ARGESIM Benchmark C13 'Crane and Embedded Control' with SIMULINK –modelled Dynamics and MATLAB-programmed Control

Georg Janisch, Miriam Leopoldseder*, Andreas Körner

Institute for Analysis and Scientific Computing, Vienna Univ. of Technology Wiedner Hauptstraße 8
1040 Vienna, Austria; *miriam.leopoldseder@tuwien.ac.at*

Simulation Notes Europe SNE 25(3-4), 207 - 209
DOI: 10.11128/sne.25.bn13.10318
Received: February 20, 2015; Revised September 20, 2015;
Accepted: October 10, 2015;

Abstract. ARGESIM Benchmark C13 'Crane and Embedded Control' is based on nonlinear and a linear continuous dynamics for a crane crab and on discrete control by a linear observer model. This solution makes use of the MATLAB/Simulink system in a 'mixed' manner. The continuous dynamics are modelled graphically in the Simulink environment – using partly connected blocks and partly MATLAB functions. The discrete control with observer model and the diagnosis is programmed in MATLAB, framed by a sampled date discrete submodel in Simulink connected to the continuous submodels for the dynamics. This approach allows fast modelling, but deeper insight into the simulation environment is necessary.

Introduction

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming [1]. It is based on numerical vector and matrix manipulation. SIMULINK is a graphical block diagram environment for multidomain simulation and model-based design [2]. Within Simulink, MATLAB functions can be used, with high complexity. This solution makes use of the MATLAB/Simulink system in a 'mixed' manner.

The continuous dynamics are modelled graphically in the Simulink environment – using partly connected blocks and partly MATLAB functions. The discrete control with observer model and the diagnosis is programmed in MATLAB, framed by a sampled date discrete submodel in Simulink connected to the continuous submodels for the dynamics.

1 Modelling

Basis for the dynamic model are the linear and nonlinear mechanical equations for the crane crab, and the 'electrical equation for the DC motor

Linear model for crane crab:

$$\ddot{x}_c = \frac{f_c}{m_c} + g\frac{m_l}{m_c}\alpha - \frac{d_c}{m_c}\dot{x}_c \tag{1}$$

$$r\ddot{\alpha} = -g\left(1+\frac{m_l}{m_c}\right)\alpha + \left(\frac{d_c}{m_c} - \frac{d_l}{m_l}\right)\dot{x}_c - r\frac{d_l}{m_l}\dot{\alpha} - \frac{f_c}{m_c} + \frac{f_d}{m_l} \tag{2}$$

$$x_l = x_c + r\alpha$$

Nonlinear model for crane crab:

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

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

$$x_l = x_c + r\sin\alpha$$

Linear model for DC motor:

$$t_m\dot{f}_c + f_c = k_m v$$

In general, for dynamic model parts Simulink submodels with continuous blocks were used. For discrete model parts – i.e. the controller and the diagnosis – Simulink submodel(s) with MATLAB functions inside were used (Figure 1 – modelling overview)

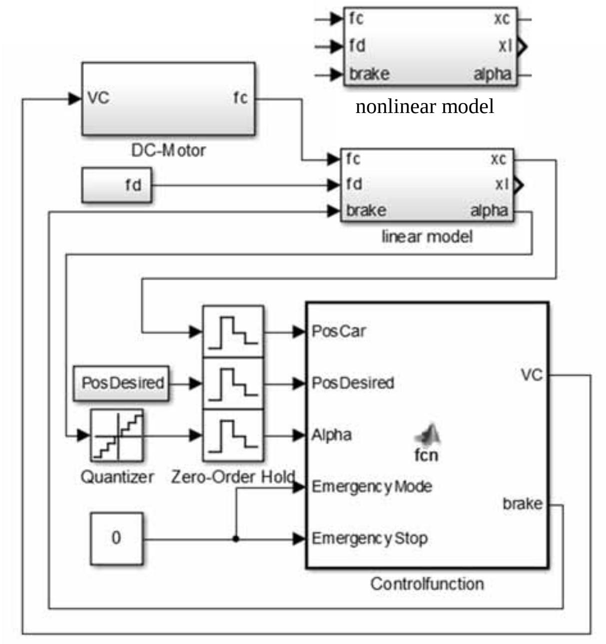

Figure 1. Overall model with continuous dynamics (linear / nonlinear) and discrete control.

The crane dynamics and the DC-motor were built in SIMULINK using continuous blocks with differnet levels of granularity.

The control is based on a linear observer which operates discrete. With observer matrices A, B and K and observer state vector $\vec{q}$ control v is computed in the main by

$$\vec{q}_{n+1} = A\vec{q}_n + B\left[v_n \;\; x_{c,n}\right]^T, \qquad y_n = K\vec{q}_n$$
$$v_{n+1} = k_p(pos_{desired-}(x_{c,n} + r\alpha)) - y_{n+1}$$

The controller algorithm and the diagnose functions were programmed in MATLAB and implemented in SIMULINK as a MATLAB function block. These blocks are supposed to work discrete. To assure that Zero-Order-Hold blocks are used. The angle sensor is simulated by a quantizer block and by comparing the current value to maximum and minimum defined in the control function (Figure 1).

The diagnosis is also implemented in form of a MATLAB-function. Fortunately Simulink allows different sample times for discrete blocks, so for the controller function a sample time of 10 ms and for the diagnosis functions a sample time of 1 ms is set.

2 Task a: Comparison of Nonlinear and Linear model without Control

The linear and nonlinear models were compared without brake and controller, so that from the model in Figure 1 only the nonlinear submodel and the linear submodel for the crane crab were used, without feedback control, only with given input f_c and disturbances *fd*.

The nonlinear model is an implicit. Before implementation, the model was transformed symbolically into an explicit state space model (with four states for the crane crab, and one state for the motor). As the nonlinearities are relatively comprehensive (after making the model explicit), no basic Simulink blocks were used: the nonlinearities were calculated in a MATLAB function, and fed into a Simulink integrator block.

The dynamic calculations are done by an ODE45-solver which is the standard MATLAB solver for non-stiff ODEs. Below the steady-state difference of x_l and f_d disturbance is shown.

f_d disturbance	-750	-800	-850
x_l difference	-5.3086	-6.8225	-7.8131

Table 1. A-Task: steady state difference

3 Task b: Simulation of the Linear Fully Controlled System

The task was to simulate the fully controlled system in a given scenario. The car starts out at position zero with no external force affecting it and with the desired position +3. After 16 seconds the target position is changed to -0.5 and then after 36 seconds changed again to 3.8. At time t = 42 f_d is set to -200 for 1 second and then to zero. The simulation stops at 60 seconds.

The controller and the brake are implemented via the DC-motor interacting with the linear model (Figure 1) following a linear discrete observer control.

The signal d_c of the model stands for the friction coefficient of the car. To implement the brake, the value of d_c is increased to simulate a mechanical brake. Again, the brake mechanism is described in the MATLAB controller function. When the break is applied the internal control parameters are held constant and the VC output is set to zero. In the linear model the instant stop of the car is achieved by increasing the friction coefficient to 10 000 s^{-1}.

The dynamic calculations are done by an ODE45-solver which is the standard MATLAB solver for non-stiff ODEs, and which also works well with discrete model parts. The simulation results are shown in Figure 2 and Figure 3.

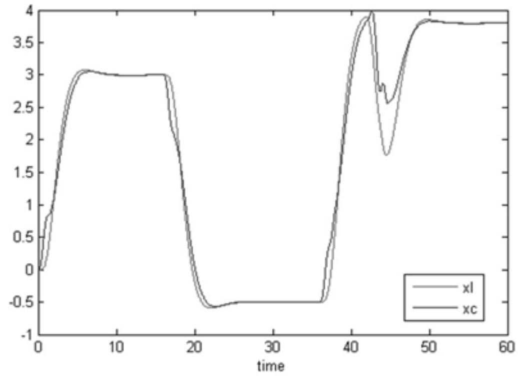

Figure *2.* B-Task: Output parameters xl and xc over time, reacting to changing input signal PosDesired.

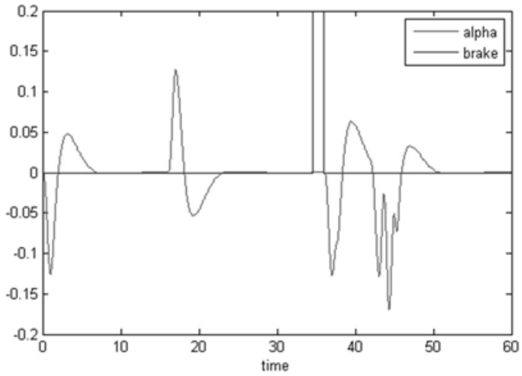

Figure 3. B-Task: Brake is activated at time = 34.59 s and resetted at time =36 s.

4 Task c: Simulation of Controlled System with Sensor Diagnosis

In this task the sensor diagnosis was added to the system. During the 60 second simulation time the desired position is shifted from +3 to -0.5. After 18 seconds the angle sensor is forced to trigger the Emergencymode. Then at t = 36 the desired position is changed again to 3.8 after which f_d is set to 200 for 1 s, before resetting it to zero.

The diagnosis is described in a MATLAB block with a sample time of 1 ms. The range sensors are programmed in the diagnosis MATLAB function.

If the range sensors for the maximum or minimum position are activated for more than 20 ms during a 100 ms time period the Emergencystop is initiated. To assure that, every millisecond the states of the range sensors are stored in an array. It has exactly 100 slots and if the sum exceeds 20 the Emergencystop is set.

The Emergencymode is handled the same way, except that it is activated when the position limits are exceeded for more than 50 ms.

The break is applied when the car reaches its maximum position and the Emergencystop is activated. The load oscillates after the stop (results see Figure 4 and Figure 5.

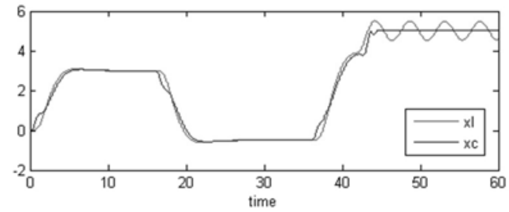

Figure 4. C-Task: Output parameters xl and xc over time, reacting to changing input signal PosDesired.

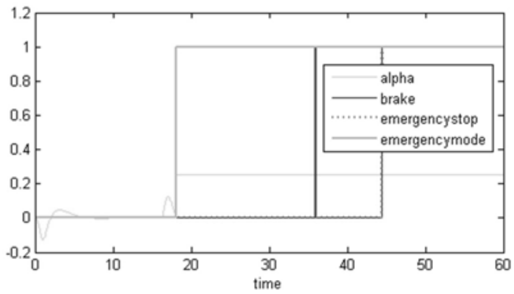

Figure 5. C-Task: Reaction of the model to broken angle sensor.

The interaction between the controller and the model is extremely sensible. Minor changes of the circle time have a great impact. They result in a decreased performance of the control.

References

[1] http://www.mathworks.de/de/help/matlab/learn matlab/ product-description.html

[2] http://www.mathworks.de/products/simulink/

[3] Scheikl J, Breitenecker F, Bausch-Gall I. C13 Crane and Embedded Control Revised Defintion. SNE 35/36, p 69-71, December 2002

ASIM ASIM ASIM

ASIM - Buchreihen / ASIM Book Series

Proceedings

Simulation in Production und Logistics 2015 - 16. ASIM-Fachtagung Simulation in Produktion und Logistik
M. Raabe, U. Clausen (Hrsg.); ISBN 978-3-8396-0936-1, Stuttgart: Fraunhofer Verlag, 2015. (O. P.)

Simulation in Produktion und Logistik 2013: Entscheidungsunterstützung von der Planung bis zur Steuerung
W. Dangelmaier, C. Laroque, A. Klaas (Hrsg.); ISBN 978-3-942647-35-9, HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut, Paderborn, 2013 (O. P.)

Modellierung, Regelung und Simulation in Automotive und Prozessautomation – Proc. 5. ASIM-Workshop Wismar 2011. C. Deatcu, P. Dünow, T. Pawletta, S. Pawletta (eds.), ISBN 978-3-901608-36-0, ASIM/ARGESIM, Wien, 2011 (O. A.)

Simulation in Produktion und Logistik 2010: Integrationsaspekte der Simulation - Technik, Organisation und Personal. G. Zülch, P. Stock, (Hrsg.), ISBN 978-3-86644-558-1, KIT Scientific Publ. Karlsruhe, 2010 (O.P.)

Monographs

Simulation und Optimierung in Produktion und Logistik – Praxisorientierter Leitfaden mit Fallbeispielen.
L. März, W. Krug, O. Rose, G. Weigert , G. (Hrsg.); ISBN 978-3-642-14535-3, Springer, 2011 (O.P.)

Verifikation und Validierung für die Simulation in Produktion und Logistik - Vorgehensmodelle und Techniken.
M. Rabe, S. Spieckermann, S. Wenzel (eds.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008 (O. P.)

Qualitätskriterien für die Simulation in Produktion und Logistik – Planung und Durchführung von Simulationsstudien. S. Wenzel, M. Weiß, S. Collisi – Böhmer, H. Pitsch, O. Rose (Hrsg.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008

Series Fortschrittsberichte Simulation

P. Einzinger: **A Comparative Analysis of System Dynamics and Agent-Based Modelling for Health Care Reimbursement Systems.**
FBS 25, ASIM/ARGESIM Vienna, 2014; ISBN 978-3-901608-75-9, ARGESIM Report 75 (O. A.)

M. Bruckner: **Agentenbasierte Simulation von Personenströmen mit unterschiedlichen Charakteristiken**
FBS 24, ASIM/ARGESIM Vienna, 2014; ISBN 978-3-901608-74-2, ARGESIM Report 74 (O. A.)

S. Emrich: **Deployment of Mathematical Simulation Models for Space Management**
FBS 23, ASIM/ARGESIM Vienna, 2013; ISBN 978-3-901608-73-5, ARGESIM Report 73 (O. A.)

G. Maletzki: **Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBC-Ansatzes.** FBS 22, ASIM/ARGESIM Vienna, 2014; ISBN 978-3-901608-72-8, ARGESIM Report 72 (O. A.)

X. Descovich: **Lattice Boltzmann Modeling and Simulation of Incompressible Flows in Distensible Tubes for Applications in Hemodynamics**
FBS 21, ASIM/ARGESIM Vienna, 2012; ISBN 978-3-901608-71-1, ARGESIM Report 71 (O. A.)

F. Miksch: **Mathematical Modeling for New Insights into Epidemics by Herd Immunity and Serotype Shift**
FBS 20, ASIM/ARGESIM Vienna, 2012; ISBN 978-3-901608-70-4, ARGESIM Report 70 (O. A.)

S. Tauböck: **Integration of Agent Based Modelling in DEVS for Utilisation Analysis: The MoreSpace Project at TU Vienna;** FBS 19, ASIM/ARGESIM Vienna, 2012; ISBN 978-3-901608-69-8, ARGESIM Report 69 (O. A.)

Ch. Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse**
FBS 18, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-68-1, ARGESIM Report 68 (O. A.)

O. Hagendorf: **Simulation-based Parameter and Structure Optimisation of Discrete Event Systems**
FBS 17, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-67-4, ARGESIM Report 67 (O. A.)

Orders via ASIM (O. A.) **or via Publisher** (O. P.)**:**
ASIM/ARGESIM Office Germany, Hochschule Wismar, PF 1210, 23952 Wismar, Germany
ASIM/ARGESIM Geschäftsstelle Österreich, c/o DWH, Neustiftgasse 57, 1040 Vienna, Austria
Download (some books) via ASIM webpage in preparation
Info: www.asim-gi.org, info@asim-gi.org