Pendulum set description

Description of the pendulum setup in this section refers mostly to the control problems. For connection, interface and explanation on how are the signals measured and transferred to the PC refer to the '*Installation & Commissioning*' manual.

As shown in Figure 1 the pendulum setup consists of a cart moving along the 1 [m] length track. The cart has a shaft, to which two freely hanging pendulums are attached. The cart can move back and forth causing the pendulums to swing.

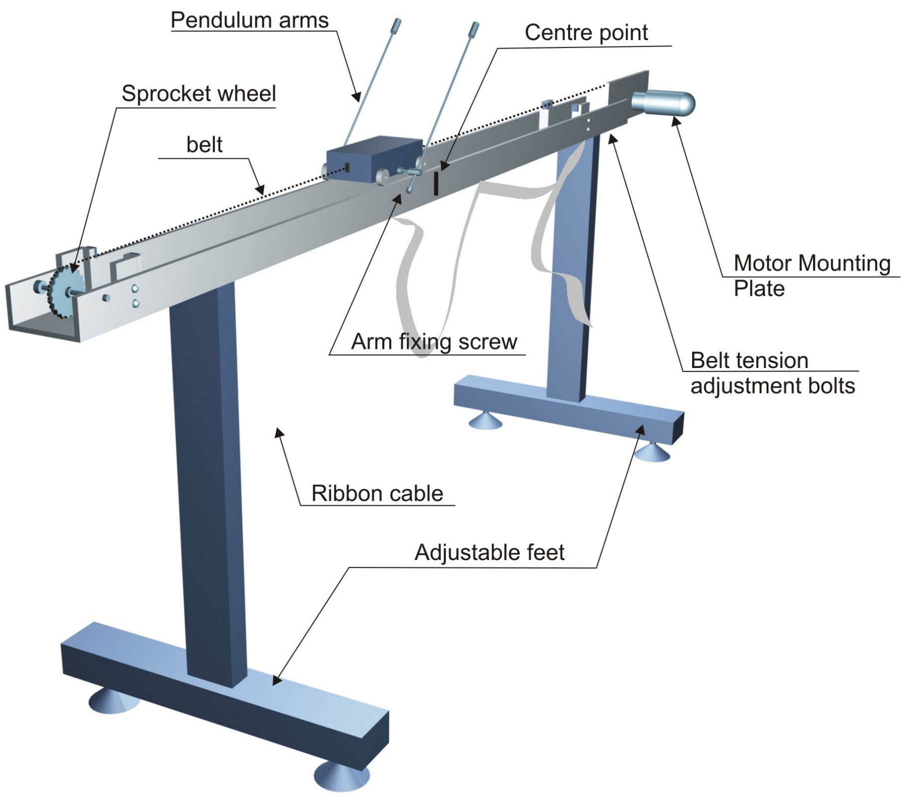


Fig. 1: Digital Pendulum Mechanical Unit.

The movement of the cart is caused by pulling the belt in two directions by the motor attached at the end of the rail. Applying voltage to the DC motor we control the force, with which the cart is pulled. The value of the force depends on the value of the control voltage. That voltage is our control signal. The two variables that are read from the pendulum through the encoders are the pendulum position (angle) and the cart position on the rail. The controller's task will be to change the DC motor voltage depending on these two variables, in such a way that the desired control task is fulfilled (stabilizing in upright position, swinging or crane control). Figure 2 presents how the control system is organised.

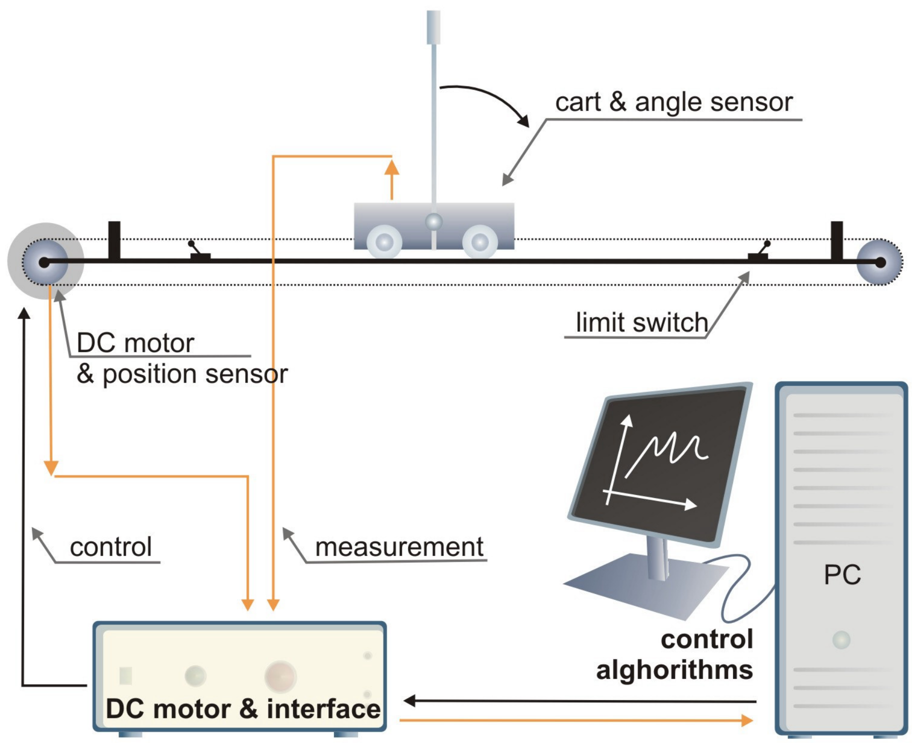


Fig. 2: Pendulum control system.

In order to design any control algorithms one must understand the physical background behind the process and carry out identification experiments. The next section explains the modelling process of the pendulum.

Pendulum model

Every control project starts with plant modelling, so as much information as possible is given about the process itself. The mechanical model of the pendulum is presented in Figure 3.

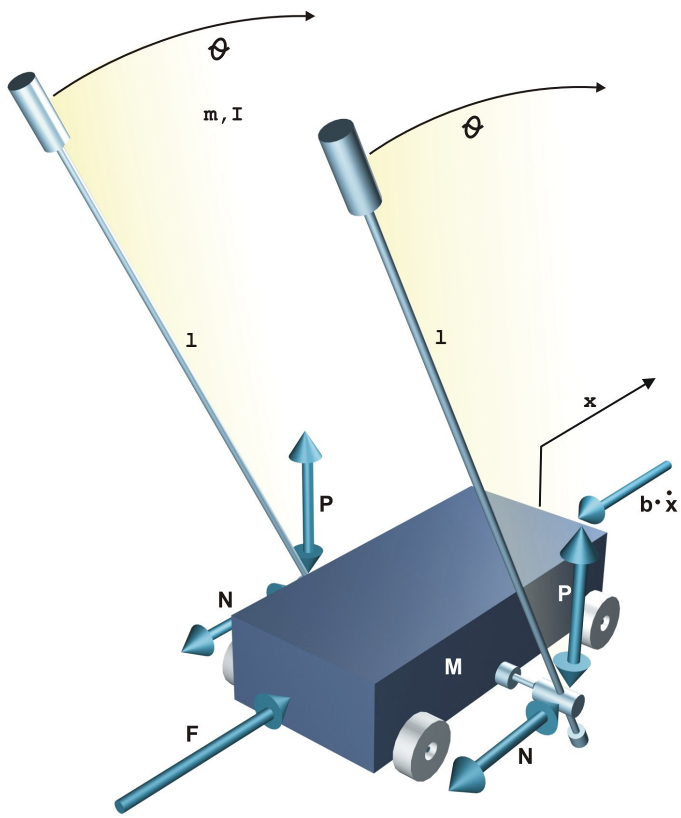


Fig. 3: Pendulum phenomenological model.

The phenomenological model of the pendulum is nonlinear, that means that at least one of the states (x and its derivative or θ and its derivative) is an argument of a nonlinear function. Such a model to be presented as a transfer function (a form of linear plant dynamics' representation used in control engineering) has to be linearised.

Summing the forces working on the pendulum and cart system and the moments we obtain the following nonlinear equations of motion:

$$(m+M)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F, \qquad (1)$$

$$(I+ml^2)\ddot{\theta} - mgl\sin\theta + ml\ddot{x}\cos\theta + d\dot{\theta} = 0. \qquad (2)$$

Very often control algorithms are tested on such nonlinear models. However for the purpose of controller design the models are linearised and presented in the form of transfer functions. Such a linear equivalent of the nonlinear model is valid only for small deviations of the state values from their nominal value. Such a nominal value is often called the equilibrium point. The pendulum has two of these, one is when $\theta = 0$ (inverted pendulum) and the other when $\theta = \pi$ (hanging freely – crane control).

The inverted pendulum is an unstable system, which in terms of behaviour means that the plant left without any controller reaches an unwanted, very often destructive state. Thus for such plants it is useful to carry out simulation tests on the models before approaching the real plant.

To complete the model given by motion equations (1) and (2), we must introduce the value of all parameters. The following table gives the values of the parameters:

Table 1. Pendulum parameters

Parameter	Value
g- *gravity*	*9.81 m/s²*
l- *pole lenght*	*0.36 to 0.4 m* *- depending on the configuration*
M- *cart mass*	*2.4 kg*
m- *pole mass*	*0.23 kg*
I- *moment of interia of the pole*	*about 0.099 kg·m²* *- depends on the configuration*
b - *cart friction coefficient*	*9.81 m/s*
d - *pendulum damping coefficient*	*although negligible,necessary in the model- 0.005 Nsm/rad*

Two of the things have to be kept in mind when designing the controllers. Both the cart position and the control signal are bounded in real time application. The bound for the control signal is set to [–2.5V .. +2.5V] and the generated force magnitude of around [–20.0N .. +20.0N]. The cart position is physically bounded by the rail length and is equal to [-0.5m .. +0.5m].

The pendulum is a SIMO plant – single input multiple output (Figure 4). The model described by (1),(2) is still missing the translation between the force F and the actual control signal, which is the control voltage u that we supply with the PC control card. Assuming the relation between the control voltage u and the generated cart velocity is linear, we might add the velocity vector generated by the motor to the model and ignore the F vector, or translate the control voltage u to the generated force F under the assumption that constant voltage will cause the cart to move with constant velocity:

$$F = k_{Fu} \cdot \frac{du}{dt}, \tag{3}$$

where k_{Fu} – is the gain between the u voltage derivative and the F force. However one must remember that derivative introduction in models especially in Simulink may cause simulation problems.

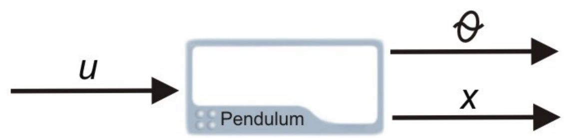


Fig. 4: Pendulum model

Exercise 1 – Nonlinear model

Introduction

For the initial exercise the user has been provided with the pendulum model described by equations (1) and (2). The model shown in Figure 4 can be opened in Simulink - '***pendmod_nonlin.mdl***'.

Task

For the beginning the user is advised to check the responses of such model in the situation where zero u voltage is applied. You can change the value of the θ_0 (theta0 – initial pendulum angle) and see how the pendulum responds.

Exemplary results and comments

Two of the values of the angles θ_0 are particularly interesting ($\theta_0 = 0$, $\theta_0 = \pi$). First, which is an equilibrium point for which small disturbance for an open loop system will cause the pendulum to fall down, swing and finally settle down in $\theta_0 = \pi$. The small disturbance can be introduced by small initial angle selection, for example $\theta_0 = 0.001$. The second $\theta_0 = \pi$ is an equilibrium in which the pendulum will always settle when no F force is applied.
The results for initial θ_0 value of $\theta_0 = 0.001$, are presented in Figure 5. Because of the friction forces the pendulum swings until it settles in $\theta_0 = \pi$.

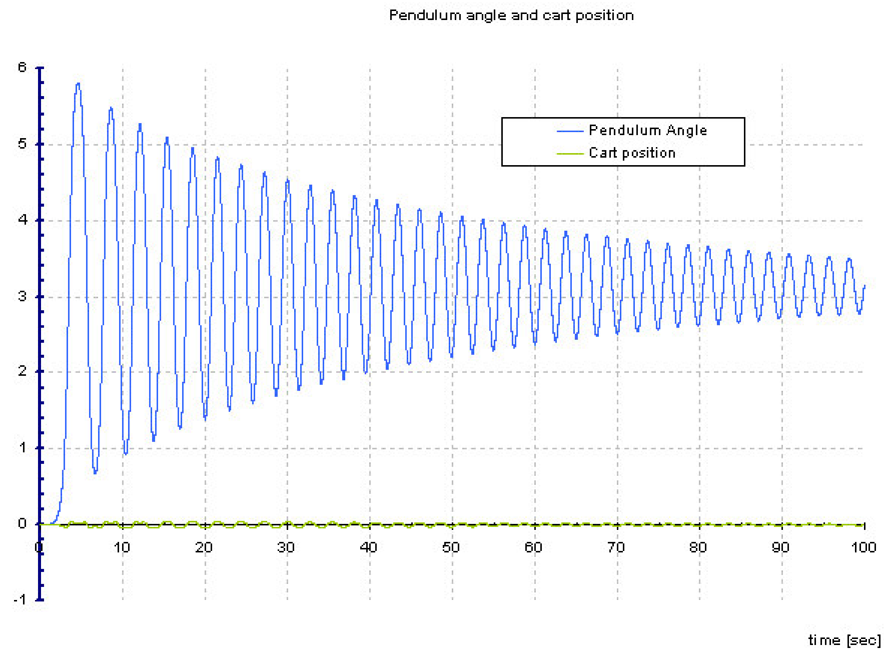


Fig. 5: Pendulum model results for $\theta_0 = 0.001$.

Model linearization

To carry out model dynamics analysis for open loop[1] systems like Bode plots, poles and zeros maps, Nyquist plots, root locus (for closed loop[2] systems only) the model has to be linearised. Linearization of a given phenomenological model can be done for the pendulum and in the given equations (1) and (2) we could substitute the nonlinear functions (sine and cosine) with their linear equivalent. Such a linearization in a working point is done with Taylor approximation of the nonlinear functions. For small angle deviations in an equilibrium point of $\theta = 0$ (inverted pendulum) we can assume that the following functions can be linearised:

$$\sin\theta \cong \theta, \tag{4}$$

$$\cos\theta \cong 1, \tag{5}$$

$$\dot{\theta}^2 = 0. \tag{6}$$

Thus the motion equations (1) and (2) take the form:

$$(m+M)\ddot{x} + b\dot{x} + ml\ddot{\theta} = F, \tag{7}$$

$$(I + ml^2)\ddot{\theta} - mgl\theta + ml\ddot{x} + d\dot{\theta} = 0. \tag{8}$$

One must remember that the equations (7) and (8) will only be valid for $\theta = 0$. For the position where $\theta = \pi$ (crane control) the following substitutions have to be made:

$$\sin\theta \cong -\theta, \tag{9}$$

$$\cos\theta \cong -1, \tag{10}$$

$$\dot{\theta}^2 = 0. \tag{11}$$

Thus the motion equations (1) and (2) take the form:

$$(m+M)\ddot{x} + b\dot{x} - ml\ddot{\theta} = F, \tag{12}$$

$$(I + ml^2)\ddot{\theta} + mgl\theta - ml\ddot{x} + d\dot{\theta} = 0. \tag{13}$$

Linear model of the pendulum, just as the nonlinear has one input – force F, and two outputs which are the θ angle and the cart position x (Figure 4). However in the inverted pendulum task we are mostly interested in the θ angle stabilization thus we may treat the cart position as

[1] Open loop system – the plant without a controller

[2] Closed loop system – the plant and controller in negative feedback loop, see "Control" section for more information.

an uncontrolled output. With one input F and one output θ two linear models in the form of transfer functions can be obtained, each for small deviations of the θ angle form two equilibrium points of $\theta = [0, \pi]$. Remember that the translation between the control voltage and the force should be added (equation 3).

Exercise 2 – Linear models

Introduction

Two of the linear models described by the equations (7),(8) and (12),(13) have been created. (*The user is advised to transform (7),(8) and (12),(13) into transfer function form on his own*). One must remember that both of these transfer functions resemble the behaviour of the real plant only for small deviations of the θ angle. The models '***pendmod_lin_stable.mdl***' and '***pendmod_lin_unstable.mdl***' hold the appropriate transfer functions:

$$G(s) = \frac{\theta(s)}{F(s)} = \frac{s}{d_3 \cdot s^3 + d_2 \cdot s^2 + d_1 \cdot s \; + d_0} \qquad (14)$$

For both of these models the denominators' parameters' signs will differ. The x position output is calculated based upon the output of the *G(s)* transfer function, which is θ. For the *stable* transfer function, that is the representation of the pendulum behaviour in the surrounding of the $\theta = \pi$, an offset of $o_{ffset} = \pi$ has to be added to the output of the transfer function. That is our initial condition.

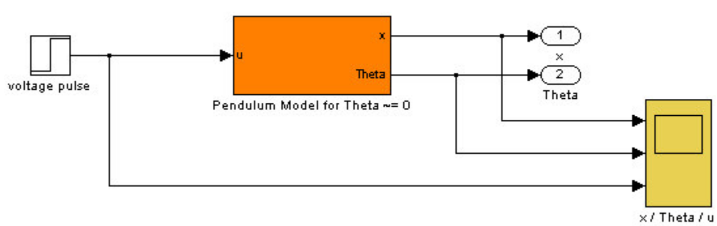


Fig. 6: Example of linear Simulink model.

Task

The user is advised to compare the responses of the linear and nonlinear pendulum models in open loop system (no controller). Also with the use of Matlab the Bode diagrams, zeros and poles maps can be drawn to carry out initial dynamic response analysis of the pendulum. In the transfer function form of the model the initial condition is equal *0* thus in order to see the linear model response we have to stimulate it with a control voltage pulse to see the reaction.

Exemplary results and comments

The dynamic response of the unstable transfer function will be different from the nonlinear system. Distorted by small voltage u impulse will not settle in the equilibrium $\theta = \pi$**.** The transfer function is valid for small deflections of θ thus it is unstable and the response grows to infinity without any control action.

The other transfer function valid for small deviations of θ but for an equilibrium of $\theta = \pi$ will behave similarly to the nonlinear model around the point $\theta = \pi$.

Model identification

The phenomenology analysis delivers a model that we 'think' fits the pendulum the best. However we know that it is just some approximation. We might have made mistakes analysing the phenomenology – wrong model structure choice, or could choose wrong parameters' values. To have an adequate model we could tune the phenomenological model. Because of the fact that it is nonlinear, the identification and tuning of that model can be a very difficult task (gradient methods). To simplify the identification, modelling and control the control algorithms will be designed based upon the dynamics of the linear models. They will be tested however on the nonlinear model and plant. Furthermore the identified models will be discrete as such models are obtained in the course of the Least Mean Square identification methods implemented in Matlab. Any of the continuous models can also be transformed for the comparison purpose into the discrete form. The obtained discrete models can be also transformed into continuous equivalent.

The plant identification theory is very broad and solves numerous problems. In the identification experiments of the linear pendulum models it is convenient however to use the identification Matlab tool.

Before any model identification procedures of the pendulum setup will be carried out we have to describe, through a simple real time simulation, the character of the dead zone. The dead zone is a nonlinearity, which could influence the model validity. Because of the static friction forces with infinitesimally small values of the control signal the cart will not move. Furthermore the static friction force may not be symmetrical. We can compensate that by simply adding additional voltage offset value. What is the value of that offset that we have to add is the outcome of the exercise 3.

Exercise 3 – Static friction compensation

Introduction

Since the control signal is the voltage that we supply the dead zone value will be expressed in volts. As explained it will be different for two movement directions. It has to be identified because the pendulum motor belt can be set up differently by the user and thus cause different static friction forces. The friction identification can be carried out with '***PendulumFriction.mdl***'. In the simulation the control voltage is increased until the cart moves in the positive direction. The cart is stopped. Then the control is decreased until the cart moves into the negative direction. The voltage values for which the cart begins to move are these offset values.

Task

According to the '*Installation & Commissioning*' manual run the friction identification simulation on the pendulum.

Exemplary results and comments

The results will be presented in displays. You can correct the friction compensation value in all simulations the '*Friction Compensation'* block (Figure 7 and 8), which is placed in the '*Feedback DAC*' block. However if the result will strongly differ form 0.1 [V] it may appear to be biased by other nonlinearities and should be discarded.

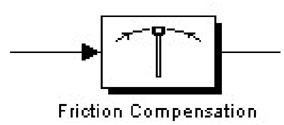


Fig. 7: Friction compensation block

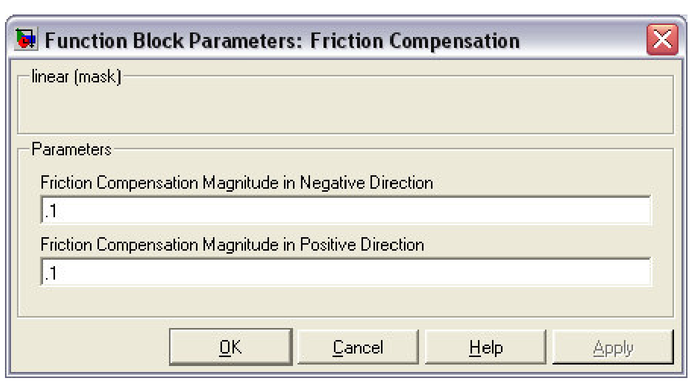


Fig. 8: Friction compensation block menu (double click the block to open).

With the static friction identification carried out we can try to identify the first dynamic model of the pendulum. Although the most interesting signal relation is the one between the control signal u and the angle θ we can first try to identify the linear model between the control signal u and the cart position x. That model will be used for first PID controller design and tests in the "*Pendulum control*" section.

There are few important things that the control system designer has to keep in mind when carrying out an identification experiment:

- *Stability problem* – if the plant that is identified is unstable, the identification has to be carried out with a working controller, which introduces additional problems that will be discussed further on. If the plant is stable and does not have to work with a controller the identification is much simpler.
- *Structure choice* – a very important aspect of the identification. For the linear models it comes down to the numerator and denominator order choice of the transfer function. It applies both for the continuous and discrete systems.
  As far as the discrete models are concerned the structures are also divided in terms of the error term description: ARX, ARMAX, OE, BJ[3].
- *Sampling time* – the sampling time choice is important both for the identification and control. It cannot be to short nor can it be too long. Too short sampling time might influence the identification quality because of the quantization effect introduced by the AD. Furthermore the shorter the sampling time the faster the software and hardware has to be and more memory is needed. However short sampling time will allow for aliasing effect elimination and thus anti aliasing filters[4] will not have to be introduced. Long sampling times will not allow for including all of the dynamics.
- *Excitation signal* – for the linear models the excitation choice is simple. Very often designers use white noise however in industrial application it is often disallowed. It is attractive however because of the fact that it holds very broad frequency content thus the whole dynamics of the plant can be identified. If the dynamics are not to complex several sinusoids with different frequencies can be summed to produce a satisfactory excitation signal.
- *Identification method* – usually two methods are being used, the Least Mean Square (LMS) method and the Instrumental Variable method. The LMS method is the most popular and implemented in Matlab. The method minimizes the error between the model and plant output. The optimal model parameters, for which the square of the error is minimal is the result of the identification.

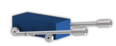
[3] More information about these structures can be obtained during the System Identification courses.

[4] These are the basics of the Digital Signal Processing course. For more insight the user is asked to study more on signal processing and digital control.

Cart model identification

The following exercise includes all of the mentioned facts and provides an identification experiment, which results in a discrete model of the moving cart due to the control voltage application. At this point the pendulum is ignored, its movement is treated as a distortion. It would be best to immobilize the pendulum to reduce the disturbance.

Exercise 4 – First model identification



Introduction

All of the control real time simulations are carried out with a sampling time of Ts = 0.001 [s]. In the identification experiments the sampling time varies. For the identification the Matlab identification interface is used. Here the sampling time is set to $Ts = 0.05$ [s].

The identification experiment is carried out with the '***CartIdent.mdl***'. The excitation signal is composed of several sinusoids. The experiment lasts 20 seconds, two signals are collected in a form of vectors and are available in the Workspace.

Task

Carry out the identification experiment, collect the data. With the use of Matlab identification interface (type *'ident'* at the command line - the identification interface will open (Figure 9)) identify a discrete model.

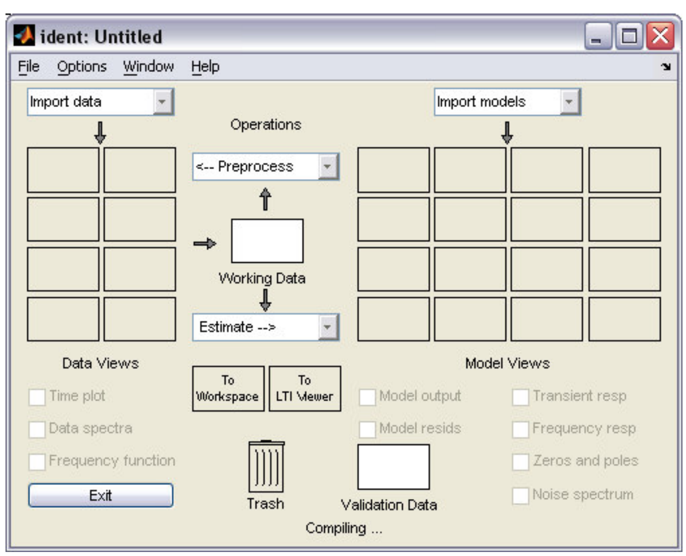


Fig. 9: Matlab identification interface.

Upload the signals for the identification from Workspace. For simplicity assign the input output data to the *u* and *y* vectors:

u = simout(:,3);
y = simout(:,1);

Make sure you specify the proper sampling time in the *ident Data Import* interface. Select the start time to be *0*. To identify a parametric model select the proper structure of the model (e.g. OE 2 4 1). Press 'estimate'. You can check the quality of the response of the identified model by the step response analysis, transient response, pole and zeros map, frequency response and model residuals.

Exemplary results and comments

The step response of the identified system can be similar to the one presented in Figure 10. If the model is transferred into Workspace it can be compared against the discrete equivalent of the continuous transfer function. In order to obtain the discrete form use the '*c2d*' command. Make sure you specify the proper sampling time.

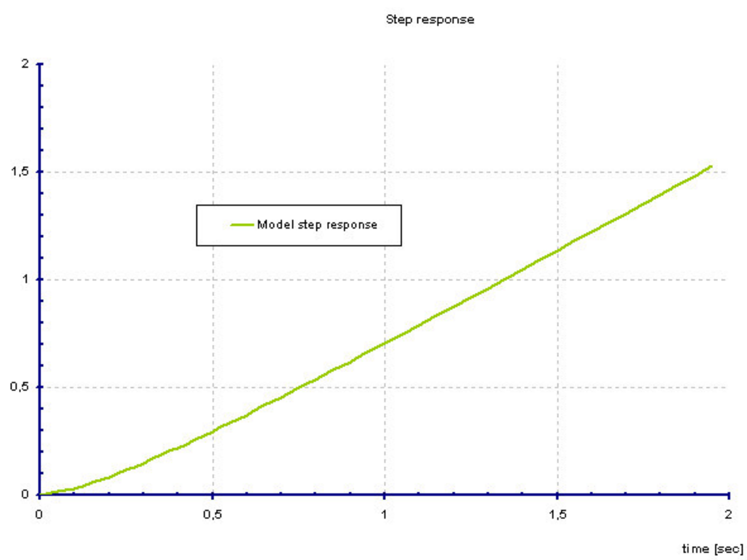


Fig. 10: Step response of the model.

Compare the step responses or bode plots of the two systems ('step' and 'bode' commands). You can also transform the discrete models into continuous equivalent with the means of '*d2c*' command.
The obtained model is used in the first PID control exercise.

Crane identification

Similar identification experiment can be carried out when identifying the transfer function between the control voltage u and the pendulum angle θ. As presented in the '*Model linearization*' section two linear models can be identified. First which holds for small deviations of angle θ around the $\theta = 0$ point, and the other when $\theta = \pi$. In this section the crane function of the pendulum is considered, thus $\theta = \pi$.

Exercise 5 – Crane linear model identification

Introduction

The crane linear discrete model can be identified in the same way as the cart model. For the identification the Matlab identification interface is used. The identification experiment is carried out with the '***CraneIdent.mdl***'. The excitation signal is composed of several sinusoids. The experiment lasts 20 seconds, two signals are collected in a form of vectors and are available in the Workspace. The sampling time is set at $Ts = 0.05$ [s].

Task

Carry out the identification experiment, collect the data. Upload the signals for the identification from Workspace. For simplicity assign the input output data to the u and y vectors:

$$u = simout(:,3); \;\; y = simout(:,2);$$

With the use of Matlab identification interface identify a discrete model.
Upload the signals for the identification from Workspace. Make sure you specify the proper sampling time. Select the proper structure of the model (e.g. OE 2 6-{or higher} 1). Press 'estimate'. You can check the quality of the response of the identified model by the step response analysis, transient response, pole and zeros map, frequency response and model residuals.

Exemplary results and comments

The response of the model is compared to the pendulum output in Figure 11. If the model is transferred into Workspace it can be compared against the discrete equivalent of the continuous transfer function (14). In order to obtain the discrete form use the '*c2d*' command. Make sure you specify the proper sampling time.

The models you obtain may strongly depend on the mounting of the belt, and the way all the bolts are screwed.

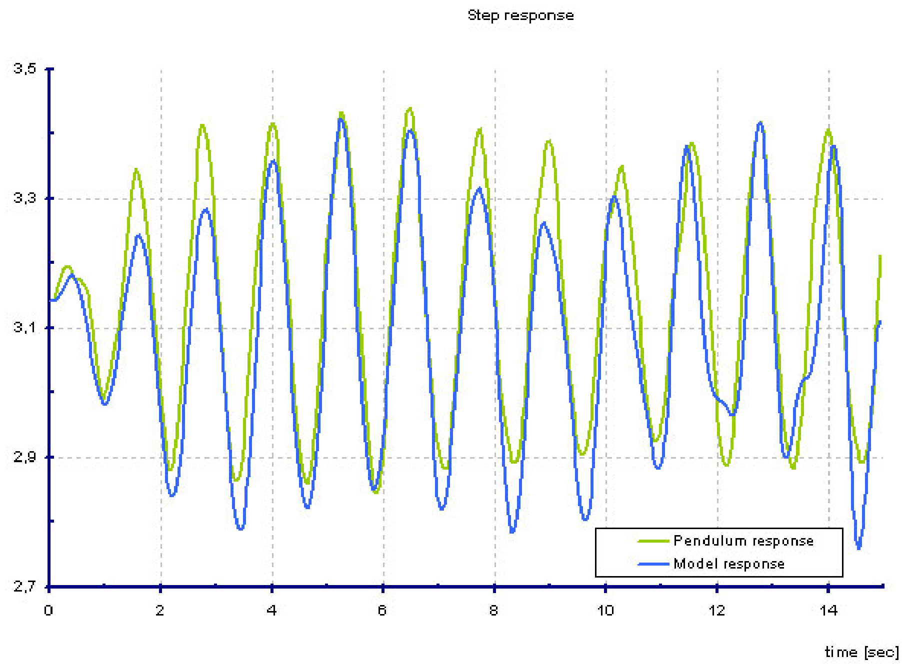


Fig. 11: Identified model and pendulum response to excitation.

Inverted pendulum identification

Inverted pendulum model identification is a difficult task mainly for one aspect – stability. The inverted pendulum is unstable and has to be identified with a running, stabilizing controller[5] – closed loop identification[6]. The controller introduces output noise and control signal correlation, which leads to model corruption. This correlation can be broken by introducing additional excitation signal, which is added to the control signal *u* (Figure 12).
If the power of the signal *r* is substantial comparing to the *n* noise power the proper model should be identified.

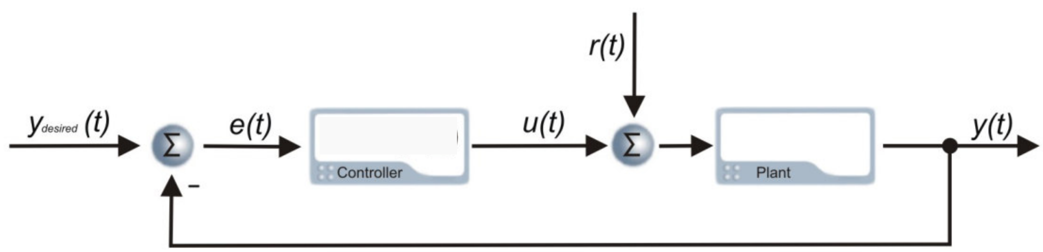
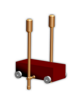


Fig. 12: Unstable system identification.

Such approach will only allow for the linear model identification of the transfer function between the voltage control signal *u* and the pendulum angle θ, for small deviations of the angle around the equilibrium point of $\theta = 0$.

More intelligent identification methods should be applied for complete nonlinear pendulum model identification (gradient methods).

Exercise 6 – Inverted pendulum linear model identification



Introduction

The inverted pendulum linear discrete model can be identified only with a controller. This is a major difference comparing to the previous exercises. For this identification task the Matlab identification interface is used. The identification experiment is carried out with the '***InvPendIdent.mdl***'. The excitation signal is composed of several sinusoids. The experiment lasts 20 seconds, two signals are collected in a form of vectors and are available in the Workspace.

---

[5] The pendulum control aspect is explained in the '*Pendulum Control*' section.

[6] Closed loop system identification is a broad topic and more advanced users are advised to refer to identification literature to get more insight.

It is IMPORTANT that before the identification experiment is started, you place the cart in the ZERO position and settle the pendulum in the bottom vertical position, $\theta = \pi$.

Task

Carry out the identification experiment, collect the data. The pendulum response to the excitation will be as presented in Figure 13.

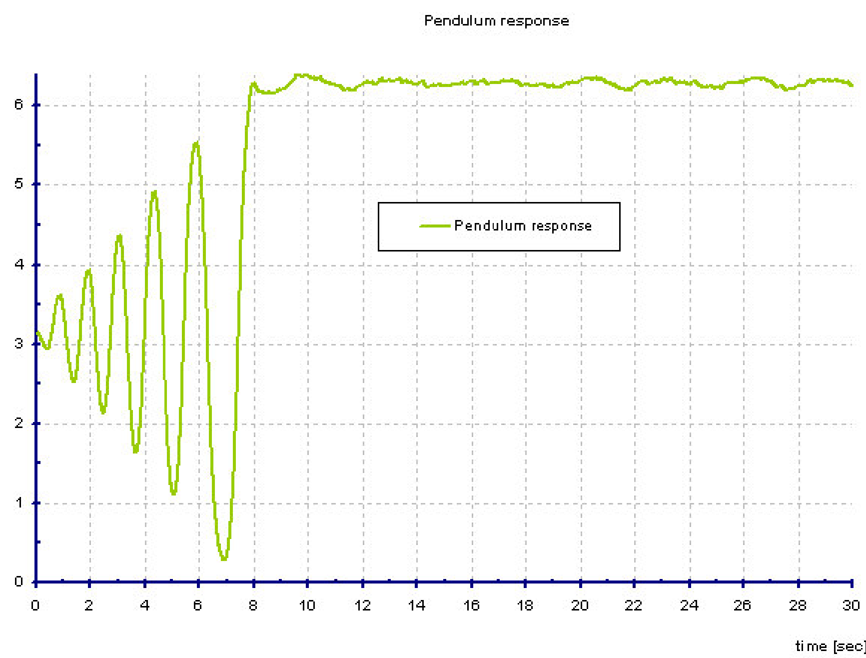


Fig. 13: Pendulum response.

Make sure that for the model identification you will use the data from the time when the pendulum is in the upright position. As shown on the plot it is safe to use data from the 10th second of the experiment. With the use of Matlab identification interface identify a discrete model.

Upload the signals for the identification from Workspace. Make sure you specify the proper sampling time $Ts = 0.01$ [s]. Select the proper structure of the model (e.g. 2 6 1). Press 'estimate'. You can check the quality of the response of the identified model by the step response analysis, transient response, pole and zeros map, frequency response and model residuals.

Exemplary results and comments

The response of the identified system can be as presented in Figure 14. If the model is transferred into Workspace it can be compared against the discrete equivalent of the continuous transfer function (14). In order to obtain the discrete form use the '*c2d*' command. Make sure you specify the proper sampling time.

The obtained model will be used in the '*Pendulum Control'* section for the controller design.

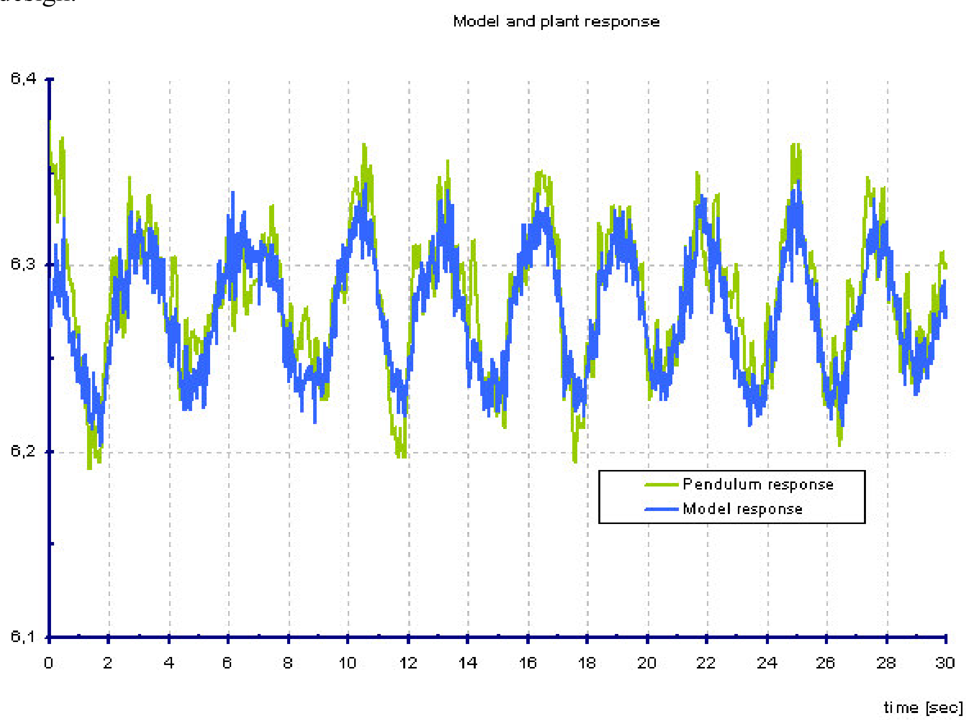


Fig. 14: Inverted pendulum model vs. plant response.