ADVANCED CONTROL OF A NON-LINEAR SYSTEM

CONTROL OF DYNAMIC SYSTEMS :

APPLICATIONS IN MOBILE ROBOTICS AND AEROSPACE

MASTER IN SMART AEROSPACE AND AUTONOMOUS SYSTEMS

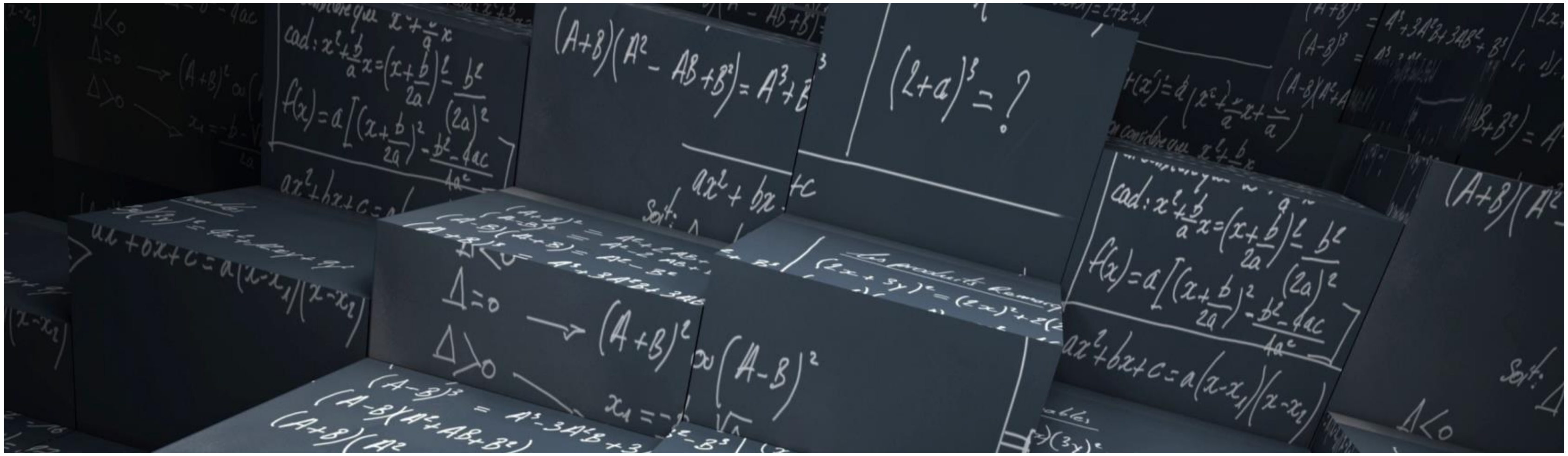
POLITECHNIKA POZNAŃSKA - POLAND

AUTEURS / AUTHORS

JIOKENG I TOKO JEAN JUNIOR MALDINI

FRIKRASSELASSIE ESHETU SEID

TABLE OF CONTENTS



INTRODUCTION

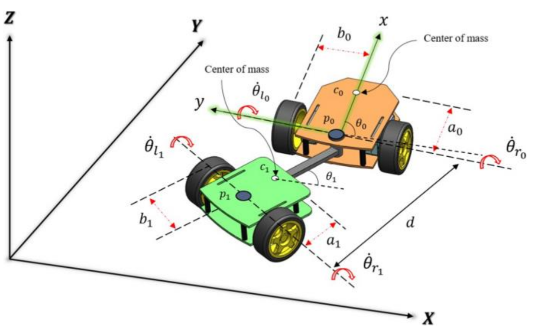
INTRODUCTION

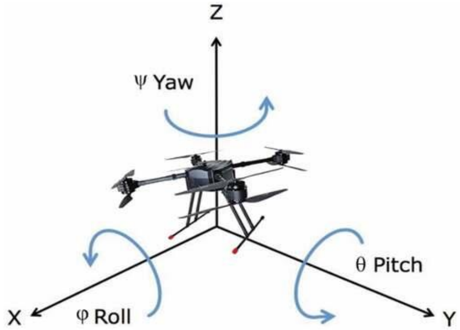
- In fields like robotics, aeronautics, and aerospace, many systems are difficult to control because they react unpredictably. For example, a drone may react suddenly in the presence of wind, and an autonomous robot may be destabilized by irregular terrain. This project explores a method called **linearization**, which simplifies these systems to make them more predictable and easier to control.

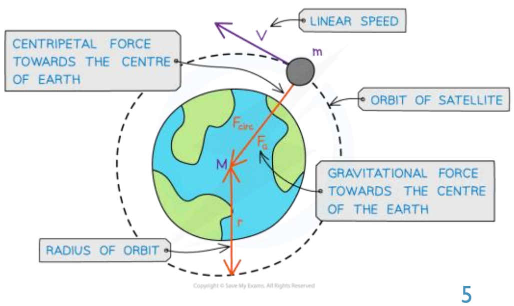
UNDERSTANDING THE NON-LINEAR SYSTEM

We studied a complex dynamic system model, represented by a mathematical equation involving several variables (the system's internal values). This model includes non-linear effects, such as a sine function, making its behaviour difficult to predict. These non-linear effects often appear in:

Mobile Robots	Drones	Satellites
Where interactions with the ground may cause unstable movements	they respond unpredictably to turbulence or wind	gravitational variations may cause orbital discrepancies.

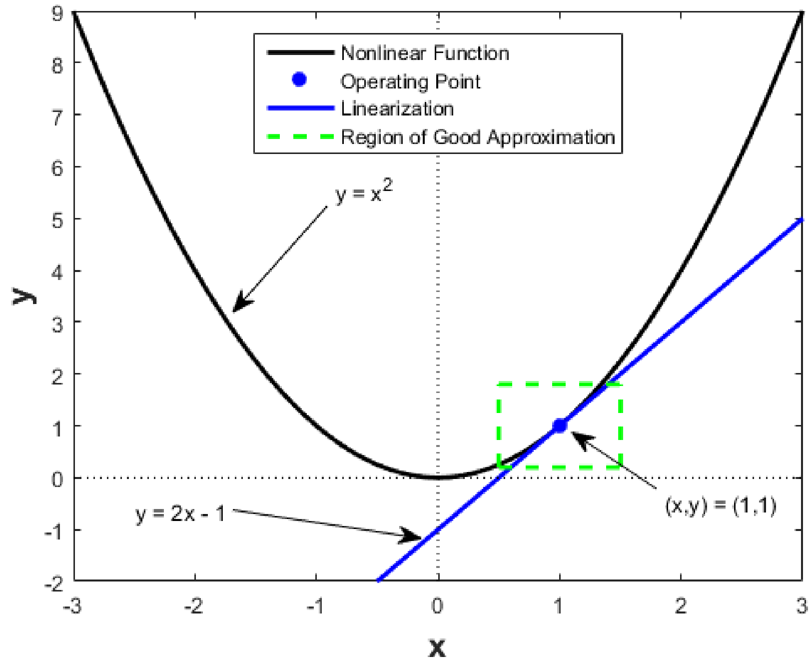






WHY LINEARIZE?

- **Linearization** is a mathematical technique that transforms systems **to behave as if they were linear** in a specific frame (around an equilibrium state).
- This means that once linearized, the system can be controlled with simple and well-established tools, making it stable and easier to manipulate.
- The system is controllable if we can express it using the state-space form: $\boldsymbol{dx = Ax + Bu}$ after linearization.



SYSTEM STUDY

WORKING STEPS

In this project, we explored the linearization of a complex dynamic system using two specific techniques:

- **State Transformation**
- **Feedback Control**

By simplifying and stabilizing this non-linear system, we obtained a model that is easier to analyse and control. This approach is particularly useful in areas like aerospace and autonomous systems, where stability and precision are essential.

METHODOLOGY AND CALCULATIONS

- 1. Identify State Transformation:
  - Find $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ such that the system dynamics in terms of $\boldsymbol{z}$ are linear.
- 2. Derive the New Dynamics:
  - Calculate $\boldsymbol{dz}$ and replace the original non-linear dynamics.
- 3. Select the Command Input Transformation:
  - Choose $\boldsymbol{u} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\beta}(\boldsymbol{x})\boldsymbol{v}$ to achieve a linear form.
- 4. Simplify the Linear Form:
  - Write the dynamics as $\boldsymbol{dz} = \boldsymbol{Az} + \boldsymbol{Bv}$**,** where A & B are matrices, simplifying control design.

PRESENTING OUR NON-LINEAR SYSTEM

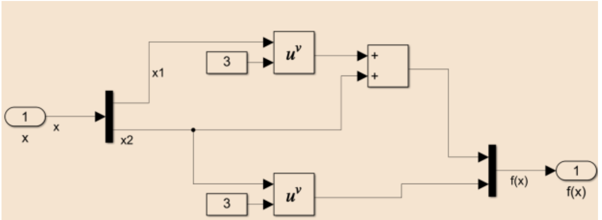
- We worked with a non-linear model represented by the equation :

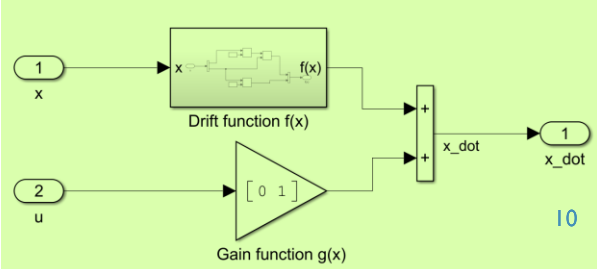
$$f(x) = \begin{bmatrix} x_1^3 + x_2 \\ x_2^3 \end{bmatrix}, g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad \dot{x} = \begin{pmatrix} x_1^3 + x_2 \\ x_2^3 + u \end{pmatrix}$$

Where $x1$ and $x2$ are the system states, and u is the control input.

- The system presents non-linearity due to cubic terms ($x1$ and $x2$), **which make the system's dynamics dependent on the state amplitudes**. These terms can cause unpredictable behaviours, such as oscillations or divergences, depending on the input u and initial conditions.

Representation of the system in MATLAB Simulink





LINEARISATION

Linearization by State Transformation

- By applying a state transformation, we rewrite the equations to obtain a simplified version.
- After calculation, we find that the state transformation to apply is as follows:

$$z_1 = -x_1$$
$$z_2 = -x_1^3 - x_2$$

=> $z = T(x) = \begin{bmatrix} -x_1 \\ -x_1^3 - x_2 \end{bmatrix}$

New Dynamics:

- Expressing the system in the new base (z_1, z_2) and replacing the original equations, we obtain an equivalent linearized model for control.
- This allows us to reformulate the system in terms of new state variables z:

$$\boxed{\dot{z} = Az + Bv}$$

PRESENTATION OF OUR NON-LINEAR SYSTEM

- We worked with a non-linear model represented by the equation:

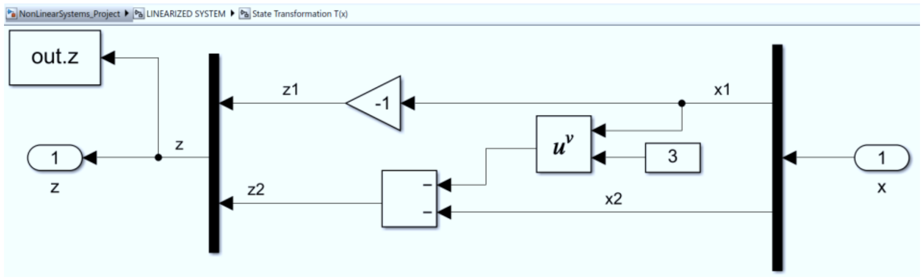
$$\dot{x} = \begin{pmatrix} x_1^3 + x_2 \\ x_2^3 + u \end{pmatrix}$$

$$z = T(x) = \begin{bmatrix} -x_1 \\ -x_1^3 - x_2 \end{bmatrix}$$

$$\dot{z} = \frac{\partial z}{\partial x}\dot{x} = \begin{bmatrix} -1 & 0 \\ -3x_1^2 & -7 \end{bmatrix} \begin{bmatrix} x_1^3 + x_2 \\ x_2^3 + v \end{bmatrix}$$

$$\dot{z} = \begin{bmatrix} -x_1^3 - x_2 \\ -3x_1^2(x_1^3 + x_2) - x_2^3 - 21 \end{bmatrix}$$

Representation of the system in MATLAB Simulink



DESIGN OF FEEDBACK CONTROLLER

Calculation of Control with the Pole Placement Method

- To ensure the stability of our system, we used the pole placement method. We calculated the controller gain **K** for state feedback. The gain is adjusted to position the poles of the linearized system. The desired poles are: **λ1=0 & λ2=−1,** These poles correspond to the eigenvalues of the matrix (A−BK) after applying the controller.
- The state feedback controller is defined by : $\boldsymbol{v} = -\boldsymbol{Kz}$

$$\dot{z}_1 = z_2 = -x_1^3 - x_2$$

where K is the gain vector to be determined. $K = [k_1 \; k_2]$

$$\dot{Z} = AZ + BV$$

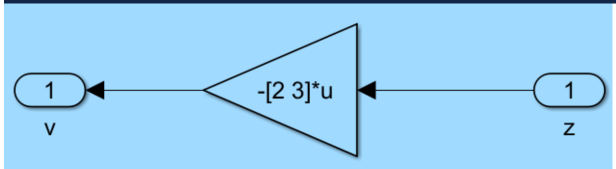
$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \qquad A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$A_{\text{cl}} = A - BK = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_1 & -1 - k_2 \end{bmatrix}$$

det (Acl−λI)=0 => $\left| \begin{bmatrix} -\lambda & 1 \\ -k_1 & -1 - k_2 - \lambda \end{bmatrix} \right| = 0$

$$\boxed{K = [2 \; 3]}$$

Representation of the system in MATLAB Simulink



CONTROLLER WITH LINEARIZED INPUT

- Simulations were conducted in Simulink (MATLAB environment) to validate our approach and calculations :

$$\dot{z}_1 = z_2 = -x_1^3 - x_2$$

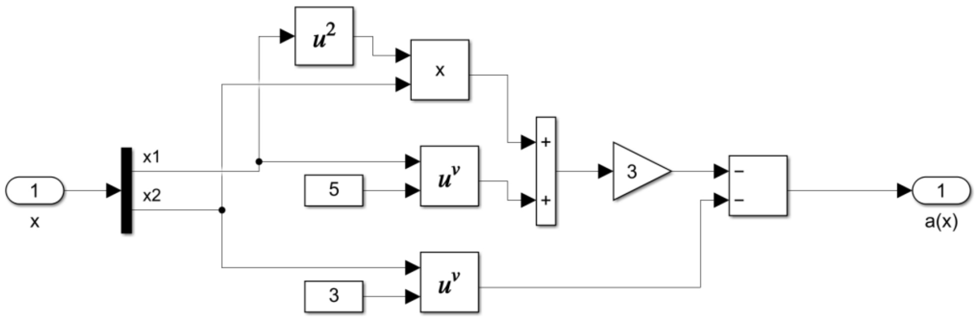
$$\dot{Z} = AZ + BV$$

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v$$

$$V = -3x_1^2(x_1^3 + x_2) - x_2^3 - u$$

$$U = -V - 3x_1^2(x_1^3 + x_2)$$

Representation of the system in MATLAB Simulink

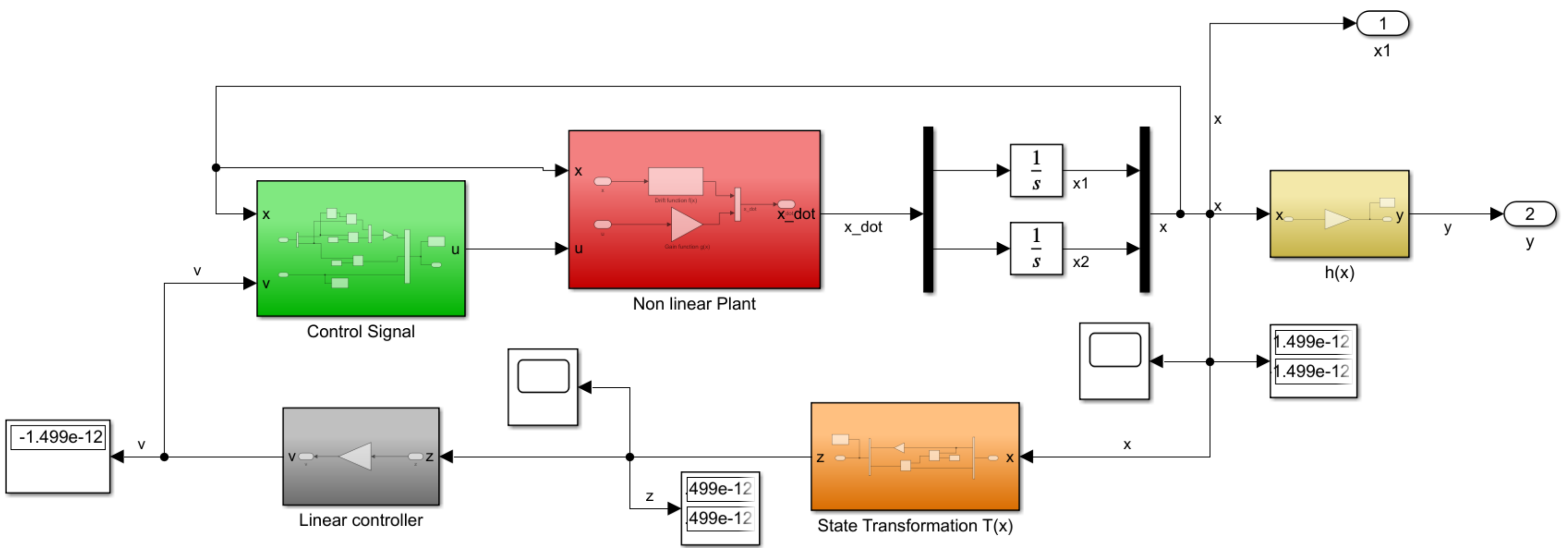


MATLAB SIMULINK

IMPLEMENTATION OF OUR MODEL

SIMULINK - LINEARIZED SYSTEM

- Simulations will be performed in Simulink (MATLAB environment) to validate our approach and calculations. Here's our non-linear system in red, and the setup to linearize its state x.

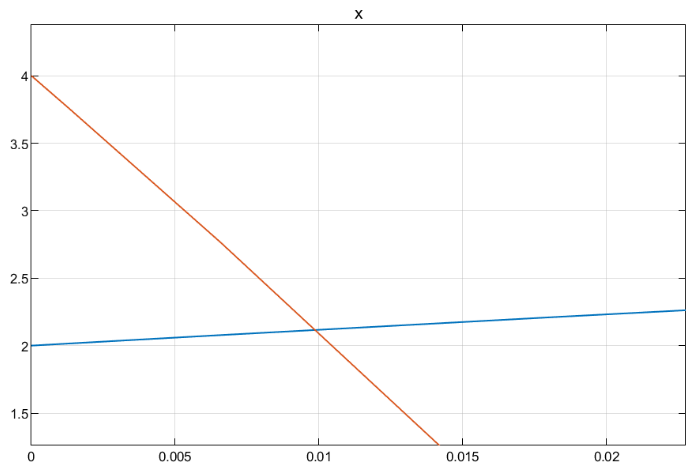


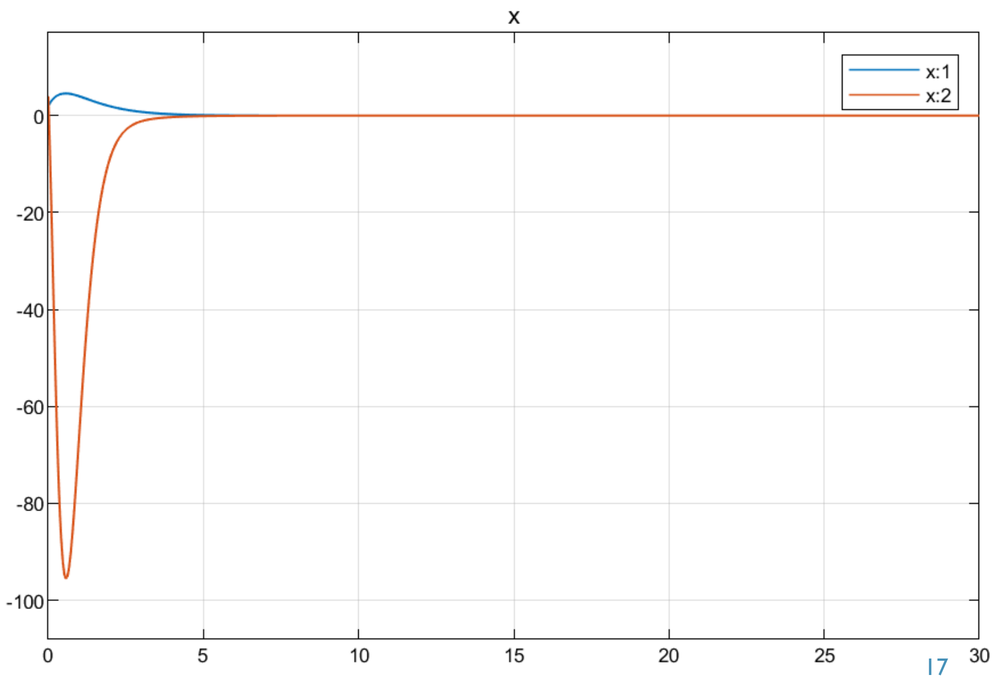
SIMULINK - LINEARIZED SYSTEM

LINEARIZED SYSTEM STABILITY TEST

- **1st Test : $x_1 = 2$ & $x_2 = 4$**

We initialize our system at a position different from its equilibrium point x=[0; 0], to verify it will return to the equilibrium position.





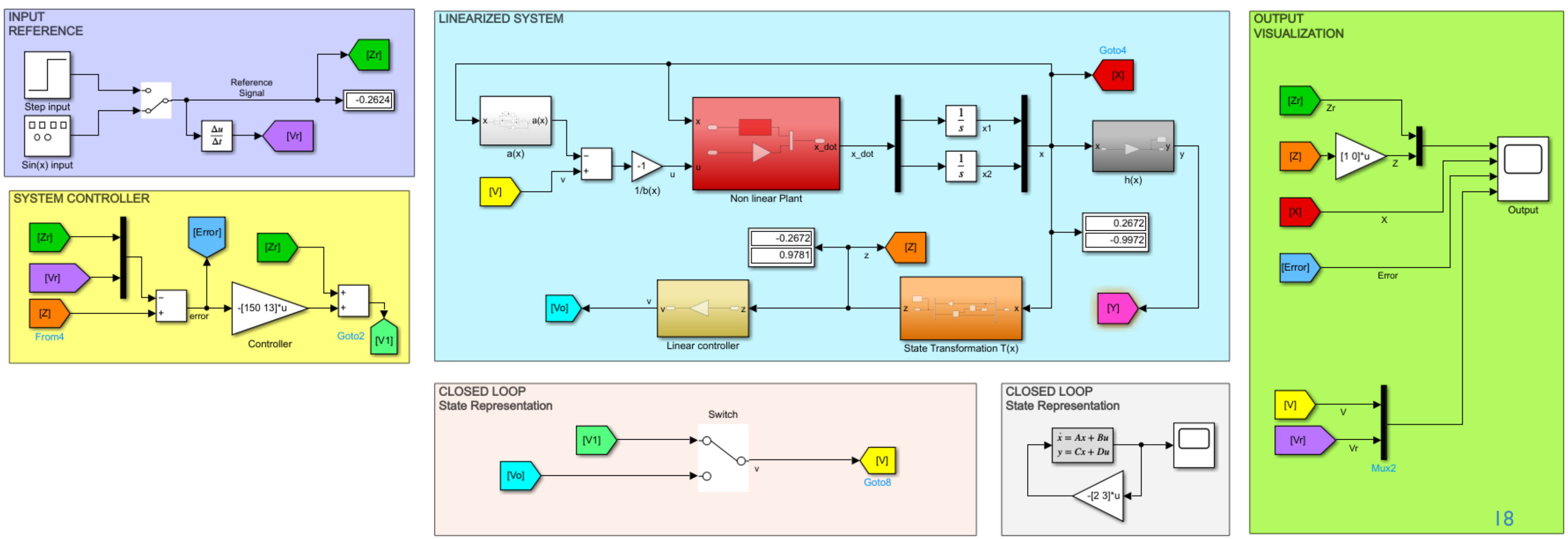
CONCLUSION

Initially, the system does deviate from defined positions, but after 5 sec, the system stabilizes.

LINEARIZED SYSTEM WITH REFERENCE INPUT

CONTROL OF THE LINEARIZED SYSTEM

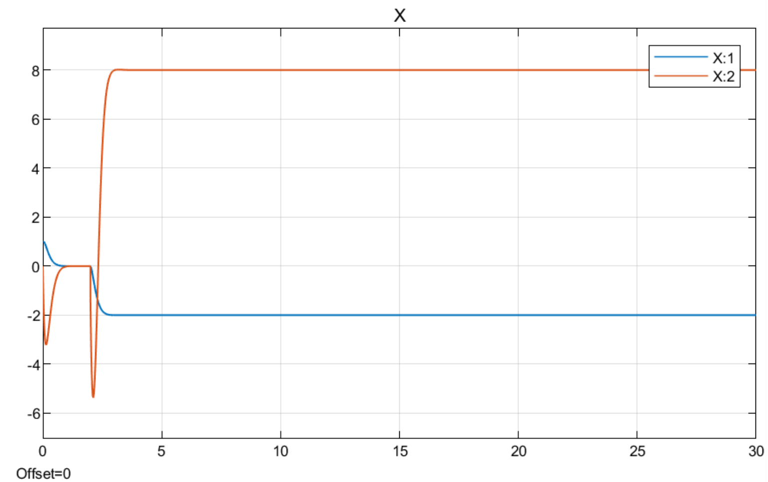
- **Principle :** We now have the complete representation of our system equipped with an input. The principle is simple: define a signal that serves as a reference for our model. With our controller gain K, we can command our robot to track this reference.

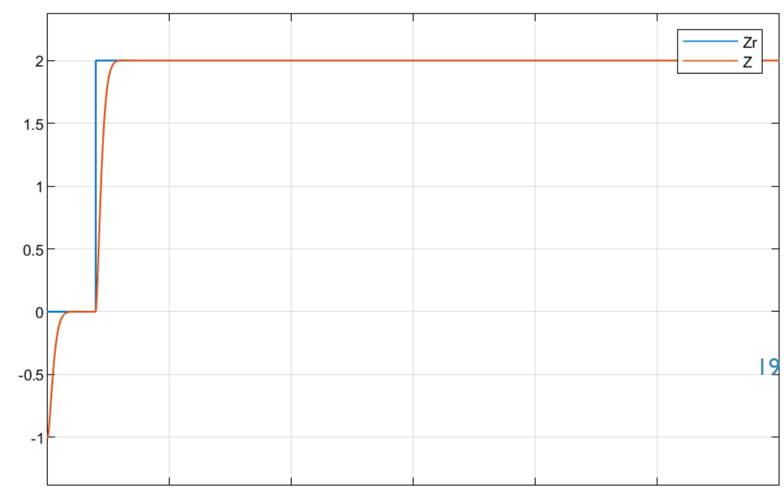


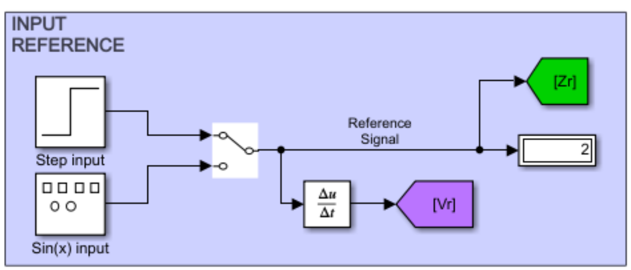
SIMULINK - LINEARIZED SYSTEM

LINEARIZED SYSTEM STABILITY TEST

- **2nd Test :**
- We send a pulse of value 2, which will trigger at 2 sec. This represents our **reference trajectory Zr**; Since we left the system in an unstable position ($x \neq 0$), we will now observe the behaviour of the **output Z with respect to Zr.**

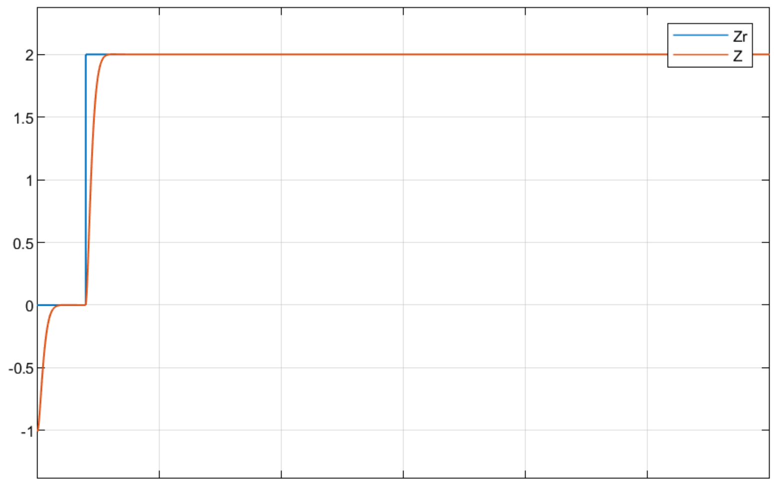


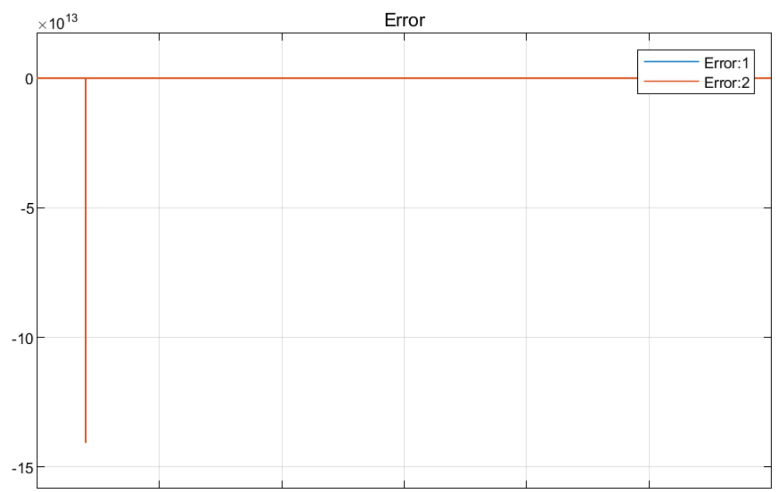




SIMULINK - LINEARIZED SYSTEM

LINEARIZED SYSTEM STABILITY TEST





- **CONCLUSION**

At first, the system starts from unstable positions. **1 sec** is enough for it to stabilize and begin tracking the reference. It experiences a slight delay due to the sudden impulse at **2 sec**, but it reaches the reference at around **1.8 sec**, and the tracking becomes perfect afterward **(Error → 0).**

In this case, tracking is more obvious because our **reference trajectory is a linear function (y = Zr = 2).**

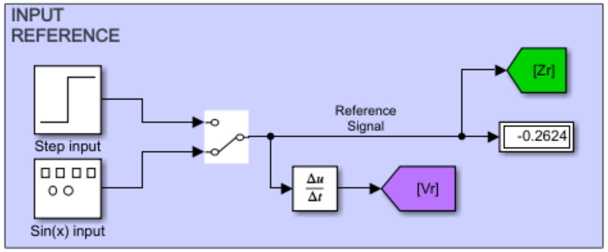
What would we have in terms of response time, robustness, and stability with respect to a non-linear reference trajectory Zr?

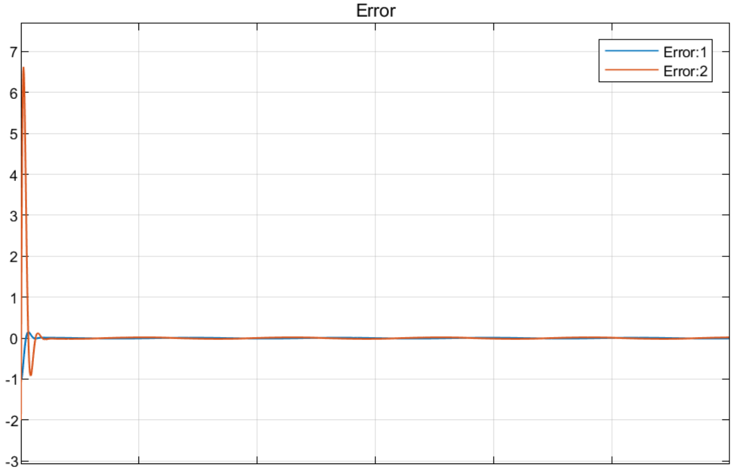
SIMULINK - LINEARIZED SYSTEM

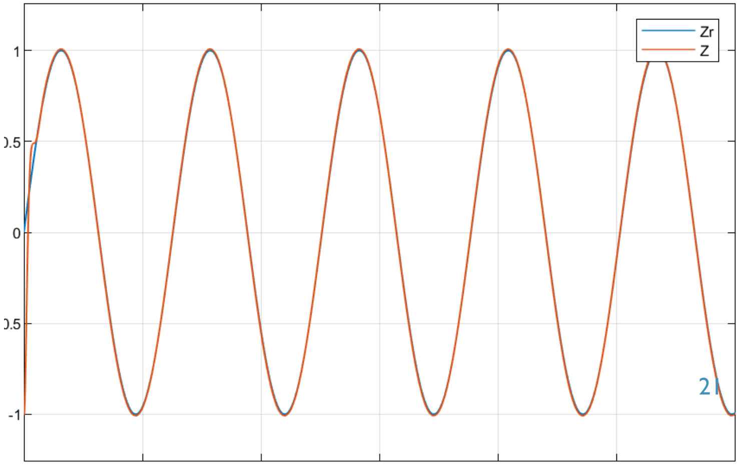
LINEARIZED SYSTEM STABILITY TEST

- **3rd Test :**
- Our **reference trajectory Zr is a sinusoidal function** ;

Since we left the system in an unstable position (x ≠ 0), we will now study the behaviour of the **output Z with respect to Zr = sin(x).**

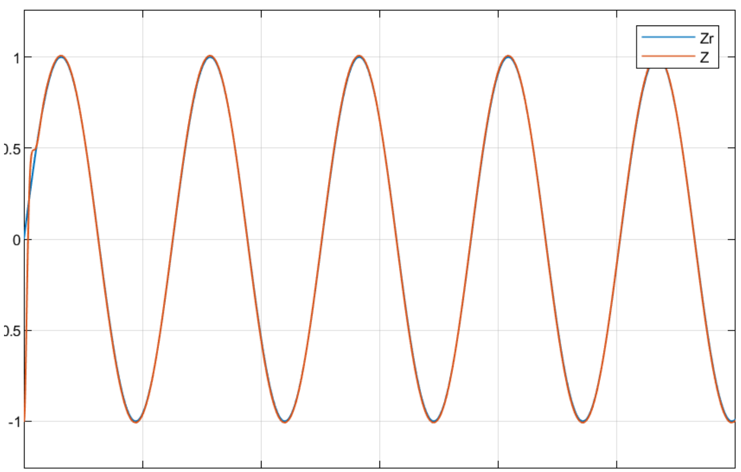


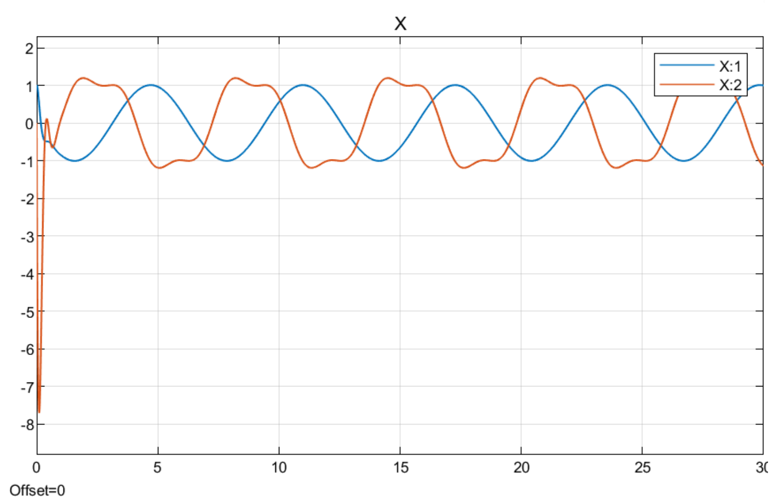




SIMILINK - LINEARIZED SYSTEM

LINEARIZED SYSTEM STABILITY TEST





- **CONCLUSION**

At first, the system starts from unstable positions, but it successfully tracks the target within approximately **0.6 sec**, and then the tracking becomes perfect **(Error → 0)**. The tracking is even faster as we increased the controller gain, knowing that the trajectory was non-linear, which initially caused some state drift, but the system quickly resolved this deviation.

So, we have the proof! We linearized our system and applied a control command while maintaining its stability.

- Therefore, our system can even track a mythical dragon, if we consider Zr as the movement curve of a dragon, for example!

CONCLUSION

- The results obtained confirm that linearization is a powerful method to **transform complex dynamic systems into controllable and predictable models**. With this approach, we were able to improve the **system's stability**, **robustness** against disturbances, and the **speed of the system's responses**. These qualities are essential in many modern applications where precision and reliability are critical.
- The results show that **linearization is not just a mathematical theory**; it is a concrete and effective solution to tackle the challenges of non-linear systems in the real world. This advancement opens the way for innovations that are safer and more efficient in fields like robotics, aeronautics, and aerospace.
- By mastering these techniques, we are preparing the ground for tomorrow's technologies, capable of meeting the most demanding challenges.

- **Concrete Application Examples**
- **Delivery Drones**: With increased stability, they can fly safely, even in strong winds, to deliver packages in hard-to-reach areas.
- **Search and Rescue Robots**: On rugged terrain, robustness allows these robots to remain operational despite unforeseen conditions.
- **Communication Satellites**: A quick and predictable response ensures perfect alignment for continuous network coverage.
- **Autonomous Vehicles**: By maintaining stability in complex situations, linearization contributes to safer and more reliable driving.

THANK YOU FOR YOUR ATTENTION!

CONTACT US

LINKEDIN :

HTTPS://LINKEDIN.COM/IN/TOKO1EJM

TOKO 1ER JEAN MALDINI

STUDENT SMART AEROSPACE AND AUTONOMOUS SYSTEMS

LINKEDIN :

HTTPS://LINKEDIN.COM/IN/FIKRESELASSIE-SEID

FIKRESELASSIE ESHETU SEID

STUDENT SMART AEROSPACE AND AUTONOMOUS SYSTEMS

THANKS TO

MOHAMMED A M SAFARINI

ASSISTANT PROFESSEUR, ÉTUDIANT EN PHD