ARGESIM SIMULATION BENCHMARKS

‘Crane and Embedded Control’ – Definition of an ARGESIM Benchmark with Implicit Modelling, Digital Control and Sensor Action

Revised Definition – Comparison 13revised

Alexander Schiftner, Felix Breitenecker, Horst Ecker, Vienna University of Technology
{Felix.Breitenecker, Horst.Ecker}@tuwien.ac.at, aschift@osiris.tuwien.ac.at

The ARGESIM Comparison / Benckmark C13 ‘Crane and Embedded Control’ is based on modeling and controlling a crane crab and addresses digital elements together with implicit continuous model description. The comparison goes back to a formulation as test example for VHDL-AMS and has been defined first as ARGESIM Comparison C13 in SNE 35/36. This redefinition presents an improved observer based digital control with more stable behaviour. The revised tasks ask for comparison of uncontrolled nonlinear and linearised system behaviour, for modeling and simulation of digital control with sensor action, and for simulation of a diagnosis of sensor action.

Introduction

This benchmark originates from a publication of E. Moser and W. Nebel in the Proceedings of the conference DATE’99 [2]. The authors set up a benchmark mainly for testing the VHDL-AMS model description. Therefore, the benchmark comprises digital elements (digital controller, sensor action and diagnosis) as well as a continous model description.

The first definition as ARGESIM Comparison C13 *Crane and Embedded Control* extended this VHDL-AMS benchmark for simulators of any kind. Experiences with solutions sent in showed, that the design of the control is not really adequate, leading to misinterpretations and to a too narrow stability region. Consequently, for this revised definition, the design of the control has been improved significantly. Furthermore, the tasks to be performed with the modelled system and required control are formulated more precisely, so that solutions can be compared better.

1 Definition of Crane Dynamics

The crane consists of a horizontal track, a car moving along this track, and a load that is connected to the car via a cable of length r as shown in Figure 1.
The car is driven by the force f_c, which is exerted by a motor controlled by a digital controller. A disturbance is modelled as the disturbing force f_d, accelerating the load in horizontal direction.
Several sensors provide information about the current state of the system. The actuators for steering the crane are the motor and a brake.

In the following the nonlinear and linearized equations for the system are given. The linear model description originates from [1], where a detailed version can be found. In this comparison also the nonlinear model is to be investigated ([3]). The basic model parameters can be found in Table 1.

Linear Model / linearised model

$$\ddot{x}_c = \frac{f_c}{m_c} + g\frac{m_l}{m_c}\alpha - \frac{d_c}{m_c}\dot{x}_c$$

$$r\ddot{\alpha} = -g(1+\frac{m_l}{m_c})\alpha + (\frac{d_c}{m_c} - \frac{d_l}{m_l})\dot{x}_c -$$

$$- r\frac{d_l}{m_l}\dot{\alpha} - \frac{f_c}{m_c} + \frac{f_d}{m_l}, \quad x_l = x_c + r\alpha$$

Nonlinear model
It is to be noted that the nonlinear model is an implicit one, of type

$$\mathbf{M}(\mathbf{x})\,\ddot{\mathbf{x}} = \mathbf{g}\,(\mathbf{x}, \dot{\mathbf{x}})$$

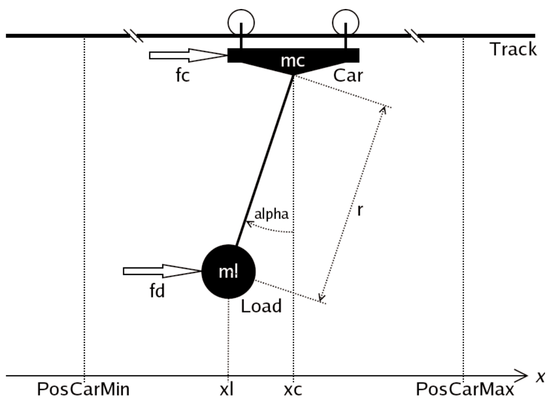

Figure 1: Schematic overview of crane model.

$$\ddot{x}_c = \left[m_c + m_l \sin^2(\alpha)\right] = -d_c \dot{x}_c + f_c + f_d \sin^2(\alpha) +$$
$$+ m_l \sin(\alpha)\left[r\dot{\alpha}^2 + g\cos(\alpha)\right] - d_l \dot{x}_c \sin^2(\alpha)$$

$$r^2\ddot{\alpha}\left[m_c + m_l \sin^2(\alpha)\right] = \left[f_d \frac{m_c}{m_l} - f_c + d_c \dot{x}_c\right] r\cos(\alpha) -$$
$$-\left[g(m_c + m_l) + m_l r\dot{\alpha}^2 \cos(\alpha)\right] \sin(\alpha) -$$
$$-d_l\left[\frac{m_c}{m_l}(\dot{x}_c r\cos(\alpha) + r^2\dot{\alpha}) + r^2\dot{\alpha}\sin^2(\alpha)\right]$$
$$x_l = x_c + r\sin(\alpha)$$

Depending on the simulation system used, these DAEs may be used directly, or they must be made explicitly by analytical or by numerical means.

Description	*Name*	*Value*
mass of car	m_c	10 kg
mass of load	m_l	100 kg
length of cable	r	5 m
gravity	g	9.81m/s^2
friction coefficient of car	d_c	0.5kg/s
friction coefficient of car with activated brake	d_c^{Brake}	100000 kg/s
friction coefficient of load	d_l	0.01kg/s
maximum position of car	*PosCarMax*	5m
minimum position of car	*PosCarMin*	-5m

Table 1: Basic model parameters.

2 Specification of the Control

The control includes the sensors, actuators, the digital controller and the diagnosis. The variable *PosDesired* is used as input to the controller and controls the position of the car (*PosCar*).

Actuators. The car is driven by a motor which exerts the force f_c on the car. As a model for the motor, including a specific controller for it, a first-order transfer function is used:

$$\dot{f}_c = -4\,(f_c - f_c^{Desired})$$

Activation of the brake is given by the following actions:

$$f_c^{Desired} := 0, \quad d_c := d_c^{Brake}$$

Sensors

Three sensors give information about the status of the system, one measuring position of the car and the other ones informing about reaching limits (Table 2).

Name	*Type*	*Description*
PosCar	Real	reports the position of the car (x_c)
SwPosCarMin	Boolean	true if $x_c < PosCarMin$, else false
SwPosCarMax	Boolean	true if $x_c > PosCarMax$, else false

Table 2: Sensor Variables / parameters.

Definition of the digital controller

The digital controller is implemented as a cycle based controller using a fixed cycle time of 10 ms. A discrete state space observer calculates the 'fictive' states **q** based only on the observation of *PosCar*:

$$\mathbf{q} = (\tilde{f}_c, \tilde{x}_c, \dot{\tilde{x}}_c, \alpha, \dot{\alpha})^T$$

The vector **q** is then fed into a state regulator. In the following the control algorithm is given, where n numbers the controlling cycles (a schematic overview of the controller is given in Figure 2).
The parameters for the controller are $V = 109.5$, *ForceMax* = 160, and *BrakeCondition* = 0.01, the vector and matrix parameters are given in the following:

$$\mathbf{q}_{n+1} := (\mathbf{M} - \mathbf{d}\,\mathbf{c}^T)\,\mathbf{q}_n + PosCar_n\,\mathbf{d} + f_c^{Desired}{}_n\,\mathbf{b}$$
$$u_{n+1} := V\ PosDesired - \mathbf{h}^T \mathbf{q}_{n+1}$$
$$f_c^{Desired}{}_{n+1} := \max\{\min\{u_{n+1}, ForceMax\}, -ForceMax\}$$

State matrix **A**, input vectors $\mathbf{b_1}$ and $\mathbf{b_2}$, and output vector **c** are given by the linear model.

$$\mathbf{M} = \begin{pmatrix} 0.96 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.01 & 0 & 0 \\ 0.001 & 0 & 0.9995 & 0.981 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -0.0002 & 0 & 0.0001 & -0.2158 & 1 \end{pmatrix}$$

$$\mathbf{c} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{d} = \begin{pmatrix} 34.5724 \\ 0.2395 \\ 2.0322 \\ 0.0164 \\ -0.1979 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0.04 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{h} = \begin{pmatrix} 2.9 \\ 109.5 \\ 286.0 \\ 1790.6 \\ 44.5 \end{pmatrix}$$

Diagnosis

The diagnosis runs concurrently to the digital controller. It is used to ensure the car stays within the given limits *PosCarMin* and *PosCarMax*. Therefore a boolean value *EmergencyMode* is introduced, which defaults to *false* and will not be reset once set to *true*.

In parallel, a condition for activating the brake while the car is standing still, is observed:

- if *PosCar* > *PosCarMax* then set *EmergenceMode* = *true*
- if *PosCar* < *PosCarMin* then set *EmergenceMode* = *true*
- if *EmergencyMode* or if for more than 3s $((\,|\,(f_c^{Desired})\,| < BrakeCondition$ then activate the brake

3 Tasks

First present the general approach, the implementation idea and the simulation system used. Especially, make clear how the implicit nonlinear model was handled, and how the digital controller was implemented. Furthermore it is of interest how the experiments were managed, especially in tasks b and c (features of the simulation environment).

The initial states for all of the following tasks should be zero:

$$x_c = 0, \dot{x}_c = 0, \alpha = 0, \dot{\alpha} = 0$$

A - Task: Nonlinear vs linear model. Implement the model (crane and motor) once using the linear equations for the crane dynamics and once using the nonlinear equations. Give details about the handling of the implicit nonlinear model (transformation to explicit model, or use of algorithms for implicit models indicating the nature of the algorithm).

Compare the linear and nonlinear models without controller and without brake, with following scenario:

- Initial state, $f_d = 0$
- At time $t = 0$: set $f_c^{Desired} = 160$ for 15s, then $f_c^{Desired} = 0$
- At time $t = 4$: set $f_d = Dest$ for 3s, then set $f_d = 0$

Print a table showing the steady-state difference (reached after about 2.000s) in the position of the load (x_l) for three values of *Dest* , *Dest* = -750,-800,-850.

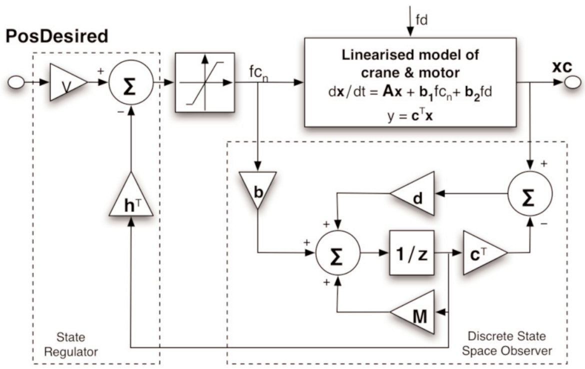

Figure 2: Schematic overview of controller.

B - Task: Controlled system. Implement the controller and brake and use the nonlinear equations for the crane dynamics. Describe how the continuous system and the discrete controller work together and how the brake is implemented.

Simulate the following scenario:

- Initial position, $f_d = 0$
- At time $t = 0$: *PosDesired* = 3
- At time $t = 16$: *PosDesired* = -0.5
- At time $t = 36$: *PosDesired* = 3.8
- At time $t = 42$: $f_d = -200$ for 1s, then $f_d = 0$
- At time $t = 60$: stop simulation

Results should be displayed as graph of position of car (x_c), position of load (x_l), angle α, and the state of the brake over time.

Task c - Controlled system with diagnosis

Add the Diagnosis to the controller. State how the *EmergencyStop* event is handled.

Simulate the following scenario:

- Initial position, $f_d = 0$
- At time $t = 0$: *PosDesired* = 3
- At time $t = 16$: *PosDesired* = -0.5
- At time $t = 36$: *PosDesired* = 3.8
- At time $t = 42$: $f_d = 200$ for 1s, then $f_d = 0$
- At time $t = 60$: stop simulation

Results should be displayed as graph of position of car (x_c), position of load (x_l), angle α, state of the brake and status of *EmergencyStop* over time.

For a solution, please follow the guidelines at the ARGESIM website WWW.ARGESIM.ORG/comparisons for and include your model source code files with the solution you send in.

References

[1] O. Foellinger: *Regelungstechnik*. Huethig, 5. Auflage, 1985

[2] E. Moser, W. Nebel. *Case Study: System Model of Crane and Embedded Controller*. Proc. DATE'99, pages 721-724, 1999.

[3] J. Scheikl, F. Breitenecker, I. Bausch-Gall: Comparison C13 *Crane and Embedded Control – Definition*. SNE 35/36, Dec. 2002; 69 – 71.

Corresponding author: Felix Breitenecker
F. Breitenecker, A. Schiftner, Inst. f. Analysis & Scientific Computing, *{fbreiten,aschift}@osiris.tuwien.ac.at*
H. Ecker, Inst. Machine Dynamics, *Horst.Ecker@tuwien.ac.at*
Vienna University of Technology,
Wiedner Hauptstrasse 8 -10, 1040 Vienna, Austria

Received: May 5, 2006
Revised: July 15, 2006
Accepted: July 20,2006