الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and scientific research
المدرسة الوطنية العليا للتكنولوجيا
National Higher School of Technology
Department of Electrical Engineering And Industrial Computing

---

Final Year Project to Obtain the Diploma of Engineering

Field: Automation

Speciality: Automation and Industrial Computing

Model free control with meta-heuristic optimization for 3-axis camera gimbal stabilization

by
Abdeldjabar Mohammed GUESMIA
Mouncif Mohammed CHAKER

Examination Committee:

Boukhalfa NAIT SLIMANI	M.A.A	Chair
Amar REZOUG	M.C.A	Supervisor
Samir LADACI	Professor	Examiner
Reda DERMOUCHE	M.A.A	Examiner

Algiers, the 04/07/2023

Academic year 2022 –2023

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and scientific research
المدرسة الوطنية العليا للتكنولوجيا
National Higher School of Technology
Department of Electrical Engineering And Industrial Computing

---

Final Year Project to Obtain the Diploma of Engineering

Field: Automation

Speciality: Automation and Industrial Computing

Model free control with meta-heuristic optimization for 3-axis camera gimbal stabilization

by
Abdeldjabar Mohammed GUESMIA
Mouncif Mohammed CHAKER

Examination Committee:

Boukhalfa NAIT SLIMANI	M.A.A	Chair
Amar REZOUG	M.C.A	Supervisor
Samir LADACI	Professor	Examiner
Reda DERMOUCHE	M.A.A	Examiner

Algiers, the 04/07/2023

Academic year 2022 –2023

Dedication

If the beginning of the journey was difficult, its end is to achieve a dream. If the start was a tear, its end is a smile. Every beginning requires an end. And so the years have passed, and the dream is coming true. Thank you, Allah, for giving me the courage to accomplish this task.

I dedicate this work to the one who raised me and fought for me, to the one who illuminated my path. I pray to Allah to prolong his life and may his words remain stars that will guide my path today, tomorrow, and always, my dear father.

And to the one who stayed by my side and was with me in my worst moments, thank you for giving me the strength to complete this journey. And here is the day you've been waiting for, dear mother, your son has finally graduated.

To my brothers, my support in life, and my dear sisters, may God protect them. To the dear Chaker family.

As well as to my dear wife Amel BOUZID, who accompanied me throughout this work and supported me in everything until my dream came true, and her family.

To my friends and all those who have been by my side, to all my dear teachers who never hesitated to help me. I present this work to you, and I hope it will meet your satisfaction.

Mouncif Med.CHAKER

Acknowledgments

We would like to express our deepest gratitude to our supervisor, Dr. REZOUG Amar for his unwavering guidance, invaluable feedback, and constant support throughout the entire process of conducting this research. His expertise and encouragement have been instrumental in shaping this project.

We extend our appreciations to Ms. NAITSLIMANI Boukhalfa who generously dedicated his time and shared his insights, without whom this project would not have started, and with whom we added depth to this work.

We would also like to thank Ing. LAGOUNE Oussama for sharing his expertise and knowledge. The dedicated time he provided for us made this project possible.

Finally to the family, friends and academic community who showed unbelievable support and motivation through this challenging journey.

Grateful we are for this scientific journey.

Contents

I General Introduction **1**

II Three-axis camera gimbal conception and modeling **3**

II.1 Introduction . . . 3
II.2 Gimbal conception . . . 3
II.3 Modeling of the three-axis camera gimbal . . . 8
II.3.1 Assumptions . . . 8
II.3.2 Geometric modeling of the system . . . 8
II.3.2.1 Framing . . . 8
II.3.2.2 Framing validation . . . 9
II.3.2.3 Denavit-Hartenberg parameters . . . 9
II.3.2.4 Transformation matrices . . . 11
II.3.2.5 Framing validation through forward kinematics . . . 11
II.3.2.6 Camera mounting offset . . . 12
II.3.2.7 Position vectors . . . 14
II.3.3 Dynamics modeling . . . 14
II.3.3.1 Inertia matrices . . . 14
II.3.3.2 translational velocity vectors . . . 15
II.3.3.3 Rotational velocity matrices . . . 16
II.3.3.4 Kinetic energy . . . 18
II.3.3.5 Potential energy . . . 18
II.3.3.6 Equations of motion . . . 19
II.3.3.7 Friction effect . . . 20
II.3.3.8 Model validation . . . 21
II.4 Conclusion . . . 24

III Meta-heuristic optimization methods **25**

III.1 Introduction . . . 25
III.2 Generalized Normal Distribution Optimizer . . . 25
III.3 Gray Wolf Optimizer . . . 29
III.4 Ant Colony Optimizer . . . 31
III.5 Conclusion . . . 34

IV Model-free control of three-axis camera gimbal **35**
IV.1 Introduction 35
IV.2 Proposed control 35
IV.2.1 Model-free control 35
IV.2.2 Dynamics estimation 37
IV.2.3 Meta-heuristic optimization of the model-free control 37
IV.3 Advantages and constraints 38
IV.4 Conclusion 39

V Results and discussion **40**
V.1 Introduction 40
V.2 Optimization results 40
V.3 System response results 46
V.4 Robustness test 55
V.5 Conclusion 56

VI Conclusion **57**

VIIReferences **58**

ListofFigures

II.1 Real-world gimbal platforms and applications. 4
II.2 Joints length (in mm). 5
II.3 Electronic board housing. 6
II.4 Electronic board housing. 6
II.5 Motors housings. 7
II.6 Motors housings. 7
II.7 Denavit-Hartenberg based referencing. 8
II.8 Gimbal configuration. 13
II.9 Offset. 13
II.10 A gimbal joint with respect to assumptions 1, 2 and 3 [12]. 15
II.11 Third joint attached frame. 16
II.12 Dynamics model and concept angular positions (No disturbances applied). 21
II.13 Difference in final angular positions illustration. 22
II.14 Difference in center of gravity offset illustration. 22
II.15 Transitional response difference illustration in the first angular position 23
II.16 Transitional response difference illustration in the second angular position 23
II.17 Resonance observation on the models behaviors. 24

III.1 Normally distributed phenomenon for different parameters 26
III.2 Randomly initialized to optimized distribution 26
III.3 Grey wolves hunting illustration [15] 30
III.4 ACO: ant choosing the next step [14]. 32
III.5 ACO: ant paths [14]. 33

IV.1 Control schema 36
IV.2 Optimization schema 38

V.1 GNDO cost function. 42
V.2 GNDO cost function mean. 42
V.3 GWO cost function mean. 43
V.4 ACO cost function mean. 43
V.5 Cost function means (10 trial). 44

V.6 Camera angular velocities with respect to optimized control parameters with applied disturbances 45
V.7 Camera angular velocities with respect to optimized control parameters with applied disturbances 46
V.8 Camera angular velocity with control and without applied disturbances 47
V.9 Control torque without applied disturbances 47
V.10 Camera angular velocities with applied disturbances 48
V.11 Camera angular velocities 49
V.12 Camera angular positions with applied disturbances 49
V.13 Simscape capture 50
V.14 Zoomed-in camera angular positions 51
V.15 Control torque with applied disturbances 51
V.16 Control torque with applied disturbances 52
V.17 Control torque with applied disturbances 52
V.18 Closed-loop estimation output 53
V.19 Closed-loop estimation output 54
V.20 Camera angular velocities. 54
V.21 Smartphone gimbal. 55
V.22 Smartphone angular velocities 56

ListofTables

II.1 DSLR examples 5
II.2 Electronic broad examples 5
II.4 Denavit-Hartenberg Parameters for the gimbal. 9
II.3 Framing validation. 10

III.1 GNDO parameters and terms significance 27
III.2 GNDO algorithm 28
III.3 GWO parameters and variable significance 30
III.4 Grey wolf optimizer algorithm. 31
III.5 ACO parameters and variables significance 33
III.6 ACO algorithm 34

V.1 Shared initial parameters for the optimization algorithms. 40
V.3 Statistics on the optimization results. 41
V.2 Best parameters resulted via optimization. 41

Abbreviations

ACO	***A**nt **C**olony **O**ptimization*
CACSD	***C**omputer **A**ided **C**ontrol **S**ystem **D**esign*
CAD	***C**omputer **A**ided **D**esign*
DSLR	***D**igital **S**ingle **L**ens **R**eflex camera*
GND	***G**eneralized **N**ormal **D**istribution*
GNDO	***G**eneralized **N**ormal **D**istribution **O**ptimization*
GWO	***G**rey **W**olf **O**ptimizer*
SMC	***S**liding **M**ode **C**ontrol*
XML	E***x**tensible **M**arkup **L**anguage*

I. General Introduction

Most common technology among civilians has been used in military or/and industrial applications before it has been introduced to the common people. Videography taking a major part in our modern life reflects how much it has been abused in both mentioned fields. Stabilization is one of the major problems of videography. This problem can be solved with high shutter speed which admittedly wasn't accessible for common use unlike nowadays. In contrast, high shutter speed would invoke a list of problems in videography in the case where the shouter is on move, a man, a drone,etc. In filming industry, a first solution has been the Glidecam, which uses a counter-weight to balance the camera, which fit the purpose for most of their DSLRs and ever so their newer Mirrorlesses. However, the Glidecam still is for handheld use only, it has no way to adapt it with vehicle uses especially drones. Other mechanical solution is Optical Image Stabilization, this one however fits under In-Body Image Stabilization.

Another path was Digital Image Stabilization, which has started by a simple technique that would use the visible frame outside pixels to shift the electronic images accordingly in order to compensate outer vibrations in real time. This one has been developed to an umbrella of numerical methods that fit the purpose, such methods has been reviewed in [1] and some specific approaches of these have been discussed in [2] and [3]. Having no mechanical interfering with the camera, these digital methods are suitable for cameras that are used for vehicle and drone applications. However, they frequently produce video parts that feel anormal to the eye due to their interfering with the image itself. Then, the Three-Axis Gimbal has invaded the field. The gimbal head is a robotic joint that has three co-centered articulations that would insure a full 3-dimensional spherical movement. It has started as two-axis head only and it has been used for sensor stabilization in military missiles for target tracking, an example is given in [4] and [5]. For visual surveillance, two-axis has been enough for two dimensional stabilization but they has not been for visual tracking, so the third axis had to be introduced. Admittedly, it made the theory behind the system quite heavy but it opened the perspectives for gimbal applications. The gimbal head is used primarily for visual surveillance, tracking and precise remote control in military drones, similar applications are found in [6] and [7]. In the filming industry, the gimbal took over, it has been adapted to be mounted on vehicles, drones and for handheld platforms. the gimbal was far lighter than the glidecam and it took less space.

Gimbals have limitations that the scientific community still is working on. The first is that drone gimbals couldn't take on DSLRs due to their non-symmetric shape and heavy weight. The major issue with DSLR gimbals is the offset between the camera center of gravity and the rotation axes. The offset provides more complexity to the system dynamics and a great challenge for the gimbal control. One solution to this issue while keep on the DSLR is an effective model-free control. From the dynamics perspective the complex modeling can be avoided through a combination of computer aided design and computer aided control system design. Also that the scientific community has not taken the handheld gimbals into subject yet as we suppose for it do not represent the heavy engineering behind it compared to the application and being a widely commercial configuration. Moreover, we can't find a source that sets a complete modeling example for this specific system even thou it has been widely used in the last decade.

Gimbal dynamics modeling is similar to robotic arms modeling using Denavit Hartenberg method. Denavit-Hartenberg method involves finding a set of parameters that fully define the position and orientation of a manipulator's structure based on its joint positions. Several models have been suggested to describe kinematic structures, including the Standard and Modified Denavit-Hartenberg (DH) convention [8]. The Denavit-Hartenberg parameters are used to define the kinematic dimensions of a serial chain, consisting of joints with a single degree of freedom [9].

The same DH parameters and references are used as a base knowledge of the system geometry to model its dynamics through Lagrange equations of motion.

Another approach to simulates the gimbal dynamics is through a combination of computer aided design and computer aided control system design. Simscape is the link between both of those. Simscape Multibody provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear. You can model multibody systems using blocks representing bodies, joints, constraints, force elements, and sensors. Simscape Multibody formulates and solves the equations of motion for the complete mechanical system. You can import complete CAD assemblies, including all masses, inertias, joints, constraints, and 3D geometry, into your model. An automatically generated 3D animation lets you visualize the system dynamics [10].

This work aims to develop a gimbal head that would be suitable for both handheld and drone applications, it should be able to take on DSLR cameras and therefore it should adapt to the non-homogenous shape of the camera and its heavy weight, also it should adapt with the different natures of the disturbances between the two applications.

We have started by three-axis camera gimbal conception and modeling. We have then introduced a very recent meta-heuristic optimization algorithm, GNDO, and 2 more algorithms ACO and GWO. We have introduced iPI, a model-free control [11]. This control takes advantage of the classic PI control and an error between the system dynamic and a givin desired dynamic. Than, we have discussed the results of optimizing this control unsing GNDO for the gimbal control. In the results we have discussed the GNDO performance relatively to the other two meta-heuristics, and the control performance.

II. Three-axis camera gimbal conception and modeling

II.1 Introduction

The overall project is model-free control for a 3-axis camera gimbal. This means we have no need for the gimbal dynamics model to synthesis the control. However, we need a gimbal for implementation. The idea is to make a digital twin for the system in order to implement the synthesized control. A digital twin is a replica of the system that has been made using computer aided design. This technique helps avoid complex dynamics modeling. It also avoids imprecise and limited modeling. This section presents our conception of the 3-axes camera gimbal. Nonetheless, we have found gimbal dynamics modeling to be an understudied field, and the resources on this particular subject are very scarce. These circumstances have emphasized the need for a detailed modeling example of such system. Therefore, this section holds in its second part, dynamics modeling of the digital twin. Then, we have established a comparison between the digital twin dynamics and the dynamics model. The comparison has showcased the modeling technique accuracy. Based on the same comparison, we have outlined the reasons for some of the gaps between the real system and dynamics model responses.

II.2 Gimbal conception

The conception is held in SolidWorks, 2020 Professional. It is a well-known Computer Aided Design platform. We have chosen this software among others such Fusin360 because it has a plugin named by Simscape Multibody Link. This tool generates an XML file that can be read through MATLAB. MATLAB generates using the XML file a Simulink system that represents the designed CAD conception. MATLAB has also a built-in Simscape tool that visuals the CAD conception during simulation in 3D, in motion.

The gimbal conception is inspired from handheld filming platforms. We also made it suitable for drone and vehicle platforms. The gimbal head have to be able to mount directly into a handheld extension and a drone platform. It however does not have to be able to mount directly into vehicle platforms. These are very specific in conceptions

and any gimbal would need an extension to be mounted to these vehicle platforms. Figure II.1 shows real-world gimbal platforms we were inspired from along with our gimbal head.



Figure II.1: Real-world gimbal platforms and applications.

The conception is made to support a DSLR camera. Our main focus cameras are under $140.00mm$ in Width, $110.00mm$ in height and $110.00mm$ in depth. This had to be taken into account in the joints dimensions. We have set the joints lengths to accommodate the big volume of this type of cameras as figure II.2 illustrates with according lengths. These joints lengths have been set to keep the center of gravity of the camera as close as possible to the axis of rotation. Yet, we kept a small offset to keep some translational

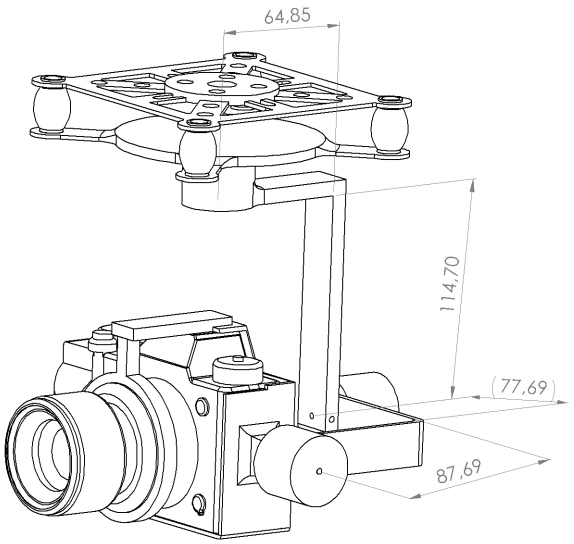


Figure II.2: Joints length (in mm).

movements in the camera. This we did to tackle the offset issue in the gimbal dynamics modeling and in the control. Table II.1 illustrates some DSLRs dimensions we have

Table II.1: DSLR examples

Camera	Width	Height	Depth	Weight
Fujifilm XT-3	132.50 mm	92.80 mm	58.80 mm	539.00 g
Canon 200d	122.40 mm	92.60 mm	69.80 mm	453.00 g
Nikon 50d	133.00 mm	102.00 mm	76.00 mm	620.00 g

kept in mind while designing the system.

The upper part, highlighted in figure II.3 below, shows a dedicated housing for control chip mounting. The housing base is $104.00mm$ as the same figure shows. This we have made to accommodates most of the common control boards. These boards are under $90.00mm$ in length in most of our considerations. Table II.2 below shows some of our

Table II.2: Electronic broad examples

Board	Length	Width
STM32 Nucleo64 STM32F401RE	82.50 mm	70.00 mm
Arduino UNO	68.60 mm	53.40 mm

considerations.

The upper part of the board housing is pierced as highlighted in figure II.4. This have been made to fix the gimbal head in handheld platforms and drone platforms. Many pierces have been made to make the gimbal suitable for a variety of platforms. Other highlighted pierced designs in this part are made to reduce weight.

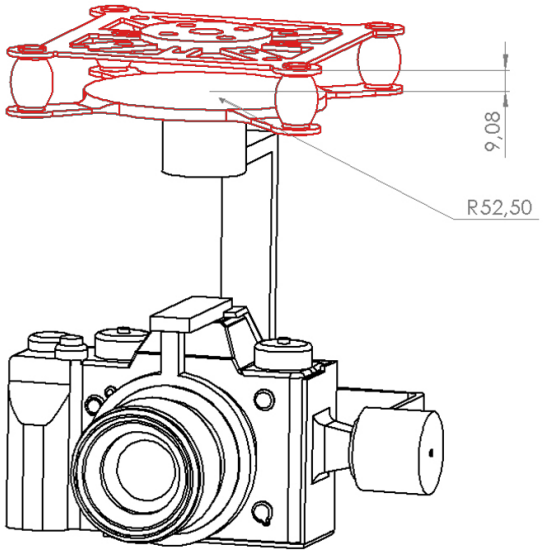


Figure II.3: Electronic board housing.

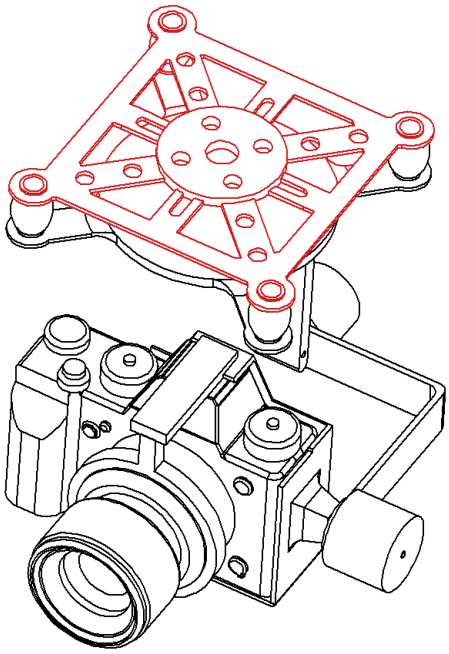


Figure II.4: Electronic board housing.

Figure II.5 highlights the motors housings. These we have dimensioned to accommo-

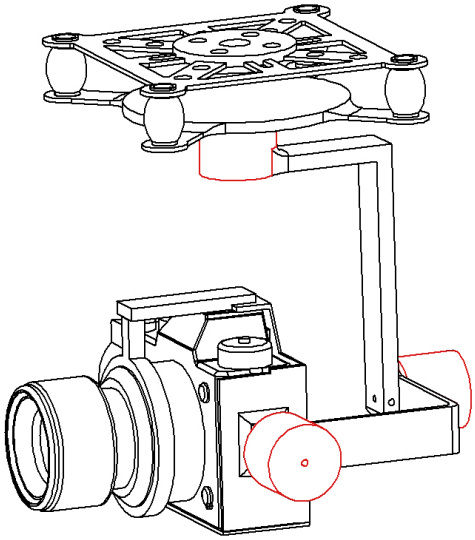


Figure II.5: Motors housings.

dates brushless type motors. They are dimensioned accordingly, $30.00mm$ in length and $36.00mm$ in diameter. We have not been able to accommodate motor digital twins in the conception in-time. So, we had to fill-up the motors housings with material. The same as the housing material. To compensate for the motors weights during motion simulation. Figure II.6 illustrates other main dimensions of the system using SolidWorks drawings.

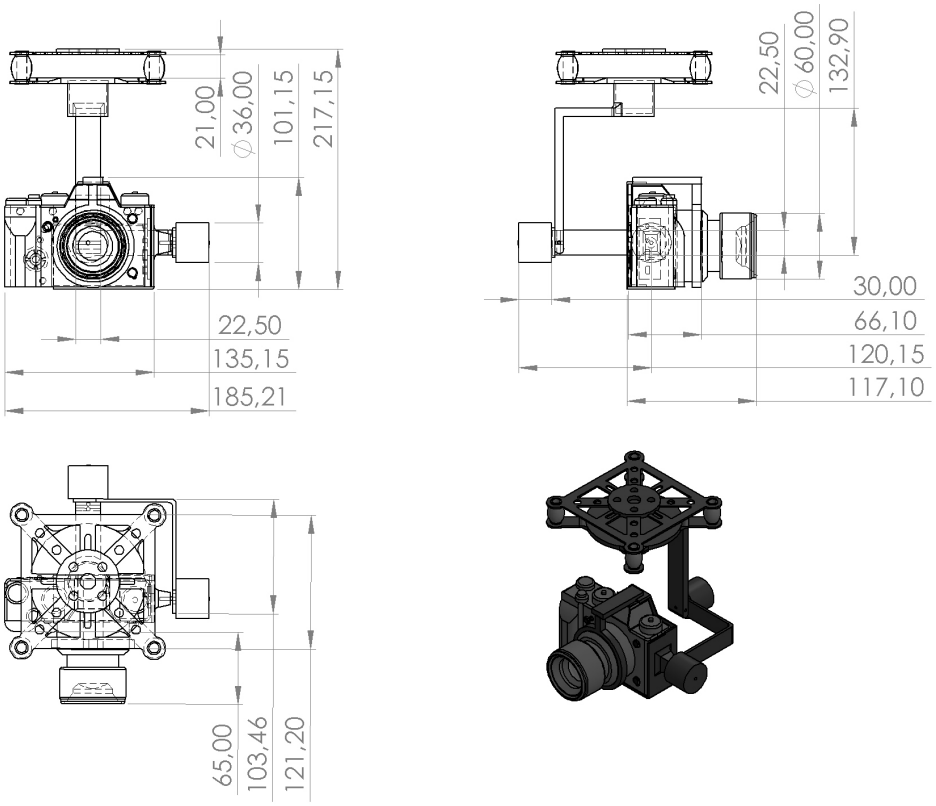


Figure II.6: Motors housings.

II.3 Modeling of the three-axis camera gimbal

II.3.1 Assumptions

Here are few assumptions that we have taken when modeling the system. These assumptions do not compromise the model form, they simplify some of the parameters.

- Assumption 1: the camera is assumed cubical in order to make the inertia its matrix diagonal.
- Assumption 2: the joints are assumed cuboidal also to make their inertia matrices diagonal.
- Assumption 3: the motors kinetics are not taken into account along with their potential energy.

II.3.2 Geometric modeling of the system

II.3.2.1 Framing

In figureII.7 the axis based on Denavit-Hartenberg method have been implemented. These help extract parameters needed for next calculations; model parameters (e.g joints angles), transformation matrices, inertia matrices for each joint and position vectors for each joint center of gravity.

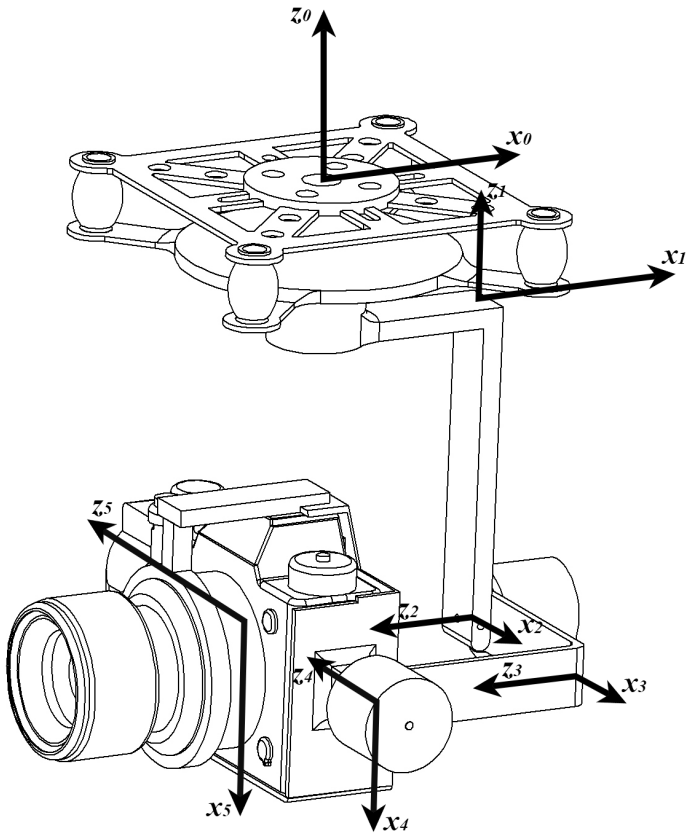


Figure II.7: Denavit-Hartenberg based referencing.

D-H frames satisfy 4 conditions. For two consecutive frames R_i and $R_i + 1$ respectively, the D-H conditions are:

- The axis z_i represents the center of rotation of the rotoid articulation i. For a translational articulation, z_i represents the axis of motion of the same articulation.
- The axis x_{i+1} must be perpendicular to the axis z_i.
- The axis x_{i+1} must intersect the axis z_i.
- The directions of z_i and x_i are chosen arbitrary, but y_i must verify the right hand rule.

D-H method states that an n joints arm should have a set of $n + 1$ frames. A common sense would state to satisfy with 4 frames in our case. However, we couldn't establish the D-H conditions with only 4 frames. So, we had to extend our set to 6 frames in order to satisfy D-h conditions at all consecutive frames.

II.3.2.2 Framing validation

Table II.3 illustrates how our proposed frames verify these conditions.

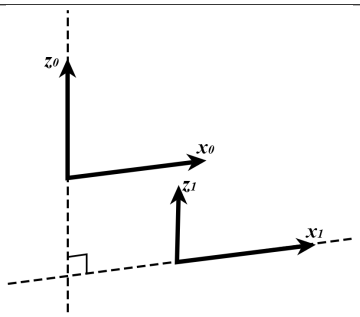
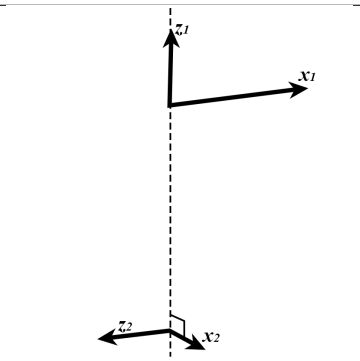
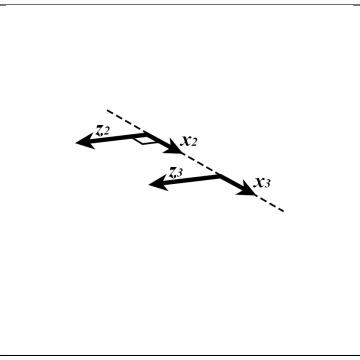
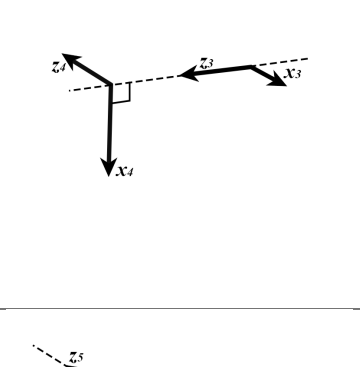
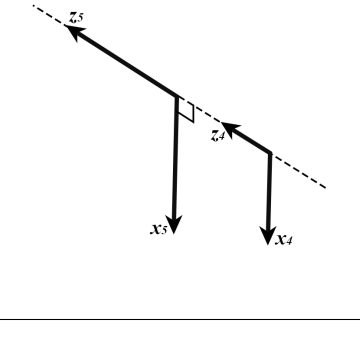
II.3.2.3 Denavit-Hartenberg parameters

According to the established frames we have extracted the D-H parameters illustrated in table II.4. Where a_i and d_i are the length and offset respectively. They represent alternately the joints length except for the initial frame R_0. Where d_i represents the offset of the second joint from the initial frame. α_i and θ_i represent the twist between two consecutive joints and the angle respectively.

Table II.4: Denavit-Hartenberg Parameters for the gimbal.

Reference	a	α	d	θ
1	l_1	0	$-l_0$	θ_1
2	0	$\pi/2$	$-l_2$	$-\pi/2$
3	l_3	0	0	θ_2
4	0	$\pi/2$	l_4	$-\pi/2$
5	0	0	l_5	θ_3

Table II.3: Framing validation.

Frames	Drawing	Validation
R_0 and R_1		x_1 is perpendicular to z_0 x_1 intercepts z_0
R_1 and R_2		x_2 is perpendicular to z_1 x_2 intercepts z_1
R_2 and R_3		x_3 is perpendicular to z_2 x_3 intercepts z_2
R_3 and R_4		x_4 is perpendicular to z_3 x_4 intercepts z_3
R_4 and R_5		x_5 is perpendicular to z_4 x_5 intercepts z_4

II.3.2.4 Transformation matrices

The transformation matrices we denote T_i^j. i represent the current reference frame and j represents the reference frame. These matrices are as follows:

$$T_0^1 = \begin{pmatrix} C_1 & -S_1 & 0 & C_1\,l_1 \\ S_1 & C_1 & 0 & S_1\,l_1 \\ 0 & 0 & 1 & -l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.1)}$$

$$T_0^2 = \begin{pmatrix} S_1 & 0 & -C_1 & C_1\,l_1 \\ -C_1 & 0 & -S_1 & S_1\,l_1 \\ 0 & 1 & 0 & -l_0 - l_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.2)}$$

$$T_0^3 = \begin{pmatrix} C_2\,S_1 & -S_1\,S_2 & -C_1 & C_1\,l_1 + C_2\,S_1\,l_3 \\ -C_1\,C_2 & C_1\,S_2 & -S_1 & S_1\,l_1 - C_1\,C_2\,l_3 \\ S_2 & C_2 & 0 & S_2\,l_3 - l_2 - l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.3)}$$

$$T_0^4 = \begin{pmatrix} S_1\,S_2 & -C_1 & -C_2\,S_1 & C_1\,l_1 - C_1\,l_4 + C_2\,S_1\,l_3 \\ -C_1\,S_2 & -S_1 & C_1\,C_2 & S_1\,l_1 - S_1\,l_4 - C_1\,C_2\,l_3 \\ -C_2 & 0 & -S_2 & S_2\,l_3 - l_2 - l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.4)}$$

$$T_0^5 = \begin{pmatrix} C_3\,S_1\,S_2 - C_1\,S_3 & -C_1\,C_3 - S_1\,S_2\,S_3 & -C_2\,S_1 & C_1\,l_1 - C_1\,l_4 + C_2\,S_1\,l_3 - C_2\,S_1\,l_5 \\ -S_1\,S_3 - C_1\,C_3\,S_2 & C_1\,S_2\,S_3 - C_3\,S_1 & C_1\,C_2 & S_1\,l_1 - S_1\,l_4 - C_1\,C_2\,l_3 + C_1\,C_2\,l_5 \\ -C_2\,C_3 & C_2\,S_3 & -S_2 & S_2\,l_3 - l_2 - l_0 - S_2\,l_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.5)}$$

Where a term S_i represents $sin(\theta_i)$ and C_i represents $cos(\theta_i)$ for compacting reasons.

II.3.2.5 Framing validation through forward kinematics

We have used forward kinematics to verify our framing. Forward kinematics set the joints angles and the end-effecter, represented by the camera, should follow. Unlike inverted kinematics which set the end-effecter configuration and deduce the joints angles. We have set 2 parameters combinations. The first represents arbitrary joints angles. Which we have implemented to the conception through SolidWorks. In order to compare the resulted orientation and position of transformation matrix T_0^5 using the same parameters. The parameters are

$$\theta_1 = \frac{\pi}{2}, \quad \theta_2 = \frac{-\pi}{2}, \quad \theta_3 = \frac{\pi}{2} \text{(II.6)}$$

The resulted transformation matrix is

$$T_0^5 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & l_1 - l_4 \\ 0 & 0 & 1 & -l_3 - l_2 - l_0 + l_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.7)}$$

Using the acctual joints length, T_0^5 becomes

$$T_0^5 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -22.84mm \\ 0 & 0 & 1 & -172.69mm \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.8)}$$

To have an idea on the camera orientation through T_0^5, we need a relation between the last frame axis and the first frame axis. This relation have been found using the rotational matrixR_0^5 extracted fromT_0^5 as follows.

$$T_i^j = \begin{pmatrix} R_i^j & {t_{ri}}^j \\ 0 & 1 \end{pmatrix} \text{(II.9)}$$

$$R_0^5 \times \vec{x_5} = -\vec{y_0} \text{(II.10)}$$

$$R_0^5 \times \vec{y_5} = \vec{x_0} \text{(II.11)}$$

$$R_0^5 \times \vec{z_5} = \vec{z_0} \text{(II.12)}$$

${t_{ri}}^j$ is the translation vector, it represents the coordinates of the j^{th} frame in the i^{th}. Figure II.8 confirms the same results from the conception using SolidWorks.

The second configuration consists of the initial orientation but equal joints length. Of course, this cannot be achieved in the conception configuration. However equal joints length, in theory, should free the camera from any translational movements. This should appear on the transformation matrixT_0^5 as $0mm$ position in both x_0 and y_0 axis, with only the fixed camera height with respect to z_0 axis, in any configuration. Using equal joints length results the following transformations matrix which confirms our framing.

$$T_0^5 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -l_2 - l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{(II.13)}$$

II.3.2.6 Camera mounting offset

In this section we introduce couple more parameters, l_6 and l_7. These represents the camera center of gravity offsets from the z_5 axis. They are along x_5 and y_5 axis re-

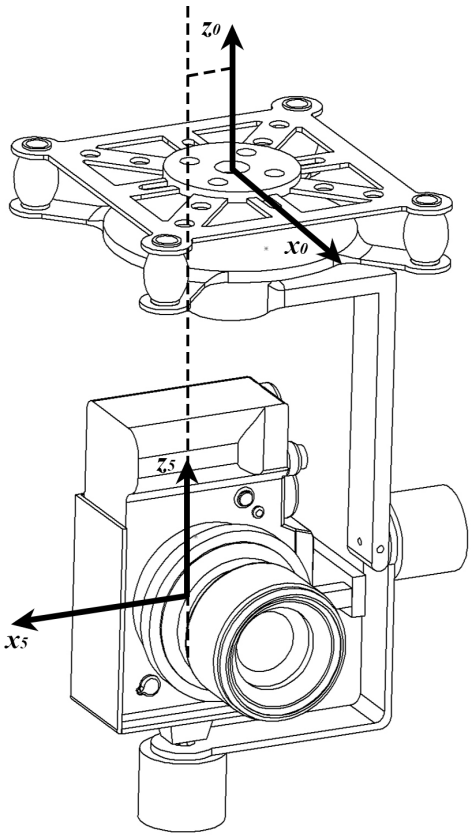


Figure II.8: Gimbal configuration.

spectively. The center of gravity offset is crucial to dynamics modeling as it generates translational movements that are only related the third rotoid articulation. These offsets are exclusive to the camera mounting and cannot be integrated with the parameters we have extracted using D-H method. Figure II.9 illustrates l_6 and the translational movement path resulted by this offset.

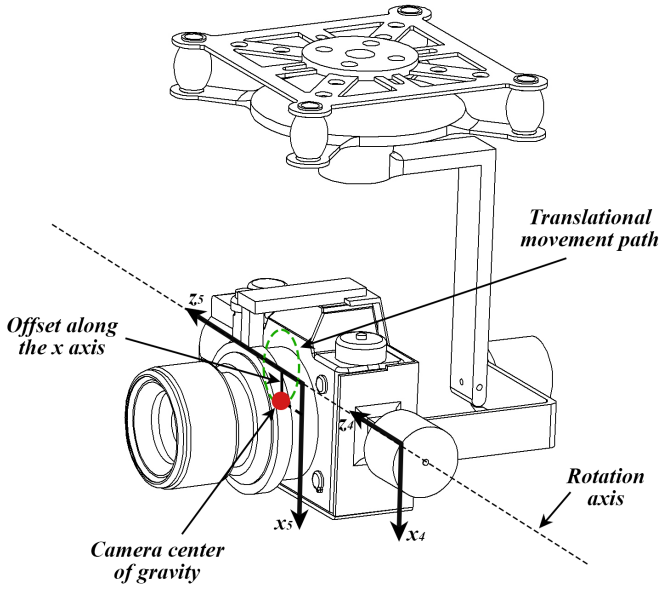


Figure II.9: Offset.

II.3.2.7 Position vectors

We know all joints and the camera have complex movement; a combination of rotational and translational movement. In order to conclude the translational velocities of each joint we need to establish their center of gravities positions. To extract these we first have established the relation the center position and the respected transformation matrix as II.14 dictates. The notation $\vec{r_{c_i/R_j}}$ represents the fact that the vector $\vec{r_{c_i}}$ coordinates are with respect to the frame R_j Then, the position vectors are

$$\vec{r_{c_i/R_0}} = R_0^i \times \vec{r_{c_i/R_5}} + \vec{t_{r0/R_0}^i} \text{(II.14)}$$

$$\vec{r_{c_1}} = \begin{pmatrix} \frac{C_1 l_1}{2} \\ \frac{S_1 l_1}{2} \\ -l_0 \end{pmatrix} \text{(II.15)}$$

$$\vec{r_{c_2}} = \begin{pmatrix} C_1 l_1 \\ S_1 l_1 \\ \frac{l_1}{2} - l_0 - l_2 \end{pmatrix} \text{(II.16)}$$

$$\vec{r_{c_3}} = \begin{pmatrix} C_1 l_1 + \frac{C_2 S_1 l_3}{2} \\ S_1 l_1 - \frac{C_1 C_2 l_3}{2} \\ \frac{S_2 l_3}{2} - l_2 - l_0 \end{pmatrix} \text{(II.17)}$$

$$\vec{r_{c_4}} = \begin{pmatrix} C_1 l_1 - \frac{C_1 l_4}{2} + C_2 S_1 l_3 \\ S_1 l_1 - \frac{S_1 l_4}{2} - C_1 C_2 l_3 \\ S_2 l_3 - l_2 - l_0 \end{pmatrix} \text{(II.18)}$$

$$\vec{r_{c_5}} = \begin{pmatrix} C_1 l_1 - C_1 l_4 - l_6 (C_1 S_3 - C_3 S_1 S_2) - l_7 (C_1 C_3 + S_1 S_2 S_3) + C_2 S_1 l_3 - C_2 S_1 l_5 \\ S_1 l_1 - S_1 l_4 - l_6 (S_1 S_3 + C_1 C_3 S_2) - l_7 (C_3 S_1 - C_1 S_2 S_3) - C_1 C_2 l_3 + C_1 C_2 l_5 \\ S_2 l_3 - l_2 - l_0 - S_2 l_5 - C_2 C_3 l_6 + C_2 S_3 l_7 \end{pmatrix} \text{(II.19)}$$

II.3.3 Dynamics modeling

II.3.3.1 Inertia matrices

First we define the inertia moment of an object as its resistance towards rotation along an axis. I_{xx}, I_{yy} and I_{zz} represent the inertia moment of an object along the $\vec{x}$, $\vec{y}$ and $\vec{z}$ axis respectively. Other moments are the product of each one of these with respect to the plane of rotation. As if the plane of rotation is $(O; \vec{x}, \vec{y})$ the correspondent inertia moment is I_{xy}.

An inertia matrix of a joint, or the camera, is of the form II.20 and it represents an

object resistance towards rotation in a 3 dimensional space.

$$I = \begin{pmatrix} \mathrm{I_{xx}} & \mathrm{I_{xy}} & \mathrm{I_{xz}} \\ \mathrm{I_{xy}} & \mathrm{I_{yy}} & \mathrm{I_{yz}} \\ \mathrm{I_{xz}} & \mathrm{I_{yz}} & \mathrm{I_{zz}} \end{pmatrix} \tag{II.20}$$

Based on Assumption 1, 2 and 3, the inertia matrix becomes

$$I = \begin{pmatrix} \mathrm{I_{xx}} & 0 & 0 \\ 0 & \mathrm{I_{yy}} & 0 \\ 0 & 0 & \mathrm{Izz} \end{pmatrix} \tag{II.21}$$

with I_{xx}, I_{xx} and I_{xx} equal to:

$$I_{xx} = \frac{m}{12}\left(b^2 + c^2\right) \tag{II.22}$$

$$I_{yy} = \frac{m}{12}\left(a^2 + c^2\right) \tag{II.23}$$

$$I_{zz} = \frac{m}{12}\left(a^2 + b^2\right) \tag{II.24}$$

and a, b and c are the width, length and height of the joint as we illustrate in figure II.11

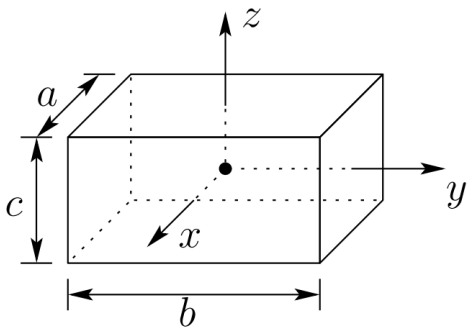


Figure II.10: A gimbal joint with respect to assumptions 1, 2 and 3 [12].

II.3.3.2 translational velocity vectors

Each joint, and the camera, have complex movement. Their movement is a combination of rotation and translation. In order to find the kinetics related to the joints translational movement we need to find the velocities of each center of gravity. These velocities are derived from each center position vectors via differentiation with respect to time. The differentiation is valid as long as the position vectors are represented in the same frame of differentiation. In our case frame R_0, which we did in the previous section.

Otherwise, mobile base theorem should be involved in the differentiation. Mobile base theorem is a vectors differentiation technique that would allow to differentiate a vector represented a frame R_i in another frame R_j. The differentiation process have been done using MATLAB and the results are as follows:

$$\dot{\vec{r}}_{c_1} = \begin{pmatrix} -\frac{S_1\, l_1\, \dot{x_1}}{2} \\ \frac{C_1\, l_1\, \dot{x_1}}{2} \\ 0 \end{pmatrix} \text{(II.25)}$$

$$\dot{\vec{r}}_{c_2} = \begin{pmatrix} -S_1\, l_1\, \dot{x_1} \\ C_1\, l_1\, \dot{x_1} \\ 0 \end{pmatrix} \text{(II.26)}$$

$$\dot{\vec{r}}_{c_3} = \begin{pmatrix} \frac{l_3\,(C_1\, C_2\, \dot{x_1} - S_1\, S_2\, \dot{x_2})}{2} - S_1\, l_1\, \dot{x_1} \\ \frac{l_3\,(C_2\, S_1\, \dot{x_1} + C_1\, S_2\, \dot{x_2})}{2} + C_1\, l_1\, \dot{x_1} \\ \frac{C_2\, l_3\, \dot{x_2}}{2} \end{pmatrix} \text{(II.27)}$$

$$\dot{\vec{r}}_{c_4} = \begin{pmatrix} l_3\,(C_1\, C_2\, \dot{x_1} - S_1\, S_2\, \dot{x_2}) - S_1\, l_1\, \dot{x_1} + \frac{S_1\, l_4\, \dot{x_1}}{2} \\ l_3\,(C_2\, S_1\, \dot{x_1} + C_1\, S_2\, \dot{x_2}) + C_1\, l_1\, \dot{x_1} - \frac{C_1\, l_4\, \dot{x_1}}{2} \\ C_2\, l_3\, \dot{x_2} \end{pmatrix} \text{(II.28)}$$

$$\dot{\vec{r}}_{c_5} = \begin{pmatrix} (C_1\, C_2\, \dot{x_1} - S_1\, S_2\, \dot{x_2})\,(l_3 - l_5) + l_6\,(S_1\, S_3\, \dot{x_1} - C_1\, C_3\, \dot{x_3} + C_1\, C_3\, S_2\, \dot{x_1} + C_2\, C_3\, S_1\, \dot{x_2} - S_1\, S_2\, S_3\, \dot{x_3}) \\ -l_7\,(C_1\, S_2\, S_3\, \dot{x_1} - C_1\, S_3\, \dot{x_3} - C_3\, S_1\, \dot{x_1} + C_2\, S_1\, S_3\, \dot{x_2} + C_3\, S_1\, S_2\, \dot{x_3}) - S_1\, \dot{x_1}\,(l_1 - l_4) \\ (C_1\, C_2\, \dot{x_1} - S_1\, S_2\, \dot{x_2})\,(l_3 - l_5) + l_6\,(S_1\, S_3\, \dot{x_1} - C_1\, C_3\, \dot{x_3} + C_1\, C_3\, S_2\, \dot{x_1} + C_2\, C_3\, S_1\, \dot{x_2} - S_1\, S_2\, S_3\, \dot{x_3}) \\ -l_7\,(C_1\, S_2\, S_3\, \dot{x_1} - C_1\, S_3\, \dot{x_3} - C_3\, S_1\, \dot{x_1} + C_2\, S_1\, S_3\, \dot{x_2} + C_3\, S_1\, S_2\, \dot{x_3}) - S_1\, \dot{x_1}\,(l_1 - l_4) \\ l_6\,(C_3\, S_1\, \dot{x_2} + C_2\, S_3\, \dot{x_3}) + l_7\,(C_2\, C_3\, \dot{x_3} - S_2\, S_3\, \dot{x_2}) + C_2\, \dot{x_2}\,(l_3 - l_5) \end{pmatrix} \text{(II.29)}$$

II.3.3.3 Rotational velocity matrices

When the complex movement of a joint is decomposed, its rotation velocity is primarily expressed in its own frame. As if the joint is rotating at itself as figure illustrates an

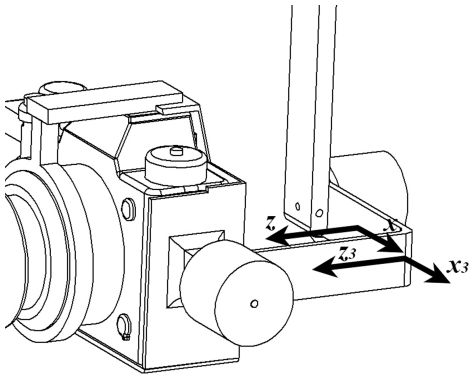


Figure II.11: Third joint attached frame.

example of the third joint. This angular velocity is represented in 3×3 matrix form to help express all combinations of the joint dynamics related to its inertia matrix and angular velocity.

Let us take the third joint example. In its attached frame, the joint rotational kinetic energy is expressed as the result of its rotation along the y axis and z axis. Respectively by $\dot{\theta}_1$ and $\dot{\theta}_2$. It is then expressed as follows

$$K = \frac{1}{2} I_{yy} {\dot{\theta}_1}^2 + \frac{1}{2} I_{zz} {\dot{\theta}_2}^2 \text{(II.30)}$$

This energy general expression is as follows

$$K = \frac{1}{2} \dot{X}^T [J_{\omega_i}^T(X) I J_{\omega_i}(X)] \dot{X} \text{(II.31)}$$

Where the vector X contains the generalized coordinates.

$$X = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \text{(II.32)}$$

The only possible value for the matrix $J_{\omega_3}(X)$ that would set II.30 equal to II.31 is

$$J_{\omega_3}(X) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \text{(II.33)}$$

Using the same technique, we have extracted the rest angular velocities. They are as follows

$$J_{\omega_1}(X) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{(II.34)}$$

$$J_{\omega_2}(X) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{(II.35)}$$

$$J_{\omega_4}(X) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \text{(II.36)}$$

$$J_{\omega_5}(X) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{(II.37)}$$

II.3.3.4 Kinetic energy

The kinetic energy of the system is the sum of each joint, and the camera kinetics. It is established in II.38

$$K = \frac{1}{2}\dot{X}^T \sum_{i=1}^{5} [m_i J_{v_i}^T(X) J_{v_i}(X) + J_{\omega_i}^T(X) R_0^i(X) I_i {R_0^i}^T(X) J_{\omega_i}(X)]\dot{X} \text{(II.38)}$$

The first term II.39 represents the sum of the joints translational kinetics. the term $J_{v_i}(X)$ represents the translational velocity of the a joint i. Which we have established before as $\dot{r_{c_i}}$. The terms II.39 becomes then as in II.40

$$K_{t_r} = \frac{1}{2}\dot{X}^T \sum_{i=1}^{5} [m_i J_{v_i}^T(X) J_{v_i}(X)]\dot{X} \text{(II.39)}$$

$$K_{t_r} = \frac{1}{2} \sum_{i=1}^{5} [m_i \dot{r_{c_i}}^T \dot{r_{c_i}}] \text{(II.40)}$$

The second energy term II.42 represents the rotational kinetics. A joint inertia in its attached frame as we established before, differs from its inertia in the other frames as it rotates through time. The term II.42 helps transform the inertia of a joint i to the inertial frame. The product $J_{\omega_i}(X)\dot{X}$ represents the angular velocities we have established before.

$$K_r = \frac{1}{2}\dot{X}^T \sum_{i=1}^{5} [J_{\omega_i}^T(X) R_0^i(X) I_i {R_0^i}^T(X) J_{\omega_i}(X)]\dot{X} \text{(II.41)}$$

$$R_0^i(X) I_i {R_0^i}^T(X) \text{(II.42)}$$

The overall kinetic energy represents heavy calculations. Therefore we have held them on MATLAB. The resulted equation is extremely large and have no use-case to be put in the down here, therefore, we have appended it in appendix **??**.

II.3.3.5 Potential energy

Just as the kinetics, the potential energy of the whole system is the sum of each joint, and the camera, potential energies. It is equal to II.43

$$P_i = m_i \vec{g} \vec{r_{ci}} \text{(II.43)}$$

The term $\vec{g}$ represents the gravity vector which is equal to II.44 in the inertial frame.

$$\vec{g} = \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \text{(II.44)}$$

$\vec{r_{ci}}$ is the center of gravity position vector for each joint i. We have already established

those. And m_i is a joint mass.

Unlike the kinetic energy of the system, the overall potential energy of is relatively simple. So, we have held the calculation by hand and the resulted energy is

$$\begin{aligned} P = \quad & -m_1 g l_0 - m_2 g \left(l_0 + \frac{l_2}{2}\right) \\ & - m_3 g \left(l_0 + l_2 - \frac{l_3}{2} S_2\right) - m_4 g \left(l_0 + l_2 - l_3 S_2\right) \\ & - m_5 g \left(l_6 C_2 C_3 - l_7 C_2 S_3 - \left(l_3 - l_5\right) S_2 + l_2 + l_0\right) \end{aligned} \tag{II.45}$$

II.3.3.6 Equations of motion

We have established the system motion equation using Lagrange equation system II.55 Where L is defined in II.46. τ_c is the control command in Nm and τ_p is the external disturbances in the same Newton-meter unit. In sake of simplification of the equations manipulation we have referred from part on to θ_i by x_i

$$L = K - P \tag{II.46}$$

$$\begin{cases} \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x_1}}\right] - \frac{\partial L}{\partial x_1} = \tau_{c_1} + \tau_{p_1} \\ \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x_2}}\right] - \frac{\partial L}{\partial x_2} = \tau_{c_2} + \tau_{p_2} \\ \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x_3}}\right] - \frac{\partial L}{\partial x_3} = \tau_{c_3} + \tau_{p_3} \end{cases} \tag{II.47}$$

The resulted system is of the form II.48.

$$M\left(X\right)\ddot{X} + C\left(X, \dot{X}\right) + G\left(X\right) = \tau_c + \tau_p \tag{II.48}$$

We have divide the dynamics equation system we have established in II.55 in a form of II.49 to help write the extremely long system. These calculation were just as the kinetics held in MATLAB on several steps. The resulted equations system we have appended in appendix **??** for the its length.

$$\begin{cases} subsys_1 : M_1\left(X\right)\ddot{X} + C_1\left(X, \dot{X}\right) + G_1\left(X\right) = \tau_{c_1} + \tau_{p_1} \\ subsys_2 : M_2\left(X\right)\ddot{X} + C_2\left(X, \dot{X}\right) + G_2\left(X\right) = \tau_{c_2} + \tau_{p_2} \\ subsys_3 : M_3\left(X\right)\ddot{X} + C_3\left(X, \dot{X}\right) + G_3\left(X\right) = \tau_{c_3} + \tau_{p_3} \end{cases} \tag{II.49}$$

Where

$$M(X) = \begin{pmatrix} M1(X) \\ M2(X) \\ M3(X) \end{pmatrix} \tag{II.50}$$

$$C(X) = \begin{pmatrix} C1(X, \dot{X}) \\ C2(X, \dot{X}) \\ C3(X, \dot{X}) \end{pmatrix} \text{(II.51)}$$

$$G(X) = \begin{pmatrix} G1(X) \\ G2(X) \\ G3(X) \end{pmatrix} \text{(II.52)}$$

II.3.3.7 Friction effect

Simscape helps simulates real-world systems. In real world, robotic joints have damping coefficients due to actuators internal damping and friction. This damping resists the articulations motion and stops the system from oscillating over time. It is a common behavior of gimbals under no power. Therefore, we have attached a damping coefficient C_{d_i} to each rotoid articulation. The effect of these damping coefficients on the system dynamics is represented through a dissipation function. We establish by here the dissipation function $D(X)$.

$$D = \frac{1}{2}\left(C_{d_1}\dot{x_1}^2 + C_{d_2}\dot{x_2}^2 + C_{d_3}\dot{x_3}^2\right) \text{(II.53)}$$

where a single articulation dissipation function is

$$D_i = \frac{1}{2}C_{d_i}\dot{x_i}^2 \text{(II.54)}$$

The Lagrange equation system becomes as in II.55 and of the form II.56

$$\begin{cases} \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x_1}}\right] - \frac{\partial L}{\partial x_1} + \frac{\partial D}{\partial \dot{x_1}} = \tau_{c_1} + \tau_{p_1} \\ \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x_2}}\right] - \frac{\partial L}{\partial x_2} + \frac{\partial D}{\partial \dot{x_2}} = \tau_{c_2} + \tau_{p_2} \\ \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x_3}}\right] - \frac{\partial L}{\partial x_3} + \frac{\partial D}{\partial \dot{x_3}} = \tau_{c_3} + \tau_{p_3} \end{cases} \text{(II.55)}$$

$$M(x)\ddot{x} + C(x, \dot{x}) + D\dot{x} + G(x) = \tau_c + \tau_p \text{(II.56)}$$

Where the derivatives $\frac{\partial D}{\partial \dot{x_i}}$ are respectively

$$\frac{\partial D}{\partial \dot{x_1}} = C_{d_1}\dot{x_1} \text{(II.57)}$$

$$\frac{\partial D}{\partial \dot{x_2}} = C_{d_2}\dot{x_2} \text{(II.58)}$$

$$\frac{\partial D}{\partial \dot{x_3}} = C_{d_3}\dot{x_3} \text{(II.59)}$$

II.3.3.8 Model validation

This section establish modeling accuracy with regards to the assumptions we have used. In figure II.12 below we illustrate the three angular positions of the gimbal concept, with mounted camera, opposed to the reached dynamics model. This last is obviously with regards to assumptions 1 to 3. The system has no disturbances nor control applied to it. This would dictates if the modeling phase has ended with success.

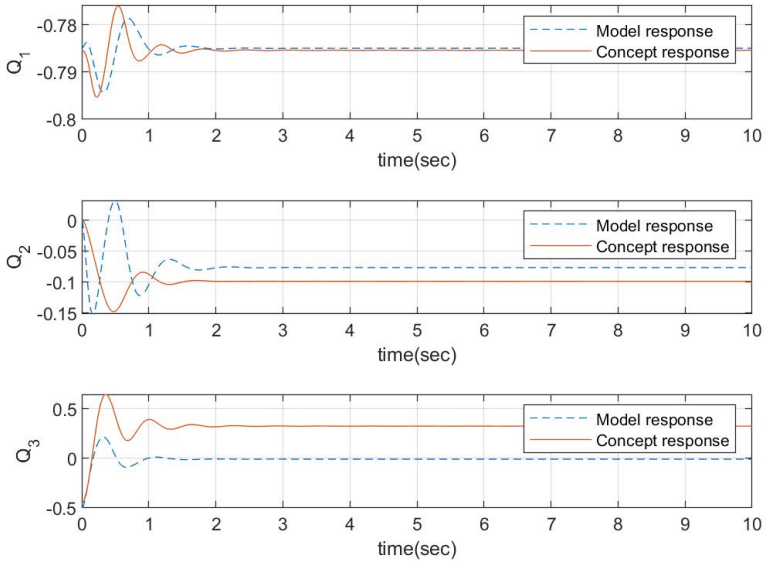


Figure II.12: Dynamics model and concept angular positions (No disturbances applied).

Similar behaviors are observed, with three key differences. First, the concept final angular positions go further than the model positions(figure II.13). This is due the camera position offset, which differs from the camera concept to the cuboid we assumed for camera in the modeling process. As figure II.14 illustrate, the offset became shorter. Therefore, the gravity torque on the camera becomes weaker and the damping on the gimbal joints affect the movement larger. A simple solution to the observed characteristic is to lengthen the offset in the dynamics model. However, this would affect the dynamics of the model in the transitional slot to alter from the concept. Reason why is for the system is highly coupled which leads us to the second key difference. A slower response is observed on the first angular position(figure II.15).

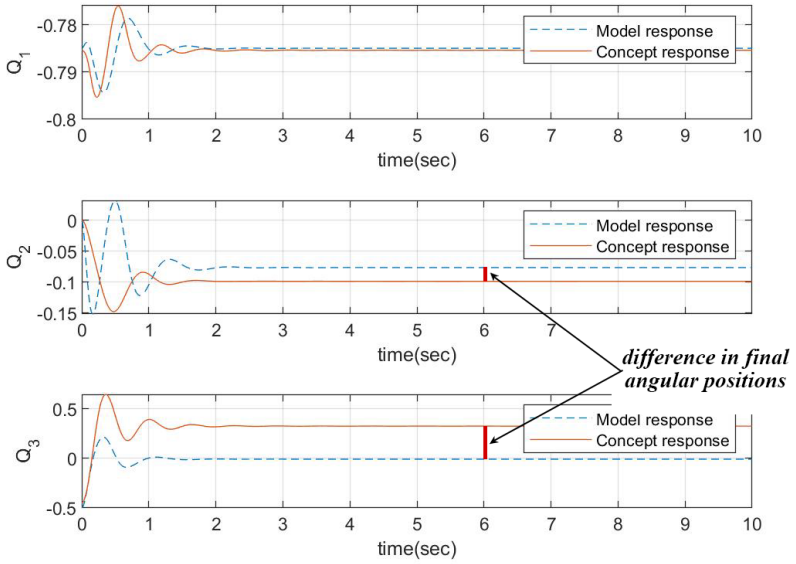


Figure II.13: Difference in final angular positions illustration.

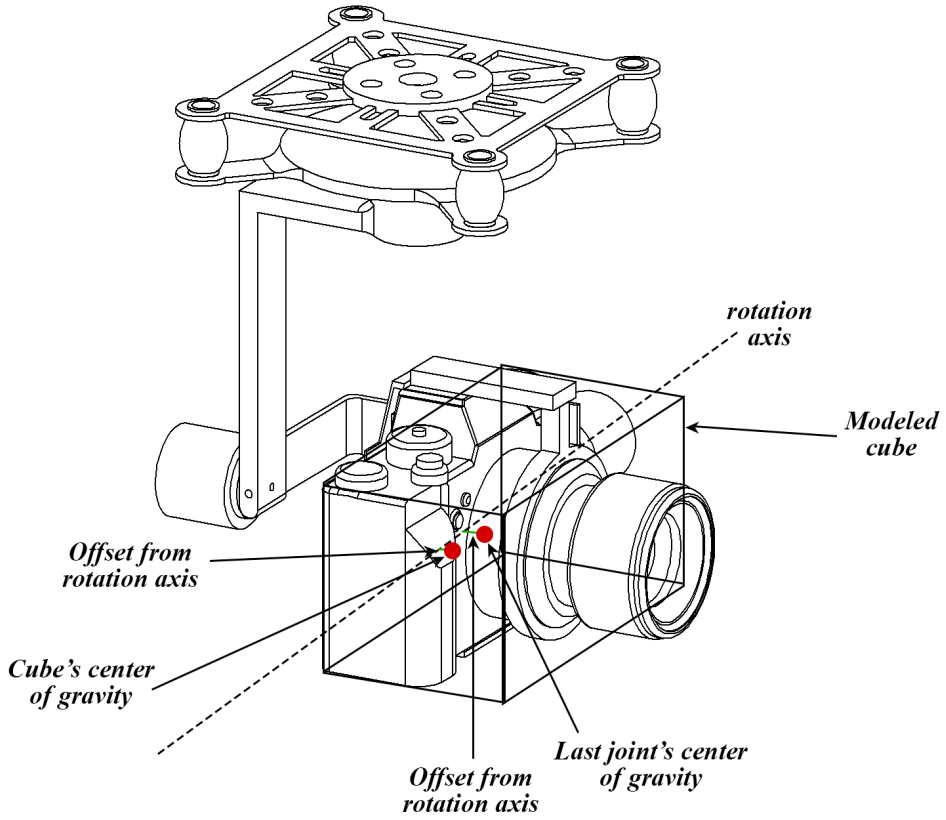


Figure II.14: Difference in center of gravity offset illustration.

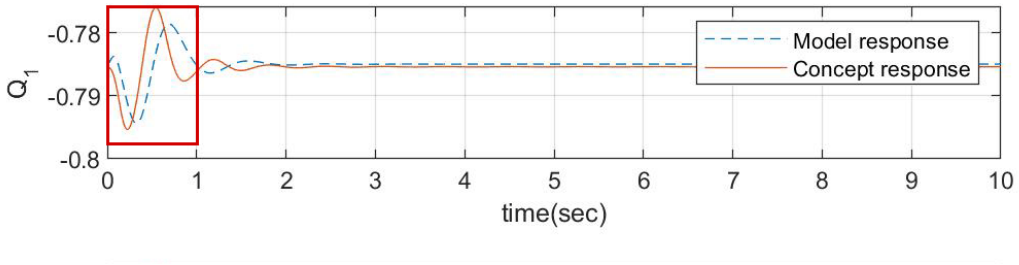


Figure II.15: Transitional response difference illustration in the first angular position

Which by contrast is caused by the faster response of the system in its second position(figure II.16).

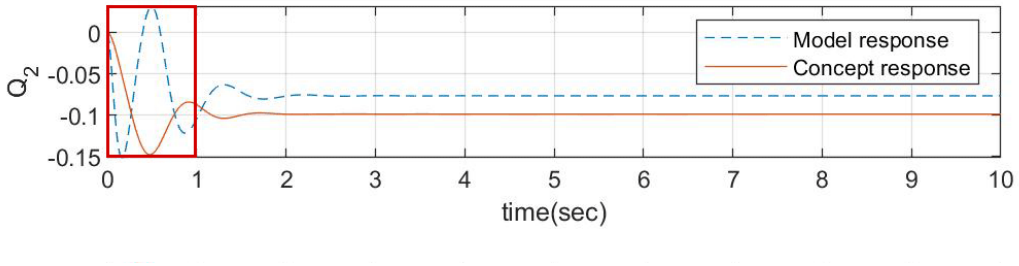


Figure II.16: Transitional response difference illustration in the second angular position

The second affect on the first is caused by the coupled dynamics of the system that are modeled through transferring the inertia matrices to the inertial frame as we we have done in equation II.42. However, the second response is faster due to effect of assumptions 1 and 3. As we have supposed the camera to be cubic we have enlarged the camera mass especially with regards to its width. This affect most the second angle as the gravity force becomes larger and its resulting torque on the camera too.

The last difference that sets the two behaviors apart is occasional moments of partial resonance. These would appear as augmented amplitudes in one of the angles. They are caused by either slack or faster response of the dynamics model compared to the concept as illustrated in figure II.17 below.

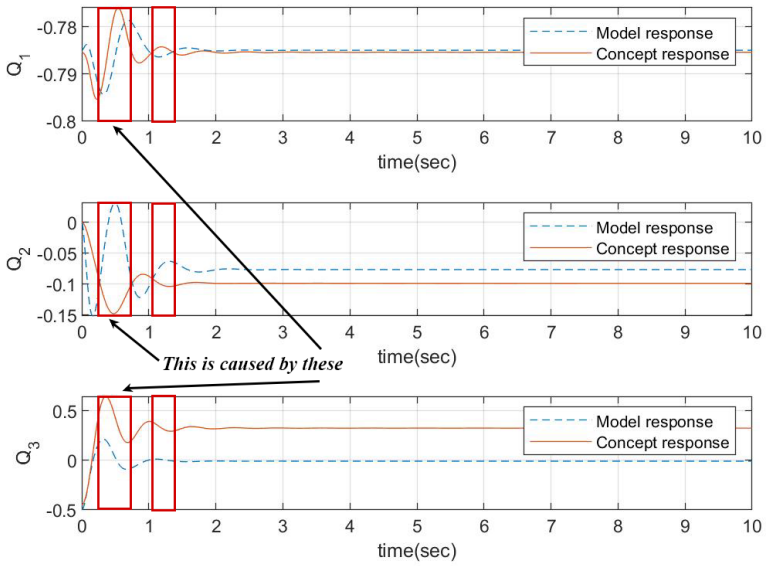


Figure II.17: Resonance observation on the models behaviors.

II.4 Conclusion

We have concluded this section with a CAD conception of the gimbal. This conception is now ready for XML generation through Simscape SolidWorks plugin. This means we now have a gimbal plant ready for control implementation. We have established a dynamics model for this digital twin. We have confirmed the model through control-free simulations. The model have behaved in a same manner as the conception, however, it have shown some differences resulted from the assumptions we have used. An extremely precise dynamics model of the system would need eliminating these assumptions. To exclude the assumptions 1 and 2 we would need only to introduce precise inertia matrices for the joints and the camera. To exclude assumption 3 we would need to add 3 more sets of energies to the total energy of the system before using the Lagrange motion equations system. These 3 sets represent each motor kinetic and potential energies.

III. Meta-heuristic optimization methods

III.1 Introduction

In this section we present the meta-heuristics GNDO [13]. This algorithm we have used to optimize our proposed control. We describe here the mechanisms of the algorithm, how they collaborate and the main inspiration of the optimizer. We also take a dive of other algorithms such as ACO [14] and GWO [15]. We have used these algorithms in order to have a reference in optimizers performances. This comparison tels to have large perspective on the GNDO performances.

III.2 Generalized Normal Distribution Optimizer

In theory, natural random phenomena tend to have normal probability distributions. Couple of parameters set these distributions apart from each other, the first parameter is the location parameter, it defines the main value, that phenomenon other values tend to be around. In the probability theory, we refer to this parameter as the mean μ. The second parameter defines how much these values can vary from each other, it is called the scale parameter and in theory it represents as the standard variance δ. The combination of these parameters using the Gaussian law is given in equation III.1, which defines a probability distribution for natural phenomena. Figure III.1 represents a normally distributed phenomenon for different parameters.

$$N(x) = \frac{1}{\sqrt{2\pi\delta}} \exp\left(-\frac{(x-\mu)^2}{2\delta^2}\right) \quad \text{(III.1)}$$

As every meta-heuristic optimization strategy, the GNDO starts with random values. The key of this algorithm is that the position of these random values with respect to their optimum values is represented by a normal distribution of two parameters. Location is the optimum value and scale is how these values vary from their optimum values. The start distribution of a variable is naturaly different from its optimum one, by the end of the optimization both distributions should be nearly identical as illustrated in figure III.2.

The GNDO optimization strategy is described by the equations III.2 to III.11 and their

significance are in the table III.1. Table III.2 illustrates the GNDO algorithm process. Both tables are a resume of the work done in [13].

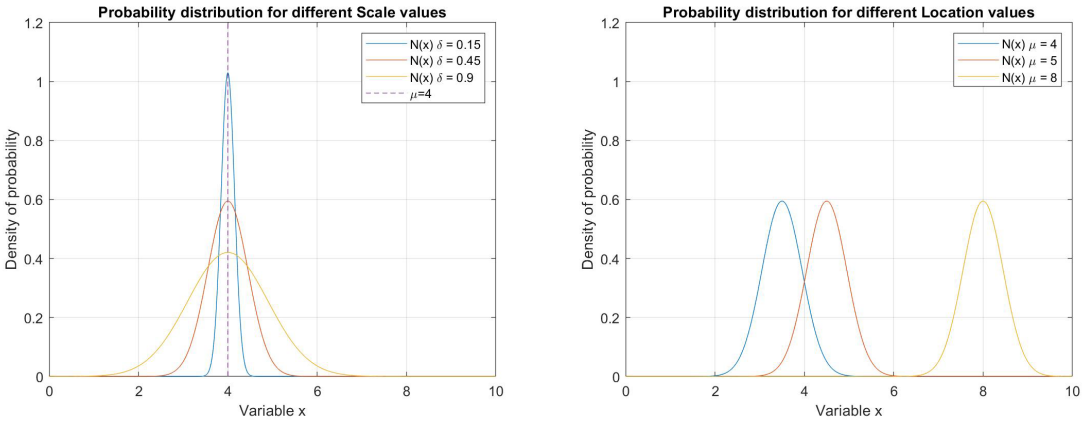


Figure III.1: Normally distributed phenomenon for different parameters

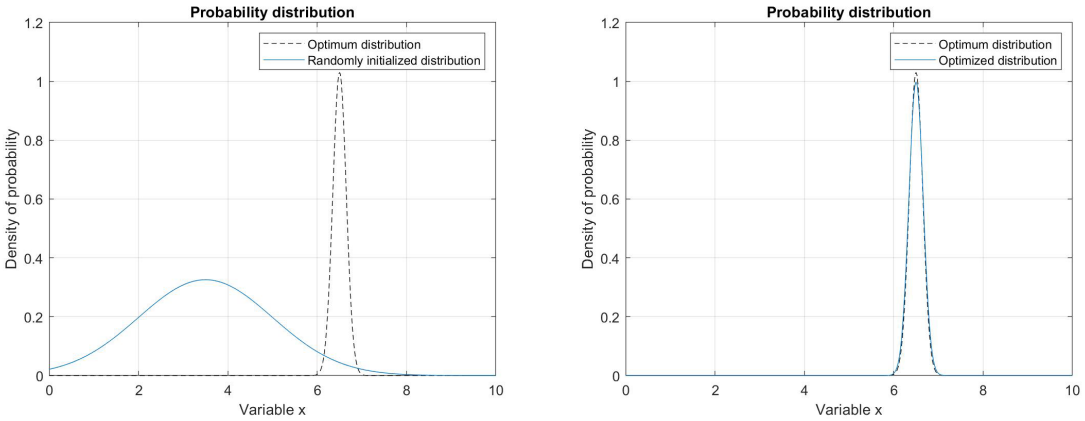


Figure III.2: Randomly initialized to optimized distribution

$$x_i = l_i - \lambda_5 \left(l_i - u_i\right) \text{(III.2)}$$

$$M = \frac{1}{N} \sum_{i=1}^{N} x_i \text{(III.3)}$$

$$\mu_i = \frac{1}{3} \left(x_i + x_{best} + M\right) \text{(III.4)}$$

$$\delta_i = \sqrt{\frac{1}{3} \left[\left(x_i - \mu_i\right)^2 + \left(x_{best} - \mu_i\right)^2 + \left(M - \mu_i\right)^2\right]} \text{(III.5)}$$

$$\eta = \begin{cases} \sqrt{-\log \lambda_1} \cos\left(2\pi\lambda_2\right) \\ \sqrt{-\log \lambda_1} \cos\left(2\pi\lambda_2 + \pi\right) \end{cases} \text{(III.6)}$$

$$\nu_i = \mu_i + \delta_i \times \eta \text{(III.7)}$$

$$x_i = \begin{cases} \nu_i & if \quad f(\nu_i) < f(x_i) \\ x_i & otherwise \end{cases} \tag{III.8}$$

$$\nu_i = x_i + \beta \times (|\lambda_3| \times \nu_1) + (1 - \beta) \times (|\lambda_4| \times \nu_2) \tag{III.9}$$

$$\nu_1 = \begin{cases} x_i - x_{p_1} & if \quad f(x_i) < f(x_{p_1}) \\ x_{p_1} - x_i & otherwise \end{cases} \tag{III.10}$$

$$\nu_2 = \begin{cases} x_{p_2} - x_{p_3} & if \quad f(x_{p_2}) < f(x_{p_3}) \\ x_{p_3} - x_{p_2} & otherwise \end{cases} \tag{III.11}$$

We have set here in table III.1 the significance of each parameter and variable of the GNDO algorithm.

Table III.1: GNDO parameters and terms significance

Parameter	Significance
α	Random variable in $[0; 1]$, to alternate between exploration and exploitation.
β	Random variable in $[0; 1]$, to balance between local and global information.
δ_i	The variance for a distribution x_i.
η	Penalty factor.
$f(x_i)$	Cost function of a distribution x_i.
l	Decision variables lower bounds vector.
λ_1, λ_2, λ_3	Random values in $[0; 1]$.
λ_3, λ_4	Normally distributed random values in $[-1, 1]$.
M	The mean of a population X.
x_{best}	The best distribution along the population x.
μ_i	The mean of: a single distribution x_i, the best distribution x_{best} and the mean of the distributions population M.
N	The size of the population.
ν_i	New distributions population.
ν_1	New distributions of index p_1.
ν_2	New distributions of index p_2.
p_1, p_2, p_3	Random integer values within the population size.
x	The population vector.

Table III.2: GNDO algorithm

Input: population size N, the upper limits of variables u, the lower limits of variables l, the current number of iterations $t = 0$ and the maximum number of iterations T_{max}.

/* Initialization */

1. Initialize population by equation III.2.
2. Calculate the fitness value of every individual and achieve the optimal solution x_{best} of the initialized population.
3. Update the current number of iterations $t = t + 1$.

/* Main loop */

while $t \leq T_{max}$ **do**
  for $i = 1 : N$ **do**
    Generate a random number α between 0 and 1.
    if $\alpha > 0.5$ **then**
      /* Local exploitation strategy */
    1. Select the current optimal solution x_{best} and calculate the mean position M by equation III.3.
    2. Compute generalized mean position μ_i, generalized standard variance δ_i and penalty factor η by equation III.4, equation III.5 and equation III.6, respectively.
    3. Perform the local exploitation strategy by equation III.7 and equation III.8.
    else
      /* Global exploration strategy */
    1. Perform the global exploration strategy by equation III.9, equation III.10, equation III.11 and equation III.8.
    end if
  end for
  Update the current number of iterations by $t = t + 1$.
end while

Output: the optimal solution x_{best}.

III.3 Gray Wolf Optimizer

As the name states, this technique simulates the grey wolves pack behavior. Greys are very hierarchical species. We find at the top the Alpha wolf which is the leader of all the pack. Alpha has the best characteristics combination among the pack such as strong, smart,etc. Right downwards lays the Beta. We find in a wolves pack several Beta wolves. One of them is meant to take the leadership and replaces the alpha when it dies. The other existed wolves level is the Delta group. These are not meant to lead but they are very active and useful to the pack. Other types of wolves exist, but they all rests around the bottom of the hierarchy. These are considered all as Omegas.

Grey wolves hunt in packs. The hunting has three main phases, tracking the hunt, encircling and attacking. Alpha,Beta and delta lead the pack through the hunts. GWO algorithm uses mathematical equations to simulate the hunting phases through every iteration. By each iteration the algorithm extracts three consecutive best solutions from the wolf's population, which are alpha, beta and delta. Then, it uses the three values for the exploitation phase. Otherwise, the algorithm explores the current population.

We based our work on [15]. The mathematical equations of the algorithm are given from III.12 to III.21 and the pseudo-code based on these equations is in table III.4.

$$a = \begin{cases} 2 & initialization \\ 2 - \left(\frac{2k}{T_{max}}\right) & otherwise \end{cases} \quad \text{(III.12)}$$

$$A = 2 \times a \times r_1 - a \text{(III.13)}$$

$$C = 2 \times r_2 \text{(III.14)}$$

$$D_\alpha = |C_1 \times X_\alpha - X(k)| \text{(III.15)}$$

$$D_\beta = |C_2 \times X_\beta - X(k)| \text{(III.16)}$$

$$D_\delta = |C_3 \times X_\delta - X(k)| \text{(III.17)}$$

$$X_1(k+1) = X_\alpha(k) - A_1 \times D_\alpha \text{(III.18)}$$

$$X_2(k+1) = X_\beta(k) - A_2 \times D_\beta \text{(III.19)}$$

$$X_3(k+1) = X_\beta(k) - A_3 \times D_\beta \text{(III.20)}$$

$$X(k+1) = \frac{1}{3}\sum_{j=1}^{3} X_j(k+1) \text{(III.21)}$$

Table III.3 below illustrates the significance of the GWO parameters and variables mentioned in the equations above.

Table III.3: GWO parameters and variable significance

Parameter	Significance
a	A linearly decreased function. It represents the effect of the α on the other wolves decisions III.3.
A_i	Random coefficient.
α	First best solution index.
β	Second best solution index.
δ	Third best solution index.
C	Random function. It balances between exploration and exploitation.
D	Simulation model of the encircling behavior of the wolves. It represents the distances of each wolf from the prey(figure III.3).
$f(x_i)$	Cost function of the individual x_i.
k	Iteration index.
r_1, r_2	Random numbers between 0 and 1.
T_{max}	Number of iterations.
X	Wolves population.

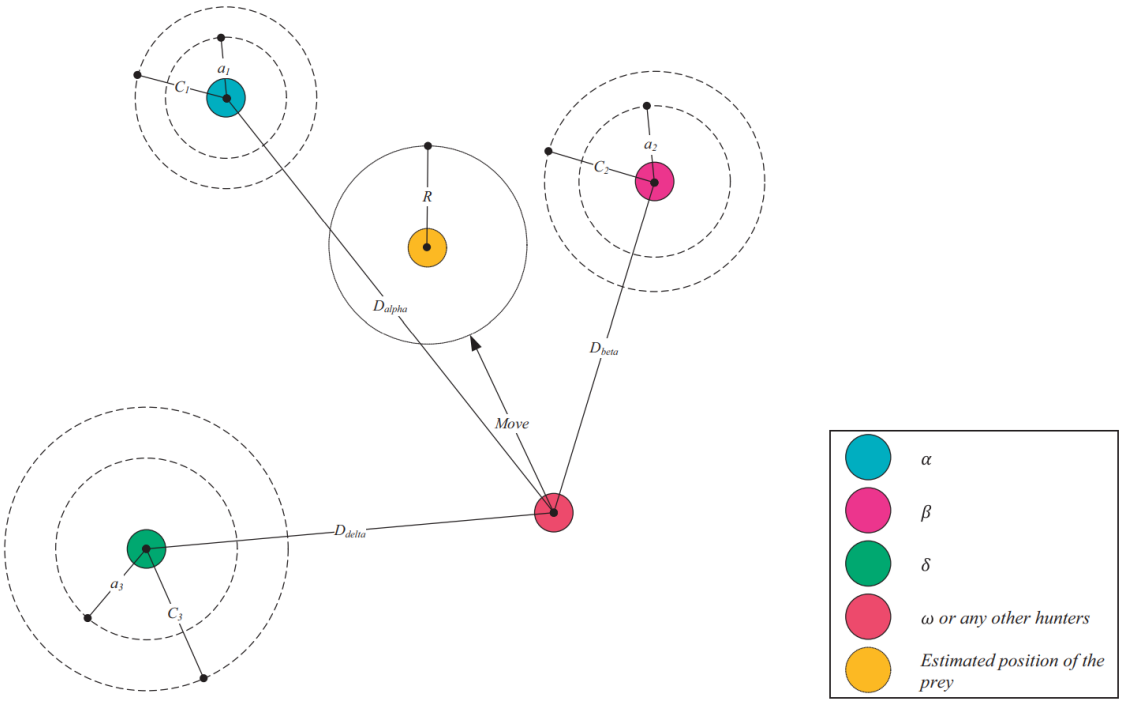


Figure III.3: Grey wolves hunting illustration [15]

Table III.4: Grey wolf optimizer algorithm.

Input: population size N, the upper limits of variables u, the lower limits of variables l, the current number of iterations $t = 0$ and the maximum number of iterations T_{max}

/* Initialization */

1. Randomly initialize the wolves population.
2. Initialize a, A and C by equations III.12, III.13 and III.14 respectively.
3. Calculate the fitness value of every individual and achieve the three optimal solutions x_α, x_β and x_δ of the initialized population.
4. Update the current number of iterations $t = t + 1$.

/* Main loop */

while $t \leq T_{max}$ **do**
  for $i = 1 : N$ **do**
    1. update the position of the current wolf by equations III.15 to III.21
  end for
  Re-evaluate the fitness of all individuals.
  Update x_α, x_β and x_δ.
  Update a, A and C by III.12, III.13 and III.14
  Update the current number of iterations by $t = t + 1$
end while

Output: the optimal solution x_α

III.4 Ant Colony Optimizer

ACO is an optimization algorithm that is based on the concept of Stigmergy which is the stimulation of the workers by the performance they have achieved. It is a common form of communication among ants. Initially, ants march towards a food source individually in random paths. In their way, each ant deposits a substance called Pheromone. The other ants are stimulated by this substance and tend to follow the path with the highest concentration of this pheromone. Over time, and going forward and backward to the food source, the shortest path towards the food becomes most concentrated with pheromone as more and more ants are attracted to it and use it. By several loads, the shortest path becomes the only used path by the ants.

The first step in this species behavior simulation is to divide the search space with a step size h. Each step characterized with a pheromone amount which initialized equally

at 0 along the search space. Each ant is given a probabilistic overview on the paths pheromone amounts using the roulette wheel concept. An ant then takes a path according to this probability. Each step the ant takes, the algorithm calculates the amount of pheromone at that step and compare it to the other steps amounts. This helps extract two pheromone values, best and worst using equations III.24 and III.25. According to these values the ant figures a new probability using equation and chooses the new path. Equation III.25 simulates the reduction of the worst path pheromone amount and III.24 simulates the increase of it.

The ACO algorithm is mimicked using simulating equations from III.22 to III.25 [14], in addition, pseudo-code established in table III.6.

$$p_j^k = \frac{\tau_j}{\sum_{j=1}^{m} \tau_j} \text{(III.22)}$$

$$\Delta\tau_j^{(k)} = \frac{\gamma f_{best}}{f_{worst}} \text{(III.23)}$$

$$\tau_j^{new} = \tau_j^{old} + \sum_k \Delta\tau_j^k \text{(III.24)}$$

$$\tau_j^{new} = (1-\rho)\,\tau_j^{old} \text{(III.25)}$$

Table III.5 illustrates each ACO parameter and variable significance

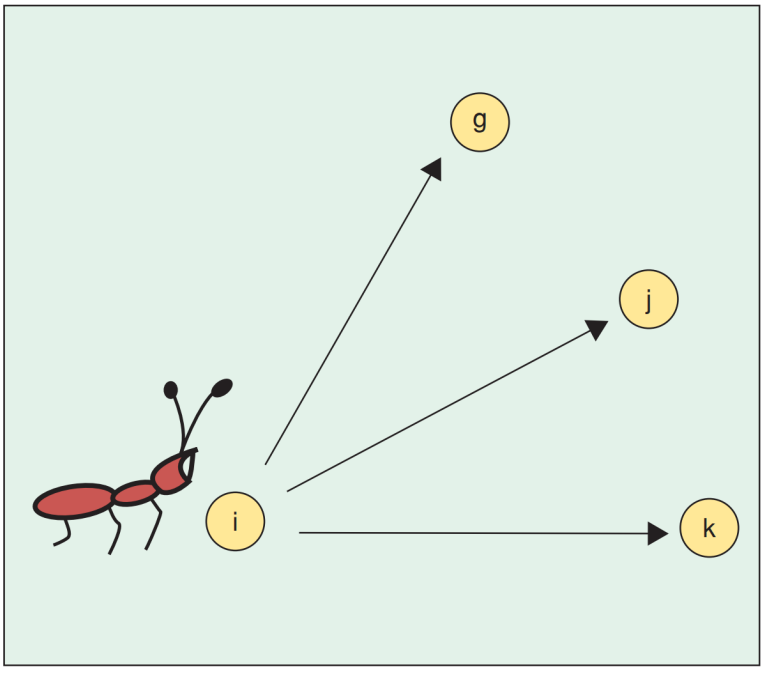


Figure III.4: ACO: ant choosing the next step [14].

Table III.5: ACO parameters and variables significance

Parameter	Significance
u_b	Upper bound of search space
l_b	Lower bound of search space
N	Group of all ants (Population size)
T	Number of iterations
γ	Scaling parameter (positive parameter)
ρ	Evaporate rate (positive parameter)
h	Step size for divided search space
m	Number of discrete values for each of the design variable
k	Ants: Any possible solution
j	Path possible
p_j^k	Propability to select discrete values of design variables
$\Delta\tau_j$	Factor of best solution of ants
r	Random number in range (0,1) for each ant
τ	Pheromone trail
$f(x_i)$	Objective function
f_{best}	Best Cost function result
f_{worst}	Worst Cost function result
x_{best}	Optimal solution

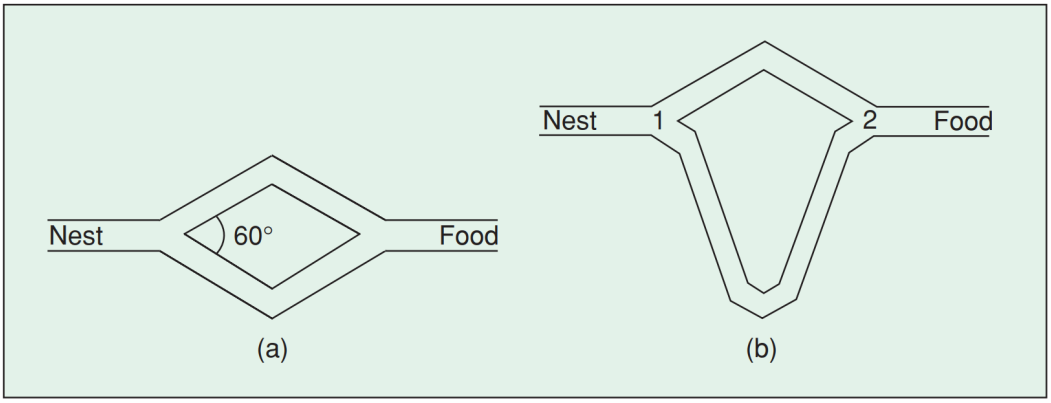


Figure III.5: ACO: ant paths [14].

Table III.6: ACO algorithm

Input: upper bound u_b, lower bound l_b, population size N, number of iterations T, scaling parameter γ, evaporate rate ρ,step size h.

/* Initialization */

1. Assume a set of permissible discrete values m for each of design variable.
2. Initialized all of design variables m with equal amount of phermone τ

/* Main loop */
while $t \leq T$ **do**

1. Find probability using III.22.
2. Find the cumulative probability ranges.
3. **for** i=1:N **do**
   - Generate a random number r.
   - Find corresponding discrete value.
   - Evaluate the objective function $f(x_i)$.
4. **end for**
5. Determine both best and worst objective function values f_{best} and f_{worst}
6. Update best path using equations III.23 and III.24.
7. Update the other paths using equation III.25

end while

Output: the optimal solution x_{best}

III.5 Conclusion

Overall, GNDO, GWO, and ACO offer different approaches to optimization and have their own strengths and weaknesses. Besides the simulation equations, each one performance depends also on the optimized plant.

IV. Model-free control of three-axis camera gimbal

IV.1 Introduction

The gimbal control is a challenging problem since the system is highly coupled and the dynamics are very complex. This makes model-based control techniques exceptionally complex to synthesis. This sheds the light on model-free techniques, however, these are ignorant of the system dynamics. Nonetheless, latest researches are introducing adaptiveness terms to model-free control that may compensate the model errors. A major reference on model-free control is [16] which has been our main source on the subject. More approaches are found in [17] and [18].

IV.2 Proposed control

IV.2.1 Model-free control

The main reason for control is to impose the system to behave in certain manner which means for it to follow a predefined trajectory. The obvious way to evaluate the behavior of the system is the deviation from the imposed behavior which is represented by the tracking error $e = y - y_{ref}$.

In order to design a model-free controller, we admit that we don't have any idea about the system dynamics,we are dealing with a system that consists of one control variable, denoted as u, and one output variable, denoted as y. Instead of using the unknown and intricate "complex" mathematical model. In order to use the model-free control method the system can be presented by an Ultra-Local Model [17] and [18] which is defined in equation IV.1.

$$y^{(n)}(t) = F(t) + \alpha U(t) \text{(IV.1)}$$

The term $U(t)$ represents the control term, the term $F(t)$ represents both unmodeled dynamics and disturbances and these need to be estimated in real-time, the term α which is a non-physical constant, is chosen by the practitioner in such a way that the magnitudes of αu and $y^{(n)}$ are comparable. . We denote the dynamics estimations $\hat{F}(t)$ and we then extract the control formula from equation IV.1, and we state it in

equation IV.2.

$$U(t) = \frac{1}{\alpha}\left(y^{(n)}(t) - \hat{F}(t)\right) \text{(IV.2)}$$

Back to the error, this has a separate dynamics which we model it through a linear combination of its derivatives as stated in IV.3.

$$e^{(n)} + k_{n-1}e^{(n-1)} + \cdots + k_0 e = 0 \text{(IV.3)}$$

We then replace the system response in the control formula IV.2 by a function of the error dynamics IV.3 and the reference dynamics as we have $y_{ref}^{(n)} = y^{(n)}(t) + e$.

$$U(t) = \frac{1}{\alpha}\left(-\hat{F}(t) + y_{ref}^{(n)}(t) - k_{n-1}e^{(n-1)} - \cdots - k_0 e\right) \text{(IV.4)}$$

The error dynamics can not be of a higher order than the system dynamics. The studied system has a second order degrees therefore only a PD controller can be used within the main controller. However, we can integrate the error several times independently of the system order, depending on the response we seek. In this fact the general formula of the proposed controller can be written as in IV.5

$$U(t) = \frac{1}{\alpha}\left(-\hat{F}(t) + y_{ref}^{(n)}(t) - k_{n-1}e^{(n-1)} - \cdots - k_0 e - k_{i_1}\int e - k_{i_2}\iint e - \ldots\right) \text{(IV.5)}$$

For the studied system, we propose a single integral tracker, and as the system dynamics is of a second order, we end-up with an intelligent PI as called in [16]. We state the controller formula in equation IV.6.

$$U(t) = \frac{1}{\alpha}\left(-\hat{F}(t) + \ddot{y}_{ref}(t) - k_p e - k_i\int e\right) \text{(IV.6)}$$

We show in figure IV.2 the control schema.

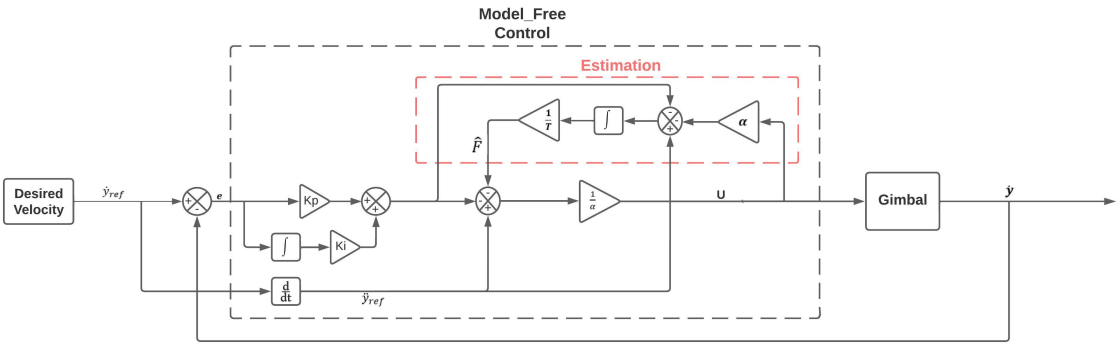


Figure IV.1: Control schema

Another technique that lays under this model-free control is Limited Model Control [11]. If we supposedly have a restricted knowledge of the system dynamics, we than can enhance the model free controller for better estimation and better performance. For this, the dynamics estimate $\hat{F}(t)$ would contain both the known and the unknown dynamics. Nonetheless, more information on this controller in [16].

IV.2.2 Dynamics estimation

Real time estimation is a vast theory and the methods are several,The classic approach in mathematical analysis involves approximating an integrable function, which is a general function defined on the interval [a, b] where a and b are real numbers with $a < b$, with a step function F_{approx}. A step function is a piecewise constant function. In order to obtain a suitable approximation of F, the key task is to estimate the constant parameter $\hat{F}$ within a sufficiently "small" time interval. Intelligent controllers can also employ similar estimations of F to achieve their objectives. However, we are interested in Closed the loop with iPID techniques for which we propose IV.7.

$$\hat{F}(t) = \frac{1}{T}\int_{T} y^{(2)}(\tau)\, d\tau \text{(IV.7)}$$

Equation IV.7 is a simple form of parametric estimation, we have inspired from [18]. This will make calculations to fast for real time applications. Moreover, integrals have filtering properties which helps make the estimation accurate with regards to noises. When the estimation window T is relatively large the estimation becomes robust against noises, however, it is less accurate. When the estimation window is small the estimation is more accurate,however, it becomes vulnerable towards noises. Which sets the estimation window to have very crucial parameter and affect the controller performance.

We now the estimation is proportional to the error between the desired dynamics and the system actual dynamics [18], [17], also to the error between the desired dynamics and the control [11]. Based on that, the estimation function above have been tailored as in IV.8 .

$$\hat{F}(t) = \frac{1}{T}\int_{T} \left(\ddot{y}_{ref}(\tau) - \alpha U(\tau) - k_p e(\tau) - k_i \int e(\tau) \right) d\tau \text{(IV.8)}$$

IV.2.3 Meta-heuristic optimization of the model-free control

From IV.6 and IV.8 the proposed control approach is concluded which is a function of parameters: α, T, K_p and K_i. Tuning these parameters is a tedious job which set the purpose for their optimization. For that, we propose GNDO algorithm as a main optimizer.

Even if the manual tuning results good parameters, we can still optimize these parameters. Nonetheless, for meta-heuristic algorithms starting with random values, many of the starting combinations can lead to singularities, heavy time consuming or misbehavior of the controller. Ever so, a manual tuning at the start is needed to establish an interval for each parameter.

A major key to an optimization process performance is the selection of an efficient cost function. We discuss here the constraints that characterised the cost function to be used during the optimization process of the model-free control.

The obvious specification is that the cost needs to be proportional to the difference between the system dynamics and the desired dynamics. We know the camera angular velocities have an oscillating characteristic, this means the error can vary from positive

to negative values. Under summation, these values can under- affect each other. To avoid this issue a quadratic form of the error is needed.

The considered error should be along the whole running course of the system. Mathematically, it is translated to a sum of every abscissa error along the course of simulation. More precisely, an integration over the course.

In theory, the error between desired and actual dynamics needs to converge to zero over time. The amount of time it takes to converge affect the system response time. A common implementation for this through optimization is a proportional term to the time IV.9.

All three specifications have lead us to the cost function IV.9. It is a commonly used function in control optimization due to the specification that we have explained before.

$$f = \gamma \int_t e^2 t \, dt \text{(IV.9)}$$

where γ is an adjustment proportional gain, t represents time and e the error between the system dynamic and the desired dynamic. Figure IV.2 illustrates the optimization schema.

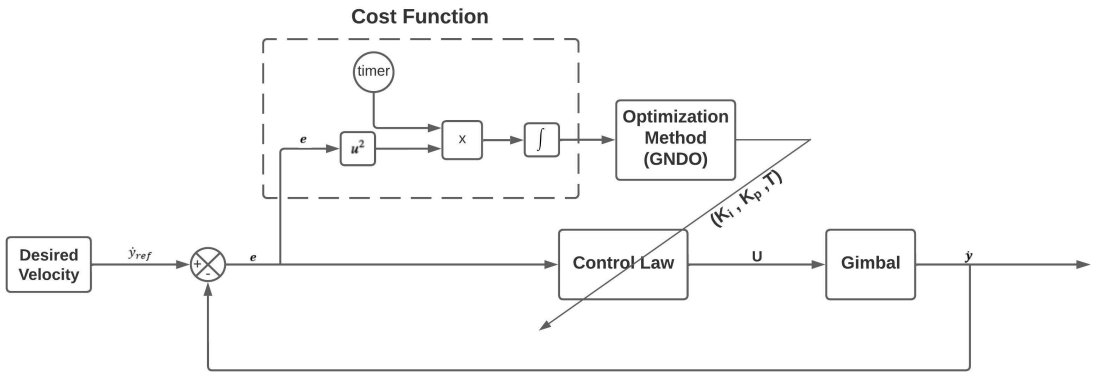


Figure IV.2: Optimization schema

IV.3 Advantages and constraints

- Model free control is highly flexible and can be applied without requiring an accurate mathematical model. This makes it suitable for controlling complex, nonlinear systems as the three axes gimbal.
- Model free control excel in adaptive control scenarios. It can adapt to change in the system dynamics, uncertainties, and disturbances. This adaptability makes it well-suitable for DSLR integration to gimbal's drone applications.
- This control method is generally easier to implement compared to model-based approaches. It involves fewer computational requirements and less complex algorithmic design. This simplicity reduces development and implementation time. It also reduces real-time computational process.

- Model-free strategy is robust against uncertainties and disturbances. It implicitly accounts for disturbances that may not be explicitly modeled like interchanged signals disturbances. This provides robustness even in the presence of unpredictable external factors.
- Model free control lacks optimality in terms of precision. Even if the performance is within large safety bounds it still is not optimal. Unlike model-based control techniques which take into account the specific dynamics of the three axes gimbal.

IV.4 Conclusion

In conclusion, the complex dynamics of the three axes gimbal has lead us to model free control instead of model-based approaches. Classic model free control lacks observance and adaptiveness which we had to compensate through intelligent PIDs. The proposed control involves an ultra-local model to represent the system dynamics and an intelligent PID to cancel disturbances. Real-time estimation technique was also discussed with a focus on a simple form for faster calculations. We have emphasized the need for optimizing the controller parameters with the GNDO algorithm. We have shed the lights on model free control advantages; flexibility, adaptability, simplicity and robustness.

V. Results and discussion

V.1 Introduction

This chapter discusses the obtained results. We have established here the system response to the control in the presence of external disturbances. We discuss how the velocities were eliminated and their effects on the camera angular positions. We have also discussed the control response to the disturbances which we have found in very interesting observations. Also, how much torque the system needed to keep the camera stable. The GNDO behavior have been discussed here with regards to the other optimizers performance. This section shows the optimizers differences and challenges.

V.2 Optimization results

Before we dive into the stats on the optimizers behavior we have listed some used parameters in table V.1. In order to keep the comparison just, the parameters shared among all four algorithms (ACO,GNDO and GWO) are the same.

Table V.1: Shared initial parameters for the optimization algorithms.

Parameter	Value
Upper bound	1000
Lower bound	0
Population size	50
Number of iterations	40

We have summarized in table V.2 the best extracted parameters we have extracted through the optimization process with some performance statistics.

Table V.3: Statistics on the optimization results.

Statistic parameter	Value
Standard deviation in the proportional gains	$[95.9237 \quad 121.0353 \quad 84.1273]$
Standard deviation in the integral gains	$[58.9873 \quad 163.9545 \quad 69.2524]$
Standard deviation in the estimation windows	7.0711×10^{-04}
Process time relatively to the highest	57%
Process time relatively to the lowest	157%
Accuracy relatively to the lowest	300%

Table V.2: Best parameters resulted via optimization.

Algorithm	Process time	Best result	Accuracy
ACO	8h 11 min	$K_p = [0.9350 \quad 0.9420 \quad 0.9950]$ $K_i = [0.8770 \quad 0.9770 \quad 0.0680]$ $T = 0.0211$	10^{-1}
GNDO	4h 38 min	$K_p = [260.1172 \quad 343.2796 \quad 213.4674]$ $K_i = 111.5818 \quad 442.9332 \quad 115.9526]$ $T = 0.0201$	10^{-4}
GWO	2h 57 min	$K_p = [200 \quad 171.0980 \quad 200]$ $K_i = [164.3281 \quad 343.5798 \quad 194.7709]$ $T = 0.0191$	10^{-2}

The accuracy column in table V.2 is the resulted error between the camera velocity and the desired velocity. This is when the system is controlled with the extracted parameters.

In term of process time, GNDO has taken a medium amount of time relatively to the other algorithms. Statistically just under 57% of the highest amount of time needed for ACO to do the same amount of iteration. Just so, GNDO has taken just under 157% of the lowest amount of time needed for GWO to do the iterations. Interestingly, even if GNDO has taken less time than ACO it has output 300% more accurate result. This time consumption is due to the relatively high number of simulations the GNDO needs per iteration to execute either exploration or exploitation.

In terms of gains, we have observed a great dispersion between the obtained gains from each algorithm. This does not provide a problem, because in the same parameters spaces, a control plant can have many combinations that gives good performance. While all algorithms have the same search spaces they have gone different ways through iterations. We have evaluated this dispersion through standard deviation which we have established in table V.3. The standard deviation in the proportional and integral gains

are very close to the actual gains values. This suggest a high dispersion. We however have found the estimation window values to be far less deviated. This is justified by the low standard deviation value.

To establish an overview on the GNDO behavior during the optimization problem, we have set 10 trials for each algorithm. Out of these 10 trials we have shown one of the resulted graphs that describes the GNDO best cost function (figure V.1). The graph represents the cost function behavior over iterations. In order to show the GNDO characteristics, we have established also the graph in figure V.2 which represents the mean cost function value over the 10 trials.

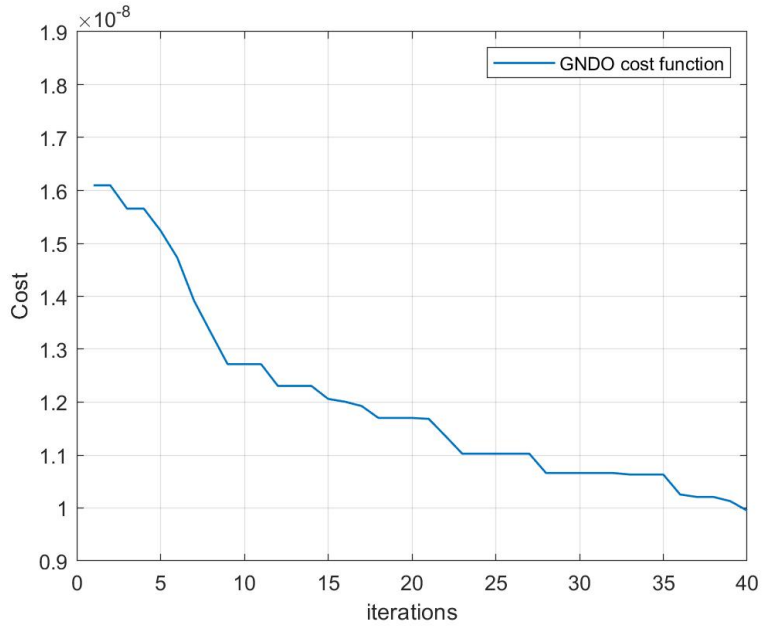


Figure V.1: GNDO cost function.

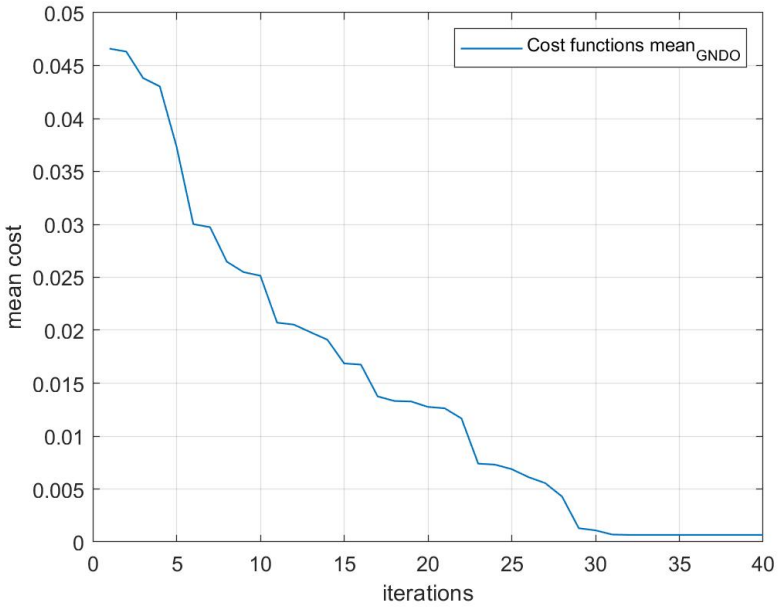


Figure V.2: GNDO cost function mean.

On figure V.2, we have observed the relatively constant convergence rate which is a positive indicator. It indicates a constant optimization over iterations. This is a favorable behavior in optimization regards. However, this rate is relatively low when compared to GWO converging rate as figure V.3 shows. Nonetheless, the same figure shows how the GNDO convergence rate is higher that the ACO's(figure V.4). The observed similarities between figure V.1 and V.2 shows the robustness of the algorithm towards its initial values. The algorithm has behaved the same over trials hence the similarities between a single trial graph and the mean graph.

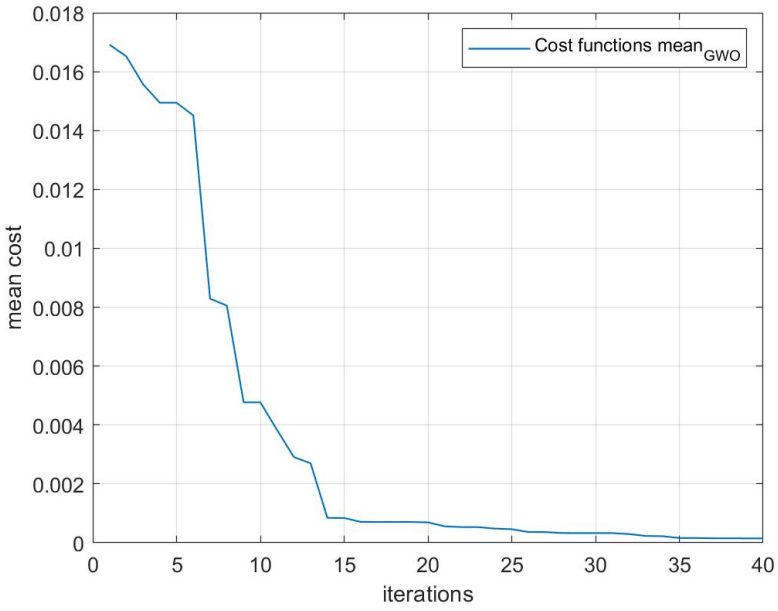


Figure V.3: GWO cost function mean.

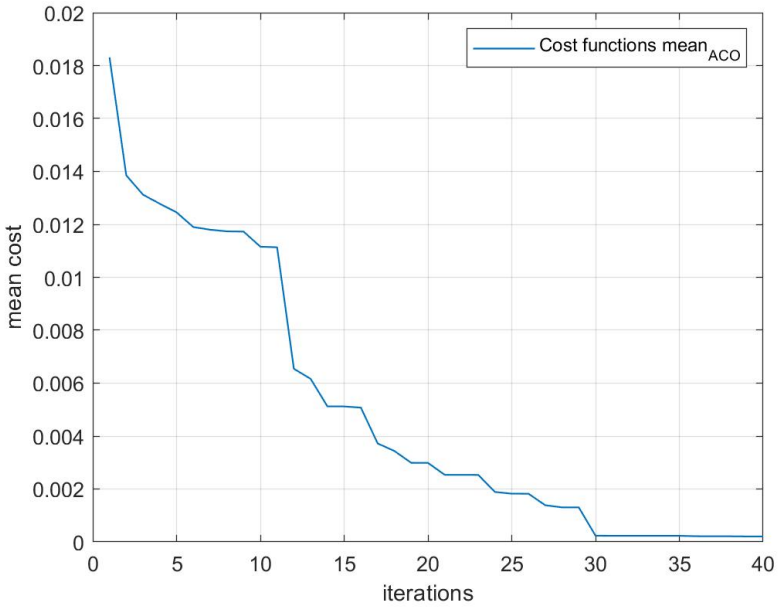


Figure V.4: ACO cost function mean.

Figure V.5 illustrates the costs means of all algorithms for more comprehensive overview.

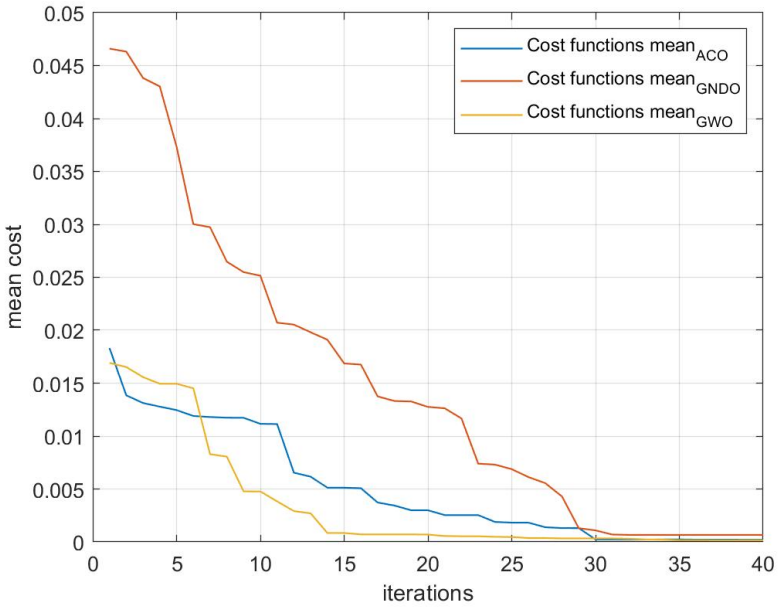


Figure V.5: Cost function means (10 trial).

Figure V.6 illustrates all resulted velocities opposed to the control-free system response.

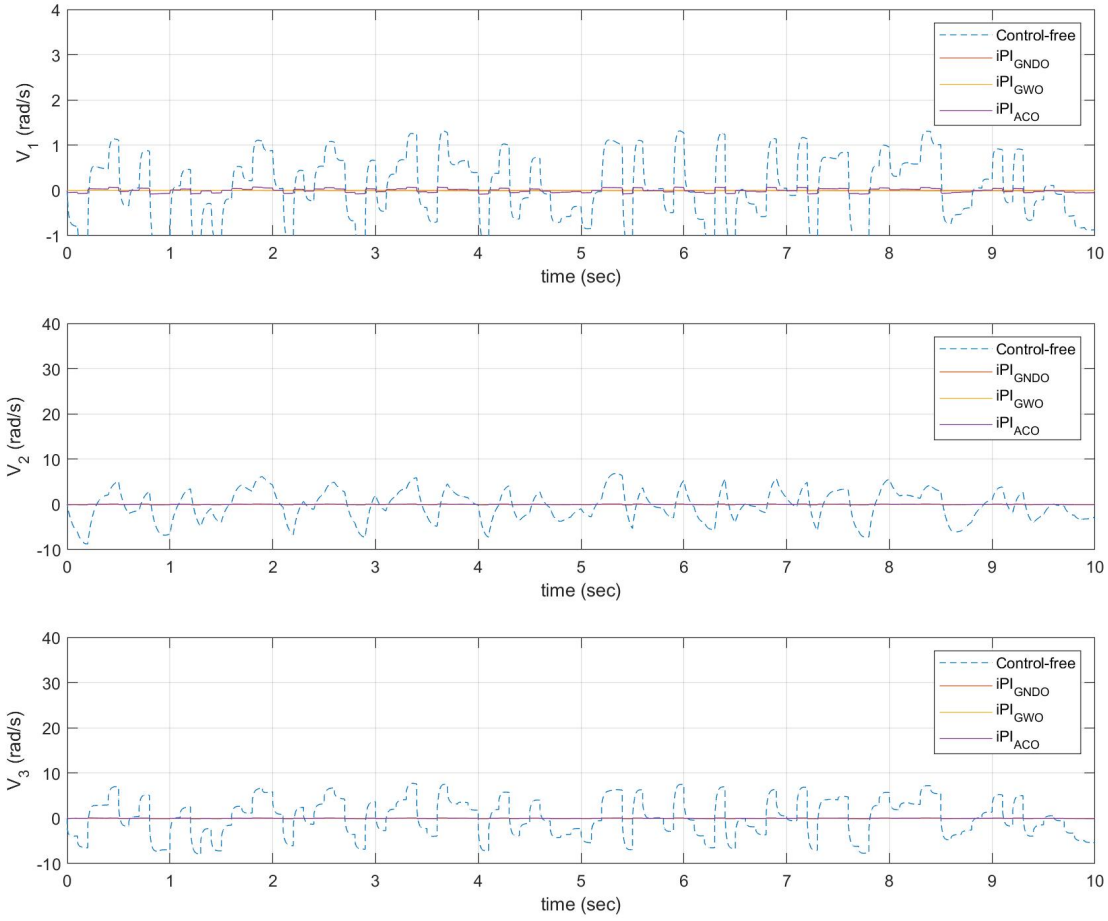


Figure V.6: Camera angular velocities with respect to optimized control parameters with applied disturbances

On this figure we have observed neutral velocity on the last two axes. On the first rotation axis we observe miner vibrations. These vibrations are due to the ACO optimized parameters as figure V.7 shows. On the same figure we have observed how the camera velocities has begone to fluctuate around zero when using the GWO optimized parameters.

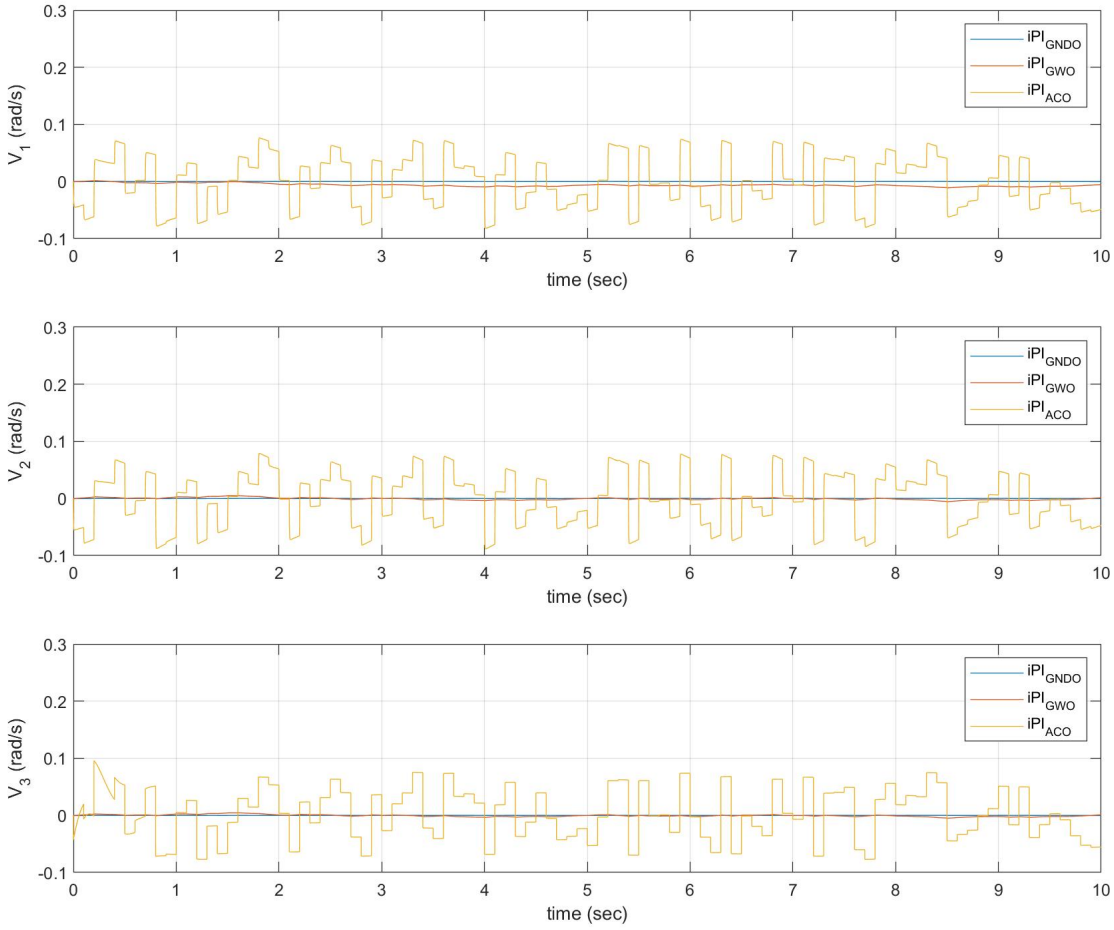


Figure V.7: Camera angular velocities with respect to optimized control parameters with applied disturbances

V.3 System response results

The system works under external disturbances that cause the camera vibration. However, we start by an ideal trial without applied disturbances to see the control the control response. Figure V.8 shows the camera angular velocities compared to the reference velocity.

The figure shows a low amplitude transition phase that maxed at 6.4×10^{-5} at the last angular velocity.

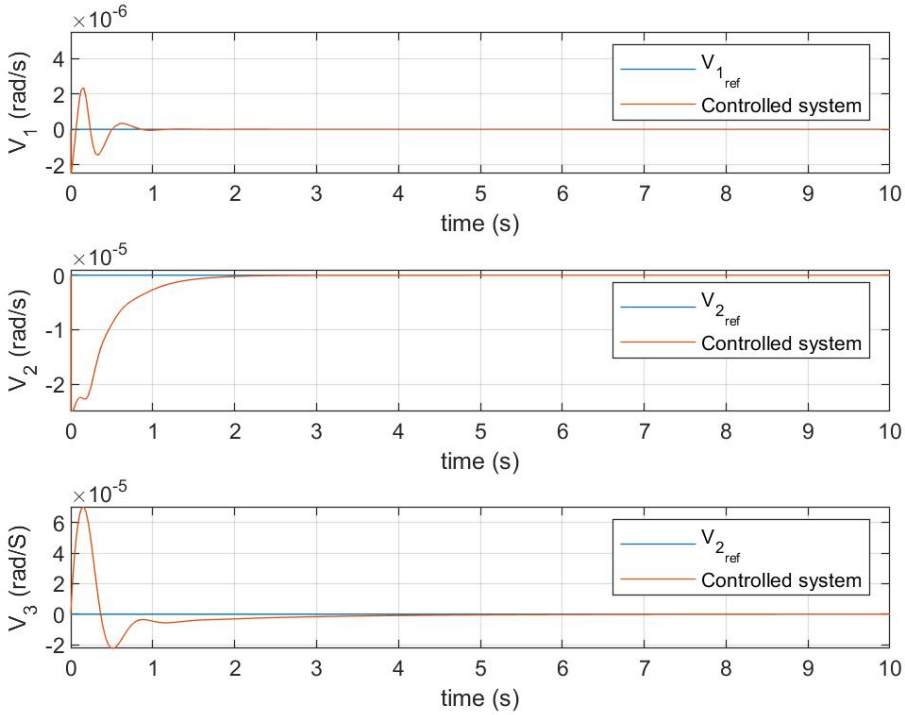


Figure V.8: Camera angular velocity with control and without applied disturbances

Figure V.9 illustrates the control torque with respect to this ideal trial. We observed how the control torque in the transition regime is high compared to the torque in the permanent regime.

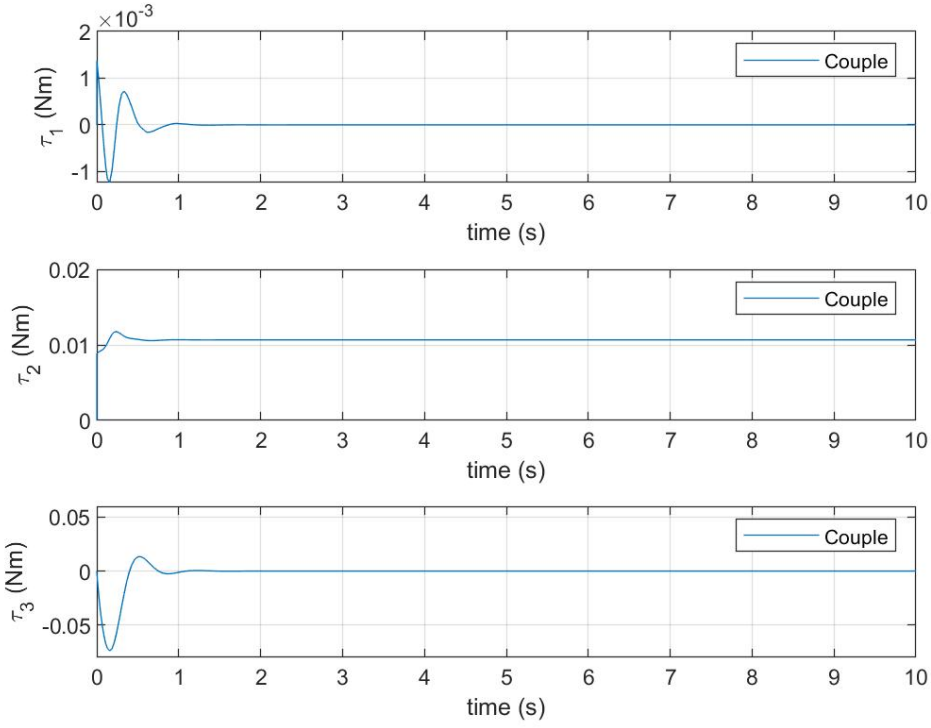


Figure V.9: Control torque without applied disturbances

We also observed that the torque in the second angle stabilized at a value different than zero to keep hold of the camera neutral velocity.

In order to show the system response with applied disturbances, with control parameters resulted by GNDO optimization, we illustrate the camera velocity (figure V.10).

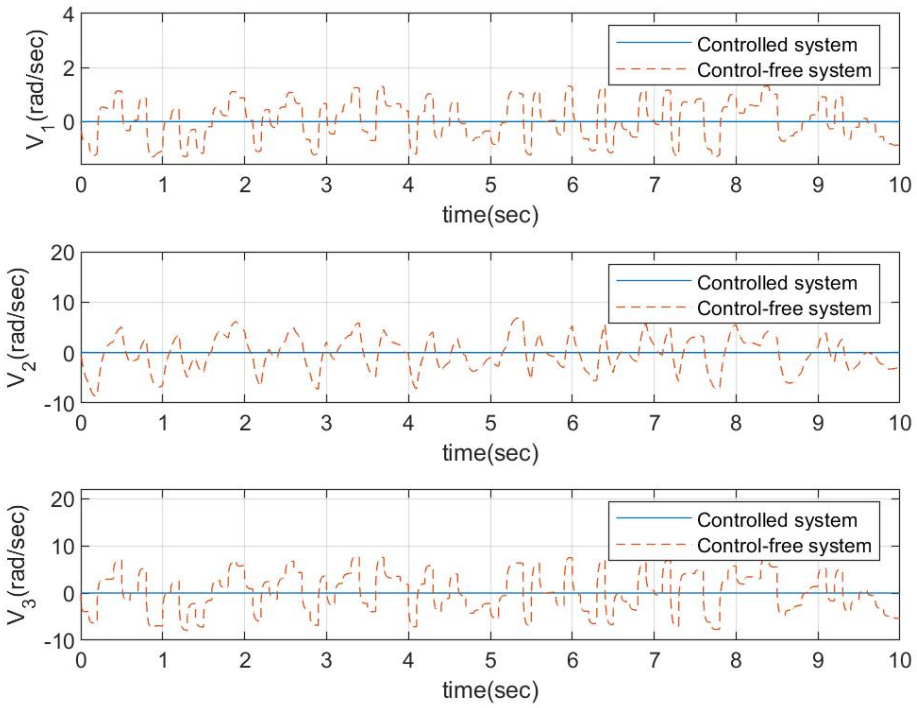


Figure V.10: Camera angular velocities with applied disturbances

We observe zero values on all three axes. In contrast of the velocities of the camera on the uncontrolled system which are fluctuated and picked up north of $7rad/s$ as shown in V_3 figure V.10. Both systems, controlled and uncontrolled have been subject to the same disturbances which are random signals up to $0.8Nm$. These disturbances range is high in camera vibration regards. Overall, the velocities were eliminated.

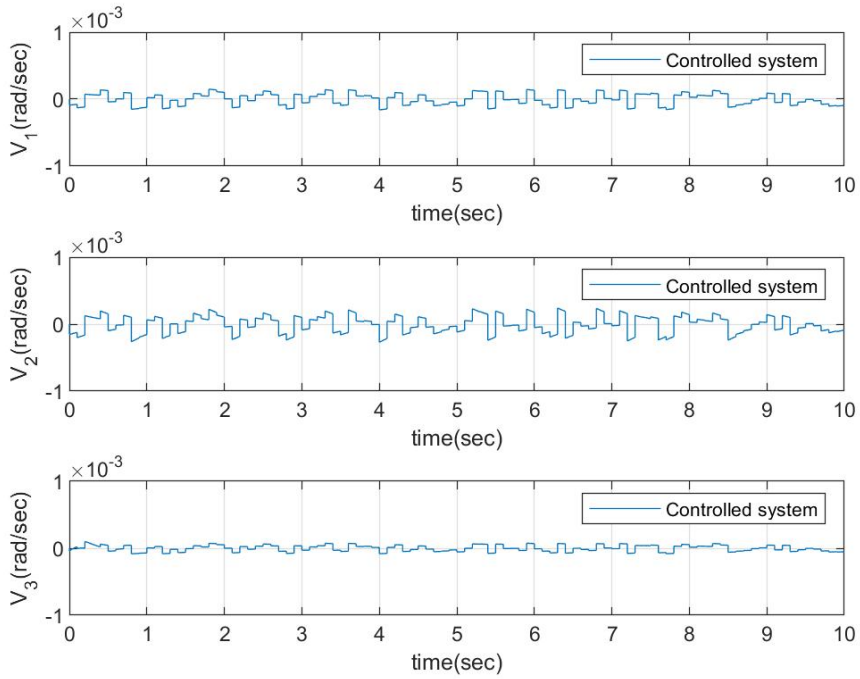


Figure V.11: Camera angular velocities

Figure V.11 shows a zoomed-in version of figure V.10, of the three angular camera velocities. It has shown an accuracy of a $10^{-4} rad/s$. This accuracy is enough to not appear on the camera as fluctuation. This results cane be confirmed through the camera angular position. The position have sticked to their initial values despite the external disturbances as figure V.12 shows.

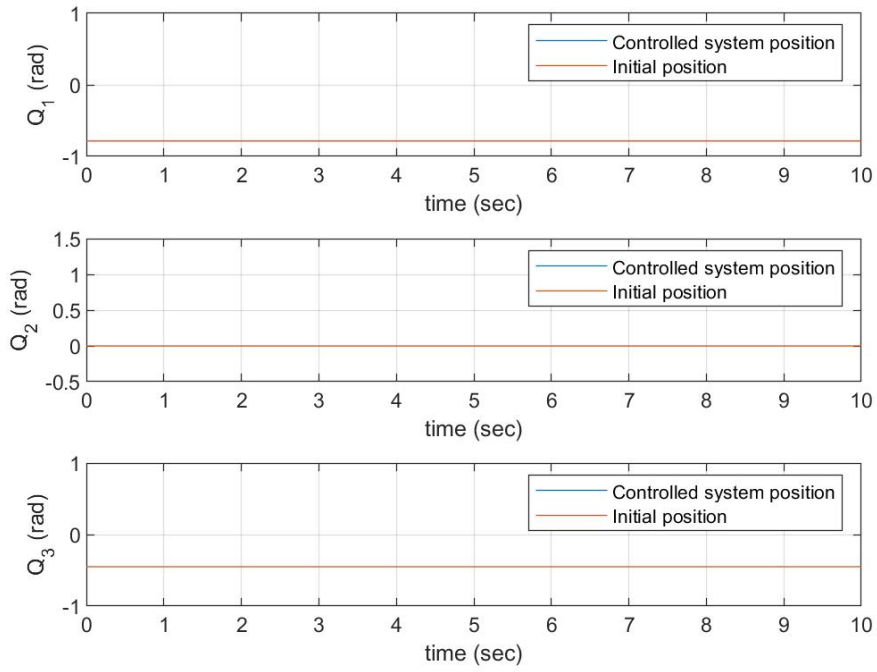


Figure V.12: Camera angular positions with applied disturbances

The initial values of angular position 1 and 3 are different to 0 due to conception planes. The conception is an assembly of many parts and each part needs at least one main sketching plan with attached axes. In assembly the differences in these planes orientations appear as angular positions. We joint here a capture of Simscape 3D visualizer during the trial to give a glam about this visualization process.

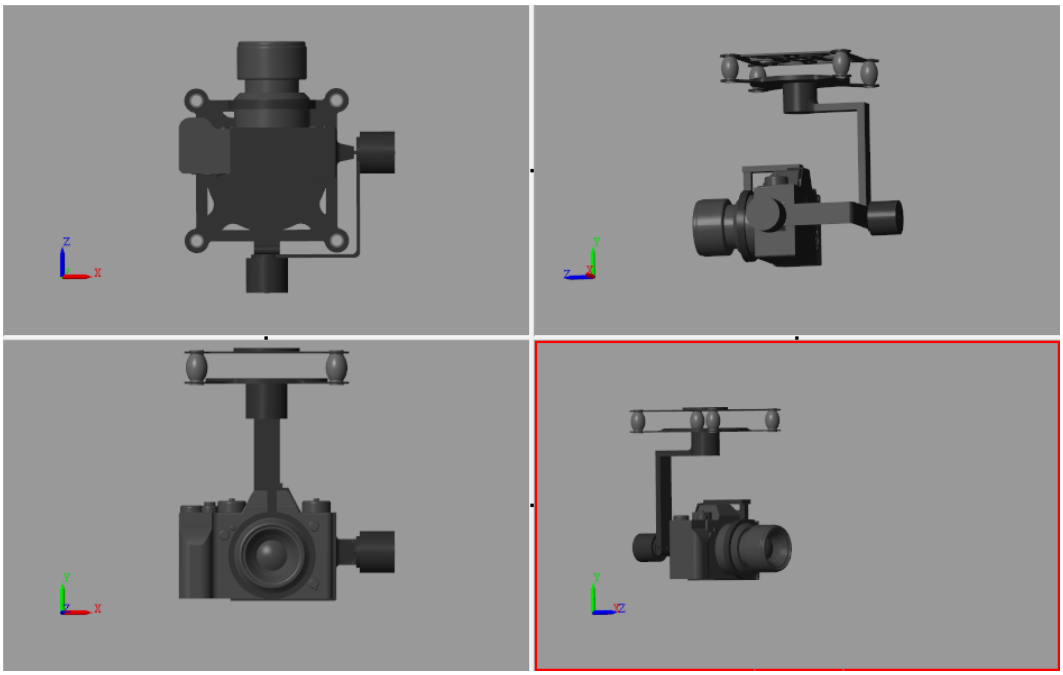


Figure V.13: Simscape capture

Figure V.14 shows a zoomed-in version of figure V.12 which have shown a $10^{-5} rad$ accuracy. The position stabilization due to vibrations velocities elimination has an accuracy too.

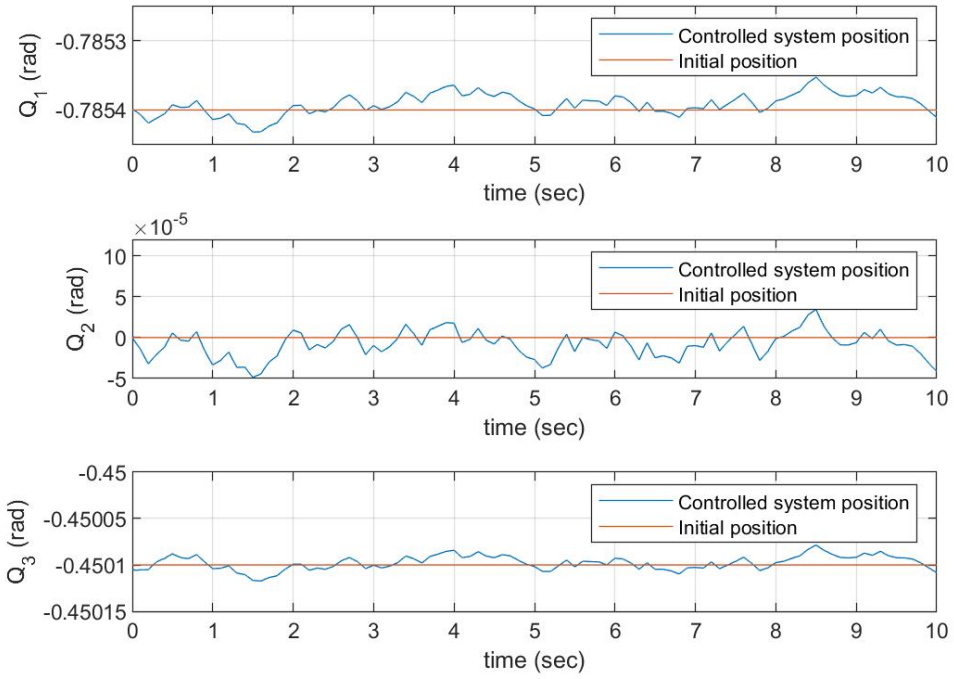


Figure V.14: Zoomed-in camera angular positions

The iPI is opposing fluctuating disturbances. As a result, the control should have a fluctuating behavior which is shown in figure V.15. The fluctuations are generated torques along the three axes of rotation. These torques represents the needed amount of couple in Nm to oppose the disturbances effect on the camera.

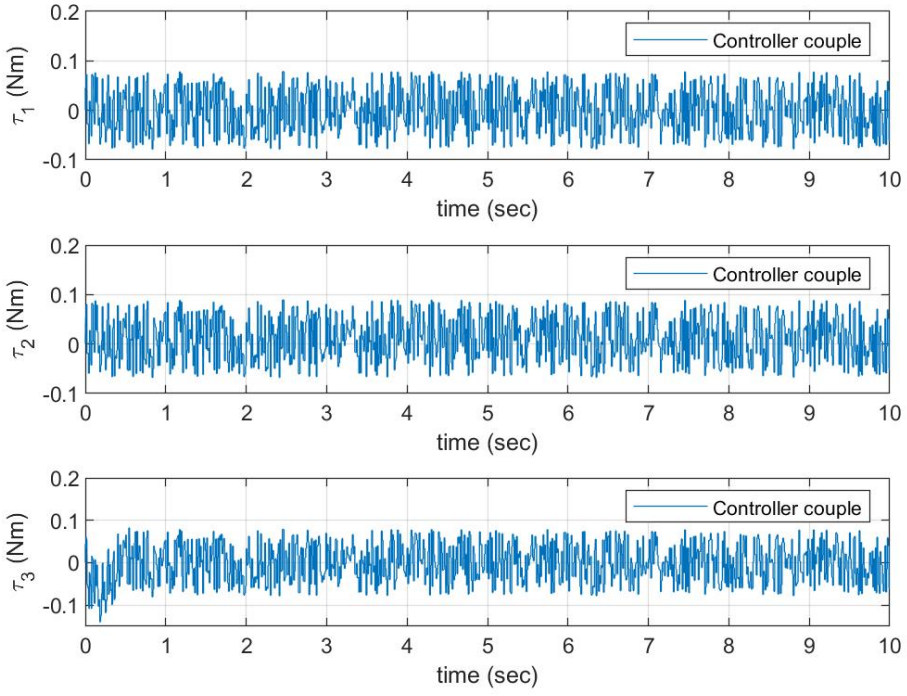


Figure V.15: Control torque with applied disturbances

An interesting behavior is observed when disturbances frequency is decreased. in this case the control have decreased its frequency accordingly as shown in figure V.16.

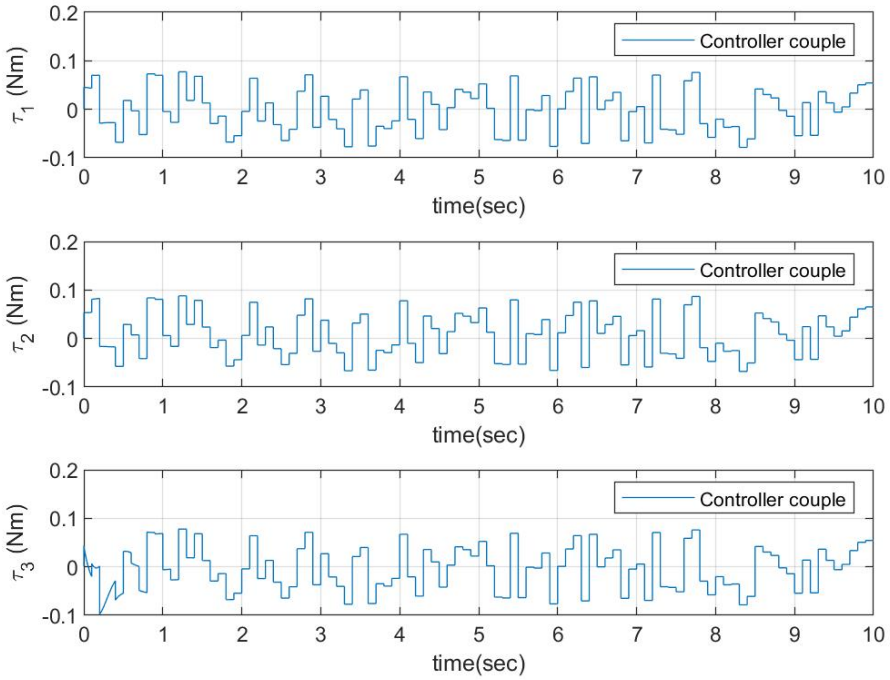


Figure V.16: Control torque with applied disturbances

In the same manner, when the disturbances frequency is increased the control generated torque frequency has increased dependently as figure V.17 shows.

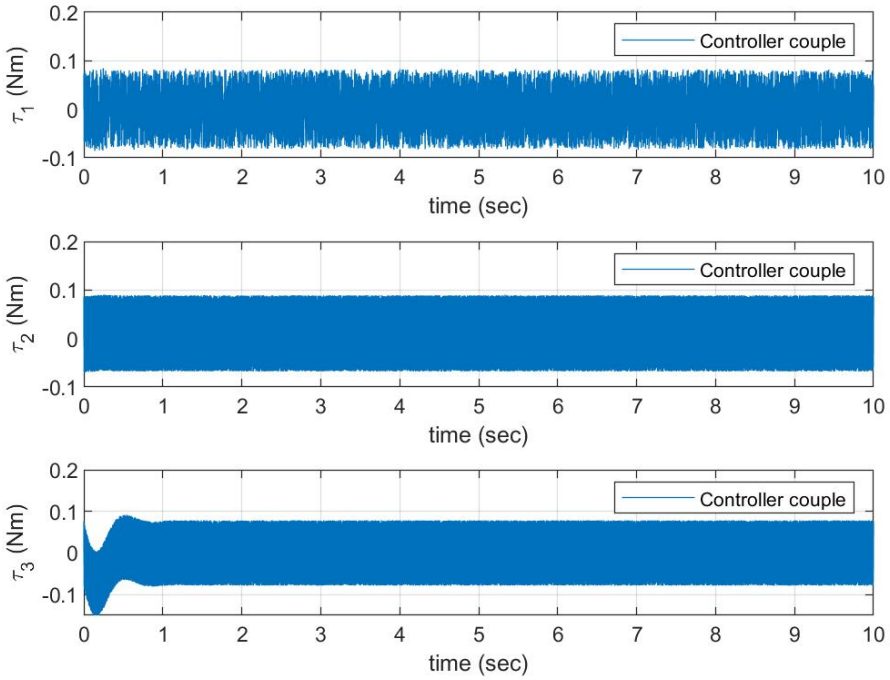


Figure V.17: Control torque with applied disturbances

The torque generated by the control has reached a maximum value of $0.1Nm$ in both positive, and negative directions. It is equivalent to a $1.019Kgfcm$. A $Kgfm$ or kilogram force centimeter, is the torque generated when applying $1Kg$ force to a $1cm$ length arm. In the longest case, the axis of rotation is no further than $8cm$ from the extreme vertex of the camera. The overall camera weight is north $400g$. All things considered, the generated torque is high enough to the expectations.

In figure V.18 we have illustrated the dynamics estimation $\hat{F}$ with respect to the on-control system acceleration. On the first angular position we have observed oscillations in the estimation around the system acceleration just before $1.5s$ of the simulation time. Over that, the estimation is superposed to the acceleration. Under highly zoomed in figure of the same first angular acceleration graph (figure V.19), we have observed the estimation to be actually an oscillating response within near zero value.

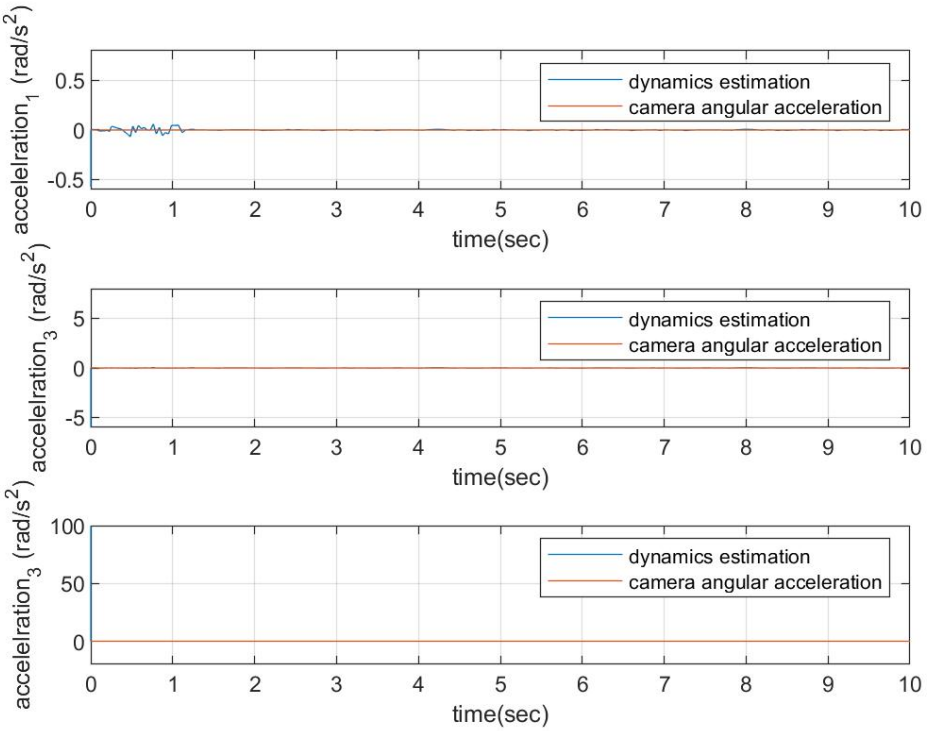


Figure V.18: Closed-loop estimation output

Both second and third estimation has shown also the same oscillating behavior with a sufficient accuracy. They however have started with a high estimated value, precisely $-6rad/s^2$ on the first angular position and $100rad/s^2$ on the third.

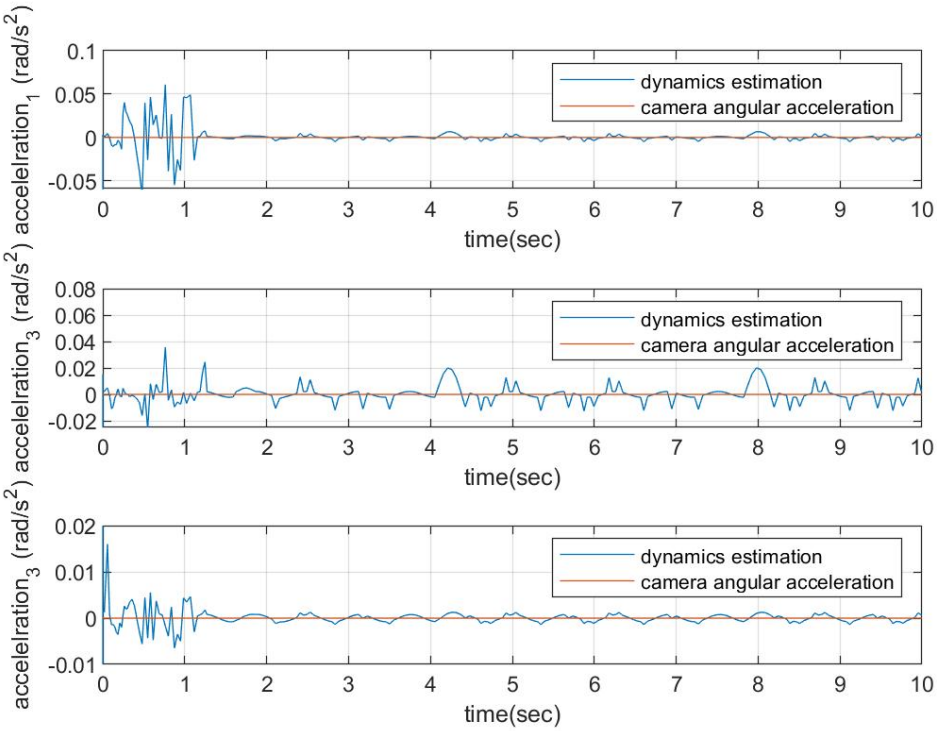


Figure V.19: Closed-loop estimation output

We have opposed the proposed model-free control to a classic PID in objective of comparison. Model-free control proved to be more efficient in this case, as figure V.20

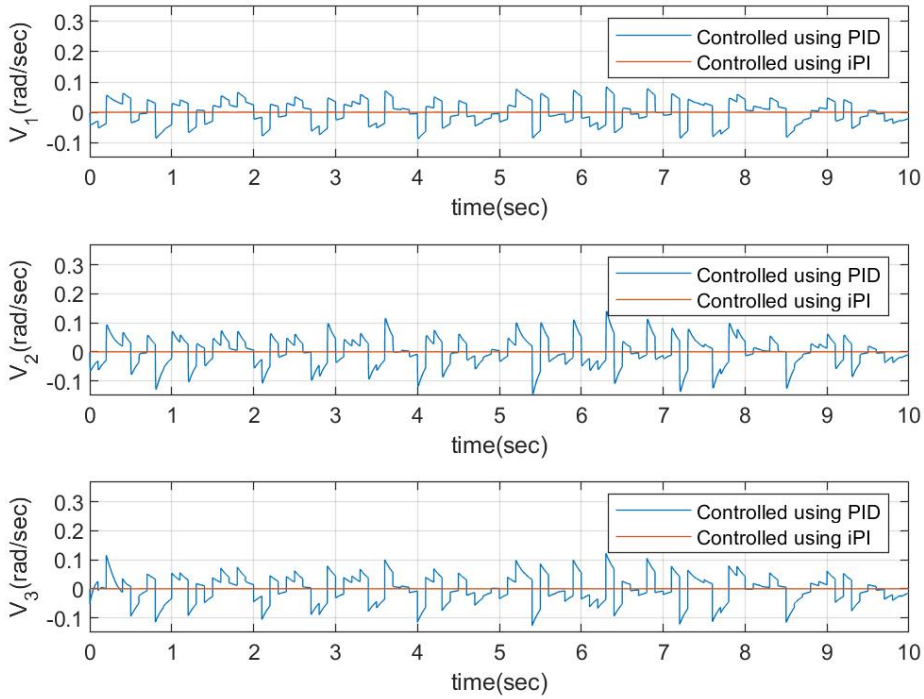


Figure V.20: Camera angular velocities.

shows the results in a zoomed-in capture of the camera velocities. The PID had an

accuracy of 10^{-1} unlike the 10^{-4} established with model-free control. However, it is only fair to state the unexpected good resulted shown by the PID. It was unexpected due the system dynamics complexity and the fluctuations in its response.

V.4 Robustness test

We have indicated previously that the proposed controller is robust against a range of shifts in the system dynamics. To prove the controller worthy we have extended the trial to a second system. The second system is a 3-axis smartphone gimbal. This system shares its main part with the first with slight differences in joints lengths and a smartphone instead of the camera (different mass, dimensions ...etc.). Figure V.21 shows the second system for illustration.

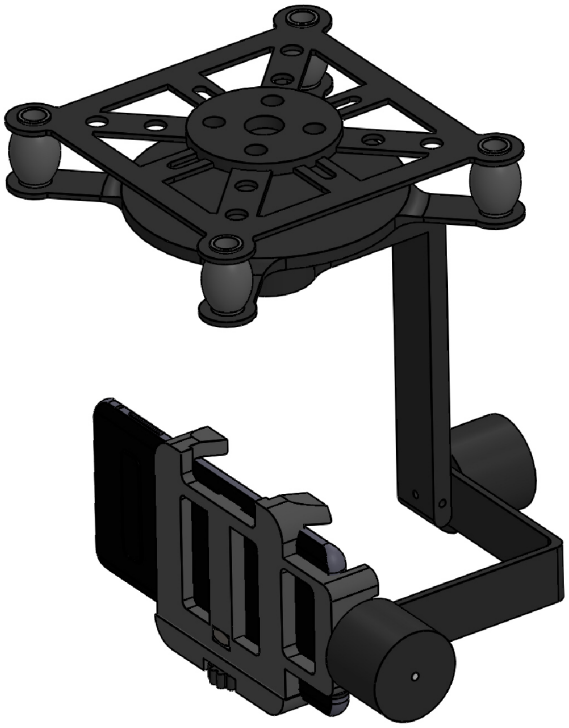


Figure V.21: Smartphone gimbal.

We have used the same controller with the same parameters. Figure V.22 shows the resulted smartphone angular velocities. The controlled system provided zero velocities again. Unlike the uncontrolled system which have provided similar behavior to the camera we have used before. The alteration we have made between the two systems provides changes in dynamic related parameters; inertia matrices, joints lengths and masses. Overall, we have proved the robustness of the model-free control.

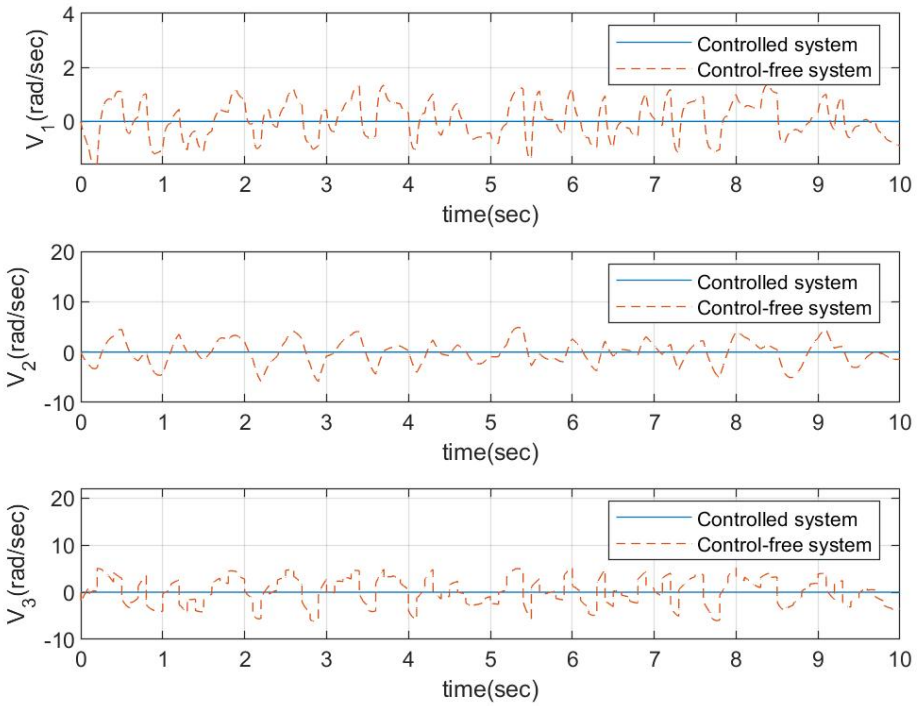


Figure V.22: Smartphone angular velocities

V.5 Conclusion

In summary, we have been able to neutralize the effect of external disturbances using GNDO optimized model-free control within a 10^{-4} accuracy. The control has been able to adapt to changes in the system dynamics, where we have used the same control parameters on a second gimbal system. The control has been also reactive to disturbances frequency as we have demonstrated. From the optimization perspective, GNDO has provided very sufficient performance compared to ACO and GWO within this context. We have found it however time and hardware consuming. GNDO has also shown a more consistent behavior of the trials.

VI. Conclusion

In this work we have synthesed an effective model-free control for a complex 3 axis camera gimbal that would support drone and handheld platforms for DSLR camera. This project went through several phases. First we have designed the gimbal head in computer aided design. We then have mounted a DSLR camera and a smartphone on the gimbal in two separate conceptions. Next we have modeled the system dynamics, with the camera mounted. We have observed from the comparison between the dynamic model response and the conception response some differences. Then, we have synthesized an iPI, a model-free controller which we have optimized using GNDO. The controller was able to elimante the camera angular velocities up to an error of 10^{-4}. Then we have tested the controller robustness using smartphone CAD conception. First, the test has shown the robustness of the controller. Second, it has proved the system capability for camera alternance. The GNDO algorithm is found to be greedy for simulations per iteration compared to the other used algorithms ACO and GWO. This makes it both hardware and time consuming. Also, the greediness of the optimizer for simulations has shed the light on a major issue with the proposed controller. An iPI or an $iPID$ is very sensitive towards its parameters. We have easily felt in flawed parameters combination through random initialization, exploration or even exploitation. This have made the optimization process very tedious. Overall, we have succeeded to create a virtual three-axis camera gimbal head with respected dynamics model and a robust control, for a variety of camera sets and platforms. This system can ensure mechanical camera stabilization. This project represents a substantial step towards a fully operational gimbal prototype. At this end, we have laid the foundation for it. As perspectives, we propose hardware realisation of the system and validation of this work through the real system.

VII. References

[1] W. Guilluy, L. Oudre **and** A. Beghdadi, **?**Video stabilization: Overview, challenges and perspectives,**?** *Signal Processing: Image Communication*, **jourvol** 90, **january** 2021, issn: 09235965. doi: 10.1016/j.image.2020.116015.

[2] M. R. Souza **and** H. Pedrini, **?**Digital video stabilization based on adaptive camera trajectory smoothing,**?** *Eurasip Journal on Image and Video Processing*, **jourvol** 2018, 1 **december** 2018, issn: 16875281. doi: 10.1186/s13640-018-0277-7.

[3] P. Rawat **and** J. Singhai, **?**Review of Motion Estimation and Video Stabilization techniques For hand held mobile video,**?** *Signal Image Processing : An International Journal*, **jourvol** 2, **pages** 159–168, 2 **june** 2011, issn: 22293922. doi: 10.5121/sipij.2011.2213.

[4] M. M. Abdo, A. R. Vali, A. R. Toloei **and** M. R. Arvan, **?**Improving two axes gimbal seeker performance using cascade control approach,**?** *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, **jourvol** 229, **pages** 38–55, 1 **january** 2015, issn: 20413025. doi: 10.1177/0954410014525130.

[5] B. Kürkçü **and** C. Kasnakoğlu, **?**Estimation of Unknown Disturbances in Gimbal Systems,**?** *Applied Mechanics and Materials*, **jourvol** 789-790, **pages** 951–956, **september** 2015. doi: 10.4028/www.scientific.net/amm.789-790.951.

[6] A. Altan **and** R. Hacıoğlu, **?**Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances,**?** *Mechanical Systems and Signal Processing*, **jourvol** 138, **april** 2020, issn: 10961216. doi: 10.1016/j.ymssp.2019.106548.

[7] R. J. Rajesh, **?**Camera Gimbal Stabilization Using Conventional PID Controller And Evolutionary Algorithms.**?**

[8] M. Granja, N. Chang, V. Granja, M. Duque **and** F. Llulluna, **?**Comparison between standard and modified Denavit-Hartenberg Methods in Robotics Modelling,**?** Avestia Publishing, 2016, isbn: 9781927877272. doi: 10.11159/icmie16.118.

[9] P. I. Corke, **?**A simple and systematic approach to assigning Denavit-Hartenberg parameters,**?** *IEEE Transactions on Robotics*, **jourvol** 23, **pages** 590–594, 3 **june** 2007, issn: 15523098. doi: 10.1109/TRO.2007.896765.

[10] **?**Simscape™ Multibody™ Getting Started Guide R2023a,**?** 2002. **url**: www.mathworks.com.

[11] M. Fliess **and** C. Join, **?**Model-free control,**?** **may** 2013. doi: 10.1080/00207179.2013.810345. **url**: http://arxiv.org/abs/1305.7085%20http://dx.doi.org/10.1080/00207179.2013.810345.

[12] M. W. Spong, S. Hutchinson **and** M. Vidyasagar, **?**Robot Modeling and Control First Edition.**?**

[13] Y. Zhang, Z. Jin **and** S. Mirjalili, **?**Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models,**?** *Energy Conversion and Management*, **jourvol** 224, **november** 2020, issn: 01968904. doi: 10.1016/j.enconman.2020.113301.

[14] M. Dorigo, M. Birattari **and** T. Stutzle, **?**Ant colony optimization,**?** *IEEE Computational Intelligence Magazine*, **jourvol** 1, **pages** 28–39, 4 **november** 2006, issn: 1556-603X. doi: 10.1109/MCI.2006.329691. **url**: http://ieeexplore.ieee.org/document/4129846/.

[15] S. Mirjalili, S. M. Mirjalili **and** A. Lewis, **?**Grey Wolf Optimizer,**?** *Advances in Engineering Software*, **jourvol** 69, **pages** 46–61, 2014, issn: 09659978. doi: 10.1016/j.advengsoft.2013.12.007.

[16] M. Fliess **and** C. Join, **?**Model-free control,**?** *International Journal of Control*, **jourvol** 86, **pages** 2228–2252, 12 **december** 2013, issn: 00207179. doi: 10.1080/00207179.2013.810345.

[17] M. Fliess **and** C. Join, **?**Commande sans modèle et commande à modèle restreint Michel Fliess, Cédric Join. Commande sans modèle et commande à modèle restreint. e-STA Sciences et Technologies de l'Automatique Commande sans modèle et commande à modèle restreint,**?** 2008, **pages** 1–23. **url**: https://hal.inria.fr/inria-00288107v3.

[18] M. Fliess, **?**Analyse non standard du bruit,**?** **jourvol** 342, 10 2006. doi: 10.1016/j.crma.2006.02.037ï. **url**: https://hal.inria.fr/inria-00001134.

ملخص

في هذا العمل، لقد قمنا باستعمال التحكم بدون نموذج على منصة ثلاثية المحاور بهدف الغاء اهتزازات الكامرا. قمنا أولا بإنشاء تصميم ثلاثي الأبعاد للمنصة. ثم قمنا بتطبيق التحكم على المنصة، و قد استطعنا إلغاء تأثير الاهتزازات على الكامرا. قمنا أيضا بعمل نموذج ديناميكي للمنصة و مقارنته مع ديناميكية التصميم. كذلك، قمنا بتحسين التحكم بدون نموذج للمنصة عن طريق عدة خوارزميات و قمنا بمقارنة أداء هته الخوارزميات.

كلمات مفتاحية: التحكم بدون نموذج، خوارزميات الأدلة العليا، منصة إلغاء الاهتزازات لدى الكامرا ثلاثية المحاور.

Abstract

In this work, we have synthesized a model-free control for a 3-axis camera gimbal in aim for camera stabilization. First we have made a CAD conception of the gimbal using SolidWorks. The control then was implemented on the conception using Simscape Multibody, and we were able to stabilize the camera on the three-axis. We have also made a dynamic model for the gimbal and compare it to the conception dynamics. We have also optimized the model-free control for the gimbal using GNDO, ACO and GWO and compare their performances.

Keywords : Model-free control, meta-heuristic optimization, 3-axis camera gimbal.

Résumé

Dans cette étude, nous avons réalisé une stabilisation de caméra sur une platforme à trois axes en utilisant une commande sans modèle. Tout d'abord, nous avons réalisé une conception CAO du plateforme à l'aide de SolidWorks. Ensuite, le contrôle a été implémenté sur la conception en utilisant Simscape Multibody, nous avons pu alors de stabiliser la caméra sur les trois axes. Nous avons également établi un modèle dynamique pour le plateforme et l'avons comparé à la dynamique de la conception. De plus, nous avons optimisé la commande sans modèle du cardan en utilisant GNDO, ACO et GWO, et avons comparé leurs performances.

Mots-clés : Commande sans modèle, optimisation méta-heuristique, plateforme de caméra (gimbal) à 3 axes.