How to Implement the Virtual Machine Concept Using xPC Target

Jens Bathelt, Swiss Federal Institute of Technology Zurich, Institute of Mechanics, CH-8092 Zurich, Switzerland, bathelt@imes.mavt.ethz.ch

Anders Jönsson, Blekinge Institute of Technology, Department of Mechanical Engineering, SE-371 79 Karlskrona, Sweden, anders.jonsson@bth.se

Abstract

This paper presents an overview of the historical background of the virtual machine, containing a real *control,* a machine *simulation* and a 3D machine *visualisation*. This setup enables realistic system simulations, since the simulation input comes from a real control. The first known implementation of the virtual machine using xPC Target is described with an existing water jet cutting machine as an example.

MathWorks products offer all necessary software for the presented setup, except the interface for the actual speed value from the simulated incremental encoder to the control. The unique xPC driver implementation, as a noninlined C-MEX S-Function, is presented at the end of this paper.

The successful implementation of the virtual machine demonstrates the feasibility of the presented approach.

Keywords: virtual machine, xPC Target, machine control, incremental encoder, S-Function

1. Virtual machine concept

Modern manufacturing machines are highly multidisciplinary, and with demands on short time-to-market, product development based on traditional prototype testing has become impractical. By using virtual models, it is possible to test large numbers of variants and optimise the product with the aid of a minimum of physical prototypes.

The virtual prototype presented in this work is called virtual machine and consists of a real machine *control* bi-directional connected to a machine *simulation*, which is bi-directional connected to a three dimensional *visualisation* of the machine as shown in figure 1.

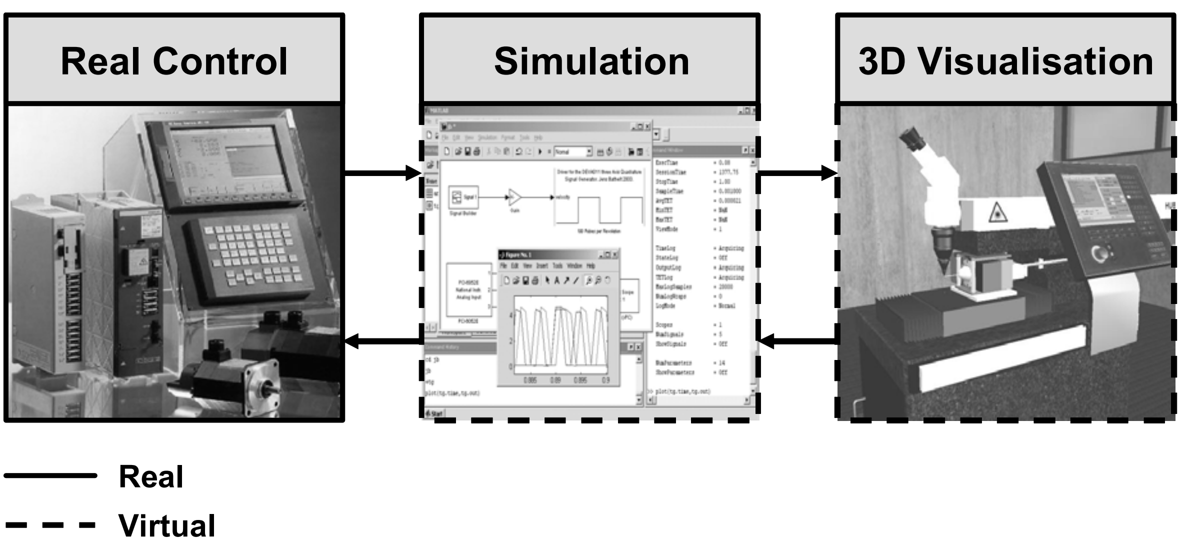

Figure 1. Virtual machine concept

Why a real control and not a simulation of a control? The idea of the virtual machine is to get the input for the machine simulation like in reality. The output of the simulation must then be returned to the control in real-time. This setup ensures a proper setup of the simulation, but the simulation must also fulfil the real-time requirement from the control system.

Why a 3D Visualisation at all? The visualisation is not necessary to run the simulation, but it gives instant feedback over the global machine behaviour and collisions can be observed and detected. There are no strong real-time requirements between the visualisation and the simulation.

The first virtual machine was implemented by Karsten Kreusch [1]. The motivation in that project was to verify machine controls. The starting and focal point was the CNC (Computer Numeric Control). From that the path via virtual servo motors / encoders to the real-time operating system VxWorks and the machine simulation was discovered. After that AnySIM was used to visualise the machine.

The second virtual machine was implemented by Stefan Dierssen [2]. In that work the virtual machine concept, as shown in figure 1, is explicitly described. The starting and focal point of this work was the 3D visualisation. High-end VR systems like VD2 and dVISE where used to visualise the machine. Analogue to Karsten Kreusch, TCP/IP was used to connect to a second computer, where the event oriented software WinMOD was simulating the machine. Different controls where connected to WinMOD: PLC (Programmable Logic Controller), CNC and Soft-CNC.

In this work the third known virtual machine is implemented. The starting and focal point of this work was the machine simulation. The complementary cooperation with Stefan Dierssen and Karsten Kreusch was leading to the virtual machine described in the next section.

The virtual machine concept, shown in figure 1, offers a broad scalability. According to the user requirements, the machine visualisation can be done by high-end VR systems or freeware VRML-viewers. The machine simulation can be advanced or just passing the control values direct to the visualisation.

Three user groups for the virtual machine where identified by Karsten Kreusch [1]:

1. The control manufactures
   - Testing the control
2. The machine builders
   - Analysing the machine behaviour
   - Virtual initial operation
   - Internal/external education
   - Sales, marketing
3. The machine users
   - Verifying the machine code
   - Training on the virtual machine

The specific business case will imply the requirements to scale the virtual machine. Data and software from earlier steps in the development process can be reused. For instance the simulation and visualisation setup for the development of the machine can also be used for the education of the machine users. The advertising department can use a full scale virtual machine at exhibitions and simple screen shot videos for a webpage.

2. Real machine setup

The virtual machine presented in this work is carried out together with the industrial partner Water Jet Sweden AB, Ronneby, Sweden. The primary business case is an analysis of water jet cutting machines. The long-term goal is a virtual machine to be used by the company for multidisciplinary systems optimisation. Figure 2 shows a possible configuration for such a real machine.

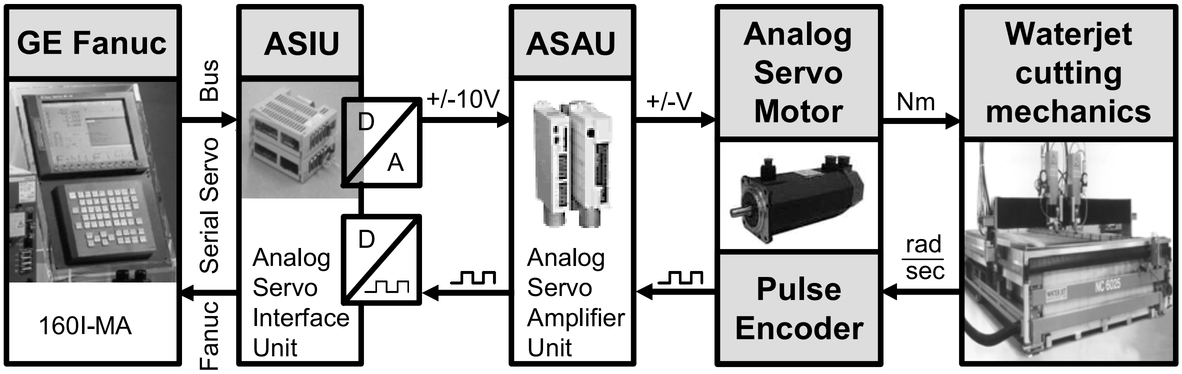

Figure 2. Real machine setup

A GE Fanuc CNC is connected via an optical fibre cable using a digital interface to an analog servo interface unit (ASIU). The target speed value is converted to a voltage between –10V and +10V and send to the analog servo amplifier unit (ASAU). The ASAU is amplifying the voltage and provides it together with a proper current to the analog servo motor, which results in a torque output from the motor. The torque acts on the mechanics giving a resulting motion. A pulse encoder is reading the velocity which is passed via the ASAU as a pulse to the ASIU. The ASIU is digitising the signal and sending it to the CNC. The analog servo interface is a world wide accepted industrial standard to connect a control and a servo motor.

3. Virtual machine setup

The broad range of the MathWorks products in combination with third-party products turned out to be a good platform for the virtual machine, since the primary goal of the ongoing project is the analysis of the dynamic behaviour of the water jet cutting machine for the machine builder. The resulting virtual machine setup is shown in figure 3.

The control and its interface is the same as for the real machine setup shown in figure 2. The real-time operating system xPC Target was chosen to guarantee that the cycle time on the simulation PC is less than the cycle time in the ASIU. The presented implementation of a virtual machine is the first one using xPC Target.

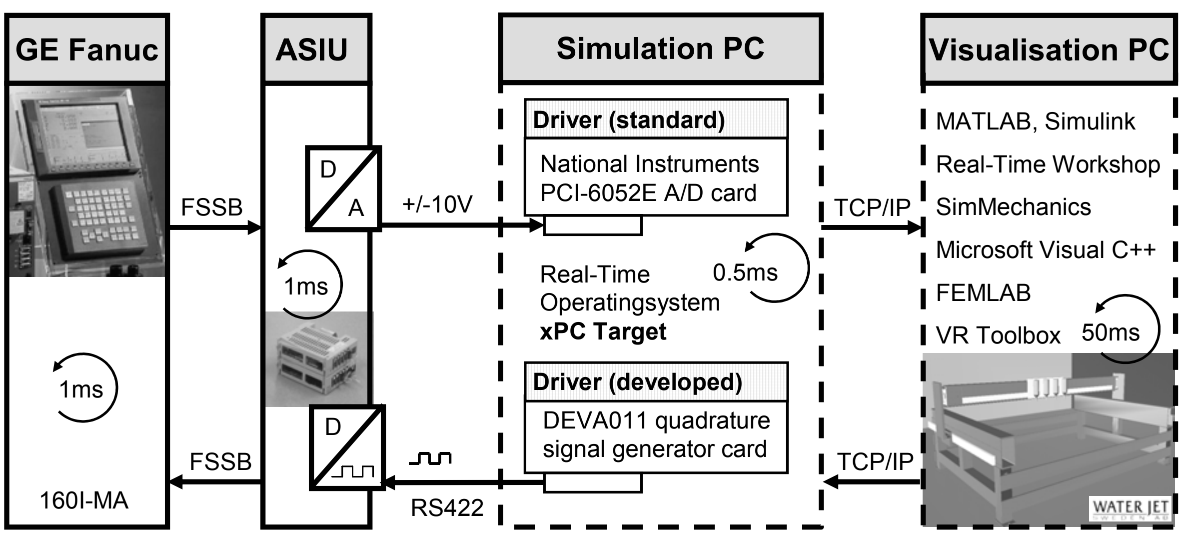

Figure 3. Virtual machine setup

The *target* speed value is a voltage between –10V and +10V, according to the target speed forced by the control. Instead of sending this voltage to the ASAU it is send to an A/D card (NI PCI-6052E). The digitised target speed value is the input for the simulation of the ASAU, the servo motor and the mechanics.

The *actual* speed value is based on the simulation result from the mechanics. The ASIU expects two square wave signals phase shifted by 90°. Usually, this signal comes from an incremental pulse encoder with frequencies up to several MHz. Cards capable of generating such signals are very rare and not supported by xPC Target. In this work the DEVA011 quadrature signal generator card is used. The driver developed receives the actual speed value, converts the speed into the quadrature waveform specification and will finally access the card with this data.

A common PC is used to run the MathWorks products MATLAB, Simulink, Real-Time Workshop, SimMechanics and the Virtual Reality Toolbox. State-space models generated from FEMLAB can also be used in Simulink. A VRML model is visualising the machine using the VR Toolbox, which is connected to the machine simulation running on xPC target. A frame rate of 20 frames per seconds is enough to get a good impression of the overall behaviour of the machine. Models for all 'non-CNC' blocks shown in figure 2 are implemented in Simulink, see figure 4.

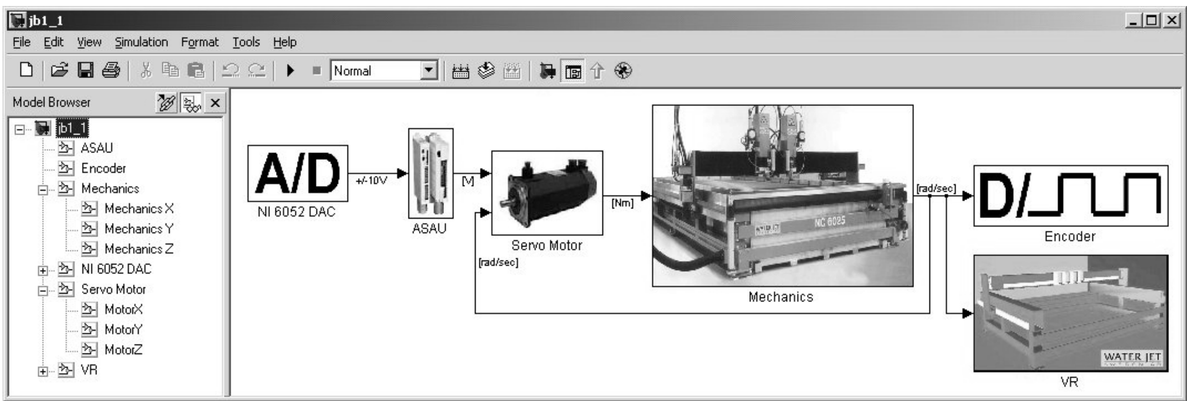

Figure 4. Main Simulink model

All necessary software, except the driver for the encoder, is available from MathWorks and third parties. The implementation of the driver will be discussed in the next section.

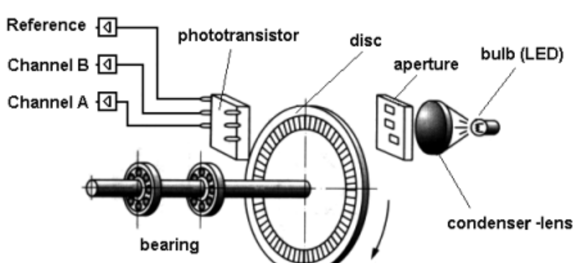

Figure 5. Incremental encoder

3. The XPC Target driver for the incremental encoder

An incremental encoder is converting motion as shown in figure 5 into a sequence of pulses as shown in figure 6.

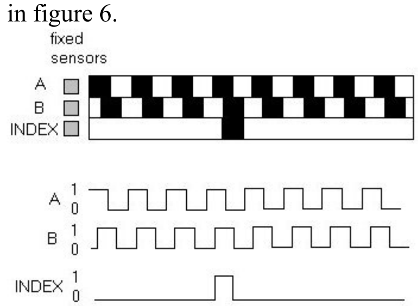

Figure 6. Two squarewave TTL signals phase shifted by 90° and a reference pulse produced by the encoder

The frequency of the A channel shown in figure 6 is corresponding to the actual rotational speed of the motor. Channel A in combination with Channel B is used to determine the direction of rotation by assessing which channel "leads" the other. The reference index yields one pulse per revolution, which is useful when counting full revolutions, as a reference to define a zero position.

Since the frequency can reach several MHz a special hardware is needed to produce these pulses. One of the very rare solutions available on the market is the DEVA011 three-axis quadrature signal generator card. The purpose of this card is to produce the signals shown in figure 6.

The device driver for this hardware is written as a noninlined C-MEX S-Function. In the mask of the S-Function all constant parameters for the card – such as the pulses per revolution and the width of the reference marker (0, 1, 2, 4) of the simulated encoder – can be defined, see figure 7. The S-Function expects only one input per axes: the actual speed value for the axes in radians per seconds.

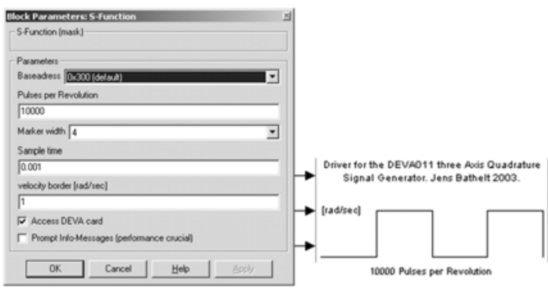

Figure 7. Mask and block of the driver

The output of this driver is a certain number of pulses per cycle. The resulting frequency is

$$\text{frequency}\left[\frac{\text{pulse}}{\text{sec}}\right] = \text{ppr}\left[\frac{\text{pulse}}{\text{revolution}}\right] \cdot \text{velocity}\left[\frac{\text{rad}}{\text{sec}}\right] \Big/ 2\pi \left[\frac{\text{rad}}{\text{rev.}}\right]$$

and the number of pulses per cycle send to the control is

$$\text{ppc}\left[\frac{\text{pulse}}{\text{cycle}}\right] = \text{frequency}\left[\frac{\text{pulse}}{\text{sec}}\right] \cdot \text{cycle time}\left[\frac{\text{sec}}{\text{cycle}}\right]$$

5. Conclusions and future work

A generic and scaleable setup for the virtual machine using xPC Target is described and exemplified. The real control used sends and receives signals to and from the simulation of the machine, as it would when connected to a real machine. This makes it possible to simulate the system accurately including the interaction between the machine and the control.

The interface between the control and the simulation is important. In this work the xPC target driver for the hardware that generates the signals from the simulation to the control is described. Since this driver delivers a standard incremental encoder signal, it can be used in conjunction with all numeric controls.

Although the system simulation is build using simplified models of the motor and the mechanics, the structure used for the virtual machine concept is scalable. This makes it easy to improve the level of detail for the simulation and the VR-model according to the demands from the usage.

The presented setup does only contain a simple machine model. A realistic model will be developed in the near future, according to the coordinated approach implemented at the Blekinge Institute of Technology [3]. The coordinated approach contains a loop consisting of 'Theoretical Modelling' ➔ 'Simulation' ➔ 'Experimental Verification' ➔ 'Optimisation' and back to 'Theoretical Modelling' again. Parallel to the presented work in this paper; measurements where done on a real water jet cutting machine in order to improve the machine model [4]. Improving the VR model is another option for the future.

References

[1] K. Kreusch. Verifikation numerischer Steuerungen an Virtuellen Werkzuegmaschinen, Berichte aus der Steuerungs- und Regelungstechnik, Shaker Verlag, Aachen, 2002.

[2] S. Dierssen. Systemkopplung zur komponenten-orientierten Simulation digitaler Produkte, Fortschr.-Ber., VDI Reihe 20 Rechnerunterstützte Verfahren, Nr. 358 VDI-Verlag, Düsseldorf, 2002.

[3] A. Jönsson. Modelling, Simulation and Experimental Investigation of a Rammer Compactor Machine, Licentiate Thesis, Luleå University of Technology, 2001.

[4] J. Wall, T. Englund, A. Berghuvud. Identification and Modelling of Structural Dynamic Characteristics of a Water Jet Cutting Machine, Accepted for publication at IMAC XXII, Hyatt Regency Dearborn, Dearborn, USA, 2004.