Realization of the Explicit Solution of Model-Based Predictive Control for Electric Drive Applications Using RT-Linux

Nael AL Sheakh Ameen
University of Wuppertal, Electrical Machines and Drives
Building FH,Rainer-Gruenter-Str. 21, 42119 Wuppertal, Germany
nael_nas@uni-wuppertal.de

Ralph M. Kennel
Technische Universitaet Muenchen,Electrical Drive Systems and Power Electronics
Arcisstr. 21, 80333 Muenchen, Germany
kennel@ieee.org

Abstract

This paper deals with the experimental design and implementation of a rapid prototype system dictated for electric drive applications using RTOS. The developed system with small size, low cost, and high efficiency, depends basically on the personal computer (PC) processor without the need to any external processor (e.g. digital signal processor (DSP)) to get the fast execution of control algorithms, and to support floating point operations. The PC-based rapid-prototyping system consists of an extension kit with all required interface boards to do the simulation and implementation of electrical drive algorithms. Almost all designed boards are based on programmable-logic devices (PLD) to reduce cost and size. To operate the hardware of the proposed system, a free available open source real-time operating system (RTOS) kernel under Linux is used. As case study from electric drive applications a Direct Model-based Predictive Current Control MBPC is selected as one of most complicated closed-loop control algorithm in this field, and some of experimental results are given in order to demonstrate the good functionality of the conceived system.

1 Introduction

Nowadays, digital control techniques are mostly carried out with micro-controllers or digital signal processors (DSPs), because almost all famous operating systems are desktop-oriented operating systems and do not support hard real-time applications. Thus, DSP controllers are considered by many engineers as an appropriate solution [1]. These controllers have an arithmetic logic unit particularly dedicated to the real-time computation. They also integrate peripheral units like analog-to-digital converters (ADCs) and timers, which are adapted to the needs of electrical drives. Nevertheless, the developed rapid-prototyping system combined with DSP has to contain all required software and packages to enhance the real-time applications. In such systems the PC is only used for developing, compiling the control software, collecting and showing the results, whereas the real-time tasks run completely on the DSP hardware. The required software to debug, compile and transfer the output binary file to the DSP system is always associated with cost and license. Also having a very cheap DSP evaluation-board doesn't mean that this system is ready to be used in drive applications without any hardware or software extension for debugging and implementation phases. High cost and license constraints associated with the high efficiency commercial prototype system like dSPACE [2], are a big problem in universities and technical institutes, because this cost appears for each researcher set-up. Because of this reason, the teaching staff at the universities always try to build their own set-up and prototype which fulfill their requirements [3] [4]. To overcome cost problem associated with such controller boards, the idea comes to design a rapid-prototyping system with small size, low cost and high efficiency, which depends on programmable logic devices (PLD) and free available real-time application interfaces under Linux. This work introduces a fully digital, hard real-time system depending basically on the PC-processor. It is designed to meet the requirements of modern rapid control prototyping and to perform the necessary simulation and implementation of electrical drives algorithms. The PC-based proposed system,like in [5], [6], does not need any external floating-point processor (as in other applications) to support the complex arithmetic operations , and it provides a wide selection of interfaces, like analogue to digital converters (ADC) board, digital to analogue converter (DAC) board, pulse width modulation (PWM) board, encoder interface board etc. The proposed system is also supported by high efficiency RTOS, a user-friendly programming environment and high level programming language. The PC-based rapid-prototyping system is developed in the Electrical Machines and Drives laboratory at Wuppertal University, and many researches and applications were applied on it ranging from simple to complicated control algorithms such as open-loop voltage frequency control [7], field-oriented control [6], sensor-less control [8], Grid applications [9], and predictive control [10]. This paper is organized as follows. In Section II, the construction and properties of the proposed system are firstly presented. Then, Section III introduces the chosen operating system and the developed programming environment. Finally, a case study from the electrical drive area is implemented in Section IV and experimental results are given to verify the validity of the developed system and to evaluate its performances.

2 CONSTRUCTION AND PROPERTIES OF THE PROPOSED SYSTEM

The designed system (see figure 1), consists of all required interface boards of electric drive, which assure safe connections and adaptation between the set-up equipments (e.g.: differential voltage probe, current sensor, position sensor, inverter and oscilloscope) and the PC. These interface boards are connected by an extension kit, which guarantees the parallel communication between the PC-mother board and other interface boards by the industrial standard adapter (ISA). A new PC with high efficiency and high processor speed is not necessary. It is sufficient to use a Pentium2- or a Pentium3 -PC with ISA slot. Since PCs of this type are not always available in the market or university storage, the PC interface board of the proposed system is designed to be connected with industrial PC (PC/104-equipped with ISA slot) as well, which is always available in market at acceptable price. The extension kit consists of 13 slots, which are equipped to be connected with 13 various interface boards. In the following paragraphs, main components of this system are described.

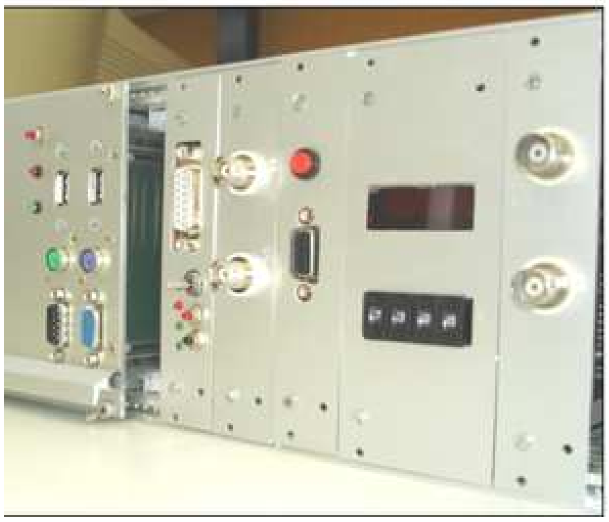

FIGURE 1: *The proposed system*

2.1 Extension Board

It is the spinal column of the proposed system and designed to provide safe parallel communication and data exchange between PC interface board and other interface boards. In combination with a suitable attached power supply, the extension board provides the required voltages (5V, +/- 15V) for other boards.

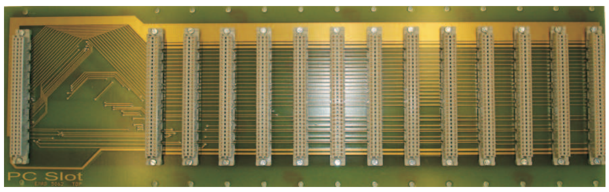

FIGURE 2: *Extension board*

Extension board gives the system more flexibility and extendibility by containing 12 slots with different addresses. Each slot is equipped with 2-physical addresses to communicate with PC and one of three available interrupt signals Irq3,5,10. One extra slot is added without addresses but with an interrupt signal Irq3. This slot can be used to insert interrupt generating board. As shown in figure 2 the first slot is designed to be connected with PC-interface board.

2.2 PC and PC-104 Interface Board

The designed PC interface board is so adapted to be connected to ISA slot of a normal desktop PC or of an industrial PC (PC-104 module) if the first one is not easily available. This board supplies the extension kit with 16-bit data bus, 24 enable signals to address 12 various interface boards and all other required interrupt and control (IOREAD, IOWRITE) signals. PLD-IC is used in this board as address decoder to reduce the cost, size and design complexity.

2.3 Pulse Width Modulation (PWM) Board

The most important expansion board in the proposed system is PWM board. It contains a field-programmable gate array (FPGA) that can be reprogrammed on the board (In-System Re-programming) by VHDL. The designed board is responsible to produce switching signals for driving two- or multi-level voltage source inverters (VSI). VSIs are nowadays commonly used in variable speed AC motor drives to produce variable, three-phase AC output voltages from a constant DC voltage.

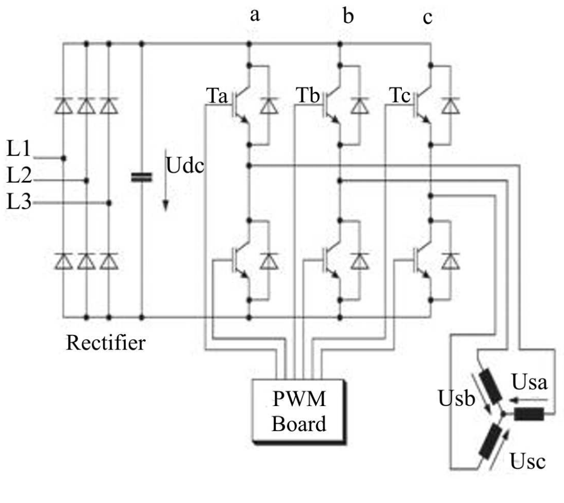

FIGURE 3: *The controlled system*

In this board, the desirable PWM strategy can be implemented to control the average output voltages of a voltage source inverter over a sufficiently small period, called switching period, by producing pulses of variable duty-cycle (T_a, T_b, T_c), (see figure 3). The reference voltages, which are computed on the PC, are refreshed exactly at the triangular PWM carrier vertex instants. Therefore, the mean values of the applied VSI voltages are equal to the reference ones [11], [12].PWM board has another function, which is the dead-time insertion between switching signals. Inserted dead time protects the power switches of the inverter against short circuit, when two switches on one leg are to be conducted at the same time. To set the wanted dead-time, a group of dip-switches are available on board surface. Beside, protection mechanism is implemented by VHDL program, and can easily be activated and modified. The third assigned task to the PWM board is an interrupt signal generation. As a result of this interrupt, the PC-processor scans the associated interrupt service routine. The interrupt frequency can be given in the main C-program.

2.4 Analogue to Digital Converter

In every controlled system some of analogue values (like currents and voltages) have to be sensed or measured from the control system. To satisfy this need, high resolution ADC interface board is introduced. Each ADC interface board contains two channels with 1.625μs conversation time and 4.992 mV resolution. A low pass filter can be activated in the input channels, and the amplitude of the analogue input signals can reach +/-10 V. The two channels have different addresses and different operating modes:

-First, they can be enabled from hardware; synchronized with an internal interrupt signal or with other signals.

-Second, they can be enabled by software commands to activate one or two channels at the same time.

2.5 Digital to Analogue Converter

For control and test purposes during simulation and implementation stages, there are always some signals or values in C-program to be showed on the oscilloscope. To response to this purpose DAC interface board is designed. Each DAC interface board contains two channels with 3μs settling time and with 4.885mV resolution. Each channel has a different address and the conversation process is initialized by software commands. For signal adjusting purpose, DAC interface board is equipped with trim resistors.

2.6 Hex Value Entering and Displaying Board

Hex-board is equipped with four hex-values seven segments display units and four hex-values dip switches. The hexadecimal entered- and displayed-values are software adapted and enable the programmer to change the operating condition of the controlled machine. To reduce the cost and size of this board, PLD-ICs are used for 7-segments decoding operation.

2.7 Incremental Encoder Interface Board

Rotary incremental encoders are common in position and motion sensing and they are preferred when low cost is important, or when only relative position is needed. Rotary encoders typically produce two TTL-outputs and their inverse. These output signals are shifted 90^o from each other and their sequence decides the rotation direction of motor shaft. Incremental rotary encoders often have a third reference signal, which comes one time in each shaft rotation. All three output signals and their inverse are considered in the developed board to remove the possible noise in work area. The designed interface board deals with incremental encoders with resolution of 16384 pulses/cycle maximum (14-bit). To protect the encoder and the rest of the board from possible short circuit, an optocoupler isolation circuit is integrated. An interface board with another type of encoders like absolute encoders or resolvers can be developed in the same way and inserted in the proposed system if it is required.

3 EMPLOYED SOFTWARE

3.1 Chosen Operating System

Since in the proposed digital system a desktop or industrial PC is employed, it is necessary to choose a proper operating system, which allows achieving: high efficiency, real-time application support, low interrupt latency, user-friendly interface and low cost. In the last decades the interest in Linux as desktop-oriented and as real-time operating system for industrial applications has been increased more and more. Linux distributions have long been used as server operating systems, and have risen to prominence in that area, also they have become increasingly popular on mainframes to pricing, compared to other mainframe operating systems. Due to its low cost and ease of modification, an embedded Linux is often used in embedded systems and it is to be considered as a major competitor to smart phones' operating systems. From this point of view and among all operating systems available from Microsoft [13], Linux, Macintosh, and others; Linux is selected as an appropriate OS for the proposed system, due to the following reasons [14]:

- It is subject to generalized public license (GPL) (free available, open source).

- Many variations are available by its different distributions.

- It depends on UNIX platform, which provides high security and performance.

- Its kernel can be adapted to support real-time applications with very low interrupt latency.

It supports loadable modules:

- Kernel modules can be inserted in kernel space by commands in user space.

- No need to restart the PC or reload the kernel after modifying the program.

These favorable characteristics of Linux as desktop oriented (general purposes) operating system are not enough for RT-applications. Linux as known is based on UNIX, which is designed as time sharing system to optimize its average performance (weighted by priorities), and Linux preserves this nature. This means that the RT-processes will suffer the most if the load increases. Currently Linux does not support an interface for informing the OS that a task is real-time task or informing it of any timing constraints on a task. As the basic Linux scheduler is time slice based on priorities, Linux as an operating system is not recommended for RT-applications for the following reasons:

- The time slice resolution may be higher than required for the RT-tasks and so it may be impossible to meet the timing demands.

- Increase the priority of real-time tasks will cause missing the deadline of the lower priority tasks.

- Schedulers fairness is not a desirable quality in real-time systems.

- Linux does not provide a reliable mechanism to wake a task up at a certain time. The sleep timer can only promise to wake a task up after a certain period.

- To protect OS data from corruption, the external interrupts are disabled at several sections in the basic Linux kernel . This adds unpredictability to the amount of time that takes to respond to real-time I/O.

Nevertheless, it is possible to obtain hard RT-Linux from general purposes operating system with saving most of advantages of Linux environments [18]. The main idea is to add a hard RT-scheduler, which stands for scheduling of RT-tasks. These tasks will be granted higher priorities than non RT tasks as described as follows(figure 4): The added RT-scheduler splits the RT-programs into two parts: small light parts with hard RT constraints and larger parts doing most of the processing. The light RT-parts are scheduled in the RT-scheduler, while the bigger parts run as normal processes under Linux. The two parts can communicate through a RT-FIFO or a non-blocking queue. RT-tasks are implemented as kernel modules, where their init-functions send all information about their release times, periods and deadlines to the RT-scheduler. RT-Linux scheduler bases on EDF (Earliest Deadline First) algorithm and runs the basic operating system process as the lowest priority process and will execute it as long as no RT-process is available. While Linux is running, it uses its own scheduler to schedule running tasks. Linux task runs as a process under the RT-scheduler and as a result can be interrupted by it at any time. The main performance characteristic of a RTOS is how fast it responses to internal or external events. These events can be internal software interrupts, external hardware interrupts etc. A Standard criteria can be introduced to compare the available RTOSs; interrupt latency, which can be defined as the time between an interface device requesting service (by raising an interrupt flag) and the time that the CPU needs to start the interrupt process.

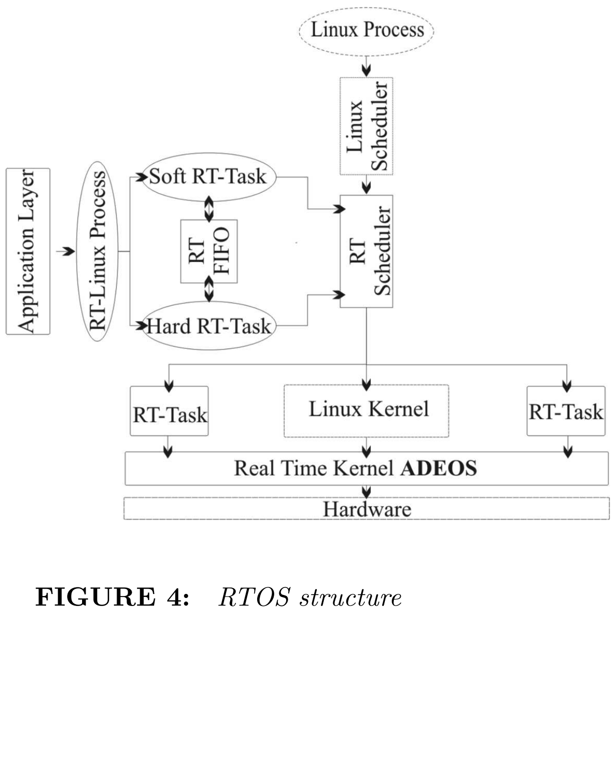

FIGURE 4: *RTOS structure*

All variants of RT-Linux have introduced modifications at the kernel level to reduce the interrupt latency between periodic interrupts to the microsecond range, allowing for faster response to external events and higher timing resolution. Table 1 shows the average interrupt latency for standard windows and all RT-Kernels under Linux. Depending on the performance criteria, the proper kernel can be selected. These kernels include a standard RT-kernel as described in IEEE 1003.1d, commercial RTOS-kernels, Micro-kernels with subject to patents [15] as well as free available Nano-kernel ADEOS. The adaptive domain environment for operating systems (ADEOS), released as an open source project in June 2002, provides an interface between the standard kernel and the hardware [16].

Table 1: Comparsion between RTOSs

Operating Systems	Application	Avg. Latency
Standard Windows	Non Real Time	100µ to 100mSec
Standard Linux	Soft Real Time	>10mSec
IEEE 1003.1d Linux - KURT: Kansas - Time Sys	Hard Real Time	10 to 100µSec
Commerical RTOS Kernels - Concurrent Computer Corporation - FSM Labs Inc. - Monta Vista Software Inc. - Quality Real Time System QRTS - RED Sonic Inc.	Hard Real Time	1 to 10µSec
Micro-Kernel Linux - RT Linux: New Mexico - RTAI: Milano	Hard Real Time	1 to 10µSec
Nano Kernel - ADEOS	Hard Real Time	1 to 10µSec

The free available ADEOS kernel is similar to the micro kernel approach mentioned above, but it is implemented in a way to prevent conflicts with the claims of its patent. The background history and design for ADEOS is described in a white paper published in 2001 [17]. ADEOS kernel was chosen and used in the proposed system. After patching the kernel as explained in [18] on the selected Linux distribution (Suse) some measurements are done on the hardware with the mathematical machine model before the real model is taken in operation. These measurements prove the efficiency of this kernel with the proposed hardware as a stable hard RT system. Interrupt latency test is done on the proposed system [7] by using the PWM board as interrupt generator. The operating system receives this interrupt signal and starts to run the related interrupt service routine (ISR), which starts its instructions with output command of logic "1" on digital to analogue board, output ("1", d2a-adr). Through the digital to analogue board and available interrupt test point on PWM board, the interrupt latency can be measured on the oscilloscope. This measurement was done under the following conditions:

- All system boards are inserted in their slots.

- 2PI current controllers are applied in the ISR (this means, there are some calculations in ISR).

- There are no programs running in the background.

- 400MHz Pentium-II Processor.

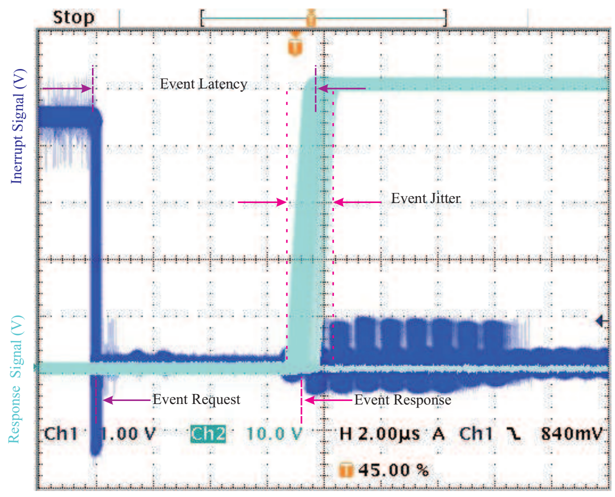

FIGURE 5: *Interrupt-latency time of the proposed system*

Figure 5 shows the measured interrupt-latency time, which equal to $8\mu s$ including DA conversion time (settling time of DAC is equal to $3\mu s$). These measurements were repeated many times to show the average latency in different load conditions of the operating system, where the max average interrupt latency does not exceed $8\mu s$. This small interrupt-latency time makes the developed system with the proposed Linux kernel suitable for almost all open- and closed-loop electric drive applications.

3.2 Developed Programming Environment

This section introduces an easy and user-friendly programming environment that is supported by a high level programming language (standard C). All required routines to communicate with designed interface boards are introduced in this framework to keep the user or researcher far away from hardware layer complications. Thus, in order to make the design of control algorithms more manageable and less intuitive, the designer has to strictly follow a set of steps and rules, which consist of an efficient design methodology. The main characteristics of this kind of methodologies are the re-usability of the already made designs, the consideration of the control performances, and finally, the reduction of the development time. The structural programming environment (figure 6) is organized in such way, that it lets the researcher easily insert an algorithm code in Interrupt Service Routine (ISR) with few modifications on other files. Start and stop files are responsible to initialize the system and save/restore the status of processor registers before entering/exiting the ISR. ISR should contain the main program or the implemented algorithm. Some definitions of assigned interface board addresses, required RT-libraries, and I/O functions are given in "Libraries.h". All required instructions to compile and execute source files are implemented in "Exec. Files group". There is no compiler or programming environment to be bought like visual studio of Microsoft, but the system depends on free available GCC compiler and RT-libraries from Linux under GPL agreement. And the portability characteristic of the developed programming environment gives the user the right to choose the proper Linux distribution freely from any constraints, but of course with some necessary modifications in the code.

4 CASE STUDY

In [7] the authors concentrated on the developed system and the integrated programming environment more than on the introduced application. Therefore, as a case study, one of the most well-known open-loop algorithms in electrical drives ($v/f = ct$) was applied to evaluate applicability and the performance of the proposed system. To prove the efficiency of the proposed system for much more complicated approaches in electric drives, Direct Model-based Predictive Current Control MBCPC [10]is selected here as one of most complicated closed-loop control algorithm in this field, and some of experimental results are given in order to demonstrate the good functionality of the conceived system.

4.1 Direct Model-Based Predictive Control

MPC is an optimal-control based method for constrained feedback control, where the optimization problem is solved at each time step starting from the current state and over a finite horizon (called prediction horizon Np). With respect to all constraints on states, outputs and inputs only the first element of the resulting optimal control sequence $\mathbf{u}$ is applied to the plant while the rest are discarded. At the next time step, this computation is repeated with

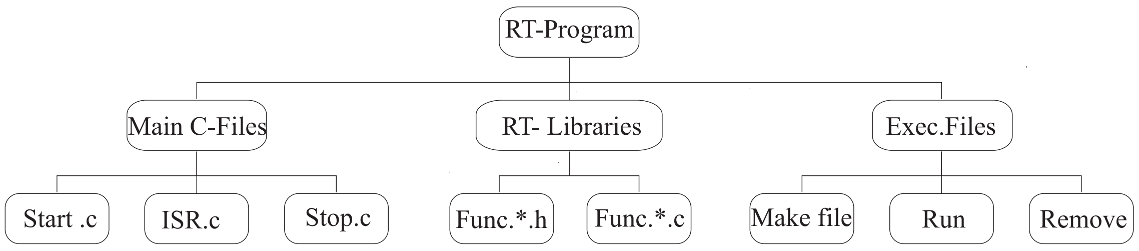

FIGURE 6: *RT-Program structure*

new state and over shifted horizon. Therefore an MPC problem can be addressed as constrained optimization problem (with 1, 2 or Inf norm) with receding horizon policy (RHC). Unfortunately, MPC still suffers some limitations make it not widely used in fast industrial applications. RHC policy with short horizon may be also steering the state to a part of state space where no solution satisfies the constraints. Therefore the feasibility and stabilities of RHC are not guaranteed in general. The lack of feasibility and stabilities of RHC is addressed in literature intensively and solved by adding the terminal set constraints for finite time optimization problem [19], or by solving the problem for infinite time *Np=Inf*. The main limitation of MPC to be a standard in the electric drive applications is the extensively online computation of the optimization problem, which increased dramatically with number of inputs and prediction horizon *Np*. In [20], the authors proposed multi-parametric programming as a solution to move the time-consuming part of the model predictive control off-line. In this case, the state vector is introduced as free parameter for the optimization problem. Thus the implicit solution of the MPC's optimization problem is converted into an explicit solution. The resulting control action will be a piecewise affine (PWA) function of the state, i.e. the state-space is divided into polytopes and inside each of them one single linear control law being valid. The explicit solution has the same characteristics concerning control performance and stability than the full on-line solution of MPC. For on-line implementation there are some methods introduced in literature to accelerate the execution depending on the system (continuous or discrete state/input variables) and on the optimization norm (1, 2 or Inf norm) [21] [22] [23] [24], and as results we get a look-up-table containing all feasible regions in state space and the associated control laws. To control the plant the controller simply has to find out in real time, in which of the polytopes the current state is located and to evaluate the appropriate linear control law. The optimal behavior of the controlled system is determined in advance either by pre-calculation or by selection of an optimal trajectory. Even if the chosen behavior does not fulfill all demands of the application, the user can be sure that this is the best possible behavior. In this paper, the authors aim to introduce an explicit solution of MBPC [10]to achieve direct control of an asynchronous machine fed by a two-level inverter using the developed Pentium system under RT-Linux [7].

4.2 MPC FORMULATIONS

The following matrix equations describe the behavior of the discrete-time linear time-invariant system:

$$\begin{aligned} x_{t+1} &= A \cdot x_t + B \cdot u_t \\ y_t &= C \cdot x_t \end{aligned} \tag{1}$$

The pair(A,B)is stabilizable, and the pair (A,C) is detectable. While fulfilling the state/output/input constraints:

$$\begin{aligned} x \in \mathbf{X} \subset \Re^n&, \mathbf{u} \in \mathbf{U} \subset \Re^m \\ \mathbf{X} &= \{x | x_k \in \mathbf{X}, k \geq 0\} \\ \mathbf{U} &= \{\mathbf{u} | u_k \in \mathbf{U}, k \geq 0\} \end{aligned} \tag{2}$$

These constraints are to be given at first as inequalities or as closed, bounded and convex set or polyhedral set. The full measurements of the state x(t) are assumed to be available at the current time t and no observer is required.

$$\begin{aligned} J^*(\mathbf{u}, x(t)) = &\min_{\mathbf{u}=u_t \cdots u_{t+N_u-1}} (\left\| P \cdot x_{t+N_p|t} \right\|_{p=1} + \\ &+ (\sum_{k=0}^{N_p-1} \left\| Q \cdot x_{t+k|t} \right\|_{p=1} + \\ &+ \left\| R \cdot u_{t+k|t} \right\|_{p=1})) \\ Q \geq 0 \quad , &\quad R > 0 \end{aligned} \tag{3}$$

subject to:

$$\begin{aligned} x_{min} \leq\ & x_{t+k|t} \leq x_{max},\ k = 1...N_c \\ y_{min} \leq\ & y_{t+k|t} \leq y_{max},\ k = 1...N_c \\ u_{min} \leq\ & u_{t+k} \leq u_{max},\ k = 0...N_c \end{aligned} \tag{4}$$

$$
\begin{array}{llll}
x_{t|t} & = x_t & & \\
x_{t+k+1|t} & = A \cdot x_{t+k|t} & +B \cdot u_{t+k|t} & ,\forall k \geq 0 \\
y_{t+k|t} & = C \cdot x_{t+k|t} & & ,\forall k \geq 0 \\
u_{t+k} & = K_{LQR} \cdot x_{t+k|t} & & ,N_u \leq k < N_p
\end{array}
\tag{5}
$$

$$
\begin{array}{ll}
K_{LQR} & = -(R + B' \cdot P \cdot B)^{-1} \cdot B' \cdot P \cdot A \\
P & = (A + B \cdot K_{LQR})' \cdot P \cdot (A + B \cdot K_{LQR}) \\
 & \quad +K'_{LQR} \cdot R \cdot K_{LQR} + Q
\end{array}
\tag{6}
$$

For constrained infinite-time optimal controllers the constraints are to be applied on the predicted states and on the calculated control laws for a finite constraint horizon Nc as in Eq. 4, therefore the constraints satisfaction in this horizon and system stability should be guaranteed. The stability of MPC problem depends mainly on the proper choice of Nu, Np, Nc, P, Q and R; therefore, reducing this dependency is useful if the stability and feasibility is still assured. This can be done by imposing some constraints on the Nu, Np, Nc and P to enforce the state trajectory with the time to reach some invariant set at the end of prediction horizon, and let $Q \geq 0$, $R > 0$ to be freely chosen as tuning parameters affect on the performance index [25]. If $N_u = N_p \leq N_\infty (N_\infty$ is the associated horizon with the maximal controllable invariant set)gives a stable system for a controllable invariant set covers all bounded state space, it is not necessary to increase the prediction horizon Np over N_∞ since that will increase also the complexity of the controller. This optimization problem is to be solved at each time step t, where the $x_{t+k|t}$ is the predicted state vector at (t+k) with respect to the initial state at the time t. Eq. 5 implies that after Nu time steps, the control is switched to unconstrained LQR (Linear Quadratic Regulator) depending on the feedback gain K_{LQR}, and P as solution of discrete Riccati Equation [26] [27].

4.3 System Model

The model used in this paper combines the model of the machine and the model of the inverter in a way that we do not need any more any modulation strategy like sub-harmonic or space vector modulation, where the predictive controller produces directly the optimal switching states (see figure 7). Two-level voltage-source inverters are widely used to control the machine, and it provides finite combinations of possible switching states to produce the required voltage at the input of the machine. Eight possible discrete-combinations of the switching states for 2-Level inverter and the continuous current and voltage signals of the machine, result in the complete model (machine + inverter) the hybrid nature. Hybrid systems with multi-parametric programming are addressed also in the literature [20] [28] [29] with different degrees of computation complexity [30]. In this paper, the authors used linear cost function introduced in [20] to solve the optimization problem .The complex representation of the induction machine introduced in [31] in stationary reference frame is considered in this part. Regarding the derived model of induction machine; there exists some non-linearity originating from the cross-coupling between the $is\alpha, is\beta$ currents. Due to the low influence of this cross-coupling on the machine performance ([32], see figure 7) it will not be considered in the machine model.

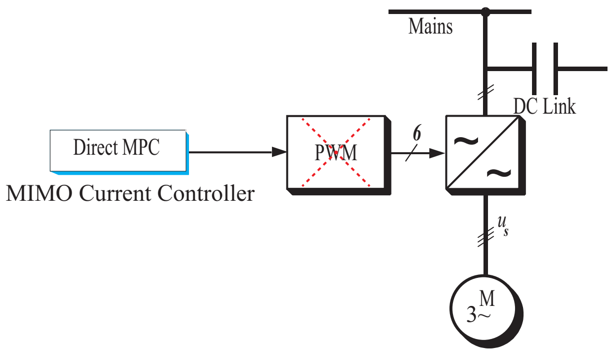

FIGURE 7: *System Model*

The resulting machine model without considering this disturbance could be restated as simple 2-PT1 element. Where the stator voltages are the input variables of the controlled system and the stator currents are output/state variables. With this MIMO-nature described by simple 2-PT1 elements the SISO structure of the control loop disappears in field oriented control, figure 6. The inverter is usually modeled in PI-control as time-delay element [11].

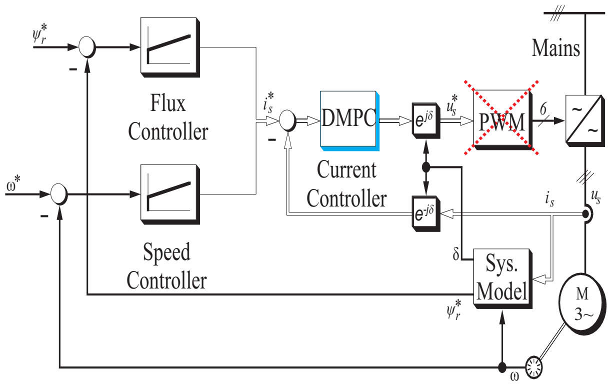

FIGURE 8: *Control Structure*

The caused time-delay from the inverter and from digital system is considered in the proposed model as a single time-step delay. Realization of direct control of the inverter without using any modulation method introduces the switching states S_a, S_b, S_c as new system inputs (figure 8). The discrete-time LTI-system model of the complete system can be given as:

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}_{k+1} = A_d \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}_k + B_d \cdot \begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix}_k \qquad (7)$$

where:

$$A_d = \begin{bmatrix} 1 - \frac{T_s}{\tau'_\sigma} & 0 \\ 0 & 1 - \frac{T_s}{\tau'_\sigma} \end{bmatrix} \qquad (8)$$

$$B_d = \begin{bmatrix} \frac{2 \cdot T_s}{3 \cdot r_\sigma \cdot \tau'_\sigma} & \frac{-T_s}{3 \cdot r_\sigma \cdot \tau'_\sigma} & \frac{-T_s}{3 \cdot r_\sigma \cdot \tau'_\sigma} \\ 0 & \frac{T_s}{\sqrt{3} \cdot r_\sigma \cdot \tau'_\sigma} & \frac{-T_s}{\sqrt{3} \cdot r_\sigma \cdot \tau'_\sigma} \end{bmatrix} \qquad (9)$$

$$\tau'_\sigma = \frac{\sigma \cdot l_s}{r_\sigma}, \; r_\sigma = r_s + r_r \cdot k_r^2, \; k_r = \frac{l_m}{l_r}, \; \sigma = 1 - \frac{l_m^2}{l_s \cdot l_r}$$

$S_{a,b,c} \in \{On, Off\}$

With the constraints implied on the state/input variables, the controllable invariant set with prediction horizon $N_p = 3 < N_\infty$,figure 9, covers the complete bounded system space. This means, that there are no initial state x_0 of the bounded state space, cannot be steered with a control law to the origin. Steering the state to the origin is not useful when the state variable has to track some reference signal. Then the system model with discrete inputs shall be reformulated to consider the zero-offset tracking in steady state and the delay [33] [34]. A toolbox for Matlab (developed at the Zurich University ETHZ [35] [36]) has been used to prove the stability and to get the explicit solution of the optimization problem.

4.4 EXPERIMENTAL RESULTS

During the implementation, the MPC controller has to determine the number (r) of the respective region containing the current state and then it has to evaluate the proper affine function Eq. 10. As the generated regions, figure 9 , are not sorted the author in [22] suggests an algorithm to reduce the on-line computations to make the explicit solution of MPC applicable in fast processes like electrical drives. The suggested algorithm aims to build a balance binary search tree contains in its leaf nodes the regions and the associated control laws. Finally the explicit solution of model-based predictive control using binary search tree strategy will be reduced to a look-up table (LUT). The look-up table contains the regions coordinates and the control law parameters F^r, G^r.

$$u_k = F^r \cdot x_k + G^r \qquad (10)$$

,where the symbol r refers to the region number contains the current state x_k. LUT is stored in a separate file in the programming environments to be loaded into the Linux kernel space as a part of the final object file during the execution.

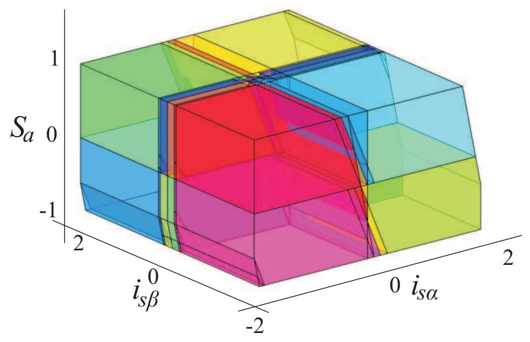

FIGURE 9: *The explicit controller regions*

Figure 10 shows one of the applied control laws Sa and the step response of the stator current i_{sq} with $400\mu s$ rise time. Step signal was only applied on the stator current i_{sq}, whereas i_{sd} was plotted without excitation to show the cross-coupling effect between the currents.

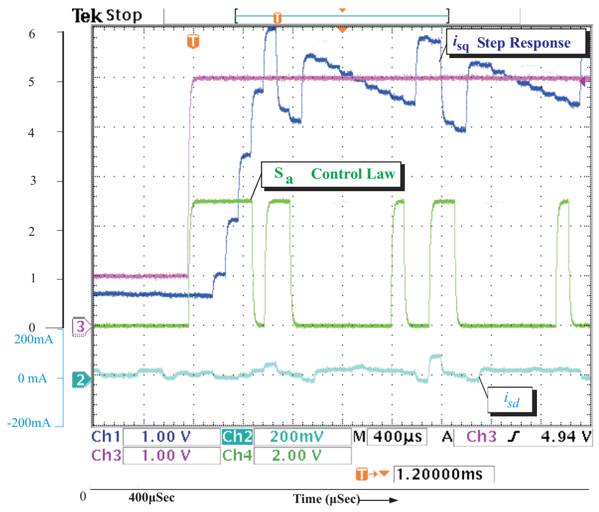

FIGURE 10: *Machine currents i_{sd} , i_{sq} and the control input S_a*

The current oscillation around the reference signal appears as expected results from the direct control of the inverter without any kind of modulation. The modulator provides a variable switching instant during the sampling time, whereas with direct control the switching states are applied to the inverter at the beginning of each cycle. Figure 11 shows machine currents $is\alpha$,$is\beta$ at half the nominal speed of the machine and their references. The sampling time used for this experiment is set to $T_s = 102\mu s$, and the consuming-time during the evaluation process of DMBP-Controller is not more than $5\mu s$.Having a constant reference of 50 Hz with amplitude of 1pu, the off-line optimization method develops a switching frequency of 890 Hz with Total Harmonic Distortions THD of 6.53%.

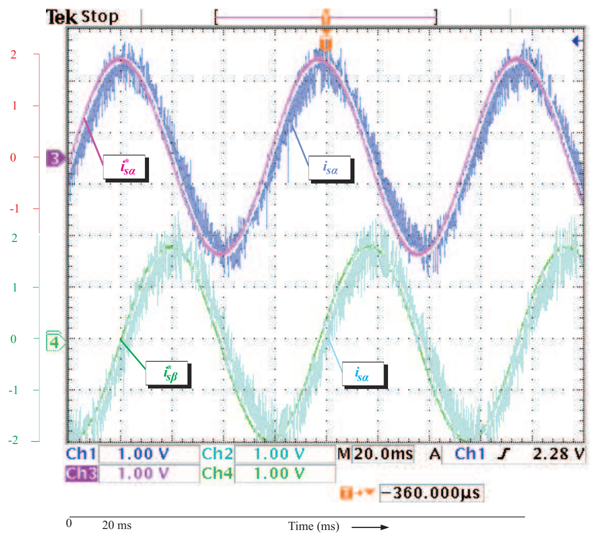

FIGURE 11: *Machine currents $is\alpha$, $is\beta$*

5 CONCLUSION

In this paper a low cost high efficiency rapid-prototyping designed to support research applications in electrical drives has been presented. It is based on personal computer and has benefited from its capabilities and high processor frequency to support sophisticated algorithm implementations and floating point operations. Concerning the operating system, a free available, RT-Linux kernel with high efficiency and very low interrupt latency time has been used. The proposed system has been supported with an easy environment for a high level programming language (C language). As case study, direct model-based current predictive control DMBCPC strategy for the control of an induction machine in a closed loop fashion, has been presented and the obtained results gave proof that such solution can reach the efficiency and performance level of their commercial counterparts, keeping at the same time a high degree of flexibility, programmability, extendibility. The explicit solution of DMBCPC ,represented as look-up table LUT , was stored in a separate file in the programming environment to be compiled and inserted in the kernel space. So the main challenge of realizing the explicit solution using RT-Linux is the size of the LUT , especially for high number of switching states like in multi-level inverters.

6 ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Syrian Ministry of Higher Education.

References

[1] *Microprocessors for power electronics and electrical drives applications*,C. Cecati,Sep. 1999,IEEE . Electron. Soc. Newslett., vol. 46, no. 3, pp. 5-9.

[2] *Rapid Control Prototyping of a Permanent Magnet DC Motor Drive System using dSPACE and Mathworks Simulink* ,K. Meah, S. Hietpas, and S. Ula,25 Feb. - 1 Mar., 2007,Applied Power Electronics Conference, APEC'07 , pp. 856-861.

[3] *UFRJ Power Electronics Teaching Lab: Ten Years,* L. G. B. Rolim , R.M. Stephan , W. I. Suemitsu , and J.L. da Silva Neto ,2005,IEEE Power Eletronics Education Workshop, pp. 69-73.

[4] *Design and implementation of a power-electronic remote-laboratory (ELEPOT-rLab),* R. M. Fernandes , L. G. B. Rolim , and W. I. Suemitsu, 2003,ISIE'03 - International Simposium on Industrial Electronics. Anais do ISIE'03 - International Simposium on Industrial Electronics, vol.1, pp.307-311.

[5] *Real-Time, PC-Based Simulator of Electric Systems and Drives,* S. Abourida, C.Dufour, J. Belanger, G. Murere, N. Lechevin, and Biao Yu,2002,APEC seventeenth Annu. IEEE, vol.1, pp. 433-438.

[6] *A Rapid-Prototyping System Based on Standard PCs with RTAI as Real-Time Operating System,* A. Linder, ,Nov. 26 - 29, 2001,Third Real-Time Linux Workshop, Milano, Italy.

[7] *Design of a Digital System Dedicated for Electrical Drive Applications*,N. AL Sheakh Ameen, A. A. Naassani , R. M. Kennel,2010,European

Power Electronics And Drives Association, EPE Journal.

[8] *Position sensorless control of the Reluctance Synchronous Machine considering High Frequency inductances,*H.W. de Kock, M.J.Kamper , O.C. Ferreira, R.M. Kennel, 2007,PEDS '07. 7th International Conference on,pp. 812 - 821.

[9] *Sensorless Control of 3-Phase PWM Rectifier in Case of Grid Phase Disconnection,* P. Szczupak, R. Kennel, T.Boller, 2005 ,Power Electronics Specialists Conference. PESC '05. IEEE 36th, pp. 2019 - 2022.

[10] *Comparative Analysis of On-Line and Off-Line Explicit Solutions, applied in a Predictive Direct Current Control,* J. C. Ramirez Martinez, R. Kennel, and N. AL Sheakh Ameen,2010,5th International Conference on Power Electronics, Machines and Drives PEMD ,Brighton, UK.

[11] *Pulse Width Modulation for Electronic Power Conversion,*J. Holtz,Aug.1994,Proceedings of the IEEE, vol.82, no. 8, pp.1194-1214.

[12] *Improved digital current control methods in switched reluctance motor drives,* F. Blaabjerg, P. C. Kjaer, P. O. Rasmussen, and C. Cossar, May. 1999,IEEE Trans. Power Electron., vol. 14, no. 3, pp. 563-572.

[13] *Windows NT Real-Time Extensions better or worse?,Available: http://www.realtime-info.be,*M. Timmerman, B. V. Beneden, and L. Uhres ,1998,.Real-Time Magazine, 3(98), pp.11-19.

[14] *Introduction to Linux for Real-Time Control, Available: http://www.isd.mel.nist.gov/projects/rtlinux/intro-rtl.pdf,*S. Hill, and B. Krishnamurthy,prepared for National Institute of Standards and Technology by Aeolean Inc.

[15] *Adding real-time support to general purpose operating systems,* V. Yodaiken, 1999.

[16] *ADEOS Project Homepage, http://www.opersys.com/adeos,* 2002.

[17] *Adaptive Domain Environment for Operating Systems,http://opersys.com/ftp/pub/Adeos/adeos.pdf,* K. Yaghmour, 2001,Technical report, OperSys.com.

[18] *RTAI Installation Guide, http://www.captain.at/programming/rtai/kernel-2.4.php*

[19] *Constrained Model Predictive Control: Stability and Optimality,*D. Q. Mayne, J. B. Rawlings, C.V. Rao and P.O.M. Scokaert,Jun. 2000,Automatica, vol. 36(6), pp. 789814.

[20] *An Algorithm for Multi-Parametric Mixed-Integer Linear Programming Problems,*V. Dua, and E. N. Pistikopoulos,2000,Annals of operations research, 2000 - Springer.

[21] *Piecewise Linear Optimal Controllers for Hybrid Systems,*A. Bemporad, F. Borrelli and M. Morari,,2000,ACC 2000 American ControlConference, pp. 1190-1194, Chicago .

[22] *Evaluation of Piecewise Affine Control via Binary Search Tree,*P. Tndel, T. A. Johansen, A. Bemporad,2003,.Automatica, vol. 39, pp.945-950

[23] *Complexity Reduction in Explicit Model Predictive Control,,*P. Tndel and T. A. Johansen,2002,.IFAC World Congress, Barcelona.

[24] *Expilicit Suboptimal Linear Quadratic Regulation with State and Input Constraints,*T. A. Johansen, I. Petersen and O. Slupphaug,Jul. 2002,Automatica, vol.38-7, Pages 1099-1111.

[25] *Robust Model Predictive Control: a Survey,*A. Bemporad and M. Morari,1999,Robustness in Identification and Control, vol. 245, pp. 207-226.

[26] *On Constrained Infinite-Time Linear Quadratic Optimal Control,*D. J. Chmielewski and V. Manousiouthakis,Nov. 1996,Systems and Control Letters, vol. 29-3, pp. 121 129.

[27] *Constrained Linear Quadratic Regulation,*P. O. M. Scokaert and J. B. Rawlings,1998,IEEE Transactions on Automatic Control, vol. 43-8,pp.1163-1169.

[28] *An Efficient Algorithm for Computing the State Feedback Optimal Control Law for Discrete Time Hybrid Systems,*F. Borrelli, M. Baotic, A. Bemporad and M. Morari,Jun. 2003,In Proc. 2003 American Control Conference, Denver, Colorado, USA.

[29] *A new Algorithm for Constrained Finite Time Optimal Control of Hybrid Systems with a Linear Performance Index,*M. Baotic, F.J. Christophersen and M. Morari, Sep. 2003,In European Control Conference, Cambridge, UK.

[30] *Low Complexity Control of Piecewise Affine Systems with Stability Guarantee,*P. Grieder, M. Kvasnica, M. Baotic and M. Morari,Jun. 2004,InAmerican Control Conference, Boston, USA.

[31] *The representation of AC Machine Dynamics by Complex Signal Flow Graphs,* J. Holtz, 1995,IEEE Transactions on Industrial Electronics, vol. 42, no.3, pp.263-271.

[32] *Direct Model Predictive Control-A new Direct Predictive Control Strategy for Electrical Drives,*A. Linder, R. Kennel,2005,Power Electronics and Applications, European Conference on 0-0 0 Page(s):10 pp. - P.10.

[33] *Model-Based Predictive Control of Electric Drives,*A. Linder,R. S. Kanchan,R. M. Kennel,P. Stolze, 2010, Cuvillier verlag Goettingen, Germany.

[34] *Optimal Control of PSM using Model Predictive Control with Integrator*,S. Matsutani,T. Zanma, Y. Sumiyoshi, M. Ishida,A. Imura, M. Fuitsuna,18-21 Aug. 2009,ICROS-SICE International Joit onference, Japan.

[35] *Multi-Parametric Toolbox(MPT),*M. Kvasnica, P.Grieder, M. Baotic, M. Morari, Mar. 2004,Lecture Notes in Computer Science, Vol. 2993,pp.448-462.

[36] *Design and Implementation of Model Predictive Control using Multi-Parametric Toolbox and YALIMP,* M. Kvasnica,M. Fikar,8-10 Sep. 2010,IEEE International Symposium on Computer-Aided Control System Design, Yokohama, Japan.