March 11, 2015

# Aerospace System Guidance and Control
# Lesson IV

---

# Introduction to Simulink

POLITECNICO DI MILANO

DIPARTIMENTO DI
INGEGNERIA AEROSPAZIALE


Skyward Experimental Rocketry

Politecnico di Milano

**Author**: Luigi Cocco

**Editor**: Francescodario Cuzzocrea

**Abstract**
Simulink is an environment for modeling, analysis, and the simulation of dynamic systems, it supports the simulation of linear systems, and/or non- linear, operating with continuous signals and/or discrete signals, continuous and/or discrete time.

**Website**:
http://www.skywarder.eu

**E-mail**:
luigi.cocco@skywarder.eu

# Contents

# Chapter 1

# Introduction

## 1.1 Simulink

Simulink provides a graphical interface that allows you to dene and build the model through its block diagram. The blocks are transported and moved on the work surface by the mouse. Simulink includes a large number of libraries that contain many blocks capable of performing operations (more or less elementary) on signals. Simulink fully integrates with MATLAB. Simulink starts by typing << Simulink >> on the MATLAB prompt or press the Simulink icon in the MATLAB toolbar 1.1.

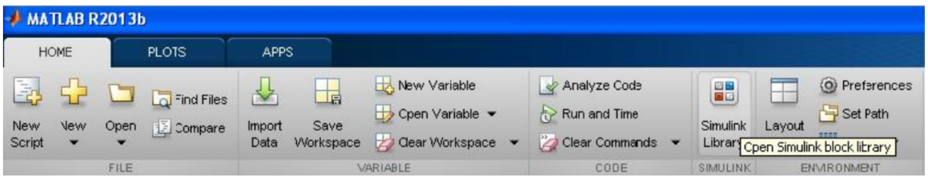


Figure 1.1: Simulink

• Simulink and MATLAB interacting through the workspace;

• The model parameters can be specied within the Simulink blocks using symbolic variables, whose value is stored in the workspace of MATLAB;

• Simulation results can be stored by block called To workspace

## 1.2 Getting started

Simulink new model can be open by clicking New model in the upper left

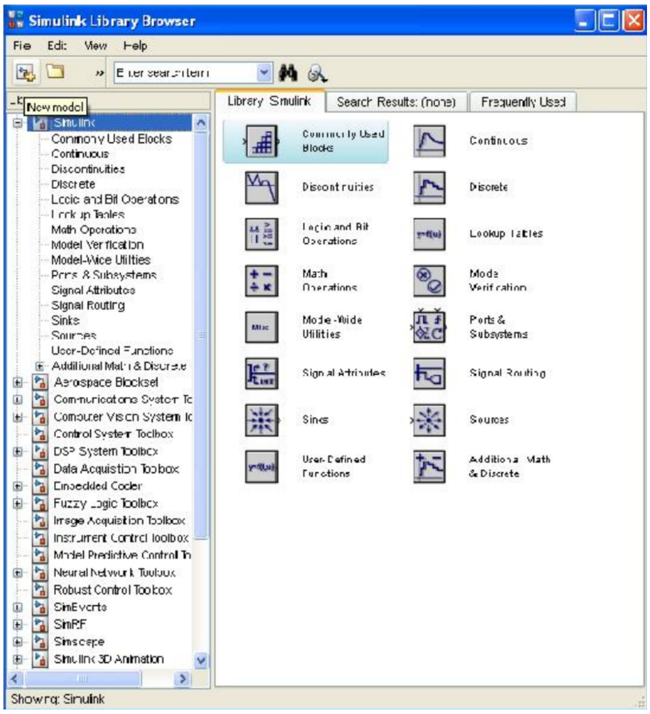


Figure 1.2: New Model

In order to create a block model, following instruction has to be done • Copying blocks (subsystems) necessary to create the calculation model you want inside the working area • Connect blocks with arrows made with mouse • Select the simulation parameters • Start the simulation and results analysis The Blocks of the model can be choosen by Simulink libraries or dened by user, and the copy can be made dragging blocks from libraries to workspace. Libraries can be selected by icons shown in Fig.1.2

## 1.3 Most important blocks

The Simulink's libraries as said in the previous section oers a lot of blocks but most used blocks can be found in the libraries showed in the figure below :

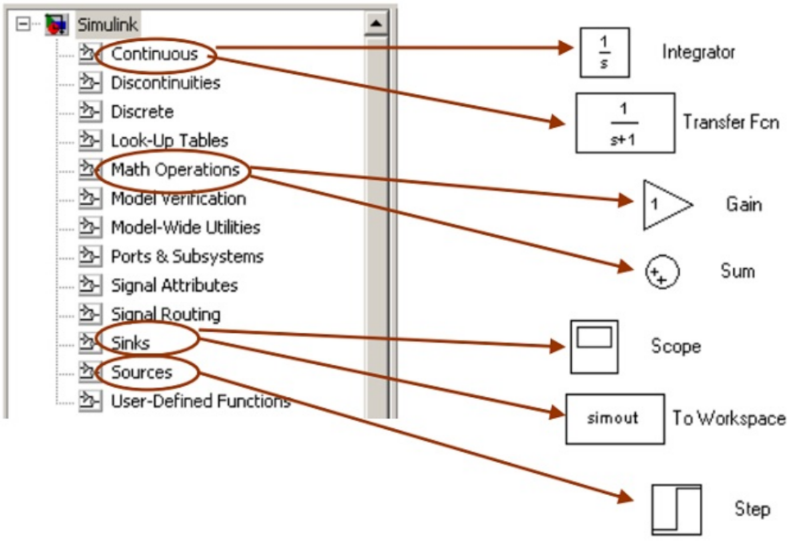


Figure 1.3: Most important blocks

# Chapter 2

# Pulse Response

In this chapter the pulse response of a generic system will be studied the system will be build on simulink and system's step response will be tested, this test is a preliminary analysis of system's stability. The generic system of this example is described by the following transfer function

$$G(s) = \frac{-s+1}{s^2+2s+2}$$

## 2.1 Simulink Model

The model is composed by a transfer function, scope, and pulse input, these blocks can be found respectively from Continuos,Sinks,Sources libraries, so the simulink model is showed in Fig.2.1

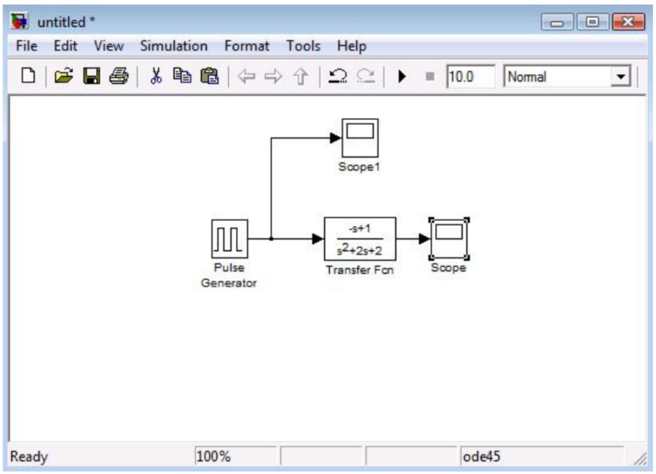


Figure 2.1: Simulink Model

and the output (scope) is showed below :

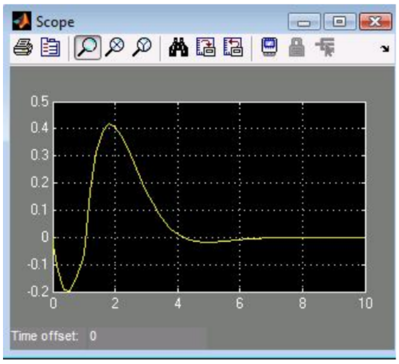


Figure 2.2: Scope

# Chapter 3

# Blimp system's stability

Blimp dynamic model can be written in a Laplace domain and the system's transfer function can be implemented on Simulink in order to study the dynamic stability of the system excited by a pulse input. Blimp's equations of motion are reported below
• Altitude

$$z(t) = \frac{1}{m}((T_1 + T_2)\Theta(t) - 2\rho z(t)\dot{S}c_D)$$

in Laplace domain it become

$$z(s) = \frac{T}{(ms^2 + 2\rho S c_D s)}\Theta(s)$$

Where $T = T_1 + T_2$;
•Yaw

$$\ddot{\psi}(t) = \frac{1}{J}((T_1(t) - T_2(t))d - \rho S c_D \frac{L^2}{4}\psi\dot{(t)})$$

in the Laplace domain

$$\psi(s) = \frac{(T_1(s) - T_2(s))}{(Js^2 + \rho S c_D \frac{L^2}{4} s)}d$$

where $T_1$ is the righet engine and $T_2$ is the left.

## 3.1 Altitude stability

In this section, altitude's transfer function will be implemented on simulink and it's stability will be tested by using pulse input. This analysis allow to understood the dynamic behavior of the system under the external perturbation.

### 3.1.1 Simulink Model

As already seen in chapter 2 the transfer function block can be use to build the laplace model of the system, however in this example the coecients of numerator and denominator of transfer function have to be calculated
**Data**
• ρ= 1.225Kg/$m^3$
• T = 10 N

- $m_{tot}$ = 2 Kg
- $c_D$ = 0.3
- S =$\pi R^2$= $\pi * 0.2^2$ = 0.1257$m^2$

And so, the transfer function is

$$z(s) = \frac{10}{(2s^2 + 0.094s)}\Theta(s)$$

and below is showed the relative model :

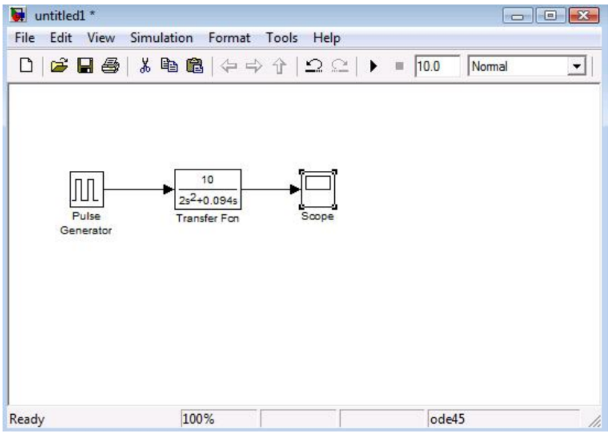


Figure 3.1: Altitude Simulink Model

and the system response:

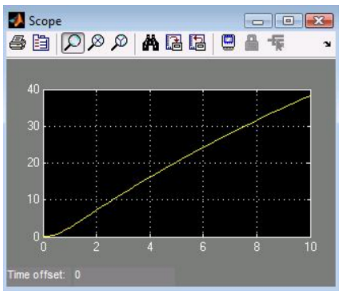


Figure 3.2: System response

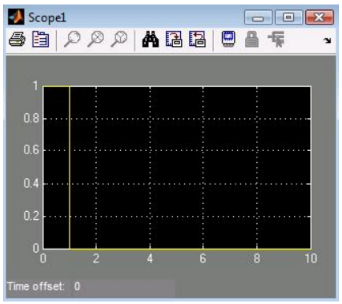


Figure 3.3: Pulse

As showed by the figure, the system is unstable, so if the system is excited by any input (perturbation) it will never come back to original position. Now it should be clear that the system must be controlled in order to keep desired position (altitude) or direction (angle between longitudinal axis and target) and to stabilize it against the perturbations.

### 3.1.2 Controlled altitude

In this section will be implemented a simple PID controller, the purpose is to show that the controller makes the system stable and it allow to reach the nal altitude (target).

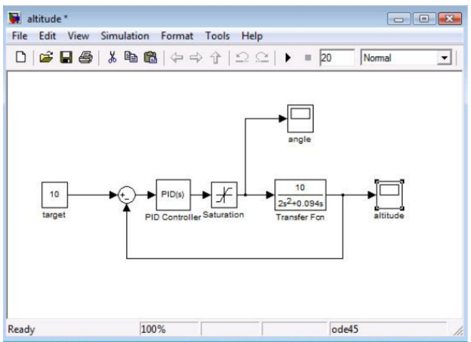


Figure 3.4: Simulink Controlled Altitude

Fig.3.5 shows the altitude and the engine angle that the controller have used in order to reach the target :

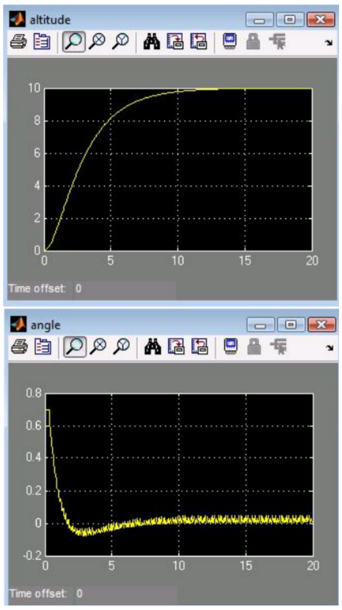


Figure 3.5: Altitude and engine angle

The constants of controller PID used for this example are 0.1 for the proportional term and 0.3 for the derivative. These constants are not obtained with some algebraic equations but by some dierent shooting , that isn't a rigorous method but it's used in most of pratical cases.

## 3.2 Yaw stability

As was done in the previous section the yaw's stability will be investigated, it will be showed that the system isn't stable like altitude's system, instead the control of yaw will be developed in the next lessons.

### 3.2.1 Simulink model

**Data**

- $J = m_1\frac{L^2}{4} + m_2\frac{L^2}{4} = m_{tot}\frac{L^2}{2} = 1Kg * m^2$
- $\rho$= $1.225Kg/m^3$
- $m_{tot}$ = 2 Kg
- $c_D$ = 0.3
- S =$\pi R^2$= $\pi * 0.2^2$ = $0.1257m^2$
- d = 0.1

Transfer function is

$$\psi(s) = \frac{0.1}{(s^2 + 0.0115s)}T(s)$$

where $T(s) = T_1(s) - T_2(s)$. The respective model is

 

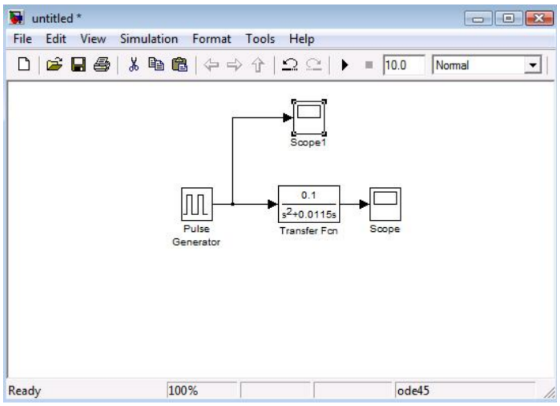


Figure 3.6: Simulink Yaw Model

The response is showed below

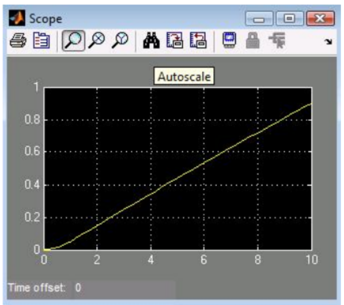


Figure 3.7: Pulse response

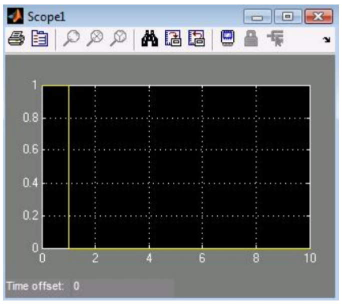


Figure 3.8: Pulse

As said at the beginning of this section this system is unstable like the altitude system, thus the control is necessary to stabilize it.

### 3.2.2 Yaw PID

As the altitude system, The yaw PID controller was implemented in order to reach a target angle, the constants of PID used for this example are obtained in the same manner of the altitude system by shooting . The Fig.3.9 shows the Simulink Model

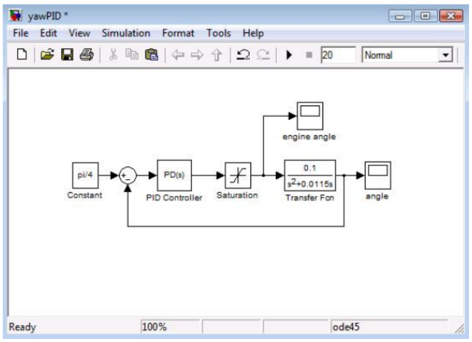


Figure 3.9: Yaw Simulink Model

the response is showed below

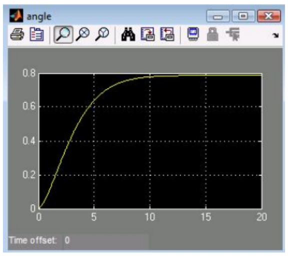


Figure 3.10: Yaw response

The PID constants are P=3, D=10.

# Chapter 4

# Direct integration

In this chapter a direct integration of equations of motion will be implemented on Simulink in order to show that the system is unstable also in time domain.

## 4.1 Altitude time domain

Rearranging altitude's equation of motion it become

$$\ddot{z}(t) = \frac{1}{m}(T\Theta(t) - 2\rho \dot{z}(t) S c_D)$$

and it can be implemented on Simulink as showed below

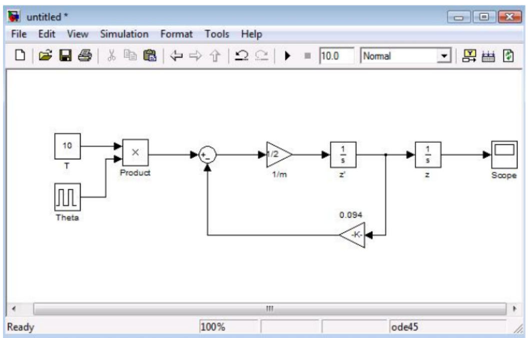


Figure 4.1: Altitude time domain integration

and the response is showed in Fig.4.2

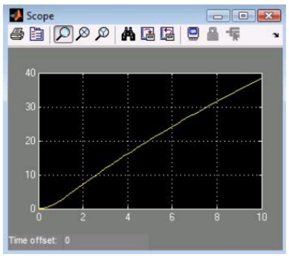


Figure 4.2: Altitude time pulse response

Comparing the response obtained with the Laplace transformation, it's easy to see that these are the same graphics, as showed below

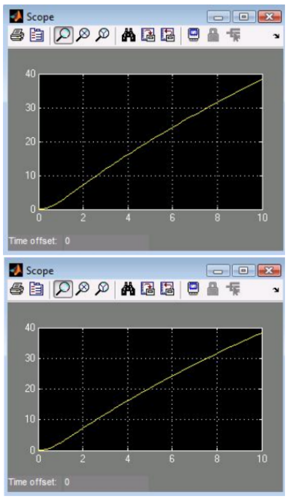


Figure 4.3: Direct integration response and laplace response

### 4.1.1 Yaw time domain

Yaw time domain equation of motion is

$$\ddot{\psi}(t) = \frac{1}{J}((T_1(t) - T_2(t))d - \rho S c_D \frac{L^2}{4}\dot{\psi}(t))$$

call $T = T_1 - T_2$

$$\ddot{\psi}(t) = \frac{1}{J}(T(t)d - \rho S c_D \frac{L^2}{4}\dot{\psi}(t))$$

the model is

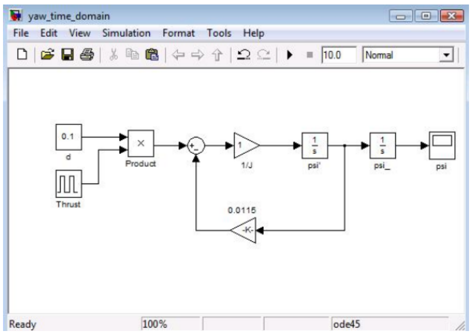


Figure 4.4: Yaw time domain integration

The Fig.4.5 shows the comparison of the responses

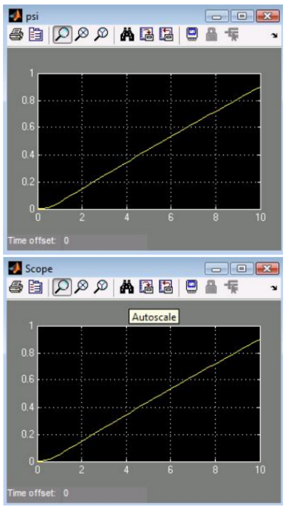


Figure 4.5: Yaw time domain integration

So the responses are the same like in the altitude system.

 