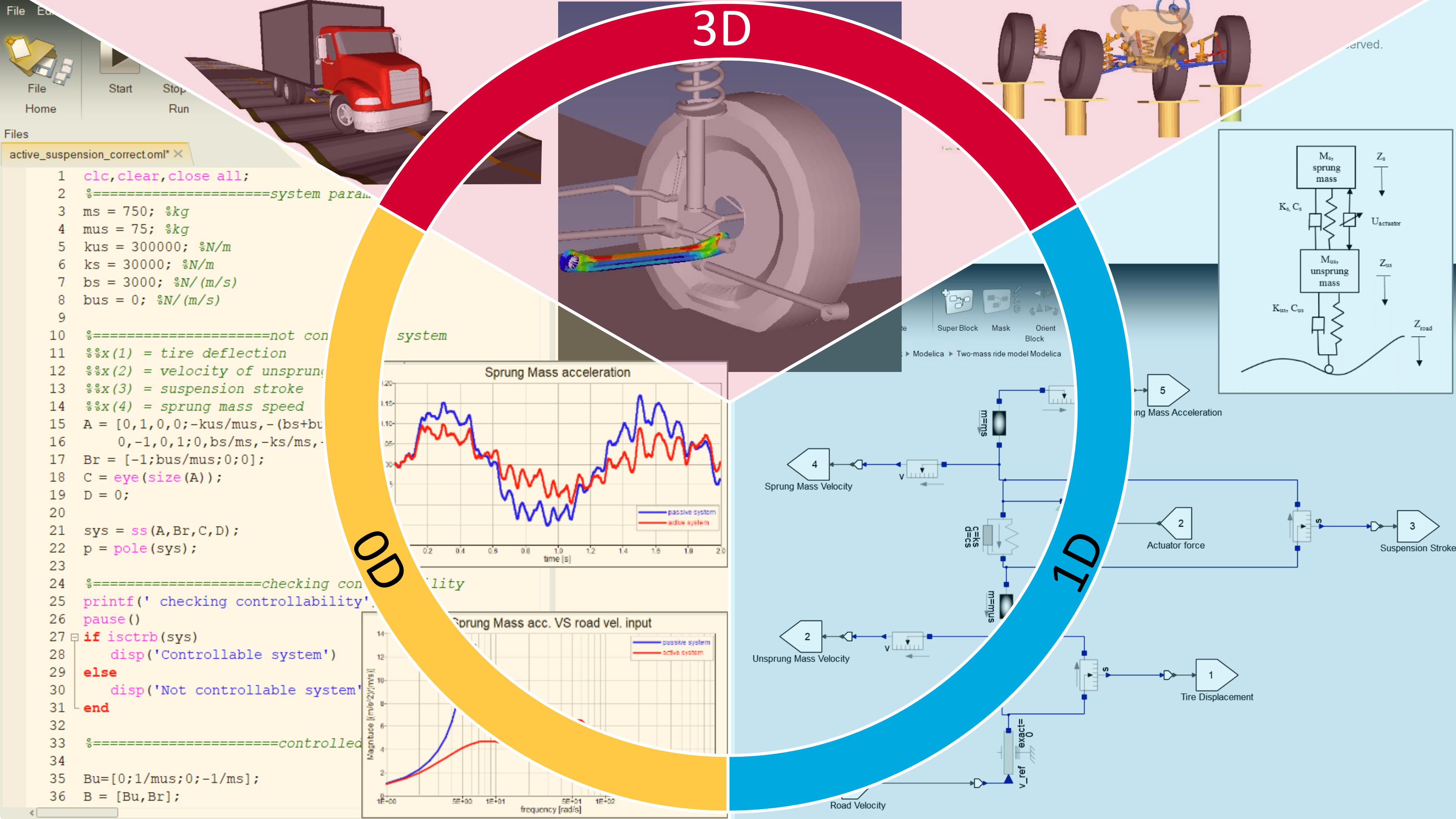
John Straetmans - Livio Mariano

jstraetmans@altair.com Livio.Mariano@altair.com

Step 1 - Drone Modeling, Control and Validation


3D
0D
1D
File
Home
Start
Stop
Run
Files
active_suspension_correct.oml*
clc,clear,close all;
%====================system param
ms = 750; %kg
mus = 75; %kg
kus = 300000; %N/m
ks = 30000; %N/m
bs = 3000; %N/(m/s)
bus = 0; %N/(m/s)
%====================not con
system
%%x(1) = tire deflection
%%x(2) = velocity of unsprun
%%x(3) = suspension stroke
%%x(4) = sprung mass speed
A = [0,1,0,0;-kus/mus,-(bs+bu
0,-1,0,1;0,bs/ms,-ks/ms,-
Br = [-1;bus/mus;0;0];
C = eye(size(A));
D = 0;
sys = ss(A,Br,C,D);
p = pole(sys);
%====================checking con
lity
printf(' checking controllability'
pause()
if isctrb(sys)
disp('Controllable system')
else
disp('Not controllable system'
end
%====================controlled
Bu=[0;1/mus;0;-1/ms];
B = [Bu,Br];
Sprung Mass acceleration
passive system
active system
time [s]
prung Mass acc. VS road vel. input
Magnitude [(m/s^2)/(m/s)]
frequency [rad/s]
Super Block
Mask
Orient
Block
Modelica
Two-mass ride model Modelica
Sprung Mass Velocity
Unsprung Mass Velocity
Actuator force
Suspension Stroke
Tire Displacement
Road Velocity
ing Mass Acceleration
m=ms
m=mus
c=cs
d=ks
exact=0
v_ref
Ms, sprung mass
Mus, unsprung mass
Ks, Cs
Kus, Cus
Uactuator
Zs
Zus
Zroad

Agenda

1- SYSTEM INTRODUCTION

- Introduction of the project
- Explanation of the body and inertial frames

2- SYSTEM DYNAMICS & MODELING

- Explanation of the computation of the attitude and the position of the drone

3- VALIDATION

- The control Inputs
- Comparing data with the paper

4- CONCLUSION

- Summary of process and results
- Discuss what is next

Agenda

1- SYSTEM INTRODUCTION

- Introduction of the project
- Explanation of the body and inertial frames

2- SYSTEM DYNAMICS & MODELING

- Explanation of the computation of the attitude and the position of the drone

3- VALIDATION

- The control Inputs
- Comparing data with the paper

4- CONCLUSION

- Summary of process and results
- Discuss what is next

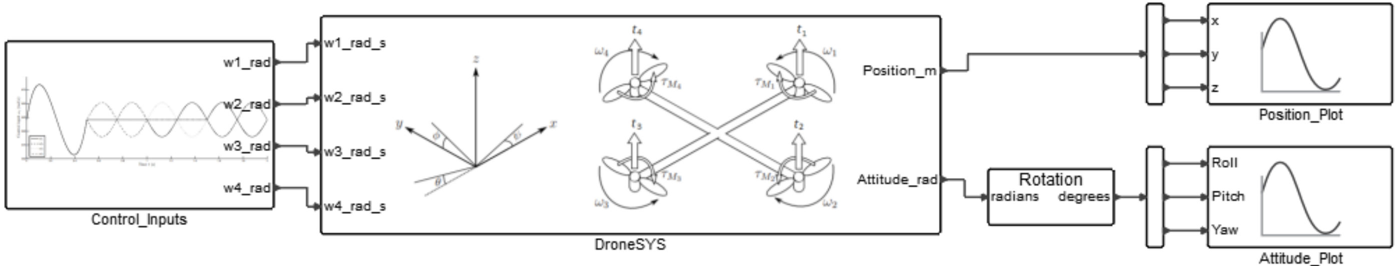
SYSTEM INTRODUCTION

Introduction

Quadcopters are a type of four rotor drone

In this project, a system is developed to model the dynamics of a quadcopter with the rotor speeds as the four primary inputs

The data from this model is then compared to the data found in the included paper in order to verify its accuracy



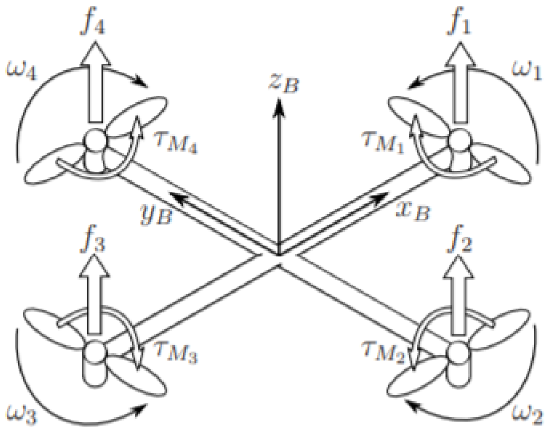
Simulates the behavior of the drone in response to the control inputs of found in for the rotor speeds based on the values found in chapter three of the included paper

The Body and Inertial Frames

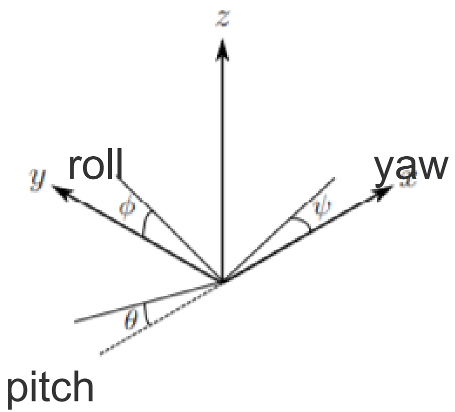
The quadcopter has six degrees of freedom. The analysis of this system is simplified by using two different frames of reference: the body frame and the inertial frame

The body frame is ideal for initially computing all of the forces that act on the drone

The inertial frame is used in order to be able to compare to a reference position and attitude when we plot the results



The Body Frame



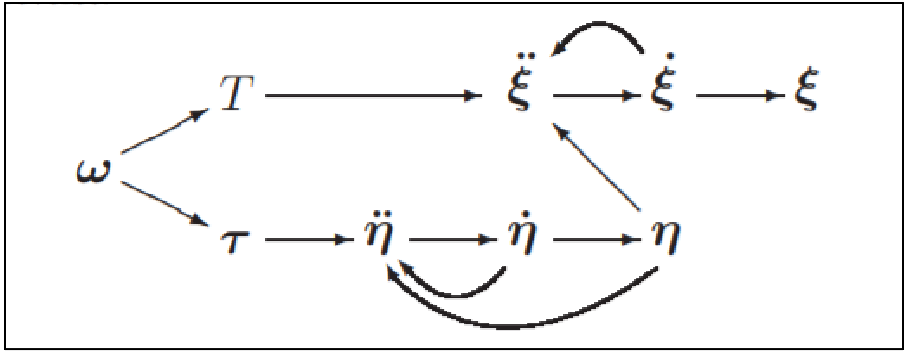
The Inertial Frame

ALTAIR

System Organization

The model is organized into two main subsystems, the computation of the attitude ($\eta = [\varphi, \theta, \psi]$) and the computation of the position ($\xi = [x,y,z]$) The two systems are coupled: the attitude is computed first, as the position relies on the attitude, due to the conversion of the thrust into the body frame

After computing the thrust (T) and the torque (τ) from the rotor speeds (ω), we then calculate the attitude and position serperately



Agenda

1- SYSTEM INTRODUCTION

- Introduction of the project
- Explanation of the body and inertial frames

1- SYSTEM DYNAMICS & MODELING

- Explanation of the computation of the attitude and the position of the drone

3- VALIDATION

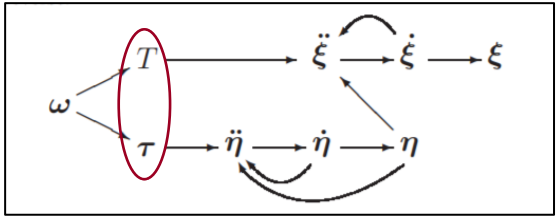
- The control Inputs
- Comparing data with the paper

4- CONCLUSION

- Summary of process and results

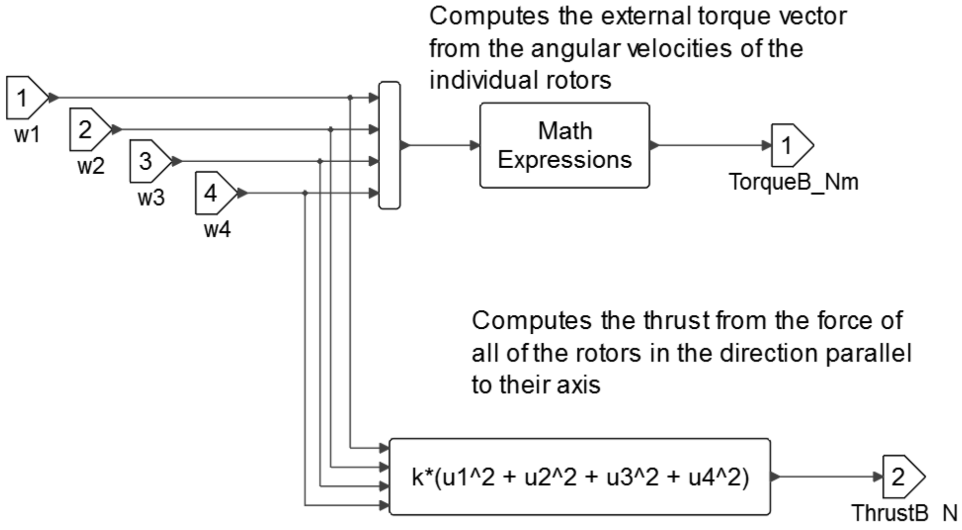
SYSTEM DYNAMICS & MODELING

Thrust and Torque in the Body Frame



In order to know the dynamics of the system, the thrust and torque in the body frame must be computed from the values of the rotor speeds

k, *b*, and *l*, are the drag, the lift, and the distance from the center of mass of the drone to the rotors respectively



Torque

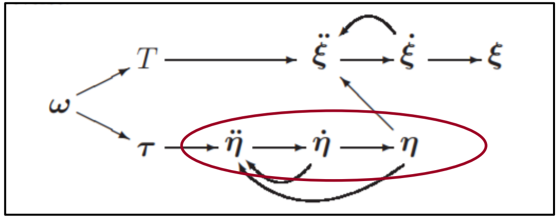
$$\boldsymbol{\tau}_a = \begin{pmatrix} \tau_a^1 \\ \tau_a^2 \\ \tau_a^3 \end{pmatrix} = \begin{pmatrix} bl(\omega_4^2 - \omega_2^2) \\ bl(\omega_3^2 - \omega_1^2) \\ k(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{pmatrix}$$

Thrust

$$T = \sum_{i=1}^{4} f_i = b \sum_{i=1}^{4} \omega_i^2, \quad \boldsymbol{T}^B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

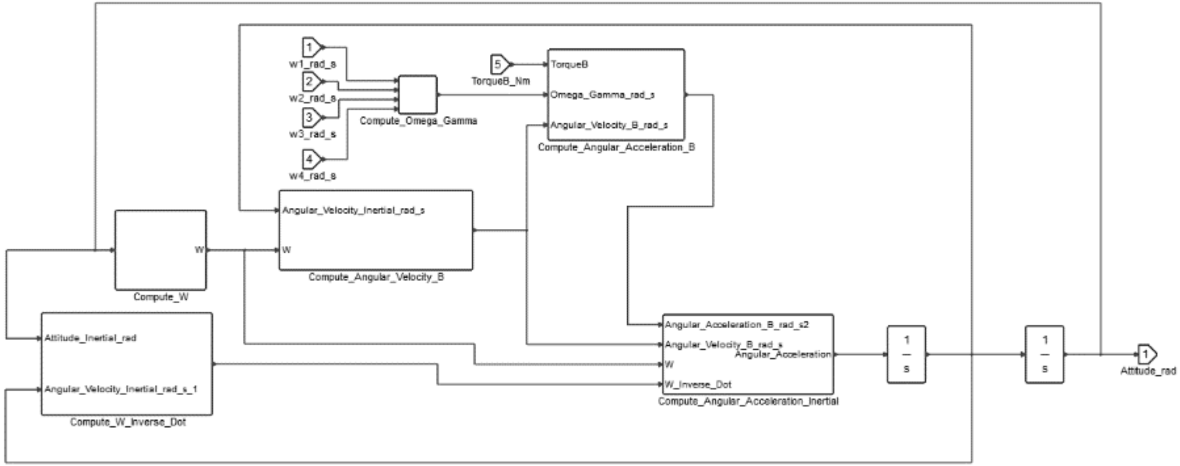
ALTAIR

Altitude



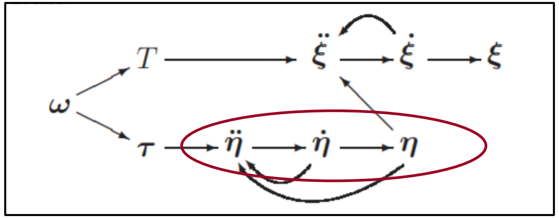
By taking advantage of the body frame, we can use the torque vector in the body frame to calculate the angular acceleration in the body frame. Then we can convert this back to the inertial frame afterwards.

This is easier than simply computing the angular acceleration in the inertial frame straightaway, since the torque vector is very simple in the body frame as reported previously



Computes the orientation of the drone using the torque created by each of the rotors

The Transformation Matrix

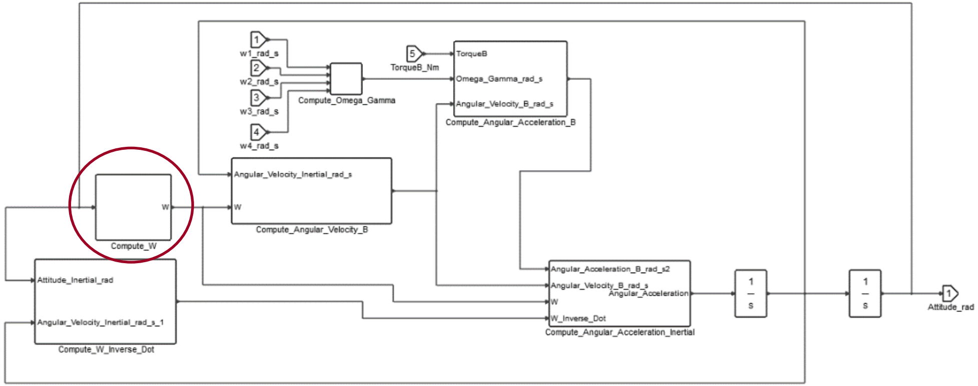


In order to convert the angular acceleration and velocites from the body frame, we must compute the transformation matrix (*W*)

Here, ν is the angular velocity in the body frame

$$\dot{\boldsymbol{\eta}} = \boldsymbol{W}_\eta^{-1}\boldsymbol{\nu}, \qquad \begin{bmatrix}\dot\phi\\ \dot\theta\\ \dot\psi\end{bmatrix} = \begin{bmatrix}1 & S_\phi T_\theta & C_\phi T_\theta\\ 0 & C_\phi & -S_\phi\\ 0 & S_\phi/C_\theta & C_\phi/C_\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix},$$

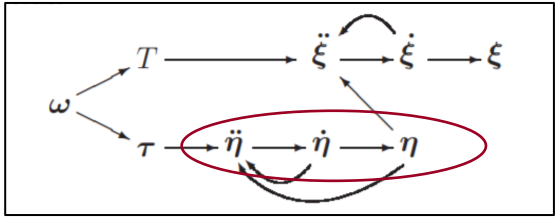
$$\boldsymbol{\nu} = \boldsymbol{W}_\eta\dot{\boldsymbol{\eta}}, \qquad \begin{bmatrix}p\\ q\\ r\end{bmatrix} = \begin{bmatrix}1 & 0 & -S_\theta\\ 0 & C_\phi & C_\theta S_\phi\\ 0 & -S_\phi & C_\theta C_\phi\end{bmatrix}\begin{bmatrix}\dot\phi\\ \dot\theta\\ \dot\psi\end{bmatrix},$$



Computes the orientation of the drone using the
torque created by each of the rotors

The Angular Acceleration (Body)

Next we can compute the angular acceleration in the body frame from the torque and the current gyroscopic forces

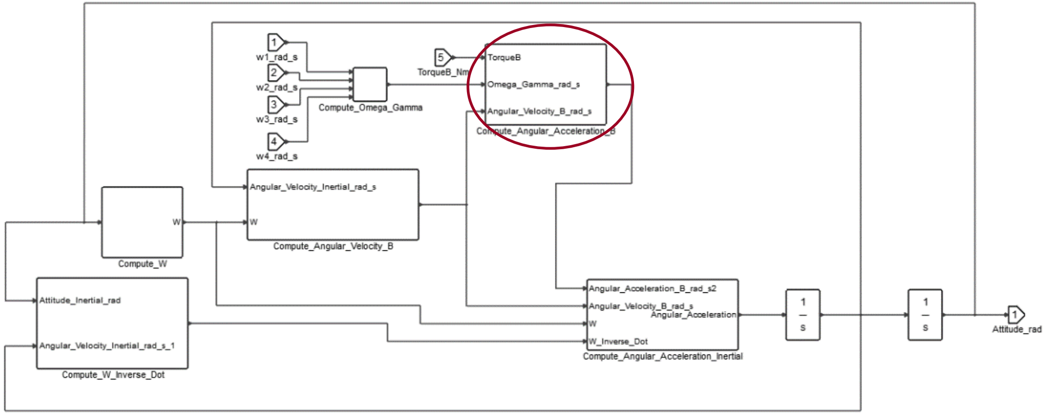


The velocity in the body frame is needed for this step, which can be computed using *W*

I is the moment of inertia matrix of the drone and I_r is the moment of inertia of the rotors

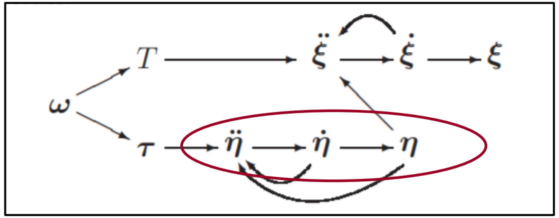
$$\omega_\Gamma = \omega_1 - \omega_2 + \omega_3 - \omega_4$$

$$\dot{\boldsymbol{\nu}} = \boldsymbol{I}^{-1}\left(-\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}\,p \\ I_{yy}\,q \\ I_{zz}\,r \end{bmatrix} - I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_\Gamma + \boldsymbol{\tau}\right)$$



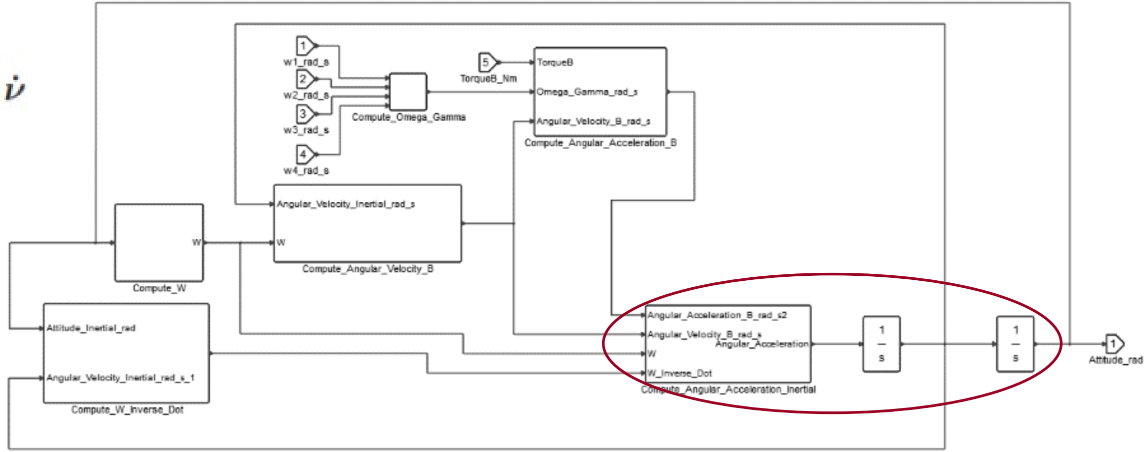
Computes the orientation of the drone using the torque created by each of the rotors

The Angular Acceleration (Inertial)



Finally, we can convert the angular acceleration to the inertial frame using W, and integrate twice to yield the attitude

$$\ddot{\eta} = \frac{\mathrm{d}}{\mathrm{d}t}\left(W_\eta^{-1}\nu\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left(W_\eta^{-1}\right)\nu + W_\eta^{-1}\dot{\nu}$$

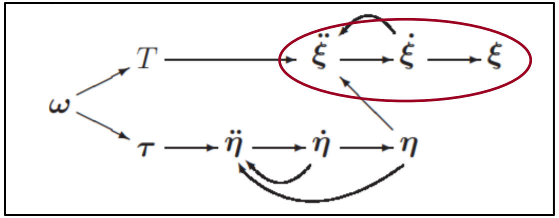


Computes the orientation of the drone using the torque created by each of the rotors

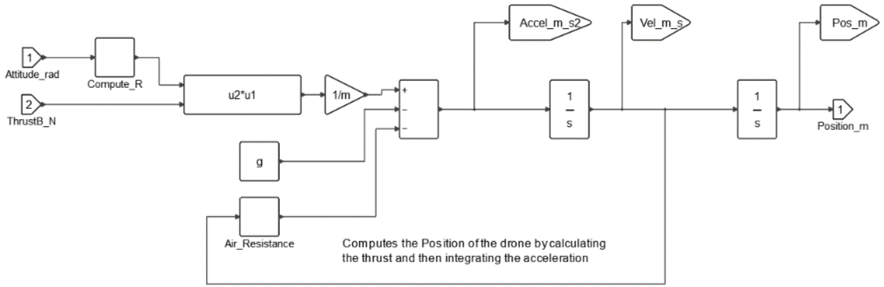
ALTAIR

Position

For the position, we must convert the thrust in the body frame into the inertial frame. Therefore, we must compute the transformation matrix R

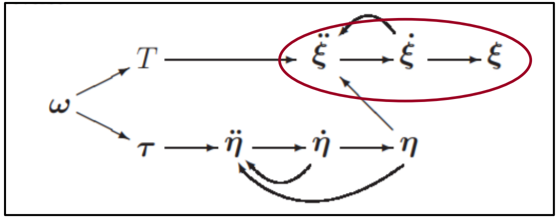


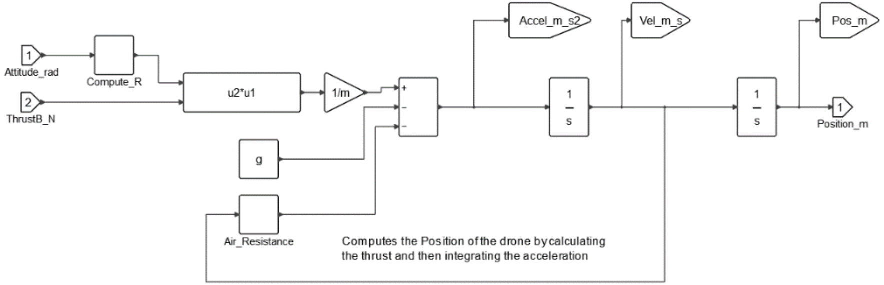
$$R = \begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix}$$



Position

Next, the acceleration is computed from the thrust, gravity, and the aerodynamics, which can then be integrated twice to obtain the position





Aerodynamical Effects

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \underbrace{\boxed{\begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix}}}_{R} \boxed{- \frac{1}{m} \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}}$$

Agenda

1- SYSTEM INTRODUCTION

- Introduction of the project
- Explanation of the body and inertial frames

2- SYSTEM DYNAMICS & MODELING

- Explanation of the computation of the attitude and the position of the drone

3- VALIDATION

- The control Inputs
- Comparing data with the paper

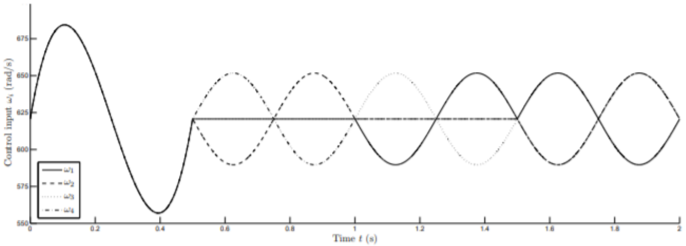
4- CONCLUSION

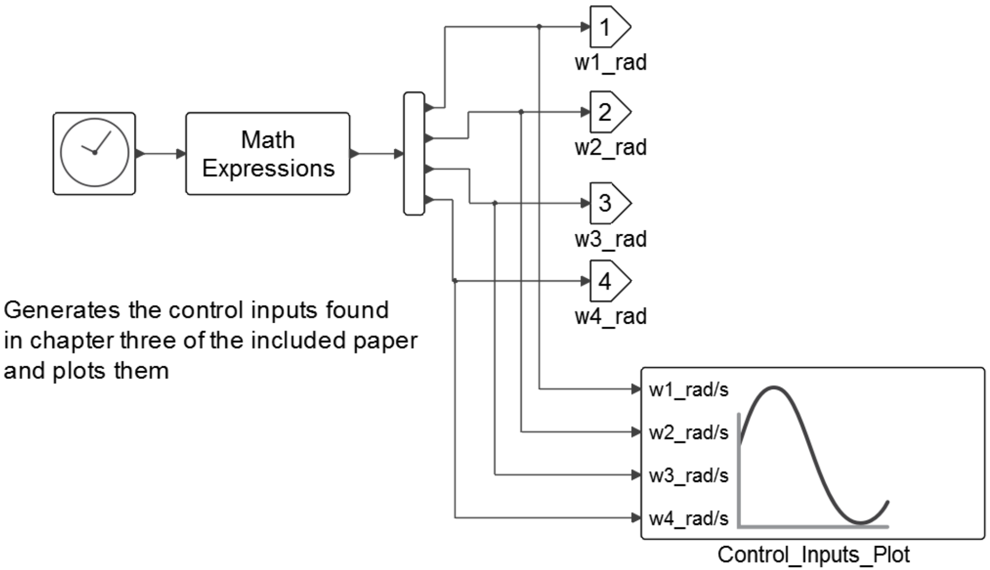
- Summary of process and results

VALIDATION

Using the Inputs Provided in the Paper

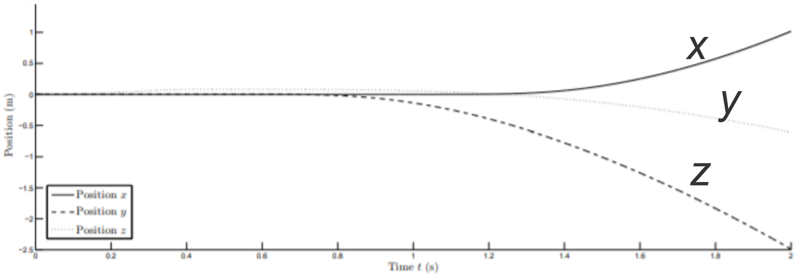
The inputs used in chapter three of the Luukkonen paper have been generated and then inputed to the system in order to compare results



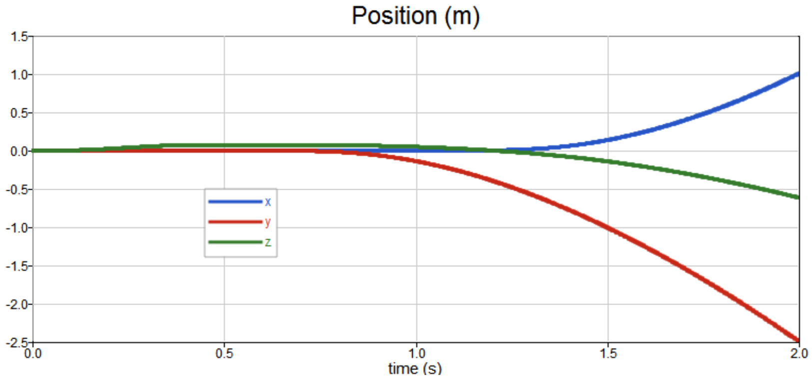


Position Comparison

Luukkonen's Data



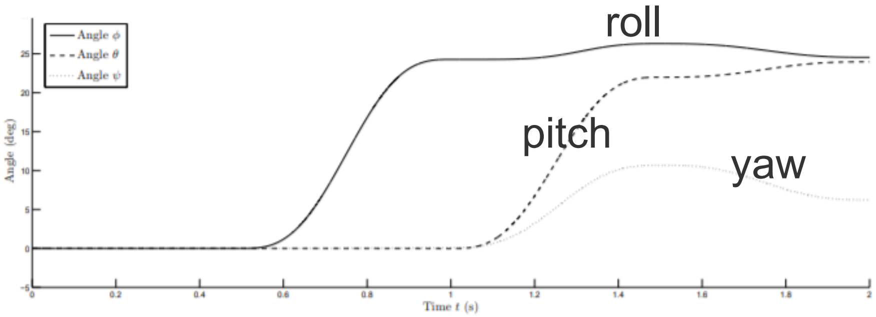
Collected Data



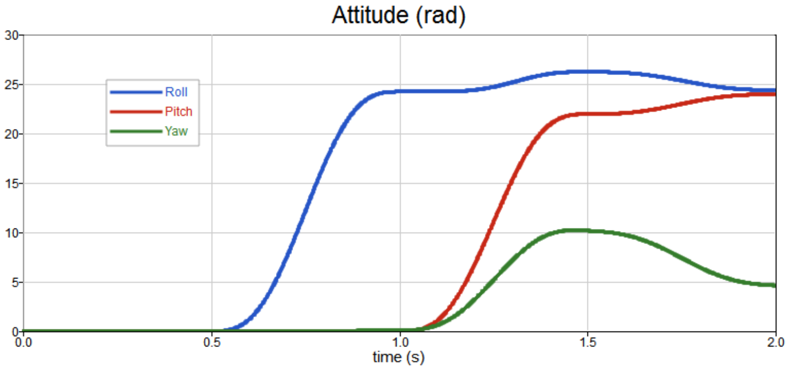
The data for the position of the drone appears to be in very good agreement with the reference data from the paper

Attitude Comparison

Luukkonen's Data



Collected Data



The altitude data appears to agree as well

ALTAIR

Agenda

1- SYSTEM INTRODUCTION

- Introduction of the project
- Explanation of the body and inertial frames

2- SYSTEM DYNAMICS & MODELING

- Explanation of the computation of the attitude and the position of the drone

3- VALIDATION

- The control Inputs
- Comparing data with the paper

4- CONCLUSION

- Summary of process and results
- Discussion of the next step

Conclusion

In conclusion, a model has been developed to simulate the dynamics of a drone

This model was then verified by comparing the dynamics generated by a set of predetermined input rotor speeds to the dynamics displayed in the associated paper

Now that we have validated the model and know that it is accurate, we can move on to the next step: implementing a PD controller to allow the drone to navigate to a reference position and altitude

THANK YOU

altair.com

#ONLYFORWARD

ALTAIR