Crane with Complex Embedded Control – ARGESIM Benchmark C13R with Implicit Modelling, Digital Control and Sensor Action

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

The ARGESIM Bencmark C13R 'Crane with Complex Embedded Control' is based on modelling and digital control of a crane crab. The discrete control is designed by means of a state space observer, and by state space control. On modelling level, setup and handling of implicit nonlinear model descriptions are investigated, and nonlinear model and linear model (linearised model) are compared. Thus, the benchmark addresses as well graphical model description of standard input/output type and of a-causal element type, as well as textual descriptions given by DAEs or implicit laws. Furthermore, graphical or textual modelling features for discrete control are investigated, together with additional sensor action modelling. On simulation level, simulation results for nonlinear and linear dynamics without feedback control are to be compared, and two time-domain scenarios with changing set points and with disturbances are investigated of the fully controlled nonlinear system are to be investigated. Thereby, the synchronisation of digital control, DAE or ODE solvers and isolated disturbance events is of importance.

SNE 17/1, April 2007

Introduction

This benchmark originates from a publication of E. Moser and W. Nebel ([2]), where the authors set up a first benchmark mainly for testing the VHDL-AMS model description in a case study for digitally controlled crane crab. This first benchmark addressed mainly modelling and simulation of discrete control with complex sensor actions (a standard task for VHDL) in combination with modelling and simulation of the continuous mechanical dynamics of the crane crab (a task, which VHDL-*AMS*, the extension of VHDL for *A*nalog and *M*ixed *S*ignals, is designed for).

In 2003, the definition of the ARGESIM Comparison C13 *Crane and Embedded Control* extended the previous VHDL-AMS benchmark for simulators of any kind, keeping the control structure and sensor structure given, and adding nonlinear dynamics, DAE modelling, and complex scenarios to be simulated. Experiences with the few solutions sent in showed, that the design of the control was not really adequate, was leading to mis-interpretations, as well as the comparison of linear and nonlinear dynamics was not representative.

Consequently, this new revised definition as ARGESIM Benchmark C13R *Crane with Complex Embedded Control* presents a more precise nonlinear and linear model for the crane dynamics, a new designed controller with much more stable behaviour, well-defined rules for sensor actions, and for simulation purposes, a representative scenario for comparing linear and nonlinear model, and two scenarios for investigating the time domain behaviour of the fully controlled nonlinear model with sensor actions.

1 Model for Crane Dynamics

The crane crab 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. Three sensors provide information about the current state of the system: current position, left stop signal and right stop signal. The actuators for steering the crane are the motor for the car, and a brake acting in case the car has reached a steady state at set point, or in case of emergency.

The nonlinear model can be derived by Lagrange approach, resulting in two nonlinear coupled 2nd order ODEs for car position x_c and for angle α, respectively; the position of the load x_l follows from x_c and α. It is to be noted, that the nonlinear model is of implicit type, and can be expressed with mass matrix $\mathbf{M}_{nl}$, general coordinates $\mathbf{p}$, and generalised forces $\mathbf{g}$:

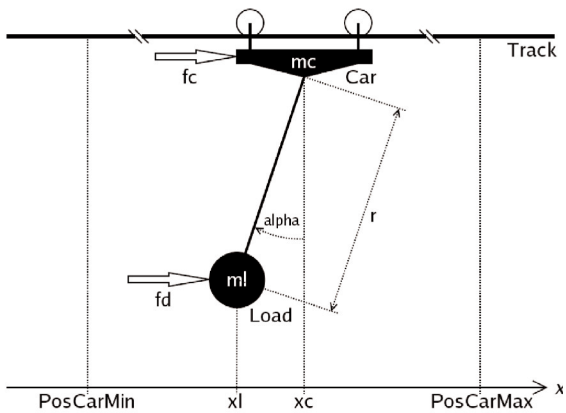

Figure 1: Schematic overview of crane model.

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

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

$$x_l = x_c + r \cdot \sin(\alpha); \mathbf{M}_{nl}(\mathbf{p}) \cdot \ddot{\mathbf{p}} = \mathbf{g}(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{g}), \mathbf{p} = (x_c, \alpha)^{\mathrm{T}}$$

The basic model parameters can be found in Table 1. The linear model (details in [2]; identical to model above linearised at zero states) is given by two coupled explicit 2nd order linear ODEs for car position x_c and for angle α, resp. (linear state space of 4th order):

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

$$r\ddot{\alpha} = -g(1 + \frac{m_l}{m_c})\alpha + (\frac{d_c}{m_c} - \frac{d_l}{m_l})\dot{x}_c - \\ - r\frac{d_l}{m_l}\dot{\alpha} - \frac{f_c}{m_c} + \frac{f_d}{m_l}, \quad x_l = x_c + r\alpha$$

$$\dot{\mathbf{x}}_L = \mathbf{A}_L \cdot \mathbf{x}_L + \mathbf{b}_c f_c + \mathbf{b}_d f_d, \quad \mathbf{x}_L = (x_c, \dot{x}_c, \alpha, \dot{\alpha})^{\mathrm{T}}$$

2 Specification of the Control

The control includes the sensors, the actuators, the digital controller and the diagnosis. The variable *PosDesired* is used as input to the controller (set point) and controls the position of the car *PosCar*, being x_c, the only measured state of the crane.

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 (the digital control variable computed in the digital controller is $f_c^{Desired}$):

Description	**Name**	**Value**
mass of car	m_c	10 kg
mass of load	m_c	100kg
length of cable	r	5 m
gravity	g	9.81 m/s²
friction coefficient of car	d_c	0.5 kg/s
friction coefficient of car with activated brake	d_c^{Brake}	10^5 kg/s
friction coefficient of car	d_l	0.001 kg/s
maximum position of car	*PosCarMax*	5 m
minimum position of car	*PosCarMin*	-5 m

Table 1: Basic model parameters.

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

As second actuator, a brake becomes active either in case of emergence, or in case the car has reached a steady state or a set point resp.

This second case is given by a control variable $f_c^{Desired}$ being for more than *TimeSteady* seconds less than a minimal control force *BrakeCondition*:

if $|\,(f_c^{Desired})\,| < $ *BrakeCondition*
for more then *TimeSteady* seconds
then activate the brake

Activation of the brake is given by the following actions (setting control variable to zero, and setting the friction coefficient of the car to a maximal value):

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

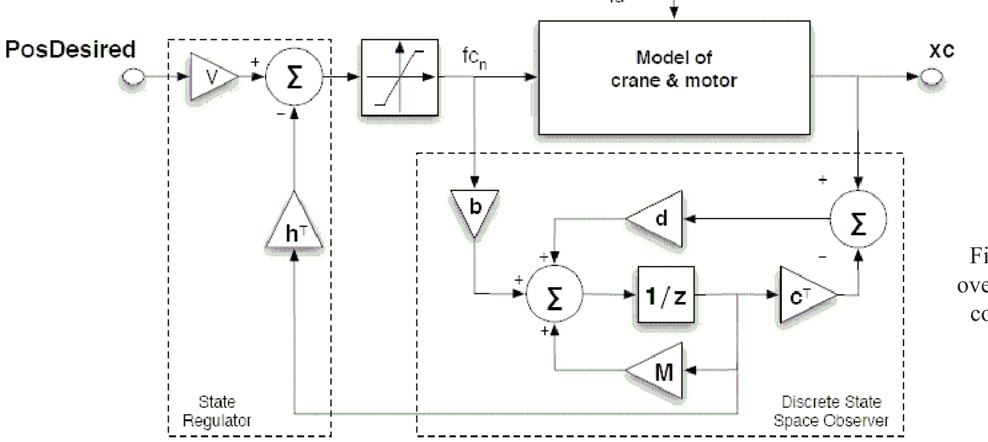

Figure 2: Schematic overview of the digital controller with state space observer.

Sensors. Three sensors give information about the status of the system, one measuring the position of the car (*PosCar*) and the other ones (*SwPosCarMax*, *SwPosCarMin*) signaling that the car has reached the outmost left or outmost right position (Table 2).

Definition of the digital controller

The digital controller (schematic overview given in Figure 2) is implemented as a cycle based controller using a fixed cycle time of *SampleTime* ms. A discrete state space observer calculates the ‘fictive’ states **q**, $\mathbf{q} = (\tilde{f}_c, \tilde{x}_c, \dot{\tilde{x}}_c, \tilde{\alpha}, \dot{\tilde{\alpha}})^T$ based only on observation of *PosCar*.

The vector **q** is then fed into a state regulator. The discrete control algorithm calculates the new control variable $f_c^{Desired}{}_{n+1}$ based on the system output value $PosCar_n$ and control variable $f_c^{Desired}{}_n$ from the previous control cycle (n numbers the controlling cycles; graphical presentation given in Figure 2):

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

The discrete state space observer was derived from the linear model equations for crane and motor, combining the linear state space $\mathbf{x}_L$ for the crane with the first order transfer function for the control force f_c without disturbance input f_d, giving a linear system of 5th order:

$$\dot{\mathbf{x}} = \mathbf{A}\cdot\mathbf{x} + \mathbf{b}_m\,f_c^{Desired}, \quad x_c = \mathbf{c}^{\mathrm{T}}\mathbf{x}$$
$$\mathbf{x} = (f_c, x_c, \dot{x}_c, \alpha, \dot{\alpha})^{\mathrm{T}}, \mathbf{b}_m = (4,0,0,0,0)^{\mathrm{T}}$$

As the above linear system is observable and controllable, the state regulator and the state space observer could be treated independently. For a detailed description of how to choose **d** refer to [3].

The resulting continuous state space observer was discretised using an explicit Euler integration scheme using the fixed cycle time *SampleTime*, which results in the following parameter vectors:

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.

Description	Name	Value
sample time of discrete controller	*SampleTime*	0.1 s
gain	*V*	109.5 N/m
minimal control force	*BrakeCondition*	0.01 N
recognition interval for *BrakeCondition*	*TimeSteady*	3.0 s
maximal control force	*ForceMax*	160 N

Table 3: Controller parameters.

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

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

There, the state regulator, represented by **h**, was chosen as a Ricatti-type regulator using an error metric emphasising the state variables in favour of the controlling variable (given the linear system as before, MATLAB’s function `lqr` was used to calculate the control vector **h)**. Further parameters for the control are given in Table 3.

Diagnosis. The diagnosis runs concurrently to the digital controller. It is used to ensure the car stays within the given soft 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, the condition for activating the brake while the car is standing still, can be observed.

In each cycle of the controller, the following conditions have to be checked:

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

3 Tasks - Experiments

The classical ARGESIM Comparisons required three tasks to be performed with the defined dynamic system, mostly addressing investigations and analysis in the time domain; furthermore information on the simulator used and a short description of the model implementation should be given - both to be presented within one page SNE. The new or revised ARGESIM Benchmarks extend the three tasks - Task **A**, Task **B**, Task **C** - and the simulator description - Task **S**imulator Description - by requesting a detailed description of the model implementation, whereby also different modelling approaches may be presented - Task **M**odelling, and by a short summary for the benchmark solution - Task **R**ésumé, trying also a classification of the approach. For documentation of all tasks two pages SNE may be used, (Task **M**odelling more than ½ page, and tasks **A**, **B**, and **C** about 1 page). Furthermore, model source files should be sent in. More details at ARGESIM website WWW.ARGESIM.ORG, menu *SNE*.

Modelling. First present the general approach, the implementation idea in the simulation system used. Especially, make clear how the implicit nonlinear model was handled, and how the digital controller was implemented. Was the DAE model used directly, or must it be made explicitly by analytical or by numerical means ? Furthermore it is of interest, how the synchronisation of digital control, DAE or ODE solvers and isolated disturbance events are performed, and how the scenarios in the tasks were managed.

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 implementation of both models in parallel. Compare linear and nonlinear model without controller and without brake, with following scenario:

- Initial state: all states zero, $f_d = 0$
- 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 2.000 s) in the position of the load (x_l) for three values of *Dest*, *Dest* = -750, -800, -850.

B - Task: Controlled System. Show implementation of controller with brake and of brake action in the nonlinear equations for the crane dynamics.

Give comments on work of discrete model parts and simulate the following scenario:

- Initial position: zero states, $f_d = 0$
- 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 position x_c, load x_l, angle α, and of the status of the brake over time.

C - Task: Controlled System with Diagnosis. Add the diagnosis to the controller. State how the *EmergencyStop* event is handled within the controller, and simulate the following scenario:

- Initial position: zero states, , $f_d = 0$
- 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 $t = 46$: $f_d = 200$ for 1s, then $f_d = 0$
- At time $t = 60$: stop simulation

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

For a solution, please follow the guidelines at the ARGESIM website WWW.ARGESIM.ORG and include your model source code files with the solution sent 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
Felix Breitenecker, Alexander Schiftner,
Institute for Analysis & Scientific Computing
Felix.Breitenecker@osiris.tuwien.ac.at
Horst Ecker, Institute for 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; December 9, 2006
Accepted: January 22, 2007