ELEKTRO 2012

A Complex Overview of the Rotary Single Inverted Pendulum System

13. 10. 2012

Ing. Slávka JADLOVSKÁ

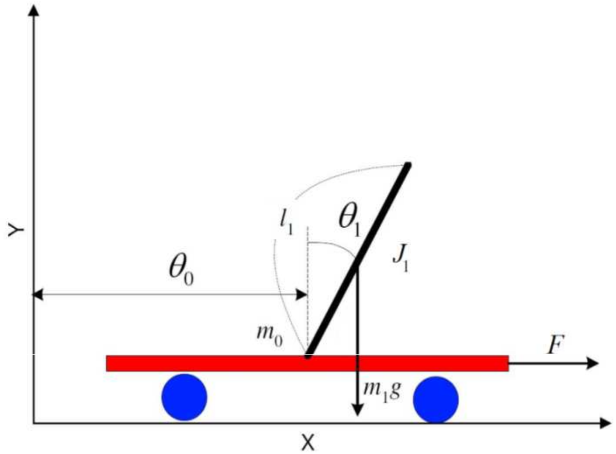
prof. Ing. Ján SARNOVSKÝ, CSc.

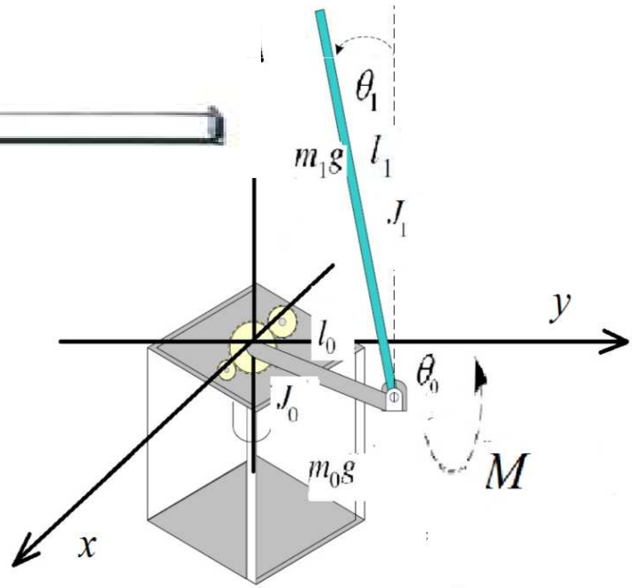
Department of Cybernetics and Artificial Intelligence

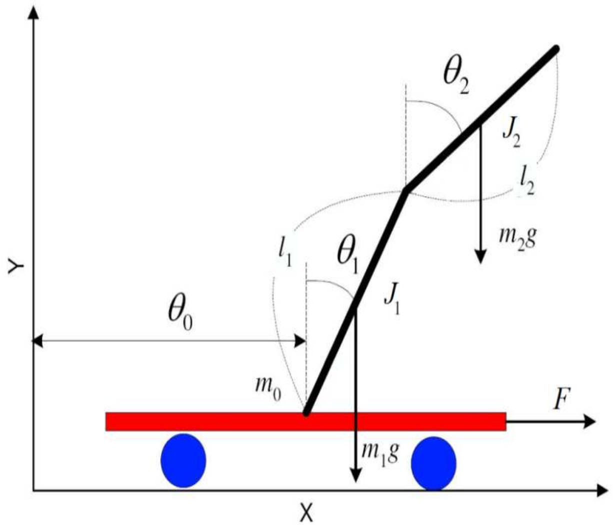
Faculty of Electrical Engineering and Informatics

Technical University of Košice

Inverted Pendula Systems – a class of mechanical systems significant for control theory







Main Points of the Presented Problem

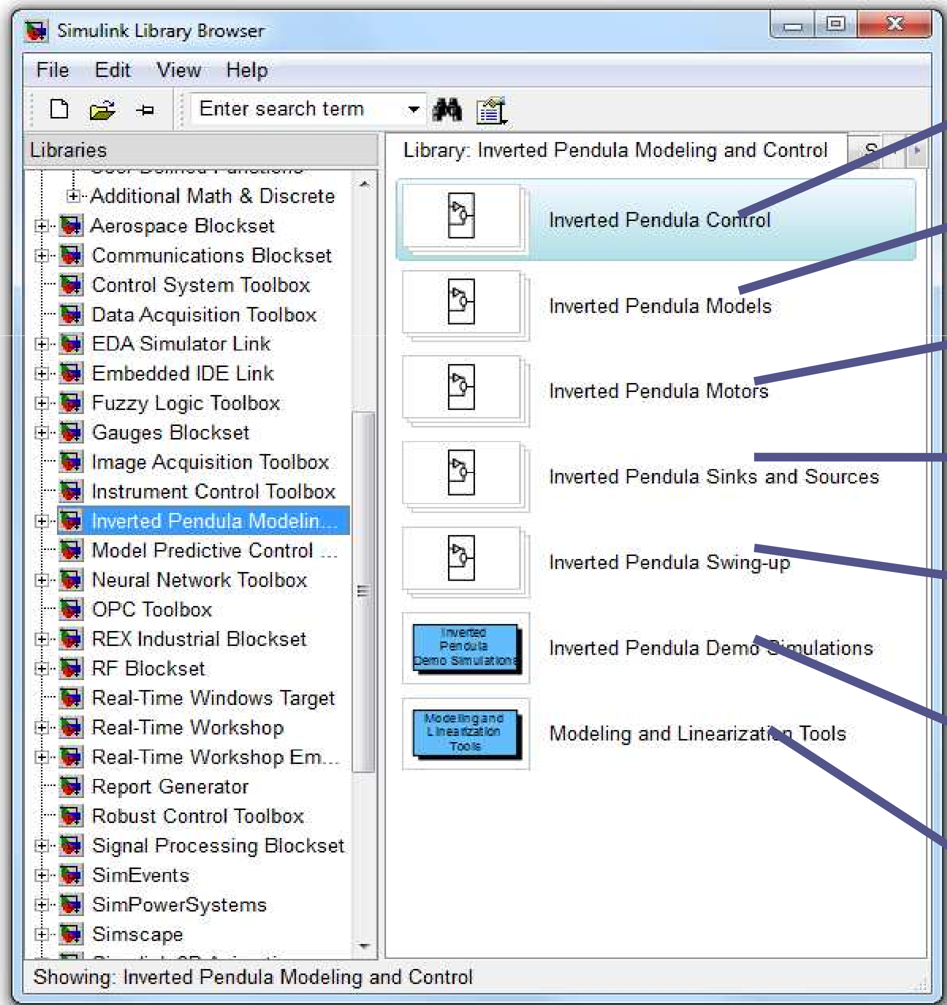
I. **mathematical modeling and simulation of the rotary inverted pendulum system**
   - automatic derivation of motion equations
   - open-loop dynamical analysis

II. **stabilization of the rotary single inverted pendulum via state-feedback control techniques**
   - automatic linear approximation of the system
   - state-feedback control with a state estimator
   - state-feedback control with permanent disturbance compensation

III. **conclusion and evaluation of the achieved results**

13. 10. 2012

Inverted Pendula Modeling and Control - IPMaC (Simulink block library)



- state-feedback control
- simulation models of inverted pendula
- direct-current motor
- input/output blocks
- swing-up into the upright position
- demonstrations of the block functionality
- GUI applications

A.
Mathematical Modeling and Simulation of the Rotary Single Inverted Pendulum

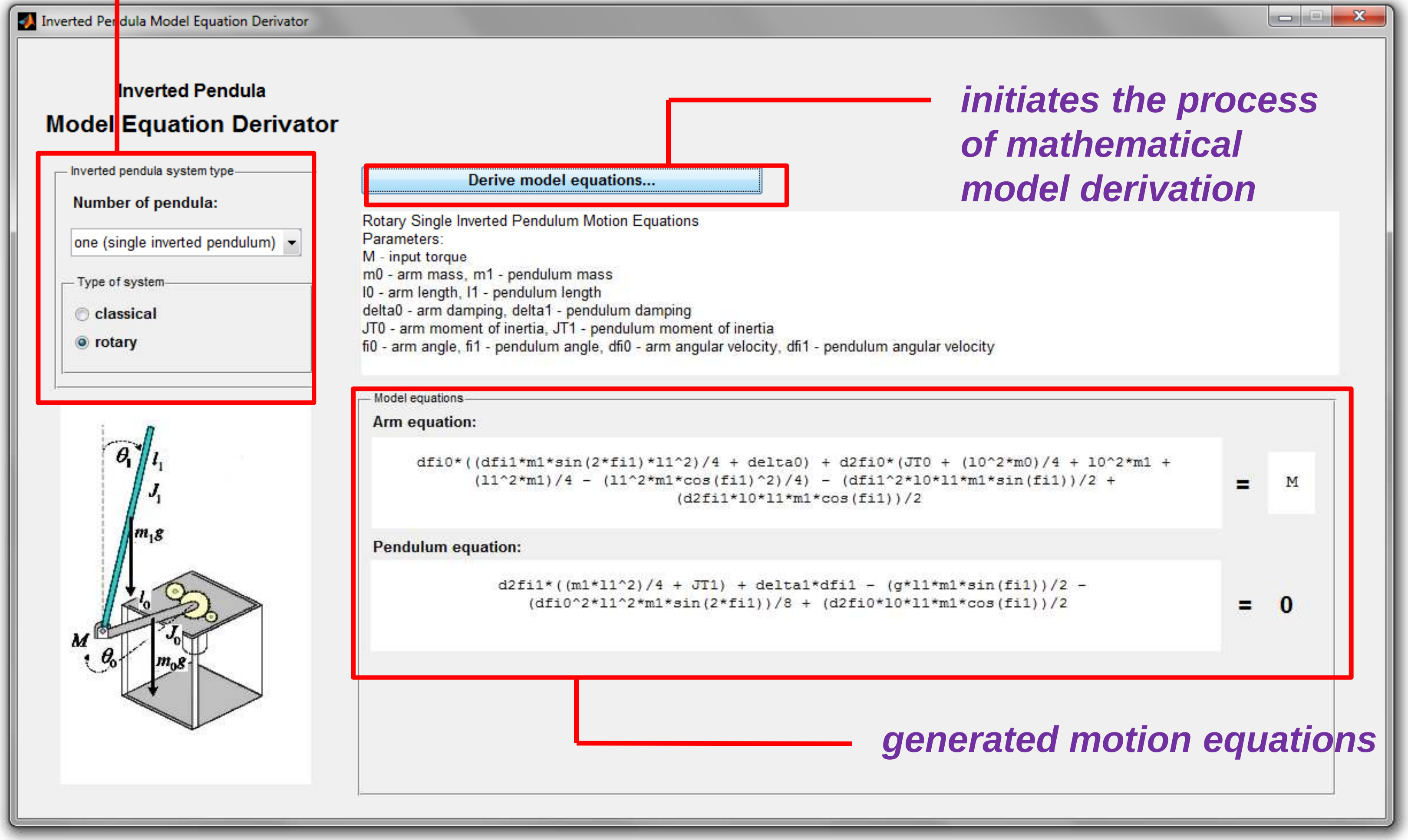
A.

13. 10. 2012

Inverted Pendula Model Equation Derivator (automatic derivation of motion equations)

selection of system type & number of pendulum links

initiates the process of mathematical model derivation

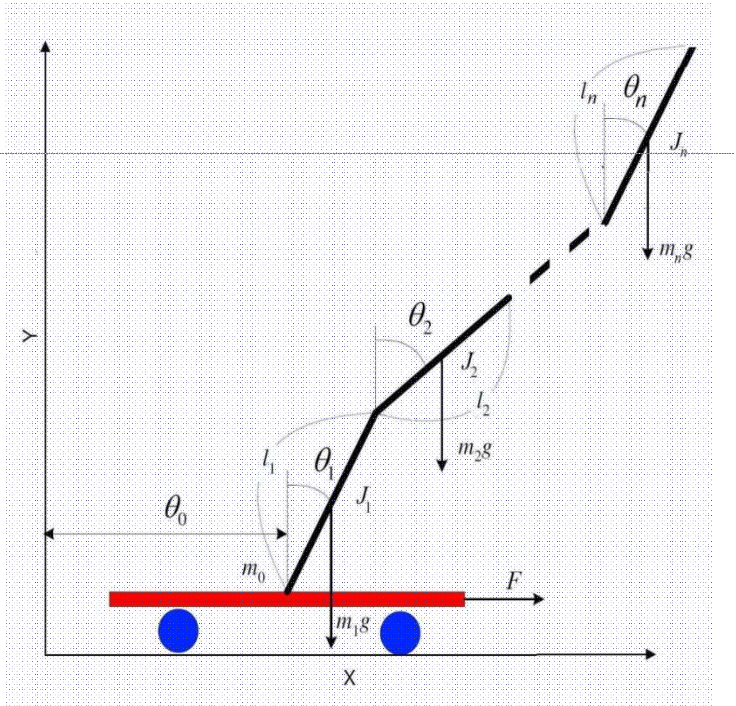


generated motion equations

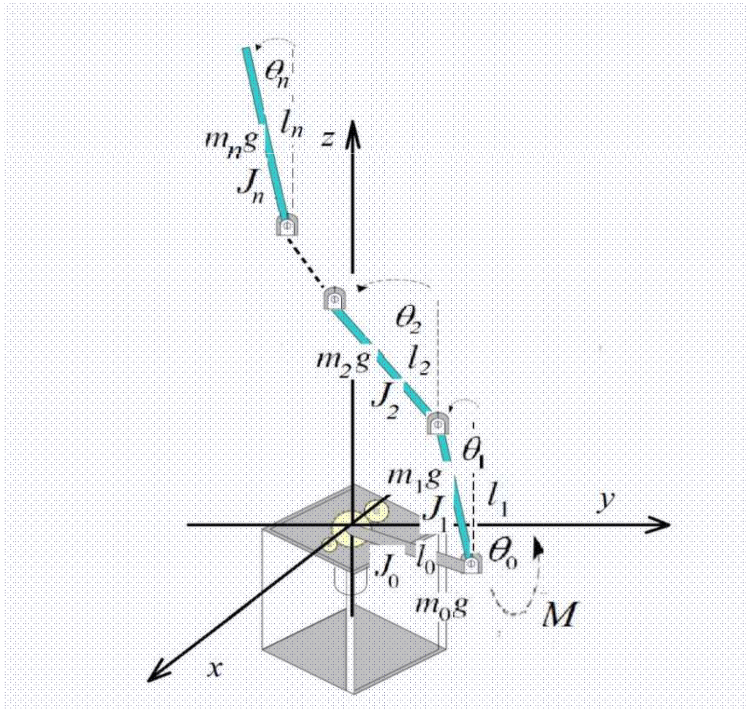
A.
Generalized approach to inverted pendula modeling

System description: $\boldsymbol{\theta}(t) = (\theta_0(t) \quad \theta_1(t) \quad \dots \quad \theta_n(t))^T$

cart/arm position — pendula angles



system of *n* classical inverted pendula



system of *n* rotary inverted pendula

A.

General procedure of mathematical model derivation for inverted pendula systems - brief outline

Vector of generalized coordinates:

$$\boldsymbol{\theta}(t) = \left(\theta_0(t) \quad \theta_1(t) \quad \ldots \quad \theta_n(t)\right)^T$$

Lagrange equations of the second kind:

$$\frac{d}{dt}\left(\frac{\partial L(t)}{\partial \dot{\boldsymbol{\theta}}(t)}\right) - \frac{\partial L(t)}{\partial \boldsymbol{\theta}(t)} + \frac{\partial D(t)}{\partial \dot{\boldsymbol{\theta}}(t)} = \mathbf{Q}^*(t)$$

`invpenderiv.m`

`rotinvpenderiv.m`

MATLAB functions which derive the motion equations for a system with a given number of pendulum links

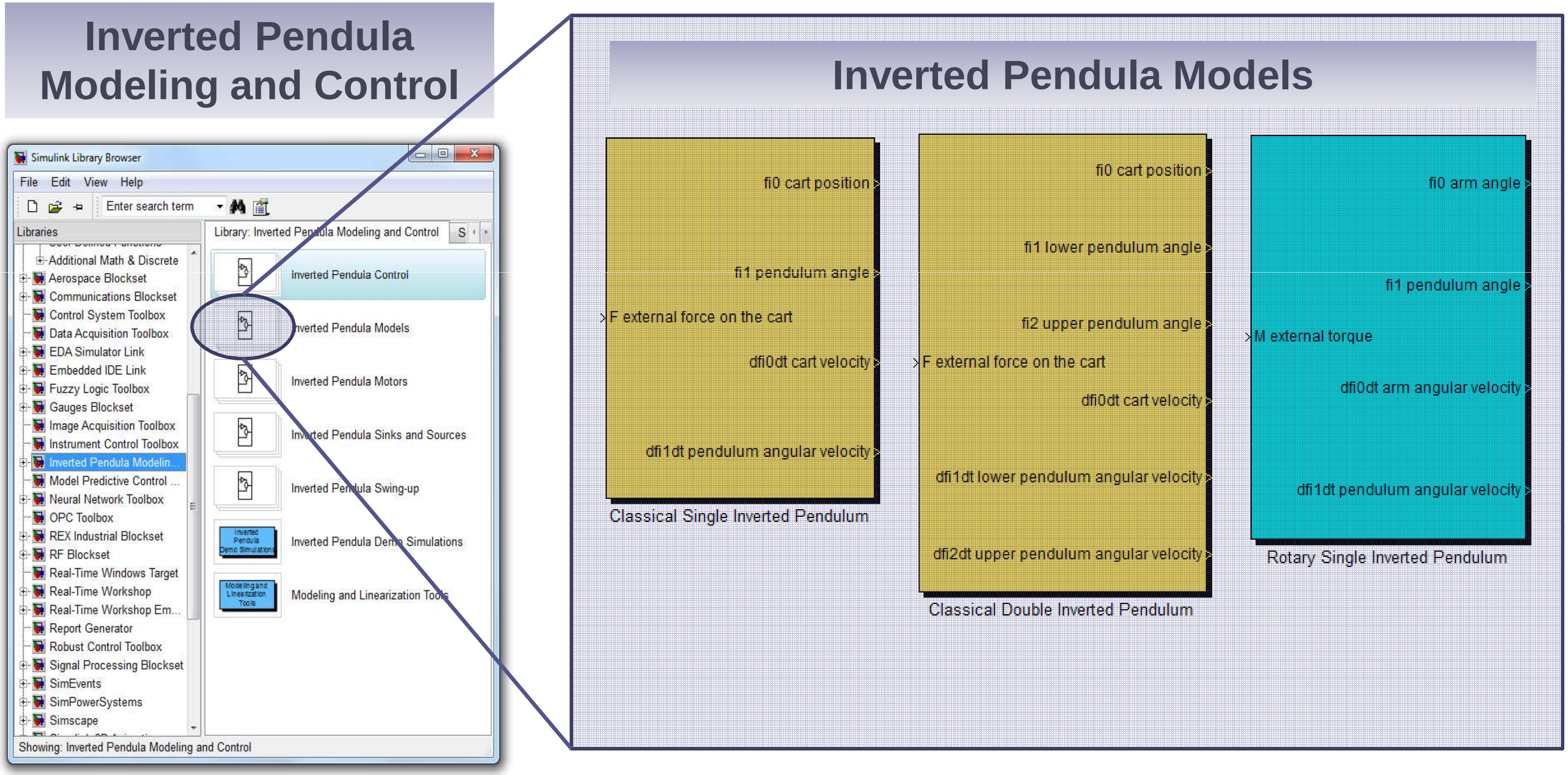
Rearrangement into standard minimal ODE form:

$$\boldsymbol{M}\left(\boldsymbol{\theta}(t)\right)\ddot{\boldsymbol{\theta}}(t) + \boldsymbol{N}\left(\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t)\right)\dot{\boldsymbol{\theta}}(t) + \boldsymbol{P}\left(\boldsymbol{\theta}(t)\right) = \boldsymbol{V}(t)$$

A.

13. 10. 2012

Simulation models of selected inverted pendula systems (dynamic-masked *Simulink* library blocks)

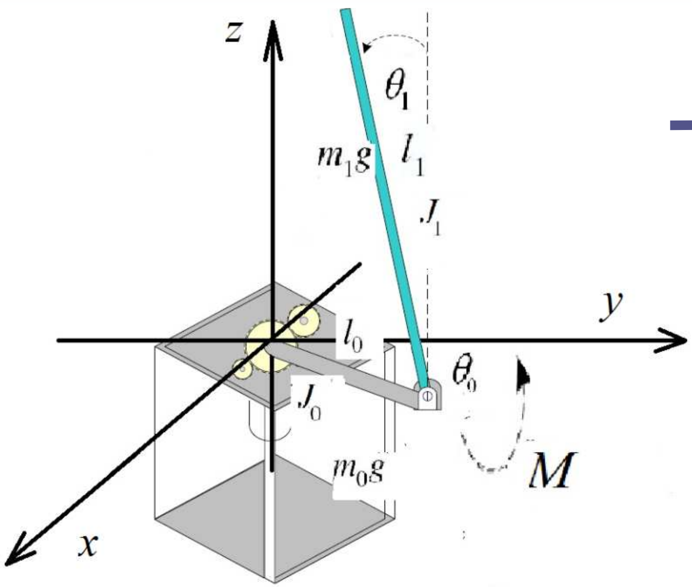


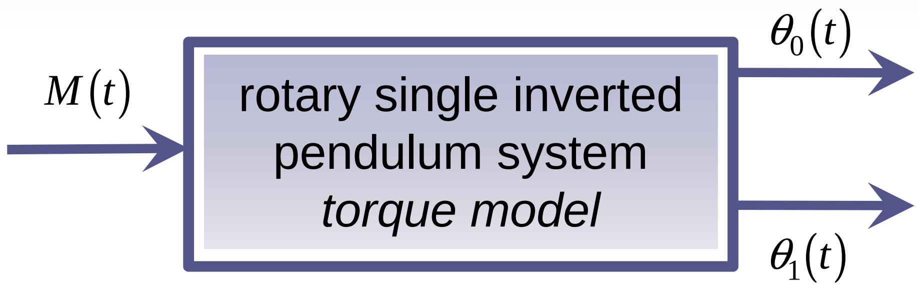
A.

Rotary single inverted pendulum system - scheme and generated mathematical model

$$\begin{pmatrix} J_0 + m_1 l_0^2 + \frac{1}{4} m_1 l_1^2 \sin^2\theta_1(t) & \frac{1}{2} m_1 l_0 l_1 \cos\theta_1(t) \\ \frac{1}{2} m_1 l_0 l_1 \cos\theta_1(t) & J_1 \end{pmatrix} \begin{pmatrix} \ddot{\theta}_0(t) \\ \ddot{\theta}_1(t) \end{pmatrix} + \begin{pmatrix} \delta_0 + \frac{1}{4} m_1 l_1^2 \dot{\theta}_1(t) \sin 2\theta_1(t) & -\frac{1}{2} m_1 l_0 l_1 \dot{\theta}_1(t) \sin\theta_1(t) \\ -\frac{1}{8} m_1 l_1^2 \dot{\theta}_0(t) \sin 2\theta_1(t) & \delta_1 \end{pmatrix} \begin{pmatrix} \dot{\theta}_0(t) \\ \dot{\theta}_1(t) \end{pmatrix} +$$

$$+ \begin{pmatrix} 0 \\ -\frac{1}{2} m_1 g l_1 \sin\theta_1(t) \end{pmatrix} = \begin{pmatrix} M(t) \\ 0 \end{pmatrix}$$

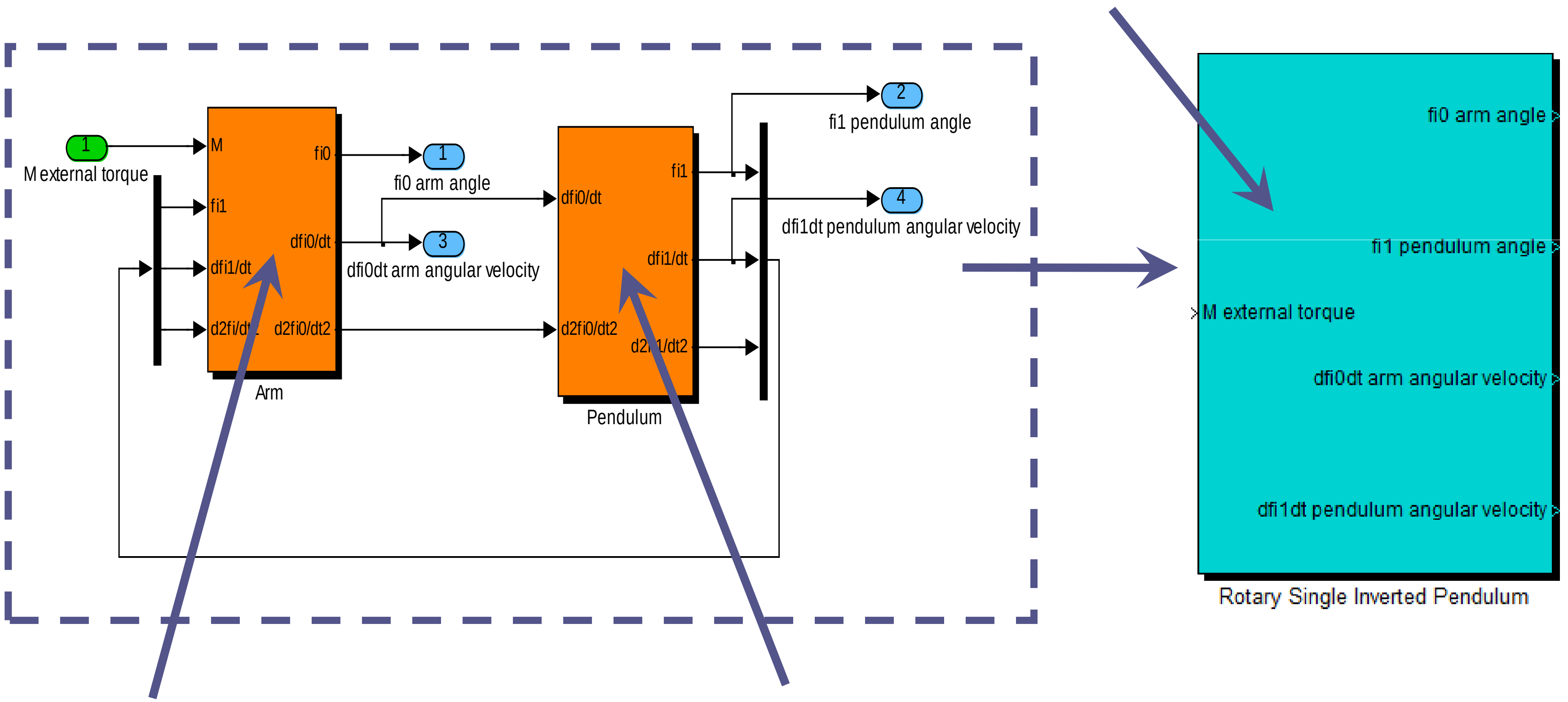




$$m_0 = 0{,}5kg \quad m_1 = 0{,}275kg$$
$$l_0 = 0{,}6m \quad l_1 = 0{,}5m$$
$$\delta_0 = 0{,}3kgs^{-1} \quad \delta_1 = 0{,}011458s^{-1}$$

A.
Rotary single inverted pendulum system - library block structure

Rotary Single Inverted Pendulum function block



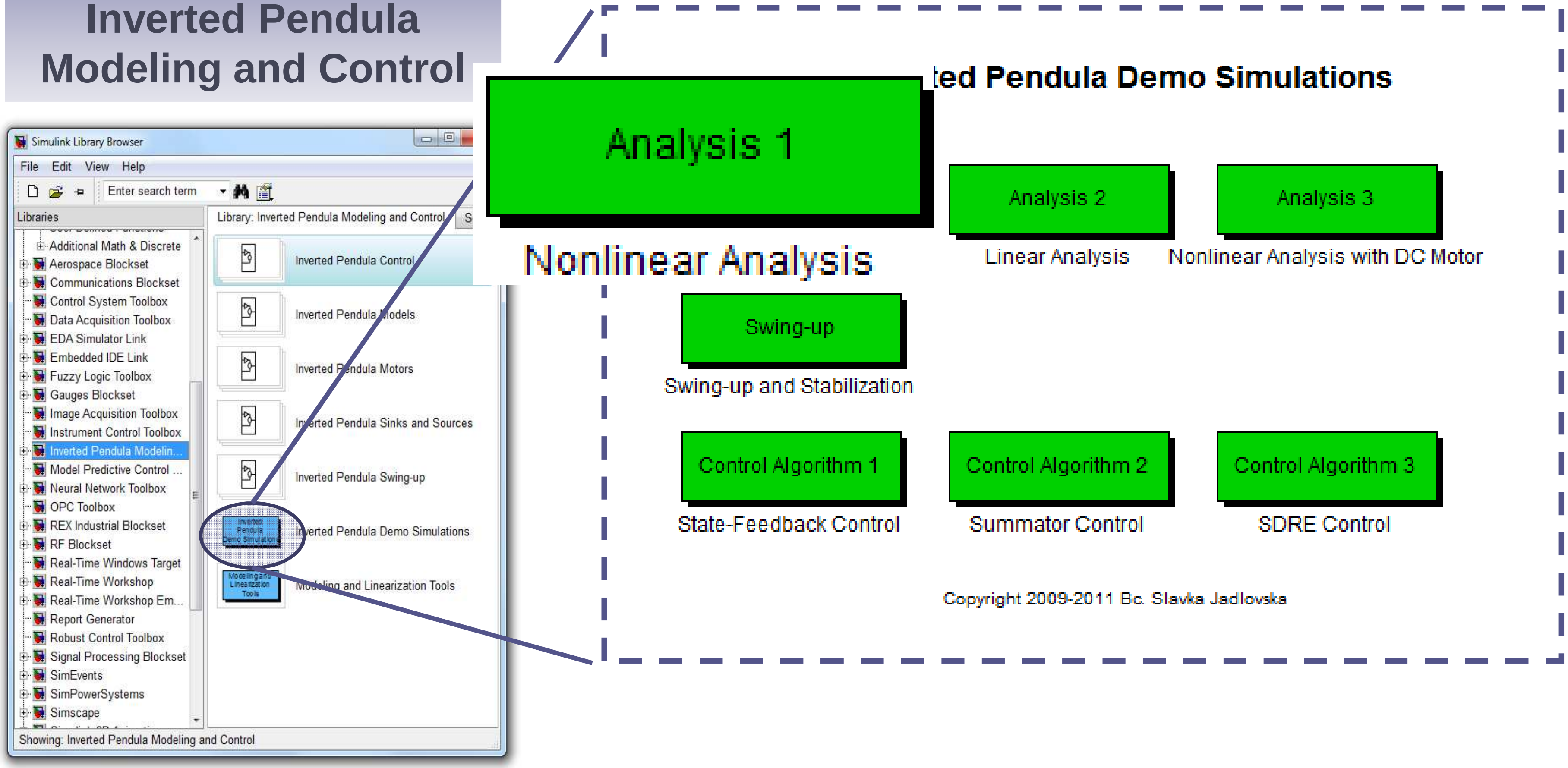
rotary arm subsystem

pendulum subsystem

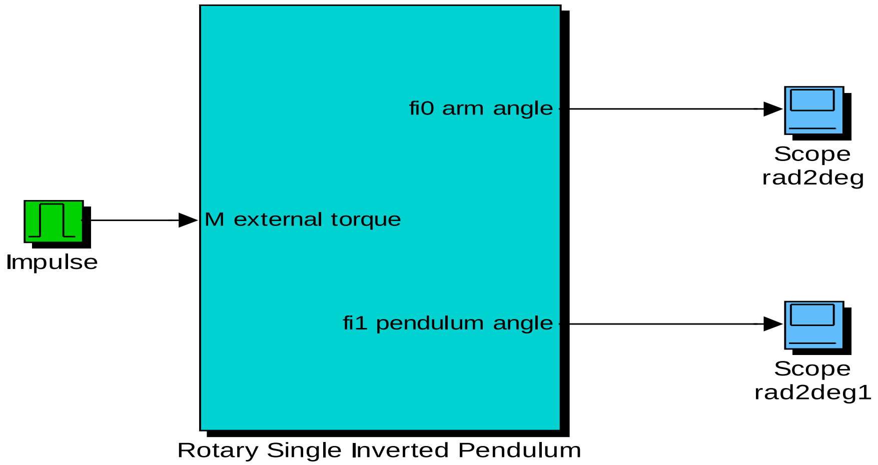
A.

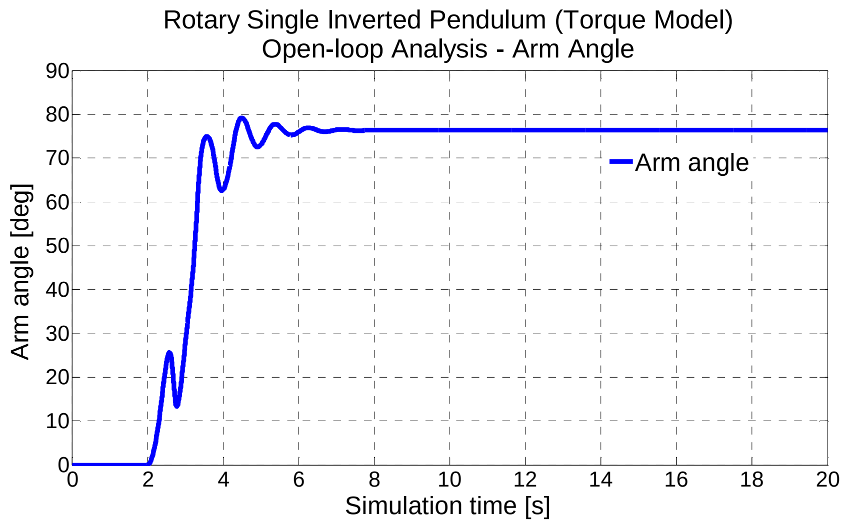
Demo Simulations I: Open-loop dynamical analysis for nonlinear force-torque models of inverted pendula systems

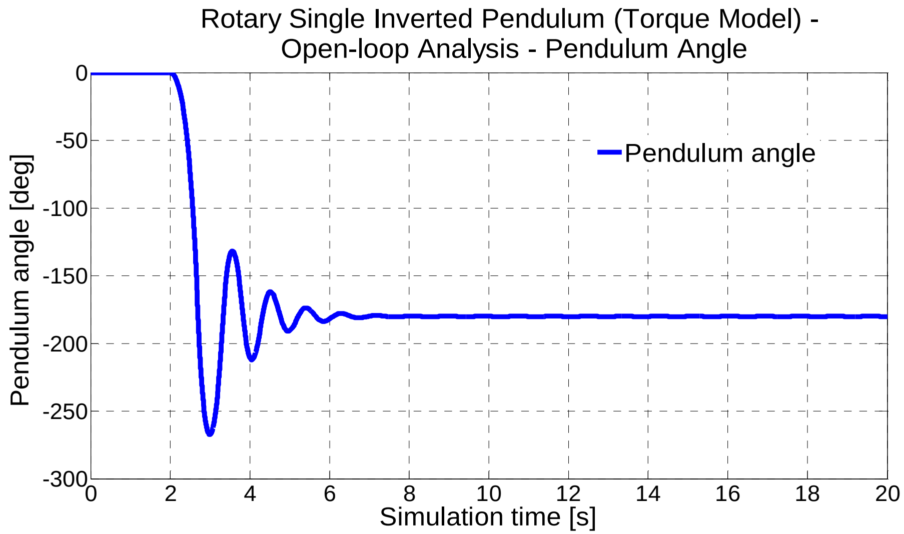
Inverted Pendula Modeling and Control



A. Rotary single inverted pendulum (torque model) - open-loop dynamical analysis





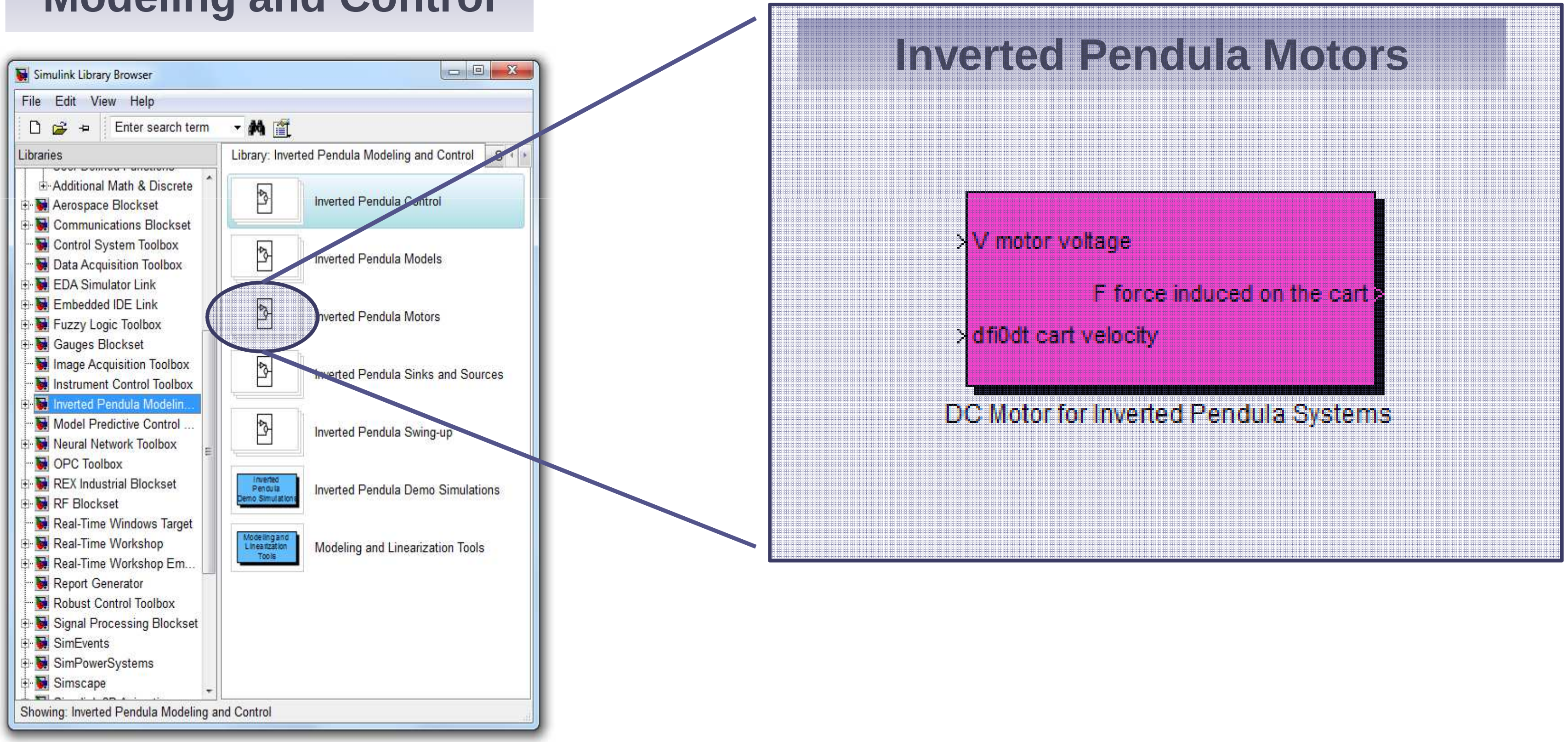


A.

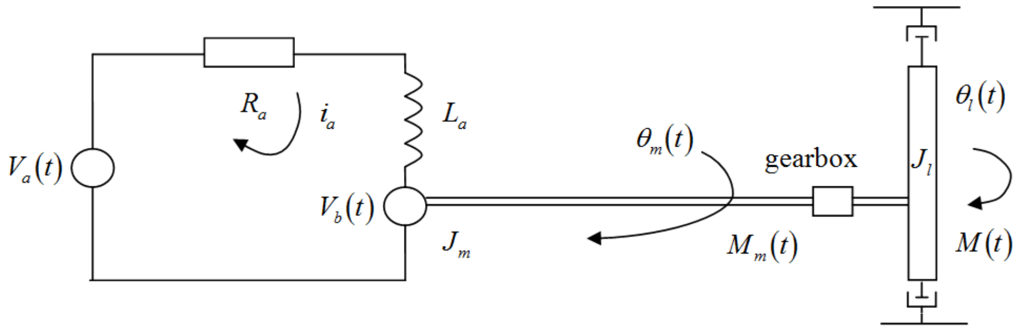
13. 10. 2012

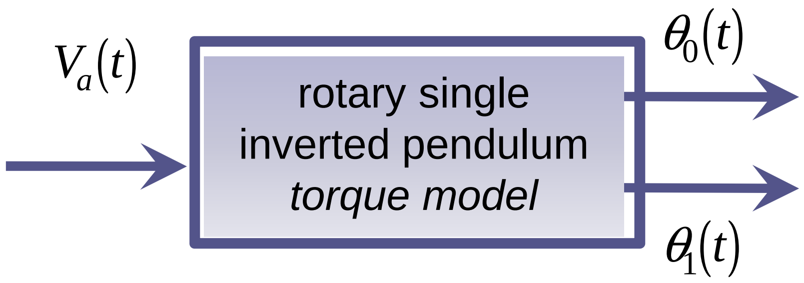
Simulation model of actuating mechanism (DC motor) for inverted pendula systems (*Simulink* library block)

Inverted Pendula Modeling and Control



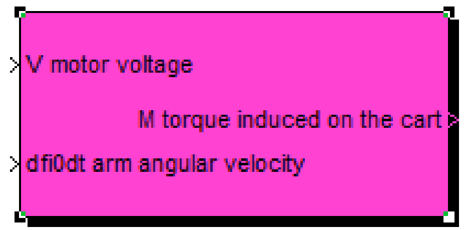
A. *DC motor model* in form of a voltage-to-torque conversion relationship





rotary single inverted pendulum system:

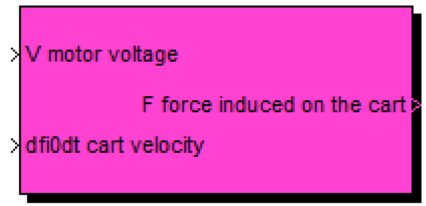
$$M(t) = \frac{k_m k_g}{R_a} V_a(t) - \frac{k_m^2 k_g^2}{R_a} {}^r\dot{\theta}_0(t)$$



DC Motor for Inverted Pendula Systems

classical single inverted pendulum system

$$F(t) = \frac{k_m k_g}{R_a r} V_a(t) - \frac{k_m^2 k_g^2}{R_a r^2} {}^c\dot{\theta}_0(t)$$



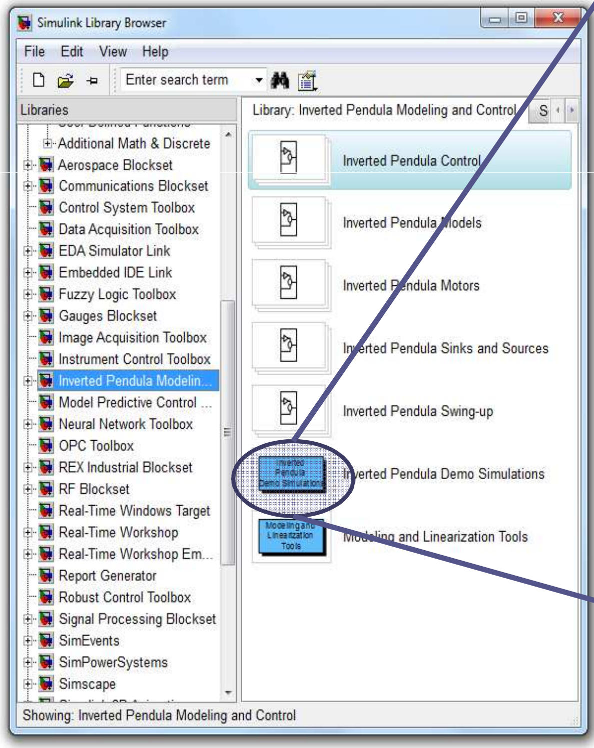
DC Motor for Inverted Pendula Systems

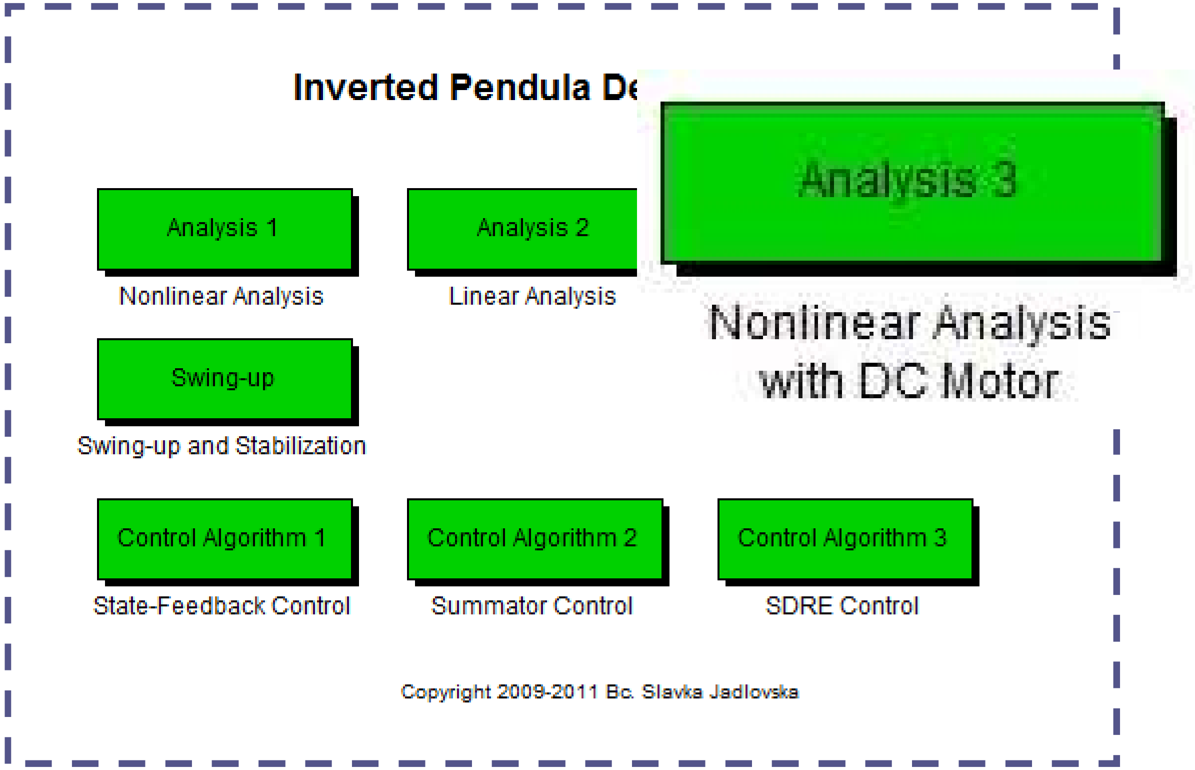
A.

13. 10. 2012

Demo Simulations II: Open-loop dynamical analysis for nonlinear voltage models of inverted pendula systems

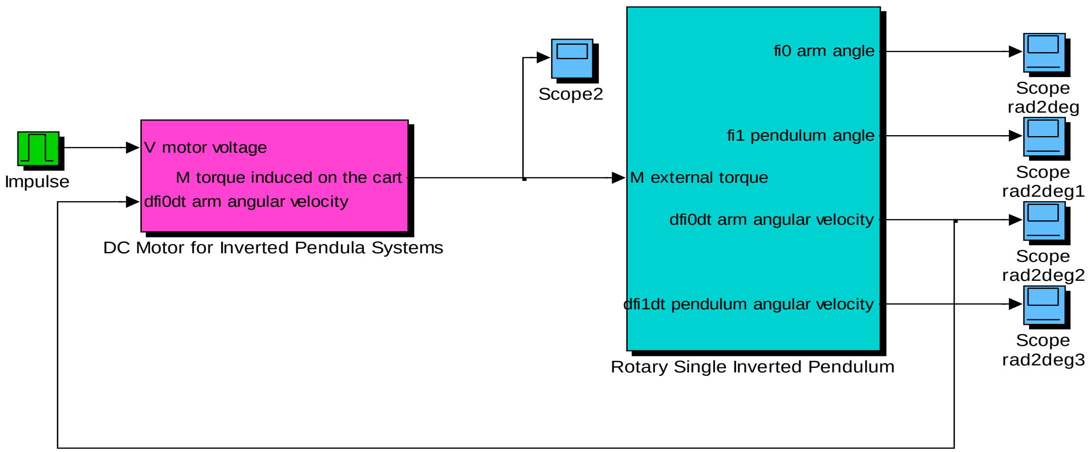
Inverted Pendula Modeling and Control

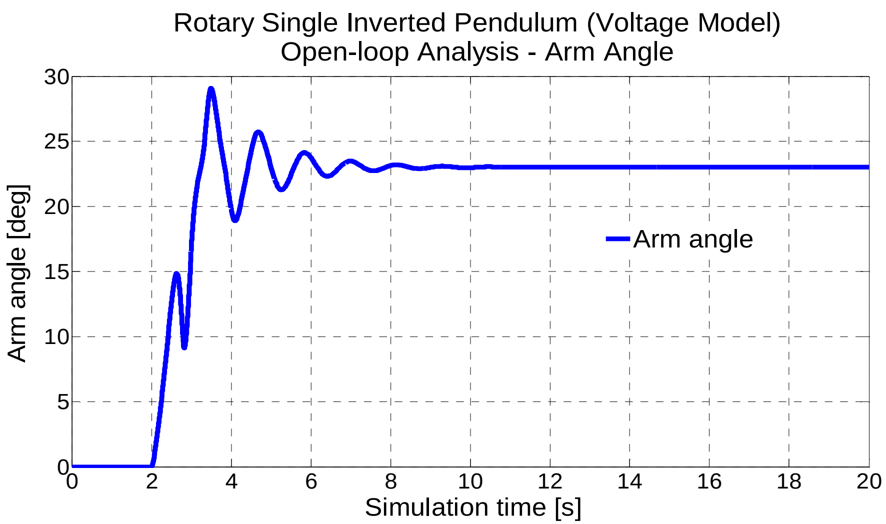


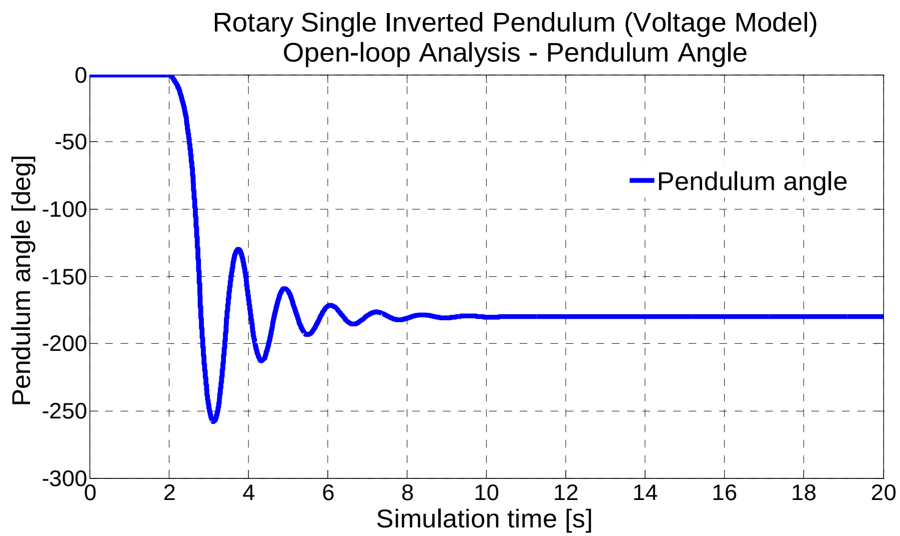


A.

Rotary single inverted pendulum (voltage model) - open-loop dynamical analysis







B.
Stabilization of the Rotary Single Inverted Pendulum via State-Feedback Control Techniques

B.

13. 10. 2012

Control techniques for inverted pendula systems supported by the *IPMaC* block library

IPMaC

Control Objective

Derivator

Stabilization of the pendulum in the upright position

Inverted Pendula Models

Swing-up into the upright position

Linearizator

Linear state-space pendulum model

- pole-placement algorithm
- linear quadratic optimal control method (LQR)
- *PID/PSD algorithm*

Nonlinear state-space pendulum model

- state-dependent Riccati equation
- *input-output linearization*
- *artificial intelligence*

Nonlinear state-space pendulum model

- energy-based methods
- *artificial intelligence*

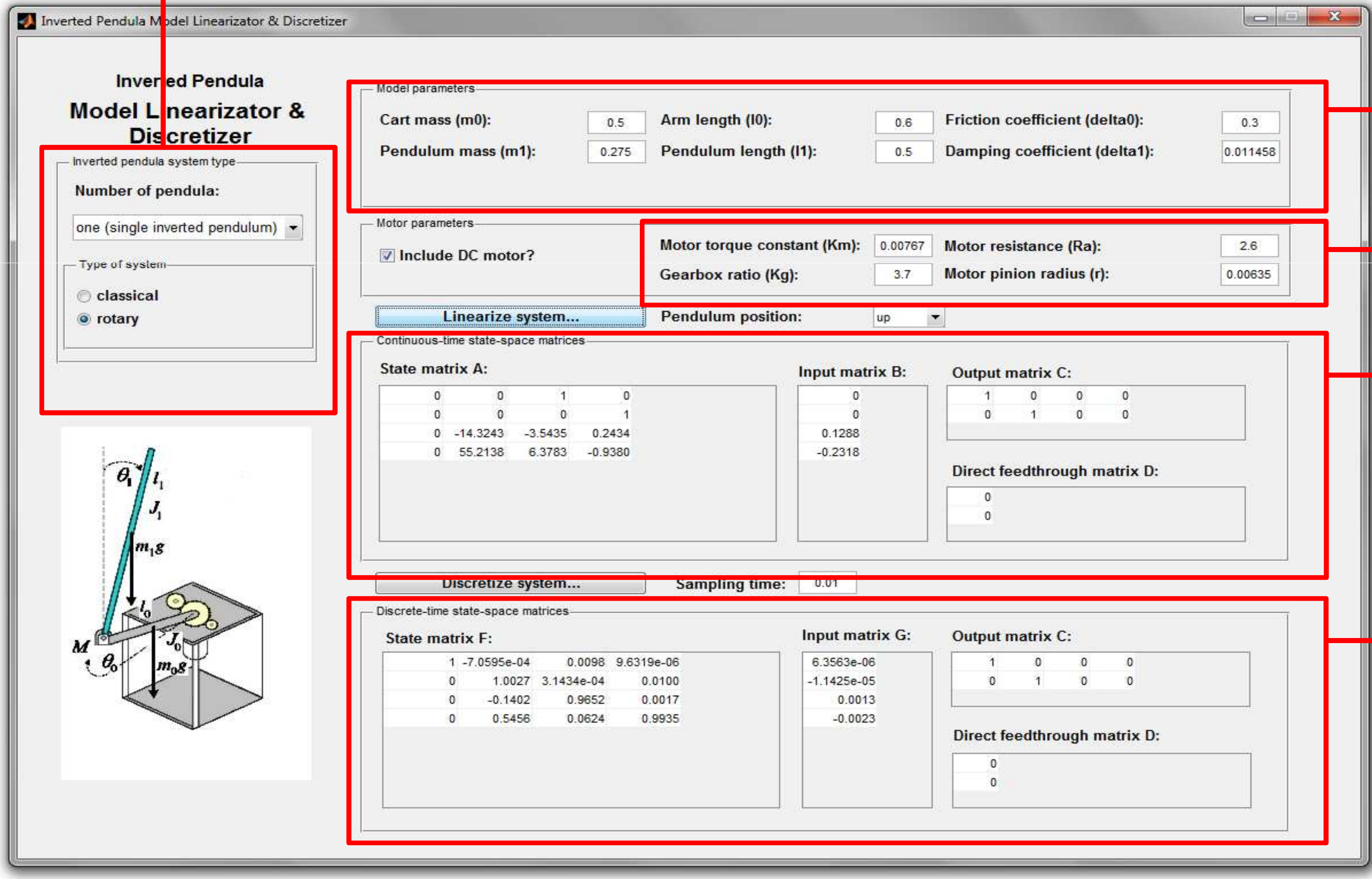
Inverted Pendula Control

Inverted Pendula Swing-up

13. 10. 2012

B. 1)
Inverted Pendula Model Linearizator & Discretizer

selection of system type & number of pendulum links



system parameters

motor parameters

matrices of the linearized system

matrices of the discretized system

B. 1)
State-space description of inverted pendula systems

$$\boldsymbol{x}(t)=\left(\theta(t)\quad \dot{\theta}(t)\right)^T$$

state vector

Standard minimal ODE form

$$\boldsymbol{M}(\theta(t))\ddot{\theta}(t)+\boldsymbol{N}\left(\theta(t),\dot{\theta}(t)\right)\dot{\theta}(t)+\boldsymbol{P}(\theta(t))=\boldsymbol{V}(t)$$

$$\ddot{\theta}(t)=\left(\boldsymbol{M}(\theta(t))\right)^{-1}\left(V(t)-\boldsymbol{N}\left(\theta(t),\dot{\theta}(t)\right)\dot{\theta}(t)+\boldsymbol{P}(\theta(t))\right)$$

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{f}\left(\boldsymbol{x}(t),u(t),t\right)$$

$$\boldsymbol{y}(t)=\boldsymbol{g}\left(\boldsymbol{x}(t),u(t),t\right)$$

nonlinear state-space description

B. 1) Linear approximation of inverted pendula systems (upright position)

13. 10. 2012

$$\mathbf{x}(t) = \left(\boldsymbol{\theta}(t) \quad \dot{\boldsymbol{\theta}}(t)\right)^T$$

state vector

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}\left(\mathbf{x}(t), u(t), t\right)$$
$$\boldsymbol{y}(t) = \boldsymbol{g}\left(\mathbf{x}(t), u(t), t\right)$$

$$\mathbf{x}_S = \mathbf{0}^T$$

$$f_i^*\left(\mathbf{x}(t), u(t)\right) \approx f_i\left(\mathbf{x}_S, u_S\right) + \sum_{k=1}^{2n+2} \left.\frac{\partial f_i\left(\mathbf{x}(t), u(t)\right)}{\partial x_k(t)}\right|_{\mathbf{x}_S, u_S} \left(x_k(t) - x_{kS}\right) + \left.\frac{\partial f_i\left(\mathbf{x}(t), u(t)\right)}{\partial u(t)}\right|_{\mathbf{x}_S, u_S} \left(u(t) - u_S\right)$$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t)$$
$$y(t) = \mathbf{C}x(t) + du(t)$$

matrices_single.m

matrices_double.m

matrices_rotary.m

B. 1)

Linearized (continuous-time) and discretized (discrete-time) state-space description of rotary single inverted pendulum

```
[A,B,C,D] = matrices_rotary([0.5 0.6 0.275 0.5
0.3 0.011458],'up','m')
```

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -14{,}3243 & -3{,}5435 & 0{,}2434 \\ 0 & 55{,}2138 & 6{,}3783 & -0{,}938 \end{pmatrix} \quad \boldsymbol{b} = \begin{pmatrix} 0 \\ 0 \\ 0{,}1288 \\ -0{,}2318 \end{pmatrix} \quad \boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \boldsymbol{d} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

```
Ts=0.01
[F,G]=c2d(A,B,Ts)
```

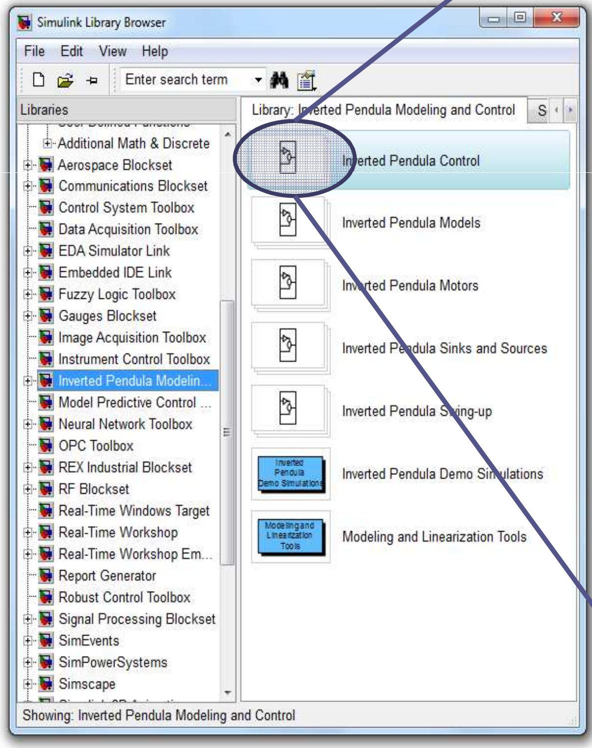
$$\boldsymbol{F} = \begin{pmatrix} 1 & -0{,}0007 & 0{,}0098 & 0 \\ 0 & 1{,}0027 & 0{,}0004 & 0{,}01 \\ 0 & -0{,}1402 & 0{,}9652 & 0{,}0017 \\ 0 & 0{,}5456 & 0{,}0624 & 0{,}9935 \end{pmatrix} \quad \boldsymbol{g} = \begin{pmatrix} 0{,}000006 \\ -0{,}000011 \\ 0{,}0013 \\ -0{,}0023 \end{pmatrix} \quad \boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \boldsymbol{d} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

B. 2)

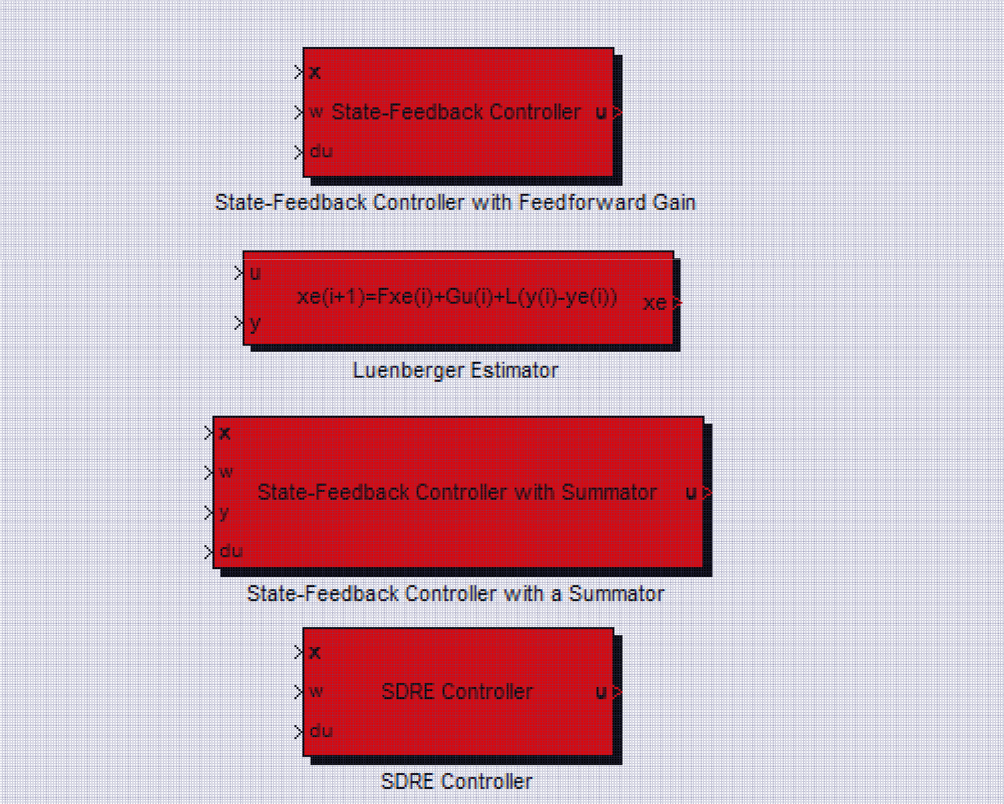
13. 10. 2012

Software support for state-feedback controller design (*Simulink* library blocks)

Inverted Pendula Modeling and Control

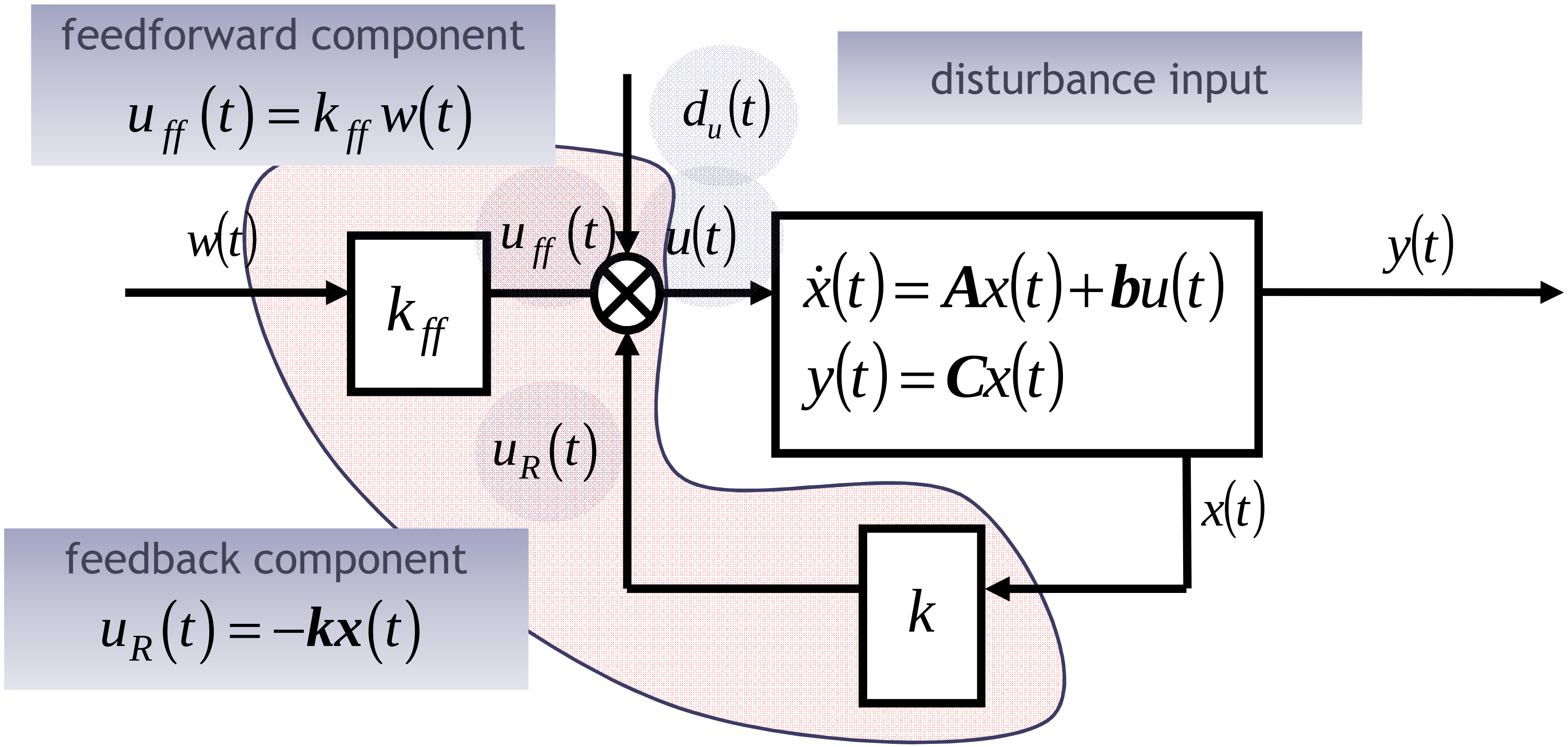


Inverted Pendula Control



B. 2)

State-feedback control - basic control scheme



$$u(t) = u_R(t) + u_{ff}(t) + d_u(t) = -\boldsymbol{kx}(t) + k_{ff} w(t) + d_u(t)$$

B. 2)

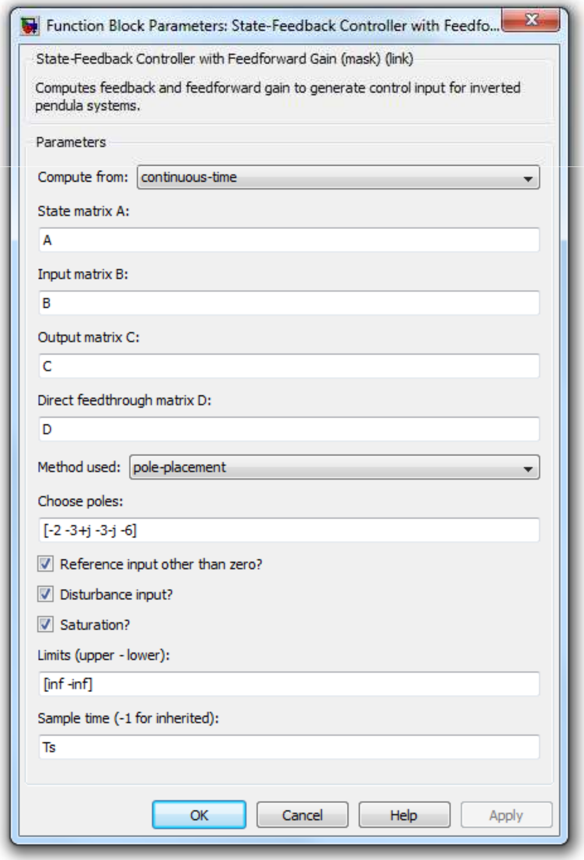
Library block *State-Feedback Controller with FeedForward Gain*

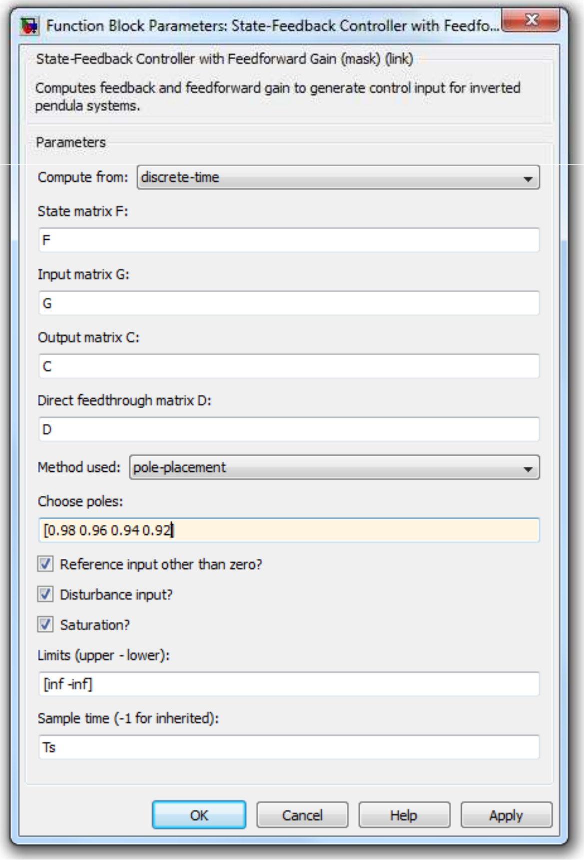
Computation of the feedback gain vector *k*

a) using the pole-placement algorithm

continuous-time state-space model

discrete-time state-space model



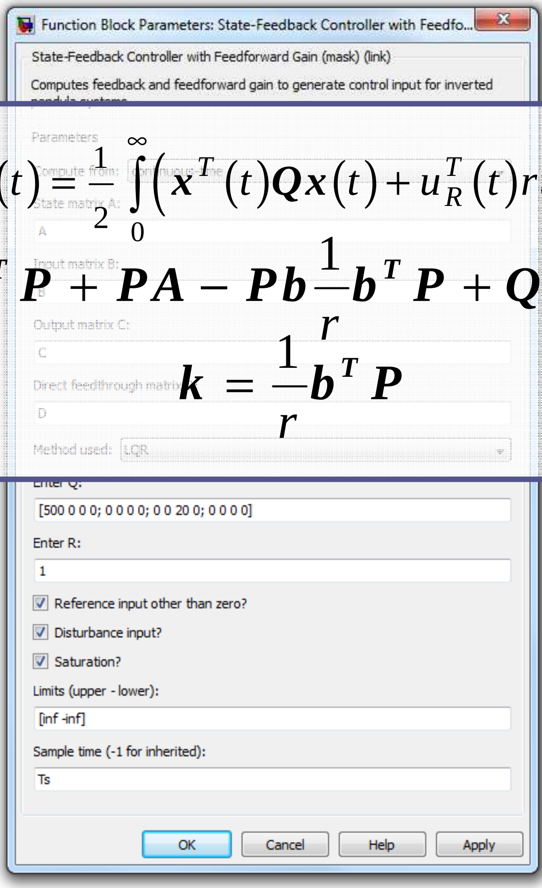


B. 2) Library block *State-Feedback Controller with FeedForward Gain*

Computation of the feedback gain vector *k*

2) using the linear quadratic optimal control method (LQR)

continuous-time state-space model

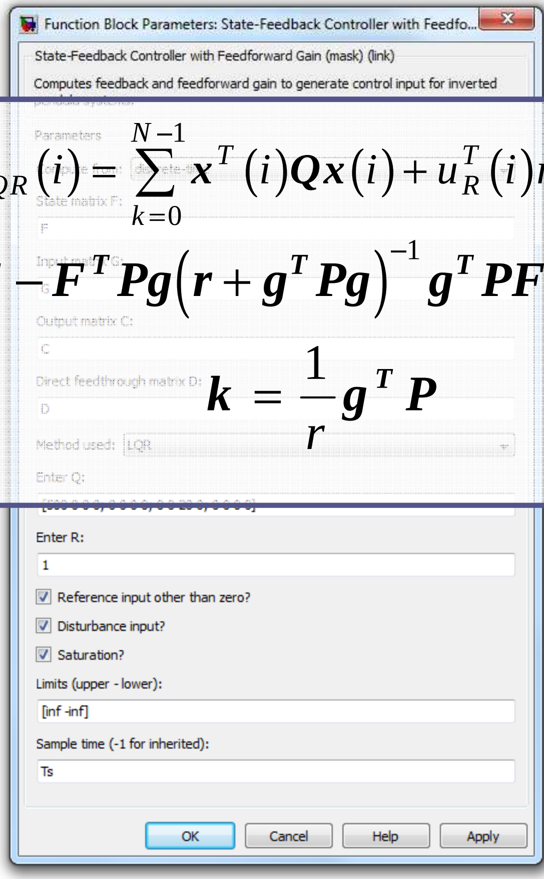


$$J_{LQR}(t)=\frac{1}{2}\int_0^{\infty}\left(\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)+u_R^T(t)ru_R(t)\right)dt$$

$$\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}-\boldsymbol{P}\boldsymbol{b}\frac{1}{r}\boldsymbol{b}^T\boldsymbol{P}+\boldsymbol{Q}=0$$

$$\boldsymbol{k}=\frac{1}{r}\boldsymbol{b}^T\boldsymbol{P}$$

discrete-time state-space model



$$J_{LQR}(i)=\sum_{k=0}^{N-1}\mathbf{x}^T(i)\mathbf{Q}\mathbf{x}(i)+u_R^T(i)ru_R(i)$$

$$\boldsymbol{F}^T\boldsymbol{P}\boldsymbol{F}-\boldsymbol{F}^T\boldsymbol{P}\boldsymbol{g}\left(\boldsymbol{r}+\boldsymbol{g}^T\boldsymbol{P}\boldsymbol{g}\right)^{-1}\boldsymbol{g}^T\boldsymbol{P}\boldsymbol{F}+\boldsymbol{Q}=0$$

$$\boldsymbol{k}=\frac{1}{r}\boldsymbol{g}^T\boldsymbol{P}$$

B. 2)

Library block *State-Feedback Controller with FeedForward Gain* - supported control objectives

- Initial deflection of the pendulum (nonzero initial conditions)
- Time-constrained and permanent disturbance input compensation ($d_u \neq 0$)
- Tracking a desired reference trajectory by the cart or arm

$$k_{ff} = \frac{-1}{c_1 \left(\boldsymbol{A} - \boldsymbol{bk} \right)^{-1} \boldsymbol{b}} \qquad k_{ffD} = \frac{1}{c_1 \left(\boldsymbol{I}_{2n+2} - \left(\boldsymbol{F} - \boldsymbol{gk} \right) \right)^{-1} \boldsymbol{g}}$$

State-Feedback Controller with Feedforward Gain

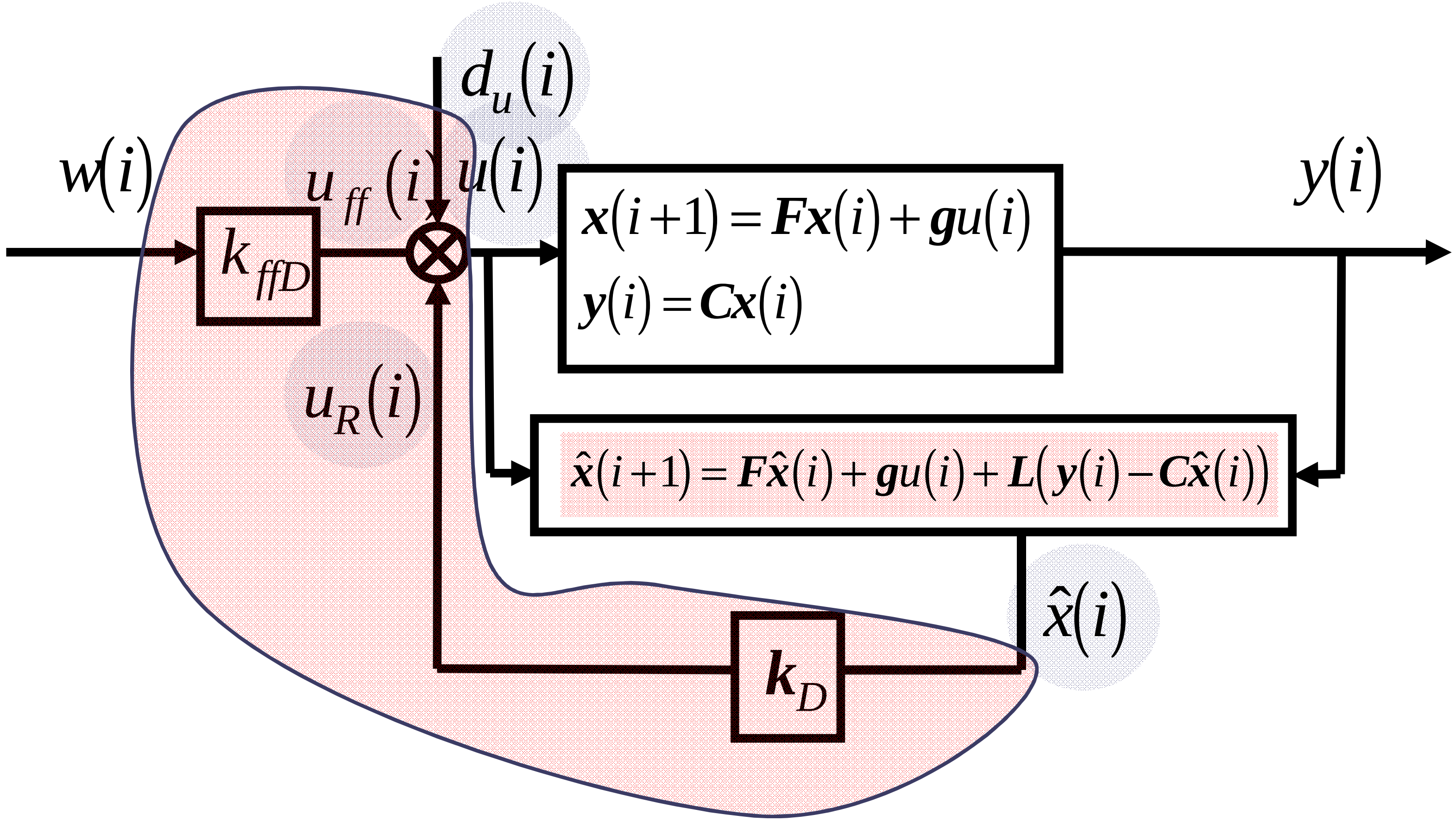
State-Feedback Controller with Feedforward Gain

State-Feedback Controller with Feedforward Gain

State-Feedback Controller with Feedforward Gain

B. 2)

State-feedback control with a discrete-time state estimator



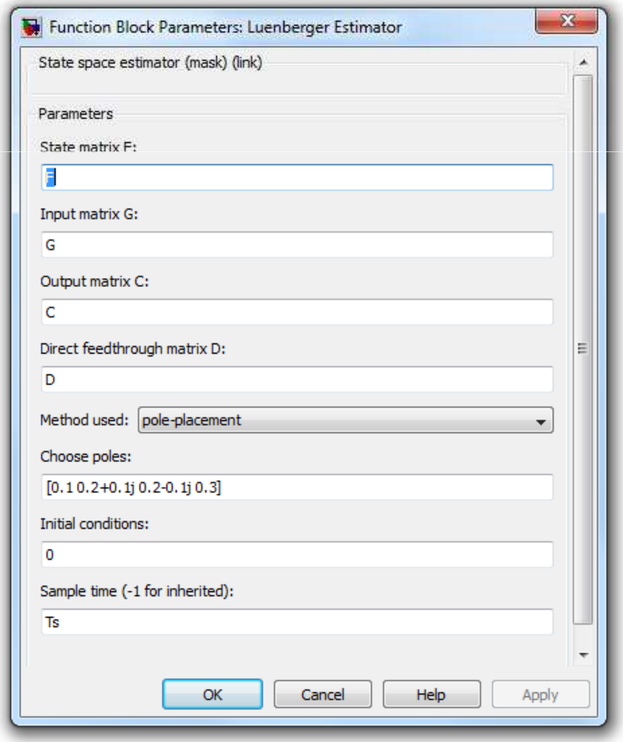
$$u(i) = u_R(i) + u_{ff}(i) + d_u(i) = -\mathbf{k}_D\hat{\mathbf{x}}(i) + k_{ffD}w(i) + d_u(i)$$

B. 2)
Library block *Luenberger Estimator*

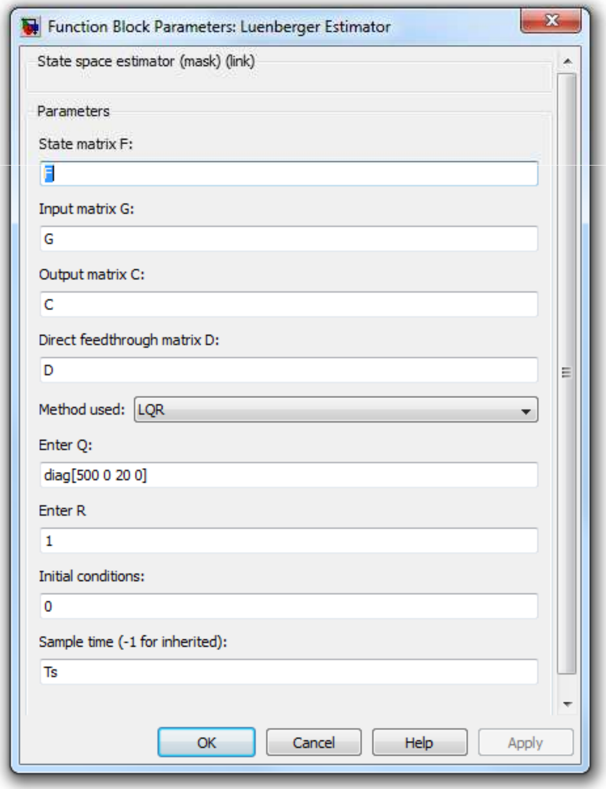
13. 10. 2012

Computing the state estimator matrix ***L***

pole-placement algorithm



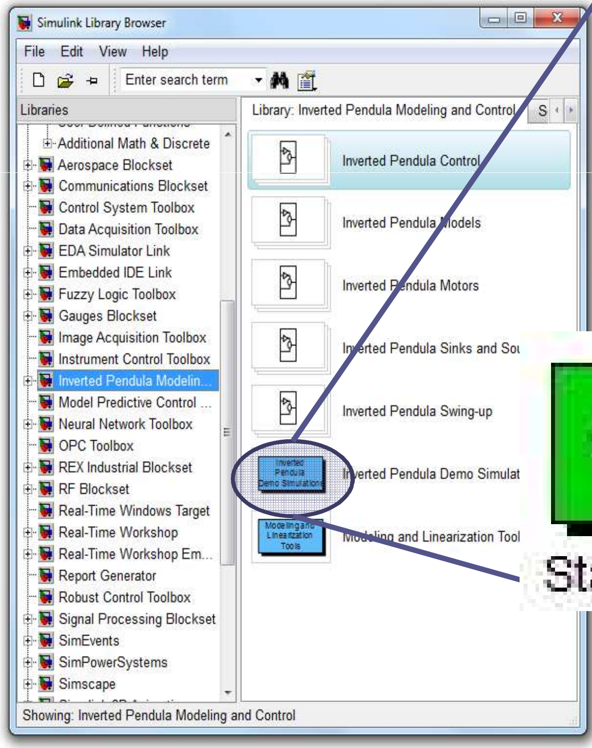
optimal control method (LQR)

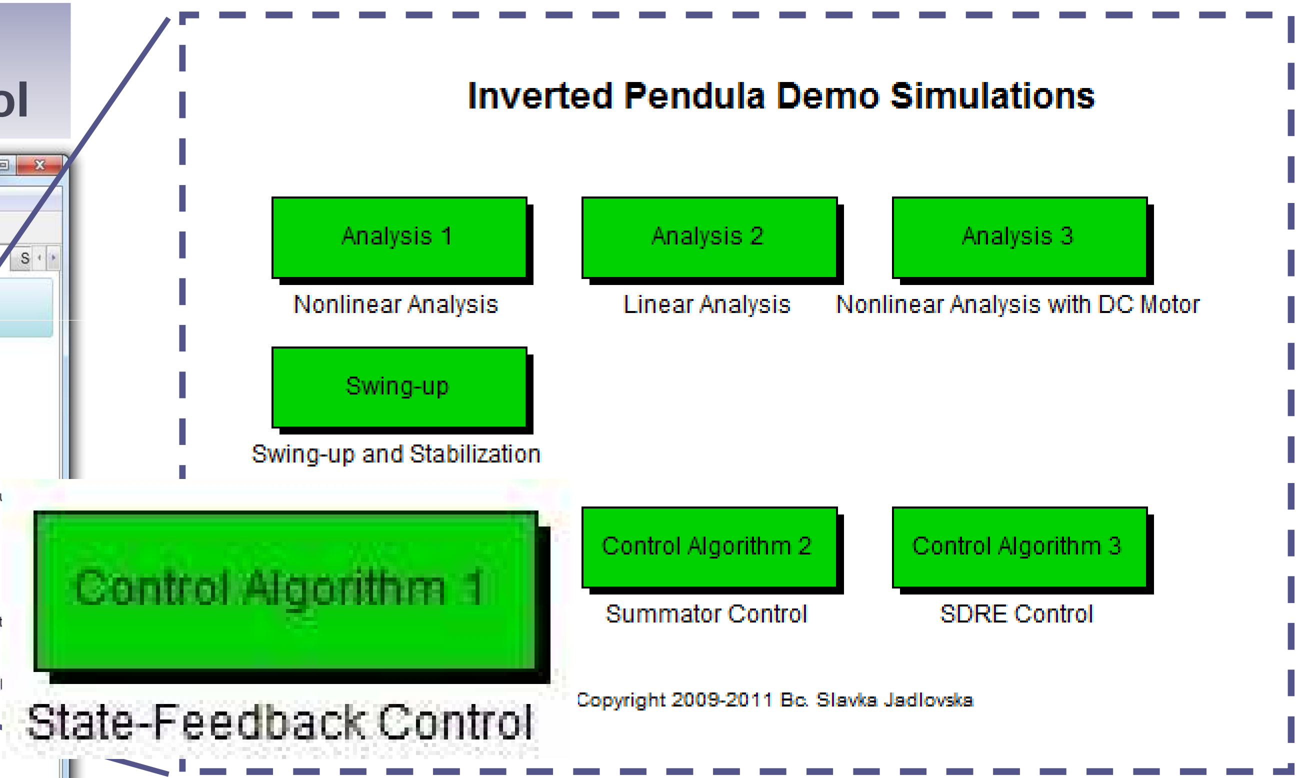


B. 2)

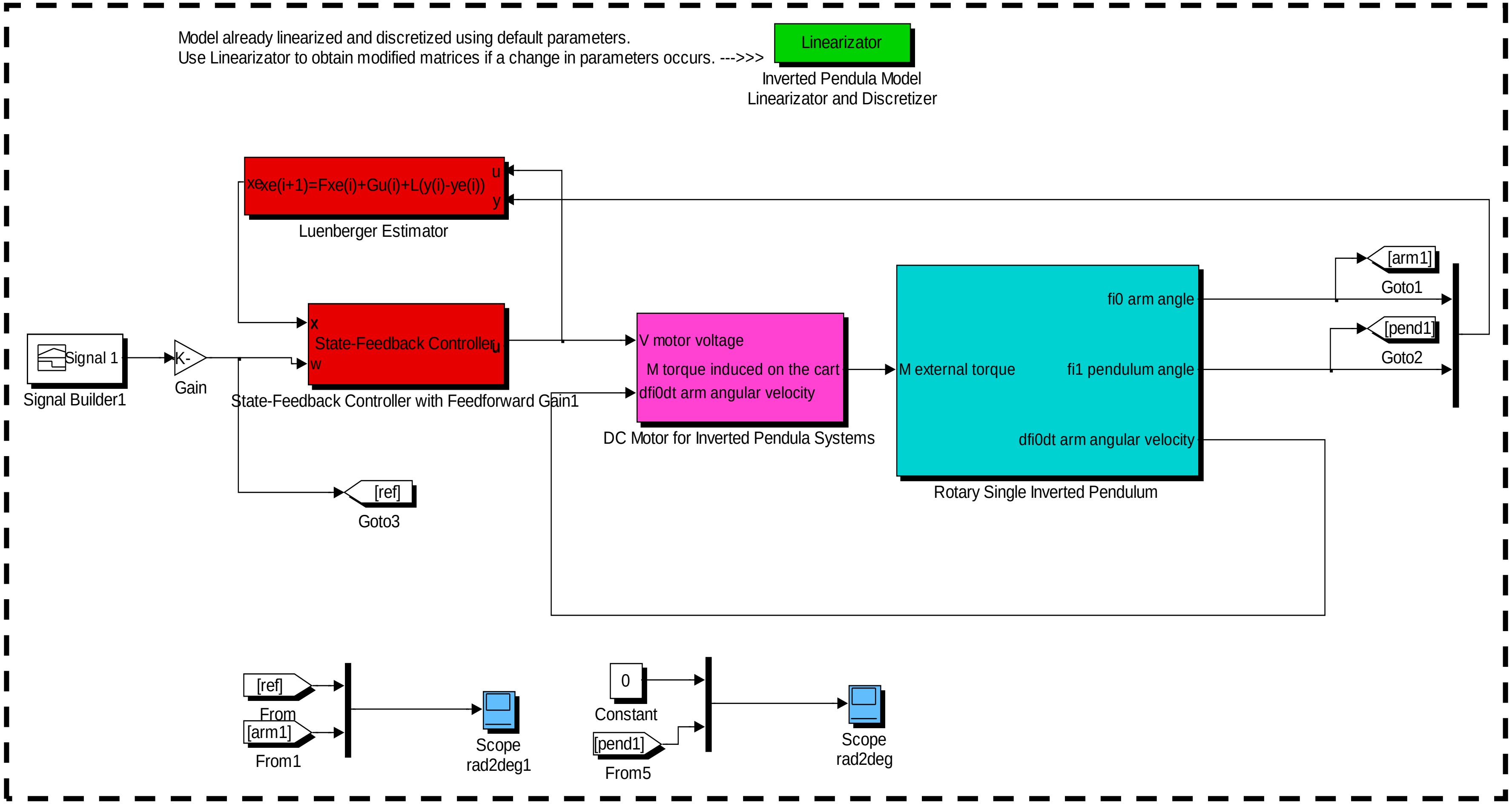
Demo Simulations III: State-feedback control for nonlinear force-torque / voltage models of inverted pendula systems

Inverted Pendula Modeling and Control



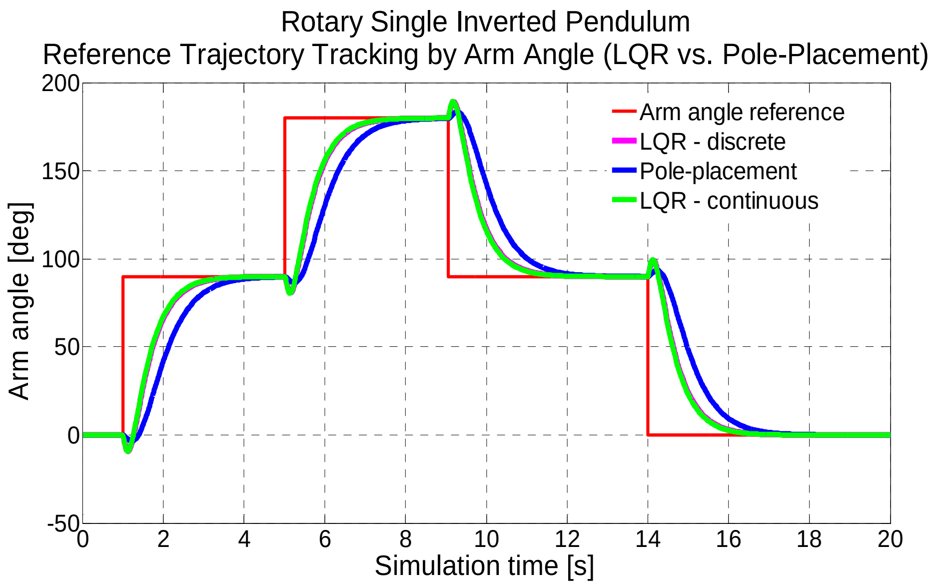


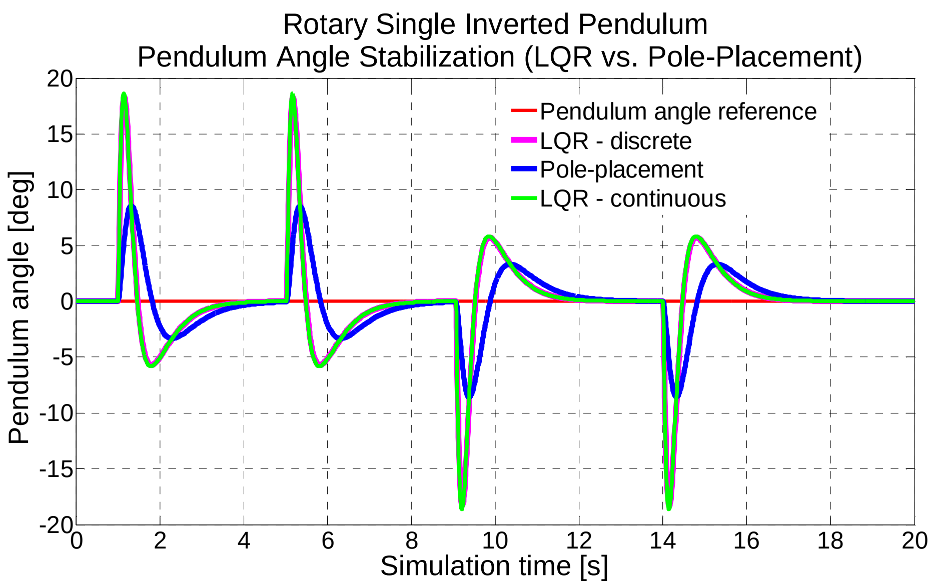
B. 2) Rotary single inverted pendulum (voltage model) - general simulation setup for state-feedback control



B. 2) Rotary single inverted pendulum - simulation results for LQR control compared to pole-placement

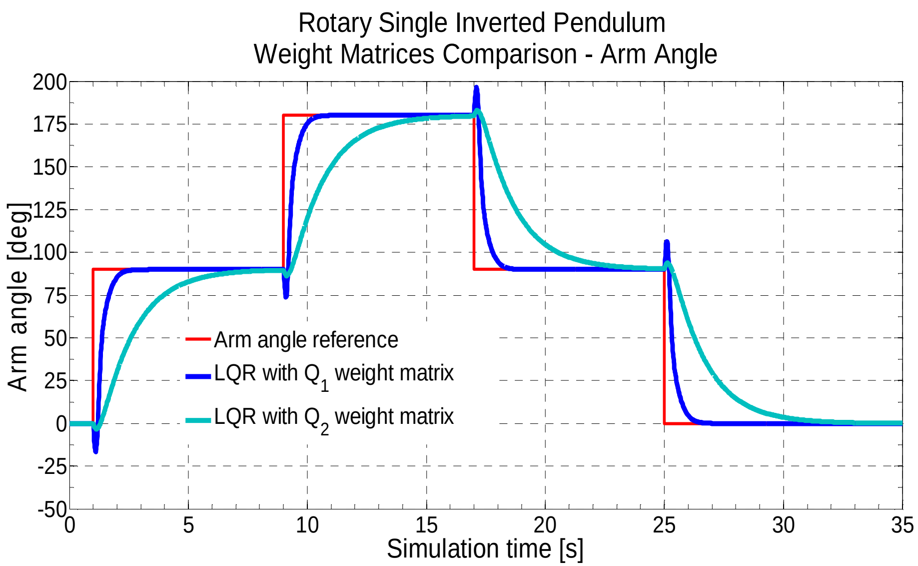
- control objective:
  - the arm should rotate for a total of half a circle and stop every quarter-turn to stabilize before returning to its initial position;
  - the pendulum should be kept upright all the time
- state-feedback control designed using continuous-time & discrete-time LQR and continuous-time pole-placement

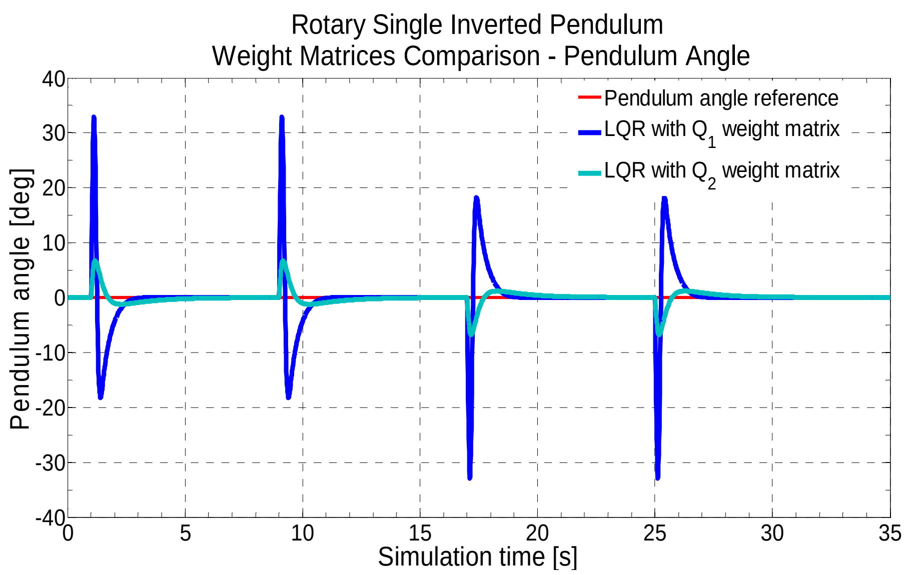




B. 2) Rotary single inverted pendulum - evaluation of the impact of weight matrices on system performance

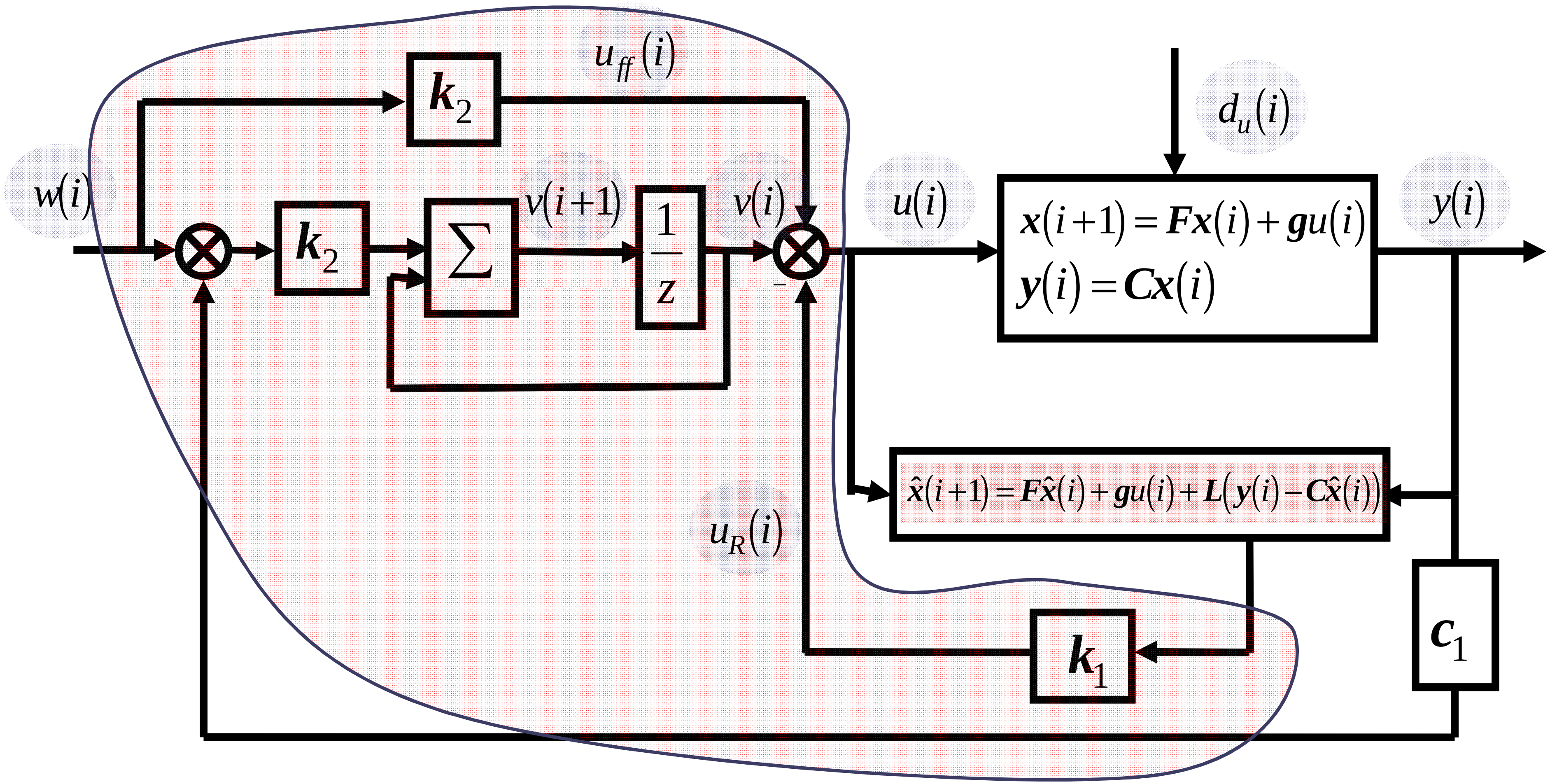
- tuning the functional weight matrices enables us to stress one of the two contradictory control objectives:
  - bringing the rotary arm into the desired position in the shortest time possible
  - stabilization of the pendulum in the upright position with the lowest possible overshoot



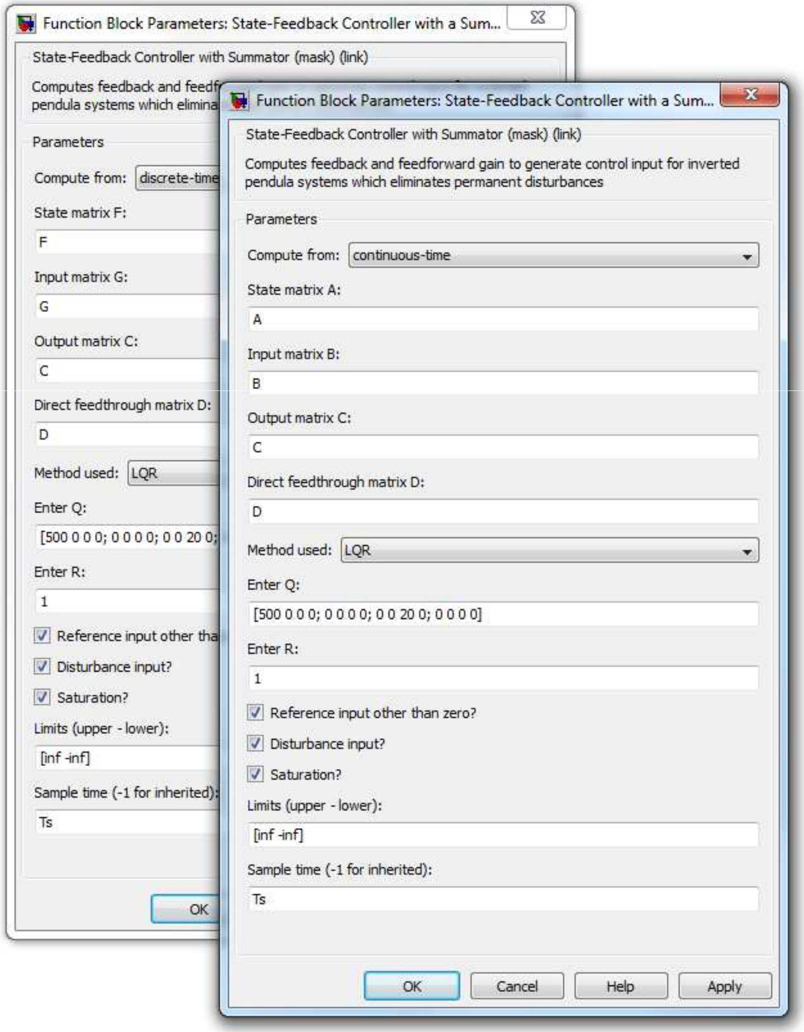


B. 2)

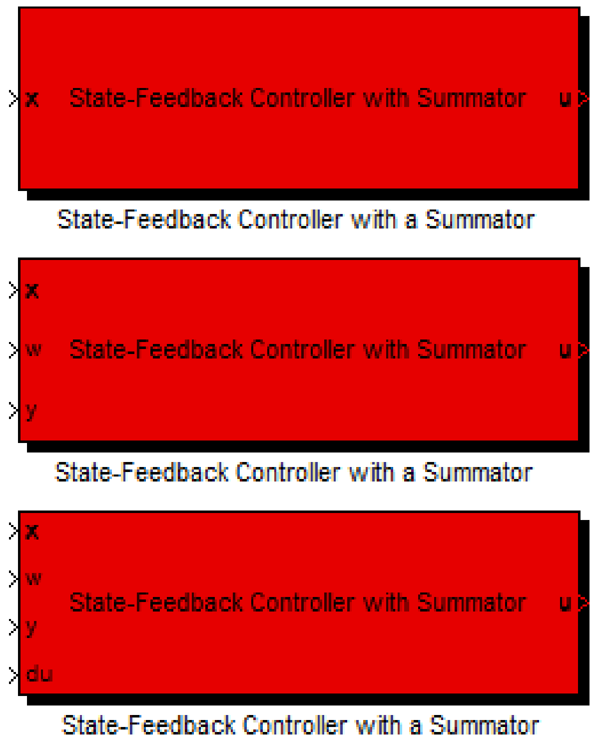
State-feedback control with permanent disturbance compensation using a summator - control scheme



B. 2)
Library block *State-Feedback Controller with a Summator*



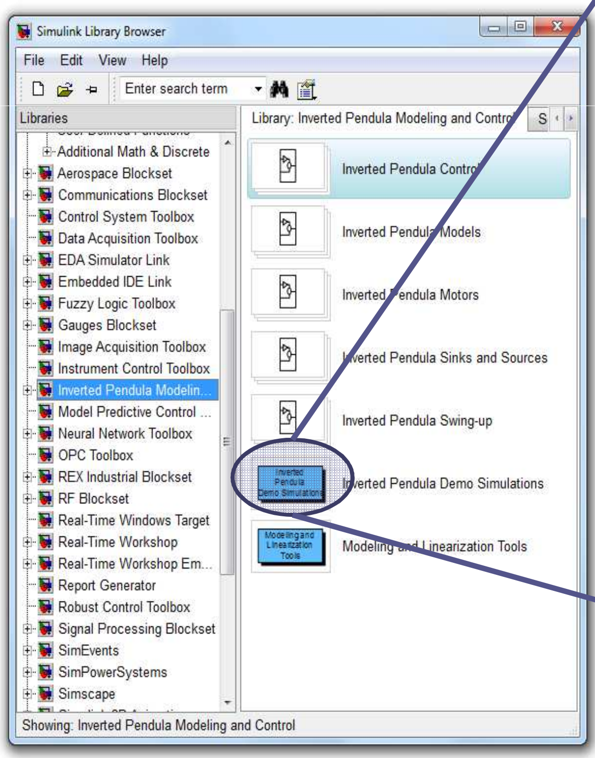
Control objectives:

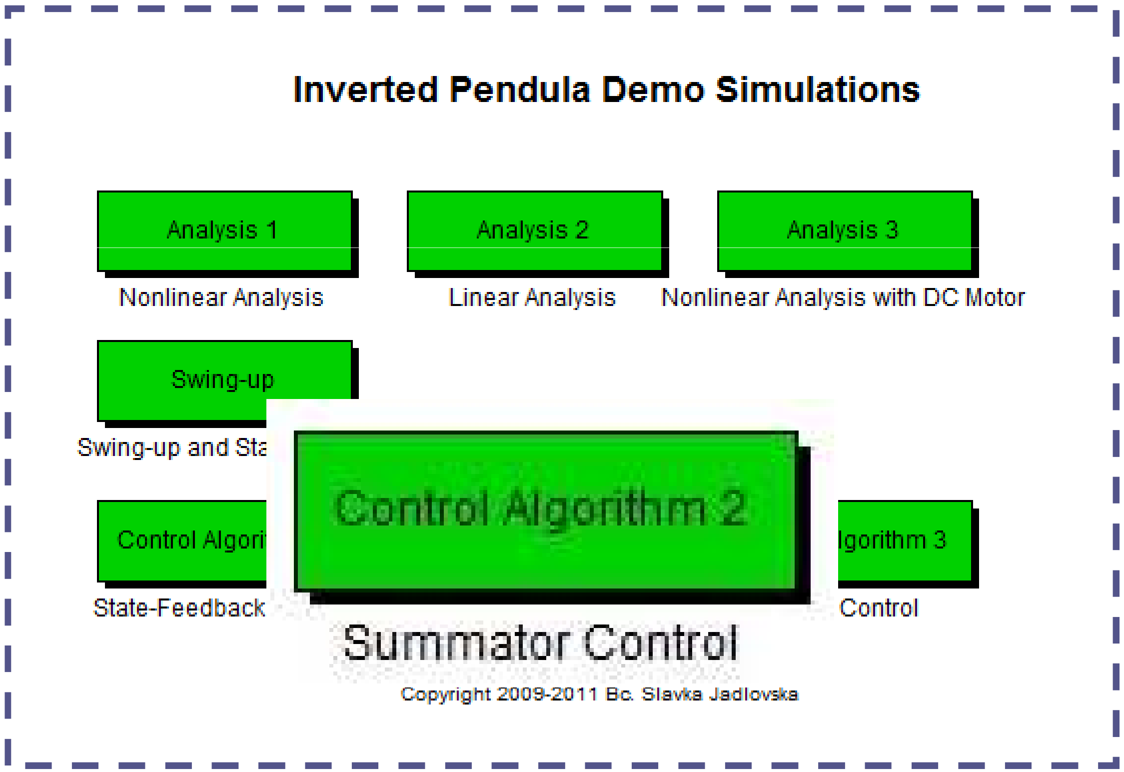


B. 2)

Demo Simulations IV: State-feedback control with a summator for nonlinear force-torque / voltage models of inverted pendula systems

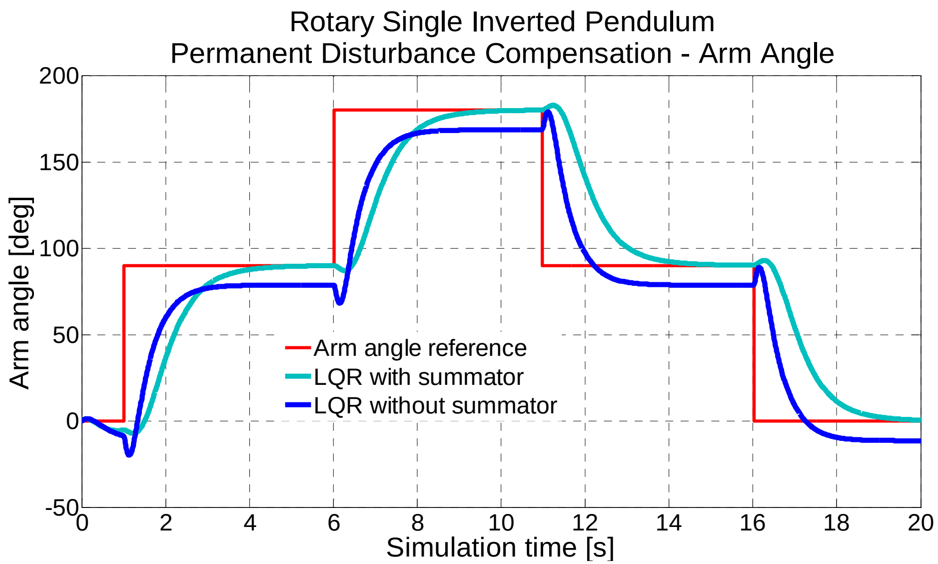
Inverted Pendula Modeling and Control

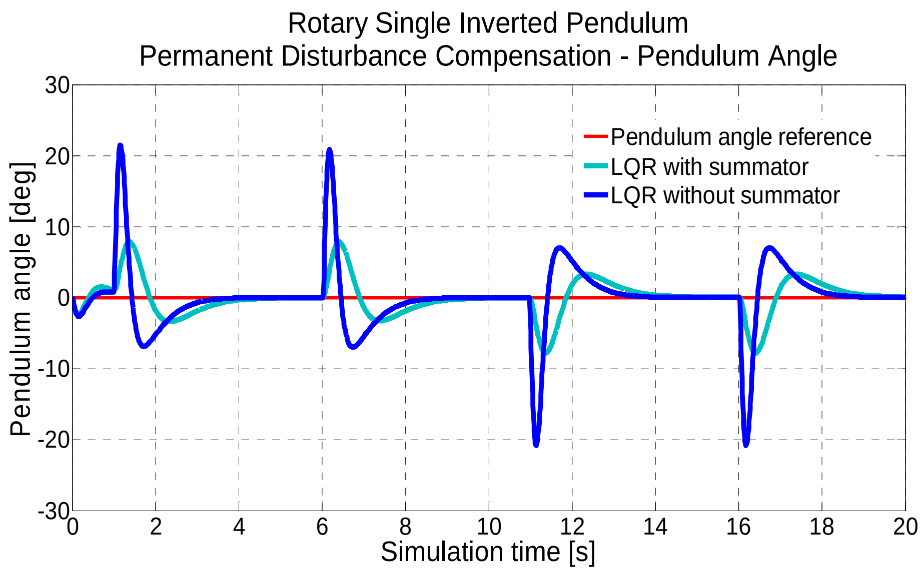




B. 2)
Rotary single inverted pendulum - simulation results for LQR control with a summator

- control objective: to track the reference trajectory & keep the pendulum upright with a constant disturbance input present
  - the **conventional LQR controller** fails to track the reference trajectory without producing steady-state error
  - permanent disturbances are successfully compensated by a **LQR algorithm with a summator included** in the control structure





Conclusion and Evaluation of Results

- comprehensive approach to modeling and control of the rotary single inverted pendulum system
- custom-designed Simulink block library ***Inverted Pendula Modeling and Control***
  - software framework for all covered issues (model derivation, open-loop analysis, linearization, state-feedback controller design)
  - provides suitable library blocks and original GUI applications to support every step of the process

Thank you for your attention.