Automatic Control Project

2005 / 2006

The Pendubot

Aurélie Dufour
Erik Henriksson
Eric Blanquer
Jacob Riback
Benjamin Mairesse

Final Presentation

Modeling

Control

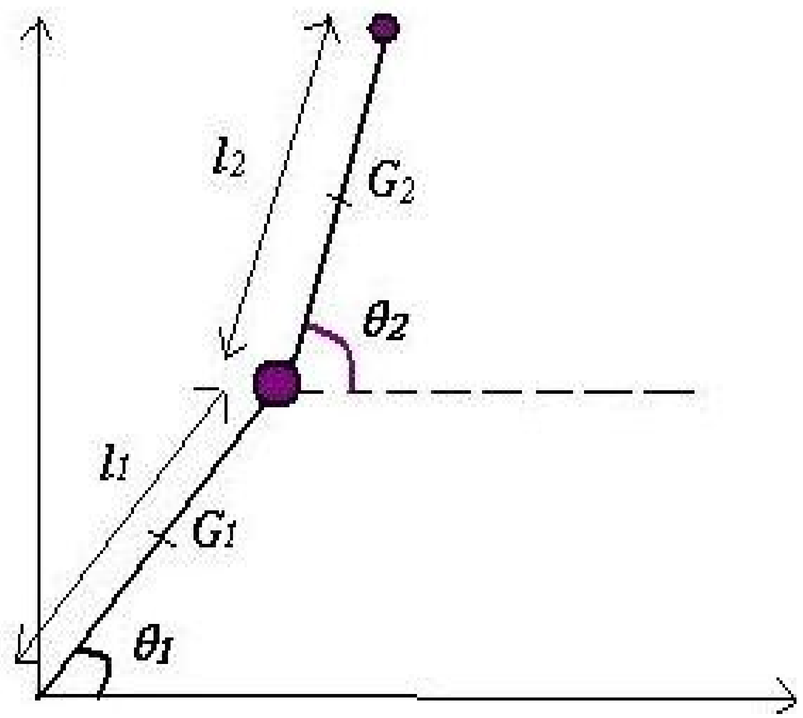
Implementation

Objectives with the project

- Stabilize in up-up and down-up
- Swing-up
- Peripheral movement
- Safety net

Description of the Pendubot

- One motor
- Two links
- Two encoders



Modeling

- Lagrange method → Energy based method:

$$L = E_k - E_p$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \sum Non\ conservative\ torques$$

- $\{q_i\}$ is a set of parameters that entirely describes the system.

PENDUBOT

State-Space Form

$$\ddot{\theta}_1 = \frac{1}{\frac{B_1 A_2}{B_2}\cos^2(\theta_1 - \theta_2) - A_1}\left[\frac{A_2 B_3}{2B_2}\sin\left(2(\theta_1 - \theta_2)\right)\dot{\theta}_1^2 + A_3\sin(\theta_1 - \theta_2)\dot{\theta}_2^2 - A_2\frac{B_4}{B_2}\cos\theta_2\cos(\theta_1 - \theta_2) + A_4\cos\theta_1 - \tau\right]$$

$$\ddot{\theta}_2 = \frac{1}{\frac{B_1 A_2}{A_1}\cos^2(\theta_1 - \theta_2) - B_2}\left[-B_3\dot{\theta}_1^2\sin(\theta_1 - \theta_2) - B_1\frac{A_3}{2A_1}\sin\left(2(\theta_1 - \theta_2)\right)\dot{\theta}_2^2 - B_1\frac{A_4}{A_1}\cos\theta_1\cos(\theta_1 - \theta_2) + B_4\cos\theta_2 + \frac{B_1}{A_1}\cos(\theta_1 - \theta_2)\tau\right]$$

Linearization

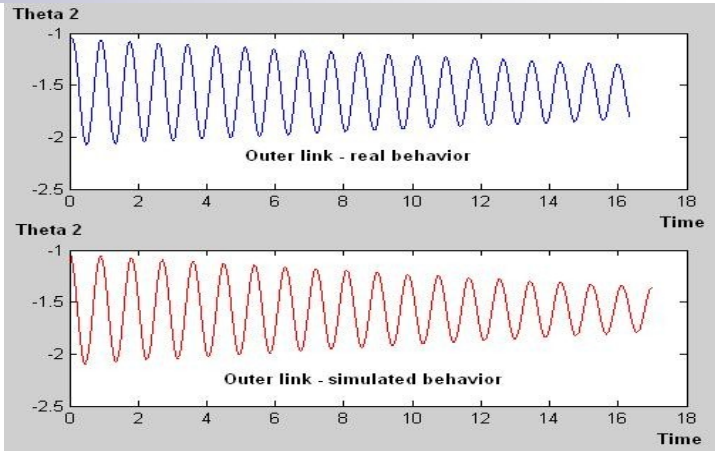
- Linearized equations: $\Delta\dot{X} = A\Delta X + B\Delta U$

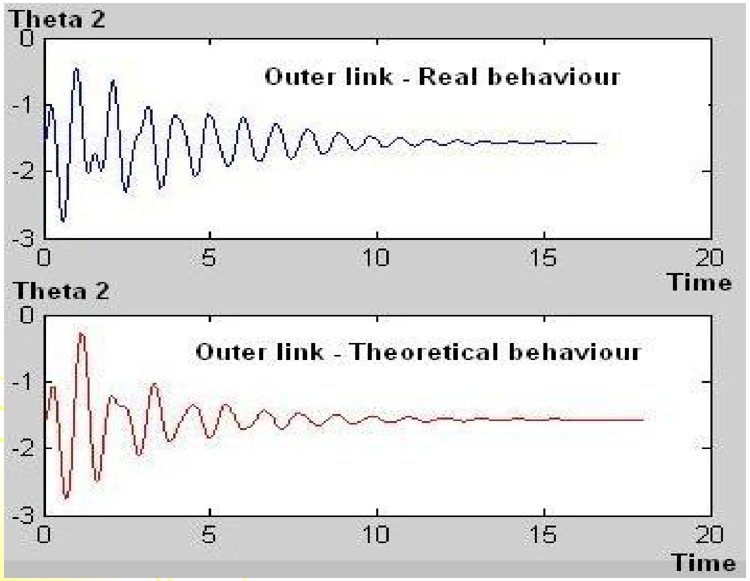
  - Where the matrices A and B were computed using Maple as follows:

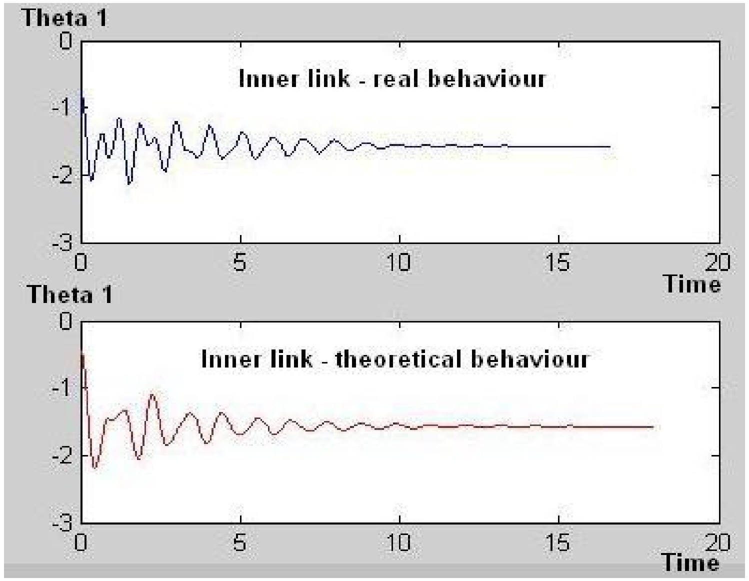
$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial \dot{\theta}_1} & \frac{\partial f_1}{\partial \dot{\theta}_2} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \frac{\partial f_2}{\partial \dot{\theta}_1} & \frac{\partial f_2}{\partial \dot{\theta}_2} \end{bmatrix}_{[\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^T = [\theta_{1eq}, \theta_{2eq}, \dot{\theta}_{1eq}, \dot{\theta}_{2eq}]^T} \qquad B = \begin{bmatrix} 0 \\ 0 \\ \frac{\partial \tau}{\partial \tau} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Model validation

- **No satisfying friction model found.**
- **Chaotic system.**
- **Model good enough for control purpose.**







Control strategy

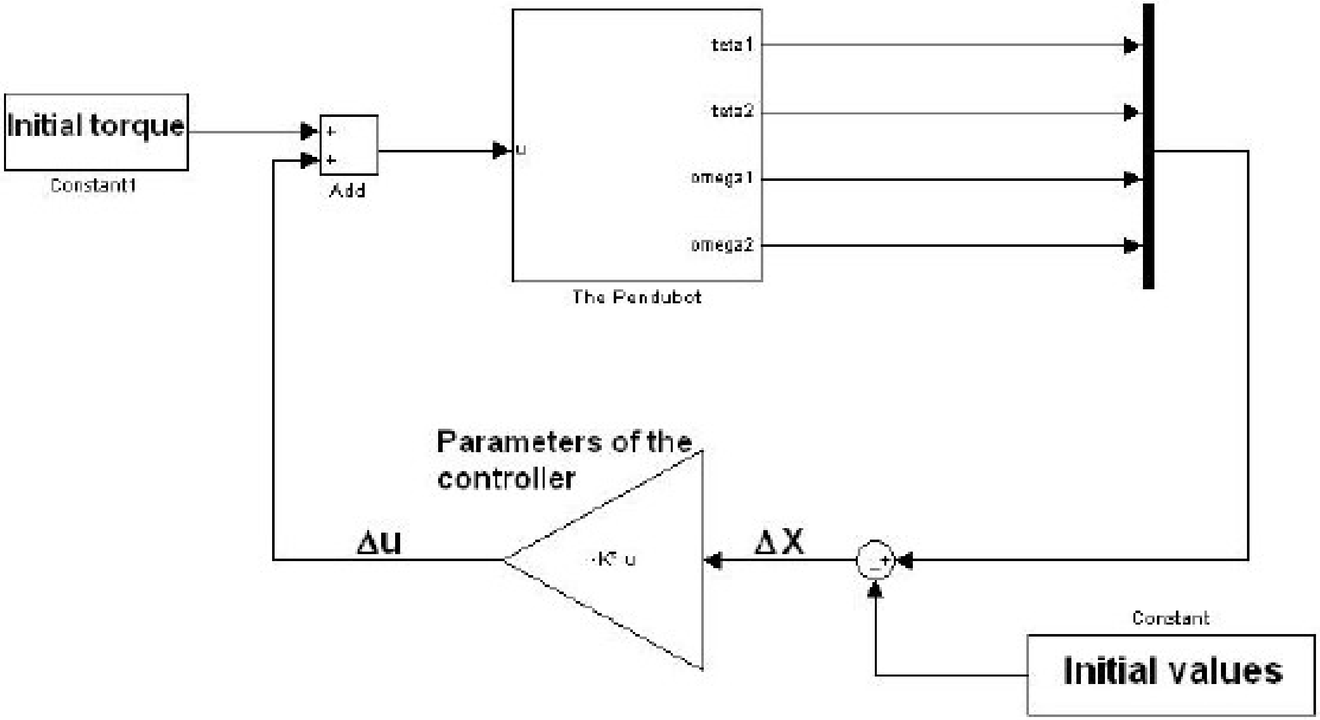
- Before implementation with *LabView*, the strategies were developed using *Matlab* & *Simulink*

- Balancing control
- Swing-up control
- Safety net
- Switched system

PENDUBOT

Balancing control

- LQ approach:
  - Linearization + State-feedback



PENDUBOT

Partial feedback linearization

- *Underactuated* system:
  → Partial feedback linearization

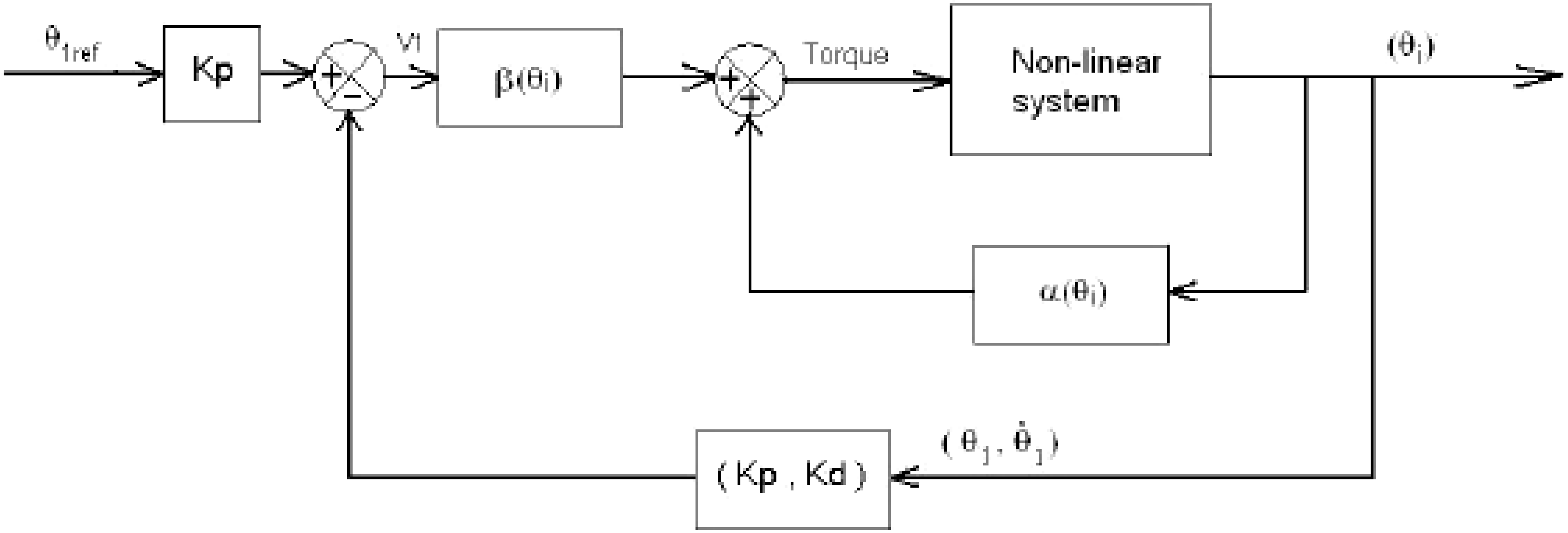
- Feeding the system with the input: $\tau = v_1\beta(\theta_i) + \alpha(\theta_i)$
  gives the new following set of equations:

$$\left\{ \begin{array}{c} \ddot{\theta}_1 = v_1 \\ \\ \ddot{\theta}_2 = \frac{1}{B_2}\left[\theta_1^2 A_2 \sin(\theta_1 - \theta_2) - B_4\cos(\theta_1 - \theta_2) - v_1 A_2 \cos(\theta_1 - \theta_2))\right] \end{array} \right\}$$

PENDUBOT

Global Scheme

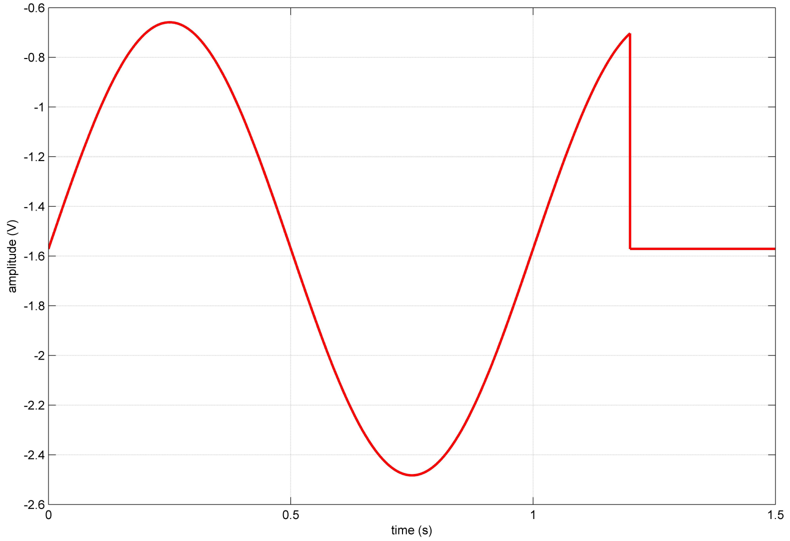
- An inner-loop to linearize
- An outer-loop to control



Swing-up

- Partial feedback linearization to control the *inner link*
- PD-control tuned to handle the *outer link*

- Suitable reference signals (static and dynamic)



PENDUBOT

Safety net

- Engage an emergency procedure when the system is *unsafe*

  unsafe = supply voltage > 9.9 Volts (10 Volts Max)

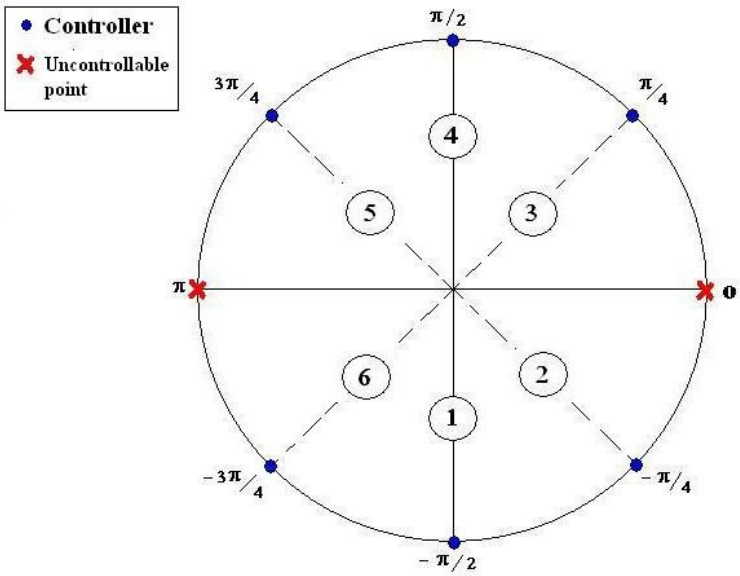
- Force the inner link to point downwards and catch it with a balancing controller

Switching areas - inner and outer links -

PENDUBOT

Switched system

- For peripheral movement
- Multiple sectors needed
- One controller per sector
- Final control system is hybrid automata



PENDUBOT

Crossing the uncontrollable state

- Go as close as possible to the uncontrollable state with a balancing controller
- ‘Jump’ using the partial feedback linearization
- Catch the Pendubot with a new balancing controller

PENDUBOT

Implementation

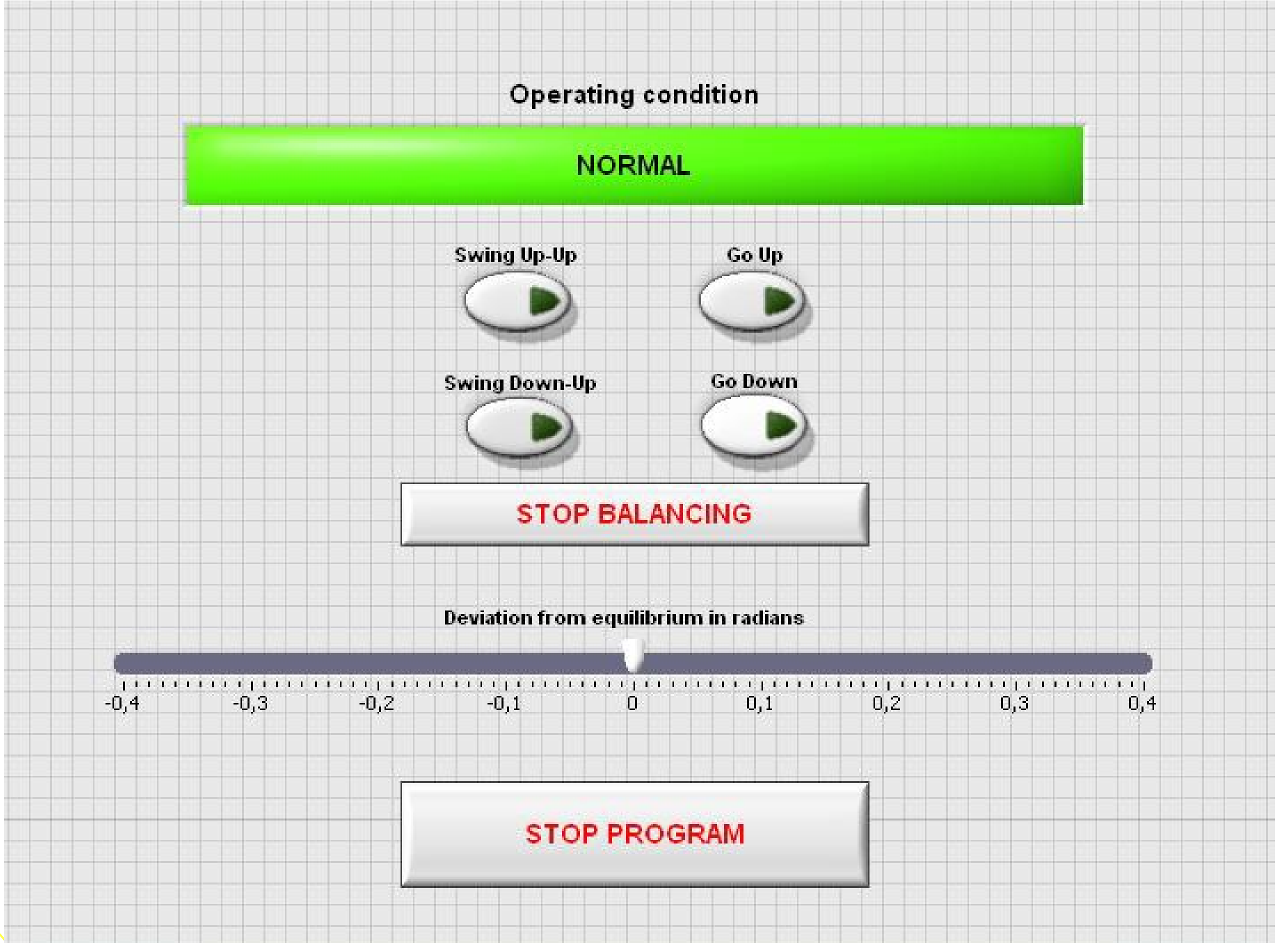
- Using LabView
- 100 Hz Sampling frequency
- Program consists of three main building blocks

PENDUBOT

Overall structure

- Flow chart

Front Panel



PENDUBOT

Results

- All the ordered functionality is implemented and working.
- Documentation is handed for comments.
- Software is ready for distribution.

Conclusions

- The group had good knowledge entrance.
- Feedback linearization is the key to success.
- Friction is very hard to model.

Conclusions cont.

- Implementation is the large part. About 40% of the total time spent.
- Large difference between having all parts of a program working and having a working program.

PENDUBOT

Questions