

Adaptive Friction Compensation

Adam Jalkemo `adam@jalkemo.se`

Alexander Israelsson `israelsson.alexander@gmail.com`

Emil Westenius `emil@westenius.se`

Jonathan Andersson `mat11ja1@student.lu.se`

Supervisor: Gabriel Ingesson

May 13, 2016

1 Introduction

In this project a Furuta pendulum process will be used. The Furuta pendulum consists of a pendulum, that is free to rotate in the vertical plane, attached to the end of an horizontally rotating arm that can be controlled. The pendulum will be stabilized in the inverted position using a swing up controller and a top controller. When controlling the pendulum in the top position limit cycles will occur due to friction. By compensating for the friction we aim to stabilize the pendulum, the friction will be estimated using an adaptive method. To do this a Java controller and interface will be implemented and run on a linux computer.

2 Program structure

For our suggested solution we will need one thread for the GUI and one for the controller. To communicate with the Furuta process we will use the analog box normally used during the control labs at LTH. The realtime package from Automatic Control at LTH will be used for communication in the GUI and controller implementation. It will be important to synchronize the controller calculations with the user inputs. A general overview can be seen in figure 1.

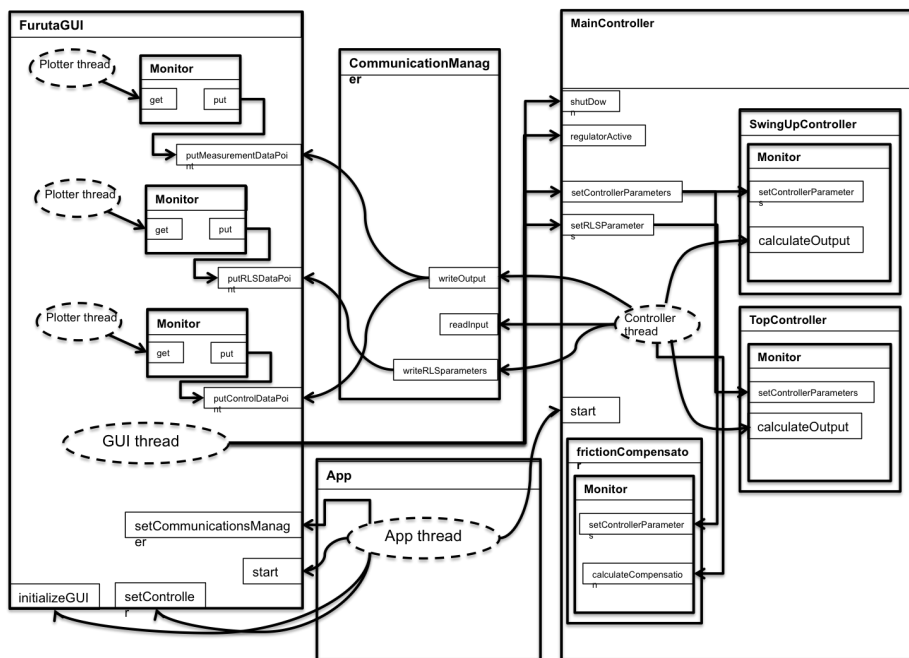


Figure 1: Overview of the program structure

3 System model

Due to the non-linear dynamics of the rotating arm and pendulum the modelling of the process becomes complicated. We have used the model given in Lab2 in "Non-linear control" where the model have been simplified with the help of Lagrange theory. The dynamics of the model is then given by:

$$(J_p + Ml^2)(\ddot{\theta} - \dot{\varphi}^2 \sin \theta \cos \theta) + Mrl\ddot{\varphi} \cos \theta - gl(M + m/2) \sin \theta = 0$$

$$Mrl\ddot{\theta} \cos \theta - Mrl\dot{\theta}^2 \sin \theta + 2(J_p + ml^2)\dot{\theta}\dot{\varphi} \sin \theta \cos \theta + (J + mr^2 + Mr^2 + (J_p + ml^2) \sin^2 \theta)\ddot{\varphi} = u \quad (1)$$

where θ is the angle of the pendulum, φ is the angle of the arm and u is the motor torque on the arm. The states θ , $\dot{\theta}$, φ and $\dot{\varphi}$ will all be measured from the process and u will be our control signal. The approximated coefficients are also taken from Lab2 and are given by:

$$l = 0.413m \quad r = 0.235m$$

$$M = 0.01kg \quad J = 0.05kgm^2$$

$$J_p = 0.0009kgm^2 \quad m = 0.02kg$$

$$g = 9.8$$

Where l is the length of the pendulum, r is the length of the arm, M is the mass of the weight, J is moment of inertia for pendulum, J_p is the moment of inertia for the arm, m is the mass of the pendulum and g is the gravity constant.

3.1 Linearization

The model in 1 is then linearized around

$$x = (\theta \quad \dot{\theta} \quad \varphi \quad \dot{\varphi}) = (0 \quad 0 \quad 0 \quad 0) \quad (2)$$

which yields

$$\dot{x} = Ax + Bu = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{bd}{ab-c^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-cd}{ab-c^2} & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ \frac{-cg}{ab-c^2} \\ 0 \\ \frac{ag}{ab-c^2} \end{pmatrix} u \quad (3)$$

where

$$a = J_p + Ml^2 \quad b = J + Mr^2 + mr^2$$

$$c = Mrl \quad d = lg(M + m/2)$$

This corresponds to the arm fixed at an angle corresponding to zero and the pendulum fixed in the top position. It is in this position that the friction will create oscillations and where the adaptive friction compensation will be applied.

3.2 Discretization

The linearized model is then discretized using ZoH. A simplified ZoH method have been implemented in java which makes it possible to change the sample time online. For the most part a sample time of $h = 0.01$ have been used since this suited the system best. With $h = 0.01$ the ZoH discretization of the model yields in

$$x_{k+1} = A_d x_k + B_d u_k = \begin{pmatrix} 1.0016 & 0.01 & 0 & 0 \\ 0.3133 & 1.0016 & 0 & 0 \\ 0 & 0 & 1 & 0.01 \\ -0.0059 & 0 & 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} -0.0036 \\ -0.7127 \\ 0.0096 \\ 1.9125 \end{pmatrix} u_k \quad (4)$$

4 Controller

There are six measurement signal available for the controllers to use. Two for the position θ and velocity $\dot{\theta}$ of the pendulum each (one has higher resolution and is used at the top position) and one for the arm position φ and velocity $\dot{\varphi}$.

4.1 Swing up

For the swing up of the pendulum a Lyapunov based controller is used. For the Lyapunov candidate the total energy (kinetic and potential) of the pendulum will be considered and is given by

$$E = Mgl(\cos\theta - 1) + \frac{J_p}{2} \dot{\theta}^2 \quad (5)$$

The Lyapunov candidate is then chosen as $V(x, a) = E^2$. By then taking the derivative of the candidate a controller $a = F(x)$ is derived which will make $\dot{V} \leq 0$ for all x . The controller will always output the maximal control signal possible in order to minimize the Lyapunov function.

4.2 Top controller

In the top position a LQR controller is used. For this the linearized model in eq. 3 is used to calculate L. In order to change the cost of matrices Q and R online a way to calculate the state feedback gain vector L was implemented and given by

$$L = (R + B^T P B)^{-1} (B^T P A)$$

where P is calculated by iterating

$$P_{k-1} = A^T P_k A - (A^T P_k B)(R + B^T P_k B)^{-1} (B^T P_k A) + Q$$

backwards in time with $P_N = Q$. This solution for the Riccati problem will not work for all systems but in this case Fredrik Bagge verified that it should work.

4.3 Controller switch

In order to switch between the two controllers good conditions is necessary to enable the pendulum to be in a good position for when the top controller takes over. Since the model is linearized around

$$x = (\theta \quad \dot{\theta} \quad \varphi \quad \dot{\varphi}) = (0 \quad 0 \quad 0 \quad 0)$$

the pendulum needs to have both a low velocity and be close to the top for the top controller to work properly. The switching conditions is chosen as an ellipse made from $\dot{\theta}$ and θ which will give good estimations of when to switch.

5 Friction compensation

When controlling the pendulum in the top position there will be limit cycles due to the friction. To minimize these cycles RLS based friction compensation is added to the top controller. When the pendulum is close to the top position a dead-zone is added in order to keep the noise from the measurements from making the friction compensation add control signal in the wrong direction. There are a lot of different ways to model friction. Some really complicated and some pretty simple. In fig 2 measured friction of the process can be seen for different velocities. The oscillating behaviour is believed to be due to different friction in different directions. Since the average friction for the different velocities did not vary that much and the measured signals are noisy the models of the frictions were kept simple. The models used can be seen in fig 3.

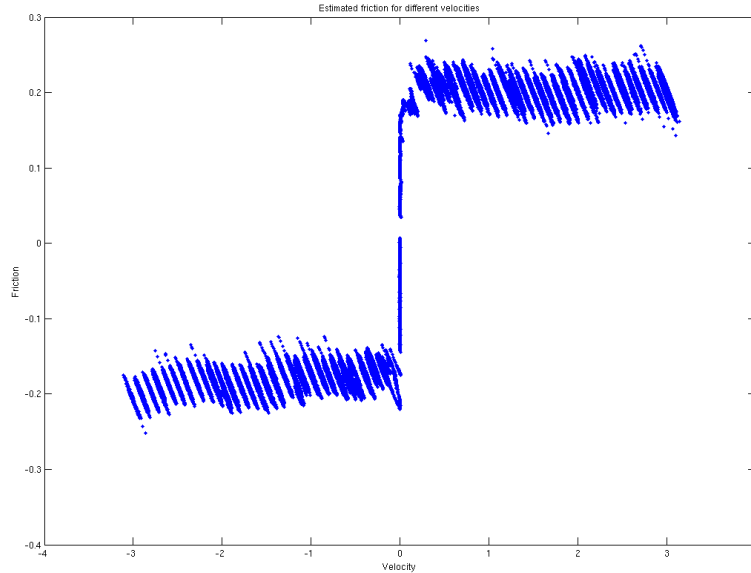


Figure 2: Measured friction of the process for different velocities

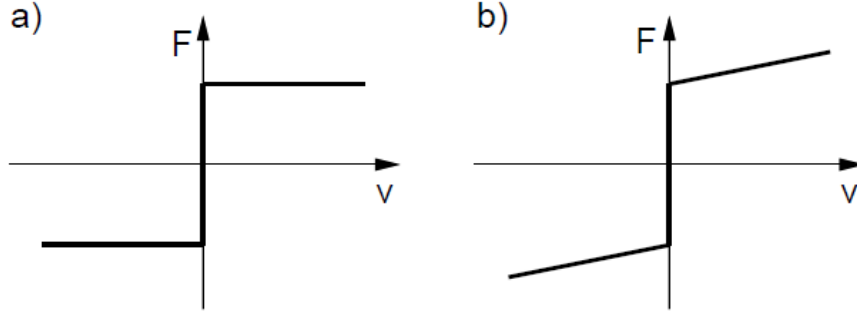


Figure 3: a) Friction model with coulomb friction. b) Friction model with coulomb and viscous friction.

5.1 Friction estimation

To manage the friction an estimate of a simple friction model is derived using RLS. The models in fig 3 will respectively give the following friction expressions

$$f = f_c \cdot \text{sign}(\dot{\varphi}) \quad (6)$$

$$f = f_c \cdot \text{sign}(\dot{\varphi}) + f_v \cdot \dot{\varphi} \quad (7)$$

where 6 is model a and only depends on the direction of the velocities and 7 is the model in b where a viscous term that depends on the velocity is added.

The regression models for the two friction models will take the form of ??.

$$\phi = \text{sign}(\dot{\varphi}), \beta = f_c \quad (8)$$

$$\phi = \begin{bmatrix} \text{sign}(\dot{\varphi}) \\ \dot{\varphi} \end{bmatrix}, \beta = \begin{bmatrix} f_c \\ f_v \end{bmatrix} \quad (9)$$

Since the friction can be seen as a direct negative torque on the arm, it can be directly subtracted from the control signal which gives us the system in 4

$$x_{k+1} = A_d x_k + B_d(u_k - f)$$

To get the RLS to work we take all measurements one step back in time and try to estimate the parameters from ??. We also aim to use the measurement from the arm velocity where we get values from both the pendulum position and the arm velocity, see the state equation for φ in eq. 4. After placing all known variables on the left hand side and we the following

$$\frac{\dot{\varphi}_k + 0.0059\theta_{k-1} - 1.9125u_{k-1} - \dot{\varphi}_{k-1}}{-1.9125} = (f_c \quad f_v) \begin{pmatrix} \dot{\varphi}_{k-1} \\ \text{sign}(\dot{\varphi}) \end{pmatrix} \quad (10)$$

Now we can apply the normal RLS algorithm with some forgetting factor λ .

As we see in equation 4 we probably can get different results in the friction estimation by choosing different equations in the same form as in equation 10. Initially we want to estimate the friction when the pendulum is hanging straight

down and the arm is rotating at a constant speed. If there were no friction, no control signal would be necessary to keep the velocity of the arm constant, and therefore the control signal required corresponds to the signal necessary for compensating the coulumb friction. We also want to estimate the friction in the top position and then the equation corresponding to the state $\dot{\varphi}$ in eq. 4 might not be the one which yields the best result. More research will be put into this at a later stage.

5.2 RLS ALG

$$\hat{\theta}_k = \hat{\theta}_{k-1} + P_k \phi_k \epsilon_k \quad (11)$$

$$\epsilon_k = y_k - \phi_k^T \hat{\theta}_{k-1} \quad (12)$$

$$P_k = \frac{1}{\lambda} \left(P_{k-1} - \frac{P_{k-1} \phi_k \phi_k^T P_{k-1}}{\lambda + \phi_k^T P_{k-1} \phi_k} \right) \quad (13)$$

6 GUI

The operator should be able to change controller parameters for both types of controller and for the friction estimator. For the top controller, the Q and R matrices. For the swing up controller, the switching area and its limit and the gain. Other general parameters for the controller such as sampling time h (if possible), and ω_0 . And for the friction estimation the forgetting factor λ and the initial values P_0 and ω_0 .

The GUI will plot the outputs from the Furuta pendulum, the friction estimation, control signal, both uncompensated as well as the compensated signal. We aim to have different regressor modes for the friction compensation and the operator can therefore change different parameters depending on the regressor chosen.

7 Results

8 Conclusion

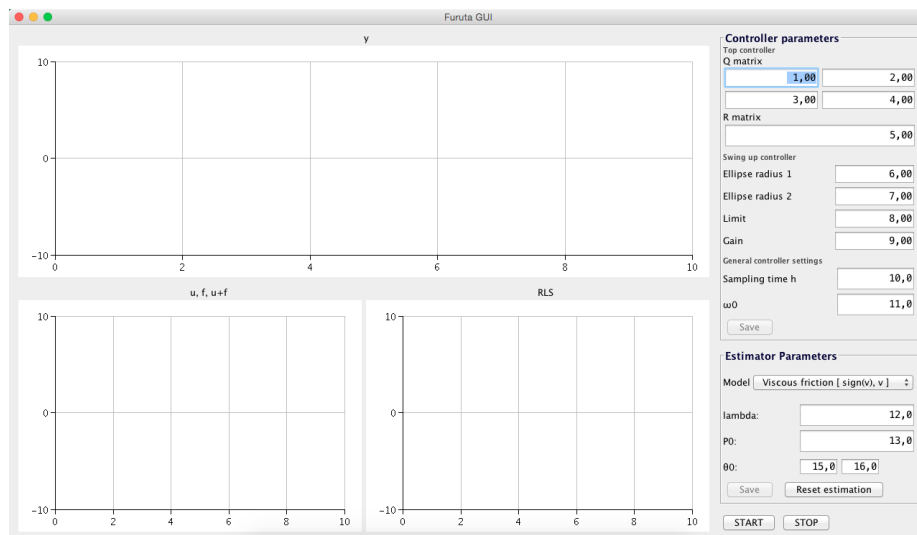


Figure 4: Overview of the program structure