

## Article

# Autonomous Vehicle Control Comparison

Pruthvi Banginwar <sup>1</sup>  and Timothy Sands <sup>2,\*</sup> <sup>1</sup> Department of Applied Engineering Physics, Cornell University, Ithaca, NY 14853, USA<sup>2</sup> Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

\* Correspondence: tas297@cornell.edu

**Abstract:** Self-driving features rely upon autonomous control of vehicle kinetics, and this manuscript compares several disparate approaches to control predominant kinetics. Classical control using feedback of state position and velocities, open-loop optimal control, real-time optimal control, double-integrator patching filters with and without gain-tuning, and control law inversion patching filters accompanying velocity control are assessed in Simulink, and their performances are compared. Optimal controls are found via Pontryagin's method of optimization utilizing three necessary conditions: Hamiltonian minimization, adjoint equations, and terminal transversality of the endpoint Lagrangian. It is found that real-time optimal control and control-law patching filter with velocity control incorporating optimization are the two best methods overall as judged in Monte Carlo analysis by means and standard deviations of position and rate errors and cost.

**Keywords:** controller; simulink; Monte Carlo; Pontryagin; optimization; patching; filter



**Citation:** Banginwar, P.; Sands, T. Autonomous Vehicle Control Comparison. *Vehicles* **2022**, *4*, 1109–1121. <https://doi.org/10.3390/vehicles4040059>

Academic Editor: Lei Zhang

Received: 2 September 2022

Accepted: 30 September 2022

Published: 10 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Vehicle kinetics are commonly studied using predominant kinetics represented by double integrators equivalently for translation and rotation, neglecting coupling cross-products generated by the transport theorem for representing motion in the basis coordinates of rotating reference frames. Invocation of the Michel Chasles's Theorem [1] permits a full description of six-degrees-of-freedom motion by assertion of Euler's [2] and Newton's [3] equations. Controlling mechanical motion has an even longer pedigree culminating in utilization of deterministic artificial intelligence as a burgeoning field proposed by Cooper/Heidlauf in 2017 [4], expanded to space vehicles by Smeresky/Rizzo [5] and after the introduction of autonomous trajectory generation by Baker et al. [6] formalized as a process that very same year [7] and proven to be optimal. The provenance of the approach stems from nonlinear adaptive control as proposed by Slotine [8], modified by Fossen [9], and improved in 2012 with experimental validation [10], while Cooper/Heidlauf formulated the Slotine/Fossen feed-forward element as a deterministic statement of self-awareness. The physics-based methods of Lorenz [11,12] provided the formalization which amplified the importance of the first principles [1–3] to establish physics-based statements of self-awareness. In Reference [13], simple adaption techniques remained prevalent until the codification of optimal learning by Smeresky/Rizzo in 2020. One noteworthy feature is the ubiquitous applicability to such disparate fields as described so far include: spacecraft, robotics, and power electronics. Additionally, Rätze et al. [14] leverage the physics-based methods for optimal control of a CO<sub>2</sub> methanation reactor. Bukhari et al. [15] leveraged the methods to monitor and mitigate megacities air pollution. Thus, the utility of this present study is manifest. Deterministic artificial intelligence necessitates autonomously generated trajectories [16], therefore presented here is a comparison of techniques to autonomously generate optimal trajectories that minimize fuel usage. Trajectory research was evident immediately by Cooper/Heidlauf following their seminal publication, the work that immediately followed concentrated on autonomous vehicle trajectory generation [17] and substantiation of sensor-less state error tracking [18] since the optimal regulator performed so poorly. Shortly

thereafter, Cooper collaborated with Smeresky to produce the combination of their efforts to date [19]. This combined work initiated the several lines of research proposals since 2020 seeking to optimize the current instantiation of deterministic artificial intelligence.

Optimization [20] can take many forms: linear quadratic optimal feed-back regulator [21], time-optimal [22], minimum fuel [23], minimum tracking error [24], etc., where minimum tracking error methods were proposed to facilitate autonomous vehicle collision avoidance [25]. These are directly compared in [26] using several figures of merit in the face of parameter uncertainty and noisy sensor fusion: state tracking error, rate tracking error, fuel consumption, and computational burden.

This article recommends the best method (of six methods compared) for moving a vehicle from an initial normalized state of zero to a final normalized state of unity in a scaled-time of one second in the presence of uncertain mass and mass moments of inertia, and noisy state and rate sensor data. Uncertainty in the moment of inertia is assumed uniform  $\pm 10\%$ , while uncertainty in state and rate is assumed Gaussian with zero mean  $\mu$ , and standard deviation  $\sigma$  of 0.01.

Six methods are derived, modeled, and compared in simulation experiments:

1. P+V Control (proportional control plus velocity negation);
2. Open-loop optimal control (feedforward);
3. Real-Time Optimal Control (RTOC) (feedforward plus feedback);
4. Double-Integrator patching filter with P+V control;
5. Double-integrator patching filter with gain-tuning for P+V control;
6. Control law inversion patching filter with P+V control.

The methods for implementing these control schemes is outlined in the following sections. Section 2 establishes a benchmark control for comparison, namely velocity control augmenting state feedback. Section 3 describes the methods for ascertaining the control minimizing state, rate, and acceleration trajectories and optimal control utilizing Pontryagin's minimization condition, the adjoint equations, and terminal transversality of the endpoint Lagrangian. Following the introduction of optimization methods, slight modifications are introduced to enable real-time optimization based on noisy sensor data. Section 4 introduces methods to incorporate the optimization results from Section 3 into a pre-existing system designed without optimization in mind. Three so-called patching filters are introduced: the double-integrator patching filter, a gain-tuned double-integrator patching filter, and a control law inverting patching filter. Section 5 presented the results of one-thousand simulation experiments using the models developed in Sections 3 and 4 and figures of merit are presented to reveal relative superiority compared to the classical benchmark controller.

## 2. Materials and Methods

Methods are developed with variables defined in proximal tables (e.g., Tables 1 and 2), where simulation topologies are provided in Appendix A to aid repeatability.

### 2.1. Classical Benchmark Control

Using an end time of  $t_f = 1$ , a P + V Control System is used with gain parameters tuned to performance specifications using the closed-loop system Equation (1) below [26]. The following equation is used to match desired rise and settling times to position and velocity gains  $K_P$  and  $K_V$ .

$$I\ddot{x} + K_V\dot{x} + K_Px \iff u = K_P(x_d - x) - K_P\dot{x} \quad (1)$$

$$\frac{x(s)}{x_d(s)} = \frac{K_P}{Is^2 + K_Vs + K_P} \rightarrow C.E. : s^2 + K_Vs + K_P|_{I=1} = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (2)$$

The damping coefficient is  $\zeta = 0.7$ , and the desired settling time is  $t_s = 0.6$ . The settling time is defined as the time it takes for oscillation to stabilize within 2% of steady state  $t_s = \frac{4.6}{\zeta\omega_n} \rightarrow \omega_n = \frac{4.6}{t_s\zeta} \rightarrow \omega_n = 10.95 \rightarrow K_V = 2\zeta\omega_n \rightarrow K_V = 15.33$ . The rise time is

$t_r = \frac{1.8}{\omega_n} = 0.164 \rightarrow K_P = \omega_n^2 = 119.95$ , giving  $K_P = 119.95$  and  $K_V = 15.33$ .  $t_s$  is set to 0.6 s to ensure settling time before the stop time of 1, so that  $\theta_f$  and  $\omega_f$  can be compared to their desired final values of 1 and 0, respectively. In the following analysis,  $\mu_{\theta_f}$ ,  $\sigma_{\theta_f}$ ,  $\mu_{\omega_f}$ ,  $\sigma_{\omega_f}$ ,  $\mu_{J_{cost}}$ , and  $\sigma_{J_{cost}}$  are figures of merit compared across the four error simulations specified in the previous section. Here,  $\mu$  denotes the average value of the subscripted parameter over  $N = 1000$  Monte Carlo simulations, and  $\sigma$  denotes the standard deviation.

**Table 1.** Proximal variable definitions.

| Variable | Definition               | Variable   | Definition             |
|----------|--------------------------|------------|------------------------|
| $x_d$    | Desired state trajectory | $\xi$      | Critical damping ratio |
| $K_P$    | Proportional gain        | $\omega_n$ | Natural frequency      |
| $K_V$    | Velocity gain            | $t_s$      | Settling time          |
| $I$      | Moment of inertia        | $t_r$      | Rise time              |

## 2.2. Finding the Optimal Control

To develop a real-time optimal controller using Pontryagin's Optimization Principles, an analytic solution for optimal state and rate trajectories will be found given certain constraints and arbitrary initial and final conditions  $(\theta_0, \omega_0)$  and  $(\theta_f, \omega_f)$ , with arbitrary start and end times  $t_0$  and  $t_f$ . Here,  $I$  is the moment of inertia of the vehicle, and  $\tau$  is the torque applied.

The goals are as indicated in Equation (3):

$$\begin{aligned}
 \text{Minimize } J_{cost}[x(\cdot), u(\cdot)] &= \frac{1}{2} \int_{t_0}^{t_f} \tau^2(t) dt \\
 \text{Subject to } \dot{\theta} &= \omega \\
 \dot{\omega} &= \frac{\tau}{I} \\
 &(\theta_0, \omega_0) \\
 &(\theta_f, \omega_f) \\
 &t_0 \\
 &t_f
 \end{aligned} \tag{3}$$

$J_{cost}$ , the quadratic cost function, is chosen to be  $\frac{1}{2} \int_{t_0}^{t_f} \tau^2(t) dt$  because it scales with applied torque (and thus proportional to fuel usage), is positive-definite, and increases with time.

A Hamiltonian is defined as a new cost function that allows one to define a running cost  $F$  that can increase as a function of time, an endpoint cost  $E(\theta_f)$  that penalizes inaccuracy in final state, and endpoint constraints. Minimizing this and solving for optimal constants via Pontryagin's method leads to an optimal control scheme. To solve this double-integrator quadratic control (DQC) problem, the following steps will be used:

1. Formulate the Hamiltonian;
2. Minimize the Hamiltonian;
3. Formulate the Adjoint Equations;
4. Apply Terminal Transversality of the endpoint Lagrangian.

Finally, the resulting solution will be given in a matrix form which allows for the optimal solution to be found at every time step, leading to RTOC.

### 2.2.1. Formulate the Hamiltonian

There is no endpoint cost  $E(\theta_f)$  specified and only the running cost  $F$  is present.  $F$  is defined as the integrand of  $J_{cost}$ :

$$F = \frac{1}{2} \tau^2 \tag{4}$$

The state-space form of  $\dot{x} = f(x, u, p)$  is

$$f(\theta, \tau, I) = \begin{Bmatrix} \omega \\ \frac{\tau}{I} \end{Bmatrix} \quad (5)$$

$$H = F + \lambda^T f(\theta, \tau, I) = \frac{1}{2}\tau^2 + \lambda_\theta \omega + \lambda_\omega \frac{\tau}{I} \quad (6)$$

### 2.2.2. Minimize the Hamiltonian

$$\begin{aligned} \frac{\partial H}{\partial \tau} &= 0 \\ \tau + \frac{\lambda_\omega}{I} &= 0 \\ \tau^* &= -\frac{\lambda_\omega}{I} \end{aligned} \quad (7)$$

Here, the  $'^*$  donates the optimal control or trajectory.

### 2.2.3. Formulate the Adjoint Equation

$$\begin{aligned} \frac{\partial H}{\partial \theta} &= -\dot{\lambda}_\theta \\ 0 &= \dot{\lambda}_\theta \\ \Rightarrow \lambda_\theta &= c_1 \end{aligned} \quad (8)$$

where  $c_i$  are constants.

$$\begin{aligned} \frac{\partial H}{\partial \omega} &= -\dot{\lambda}_\omega \\ \lambda_\theta &= -\dot{\lambda}_\omega = c_1 \\ \Rightarrow \lambda_\omega &= -c_1 t + c_2 \end{aligned} \quad (9)$$

$$\begin{aligned} \Rightarrow \tau^* &= -\frac{\lambda_\omega}{I} = \frac{1}{I}(c_1 t - c_2) = I\dot{\omega}^* \\ (c_1 t - c_2) &= I^2 \dot{\omega}^* \\ I^2 \omega^* &= \frac{1}{2}c_1 t^2 - c_2 t + c_3 \\ I^2 \theta^* &= \frac{1}{6}c_1 t^3 - \frac{1}{2}c_2 t^2 + c_3 t + c_4 \end{aligned} \quad (10)$$

In addition, using initial conditions and assuming  $I = 1$ ,

$$(\theta_0, \omega_0) \Rightarrow c_3 = \omega_0, c_4 = \theta_0 \quad (11)$$

Using  $t'_f = t_f - t_0$  and  $t' = t - t_0$ , the final conditions imply as in Equation (12):

$$\begin{aligned} &(\theta_f, \omega_f) \\ \Rightarrow c_2 &= \frac{1}{2}c_1 t_f + \frac{\omega_0 - \omega_f}{t_f} \\ \text{and } \theta_f &= \frac{1}{6}c_1 t_f^3 - \frac{1}{2}c_2 t_f^2 + c_3 t_f + c_4 \\ \Rightarrow c_1 &= 12\left(\frac{\theta_0 - \theta_f}{t_f^3}\right) + 6\left(\frac{\omega_0 + \omega_f}{t_f^2}\right) \\ \Rightarrow c_2 &= 6\left(\frac{\theta_0 - \theta_f}{t_f^2}\right) + \frac{4\omega_0 - 2\omega_f}{t_f} \end{aligned} \quad (12)$$

Leading to the optimal trajectory and control:

$$\begin{aligned}\tau^* &= \frac{1}{I}(c_1 t' - c_2) \\ I^2 \omega^* &= \frac{1}{2} c_1 t'^2 - c_2 t' + c_3 \\ I^2 \theta^* &= \frac{1}{6} c_1 t'^3 - \frac{1}{2} c_2 t'^2 + c_3 t' + c_4\end{aligned}\quad (13)$$

This control is used for the open-loop guidance control.

#### 2.2.4. Terminal Transversality of the Endpoint Lagrangian

The endpoint cost  $E(\theta_f)$  is kept as 0:

$$E(\theta_f) = 0 \quad (14)$$

The endpoint function  $e(\bar{\theta}_f)$  is set to equal zero:

$$e(\bar{\theta}_f) = \begin{bmatrix} \theta_f \\ \omega_f \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (15)$$

Thus, the endpoint Lagrangian is

$$\bar{E}(\theta_f, \nu) = E + \nu^T e(\bar{\theta}_f) = 0 \quad (16)$$

where  $\nu$  is defined as the covector and is composed of the adjoints ( $\lambda$ ) found in Section 2.2.3

$$\nu = \begin{bmatrix} \lambda_\theta \\ \lambda_\omega \end{bmatrix} \quad (17)$$

Because  $\bar{E}(\theta_f, \nu) = 0$ , there is no terminal transversality condition to be applied. The optimal state and rate trajectory are shown in Figure 1.



**Figure 1.** Artemis missions, NASA will land the first woman and first person of color on the Moon, using innovative technologies to explore more of the lunar surface than ever before image courtesy NASA [27], image use in accordance with NASA image use policy: [28].

### 2.2.5. Real-Time Optimal Control (RTOC)

At each time step, the simulation solves a matrix equation to find the best constants  $\vec{p} = [c_1 \ c_2 \ c_3 \ c_4]$  that will achieve the desired end state and rate given the current state and rate. If no noise or external disturbances are present, these should always be the optimal constants found in Equations (8) and (9). However, the presence of noise and disturbances necessitates adjustment to reach the end goal. The form of the matrix equation is taken directly from the set of linear equations found in Equations (10), (12), and (13) and is presented as follows:

$$[T] \vec{p} = \vec{b} \quad (18)$$

$$\begin{bmatrix} \frac{t'^3}{6} & \frac{-t'^2}{2} & t' & 1 \\ \frac{t_f^3}{6} & \frac{-t_f^2}{2} & t_f & 1 \\ \frac{t'^2}{2} & -t' & 1 & 0 \\ \frac{t_f^2}{2} & -t_f & 1 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} \theta(t') \\ \omega(t') \\ \theta_f \\ \omega_f \end{bmatrix} \quad (19)$$

$$[T]^{-1} \vec{b} = \vec{p} \quad (20)$$

$[T]$  is defined as the matrix in Equation (19).  $\vec{p}$  is defined as the vector of constants  $c_i$ , where  $I$  is absorbed into these constants when the matrix equation is solved.  $\vec{b}$  is defined as the vector of current and final state and rate as in Equation (19). Performing the matrix inverse in Equation (20) solves for the optimal constants  $\vec{p}$  at every time step.

The simulation was run with the optimal control, a moment of inertia of  $I = 1$ ,  $t_0 = 0$ ,  $t_f = 1$  s,  $(\theta_0, \omega_0) = (0, 0)$ , and  $(\theta_f, \omega_f) = (1, 0)$ . Multiple matrix inverse solvers were tested to see which performed best. The results are summarized in Table 2.

When  $|\det[T]| \leq 0.001$ , the control is switched to the open-loop optimal solution, as a matrix with zero determinant is noninvertible, and the matrix inversion would provide a diverging solution.

**Table 2.** Figures of Merit for RTOC implementing various MATLAB inversion techniques while adding uncertainty. Gaussian noise in  $\hat{\theta}$  and  $\hat{\omega}$  of  $\sigma = 0.01$  and Uniform Distribution of  $I = 1 \pm 0.1$  ( $N = 1000$ ). Overall step size  $h=0.001$  s. RTOC update rate  $h = 0.01$  s.

| Figure of Merit            | $\frac{1}{[T]}$         | $\text{pinv}([T])$ |
|----------------------------|-------------------------|--------------------|
| $\mu_{\hat{\theta}_f}$     | -0.6178                 | 1.0014             |
| $\sigma_{\hat{\theta}_f}$  | $1.5996 \times 10^{-4}$ | 0.0035             |
| $\mu_{\hat{\omega}_f}$     | -1.9737                 | 0.0029             |
| $\sigma_{\hat{\omega}_f}$  | $2.0012 \times 10^{-4}$ | 0.0094             |
| $\mu_{J_{\text{cost}}}$    | 3.1891                  | 6.0167             |
| $\sigma_{J_{\text{cost}}}$ | $2.8279 \times 10^{-4}$ | 0.0218             |

MATLAB's  $\text{pinv}([T])$  was used to solve RTOC because of its greatly improved mean error in state and rate over the  $1/[T]$  method, and ability to prevent singularities. Other methods were considered but were unusable. MATLAB was unable to use the  $[T]^{-1}$  function on this particular matrix. The  $\text{inv}([T])$  and LU Inverse functions caused the matrix values to diverge to infinity at the end of the simulation and were unusable. While the determinant switch can switch to open-loop control and bypass the divergent solution, MATLAB still attempts to calculate the matrix inverse, and fails due to the presence of singularities. Both the overall integration time step and the update time step of the RTOC matrix inversion are  $h = 0.01$  s. Decreasing  $h$  results in higher accuracy but higher computation time as well. Depending on the use case, higher accuracy may be preferable, especially in the presence of greater uncertainty and external forces.  $h = 0.01$  s was found to perform well in accuracy while not requiring significant computation time for the 1 s simulation.

The resulting constants from  $\vec{p}$  are fed as the control as in Equation (13).

### 2.3. Patching Filters

Instead of feeding in the desired end state to the P+V controller, patching filters take in the optimal control and feed in the desired current state as determined by the optimal control. Here, except for the case of gain-tuning (Section 2.3.2), it is assumed that the gains of the P+V controller cannot be altered from their form found in Section 2:

$$u(t) = (\theta_d(t) - \hat{\theta}(t))K_P - \hat{\omega}(t)K_V \quad (21)$$

where  $\theta_d(t)$  is calculated in a few different ways:

#### 2.3.1. Double-Integrator Patching Filter

Here, the open-loop optimal solution is fed into a double-integrator and fed as an input to the P+V controller. The double-integrator model essentially feeds in exactly the optimal trajectory  $\theta^*(t) = \theta_d(t)$  to the P+V controller instead of  $\theta_{f,d}$ , the desired final state.

#### 2.3.2. Double-Integrator Patching Filter with Gain-Tuning

Here, it is assumed that the gains in the P+V controller *can* be altered. By performing manual tuning,  $K_P = 280$  and  $K_V = 15.33$  were found to work best in arriving at the desired end state, correcting for errors that arose in the case of the double-integrator patching filter without gain-tuning. Higher or lower velocity gains worsened accuracy, while higher position gains only increased the velocity. These newfound gains are used in the simulation for the double-integrator patching filter with gain-tuning.

#### 2.3.3. Control Law Inversion Patching Filter

Here, the open-loop optimal solution is fed into the following transfer function, which is then fed as an input to the P+V Controller:

$$\frac{1}{K_P} \left[ \frac{s^2 + K_V s + K_P}{s^2} \right] \quad (22)$$

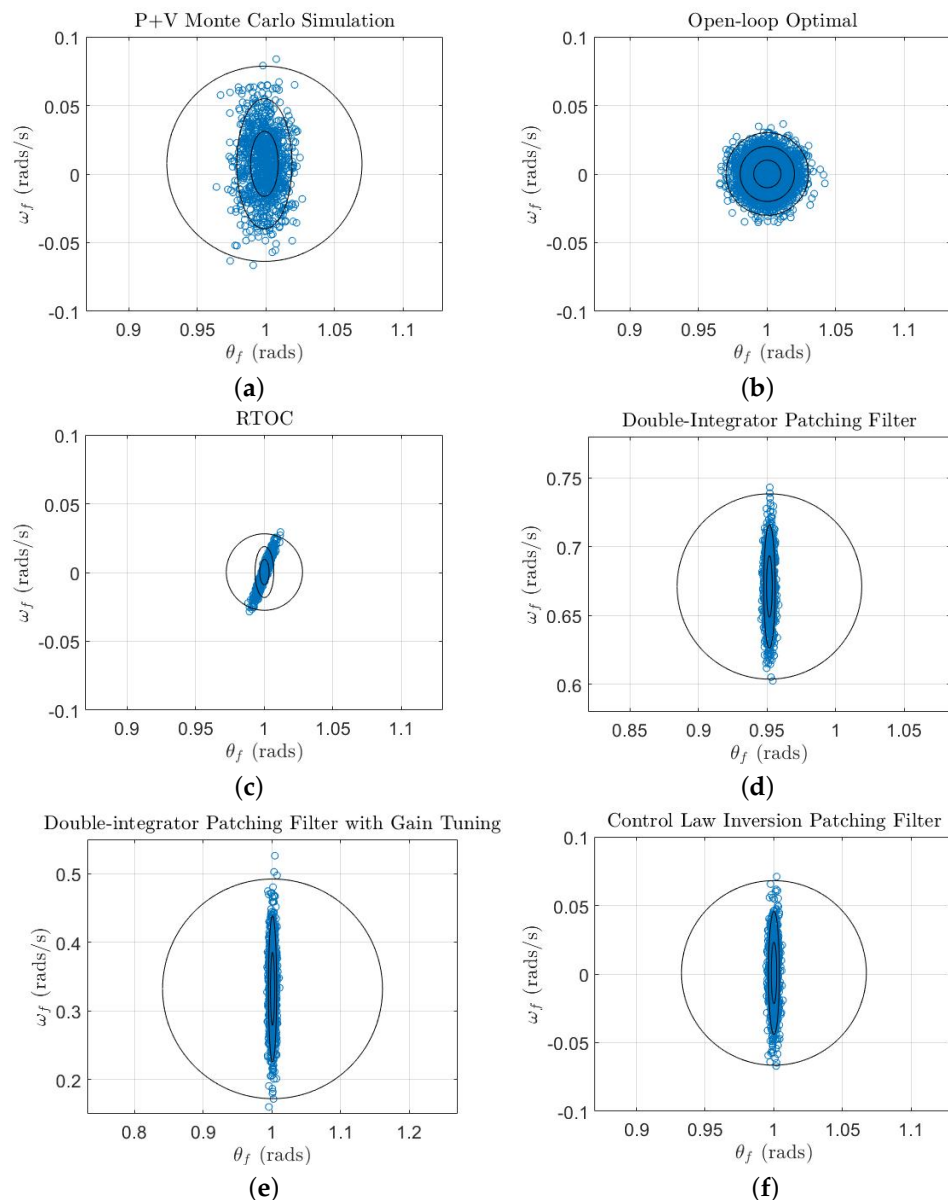
This takes into account the fact the P+V controller gains will inherently change the resulting  $\theta(t)$  to one that is not the  $\theta^*(t)$  fed in to the filter. This transfer function aims to predict the effects that the P+V controller will have and input a  $\theta_d(t)$  that counters that effect to retain optimality. According to Equation (2), this inverts the effect of the P+V controller, and adds a double-integrator (a factor of  $\frac{1}{s^2}$ ) to it.

## 3. Results

A Monte Carlo simulation was run for each of the six methods using a uniform  $\pm 10\%$  uncertainty in  $I$  so that  $I$  varied from 0.9 to 1.1. The state and rate sensors are subject to a Gaussian distribution of “white noise” with mean 0 and a standard deviation of 0.01.  $N = 1000$  simulations are done for each method. MATLAB’s ode4 Runge–Kutta integration solver was used with a step size of  $h = 0.01$  s.

State and rate data are taken prior to going through the noisy sensors. The large, systematic diagonal spread in final state and rate values is most likely to stem from the uniform uncertainty in  $I$ , not the noisy sensor data.

In Figure 2, the standard deviation ellipses plotted are those of the measured data in  $\theta_f$  and  $\omega_f$ . In addition, 1, 2, and 3 standard deviations are plotted.



**Figure 2.** Monte Carlo simulations of six different controller schemes,  $n = 1000$ . Gaussian noise in state and rate sensors with zero mean and  $\sigma = 0.01$  and uniform uncertainty in  $I = 1 \pm 0.1$ . All simulations are run with ode4 Runge-Kutta integration with a step size of  $h = 0.01$  s. (a) P+V control, (b) open-loop optimal feedforward control, (c) real-time optimal control, (d) double-integrator patching filter with P+V control.  $K_P = 119.95$  and  $K_V = 15.33$ , (e) double-integrator patching filter with gain-tuning of P+V control.  $K_P = 280$  and  $K_V = 15.33$ , (f) control law inversion patching filter with P+V control.  $K_P = 119.95$  and  $K_V = 15.33$ .

#### 4. Discussion

The results summarized in Table 3 will be compared and contrasted. It is noted that state and rate data presented are “actual” final states and rates before passing through noisy sensors.

It is clear from the results that the double-integrator patching filter without gain tuning is the least accurate method. The errors in accuracy are four orders of magnitude higher in state and rate than RTOC. While the quadratic cost of the maneuver is 28% lower than the case of open-loop optimal control and RTOC, the loss in accuracy is too large for this method to be considered a good option. With gain-tuning, the state error was decreased by two orders of magnitude, making it the second-most accurate method for state out of

the six. However, it still has a large, systematic error in rate, as well as a higher cost than the double-integrator patching filter without gain-tuning and the control law inversion patching filter. This systematic error makes the gain-tuning with double-integrator patching filter also a relatively bad method.

**Table 3.** Figures of Merit for adding uncertainty to various control schemes. Gaussian noise in  $\hat{\theta}$  and  $\hat{\omega}$  of  $\sigma = 0.01$  and Uniform Distribution of  $I = 1 \pm 0.1$  ( $N = 1000$ ). ode4 (Runge–Kutta) integration with  $h = 0.01$  s used.

| Figure of Merit              | P+V Control:<br>$K_P = 119.95$ ,<br>$K_V = 15.33$ | Open-Loop<br>Optimal<br>Guidance | RTOC                     | Double-<br>Integrator<br>Patching Filter<br>$K_P = 119.95$ ,<br>$K_V = 15.33$ | Gain-Tuning<br>with Double-<br>Integrator<br>Patching Filter<br>$K_P = 280$ ,<br>$K_V = 15.33$ | Control Law<br>Inversion<br>Patching Filter<br>$K_P = 119.95$ ,<br>$K_V = 15.33$ |
|------------------------------|---------------------------------------------------|----------------------------------|--------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| $\mu_{err_{\hat{\theta}_f}}$ | 0.001                                             | 0.0002                           | $-6.4854 \times 10^{-5}$ | −0.0484                                                                       | $7.3814 \times 10^{-4}$                                                                        | $1.4422 \times 10^{-5}$                                                          |
| $\sigma_{\hat{\theta}_f}$    | 0.0101                                            | 0.099                            | 0.0035                   | 0.0021                                                                        | 0.0030                                                                                         | 0.0020                                                                           |
| $\mu_{err_{\hat{\omega}_f}}$ | 0.0075                                            | $9.1195 \times 10^{-5}$          | $-9.4813 \times 10^{-5}$ | 0.6711                                                                        | 0.3322                                                                                         | $8.2613 \times 10^{-4}$                                                          |
| $\sigma_{\hat{\omega}_f}$    | 0.0237                                            | 0.0100                           | 0.0093                   | 0.0224                                                                        | 0.0534                                                                                         | 0.0224                                                                           |
| $\mu_{J_{cost}}$             | 236.72                                            | 6.0110                           | 6.0168                   | 4.3251                                                                        | 9.1355                                                                                         | 6.6956                                                                           |
| $\sigma_{J_{cost}}$          | 13.4524                                           | 0.6917                           | 0.0213                   | 0.1535                                                                        | 0.6566                                                                                         | 1.660                                                                            |

The P+V controller has accuracy and precision within the same order of magnitude as that of open-loop optimal control, except in the rate, where open-loop optimal wins by three orders of magnitude. The major issue with P+V control is cost. For essentially the same accuracy, the maneuver costs 1.5 orders of magnitude more. The benefit of P+V control over open-loop optimal, however, is that it is a feedback control mechanism rather than a feedforward mechanism. This allows it to correct for external disturbances. A feedforward mechanism, on the other hand, is unaffected by sensor noise (even if it is of nonzero mean). Which one is more desirable in a given scenario is then determined by the context of the engineering problem and the presence of disturbances and noise of nonzero noise, which is not tested here.

Open-loop optimal control may have the highest accuracy in rate, but the control law inversion patching filter has the highest accuracy in state, and outperforms all other methods except RTOC in rate accuracy. Its precision is lower than that of RTOC, but its increase in accuracy outweighs this. The control law inversion patching filter has a slightly higher mean cost than RTOC and open-loop optimal guidance but is relatively insignificant. The spread in cost is also higher. RTOC's slightly 0.5 order of magnitude lower state accuracy and one order of magnitude higher rate accuracy, combined with improved slightly improved rate precision and lower state precision, makes it a good competitor with the control law inversion patching filter, depending on desirable traits.

All figures in Figure 2 are made with equal axes. It should be noted that Figure 2d,e has a shifted center of the graph to encompass the data points. These points center on a different point than the other methods due to a systematic error inherent in the double-integrator patching filter. These graphs also provide visual cues indicating the spread patterns of the various methods. The open-loop optimal feedforward control has a relatively circular distribution of points, as it is not subject to noise in the state and rate sensors, but just uncertainty in the moment of inertia. The spreads of the other methods are more elliptical in nature. RTOC has a smaller spread than any other method, but has a noticeable diagonality to it—indicating a bias from the changing moment of inertia.

Computational burden was not measured as a figure of merit due to the inability to obtain meaningful results. Other processes were running on the same computer during these simulations, thus affecting run time, or computational burden, measurements.

## 5. Conclusions

The control law inversion patching filter and RTOC work best out of the six options for general purposes. The decision for which control scheme is best ultimately depends on the relative use case, context, and requirements. Strict cost requirements or precision requirements may outweigh benefits that the control law inversion patching filter provides over RTOC, for example. The control law inversion filter also provides a benefit over RTOC in computational burden due to the lack of a matrix inversion scheme.

Using optimality will reduce cost significantly over traditional methods, as indicated by the steep decrease in  $J_{cost}$  between P+V control and any optimal control scheme. If traditional methods are being used, one can incorporate optimality by the use of patching filters and combine the benefits of traditional feedback controllers with the cost benefits of Pontryagin optimization.

A future study could be done running the control law inversion patching filter with RTOC and measuring its performance. Future study may also be done studying the effects of these control schemes with the inclusion of a full 6-degrees-of-freedom, three-dimensional, coupled, nonlinear equations of motion with external forces and Coriolis forces present.

**Author Contributions:** Conceptualization, P.B. and T.S.; methodology, P.B. and T.S.; software, P.B. and T.S.; validation, P.B. and T.S.; formal analysis, P.B. and T.S.; investigation, P.B. and T.S.; resources, T.S.; data curation, P.B.; writing—original draft preparation, P.B.; writing—review and editing, P.B. and T.S.; visualization, P.B.; supervision, T.S.; project administration, P.B. and T.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## Abbreviations

The following abbreviations and symbols are used in this manuscript:

|               |                                                     |
|---------------|-----------------------------------------------------|
| P+V           | Proportional and velocity                           |
| RTOC          | Real-time optimal control                           |
| MAE           | Mechanical and aerospace engineering                |
| $J_{cost}$    | Quadratic cost                                      |
| $\mu_x$       | Mean value of parameter $x$                         |
| $\sigma_x$    | Standard deviation of parameter $x$                 |
| $F$           | Running cost                                        |
| $E(\theta_f)$ | Endpoint cost                                       |
| $\theta$      | State or angle (synonymous)                         |
| $\theta_f$    | Final state, or angle (synonymous)                  |
| $\omega$      | Rate or angular velocity (synonymous)               |
| $\omega_a$    | Final rate or angular velocity (synonymous)         |
| $x^*$         | Optimal value of parameter $x$                      |
| $\hat{x}$     | Value of parameter $x$ from sensor data             |
| $\tau$        | Applied torque                                      |
| $I$           | Moment of inertia of vehicle                        |
| $K_p$         | Position gain                                       |
| $K_v$         | Velocity gain                                       |
| $t_r$         | Rise time                                           |
| $t_s$         | Settling time                                       |
| $u$           | Control parameter. In this case, $u = \tau$ .       |
| $[T]$         | Matrix of time coefficients for RTOC                |
| $\vec{p}$     | Optimal coefficients at each time step for RTOC     |
| $\vec{b}$     | Vector of current and final state and rate for RTOC |

## Appendix A. MATLAB Simulink Model

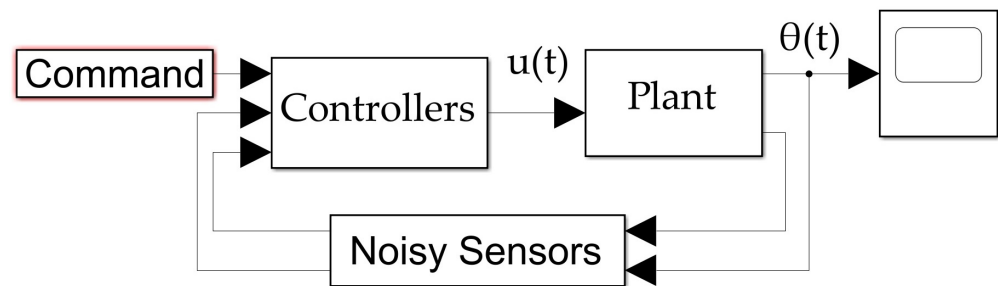


Figure A1. Upper level Simulink model.

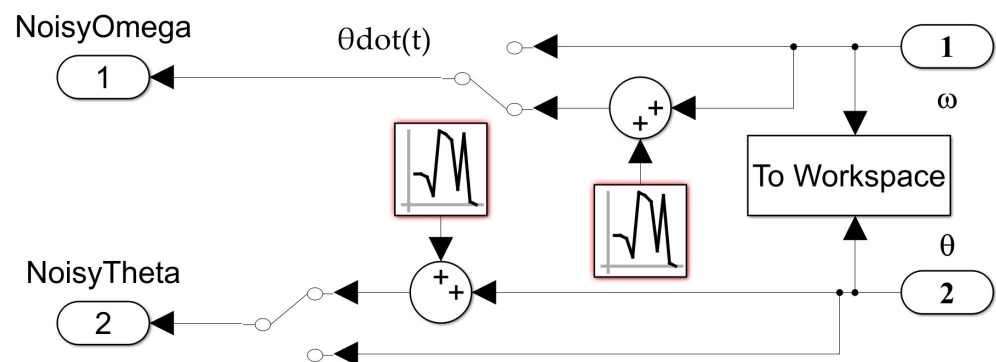


Figure A2. Addition of Gaussian noise to sensor data.

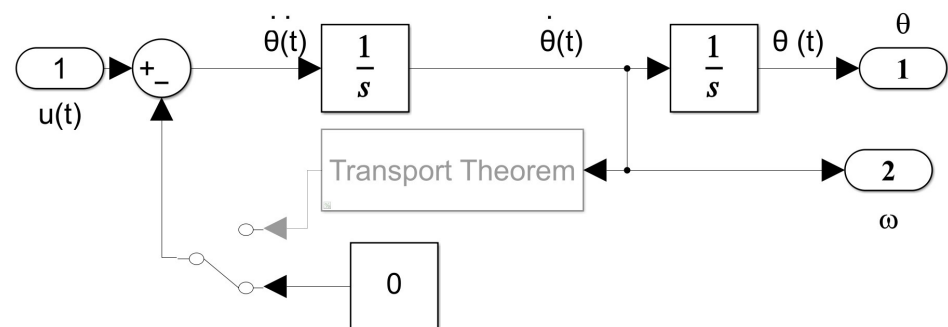


Figure A3. Double-integrator plant to convert input control  $\tau$  into state and rate data. Including the transport theorem was not done; however, one could add it here if desired.

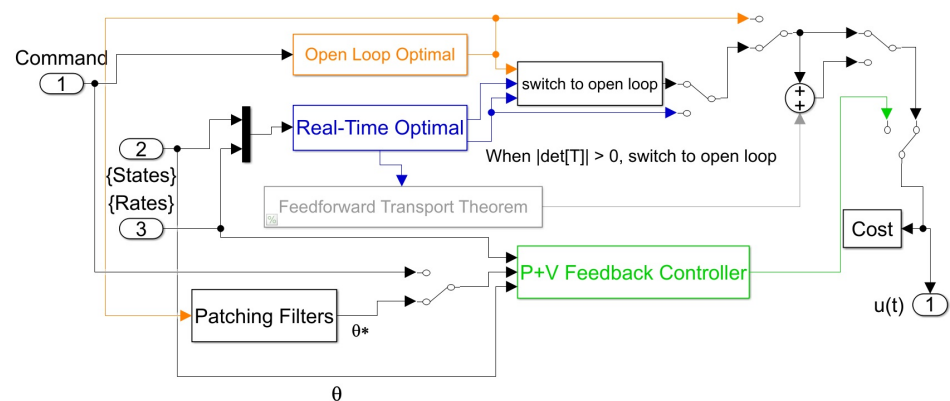
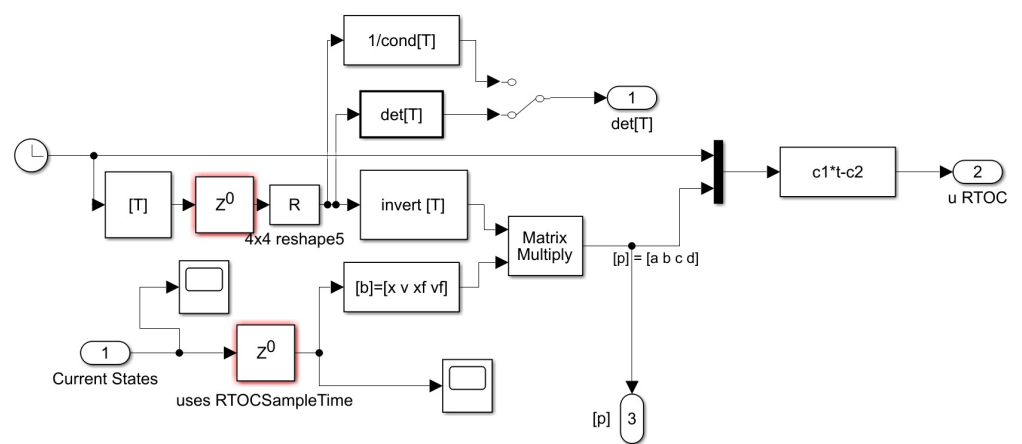
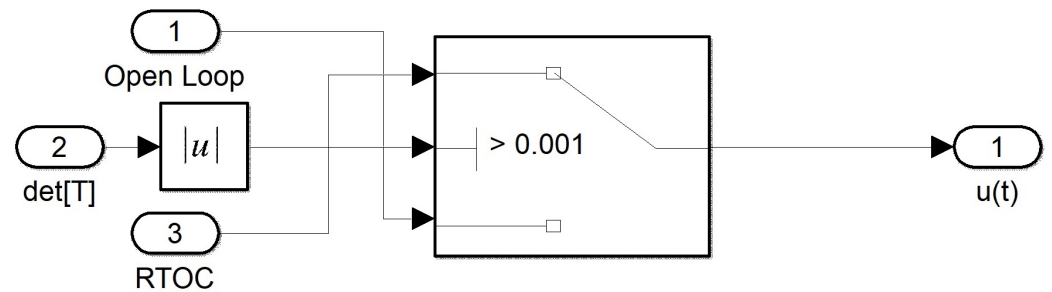


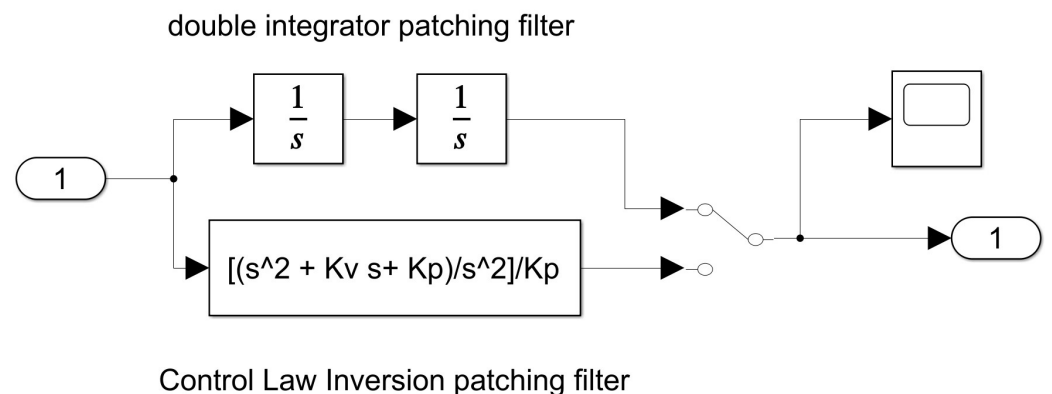
Figure A4. Controllers Simulink block.



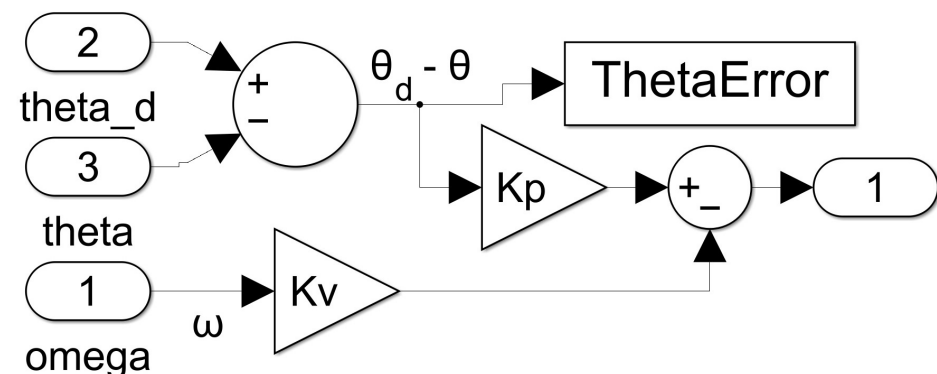
**Figure A5.** Real-time optimal control Simulink model.



**Figure A6.** Determinant switching logic for RTOC. When the absolute value of  $\det[T] < 0.001$ , the switch flips to open-loop optimal control.



**Figure A7.** Simulink implementation of patching filters. A manual switch is present to switch between a simple double-integrator patching filter and a control law inversion patching filter.



**Figure A8.** Simulink implementation of a P+V feedback controller.

## References

- Chasle, M. Note sur les propriétés générales du système de deux corps semblables entr'eux. *Bull. Sci. Math. Astron. Phys. Chem.* **1830**, *14*, 321–326. (In French)
- Euler, L. Formulae Generales pro Translatione Quacunque Corporum Rigidorum (General Formulas for the Translation of Arbitrary Rigid Bodies.) *Novi Comment. Acad. Sci. Petrop.* **1776**, *20*, 189–207. Available online: <https://scholarlycommons.pacific.edu/euler-works/478/> (accessed on 31 August 2022).
- Newton, I. *Principia, Jussu Societatis Regiæ*; (Ac Typis Joseph Streater); Cambridge University Library: London, UK, 1687.
- Cooper, M.; Heidlauf, P.; Sands, T. Controlling Chaos—Forced van der pol equation. *Mathematics* **2017**, *5*, 70. [CrossRef]
- Smeresky, B.; Rizzo, A.; Sands, T. Optimal Learning and Self-Awareness Versus PDI. *Algorithms* **2020**, *13*, 23. [CrossRef]
- Baker, K.; Cooper, M.; Heidlauf, P.; Sands, T. Autonomous trajectory generation for deterministic artificial intelligence. *Electr. Electron. Eng.* **2018**, *8*, 59–68.
- Sands, T. Development of Deterministic Artificial Intelligence for Unmanned Underwater Vehicles (UUV). *J. Mar. Sci. Eng.* **2020**, *8*, 578. [CrossRef]
- Slotine, J.-J.E.; Weiping, L. *Applied Nonlinear Control*; Prentice-Hall: Hoboken, NJ, USA, 1991.
- 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. [CrossRef]
- Sands, T.; Kim, J.; Agrawal, B. Spacecraft Adaptive Control Evaluation. In Proceedings of the InInfotech@ Aerospace, Garden Grove, CA, USA, 19–21 June 2012.
- University of Wisconsin. Staff-Authored News from College of Engineering. 2019. Available online: <https://engineering.wisc.edu/news/> (accessed on 6 October 2022).
- Malecek, A. Robert Lorenz, Pioneer in Controls Engineering, Passes Away. 2019. Available online: <https://engineering.wisc.edu/news/robert-lorenz-pioneer-controls-engineering-passes-away/> (accessed on 6 October 2022).
- Sands, T.; Lorenz, R. Physics-based automated control of spacecraft. In Proceedings of the AIAA Space 2009, Pasadena, CA, USA, 14–17 September 2009.
- Rätze, K.; Bremmer, J.; Biegler, L.; Sundmacher, K. Physics-Based Surrogate Models for Optimal Control of a CO<sub>2</sub> Methanation Reactor. *Comput. Aided Chem. Eng.* **2017**, *40*, 127–132.
- Bukhari, A.; Raja, M.; Asif, Z.; Shoaib, M.; Kiani, A. Fractional order Lorenz based physics informed SARFIMA-NARX model to monitor and mitigate megacities air pollution. *Chaos Solit. Fract.* **2022**, *161*, 112375. [CrossRef]
- Sandberg, A.; Sands, T. Autonomous Trajectory Generation Algorithms for Spacecraft Slew Maneuvers. *Aerospace* **2022**, *9*, 135. [CrossRef]
- Cooper, M.; Heidlauf, P. Nonlinear Lyapunov control improved by an extended least squares adaptive feed forward controller and enhanced Luenberger observer. In Proceedings of the International Conference and Exhibition on Mechanical & Aerospace Engineering, Las Vegas, NV, USA, 2–4 October 2017.
- Cooper, M.; Heidlauf, P. Nonlinear feed forward control of a perturbed satellite using extended least squares adaptation and a luenberger observer. *J. Aero. Aerosp. Eng.* **2018**, *7*, 1. [CrossRef]
- Cooper, M.; Smeresky, B. An overview of evolutionary algorithms toward spacecraft attitude control. In *Advances in Spacecraft Attitude Control*; IntechOpen: London, UK, 2020.
- Chow, G.C. *Analysis and Control of Dynamic Economic Systems*; Krieger Publ.: Malabar, FL, USA, 1986; ISBN 0-89874-969-7.
- Borggaard, J.; Zietsman, L. The Quadratic-Quadratic Regulator Problem: Approximating feedback controls for quadratic-in-state nonlinear systems. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 818–823.
- Kreindler, E. Contributions to the theory of time-optimal control. *J. Frank. Inst.* **1963**, *275*, 314–344. [CrossRef]
- Sciarretta, A.; Guzzella, L. Fuel-Optimal Control of Rendezvous Maneuvers for Passenger Cars (Treibstoffoptimale Annäherung von Straßenfahrzeugen). *Automatisierungstechnik* **2005**, *53*, 244–250. [CrossRef]
- Sensing and Controls Pioneer Lorenz Named to National Academy of Engineering. 2019. Available online: <https://engineering.wisc.edu/blog/sensing-controls-pioneer-lorenz-named-national-academy-engineering/> (accessed on 1 September 2022).
- Raigoza, K.; Sands, T. Autonomous Trajectory Generation Comparison for De-Orbiting with Multiple Collision Avoidance. *Sensors* **2022**, *22*, 18. [CrossRef]
- Sands, T. Treatise on Analytic Optimal Spacecraft Guidance and Control. *Front. Robot. AI Robot. Control Syst.* **2022**, *9*, 884669. <https://doi.org/10.20944/preprints202201.0467.v1> [CrossRef]
- NASA Artemis. Available online: <https://www.nasa.gov/specials/artemis/> (accessed on 1 September 2022).
- Media Usage Guidelines. Available online: <https://www.nasa.gov/multimedia/guidelines/index.html> (accessed on 1 September 2022).