

Article

Proposals for Surmounting Sensor Noises

Andre Pittella¹ and Timothy Sands^{2,*}¹ Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14850 USA² Department of Mechanical Engineering (CVN), Stanford University, Stanford, CA 94305, USA

* Correspondence: dr.timsands@alumni.stanford.edu

Abstract: Classical and optimal control architectures for motion mechanics with fusion of noisy sensors use different algorithms and calculations to perform and control any number of physical demands, to varying degrees of accuracy, precision, and cost. Their performances are tested for the purpose of comparison through the means of a Monte Carlo simulation that simulates how different parameters might vary under noise, representing real-world imperfect sensors. We find that improvements in one figure of merit often come at a cost in the performance in the others, especially depending on the presence of noise in the system sensors. If sensor noise is negligible, open-loop optimal control performs the best. However, in the overpowering presence of sensor noise, using a control law inversion patching filter performs as the best replacement, but has significant computational strain.

Keywords: sensor fusion; sensor noise; optimization; feedback; real-time optimization; velocity-based controller

1. Introduction

Artemis I will be the first integrated flight test of NASA's deep space exploration system: the Orion spacecraft, Space Launch System (SLS) rocket and the ground systems at Kennedy Space Center in Cape Canaveral, Florida. The first in a series of increasingly complex missions, Artemis I will be an uncrewed flight that will provide a foundation for human deep space exploration and demonstrate our commitment and capability to extend human existence to the Moon and beyond. During this flight, the uncrewed Orion spacecraft will launch on the most powerful rocket in the world and travel thousands of miles beyond the Moon, farther than any spacecraft built for humans has ever flown, over the course of about a three-week mission. [1]



Figure 1. Lunar lander concept sought for the Artemis program, image courtesy NASA [2], image use in accordance with NASA image use policy: [3].

Dealing with the fusion of potentially poor, noisy sensors is a ubiquitous challenge that has a long lineage leading to several disparate approaches. Some researchers focus on improving the sensor itself or its signal using internal algorithms. Others seek to develop architectures that prove robust to fused noisy sensor data. Wang, et al. proposed a novel method for accurate, autonomous and real-time orbit determinations for geo-referencing with a standalone global positioning system receiver [4]. Xiong, et al. addressed systematic centroid errors and poor attitude accuracy by augmenting star trackers with an image intensifier [5]. Kim, et al. proposed an algorithm for determining the orbit of a geostationary satellite using single-epoch measurements from a global positioning system receiver with sparse visibility of the global positioning system satellites [6]. Takayama illustrated weaknesses with global navigation system signals and proposed novel sensor noise models used to enhance sensor sensitivity [7].

Leake, et al. proposed dealing with sensor issues with improved sensor algorithms, proposing a non-dimensional star identification algorithm compared in terms of accuracy, speed, and robustness to the so-called pyramid algorithm. [8] Marin, et al. sought to enhance star trackers by sensor and sensor fusion algorithms to provide a smoother and faster output. [9] Perov, et al. sought to utilize the principle of phase interferometer, using multiple receiving antennas [10]. Wang, et al. sought to integrate communications navigation with global positioning system sensors. [11] Christian proposed autonomous augmentation using optical navigation by relativistic perturbation of starlight [12]. Fan, et al. investigated a plume noise suppression algorithm based on star point shape and the angular distance between stars [13]. Opromolla, et al. proposed dealing with sensor issues algorithmically by using a model-based three-dimensional template matching technique for pose acquisition of an uncooperative space object [14].

Rather than dealing with sensor issue algorithmically as proposed by Opromolla, Chen, et al. instead focused on the control algorithm using the sensor measurements and proposed a velocity-based impedance control scheme illustrating efficacy in both speed and robustness [15]. This manuscript parallels the insights of Chen, et al. and enhances the velocity-based logic with optimization methods of Pontryagin akin Sandberg's recent recitation [16]. Sandberg's very recent improvements follow a lineage of small improvements from its provenance in the nonlinear adaptive control methods of the 1990's as proposed by Slotine [17] and afterward improved by Fossen [18], Sands [19,20] who offered experimental validation in [21]. Most recently Raigoza [22] who augmented the method with autonomous collision avoidance, and Wilt [23] who evaluated efficacy in the face of uniform variations in mass moment of inertia (e.g., from fuel slosh). This manuscript proposes and illustrates methods that assume velocity-based control logic like [Chen] but seek to induce open-loop optimal results like [Sandberg, Raigoza, and Wilt] and thus mitigate deleterious effects of noisy sensors.

Controllers are differential equations that are designed to be able to manipulate some state variable by arbitrarily changing a variable connected to it through a physical law or equation. They often feature several points of adjustment and tunability to fit figures of merit desired by the designer. The figures of merit utilized here include spacecraft tracking accuracy (how well the controller reaches the desired target), precision (as measured by deviation), and cost (numerical evaluation of the resources used). Computational burden is also monitored since difficulties such as singular matrix inversions are involved. A variety of control system architectures were evaluated, and their ability is assessed to perform the task of a rest-to-rest reorientation normalized to unity.

1.1. Novelties presented

Numbered lists can be added as follows:

1. Open loop optimal results are analytically calculated providing a performance benchmark for comparing other methods.
2. Inspired by Opromolla, 2015 [10], velocity-based classical control is investigated, especially since it was utilized as the comparative benchmark by Sandberg 2022

- [16], Raigoza 2022 [22], and Wilt 2022 [23], while the open-loop optimal results in item #1 are used as comparative benchmark here.
3. Real-time optimal control is compared since it was the latest cited proposals of 2022.
 4. Double-integrator patching filters are implemented seeking to match open loop optimal results amidst fusion of noisy sensors.
 5. System-inverting patching filters are implemented also seeking to match open loop optimal results amidst fusion of noisy sensors.

2. Control System Architectures

2.1 The task at hand

The controllers will perform a rest-to-rest reorientation of one unit of rotation (indicated by variable θ whose rate is indicated by variable ω) scaled to unity over unit of time scaled to unity. The equations for such a maneuver are listed in equations (1) – (3). Controller cost will be used as a key figure of merit, where a quadratic cost computation will be used indicated by variable J in equation (4).

$$\theta(0) = \omega(0) = 0 \quad (1)$$

$$\theta(1) = \theta_d = 1 \quad (2)$$

$$\omega(1) = 0 \quad (3)$$

$$J = \frac{1}{2} \int_0^1 u(t)^2 dt \quad (4)$$

with state $\theta(t)$, rate $\omega(t)$, desired state θ_d , quadratic cost functional $J(t)$, and control variable $u(t)$. The state, rate, and control are connected in accordance with equations (5) and (6).

$$\omega = \dot{\theta} \quad (5)$$

$$u = \frac{1}{I} \dot{\omega} \quad (6)$$

for moment of inertia I , where dotted variables indicate derivatives in time.

2.2. Proportional plus velocity (P+V) control

The P+V controller is a form of classical control, tuned using two different gains: a proportional gain K_p applied to the state error, added to a velocity gain K_v applied to the rate, *not the rate error* (see (14)). P+V controllers can be easily tuned to fit desired settling times and damping ratios. For a maximum tolerance of 2% error at the desired settling time, equation (7).

$$t_s = -\frac{\ln(0.02 * \sqrt{1 - \zeta^2})}{\zeta \omega_n} \quad (7)$$

can be used, where ζ is the damping ratio of the controller and ω_n is its natural frequency response. For a P+V controller, these are known to be equations (8) and (9).

$$\omega_n = \sqrt{K_p} \quad (8)$$

and

$$\zeta = \frac{K_v}{2\sqrt{K_p}} \quad (9)$$

For a settling time of ≤ 1 second and a damping ratio of 0.7, the gains are calculated to be $K_p = 37$ and $K_v = 8.5$.

P+V controllers are always able to reach the desired value, but it is the selection of gains that affects how quickly it is reached. This ability is hampered by noise, especially on the rate sensor. Because the end rate is 0, any error in rate detection will cause a control to move the system off of the desired value. Additionally, the settling time is only for the state target, so no estimation of the rate target settling time is used to set the gains.

2.3. Open-loop Optimal Control

For a given problem with dynamics, boundary conditions, and a cost functional, an optimal solution exists that minimizes the output of the cost functional, which is found using Pontryagin's method [24]. Using equations (1)-(6), the task described in *the task at hand* (from section 2.1) has an optimal control u^* of the form in equation (10).

$$u^*(t) = \theta_d(at + b) \quad (10)$$

By integrating twice and applying the initial and final boundary conditions, the linear system in equation (11) is found.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1/6 & 1/2 & 1 & 1 \\ 1/2 & 1 & 1 & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \theta_d \\ 0 \end{Bmatrix} \quad (11)$$

where the solution to equation (11) can be applied to (10) to yield equation (12).

$$u^*(t) = \theta_d(-12t + 6) \quad (12)$$

which has a minimal quadratic cost of $J^* = 6$ and exact achievement of the target end conditions. While open-loop optimal control yields the perfect results on paper, it is completely blind to noise and perturbation.

2.4. Real-time Optimal Control

Real-time optimal control, or RTOC, is a modification of the previous iteration of optimal control that considers the current state of the system, allowing the controller to adjust for noise and perturbations. The top two rows of the matrix in (11) are the equations for $\theta^*(t)$ and $\omega^*(t)$, the optimal forms of the state and rate variables, evaluated at the initial conditions. By modifying these rows such that the forms are evaluated at the current time t_0 , the linear system changes to:

$$\begin{bmatrix} t_0^3/6 & t_0^2/2 & t_0 & 1 \\ t_0^2/2 & t_0 & 1 & 0 \\ 1/6 & 1/2 & 1 & 1 \\ 1/2 & 1 & 1 & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} \theta(t_0) \\ \omega(t_0) \\ \theta_d \\ 0 \end{Bmatrix} \quad (13)$$

The vector of unknowns on the left-hand side is calculated by inverting the matrix and multiplying it by the vector of knowns on the right-hand side. The unknowns a and b are then used in (10) to make an instantaneous optimal control, calculated for the current state of the system. Therefore, the controller can adjust its control output in the presence of noise and perturbations.

While being able to change course based on the current state of the system is an advantage over open-loop optimal control, RTOC has a large computational demand in its inversion of a 4x4 matrix. Additionally, the matrix becomes singular as $t_0 \rightarrow 1$, which causes problems as singular matrices are uninvertible. In simulation, the effect of this is the control shooting off to infinity towards the end of the runtime, leading to huge error in the state and/or rate. To avoid this behavior, a switch is worked in so that, when the matrix determinant approaches 0, the controller switches to open-loop optimal control for the remainder of the runtime.

2.5. Patching Filter: Double Integrator

One way to combine the analytical accuracy of optimal control and the computational simplicity & tunability of classical P+V control is to impose a 'patching' filter that integrates the optimal control $u^*(t)$ twice to find the optimal state function $\theta^*(t)$, which is given to a P+V controller as the desired value instead of a constant command θ_d .

2.6. Patching Filter: Double Integrator, Tuned

However, the result of the previous architecture may not be desirable. Should the gains of the original P+V controller be modifiable, tuning them around the patching filter is a viable strategy. This architecture is not as well-studied as the original P+V control, so handy equations based around desired settling time and damping ratio are not available. Therefore, the gains will be tuned by hand, i.e., trial and error, until the result meets acceptable figures of merit.

2.7. Patching Filter: Control Law Inversion

Should the gains not be modifiable, there is another way to adjust the architecture. Instead of directly integrating the optimal control, the control law for the P+V controller can be solved for an optimal input such that, after all the gains are applied and the control calculated, the P+V controller output matches that of the optimal control $u^*(t)$, where the asterisk '*' indicates optimality. The P+V control law is listed in equation (14). Solving for the desired state results in equation (15).

$$u = K_P(\theta_d - \theta) - K_V\omega \quad (14)$$

$$\theta_d = \frac{1}{K_P}u + \frac{K_V}{K_P}\omega + \theta \quad (15)$$

Entering the optimal values u^* , ω^* , and θ^* gives the optimal input. For a patching filter that works in the Laplace domain as a transfer function, this can be rewritten as equation (16) which can be simplified to become equation (17).

$$\Theta_d(s) = U^*(s) \left(\frac{1}{K_P} + \frac{K_V}{K_P s} + \frac{1}{s^2} \right) \quad (16)$$

$$\frac{\Theta_d(s)}{U^*(s)} = \left(\frac{s^2 + K_V s + K_P}{K_P s^2} \right) \quad (17)$$

Equation (17) is the patching filter to be applied to the optimal control yielding the optimal state trajectory input for the P+V controller.

3. Results and Analysis

The following simulations were performed in MATLAB Simulink, using the ode4 (Runge-Kutta) integrator and a 0.01s step size. Using a random number generator with a Gaussian distribution, a noise signal was added to the state and rate with variances of 0.0001 with zero mean. The moment of inertia was also varied randomly but uniformly to $\pm 10\%$ of its true value.

3.1. Monte Carlo Simulation

Displayed in Figure 2 are the results of 1000 simulations of each of the control architectures outlined in the previous section. Table 1 contains the numerical performances of each architecture for direct comparison.

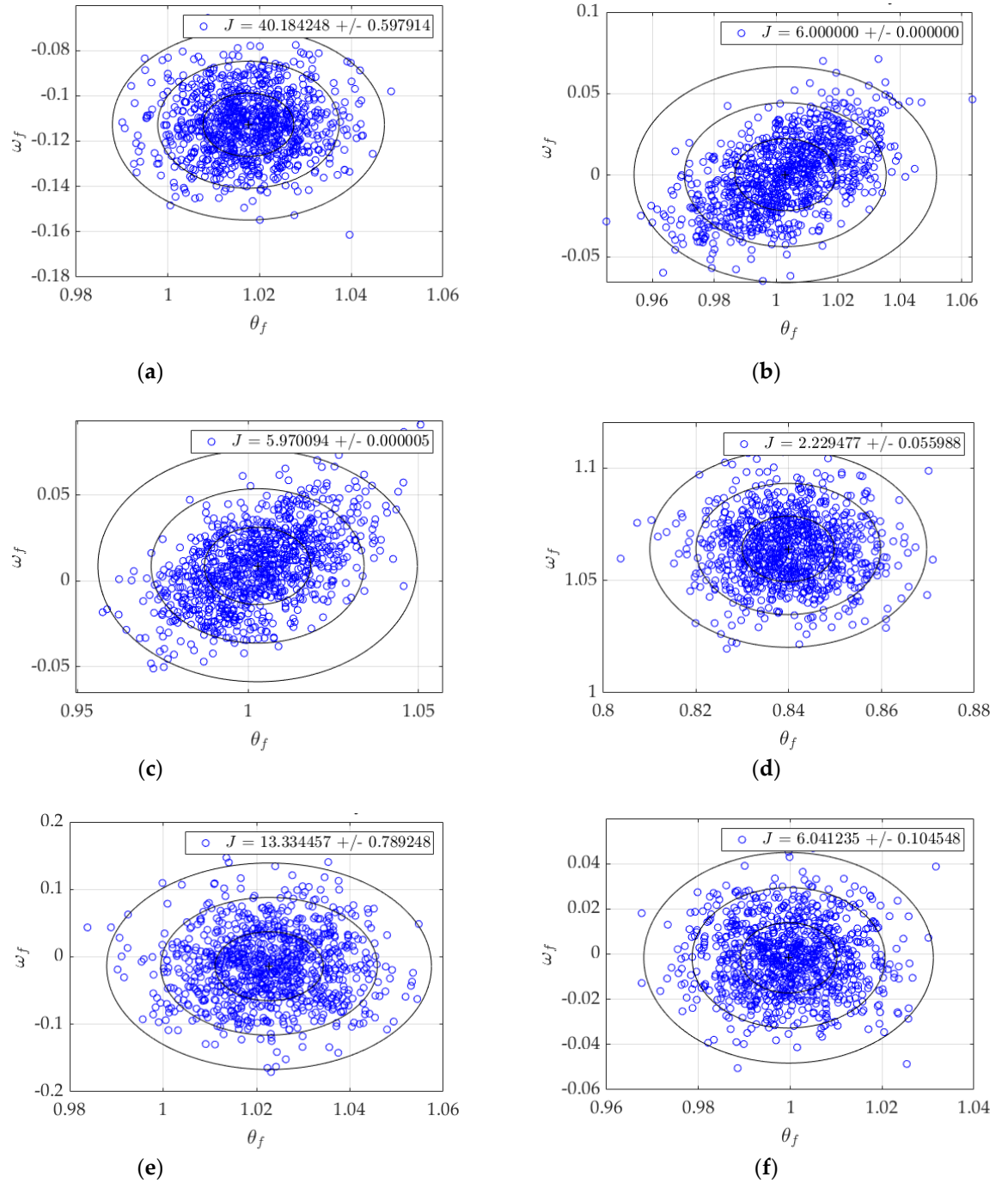


Figure 2. Monte Carlo plots, $N = 1000$. Rings show 1-sigma, 2-sigma, and 3-sigma deviations from the mean, denoted by a +. (a) P+V control; (b) open-loop optimal control; (c) RTOC; (d) double integrator patching filter; (e) double integrator, tuned; (f) control law inversion patching filter.

From Table 1, the relationships between accuracy (the final values of θ & ω), precision (the magnitude of their spread, seen in σ), and cost (J) can be evaluated. The P+V controller, while having the smallest standard deviations for both values, has notable error in ω_f and the highest quadratic cost, 301% the cost of the next highest cost. Open-loop optimal has some of the most accurate mean values, but the second-highest spread in both state and rate. RTOC is similar, with lower cost (-0.5%) and rate spread (-29%) but higher state spread (38%) and final rate error (218%). The double integrator patching filter has the second-lowest spreads in both state and rate and the lowest cost, but the final values are completely off the mark; 16% error in state and 943% more rate error than the next highest architecture. After some tuning, it performs similarly to the P+V controller but with higher spreads (17.2% and 264% respectively), lower cost (-66.8%), and a more

accurate final rate (12% the error of the P+V). Finally, the control law inversion patching filter strikes a balance between all figures of merit: lowest state error, third lowest state spread, second lowest rate error, third lowest state spread, and a cost only marginally higher (0.7%) than the optimal $J^* = 6$, where the asterisk “*” indicates optimality. However, it took the longest to run after RTOC.

Table 1. Numerical results of Monte Carlo simulations, $N = 1000$. σ refers to the standard deviation of a quantity. All other numerical results represent the mean over all simulations.

Architecture	θ_f	σ_θ	ω_f	σ_ω	J	σ_J
P+V ¹	1.0176	0.0099	-0.1128	0.0140	40.1842	0.5979
Open-loop optimal	1.0029	0.0163	0.0004	0.0220	6.0000	0.0000
RTOC	1.0028	0.0225	0.0087	0.0156	5.9701	0.00001
Double Integrator ¹	0.8399	0.0100	1.0640	0.0146	2.2295	0.0560
Double Int., Tuned ²	1.0227	0.0116	-0.0140	0.0510	13.3345	0.7892
Control Inversion ¹	0.9996	0.0105	-0.0016	0.0155	6.0412	0.1045
P+V ¹	1.0176	0.0099	-0.1128	0.0140	40.1842	0.5979

¹ $K_p = 37, K_V = 8.5$

² $K_p = 78, K_V = 0.65$

Table 2. Numerical results of Monte Carlo simulations, $N = 1000$. σ refers to the standard deviation of a quantity. All other numerical results represent the mean over all simulations. Percent differences are compared to the benchmark open loop optimal results.

Architecture	θ_f	σ_θ	ω_f	σ_ω	Cost	σ_J
P+V ¹	1%	-39%	-28300%	-36%	570%	60%
Open-loop optimal	--	--	--	--	--	--
RTOC	0%	38%	2075%	-29%	0%	0%
Double Integrator ¹	-16%	-39%	265900%	-34%	-63%	6%
Double Int., Tuned ²	2%	-29%	-3600%	132%	122%	79%
Control Inversion¹	0%	-36%	-500%	-30%	1%	10%

¹ $K_p = 37, K_V = 8.5$

² $K_p = 78, K_V = 0.65$

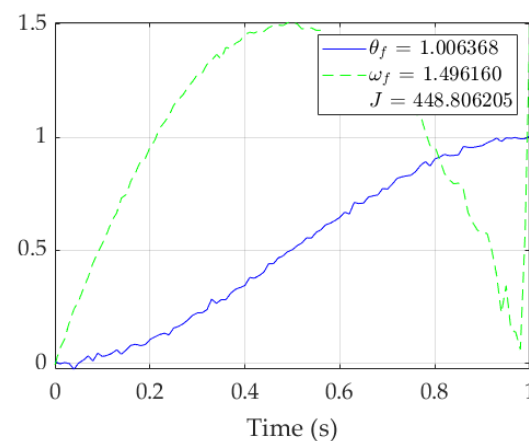


Figure 3. Singular run of RTOC using $\text{pinv}(A)$ without open loop switching. Note the deterioration in $\omega(t)$ starting after time = 0.8 seconds, indicating the function’s inability to reliably solve the system as it approaches singularity.

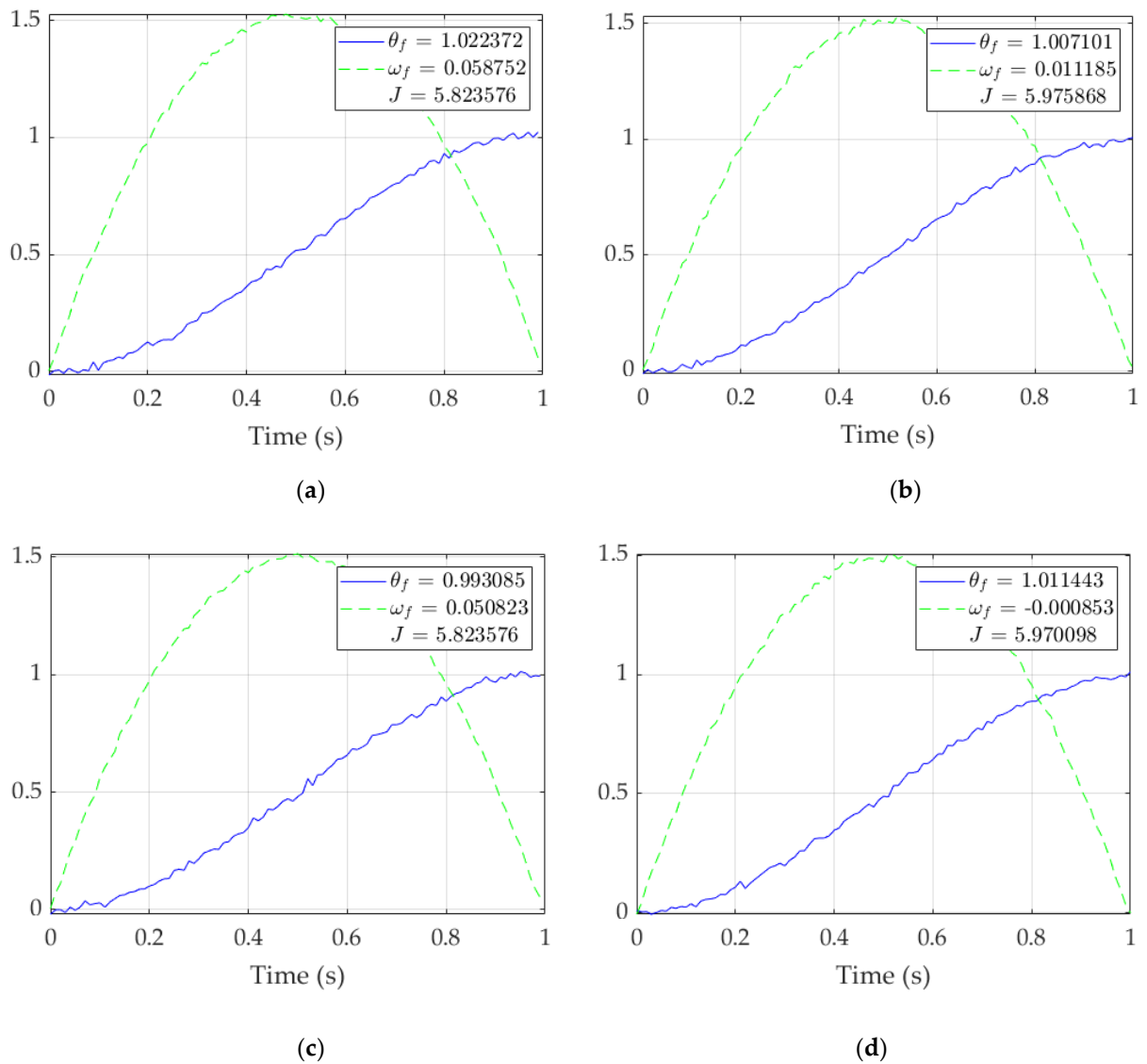


Figure 4. Singular runs of RTOC for each matrix inverse procedure. **Solid blue line:** $\theta(t)$; **dashed green line:** $\omega(t)$. (a) $(A)^{-1}$; (b) $1 \setminus A$; (c) $\text{inv}(A)$; (d) $\text{pinv}(A)$. Note the failure of plots (a) and (c) to reach the right side, where time = 1 second.

4. Conclusions

From the results shown in Section 3 summarized in table 2, there is no one control that performs the best in every single regard. The P+V controller has high cost and rate error, the optimal controls have low cost and high accuracy but higher susceptibility to sensor noise, while the double integrator patching filter is difficult to tune and costly. The control law inversion patching filter, however, can be seen as the best alternative to open-loop optimal control especially in the presence of sensor noise and inertia uncertainties, as it had a cost only slightly higher (0.7%) with smaller standard deviations (35.6% less spread in state, 29.5% less spread in rate) and final results more accurate than any other architecture (14.3% the state error of the runner up, 18.4% the rate error of the runner up). The key tradeoff was computational burden since control inversion optimization took 25-40% longer to run than some of the other architectures.

References

1. Hambleton, K. Artemis I Map, February 9, 2018. Available online: <https://www.nasa.gov/image-feature/artemis-i-map> (accessed 12 November 2022)
2. Mahoney, E. Fast-Track to the Moon: NASA Opens Call for Artemis Lunar Landers, Sep 30, 2019. Available online: <https://www.nasa.gov/feature/fast-track-to-the-moon-nasa-opens-call-for-artemis-lunar-landers/> (accessed 12 November 2022)

3. Media Usage Guidelines. Available online: <https://www.nasa.gov/multimedia/guidelines/index.html> (accessed on 12 November 2022).
4. Wang, F.; Gong, X.; Sang, J.; Zhang, X. A Novel Method for Precise Onboard Real-Time Orbit Determination with a Standalone GPS Receiver. *Sensors* **2015**, *15*, 30403–30418.
5. Xiong, K.; Jiang, J. Reducing Systematic Centroid Errors Induced by Fiber Optic Faceplates in Intensified High-Accuracy Star Trackers. *Sensors* **2015**, *15*, 12389–12409.
6. Kim, G.; Kim, C.; Kee, C. Coarse Initial Orbit Determination for a Geostationary Satellite Using Single-Epoch GPS Measurements. *Sensors* **2015**, *15*, 7878–7897.
7. Takayama, Y.; Urakubo, T.; Tamaki, H. Novel Process Noise Model for GNSS Kalman Filter Based on Sensitivity Analysis of Covariance with Poor Satellite Geometry. *Sensors* **2021**, *21*, 6056.
8. Leake, C.; Armas, D.; Mortari, D. Non-Dimensional Star-Identification. *Sensors* **2020**, *20*, 2697.
9. Marin, M.; Bang, H. Design and Simulation of a High-Speed Star Tracker for Direct Optical Feedback Control in ADCS. *Sensors* **2020**, *20*, 2388.
10. Perov, A.; Shatilov, A. Deeply Integrated GNSS/Gyro Attitude Determination System. *Sensors* **2020**, *20*, 2203.
11. Wang, L.; Lü, Z.; Tang, X.; Zhang, K.; Wang, F. LEO-Augmented GNSS Based on Communication Navigation Integrated Signal. *Sensors* **2019**, *19*, 4700.
12. Christian, J.A. StarNAV: Autonomous Optical Navigation of a Spacecraft by the Relativistic Perturbation of Starlight. *Sensors* **2019**, *19*, 4064.
13. Fan, Q.; Cai, Z.; Wang, G. Plume Noise Suppression Algorithm for Missile-Borne Star Sensor Based on Star Point Shape and Angular Distance between Stars. *Sensors* **2019**, *19*, 3838.
14. Opromolla, R.; Fasano, G.; Rufino, G.; Grassi, M. A Model-Based 3D Template Matching Technique for Pose Acquisition of an Uncooperative Space Object. *Sensors* **2015**, *15*, 6360–6382.
15. Chen, C.; Nie, H.; Chen, J.; Wang, X. A Velocity-Based Impedance Control System for a Low Impact Docking Mechanism (LIDM). *Sensors* **2014**, *14*, 22998–23016.
16. Sandberg, A.; Sands, T. Autonomous Trajectory Generation Algorithms for Spacecraft Slew Maneuvers. *Aerospace* **2022**, *9*, 135.
17. Slotine, J.-J.E.; Weiping, L. Applied Nonlinear Control; Prentice-Hall: Hoboken, NJ, USA, 1991.
18. Fossen, T. Comments on Hamiltonian adaptive control of spacecraft by Slotine, J.J.E. and Di Benedetto, M.D. *IEEE Trans. Autom. Control*. **1993**, *38*, 671–672.
19. 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; International Astronautical Federation: Paris, France, 2007.
20. Sands, T.; Lorenz, R. Physics-Based Automated Control of Spacecraft. In Proceedings of the AIAA Space Conference & Exposition, Pasadena, CA, USA, 14–17 September 2009.
21. Sands, T.; Kim, J.J.; Agrawal, B.N. Spacecraft Adaptive Control Evaluation. In Proceedings of the Infotech@ Aerospace, Garden Grove, CA, USA, 19–21 June 2012; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2012; pp. 2012–2476.
22. Raigoza, K.; Sands, T. Autonomous Trajectory Generation Comparison for De-Orbiting with Multiple Collision Avoidance. *Sensors* **2022**, *22*, 7066.
23. Wilt, E.; Sands, T. Microsatellite Uncertainty Control Using Deterministic Artificial Intelligence. *Sensors* **2022**, *22*, 8723.
24. Ross, I. M. (2015). A primer on Pontryagin's principle in optimal control. Collegiate Publisher.