Assignment 2

Advanced Flight Mechanics 2021-22

Eduardo Andrés
Abolfazl Simorgh

A2. Navigation

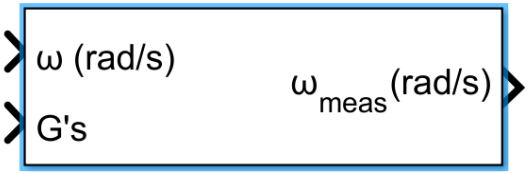
- Session 1: Sensors
- Session 2: Kalman Filter

Session 1

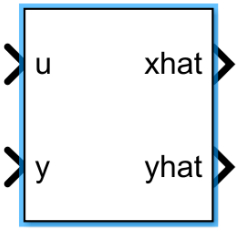
Sensors

1. Introduction and objectives

- Implement the linear model in Simulink
  - You will only need A_{long} and A_{lat} from Assignment1.
  - Excite the Phugoid and the Dutch Roll.
- Get measures from a 3-axis gyroscope.



- Filter the outputs with a Kalman Filter.



- Compare the ideal, the measured and the filtered angular rates.

2. Linear model (I)

One model for longitudinal and lateral dynamics.

- Dynamics $\dot{\mathbf{x}} = A\mathbf{x}$
- Output $\mathbf{y} = C\mathbf{x}$

Such that:

- $\mathbf{x} = \begin{bmatrix} u & w & q & \theta & v & p & r & \psi \end{bmatrix}^T$
- $\mathbf{y} = \begin{bmatrix} p & q & r \end{bmatrix}^T$

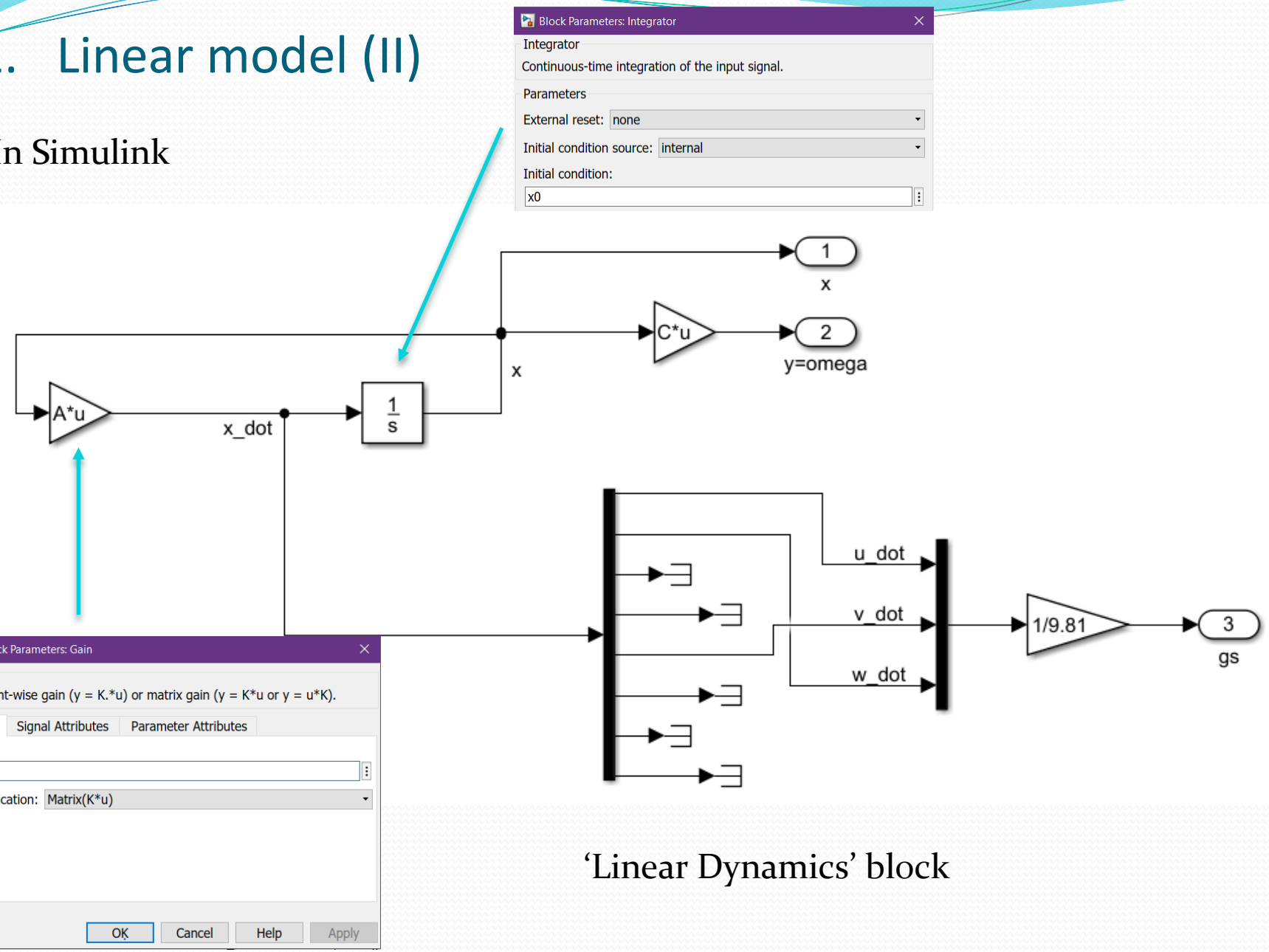
Then

$$A = \begin{bmatrix} A_{long} & 0_{4\times4} \\ 0_{4\times4} & A_{lat} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

2. Linear model (II)

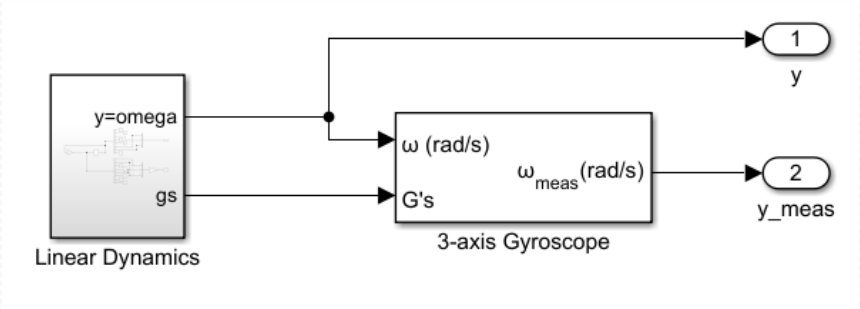
In Simulink



'Linear Dynamics' block

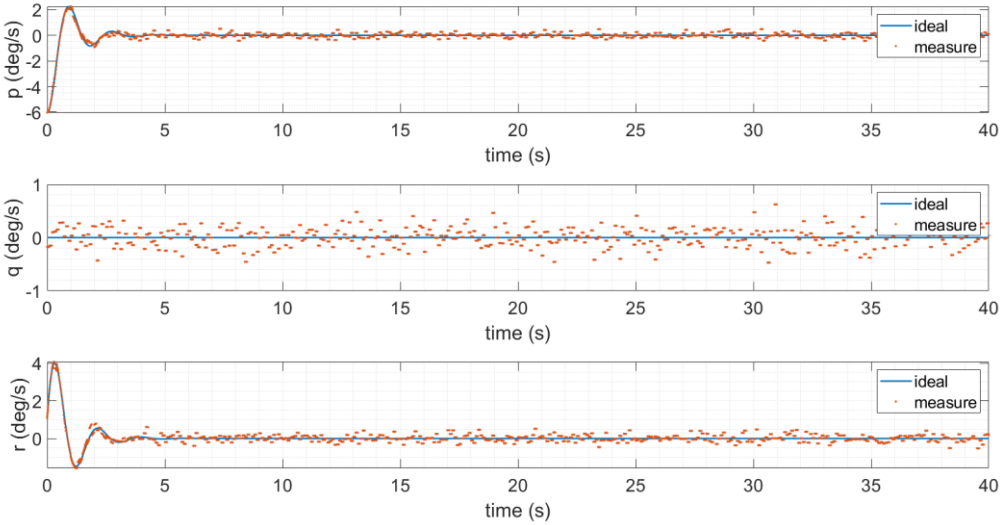
3. Gyroscope

- From the linear model:
  - $\omega = \mathbf{y} = \begin{bmatrix} p & q & r \end{bmatrix}^T$
  - $G's = \frac{1}{g}\begin{bmatrix} \dot{u} & \dot{v} & \dot{w} \end{bmatrix}^T$

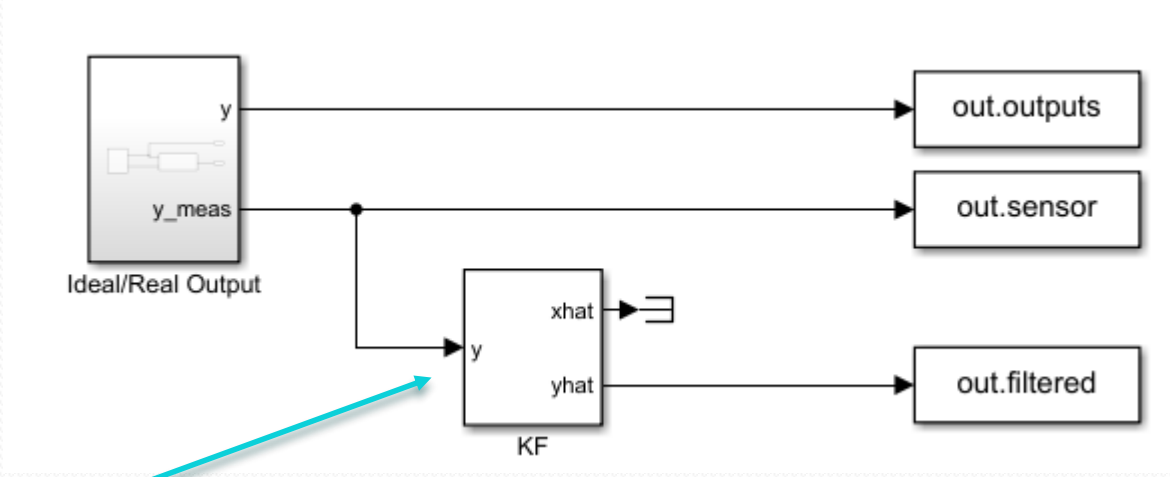


'Ideal/Real Output' block

- Gyroscope settings:
  - In 'main', first order dynamics.
  - In 'noise', *noise power* $= (10^{-6}, 10^{-6}, 10^{-6})$



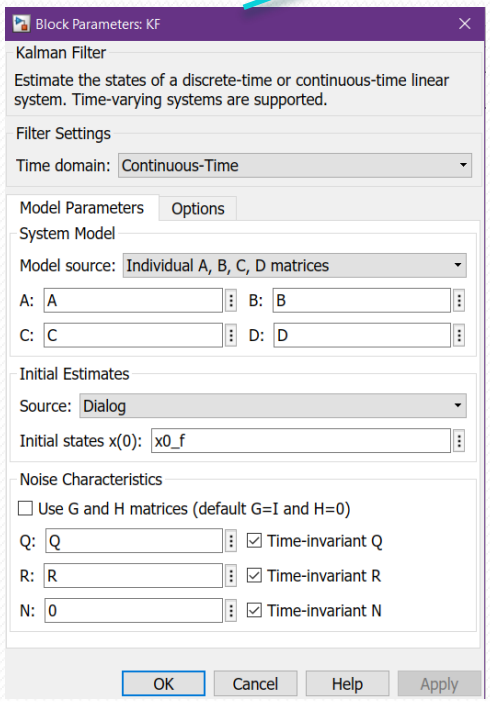
4. Kalman Filter

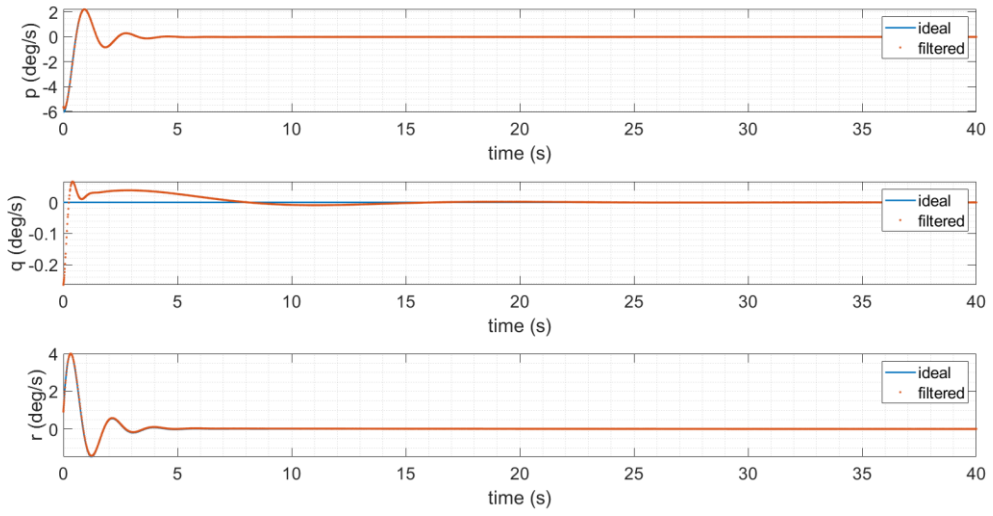


Ideal dynamics **y**

Measures, observations

Estimation $\hat{\mathbf{y}}$





5. Tips for Simulink

- Always run the model from Matlab.
- Do not include numerical values inside Simulink blocks. Use variables from your Matlab workspace. It is easier to assign values from there.
- Be careful with vectors, maybe you need a column instead of a row.
- Use 'ToWorkspace' block to export results to Matlab and work with them.

Matlab Code:

```
out = sim('model', t);
time = out.tout;
p_real = out.sensor.Data(:, 1);
q_real = out.sensor.Data(:, 2);
r_real = out.sensor.Data(:, 3);
```