the mass center was proposed [20–22] by electrostatic suspension of a proof mass in the satellite in an electrode cage which is rigidly attached to the spacecraft. Accelerations of the proof mass were determined, and then then relative accelerations between the spacecraft and proof mass were used to estimate the mass center. The procedure was used periodically to provide mass center location estimates. Follow–on experiments were presented in [23]. Arguably, this proof mass–electrode cage may be considered a state of the art for spacecraft so equipped. The calculations involve estimation of angular velocities and quaternion fitting using regression. (ASIDE: the methods proposed in this paper instead involve estimation of all inertia components (moments and products) and uses those estimates to deterministically calculate mass center location).

CFD methods were used in [24] to investigate the aerodynamic performance of the free-fall wing, and the results indicate wing fluttering and tumbling motions' tumbling frequencies are driven by changing locations of the moments of inertia around the tumbling axis. Furthermore, the changing position of the center of mass complicates the free-fall motion. The study in reference [25] used least squares, maximum likelihood, and Kalman filter estimators to estimate mass, center of mass, and moments of inertia of a multi–lift slung aircraft load. The study in reference [26] used image processing to locate centers of mass using Python–based programming.

Regression–based learning may function as a projection operator to enable data-driven learning of the operators in the Mori–Zwanzig formalism, where [27] proposed a principled method to extract the Markov and memory operators for any regression models. The research found that more

expressive nonlinear regression models fill in the gap between the highly idealized and computationally efficient Mori's projection operator and the most optimal yet computationally infeasible Zwanzig's projection operator.

For certain applications, e.g. on-orbit debris inspection of orbital natural objects, defunct satellites, and debris, etc. estimating the mass center requires mapping from a moving observer. Reference [28] applies an observer to estimate internal angular orientation, angular velocity, and angular acceleration. Next, pose-graph mapping is with visual odometry produces trajectory relative to a target-fixed frame.

Distinguished from these disparate approaches to autonomously estimate the mass center (while maintaining usage of observers), this present study seeks to determine time varying inertia identification and proposed novel online calculation of time varying location of the combined system's center of mass using two norm optimal nonlinear, projection regression-based learning augmented with enhanced, nonlinear Luenberger observers in direct comparison to benchmark methods.

Box 1. Problem Statements, research objectives, and key contributions.

Problem statements: At all times, find a system's mass distribution and the location of the center of mass. How may the location be calculated given only timed, angular position data? To what degree of accuracy can metrics be discerned? Requiring how much computational burden? Requiring how much effort?

Research objectives: 1) Using only attitude angle data, discern mass inertia imbalance and 2) express that imbalance as an offset center of gravity, and then discern the location of the center of mass.

Key contributions: Center of gravity location estimation convergence is improved over the state-of-the-art comparative benchmark [21–23] from millimeter convergence over hundreds of days, to hundreds of millimeters in minutes achieved in this study.

1.2. State of the Art Benchmarks

The following list highlights the current state of the art inertia moment and products determination and locating the mass center:

1. In 2022, reference [1] provided an overview survey of the algorithms and test and validation strategies for grappling robotic free-flying spacecraft, de-emphasizing spacecraft docking and berthing. Core technologies identified includes orientation control (including distributed control systems) and the ability to adapt to unexpected loads. Those core technologies motivate the developments presented in this present study.
2. Nonlinear dynamics are highlighted as key, and therefore nonlinear approaches are adopted.
3. Despite insistence on nonlinear dynamics, linear time invariant. PID control laws are emphasized to be "straightforward to design and relatively easy to tune and provide performance sufficient to meet most operating requirements".
4. Since the dynamics are Hamiltonian, nonlinear adaptive control is also highlighted and is also adopted in this present study.
5. Trajectory planning is highlighted as well, and autonomous trajectory generation is adopted to permit nonlinear control proposals.
6. Air-bearing tables are highlighted to aid validation using laboratory experiments, and one such free-floating, highly flexible robotic gripper arm is described.

1.3. Novelties Presented

The following short list of novelties are combined sequentially to produce novel, autonomous, real-time space robotics:

1. Time varying estimates of mass locations are found with classical adaptive methods compared to optimal learning leveraging nonlinear, projection regression-based methods. Estimates are supported in-part by the novel use of nonlinear enhanced Luenberger observers.
2. Time varying estimates of mass locations are used to find time varying estimates of the location of the system center of mass. Estimates are supported in-part by the novel use of nonlinear enhanced Luenberger observers applying the fundamental relationships between products of inertia and mass center location.
3. Requirement to diagonalize the matrix of mass moments of inertia is eliminated.
4. Requirement to linearize the governing differential equations is eliminated.
5. Requirement to simplify governing equations (e.g. small angle assumption, etc.) is eliminated.
6. Analysis precedes modeling and simulation to verify the design, and then spaceflight experiments are proposed for the sequel to validate the simulation results.

1.4. Conveniences of Presentation

Periodic tables of proximal variables and nomenclature definitions is provided to aid reading without necessitating flipping back and forth between pages seeking definitions of terms. Observers are presented topologically rather than using state space equations to aid remembrance and understanding. Lastly, simulations used to produce the data presented in the manuscript are used to produce the graphic topologies, and furthermore MATLAB®/SIMULINK® subsystems are included in the article appendix aiding repeatability by the readership.

2. Materials and Methods

Physics-based dynamics are foremost detailed in section 2.1. Next, feedforward control using physics-based dynamic embodiment is presented in section 2.2. followed by feedback instantiations: both nonlinear adaption and projection regression-based learning of fully populated inertia matrix components. Nonlinear, enhanced Luenberger observers are introduced to support both feedback instantiations, and the observers are juxtaposed quantitatively to a comparative benchmark. Finally, location of the mass center is estimated using the estimated inertia components with the Final Value Theorem, which is briefly reviewed in the analysis of section 2.

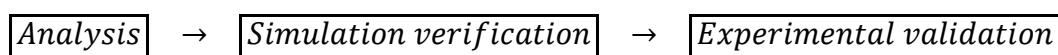


Figure 2. Process stages adopted in this study: analyze and hypothesize followed by verification of analysis with modeling and simulation validated experimentally.

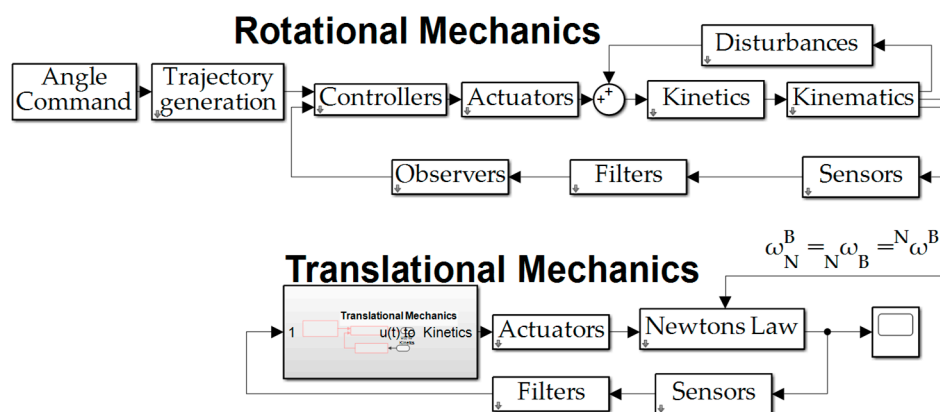


Figure 3. Space robot system topology represented by the MATLAB®/SIMULINK® simulation used to produce the quantitative results presented in this manuscript. Internal content of simulation subsystems is depicted in

Appendix B. Notice the translational subsystems of the six-degree-of-freedom motion simulation are deactivated reducing superfluous computational burden.

Foreshadowing the summarized methods, Table 1 includes a pithy implementation procedure and lists the relevant sections the details may be found.

Table 1. Implementation procedure.

1. Use the physics-based governing dynamics to embody the control (section 2.1)
 - a. Do not diagonalize the inertia matrix
 - b. Do not linearize the governing equations of motion
 - c. Do not use small angle assumption to simplify the system equations
 - d. Establish classical feedback control as comparative benchmark
2. Use nonlinear, enhanced Luenberger observers (section 2.4–2.6) to estimate all motion states and control.
3. Use projection-based non-linear regression learning to learn the fully populated mass moments of inertia (masses and mass locations) including principal moments and off-diagonal products of inertia. (section 2.3)
4. Use off-diagonal products of inertia to estimate location of the center of mass (section 2.8).

2.1. Physics-Based Dynamics

The well-known Euler's moment equations (equation (1)) govern rotations of objects of mass. The system of equations is nonlinear and coupled (especially for a fully populated inertia matrix $[J]$ with both moments and products of inertia, particularly noting the presence of the cross-product operation which puts terms from the second and third equation into the first equation in the system. The equations are ubiquitously linearized and simplified by changing the basis of measurement to diagonalize the inertia matrix.

$$\tau = [J]\{\dot{\omega}\} + \{\omega\} \times [J]\{\omega\} \quad (1)$$

One motivation for linearization is to aid expression of the system of equations in a standard regression form (a matrix of 'knowns' multiplied by a vector of presumed 'unknowns'). Particularly stemming from a common practice to put the motion states into the vector of unknowns, often referred to as the "state vector", attempts to express equation (1) into regression form in equation (2).

$$\tau = \underbrace{\begin{bmatrix} J_{xx} & J_{xy} & J_{xz} & ? & ? & ? \\ J_{yx} & J_{yy} & J_{yz} & ? & ? & ? \\ J_{zx} & J_{zy} & J_{zz} & ? & ? & ? \end{bmatrix}}_{[\Phi?]} \underbrace{\begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \\ ? \\ ? \\ ? \end{Bmatrix}}_{\{\beta?\}} \quad (2)$$

Table 2. Table of proximal variables and nomenclature ¹.

Variable/ acronym	Definition	Variable/ acronym	Definition
$[J] = J$	Inertia matrix or tensor	J_{xx}	Moment of inertia about \hat{x}
$\{\dot{\omega}\} = \dot{\omega}$	Angular acceleration	$J_{xy} = J_{yx}$	$\hat{x}\hat{y}$ inertia product
$\{\omega\} = \omega$	Angular velocity	$J_{xz} = J_{zx}$	$\hat{x}\hat{z}$ inertia product
$\dot{\omega}_x$	Angular acceleration in the \hat{x} -direction	J_{yy}	Moment of inertia about \hat{y}
$\dot{\omega}_y$	Angular acceleration in the \hat{y} -direction	$J_{yz} = J_{zy}$	$\hat{y}\hat{z}$ inertia product
$\dot{\omega}_z$	Angular acceleration in the \hat{z} -direction	J_{zz}	Moment of inertia about \hat{z}
$[\Phi?] = \Phi?$	Regression matrix of knowns	$\{\beta?\} = \beta$	Unknown, predicted variables

¹ Such tables are offered throughout the manuscript to aid readability.

The inertia matrix $[J]$ is most often diagonalized using the eigenvalue coordinate transformation, or spectral decomposition. The transformation begins with relationships expressed in coordinates of the body fixed axes (where the inertia matrix is fully populated with moments and products). Transformation to a new basis vector set (i.e., the eivenvectors) lead to a diagonalized inertia matrix with only moments and no products, and the diagonalized matrix simplifies subsequent mathematics that are performed with the inertia matrix. [29]

A common engineering practice is to simply assume the eigenvector basis is equivalent to the body frame basis vectors, where the engineer is comfortable accepting the accompanying error of such an assumption.

The inertia matrix will not be diagonalized in this present study, instead favoring increased accuracy. The inability of expressing equation (2) in standard regression form must be re-addressed.

Noting the difficulties of equation (2), omitting the common practice establishing the vector of unknowns, the mass moments and products may instead be used (equation (3)). A more elaborate derivation from equation (1) to equation (3) is offered in appendix A.

$$\tau = \underbrace{\begin{bmatrix} \dot{\omega}_x & \dot{\omega}_y - \omega_x\omega_z & \dot{\omega}_z - \omega_x\omega_y & -\omega_y\omega_z & 0 & \omega_y\omega_z \\ \omega_x\omega_z & \dot{\omega}_x + \omega_y\omega_z & 0 & \dot{\omega}_y & \dot{\omega}_z - \omega_x\omega_y & -\omega_x\omega_z \\ -\omega_x\omega_y & 0 & \dot{\omega}_x - \omega_y\omega_z & \omega_x\omega_y & \dot{\omega}_y + \omega_x\omega_z & \dot{\omega}_z \end{bmatrix}}_{[\Phi]} \underbrace{\begin{Bmatrix} J_{xx} \\ J_{xy} \\ J_{xz} \\ J_{yy} \\ J_{yz} \\ J_{zz} \end{Bmatrix}}_{\hat{\beta}} \quad (3)$$

Equation (3) establishes the physics-based embodiment of the control with specific instantiations in both feedforward and feedback.

2.2. Feedforward Control Using Physics-Based Dynamic Embodiment

The feedforward instantiation of equation (3) is presented in equation (4) where the matrix of “knowns” necessitates having known angular velocities and angular accelerations. This necessity motivates the use of sinusoidal commanded trajectories, which will be shown to also aid estimation identifiability in section 2.9.

$$\tau_{ff} = \underbrace{\begin{bmatrix} \dot{\omega}_x & \dot{\omega}_y - \omega_x\omega_z & \dot{\omega}_z - \omega_x\omega_y & -\omega_y\omega_z & 0 & \omega_y\omega_z \\ \omega_x\omega_z & \dot{\omega}_x + \omega_y\omega_z & 0 & \dot{\omega}_y & \dot{\omega}_z - \omega_x\omega_y & -\omega_x\omega_z \\ -\omega_x\omega_y & 0 & \dot{\omega}_x - \omega_y\omega_z & \omega_x\omega_y & \dot{\omega}_y + \omega_x\omega_z & \dot{\omega}_z \end{bmatrix}}_{[\Phi]_d} \underbrace{\begin{Bmatrix} \hat{J}_{xx} \\ \hat{J}_{xy} \\ \hat{J}_{xz} \\ \hat{J}_{yy} \\ \hat{J}_{yz} \\ \hat{J}_{zz} \end{Bmatrix}}_{\hat{\beta}} \quad (4)$$

This embodiment may also be used in feedback, where actual trajectories are substituted for desired trajectories, and that instantiation is described later in section 2.8 Combined control.

Table 3. Table of proximal variables and nomenclature ¹.

Variable/ acronym	Definition	Variable/ acronym	Definition
$[\omega_x]_d$	Desire angular velocity about \hat{x}	$[\dot{\omega}_x]_d$	Desire angular acceleration about \hat{x}
$[\omega_y]_d$	Desire angular velocity about \hat{y}	$[\dot{\omega}_y]_d$	Desire angular acceleration about \hat{y}
$[\omega_z]_d$	Desire angular velocity about \hat{z}	$[\dot{\omega}_z]_d$	Desire angular acceleration about \hat{z}
$[\Phi] = \Phi$	Regression matrix of sensor data	$[\Phi]_d = \Phi_d$	Regression matrix of desired states
β	Unknown, predicted variables	$\hat{\beta}$	Estimated variables
τ	Total control signal	τ_{ff}	Feedforward control signal
\hat{J}_{xx}	Estimated Moment of inertia about \hat{x}	$\hat{J}_{xy} = \hat{J}_{yx}$	Estimated $\hat{x}\hat{y}$ inertia product

\hat{J}_{yy}	Estimated Moment of inertia about \hat{y}	$\hat{J}_{xz} = \hat{J}_{zx}$	Estimated $\hat{x}\hat{z}$ inertia product
\hat{J}_{zz}	Estimated Moment of inertia about \hat{z}	$\hat{J}_{yz} = \hat{J}_{zy}$	Estimated $\hat{y}\hat{z}$ inertia product

¹ Such tables are offered throughout the manuscript to aid readability.

2.3. Regression-Based Projection for Learning

Feedback may be used to project trajectory tracking errors (equation (5)) on the nonlinear, coupled system equations expressed in regression for in equation (4). The estimated torque is not likely provided by motion sensors (e.g. star trackers, sun sensors, limb lookers, etc.), so observers are investigated to provide the information (in addition to full state trajectories described later in section 2.8).

$$\tau_{fb} = [\Phi]_d([\Phi]_d^T[\Phi]_d)^{-1}[\Phi]_d\delta\tau \quad (5)$$

$$\delta\tau = \tau_{ff} - \hat{\tau} \quad (6)$$

2.4. Luenberger Observers

Luenberger observers are sometimes referred to as state observers or simply observers and is well described in reference [30] for observable systems recently ubiquitously defined in so-called “state space” form, or alternatively formulated in state-variable form. The discrete form of the estimator is often represented by the form presented in equation (7) for discrete time index k , states designated x , the “hat” $\hat{\cdot}$ indicating an estimate, output y , where A_d is discretized state matrix and B_d is discretized observer gain matrix respectively.

$$\hat{x}(k+1) = A_d\hat{x}(k) + B_d u(k) + L_d[y(k) - \hat{y}(k)] \quad (7)$$

Rather than adopting the ubiquitous form of equation (7), observers are presented in the following sections in topological form to highlight the enhancements to the observers.

2.5. Enhanced Luenberger Observers

Notice in Figure 4 the three channels on the right-hand side of the figure. The ubiquitous derivative channel is past forward (to the left of the graphic) of the next integrator compared to the integral and proportional channels. Since the signal to the left of an integrator (the integrator output) is the integral of the input signal, the input signal (to the right) is the derivative of the output signal, thus inducing derivative action on the input. This commonplace enhancement induces derivative action into the estimator without numerically differentiating noisy input signals.

Secondly, notice the top-center topology in Figure 4 contains a manual switch with the control as a potential input. This enhancement is often applied to reduce or eliminate phase lag in the estimator.

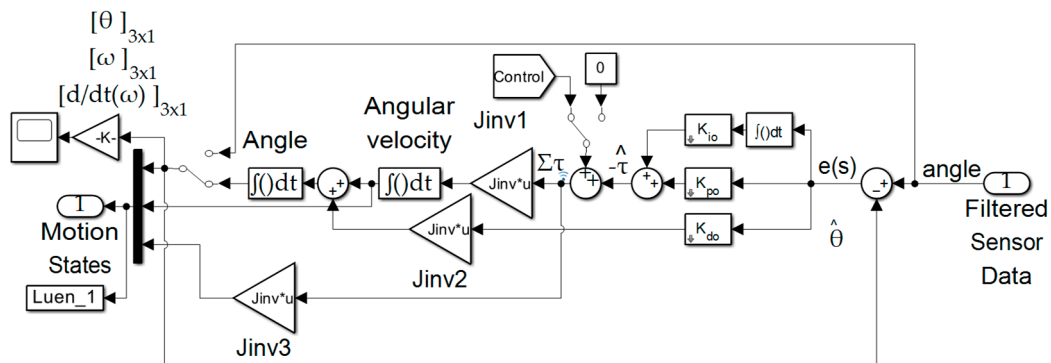


Figure 4. Luenberger and Enhanced Luenberger Observers built in MATLAB®/SIMULINK®.

Lastly notice the center channel or line passing down the middle of the estimator from right to left consists of a double-integrator of states lacking the cross-products of motion stemming from the transport theorem of mechanical motion relating the time derivative of a Euclidean vector as evaluated in a non-rotating coordinate system to its time derivative in a rotating reference frame. [31] Adoption of enhancements accounting for transport theorem are presented in the following including nonlinear enhancements accounting for the nonlinear coupling nature of the transport theorem terms (the cross-products in equation (1)).

2.6. Nonlinear Enhanced Luenberger Observers

Notice Figure 5 contains predominantly the same content as Figure 4, the enhanced Luenberger observer. The new enhancements are depicted in the center of Figure 5 where the angular velocity signal is extracted, scaled by the mass moments of inertia, then used to create the cross-products which are added to the baseline double-integrator signal.

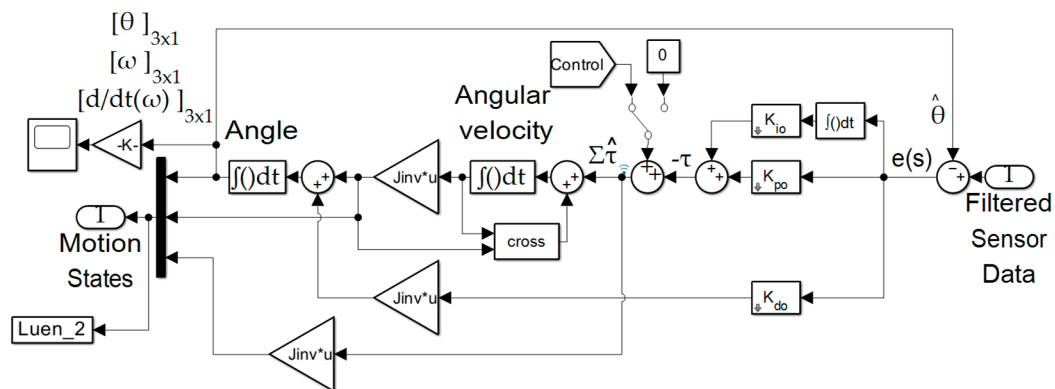


Figure 5. Nonlinear Enhanced Luenberger Observer built in MATLAB®/SIMULINK®.

2.6. Classical Feedback Adaption

Following initial parameterization in the inertial reference frame, classical feedback adaption was presented in [32] in the body reference frame, greatly simplifying the algorithms, enhancing its usefulness as a contemporary replacement of the formerly ubiquitous M.I.T. Rule. [33] The subsystem is displayed in Figure B.31 of Appendix B codifying equation (8).

2.7. Two Norm Optimal Nonlinear, Projection Regression-Based Learning

An alternative to the classical feedback adaption approach is to project trajectory tracking errors onto the governing system equations parameterized into nonlinear regression form (without truncation, approximations, linearization, diagonalization, or other implications). The regression form is illustrated in the feedforward (τ_{ff}) portion of equation (9)'s combined control, while the projection onto the system equations is displayed in the second term (τ_{fb}) using the two norm optimal pseudoinverse application of least squares. The subsystem is displayed in Figure B.32 of Appendix B.

2.8. Combined Control

The feedforward control portion is identically the nonlinear, coupled governing equations of motion parameterized as a matrix $[\Phi]$ multiplied by a vector $\hat{\beta}$ where the matrix is built using desired (commanded) trajectories annotated by the subscribe "d", while in feedback form, actual trajectories are used in accordance with equation (5). The need for actual trajectories justifies the investigation of contemporary enhancements to Luenberger observers in sections 2.4 and 2.5.

$$\tau = \underbrace{[\Phi]_d \{\beta\}}_{\tau_{ff}}, \text{ where } \beta = \int \Gamma[\Phi]s(\delta\omega + \lambda\delta\dot{\omega}) dt \quad (8)$$

$$\tau = \underbrace{[\Phi]_d \{\beta\}}_{\tau_{ff}} + \underbrace{[\Phi]_d ([\Phi]_d^T [\Phi]_d)^{-1} [\Phi]_d \delta\tau}_{\tau_{fb}} \quad (9)$$

$$\delta\tau = \text{Torque estimate from Figure 4– Figure 5} \quad (10)$$

2.8. Estimating Location of the Center of Mass

Since the matrix of mass moments of inertia has not been diagonalized, the cross-products of inertia (off-diagonal matrix terms) may be used to locate the center of mass using the parallel axis theorem. Figure 6 provides an illustrative example. Consider the desire to have a mass center located along the depicted central T_c axis. If the actual location is suspected to be at some other point, e.g. about the depicted y axis, the parallel axis theorem may be used to parameterize the problem.

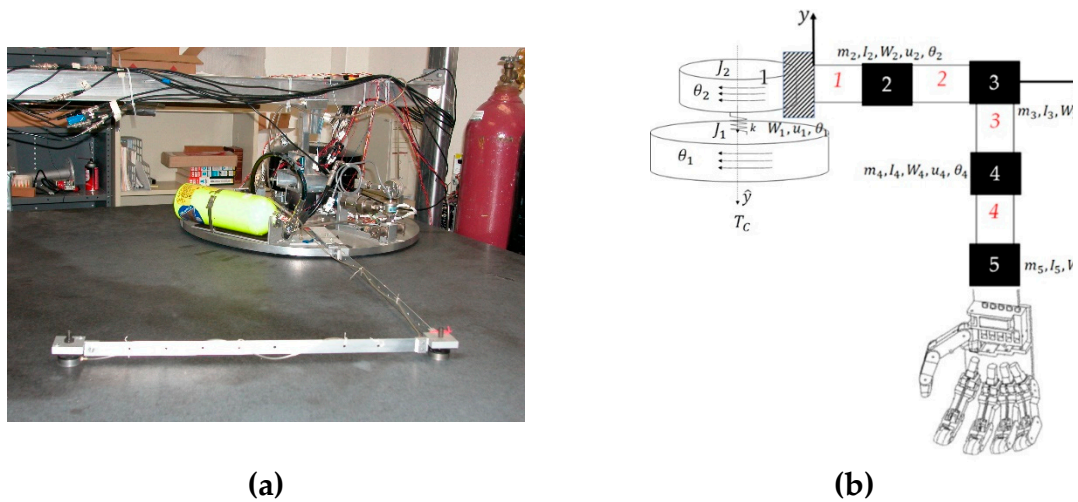


Figure 6. Benchmark free-floating laboratory space gripper robot used in prequel studies. [35] (a) photo of laboratory hardware, (b) schematic used for prequel engineering analyses.

Table 4. Table of proximal variables and nomenclature ¹.

Variable / acronym	Definition	Variable/ acronym	Definition
$m_i \forall i = 2, \dots, 5$	Mass index	$[\Phi]_d$	Defined in equation (4)
$I_i \forall i = 2, \dots, 5$	Moment of inertia index	$\{\beta\}$	Defined in equation (4)
$J_i \forall i = 1, \dots, 2$	Moments of inertia	Γ	Adaption gain
$u_i \forall i = 1, \dots, 5$	Translational displacements	λ	Positive constant reference trajectory gain
$\theta_i \forall i = 1, \dots, 5$	Rotational displacements	$[\Phi]$	Dynamic system matrix
T_c	Control torque	s	Combined error measure
τ_{ff}	Feedforward control	$\delta\omega$	Differential angular velocity
$\delta\tau$	Control difference	$\delta\dot{\omega}$	Differential angular acceleration
τ_{fb}	Feedback control	dt	Differential time

Following a brief presentation of the parallel axis theorem with a small proof, the parameterization is used to express three-dimensional locations (x, y, z coordinates) are revealed to be expressed strictly using the products of inertia. Thus, ubiquitous diagonalization of the inertia matrix by transforming basis vectors would eliminate the ability to use this formulation presented in equation (12).

Theorem. *Parallel axis theorem.* The moment of inertia $J_y = J_o$ about some point other than the center of mass is the sum of the moment of inertia J_{cm} about the center of mass and the moment of inertia relative to the fixed distance between axes (y and \hat{y} in Figure 6) in the rigid body represented by the product of the total mass and the square of the distance (See equation (11)).

$$J_o = J_{cm} + MR_o^2 \quad (11)$$

The following proof mirrors [32], since the version most closely illuminates the use of parallel axis theorem in this manuscript to reveal the three-dimensional location of the center of mass. The nomenclature of the proof in [32] has been mapped to the nomenclature of the formerly investigated space robot. [33]

Proof. Consider a rigid system of particles of mass $M = \sum_i m_i$, rotating about a fixed axis O along axis y . Place the origin of the coordinate system at the center-of-mass (cm) of our system of particles. The general point i has coordinates (x_i, y_i) , the x coordinate of O is a , and the y coordinate of O is b . Since the center of mass is at $(0,0)$, by the definition of the center-of-mass (the unique point at any given time where the weighted relative position of the distributed mass sums to zero, equation (12) results.

$$x_{cm} = \sqrt{\frac{J_{xy}J_{xz}}{mJ_{xy}}}, \quad y_{cm} = \sqrt{\frac{J_{xy}J_{yz}}{mJ_{xz}}}, \quad z_{cm} = \sqrt{\frac{J_{xz}J_{yz}}{mJ_{xy}}} \quad (12)$$

By definition, $J_o = \sum_i m_i r_{io}^2$, where r_{io} is measured from O , not from mass center. Thus, $r_{io}^2 = (x_i - a)^2 + (y_i - b)^2$ leading to equation (13). But since $x_{cm} = \sum_i \frac{m_i x_i}{M} = 0$ and similarly for y , the cross terms in equation (13) go to zero once the sum over i is done leading to equation (14). Since $(x_i^2 + y_i^2) = r_i^2$, where r_i is now measured from the mass center, and $(a^2 + b^2) = R_o^2$ is just R_o^2 , where R_o is the distance from O to the mass center, equation 32 results. \square

$$J_o = \sum_i m_i [(x_i - a)^2 + (y_i - b)^2] = \sum_i m_i [x_i^2 - 2ax_i + a^2 + y_i^2 - 2by_i^2 + b^2] \quad (13)$$

$$J_o = \sum_i m_i (x_i^2 + y_i^2) + \sum_i m_i (a^2 + b^2) \quad (14)$$

$$J_y = J_{cm} + MR_y^2 \quad (15)$$

Table 5. Table of proximal variables and nomenclature ¹.

Variable / acronym	Definition	Variable / acronym	Definition
$J_y = J_o$	Inertia moment about a fixed axis O along y	y	Robot arm base parallel axis
J_{cm}	Inertia moment about mass center	\hat{y}	Body center parallel axis
M	Total system mass	r_{io}	Radius from parallel axis (not mass center)
R_o	Distance from O to the mass center	r_i	Radius vector to the mass center
R_y	Distance to axis y	m_i	Differential mass element
O	Fixed axis along axis y	(x_i, y_i)	Center of mass coordinates
a	x coordinate of O	b	y coordinate of O

¹ Such tables are offered throughout the manuscript to aid readability.

Equation (12) is extracted from the short proof of parallel axis theorem, since it provides the time varying location of the mass center giving known (or estimated) values of the products of inertia (which would not be available if the matrix had been previously diagonalized by transformation from the body axes basis to the eigenvector basis frame. Thus, the next necessary step of analysis is to elaborate the ability to estimate the inertia matrix, where moments and products together are used to embody the control, while the products of inertia alone are used to parameterize the location of the mass center.

2.9. Persistent Excitation of Estimation

Simple step commands do not provide sufficiently rich input signals to produce faithful estimates of system parameters and locations of the center of mass. More “rich” input signals are utilized to increase estimation confidence.

Definition. *Persistent excitation.* Condition placed upon input commands to ensure the signals provide sufficient information over time to accurately estimate system parameters (eventually manifest as estimated locations of the system center of mass).

A finite impulse response model of order n involves the inversion of the $n \times n$ matrix in equation (16) where $\varphi(t)$ is defined in equation (17) for control input signal u . Estimation will have a unique solution (i.e. the process is “identifiable”) if and only if the matrix in equation (16) is non-singular or invertible. [36]

$$\frac{1}{N} \sum_{i=1}^N \varphi^T(t) \varphi(t) \quad (16)$$

$$\varphi(t) = [u(t-1) \quad u(t-2) \quad \dots \quad u(t-n)] \quad (17)$$

Table 6. Table of proximal variables and nomenclature ¹.

Variable / acronym	Definition	Variable / acronym	Definition
N	Covariance dimension	u	Control
n	Dimension of the control	$(t-i) \forall i = 1, \dots, n$	Indexed time
φ	System equation	φ^T	Transposed system equation

¹ Such tables are offered throughout the manuscript to aid readability.

Impacts of persistent excitation on parameter convergence are described in [37]. Modeling accuracy requires convergence of estimated parameters necessitating persistent excitation. Increased precision and faster convergence stems from relatively stronger excitation. Closed-loop stability of nonlinear, adaptive (or learning) systems is increased by bounding parameter estimation which is also aided by persistent excitation. A step command has one instance of excitation and accordingly only leads to reliable estimation of one parameter. Meanwhile, a sinusoidal command has two instances of excitation (as the curve oscillatory reverses direction twice) leading to reliable estimation of two parameters. Random signals (i.e. Gaussian) are very rich excitation, since values at any pair of times are identically distributed and statistically independent (and hence uncorrelated); and accordingly aid reliable estimation of any number of parameters. Accordingly, white noise is added to command signals in this investigation.

Table 7. Persistent excitation of various input signals ¹.

Input signal	Order of persistent excitation
Step command	1
Sinusoidal commands	2
White noise commands	n

¹ Such tables are offered throughout the manuscript to aid readability.

Propositions for achieving exponentially convergent estimation when lacking persistent excitation is presented in [38–39]. In this research, the approach taken is to add small amplitude white noise to all input command signals, which are based on sinusoids as constructed in Figure B.27 (sinusoids) Figure B.29 (additional white noise) in Appendix B.

Box 2. Input signal recommendation.

Recommended input command: To maximize persistent excitation, autonomous sinusoidal trajectories are recommended with white noise added.

2.10 Generic Robotics On-Orbit Trainer (GROOT) and Long Duration Propulsive EELV Secondary Payload Adapter (LDPE ESPA)

A generic, customizable simulation is the basic orbital and robotic simulation (BOARS), developed and maintained by NASA simulations and graphics branch [40] including two orbital vehicles as depicted in Figure 7 in arbitrary orbits. A robotic dynamic manipulator developed in collaboration with the United States Space Force (USSF) and used at the U.S. Air Force Test Pilot School is attached to one of the vehicles called the Generic Robotics On-Orbit Trainer (GROOT). [41] The software simulations developed and used to produce the results in section 3 of this manuscript utilize configurations scaled from those currently in use for GROOT.

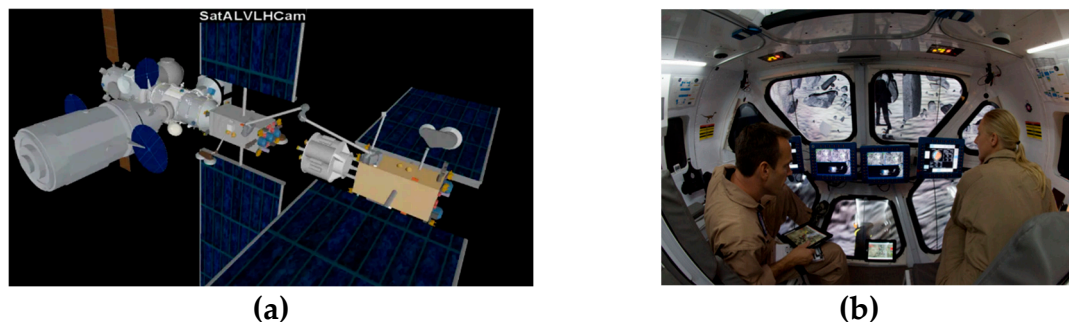


Figure 7. (a) NASA Basic Orbital and Robotic Simulation (BOARS), [38] (b) NASA Exploration Systems Simulations [41] for mission concept evaluation and operational insight to flight crew. Image credit: NASA, used in accordance with published image use policy. [5].

2.11. Implementation Procedures

From section 2, an implementation procedure emerges, summarized in Table 8. The procedure was implemented using MATLAB®/SIMULINK® where the subsystems are depicted in Appendix B to aid repeatability by the readership.

Table 8. Implementation procedure.

1. Use the physics-based governing dynamics to embody the control (section 2.1)
 - a. Do not diagonalize the inertia matrix
 - b. Do not linearize the governing equations of motion
 - c. Establish classical feedback control as comparative benchmark
2. Implement nonlinear feedforward in equation (4) (section 2.2)
3. Use nonlinear, enhanced Luenberger observers (section 2.5) to estimate all motion states and control.

4. Use projection-based non-linear regression learning to learn the fully populated mass moments of inertia (masses and mass locations) including principal moments and off-diagonal products of inertia. (section 2.7)
5. Use off-diagonal products of inertia to estimate location of the center of mass (section 2.8).

Using the implementation procedure in Table 9, simulations were created (as depicted in Appendix B), where the results are next presented in section 3.

Table 9. Algorithm 1: Pseudocode ¹.

Input:	Desired end state: final velocity and final acceleration
Output:	Inertia matrix components and three-dimensional coordinates of the center of gravity
1 begin	
2	Path planning using sinusoidal trajectories:
3	DesiredVelocity = DesiredFinalVelocity * SineEquation
4	DesiredAcceleration = DesiredFinalAcceleration * CosineEquation
5	Assemble feedforward control
6	Parameterize Euler's moment equations into standard regression form
7	Control_1 = DesiredStateMatrix * UnknownInertiaRegressionVector
8	OptionalFeedback = DesiredStateMatrixInverse * EstimatedTorque
9	EstimatedInertia = DesiredStateMatrixInverse * AppliedTorque
10	MassCenterCoordinates = Function(InertiaCrossProducts)
11 end	

¹ Corresponding to Table 8.

3. Results

This section presents the results of simulations of the proposed autonomous, online center of mass determination together nonlinear, coupled control and inertia identification and nonlinear, enhanced Luenberger observers.

3.1. Section Description

Section 3.2 describes the developed results incrementally beginning with the unperturbed system followed by the comparative benchmark. Introduced next is dynamic nonlinear feedforward augmented by 1) PID feedback, and 2) inertia adaption. Next, dynamic, nonlinear feedforward with nonlinear, projection-based regression learning is applied. Estimation performance is compared next to evaluate Luenberger observer performance.

Section 3.3 performs identification of mass moments and products. Identification is compared using nonlinear adaption versus nonlinear, projection-based learning. Section 3.4 extends the results of section 3.3 to reveal the time varying location of the center of mass again comparing nonlinear adaption versus nonlinear, projection-based learning. While the baseline maneuver was a single thirty-degree yaw maneuver, compound maneuvers are evaluated next before summary results are presented in section 3.5.

3.2. Incremental Development and Presentation of Results

First, an unperturbed system is analyzed with classical feedback control. Next, a comparative benchmark for the study is analyzed, a highly perturbed system with classical feedback control. The perturbations include uniform variations to all values in the inertia matrix; random variations (white noise) added to the control to aid estimation; random noise added to sensor outputs;

magnetic disturbance torques, aerodynamic disturbance torques, solar radiation pressure disturbance torques, and gravity gradient disturbance torques.

3.2.1. Unperturbed System

The unperturbed system is used to aid software troubleshooting and for investigating numeric precision. Numerical precision is studied using the quaternion normalization condition [42] displayed in equation (18) as a figure of merit to iterate numerical integration solvers and evaluation step-sizes. Solvers and step-sizes were iterated until the error evaluating the quaternion normalization condition is always zero (or numeric zero) as displayed in Figure 9. Also displayed in Figure 9(b) is a display of the maneuver cost versus time, where cost is calculated as one half the time integral of the square of the control. Each case investigated is compared in part by the amount of control cost to perform the maneuver as one comparative figure of merit in addition to trajectory tracking error means and standard deviations.

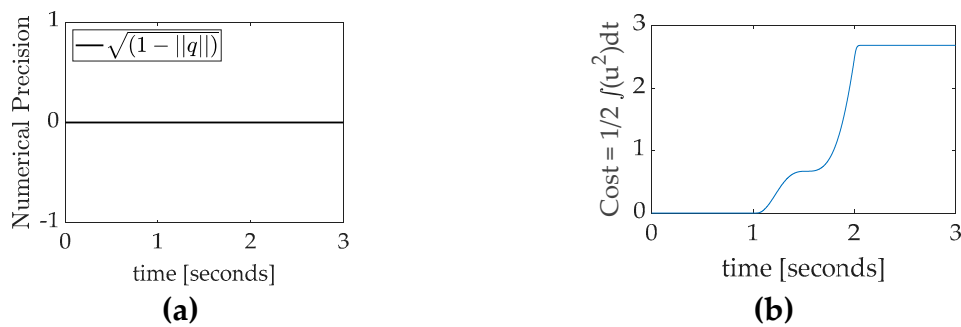


Figure 9. (a) Quaternion normalization condition from equation (18) used to validate numerical precision of simulation results. (b) Quadratic cost of maneuver.

$$\underline{q}(t) \equiv \{q_1 \quad q_2 \quad q_3 \quad q_4\}^T \rightarrow \|\underline{q}\| = 1 \quad (18)$$

The initial case investigated is classical PID control of an unperturbed system without feedforward or inertia adaption or learning, and the case is presented in Figure 10–Figure 11 with quantitative data presented in Table 10.

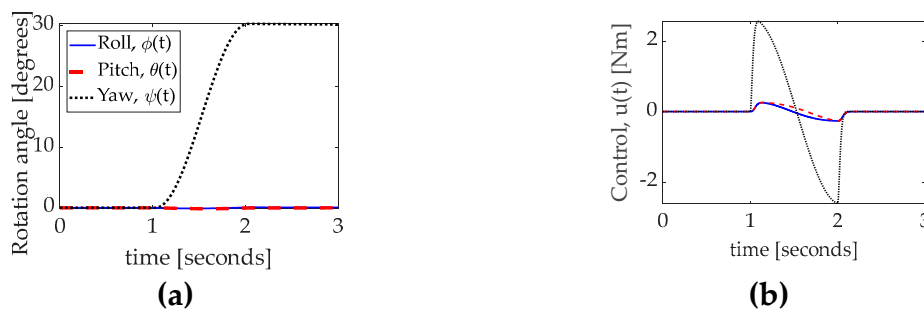


Figure 10. (a) Rotation during unperturbed system maneuver: thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant with roll as solid blue line, pitch as dashed red line, and yaw as dotted black line. The three-dimensional maneuver data presented together in Figure 9(a) is elaborated separately in Figure 11; (b) control utilized during the maneuver. Quantitative data corresponding to the qualitative display in Figure 10 is presented in Table 10.

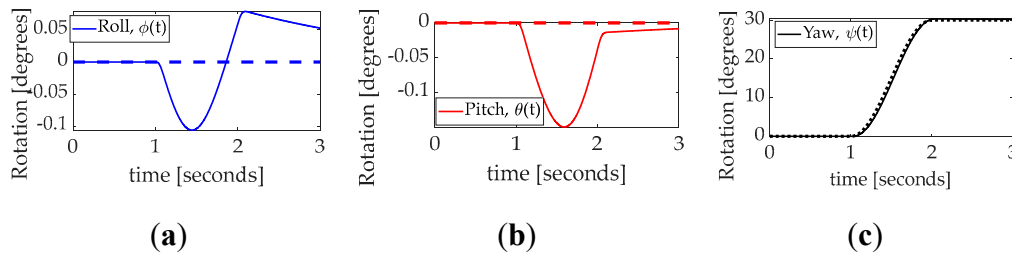


Figure 11. Commanded rotation versus actual rotation during unperturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Commanded rotation versus actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Quantitative data corresponding to the qualitative display in Figure 11 is presented in Table 10.

Table 10. PID controlled unperturbed system.

Cost = 0.0083041	Roll [degrees]	Pitch [degrees]	Yaw [degrees]
Mean	-8.7655×10^{-5}	0.00062823	0.0026512
Standard deviation ¹	0.00096123	0.00088184	0.0084459

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

3.2.2. Comparative Benchmark

The comparative benchmark case of classical PID control without feedforward or learning controlling a highly perturbed system with sensor noise is presented in Figure 12–Figure 13 with quantitative data presented in Table 11. Figure 12a simultaneously displays the roll, pitch, and yaw angles with respect to time. The commanded rotational maneuver is a thirty-degree yaw tracking a sinusoidal trajectory with no commanded roll or pitch, and the maneuver is performed with precision. The three axis controls are displayed in Figure 12b, notice the largest control corresponds to the yaw channel, while the relative smaller two channels of control counter cross-coupled motion inherent with a fully populated inertia matrix. Figure 13 displays the same maneuver data as Figure 12a separated into three displays, one for each axis of motion: roll, pitch, and yaw. The three plots present qualitative information that matches quantitative presentation of data in Table 11.

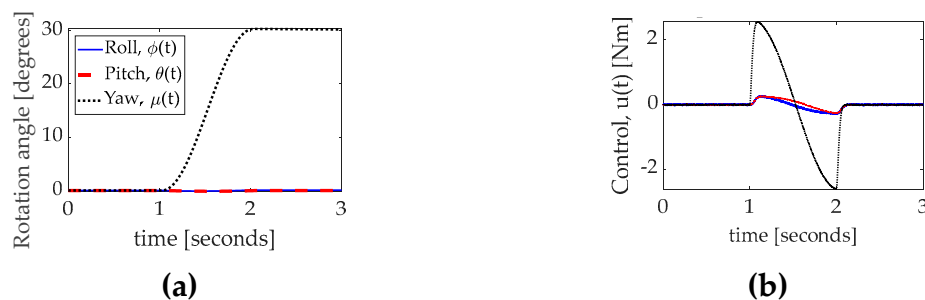


Figure 12. (a) Rotation during unperturbed system maneuver: thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant with roll as solid blue line, pitch as dashed red line, and yaw as dotted black line. The three-dimensional maneuver data presented together in Figure 9(a) is elaborated separately in Figure 11; (b) control utilized during the maneuver. Quantitative data corresponding to the qualitative display in Figure 10 is presented in Table 10.

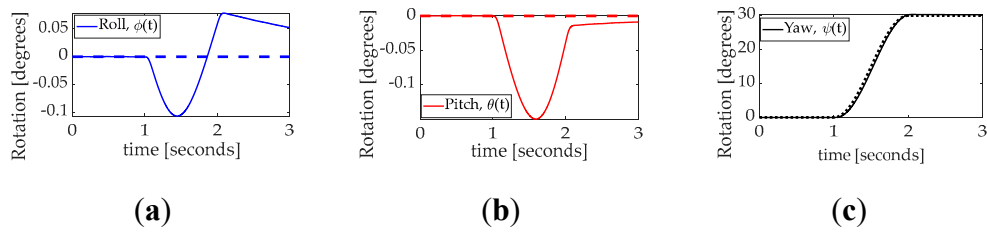


Figure 13. PID controlled highly perturbed system: uniformly varied inertia, sensors with white noise. Commanded rotation versus actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Commanded rotation versus actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Quantitative data corresponding to the qualitative display in Table 11 is presented in Table 13.

Table 11. PID controlled highly perturbed system: uniformly varied inertia, sensors with white noise, thirty-degree yaw maneuver tracking errors.

Cost = 0.0083917	Roll [degrees]	Pitch [degrees]	Yaw [degrees]
Mean	-8.5263×10^{-5}	0.0006306	0.0026536
Standard deviation ¹	0.00096242	0.00088425	0.0084479

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

3.2.3. Dynamic Nonlinear Feedforward with PID Feedback and Inertia Adaption

Having foremost established idealized performance controlling an unperturbed system, next the controlled system was perturbed with some performance degradation. Next, this section displays the results of adding dynamic nonlinear feedforward and inertia adaption. Efforts were not made to optimize the adaption or the feedback gains, since tracking errors assist estimation of mass center location (the goal of the study).

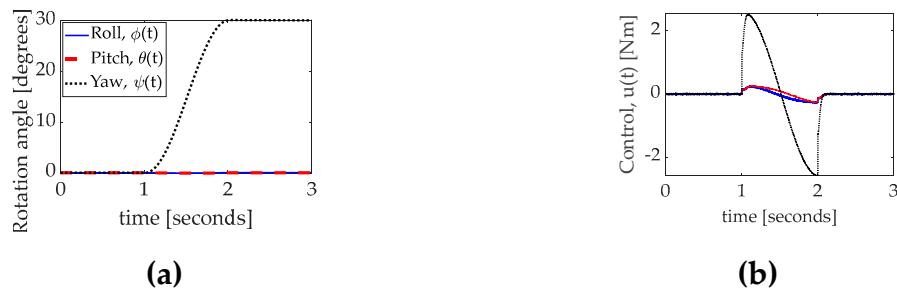


Figure 14. Dynamic nonlinear feedforward with PID feedback and inertia adaption. (a) Rotation during highly perturbed (uniformly varied inertia, sensors with white noise) thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant with roll as solid blue line, pitch as dashed red line, and yaw as dotted black line. Quantitative data corresponding to the qualitative display in Figure 14 is presented in Table 13.

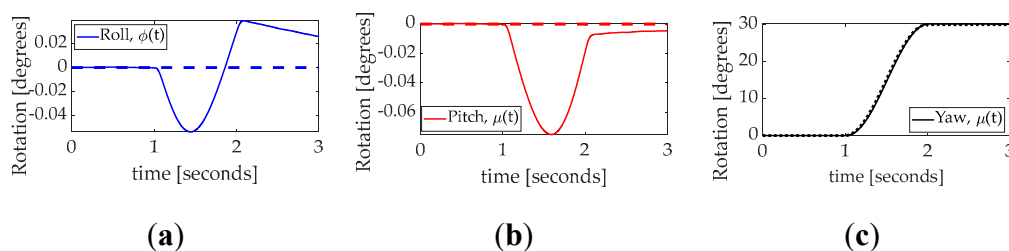


Figure 15. Dynamic nonlinear feedforward with PID feedback and inertia adaption. Commanded rotation versus actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Commanded rotation versus actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Quantitative data corresponding to the qualitative display Figure 15 is presented in Table 13.

Table 12. Dynamic nonlinear feedforward with PID feedback and inertia adaption. Highly perturbed (uniformly varied inertia, sensors with white noise) thirty-degree yaw maneuver tracking errors.

Cost = 0.0075635	Roll [degrees]	Pitch [degrees]	Yaw [degrees]
Mean	-2.7565×10^{-5}	0.0003158	0.0013581
Standard deviation ¹	0.00047434	0.00044734	0.0042131

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

3.2.4. Dynamic Nonlinear Feedforward with PID Feedback and Nonlinear Projection-Based Regression Learning

Having in the previous section displayed the results of adding dynamic nonlinear feedforward and inertia adaption, this section replaces inertia adaption with nonlinear projection-based regression learning. As before, efforts were not made to optimize the adaption or the feedback gains, since tracking errors assist estimation of mass center location (the goal of the study).

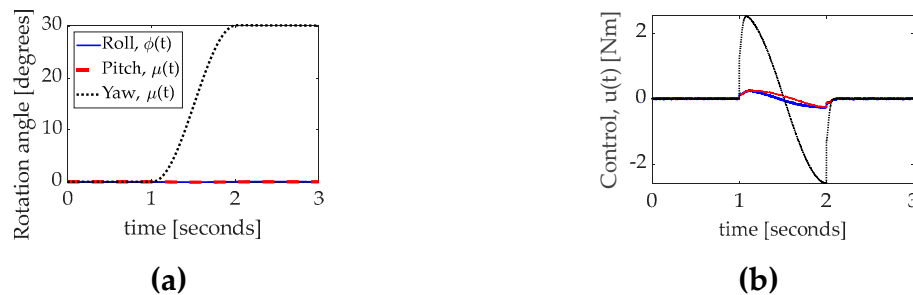


Figure 16. Dynamic nonlinear feedforward with PID feedback and nonlinear projection-based regression learning. (a) Rotation during highly perturbed (uniformly varied inertia, sensors with white noise) thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant with roll as solid blue line, pitch as dashed red line, and yaw as dotted black line. Quantitative data corresponding to the qualitative display in Figure 16 is presented in Table 13.

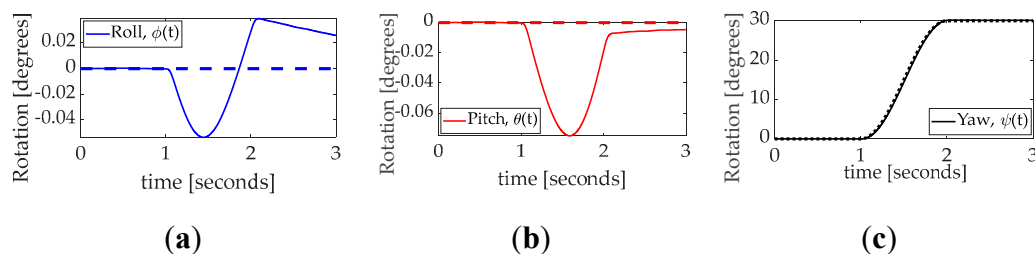


Figure 17. Dynamic nonlinear feedforward with PID feedback nonlinear projection-based regression learning. Commanded rotation versus actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Commanded rotation versus actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Quantitative data corresponding to the qualitative display in Figure 17 is presented in Table 13.

Table 13. Dynamic nonlinear feedforward with PID feedback and nonlinear feedforward nonlinear projection-based regression learning. Highly perturbed (uniformly varied inertia, sensors with white noise) thirty-degree yaw maneuver tracking errors.

Cost =0.0075635	Roll [degrees]	Pitch [degrees]	Yaw [degrees]
Mean	-2.7565×10^{-5}	0.0003158	0.0013581
Standard deviation ¹	0.00047434	0.00044734	0.0042131

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

Box 3. Identical tracking: adaption vs. nonlinear projection-based regression learning.

Estimation vs. tracking control: Notice identical trajectory tracking results in Table 12 and Table 13. Adaption and nonlinear projection-based regression learning methods are strictly used to estimate inertia components and mass center location and do not affect trajectory tracking performance.

3.2.5. Estimator Performance Comparison

With the backdrop of introducing tracking control options, the remaining topic of study is the performance of state estimation by disparate Luenberger observers. Those estimated states are used by both the inertia adaption scheme and the nonlinear projection-based regression learning. The goal of the comparison is merely to choose the Luenberger instantiations. The observer comparison is presented qualitatively in Figure 18 and quantitatively in Table 14. The nonlinear, enhanced observer was slightly superior in the maneuver axis, but relatively substantially inferior in the non-maneuver axes.

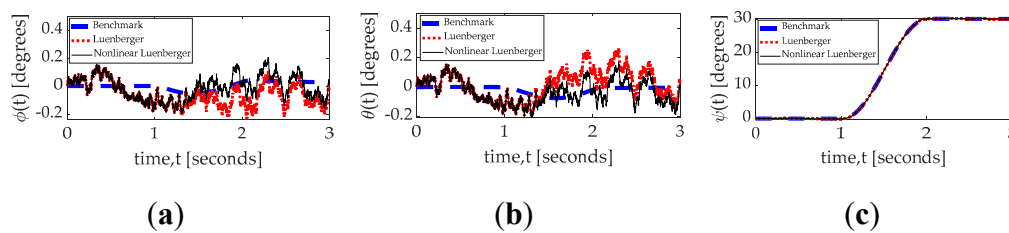


Figure 18. Estimated actual rotation during highly perturbed thirty-degree yaw with time in seconds on the abscissa and estimated rotation angle in degrees on the ordinant. (a) roll; (b) pitch; (c) yaw. Actual rotation angle (dashed blue line), rotation angle estimated by enhanced Luenberger observer (dotted red line), and rotation angle estimated by enhanced, nonlinear Luenberger observers. (a) roll; (b) pitch; (c) yaw. Quantitative data corresponding to the qualitative display in Figure 18 is presented in Table 14.

Table 14. Observer estimation percent performance comparison. Highly perturbed (uniformly varied inertia, sensors with white noise) thirty-degree yaw maneuver tracking errors.

	Mean roll [degrees]	Mean pitch [degrees]	Mean yaw [degrees]
Benchmark Luenberger	--	--	--
Nonlinear Luenberger ¹	77%	144.58%	-1.02%

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

Box 4. Observer recommendation.

Observer recommendation: Simultaneous use of both Luenberger styles is recommended to leverage their relative strengths in the maneuver-axis and non-maneuver axis.

3.3. Identification of Mass Moments and Products

The estimations from the Luenberger observers are used to estimate inertia moments and products using two disparate methods: nonlinear adaption, and nonlinear projection-based regression learning described respectively in sections 3.3.1 and 3.3.2.

3.3.1. Identification of Mass Moments and Products by Nonlinear Adaption

Nonlinear adaption as proposed in [43] and improved in [44] and [45] were utilized to estimate the six inertia moments and products. The method was simulated in [46] and experimentally validated in [47].

A combined measure of error linearly combines angular acceleration and velocities using a constant gain. The gained, combined measure of error is multiplied by the system matrix with an adaption gain to establish the adaption rate which is integrated to produce inertia component adaptions. The MATLAB®/SIMULINK® simulation subsystems are presented in appendix B.

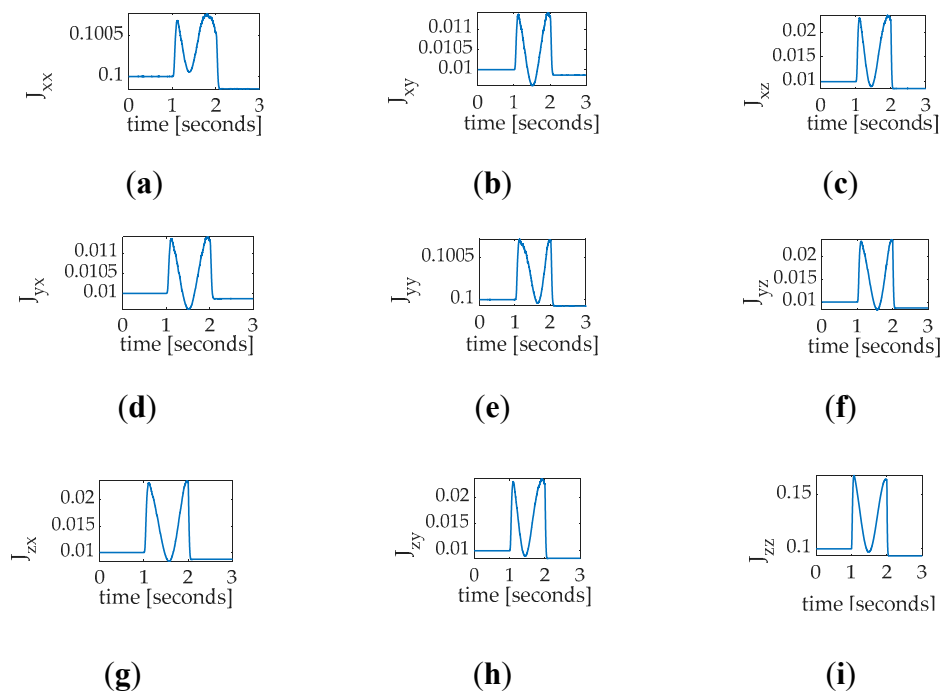


Figure 19. Nonlinear adaption derived inertia moments and products: Time varying mass moments and products for a single, thirty-degree yaw maneuver with coupled motion in roll and pitch, where the command has white noise added to aid persistent excitation (a) J_{xx} ; (b) J_{xy} ; (c) J_{xz} ; (d) J_{yx} ; (e) J_{yy} ; (f) J_{yz} ; (g) J_{zx} ; (h) J_{zy} ; (i) J_{zz} . Notice the initial flat-line behavior during the initial quiescent period is followed by rapid location learning during the maneuver excitement, lastly followed by a slow, oscillatory settling following the maneuver during regulation. The moments and products of inertia seem to converge towards a certain value during the maneuver excitement but not during either period of regulation, lacking persistent excitation. .

3.3.2. Identification of Mass Moments and Products by Nonlinear Projection-Based Regression Learning

Nonlinear projection-based regression learning as proposed just last year in [48] to learn the six inertia moments and products. Referring to Figure 20, the method utilized included analysis and simulation, while validating experimentation is proposed in this present manuscript.

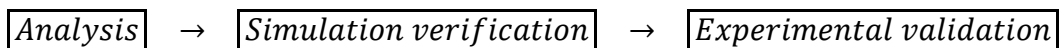


Figure 20. Process stages adopted in this study: analyze and hypothesize followed by verification of analysis with modeling and simulation validated experimentally.

Angle trajectory tracking error alone (not in a combine error measure as done with nonlinear adaption) is projected on the system matrix used to embody the nonlinear feedforward control parameterized in regression form without linearization, diagonalization, or otherwise simplification. Figure 21 reveals the mass moments and products estimated by the learning algorithm, while the corresponding MATLAB®/SIMULINK® simulation subsystems are presented in appendix B.

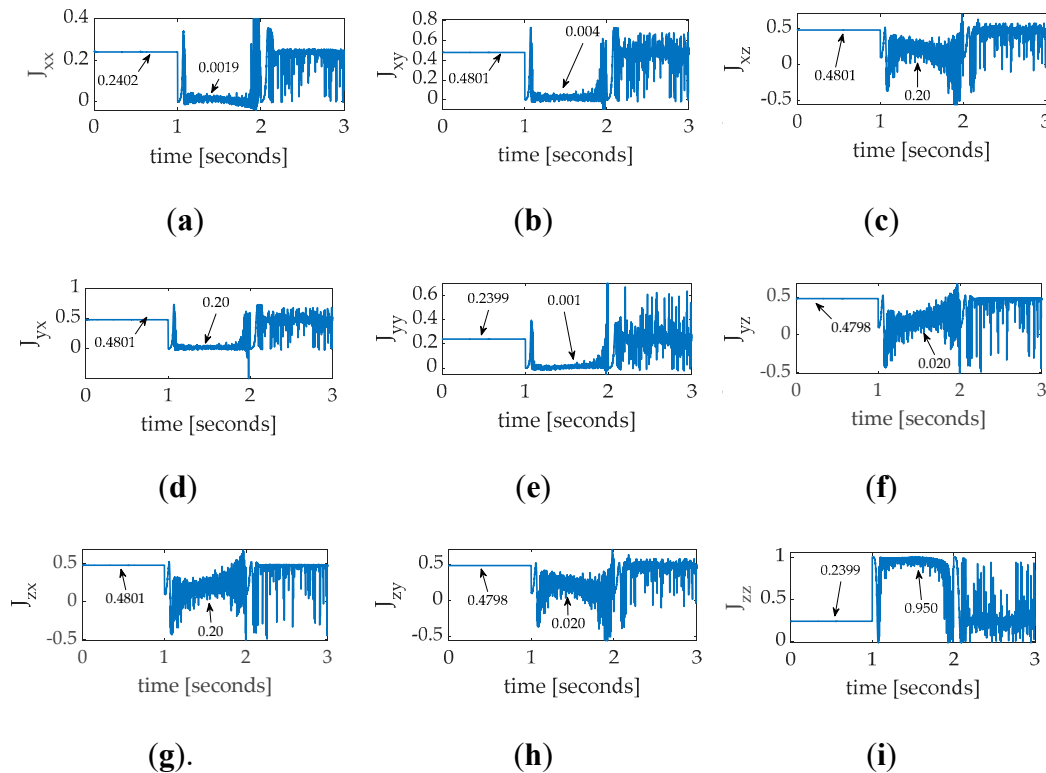


Figure 21. Projection regression-based learning inertia moments and products: Time varying mass moments and products for a single, thirty-degree yaw maneuver with coupled motion in roll and pitch, where the command has white noise added to aid persistent excitation (a) J_{xx} ; (b) J_{xy} ; (c) J_{xz} ; (d) J_{yx} ; (e) J_{yy} ; (f) J_{yz} ; (g) J_{zx} ; (h) J_{zy} ; (i) J_{zz} . Notice the initial flat-line behavior during the initial quiescent period is followed by rapid location learning during the maneuver excitement, lastly followed by a slow, oscillatory settling following the maneuver during regulation. The moments and products of inertia seem to converge towards a certain value during the maneuver excitement but not during either period of regulation, lacking persistent excitation.

3.4. Location of Mass Center

Time varying estimates of mass moments and products may be used with equation (12) to directly calculate the location of the mass center (as a reminder that the inertia matrix must not be diagonalized). The estimates of inertia moments and products are produced with two disparate methods: nonlinear adaption, and nonlinear projection-based regression learning respectively in sections 3.4.1. and 3.4.2.

3.4.1. Identification of Mass Center Coordinates by Nonlinear Adaption

Notice in Figure 22 adaption occurs twice: at the beginning and the end of trajectory tracking. The roll channel depicted in subfigure (a) seems to indicate a proper mass center location roughly at 0.15 in the x direction. Meanwhile, the pitch channel adaption deviates from 0.1 ± 0.02 in the y direction, while the yaw channel location adaption displayed in subfigure (c) behaves similar to the

roll channel, seemingly indicating a mass center location roughly 0.225 in the z direction. Lastly, notice during the post-maneuver quiescent period, the location estimate does not return to the initial estimate (at the beginning of the pre-maneuver quiescent period). As a reminder the control signal contains a sinusoidal trajectory command (which is persistently exciting of order 2) with white noise added (which is persistently exciting of order n).

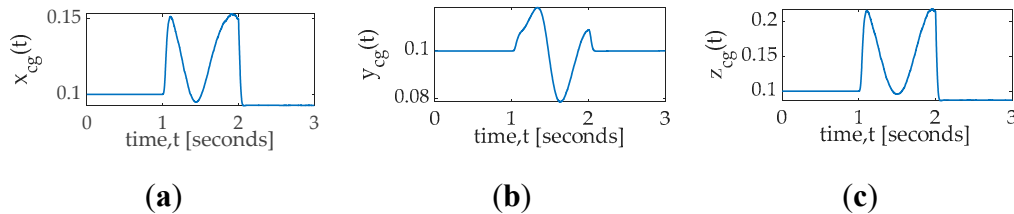


Figure 22. Nonlinear adaption time varying location of the center of mass: For a single, thirty-degree yaw maneuver with coupled motion in roll and pitch, where the command has white noise added to aid persistent excitation (a) x coordinate; (b) y coordinate; (c) z coordinate. Notice the initial flat-line behavior during the initial quiescent period is followed by rapid location learning during the maneuver excitement, lastly followed by a slow, oscillatory settling following the maneuver during regulation. The location seems to approach the body axes origin in the x and y directions with slight movement away in z direction. (d) three-dimensional plot of learned cg location using projection regression-based learning; (e) three-dimensional plot of learned cg location using nonlinear adaption for inertia calculations. Both (d) and (e) use identical deterministic relationships to locate the center of mass using the products of inertia.

3.4.2. Identification of Mass Center Coordinates by Nonlinear Projection-Based Regression Learning

Notice in Figure 23 adaption occurs many more times than twice. Especially focusing on the period between the beginning and the end of trajectory tracking, the roll channel depicted in subfigure (a) seems to indicate a proper mass center location roughly at 0.44 in the x direction. Meanwhile, the pitch channel adaption deviates from 0.20 in the y direction, while the yaw channel location adaption displayed in subfigure (c) seemingly indicating a mass center location roughly 1.35 in the z direction. Again, notice during the post-maneuver quiescent period, the location estimate does not return to the initial estimate (at the beginning of the pre-maneuver quiescent period). As a reminder the control signal contains a sinusoidal trajectory command (which is persistently exciting of order 2) with white noise added (which is persistently exciting of order n).

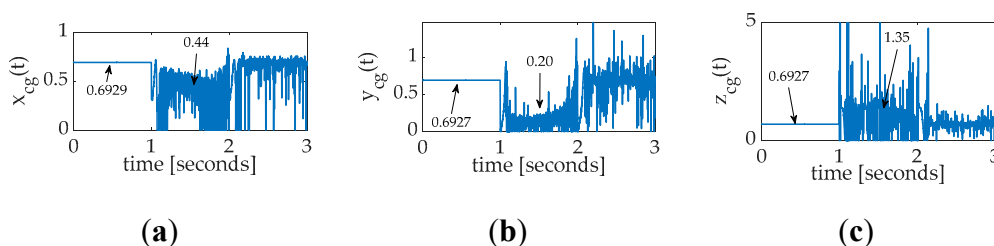


Figure 23. Projection regression-based learning time varying location of the center of mass: Three-dimensional coordinate estimates (a) \hat{x} , (b) \hat{y} , and (c) \hat{z} ; for a single, thirty-degree yaw maneuver with coupled motion in roll and pitch, where the command has white noise added to aid persistent excitation. Notice the initial flat-line behavior during the initial quiescent period is followed by rapid location learning during the maneuver excitement. Furthermore notice the degree of convergence spans roughly 253, 493, and 657 millimeters in each axis respectively in mere seconds.

Box 5. Mass center recommendation.

Locating the mass center: Compared to using nonlinear adaption, use of nonlinear projection–based regression learning was more sensitive to identical cases of persistent excitation and accordingly produced better location of mass center.

3.4.3. Location of Center of Mass: Three–Dimensional Comparison

Both methods of inertia matrix components identification (adaption and learning) were implemented and compared in Figure 24. Immediately evident upon first glance is the fact the adaptive approach converges relatively slowly compared to projection regression two norm optimal learning. During developing the learning approach in section 2 of this manuscript, the notion of having optimal estimates at every time step seemed attractive, but clearly has drawbacks illustrated in Figure 23a and Figure 24 compared to the nonlinear adaptive approach whose results are displayed in Figure 23b and Figure 22. The two–norm optimal pseudoinverse calculation in the learning approach seems to be highly excited by the presence of so many perturbations (indicating potentially poor applicability to deployed space robotic systems).

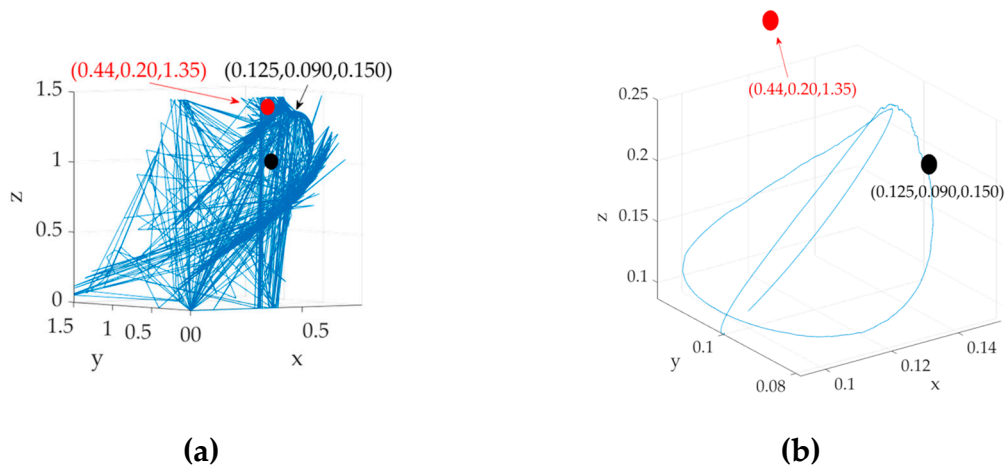


Figure 24. Comparison, time varying location of the center of mass: (a) three–dimensional plot of learned cg location using **projection regression–based learning**; (b) three–dimensional plot of learned cg location using **nonlinear adaption** for inertia calculations. Both (a) and (b) use identical deterministic relationships to locate the center of mass using the products of inertia.

3.4. Compound Maneuvers

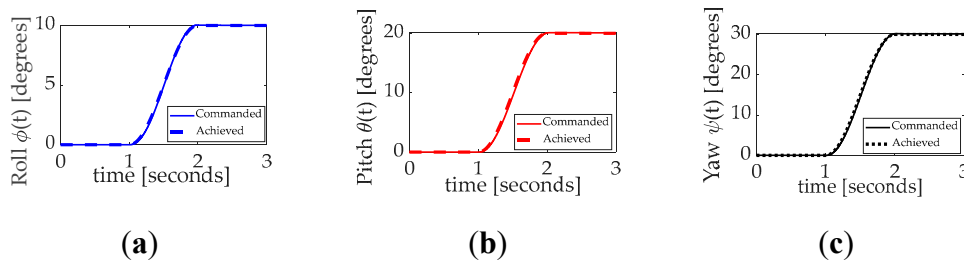


Figure 25. Compound maneuver trajectory tracking errors. Projection regression–based learning with nonlinear Luenberger observers. (a) roll; (b) pitch; (c) yaw.

Table 15. Compound maneuver trajectory tracking errors. Projection regression–based learning with nonlinear Luenberger observers.

	Roll [degrees]	Pitch [degrees]	Yaw [degrees]
Error mean	0.00028129	-4.1077×10^{-5}	0.0027575

Error standard deviations	0.0084356	0.0054244	0.011515
1			

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

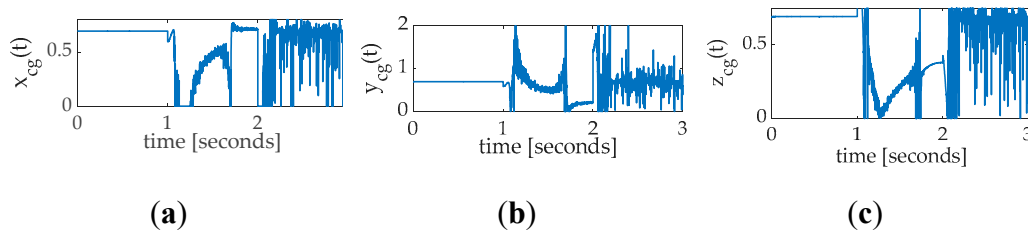


Figure 26. Compound maneuver estimated mass center location. Projection regression-based learning with nonlinear Luenberger observers. (a) \hat{x} -coordinate; (b) \hat{y} -coordinate; (c) \hat{z} -coordinate.

Box 6. Compound maneuver recommendation.

Single axis versus compound axis maneuvers: Compound maneuvers eliminated learning convergence experienced with single-axis maneuvers. Single axis maneuvers should be sequenced sequentially amongst all three axes.

3.5. Summary of Results

Results displayed through the manuscript are consolidated in this section and presented as percent performance difference relative to a comparative benchmark. Narrative results are subsequently discussed in Section 4 Discussion aiming to highlight substantial results in plain verbiage (e.g. no use of “microradian”, etc.) in favor of percent performance difference. Superlative results are highlighted in bold font to amplify information transmission when readers scan the manuscript.

Table 16. Attitude control trajectory tracking percent performance difference.

Maneuver case	Roll errors		Pitch errors		Yaw errors	
	Mean	Deviation	Mean	Deviation	Mean	Deviation
Unperturbed PID unperturbed system*	3%	0%	0%	0%	899%	0%
Benchmark case **	--	--	--	--	--	--
Feedforward + feedback + adaption*** or learning ****	-68%	-51%	-50%	-49%	-49%	-50%

* Classical PID controlled perturbed slew without feedforward or learning controlling the highly perturbed† system ** Classical PID controlled perturbed slew with feedforward but no adaption or learning controlling the highly perturbed† system *** Classical PID controlled perturbed slew with feedforward plus nonlinear inertia adaption controlling the highly perturbed† system **** Classical PID controlled perturbed slew with feedforward plus inertia learning controlling the highly perturbed† system † Assumed mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

Table 17. Observer estimation percent performance comparison. Highly perturbed (uniformly varied inertia, sensors with white noise) thirty-degree yaw maneuver tracking errors.

	Mean roll [degrees]	Mean pitch [degrees]	Mean yaw [degrees]
Benchmark Luenberger	--	--	--
Nonlinear Luenberger ¹	77%	144.58%	-1.02%

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

Table 18. Performance comparison of controlling highly perturbed system (uniformly varied inertia, sensors with white noise) ¹ unless otherwise stated for a thirty-degree yaw maneuver.

Case	Cost
PID controlled unperturbed system	-1%
Benchmark PID feedback	--
Nonlinear feedforward with inertia adaption, and PID feedback	-10%
Nonlinear feedforward with regression-based inertia learning	-10%

¹ Mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

Table 19. Locating center of mass performance comparison. ****+.†.

Maneuver case	x coordinate	y coordinate	z coordinate
Nonlinear adaption	7%	0%	13%
Nonlinear projection-based regression learning	36%	71%	-95%

**** Classical PID controlled perturbed slew with feedforward plus inertia learning controlling the highly perturbed† system † Assumed mass moments uniformly perturbed 50%, control contains noise with variance 0.0001, and white sensor noise with 0.0334 degrees deviation.

4. Discussion

Nonlinear adaptive methods for locating the mass center proved effective, but less so than nonlinear projection-based regression learning. Using identical excitation, nonlinear adaptive methods were able to establish an average seven percent difference in coordinates of the mass center, while an average over seventy percent difference was achievable using nonlinear projection-based regression learning. Listed in Table 20, several recommendations are advised to repeat these results.

Table 20. Most significant recommendations.

1. Use the physics-based governing dynamics to embody the control (section 2.1)
2. To maximize persistent excitation, autonomous sinusoidal trajectories are recommended with white noise added. (section 2.9)
3. Simultaneous use of both Luenberger styles is recommended to leverage their relative strengths in the maneuver-axis and non-maneuver axis. (section 3.2)
4. Compared to using nonlinear adaption, use of nonlinear projection-based regression learning was more sensitive to identical cases of persistent excitation and accordingly produced better location of mass center. (section 3.4.2)
5. Compound maneuvers eliminated learning convergence experienced with single-axis maneuvers. Single axis maneuvers should be sequenced sequentially amongst all three axes. (section 3.4.3)

4.1 Future Research Directions

While this prequel study introduces the first two steps of investigation (analysis and verification in simulations), validation with spaceflight experiments is planned for March 2025, after which a sequel will be published with the results of the spaceflight tests.

4. Conclusion

Arguably, reference [22] presents the best recently proposed comparative benchmark, since that method was experimentally validated on the GRACE-C satellite. The method effectively generated convergent estimates of mass center coordinates, where the convergence spanned just greater than 0.3 millimeters over several hundred days. Meanwhile convergence using the proposed methods (in simulations) span roughly 200–600 millimeters respectively in each axis, and convergence occurs in mere seconds assuming a single-axis slew maneuver. Multi-axis, compound slew maneuvers were completely ineffective using the proposed approach to estimate coordinates of the center of mass. Succinctly, the proposed methods seem to indicate advantages of speed and quantity of coordinate convergence when compared to the state-of-the-art comparative benchmark. Spaceflight experiments planned for March 2025 will evaluate the efficacy of this proposed approach empirically.

Author Contributions: “Conceptualization, T.S.; methodology, T.S.; software, T.S.; validation, T.S.; formal analysis, T.S.; investigation, T.S.; resources, T.S.; data curation, T.S.; writing—original draft preparation, T.S.; writing—review and editing, T.S.; visualization, T.S.; supervision, T.S.; project administration, T.S.; funding acquisition, T.S. All authors have read and agreed to the published version of the manuscript.” Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding. The APC was funded by the corresponding author.

Data Availability Statement: Data supporting reported results can be obtained by contacting the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The nonlinear feedforward is developed step-by-step in this appendix.

$$\tau = [J]\{\dot{\omega}\} + \{\omega\} \times [J]\{\omega\} \quad (20)$$

$$\tau = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{Bmatrix} + \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \times \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \quad (21)$$

$$\tau = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{Bmatrix} + \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \times \begin{Bmatrix} J_{xx}\omega_x + J_{xy}\omega_y + J_{xz}\omega_z \\ J_{yx}\omega_x + J_{yy}\omega_y + J_{yz}\omega_z \\ J_{zx}\omega_x + J_{zy}\omega_y + J_{zz}\omega_z \end{Bmatrix} \quad (22)$$

$$\tau = \begin{bmatrix} J_{xx}\dot{\omega}_x + J_{xy}\dot{\omega}_y + J_{xz}\dot{\omega}_z \\ J_{yx}\dot{\omega}_x + J_{yy}\dot{\omega}_y + J_{yz}\dot{\omega}_z \\ J_{zx}\dot{\omega}_x + J_{zy}\dot{\omega}_y + J_{zz}\dot{\omega}_z \end{bmatrix} + \begin{bmatrix} J_{yy}\omega_y\omega_z + J_{zz}\omega_y\omega_z \\ J_{xx}\omega_x\omega_y - J_{zz}\omega_y\omega_z \\ -J_{xx}\omega_x\omega_y + J_{yy}\omega_y\omega_z \end{bmatrix} \quad (23)$$

$$\tau = \begin{bmatrix} J_{xx}\dot{\omega}_x + J_{xy}\dot{\omega}_y + J_{xz}\dot{\omega}_z + J_{yy}\omega_y\omega_z + J_{zz}\omega_y\omega_z \\ J_{yx}\dot{\omega}_x + J_{yy}\dot{\omega}_y + J_{yz}\dot{\omega}_z + J_{xx}\omega_x\omega_y - J_{zz}\omega_y\omega_z \\ J_{zx}\dot{\omega}_x + J_{zy}\dot{\omega}_y + J_{zz}\dot{\omega}_z - J_{xx}\omega_x\omega_y + J_{yy}\omega_y\omega_z \end{bmatrix} \quad (24)$$

$$\tau = \underbrace{\begin{bmatrix} \dot{\omega}_x & \dot{\omega}_y & \dot{\omega}_z & -\omega_y\omega_z & 0 & \omega_y\omega_z \\ \omega_x\omega_y & \dot{\omega}_x & 0 & \dot{\omega}_y & \dot{\omega}_z & -\omega_y\omega_z \\ \omega_x\omega_z & 0 & \dot{\omega}_x & \omega_x\omega_y & \dot{\omega}_y & \dot{\omega}_z \end{bmatrix}}_{[\Phi]} \underbrace{\begin{Bmatrix} J_{xx} \\ J_{xy} \\ J_{xz} \\ J_{yy} \\ J_{yz} \\ J_{zz} \end{Bmatrix}}_{\beta} \quad (25)$$

Appendix B

This appendix displays all the simulation subsystems permitting repeatability by the readership using MATLAB®/SIMULINK®. Implementation steps follow:

1. Create a scheme of command. A sinusoidal scheme is depicted in Figure B.27 and may be constructed by dragging and dropping the blocks depicted from the library to the simulation. The switching logic produces smooth sinusoids for angular displacement, velocity, and acceleration. The switch institutes to regulation at the final commanded value smoothly exiting the sinusoidal curves respectively. The syntax for the sinusoidal commands is displayed in Figure B.28
2. The attitude controllers are displayed in Figure B.29 including both feedforward and feedback controllers. The project-based learning algorithm is also included. Additionally, notice there is a manual switch away from the input sinusoidal trajectories and optionally towards a simple step command. Lastly, notice artificially faked inertia values and random perturbations are included as switchable options as well.
3. Projection-based learning and/or nonlinear adaption feedback is displayed in Figure B.30, Figure B.31, and Figure B.32,

The implementation steps elaborated so far in Appendix B yield an ability to produce endpoint commands, autonomous trajectory generation, and trajectory tracking abilities, where the outputs include the control torque (to be sent to the space robot). Additional outputs include time-varying inertia estimates and location of the center of mass. The control signal is routed the actuator on the space robot.

4. The reaction wheel actuator is displayed in Figure B.35, where the output torque from the actuator is routed to the governing equations of motion depicted in Figure B.36 with expanded drilldown elaboration in Figure B.37 and Figure B.38. The governing equations of motion produce angular velocities relative to the inertial reference frame but need to be expressed in the coordinates of a body frame of interest.
5. Figure B.38 displays the high-level kinematics topology. Drill downs for quaternions is displayed in Figure B.39 where the quaternions are used to construct a nine-element direction cosine matrix transformation between inertial and body coordinates. The direction cosine matrix is used to extract Euler angles where a 3–2–1 rotation sequence is assumed, and that code is depicted in Figure B.40.
6. While the true Euler angles cannot be known by the robot operators, sensors are simulated with additive noise and filters designed to counteract the noise in Figure B.41.
7. The direction cosine matrix conveniently reveals the vector locations of the body axes and is used to calculated environmental disturbance torques.

The implementation steps elaborated so far in Appendix B now additionally include kinetics, kinematics, noisy sensors and filters. Lastly, the simulation of environmental disturbances is elaborated.

8. The top-level topology of environmental disturbances is displayed in Figure B.42 where each subsystem is individually displayed in Figure B.43 for atmospheric, magnetic, solar radiation, and gravity gradient disturbances.

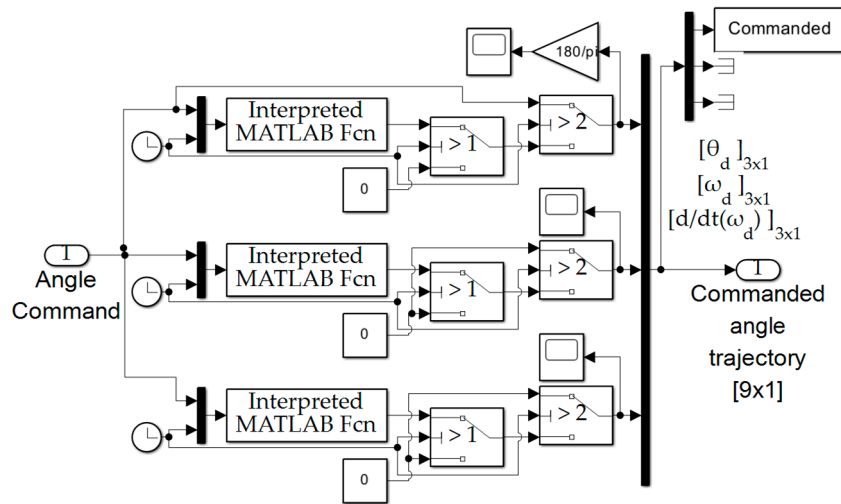


Figure B.27. Internal content of trajectory generation subsystem depicted in Figure 3. Notice the sinusoidal attitude trajectories are formulated from the single input of angle commanded outside the space robot (presumably either autonomously by the demands of the translational motion subsystem or alternatively externally supplied by a human operator aiding the space robot’s motion with remote-controls).

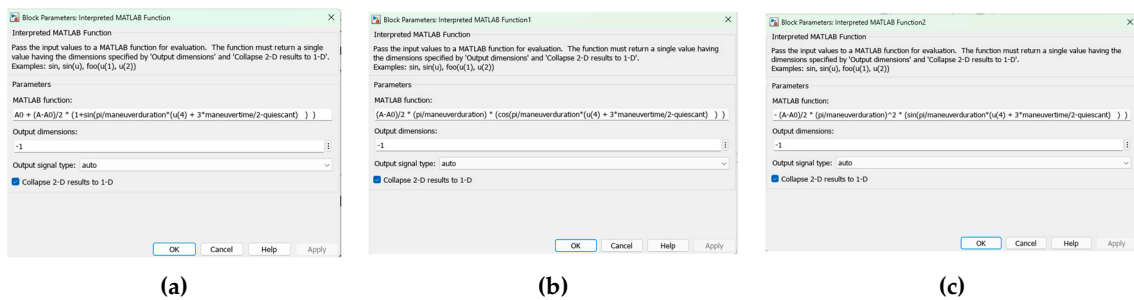


Figure B.28. Pseudocode (labeled “Block Parameters”) used in the three “Interpreted MATLAB Fcn” blocks in Figure B.27 encoding the angle, angular rate, and angular acceleration commands respectively. Top-to-bottom subsystems in Figure B.27 correspond to left-to-right block parameters in this figure. The pseudocode is provided to aid repeatability of the research results.

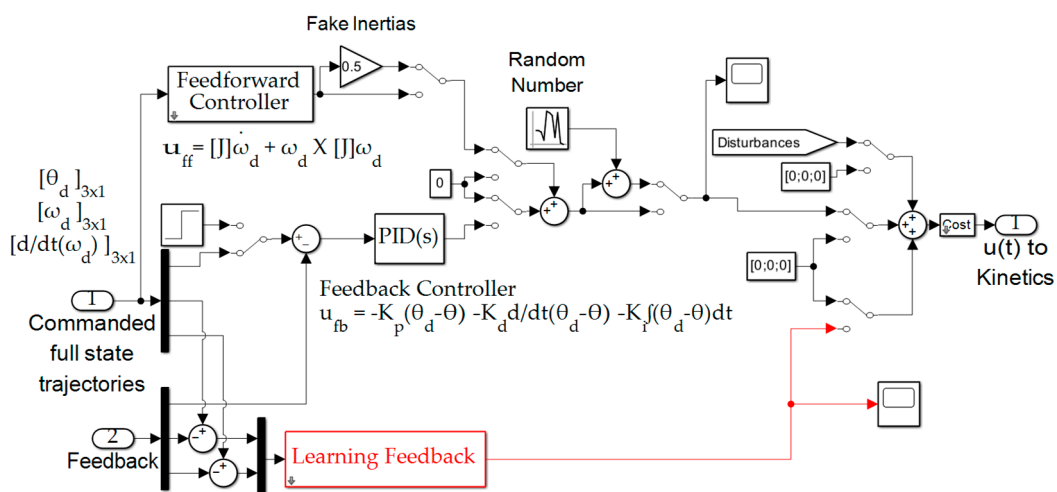


Figure B.29. Internal content of “Controllers” subsystem depicted in Figure 3. Notice the use of manual switches permitting simulation iteration where all facets are held constant except the facet being investigated. Furthermore, notice the “fake inertia” gain block at the center–top of the figure permits simulation users to

artificially make the feedforward command (based on the assumed values of the mass moments of inertia) artificially smaller or larger than assumed. Doing so permits simulation debugging, e.g. if the user artificially makes the mass moments to be assumed much larger, the learning feedback should produce adaptation of mass moments towards the values manually altered in the gain block.

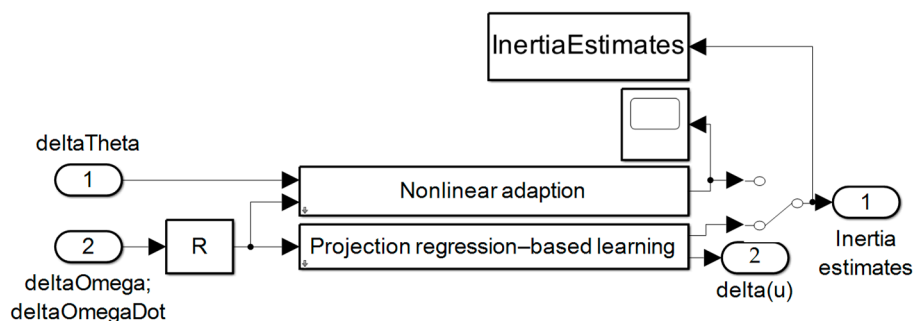


Figure B.30. Internal content of “Learning Feedback” subsystem.

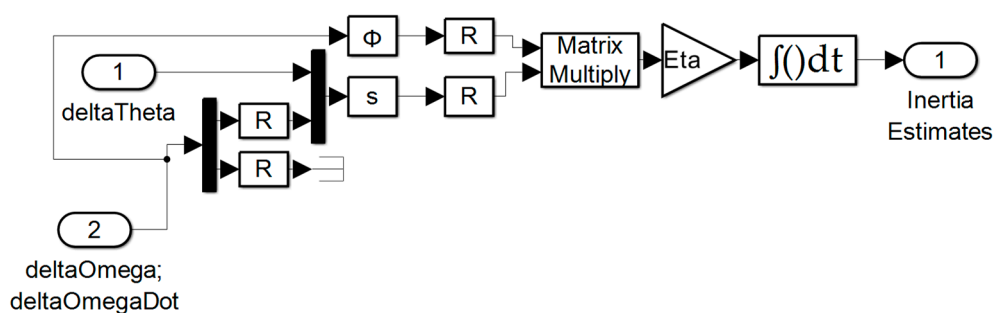


Figure B.31. Internal content of “Nonlinear adaption” option in the “Learning Feedback” subsystem.

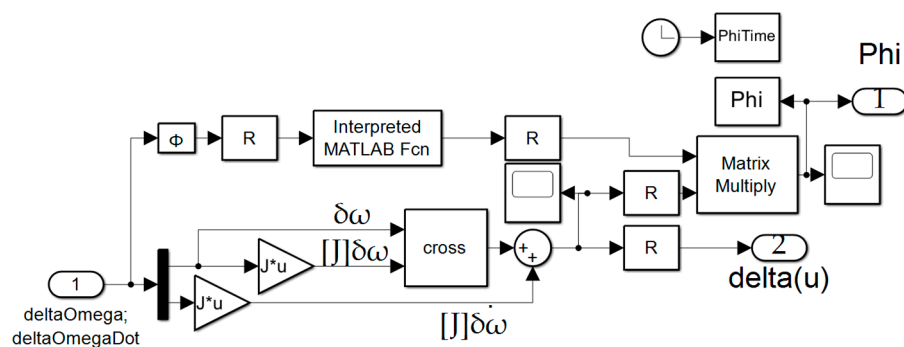


Figure B.32. Internal content of “Projection regression-based learning” option in the “Learning Feedback” subsystem.

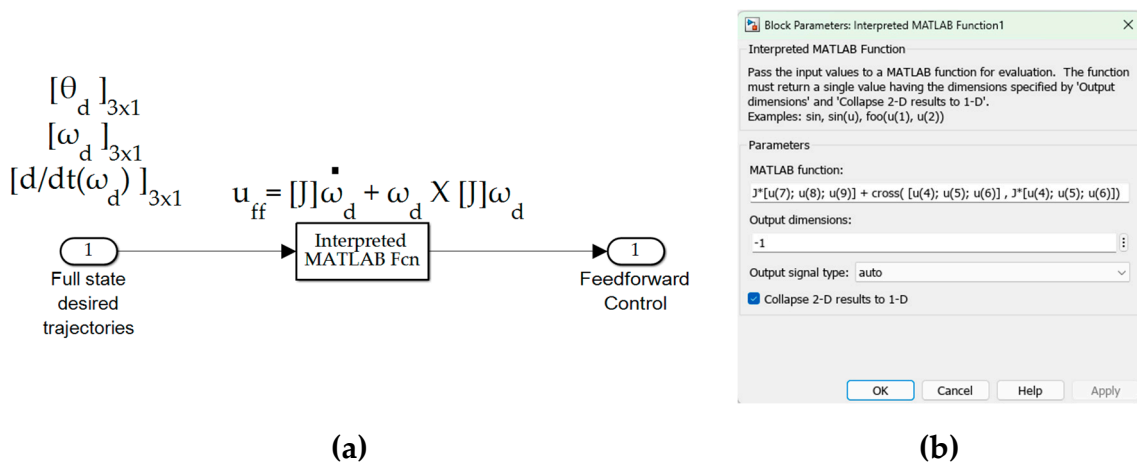


Figure B.33. (a) Internal content of “Feedforward Control” subsystem in **Figure B.29**. (b) internal pseudocode of the “Interpreted MATLAB FCN” block in subfigure (a).

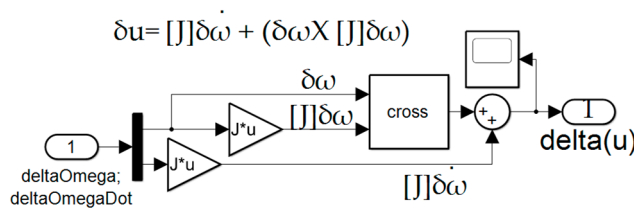


Figure B.34. Internal content of “Learning Feedback” subsystem depicted in **Figure B.29**.

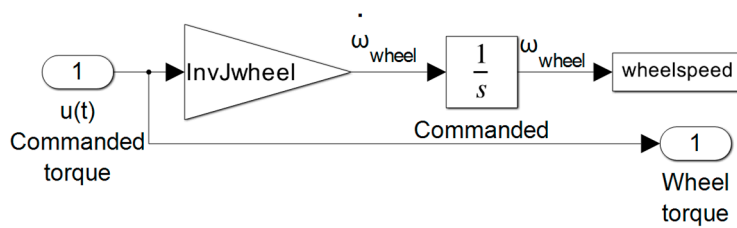


Figure B.35. Internal content of “Actuators” subsystem depicted in **Figure 3**.

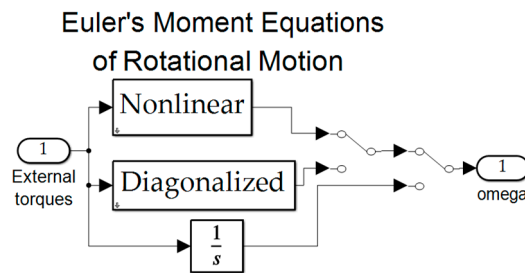


Figure B.36. Internal content of “Kinetics” subsystem depicted in **Figure 3**. The upper two subsystems are elaborate in **Figure B.37**, while the final (lowest) kinetic system is the simple integrator depicted on the bottom of this figure.

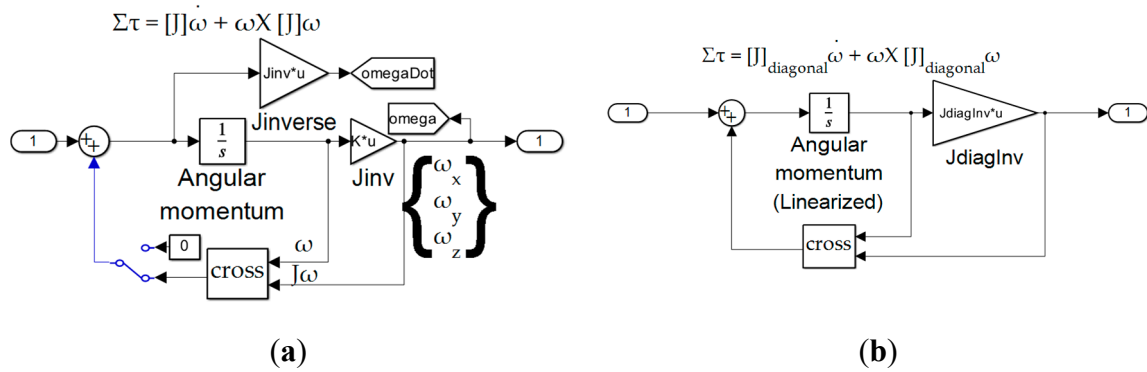


Figure B.37. (a) Internal content of "Nonlinear" kinetics subsystem depicted in Figure B.36. (b) Internal content of "Diagonalized" kinetics subsystem depicted in Figure B.36.

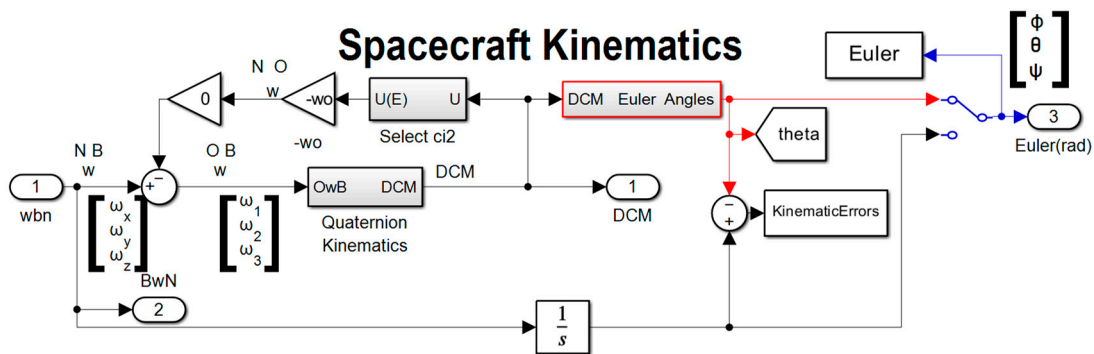


Figure B.38. Internal content of "Kinematics" subsystem depicted in Figure 3.

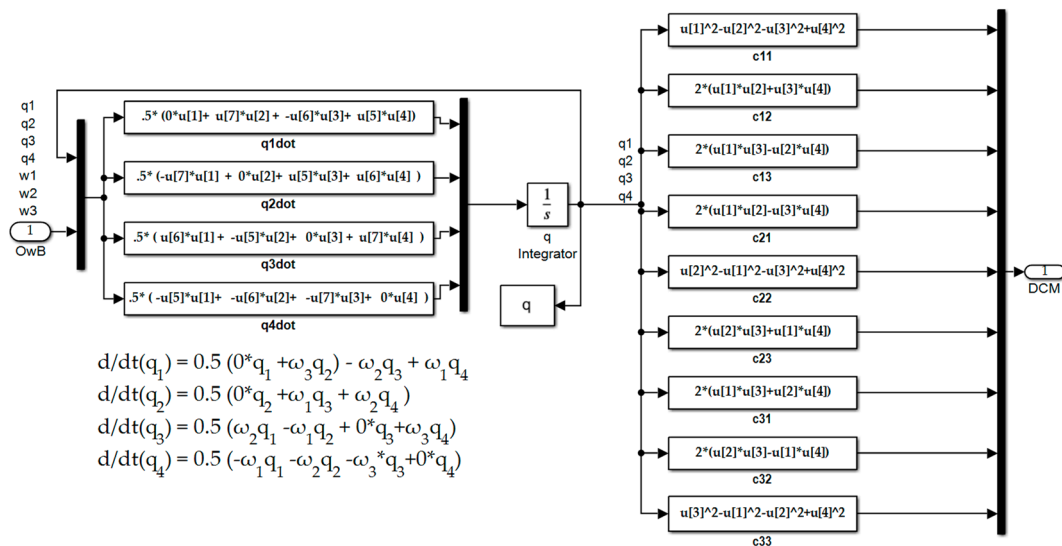


Figure B.39. Internal content of "Quaternion Kinematics" subsystem depicted in Figure B.38.

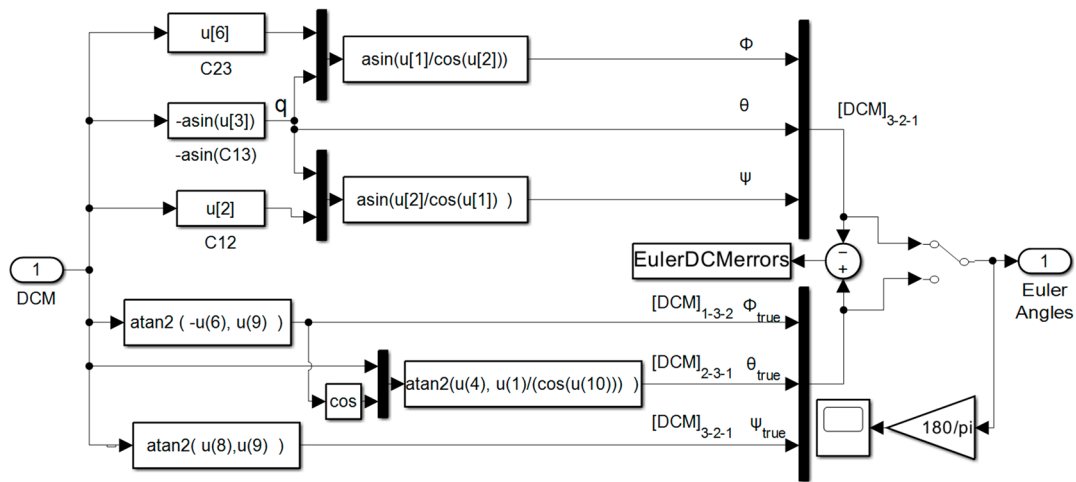


Figure B.40. Internal content of “DCM Euler Angles” subsystem depicted in Figure B.38.

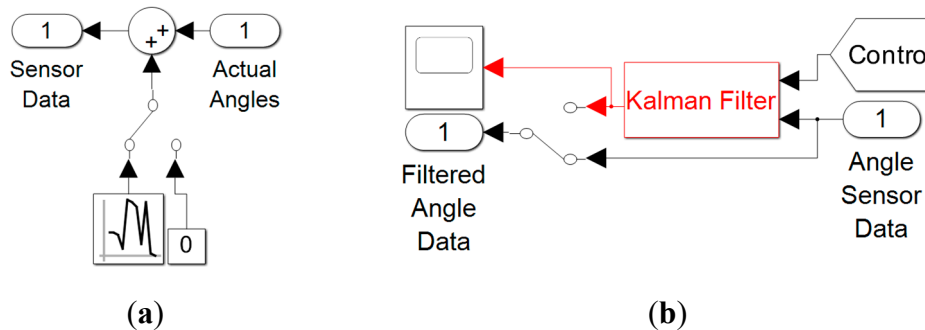


Figure B.41. Internal content of “Filters” subsystem and “Sensors” subsystems depicted in Figure 3 illustrated respectively in this figure’s (a) and (b).

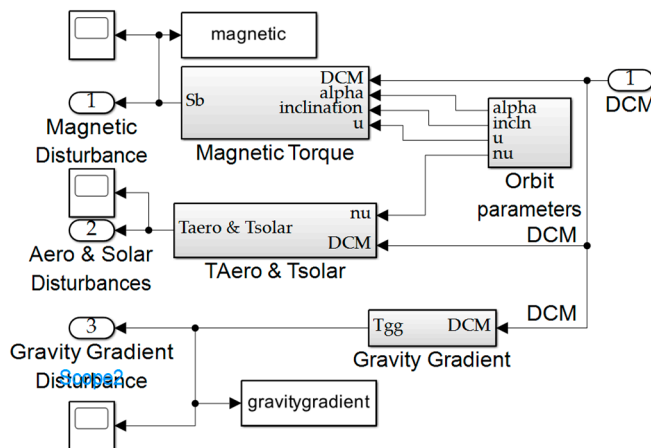


Figure B.42. High-level topology of environmental disturbances.

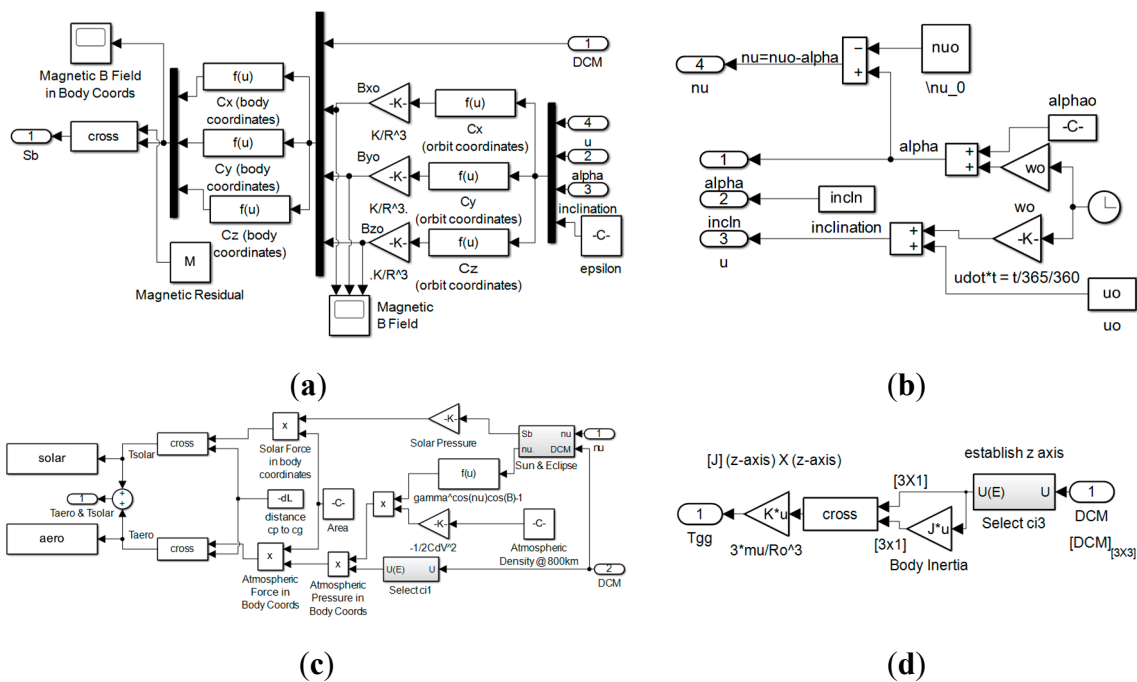


Figure B.43. Internal content of “Space Environment Disturbances” depicted in Figure B.42 illustrated respectively in this figure’s (a), (b), (c), and (d). (a) subsystem labeled “Magnetic Torque” subsystem; (b) orbit parameters; (c) TAero & Tsolar; (d) Gravity Gradient.

References

- Henshaw, C.; Glassner, S.; Naasz, B.; Roberts, B. Grappling Spacecraft. *Ann. Rev. Con., Rob., Auto. Sys.* **2022**, *5*, 137-159.
- JPL Robotics. Available online: <https://www-robotics.jpl.nasa.gov/what-we-do/research-tasks/prototype-grapple-arm-for-space-exploration-vehicle/> (accessed on 18 November 2024).
- PickNik awarded NASA SBIR Phase 2 for in-orbit robot autonomy. Available online: <https://www.aerospacemanufacturinganddesign.com/news/picnik-robotics-awarded-nasa-sbir-phase-2-robot-autonomy/> (accessed on 18 November 2024).
- Robotic Refueling Mission 3 (RRM3). Available online: <https://www.nasa.gov/nexis/robotic-refueling-mission-3/> (accessed on 18 November 2024).
- NASA Images and Media Usage Guidelines. NASA content – images, audio, video, and media files used in the rendition of 3-dimensional models, such as texture maps and polygon data in any format – generally are not subject to copyright in the United States. Available online: <https://www.nasa.gov/nasa-brand-center/images-and-media/> (accessed on 18 November 2024).
- Maclay, T.; Goff, J.; Sheehan, J.; Han, E. The development of commercially viable ADR services: Introduction of a small-satellite grapple interface. *Journal of Space Safety Engineering* **2020**, (7)3, 364–368.
- IEEE Robotics & Automation Society, Technical Committee for Space Robotics. Available online: <https://www.ieee-ras.org/space-robotics> (accessed on 18 November 2024).
- Hernández-Arias, H.; Prado-Molina, J. On-Orbit Center Of Mass Relocation System For A 3U Cubesat. *Int. J. Sci. Tech. Res.* **2018**, (7)7, 44–51.
- Calaon, R.; Kiner, L.; Allard, C.; Schaub, H. Momentum management of a spacecraft equipped with a dual-gimbal electric thruster. *AAS Guidance and Control Conference*, Breckenridge, CO, Feb. 2–8 2023. Paper No. AAS-23-178
- Gracey, W. The experimental determination of the moments of inertia of airplanes by a simplified compound-pendulum method. *National Advisory Committee for Aeronautics Technical Note* **1948**, 1629.

11. Soule, H.; Miller, M. The experimental determination of the moments of inertia of airplanes. *National Advisory Committee for Aeronautics Report 467* **1934**, 1629.501–513.
12. Genta, G.; Delpreta, C. Some considerations on the experimental determination of moments of inertia. *Meccanica* **1994**, *29*, 125-141.
13. Alessia Nocerino, A.; Opromolla, R.; Fasano, G.; Grassi, M.; Balaguer, P.; John, S.; Cho, H.; Bevilacqua, R. Experimental validation of inertia parameters and attitude estimation of uncooperative space targets using solid state LIDAR. *Acta Astronautica* **2023**, *240*, 428-436.
14. Bourabah, D.; Field, L.; Botta, E. Estimation of uncooperative space debris inertial parameters after tether capture. *Acta Astronautica* **2023**, *202*, 909-926.
15. Field, L.; Bourabah, D.; Botta, E. Online Control and Moment of Inertia Estimation of Tethered Debris. In Proceedings of the AIAA SCITECH 2024 Forum, Orlando, USA, 8–12 January 2024.
16. Sonobe, M.; Inoue, Y. Center of Mass Estimation Using a Force Platform and Inertial Sensors for Balance Evaluation in Quiet Standing. *Sensors* **2023**, *23*(10), 4933.
17. Zatsiorsky, V.M.; King, D.L. An algorithm for determining gravity line location from posturographic recordings. *J. Biomech.* **1997**, *31*, 161–164
18. Caron, O.; Faure, B.; Brenière, Y. Estimating the centre of gravity of the body on the basis of the centre of pressure in standing posture. *J. Biomech.* **1997**, *30*, 1169–1171.
19. Germanotta, M.; Mileti, I.; Conforti, I.; Del Prete, Z.; Aprile, I.; Palermo, E. Estimation of Human Center of Mass Position through the Inertial Sensors-Based Methods in Postural Tasks: An Accuracy Evaluation. *Sensors* **2021**, *21*(2), 601.
20. Wang, F.; Bettadpur, S.; Save, H. Determination of the center-of-mass of gravity recovery and climate experimental satellites. *J. Space. Roc.* **2010**, *47*(2), 371.
21. Kornfeld, R.; Arnold, B.; Gross, M.; Dahya, N.; Klipstein, W.; Gath, P.; Bettadpur, S. GRACE-FO: The Gravity Recovery and Climate Experiment Follow-On Mission. *J. Space. Roc.* **2019**, *56*(3).
22. Huang, Z.; Li, S.; Cai, L.; Fan, D.; Huang, L. Estimation of the Center of Mass of GRACE-Type Gravity Satellites. *Remote Sens.* **2022**, *14*(16), 4030.
23. Pan, Z.; Xiao, Y. Data Quality Assessment of Gravity Recovery and Climate Experiment Follow-On Accelerometer. *Sensors* **2024**, *24*(13), 4286.
24. Dou, Y.; Wang, K.; Zhou, Z.; Thomas, P.R.; Shao, Z.; Du, W. Investigation of the Free-Fall Dynamic Behavior of a Rectangular Wing with Variable Center of Mass Location and Variable Moment of Inertia. *Aerospace* **2023**, *10*(5), 458.
25. Geng, J.; Langelan, J. Junyi Geng and Jack W. Langelan. Estimation of Inertial Properties for a Multilift Slung Load. *J. Guid., Con., Dyn.* **2021** (44)2, 220-237.
26. Gahramanova, A. Locating Centers of Mass with Image Processing. *Und. J. Math. Mod.* **2019**, (10)1.
27. Lin, Y.T.; Tian, Y.; Perez, D.; Livescu, D. Regression-Based Projection for Learning Mori-Zwanzig Operators. *SIAM J. App. Dyn. Sys.* **2023**. (22)4.
28. Setterfield, T.; Miller, D.; Leonard, J.; Saenz-Otero A. Mapping and determining the center of mass of a rotating object using a moving observer. *Inter. J. Rob. Res.* **2018**, *37*(1), 83-103.
29. Diagonalize the Inertia Tensor. Available online: [https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Variational_Principles_in_Classical_Mechanics_\(Cline\)/13%3A_Rigid-body_Rotation/13.07%3A_Diagonalize_the_Inertia_Tensor](https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Variational_Principles_in_Classical_Mechanics_(Cline)/13%3A_Rigid-body_Rotation/13.07%3A_Diagonalize_the_Inertia_Tensor) (accessed on 26 November 2024).
30. Luenberger Observer. Available online: <https://www.mathworks.com/help/sps/ref/luenbergerobserver.html> (accessed on 19 November 2024).
31. Transport theorem. Available online: https://en.wikipedia.org/wiki/Transport_theorem (accessed on 19 November 2024).
32. Fossen, T. Comments on 'Hamiltonian adaptive control of spacecraft'. *IEEE Transactions on Automatic Control* **1993**, *38*(4), 671–672.
33. Iven, M.; Mareels, B.; Anderson, R.; Bitmead, R.; Bodson, M.; Sastry, S. Revisiting the MIT rule for adaptive control. *IFAC Adaptive Systems in Control and Signal Processing* **1987**, *20*(2), 161–166.

34. Adams, G. Arizona State University Physics Department Proof of the Parallel Axis Theorem. Available online: <https://www.public.asu.edu/~gbadams/sum00/parallelaxisT.pdf> (accessed on 19 November 2024).
35. Kuck, E.; Sands, T. Space Robot Sensor Noise Amelioration Using Trajectory Shaping. *Sensors* **2024**, *24*(2), 666.
36. Zhu, Y. Identification Test Design and Data Pretreatment. In *Multivariable System Identification for Process Control*; Zhu, Y. (Eds); Elsevier Science: Maryland Heights, USA, **2001**; pp. 31–63.
37. Persistent excitation conditions and their implications. Section 8.1 of *Study Guide on Adaptive and Self-Tuning Control*. Available online: <https://library.fiveable.me/adaptive-and-self-tuning-control/unit-8/robustness-issues-adaptive-control/study-guide/yLKsaWxBlamk2I48> (accessed on 22 November 2024).
38. Marino, R.; Tomei, P. On exponentially convergent parameter estimation with lack of persistency of excitation. *Systems & Control Letters* **2022**, *159*, 105080.
39. Korotina, M.; Romero, J.; Aranovskiy, S.; Bobtsov, A.; Ortega, R. A new on-line exponential parameter estimator without persistent excitation. *Systems and Control Letters* **2022**, *159*, 105079.
40. Software, Robotics, and Simulation Division. Available online: <https://www.nasa.gov/software-robotics-and-simulation-division/> (accessed on 21 November 2024).
41. Dynamics Operations Training Systems. Available online: <https://www.nasa.gov/general/generic-robotics-on-orbit-trainer/> (accessed on 21 November 2024).
42. Dai, Q.; Dai, Q.; Xiao, G.; Zhou, G.; Ye, Q.; Han, S.; Han, S.; A novel Gaussian sum quaternion constrained cubature Kalman filter algorithm for GNSS/SINS integrated attitude determination and positioning system. *Frontiers in Neurorobotics* **2024**, *18*, 1374531.
43. Slotine, J.; Di Benedetto, M. Hamiltonian adaptive control of spacecraft. *IEEE Transactions on Automatic Control* **1990**, *35*(7), 848–852.
44. Fossen, T. Comments on ‘Hamiltonian adaptive control of spacecraft’. *IEEE Transactions on Automatic Control* **1993**, *38*(4), 671–672.
45. Sands, T.; Kim, J.; Agrawal, B. Improved Hamiltonian Adaptive Control of spacecraft. In Proceedings of the IEEE Aerospace conference, Big Sky, MT, USA, 7–14 March 2009.
46. Sands, T.; Kim, J.; Agrawal, B. Spacecraft fine tracking pointing using adaptive control. In Proceedings of the 58th International Astronautical Congress, Hyderabad, India, 24–28 September 2007.
47. Sands, T.; Kim, J.; Agrawal, B. Spacecraft adaptive control evaluation. In Proceedings of Infotech@Aerospace, Garden Grove, California, USA, 19–21 June 2012.
48. Huang, B.; Sands, T. Novel learning for control of nonlinear spacecraft dynamics. *J. Applied Math* **2023**, *1*(1), 42.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.