Crane Damping

Applied Robotics - FRTF20

HT-2020 Group10

Hicham Mohamad

Ola Nilsson

Crane Damping Project

- System Overview
- Model Description
- Control Design Approach
- Angle Estimation using IMU
- Network Connection

System Overview

System consists of a robot arm that is driven by:

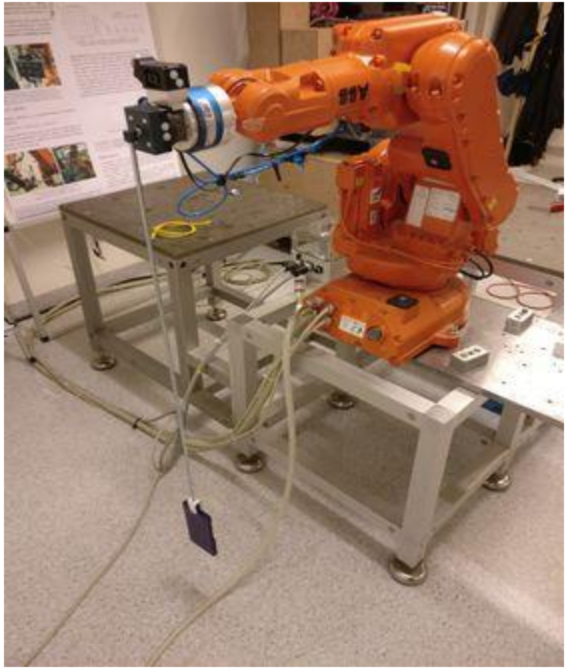
- An **ABB IRB140** robot with 6 joints/links
- A free swinging **pendulum** attached to the pivot point

IRC5: Embedded main-computer + PID controller

Dc motors at the 6 joints

Mobile phone tracker added to the system

IMU app called IMU+GPS-Stream to gather information



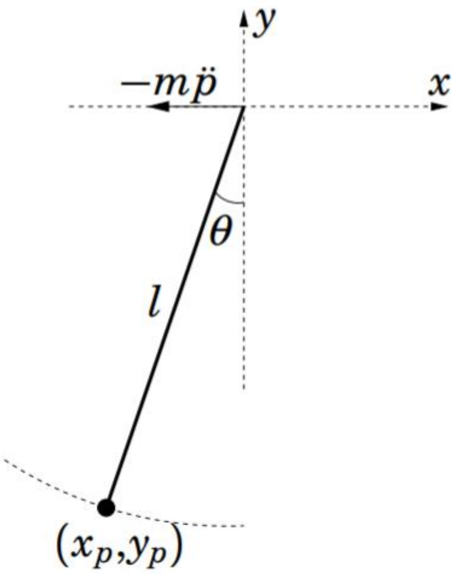
Project Objective

Damping the load swinging in a planar orbit:

- The main task of our project is to control ABB140 robot arm such that it will be able to damp the oscillations of the swinging pendulum

- using the ABB controller **IRC5** and the LTH-made sensor software **interface ExtCtrl**

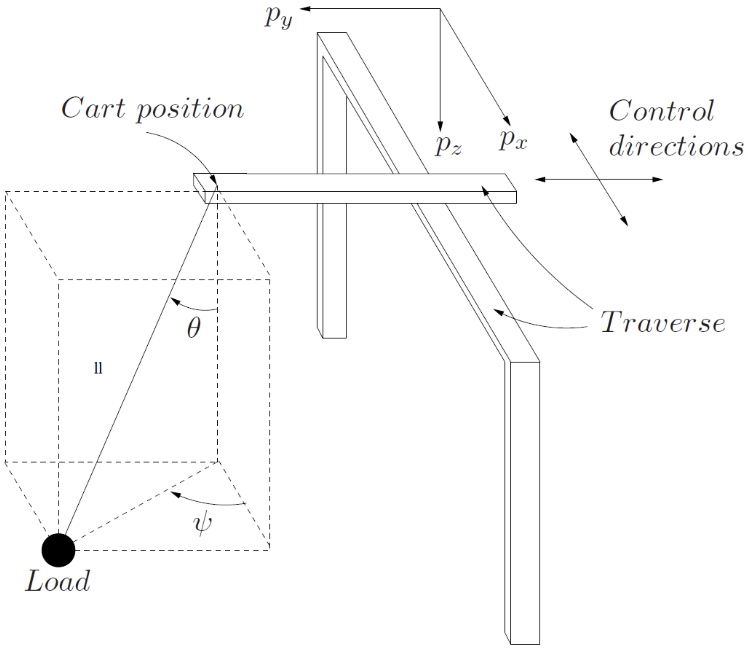
- Alternative: Circular swinging process



Modeling of the Crane Process

The Cart-pendulum model

- Our process can be modelled as a **Pendulum** mounted to a **motorized cart**.
- The **cart motion** tries to keep the leaning angle, θ, at a small level and damp the swinging pendulum.
- The model should be close to the **real process behaviour**.

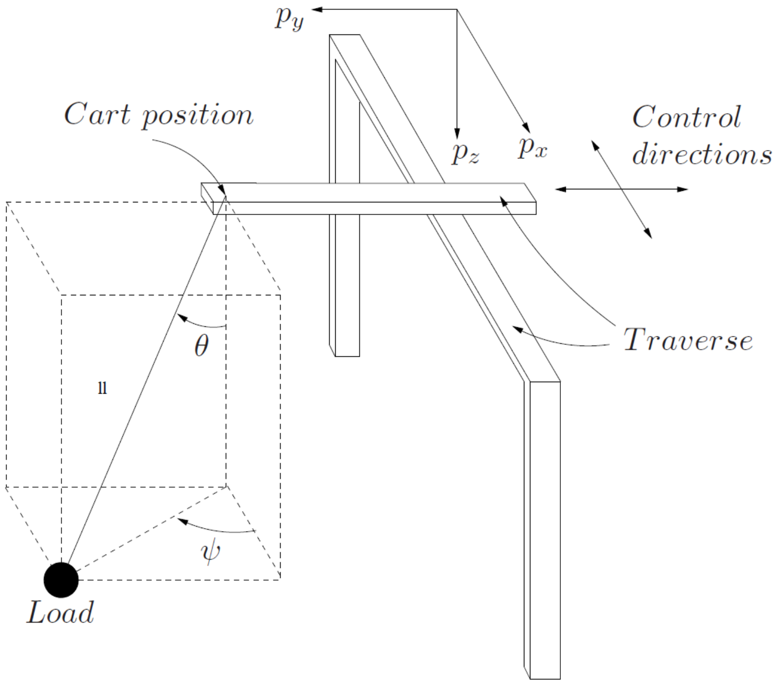


Characteristics of The Physical System

- Pendulum is **oscillatory** without control.

- Dynamics of the pendulum system is **nonlinear.**

✓ **Pendulm oscillates back and forth and can be damped by a controller, i.e.** The acceleration of the DC motors tries to keep it at the downward vertical position at **zero degree** by acting on the tilt-angle θ



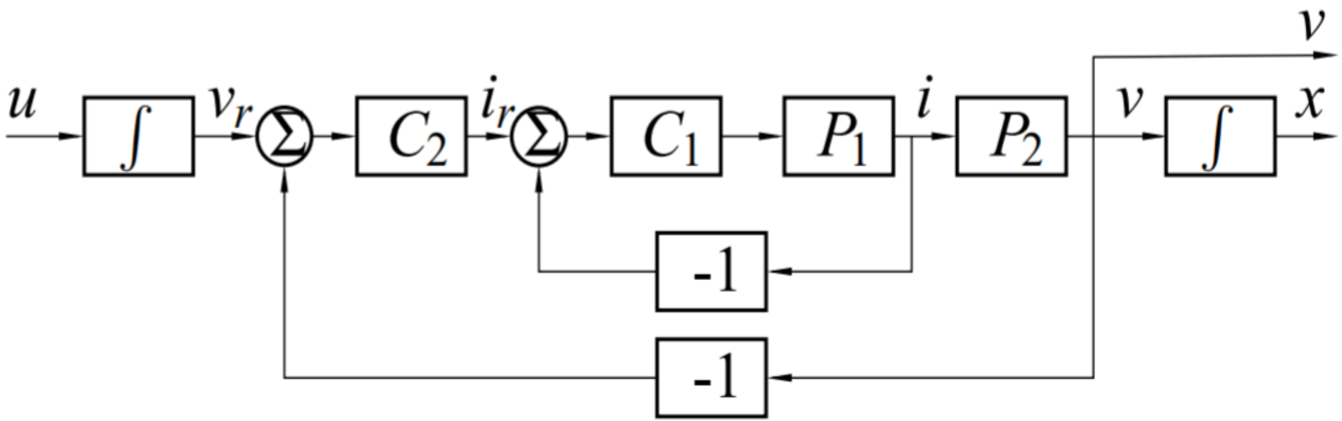
Modelling the Cart / Cascaded structure

DC-motor is controlled in a cascaded structure

Cart dynamics can be modeled as a **double integrator** from acceleration reference *u* to the position *p* of the robot arm: **Inner loop** & **Outer loop**

Double integrator

$$\ddot{p} = a_{ref} = u$$

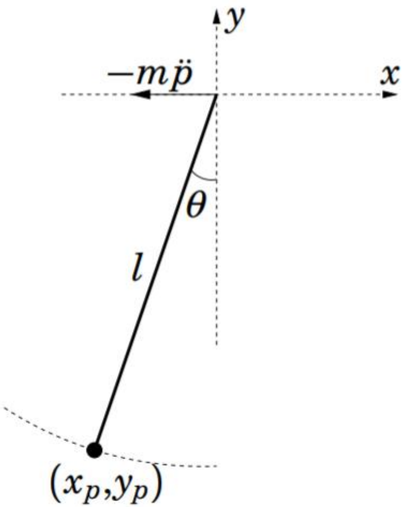


Kinetic and Potential Energy - Pendulum System

Kinetic and potential energy in the generalized coordinate

$$K = \frac{1}{2} m l^2 \dot{\theta}^2$$

$$P = -mgl\cos(\theta)$$



Lagrangian

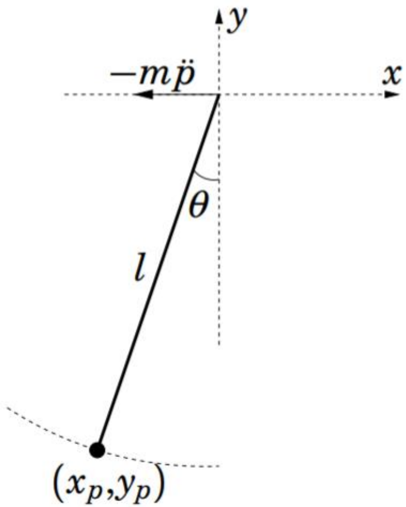
$$L(\theta, \dot{\theta}) = K - P$$

Dynamic Equations of Pendulum System

Euler-Lagrange 's Law gives the dynamic equation for the pendulum

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau = F_x \frac{\partial r_x(\theta)}{\partial \theta} = m\ddot{p}l\cos\theta$$

$$\ddot{\theta} = \frac{-g}{l}\sin\theta + \frac{\ddot{p}}{l}\cos\theta$$

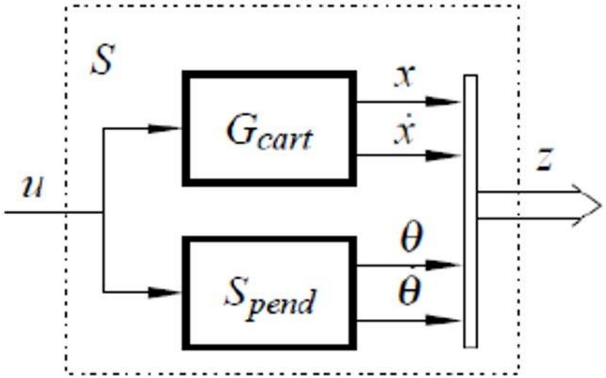


Resulting Nonlinear Model – Complete System

The two governing equations for the cart-pendulum system

$$\ddot{p} = a_{ref} = u$$

$$\ddot{\theta} = \frac{-g}{l} \sin\theta + \frac{\ddot{p}}{l} \cos\theta$$



Linearized Pendulum equation of motion

Linearization: The model can be linearized by substituting the approximations in the downward position, around $\theta = 0$

$$\cos\theta \approx 1$$
$$\sin\theta \approx \theta$$

$$\ddot{\theta} = -\frac{g}{l}\theta + \frac{u}{l}$$

Validation: This assumption can be reasonably valid if we design the controller such that the pendulum do not deviate too much from the vertically downward equilibrium position.

The Full State Space System

Linearization of the system dynamics, results in the following state space model

$$z = (z_1, z_2, z_3, z_4)^T = (p, \dot{p}, \theta, \dot{\theta})^T$$

$$\dot{z} = Az + Bu = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -g/l & 0 \end{pmatrix} z + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1/l \end{bmatrix} u$$

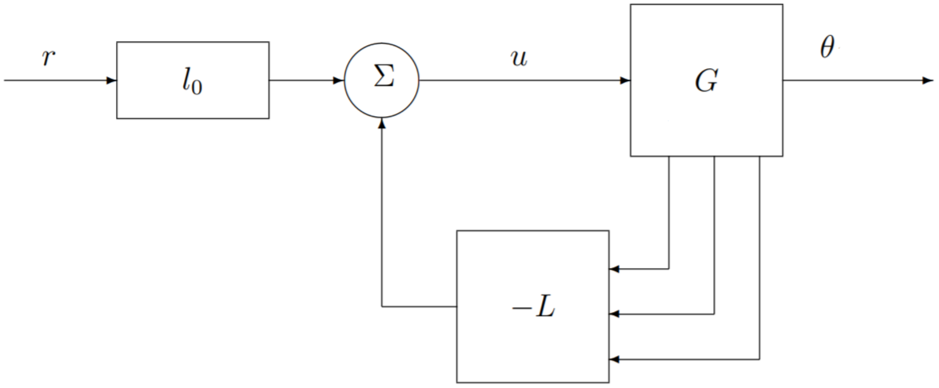
$$y = \begin{pmatrix} p \\ \theta \end{pmatrix} = Cz + Du = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} z + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Linear State Feedback Design – LQ Control

Optimal Feedback Law ***u = - L x***

- Our control design is based on **the linear model**

- **LQR design:** computing the optimal gain vector L that minimizes the cost function J given the weights Q and R:

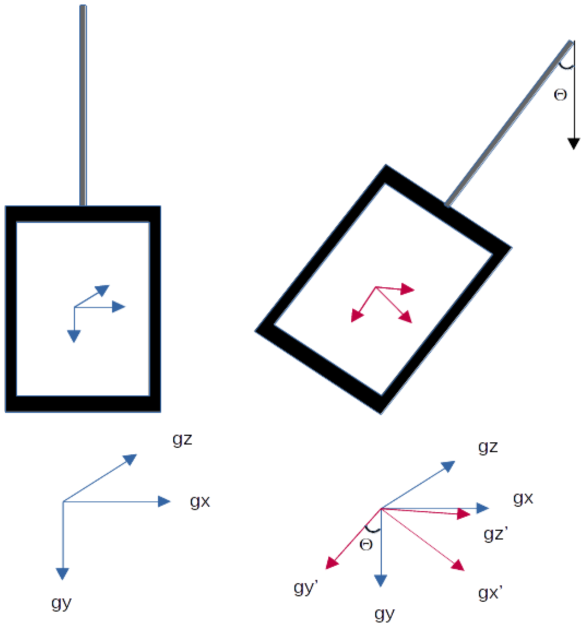


$$J = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t))dt = \int_0^\infty (x^T(t)Qx(t) + u^2(t))dt$$

Angle Estimation using IMU

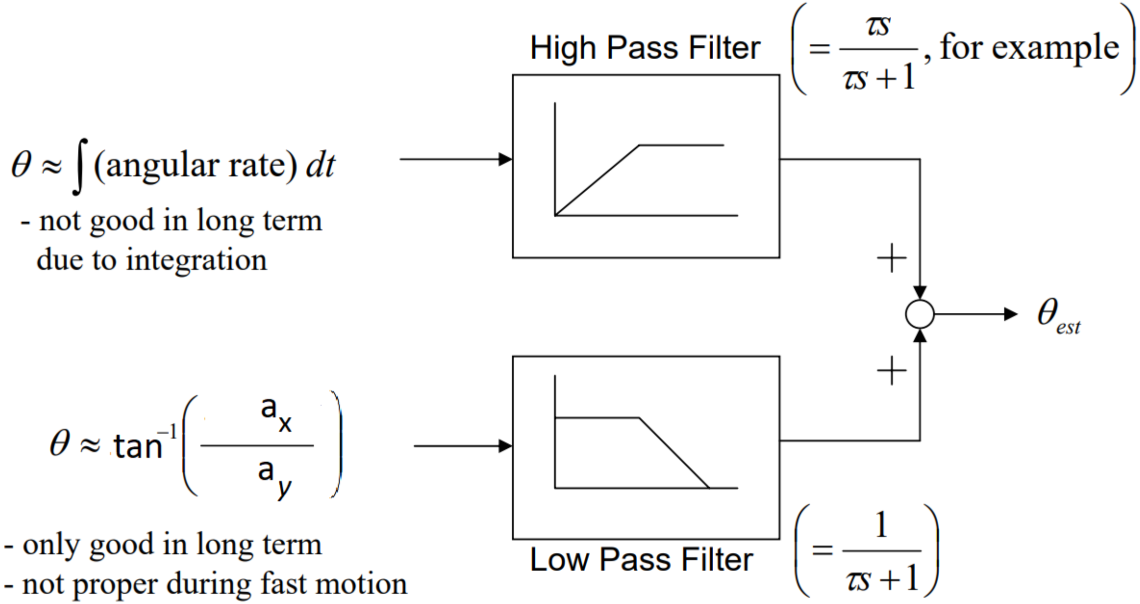
By extracting information from both the accelerometers and the gyro

Sensor fusion: Combining these using **low-pass** and **high-pass** filters, we will create a better estimate of the angle of the pendulum, compared to a naive approach of using only the gyro, or only the accelerometer.



Angle Estimation IMU - Complimentary Filter

Better estimate by using sensor fusion with Complementary Filter



Simulink, Python, C and Orca/EXTCTRL

Network communication to the robot control system

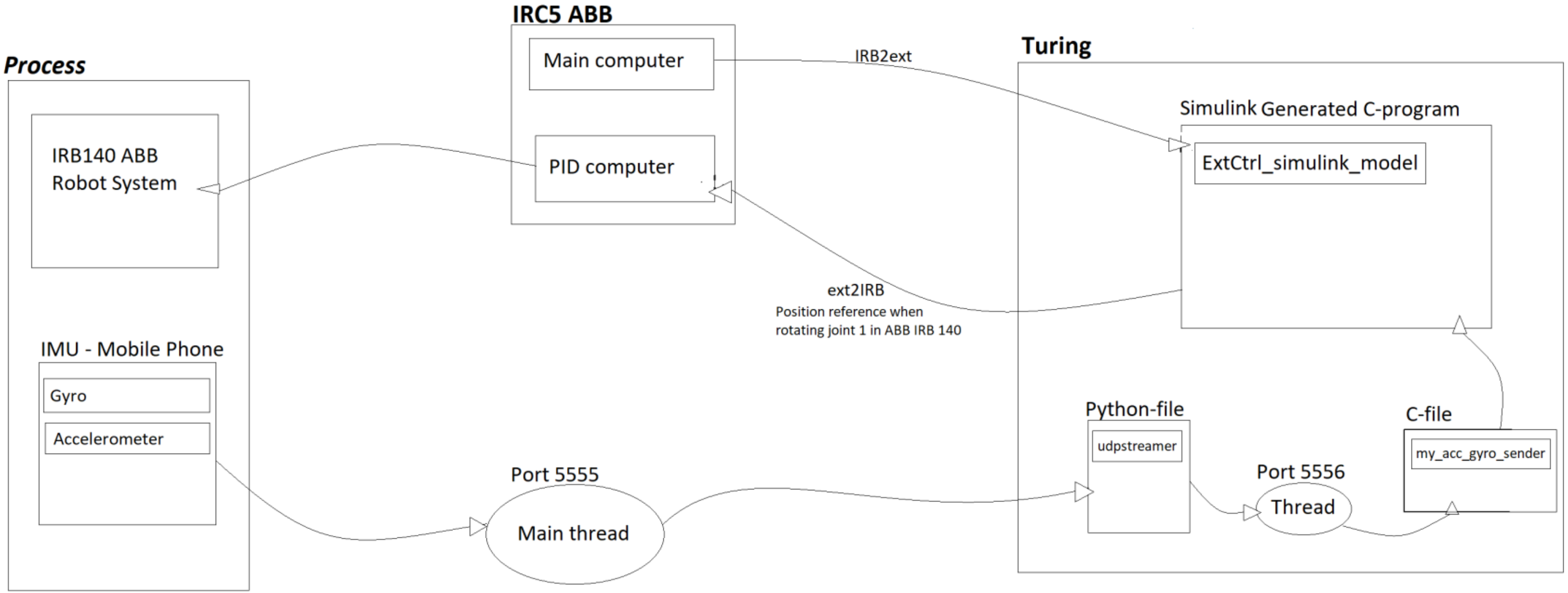
- The model and the optimization problem are described in Maltlab/Simulink
- Problem solved by a third-party LTH-made Simulink **ExtCtrl**
- Python and C is the interface that let us communicate to **ExtCtrl** and plot the result

Communication setup / Network Connection

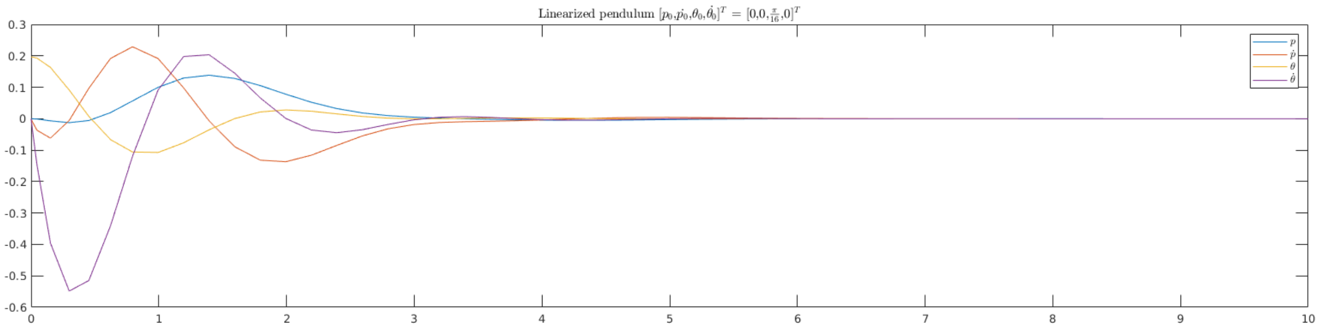
Three parts necessary in order to set up a proper communication to the robot:

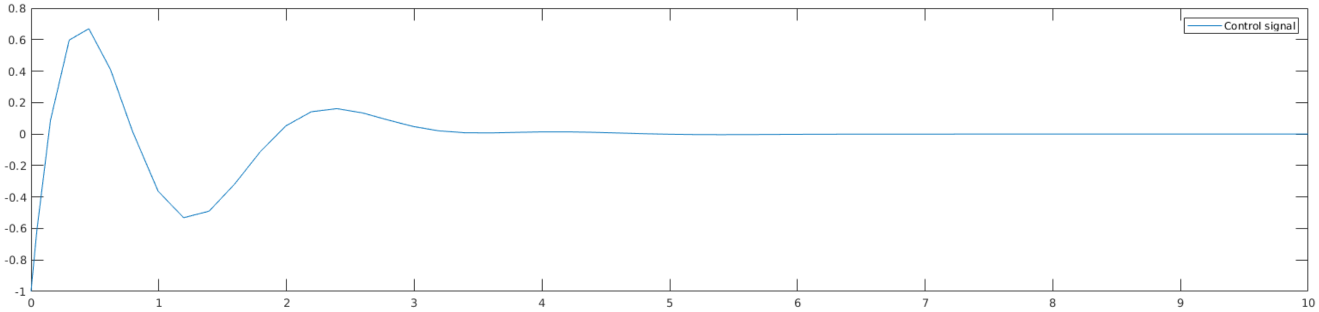
- ***.lc-file** to define the variables to send and recieve on network
- **Matlab/Simulink model** sets up the connection between Orca and **ExtCtrl**
- ***.c-file** to manipulate signals or introduce measurments devices

Communication setup with the robot

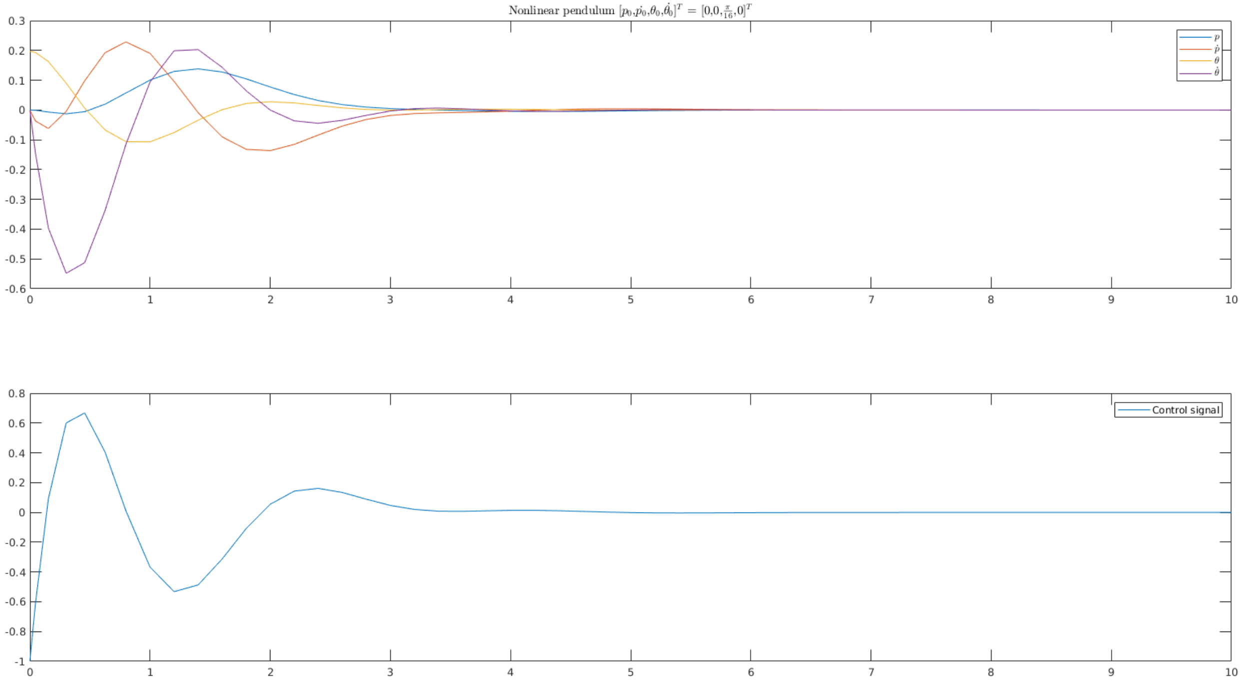


Linear simulation

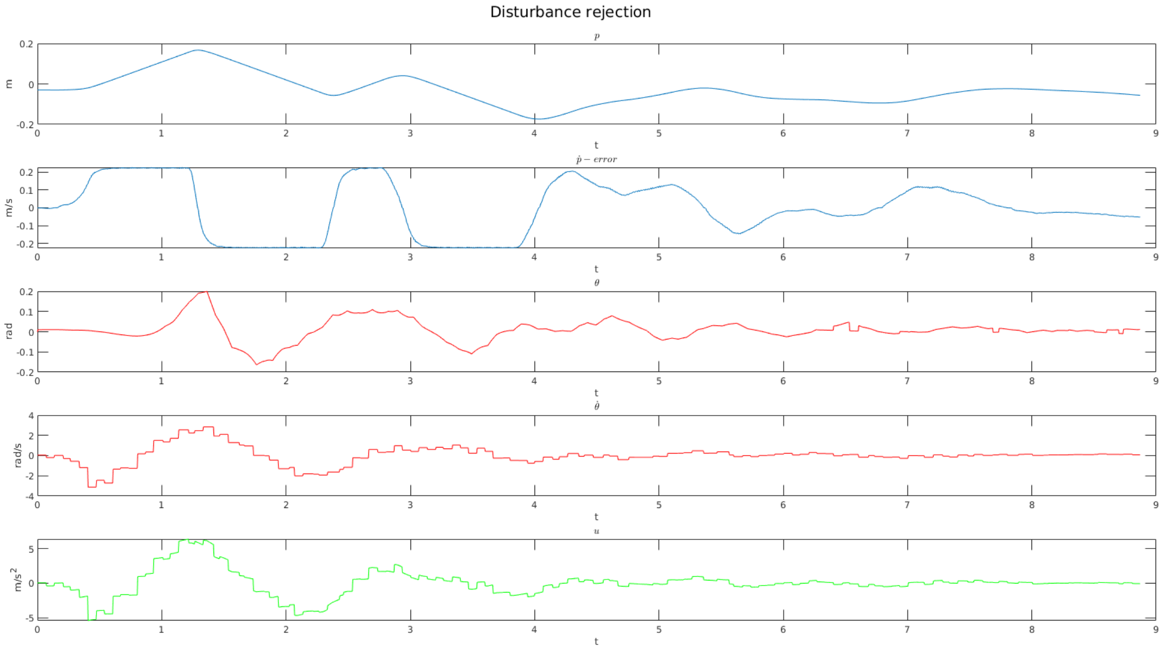




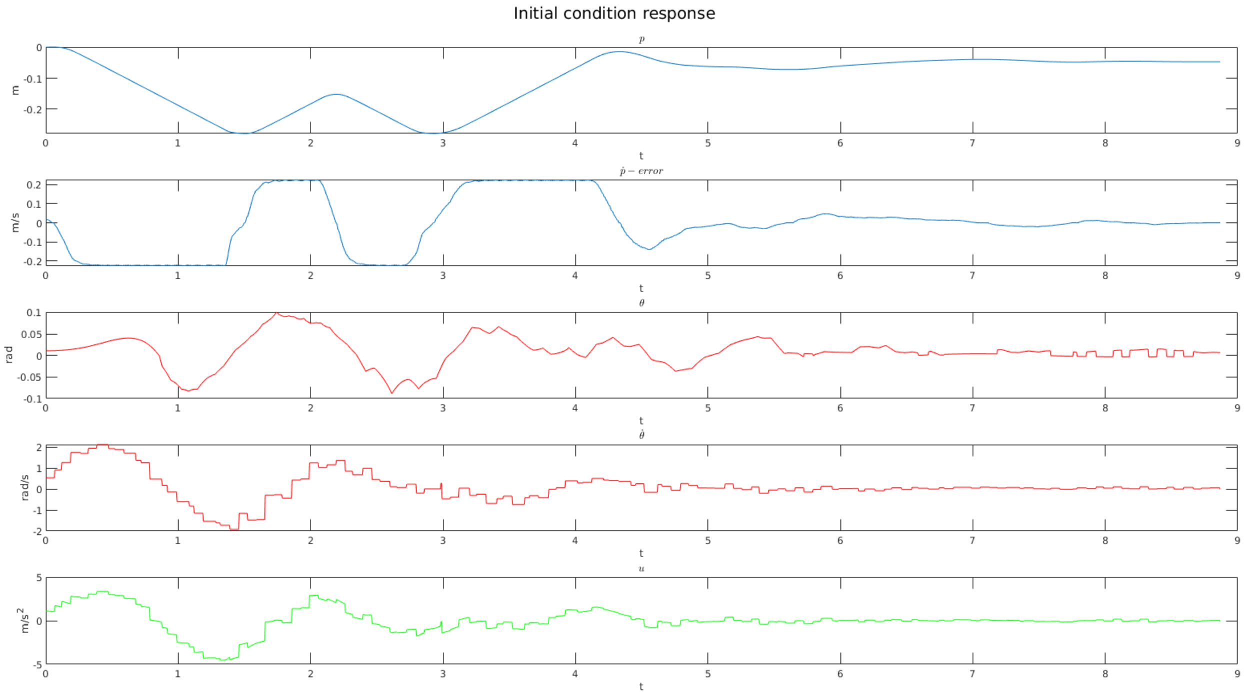
Nonlinear simulation



Distubance Rejection



Initial value response



Questions?

Crane Damping – Group 10