RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

École Nationale Polytechnique
Département d'Automatique

End of Studies Project

For the attainment of an Engineer degree in Control Engineering

---

Active Disturbance Rejection Control Design for Twin Rotor MIMO Systems: Simulations and Experiments

---

BENCHIKH Aghiles & HADJ Yasser

Under the supervision of **Pr. BOUDJEMA Farès** and **Pr. BOUKHETALA Djamel**, ENP

Publicly presented and defended on 01/07/2024

Composition of the jury:

President:	Dr. CHAKIR Messaoud, ENP
Supervisors:	Pr. BOUDJEMA Farès & Pr. BOUKHETALA Djamel, ENP
Examinator:	Pr. BOUDANA Djamel, ENP

ENP 2024

Département d'Automatique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Alger Algérie.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

École Nationale Polytechnique
Département d'Automatique

End of Studies Project

For the attainment of an Engineer degree in Control Engineering

---

Active Disturbance Rejection Control Design for Twin Rotor MIMO Systems: Simulations and Experiments

---

BENCHIKH Aghiles & HADJ Yasser

Under the supervision of **Pr. BOUDJEMA Farès** and **Pr. BOUKHETALA Djamel**, ENP

Publicly presented and defended on 01/07/2024

Composition of the jury:

President:	Dr. CHAKIR Messaoud, ENP
Supervisors:	Pr. BOUDJEMA Farès & Pr. BOUKHETALA Djamel, ENP
Examinator:	Pr. BOUDANA Djamel, ENP

ENP 2024

Département d'Automatique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Alger Algérie.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

École Nationale Polytechnique
Département d'Automatique

Mémoire de Projet de Fin d'Études

En vue de l'obtention du diplôme d'Ingénieur d'état en Automatique

---

Commande par Rejet Actif de Perturbations pour le Twin Rotor MIMO System: Simulations et Expérimentation

---

BENCHIKH Aghiles & HADJ Yasser

Sous la supervision de **Pr. BOUDJEMA Farès** et **Pr. BOUKHETALA Djamel**, ENP

Présenté et soutenu publiquement le 01/07/2024

Composition du jury:

Président:	Dr. CHAKIR Messaoud, ENP
Promoteurs:	Pr. BOUDJEMA Farès & Pr. BOUKHETALA Djamel, ENP
Examinateur:	Pr. BOUDANA Djamel, ENP

ENP 2024

Département d'Automatique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Alger Algérie.

ملخص

في هذا المشروع، تمت دراسة التحكم النشط المقاوم للإضطرابات كتقنية جديدة للتحكم في الأنظمة دون الحاجة لمعرفة كل المعلومات عنها وكمنهجية كاسرة للحدود بين الديناميات الخطية واللاخطية التي لطالما عرفتها تقنيات التغذية الراجعة التقليدية. تم إجراء تحاليل عن طريق تجارب ومحاكات على Twin Rotor MiMo System لتسليط الضوء على الصعوبات التي تواجه هذه التقنية في الميدان العملي والتجريبي. بعد ذلك، تم الاقتراح والتحقق من صحة حل بسيط لمواجهة هذه العقبات.

الكلمات المفتاحية: التحكم النشط المقاوم للاضطرابات – Twin Rotor MIMO System – Particle Swarm Optimization – الاضطرابات الكلية – تشبع المحرك.

Résumé

Dans ce projet, la Commande par Rejet Actif de Perturbations est étudiée comme une nouvelle technique de commande exploitant un minimum d'informations sur le système et brisant les frontières entre la linéarité et la non-linéarité dans la théorie de la commande par bouclage classique. Une analyse par simulation ainsi que des expérimentations sont ensuite réalisées sur le Twin Rotor MIMO System pour mettre en évidence les défis de cette approche dans des applications du monde réel, et une solution simple à ces problèmes est proposée et validée par la suite.

Mots clés : Commande par Rejet Actif de Perturbations - Twin Rotor MIMO System - Optimisation d'Essaim de Particules - Perturbations Totales - Saturation des Actionneurs.

Abstract

In this project, Active Disturbance Rejection Control is studied as a new model-free control framework that breaks the boundaries between linearity and Nonlinearity in classical feedback loop control. A simulative and experimental analysis is then conducted on the Twin Rotor MIMO System to highlight the challenges of this control approach in real-life applications. Subsequently, a simple solution to these obstacles is proposed and validated.

Keywords : Active Disturbance Rejection Control - Twin Rotor MIMO System - Particle Swarm Optimization - Total Disturbances - Actuator Saturation.

Dedications

I dedicate this work to my mother, my father and my grandfather allah yerehmo, for their constant support, their encouragements from the day i was born and their motivation.

My dedication extends to my brothers, aunts, V; and loved ones. A special dedication for Chakawi members.

Finally, To my ENSTP friends that supported me and shared with me special moments.

- Aghiles

To my loved ones.
- Yasser

Acknowledgements

We would like to thank everyone who contributed to the success of our study and who helped us throughout the process writing of this thesis.

First of all, we would like to thank our thesis supervisor, Pr. Boudjema and Pr. Boukhetala from the Ecole Polytechnique of Algiers, for their availability and their advice, which helped sharpen our thinking skills and taught us how to safely handle an experiment without losing track of our logical train of thought. More specifically, We want to thank Pr. Boukhetala for his precious effort in setting up the environment for the experiments after years of it not working, and Pr. Boudjema for being flexible with our ideas and for giving us the autonomy that we wanted while still keeping us on the right path.

We extend our gratefulness to the entire academic staff of our college, especially our dear professors who have greatly contributed to our development throughout our academic journey ever since our preparatory classes.

Contents

List of Figures

List of Tables

Acronyms

- **ADRC** : Active Disturbance Rejection Control
- **PID** : Proportional Integral Derivative
- **TRMS** : Twin Rotor MIMO System
- **ESO** : Extended State Observer
- **MIMO** : Multi-Input Multi-Output
- **LQG** : Linear Quadratic Gaussian
- **LQR** : Linear Quadratic Regulator
- **MPC** : Model Predictive Control
- **TRAS** : Twin Rotor Aerodynamical System
- **SMC** : Sliding Mode Control
- **NN** : Neural Networks
- **EKF** : Extended Kalman Filter
- **LO** : Luenberger Observer
- **LADRC** : Linear Active Disturbance Rejection Control
- **TD** : Tracking Differentiator
- **SNR** : Signal-to-Noise Ration
- **SEFC** : State Error Feedback Controller
- **SISO** : Single-Input Single-Output
- **TITO** : Two-Input Two-Output
- **NL** : Nonlinear
- **SELG** : Small Error Large Gain
- **LESG** : Large Error Small Gain
- **LESO** : Linear Extended State Observer
- **GPD** : General Proportional Derivative
- **NLPD** : Nonlinear Proportional Derivative
- **I/O** : Input/Output
- **HIL** : Hardware-In-the-Loop
- **PSO** : Particle Swarm Optimisation
- **NLADRC** : Nonlinear Active Disturbance Rejection Control
- **PDF** : Positive Definite Function

General Introduction

"It may help the newcomers to ADRC greatly if one abandons the initial How can this be right? attitude and, instead, run a few simulations of the proposed solutions and observe the results."

Jingqing Han, [1] - 2009

Motivation

Control systems are described by a chain of sensors, controllers, and actuators tied to processes. When a sensor takes the output of a process and forwards it to a controller, who in turn gives the actuators a set of commands to follow so that it affects the process in a way that makes it behave according to a desired dynamic, this chain is called a feedback loop. Feedback loops have been crowned in the classical control theory as the dominant chain of actions. The most important element in a feedback loop is the controller and the observer (equivalent of a sensor in the case where the controlled signal is not measurable). Designing a controller or an observer often relies on the mathematical model of the systems, in which linear and non-linear design techniques have been proposed.

The idea that the system's mathematical model takes part in the design process quickly became a liability rather than a functionality, and the closed loop became sensitive to modeling errors and/or parametric variations. Therefore, robustness against internal and external disturbances became an essential requirement in feedback loops. A lot of research efforts have been conducted and many solutions have been proposed to fulfill this requirement, but it was done in an implicit manner for the most part.

Motivated by the practical demands from the industry to surmount the problems with the model-based approaches (especially the PID control framework) and the new challenges for systems that are subject to more uncertainties, the idea of disturbance estimation and rejection has been fostered by Active Disturbance Rejection Control. J.Han proposed this almost model-free control technology in [1] as an experimental control ideology, where the focus was more towards industrial applications than theoretical research.

Active Disturbance Rejection Control came as a new control framework that uses very little information about the system and can be considered as a model-free control approach. The main line of thought in ADRC is that the system's dynamics are to be overcome by the control law anyway, so why not go ahead and do that explicitly? Getting rid of all dynamics of the system made the process of controller and observer design very easy and almost model-free. This gives an ADRC-based closed loop an intrinsic robustness property against anything that was overcame explicitly, and the performance requirements were easily met.

Helicopters are an important means of transportation, and safety has always been regarded as

a very crucial requirement for aerial mobile systems. Since a lot of aerodynamic phenomenons take place in a flight episode, deriving an accurate mathematical model of such systems is a challenging task, let alone automating them in a performative and robust manner. The Twin Rotor MIMO system is a helicopter lab simulator that was developed by Feedback as a control system for research purposes. This system mainly focuses on reconstructing the interconnection between a helicopter's main and tail propellers.

The TRMS control is challenging because its model is a complex high-order nonlinear system with significant cross-couplings and some inaccessible measurements. The high order of the system results in it having many resonant vibration modes, which can lead to high oscillations or instability when excited. Additionally, the aerodynamic thrusts in this multi-variable system are produced by varying the rotor speeds, which cannot change instantaneously. This leads to a time delay in the system, which is a major cause of instability in feedback systems. Therefore, many assumptions and simplifications are being made while modeling the system, but it causes the problem of model uncertainty when applying model-based control and optimization algorithms in experiments.

State of the Art

A historical perspective on control theory can be found in [4], with a detailed account of both classical and modern control theory in [5]. However, since this thesis talks more about ADRC, an overview of the theoretical progress of this control approach is more worthy of presenting here. Following the emergence of ADRC as an experimentally oriented approach, a lot of theoretical work has been done to justify why it solved many industrial challenges, even though its core idea was fairly simple. The stability characteristics of linear ADRC for nonlinear time-varying plants subject to vast dynamic uncertainties were first addressed in [6] revealing that both estimation and tracking errors are bounded with their bounds proportion to the bandwidths of linear ESO. The global and semi-global convergence of the nonlinear ADRC for a class of MIMO nonlinear systems with large uncertainty were investigated in [7]. On the other hand, An adaptive ESO with time-varying observer gains was proposed for nonlinear disturbed systems, and the convergence proof of estimation errors was presented in [8]. The backstepping ADRC design was proposed for uncertain nonlinear systems, and the corresponding closed-loop convergence was established in [9]. More work has been done to contribute to the theory of ADRC, but it remains an experimental approach at its core.

A lot of research work has been conducted to control the TRMS using different control techniques, whether model-free or model-based. [10] presented a detailed literature review of the many control algorithms applied to the TRMS. The method of input shaping was proposed for the TRMS but it does not achieve the objective of controlling the position and is often combined with feedback control methods. Various types of PID controllers have been proposed where the controller's gains are obtained either by heuristic tuning or by optimization processes. However, it has a limited operation range and often leads to oscillations in higher-order nonlinear systems. A proposed solution was augmenting the PID controller with other control algorithms (such as Fuzzy Logic and Feedforward control) to improve the performance. Linear optimal controllers were proposed to find a control law that satisfies a certain optimal criterion. LQG compensator was augmented with a command prefilter to moderate the requirement of high control energy and reduce vibration. LQR was proposed using state and output feedback. These controllers achieve robustness performance but they operate in a narrow horizon due to being limited to the linearized system. MPC uses the linearized systems of the current prediction horizon to form an objective subject to inequality constraints based on the systems'

limits. While this approach has the advantage of handling the system's constraints, it is computationally quite expensive and requires high-fidelity systems models to achieve satisfactory performance. The perk of using intelligent controllers is that they can be independent of complex nonlinear systems models. Classical and adaptive fuzzy controllers have been proposed, and combined with PID, LQR, and SM control algorithms to improve the performance. The inconvenience of fuzzy controllers is that they depend on the number of membership functions, which can be very difficult to tune. using artificial NNs, an adaptive nonlinear model inversion controller was implemented in real time for the TRMS. The feedback controller and adaptive NNs were integrated to compensate for possible model inversion errors. Reasonable tracking response was exhibited in the presence of inversion errors caused by model uncertainty.

On the other hand, SMC is popular thanks to its robustness property, but the chattering phenomenon motivated several reduction methods. Fuzzy-Sliding and Fuzzy Integral Sliding control was proposed for the decoupled TRMS model, which considered the coupling effects as uncertainties. Chattering was drastically reduced while the systems remained robust to external disturbances. Classical and adaptive 2nd order SMC have been proposed, but the calculation of the control laws was quite complex and tracking of sinusoidal signals could be further improved.

H_{inf} and L_2 controllers with nonlinear extensions were also applied. they could be interpreted as PID controllers with dynamic gains. The robustness property of these methods makes them applicable to the cross-coupled MIMO model of the TRMS, but tracking of time-varying references was not investigated. Feedback linearization, which consists of the cancellation of the systems' nonlinearities and then applying linear control methods externally on the resulting linear systems, was introduced to TRMS via an external LQR, and with a local state observer or an EKF in the case of output feedback linearization, and the output response using the EKF showed significant sine wave tracking errors.

Backstepping control is applicable in the TRMS model. Adaptive backstepping was applied along with a neuro-adaptive observer but it showed significant sine wave input tracking errors, and with a LO but it showed high overshoots resulting from step changes to the reference. A switched-step integral backstepping control scheme was designed to switch between two candidate controllers obtained at different steps of the backstepping design process. Experiments showed a reduction in power consumption, saturation of the control signal, and visible motor jerking. The controller was also robust to external disturbances.

Structure of the Thesis

In this project, we will present an in-depth study of ADRC and design an optimal ADRC structure for the control of the TRMS. Simulations and experiments will also be conducted to verify whether the simulations' results translate into an effective control structure in the experimentation. More importantly, we will address a new shortcoming of ADRC and propose a simple way around it in real-life implementations. Our thesis is structured as follows:

Chapter 1 will discuss Active Disturbance Rejection Control as a new control framework as well as its different functional blocks. The theory behind ADRC will be presented starting from some remedies to the problems with PID control to describing it as a new way of thinking when it comes to control design and feedback loops.

Chapter 2 presents the Twin Rotor MIMO System as a helicopter lab simulator. Its hardware and software components are also briefly presented. But more importantly, a nonlinear model

was derived from Feedback's manual along with its linearized model around the origin. The stability, controllability, and observability of both models are then analyzed.

Chapter 3 tackles the tuning of linear and nonlinear ADRC's SFEC and ESO through simulations-run optimization-based pole placement. The adaptation of the TRMS' mismatched nonlinear model to its matched form was first done so that the ADRC closed loop holds. Particle swarm optimization is briefly presented and executed to find the optimal parameters. A comparison between LADRC and ADRC concludes the chapter.

In **Chapter 4**, the ADRC algorithm is implemented in the real TRMS. The issues that arise are identified one by one and a solution to each one is presented and implemented as a contribution. The LADRC's controller was refined in terms of its structure and its parameters due to the control input limitations altering the compensation of the total perturbation when it takes large values. The implementation was successfully done for a stabilization problem and disturbance rejection.

Chapter 1

Active Disturbance Rejection Control

1.1 Problem Statement

In control theory, feedback loops have been the dominant structure to control all types of plants. The main element of a feedback loop is the controller: a dynamic bloc that takes as input the error between the reference signal(s) and the plant's signal(s) and outputs a control signal(s) that shapes the dynamics of the controlled system in a way that the output of the closed loop follows the desired input.

One of the most vital characteristics of a controller is performance and robustness. Many control theorists have been trying to develop a controller that apports both of these properties to the closed loop. The most commonly used controller is the Proportional Integral Derivative controlled –denoted as PID–, in which the control law is generated by the following linear combination:

$$u = K_p\, e + K_i \int e\, dt + K_d\, \frac{de}{dt} \tag{1.1}$$

where $e = y_r - y$ is the tracking error (in which $y_r = 0$ in the case of a stabilization problem), and:

- The proportional term produces an output that is proportional to the current error value. It helps reduce the error by adjusting the control output. Increasing K_p will increase the response speed but may cause overshoot and oscillations if too high.
- The integral term sums the error over time and produces an output that is proportional to the accumulated error. It eliminates the residual steady-state error that occurs with a pure proportional controller (depending on the number of integrators of the system). It ensures that the control output continues to increase (or decrease) until the error is zero. However, if is too high, it can lead to instability and increased overshoot.
- The derivative term produces an output that is proportional to the rate of change of the error and predicts based on its future trend. It provides a damping effect, which helps to reduce overshoot and improve stability. If K_d is too high, it can amplify noise in the error signal and lead to erratic control behavior.

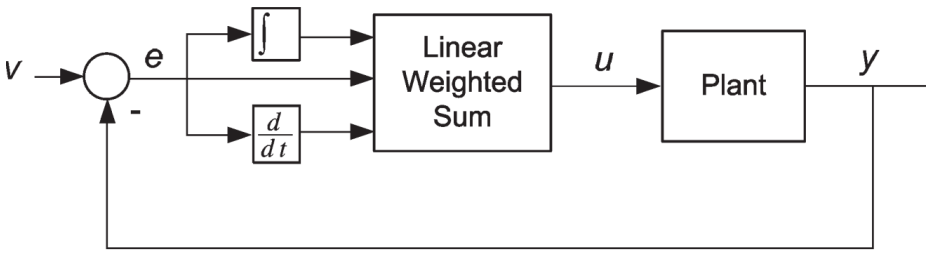


Figure 1.1: PID Control Topology [1]

With this simple yet effective combination, the control of a wide variety of plants became a very easy task, solidifying PID as the most commonly used controller for a long period of time. However, it presents many inconveniences that are hard to overlook. J.Han stated in [1] that the classical PID-based control loop presents four major inconveniences:

a. The sudden jump in the value of the reference, when described as a step signal, is not suited for all types of dynamic systems. For example, a lot of highly interconnected MIMO systems are very sensitive to any variation of any one of the output signals, which is called the cross-coupling interaction phenomenon. Therefore, asking the output(s) of these systems to make a sudden jump in value can lead to the instability of the closed loop, amounting to horrible damages in real-world applications.

b. The derivative term in the PID formula (1.1) causes a noise degradation on the controller (i.e. a loss in robustness against input noise) due to the differentiation method used. Therefore, the noise in the feedback error signal negatively affects the constructed control signal.

c. The oversimplification of the control law causes a loss in performance and robustness. For highly Nonlinear systems, the weighted sum in (1.1) is way too simple to be considered the best control law. Focusing only on the linear combination of the present, accumulative, and predictive feedback errors is not the best choice when there are much more effective combinations available.

d. The integral term in the PID formula (1.1) introduces new problems such as saturation and reduced stability margins due to the phase lag. This term is also critical to eliminate the steady-state error, making it obligatory to counter the complications it implies.

These disadvantages rendered using the PID framework safely in real-life applications, especially in the control of industrial plants, a very challenging –and sometimes impossible– task. That is the main motivation behind finding a solution to these problems in the form of a new control scheme that is easily and safely applicable to control industrial plants.

In this chapter, we will get you familiar with this new control approach. In Section 1.2, we will introduce ADRC as a new control framework that solves the problems in PID control, as well as the main concept behind it. Sections 1.3, 1.4, and 1.6 detail the main blocs of ADRC as a control scheme. Section 1.5 presents the main Nonlinear function to be used in these blocs. Finally, in Section 1.7, these blocs are put to use together in what is called the ADRC algorithm.

1.2 From PID to Active Disturbance Rejection Control

J.Han has worked on this new control paradigm for years. His work can be found in Chinese literature dating from 1989 to 2007, laying the ground for one of his biggest pieces of research in the domain. In 2009, he published [1] in English as a fundamental compilation of all his previous works, introducing remedies to the major problems with PID and formally combining them in what he called Active Disturbance Rejection Control (ADRC).

1.2.1 Remedies to the problems with PID

Within the core of ADRC lays its main idea, and around its shell are presented the remedies to the previously mentioned problems with PID control. For each issue, a solution was presented:

a. The set-point jump issue was solved by introducing a transient profile as a reference trajectory, shaping a smooth trajectory to the desired value. The convergence speed is also tuned using specific parameters in the transient profile generator's equation, which will be detailed in Section 1.3. This simple yet effective bloc is then used to make the plant's dynamic implicitly follow this smooth trajectory by trying to reach the current value of the set point signal in each time step, which presents a smooth jump to the desired value of the system output. This solution easily mitigates the complexions manifested in real-life applications.

b. The noise amplification caused by the derivative term is also countered with a simple solution, that is using a noise-tolerant differential equation when approximating the derivative of any signal. Since ADRC already has robust properties against output noise, it becomes sufficient to implement this approximation in the controller's input signal only.

c. The loss of performance in the control action can be regained using the power of nonlinear control. Instead of relying on the present, accumulative, and predictive error in the form of a linear combination, there may be better Nonlinear alternatives. Since we also want to mitigate the saturation and stability margin reductions brought by the integral term while maintaining its zero steady-state error property, it is only natural that we insert a special Nonlinear function in the equation defining the Nonlinear feedback controller. J.Han proposed $fal(\cdot)$ as a Nonlinear feedback function, but it went through many improvements over time, up until $Hfal(\cdot)$ was presented in [3].

1.2.2 Main concept behind ADRC

The points above are just remedies to the problems with PID control, the core of ADRC is in the concept of total disturbance, its estimation, and rejection. In any control loop, the control signal is inserted to overcome the system's dynamics and impose the desired behavior, thus achieving the control objective. Overcoming the system's dynamics has been done implicitly for a wide range of model-based control techniques. In Active Disturbance Rejection Control, this issue is tackled differently. Consider the control-matched input-affine flat Nonlinear time-continuous (see Definitions 1.2.2, 1.2.2, and 1.2.2) system:

$$y^{(r)} = f(y, \dot{y}, ..., y^{(r-1)}, t) + w(t) + bu(t) \tag{1.2}$$

where r is the relative degree of the output signal $y(t)$, $f(y, \dot{y}, ..., y^{(r-1)}, t)$ is the unknown dynamics of the system, b is the nominal control gain or a fair approximation of it that can be

fixed experimentally, $u(t)$ is the input control signal, and $w(t)$ is the external noise.
A control-matched system is a system where its Nonlinear uncertainties and unknown external disturbances are only in the control channel of its state space representation. Affine means linear, and an input-affine system is a non-autonomous plant where the control input is linearly introduced to the system's dynamics. A flat system is a system that contains at least a

flat output i.e. the states and the control input can be determined from such outputs and a finite number of its derivatives, Appendix A details this concept and its benefits for the ADRC framework.

If we take the state vector as $x = [y \quad \dot{y} \quad \dots \quad y^{(r-1)}]^T$ then (1.2) is written as:

$$\Sigma : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_r = f(y,t) + w(t) + bu \end{cases} \tag{1.3}$$

The ADRC approach considers and treats the system's dynamics as an internal disturbance: its effect is lumped with the effect of all the other forms of disturbances (including external noise, parametric variations, and modeling errors) in what is called total disturbance. This renders ADRC a model-free control framework, granting it robustness against these types of disturbances. The total disturbance is then explicitly estimated and pre-compensated following:

$$u = \frac{-\hat{f}(y,w,t) + u_0}{b} \tag{1.4}$$

where u_0 is the pseudo control signal and $\hat{f}(y,w,t)$ is an estimate of the total disturbance, and the estimation process is detailed in Section 1.6. This compensation produces a very simple and easy-to-control representation of the system. Under the condition that $\hat{f}(y,w,t) \approx f(y,w,t)$, the system is written as:

$$\Sigma : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_r = u_0 \end{cases} \tag{1.5}$$

which is called the canonical form of the system described in (1.2). This series of r integrators back-to-back is in fact a change of basis that is inspired by Differential Geometry. It is also important that some conditions are to be met in order for this form to be valid i.e. for the ADRC approach to be applicable as a control mechanism, see Appendix B for details on these conditions and on the process of obtaining the canonical form.

Since tracking problems are as important as stabilization problems, we will project the previous equations onto this case to achieve the tracking objectives using ADRC. Consider a tracking problem where the plant's output $y(t)$ is ought to follow a reference trajectory $y_r(t)$, the tracking error is then:

$$e = y_r - y \tag{1.6}$$

knowing that $y = x_1$, the canonical form of the tracking error is then:

$$\Sigma : \begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = e_3 \\ \quad \vdots \\ \dot{e}_r = y_r^{(r)} - u_0 \end{cases} \tag{1.7}$$

if $u_0 = y_r^{(r)} + g(e,t)$ where $g(e,t)$ is a given function of the tracking error and its derivatives up to an order that defines the order of the control action, then (1.7) becomes:

$$\Sigma : \begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = e_3 \\ \quad \vdots \\ \dot{e}_r = -g(e,t) \end{cases} \tag{1.8}$$

which represents a non-autonomous r-th order serie of integrators, controlled by $g(e,t)$. See Section 1.4 for more details on the various types of control actions and the characteristics of their performance.

PS: Please note that we considered the general case where the reference signal can be described using an r-th order function, which is less common in practice, especially in higher-order systems. For example, a high-frequency sine reference becomes very large in amplitude when derivated, so generally, only the reference signal is considered.

1.2.3 Case of multi-variable systems

ADRC covers a much wider range of applications than PID even tho it was presented by J.Han for relatively simple order plants. However, this same concept can be applied to much more complex systems, for instance Multi-Input Multi-Output systems.

While working on the decoupled plant is better, one can directly tackle the non-decoupled MIMO system using the ADRC algorithm defined in Section 1.7. However, it is more simple to decouple the plant and treat it like two separate sub-systems, which is done either explicitly or implicitly.

In an explicit manner, [2] considered the coupling terms as internal disturbances and left them for the ADRC algorithm to estimate and pre-compensate them. However, this comes at the cost of tightening the tolerance interval of ADRC's intrinsic robustness property, as it is better to relieve as much pressure from the controller as possible.

As for affine input-control systems described in (1.9), [1] decoupled them implicitly by working on controlling the derivatives of the outputs. Consider the MIMO flat affine continuous-time nonlinear system:

$$\begin{cases} \dot{x} = F(x,t) + Bu \\ y = Cx \end{cases} \tag{1.9}$$

where $B \in \Re^{n \times m}$ is the interconnexion input matrix. The derivative of the output is:

$$\dot{y} = CF(x,t) + Du \tag{1.10}$$

if $D = CB \in \Re^{m \times m}$ is a non-singular matrix with $m = size(y) = size(u)$ then the multi-variable system described in (1.9) is implicitly decoupled as:

$$\dot{y} = G(x,t) + \nu \tag{1.11}$$

where $G(x,t) = CF(x,t)$ is the new total disturbance and $\nu = Du$ is the new control signal.

Now, a SISO ADRC controller can be designed for each channel with $\dot{y}_i$ being the output of the i-th channel and ν_i its control input. Then:

$$u = D^{-1}\nu \tag{1.12}$$

PS: It is to be noted that in practice D is not to be exactly known, as some of its modeling errors can be treated as internal disturbance and thus rejected by means of ADRC.

In the following sections, we will detail the different functioning blocs of an ADRC scheme and how they are implemented together in what is called the ADRC algorithm. These blocs are necessary to build a correct configuration of ADRC because they complement each other in many areas.

1.3 Tracking Differentiator

While it is mostly ignored in control literature, avoiding the complications brought by the setpoint jump is a very important task in real-world applications. Robotic platforms and industrial plants with fast dynamics and reactions are the most concerned about this issue, as the dynamic constraints and the unmodeled dynamics can reflect grave consequences in terms of the performance of the system as well as the safety of the elements in its surrounding environment. Therefore, it is necessary to construct a transient profile that the output of the plant can follow in a smooth manner.

1.3.1 Transient profile generation

A simple way to do so is to consider the transient trajectory as a double integral system whose velocity is to be controlled. Consider the 2nd order system:

$$\begin{cases} \dot{\nu}_1 = \nu_2 \\ \dot{\nu}_2 = u \end{cases} \tag{1.13}$$

where ν_1 is the desired trajectory and ν_2 is its velocity, and $|u| < r$ is the control signal limited by the maximum acceleration of the plant, which is set according to the physical limitations of the system.

Since the objective is to make this double integration system follow a convergent dynamic to the setpoint y_r without overshoot and in the fastest way possible, J.Han used the time optimal control method to achieve this goal, which yields:

$$u = -r sign\Big(\nu_1 - y_r + \frac{\nu_2|\nu_2|}{2r}\Big) \tag{1.14}$$

the transient profile generator is then written as:

$$\begin{cases} \dot{\nu}_1 = \nu_2 \\ \dot{\nu}_2 = -r sign\Big(\nu_1 - y_r + \frac{\nu_2|\nu_2|}{2r}\Big) \end{cases} \tag{1.15}$$

which was proven, in [11], to converge to the set point for a sufficiently large maximum acceleration, according to:
$\forall \varepsilon > 0$ and $T > 0$, $\exists r_0$ such that: if $r > r_0$ then $\int_0^T |\nu_1(t) - y_r| dt < \varepsilon$ Additionally, this continuous-time transient profile generation has poor numerical properties because it could introduce considerable numerical errors in discrete-time implementations. Therefore, a discrete-time version of (1.15) was proposed in [1] as:

$$\begin{cases} \nu_1 = \nu_1 + h\nu_2 \\ \nu_2 = \nu_2 + h\ fhan(\nu_1 - y_r, \nu_2, r_0, h_0) \end{cases} \tag{1.16}$$

where h is the sampling period; r_0 and h_0 are controller parameters that can be set to r and h respectively but can be tuned to adjust the speed and smoothness of the transient trajectory following:

$$\begin{cases} d = h_0 r_0^2 \quad ; \quad a_0 = h_0 \nu_2 \\ y = \nu_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + sign(y)(a_1 - d)/2 \\ s_y = (sign(y + d) - sign(y - d))/2 \\ a = (a_0 + y - a_2)s_y + a_2 \\ s_a = (sign(a + d) - sign(a - d))/2 \end{cases} \tag{1.17}$$

and then:

$$fhan(\nu_1, \nu_2, r_0, h_0) = -r_0\left(\frac{a}{d} - sign(a)\right)s_a - r_0 sign(a) \tag{1.18}$$

1.3.2 Noise-tolerant differentiation

In numerical implementations, a pure differentiation is not feasible, that is why it is very common that a differentiation of a signal $h(t)$ is numerically approximated by:

$$Y(s) = \frac{s}{\tau s + 1} H(s) \tag{1.19}$$

which is in the time domain:

$$y(t) = \frac{h(t) - h(t - \tau)}{\tau} \approx \dot{h}(t) \tag{1.20}$$

meaning that the noise in $h(t)$ gets amplified by a factor of $1/\tau$. Therefore, this approximation is not a good representation of $\dot{h}(t)$. Instead, [1] proposed a more centered approximation:

$$\dot{h}(t) \approx \frac{h(t - \tau_1) - h(t - \tau_2)}{\tau_2 - \tau_1} \tag{1.21}$$

which can be implemented using a 2nd order transfer function of the form:

$$G(s) = \frac{1}{\tau_2 - \tau_1}\left(\frac{1}{\tau_1 s + 1} - \frac{1}{\tau_2 s + 1}\right) \tag{1.22}$$

where $\tau_2 > \tau_1 > 0$ determines the differentiation window.

However, this approximation does not counter the issue of noise amplification well enough to be considered a sufficient solution alone, and it doesn't really solve the issue of discontinuities in the reference signal. That is why PID controllers are most often used without the derivative action.

1.3.3 Putting it together: TD

This struggle motivated the combination of (1.15) and (1.21) in what is called the Tracking Differentiator (TD). If ν_1 of (1.15) can be viewed as a position signal of the trajectory then ν_2 indirectly approximates its generalized differentiation, even when $y_r(t)$ is not differentiable. This idea of using an integration system to calculate the derivative of a signal goes back to the

1920s when fractional differentiation was presented in [12], and it is the core of the TD in an ADRC configuration.

Feedback control breaks the boundary between linear and Nonlinear systems, as it can render a perfectly linear system Nonlinear just by closing the feedback loop, and TD is a good example of that. Instead of focusing on pole placement, one can look outside the box and use Nonlinearity placement even in a double integrator system. The impact TD had is profound in very different areas, as it formed a new line of thought on feedback control design. But the major advantages it brought are:

- As a noise filter, it blocks any part of the signal with an acceleration exceeding r. The physical boundary of signals is often known in terms of its acceleration rate, which can be convenient when implementing this information into TD to reject noises based on the understanding of the physics of the plant. This is not the case in traditional linear filters that can only attenuate noises based on their frequency contents.
- The frequency response of TD is very good as a filter. It has a much smaller phase shift compared to linear filters while maintaining a flat gain within its bandwidth.
- The most important advantage of TD is its ability to calculate the derivative of any signal with a good signal-to-noise ratio (SNR). Since a pure differentiation is not physically implementable, and the commonly used differentiation approximation method amplifies input noise, TD comes as a good approximator of the derivative of any signal, even when it is not differentiable, using N.Wiener's principle of differentiation by integration [12].

Following the big impact of TD, it was initially meant to act as a filter of the output and its derivative. Having as an input the noisy output of the plant and producing a good-quality estimate of it and its derivative, TD was meant to send this information to the feedback loop in the pre-ADRC configuration. Figure 1.2 shows the control topology in question, with TD as a feedback filter, a transient profile generator that can either be TD or even the 2nd order transfer function (1.22), and a Nonlinear PID controller (which is detailed in Section 1.4).

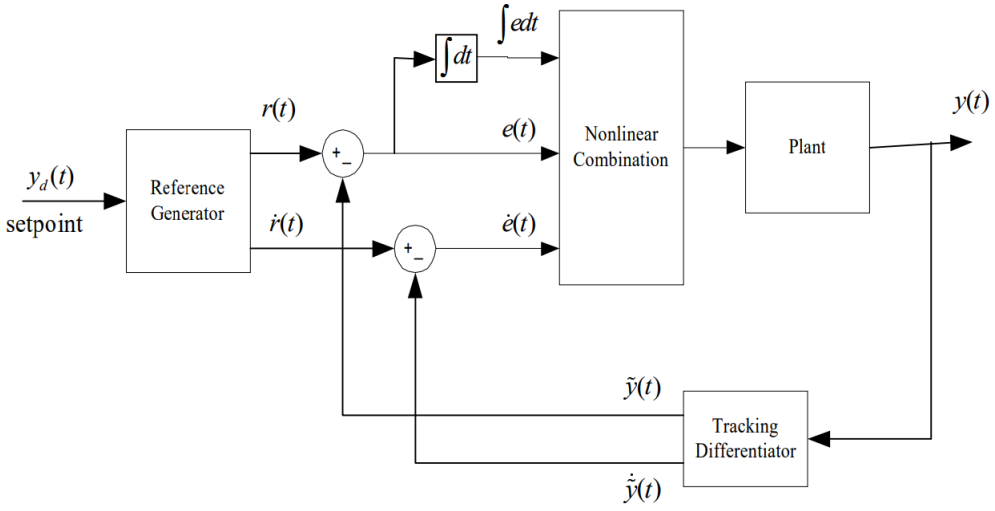


Figure 1.2: pre-ADRC Configuration [2]

In this structure, TD's role is twofold: 1) to filter out the noises in the measurements of $y(t)$ and 2) to determine $\dot{y}(t)$ via integration. But this changed quickly when ADRC's main concept of total disturbance estimation and rejection came to light, and TD is now mainly used to

generate transient trajectories for the setpoint, which provides better control of not only the plant's output but its derivatives as well.

Figure 1.3 illustrates the motion profile generated by TD for different values of r. As can be seen, r has a proportional relation with the convergence speed of the generated profile, which translates directly to the acceleration of the plant. Of course, our choice of this parameter needs to satisfy :

$$r_0 < r < \alpha_\nu \tag{1.23}$$

with r_0 the acceleration threshold defined in Theorem 1.3.1 and α_ν is the plant's maximum acceleration. As can be clearly seen from Figure 1.3, the upper bound of this inequality was not satisfied for $r = 10$ and its lower bound was merely satisfied for $r = 0.1$.

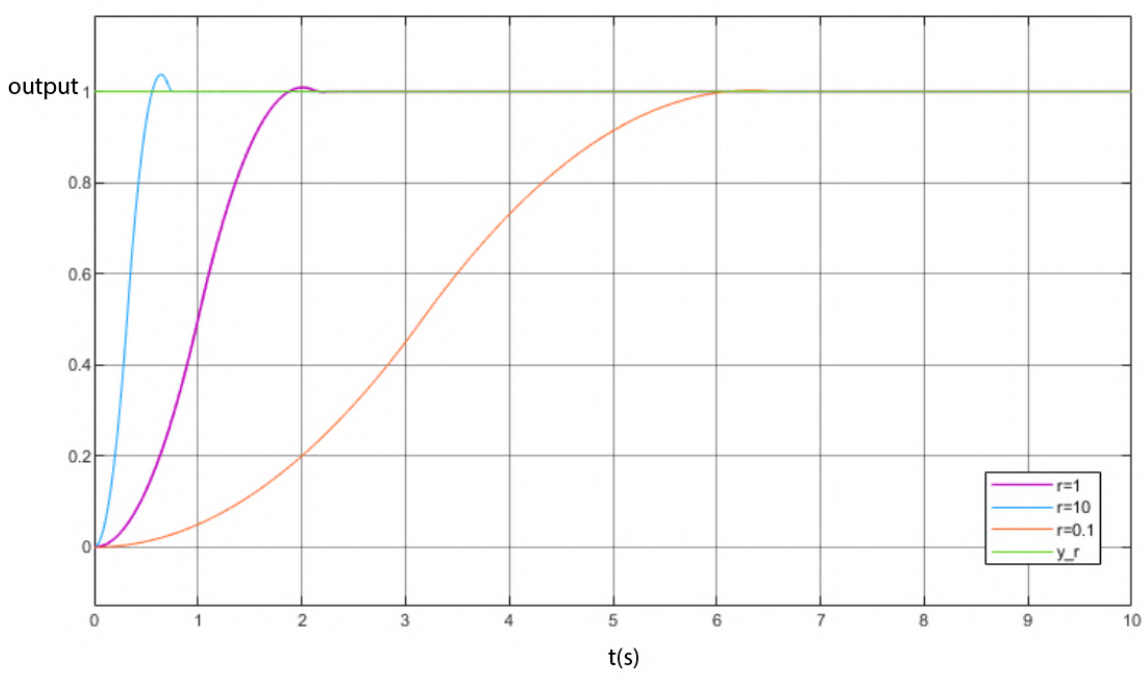


Figure 1.3: Transient Profile Generation using TD for $y_r = 1$ and $r = \{1, 10, 0.1\}$

1.4 State Error Feedback Controller

The loss of performance and robustness due to the oversimplification of the control law is tackled in this section. But first, it is good to keep in mind that since the total disturbances are pre-compensated and any system will be described in its canonical form (1.5), the SFEC is model-free and can be tuned with little information about the system.

Let us kick our study off with linear feedback control. Since feedback control techniques mainly focused on pole placement, linear PID became a very common type of controller. Regardless of its technical issues, it stood as a good solution to many theoretical applications. To make it simple, consider the stabilization problem of a 2nd order SISO system:

$$\begin{cases} \dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \end{cases} \tag{1.24}$$

where linear feedback is implemented:

$$u = -Kx = -K_1 x_1 - K_2 x_2 \tag{1.25}$$

where $K = [K_1 \quad K_2]^T$ is the feedback gain that can be easily solved by a simple pole placement that makes the characteristic polynomial of the closed loop's differential equation (1.26), is Hurwitz.

$$\ddot{x} + K_2\dot{x} + K_1 x = 0 \tag{1.26}$$

A Hurwitz polynomial is a polynomial whose roots (zeros) are located in the left half-plane of the complex plane or on the imaginary axis i.e. the real part of every root is zero or negative. Such a polynomial must have coefficients that are positive real numbers.

In the most common case of a tracking problem, this linear combination is also applied in the form of a PID controller as defined in (1.1). Unfortunately, this combination manifested issues and limited the application of PID control in many industrial applications. This leads us to explore Nonlinear feedback control, which is the main topic of this section.

Consider a single integrator system:

$$\dot{x} = u + w(t) \tag{1.27}$$

where $|w(t)| < 1 \ \forall t$ is a bounded disturbance. Instead of using a linear proportional controller to stabilize the system at the origin, one can use:

$$u = -K_p|x|^{\alpha} sign(x) \tag{1.28}$$

where smaller values of α reduces the steady state error, ranging from $\alpha = 1$ for a linear proportional controller to $\alpha << 1$ for more intense Nonlinearity of the controller.

Figure 1.4 illustrates a comparison between the linear and Nonlinear proportional controller, depending on the values of α, and assuming that there are no disturbances just for clarity. One can observe that smaller values of α not only reduce the static error but also speed up the convergence process of the closed loop. This is because (1.28) provides a higher gain when the error is small and a lower gain when the error is large for $0 < \alpha < 1$. This performance can be achieved using linear controllers but they are much more complicated to implement, which is an added advantage of Nonlinear feedback.

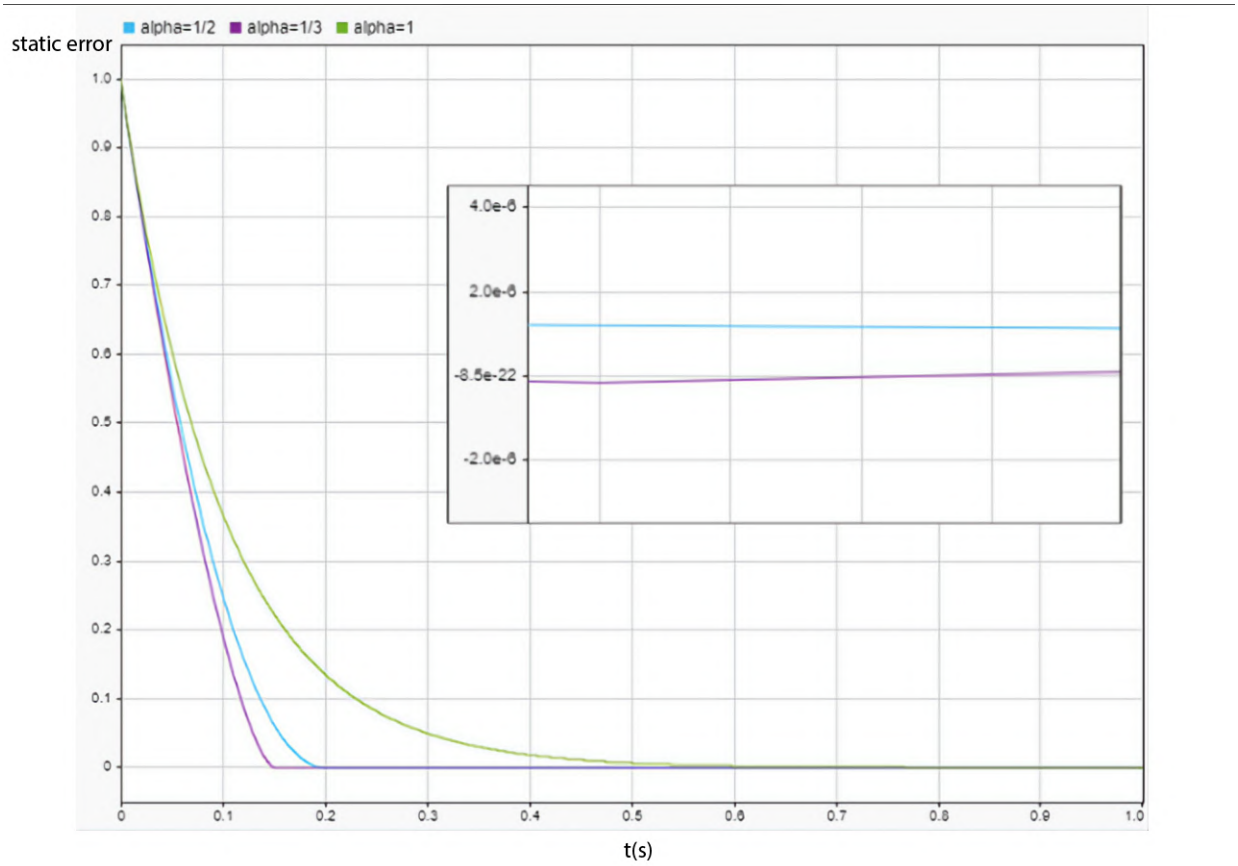


Figure 1.4: Performance of a P-controller for different degrees of Nonlinearity

J.Han proposed a systematic way to design Nonlinear feedback controllers in order to achieve good performance and robustness characteristics without worrying about any gain in the complexity of the control law. Many research efforts have been put into finding better controllers than PID, such as gain scheduling control, fuzzy logic, and optimization control. But the tuning of these controllers is problem-dependent and can be for a specific class of systems in some cases. Because of that, [13] proposed Nonlinear PID as:

$$u = K_P|e|^{\alpha_P} sign(e) + K_I|e_I|^{\alpha_I} sign(e_I) + K_D|e_D|^{\alpha_D} sign(e_D) \tag{1.29}$$

where e_I and e_D are the accumulative and rate of change of the error, respectively.

This fundamental change in the way PID controllers are designed was introduced according to the good performance of (1.28), where the proportional term stays the same. As for the integral term, inserting the integration of the past errors inside $|e_I|^{\alpha_I}$ solves the integral windup problem often encountered in practice. This is due to the fact that the integral action is reduced when the error is large (i.e. in the transient regime) if we set $\alpha_I < 1$, which prevents this term from being saturated. The main role of the derivative term is in the transient period, where $\alpha_D > 1$ speeds up the convergence process at the cost of overshooting, while $\alpha_D < 1$ reduces the overshoot but the convergence will be slower. In our case, we take $\alpha_D > 1$ to attenuate the effect of this term in the steady state (i.e. when e_D is small and, therefore, the gain will be small) because the differentiation signal possesses little information and lots of noise in this time period.

The major drawback of (1.29) is that it causes high-frequency chattering in some simulations due to excessive gains. This calls for a solution that keeps the same properties while dealing with this issue. Section 1.5 presents a solution and an improvement over it, and finishes up with a comparative study of both solutions.

1.5 *fal(·)* and *Hfal(·)* Functions

To enhance the numerical properties of the NL PID, it is better to use the $fal(\cdot)$ function to describe the error, which is defined as:

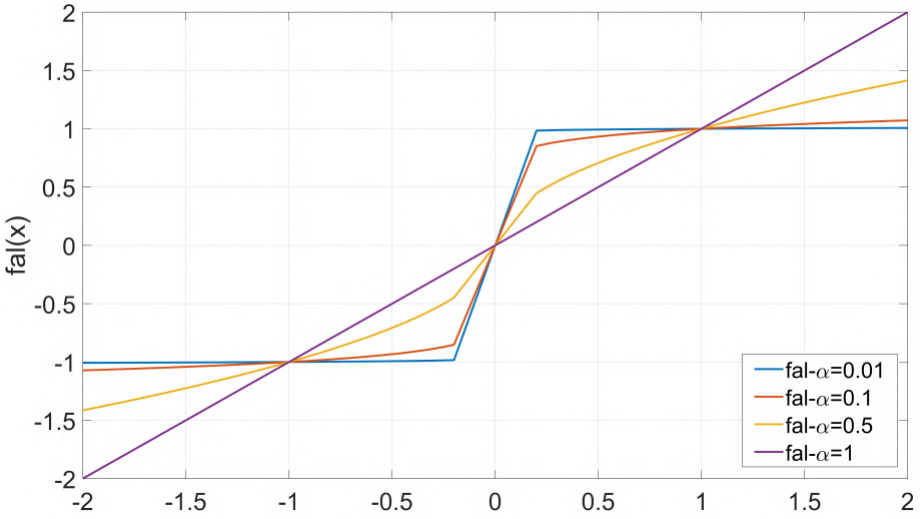
$$fal(e,\alpha,\delta) = \begin{cases} \dfrac{e}{\delta^{1-\alpha}} & ;|e| \leq \delta \\ |e|^{\alpha} sign(e) & ;|e| > \delta \end{cases} \tag{1.30}$$

where smaller values of α reduce the steady state error and speed up the convergence in the case of $\alpha < 1$, while δ denotes a commutation interval. Using this function, finite time convergence of the tracking error is ensured while holding the principle of SELG-LESG:

- LESG: When the error is large, the value of $fal(|e| > \delta, \alpha, \delta)$ ensures a small gain, in the condition that $\alpha << 1$
- SELG: when the error is small, the value of $fal(|e| \leq \delta, \alpha, \delta)$ ensures a large gain, in the conditions that $\delta >> 1$ and $\alpha >> 1$

1.5.1 *fal(·)* function analysis

Figure 1.5 illustrates the effect of parameter variations of the $fal(\cdot)$ function. First of all, and as its expression suggests, this function is divided into a linear and Nonlinear zone. Looking at the variation in the shape of the curve, one can directly observe that α is responsible for the degree of Nonlinearity of this function while δ is responsible for the width of the window between the linear and Nonlinear zones. Additionally, α and δ together are responsible for the slope and extrema of the linear region. Additionally, notice that $fal(e, \alpha = 1, \delta) = e$ i.e. the degree of nonlinearity of the feedback controller can be set to zero just by assuming $\alpha = 1$



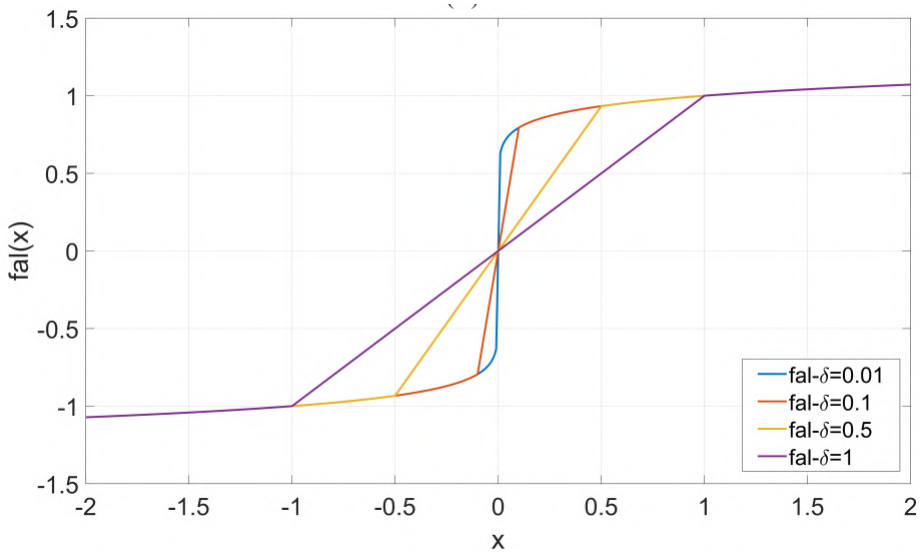


Figure 1.5: The influence of parameters α and δ on the $fal(\cdot)$ function curve [3]

As it can be seen from this figure, the $fal(\cdot)$ function takes the same characteristics of the Nonlinear controller presented in the previous section, while preventing excessive gain when the error is small, using the introduced linear region to attenuate them.

Figure 1.6 illustrates the shape of this function in comparison to a ramp. One can notice that the linear gain is larger than the Nonlinear controller for $|x| > 1$, and smaller for $|x| < 1$. This means that the fal-based Nonlinear controller applies the SELG-LESG control requirement better than its linear peer. applications without issues.

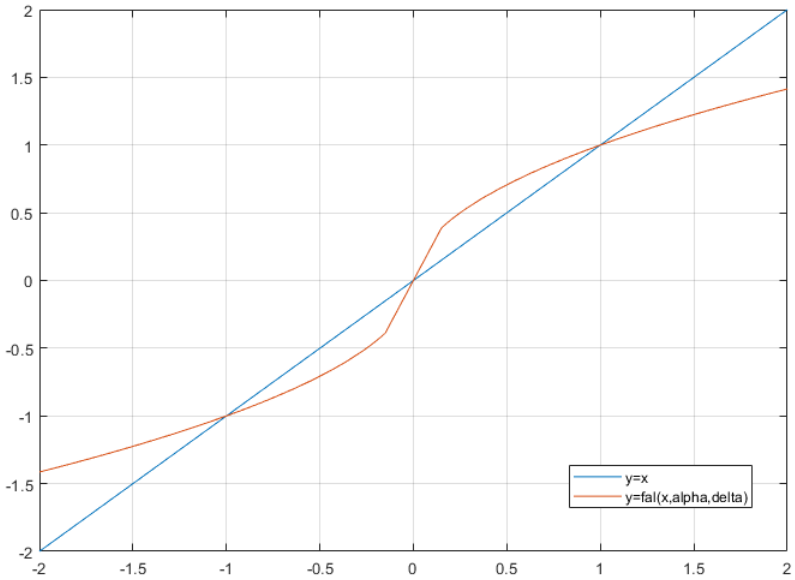


Figure 1.6: Comparison of Linear and fal-based Gains

Now that the $fal(\cdot)$ function is all set up, we can use it to replace the Nonlinearity in the NL PID expression (1.29) as:

$$u = K_P fal(e, \alpha_P, \delta_P) + K_I fal(e_I, \alpha_I, \delta_I) + K_D fal(e_D, \alpha_D, \delta_D) \tag{1.31}$$

where $0 < \alpha_P < 1$, and $0 < \alpha_I < 1$, and $\alpha_D > 1$. Using this specific Nonlinearity placement, PID becomes a much more effective controller and can be applied in numerical applications. As for the window of $fal(\cdot)$ defined by δ, it is generally set the same for all three control actions. What is different is the degree of Nonlinearity, which depends on the type of control action in the case of the controller, and on the order of the estimated derivative in the case of the

extended state observer.

1.5.2 Improving the $fal(\cdot)$ function

Over the years, the $fal(\cdot)$ function has witnessed significant improvements due to some inconvenient it manifested, most importantly:

- In the Nonlinear zone, if α is taken large then the gain will be larger, which not only contradicts the LESG control requirement but also amplifies input noise on the system.
- In the linear zone, if δ and α are taken too large to meet the SELG control requirement, the resulting gain is way too large and will cause the controller vulnerability
- Right in the intersection of both zones, there's no smoothness. This means that switching between both of them at a high frequency makes the system produce the jitter phenomenon, resulting in poor performance in terms of stability and accuracy.

Since this function is the core of the control law defined in (1.31) and other blocs of ADRC (see Sections 1.6 and 1.3), improving it means improving the performance of every bloc of ADRC and therefore achieving the control objective in an overall better strategy.

Solutions to the aforementioned issues have been worked on through a series of $fal(\cdot)$ function improvements. $faln(\cdot)$ was proposed in [14] to improve the disturbance estimation and rejection of the closed loop, but did not solve the jitter problem. $glan(\cdot)$ was constructed in [15] to improve parameter adjustment in the process of disturbance estimation, but the function itself still had inflection points. $zzxfal(\cdot)$ was proposed in [16], which improved the Nonlinear feedback control by overcoming the unsmooth characteristics of $fal(\cdot)$ and reducing the number of jitter in dynamic regulation, but the controller's gains were big when the error was big. [17] proposed $nfal(\cdot)$ to improve the disturbance estimation and rejection process as well as the closed loop feedback controller, satisfying the robustness and rapidity along its way, but this was at a very high complexity cost. As a last improvement, Y.Qin and J.Yang presented $Hfal(\cdot)$ in [3] as:

$$Hfal(e,k,w) = \begin{cases} y & ;dy > 0 \\ max(y)sign(e) & ;dy \leq 0 \\ /y = \dfrac{k^2 e}{\cosh\left(e^2w^2\right)\left(\tanh\left(e^2w^2\right)+1\right)} \end{cases} \tag{1.32}$$

where k shapes the slope of this function at the origin, while k and w together place the extreme values of the characteristic curve of this function. Figure 1.7 illustrates best the role of k and w in shaping this function's curve.

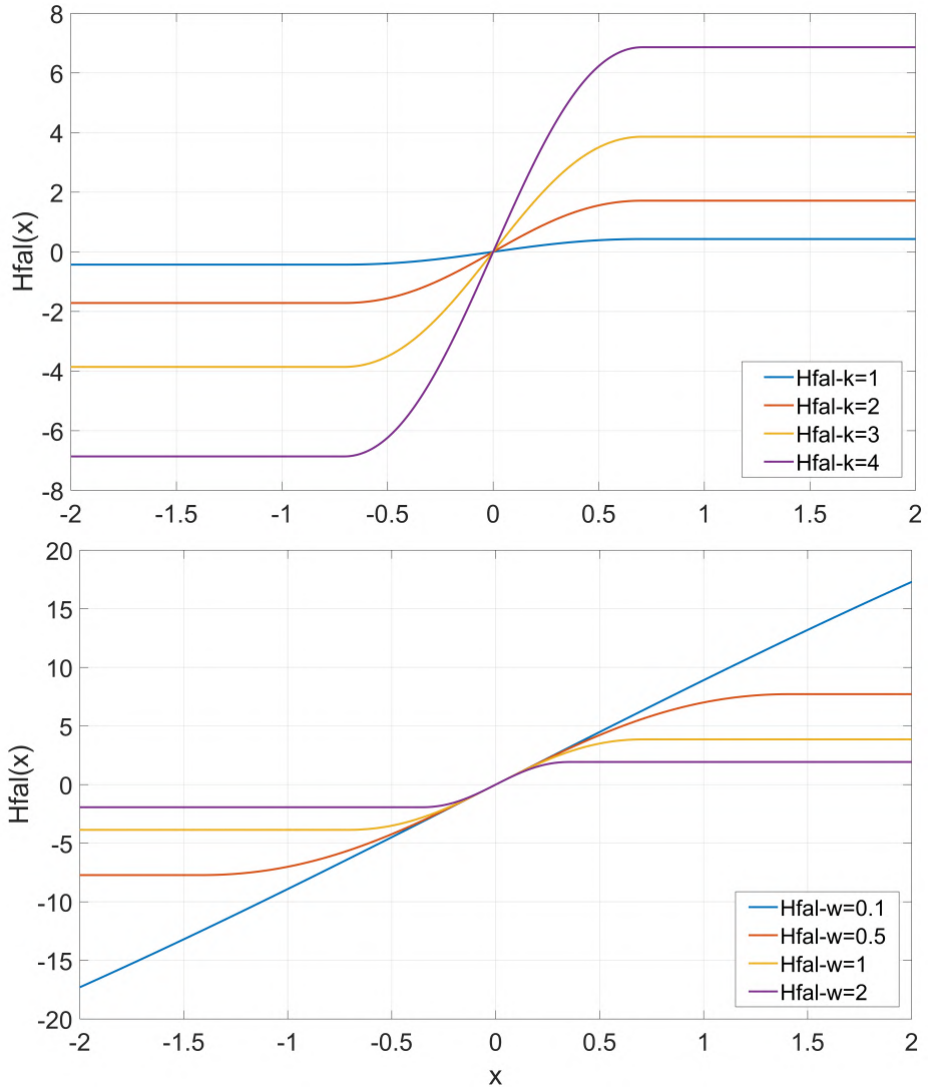


Figure 1.7: The influence of parameters k and w on the $Hfal(\cdot)$ function curve [3]

Y.Qin and J.Yang came up with $Hfal(\cdot)$ as a simpler solution that provided better smoothness, continuity, and convergence than $fal(\cdot)$, thus suppressing the jitter problem. It also applied the SELG-LESG control requirement without the issue of excessive gains, thus avoiding the problem of controller vulnerability and noise amplification.

1.5.3 Comparison between $fal(\cdot)$ and $Hfal(\cdot)$

Now that both functions have been defined and detailed, a visual comparison is needed. To illustrate the improvement of $Hfal(\cdot)$ over $fal(\cdot)$, both functions were used to approximate the derivatives at the origin using the following params:

$fal(x, \alpha, \delta)$	$Hfal(x, k, w)$
$\alpha = 0.025$	$k = 2.2$
$\delta = 0.2$	$w = 2.1$

Table 1.1: $fal(\cdot)$ and $Hfal(\cdot)$ Parameters

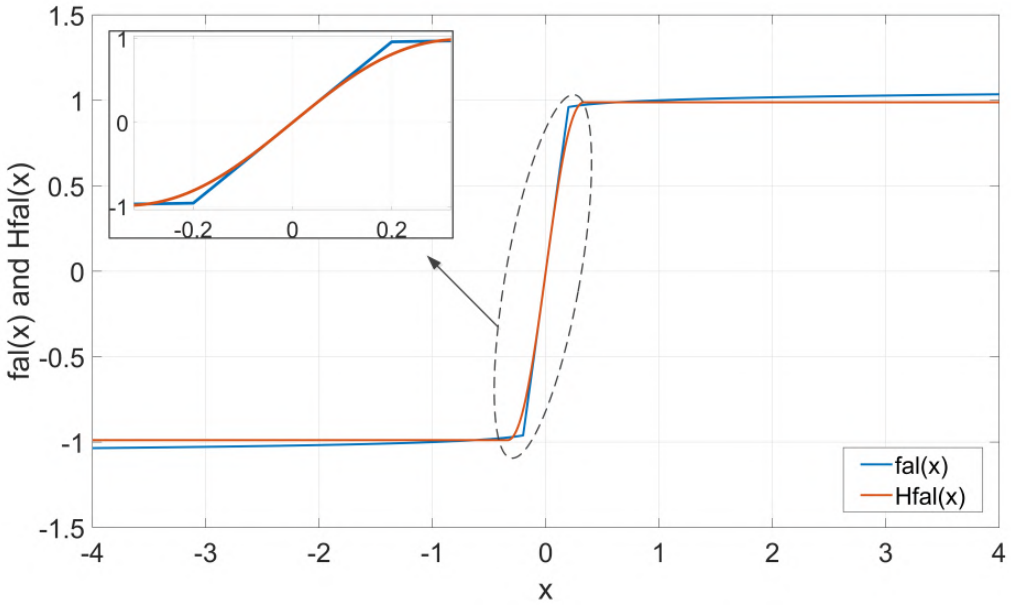


Figure 1.8: Comparison of $fal(\cdot)$ and $Hfal(\cdot)$ characteristics [3]

Figure 1.8 illustrates best that $Hfal(\cdot)$ has still the same fast convergence properties of $fal(\cdot)$ while providing better performance than $fal(\cdot)$ in many aspects:

- $Hfal(\cdot)$ is smooth and continuous in the definition domain, which effectively overcomes the jitter problem caused by the sudden change of slope of $fal(\cdot)$ at the inflection point.
- When the error is small, a smaller Nonlinear structure is used to amplify the error $Hfal(\cdot)$, which meets the SELG requirement but also avoids the vulnerability of the controlled caused by the sudden change of the gain.
- When the error is large, $Hfal(\cdot)$ has a stronger ability to suppress the gain of the system, which reduces the amplification of noise while meeting the LESG requirement.

Therefore, it is only natural to use $Hfal(\cdot)$ to implement solutions to engineering problems in a safer manner. The NL PID is then written as:

$$u = K_P \cdot Hfal(e, k_P, w_P) + K_I \cdot Hfal(e_I, k_I, w_I) + K_D \cdot Hfal(e_D, k_D, w_D) \tag{1.33}$$

In the following section, we will use these functions to design an extended state observer of the system, just like we did with the state error feedback controller just now. Demonstrating that these functions can be implemented to estimate system states and not only to solve stabilization and tracking problems. Then we will talk about the vital role of estimating the total disturbance in an ADRC scheme.

1.6 Extended State Observer

In the classical control theory, the integral action has a very profound impact on the performance of the closed loop, including suppressing the steady state error, speeding up the convergence process, and rejecting disturbances. But, as detailed in Section 1.2, this action brings inevitable lag into the system, which translates into the reduction of the system's stability margins.

The reason why this section is introduced using the integral control action is that Active Disturbance Rejection Control provides an alternative, which lies in Extended State Observer (ESO). Since the integral action is inserted to mitigate the static error, and the static error comes from the system's dynamics, and ADRC estimates and pre-compensates them, there's nothing in the system that is causing the static error (since the system will be described as (1.5)), and an integral action is therefore not needed anymore. We proposed the following Lemma, which will be experimentally verified in Chapter 4:
If there's a good estimate of the total disturbance and if it is fully pre-compensated, then the integral action is not needed anymore and a PD controller should be enough to achieve the control objective in the closed loop.

1.6.1 Generalised ESO

In the pre-ADRC configuration previously illustrated in Figure 1.2, the role of providing a quality estimation of the plant's output and its derivatives –which constitutes the ADRC canonical form's state vector– was assigned to the TD. But with the concept of total disturbance estimation and rejection, a good estimate of the total disturbance is obviously needed. Preferably, an ESO should give a good estimate of the desired signals without relying on the plant's mathematical model. [18] proposed the idea of considering the total disturbance as a differentiable state to be observed along its derivatives. Consider the following r-th order dynamic system:

$$\Sigma : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_r = f(y, w, t) + bu \end{cases} \tag{1.34}$$

if the total disturbance is an m-times differentiable signal $f(y, w, t) \in \mathcal{C}^m$ then the state vector extension $x = [y \quad \dots \quad y^{(r-1)} \quad f \quad \dots \quad f^{(m-1)}]^T$ rewrites (1.34) as:

$$\Sigma : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_r = x_{r+1} + bu \\ \dot{x}_{r+1} = x_{r+2} \\ \quad \vdots \\ \dot{x}_{r+m} = f^{(m)} \end{cases} \tag{1.35}$$

Taking into account the consecutive derivatives of the total disturbance allows the ESO to reconstruct more complex disturbances since it can track $(m-1)$-order polynomial functions. The reason behind approximating the total disturbance using a polynomial is that the ESO becomes robust against it [19], as the characteristic polynomial of the observation closed-loop would be independent of the coefficients of the disturbance's polynomial. Additionally, reconstructing the derivatives of the total disturbance comes at the cost of noise sensitivity and tuning complexity, and only the total disturbance is reconstructed by the ESO in most use cases. The augmented state space representation is then:

$$\Sigma : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_r = x_{r+1} + bu \\ \dot{x}_{r+1} = \dot{f} \end{cases} \tag{1.36}$$

and the corresponding Extended State Observer is presented in [2] as:

$$\Sigma_{ESO} : \begin{cases} \dot{z}_1 = z_2 - g_1(e,t) \\ \dot{z}_2 = z_3 - g_2(e,t) \\ \quad \vdots \\ \dot{z}_r = z_{r+1} + bu - g_r(e,t) \\ \dot{z}_{r+1} = -g_{r+1}(e,t) \end{cases} \tag{1.37}$$

where $z = [z_1 \quad \dots \quad z_{r+1}]^T$ is the reconstructed signals of the plant's output, its derivates, and the total disturbance, $g_i(e,t)$ is the correction function of the i-th estimate, and $e = z_1 - y(t)$ is the estimation error.

Generally speaking, the exact expression of $\dot{f}$ is unknown and not even needed. Therefore, it is discarded in the ESO. Additionally, the observation error correction function depends only on the present value of the observation error.

1.6.2 Estimation correction terms

The type of the observer depends on the correction function $g(e,t)$. Interestingly, if we choose $g_i(e,t) = L_i(z_1 - y) \ \forall i = \{1 : r+1\}$ then the corresponding ESO is the classical Luenberger Observer (LO). Also, if we choose $g_i(e,t) = L_i(e + k_i sign(e)) \ \forall i = \{1 : r+1\}$ then this form would be consistent with that of a variable structure observer.

In an ADRC scheme, the choice of the correction function is most generally the $fal(\cdot)$ function, which gives good results in terms of robustness and fast-tracking. Comparison between LESO and fal-based ESO should be the same as the one in Section 1.5. Using the $fal(\cdot)$ function, the ESO is written as:

$$\Sigma_{ESO} : \begin{cases} \dot{z}_1 = z_2 - L_1 fal(e, \alpha_1, \delta_1) \\ \dot{z}_2 = z_3 - L_2 fal(e, \alpha_2, \delta_2) \\ \quad \vdots \\ \dot{z}_r = z_{r+1} + bu - L_r fal(e, \alpha_r, \delta_r) \\ \dot{z}_{r+1} = -L_{r+1} fal(e, \alpha_{r+1}, \delta_{r+1}) \end{cases} \tag{1.38}$$

What is special about ESO in ADRC is that it does not depend on the model of the plant thanks to the lumping of all dynamics and disturbances into a single ready-to-reconstruct state. This breaks it from the complexations brought by modeling error and parametric variations on the performance and robustness of the observation's closed loop system. See Appendix C for details on the stability of an ADRC closed loop system via the dynamics of the corresponding ESO and SFEC as well.

1.6.3 Observer parameter tuning

The process of observer tuning is the first step when using ADRC as a control framework, because a non-accurate estimation of the plant's output will produce a bad transient profile in the TD, and a bad estimation of the total disturbance invalidates the canonical form of the system.

With a good parameter tuning of the Nonlinear correction term, a single fixed ESO can track the system's signals rather closely for a large class of control problems. In the case of a linear ESO (LESO), its gains can be calculated using a simple pole-placement of its characteristic polynomial:

$$P(s) = s^{r+1} + L_1 s^r + \ldots + L_r s + L_{r+1} \tag{1.39}$$

which is identified with:

$$G(s) = (s + w_o)^{r+1} \tag{1.40}$$

This approach, proposed by Z.Gao in [20], is in fact one of two steps to reduce the number of parameters to tune in an ADRC configuration. By placing all of the LESO poles in the left half plane at the same frequency $-w_o$, the gains $< L_i >_i$ are selected by means of identification so that the LESO characteristical polynomial is Hurwitz. This bandwidth parametrization method was further developed and projected into the High Gain LESO case as:

$$g_i = \frac{a_i}{\varepsilon^i}(\hat{y}(t) - y(t)) \qquad \forall i = \{1 : r+1\} \tag{1.41}$$

where ε is the tuning gain parameter and a_i is the parameter of the i-th observation channel. The characteristic polynomial (1.39) becomes:

$$P(s) = s^{r+1} + \frac{a_1}{\varepsilon} s^r + \ldots + \frac{a_r}{\varepsilon^r} s + \frac{a_{r+1}}{\varepsilon^{r+1}} \tag{1.42}$$

and then the same identification procedure holds. Using this parametrization, the LESO is tuned by setting ε such that it is inversely proportional to all of the estimation errors.

As for the fal-based ESO, a lot of methods can be used to tune its parameters. Most commonly optimization-based methods, empirical methods, or intuitive methods rely on the parameters of the LESO. Intuitive methods are the most useful in the context of an ADRC configuration because it is so easy to do an optimal pole placement on the augmented canonical form thanks to ADRC's concept of total disturbances rejection. Optimization-based and empirical tuning methods may or may not be the simplest design techniques in this context, and the same could be said for SFEC tuning too.

However, in higher-order ADRC, the number of parameters to be tuned grows exponentially. This is why [21] proposed a modified ESO structure, where several low-order state observers with the same parameters were put in series to replace the high-order ESO. The structure of this series of observers is shown in Figure 1.9, where k is the order of each state observer and n is the order of the system. The same tuning process is applied to the State Feedback Error Controller.

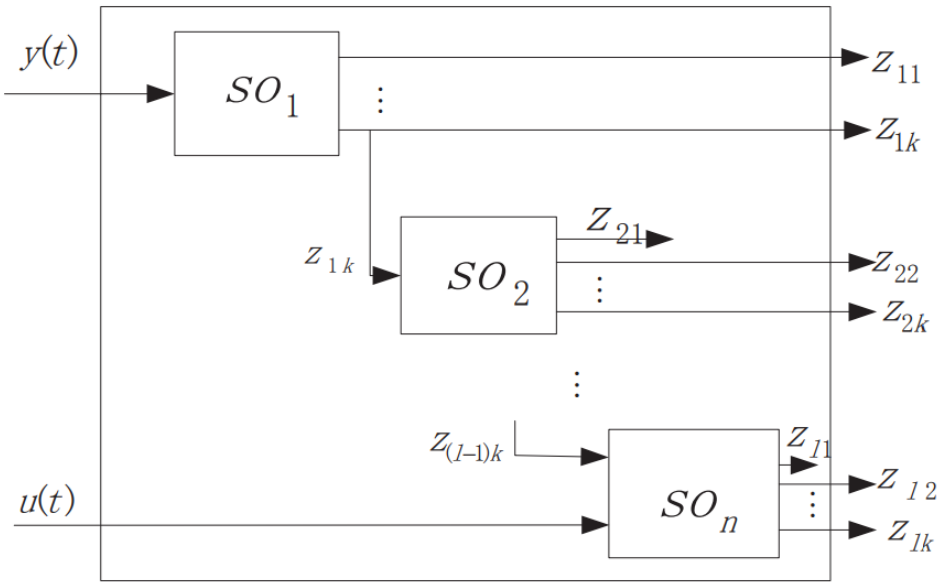


Figure 1.9: Modified ESO Topology

This modified ESO is defined as:

$$
SO: \begin{cases}
SO_1: \begin{cases} \dot{z}_{11} = z_{12} - g_1(z_{11} - y(t)) \\ \dot{z}_{12} = z_{13} - g_2(z_{11} - y(t)) \\ \quad \vdots \\ \dot{z}_{1k} = -g_k(z_{11} - y(t)) \end{cases} \\
SO_2: \begin{cases} \dot{z}_{21} = z_{22} - g_1(z_{21} - z_{1k}) \\ \dot{z}_{22} = z_{23} - g_2(z_{21} - z_{1k}) \\ \quad \vdots \\ \dot{z}_{2k} = -g_k(z_{21} - z_{1k}) \end{cases} \\
\quad \vdots \\
SO_n: \begin{cases} \dot{z}_{n1} = z_{n2} - g_1(z_{n1} - z_{(n-1)k}) \\ \dot{z}_{n2} = z_{n3} - g_2(z_{n1} - z_{(n-1)k}) \\ \quad \vdots \\ \dot{z}_{nk} = -g_k(z_{n1} - z_{(n-1)k}) \end{cases}
\end{cases} \tag{1.43}
$$

Using this observer, whatever the order of the system is, if a k-th order observer is sufficient to give a good estimate of the desired signals then only k estimation correction terms are to be tuned, which is highly advantageous, especially for high order plants.

1.7 The Complete ADRC Algorithm

Combining the Tracking Differentiator, the Extended State Observer, and the State Error Controller in a single structure, ADRC takes the topological form shown in Figure 1.10:

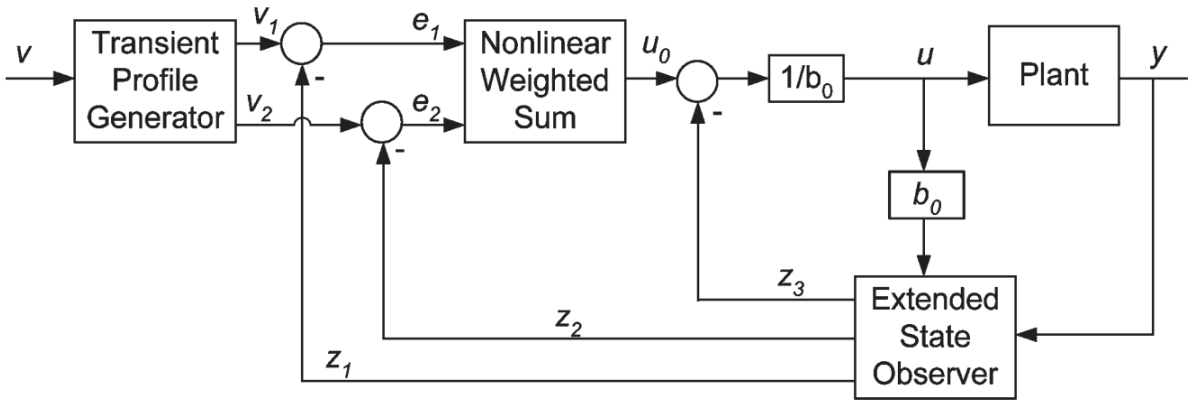


Figure 1.10: ADRC Topology

Since the main concept of ADRC is total disturbance estimation and rejection, which is done thanks to the ESO for the most part, a General PD (GPD) controller is sufficient to control the canonical form of the plant, with possible extensions to more derivatives of the r-th order. As for the observer, LESO is most commonly used since it exhibits fast convergence, good precision, and strong robust properties thanks to the idea of total disturbance system augmentation.

Active Disturbance Rejection Control is a model-free control framework that relies on little information about the system (which is the output signal, the maximum physical acceleration, and a few other variables) and discards the rest along the external disturbances. Even the input gain b can be a rough approximation within a $\pm 50\%$ range. Therefore, as long as the three blocs of ADRC are properly tuned, solid results can be achieved in simulation as in real-life applications. Appendix C details the stability analysis of the ADRC configuration illustrated in Figure 1.10 for both the linear and Nonlinear cases.

Algorithm 1.7 summarizes the complete Active Disturbance Rejection Control algorithm and defines the chronological order in which the three functioning blocs in Figure 1.10 work together to achieve the desired performance and robustness characteristics using this model-free framework. Even though it is advised to use the discrete-time functions, we will be using fal in the Nonlinear feedback controller and the time-continuous version of TD from now on, just for clarity reasons.

[H] **Inputs:** the reference signal $y_r(t)$; the output signal $y(t)$; approximation of the control gain b; maximum acceleration of the plant r.

Outputs: controlled output signal $y(t)$; the feedback error $e(t)$; the observation error $e_o(t)$

1. Internal and external disturbances are lumped into the total disturbance, and the plant is described as a series of integrators, by setting $x = [y \quad \dots \quad y^{(r-1)}]^T$, as:

$$\Sigma_T : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_r = f(y, w, t) + bu \end{cases}$$

2. ESO takes the control input and the plant's output signal and adds the total distur-

bance as a state, then precisely reconstructs the states of the augmented system without depending on the plant's model, following:

$$\Sigma_{ESO}: \begin{cases} \dot{z}_1 = z_2 - L_1 fal(e_o, \alpha_1, \delta_1) \\ \dot{z}_2 = z_3 - L_2 fal(e_o, \alpha_2, \delta_2) \\ \quad \vdots \\ \dot{z}_r = z_{r+1} + bu - L_r fal(e_o, \alpha_r, \delta_r) \\ \dot{z}_{r+1} = -L_{r+1} fal(e_o, \alpha_{r+1}, \delta_{r+1}) \end{cases}$$

3. TD generates a noise-tolerant setpoint transient profile $\nu(t)$ as the reference trajectory with minimum overshoot and controllable speed, following:

$$\begin{cases} \dot{\nu}_1 = \nu_2 \\ \dot{\nu}_2 = -rsign\Big(\nu_1 - y_r + \dfrac{\nu_2|\nu_2|}{2r}\Big) \end{cases}$$

4. ADRC pre-compensates the total disturbance using the reconstructed signal to explicitly overcome the system dynamics and avoid using integral control action, following:

$$u = \frac{-\hat{f}(y, w, t) + u_0}{b}$$

5. fal-based NL PD shapes the good performance of the plant in its canonical form while maintaining good numerical properties, following:

$$u_0 = K_P \; fal(\nu_1 - z_1, \alpha_P, \delta_P) + K_D \; fal(\nu_2 - z_2, \alpha_D, \delta_D)$$

1.8 Conclusion

Active Disturbance Rejection Control came as a new control framework to counter the issues met in industrial applications. PID control was largely used in theory but manifested many drawbacks in real-life applications of control engineering problems, rendering it useless for a large class of plants. J.Han came up with ADRC as a twofold control structure: 1) a combination of the solutions to the problems with PID and 2) a new control design ideology.

In an ADRC configuration, every bloc was designed to fulfill a specific mission in a way that mitigates the associated problem with PID control, while maintaining the best performances possible. Disturbance rejection is an old but key problem for high-performance control design, and a great amount of research effort was put into it. Since the synthesized methods usually assume knowledge of the disturbance and/or the plant's model, modeling errors became a liability to these model-based approaches. ADRC comes as a model-free control technique and assumes a minimum knowledge of the plant. In fact, it was mentioned in [2] that the key issues to be addressed were summarized in two questions: Q1) What do we have to know about the plant in order to control it? Q2) How do we go about finding effective control mechanisms?

The breakthrough brought by the idea of explicitly overcoming the system dynamics alongside the internal and external disturbances had a deep impact. First, it allowed us to avoid the use of integral control action, as the system in itself becomes a simple series of integrators. Second, it allowed for a whole bunch of dynamics to be reconstructed with precision as a single signal to be eliminated later on. Finally, it gave a strong inherent robustness property to the loop.

Aside from total disturbance estimation and rejection, the second line of thought followed in the process of developing ADRC was the Nonlinearity placement. In classical control theory, pole placement was used to impose a specific dynamic in the closed loop of the system. But that quickly became inconvenient as it is a model-dependent approach. Since feedback loop control is the border between linearity and Nonlinearity, one might as well try to place Nonlinearities into the loop. This allowed us to achieve strong performances and enhanced numerical properties without the limitations brought by linear control theory and model-based control.

Chapter 2

Twin Rotor MIMO System

2.1 Introduction

Aeronautic systems are among the most developed and investigated electro-mechanical devices. They reached such a degree of complexity and high cost that it became difficult to develop experimental algorithms and test new controls approach directly on them. The helicopter is one of these devices, it has complex dynamics and is difficult to control even with the modern tools of engineering. So it is only natural that they are given a lot of attention in the engineering community to develop solutions that allow the development of high-end control technique for such transportation machines to be highly performative and robust, the most effective solution being the development of accurate prototypes, it is a safe alternative with a significant cost reduction.

Recent developments in the field of embedded systems and mechatronics led to the emergence of accurate prototypes and test benches for these complex devices. The Twin Rotor MIMO System is one of these prototypes of a helicopter that offers the possibility to test controls before implementing them on the real system.

The first step when developing control algorithms is a good understanding of the system and its particularities. It is then necessary to translate this understanding to a mathematical model that captures a maximum of the intrinsic characteristics and external behavior in a real environment. However, more accurate models introduce nonlinearities and complex equations making the control problem harder to solve, one of the challenges of the control engineer is to deal with this accuracy-complexity trade-off.

In this chapter, we will present the Twin Rotor MIMO System (TRMS) as a mechatronic platform whose mathematical model reconstructs, at lab level, the most important phenomenons that a helicopter exhibits in a real-life situation. Section 2.2 gives a brief image of the flight principle of a real helicopter and the different physical and aeronautical phenomenons that take place during an episode in the air. Section 2.3 provides illustrative diagrams and schemes, and a presentation of the TRMS as an actual twin-rotor multi-variable system, whose hardware and software configuration essentials are provided. The mathematical nonlinear model of the system is obtained, studied and linearized in Section 2.4. This study covers stability, controllability, and observability analysis and constitutes the apple of this chapter.

2.2 Helicopter Flight Principle

Helicopters are flying devices that possess 6 degrees of freedom and can hover or maneuver freely in any direction. Unlike airplanes, the helicopter doesn't fly using engines but with spinning blades, which require having two sets of blades rotating in different directions, moreover, the speed of the helicopter doesn't depend on the rotation speed of the blades but the angle of attack. For this section, we will talk about the most common type of helicopter, the single-main rotor helicopter.

Figure 2.1: Single Main Rotor Helicopter

2.2.1 The main rotor

The three forces that act on a helicopter are :

- The force of gravity.
- The lift of the propellers.
- The drag of the air.

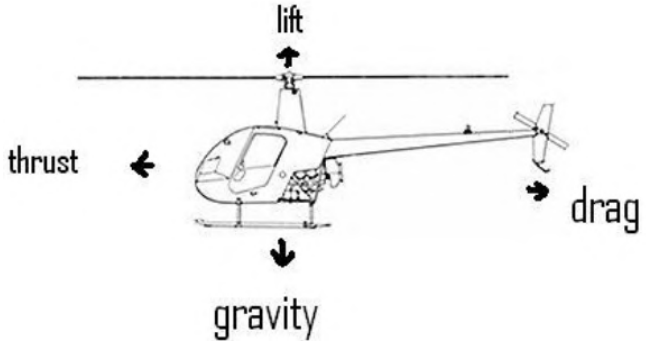


Figure 2.2: Force Diagram of a Helicopter

Using the airfoil principle, the particular shape of the rotor blades generates a lift force, the greater the angle of attack of the air on the blade the greater the lift. Since all the blades rotate at the same speed, a transmission box controlled by the pilot changes the angle of each blade.

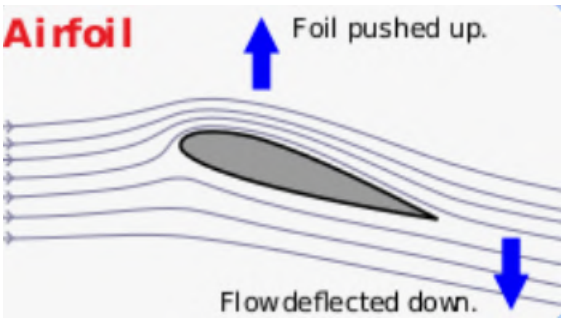


Figure 2.3: Airfoil Principle

A similar angle for all the blades will result in a variation of the total upward lift force; by balancing it with the gravitational force the helicopter can hoover at different altitudes.

Variation of the angle of attack at the front and the rear or on the sides will create different lift forces resulting in a torque. A torque applied in the pitch or roll plan will cause a shift of the position in the roll or pitch angle respectively (not the inverse!). This is explained by the gyroscopic precession principle since the blades are rotating on the horizontal plane, an angular momentum is generated perpendicular to it, with additional torque, another angular momentum will cause the total angular momentum to shift, and the resulting torque acts on the helicopter the way it is described.

2.2.2 The tail rotor

The horizontal torque caused by the spinning of the main rotor causes the helicopter to spin in the opposite direction of the blades, this is called the counter torque (3^{rd} law of Newton), to counter this effect another rotor is needed.

The tail rotor has the same principle as the main rotor; by varying the angle of attack of it blades a lift force is generated to balance the counter torque in the horizontal plane. The pilot can control this rotor to cause the helicopter to rotate in the yaw plane.

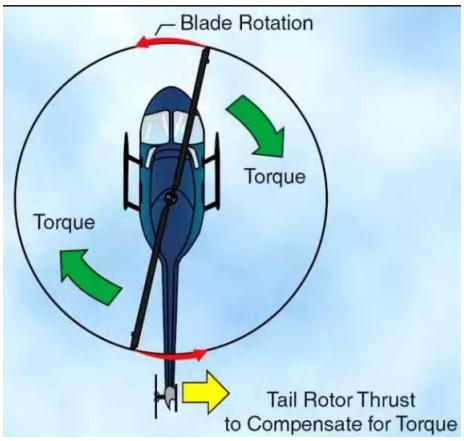


Figure 2.4: Counter Torque Compensation

2.3 TRMS Description and Configuration

2.3.1 System description and schematics

The Twin Rotor MIMO System (TRMS) is an experimental device developed by Feedback Instruments Ltd to conduct laboratory experimentation. It shares many of the characteristics of a helicopter, making it a safe way to test new control algorithms before applying them to a real flying device.

The TRMS serves as a model of a helicopter. However, some significant simplifications are made. First is the fact that TRMS is attached to a tower and second of great importance that the helicopter position and velocity are controlled through the rotor velocity variation. In the real helicopter, the rotor velocity is constant and the propulsion is varied through the rotor blade angle modification. Nevertheless, the most important dynamic characteristics present in a helicopter are captured in the TRMS model. Like in a real helicopter, there is a significant cross-coupling between the two rotors. If we activate the vertical position rotor the helicopter will also turn in the horizontal plane [22].

It consists of :

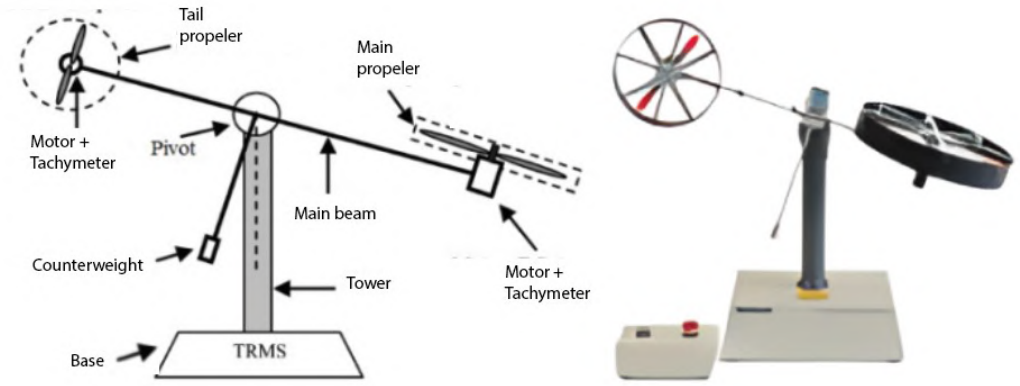


Figure 2.5: Scheme of TRMS' Components

- **The base :** To maintain the whole structure, it also contains all the necessary circuits and electronic components of the device.
- **The tower :** To maintain the beam.
- **The beam :** It can move in the horizontal and/or the vertical plane.
- **The propellers :** A horizontal and a vertical one are placed at the two ends of the beam, they are constituted of a helix, a rotor, and a shield.
- **A secondary beam with a counterweight :** Fixed in the middle of the main beam to reduce the vibrations of the vertical movements.
- **A power box :** To turn on/off the device.

The TRMS can move freely in the horizontal plane between -2.82 rad and +2.82 rad and in the vertical plane between -1.05 rad and 1.22 rad. The motion can be restrained to one plane with screws.

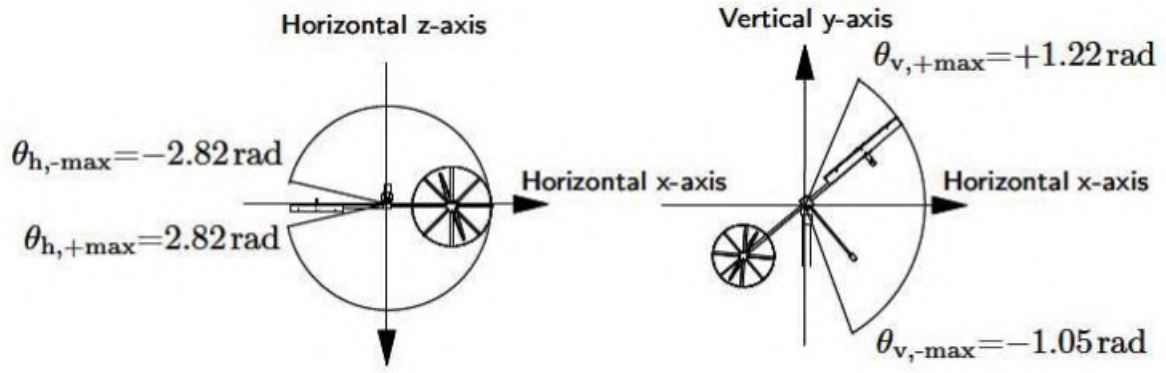


Figure 2.6: Range of Motion of the TRMS

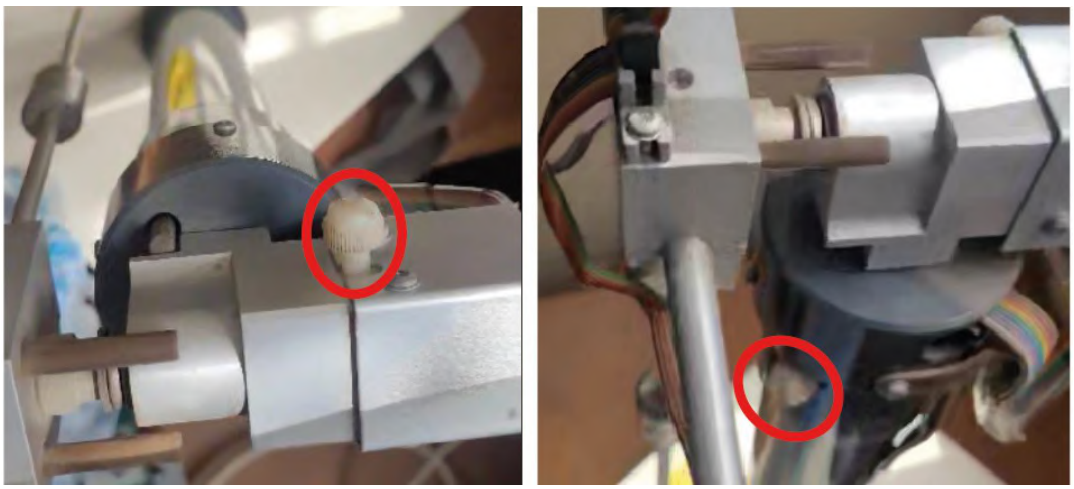


Figure 2.7: Vertical and Horizontal Screws

As for the In and Out signals, the Twin Rotor MIMO System has:

- **Two inputs :** the vertical and horizontal rotors.

  The two DC motors have the advantage of being linear, the angular speed is proportional to the voltage input. The voltage is limited to $\pm 2.5V$ and is delivered through a chopper which is controlled with PWM pulses generated from the acquisition card.

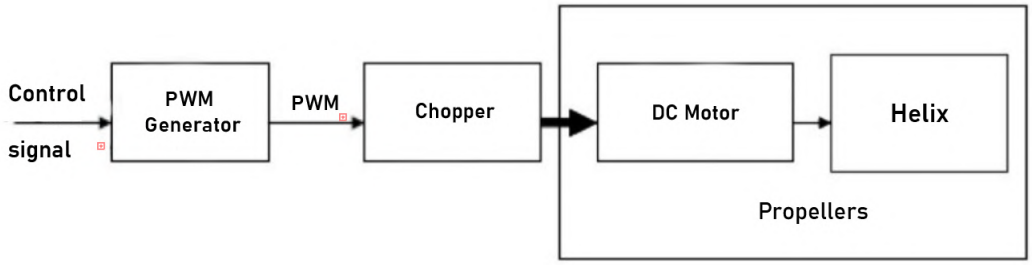


Figure 2.8: Control of the Motors

- **Two outputs :** The vertical and horizontal angle of the beam.

  The vertical and horizontal angles are measured with optic encoders placed inside the articulation of the beam. The angles are set to $0\,rad$ at the horizontal position of the system.

2.3.2 System wiring and configuration

Only two cards are needed to make a communication portal between the TRMS and a computer: an adaptation card called the SCSI BOX, and an acquisition card called the ADVANTECH PCI 1711.

The TRMS base has all the hardware needed to filter, adapt, and manage the I/O signals of the system, it contains the following elements:

- **On/Off connector :** To turn on or off the system.
- **Voltage switch selector :** To select the power voltage (110V/220V).
- **General power supply :** To connect the power supply.
- **Digital input plug :** To connect the remote On/Off switch.
- **PL1 connector** To connect with the PL1 of the adaptation card, it transfers the signal of the sensors. It is the wide ribbon 40-way cables at the bottom right.
- **CN2 connector** To connect with the PL2 of the adaptation card, it transfers the signal of the motors. It is the topmost right ribbon 20-way cable.
- **CN3 connector** To connect with the PL3 of the adaptation card, it transfers the signal of the tachometers. It is the second ribbon 20-way cable.

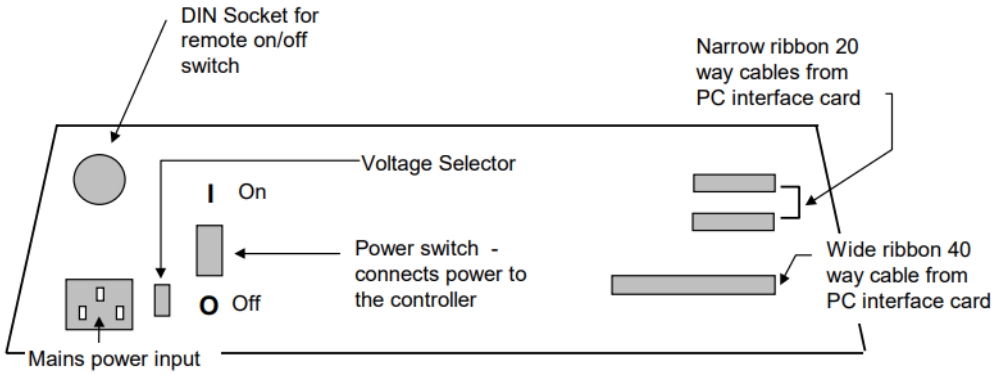


Figure 2.9: The TRMS' Base Rear Panel

The SCSI BOX card adapts the signals of the computer and the TRMS, it has 4 pins :

- **PL1 connector** to transfer the signal of the sensors.
- **PL2 connector** to transfer the signal of the motors.
- **PL3 connector** to transfer the signal of the tachometers.
- **68 pin connector** to connect the card with the ADVANTECH PCI 1711.

The ADVANTECH PCI 1711 is a universal acquisition card that is installed on the PCI port of the PC, it allows the adaptation of the digital/analog signals between the hardware of an embedded system and a control computer, in our case we use :

- **2 Digital Outputs** for the control signals of the motors.
- **2 Digital inputs** for the acquisition of the sensor's signals.
- **2 Analog inputs** for the tachometer's signals.

Figures 2.10 and 2.11 show the SCSI BOX and ADVANTECH PCI cards respectively on a real TRMS lab platform.

Now the different functioning units of the TRMS have been presented, let's move on to putting them together. Setting up the connection protocol for our system is a 3-steps process:

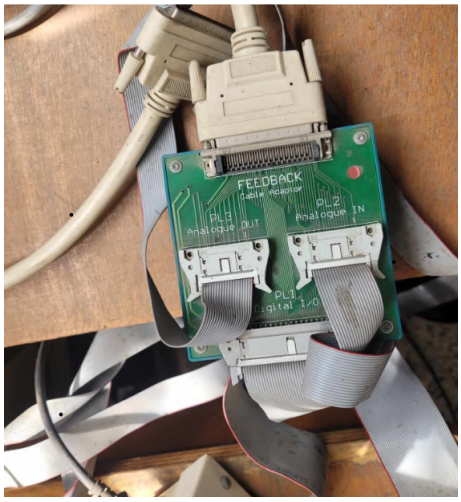


Figure 2.10: The SCSI BOX

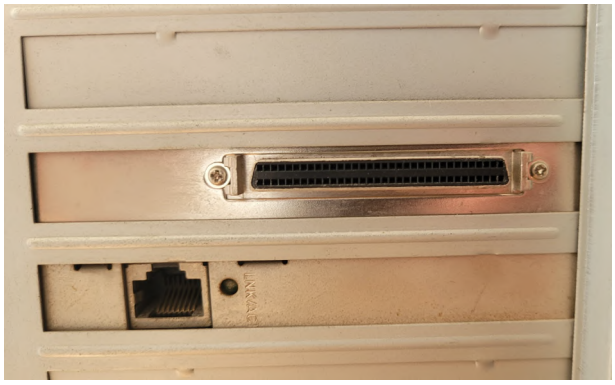


Figure 2.11: The ADVANTECH PCI 1711

1. Connecting the TRMS base's connectors to the adaptation card.
2. Connecting the adaptation card to the acquisition card.
3. Connecting the acquisition card to the computer.

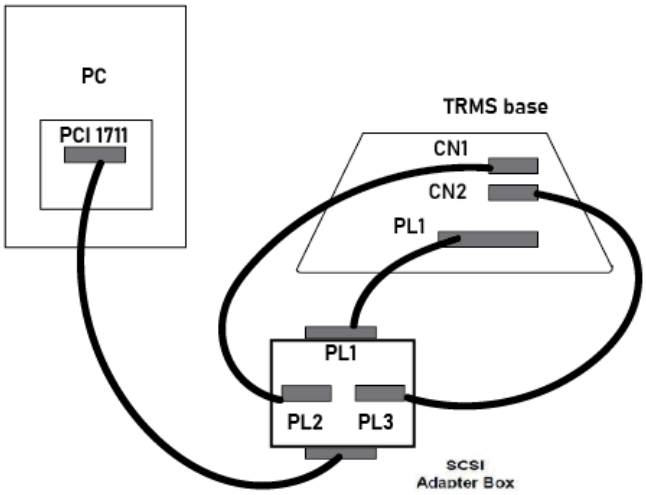


Figure 2.12: TRMS Connection Setup

2.3.3 Development environment

The TRMS uses the HIL (Hardware-In-the-Loop) simulation, this approach allows the integration of the hardware components (Actuators and Sensors) through emulation in the simulation in a computer alongside the plant of the system (the mathematical model). With this technique, the real implementation of the control algorithms is easier and shares the same program with the simulation.

After the tests are conducted in Simulink/Matlab, the TRMS is connected to the same computer and the program can be transferred to the system's hardware with the use of a Simulink library. In particular, we can plot the real outputs and input signals on the screen of the computer, improving the test duration and reducing its complexity.

The needed software programs are:

- **Matlab :** It is used as the development environment where all the programs run. The version used was matlab 6 running on Windows xp.
- **Simulink :** It is a visual interface to develop the control algorithms and plots the results that use a bloc diagram logic, the software translates the program to code lines to be executed by Matlab.
- **Real Time Workshop :** It translates the Simulink code to a C++ code to be executed by the TRMS software.
- **VISUAL C++ 6.0 PRO :** To compile the C++ code.
- **Real Time Windows Target - RTWT :** To execute in real time the Matlab code and transfer it from Windows to the TRMS.

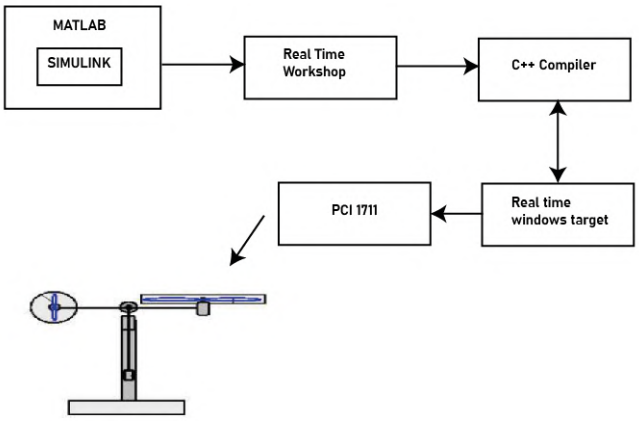


Figure 2.13: TRMS Code Execution Sequence

- **PCI1711 Toolbox :** The TRMS can be controlled using Matlab/Simulink using this toolbox, it contains blocs to communicate with the acquisition card, Simulink models, and DLL files for control and modeling. It can be used with other simulink blocs and once the program is compiled and uploaded we can use it to plot the input and output signals.

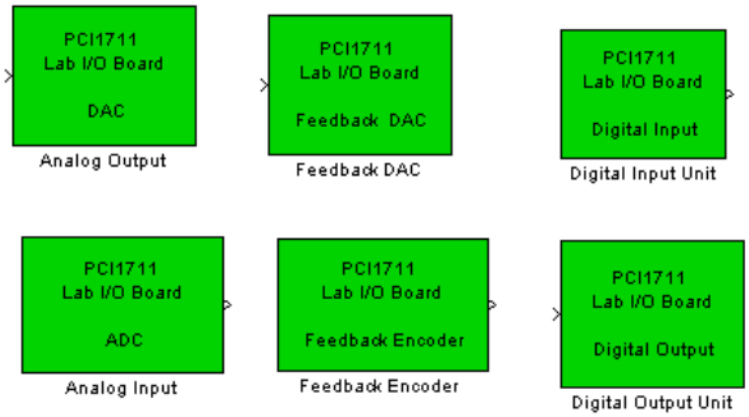


Figure 2.14: TRMS Toolbox Library

2.4 TRMS Modeling

The TRMS is an underactuated high-order Nonlinear MIMO system with input saturation. Many modeling projects were conducted on the TRMS, in [23, 24] the model was obtained using the second law of newton, in [25, 26] using lagrange equations, in [27] a linear model in the frequency domain was obtained with system identification, in [28, 29] more complex system were derived when even wire friction was taken into consideration. We will use the system in [30] inspired from the manufacturer manual [22] as it was implemented succesfully in [31]

Thanks to the properties of the ADRC approach, we only need a fair representation of the system that pictures the main characteristics with no further precision needed.

2.4.1 Nonlinear model

We consider the system as a cascade of two subsystems: the mechanical subsystem, which we split into the two planes of motion, and the electrical subsystem.

2.4.1.1 Electrical part

The relation between the input voltage u and the rotation speed of the motor τ is approximated by a first-order differential equation.

$$\tau_1 = \frac{k_1}{T_{11}s + T_{10}}u_1 \tag{2.1}$$

$$\tau_2 = \frac{k_2}{T_{21}s + T_{20}}u_2 \tag{2.2}$$

2.4.1.2 Mechanical Vertical plan

The forces that act on the TRMS beam in the vertical plan are :

- The force of gravity.
- The lift force of the vertical propeller.
- The drag or friction from the air.
- The friction of the wires.

The torques that act on the TRMS in the vertical plan are :

- The counter torque of the horizontal propeller.
- The gyroscopic torque generated by the horizontal propeller.
- The counterbalance and counterweight torque.

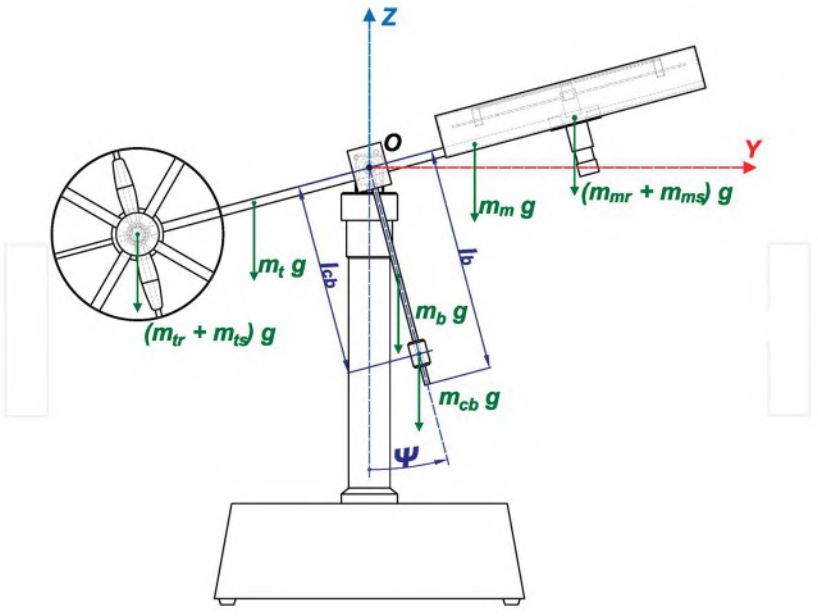


Figure 2.15: Vertical View of the TRMS

To describe the motion in the vertical plane, the manufacturer neglected some of the phenomena: the friction of the wires and the counter torque, these torques are negligible in the vertical plane compared to the gravity, the drag, and the lift forces. Using the second law of Newton we find :

$$I_1\ddot{\psi} = \sum M_\psi \tag{2.3}$$

$$I_1\ddot{\psi} = M_{p\psi} - M_g - M_{f\psi} - M_{gy} \tag{2.4}$$

where :

- I_1 : Moment of inertia of the TRMS in the vertical plane.
- ψ : The vertical angle .
- $M_{p\psi}$: The torque generated by the lift force of the vertical propellers $M_{p\psi} = a_1\tau_1^2 + b_1\tau_1$
- M_g : The torque generated by the gravity force $M_g = M_{gr}\sin\psi$
- $M_{f\psi}$: The torque generated by the friction forces $M_{f\psi} = B_{1\psi}\dot{\psi} - B_{2\psi}\,\dot{\varphi}^2\sin(2\,\psi)$
- Mgy : The gyroscopic torque $M_{gy} = K_{gy}\,M_p\,\dot{\varphi}\cos\psi$

Remark

The model used in the vertical plane is slightly different from the model of the manufacturer, the second term of the friction torque has been changed.

2.4.1.3 Mechanical Horizontal plan

The forces that act on the TRMS beam in the horizontal plane are :

- The lift force of the horizontal propeller.

- The drag or friction from the air.
- The friction of the cables.

The only torque that acts on the TRMS in the horizontal plan is the counter torque of the vertical propeller.

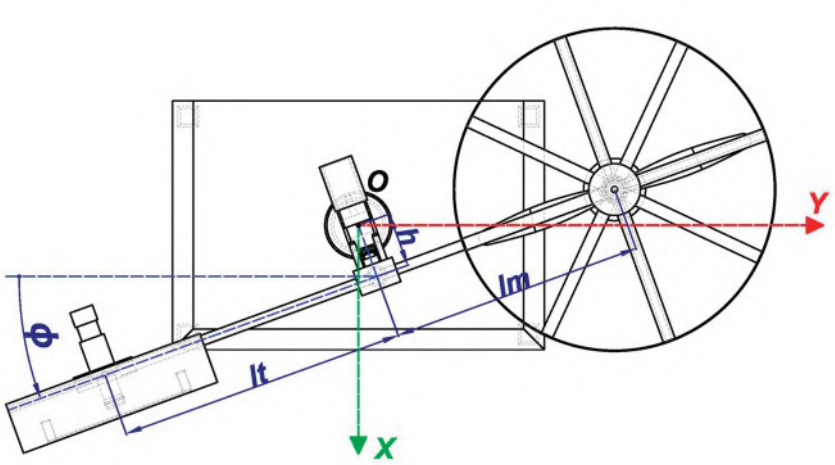


Figure 2.16: Horizontal View of the TRMS

Likewise, using the second law of Newton we find :

$$I_2\ddot{\phi} = \sum M_{\varphi} \tag{2.5}$$

$$I_2\ddot{\varphi} = M_{p\varphi} - M_{f\varphi} - M_c \tag{2.6}$$

where :

- I_2 : Moment of inertia of the TRMS in the horizontal plane.
- φ : The horizontal angle.
- $M_{p\varphi}$: The torque generated by the lift force of the horizontal propellers $M_{p\varphi} = a_2\tau_2^2 + b_2\tau_2$
- $M_{f\varphi}$: The torque generated by the friction forces $M_{f\varphi} = B_{1\varphi}\dot{\varphi}$
- M_c : The counter torque of the vertical propeller $M_c = \frac{K_c(T_0 s+1)}{T_p s+1} M_{p\psi}$

Remark

The model used in the horizontal plan is slightly different from the model of the manufacturer, the friction torque has a typo error in its first term that we corrected and we neglected the second term, the coupling torque is a function of the main rotor lift torque and not the main rotor speed.

2.4.1.4 Model parameters

According to the manual [22] provided by Feedback, the manufacturing company behind TRMS, the parameters' values are given in the following table:

Parameter	Value
$\mathbf{I}_1$ - moment of inertia of vertical rotor	$6.8 \cdot 10^{-2}$ kg $\cdot$ m^2
$\mathbf{I}_2$ - moment of inertia of horizontal rotor	$2 \cdot 10^{-2}$ kg $\cdot$ m^2
a_1 - static characteristic parameter	0.0135
$\mathbf{b}_1$ - static characteristic parameter	0.0924
$\mathbf{a}_2$ - static characteristic parameter	0.02
$\mathbf{b}_2$ - static characteristic parameter	0.09
$\mathbf{M_{gr}}$ - gravity constant	0.32 N $\cdot$ m
$\mathbf{B}_{1\psi}$ - friction function parameter	$6 \cdot 10^{-3}$ N $\cdot$ m $\cdot$ s/rad
$\mathbf{B}_{2\psi}$ - friction function parameter	$1 \cdot 10^{-3}$ N $\cdot$ m $\cdot$ s^2/rad
$\mathbf{B}_{1\varphi}$ - friction function parameter	$1 \cdot 10^{-1}$ N $\cdot$ m $\cdot$ s/rad
$\mathbf{B}_{2\varphi}$ - friction function parameter	$1 \cdot 10^{-2}$ N $\cdot$ m $\cdot$ s^2/rad
$\mathbf{K}_{gy}$ - gyroscopic parameter	0.05 s/rad

Table 2.1: Model Parameters of the Mechanical Part

$\mathbf{k}_1$ - motor 1 gain	1.1
$\mathbf{k}_2$ - motor 2 gain	0.8
$\mathbf{T}_{11}$ - motor 1 denominator parameter	1.1
$\mathbf{T}_{10}$ - motor 1 denominator parameter	1
$\mathbf{T}_{21}$ - motor 2 denominator parameter	1
T_{20} - motor 2 denominator parameter	1
T_p - cross-reaction parameter	2
T_0 - cross-reaction parameter	3.5
$\mathbf{k}_c$ - cross-reaction gain	-0.2

Table 2.2: Model Parameters of the Electric Part

2.4.1.5 State space representation

From the differential equations, the state space vector, control inputs, and outputs are:

- The state vector : $x = [\psi\ \dot{\psi}\ \varphi\ \dot{\varphi}\ \tau_1\ \tau_2\ M_c]^t$
- The inputs $u = [u_1\ u_2]^t$
- The outputs $y = [\psi\ \varphi]^t$

and the resulting state space representation is:

$$\Sigma_{NL}: \begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = \dfrac{a_1 x_5^2 + b_1 x_5 - M_{gr} \sin x_1 - B_{1\psi} x_2 + B_{2\psi} x_4^2 \sin(2x_1) - K_{gy}(a_1 x_5^2 + b_1 x_5) x_4 \cos x_1}{I_1} \\ \dot{x_3} = x_4 \\ \dot{x_4} = \dfrac{a_2 x_6^2 + b_2 x_6 - B_{1\varphi} x_4 - x_7}{I_2} \\ \dot{x_5} = \dfrac{k_1 u_1 - T_{10} x_5}{T_{11}} \\ \dot{x_6} = \dfrac{k_2 u_2 - T_{20} x_6}{T_{21}} \\ \dot{x_7} = \dfrac{K_c T_0 (2a_1 x_5 \dot{x_5} + b_1 \dot{x_5}) + K_c (a_1 x_5^2 + b_1 x_5) - x_7}{T_p} \end{cases} \tag{2.7}$$

2.4.2 Linear model

To study the TRMS model both the linear and nonlinear model were considered. This linearisation was done on the equilibrium point at the origin for all the state variables and control inputs.

Remark

Primarily, the origin of all states is not an equilibrium point. At rest, $\psi \approx 45°$, however, in practice, the TRMS is initiated and kept at the horizontal position by hand until the motors start spinning.

The state-space equations around $x_e = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ and $u_e = [0\ 0]$ are :

$$\Sigma_L: \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{2.8}$$

where :

- A is the **system matrix**, defining the dynamics of the state vector $x(t)$. It is the partial derivative of $\dot{x}(t)$ with respect to $x(t)$ evaluated at the equilibrium point (x_e, u_e):

$$A = \left.\frac{\partial \dot{x}(t)}{\partial x(t)}\right|_{x=x_e,\, u=u_e}$$

- B is the **input matrix**, defining how the input vector $u(t)$ influences the state vector $x(t)$. It is the partial derivative of $\dot{x}(t)$ with respect to $u(t)$ evaluated at the equilibrium point (x_e, u_e):

$$B = \left.\frac{\partial \dot{x}(t)}{\partial u(t)}\right|_{x=x_e,\, u=u_e}$$

- C is the **output matrix**, mapping the state vector $x(t)$ to the output vector $y(t)$. It is the partial derivative of $y(t)$ with respect to $x(t)$ evaluated at the equilibrium point (x_e, u_e):

$$C = \left.\frac{\partial y(t)}{\partial x(t)}\right|_{x=x_e,\, u=u_e}$$

The linearized model's state space matrices are therefore:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4.7 & -0.088 & 0 & 0 & 1.358 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5 & 0 & 4.62 & -50 \\ 0 & 0 & 0 & 0 & -0.909 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0.02 & 0 & -0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0.8 \\ -0.032 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.4.3 Study of the linear model

To understand our system, a theoretical study, and open loop simulations were conducted, all the conclusions of this section are guaranteed to be true only around the origin.

2.4.3.1 Stability

[Lyapunov Stability] The system is **Lyapunov stable** if for every $\epsilon > 0$, there exists a $\delta > 0$ such that

$$\|x(0)\| < \delta \implies \|x(t)\| < \epsilon \quad \text{for all}\, t \geq 0.$$

This is equivalent to saying that all the eigenvalues λ_i of the matrix A have non-positive real parts and those with zero real parts have a Jordan block of size one.

$$\mathrm{Re}(\lambda_i) \leq 0, \quad \forall\, i, \quad \text{and if} \quad \mathrm{Re}(\lambda_i) = 0, \text{ then the Jordan block corresponding to } \lambda_i \text{ is 1.}$$

[Asymptotic Stability] The system is **asymptotically stable** if it is Lyapunov stable and

$$\lim_{t\to\infty} x(t) = 0.$$

This is equivalent to saying that all the eigenvalues λ_i of the matrix A must have negative real parts :

$$\mathrm{Re}(\lambda_i) < 0, \quad \forall\, i.$$

[Exponential Stability] The system is **exponentially stable** if there exist constants $M > 0$ and $\alpha > 0$ such that

$$\|x(t)\| \leq M\|x(0)\|e^{-\alpha t}, \quad \text{for all}\, t \geq 0.$$

This is equivalent to saying that all the eigenvalues λ_i of the matrix A have strictly negative real parts :

$$\mathrm{Re}(\lambda_i) < 0, \quad \forall\, i.$$

[Bounded-Input Bounded-Output (BIBO) Stability] The system is **BIBO stable** if for every bounded input $u(t)$ satisfying

$$|u(t)| \leq M_u \quad \text{for all } t \geq 0,$$

where M_u is a finite positive constant, the output $y(t)$ remains bounded, that is,

$$|y(t)| \leq M_y \quad \text{for all } t \geq 0,$$

where M_y is a finite positive constant.

This is equivalent to saying that all the poles of the transfer function $H(s) = C(sI - A)^{-1}B$ have negative real parts.

Using the `eig()` function on Matlab for our system, we find:

$$\lambda = [-0.0440 + 2.1675i\ , -0.0440 - 2.1675i, \quad 0\ , -5\ , -0.5\ , -0.909\ , -1]$$

which are interpreted as:

- Having all eigenvalues negative and one null means the system is asymptotically stable. That means that the system won't become unstable if shifted from its equilibrium point and released freely. However, it is not BIBO stable, for a non-zero input signal the system will become unstable.
- Having the first 2 poles with an imaginary part suggests that the vertical system has an oscillatory behavior.

To confirm the Lyapunov stability, we first simulate the open loop response with no control inputs for different initial conditions:

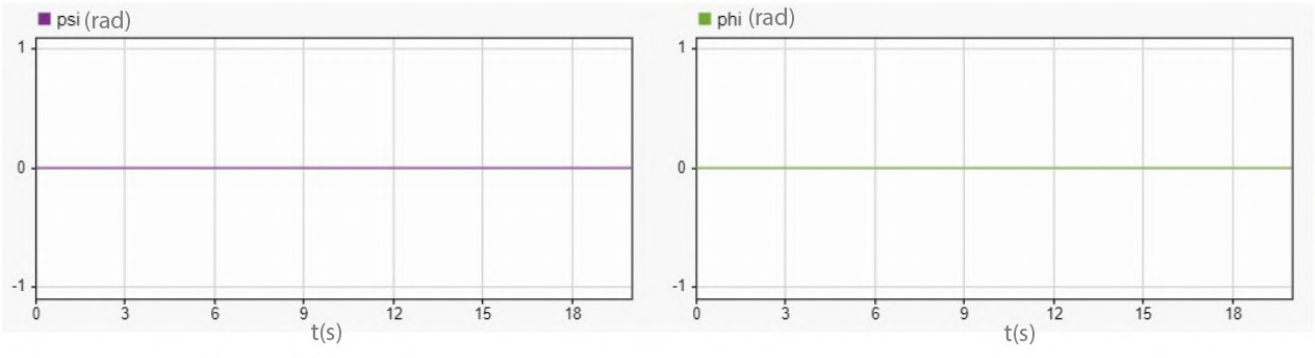


Figure 2.17: $y(t = 0) = (0\ \ 0)^T$ and u=$(0\ 0)^T$

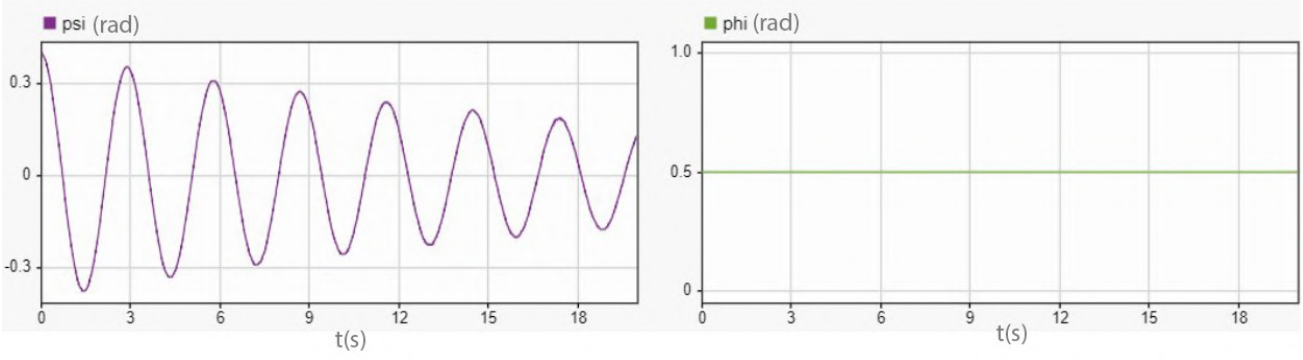


Figure 2.18: $y(t = 0)$=(0.5 0.5) and u=$(0\ 0)^T$

Remarks

- The origin is a stable equilibrium point.
- The vertical system has an oscillatory behavior but always comes back to the origin (asymptotic stability).
- For the horizontal system, all the points are equilibrium points, this is explained by the absence of a static torque acting on the horizontal plane (absence of gravity).

To confirm the BIBO stability, we simulate the open loop response to a step signal of different magnitudes on both the pitch and yaw sub-systems with initial conditions at the origin:

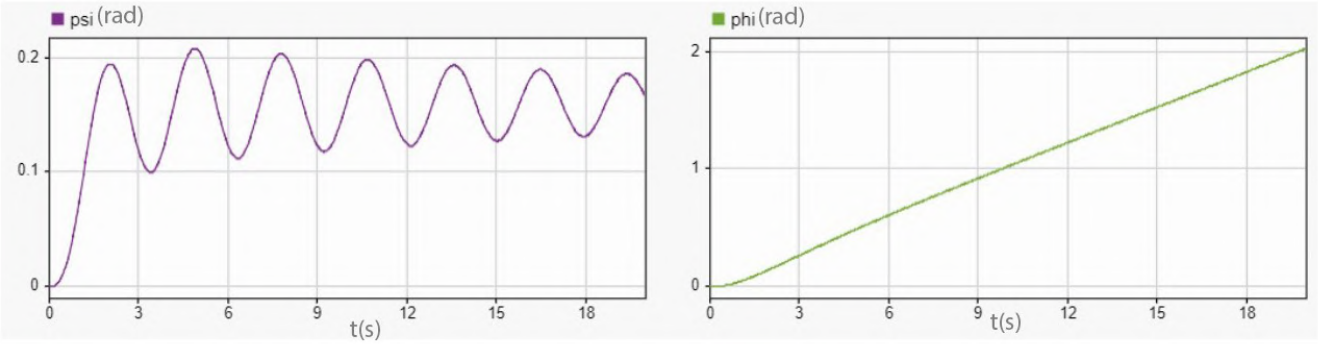


Figure 2.19: $u = (0.5 \;\; 0)^T$

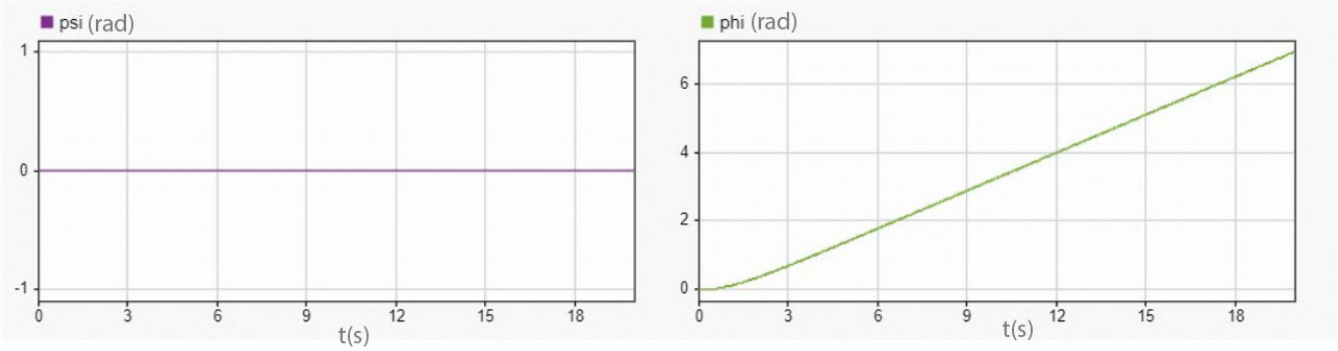


Figure 2.20: $u = (0 \;\; 0.5)^T$

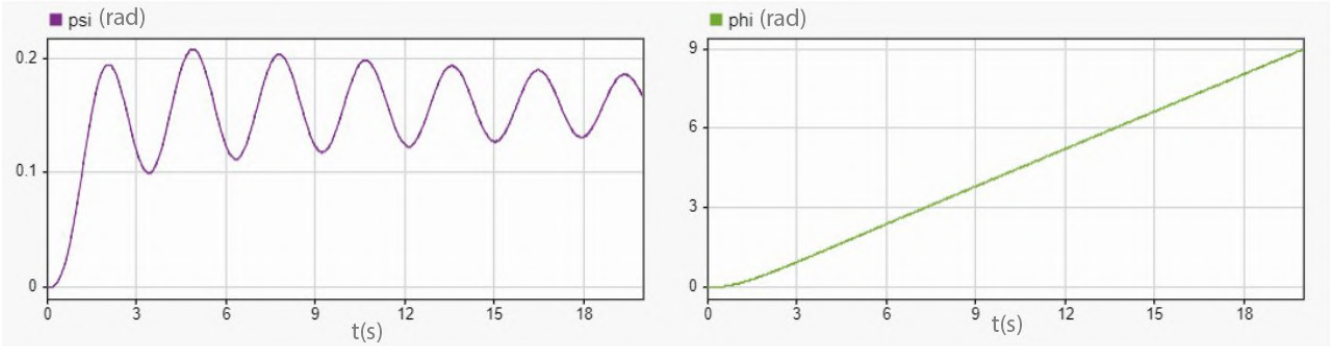


Figure 2.21: $u = (0.5 \;\; 0.5)^T$

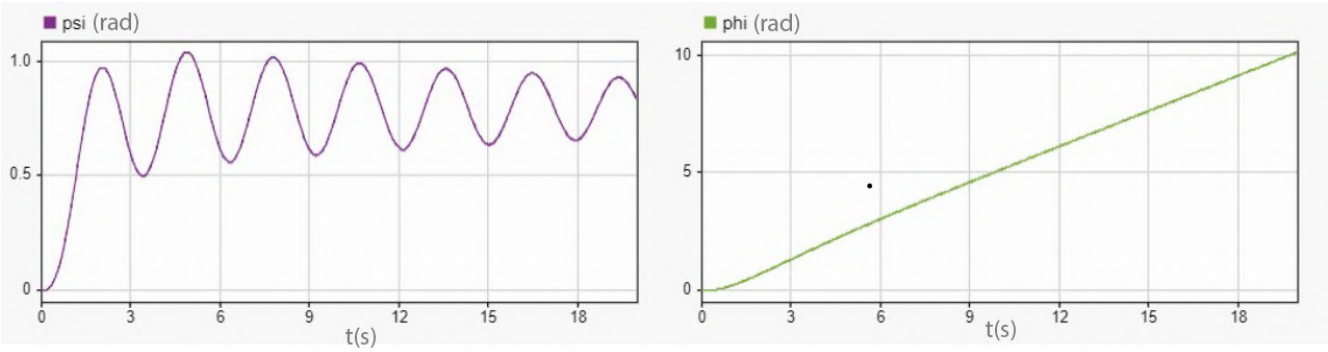


Figure 2.22: $u = (2.5 \;\; 0)^T$

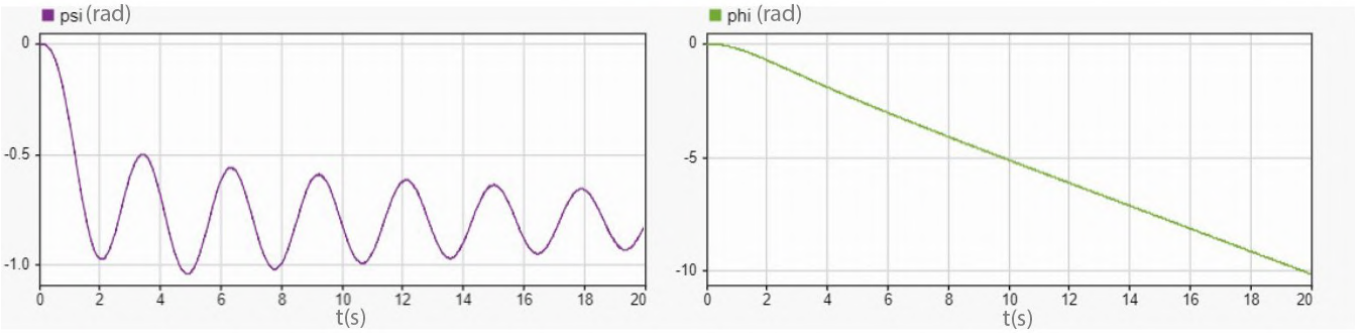


Figure 2.23: $u = (-2.5 \;\; 0)^T$

and the open loop's impulse response too:

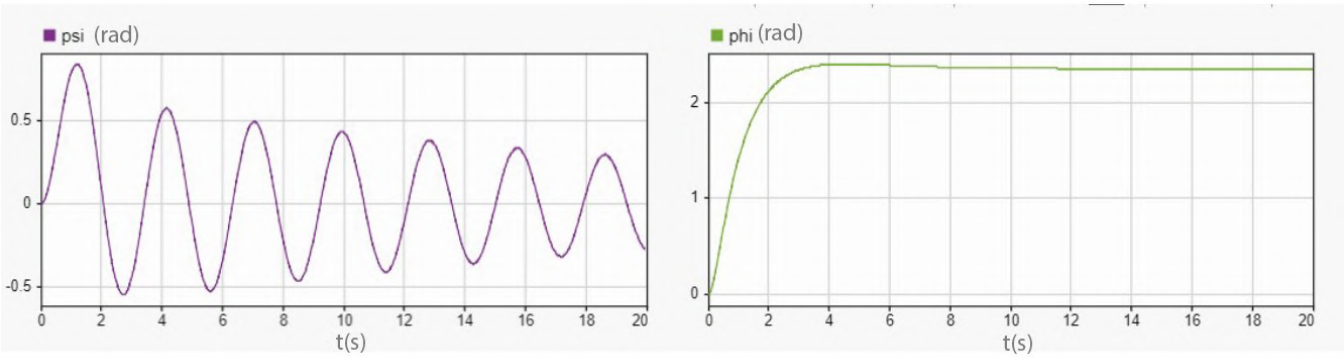


Figure 2.24: open loop reponse for an impulse input in the vertical and horizontal rotor

Remarks

- The horizontal system acts like an integrator, it is explained by the absence of dynamic torque in the horizontal plane to counteract the motion, the only force being the friction with the air that is not sufficient.
- There is a strong coupling effect that acts mainly on the horizontal system even when $u_2 = 0$, when both control signals are applied the coupling effect is amplified with no visible effect on the vertical plane (due to the damping force of gravity and the counterweight).
- The vertical system is BIBO stable, when the maximum input signal is applied the output is still bounded.
- For negative values of the control signal the same conclusion can be made.

- The Same conclusion is made for the impulse response, when an impulse is given to the system the TRMS moves to a new horizontal angle while the vertical position comes back to the equilibrium point with oscillations.

Conclusion

The linear model is Lyapunov stable, BIBO unstable, oscillatory in the vertical plane and with a strong coupling in the horizontal plane.

2.4.3.2 Controllability

A system is said to be controllable if it is possible to drive the state $x(t)$ from any initial state $x(0)$ to any final state $x(T)$ in finite time T, using an appropriate control input $u(t)$.

To determine if the system is controllable, we construct the **controllability matrix** $\mathcal{C}$:

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}$$

The system is **controllable** if and only if the controllability matrix $\mathcal{C}$ has full rank, meaning:

$$\text{Rank}(\mathcal{C}) = n$$

Using Matlab, we find that The controllability matrix has a rank of $7 = n$ so our system is controllable.

Having the linear system controllable at the origin means that the nonlinear system is also controllable for a neighborhood of the origin.

2.4.3.3 Observability

A system is said to be observable if it is possible to determine the state $x(t)$ from the output $y(t)$ over a finite time interval, using appropriate measurements.

To determine if the system is observable, we construct the **observability matrix** $\mathcal{O}$:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

The system is **observable** if and only if the observability matrix $\mathcal{O}$ has full rank, meaning:

$$\text{Rank}(\mathcal{O}) = n$$

This ensures that all components of the state $x(t)$ can be uniquely determined from the output $y(t)$ using suitable measurements. This condition is necessary in the ADRC approach where observers are a major part of it.

Using Matlab, we find that The observability matrix has a rank of $7 = n$ so our system is observable.

Having the linear system observable at the origin means that the nonlinear system is also observable for a neighborhood of the origin.

2.4.4 Study of the nonlinear model

In the following our system (2.7) is defined as

$$\Sigma : \begin{cases} \dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2 \\ y = \begin{bmatrix} h1_(x) \\ h_2(x) \end{bmatrix} \end{cases} \tag{2.9}$$

where f and g_1 and g_2 are vector fields of the appropriate dimensions, $h_1(x) = x_1$, and $h_2(x) = x_3$
The Lie derivative of a scalar function $h(x)$ with respect to a vector field $f(x)$, denoted by $L_f h$, represents the rate of change of h along the flow of f. It is defined as:

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$$

The Lie bracket $[f, g]$ of two vector fields f and g is defined as:

$$[f, g](x) = \frac{\partial g(x)}{\partial x} f(x) - \frac{\partial f(x)}{\partial x} g(x)$$

2.4.4.1 Stability

The equilibrium point $\bar{x}$ of the system is considered *Lyapunov stable* if there exists a continuously differentiable function $V(x)$ such that:

1. $V(\bar{x}) = 0$.
2. $V(x) > 0$ for all $x \neq \bar{x}$ (positive definite).
3. $\dot{V}(x) \leq 0$ for all x in a neighborhood around $\bar{x}$.

Stability Conditions:

- If $\dot{V}(x) < 0$ for all x in a neighborhood excluding $\bar{x}$, $\bar{x}$ is *asymptotically stable*.
- If $\dot{V}(x) \leq 0$ for all x in a neighborhood, $\bar{x}$ is *Lyapunov stable* (but not necessarily asymptotically stable).

However our state space model (2.7) is too complicated to find a suitable Lyapunov function, we will proceed by analyzing the result of the open loop simulations. We start by an open loop study without control signal for different input signals

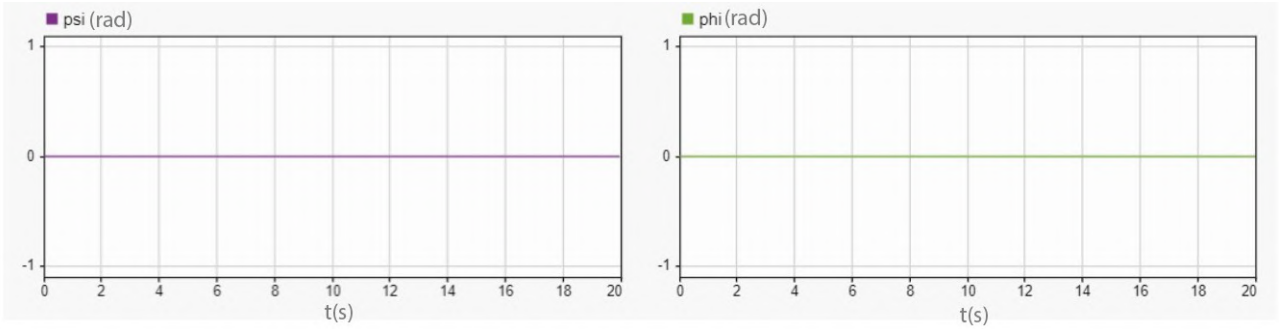


Figure 2.25: $y(t=0) = (0\ \ 0)^T$ and $u = (0\ \ 0)^T$ non linear

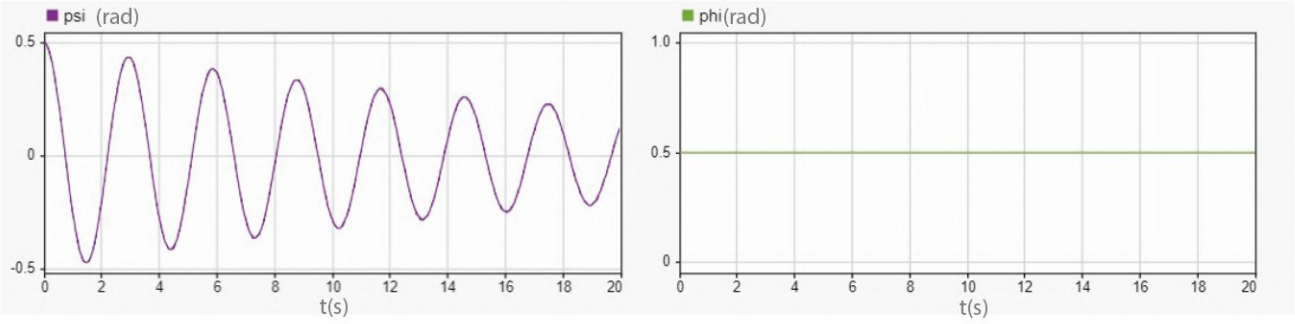


Figure 2.26: $y(t=0) = (0.5\ \ 0.5)^T$ and $u = (0\ \ 0)^T$ non linear

Remark

Here we have the same results as the linear model, for initial conditions that are further from the equilibrium point we still find the same remarks.

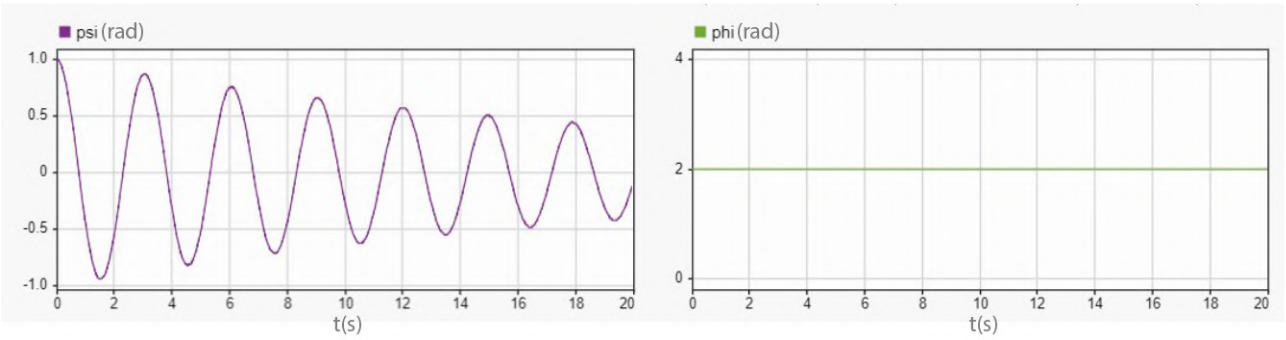


Figure 2.27: $y(t=0) = (1\ \ 2)^T$ and $u = (0\ \ 0)^T$ non linear

We will now test the stability in the presence of input signals, we simulate the open loop response to a step signal of different magnitudes on both the pitch and yaw sub-systems with initial conditions at the origin :

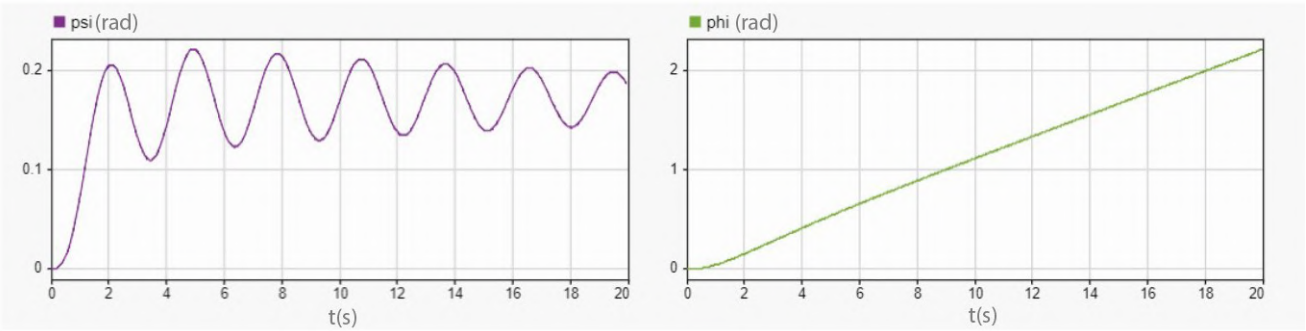


Figure 2.28: $y(t=0) = (0\ \ 0)^T$ and $u = (0.5\ \ 0)^T$ non linear

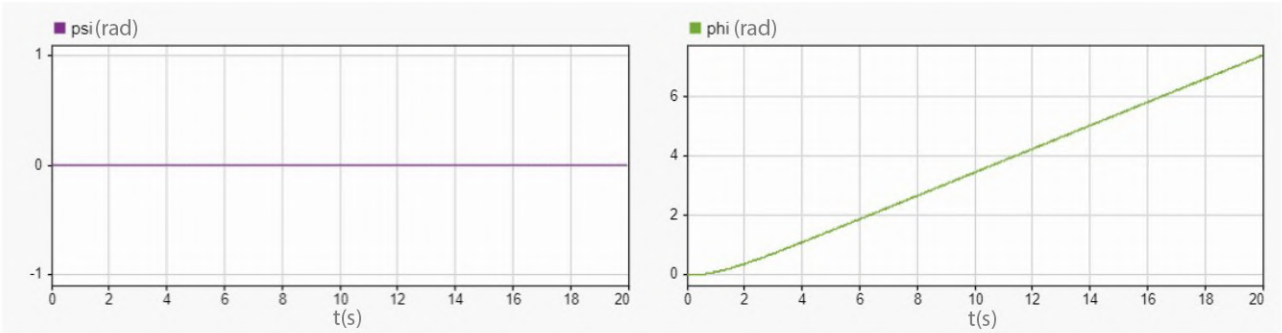


Figure 2.29: $y(t=0) = (0\ \ 0)^T$ and $u = (0\ \ 0.5)^T$ non linear

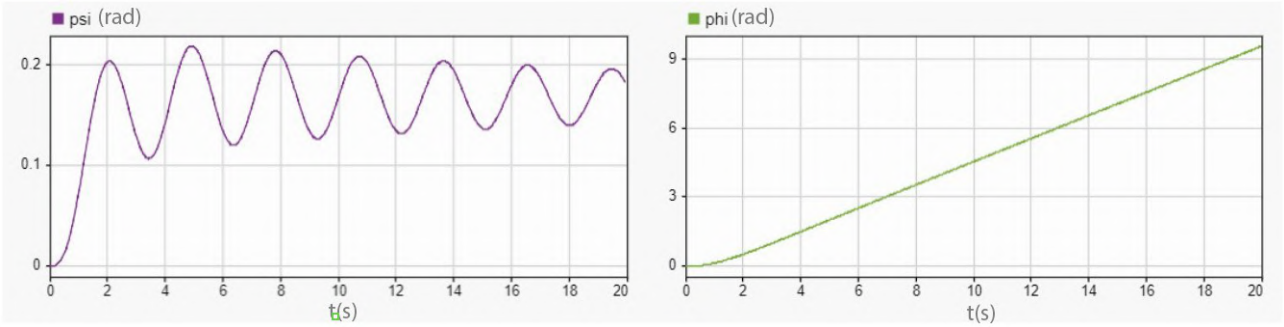


Figure 2.30: $y(t=0) = (0\ \ 0)^T$ and $u = (0.5\ \ 0.5)^T$ non linear

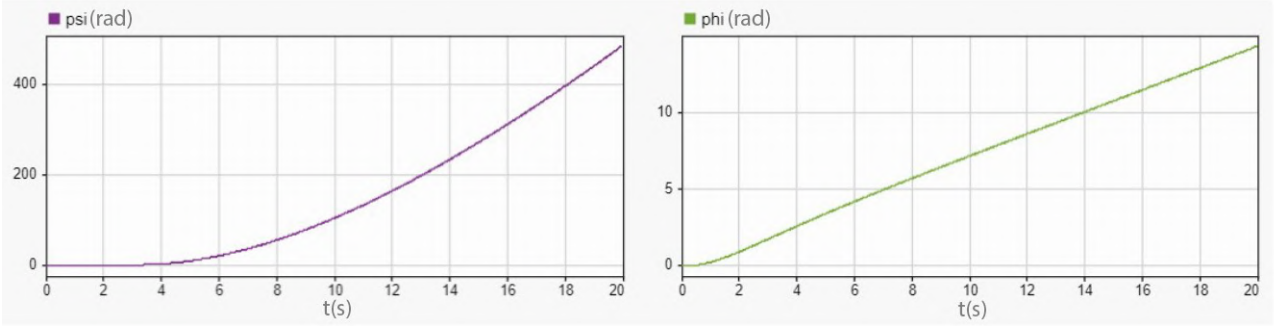


Figure 2.31: $y(t=0) = (0\ \ 0)^T$ and $u = (2.5\ \ 0)^T$ non linear

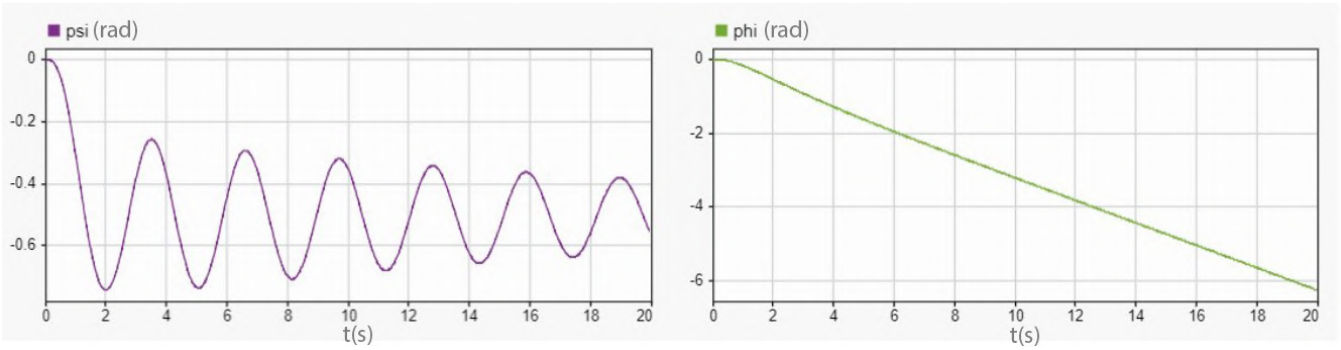


Figure 2.32: $y(t=0) = (0 \;\; 0)^T$ and $u = (-2.5 \;\; 0)^T$ non linear

The only difference is that the vertical system may be unstable too depending on the value of the input, for the negative value it is still stable but for positive $u > 2.25$ *approx* the system is unstable.

Conclusion

The nonlinear model is Lyapunov stable, BIBO unstable, oscillatory in the vertical plane, and with a strong coupling in the horizontal plane.

2.4.4.2 Controllability

[accessibility] The system is **accessible** from a state x_0 if for any neighborhood V of x_0, there exists a control input $u(t)$ and a time $t > 0$ such that the system state $x(t)$ can be driven into V. **Verification Condition**: Check if the distribution generated by the vector fields $f(x)$, $g(x)$, and their iterated Lie brackets $\{f, g_i, [f, g_i], \ldots\}$ has full rank at x_0.

$$\text{Span}\left(\{f(x_0), g(x_0), [f,g](x_0), [[f,g],f](x_0), \ldots\}\right) = \mathbb{R}^n$$

[Weak controllability] The system is **weakly controllable** if starting from any initial state x_0, it can be driven arbitrarily close to any desired state x_f as time t tends to infinity, using appropriate control inputs $u(t)$.

Verification Condition: Check if the Lie algebra generated by the vector fields $f(x)$, $g(x)$, and their iterated Lie brackets cover the state space around x_0.

$$\text{Rank}\left(\{f, g, [f,g], [[f,g],f], \ldots\}\right) = n \quad at \quad x_0$$

(Strong controllability) The system is **strongly controllable** if, for any initial state x_0 and any final state x_f, there exists a control input $u(t)$ and a finite time T such that the state $x(t)$ can be driven exactly from x_0 to x_f within the time interval $[0, T]$.

Verification Condition: Check if the Lie algebra generated by the vector fields $f(x)$, $g(x)$, and their iterated Lie brackets spans the entire state space $\mathbb{R}^n$ at every point x in the state space.

$$\text{Rank}\left(\{f, g, [f,g], [[f,g],f], \ldots\}\right) = n \quad \forall x \in \mathbb{R}^n$$

For our system, the calculation of the lie brackets is too complicated to be made by hand, Using Matlab we find that the Nonlinearmodel is also controllable.

2.4.4.3 Observability

Observability in nonlinear systems is analogous to controllability but focuses on the ability to determine the system's internal states from its outputs. It ensures that the system's states can be uniquely reconstructed from its measurements over time. The same concepts of controllability exist for observability. We will focus only on the test verification

The system is **observable** if the observability matrix $\mathcal{O}$ has full rank. The observability matrix $\mathcal{O}$ is constructed from the partial derivatives of $h(x)$ with respect to x and its Lie derivatives with respect to $f(x)$ and $g(x)$:

$$\mathcal{O} = \begin{bmatrix} \dfrac{\partial h}{\partial x} \\ \dfrac{\partial h}{\partial x} \cdot f(x) \\ \dfrac{\partial h}{\partial x} \cdot g(x) \\ \vdots \\ \dfrac{\partial h}{\partial x} \cdot f^{(n-1)}(x) \cdot g(x) \end{bmatrix}$$

The system is observable if $\text{Rank}(\mathcal{O}) = n$.

For our system, the calculation of the lie derivatives is too complicated and time-consuming when done manually, using Matlab we find that the Nonlinear model is also observable.

2.5 Conclusion

This chapter introduced the TRMS, highlighting its role as both a laboratory device and a helicopter simulator that shares key characteristics with real-world systems. The examination of the TRMS led to the development of a nonlinear model that features significant coupling between its components and limitations in the input signals, reflecting the complexities of real helicopter dynamics.

Despite being theoretically observable and controllable, the model of the TRMS is inherently unstable. This instability, coupled with the highly nonlinear behavior of the system, presents substantial challenges for control. These difficulties underscore the need for advanced control strategies capable of managing the intricate dynamics and ensuring stable operation of the TRMS.

Chapter 3

Parameter Tuning through Simulations

3.1 System Set-Up for ADRC

To be compatible with an ADRC configuration, a system needs to be written in its cascade integration form described in (1.3). This form is written, if we take the state space vector as $x = [y \quad \dot{y} \quad \dots \quad y^{(r-1)}]^T$, in a direct manner for matched systems.

However, the TRMS is described by the non-linear state space representation (2.7), which opposes Definition 1.2.2 and describes two 3rd order mismatched interconnected sub-systems, starting with the pitch (vertical) axis:

$$y = x_1 = \psi \tag{3.1}$$

$$\dot{y} = \dot{\psi} = x_2 \tag{3.2}$$

$$\ddot{y} = \ddot{\psi} = \frac{b_1}{I_1} x_5 + h_{11} \tag{3.3}$$

$$y^{(3)} = \psi^{(3)} = \frac{k_1 b_1}{I_1 T_{11}} u_1 + h_{12} \tag{3.4}$$

the compact state space representation is then:

$$\Sigma_{(x,u)} : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{b_1}{I_1} x_5 + h_{11}(x) \\ \dot{x}_3 = \dfrac{k_1 b_1}{I_1 T_{11}} u_1 + h_{12}(x) \end{cases} \tag{3.5}$$

where $< h_{1i} \in \mathcal{C}^{3-i} >_i$ represent the unknown system dynamics and are defined as:

$$h_{11}(x) = \frac{1}{I_1}\Big(a_1 x_5^2 - M_{gr} \sin x_1 - B_{1\psi} x_2 + B_{2\psi} x_4^2 \sin(2x_1) - K_{gy}(a_1 x_5^2 + b_1 x_5) x_4 \cos x_1\Big) \tag{3.6}$$

$$h_{12}(x) = \frac{\partial h_{11}}{\partial t} - \frac{b_1 T_{10}}{I_1 T_{11}} x_5 \tag{3.7}$$

It can be seen that these uncertainties are in all channels of the system's state space representation, not only in the control channel, which makes what is called a mismatched system.

Matched Perturbations:
Perturbations that enter the system through the same channels as the control input u. They affect the system dynamics in a manner similar to the control signals.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Bd(t) \\ y(t) = Cx(t) \end{cases} \tag{3.8}$$

Mismatched Perturbations:
Perturbations that enter the system through different channels than the control input u. They affect the system dynamics in a manner different from the control signals.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Ed(t) \\ y(t) = Cx(t) \end{cases} \tag{3.9}$$

Mixed Perturbations:
Perturbations that have both matched and mismatched components. They partially enter through the same channels as the control input u and partially through different channels.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Bd_1(t) + Ed_2(t) \\ y(t) = Cx(t) \end{cases} \tag{3.10}$$

Now that we've established that the TRMS non-linear model is mixed, we have to solve this issue, according to [32], by refining the total disturbance that affects the system performance to be able to proceed with ADRC's estimation/rejection strategy. Consider the following state transformation:

$$\begin{cases} \bar{x}_1 = x_1 \\ \bar{x}_2 = x_2 \\ \bar{x}_3 = \dfrac{b_1}{I_1} x_5 + h_{11}(x) \end{cases} \tag{3.11}$$

keeping the same measured controlled output, the system is then equivalently transformed into an integral chain with matched total disturbances:

$$\begin{cases} \dot{\bar{x}}_1 = \bar{x}_2 \\ \dot{\bar{x}}_2 = \bar{x}_3 \\ \dot{\bar{x}}_3 = \dfrac{k_1 b_1}{I_1 T_{11}} u_1 + h_{12}(x) = b_\psi u_1 + f_\psi \end{cases} \tag{3.12}$$

having b_ψ and f_ψ as the control input gain and the refined total disturbance, respectively.

PS: We allowed the control input to be inserted to the system through b_ψ, which depends on the system's parameters, because it has a $\pm 50\%$ error tolerance in an ADRC context and can, therefore, be considered free-of uncertainties, keeping (3.12) matched.

In the same manner, the state transformation:

$$\begin{cases} \bar{x}_4 = x_3 \\ \bar{x}_5 = x_4 \\ \bar{x}_6 = \dfrac{b_2}{I_2} x_6 + h_{21}(x) \end{cases} \tag{3.13}$$

applied to the yaw (horizontal) sub-system yields:

$$\begin{cases} \dot{\bar{x}}_4 = \bar{x}_5 \\ \dot{\bar{x}}_5 = \bar{x}_6 \\ \dot{\bar{x}}_6 = \dfrac{k_2 b_2}{I_2 T_{21}} u_2 + h_{22}(x) = b_\varphi u_2 + f_\varphi \end{cases} \tag{3.14}$$

where:

$$h_{21} = \frac{1}{I_2}\left(a_2 x_6^2 - x_7 - B_{1\varphi} x_4\right) \tag{3.15}$$

$$h_{22} = \frac{\partial h_{21}}{\partial t} - \frac{b_2 T_{20}}{I_2 T_{21}} x_6 \tag{3.16}$$

PS: It is good to note that the control cross-coupling term is included in $h_{22}(x)$ implicitly in $\dot{x}_7$ from the time derivative of $h_{21}(x)$

In Summary, the state space transformation described by:

$$\bar{x} = \begin{bmatrix} x_1 & x_2 & \frac{b_1}{I_1} x_5 + h_{11}(x) & x_3 & x_4 & \frac{b_2}{I_2} x_6 + h_{21}(x) \end{bmatrix}^T$$

yields the following matched integral chain:

$$\Sigma_{(\bar{x},u)} : \begin{cases} \dot{\bar{x}}_1 = \bar{x}_2 \\ \dot{\bar{x}}_2 = \bar{x}_3 \\ \dot{\bar{x}}_3 = b_\psi u_1 + f_\psi \\ \dot{\bar{x}}_4 = \bar{x}_5 \\ \dot{\bar{x}}_5 = \bar{x}_6 \\ \dot{\bar{x}}_6 = b_\varphi u_2 + f_\varphi \end{cases} \tag{3.17}$$

where:

$$f_\psi = \frac{\partial h_{11}}{\partial t} - \frac{b_1 T_{10}}{I_1 T_{11}} x_5 \quad ; \quad b_\psi = \frac{k_1 b_1}{I_1 T_{11}} \tag{3.18}$$

$$f_\varphi = \frac{\partial h_{21}}{\partial t} - \frac{b_2 T_{20}}{I_2 T_{21}} x_6 \quad ; \quad b_\varphi = \frac{k_2 b_2}{I_2 T_{21}} \tag{3.19}$$

The system can now be implemented in an ADRC configuration using the representation in the $(\bar{x}, u)$ basis. The following Figure illustrates the ADRC configuration used to control both the pitch and yaw axis of the TRMS.

PS: The stability proof of the change of basis is mentioned in Appendix C.

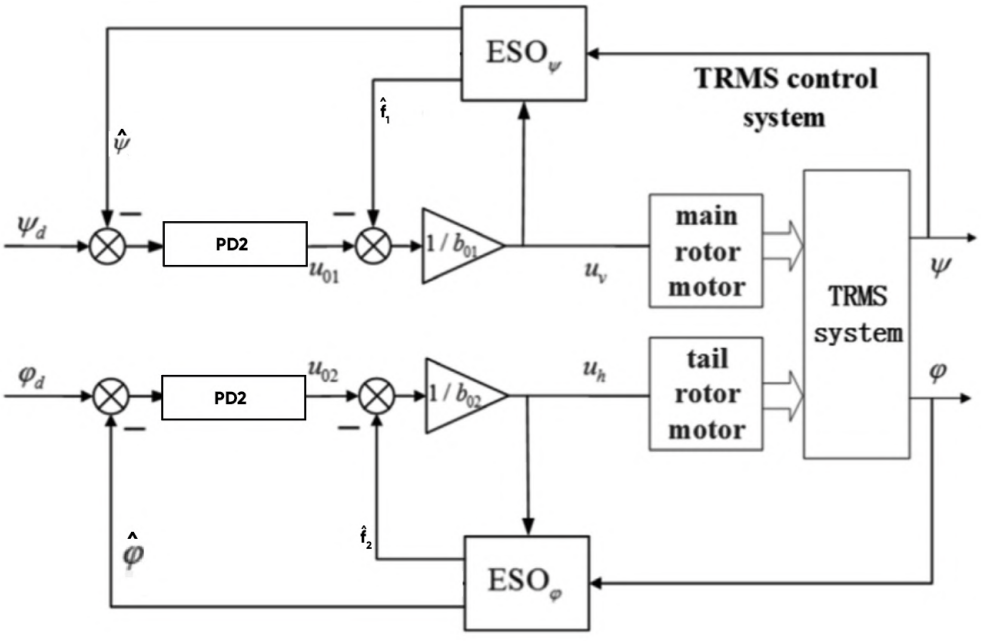


Figure 3.1: ADRC Structure on TRMS

In this chapter, we will tune the parameters of the ADRC's ESO and SFEC so that it will be used to control the TRMS's complex model through simulations. The design method is going to be either empirical or optimization-based, depending on the type of the bloc to design. In the latter case, Particle Swarm Optimization (PSO) is set as the optimization algorithm, which is presented in Section 3.2. The process of parameter tuning for both the linear non-linear ADRC structures is presented in Sections 3.3 and 3.4 respectively, and it will cover both the Extended State Observer and the State Error Feedback Controller blocs. A comparative study between LADRC and NL ADRC will then be done in terms of performance and robustness in Section 3.5.

3.2 Particle Swarm Optimization

Meta-heuristics algorithms are optimization strategies that find near-optimal solutions based on advanced (meta) procedures (heuristics). These algorithms rely on stochastic search and exploration of possible solutions in an "intelligent" way, giving generally better results than classical optimization algorithms and iterative methods. Their efficiency and the facility to implement due to the development of the hardware interfaces and the increase of the computational power made meta heuristics largely used in solving optimization problems in recent years.

Meta heuristics are classified into different categories:

- **Search strategy :** Some algorithms are used to improve a candidate's solutions in a local neighbourhood while others try to find solutions in broader search spaces.
- **Nature of the solution :** Some algorithms are single-solution types where a unique solution is improved while others have a particle-based approach where they improve simultaneously many candidates
- **Inspiration and origin :** Many effective recent algorithms are inspired by nature, they mimic the behaviour of species and natural phenomena that appear to have an optimal way of solving some problem.

Kennedy and Eberhart (1995) [33] proposed the PSO algorithm, which is a meta-heuristic particle-based algorithm inspired by the principle of swarm intelligence and the social interactions of birds flocking or fish schooling.

It can be used alone to find global solutions or in combination of other algorithms. Because it is based on social behaviour, the algorithm's principles are :

- **Intrinsic Experience :** Where the action of each particle is influenced by its personal previous results.
- **Social Influence :** involves the interaction among particles in the swarm, the action of each one of them is also influenced by the results of its surroundings.

3.2.1 The algorithm

In PSO, candidate solutions are described as particles $i, i \in 1....n$. At a time step T, we have these values :

- **The position** $x_i(T)$ **:** It is a vector in $\mathbb{X} \in \mathbb{R}^m$ the solution space for every particle. It represent a single solution for the problem (where a solution has m variables)
- **The velocity** $v_i(T)$ **:** It is a vector of the same dimension as x_i for every particle. It describes the "movement" of the particle.
- **The personal best** $P_i(T)$ **:** It is the best solution $x_i(t), t \in 1....T$ with respect to a given cost function for every particle.
- **The global best** $G(T)$ **:** It is the best solution among all particles up to time T.

Given these values, the mathematical model of PSO is given for each particle's x_{ij}, $j \in 1...m$ variable as :

$$\begin{cases} v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (P_{ij}(t) - x_{ij}(t)) + c_2 \cdot r_2 \cdot (G_j(t) - x_{ij}(t)), \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \end{cases} \tag{3.20}$$

where r_1 and r_2 are random numbers between 0 and 1, w, c_1 and c_2 are real numbers.

The coefficient w c_1 and c_2 are dynamic, as the algorithms runs c_2 increases and w decrease, the goal is to give more importance to the global best terms and converge toward it and reduce the exploration of other regions.
In 2002, clerc and kennedy [34] proposed these construction coefficient with constant values :

$$\chi = \frac{2\kappa}{12 - \Phi - \sqrt{\Phi^2 - 4\Phi}}$$

where $\kappa \in [0, 1]$, and $\Phi = \Phi_1 + \Phi_2 \geq 4$.

Taking:

$$\kappa = 1, \quad \Phi_1 = 2.05, \quad \Phi_2 = 2.05$$

we have:

$$w = \chi$$
$$c_1 = \chi\Phi_1$$
$$c_2 = \chi\Phi_2$$

3.2.2 Application to the TRMS

The first step in solving an optimization problem is the correct understanding of the objective and its translation to a mathematical problem that can be solved by an algorithm. Generally, the mathematical problem is the minimisation or maximisation of an objective function (also called criteria or cost function) that captures the characteristics of the optimization function.

In control engineering, the criteria is a function of the error $e = y - y_r$, for PSO, the most common criteria are [35] :

1. **Integral Absolute Error (IAE)** defined as

$$\text{IAE} = \int_0^\tau |e(t)|\, dt. \tag{3.21}$$

2. **Integral Square Error (ISE)** expressed by

$$\text{ISE} = \int_0^\tau e(t)^2\,dt. \tag{3.22}$$

3. **Integral Time Absolute Error (ITAE)** represented as

$$\text{ITAE} = \int_0^\tau t|e(t)|\,dt. \tag{3.23}$$

4. **Integral Time Square Error (ITSE)** denoted as

$$\text{ITSE} = \int_0^\tau te(t)^2\,dt. \tag{3.24}$$

5. **Mean Square Error (MSE)** defined as

$$\text{MSE} = \frac{1}{\tau}\int_0^\tau e(t)^2\,dt. \tag{3.25}$$

6. **Integral Error (IE)** given as

$$\text{IE} = \int_0^\tau e(t)\,dt. \tag{3.26}$$

7. **Root Mean Square Error (RMSE)** defined as

$$\text{RMSE} = \sqrt{\frac{1}{\tau}\int_0^\tau e(t)^2\,dt}$$

Remark

The criteria can be improved by the integration of a weighted sum of the other control variables like the control effort u, the overshoot, and the response time.

Parameter	**Value**
Criteria	ITAE
Population	50
Max Iteration	30
w	0.7298
c_1	1.4962
c_2	1.4962

Table 3.1: PSO Parameters

[h!] [1] Initialize the swarm with N particles each particle $i = 1, 2, \ldots, N$ Initialize particle's position x_i randomly Initialize particle's velocity v_i randomly Initialize particle's best known position $p_i = x_i$

Initialize global best known position g among all particles' positions

stopping condition not met each particle $i = 1, 2, \ldots, N$ Update particle's velocity:

$$v_i \leftarrow \omega v_i + c_1 r_1(p_i - x_i) + c_2 r_2(g - x_i)$$

where ω is the inertia weight, c_1, c_2 are acceleration coefficients, and r_1, r_2 are random numbers in $[0, 1]$

Update particle's position:

$$x_i \leftarrow x_i + v_i$$

Evaluate the fitness of the new position x_i

fitness of x_i is better than fitness of p_i Update particle's best known position $p_i \leftarrow x_i$

fitness of x_i is better than fitness of g Update global best known position $g \leftarrow x_i$

3.3 LADRC Parameter Tuning

Linear ADRC means that the ESO and SFEC are linear i.e. there's no insertion of non-linear functions in the correction terms of the observation and tracking errors. In this section, the considered LESO is a LO described by:

$$\Sigma_{LESO} : \begin{cases} \dot{z}_1 = z_2 + L_1(y - \hat{y}) \\ \dot{z}_2 = z_3 + L_2(y - \hat{y}) \\ \dot{z}_3 = z_4 + bu + L_3(y - \hat{y}) \\ \dot{z}_4 = L_4(y - \hat{y}) \end{cases} \tag{3.27}$$

and the considered SFEC is a PD2 described by:

$$u_0 = y_r^{(3)} + K_p \ e + K_d \ \dot{e} + K_{dd} \ \ddot{e} \tag{3.28}$$

Since both the controller and observer are linear, and the ADRC approach works on the canonical form of the system, a simple pole placement would be enough to give the closed-loop stable dynamics. However, our goal is to present a systematic approach to optimal parameter tuning of ADRC's blocs to solve control problems related to the highly non-linear high-order MiMo system that describes the TRMS. Optimization-based tuning is therefore required.

The controller and observer design method, whose parameters are subject to an optimization algorithm, is called bandwidth parametrization. Although Z.Gao proposed a "single cut-off frequency" version of this method to reduce the number of parameters to be tuned in [20], we will still proceed with different poles for the controller and the observer because 1) using multiple pole placement gives the freedom of changing some control actions without affecting the others i.e. without tuning constraints and 2) the parameter tuning will be done by the PSO algorithm for most of the time anyway.

Starting off with the SFEC, the differential equation described by the TRMS tracking error in an ADRC closed loop with a PD2 controller is described by:

$$e^{(3)} + K_{dd} \ \ddot{e} + K_d \ \dot{e} + K_p \ e = 0 \tag{3.29}$$

which translates to the Laplace domain in the form of a characteristic polynomial:

$$P(s) = s^3 + K_{dd} \ s^2 + K_d \ s + K_p \tag{3.30}$$

which is then identified with:

$$G(s) = (s + w_{c1})(s + w_{c2})(s + w_{c3}) \tag{3.31}$$

and by identification, the controller gains are:

$$\begin{cases} K_p = w_{c1} \ w_{c2} \ w_{c3} \\ K_d = w_{c1}(w_{c2} + w_{c3}) + w_{c2} \ w_{c3} \\ K_{dd} = w_{c1} + w_{c2} + w_{c3} \end{cases} \tag{3.32}$$

In the same manner, the LESO gains are given by:

$$\begin{cases} L_1 = w_{o1} + w_{o2} + w_{o3} + w_{o4} \\ L_2 = w_{o1} \ w_{o2} + w_{o3} \ w_{o4} + (w_{o1} + w_{o2})(w_{o3} + w_{o4}) \\ L_3 = (w_{o1} + w_{o2}) w_{o3} \ w_{o4} + (w_{o3} + w_{o4}) w_{o1} \ w_{o2} \\ L_4 = w_{o1} \ w_{o2} \ w_{o3} \ w_{o4} \end{cases} \tag{3.33}$$

Our task now is to find the optimal value of $w_c = [w_{c1} \;\; w_{c2} \;\; w_{c3}]^T$ and $w_o = [w_{o1} \;\; w_{o2} \;\; w_{o3} \;\; w_{o4}]^T$ so that the TRMS behaves in an optimal manner, subject to the fixed criterion, in its closed-loop form. Depending on the complexity of the task and/or the performance requirement, we will tune the parameters of the LO and PD2 controller in the most simple yet efficient way possible.

3.3.1 ESO Tuning

To have an idea of the location of the optimal solution, we start off by using the single pole placement :

$$w_o = \begin{pmatrix} 15 & 15 & 15 & 15 \end{pmatrix}^T$$

as proposed in [20] on the open loop TRMS system, which yields the following observer gains:

$$L = \begin{pmatrix} 60 & 1350 & 13500 & 50625 \end{pmatrix}^T$$

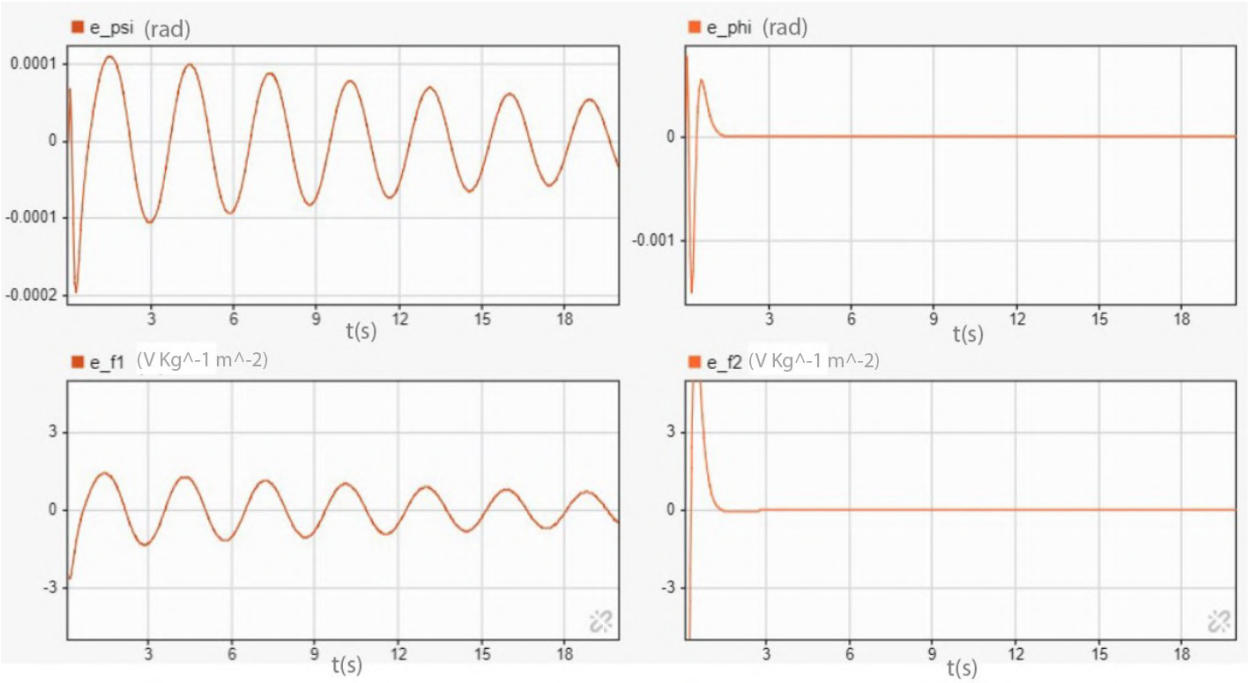


Figure 3.2: Estimation Error on the Output and Total Disturbance of the (left) pitch (right) yaw Sub-systems for $w_o = (15 \;\; 15 \;\; 15 \;\; 15)^T$

As it can be seen from Figure 3.2, the observation error on the output signals is within a 10^{-4} and 10^{-3} range for the pitch and yaw signals respectively, meaning that the output signals' reconstruction performance is already satisfactory. However, the estimation error on the total disturbance for the pitch sub-system shows a slightly damped oscillatory dynamic with a considerable amplitude. Generally, system uncertainties $f(t)$ can be estimated with a bounded error using a LESO if either $\dot{f}(t)$ or $f(t)$ is bounded, as stated in [36].

However, this observation can't be regarded as an issue because the oscillations are due to the test being performed in the open loop of this complex system i.e. there won't be any of that once we insert the observer in the closed loop system. Additionally, for applications focusing on open loop tests, enhancing the quality of the total disturbance's estimates can be done by adding its derivatives as to-be-reconstructed states by the observer, as proposed in [37].

To upgrade the performance of this LO output-wise, let us set slightly faster poles:

$$w_o = \begin{pmatrix} 17 & 17 & 18 & 18 \end{pmatrix}^T$$

then the identification (3.33) gives:

$$L = \begin{pmatrix} 70 & 1837 & 21420 & 93636 \end{pmatrix}^T$$

Before proceeding to analyze the performance of this observer, it can intuitively be seen –from a control engineer's perspective– that this observer can be regarded as a high-gain observer, where moderate changes in the observer's pole placement cause large changes in its gains.

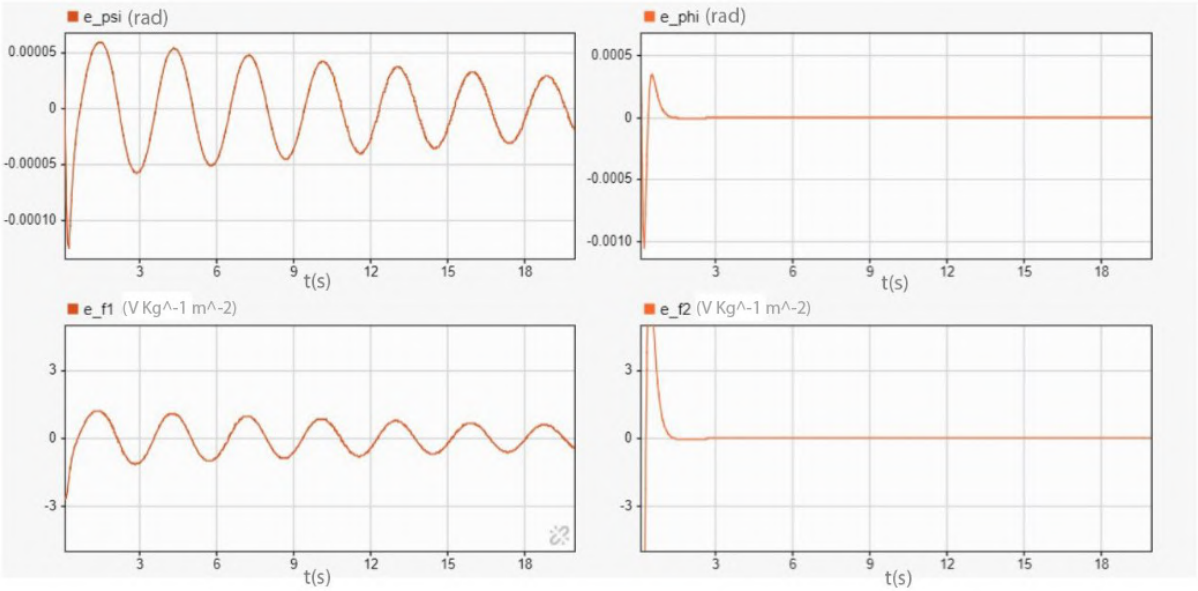


Figure 3.3: Estimation Errors for $w_o = (17\ \ 17\ \ 18\ \ 18)^T$

Figure 3.3 shows that the error on the reconstructed outputs is now about $\times 10$ times smaller than the previous one. Meaning that the performance of this LO is better than the previous one.

Even though the performance of the observer is already satisfactory, let's try and do an even faster pole placement at:

$$w_o = \begin{pmatrix} 19 & 19 & 20 & 20 \end{pmatrix}^T$$

which yields:

$$L = \begin{pmatrix} 78 & 2281 & 29640 & 144400 \end{pmatrix}^T$$

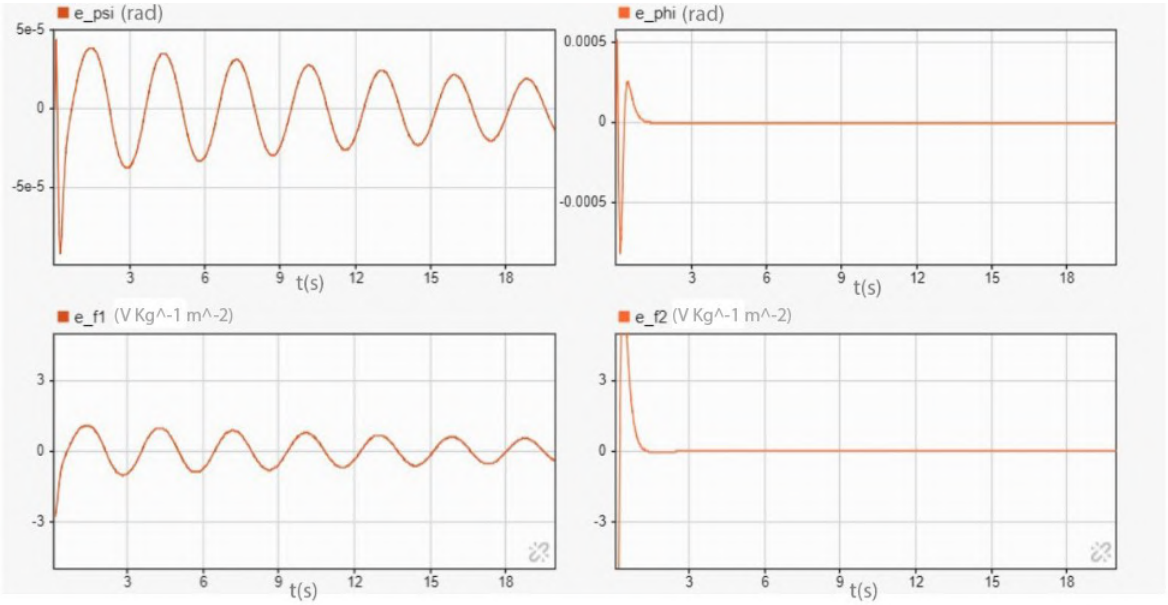


Figure 3.4: Estimation Errors for $w_o = (19\ \ 19\ \ 20\ \ 20)^T$

By observing Figure 3.4, one can say that no improvements have been made in the observer's performance and it is still in the same already-sufficient estimation error range. This means that we are at the limits of the observer's performance and can therefore be satisfied with the 2nd pole placement.

The real and reconstructed outputs and total disturbances of the TRMS' horizontal and vertical sub-systems are therefore illustrated in the following figure:

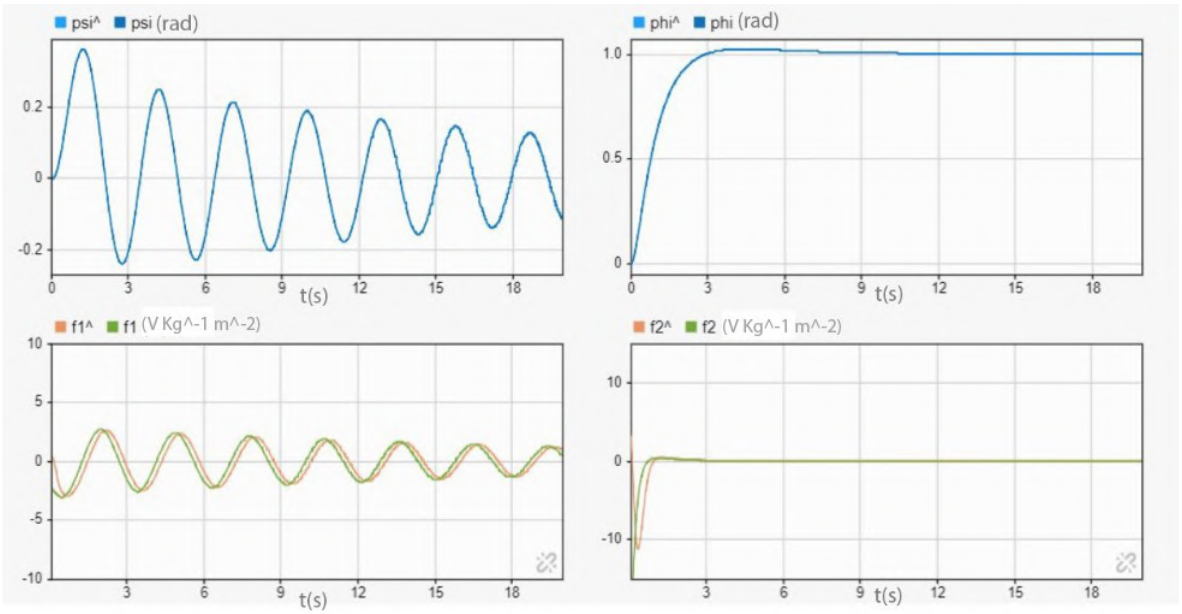


Figure 3.5: Real and Reconstructed Outputs and Total Disturbances using an Optimal LO

In short, we designed an optimal LO to reconstruct the states of the TRMS' highly non-linear model through pole placement without having to run an optimization algorithm, making this approach an effective solution for this use case.

3.3.2 SFEC Tuning

Now that the $\hat{f} \approx f$ condition is satisfied, and the TRMS non-linear matched model (3.17) can be transformed into its canonical form, the SFEC needs to be well-tuned in order to get the best of both worlds: performance and robustness.

At first glance, it might look like the PD2 controller is one parameter less than the LO and can therefore follow the same tuning method. But in reality, the observer tuning can be done to each sub-system independently of the other because 1) the control action was not touched and 2) the tuning was done in the open loop. But unlike the observer, each one of the controller's gains is associated with a different control action, meaning that careful attention should be taken as to how each control action affects its subsystem as well as TRMS plant as a whole. Therefore, Empirical tuning is challenging, and optimization algorithms are needed.

Using the PSO algorithm with the parameters described in Table 3.1 simultaneously on the pitch and yaw controllers, and taking the data from real-time simulations of the closed-loop model at the maximum values of the references (because if the maximum reference can be tracked then tracking lesser values is guaranteed as well), the optimal gains are:

$$K_\psi = \begin{pmatrix} 88.3994 & 59.5359 & 13.3648 \end{pmatrix}^T \quad ; \quad K_\varphi = \begin{pmatrix} 450 & 285 & 32 \end{pmatrix}^T$$

and the loss in the associated cost function is illustrated in the following figure:

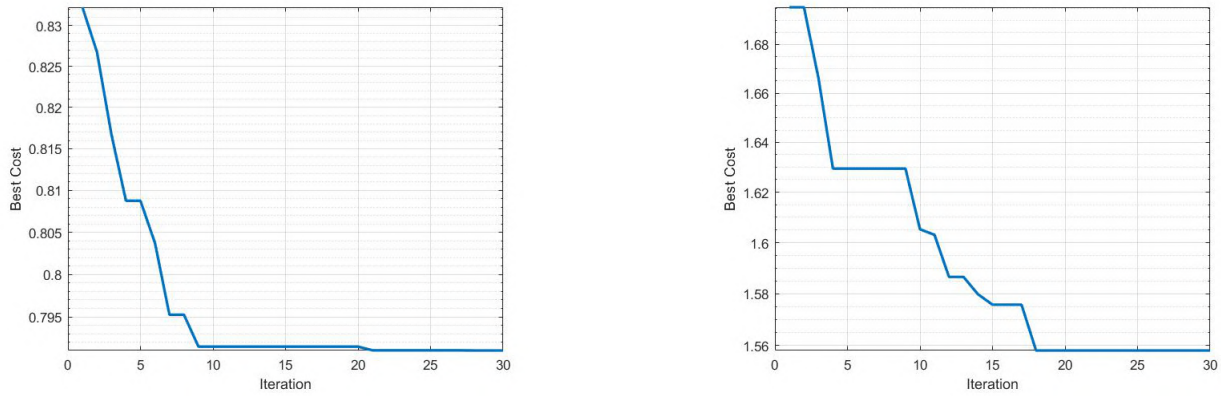
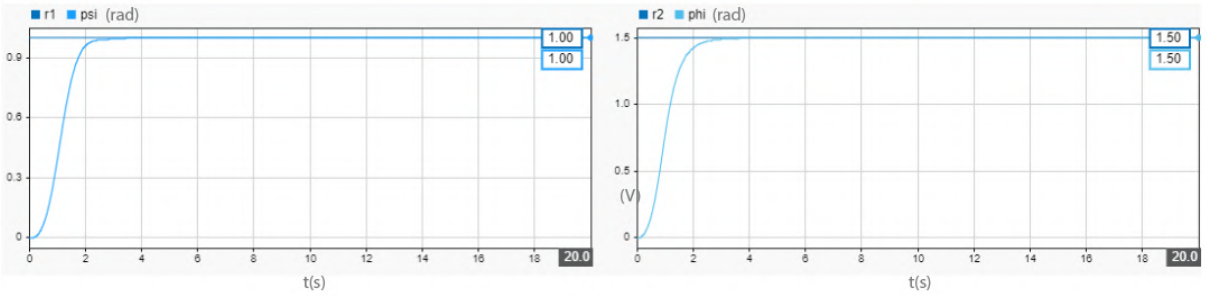
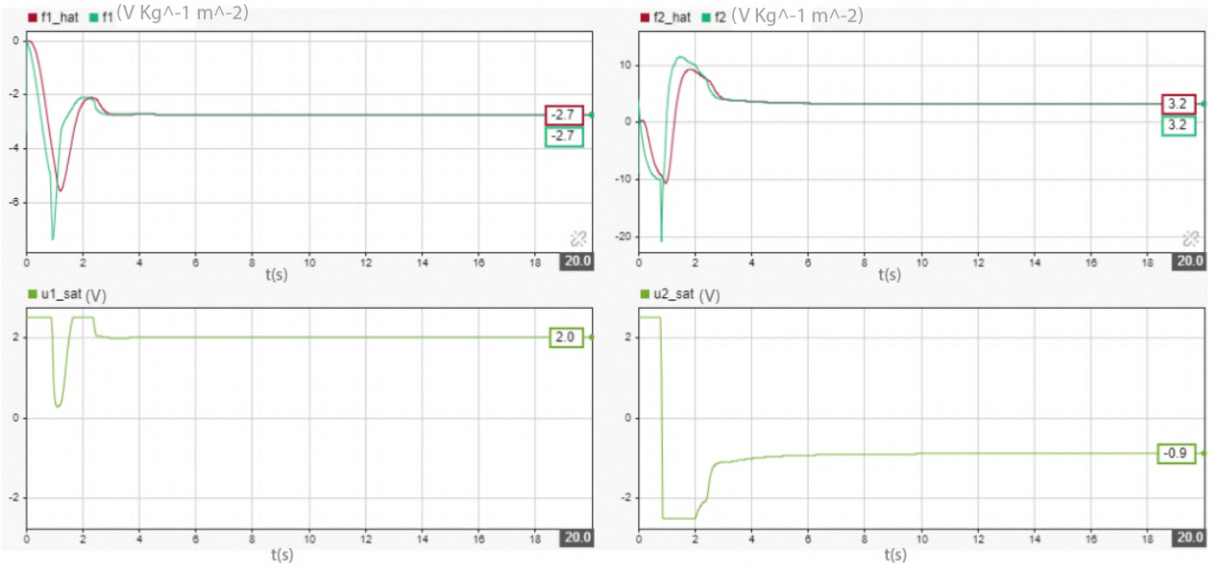


Figure 3.6: Loss in Cost Function for the (left) pitch and (right) yaw PD2 Controller



(a)

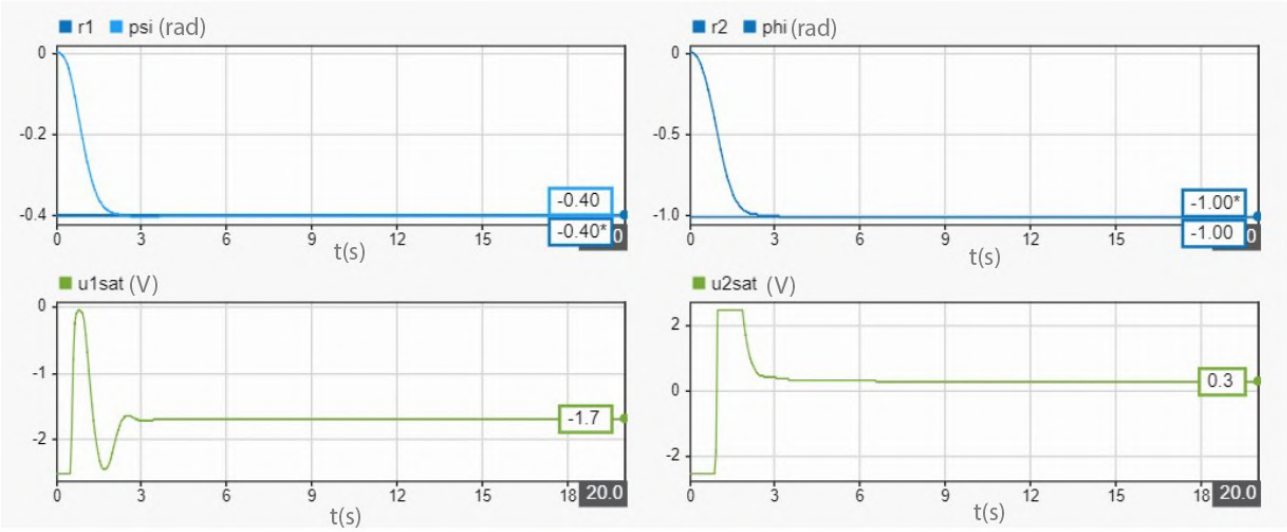


(b)

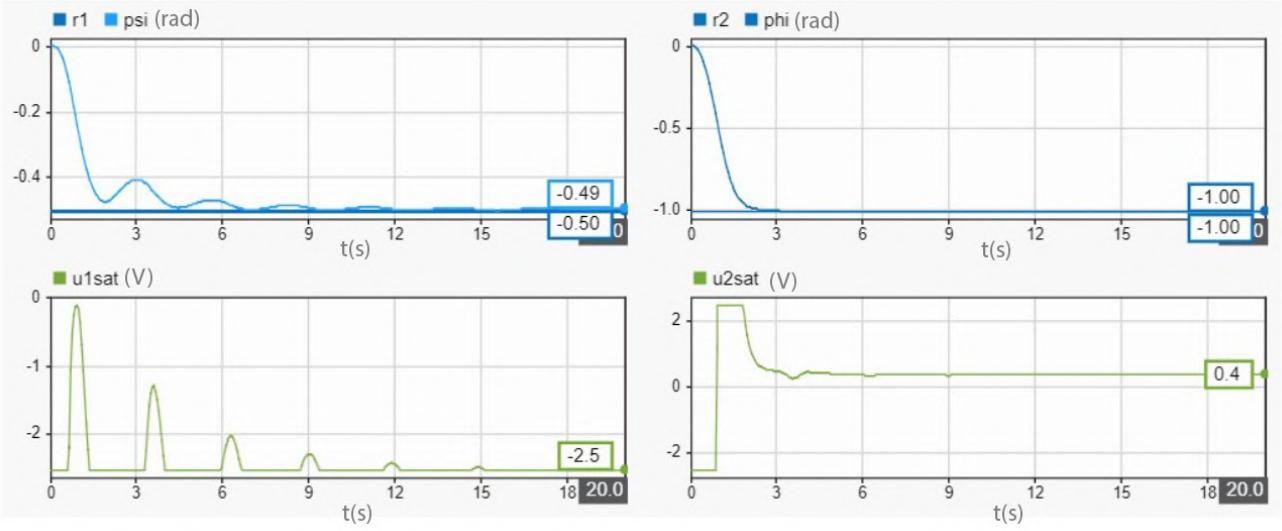
Figure 3.7: Optimal PD2 Performance for a positive Set-point Tracking: (a) output tracking (b) total disturbance estimation and control input energy

The performance of this controller is illustrated in Figure 3.7 for a positive step reference. As it can be seen, the tracking objective was achieved within 4 seconds (and more specifically, the exact moment that the total disturbances are properly estimated) and the steady state error equals zeros for both the pitch and yaw angles with no overshoot or control input saturation.

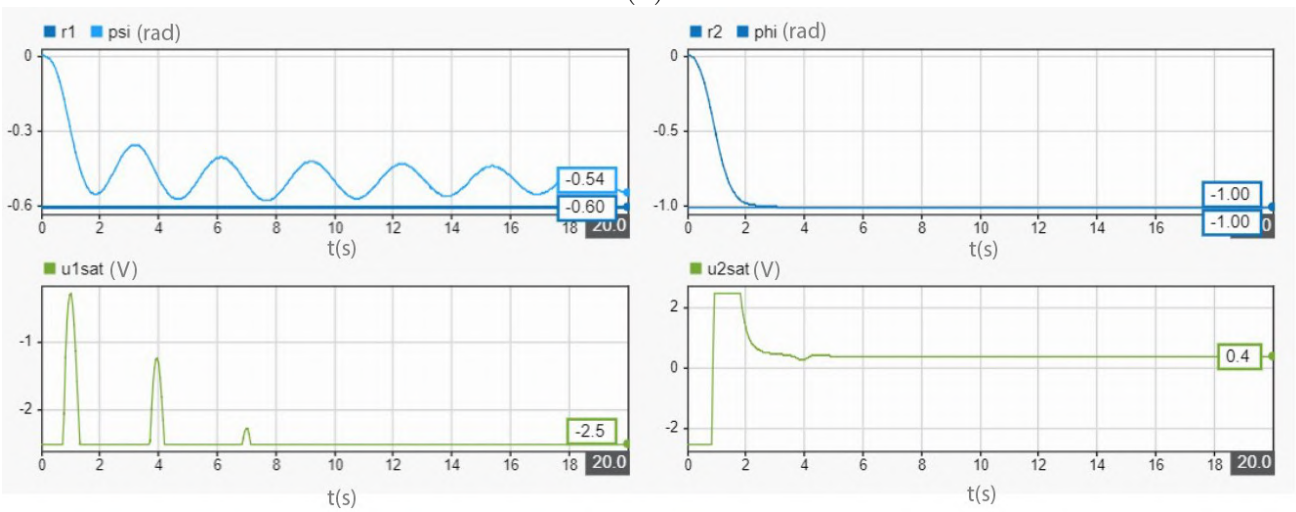
Additionally, the references were set near the maximum positive values given in Chapter 2. This means that the designed controller apports fast convergence properties to the closed loop system with no overshoot and reduced control input energy, and that is for any given positive reference, which affirms that this is the most optimal tuning of the controller.



(a)



(b)



(c)

Figure 3.8: Optimal PD2 Performance for (a) $r_1 = -0.4\,rad$ (b) $r_1 = -0.5\,rad$ (c) $r_1 = -0.6\,rad$

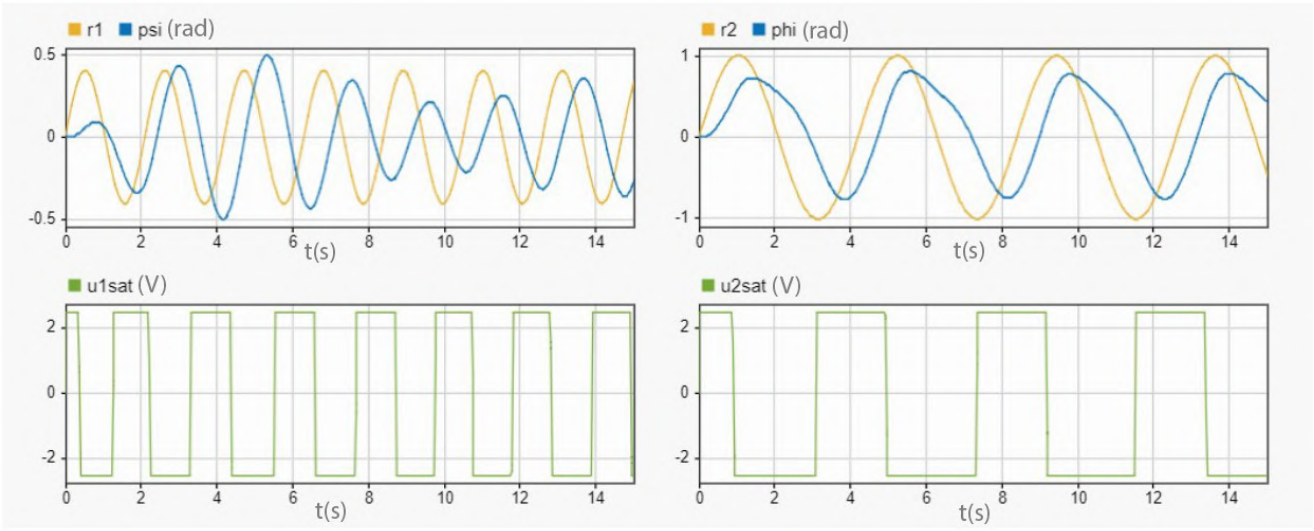
Since the tracking performance for other reference signals needs to be investigated as well, will investigate constant negative and sine wave reference tracking to analyze both static and dynamic tracking capacities of the designed controller.

By observing the closed-loop dynamics in Figure 3.8, we can notice that the same tracking performance is kept for negative pitch references that satisfy:

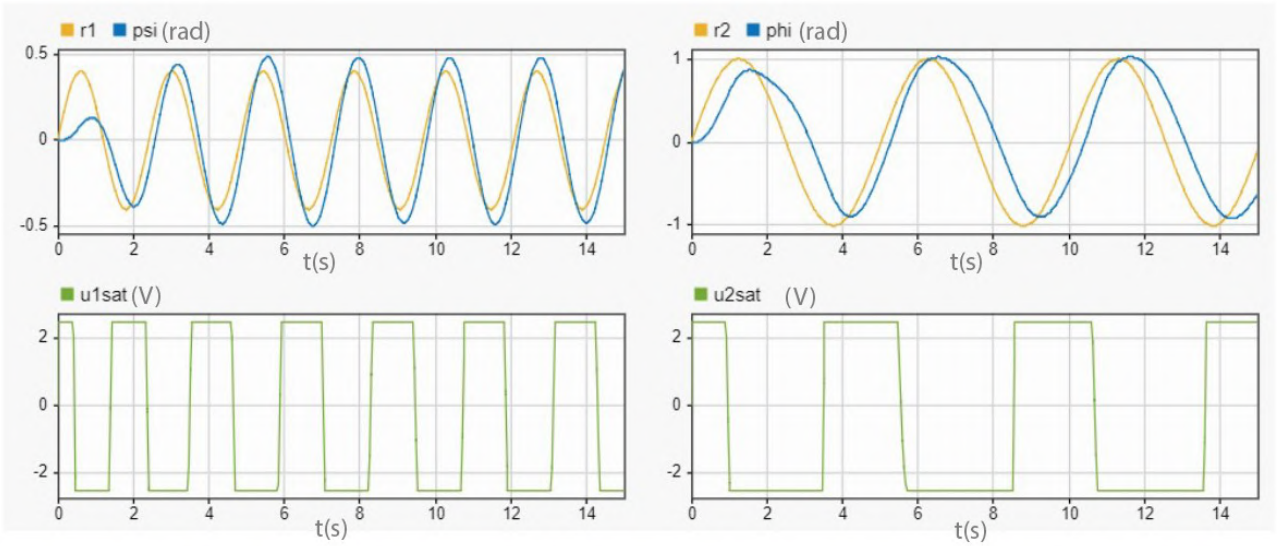
$$r_1 \geq -0.4\,rad \tag{3.34}$$

because above these limits, the TRMS's non-linear model exhibits persistent large values of the total disturbances on its vertical axis, such disturbances are estimated properly but cannot be compensated by the ADRC's compensation control law because of the $\pm 2.5\,V$ input saturation on the DC motors. Therefore, reference trajectories given for the pitch angle should respect condition (3.34).

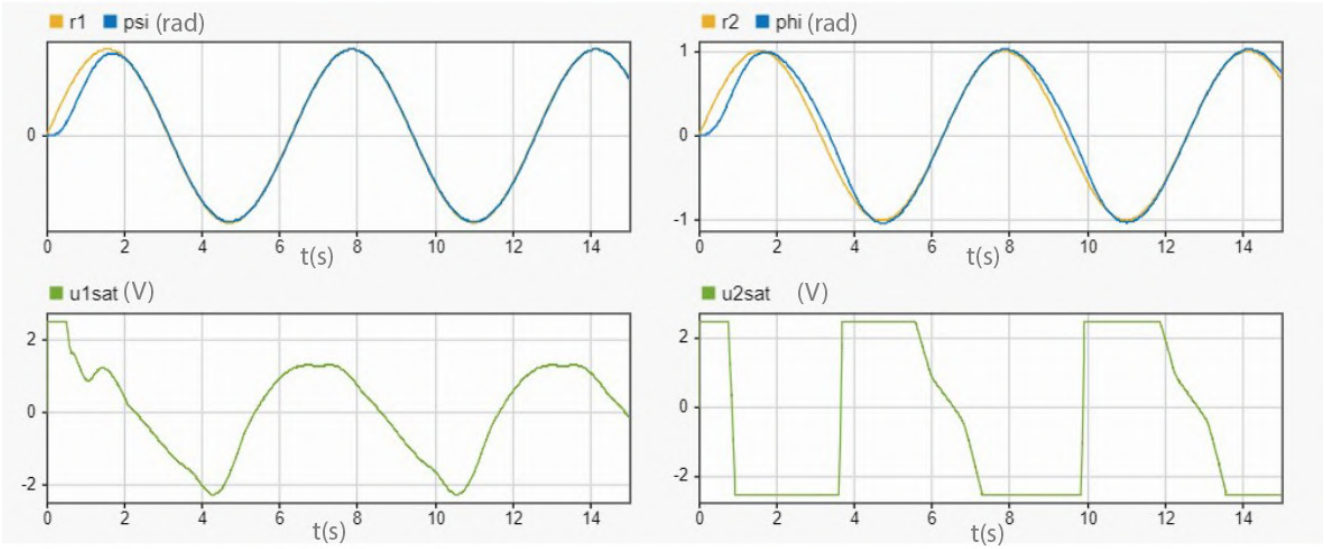
As for the yaw angle, it can be seen that the same optimal tracking performance is achieved for any negative value within the physical range of the TRMS. This comes naturally as the tail motor does not exhibit strong inertia contrary to the main rotor, meaning that more effort (and, therefore, more energy i.e. more input voltage) is needed to control the latter than the former, which can be seen in the fact that $u_1 > u_2$ at the steady state of the previous simulations.



(a)



(b)



(c)

Figure 3.9: Optimal PD2 Performance for Sine (a) $w_r = (3\ 1.5)^T$ (b) $w_r = (2.6\ 1.25)^T$ (c) $w_r = (1\ 1)^T$ References

For variable references, trajectory tracking was performed on sine wave references with different

frequencies, as shown in Figure 3.9. It can be seen that there's a speed limit to which the reference can be tracked without any lag, which is only logical because as seen in set-point tracking, the closed loop takes about 3 seconds at max to reach its desired value. This translates into sine references tracking as the frequency limit on the reference signal:

$$w_r \leq 1\, rad \tag{3.35}$$

Otherwise, the PD2 controller would try to catch up with the fast reference signal and would cause control input saturation, as shown in Figure 3.9 (a) and (b).

PS: It is good to note that the optimal closed-loop poles were always converging toward their upper bounds (set for the PSO algorithm), which is natural from a controller engineering perspective, as higher gains generally mean better performance. However, the upper bound was kept at $w_{max} = 15\, rad$ which showed satisfactory results as the previous figures illustrated.

Now that the LO and PD2 blocs are well tuned, the ADRC closed loop of the underactuated TRMS plant behaves optimally in terms of convergence speed and steady-state errors without the need to saturate the DC motors.. Also, the TD bloc is not present in this structure because the transient trajectory generation described in (1.15) uses a non-linear input term and therefore is often not included in LADRC configurations.

3.4 NL ADRC Parameter Tuning

Non-linear ADRC means that its blocs (namely the ESO and SFEC) are non-linear i.e. contain non-linear terms.

In the same manner as the previous section, we will design non-linear ESO and SFEC to tune an NL ADRC structure for the TRMS system. The used SFEC is a NL PD2 controller defined by:

$$u_0 = y_r^{(3)} + K_p\ \ fal(e, \alpha_p, \delta_p) + K_d\ \ fal(\dot{e}, \alpha_d, \delta_d) + K_{dd}\ \ fal(\ddot{e}, \alpha_{dd}, \delta_{dd}) \tag{3.36}$$

where $e = y_r - y$ is the tracking error and $K = (K_p\ K_d\ K_{dd})^T$, $\alpha = (\alpha_p\ \alpha_d\ \alpha_{dd})^T$, and $\delta = (\delta_p\ \delta_d\ \delta_{dd})^T$ are the parameter vectors that need to be set in order to design an optimal non-linear controller. Similarly, the corresponding observer is defined as:

$$\Sigma_{ESO} : \begin{cases} \dot{z}_1 = z_2 - L_1 fal(e_o, \alpha_1, \delta_1) \\ \dot{z}_2 = z_3 - L_2 fal(e_o, \alpha_2, \delta_2) \\ \dot{z}_3 = z_4 + bu - L_3 fal(e_o, \alpha_3, \delta_3) \\ \dot{z}_4 = -L_4 fal(e_o, \alpha_4, \delta_4) \end{cases} \tag{3.37}$$

where $e_o = \hat{y} - y$ is the estimation error and $< L_i >_i$, $< \alpha_i >_i$, and $< \delta_i >_i$ are the observer parameters that need to be set in order to optimize its reconstruction performance. Also, the linearity window interval is set as $\delta_i = 0.1$ for all actions in the ESO as well as the NL PD2 controller.

3.4.1 ESO Tuning

This time around, tuning the ESO is done via optimization. Using the PSO algorithm with the parameters defined in 3.1, the optimal observer parameters are:

$$w_o = \begin{pmatrix} 17 & 17 & 18 & 18 \end{pmatrix}^T \quad ; \quad \alpha_o = \begin{pmatrix} 1.2277 & 0.1 & 0.1 & 0.1 \end{pmatrix}^T$$

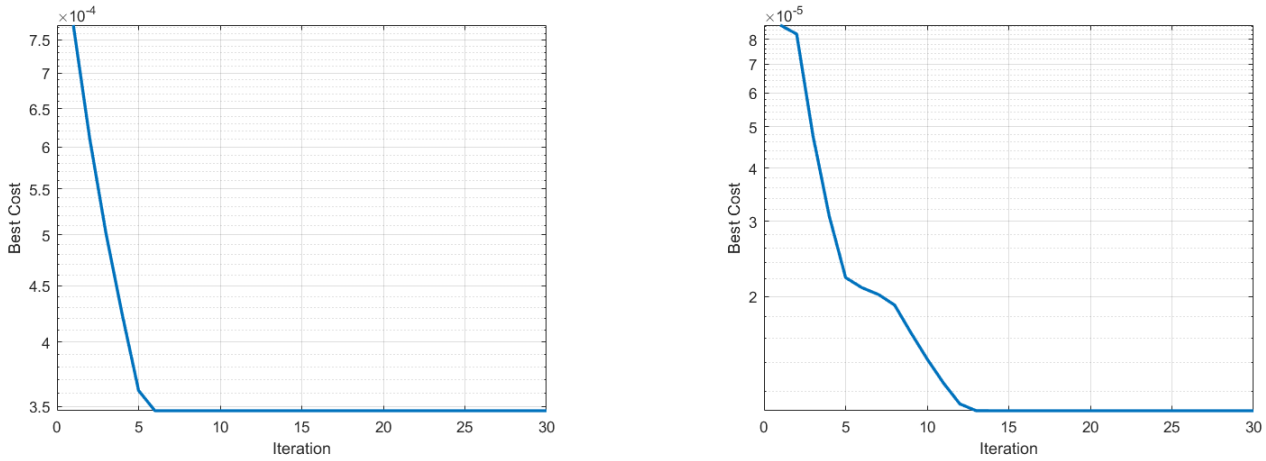


Figure 3.10: Loss in Cost Function for the (left) pitch and (right) yaw NLESO

and the associated loss in the cost function is shown in the following figure: First, notice that the optimal observer gains are equal to to the ones we designed in the linear case (see Section 3.3), which is logical since we set the upper bound of these poles in the PSO algorithm at that place. Because, as previously affirmed, not only do they present an optimal point but they also set the performance limit for the ESO. This means that if we set a higher upper bound, the optimal gains would go there. In other words, it is just that the algorithm will keep on converging to neglectable better solutions while we have the intellect to stop at the performance limits manually.

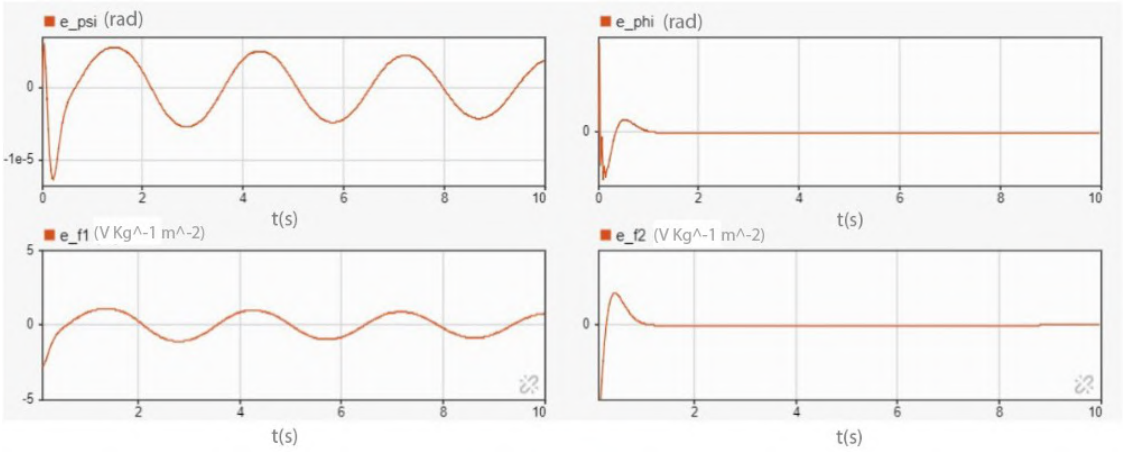


Figure 3.11: Estimation Errors for Optimal ESO

The figure above shows that the estimation errors converge to zero within 1 second for the yaw angle and total disturbance, and almost instantly for the pitch angle and total disturbance at a 10^{-4} range. This confirms the good performance of the optimized non-linear ESO.

3.4.2 SFEC Tuning

In the same manner, the NL PD2 tuning through PSO gives the optimal solutions:

$$w_c = \begin{pmatrix} 4.6692 & 5.7686 & 8.0 \\ 7.0378 & 3.99207 & 8.7527 \end{pmatrix} \quad ; \quad \alpha_c = \begin{pmatrix} 1.0811 & 1.2473 & 1.5754 \\ 1.2028 & 0.8319 & 0.6201 \end{pmatrix}$$

and the corresponding loss in the cost function is shown in the following figure:

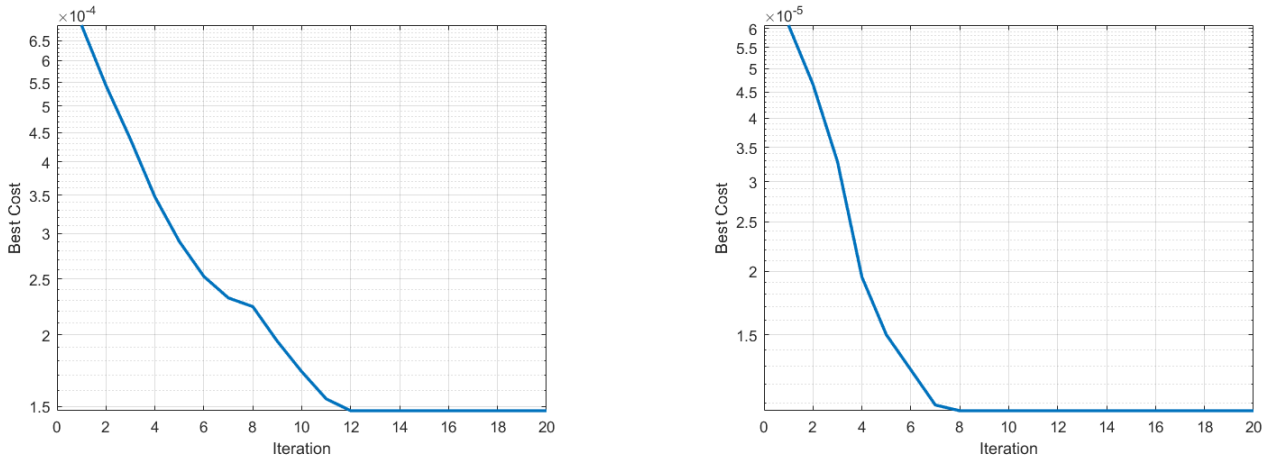
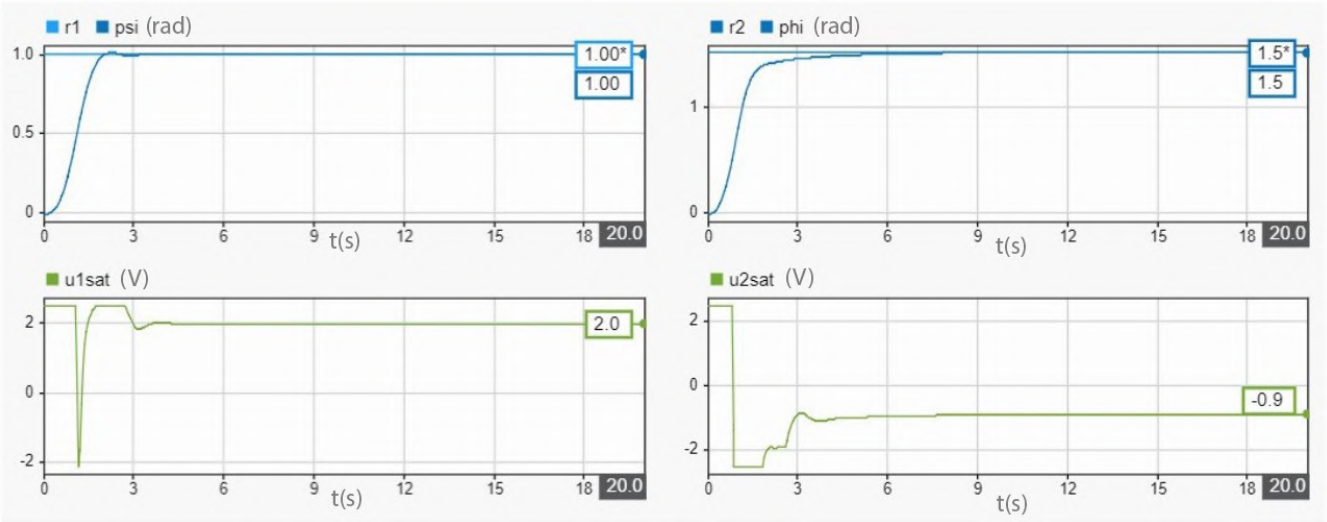
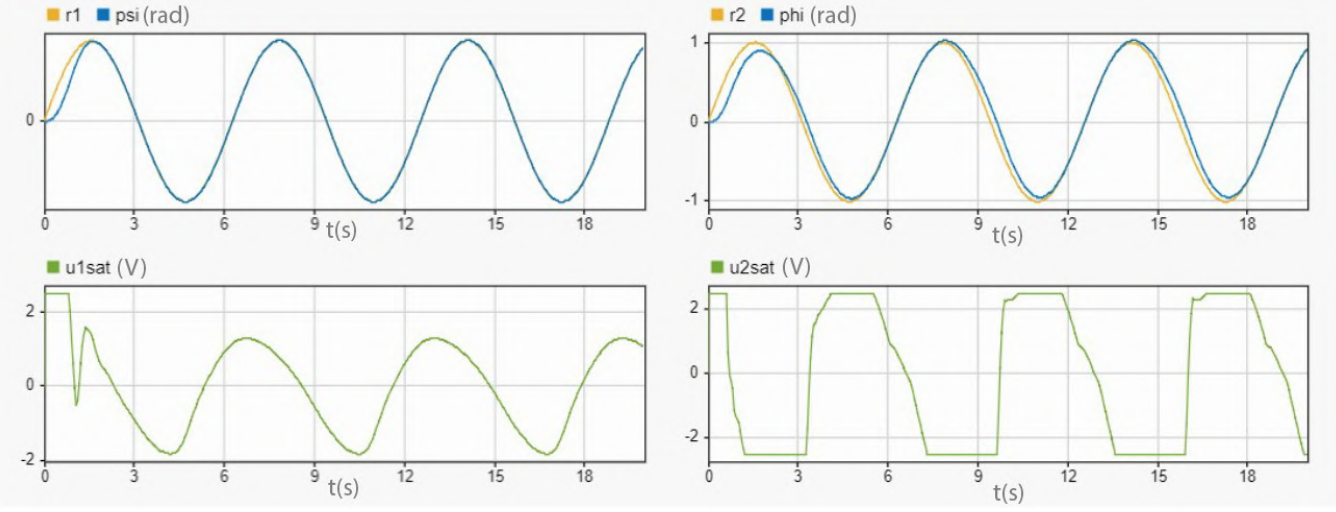


Figure 3.12: Loss in Cost Function for the (left) pitch and (right) yaw NL PD2 Controller

Similarly to the previous section, Figure 3.13 shows the tracking performance of the optimal controller for positive step as well as sine wave reference tracking. Overall, the same tracking performance was kept as the LADRC structure with a slightly faster response time. A more detailed comparative study between LADRC and NL ADRC is presented in Section 3.5.



(a)



(b)

Figure 3.13: (a) Positive Set-point (b) Sine Wave Reference tracking for Optimal NL PD2

We've now designed an optimal Non-Linear ADRC structure to ensure the performance of the closed loop through optimization-based tuning of the non-linear controller and observer and verified its static and dynamic tracking performance. But the real question now is, which one is better? LADRC or ADRC? This question will be answered in the following section.

3.5 Comparative study : LADRC and ADRC

As stated in , the linear feedback may not be the best choice of control approach, in this section, we will compare the simulation results of the optimal LADRC and ADRC algorithms obtained in different configuration.

3.5.1 Estimation performance analysis

To test the observer performance we will use the same open loop simulation we have done to tune the parameter of both observers

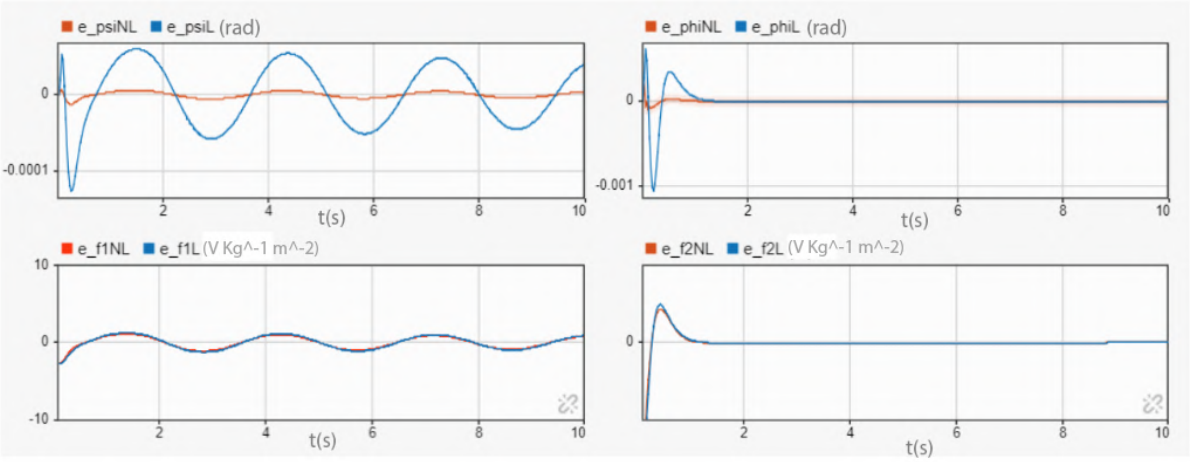


Figure 3.14: Estimation Error on the Output and Total Disturbance of the (left) pitch (right) yaw Sub-systems for the optimal LESO and NLESO

We notice that despite both the linear and non linear observer having a very small error and fast convergence, the NLESO is better in term of overshoot and time convergence for the output. Regarding the total disturbance estimation both observers have the same behaviour.
However, the gains of the NLESO are greater than the gains of the LESO as we kept the former ones at their upper bound, to have a more fair comparison and highlight the effect of the non linear function, let's put both observer with the same gains.

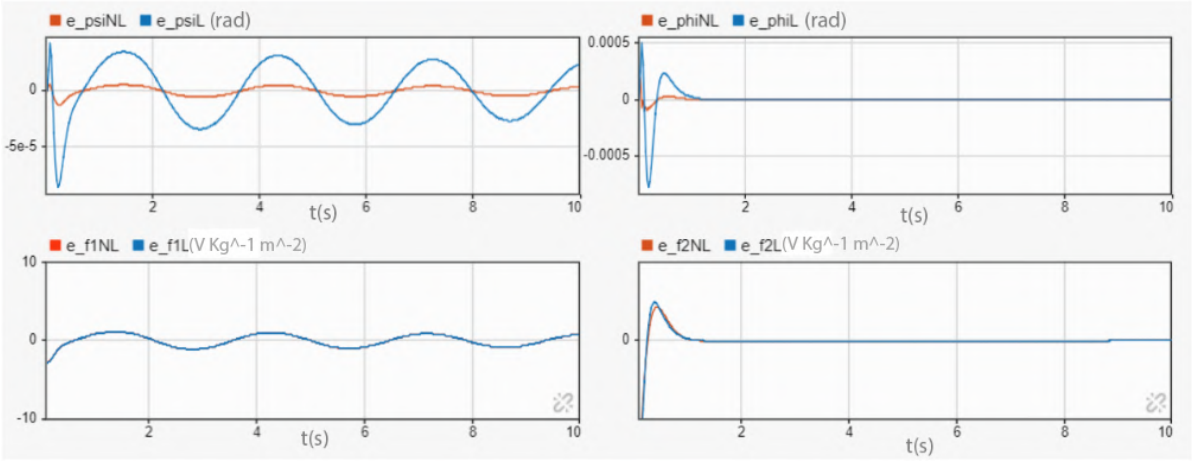


Figure 3.15: Estimation Error on the Output and Total Disturbance of the (left) pitch (right) yaw Sub-systems for LESO and NLESO at $w_0 = [20\,20\,20\,20]$

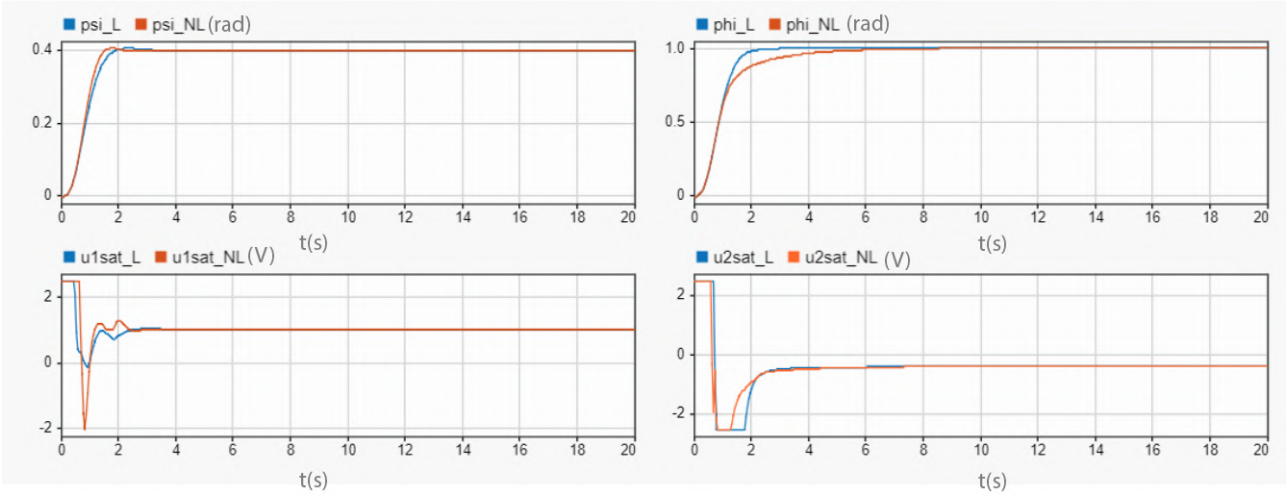
Now we can notice that despite having a better response for the linear observer, the NLESO has still a faster convergence time and less overshoot for the observation of the output, the performance for the total disturbance remains the same.

This is explained by the value of the α parameters: the PSO optimisation led all of them at the exception of one to their lower bound at 0.2. Having $\alpha < 1$ means a higher gain when $e_o < 1$, this means that when the observer is near convergence, the NLESO will be faster and since we have very small error from the beginning for the output and big ones for the total disturbance, the effect of the NLESO will be noticeable only for the output observation.

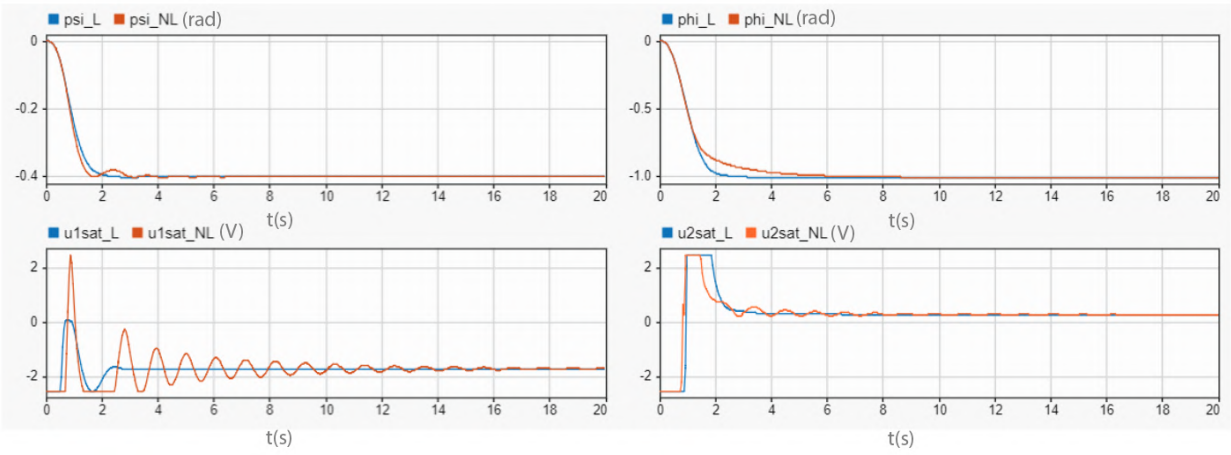
However, we should mention that this result will not have any impact on real implementations because the error is very small and not significant in both cases. Since the LESO is simpler, one can argue that it could be the best choice in real-life applications (at least for our system).

3.5.2 Tracking performance analysis

Now that we know that both observers will give similar results, the study of the LADRC vs ADRC will only depend on the performances of the feedback loop. We can notice for the simulations of the precedent section that the coupling term is compensated and the study of each plan of motion can be independent.



(a)



(b)

Figure 3.16: LADRC and ADRC Performance for a positive (a) and a negative (B) Set-point Tracking

Figure 3.16 shows the optimal response to a positive and negative step for both algorithms. For a positive step reference :

- In the vertical axis, the ADRC is slightly faster than the LADRC, however the control effort is larger in the transient time.
- For the horizontal axis, the difference is more noticeable but this time it is the LADRC that has a smaller response time.
- For both approaches the control input is the same at convergence with no saturation.

For a negative step reference:

- In the vertical axis, they seem to have a similar response the LADRC has a smoother behaviour, the control signal is also better as the NLADRC present some oscillations.
- For the horizontal axis the results are the same than for a positive step, LADRC is faster.

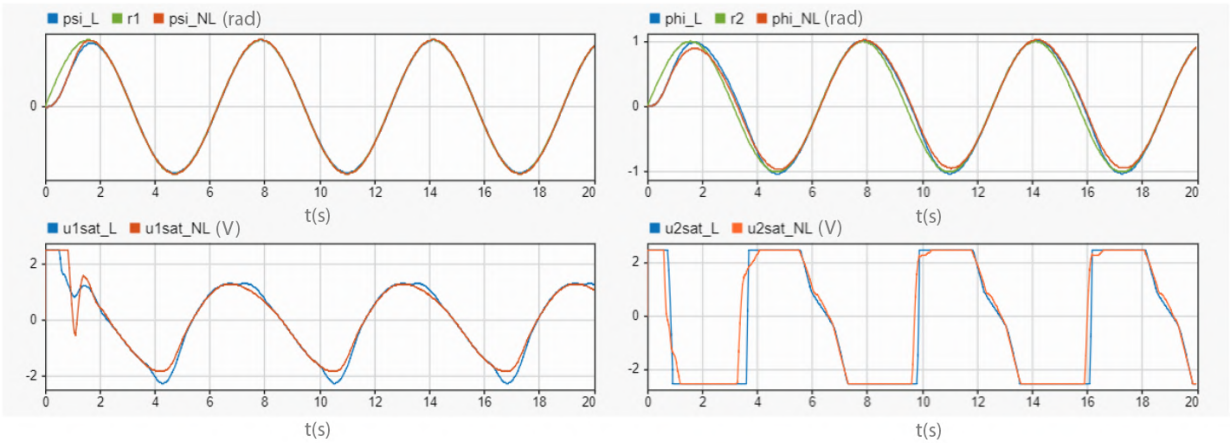


Figure 3.17: LADRC and ADRC Performance for a sinusoidal Tracking

Figure 3.17 shows the optimal response to a sin wave input signal at 1rad/s for both algorithms. We can notice that both outputs follow the reference signal for both controllers, however, the control effort is slightly smaller for NLADRC.

We can conclude from this nominal study of the ADRC and NLADRC that the linear version is better for fixed reference point tracking while the non-linear version has smaller solicitations for the actuator in the case of sin wave tracking, however, the difference is still not that significant.

3.5.3 Robustness analysis

We will now analyse the robustness of the ADRC and LADRC in the presence of perturbations, the main goal of the the ADRC approach. We will give for both sub systems a perturbations at different time, the perturbations will be a step or an impulse that is introduced in the second derivatives of the angles, they simulate the effect of a torque or a force that acts on the acceleration (2nd law of newton and equations 2.5 and 2.6)

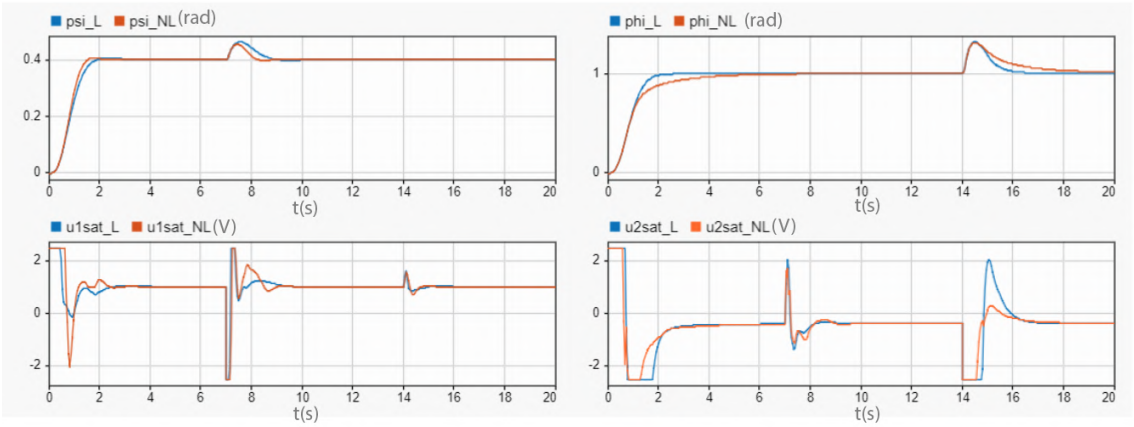


Figure 3.18: LADRC and ADRC Robustness for an impulse

For both, the system recovers from the perturbation quickly, the overshoot is relatively small, and we keep the same results of the nominal case, the ADRC recover faster in the vertical plan while the LADRC is faster for the horizontal plan, for both case the control input is greater for the faster approach.

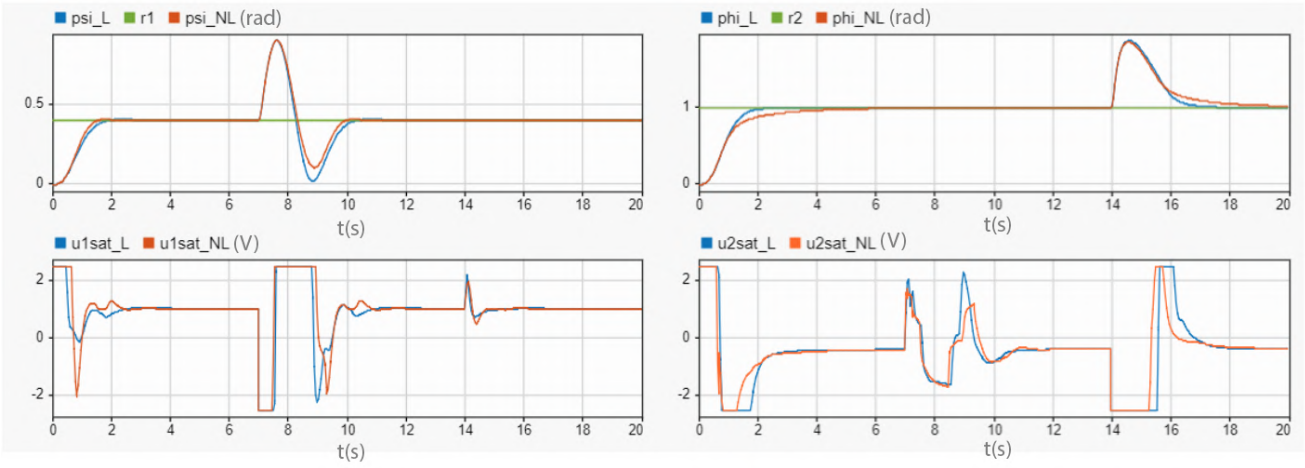


Figure 3.19: LADRC and ADRC Robustness for a bigger impulse

For a bigger value of the perturbation the NLADRC performs better in the vertical plan while it improved in the horizontal plan having now a similar behaviour to the LADRC, by increasing the value of the perturbation more, NLADRC would have better results, by having an error greater then 1, the combination of the non linear feedback seems to be more optimal, especially

for the proportional term that has $\alpha > 1$ which acts better for bigger errors, however we are already near the physical limitation of the horizontal axis around 2 rad/s, so we can say that the LADRC would act better in our case.

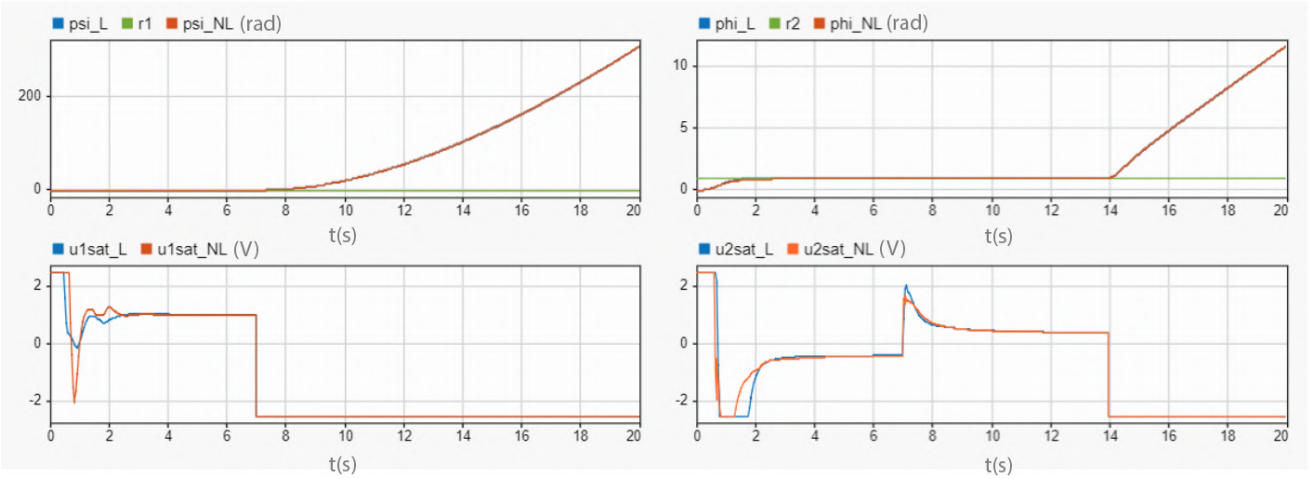


Figure 3.20: LADRC and ADRC Robustness for a constant disturbance

When applying a constant perturbation the system became unstable, this is due to the saturation of the control signal, as the perturbation persists the total disturbance increases and since it's now only partially compensated with a limitation at 2.5 the inertia of the system and the effect of the perturbation overcome the control signals and the system become unstable, this is one of the limitation of the ADRC for this class of system that will be addressed more in details in the next chapter.

Tables 3.2 and 3.3 summarize the results found throughout this whole comparative study:

		Conditions			
		Nominal positif	Nominal negatif	Small impulse	Big impulse
ADRC	Overshoot (rad)	0	0	0.47	0.55
	Response time (s)	1.75	1.5	0.5	2.5
LADRC	Overshoot (rad)	0	0	0.45	0.55
	Response time (s)	1.77	1.6	1	2.7

Table 3.2: Performance Characteristics of the Vertical Axis

		Nominal positif	Nominal negatif	small impulse	Big impulse
ADRC	Overshoot (rad)	0	0	0.4	0.8
	Response time (s)	5	4	2.5	3
LADRC	Overshoot (rad)	0	0	0.4	0.8
	Response time (s)	1.75	1.8	1.2	2

Table 3.3: Performance Characteristics of the Horizontal Axis

3.6 Conclusion

In this chapter, paramerter tuning were performed for both linear and nonlinear ADRC structures applied to the TRMS. Mostly through optimization, the TRMS model was subjected to tracking problems for different reference signals in order to design the best ESO and SFEC that guarantees the desired performance.

Before proceeding with the simulative analysis, the system's nonlinear mismatched model was adapted to its matched form so that the theory presented in Chapter 1 remains applicable to the TRMS' nonlinear model, and that the results obtained through simulations holds in an ADRC context.

PSO was presented then executed as the main optimization algorithm to tune both the controller and observer so that they achieve the optimal performance subject to the ITAE criterion. This algorithm was applied to all cases except for the LESO because finding an effective systematic tuning method is of the same importance as optimal observer design.

The chosen structure of the state observer was LO for the linear case and its $fal(\cdot)$ counterpart for the nonlinear case, and it was relatively easy to design and it showed promising results within the first design attemps. The implemented observer is high-gain observer whose gains were set at its performance limit so that the most optimal signal reconstruction was performed without useless gain augmentation.

As for the SFEC, PD2 was chosen as the controller structure for both the linear and nonlinear cases as well. PSO was performed on both of them and it showed optimal tracking performance in terms of convergence time and steady state error. A frequency and negative amplitude limits was then set for the tracking of sine and negative step references respectively, making a link between the convergence time and the reference signals' varying speed.

Finally, a comparative study was performed between LADRC and NL ADRC. The results showed that ADRC is better than LADRC in terms of state estimation, but the difference in performance is small that the LADRC was preferred than its nonlinear counterpart thanks to its simplicity. As for the tracking performance, ADRC was only better than LADRC in the case of the convergence time for a positive setpoint reference, but it still showed more control effort than LADRC with relatively the same performance. We can therefore say that LADRC is better than ADRC in terms of tracking performance, having close performance without forcing the control inputs too much.

For these reasons, LADRC was considered better than ADRC in our case, and it is chosen therefore as the control structure to be implemented in the experimental study of the following chapter.

Chapter 4

Experimental TRMS Stabilization through ADRC

Experiments are the best way to confirm the remarks and a conclusion obtained from theoretical studies, and linking any theory with its practical counterpart is vital in understanding engineering-related applications. That is why, we proceeded with a series of experiments on the real Twin Rotor MiMo System in order to stabilize it to the origin through Active Disturbance Rejection Control.

The parameters of the ADRC structure are obtained using simulations of its non-linear mathematical model and have been given in Chapter 3. In our case, we will limit ourselves to a stabilization problem using a LADRC structure on the real TRMS.

PS: Please note that, similarly to the note that was made in Chapter 3, the TD will not be used in this case because not only it is not relevant in a LADRC structure, but the real plant does not have fast dynamics and can therefore be given constant references without worrying about the set-point jump issue.

4.1 Pre-Experiment Configuration

Before proceeding to the different experiments, let us talk about some key instructions to do beforehand so that the data exported from the experiment reflects perfectly what happened.

First, we used Matlab 6.1 to create the ADRC structure as a Simulink file, which needed its settings adjusted. To run the program smoothly, open up the settings window of your Simulink file and then:

1. Set the numerical path as a ode1 fixed step of value 0.001
2. Deselect all the selected cases in the I/O Workspace tab
3. Verify that the target is `rtwinfeedback.tlc` in the Realtime Workshop tab.

Then, in the tools drag menu, set the control panel mode to "External". After that, we used the built-in Microsoft C++ compiler to build the Simulink file of the experiment into our workspace using the `Ctrl+B` shortcut.

Now that the software is all set up, it is ready to be transmitted to the hardware. To do that, the TRMS was held and maintained at $\psi = \varphi = 0\, rad$ and the Simulink file was connected to

the actual system by tapping at the following key:

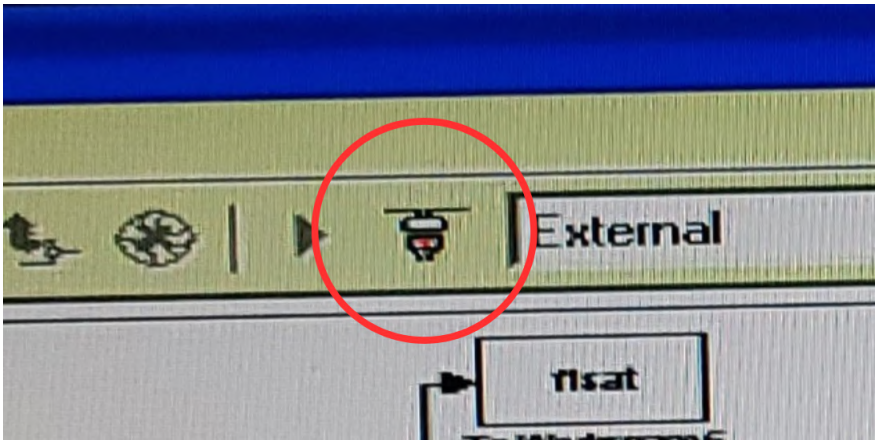


Figure 4.1: TRMS Connection button using Matlab 6.1

the reason that the connection was done in this specific position is that the sensor (i.e. position encoders) needed to be initialized at zero so that it best reflects the behavior of the physical plant. Now, the system can be let go of and the simulation is started by pressing the start button and clicking on the run command of the system simultaneously.

All that rest now is opening the scope and observing the closed-loop behavior. This process is the same and needs to be repeated for every single experiment, except for building the Simulink file which is done only when it is edited.

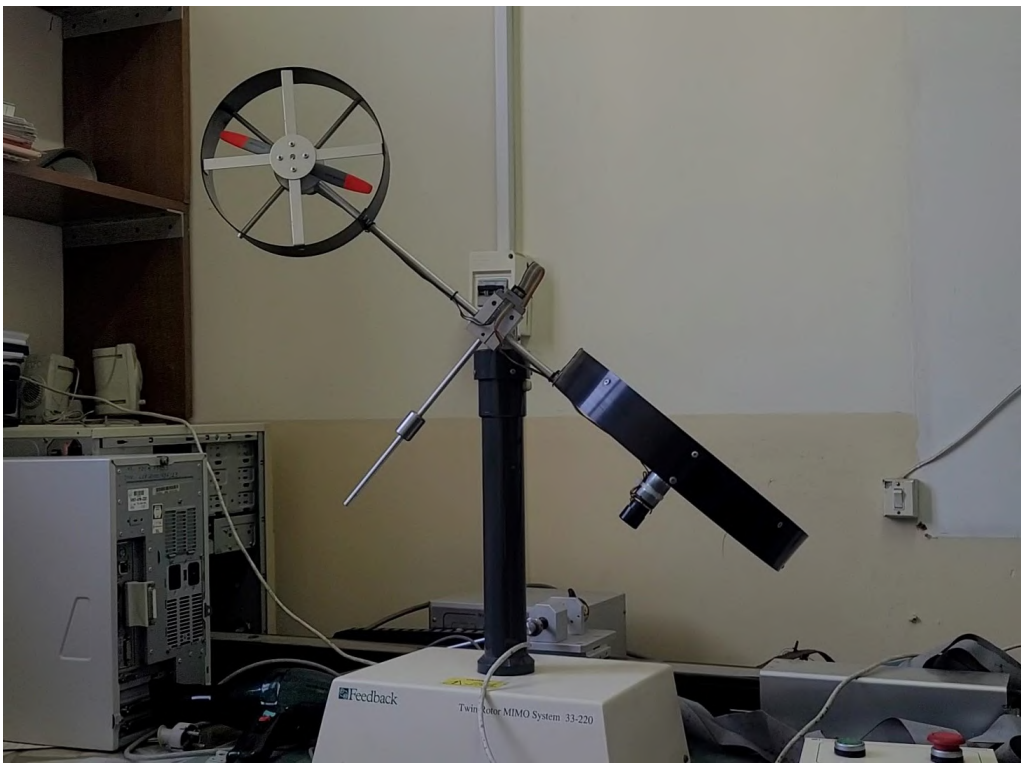


Figure 4.2: Real TRMS Experimentation

4.2 Experimental LESO Validation

Before going further into our study, it is as important to make sure that the designed LESO is producing high-quality estimates of these output signals to ensure that the SFEC is using accurate tracking errors while working on setting the closed loop dynamic. For that, the LESO is a LO whose gains were obtained in Section 3.3 of Chapter 3 as:

$$L = \begin{pmatrix} 78 & 2281 & 29640 & 144400 \end{pmatrix}^T$$

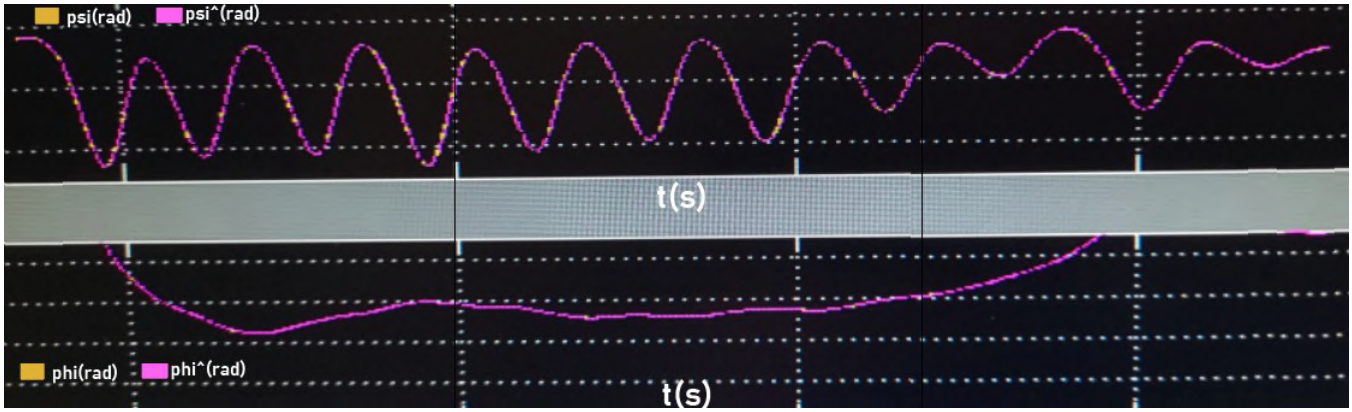


Figure 4.3: Real and Experimentally Reconstructed (top) Pitch and (bottom) Yaw Angles

The figure above shows that the estimated and actual values of the pitch and yaw angles, in a random closed loop experiment (that does not have to be optimal), are superposed to each others. Meaning that the signal reconstruction done by the ESO is very performative and guarantees high-quality reconstructions of the output signal, its derivatives, and the total disturbance.

4.3 Stabilization Experiment with LADRC

Now that the $\hat{f} \approx f$ condition is verified experimentally, the adrc compensation law (1.4) takes place and the TRMS is transformed into a chain of integrators. However, this can only be possible when the control input u is within the $\pm 2.5V$ allocated for the DC Motors, as it was adressed in Lemma 1.6. In our case, we found in Chapter 2 that the mathematical non-linear model is highly complex and highly interconnected, which will be more severe when the unmodeled dynamics are included in the real system.

Additionally, we need to make sure that the simulation-based tuning done in this chapter translates into a feasible and performative control structure when applied to the real TRMS helicopter lab simulator. From a control engineer's perspective, it can be seen that the linear and non-linear PD2 controller gains found in Sections 3.3 and 3.4 via PSO are relatively large and might damage the TRMS' DC Motors.

Because of these reasons, refining the controller's structure as well as it's parameters is needed to ensure that the total disturbances are fully compensated whatever their value is, and that the system's motors are not damaged due to excessive gains.

So while Chapter 3 confirmed the theoretical remarks made in Chapter 1 through simulations, Chapter 4 will check the validity and refine the ADRC parameters set in Chapter 3.

4.3.1 From PD2 to PID as a SFEC

While the ESO was kept the same, the SFEC was restructured from a PD2 into a PID controller. This might seem contradictory at first, because we just verified the high-quality performance of the ESO, which was meant to eliminate the integral error in some way. What follows will explain and experimentally prove the logic behind restructuring the controller.

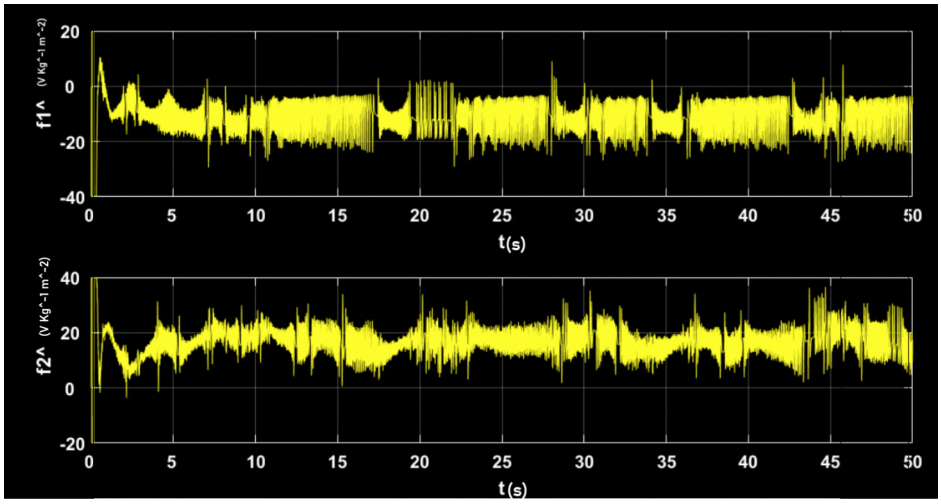


Figure 4.4: The estimated values of the total disturbances

Figure 4.4 shows the estimated total disturbances for an arbitrary open loop test. It can clearly be seen that the total disturbance is large in value for both the pitch and yaw subsystems, which was due mainly because of interconnection phenomenons between both axis. This is problematic because Theorem C and C of Appendix C stated that the system uncertainties need to be bounded for the ADRC stability.

A way around this issue is to set a limit on the value of the total disturbance that the control input can pre-compensate without saturation. Through empirical tests, this value was set at:

$$f_m = \pm 12.5\ V.kg^{-1}.m^{-2}$$

and the following figure shows the total disturbances of the pitch and yaw axis after saturation.

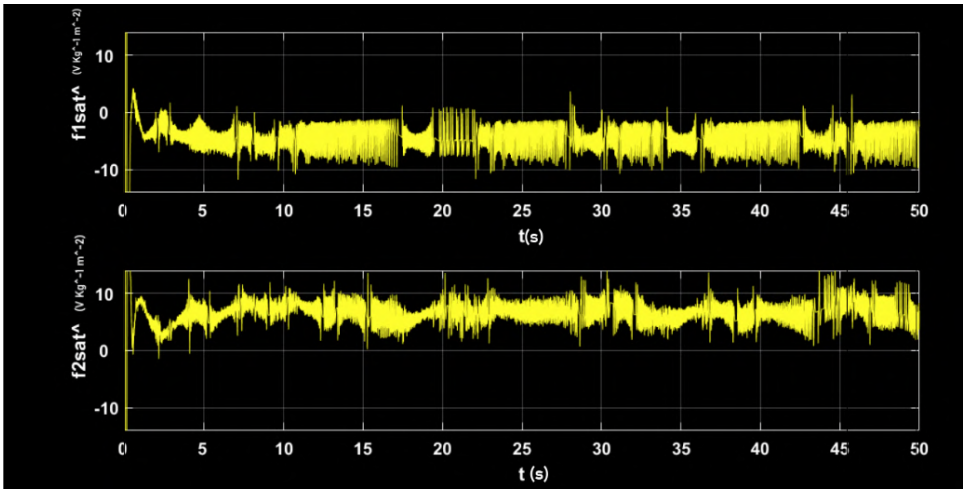


Figure 4.5: The estimated values of the total disturbances after saturation

However, the limited controller input signal is not enough to mitigate the total disturbance, and the non-compensated remaining total disturbances will still go through the system and

cause static error. For that reason, Lemma 1.6 does not hold anymore. We therefore propose a restruction of the SFEC by adding an integral action to remove the apparent steady state error. While it is true that the integral action reduces the stability margin of the closed loop, we will still proceed with it because we have the advantage of implementing it in an ADRC structure, which still keeps the observation's intrinsic robustness properties.

The new adrc structure consists then of a limited estimate of the total disturbance and a PID controller as the SFEC. The following Figure illustrates the new ADRC topology:

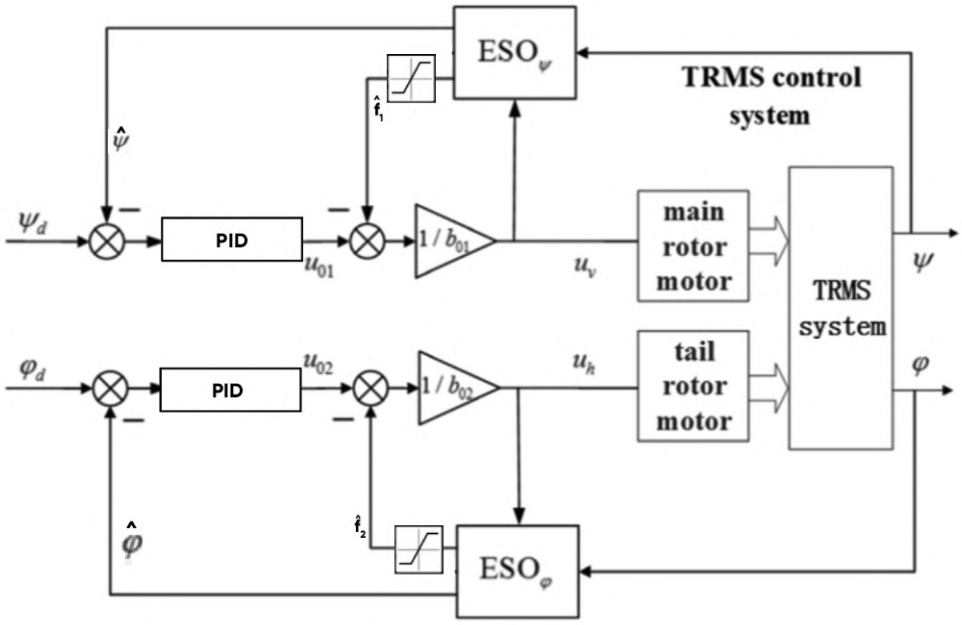
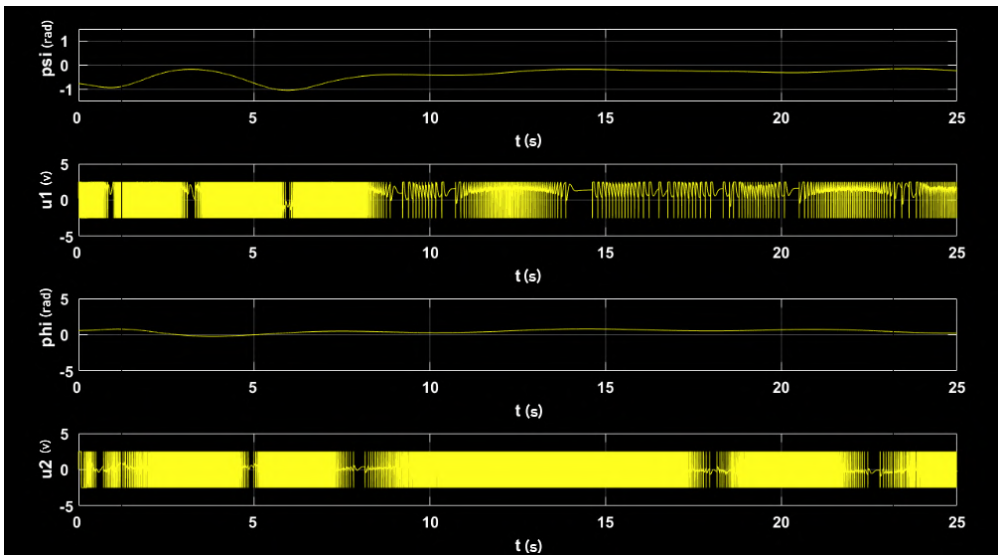


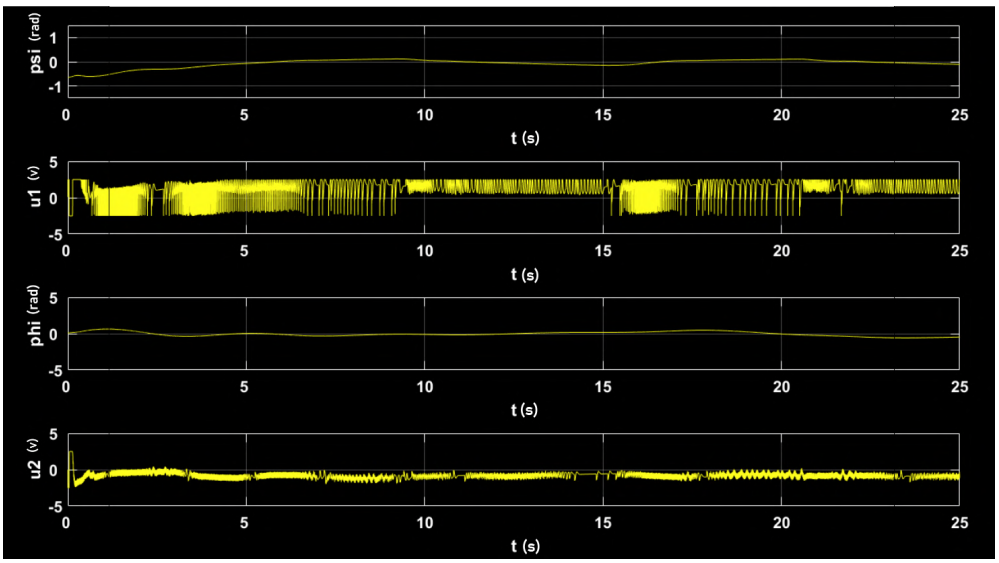
Figure 4.6: Modified ADRC Topology for Underactuated Systems of High Complexity

4.3.2 Tuning and testing the new controller

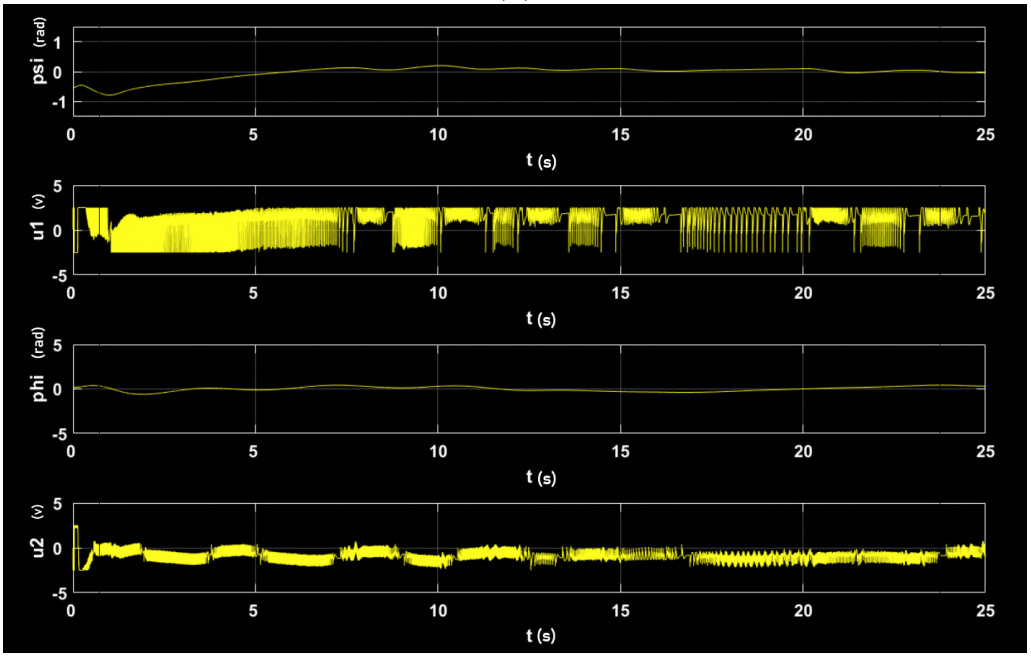
Since the issue that forced us to add an integral action in the control was not apparent in simulations. Empirical tuning is the only way to set the parameters of the new observer.



(a)



(b)



(c)

Figure 4.7: Nomnial Case Stabilization Problem for (a) $K_\psi = (0.15\ 2.5\ 1.5)^T$ and $K_\varphi = (0.005\ 1\ 0.01)^T$ (b) $K_\psi = (0.2\ 2\ 0.03)^T$ and $K_\varphi = (0.04\ 1\ 0.06)^T$ (c) $K_\psi = (0.2\ 2\ 0.03)^T$ and $K_\varphi = (6 \times 10^{-3}\ 1.5\ 0.06)^T$

Figure 4.7 (a) and (b) and (c) shows the output signals and control inputs for different values of the controller gains.

		Steady state error	Overshoot	Response time
a	Vertical axis	0.2	0	13
	Horizontal axis	1	2	2.5
b	Vertical axis	0.1	0	4.5
	Horizontal axis	0.1	2	2.5
c	Vertical axis	0.1	0	6
	Horizontal axis	0.75	2	3

As it can be seen from Table 4.3.2, the best gains out of the three are:

$$K_\psi = (0.2 \;\; 2 \;\; 0.03)^T \quad ; \quad K_\varphi = (6 \times 10^{-3} \;\; 1.5 \;\; 0.06)^T$$

which showed consistent zero-reference tracking at 6 and 3 seconds with a static error of 0.1 and 0.75 for the pitch and yaw angles respectively.

A robustness test was also performed on this modified ADRC Structure. An impulse external disturbance was modeled by a push in the vertical beam of the TRMS so that it disturbs the vertical subsystem directly and the horizontal subsystem indirectly (i.e. through the interconnection phenomenons).

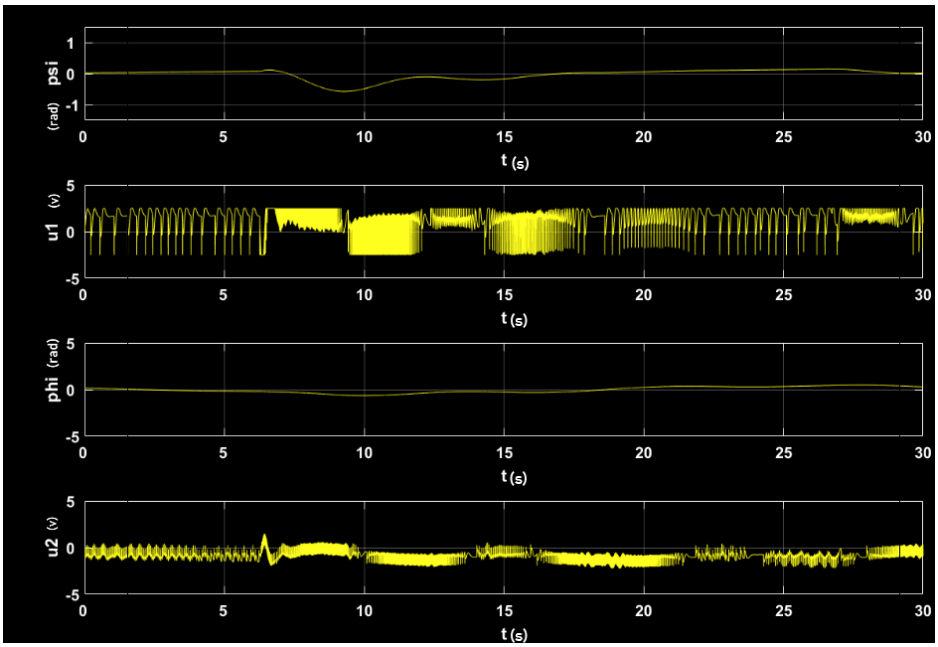


Figure 4.8: Experimental Robustness Test against External Disturbances

As it can be seen from Figure 4.8, the vertical subsystem came back to its reference position 5 seconds after the disturbance was performed. Also, there was no effect of the coupling phenomenon on the horizontal subsystem, as its controller compensated them. This means that robustness is guaranteed while maintaining the same tracking performance.

Since our goal was to develop a way around the issue of total disturbance's high values, we limited our performance analysis to a stabilization problem and our robustness analysis to a single external disturbance test. But that doesn't mean that the other types of references and disturbances are not as important. In fact, they should be carefully considered depending on the interest of the ongoing work.

4.4 Conclusion

In this chapter, we implemented an active disturbance rejection control structure to solve a stabilization problem on the twin rotor mimo sysrem via real life experiments. The parameters of the associated observer and controller were taken from the simulative results of the optimisation-based pole placement.

We found that these gains do not translate into an effective solution when projected to the real plant. This was due to the actuator's energy consumption being set at the max in the simulation-run optimisation, which called for a gain rescheduling on the ADRC's controller.

Empirical tuning allowed to set proper gains energy-wise since the simulative tuning did not reflect the constraints of the real system.

The second issue that we encountered was that the plant exhibied large total disturbances because of the inter-aerodynamics happening between the main and tail rotor. To solve this issue, a limit of the estimated values of the total disturbance was experimentally set for both the pitch and yaw according to the actuator's maximum capacity of disturbance rejection.

Following this limitation, the uncompensated total disturbances remains in the closed loop and consequently causes a static error. This a drawback of ADRC that was, to our best knowledge, never explored before, as it was always supposed that the total disturbances are bounded for the closed loop stability.

A remedy that we proposed was to restructure the PD2 controller into a PID controller in the LADRC scheme. By adding an integral action to the SFEC, the static error is erased and the tracking problem is solves.

At the end, a robustness test was performed to assess ADRC's intrinsic robustness property by modelising an impulse disturbance on the vertical beam of the TRMS. We found out that even when an integral action was added, ADRC still stood its ground as a good controller since the system recovered from the impulse came back to its reference position in no time.

General Conclusion

The work presented in this Master's thesis consisted on the study of a recent control approach, namely Active Disturbance Rejection Control (ADRC), and its implementation on the Twin Rotor MIMO System (TRMS) helicopter lab simulator.

While the Proportional Integral Derivative (PID) was the most used control algorithm, it still had several problems related to its robustness to external perturbations, modeling errors, and parametric variations. With the increase of system complexity, PID seemed to have reached its limits. Meanwhile, the control engineering community witnessed a major breakthrough with the development of observers and estimators and entered a new era. Nonlinear control techniques emerged as an alternative to the classical feedback error control, such as robust and predictive control. Despite apporting performance improvements, these new algorithms brought a lot of complications: they are either model-based approaches or they are based on the worst case scenario that make control relatively conservative. This control algorithms are either very sensitive to the choice of the model or non optimal in term of energy consumption.

It is in this context that J.Han proposed in 1989 a new paradigm combining the simplicity of classical control theory with the state feedback and the efficiency of modern control theory with the state observer : The Active Disturbance Rejection Control. In , a theoretical study of the ADRC is proposed, we discussed the major problem of the PID and the solution proposed by this new approach as well as the different linear and non linear components of the algorithm. The total disturbance observation and rejection was the main idea behind ADRC, lumping internal and external perturbations offered to the ADRC strong robustness properties. The second important contribution was the introduction of nonlinearities in the feedback loop, these gave the ADRC a certain versatility to attack control problems with different angles through the tuning of the $fal(\cdot)$ function or its more recent alternatives.

To confirm our theoretical conclusions in practice, we needed to implement the ADRC on a real system. The Twin Rotor MIMO System was an ideal candidate due to its characteristics. In chapter 2, we studied the system and presented a nonlinear model that was found to be unstable in the open loop, and to have strong coupling and input saturation. In chapter 3, we adapted the structure of the ADRC to the TRMS that was primarily not compatible because of its high complexity. Then we tuned the parameter of both LADRC and ADRC using Particle Swarm Optimization and pole placement methods. The comparative study in simulation of the two variant of the ADRC in different conditions resulted in a small difference, in most cases both approach had similar results and the less complexe LADRC was preferred.

Finally in chapter 4, LADRC was implemented in a real TRMS device for a stabilization experiment with and without disturbance. The first step was the adjustement of the parameters obtained in simulation, which was due to the fact that our mathematical model didn't reflect the real system with enough precision for tuned parameters of the ADRC's controller to work correctly. Indeed, despite its robust nature, ADRC needs a fair representation of the system for the parametrization phase, as tuning directly with the real plant presents many risks. We then

implemented the ADRC algorithm which was not successful at first. In fact, we encountered a major problem: because the limitation on the input signals, the total disturbance was not compensated fully and LADRC didn't perform as expected. After some additional refining, we successfully implemented a version of the ADRC with total disturbance saturation and an integral term with satisfactory performances and good disturbance rejection.

At the end of this work, we propose the following research perspectives :

- To further explore the performances of this new version –that we came up with– for systems with input saturation large values of the total disturbance.
- To find a systematic approach to tune the parameters of the modified ADRC version.
- To assure better parametric tuning in simulation without the need of an extreme precision of the model for highly complex systems.

Appendixes

A The Concept of Differential Flatness

Flatness is a unique property of controlled dynamic systems that offers several unique features that help simplify the process of system analysis as well as control design. This property is found in a large class of systems and should therefore be put to use by the math-inclined control engineers community. In [19], this concept was extensively detailed with many illustrative examples, but we will limit ourselves to the relevant bits only.

A.1 Definition of flatness

For simplicity reasons, we will restrict ourselves to finite-dimensional non-linear multi-variable systems. Consider the MiMo system:

$$\begin{cases} \dot{x} = F(x, u) \\ y = h(x, u) \end{cases} \tag{1}$$

where $x \in \Re^n$ is the state vector, $u \in \Re^m$ is the control inputs vector and $y \in \Re^p$ is the outputs vector. Let α, β, and γ be finite indices. System (1) is said to be differentially flat if there exists a vector

$$z = \phi(x, \dot{x}, \ldots, x^{(\gamma)}) \in \Re^m \tag{2}$$

called the set of flat outputs, equal in size to the number of system inputs, whose components are differentially independent (i.e. they do not satisfy differential equations by themselves), and –most importantly– such that there exist functions ψ and χ for which:

$$x = \psi(z, \dot{z}, \ldots, z^{(\alpha)}) \quad ; \quad u = \chi(z, \dot{z}, \ldots, z^{(\beta)}) \tag{3}$$

This means that a system is differentially flat if it contains a set of flat outputs, in which the control input and state signals of the system are differentially parametrized in terms of these flat outputs and a finite number of its derivatives. In general, the flat output of a system is not unique, and a scalar multiple of a flat output is also a flat output.

A.2 Finding the flat outputs

In the case of ADRC, the considered system was supposedly flat (see Section 1.2 of Chapter 1), because just from the differential equation describing it, or more simply from its state space representation, the flat outputs can be taken as the actual outputs and this definition would still hold. But in a more formal way, [19] presented an approach to determine whether a system is flat or not, and how to find its flat outputs in the former case.

Even though it was stated that it is more elaborate and complicated to give a systematic approach for the case of non-linear MiMo systems, we have the privilege once again to work with the SiSo case for each channel of our MiMo system thanks to ADRC's concept of total disturbances. Consider the SiSo non-linear time-invariant input-affine system:

$$\begin{cases} \dot{x} = F(x) + g(x)u \\ y = h(x) = x_1 \end{cases} \tag{4}$$

First of all, this system is said to be feedback linearizable (i.e. flat) at the point $x \in \Re^n$ if the following requirements are met:

a. The distribution $[g \;\; Ad_F g \;\; \dots \;\; Ad_F^{n-2} g]$ is involutive in a neighborhood containing x

b. The set of vectors $\{g \,;\, Ad_F g \,;\, \dots \,;\, Ad_F^{n-1} g\}$ is linearily independant at the point x

where $Ad_F^i g(x)$ represents the i-th iterated Lie Bracket (see Definition 2.4.4), and:
A distribution is called involutive if its vector subspace $\mathcal{D}$ is a Lie subalgebra i.e. for any two vector fields X and Y of $\mathcal{D}$, the Lie product $[X, Y]$ of these vectors also belongs in this subspace.
Once we verify that a scalar system is flat, we need to find the expression of its flat output. Define the invertible matrix $\kappa(x)$ as follows:

$$\kappa(x) = [g \;\; Ad_F g \;\; \dots \;\; Ad_F^{n-1} g] \tag{5}$$

and the flat output z is determined, up to a smooth factor function $\alpha(x) \neq 0 \;\; \forall x$, via integration of its gradient, which is given by:

$$\frac{\partial z(x)}{\partial x^T} = \alpha(x)[0 \;\; 0 \;\; \dots \;\; 1]\kappa^{-1}(x) \tag{6}$$

such that: $dim([0 \;\; 0 \;\; \dots \;\; 1]) = 1 \times n$

Now that the theory is set up, let's apply it to our case. To make it fast, consider the smooth 2nd order scalar system:

$$\dot{x} = \begin{pmatrix} x_2 \\ f(y,t) \end{pmatrix} + \begin{pmatrix} 0 \\ b \end{pmatrix} u = F(x) + g(x)u \tag{7}$$

where f is the total disturbance and b is the input gain. The vector field $g(x)$ is in itself a one-dimensional distribution, which is trivially involutive. The set of vectors:

$$\{g \,;\, Ad_F g\} = \{ \begin{pmatrix} 0 \\ b \end{pmatrix} ; \begin{pmatrix} -b \\ -b\dfrac{\partial f}{\partial x_2} \end{pmatrix} \} \tag{8}$$

is a linearly independent set $\forall x \in \Re^2$. So both conditions are satisfied, and the system is therefore flat. The gradient of the flat output, modulo $\alpha(x)$, is:

$$\frac{\partial z(x)}{\partial x^T} = \alpha(x)[0 \;\; 1] \begin{pmatrix} 0 & -b \\ b & -bf_{x_2} \end{pmatrix}^{-1} = \frac{\alpha(x)}{b}[-1 \;\; 0] \tag{9}$$

the flat output $z(x)$ is then defined via integration:

$$\begin{cases} z_{x_1} = -\dfrac{\alpha(x)}{b} \\ z_{x_2} = 0 \end{cases} \tag{10}$$

for our own convenience, if we take $\alpha(x) = -b$ then:

$$z(x) = x_1 \tag{11}$$

meaning that the flat output is just the plant's actual output, which is the same result intuitive result suggested at the start of this Section.

A.3 Advantages and disadvantages of flatness

Flatness is a unique property of controlled dynamic systems, which offers control engineers the ability to design controllers that have more advantages than inconveniences, but it is good to mention both of them.

Some complaints have been made regarding this property. The most common one is the lack of systematic methods to find the flat output of more "irregular" systems and even the complexity and restrictions manifested in the already established methods. It is also a fact that flatness may be assessed by inspection in the vast majority of control engineering problems. Even when the flat output could be determined, it cannot be measured in some practical cases. Flat outputs usually enjoy a distinctive physical nature in most real-life applications. Perhaps ADRC comes to the rescue this time too, as it simplifies the system's mathematical model to a degree where it becomes easy to analytically find the set of flat outputs. However, the issue of non-measurability remains unsolved.

Regarding its advantages, it is only fair to say that flatness can be seen as a property tailored around ADRC. The combination of this structural characteristic with this control framework elevates the robustness of simple controllers, even for some highly non-linear systems. In fact, any linear controllable system whose flat output coincides with the measurable output of the plant can achieve stabilization and/or tracking objectives using a linear well-tuned stable filter alone. To add on it, robust control of underactuated systems using linear controllers can be achieved for rather unrestricted trajectories using the odd-even derivatives property of the incremental flat outputs of its controllable tangent linearization models, and adding ADRC to the mix gives it robustness against any excitation that the controlled tracking may cause to the neglected non-linearities (through the process of linearization).

Another advantage of flatness is the fact that it is easy to assess even if the system in question is non-affine in the control. Additionally, flatness examines the feasibility of controllers in the context of uncertainties and mitigates the effects of inputs and state restrictions.

In general, flatness-based ADRC opens the door for the synthesis of non-complex robust controllers of a large class of problematic systems, including those affected by saturation; uncertainties; nonlinearities ...etc. This combination also disregards most inconveniences about each of its components and is therefore very important to put it to use in making breakthrough advancements whether it be in the maths or practical side of control engineering.

B Differential Geometry-Based ADRC

Differential geometry is a versatile concept that is backed by the Lie tools and founded upon the idea of changing the way we see a system. This concept helps simplify the system analysis and control design procedures by doing it on the basis of a simpler representation of the system, and then applying it to the original form of the system. If you see the core concept of ADRC, differential geometry represents it best.

B.1 Theoritical background

Differential Geometry is motivated by the following question: why do we try to control a complex non-linear system in its original basis (x, u) when we can probably do it in a basis (z, ν) where the state space model of the system in question becomes fully or partially linear?

The proposed change of basis is expressed as:

$$\begin{cases} z = \phi(x) \\ u = \alpha_1(x) + \alpha_2(x)\nu \end{cases} \tag{12}$$

where $\phi(x)$ must be bijective to allow for passing between the two bases. Loosely speaking, $\phi(x)$ is bijective if $\left|\frac{\partial \phi}{\partial x}\right| \neq 0 \ \forall x$.

The new state space representation is:

$$\Sigma_{(z,\nu)} : \left\{ \dot{z} = \frac{\partial \phi}{\partial x}\dot{x} = \frac{\partial \phi}{\partial x} F\Big(\phi^{-1}(z), \alpha_1(\phi^{-1}(z)) + \alpha_2(\phi^{-1}(z))\nu\Big) \right. \tag{13}$$

and the task is to find $\phi(x)$, $\alpha_1(x)$, and $\alpha_2(x)$ so that (13) is fully or partially linear without resorting to linearization and, therefore, avoiding its limitations in terms of locality, controllability and observability.

Using the iterative property of the Lie derivative as well as the Lie product and tying it up with the concept of relative degrees, one can find that, given a non-linear system:

$$\begin{cases} \dot{x} = F(x, u) \\ y = h(x) \end{cases} \tag{14}$$

its relative degree, as long as it is > 1, satisfies:

$$\begin{cases} L_g h(x) = L_g L_F h(x) = \ldots = L_g L_F^{r-2} h(x) = 0 \\ L_g L_f^{r-1} h(x) \neq 0 \end{cases} \tag{15}$$

where $L_g^{(i)} h(x)$ is the i-th Lie derivative of $h(x)$ with respect to $g(x)$ (see Definition 2.4.4). Using this iterative property, one can find through consecutive derivations of y that:

$$\begin{cases} y^{(i)} = L_F^i h(x) \quad \forall i = \{0 : r-1\} \\ y^{(r)} = L_F^r h(x) + L_g L_F^{r-1} u \end{cases} \tag{16}$$

Following (16), there are two study cases that need to be considered.

B.1.1 Case 01: Full linearity

Consider the non-linear system (14) where the relative degree of the output $y(t)$ is equal to the dimension of the state vector x:

$$r = size(x) = n \tag{17}$$

then the bijective change of basis:

$$\begin{cases} z = \phi(x) = \begin{pmatrix} h(x) \\ L_F h(x) \\ \vdots \\ L_F^{n-1} h(x) \end{pmatrix} \\ \\ u = \Big(L_g L_F^{n-1} h(x)\Big)^{-1} \Big(- L_F^n h(x) + \nu \Big) \end{cases} \tag{18}$$

always give the hidden fully linear form:

$$\Sigma_{(z,\nu)} : \begin{cases} \dot{z} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} z + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \nu \\ y = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix} z \end{cases} \quad ; \quad Y(s) = \frac{1}{s^n} \nu(s) \tag{19}$$

which is called the Normal Form of the non-linear system (14), which is basically a series of n-integrators from the input $u(t)$ to the output $y(t)$. This illustrates best that this approach focuses rather on the input-output connection than the complexities of that same connection, by simply changing the way we look at the system.

B.1.2 Case 01: Partial linearity

In a lot of control engineering applications, the relative degree of the plant's output is smaller than the dimension of this same plant:

$$r < size(x) = n \tag{20}$$

In this case, the same change of basis takes place for the control input, and up to the r-th order for the state vector, following:

$$\begin{cases} z = \phi(x) = \begin{pmatrix} h(x) \\ L_F h(x) \\ \vdots \\ L_F^{r-1} h(x) \\ \lambda_1(x) \\ \vdots \\ \lambda_{n-r}(x) \end{pmatrix} \\ \\ u = \Big(L_g L_F^{r-1} h(x)\Big)^{-1} \Big(- L_F^r h(x) + \nu \Big) \end{cases} \tag{21}$$

where $< \lambda_i >_i$ are arbitrary non-linear functions such that $\phi(x)$ is a diffeomorphism.

In the same manner, the normal form of the system is divided into a fully linear part $\varepsilon_1 \in \Re^r$ and a non-linear part $\varepsilon_2 \in \Re^{(n-r)}$:

$$\Sigma_{(z,\nu)} : \begin{cases} \dot{\varepsilon}_1 = A\varepsilon_1 + B\nu \\ \dot{\varepsilon}_2 = \psi(\varepsilon_1, \varepsilon_2, \nu) \\ y = C\varepsilon_1 \end{cases} \quad ; \quad \begin{matrix} \varepsilon_1 = [z_1 \ \ \dots \ \ z_r] \\ \varepsilon_2 = [z_{r+1} \ \ \dots \ \ z_{n-r}] \end{matrix} \tag{22}$$

where the linear relation between the input and the output remains only tied to the relative degree of the output:

$$Y(s) = \frac{1}{s^r}\nu(s) \tag{23}$$

A few remarks need to be issued regarding this case:

- Since ε_2 appears neither in the output nor in the state that appears in the output (namely, ε_1), we say that it is the non-observable state part. Therefore, it should be stable for the normal form to be detectable.
- Since ε_2 is non-observable, it can't be controllable through observer-based feedback, but that doesn't mean that the control signal does not reach this part. Lyapunov stability analysis is therefore needed.
- It was proven that the poles of the hidden form consist of r poles coming from ε_1 and $(n-r)$ poles coming from the zeros of ε_2. Therefore, the non-linear part shouldn't have zeros that are of positive real part for the hidden form to be stable. We are thus talking about the "zero-dynamics" of the system.
- An interesting approach to do input/output-based control is to add dynamic extensions to the linear part in the form of the $(n-k)$ missing integrators. But the new control signal becomes $u^{(n-k)}(t)$ rather than $u(t)$, and a non-linear differential equation needs to be solved every time we want to apply the control signal to the system.

B.2 Normal Form and ADRC

Now that we've given theoretical background on Differential Geometry, let's illustrate how Active Disturbance Rejection Control was inspired by it.

In fact, if you look at the state vector of the state space model (1.3) and the ADRC compensation law (1.4), it is the same bijective change of basis defined in (18). Through identification between (1.9) and (16):

$$y^{(r)} = L_F^r h(x) + L_g L_F^{r-1} u = f(y,t) + bu \tag{24}$$

therefore:

$$L_F^r h(x) = f(y,t) \quad ; \quad L_g L_F^{r-1} = b \tag{25}$$

replacing in (18):

$$u = \Big(L_g L_F^{r-1} h(x)\Big)^{-1}\Big(- L_F^r h(x) + \nu\Big) = b^{-1}(-f(y,t) + \nu) \tag{26}$$

which is the same ADRC compensation law (1.4) except the pseudo-control input is just the control input expressed in the new basis. One can also verify these results using numerical application.

Additionally, from (18) and (16):

$$\phi(x) = \begin{pmatrix} h(x) \\ L_F h(x) \\ \vdots \\ L_F^{r-1} h(x) \end{pmatrix} = \begin{pmatrix} y \\ \dot{y} \\ \vdots \\ y^{(r-1)} \end{pmatrix} \tag{27}$$

which is the same state vector of (1.3). This means that the ADRC canonical form is a change of basis using differential geometry.

Of course, we considered the case where the normal form is totally linear because we applied the ADRC approach to the system described in (1.2). [38] tackled the presence of zero dynamics for an ESO-based ADRC configuration. More importantly, differential geometry is known for its sensitivity to parametric variations, but that is not the case in the context of ADRC. In fact, the change of basis used in the context of ADRC is independent of the model of the plant because 1) b can handle approximation errors up to a $\pm 50\%$ range and 2) $f(y,t)$ is evaluated by the ESO and not through the Lie derivatives i.e. independently of the system's mathematical model.

C Stability Analysis of ADRC

J.Han designed ADRC as an experimental approach, he stated in [1] that:

> *ADRC is the result of years of investigation, largely performed experimentally in computer simulations with the scientific spirit of daring imaginations, painstaking observations careful generalization and abstraction, and truthful verifications of principles in real-world applications. It may help the newcomers to ADRC greatly if one abandons the initial How can this be right? attitude and, instead, run a few simulations of the proposed solutions and observe the results. Perhaps the facts, or data, are more convincing than mere articulation of ideas.*

However, considerable works in exploring the theory behind ADRC has been made. So it's only natural to tackle some of it here, and most importantly, the stability analysis. Z.Gao presented in [39] an analytical study of the stability of a LADRC scheme based off Lyapunov stability analysis (see Definition 2.4.4.1). He proposed the following theorem:
A LADRC design yields a BiBo stable closed-loop system if the linear observer itself and the state feedback error control law are stable, respectively. To demonstrate this result, let us start with the LESO. Consider the linear system:

$$\begin{cases} \dot{x} = Ax + Bu + Eh \\ y = Cx \end{cases} \tag{28}$$

where h is scalar input disturbances. The corresponding LESO is:

$$\left\{ \dot{z} = Az + Bu + L(y - \hat{y}) \right. \tag{29}$$

If $e = x - z$ is the observation error then its dynamics are defined by:

$$\dot{e} = (A - LC)e + Eh = A_{CL}e + d \tag{30}$$

Assuming that A_{CL} is Hurwitz (i.e. the observation error dynamics are stable), we will find the condition that assures Lyapunov stability of the –observation– dynamics that this matrix describes. Let the candidate Lyapunov PDF:

$$V = e^T P e \tag{31}$$

where P is the solution of the Lyapunov equation

$$A_{CL}^T P + P A_{CL} = -Q \tag{32}$$

and Q is a positive definite matrix. The derivative of the candidate Lyapunov function is:

$$\dot{V} = -e^T Q e + 2 d^T P e = -\left(e^T Q^{\frac{1}{2}} - d^T P Q^{-\frac{1}{2}}\right)\left(e^T Q^{\frac{1}{2}} - d^T P Q^{-\frac{1}{2}}\right)^T + \left(d^T P Q^{-\frac{1}{2}}\right)\left(d^T P Q^{-\frac{1}{2}}\right)^T \tag{33}$$

which means that $\dot{V} < 0$ if:

$$||e^T Q^{\frac{1}{2}} - d^T P Q^{-\frac{1}{2}}||_2 > ||d^T P Q^{-\frac{1}{2}}||_2 \tag{34}$$

or equivalently, and for $Q = I_{n+1}$:

$$||e||_2 > 2||Pd||_2 \tag{35}$$

This means that $||e||_2$ decreases (i.e. $\dot{V} < 0$) and is, therefore, bounded if it satisfies (35), which can only be possible if d is bounded. This result provokes:
Assuming that the observer gains $< L_i >_i$ is chosen so that A_{CL} is Hurwitz, the LESO error e is bounded for any bounded input disturbance h.

A generalization of this Lemma to the case of dynamic systems is written as:
Consider the dynamic system:

$$\dot{\eta} = M\eta + g(\eta) \tag{36}$$

where $\eta \in \Re^n$ and $M \in \Re^{n \times n}$. The state η is bounded if M is Hurwitz and $g(\eta)$ is bounded.

Combining Lemma C and C, the boundedness of the LESO error is proven, the remaining task now is to prove that $\bar{x} = [y \ \dot{y}]^T$ is bounded. Since the external disturbances are pre-compensated with all other sorts of disturbances in 1.4, the remaining system is free of disturbances and can enjoy stable dynamics so long that the SFEC gains are chosen so that the closed-loop matrix is Hurwitz. Hence Theorem C is established.

The nonlinear ESO using $fal(\cdot)$ functions and the corresponding ESO-based output feedback control were designed for nonlinear systems with mismatched uncertainties, and the convergence of the closed-loop systems was proved in [40].

The convergence of the closed-loop system for mismatched system that have the same form as the one we got in Chapter 3 is a special case of Theorem 2.1 of [41], which is summarized from [42] as the following theorem:

Suppose that there exists a positive constant C such that $\|x(0)\| \leq C$,
$\sup_{t \geq 0} \|(w(t), \dot{w}(t), \ddot{w}(t))\| \leq C$, and $\sup_{t \geq 0} |r_i(t)| \leq C$ for all $i = 1, 2, 3$.

Then, the closed-loop system has the following properties:

(i) The closed-loop state $x(t)$ is bounded: $\|x(t)\| \leq \gamma$ for all $t \geq 0$, where γ is an ϵ-independent positive constant.

(ii) The output $y(t)$ of system tracks practically the reference signal $r(t)$ in the sense that: For any $\sigma > 0$, there exists a constant $\epsilon^* > 0$ such that for any $\epsilon \in (0, \epsilon^*)$,

$$|y(t) - r(t)| \leq \sigma \quad \text{uniformly in } t \in [t_\epsilon, \infty),$$

where $t_\epsilon > 0$ is an ϵ-dependent constant. In particular,

$$\limsup_{t\to\infty} |y(t) - r(t)| \leq \sigma.$$

Bibliography

[1] Jingqing Han. From pid to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*, 56(3):900–906, 2009.

[2] Zhiqiang Gao, Yi Huang, and Jingqing Han. An alternative paradigm for control system design. volume 5, pages 4578 – 4585 vol.5, 02 2001.

[3] Y Qin, J Yang, Y Wang, and F Zhou. *Discrete ADRC method based on improved fal function and its application*. europepmc.org, 2023.

[4] LL Xie and L Guo. How much uncertainty can be dealt with by feedback? *IEEE Transactions on Automatic Control*, 2000.

[5] Z Gao. On the centrality of disturbance rejection in automatic control. *ISA transactions*, 2014.

[6] Q Zheng, LQ Gaol, and Z Gao. On stability analysis of active disturbance rejection control for nonlinear time-varying plants with unknown dynamics. *2007 46th IEEE conference on Decision and Control*, 2007.

[7] ZL Zhao and BZ Guo. On convergence of nonlinear active disturbance rejection control for siso nonlinear systems. *Journal of Dynamical and Control Systems*, 2016.

[8] Z Pu, R Yuan, J Yi, and X Tan. A class of adaptive extended state observers for nonlinear disturbed systems. *IEEE Transactions on Industrial Electronics*, 2015.

[9] S Xingling and W Honglun. Back-stepping active disturbance rejection control design for integrated missile guidance and control system via reduced-order eso. *ISA transactions*, 2015.

[10] A Haruna, Z Mohamed, AM Abdullahi, and M Basri. A review of control algorithms for twin rotor systems. *Applications of Modelling and Simulation*, 2023.

[11] J. Han. Linearity and nonlinearity in feedback systems. *Control and Decision*, 1988.

[12] N Wiener. The operational calculus. *Mathematische Annalen*, 1926.

[13] J Han. A new type of controller: Nlpid. *Control and Decision*, 1994.

[14] ZW Chen, ZZ Zhang, and YJ Cao. Fal function improvement of adrc and its application in quadrotor aircraft attitude control. *Control and Decision*, 2018.

[15] Y Wenqi, LU Jianhua, J Xu, and Y WANG. Design of quadrotor attitude active disturbance rejection controller based on improved eso. *Systems Engineering & Electronics*, 2022.

[16] Gu Y Liu D Zhou Z, Yang J. Horizontal thrust control of magnetic suspension platform based on active disturbance rejection controller. *Journal of Mechanical Engineering*, 2008.

[17] B LIU, M LI, and J YANG. Attitude control of quadrotor aircraft based on improved adrc. *Command control and simulation*, 2021.

[18] M Przybyła, M Kordasz, R Madoński, and P. HERMAN. Active disturbance rejection control of a 2dof manipulator with significant modeling uncertainty. *Bulletin of the Polish Academic of Science*, 2012.

[19] H Sira-Ramírez, A Luviano-Juárez, M Ramírez-Neria, and Eric William Zurita-Bustamante. *Active disturbance rejection control of dynamic systems: a flatness based approach*. Butterworth-Heinemann, 2017.

[20] Z Gao. Scaling and bandwidth-parameterization based controller tuning. *ACC*, 2003.

[21] D Zhang, X Yao, and Q Wu. Parameter tuning of modified active disturbance rejection control based on the particle swarm optimization algorithm for high-order system. *2016 IEEE International Conference*, 2016.

[22] Feedback Instruments Ltd. Twin rotor mimo system control experiment, 1998. 33-949s.

[23] Chin-Wang Tao, Jin-Shiuh Taur, Yeong-Hwa Chang, and Chia-Wen Chang. A novel fuzzy-sliding and fuzzy-integral-sliding controller for the twin-rotor multi-input–multi-output system. *IEEE Transactions on Fuzzy Systems*, 18:893 – 905, 10 2010.

[24] Rafet Tolga Cankurt. Modeling and control of a twin rotor system. Master of science thesis, Scuola di Ingegneria Industriale e dell'Informazione, Politecnico di Milano, July 2019.

[25] L. Belmonte, R. Morales, Antonio Fernández-Caballero, and J. Somolinos. A tandem active disturbance rejection control for a laboratory helicopter with variable-speed rotors. *IEEE Transactions on Industrial Electronics*, 2016.

[26] Sherif Abdelmaksoud, Musa Mailah, and Ayman Abdallah. Practical real-time implementation of a disturbance rejection control scheme for a twin-rotor helicopter system using intelligent active force control. *IEEE Access*, 9:4886 – 4901, 12 2020.

[27] Yang Xiaoyan, Jianwei Cui, Dazhong Lao, Donghai Li, and Junhui Chen. Input shaping enhanced active disturbance rejection control for a twin rotor multi-input multi-output system (trms). *ISA transactions*, 62, 02 2016.

[28] L. Belmonte, R. Morales, Antonio Fernández-Caballero, and J. Somolinos. *Nonlinear Cascade-Based Control for a Twin Rotor MIMO System*. 10 2016.

[29] Azamat Tastemirov, Andrea Lecchini-Visintini, and Rafael Morales-Viviescas. Complete dynamic model of the twin rotor mimo system (trms) with experimental validation. *Control Engineering Practice*, 66:89–98, 09 2017.

[30] jatin kumar Pradhan and Arun Ghosh. Design and implementation of decoupled compensation for a twin rotor multiple-input and multiple-output system. *Control Theory Applications, IET*, 7:282–289, 01 2013.

[31] Kamel Kadri, Fares Boudjema, Yasser Bouzid, Badis Ouahab, and Hassane Draris. A linear active disturbance rejection control based pso to control twin rotor mimo system with experimental validation. pages 1–7, 11 2023.

[32] ZH Wu, HC Zhou, BZ Guo, and F Deng. Review and new theoretical perspectives on active disturbance rejection control for uncertain finite-dimensional and infinite-dimensional systems. *Nonlinear Dynamics*, 2020.

[33] Kennedy JE and Eberhart. In *Particle Swarm Optimization*, volume 4, 01 1995.

[34] Maurice Clerc and James Kennedy. Kennedy, j.: The particle swarm: Explosion, stability and convergence in a multi-dimensional complex space. ieee trans. on evolutionary computation 6, 58-73. *Evolutionary Computation, IEEE Transactions on*, 6:58 – 73, 03 2002.

[35] Stephen Joseph, Emmanuel Dada, Afeez Abidemi, David Oyewola, and Ban Mohammed. Metaheuristic algorithms for pid controller parameters tuning: Review, approaches and open problems. *Heliyon*, 8:e09399, 05 2022.

[36] X Yang and Y Huang. Capabilities of extended state observer for estimating uncertainties. *2009 American control conference*, 2009.

[37] Y Zhang, Z Zhao, T Lu, L Yuan, W Xu, and J Zhu. A comparative study of luenberger observer, sliding mode observer and extended kalman filter for sensorless vector control of induction motor drives. *2009 IEEE Energy Conversion Congress and Exposition, ECCE*, 2009.

[38] T Jiang, C Huang, and L Guo. Control of uncertain nonlinear systems based on observers and estimators. *Automatica*, 2015.

[39] Zhiqiang Gao. Active disturbance rejection control: A paradigm shift in feedback control system design. volume 2006, page 7 pp., 07 2006.

[40] Zhiliang Zhao and Bao-Zhu Guo. A novel extended state observer for output tracking of mimo systems with mismatched uncertainty. *IEEE Transactions on Automatic Control*, 63:211–218, 12 2017.

[41] Bao-Zhu Guo and Ze-Hao Wu. Output tracking for a class of nonlinear systems with mismatched uncertainties by active disturbance rejection control. *Systems Control Letters*, 100:21–31, 02 2017.

[42] Ze-Hao Wu, Hua-Cheng Zhou, Bao-Zhu Guo, and Feiqi Deng. Review and new theoretical perspectives on active disturbance rejection control for uncertain finite-dimensional and infinite-dimensional systems. *Nonlinear Dynamics*, 101, 07 2020.