

Case-study: Simulator-based control system tuning and testing using an estimated process model

Finn Haugen
Telemark University College

Abstract

A mathematical model of a process to be controlled can be used for off-line tuning and testing of control systems for the process. The tuning and testing can be performed using analytical methods or using simulations. The benefit of simulations in this respect is that it is straight-forward to include non-linearities in the model. This presentation *demonstrates* the application of commercially available functions for system identification and simulation, and control design. Focus is on functions available in National Instrument's LabVIEW, but similar functions are available in other tools, as MATLAB. The presentation shows a practical application, namely a speed servo for a DC motor.

1 Introduction

1.1 Outline of the presentation

This presentation will demonstrate how to use LabVIEW toolkits for analysis, design, and simulation of a speed control system for a DC motor. The control system is finally implemented on a National Instruments Compact FieldPoint system for I/O and embedded control. The motor and the Compact FieldPoint system are described briefly below.

The following LabVIEW tools are used:

- **System Identification Toolkit** [5]
- **Simulation Module** [2]
- **Control Design Toolkit** [3]

An HTML version of this document, including linked files, is available at <http://techteach.no>.

1.2 The physical process - a DC motor

In this presentation a DC motor is modelled, simulated and controlled:

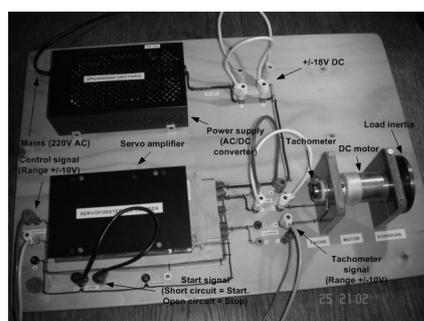


Figure 1: DC motor

The motor is produced by Faulhaber. The control signal is in the range of $\pm 10V$, and the tachometer voltage is in the range of approximately $\pm 10V$. A load inertia has been added to the motor. The time constant of the motor including load and tachometer is approximately 0.3s. A load torque can be applied to the motor by simply braking the motor (the load) by hand.

1.3 The Compact FieldPoint I/O and embedded control system

The Compact Fieldpoint system used in this application consists of the following modules:

- **cFP-2000** controller module running LabVIEW RT 7.1
- **cFP-AIO-610** analog I/O module

Figure 2 shows the Compact FieldPoint System, and Figure 3 shows how the FieldPoint system appears in the Measurement and Automation Explorer (MAX) utility where I/O channels are defined and tested before being used in a LabVIEW program.

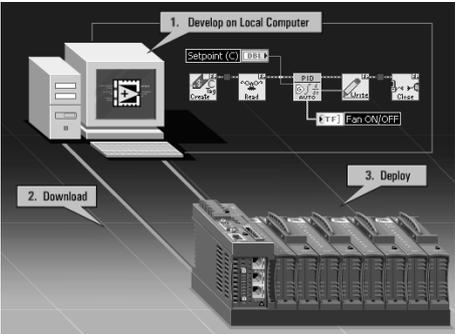


Figure 2: Compact FieldPoint

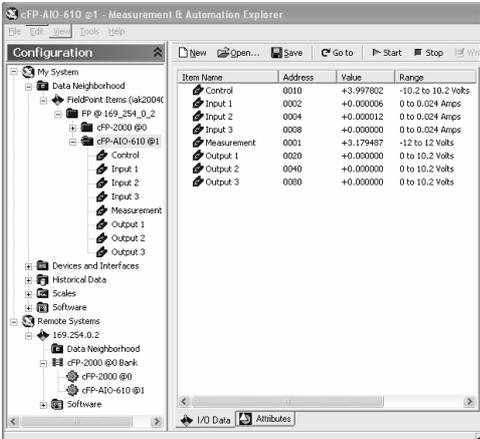


Figure 3: Measurement and Automation Explorer (MAX) showing the Compact FieldPoint system

1.4 A procedure for analysis, design and implementation of a control system

Figure 4 shows a procedure for analysis, design and implementation of a control system.

A procedure for analysis, design and implementation of a control system

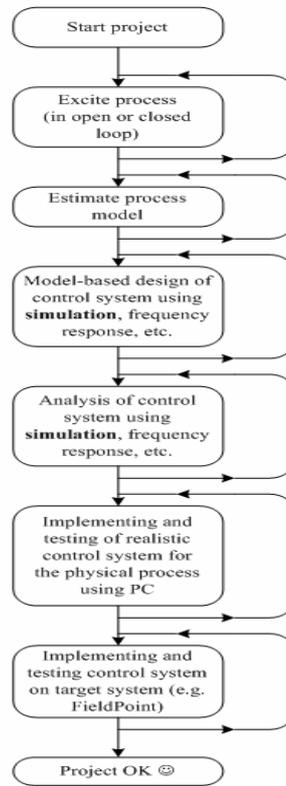


Figure 4: Procedure for analysis, design and implementation of a control system

In the present application separate files implements the steps shown above. The logged data are stored in a spreadsheet file. The estimated model is also stored in a file.

The figure below shows a block diagram of the process with controller and estimator, and blocks containing functions for analysis and design of the control system.

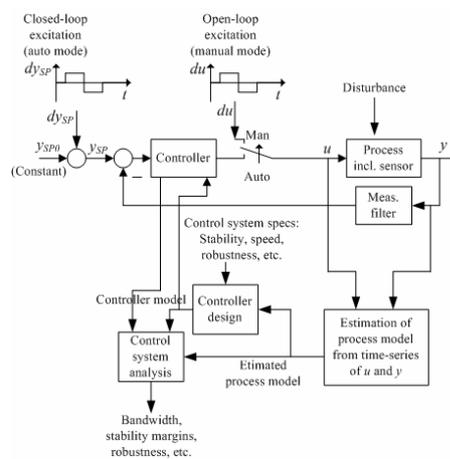


Figure 5: Block diagram of the process with controller and estimator, and blocks containing functions for analysis and design of the control system.

2 Black-box modelling of the process

2.1 Introduction

When a black-box model of a process is to be developed the process must be excited by a sufficiently "rich" signal (we can of course not expect to derive a dynamics model from constant signals). The excitation can be made in two ways, refer to the figure shown above):

- **Closed loop excitation.** This means that the process is under feedback control. The excitation is via the setpoint.
- **Open loop excitation.** This means that the process is not under feedback control. The excitation is via the control signal.

In this application *open loop excitation* is used.

2.2 Exciting the process and measuring the response

excite_and_logg.vi shown below saves time t , input signal u and measured response y on the spreadsheet file logfile1. t , u , and y are columns in this file. The sampling time is $h = 0.02s$ which is used both for analog output (control signal) and analog input (measurement signal) throughout this application.

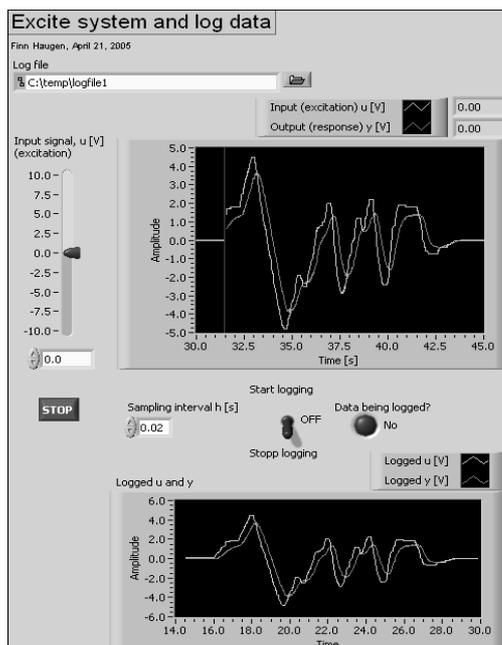


Figure 6: Front panel of excite_and_logg.vi

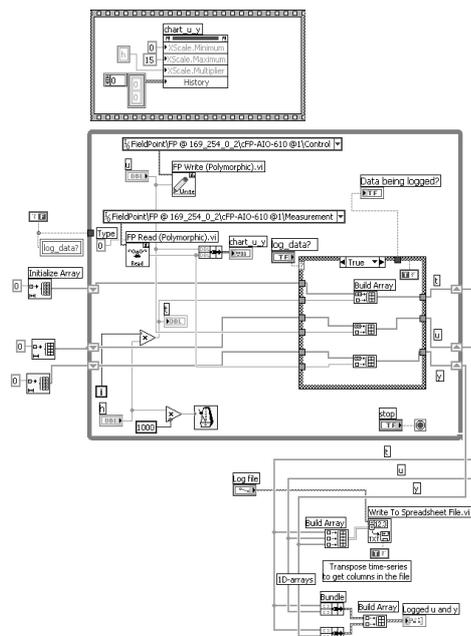


Figure 7: Block diagram of excite_and_logg.vi

Here is the log file from one specific experiment: logfile1. The columns in the file are t , u , y .

2.3 Estimating a mathematical model from logged data

A mathematical model of the motor in the form of a *discrete-time transfer function*, $H_p(z)$ is estimated. This transfer function is actually calculated from an estimated discrete-time state space model,

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\y(k) &= Cx(k) + Du(k)\end{aligned}$$

using the **Convert to Transfer Function Model of CDT** (Control Design Toolkit) function of the Control Design Toolkit.

The state-space model is estimated using the **SI Estimate State Space Model** function. This function implements a Subspace-method which is an efficient and generally applicable method for estimating black-box (canonical) state-space models. The function is based on the Subspace-method reported in [4]. The **SI Estimate State Space Model** function is on the Parametric Modeling palette of the System Identification function palette, see the figure below.

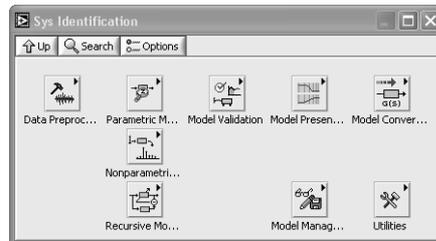


Figure 8: Functions palette of System Identification Toolkit

The program `system_ident.vi` shown below estimates a discrete-time transfer function model compatible with the Control Design Toolkit, and saves the model on a file (for later use). Important features of the program are as follows:

- The **SI Estimate Orders of System Model** function indicates the best (optimal) order based on a calculation of the number of significant singular values of the system matrix of the underlying state-space model used by the subspace estimation method. The user can still freely select the order of the estimated order.
- The user can choose to apply linear detrending of the input and output data before the data is used for system identification. The detrending is made by the **SI Remove Trend** function.
- The quality of the estimated model is indicated by plotting together the measured system output and the simulated model output which is generated using the **SI Simulate Model** function.

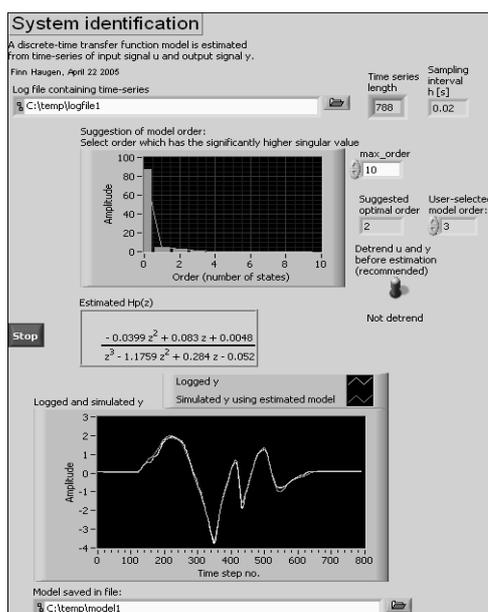


Figure 9: Front panel of `system_ident.vi`

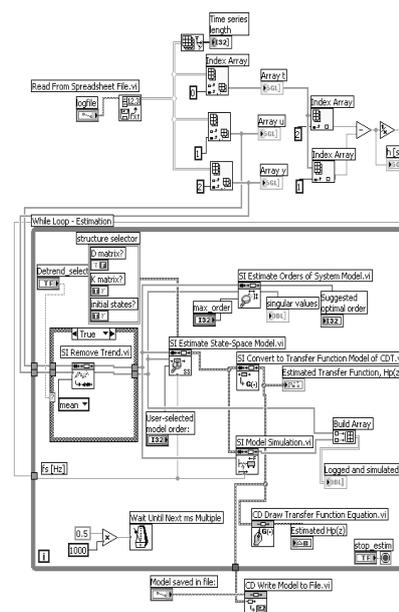


Figure 10: Block diagram of `system_ident.vi`

2.4 Is the model accurate? Comparing process simulation and measurements

In compare_process_simulation_and_measurements.vi shown below the estimated model is further evaluated. The user excites both the simulated process and the real process with an arbitrary input signal (adjusted by the user), and the simulated response and the real measurement are compared. The simulation is based on the LabVIEW Simulation Module [2]. The functions palette of the Simulation Module is shown below.

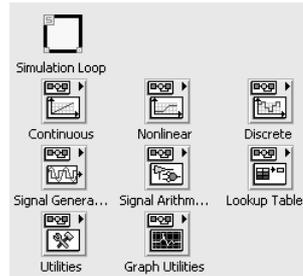


Figure 11: The functions palette of the Simulation Module

Since in general the linear model is valid *only around an operating point*, the model is excited by the deviation, du , from the operating point value, u_0 , of the input signal, and the simulated response is thus the deviation, dy , from the operating point value, y_0 , of the output signal.

In other words:

- The input signal to the simulated process is du , while the total input signal to the physical process is

$$u = u_0 + du$$

where u_0 is fixed and du is arbitrary, but small in amplitude.

- The response from the simulated process is dy_{sim} , while the response from the real process is

$$y_{meas} = y_0 + dy_{meas}$$

giving

$$dy_{meas} = y_{meas} - y_0$$

The model is accurate if the difference between dy_{meas} and dy_{sim} is small!

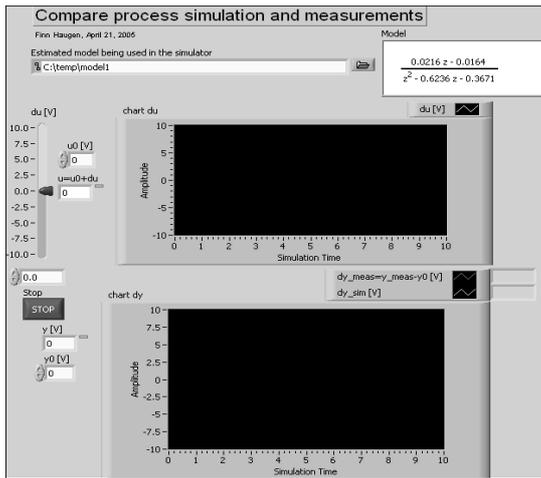


Figure 12: Front panel of compare process simulation and measurements.vi

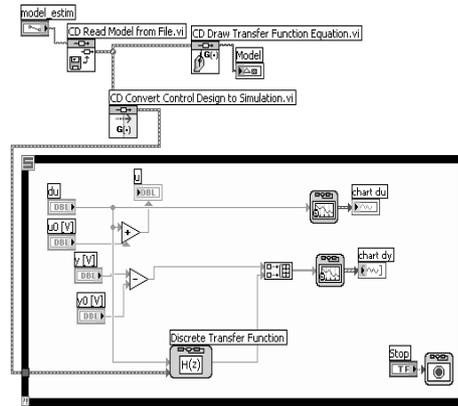


Figure 13: Block diagram of compare process simulation and measurements.vi

3 Control system analysis and design

3.1 Analysis of control system using Control Design Toolkit

The function palette of the Control Design Toolkit [3] is shown below.

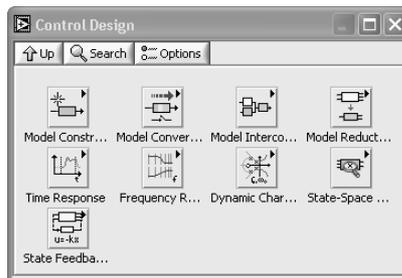


Figure 14: Function palette of the Control Design Toolkit

The figure below shows a block diagram of the control system, including a measurement low pass filter (which is a second order Butterworth lowpass filter). The blocks contains discrete-time transfer functions (i.e. z-transfer functions).

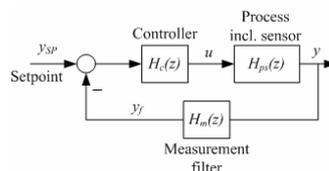


Figure 15: Block diagram of the control system

control_analysis_design.vi shown below analyses the control system with respect to the following [1]:

- Bandwidth (loop transfer function crossover frequency)
- Gain Margin GM and Phase Margin PM

- Simulated step response in the process output variable y (due to a unit step in the setpoint)

In addition, PID settings are calculated from the Ziegler-Nichols' closed-loop method interpreted in the frequency domain.

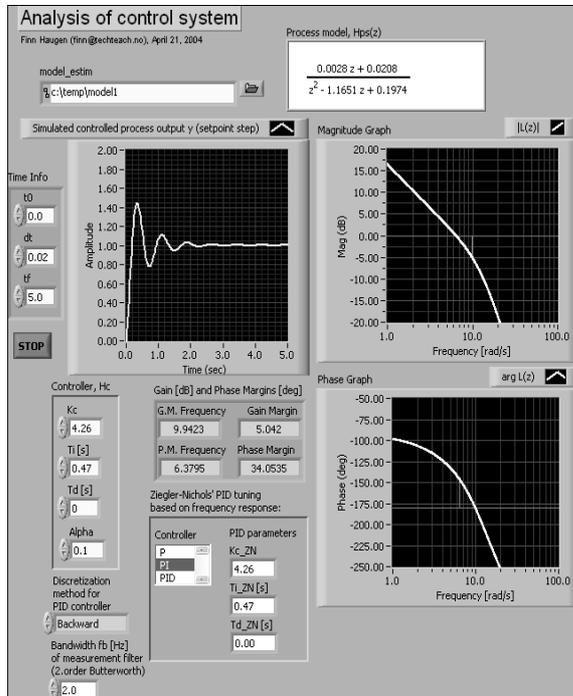


Figure 16: Front panel of control analysis design.vi

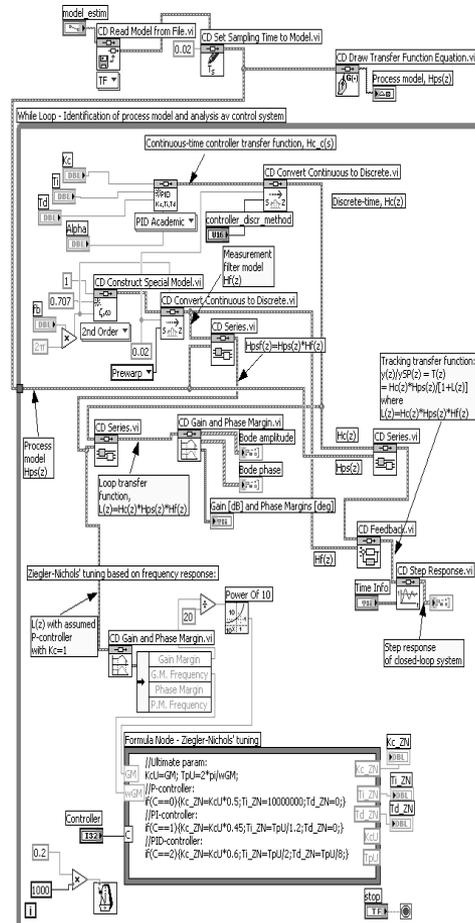


Figure 17: Block diagram of control analysis design.vi

3.2 Comparing simulated control system and real control system (target: PC)

real and simulated control system.vi shown below simulates a PID control system for the estimated process model.

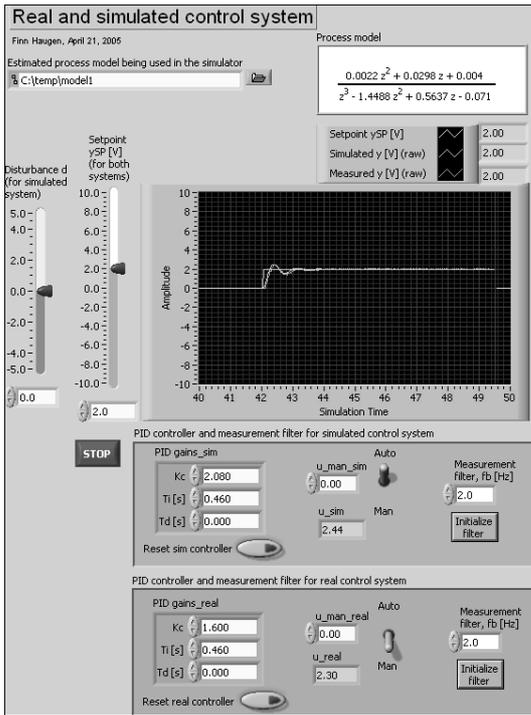


Figure 18: Front panel of real and simulated control system.vi

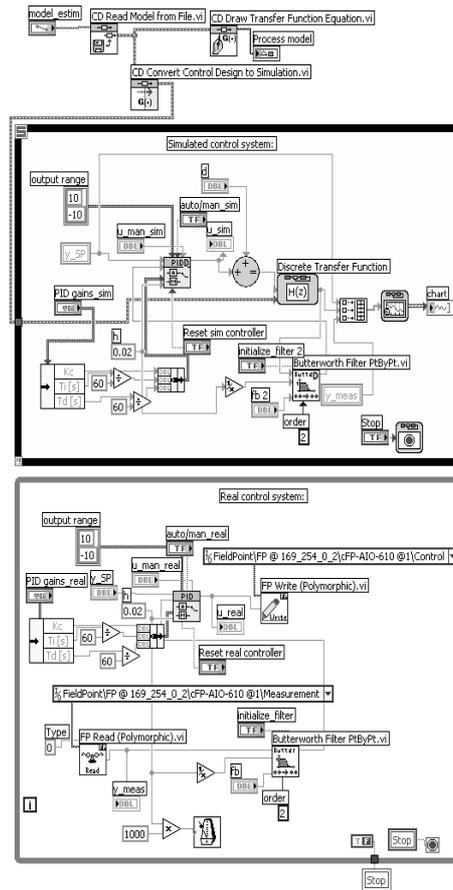


Figure 19: Block diagram of real and simulated control system.vi

4 Implementing and testing the control system on a real-time target (Compact FieldPoint)

rt_control_system.vi shown below implements the control system to be downloaded and run on the FieldPoint system.

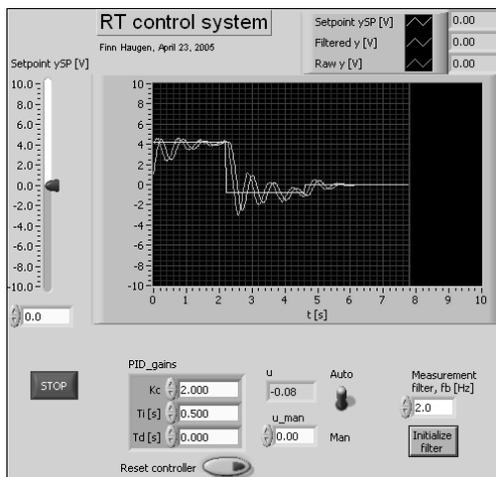


Figure 20: Front panel of rt_control_system.vi

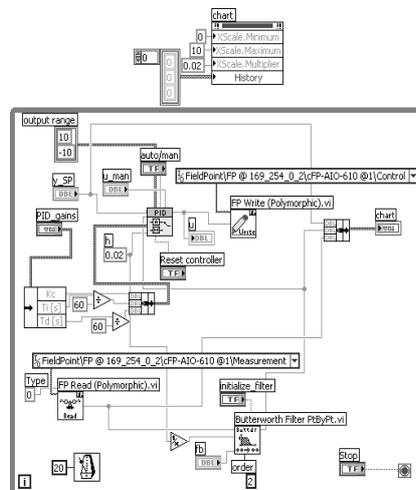


Figure 21: Block diagram of rt_control system.vi

5 Discussion, conclusion, and future work

It has been demonstrated how functions for system identification, simulation and control analysis and design can be applied for tuning and testing of a practical control system. These functions are based on an estimated discrete-time transfer function of the process to be controlled. The transfer function stems from a state-space model estimated using a subspace method. Functions available in LabVIEW were demonstrated in this presentation, but similar functions are available in other tools, as MATLAB.

The model order was selected by the user based on a visual comparison of the measured response and simulated response. Based on this visual comparison, the model order was actually selected different from the "optimal" order indicated by the Subspace-method. Although the visual comparison may be a good basis for selecting model order in a partly manual application, it can, of course, not be used as a validation method in a fully automated system. Future work will focus on building a fully automated system for model estimation and control design. Alternative validation methods and alternative estimation methods will then be considered, as the prediction error method with validation based on e.g. the FPE (Final Prediction-Error criterion) or the AIC (Akaike's Information Theoretic Criterion).

Finally, LabVIEW is a nice tool for developing application for system identification, simulation, and control system analysis and design. This is due to its graphical programming environment, easy physical I/O support, and several powerful toolkits recently added. At Telemark University College we use LabVIEW increasingly in student labs to bring theory into practical applications.

References

- [1] Haugen, F. 2005: *Discrete-time signals and systems*,
http://techteach.no/publications/discretetime_signals_systems/discrete.pdf
- [2] Haugen, F. 2005 *Introduction to LabVIEW Simulation Module*,
http://techteach.no/publications/labview_simulation
- [3] Haugen, F. 2005: *Introduction to LabVIEW Control Design Toolkit*,
http://techteach.no/publications/labview_control_design
- [4] Overschee, P. V., and B. De Moor. 1993. N4SID: *Subspace algorithms for the stochastic identification problem*. *Automatica* 29, no. 3:649–660.
- [5] National Instruments. 2004. *System Identification Toolkit User Manual*