

# Optimization imposition upon drone gimbal control electronics

Erhe Zheng<sup>1</sup>, Timothy Sands<sup>2,\*</sup>

<sup>1</sup> Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

<sup>2</sup> Department of Mechanical Engineering (SCPD), Stanford University, Stanford, CA 94305, USA

\* Corresponding author: Timothy Sands, dr.timsands@alumni.stanford.edu

---

## ARTICLE INFO

Received: 2 June 2023  
Accepted: 17 July 2023  
Available online: 25 July 2023

doi: 10.59400/jam.v1i2.69

Copyright © 2023 Author(s).

*Journal of AppliedMath* is published by Academic Publishing Pte. Ltd. This article is licensed under the Creative Commons Attribution License (CC BY 4.0).  
<https://creativecommons.org/licenses/by/4.0/>

**ABSTRACT:** The goal of the manuscript is to design a relatively good control structure for the noise suppression of a drone's camera gimbal action. The gimbal's movement can be simplified as a rest-to-rest reorientation system that can achieve the boundary result of a dynamic system. Six different control architectures are proposed and evaluated based on their ability to control the trajectory of the dynamic-system position and speed, their running time, and their quadratic cost. The robustness of the control architecture to uncertainties in inertia and sensor noise is also analyzed. Monte Carlo figures are used to assess the performance of the six control systems. The conditions for applying different architectures are identified through this analysis. The analysis and experimental tests reveal the most suitable control of the drone's camera gimbal rotation.

**KEYWORDS:** proportional plus velocity controller; double integrator patching filter; control law inversion patching filter; real-time optimal control; open loop control; feedback control; Monte Carlo model

---

## 1. Introduction

### 1.1. The introduction to the drone and its single-axis

Drones, also known as unmanned aerial vehicles, are a type of aircraft that can be controlled remotely without a pilot onboard<sup>[1]</sup>. The history of drones began after the First World War, when simple prototypes were used as weapons. In the late twentieth century, an autonomous control system was added to serve the operation of drones. Today, equipped with advanced technologies such as Global Positioning System<sup>[2]</sup> cameras, and sensors<sup>[3]</sup>, the drone has the ability for navigation, data collection, videography, or some specific objectives. The drone is becoming increasingly important in a wide range of industries, including agriculture construction<sup>[4,5]</sup>, journalism<sup>[6]</sup>, and filmmaking<sup>[7]</sup>, amongst other work. Drones of various sizes and shapes have been designed for a wide range of applications in various industries. Overall, drones are an important technological advancement and an increasingly vital tool that will continue to grow in modern society.

Cameras are essential for drones as they enable users to capture images and videos from a unique perspective that is different from that obtained by traditional cameras or from the ground. As the basis of robot perception, cameras work as eyes for aerial photography and videography, surveying and mapping, and inspection and monitoring of infrastructure and facilities<sup>[9]</sup>. Therefore, cameras are a crucial component of modern drones, and their performance will also directly impact the performance of the drone. **Figure 1a** illustrates the use of NASA's drone, and **Figure 1b** provides an approximate representation of the drone's gimbal.



**Figure 1. (a)** NASA uses drones for launch detection<sup>[8]</sup>; **(b)** The detail of the one-gimbal drone's camera.

Errors caused by camera mounting devices are considered one of the main factors affecting camera performance. The most significant source of noise affecting the performance of the camera, apart from inherent differences in its own accuracy, is the deviation and vibration caused during the moving process. Therefore, selecting a suitable mounting device and reducing noise are crucial. Various camera mounting devices are available, as described by Carmona<sup>[10]</sup>, who has designed a complex platform-style mechanical structure that can stabilize the camera and absorb vibrations. However, complex mechanical structures can be relatively cumbersome for drones that aim to be small and affordable. Danh<sup>[11]</sup> has described a dual-axis gimbal that stabilizes the camera, but this design still requires an excellent control structure to support the system's operation. The current mainstream solution on the market is to use a single-axis gimbal and electrical control structure to mount the camera, as explained in Gasparovic et al.'s article<sup>[12]</sup> where the advantages of the gimbal are clearly outlined. Therefore, using more precise control methods can ensure the accuracy of the movement of the majority of drones equipped with single-axis gimbals and cameras for visual control on the market.

## 1.2. The abstract introduction of gimbal control research

### 1.2.1. Challenges in the manipulation of the gimbal rotation

Although the motion model of the gimbal is not complex and can be simplified to a general Euler rotation model. Accurately rotating the gimbal to a specific angle is still a challenging task. This means that within a limited time frame, the gimbal is required to start from a stationary state, rotate to a specific angle, and maintain a stationary state with zero speed or acceleration at the end. Also, the operation time is limited. This rest-to-rest rigid body system is the research focus of this manuscript. With the use of electronic control structures, precise control is possible theoretically, but selecting a suitable control is still crucial realistically. Meanwhile, the noise in the position and velocity sensors poses a significant challenge to precise control. In real-life situations, such noise and uncertainty are inevitable factors that need to be taken into account.

### 1.2.2. How the challenges were addressed in the past

At the fundamental level, Kim et al.'s article<sup>[13]</sup> explains the basic control method PID and its application in the stabilization control of gimbal for drone gimbal, as well as the tuning process. Their article provides great inspiration for the control structure of the gimbal. Similar to PID control<sup>[14]</sup>, other control methods such as P+V control and some optimization control can also be applied to the drone camera gimbal while maintaining higher accuracy<sup>[15]</sup>. Lassak et al.<sup>[16]</sup> explain the importance of the filter or patching for the control structure, which helps the control structure to achieve control objectives better and faster. Actually, real-time control<sup>[17]</sup> is also crucial for a motion mechanism like a drone, and Aguilar<sup>[18]</sup> has presented a real-time image control method for drones in his paper. The approach is also applicable to gimbal control<sup>[19,20]</sup>. For simulating real environments, the noise in the sensors is a significant factor and the Monte Carlo model is particularly prominent in analyzing noise<sup>[21]</sup>. The application of

Monte Carlo by Binder<sup>[22]</sup> is a good example of data analysis. Last, Sands’s analysis of multiple controls<sup>[23]</sup> for space robots also provides a good instance for the comparison of different optimal-control structures. Although all control structures selected are optimization controls, they have their own advantages and disadvantages in different situations<sup>[24]</sup>.

### 1.2.3. How the major limitation in the project

Although this project has provided various excellent optimization control methods through the Pontryagin method, different control approaches demonstrate their own advantages in various environmental conditions. Theoretically, more outstanding control methods can be generated based on established system theory and existing output conclusions. These methods are also worth considering. Additionally, this paper adopts evaluation methods from realistic systems, including direct comparisons with declared benchmarks, comparisons using various performance data, and Monte Carlo chart comparisons. However, other analytical methods are worth introducing into the analysis process of control effectiveness.

### 1.3. The rest-to-rest model of the drone’s gimbal and six controls

The movement of the drone’s camera gimbal is simplified as a rest-to-rest rigid body system, which means that its velocities and acceleration at the beginning and end are zero, and its position changes in a limited time period. The rest-to-rest rigid-body system adheres to classical Newtonian mechanics and Euler rotation<sup>[25]</sup>.

When it comes to a rest-to-rest reorientation system, the fundamental model is a basic double integrator. The rotation equation of the reorientation system is the basis for optimization. Based on the Euler rotation theorem, the equation of the gimbal rotation is displayed in Equation (1):

$$I\ddot{\theta} = T_q = u \tag{1}$$

The name and explanations of symbols in some of the formulas are provided in **Table 1**.

**Table 1.** Proximal variable definitions.

Valuables	Definitions	Valuables	Definitions
$\theta$	Angle	$J$	the quadratic cost functional
$\omega$	Angular velocity	$F$	The dynamic cost function
$\ddot{\theta}$	Rotational acceleration	$f$	The dynamic constraint
$I$	The inertia of moment	$T_q$	System torque
$t_0$	The initial time	$t_f$	The final time
$u$	Control function of time	-	-

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

The boundary condition for the rest-to-rest maneuver is characterized by starting from a stationary position and ending with a new orientation and zero velocity. The expected duration of the maneuver is approximately one second. The boundary condition in this problem is simplified in Equation (2):

$$\begin{cases} \theta(0) = 0 \\ \omega(0) = 0 \\ \theta(1) = 1 \\ \omega(1) = 0 \end{cases} \tag{2}$$

In addition, the performance of the controller is also evaluated by the quadratic cost. The control architecture should tune the relevant parameters and select appropriate values to minimize the quadratic cost. Equation (3) is the cost equation used in this manuscript:

$$J = \frac{1}{2} \int_{t_0}^{t_f} u^2 dt = \int_{t_0}^{t_f} F dt \tag{3}$$

There are six control architectures used to achieve the control target:

- 1) Open loop quadratic-cost control with HzMAT optimal control result<sup>[26,27]</sup>.
- 2) Proportional plus velocity (P+V) feedback control.
- 3) Real-time optimal control (RTOC) with the result of the HzMAT optimal control<sup>[28]</sup>.
- 4) P+V feedback control with double-integrator patching filter<sup>[29]</sup>.
- 5) Gain-tuning P+V controller with double-integrator patching.
- 6) P+V controller with Control law inversion patching filter.

**Table 2** explains the meaning of the abbreviations in some controller names.

**Table 2.** Explanation of nouns.

Topic	Definitions
RTOC	Real-time optimal control
P+V	Proportional plus velocity
HzMAT	Optimal control using Hamilton functional
$K_p$ and $K_v$	A proportional value for P+V control

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

The main controllers used in this manuscript are the P+V controller with or without different patching, like double-integrator quadratic cost control, and the HzMAT optimal controller (which includes open-loop with HzMAT analysis results, RTOC (real-time optimal controller)). [30]

The P+V feedback controller is similar to the PID controller or PD controller. The P+V controller contains an amplifier to adjust the state values towards the target and an integrator to stabilize oscillations. The advantage of this controller is that it is simple to operate and easy to build. Compared to the open-loop controller, the P+V controller has better theoretical robustness to uncertainty. However, the P+V controller has two parameters. That is difficult to tune effectively. Therefore, the P+V controller may have poor accuracy and precision and cannot minimize the quadratic cost.

The HzMAT optimal-control algorithm is based on Pontryagin’s minimize principle, which converts the optimal control problem into a multipoint boundary value problem<sup>[31-33]</sup>. The HzMAT functions contain four main functions:

Hamiltonian minimization condition:

$$H = F + \lambda^T f(x, u) \tag{4}$$

$$\frac{\partial H}{\partial u} = 0 \tag{5}$$

Adjoint equation:

$$\frac{\partial H}{\partial x} = -\dot{\lambda}_x \tag{6}$$

Terminal transversality of Endpoint Lagrange:

$$\bar{E} = E + v^T e \tag{7}$$

$$\frac{\partial E}{\partial X_f} = \lambda_x(t_f) \tag{8}$$

Hamiltonian final value condition:

$$H(t_f) = -\frac{\partial E}{\partial t_f} \tag{9}$$

The name ‘‘HzMAT’’ is also composed of the initials of main functions’ name.

The advantage of the HzMAT optimal-control algorithm is that the influence of the quadratic cost and running time is considered in this optimal control. The final control result minimizes both the cost and running time. However, the HzMAT optimal-control algorithm requires comprehensive computation. Adding the HzMAT optimal-control algorithm result to the open-loop controller and real-time system makes two types of HzMAT optimal controllers. Overall, the HzMAT is an ideal algorithm for a position-to-position time-optimal system. The meaning of some symbols in the HzMAT algorithm has been explained in **Table 3**.

**Table 3.** Table of proximal variable definitions.

Variables	Definitions	Valuables	Definitions
$H$	Hamiltonian	$F$	Dynamic cost function
$\lambda, v$	Lagrange multipliers	$f$	Dynamic constraint
$\lambda_x$	Lagrange multiplier for $x$	$\dot{\lambda}_x$	Lagrange multiplier derivative for $x$
$x$	System states	$u$	Control (a function of time)
$\bar{E}$	Total end-point cost functional	$E$	Static cost function
$e$	Static constraint function	$t_f$	Final time

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

## 2. Materials and methods

### 2.1. The HzMAT optimal to the rest-to-rest orientation system

Based on Equation (3), the quadratic cost function is considered, and we identify that the  $F$  in Equation (4) represents a dynamic cost in the system that is generated by the quadratic cost. Therefore, the dynamics, or running cost, presents the link between the quadratic cost and the dynamic system constraint. Equation (10) is the quadratic cost function:

$$F = \frac{1}{2}u^2 \tag{10}$$

The  $f$  in the Hamiltonian condition is a dynamic constraint in the system. The external control in the dynamics depends on the cost, state, and time. The costate represents the sensitivity of the system to perturbations, and the states represent the current physical state of the system. Additionally, time is also a critical factor in external control since time determines the duration of the control action and influences the overall performance of the system. Equation (11) displays the relationship between the dynamics and these factors.

$$\dot{x} = f(x, u, t) = f(x, \lambda, t) \tag{11}$$

Equation (11) provides insight into the relationship between the dynamic’s constraints and states. Equation (12) shows that the detail of the dynamic constraint, represented by  $f$  in the Hamiltonian,

influences the evolution of the state variables over time. The dynamic constraint is simplified by expressing the constraints as a set of equations that maintain the simplest structure, consisting of a first derivative and a derivative-free function.

$$\{\bar{x}\} = \begin{Bmatrix} \theta \\ \omega \end{Bmatrix} \tag{12}$$

$$\{\dot{\bar{x}}\} = \begin{Bmatrix} \dot{\omega} \\ \tau/I \end{Bmatrix} = \begin{Bmatrix} \dot{\omega} \\ u \end{Bmatrix} \tag{13}$$

For each state variable in the system, there is a corresponding costate variable, which represents the sensitivity of the system’s performance to perturbations in that state variable. Therefore, the Hamiltonian condition function is a combination of dynamic cost and dynamic constraints by Lagrangian, which include both the state and costate variables.

$$H = F + \lambda^T f(x, u) = \frac{1}{2}u^2 + \{\lambda_\theta \quad \lambda_\omega\} \begin{Bmatrix} \dot{\omega} \\ u \end{Bmatrix} = \frac{1}{2}u^2 + \lambda_\theta \dot{\omega} + \lambda_\omega u \tag{14}$$

Based on Equation (5), the result of the partial derivative of the Hamiltonian condition functions with respect to the control value is calculated.

$$u + \lambda_\omega = 0 \tag{15}$$

Therefore, referring to Equation (6), the adjoint equations in this condition are available. Equations (15) and (16) obtained by taking the partial derivative of the  $\theta$  and  $\omega$  using the Hamiltonian function are:

$$\frac{\partial H}{\partial \theta} = 0 = -\dot{\lambda}_\theta \tag{16}$$

$$\frac{\partial H}{\partial \omega} = \lambda_\theta = -\dot{\lambda}_\omega \tag{17}$$

**Table 4** shows the definitions of some special symbols in the process of solving a practical issue by the HzMAT algorithm.

**Table 4.** Table of proximal variable definitions.

Variable	Definition
$\tau$	The torque from the controller
$\lambda_\theta$	The Lagrangian about $\theta$
$\lambda_\omega$	The Lagrangian about $\omega$
$\dot{\lambda}_\theta$	The derivative about the $\theta$ lagrangian
$\dot{\lambda}_\omega$	The derivative about the $\omega$ lagrangian

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

Equations (15) and (16) establish the relationship between the Lagrangian and constant through integral operation, while Equations (17) and (18) demonstrate the obtained result:

$$\lambda_\theta = a \tag{18}$$

$$-\lambda_\omega = at + b = u \tag{19}$$

where  $a$  and  $b$  are both constant.

By combining Equations (17) and (18), a straightforward relationship between control and time can be derived in Equation (20).

$$u = \dot{\omega} = at + b \tag{20}$$

Integrate these equations and obtain the optimal control for both velocity v and angle  $\theta$ .

$$\omega = \frac{1}{2}at^2 + bt + c \tag{21}$$

$$\theta = \frac{1}{6}at^3 + \frac{1}{2}bt^2 + ct + d \tag{22}$$

where  $a$ ,  $b$ ,  $c$ , and  $d$  are unknown constants.

Equations may contain unknown constants; therefore, constructing a solvable equation needs to be based on time and control. The solution to the equation requires the specification of boundary conditions. After satisfying these conditions, a time-optimal control under the given conditions is available.

Due to the boundary condition in Equation (2), the control function with respect to time from Equation (19) to (21) is displayed in Equations (23)–(25):

$$u = 6 - 12t \tag{23}$$

$$\omega = -6t^2 + 6t \tag{24}$$

$$\theta = -2t^3 + 3t^2 \tag{25}$$

**Table 5** explains some values in the results and it proves that the result of solving a practical issue with the HzMAT depends solely on time.

**Table 5.** Table of proximal variable definitions.

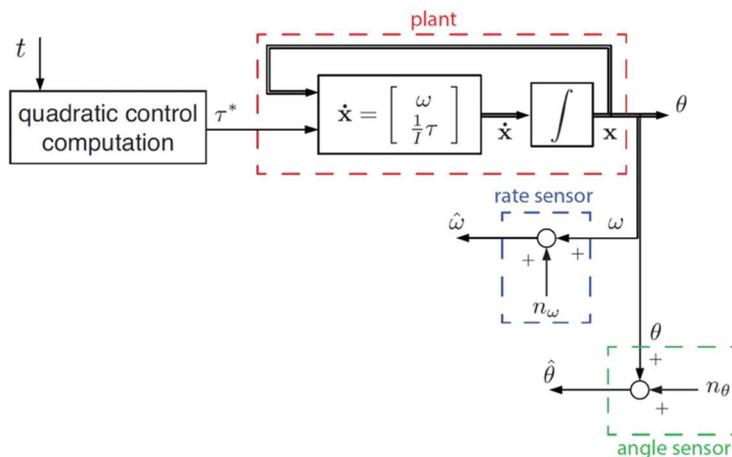
Variable	Definition
$t$	Time (dimensionless)
$a, b, c$	Constants of integration

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

## 2.2. The introduction of six control architectures

### 2.2.1. Open loop controller with the result of the HzMAT optimal analysis

By utilizing the optimal analysis result from the HzMAT with equation (22) as input, the ideal output for both theta and angular velocity can be derived through double integrator translation control. The control architecture is depicted in **Figure 2**:



**Figure 2.** Open-loop controller structure.

### 2.2.2. P+V feedback controller

Due to the feedback system, an ideal target (shown in **Figure 3**) is set and the error between the target and the output theta can be calculated. The error is then multiplied by  $K_p$  and added to  $K_v$  times the output speed, forming the system input. After passing through the double integrator, the system provides output values for both theta and speed, which are then fed back into the input, completing the feedback loop.

The values of  $K_p$  and  $K_v$  used in this system are tuned based on the analysis of oscillation. the system's steady state can be set as  $K_p = \omega_n^2$  and  $K_v = 2 \times \zeta \times \omega_n$ . Where  $\omega_n$  is the angular speed obtained from oscillation analysis, and  $\zeta$  is a parameter related to the system's running time.

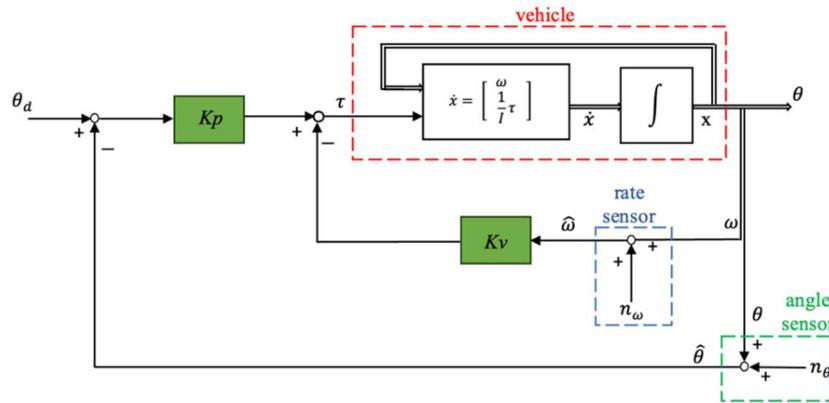


Figure 3. P+V feedback controller structure.

Table 6 explains the variable definitions in Figures 2 and 3.

Table 6. Table of proximal variable definitions for Figures 2 and 3.

Variable	Definition
$\theta_d$	The desired angle
$\theta$	The position output
$K_p$	A proportional value for P+V control
$K_v$	A proportional value for P+V control
$x$	System states
$\dot{x}$	The derivative of states
$\omega$	The velocity output
$\hat{\omega}$	The actual velocity (the velocity output with the noise)
$\hat{\theta}$	The actual angle (the position output with the noise)
$n_\theta$	The noise from the position sensor
$n_w$	The noise from the velocity sensor

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

### 2.2.3. Real-time optimal controller (RTOC)

RTOC is a real-time controller (shown in **Figure 4**) that utilizes the HzMAT optimization with different boundary conditions. The HzMAT analysis can provide reference equations like Equations (20)–(22). Various final boundary conditions can help to solve the values of a, b, c, and d. These values

will be utilized to determine the control  $u$  and subsequently calculate the output position and speed of the system. But in this manuscript, time, position, and velocity limitations keep the same for easy comparison.

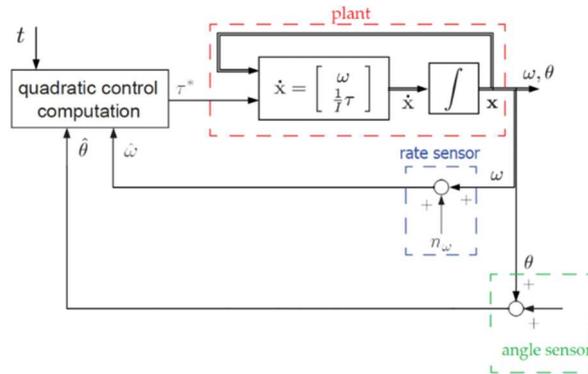


Figure 4. RTOC controller structure.

### 2.2.4. P+V feedback control with double-integrator patching filter

In the P+V feedback controller, a patching filter is introduced to optimize the basic target and enhance the system’s input to approach the ideal target. In the patching filter, there is a simple double integrator (shown in **Figure 5**) that closely represents the dynamics of the open-loop plant. The  $K_p$  and  $K_v$  parameters remain unchanged.

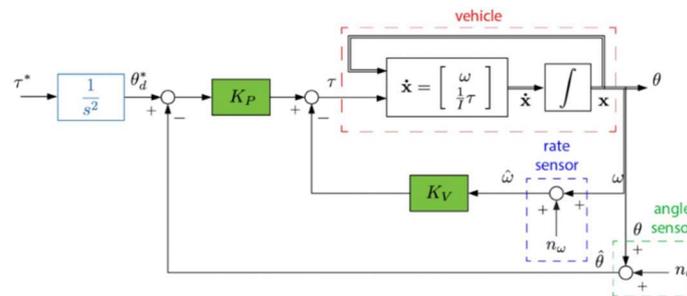


Figure 5. P+V feedback controller with double integrator patching filter.

The meanings of the specific variables appearing in the **Figures 4** and **5** are explained in **Table 7**.

Table 7. Table of proximal variable definitions for **Figures 4** and **5**.

Variable	Definition
$s$	Complex variable for Laplace Transform
$\theta_d^*$	The desired transformed angle
$\tau^*$	The desired torque input
$\omega$	The velocity output
$\theta$	The position output
$\hat{\omega}$	The actual velocity (the velocity output with the noise)
$\hat{\theta}$	The actual angle (the position output with the noise)
$n_\theta$	The noise from the position sensor
$n_\omega$	The noise from the velocity sensor

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

### 2.2.5. Gain-tuning P+V controller with double-integrator patching.

This control architecture still utilizes the P+V feedback control structure, complemented by a double integrator patching filter. However, the  $K_p$  and  $K_v$  parameters have been updated with brand-new values that are tested to improve the system's ability to approach the ideal target. After careful selection and tuning, the  $K_p$  and  $K_v$  are set to 400 and 3, respectively.

### 2.2.6. P+V controller with Control law inversion patching filter.

This controller architecture is like the previous structure, with the primary difference being the use of a control law inversion patching filter. To achieve the HzMAT optimal control with the P+V controller, the input must be designed to be close to the final output.

**Table 8** explains the variable meanings in the process of deriving the law inversion.

**Table 8.** Variables definitions.

Variables	Definitions	Variables	Definitions
$\theta_d$	The desired angle	$X_d(s)$	The input of the control structure (angle)
$\theta$	The actual angle	$U^*(s)$	The output of the control structure (control)
$u^*$	The desired control	$K_p$	A proportional value for P+V control
$u$	The actual control	$K_v$	A proportional value for P+V control

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

As a reminder, the closed-loop control law is displayed in the Equation (25):

$$u = K_p(\theta_d - \theta) - K_v\omega \tag{26}$$

Also, Equation (25) can be written as Equation (26) and (27). Equation (27) represents the ideal outcome of Equation (26).

$$\theta_d = \frac{u}{K_p} + \theta + \frac{K_v}{K_p}\omega \tag{27}$$

$$\theta_d(t) = \frac{u^*}{K_p} + \theta^*(t) + \frac{K_v}{K_p}\omega^*(t) \quad \forall t \in [t_0, t_f] \tag{28}$$

For the double integrator plant, Equation (28) is the same as Equation (27):

$$\theta_d(t) = \frac{1}{K_p}(u^* + K_v \int u^*(t)dt + K_p \iint (u^*(t)dt)dt) \tag{29}$$

Equation (28) may be written in the s-domain based on Laplace Transform as

$$X_d(s) = \frac{1}{K_p}(U^*(s) + \frac{K_v}{s}U^*(s) + \frac{K_p}{s}U^*(s)) \tag{30}$$

Therefore, an alternative patching filter is shown in the Equation (30):

$$\frac{X_d(s)}{U^*(s)} = \frac{1}{K_p} \left[ \frac{s^2 + K_v s + K_p}{s^2} \right] \tag{31}$$

The control structure is displayed in **Figure 6**:

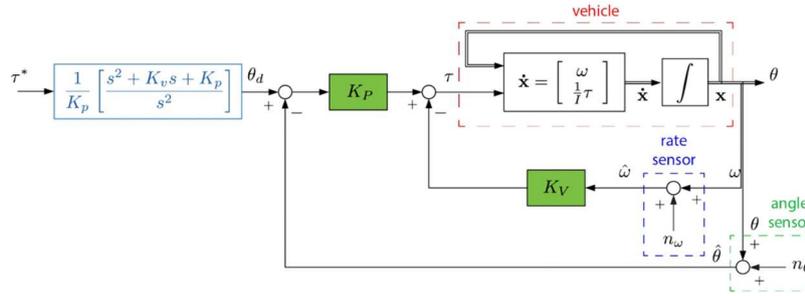


Figure 6. P+V controller with control law inversion patching filter structure.

Table 9 lists some variable definitions in Figure 6.

Table 9. Table of proximal variable definitions for Figure 6.

Variable	Definition
$s$	Complex variable for Laplace Transform
$K_p$	A proportional value for P+V control
$K_v$	A proportional value for P+V control
$\hat{\omega}$	The actual velocity (the velocity output with the noise)
$\hat{\theta}$	The actual angle (the position output with the noise)
$n_\theta$	The noise from the position sensor
$n_\omega$	The noise from the velocity sensor

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

### 2.3. The MATLAB® solver selection

In the Simulink system, the solver plays a crucial role in determining the trajectory of the position and speed variables. Thereby, the solver affects the accuracy and precision of the results. Therefore, selecting the appropriate solver and step size carefully for the analysis of the controllers' performance is essential.

For the purposes of this manuscript, using an open-loop controller to evaluate the solver is convenient. When ode1 with a step size of 0.01 is tested, the resulting trajectory appears in Figure 7a. To obtain more accurate results, using a different solver as a comparison is significant, such as ode4 with a step size of 0.01, whose trajectory is displayed in Figure 7b. To further improve the accuracy and precision of the results, reducing the step size to 0.0001 is also an option. The objective trajectory with a step size of 0.0001 is shown in Figure 7c.

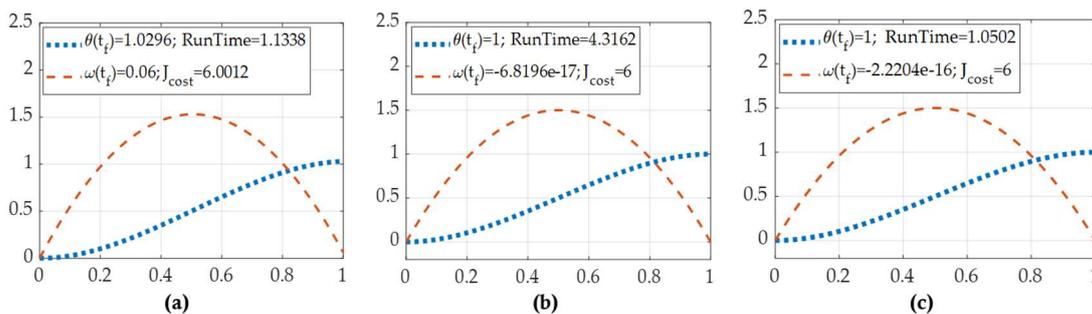


Figure 7. MATLAB® integration solver iteration with time on the abscissa and position and velocity on the ordinant. (a) the trajectory under the ode1 and step 0.01. (b) the trajectory under the ode4 and step 0.01. (c) the trajectory under the ode4 and step 0.0001.

The conclusion based on the trajectory of ode1 with a step size of 0.01 is obvious: the final angular speed fails to satisfy the rest-to-rest condition. As a result, ode1 may not be the optimal choice for our analysis. After switching to the ode4 solver, the final boundary condition appears to be better satisfied in **Figure 7b**. Switching to the ode4 solver with a step size of 0.0001 did not significantly improve the final boundary condition. However, the running time increased significantly. **Table 10** presents the results of the usage of different step sizes and solvers.

**Table 10.** The performances of the open-loop system with different solvers and step length.

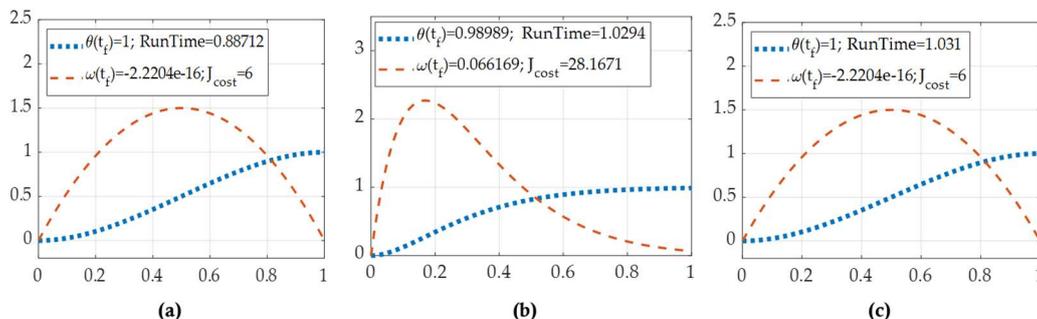
Solver and step length	End position	End velocity	Computational runtime	Quadratic cost (vehicle fuel usage)
Ode1 solver with 0.01 step length	1.023	0.060	1.1338	6.001
Ode4 solver with 0.01 step length	1.000	$2.220 \times 10^{-16}$	1.0502	6.000
Ode4 solver with 0.0001 step length	1.000	$6.820 \times 10^{-17}$	4.3162	6.000

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

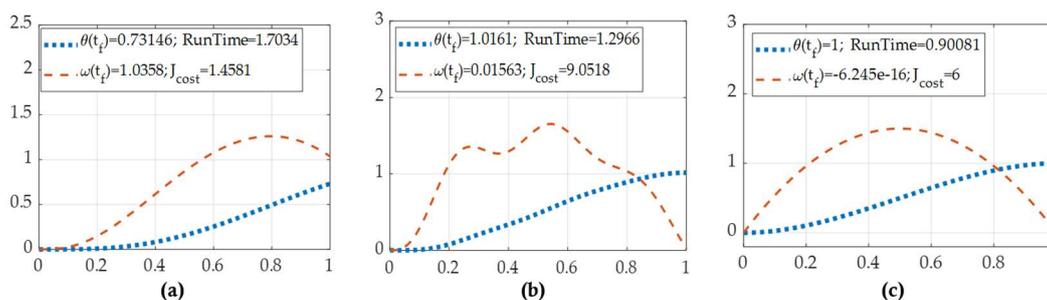
To find a compromise between accuracy and efficiency, using the ode4 solve with a step size of 0.01 is an ideal option for the simulation. This should provide reasonably accurate results while also maintaining a manageable running time.

### 3. Trajectory of the result from six architectures without any uncertainty

Once the various control structures in Simulink are implemented, the resulting trajectories can be examined. **Figures 8a–c** and **Figures 9a–c** show the trajectories obtained from six control systems without noise. **Figures 8a–c** are associated with the open loop, P+V controller, and the RTOC while **Figures 9a–c** are from the P+V controller with a double-integrator patching filter, the tuning parameter P+V controller with double integrator filter and the P+V controller with a law inversion filter.



**Figure 8.** MATLAB® simulations with no noise with time on the abscissa and position and velocity on the ordinant. (a) the trajectory from the open loop; (b) the trajectory from the P+V controller; (c) the trajectory from the RTOC.



**Figure 9.** MATLAB® simulations with no noise with time on the abscissa and position and velocity on the ordinant. (a) the trajectory from the P+V controller with double-integrator patching filter; (b) the trajectory from the tuning parameter P+V controller with double integrator filter; (c) the trajectory from the P+V controller with law inversion filter.

**Table 11** shows some essential indicators for comparing of the control result of six different control structures without noise.

**Table 11.** The performances of six control architectures without errors.

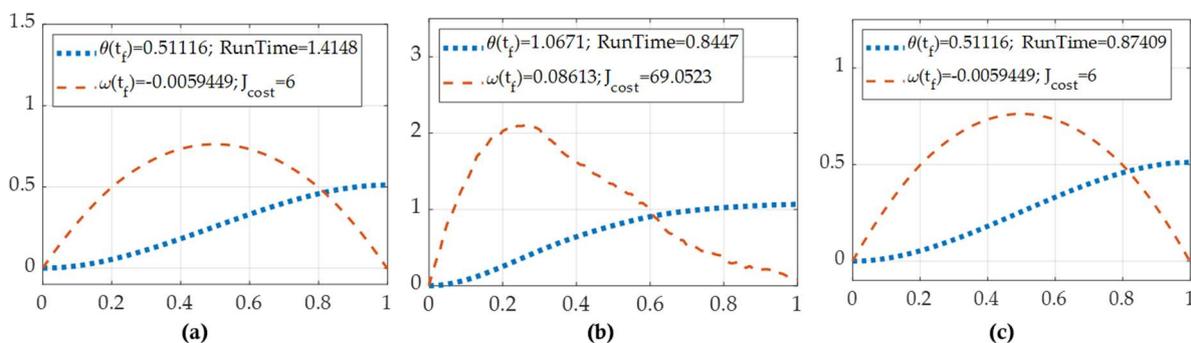
Vehicle control method	End position	End velocity	Computational runtime	Quadratic cost (vehicle fuel usage)
Open-loop optimal control, ideal case	1.000	$-6.189 \times 10^{-17}$	4.000	6.000
Classical proportional + velocity feedback	0.990	0.066	3.449	28.167
Real-time optimal control	1.000	$-6.820 \times 10^{-17}$	3.896	6.000
P+V control with double-integrator filter	0.732	1.036	3.483	1.458
Tunning parameter P+V control with filter	1.016	0.016	3.475	9.052
P+V control with law inversion filter	1.000	$4.857 \times 10^{-17}$	3.962	6.000

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

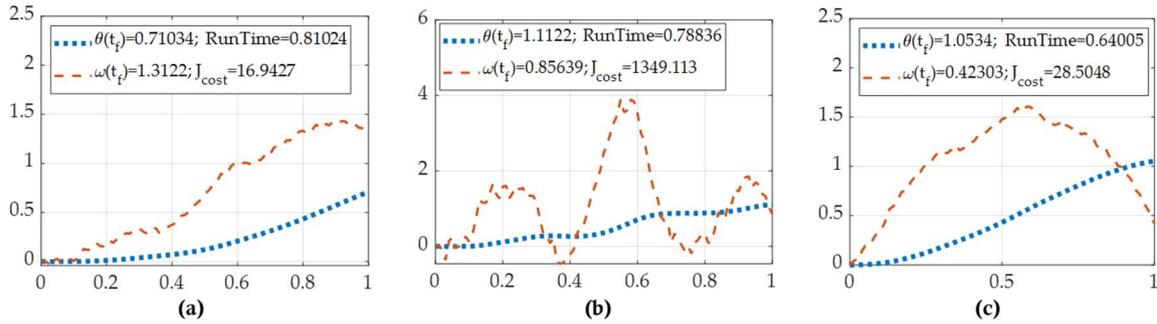
Once the trajectories from different control structures are generated, the conclusion is obvious. Firstly, the HzMAT optimal controller provides significantly better precision and accuracy than the P+V controller. However, the HzMAT controller may cause a slight overshoot in the controlling, whether it is the open-loop controller or RTOC, as evidenced by the negative final boundary value of velocity. Secondly, by comparing the three figures of the P+V controller, the P+V controller with better tuning parameters of  $K_p$  and  $K_v$  and a suitable patching filter can provide accurate enough results. Thirdly, surprisingly, the open-loop controller achieved accuracy beyond our expectations, possibly due to the HzMAT optimal-control algorithm. It is worth noting that the ideal quadratic cost is 6, and the two types of HzMAT controllers performed better in terms of cost optimization, while the P+V controller incurred higher costs, even though they produced very similar trajectories. Finally, regarding running time, all P+V controller architectures had a shorter running time than the two types of HzMAT controllers.

#### 4. Analysis of six control architectures with some uncertainties

After the uncertainties in inertia and sensor noises from the position and velocity sensors are added to the system, the trajectory demonstrates the distinct characteristics and robustness of different control architectures. **Figures 10a–c** and **11a–c** illustrate the trajectories obtained from the six control structures with noise. The trajectories from the open loop, the traditional P+V controller, and the RTOC are shown in **Figure 10**. And the trajectories from the three new types of P+V controllers are shown in **Figure 11**.



**Figure 10.** MATLAB® simulations with noise with time on the abscissa and position and velocity on the ordinant. **(a)** the trajectory from the open loop with noise; **(b)** the trajectory from the P+V controller with noise; **(c)** the trajectory of RTOC with noise.



**Figure 11.** MATLAB® simulations with noise with time on the abscissa and position and velocity on the ordinant. **(a)** the trajectory from the P+V controller with double-integrator patching filter with noise; **(b)** the trajectory from the tuning parameter P+V controller with double integrator filter with noise; **(c)** the trajectory from the P+V controller with law inversion filter with noise.

**Table 12** lists some essential indicators for comparing the control results of six different control structures with noise.

Surprisingly, despite the uncertainties, the open-loop controller’s trajectory remains like the condition without errors, which shows good robustness. It’s possible that only the uncertainty of inertia can affect the open-loop system, but its impact is limited. However, the theta result has a huge gap between the correct result set, which means the open-loop controller is not suitable for the real environment. Also, uncertainties significantly affect the theta result of RTOC’s HzMAT controller. The P+V controllers’ trajectories show oscillation and are not as smooth as those from the HzMAT controller. Compared to the no-error system figures, a significant improvement in the performance of P+V controllers is that they maintain high accuracy under the influence of errors. But the self-tuning patching filter amplified the errors’ impact, rendering the figure useless. The reason is poor tuning parameters, and this result presents the significance of tuning. In conclusion, a well-set P+V controller in error may yield a similar output to the HzMAT controller without errors. In terms of cost, the HzMAT control architectures are less expensive compared with the P+V controller.

**Table 12.** The performances of six control architectures with errors.

Vehicle control method	End position	End velocity	Computational runtime	Quadratic cost (vehicle fuel usage)
Open-loop optimal control, ideal case	0.511	0.00594	4.107	6.000
Classical proportional + velocity feedback	1.048	0.02946	3.763	63.121
Real-time optimal control	0.511	-0.00595	4.143	6.000
P+V control with double-integrator filter	1.040	-0.02946	3.609	63.121
Tunning parameter P+V control with filter	0.961	7.0317	3.617	20834.110
P+V control with law inversion filter	1.048	-0.02946	3.625	63.121

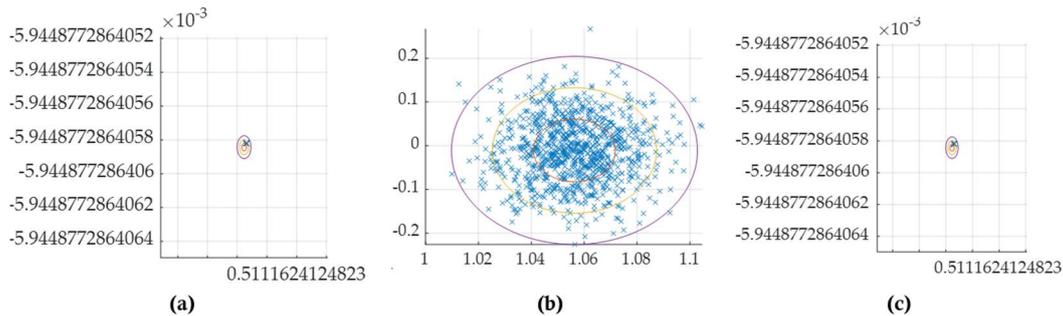
Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

## 5. Monte Carlo analysis

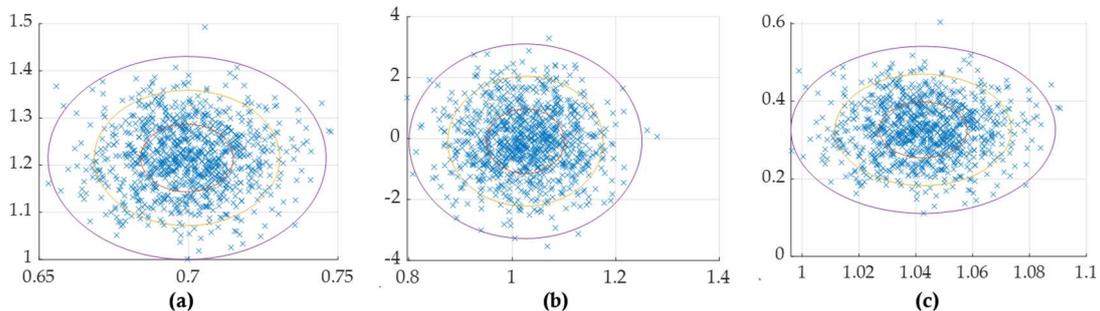
Monte Carlo analysis is a computational algorithm that uses random samples or parameters to generate a result in a target area. This Monte Carlo figure, also known as the distribution diagram, provides evidence of the impact of some specific parameters.

In this analysis, a large number of random error samples that conform to the normal distribution were selected from position and velocity sensors. By incorporating the generated values from the average

distribution into the seed of Simulink, the simulation results were made predictable and reproducible. Subsequently, Monte Carlo charts for position and velocity were generated to analyze the impact of the two sensor noises. The six images in **Figures 12** and **13** correspond to Monte Carlo plots generated from 1000 sets of samples for six different control structures in a real-noise environment. The real noise environment includes angle and velocity sensors, noise that has a normal distribution, as well as variations in inertial moments. The order of the control systems also follows the order of the images, namely the open loop, the traditional P+V controller, the RTOC, the P+V controller with a double-integrator patching filter, the tuning P+V controller with filter, and the P+V controller with a law-inversion patching filter.



**Figure 12.** MATLAB® simulations with noise with the angle on the abscissa and the angular velocity on the ordinant. **(a)** Monte Carlo figure from the open loop; **(b)** Monte Carlo figure from the single P+V controller; **(c)** Monte Carlo figure from the RTOC.



**Figure 13.** MATLAB® simulations with noise with the angle on the abscissa and the angular velocity on the ordinant. **(a)** Monte Carlo figure from the P+V controller with double-integrator patching filter; **(b)** Monte Carlo figure from the tuning parameter P+V controller with double integrator filter; **(c)** Monte Carlo figure from the P+V controller with law inversion filter.

**Table 13** lists some important indicators which can be used to analyze the Monte Carlo plots.

**Table 13.** The performances of six control architectures with errors in Monte Carlo analysis.

Vehicle control method	Mean position error	Position error deviation	Mean velocity error	Velocity error deviation
Open-loop optimal control, ideal case	0.511	$1 \times 10^{-13}$	$-5.944 \times 10^{-13}$	$1 \times 10^{-17}$
Classical proportional + velocity feedback	1.056	0.016	-0.010	0.050
Real-time optimal control	0.511	$1 \times 10^{-13}$	-0.511	$1 \times 10^{-16}$
P+V control with double-integrator filter	0.699	0.015	1.222	0.065
Tunning parameter P+V control with filter	1.029	0.069	-0.015	1.133
P+V control with law inversion filter	1.043	0.155	0.328	0.073

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined, master table of definitions is included in the appendices.

Open-loop controller is the least affected, as mentioned previously. All control architectures using P+V controllers are influenced by uncertainties. The Monte Carlo figures for the P+V controller with different filters show that  $K_p$  and  $K_v$  determine the degree of noise impact, and the patching filter is the one influenced by the uncertainty, despite its objective of closing the input to the final output. Surprisingly, the RTOC shows strong robustness. However, it's undeniable that even the RTOC's precision decreases due to uncertainties.

## 6. Discussions

The findings from the model indicate that HzMAT exhibits excellent accuracy and precision under ideal conditions. However, in real-world scenarios with system noise, both HzMAT and P+V controllers have certain drawbacks. Among the P+V controllers, the one with a law inversion filter consistently demonstrates superior performance regardless of the noise level. Other P+V controllers show satisfactory results in noisy environments.

When dealing with a fixed dynamic system without significant errors or real-time adjustment requirements, the open-loop controller with the HzMAT value becomes a preferred option as it remains unaffected by sensor noise and effectively controls the system. Nevertheless, open-loop control lacks a feedback system and isn't suitable for comprehensive dynamic systems.

The gain-tuning patching filter and double integrator filter reveal that  $K_p/K_v$  and the patching filter greatly influence the precision of the P+V controller. Hence, using the ideal of Law inversion as a basis for patching filters is a promising approach. While RTOC takes precedence when sensor noise is minimal or cost optimization is crucial, the P+V controller with a law inversion filter becomes the priority when a simple computation algorithm or a faster controller is required in real-world applications.

Simple P+V controllers or those with a double integrator can still be employed, primarily for their ease of operation. With appropriate  $K_p$  and  $K_v$  tuning, these controllers can also be effectively applied in practical scenarios.

As for patching filters, adjusting the input to approximate the desired output proves to be a beneficial strategy, as patching filters essentially work as feedforward controls that influence the input value. Compared to a single P+V controller, those with double integrator patching filters or control law inversion patching filters exhibit less robustness, despite offering improved precision through patching.

## 7. Conclusion

When it comes to the control of the drone's camera gimbal rotation, the P+V control with the law of inversion patching is the priority in the complex real environment needing high precision. Also, RTOC is suitable for general civilian products that don't need high precision but need robustness. Other combinations of control structures, like adding feedforward patching for RTOC, should be tested and added for the drone camera gimbal control in future research.

## Acknowledgments

The professor and fellow students of the Course MAE 5830 Astronautic Optimization at Cornell University provide valuable insights on optimization control. Jiren Liu provides a drone camera gimbal for research.

## Conflict of interest

The authors declare no conflict of interest.

## Data availability statement

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

## References

1. Shokirov R, Abdujabarov N, Jonibek T, et al. Prospects of the development of unmanned aerial vehicles (UAVs). *Technical Science and Innovation* 2020; 2020(3): 4–8. doi: 10.51346/tstu-01.20.3-77-0069
2. Saha A, Kumar A, Sahu AK. FPV drone with GPS used for surveillance in remote areas. In: Proceedings of the 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN); 3–5 November 2017; Kolkata, India. pp. 62–67.
3. Jain U, Rogers M, Matson ET. Drone forensic framework: Sensor and data identification and verification. In: Proceedings of the 2017 IEEE Sensors Applications Symposium (SAS); 13–15 March 2017; Glassboro, NJ, USA. pp. 1–6.
4. Mogili UR, Deepak BBVL. Review on application of drone systems in precision agriculture. *Procedia Computer Science* 2018; 133: 502–509. doi: 10.1016/j.procs.2018.07.063
5. Reinecke M, Prinsloo T. The influence of drone monitoring on crop health and harvest size. In: Proceedings of the 2017 1st International Conference on Next Generation Computing Applications (NextComp); 19–21 July 2017; Mauritius. pp. 5–10.
6. Almalki FA, Aljohani M, Algethami M, Soufiene BO. Incorporating drone and AI to empower smart journalism via optimizing a propagation model. *Sustainability* 2022; 14(7): 3758. doi: 10.3390/su14073758
7. Quiroz G, Kim SJ. A confetti drone: Exploring drone entertainment. In: Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE); 8–10 January 2017; Las Vegas, NV, USA. pp. 378–381.
8. Koehle K. NASA offers guidance for drone use viewing Antares launch. Available online: <https://www.nasa.gov/wallops/2021/press-release/nasa-offers-guidance-for-drone-use-viewing-antares-launch> (accessed on 11 February 2021).
9. Huang C, Yang Z, Kong Y, et al. Learning to capture a film-look video with a camera drone. In: Proceedings of the 2019 international conference on robotics and automation (ICRA); 20–24 May 2019; Montreal, QC, Canada. pp. 1871–1877.
10. Ansias Carmona E. *Development of Active Camera Stabilization System for Implementation on UAV's* [Bachelor's thesis]. Universitat Politècnica de Catalunya; 2012.
11. Cong Danh N. The stability of a two-axis gimbal system for the camera. *The Scientific World Journal* 2021; 2021: 9958848. doi: 10.1155/2021/9958848
12. Gašparović M, Jurjević L. Gimbal influence on the stability of exterior orientation parameters of UAV acquired image. *Sensor* 2017; 17(2): 401. doi: 10.3390/s17020401
13. Kim M, Byun GS, Kim GH, Choi MH. The stabilizer design for a drone-mounted camera gimbal system using intelligent-PID controller and tuned mass damper. *International Journal of Control and Automation* 2016; 9(5): 387–394. doi: 10.14257/ijca.2016.9.5.37
14. Rajesh RJ, Ananda CM. PSO tuned PID controller for controlling camera position in UAV using 2-axis gimbal. In: Proceedings of the 2015 International Conference on Power and Advanced Control Engineering (ICPACE); 12–14 August 2015; Bengaluru, India. pp. 128–133.
15. Duan Q, Zhou X, He Q, et al. Pointing control design based on the PID type-III control loop for two-axis gimbal systems. *Sensors and Actuators A: Physical* 2021; 331: 112923. doi: 10.1016/j.sna.2021.112923
16. Laššák M, Draganova K, Blišťanová M, et al. Small UAV camera gimbal stabilization using digital filters and enhanced control algorithms for aerial survey and monitoring. *Acta Montanistica Slovaca* 2020; 25(1): 127–137. doi: 10.46544/AMS.v25i1.12
17. Altan A, Hacıoğlu R. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. *Mechanical Systems and Signal Processing* 2020; 138: 106548. doi: 10.1016/j.ymssp.2019.106548
18. Aguilar WG, Angulo C. Real-time model-based video stabilization for microaerial vehicles. *Neural Processing Letters* 2016; 43(2): 459–477. doi: 10.1007/s11063-015-9439-0
19. Kangunde V, Jamisola RS, Theophilus EK. A review on drones controlled in real-time. *International Journal of Dynamics and Control* 2021; 9: 1832–1846. doi: 10.1007/s40435-020-00737-5

20. Altan A, Aslan Ö, Hacıoğlu R. Real-time control based on NARX neural network of hexarotor UAV with load transporting system for path tracking. In: Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT); 25–27 October 2018; Istanbul, Turkey. pp. 1–6.
21. Yoo H, Kim B, Kim JW, Lee JH. Reinforcement learning based optimal control of batch processes using Monte-Carlo deep deterministic policy gradient with phase segmentation. *Computers & Chemical Engineering* 2021; 144: 107133. doi: 10.1016/j.compchemeng.2020.107133
22. Binder K. Applications of Monte Carlo methods to statistical physics. *Reports on Progress in Physics* 1997; 60(5): 487. doi: 10.1088/0034-4885/60/5/001
23. Sands T. Inducing performance of commercial surgical robots in space. *Sensors* 2023; 23(3): 1510. doi: 10.3390/s23031510
24. Mingkhwan E, Khawsuk W. Digital image stabilization technique for fixed camera on small size drone. In: Proceedings of the 2017 Third Asian Conference on Defence Technology (ACDT); 18–20 January 2017; Phuket, Thailand. pp. 12–19.
25. Guermont JL, Nazarov M, Popov B, Tomas I. Second-order invariant domain preserving approximation of the Euler equations using convex limiting. *SIAM Journal on Scientific Computing* 2018; 40(5): A3211–A3239. doi: 10.1137/17M1149961
26. Qin S, Cramer M, Koch C, Serafini A. Optimal control for Hamiltonian parameter estimation in non-commuting and bipartite quantum dynamics. *SciPost Physics* 2022; 13(6): 121. doi: 10.21468/SciPostPhys.13.6.121
27. Breitenbach T, Borzi A. A sequential quadratic Hamiltonian method for solving parabolic optimal control problems with discontinuous cost functionals. *Journal of Dynamical and Control Systems* 2019; 25(3): 403–435. doi: 10.1007/s10883-018-9419-6
28. Jafarizadeh MA, Naghdi F, Bazrafkan MR. Time optimal control of two-level quantum systems. *Physics Letters A* 2020; 384(29): 126743. doi: 10.1016/j.physleta.2020.126743
29. Alpage D, Dörfler F, Lygeros J. An extended Kalman filter for data-enabled predictive control. *IEEE Control Systems Letters* 2020; 4(4): 994–999. doi: 10.1109/LCSYS.2020.2998296
30. Ross IM. *A Primer on Pontryagin's Principle in Optimal Control*. Collegiate Publishers; 2015.
31. Yang ZC, Rahmani A, Shabani A, et al. Optimizing variational quantum algorithms using Pontryagin's minimum principle. *Physical Review X* 2017; 7(2): 021027. doi: 10.1103/PhysRevX.7.021027
32. Wang Y, Wu Z, Chen Y, et al. Research on energy optimization control strategy of the hybrid electric vehicle based on Pontryagin's minimum principle. *Computers & Electrical Engineering* 2018; 72: 203–213. doi: 10.1016/j.compeleceng.2018.09.018
33. Dagdougui H, Ouammi A, Sacile R. Optimal control of a network of power microgrids using the Pontryagin's minimum principle. *IEEE Transactions on Control Systems Technology* 2014; 22(5): 1942–1948. doi: 10.1109/TCST.2013.2293954

## Appendix A

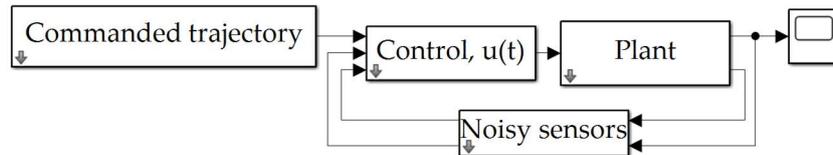


Figure A1. MATLAB® simulation with noise in plant and state and rate sensors.

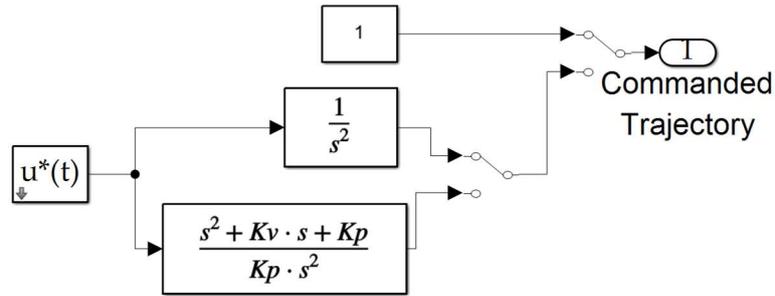


Figure A2. MATLAB® simulation of commanded trajectory options.

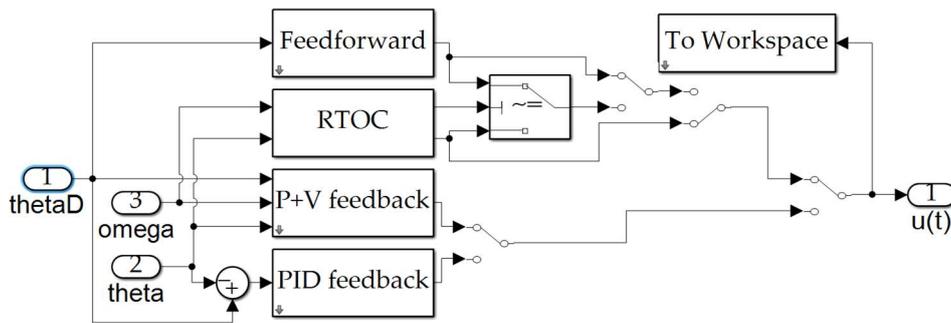


Figure A3. MATLAB® simulation of controller options.

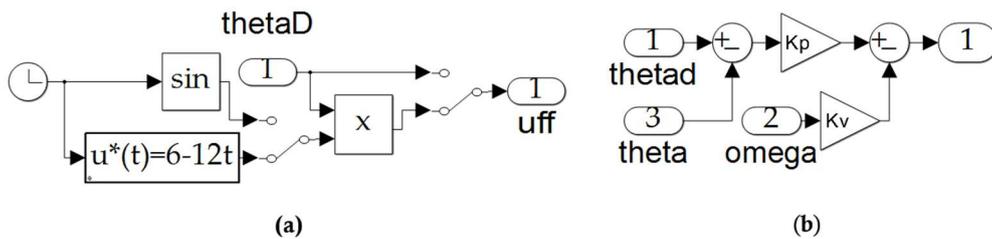


Figure A4. MATLAB® simulation of (a) feedforward control calculation; (b) P+V control.

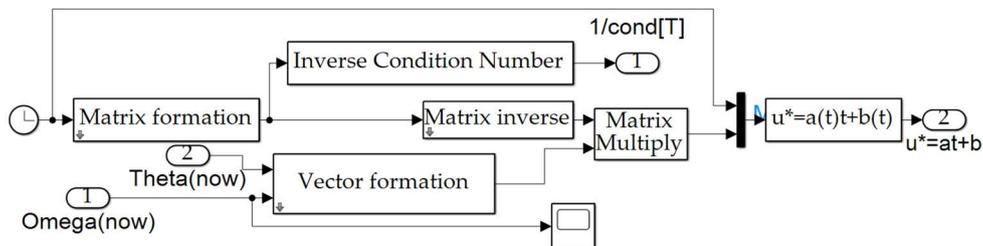


Figure A5. MATLAB® simulation of real-time optimal control.

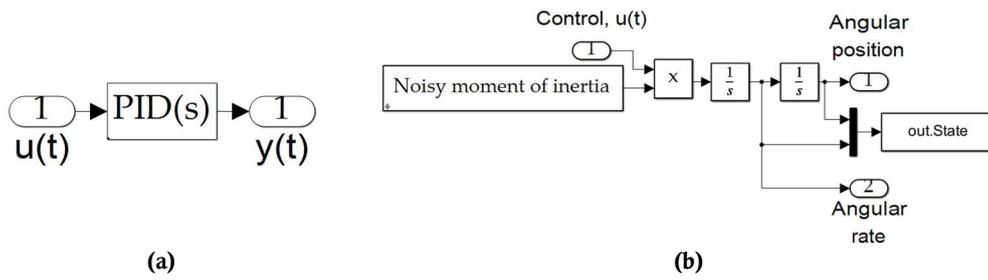


Figure A6. MATLAB® simulation of (a) PID feedback control calculation; (b) Plant.

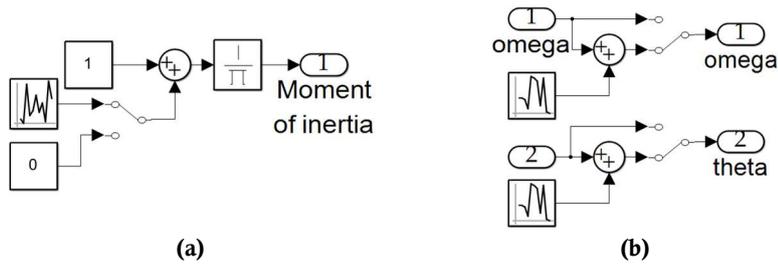


Figure A7. MATLAB® simulation of (a) noisy moment of inertia calculation; (b) noisy sensors.

## Appendix B

Definitions of variables and nomenclature.

**Table B1.** Table of variable definitions

Variables	Definitions	Variables	Definitions
$\theta$	Angle	$J$	the quadratic cost functional
$\omega$	Angular velocity	$F$	The dynamic cost function
$\ddot{\theta}$	Rotational acceleration	$f$	The dynamic constraint
$I$	The inertia of moment	$T_q$	System torque
$t_0$	The initial time	$t_f$	The final time
$u$	Control function on time	$t$	Time(dimensionless)
$H$	The Hamilton equation	$F$	The dynamic cost function
$\lambda, \nu$	The Lagrangian	$f$	The dynamic constraint
$\lambda_x$	The Lagrangian about x	$\dot{\lambda}_x$	The derivative of Lagrangian
$x$	System states	$\dot{x}$	The derivative of states
$\bar{E}$	The total end-point cost	$E$	The static cost function
$e$	The static constraint function	$\tau$	The torque from the controller
$\lambda_\theta$	Lagrange multiplier about $\theta$	$\lambda_\omega$	Lagrange multiplier about $\omega$
$\dot{\lambda}_\omega$	The derivative about the $\omega$ lagrangian	$\dot{\lambda}_\theta$	The derivative about the $\theta$ lagrangian
$s$	Complex variable for Laplace Transform	a, b, c	Constants of integration
$\theta_d^*$	The desired transformed angle	$\hat{\omega}$	The actual velocity (the velocity output with the noise)
$\tau^*$	The desired torque input	$\hat{\theta}$	The actual angle (the position output with the noise)
$n_\theta$	The noise from the position sensor	$n_w$	The noise from the velocity sensor
$\theta_d$	The desired angle	$X_d(s)$	The input of the control structure (angle)
$u^*$	The desired control	$U^*(s)$	The output of the control structure (control)