Fast Online Model Predictive Control of IPMSM Using Parallel Computing on FPGA

Michael Leuer and Joachim Böcker

Abstract—Model predictive control (MPC) offers a variety of advantages compared to conventional control methods. The problem with MPCs is the high computational cost and the associated long control cycle time. This makes the use of MPCs unattractive for processes with small time constants, as in permanent magnet synchronous motors with interior magnets (IPMSM) for electric vehicles. In this paper a model predictive control method for nonlinear systems with inherent output limitation is presented. This approach offers real-time capability for online MPCs even for processes with time constants in the millisecond range. This becomes feasible by the possibility of parallel computation, as provided by a FPGA (Field Programmable Gate Array). In the following, the functional principle of this real-time MPC approach is presented and the functionality is verified by simulation results of an IPMSM control for automotive applications.

I. INTRODUCTION

ELECTRICAL and hybrid electrical vehicles are becoming increasingly important in the public interest and the automotive industry. As traction drives permanent magnet synchronous motors with interior magnets (IPMSM) are favored due to their high power and torque density as well as efficiency [1]. To reach the full performance potential of an electric drive, an adequate control is essential. Conventional controls for IPMSM are based in general on PI control algorithms, such as introduced in [2]. From the perspective of the control algorithm, the IPMSM can be represented as a coupled nonlinear system (see equation 1). When controlling this system with a PI control structure, a decoupling of the system is recommended. However, the decoupling cannot always be guaranteed due to physical limitations. This can lead to incorrect setting of the operation points. A further problem lies in the nonlinearity that is introduced into the control loop by the control variable restriction. Since the parameters of the PI controller are calculated only once during the design process, a compromise between short rise time and minimum overshoot has to be designed. Today’s remedy was creating reconfigurable control [3]. Hereby it can be problematic to switch between the different controls algorithms because of transients. That is why the principle of the model predictive control (MPC) is more effective. MPCs are based on the solution of a dynamic optimization problem on a moving horizon. Here, a model of the plant is used to predict the trace of the state variables over a specified time horizon. The manipulated variables can be scheduled in this way so that a faster and better damped transient response is achieved in comparison to conventional concepts. Here, possible control and state variable constraints are directly taken into account. By doing so, the model predictive control is particularly suitable for the control of coupled non-linear multivariable systems. The online optimization of the controller in each cycle step enables an intelligent foresighted planning of the control variables, in contrast to the reactive character of a PI controller. This leads to an optimal utilization of the available control variables - both for small as well as large deviations from the reference set point. Despite many advantages of MPC approaches enormous demands on processing power for high-dynamic control systems, as they are common in the electric drive technology, have to be faced. However, there are new MPC approaches which allow a reduction of the online computation time. In this context, the explicit MPC [4] has to be mentioned. The optimization is performed offline in the controller design. The results of this optimization are stored in an n-dimensional table, and read by the control algorithm during run time. By doing so, the online computational complexity of the MPC is drastically reduced. Such explicit MPC for IPMSM drives have been presented in [5], [6] and [7]. As a fast online MPC for nonlinear systems the suboptimal MPC, as presented in [8], is worth mentioning. With this method, the computational complexity can be reduced by calculating only a certain number of iterations for the optimization in each sampling step. In this way, only an approximate solution is determined. By initializing the optimization algorithm with the approximate solution in the following sampling step, the next calculated solution can be improved incrementally. The established MPC calculates the mean of the optimal manipulated variable by an optimization algorithm. The new fast MPC considers the adjustable voltage vectors of the switching states to solve the optimization problem best. The procedure solves the minimization problem without using an optimization algorithm.

II. MOTOR MODEL

The model predictive control is performing its predictions based on a model of the system which has to be controlled. Therby the plant has to be identified as precisely as possible. As a starting point for the model the voltage equations of the IPMSM

\[ u_d = R_i d + L_q i_d - \omega L_q i_q \]

\[ u_q = R_i q + L_q i_q + \omega (L_d i_q + \psi_p) \]  

(1)

are used. The dynamic behavior of the inverter is simulated by one P-T1 link with the corresponding inverter time constant.
III. MODEL PREDICTIVE CONTROL

The practical use of the model predictive control can be demonstrated by the fact that it has its origin in the industry and arrived in the research at universities later. The MPC was developed in the 70s in the petrochemical industry, where coupled multivariable systems with relatively large time constants have to be controlled. Meanwhile, due to its excellent controller performance the MPC was also applied in controls of high dynamic processes. However, problems arise if the model predictive control is to be used for processes with very small time constants, such as the electrical behavior of an IPMSM. Here, it is often impossible to optimize the manipulated variable sequence and perform the predicted calculation of the state trajectory within the desired control cycle time. To estimate the computing time relative to the control cycle time, an MPC with online optimization and consideration of circular manipulated variable limits was implemented. The basic flow chart of the MPC is shown in figure 2. The MPC implemented here operates with \( n_p = 6 \) prediction steps and \( n_s = 2 \) control steps. The state variable for the next \( n_p \) time steps are pre-calculated and the next \( n_s \) control values with respect to a quality function are optimized (see figure 1). Figure 3 provides a comparison of the MPC with a PI controller, which has been designed by the magnitude optimum method. Further more the figure shows the computation times that are required in each cycle step. As computing hardware, a personal computer with an Intel Core i7 CPU (2,8 GHz) and 4 GB memory was used. As operating system and simulation environment 64 Bit bit Windows 7 with MATLAB Simulink was used. Both control algorithms have been implemented with a control cycle time of 100 \( \mu s \). From the step responses of the currents it can be clearly seen that the model predictive control has a significantly better performance than the PI controller. The curve with the respective processing times for the corresponding sampling point shows that the optimization calculation has to be performed about 300 times faster in order to be realized as a real-time capable MPC. Due to the considerable improvement of the MPC controller performance compared to PI control algorithms, a real-time implementation of MPC is desirable. One possible method of a real-time online MPC is presented below.
IV. VECTOR-MPC APPROACH

Conventional MPCs usually work with an optimization algorithm, in order to achieve an optimal control variable sequence with respect to a defined quality function. As a result a voltage mean value is calculated as the manipulated variable. These mean values can be generated from adjustable voltage vectors by using a pulse width modulation (PWM).

In the method presented here, the optimization problem is addressed differently, however. It can be concluded through systems analysis of the inverter (see Figure 4) that $2^3$ switch states and 8 voltage vectors can be realized. These voltage vectors are shown in Figure 5. Since the voltage vector $v_0$ and $v_7$ are the same, only 7 different voltage vectors are possible. The following proposal, called Vector-MPC, does not calculate the optimal control value, which has to be adjusted in turn by using a PWM from different voltage vectors averaged over time. Rather, it is investigated very specifically which of the 8 voltage vectors minimizes the quality function best. The basic procedure has been introduced in [7] as an offline approach. In the selected approach the optimal control variable

\[ u_c(t) = \min \{ \epsilon \} \]

sequence will not be determined by means of an optimization algorithm but rather by means of trial & error. While in conventional MPCs any manipulated variables must be taken into consideration, the Vector-MPC only considers 8 realizable voltage vectors. By doing so, the computational cost can be significantly reduced, so that a reduction of the controller cycle times are possible. The problem is that during a period only one switching change is possible. To maintain the current ripple to an acceptable level, further reduction of the control cycle time is required.

A. Working Principle of The Real-time Online Vector-MPC

In order to find the optimal voltage vector for the next output step, the eight possible voltage vectors will be applied to the plant-model contained in the MPC and the model response is calculated. In this way the resulting quality function for each voltage vector can be calculated. As the output manipulated variable, the voltage vector is chosen which fulfills the smallest value of the quality function. This procedure allows saving the computation and time consuming optimization algorithm. Furthermore, this algorithm is suitable for the implementation on a FPGA, as the model calculations and the calculation of the voltage vectors associated quality functions can be performed in parallel. Figure 6 represents the basic operation and Figure 7 represents the schematic functional flow chart of the MPC for one prediction step. Due to the fact, that it is only possible to switch the voltage vectors once during each controller period, a correspondingly lower control cycle time.
is necessary. With the division of the control algorithm on the dSPACE DS1006 quad-core processor board and the dSPACE DS5203 FPGA board, a control cycle time of 10 μs can be realized. As advantages of this Vector-MPC approach the possibility of parallelization (see Figure 6), the low computational complexity, as well as the optimal use of the available DC-link voltage can be mentioned. While the consideration of the output limit in an MPC optimization algorithm may require a high computational effort, the output limit is implied by the fact, that the Vector-MPC directly sets the full voltage vector. Unlike other control algorithms, the whole spanned Hexagon from the eight possible voltage vectors (see Figure 5) can be chosen as the working area. Moreover, no further processing of the manipulated variables, generated by the controller, by PWM or similar components is necessary because this approach generates directly the gate signals for the inverter.

**B. Vector-MPC Implementation**

As rapid control prototyping hardware a dSPACE RCP system consisting of a quad-core processor board and a FPGA board was used. For optimal utilization of the available hardware the control algorithm has been split up between the two dSPACE boards (see Figure 8). The actual control algorithm is performed in a parallel manner on the FPGA board to achieve a controller cycle time of 10 μs. The calculation of the state model and the update of the non-linear plant model is executed on the processor board. This is done with a cycle time of 100 μs. The caused error during this 10 times longer cycle time is only very small, as the state matrices are changing only slightly. The data exchange between the two control boards is done using the peripheral high-speed bus (PHS). For a better visualization of the distributed control algorithm, it will be explained with the phase-locked loop (PLL) which is shown in Figure 9. The measurement of the sine and cosine signal from the encoder of the IPMSM is done by the A/D-converters on the processor board. That is the reason why new measurement results are available only every 100 μs. However, for the Vector-MPC an updated rotor position signal is necessary every controller cycle step. To realize this, the integrator of the PLL is embedded on the FPGA board. In this way, the intermediate steps of the rotor position can be interpolated accordingly so that all 10 μs, an updated value is available. In addition to providing these interpolated rotor positions, the phase locked loop will be used to filter the measurement signal, and to predict the future rotor positions for the MPC. The example of the PLL shows the possibility of very properly allocating certain portions of the control algorithms to a suitable hardware, without increasing the computational effort. While splitting up the control algorithm, it is necessary to pay particular attention to the consideration of the different cycle times. Also the output control signals
have to be taken into consideration because of the limited switching frequency of the converter. If the switching of the inverter is allowed in every cycle step of the Vector-MPC, a very good controller performance will be reached. On the other hand, the inverter can be thermally destroyed due to the high switching losses. Therefore, the switching frequency was limited to 10 kHz. So the Vector MPC and the PI control with PWM will have the same maximum switching period for the following comparisons.

Following successful implementation of the Vector-MPC scheme using MATLAB Simulink and Xilinx System Generator, the performance of the Vector-MPC method was shown by simulations. For these simulations the real computing time on hardware has been considered.

C. Quality Function Design

After implementation of the parameterization the MPC has to be performed. In the model predictive control this is very easy because the MPC is based on the optimization of a quality function. Therefore it is only necessary to define a suitable quality function with appropriate weighting matrices. For the presented vector MPC following quality function is selected:

$$J = e^T Q e + u^T R u + \Delta s S + F^T \Delta f \rightarrow \min$$  \hspace{1cm} (2)

with

- $e = r_{k+1} - x_{k+1}$ ... control error after execution of the respective voltage vector;
- $\Delta u = u_k - u_{k+1}$ ... gradient of the respective voltage vector in d-q coordinates;
- $\Delta s$ ... number of necessary switching procedures for the respective voltage vector;
- $\Delta f$ ... number of switching procedures during the last five sampling periods that are higher than one;
- $Q, R, S, F$ ... weighting matrix of the control error, the gradient, the switching procedures and the switching frequency.

By strong weighting of the control error, a possible quick adjustment of the set value is reached, while a strong weighing of the voltage vector gradient results a slowly changing manipulated variable sequence. By weighing the switching operations, the switching frequency can be affected. That way, an optimal compromise between switching losses and dynamics of the control has to be found. The controller design, and the configuration of the MPC can be represented as pareto optimization. The choice of weighing matrices also depends on the scaling of the respective factor. A high scaling value in a weighting matrix means not directly that the appropriate factor in the quality function is also given more weight than any other.

V. SIMULATION RESULTS

To verify the operation of the Vector-MPC technique, a simulative comparison of the MPC with the PI controller was performed. The PI-controller was designed according to the magnitude optimum method. Since the Vector-MPC calculates the switching sequence, the switching behavior of the PI controller has also been considered rather than the average behavior as shown in Figure 3. The simulation results are shown in Figure 10. It can be seen, that this MPC approach is superior the PI control with regard to the control performance. The MPC distinguishes with short rise time and low overshoot. The weighting of the switching frequency and the gradient of the voltage vectors make it possible to reduce the switching frequency in stationary operating points compared to the PI control, without increasing the current ripple too much. To investigate the dynamics of the Vector-MPC with respect to its small-signal behavior, the simulation results for a small
step response are shown in Figure 11. Comparing Figure 10 with Figure 11, we see the benefits of the non-linear control algorithm over the linear PI-controller. While the step response of the linear PI control has the same shape for both the small signal behaviour, as well as the large-signal behaviour it can be seen that the MPC always seeks the quickest way to compensate the control error.

VI. Conclusion

Model predictive controls are characterized by their theoretical high controller performance. For implementation of traditional real-time online MPC the computational effort is so high that the solution of the optimization algorithm is likely to significantly exceed the control cycle time. However, it was shown that with the Vector-MPC a real-time online MPC can be implemented. This method does not rely on a complex and thus computationally expensive optimization algorithm and therefore turns out to be suitable for controlling processes with small time constants, such as IPMSM. Moreover the step response demonstrated that the dynamics of the drive can be improved by the use of the model predictive control.

REFERENCES


