Design choices for the prediction and optimization stage of finite-set model based predictive control

T.J. Vyncke, S. Thielemans, T. Dierickx, R. Dewitte, M. Jacxsens, and J.A. Melkebeek
Department of Electrical Energy, Systems and Automation (EESA)
Ghent University (UGent), Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
phone: +32 (0)9 264 3442, fax: +32 (0)9 264 3582, e-mail: Thomas.Vyncke@UGent.be

***Abstract*—The interest in applying model-based predictive control (MBPC) for power-electronic converters has grown tremendously in the past years. This is due to the fact that MBPC allows fast and accurate control of multiple controlled variables for hybrid systems such as a power electronic converter and its load. As MBPC is a family of possible controllers rather than one single controller, several design choices are to be made when implementing MBPC.**

In this paper several conceptual possibilities are considered and compared for two important parts of online Finite-Set MBPC (FS-MBPC) algorithm: the cost function in the optimizations step and the prediction model in the prediction step. These possibilities are studied for two different applications of FS-MBPC for power electronics. The cost function is studied in the application of output current and capacitor voltage control of a 3-level flying-capacitor inverter. The aspect of the prediction model is studied for the stator flux and torque control of an induction machine with a 2-level inverter. The two different applications illustrate the versatility of FS-MBPC.

In the study concerning the cost function firstly the comparison is made between quadratic and absolute value terms in the cost function. Comparable results are obtained, but a lower resource usage is obtained for the absolute value cost function. Secondly a capacitor voltage tracking control is compared to a control where the capacitor voltage may deviate without cost from the reference up to a certain voltage. The relaxed cost function results in better performance.

For the prediction model both a classical, parametric machine model and a back propagation artificial neural network are applied. Both are shown to be capable of a good control quality, the neural network version is much more versatile but has a higher computational burden. However, the number of neurons in the hidden layer should be suffciently high.

All studied aspects were verified with experimental results and these validate the simulation results. Even more important is the fact that these experiments prove the feasibility of implementing online finite-set MBPC in an FPGA for both applications.

***Index Terms*—MBPC, predictive control, FPGA implementation, flying-capacitor inverters, programmable digital hardware, induction motor, torque control**

I. Introduction

Model based predictive control (MBPC) provides a control technique which is very suitable for the control of power electronic converters and their load. The facts that MBPC is a multivariable and discrete-time control is indeed very advantageous in power electronics control. Furthermore specific control objectives can be defined with great flexibility. These control objectives are achieved satisfactorily because of the online optimization based on the prediction of the possible future system states. Recently the interest in using MBPC for power electronics control [1]–[5] has increased tremendously, mainly because of the availability of faster and cheaper processing power.

As the main strengths and the typical operation of a model based predictive controller are determined by the prediction and optimization, the possibilities for these parts of the algorithm should be investigated. In this paper several choices for the cost function and the prediction model are compared. For the cost function, tracking control terms with quadratic and absolute value norms are studied. In [6] a good range of weight factors is established for the quadratic terms. In this paper the influence of using absolute value norms on the established good range, the control quality and hardware usage are discussed. Furthermore the effect of replacing the tracking control term by a cost function with an tolerance band is studied. For the prediction model the effect of replacing an analytical parametric prediction model with a back-propagation artificial neural network is considered.

The effects of the design choices are illustrated in two applications. The cost function variations are studied in the context of output current and capacitor voltage control for a 3-level flying capacitor inverter. The prediction model variations are studied for the stator flux and torque control of an induction machine.

The prediction of the future system states and the optimisation, however, are computationally demanding. As the large computational burden is considered to be the main disadvantage of MBPC, the feasibility to implement the discussed model based predictive controllers is important. The provided experimental results not only validate the simulation results but also illustrate the feasibility of implementing both applications in an FPGA.

II. Finite-Set Model Based Predictive Control for Power Electronics

The principles of FS-MBPC are explained and applied to the control of power electronic converters in this section. In this paper only discrete-time controllers operating with a fixed update frequency f_u are considered. Further on in the paper two applications will be discussed. For both applications, two main control objectives for the FS-MBPC exist. In the first application, MBPC current control of a 3-level inverter, these are the tracking of the reference current and the balancing of the flying capacitor voltages. For the second application, predictive torque control of induction machines, these control

objectives are the tracking of the electromagnetic torque and stator flux references. In both applications all control objectives are achieved simultaneously by the multivariable control scheme. To this end the inputs for the FS-MBPC algorithm are the reference values and the measurements or estimations of the controlled variables. The output of the algorithm is one of the possible switch states of the converter (the finite set), without using any modulation scheme. At every update instant a new switch state can be applied and is maintained during the entire update period. This results in a spread spectrum switching frequency. The average switching frequency per switch will certainly be lower than the update frequency, f_u.

Online finite-set MBPC is a strategy to control selected state variables by a real-time optimization of the future switch states. For the optimization the future state variables need to be calculated online for all possible future switch state sequences. Three steps can be defined: estimation, prediction and optimization step.

Estimation

At update instant k, the optimal switch state obtained in the previous update period is applied and the measurements of measurable state variables (e.g. phase currents i_x^k with $x = a, b, c$) are obtained (throughout this paper superscript k denotes the kth update period). At the end of this update period the state variables will have changed due to the inverter switch state applied. The values of the state variables at $k+1$ have to be known to correctly predict the future state variables. To this end the state variables are calculated at the end of the current update period in the estimation step using the switch state which is currently applied. This change is estimated based on the measurements, the applied switch state S_{ix}^k and a system model. The estimation step is very important to start the following steps with correct values. However, although several types of system models can be used, it is not a computationally demanding step. This estimation step is needed to deal with the calculation time delay in practical implementations as discussed in [1]. The authors of [1] explicitly use the name two-step-ahead prediction for a method using an estimation step and a single prediction step. In [7] the estimation step is called the initial state projection.

Prediction

The next step, the prediction step, covers future update periods where all possible future switch state sequences are considered. The number of update periods considered in the prediction step is denoted by N, the prediction horizon. From k+1 on, the controller can use any possible output during each update period to bring the controlled variables closer to their desired values. The controller thus predicts the outcome of all possible switch states over the entire prediction horizon, from $k+1$ to $k+1+N$, based on the estimations at $k+1$.

In the prediction step a system model is needed. Often this is the same model as in the estimation step is used, although this is not necessary. As the prediction system model has to be evaluated for all possible switch state sequences, the computational complexity of the model and its accuracy will directly determine the overall computational burden and control quality.

Optimization

In the optimization step the most appropriate switch state sequence is selected, of which the first switch state is applied at the next update instant. Once the trajectories of the state variables for all possible control sequences have been calculated, the optimal sequence can be selected by evaluation of a cost function g^k. The sequence resulting in the minimal cost is then selected and the first switch state is applied by the controller at time instant $k+1$. At this time $k+1$, the algorithm is started again, resulting in a so-called receding horizon.

The cost function assigns a cost to a deviation of the state variables from their desired values. Typically a cost function with quadratic error terms is used, stemming from the mathematical treatment for offline MBPC [4], [8], [9]. For online MBPC a large number of different cost functions can be devised. Besides the freedom in choosing the form of the cost terms, there is also a choice to be made for the relative importance of each cost term. Typically each cost term expressing the error of a certain controlled variable is assigned a weight factor, the relative ratios of these weight factors allow to focus the control effort on specific controlled variables.

Similarly a weight factor can be introduced to express the relative importance of the error in update period $k+i+1$ for $i \in [1, N]$. This is not elaborated here (for a discussion, see [6]), as this papers only discusses MBPC implementations where $N=1$, i.e. a prediction horizon of one update period.

III. Design choices for the optimization step: cost function design

As mentioned in the previous section, there are different options for the design of the cost function used in the optimization step. This is illustrated here with the specific application of MBPC current and capacitor voltage control of 3-level inverter. Firstly the application is briefly introduced, after which the results obtained with a quadratic-term cost function are discussed. These results are compared with those obtained with a cost function that uses absolute values of the errors instead of the squared values. Finally a cost function formulation that allows a certain deviation without cost is discussed. All three types of cost functions are implemented in an FPGA setup and experimentally verified.

A. Application: current and capacitor voltage control of 3-level inverter

The topology of a three-phase, three-level flying capacitor converter (FCC) is depicted in figure 1. It uses 2 pairs of complementary controlled switches, $(S_{1x}, \overline{S_{1x}})$ and $(S_{2x}, \overline{S_{2x}})$ per phase x, where $x = a, b, c$. These switches make it possible to connect the flying capacitors C_{1x} in series with the load (an RL series connection). The series connection of the flying capacitor produces an intermediate output voltage. Because the flying capacitor is connected in series with the load, the voltage of the capacitor changes as the load current flows through the capacitor. The voltage of the flying capacitor C_1 in a three-level converter should always be kept at $V_{\mathrm{DC}}/2$. This choice provides optimal voltage rating of the switches as this only has to be $V_{\mathrm{DC}}/2$. Each phase has 4 (2^2) switch states of

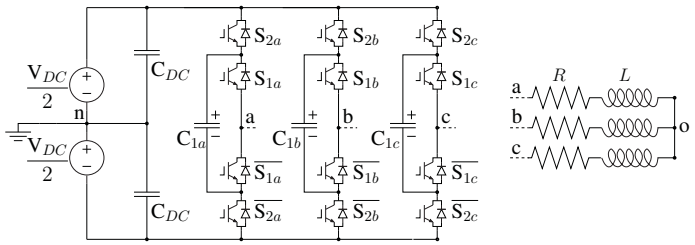


Figure 1: 3-level flying capacitor converter topology.

which 2 switch states produce the intermediate output voltage. This makes it possible to perform a correction of the capacitor voltage for every possible current direction and thus control the capacitor voltage. The two control objectives for the model-based predictive control (MBPC) with multilevel converters are the tracking of the reference currents i_x^k with $x = a, b, c$ and the balancing of the flying capacitor voltages v_{cx}^k. To this end the inputs for the MBPC algorithm are the reference values and the measurements of phase currents i_x^k and the flying capacitor voltages v_{cx}^k. The output of the algorithm is one of the 64 ($(2^2)^3$) possible switch states of the inverter. A more complete discussion of the FS-MBPC for a 3-level flying capacitor inverter can be found in [6]. In this paper we only consider a prediction horizon of 1 update period. The equations 1-5 form the estimation model (with $i = 0$) and prediction model (with $i = 1$). The system parameters of the studied system are given in table I.

$$v_{xn}^{k+i+1} = (S_{2x}^{k+i} - \frac{1}{2})V_{\text{DC}} - (S_{2x}^{k+i} - S_{1x}^{k+i})v_{cx}^{k+i} \quad (1)$$

$$v_{xo}^{k+i+1} = v_{xn}^{k+i+1} + v_{on}^{k+i+1} \quad (2)$$

$$v_{on}^{k+i+1} = \frac{v_{an}^{k+i+1} + v_{bn}^{k+i+1} + v_{cn}^{k+i+1}}{3} \quad (3)$$

$$i_x^{k+i+1} = e^{-\frac{1}{f_u}\frac{R}{L}} i_x^{k+i} + \frac{1 - e^{-\frac{1}{f_u}\frac{R}{L}}}{R} v_{xo}^{k+i+1} \quad (4)$$

$$v_{cx}^{k+i+1} = v_{cx}^{k+i} + \frac{1}{2Cf_u}(i_x^{k+i} + i_x^{k+i+1})(S_{2x}^{k+i} - S_{1x}^{k+i}) \quad (5)$$

V_{DC}	120V	L	14.5 mH
f_u	20 kHz	R	4.5 Ω
C	110 μF		

Table I: System parameters

V_{base}	400 V
I_{base}	10 A
ω_{base}	100π rad

Table II: Per unit base values

B. Tracking control with quadratic error term for the flying capacitor voltages

In the online optimization a cost function has to be calculated for all of the 64 possible switch states to be applied from $k+1$ to $k+2$. This can be done by using equation 6 as a phase

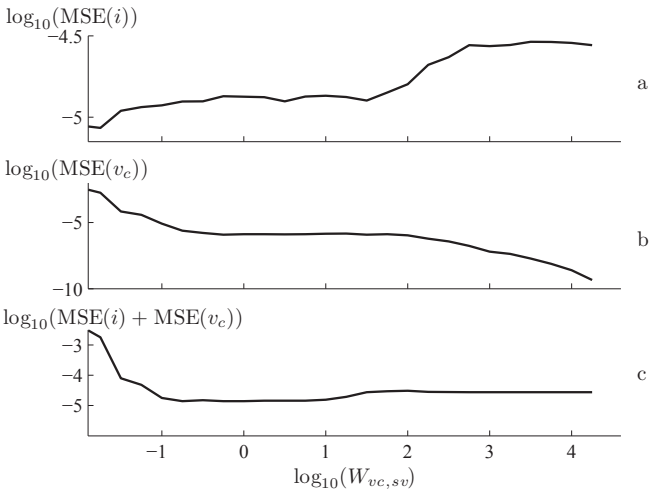


Figure 2: The MSE values for ouput current (a), capacitor voltage (b) and total error (c) as a function of the capacitor voltage weight factor $W_{vc,sv}$, simulations

cost function, where the error between the reference values and actual values is squared to give a quadratic error term. Each error term is weighted in the sum, the relative importance of the error of the capacitor voltage is expressed by $W_{vc,sv}$. The weight factor $W_{vc,sv}$ is a dimensionless parameter as we will use per unit values for all currents and voltages (both in the simulations and the FPGA implementation, base values are given in table II).

$$g_x^k = (i_{x,r}^{k+2} - i_x^{k+2})^2 + W_{vc,sv}(v_{c,r}^{k+2} - v_{cx}^{k+2})^2 \quad (6)$$

The best switching action is found by minimising the total cost function g^k, which is the sum of all g_x^k. The results obtained with this cost function formulation for a reference current of 2A, 50Hz are shown in figure 2, the top figure shows the MSE value of the current as a function of $W_{vc,sv}$, the middle figure the MSE value of the capacitor voltage and the bottom figure shows the total MSE. The MSE is a good measure to evaluate the control quality as it expresses the average squared deviation from the reference value. It is defined by:

$$\text{MSE} = \frac{\sum_{k=1}^{m}(x_{ref}^k - x^k)^2}{m} \quad (7)$$

where the x_{ref}^k and x^k variables are dimensionless in the per unit system. For very low $W_{vc,sv}$ the current control has priority and very good current control (low MSE) is obtained at a cost of higher capacitor voltage deviations. The best operating area is for an intermediate weight factor where both current and capacitor voltage are well controlled. For high $W_{vc,sv}$ the current control is neglected to achieve good capacitor voltage control. The existence of a good range for the weight factor is important as the selection of the weight factor is still an unresolved issue in the literature. This analysis is performed more extensively in [6] where the MSE-method is discussed to analyse several other design choices not treated in this paper. These include the model simplification proposed in [2] and larger prediction horizons $N > 1$. Furthermore the

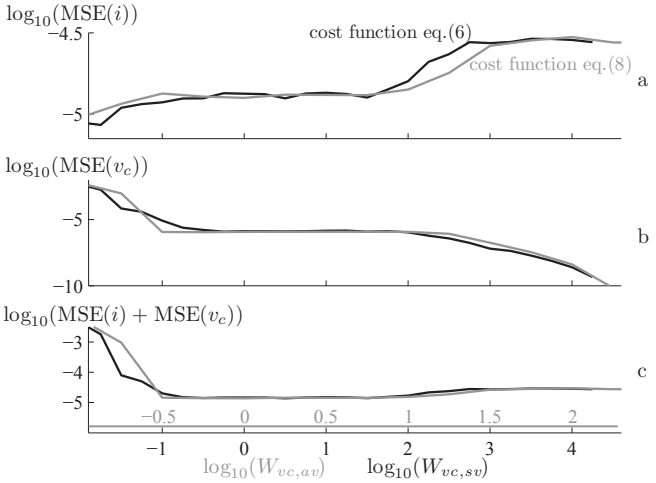


Figure 3: Comparison of the MSE values for ouput current (top), capacitor voltage (middle) and total error (bottom) for the quadratic and absolute value cost functions where $W_{vc,sv} = W_{vc,av}^2$, simulations

sensitivity of the good range of weight factors to the current amplitude is shown to be very low.

C. Tracking control with absolute value error term for the flying capacitor voltages

The quadratic cost terms as discussed before clearly allow good control quality. However, the calculation of these terms demands considerable FPGA resources. With an update frequency of 20kHz and an acceptable control, the predicted errors of the controlled variables at $k+2$ are small. Squaring these small per-unit values in a fixed-point format demands large data types to retain enough precision for the optimization, this results in a higher usage of FPGA slices and embedded multipliers than in the case of a cost function using absolute values of the error terms:

$$g_x^k = \left|i_{x,r}^{k+2} - i_x^{k+2}\right| + W_{vc,av}\left|v_{c,r}^{k+2} - v_{cx}^{k+2}\right| \tag{8}$$

In [8] different types of norms for the cost function terms are discussed. However, general observations on the cost function design for common MBPC schemes as Generalized Predictive Control (GPC) do not always hold for FS-MBPC with $N = 1$ as steady state is never achieved. In [1] different cost function definitions are presented and it is stated (but not shown) that with small update periods the influence is not important. In order to consider using absolute value cost functions to reduce the FPGA utilization, the control quality and the range of good weight factors have to be investigated. In figure 3 the control quality is again studied by the MSE-values of the controlled variables and compares the obtained quality of a quadratic and absolute-value cost function. On the horizontal axis $W_{vc,sv}$ (black axis and curve) and $W_{vc,av}$ (grey axis and curve) are shown. Clearly the control quality is very comparable and in the area of good simultaneous control it is identical if a scaling $W_{vc,av}^2 = W_{vc,sv}$ is chosen. This can be understood easily by using the notations $\Delta i = i_r^{k+2} - i_x^{k+2}$ and $\Delta v_c = v_{c,r}^{k+2} - v_{cx}^{k+2}$

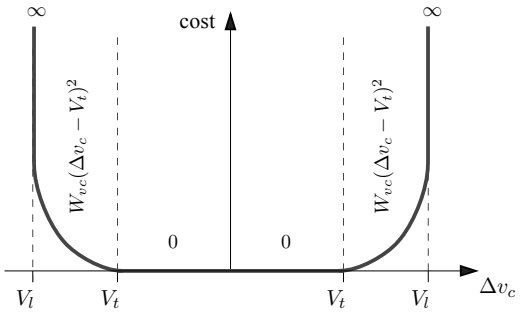


Figure 4: Graphical illustration of the cost for the capacitor voltage deviation

to rewrite the equations 6-8

$$|\Delta i| + W_{vc,av}|\Delta v_c| \tag{9}$$

$$\Delta i^2 + W_{vc,av}^2 \Delta v_c^2 + 2\Delta i \Delta v_c W_{vc,av} \tag{10}$$

As the only difference is the term containing $\Delta i \Delta v_c$, which is small in the area of good simultaneous control, the expression

$$W_{vc,av}^2 = W_{vc,sv} \tag{11}$$

holds in the area of interest. This means that the weight factor selection for both cost functions is equally straightforward. Furthermore it shows that the absolute-value based cost function can be used.

D. Allowed deviation control for the flying capacitor voltages

Two cost function formulations have been studied that achieve a very good tracking of the capacitor voltage. However, perfect tracking of the capacitor voltages often is not required and sometimes undesirable (for high weight factors the controller neglects the current control). In most cases the capacitor voltage only needs to be in the vicinity of the reference $V_{\mathrm{DC}}/2$ to respect the voltage rating of the switches, but a certain deviation can be allowed. As such the following cost function is proposed

$$g_x^k = \begin{cases} \Delta i^2 & 0 \le |\Delta v_c| \le V_t \\ \Delta i^2 + W_{vc}(|\Delta v_c| - V_t)^2 & V_t \le |\Delta v_c| \le V_l \\ \infty & |\Delta v_c| > V_l \end{cases} \tag{12}$$

A graphical interpretation of the capacitor voltage cost term is shown in figure 4. Deviations of the capacitor voltage of less than V_t remain within the tolerance band and have no cost. For deviations that are larger, but remain below the safety voltage limit V_l, a quadratic cost term is calculated that expresses the deviation from the allowed value $|\Delta v_c| - V_t$. If the predicted capacitor voltage deviation is larger than V_l an infinite cost insures that the inverter will not operate in the unsafe area.

In figure 5 the proper operation of the proposed cost function is shown for a fixed threshold $V_t = 3.5V$ and $V_l = 7.5V$ with three values of the weight factor. Clearly the capacitor voltage approaches the safety limit only for low weight factors but never crosses the limit. For the low cost the capacitor voltage deviation often is between V_t and V_l. For high W_{vc} even stays within the tolerance band and never approaches the safety limit.

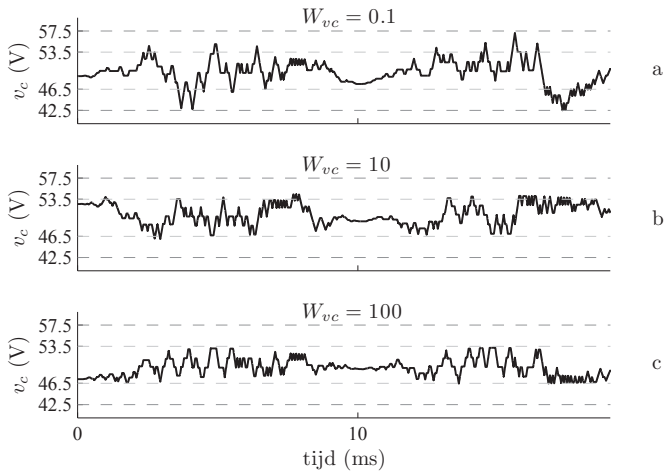


Figure 5: Simulated capacitor voltages for the cost function of equation 12 with a fixed tolerance band $V_t = 3.5V$ and limit $V_l = 7.5V$ with three values of the weight factor

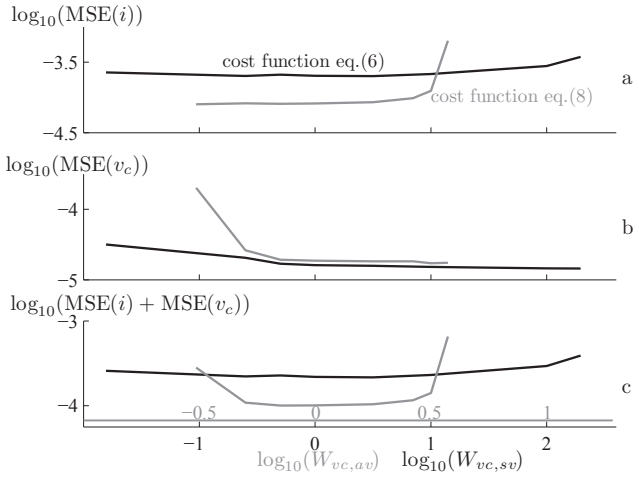


Figure 6: Measured MSE values for ouput current (top), capacitor voltage (middle) and total error (bottom) for the quadratic and absolute value cost functions where $W_{vc,sv} = W_{vc,av}^2$

E. Experimental results

The experimental setup is a 3-level flying capacitor converter constructed from in-house, half-bridge power electronic building blocks (PEBBs). The FC converter is controlled with an Xilinx VirtexII-Pro FPGA (XUPV2P-30), clocked at 100 MHz and an update period of 20 kHz. More details of this setup can be found in [6]. The three different options for the cost function presented here, were implemented in the FPGA. In figure 6 the comparison between the quadratic and absolute-value cost functions is made. The results validate the conclusion that both cost functions perform well in the area of good weight factors. In the experiments the absolute-value cost function performs better than the quadratic cost function.

In figure 7 the measured capacitor voltages validate the proper operation with the cost function defined by equation 12. The results are almost the same as those in simulation, but due to measurement noise and parameter mismatch the

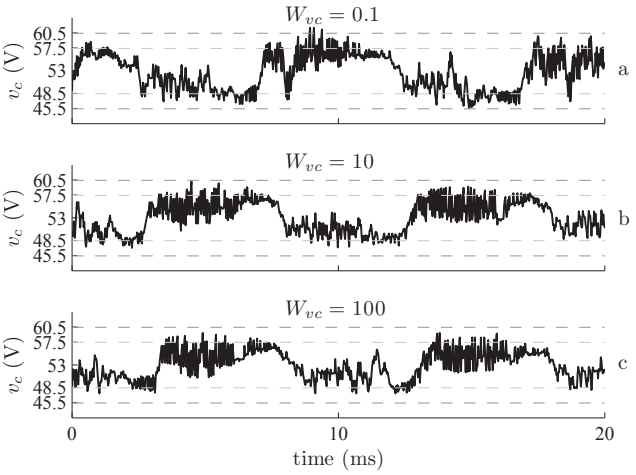


Figure 7: Measured capacitor voltage for tolerance band $V_t = 4.5$V and limit $V_l = 7.5$V

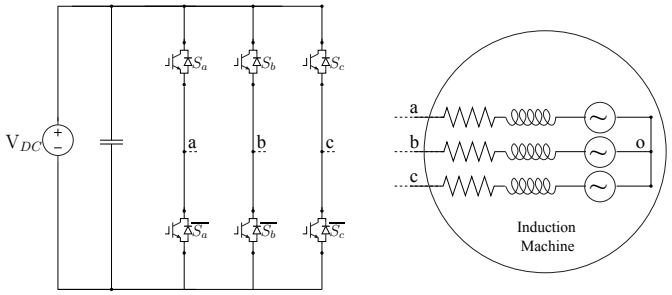


Figure 8: 2-level inverter topology and induction machine load.

voltage deviations are slightly larger. As a result the capacitor voltage can cross the safety limit for low W_{vc} and crosses more often out of the tolerance band. As a result W_{vc} should be selected higher than in the simulations. The result with $W_{vc} = 100$ is very satisfactory.

IV. Design choices for the prediction step: model type

In the previous section the possibilities for the formulation of the cost function in the optimization phase have been explored. The computational effort and the accuracy for the predictions on which the optimizations are performed, are also important aspects of an MBPC implementation. As these aspects are directly determined by the prediction model that is used, two possible prediction models are discussed here. In this section the application is a predictive torque control of induction machines. Again the simulation results are verified with experimental results.

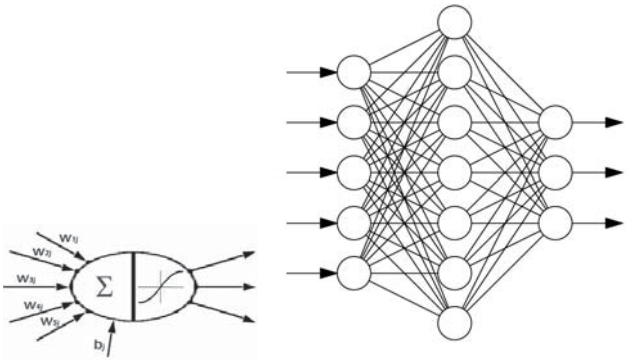
A. Application: flux and torque control for induction motor

Predictive flux and torque control is related to direct torque control (DTC) of induction motors. As in DTC the stator flux and motor torque are the controlled variables and the inverter switch state is selected directly to control these variables. Unlike in DTC the effect of each possible switch state is calculated and evaluated to obtain the optimal switch sequence.

more information on predictive torque and flux control can be found in the literature [3], [10]. In this application a 2-level inverter is used as shown in figure 8. For a 2-level inverter only 8 switch states can be applied.
The prediction horizon again is set to 1 update period but a two-step-ahead FS-MBPC is used (including estimation step for the time delay and prediction step). For the optimization step a quadratic cost function with stator flux error and torque error terms are used with an equal weighing of the per unit error terms. The parameters of the induction machine and inverter can be found in table III.

$R_s = 1.26\Omega$	$R_r = 0.75\Omega$
$L_m = 0.292$H	$L_r = L_s = 0.304$H
$P_{nom} = 4$ kW	$N_{nom} = 2900$ rpm
V_{DC}=560V	$f_u = \frac{1}{T_u} = 20$ kHz

Table III: System parameters of induction machine and inverter



a b

Figure 9: (a) Single neuron (b) Interconnected neurons form ANN

B. Analytical machine model

In most MBPC implementations of induction motor torque control an analytical machine model, using the machine parameters, is utilized as a prediction model. Several formulations of the induction machine model exist, depending on the chosen reference frame and state variables. In equation 13 a prediction model is given with the stator flux ($\Psi_{s\alpha}, \Psi_{s\beta}$) and stator current ($I_{s\alpha}, I_{s\beta}$) components in the stationary reference frame as state variables.

The stator currents ($I_{s\alpha}, I_{s\beta}$) and the motor speed ω are measured. As the pure integration of the back EMF to obtain the stator flux is unstable, the stator flux components $\Psi_{s\alpha}, \Psi_{s\beta}$ are estimated with a low-pass-filter (corrected for phase and amplitude errors). From the stator flux components the stator flux magnitude $|\hat{\Psi}_s|$ and torque T are calculated. The obtained stator flux and torque control are shown in figure 10, the reference for the stator flux is set to the nominal value $|\Psi_s|_{nom} = 1$Wb and the torque setpoint varies stepwise to random values between $+T_{nom}$ and $-T_{nom}$. In this way the entire operating region (both in torque and speed) of the motor is covered. In figure 10 clearly both controlled variables remain close to the references over the entire simulation range.

C. Neural network machine model

The prediction of the future stator flux and torque values is based on the prediction of the future stator flux and current components. The prediction of these components constitutes a mapping from the machine state and possible stator voltages to $\Psi_{s\alpha}, \Psi_{s\beta}, I_{s\alpha}, I_{s\beta}$. Back-propagation artificial neural networks (ANN) can be trained to perform such mappings, a thorough description of ANNs can be found in [11]. Here a back-propagation ANN is proposed as a prediction model for the calculation of the state variables $[\Psi_{s\alpha}, \Psi_{s\beta}, I_{s\alpha}, I_{s\beta}]$ instead of an analytical prediction model.
An ANN consists of interconnected neurons, as shown in figure 9b, in three layers: the input layer, hidden layer and output layer. Each neuron consists of the weighted sum of its inputs and a bias value, together with an activation function, as shown in figure 9a. The ANN is trained by the back propagation method and as such can model any machine. By performing an online back-propagation the ANN could be made adaptive to respond to changes in the controlled system. This offers some advantages over adaptive implementations of analytical

$$
\begin{cases}
\frac{\Delta\hat{\Psi}_{s\alpha}}{T_u} &= V_{s\alpha} - R_s I_{s\alpha} \\
\frac{\Delta\hat{\Psi}_{s\beta}}{T_u} &= V_{s\beta} - R_s I_{s\beta} \\
\frac{\Delta\hat{I}_{s\alpha}}{T_u} &= \frac{1}{\sigma L_s}(-R_r \frac{L_s}{L_r} I_{s\alpha} - \omega\sigma L_s I_{s\beta} + \frac{R_r}{L_r}\Psi_{s\alpha} + \omega\Psi_{s\beta} + \frac{\Delta\hat{\Psi}_{s\alpha}}{T_u}) \\
\frac{\Delta\hat{I}_{s\beta}}{T_u} &= \frac{1}{\sigma L_s}(-R_r \frac{L_s}{L_r} I_{s\beta} + \omega\sigma L_s I_{s\alpha} + \frac{R_r}{L_r}\Psi_{s\beta} - \omega\Psi_{s\alpha} + \frac{\Delta\hat{\Psi}_{s\beta}}{T_u}) \\
\\
\hat{\Psi}_{s\alpha}^{k+1} &= \Psi_{s\alpha}^{k} + \Delta\hat{\Psi}_{s\alpha} \\
\hat{\Psi}_{s\beta}^{k+1} &= \Psi_{s\beta}^{k} + \Delta\hat{\Psi}_{s\beta} \\
\hat{I}_{s\alpha}^{k+1} &= I_{s\alpha}^{k} + \Delta\hat{I}_{s\alpha} \\
\hat{I}_{s\beta}^{k+1} &= I_{s\beta}^{k} + \Delta\hat{I}_{s\beta} \\
\\
|\hat{\Psi}_s|^{k+1} &= \sqrt{\left(\hat{\Psi}_{s\alpha}^{k+1}\right)^2 + \left(\hat{\Psi}_{s\beta}^{k+1}\right)^2} \\
\hat{T}^{k+1} &= \frac{3}{2}N_p(\hat{\Psi}_{s\alpha}^{k+1}\hat{I}_{s\beta}^{k+1} - \hat{\Psi}_{s\beta}^{k+1}\hat{I}_{s\alpha}^{k+1})
\end{cases}
\tag{13}
$$

models as any system change is correctly correctly modelled by the ANN and not only those captured by the parameters. The plasticity of the network to model any machine and non-linearity is the most interesting feature of using ANNs as prediction models.

The ANNs used here have tan-sigmoid ($tanh$) activation functions for all neurons. The ANNs were constructed and trained using the Neural Network Toolbox from Matlab. The network has 7 inputs $[\Psi_{s\alpha}, \Psi_{s\beta}, I_{s\alpha}, I_{s\beta}, \omega, V_{s,\alpha}, V_{s,\beta}]$ and 1 output (one of the flux or current components). For the hidden layer different numbers of neurons were used. Using simulations the capability of the ANN to identify the induction machine was evaluated for all of these situations.

If a large number of neurons in the hidden layer is chosen the predictions of the ANN can be as accurate as the analytical prediction model, simulations have shown that 15 neurons in the hidden layer are more than sufficient for both the flux prediction ANN and the current prediction ANN. To reduce the computational effort a lower number can be selected. For 1 and 2 neurons in the hidden layer for the flux prediction ANN and the current prediction ANN respectively, acceptable predictions are obtained and control quality is not much reduced. The results obtained with these ANNs is shown in figure 11. Again the stator flux and torque remain close to the reference values. Some ondulation of the torque trajectory is noticeable, this results from the limited capabilities of the small (1 and 2 hidden neurons respectively) ANN to identify the machine.

D. Experimental results

Both prediction models were implemented on an experimental FPGA-based setup ($V_{DC} = 150$ V, $f_u = 12$ kHz, other parameters as in table III). For the $tanh$ activation functions a piece-wise linear approximation was used, the implemented ANN was trained in simulation. Figures 12 and 13 show the estimated stator flux and estimated torque for the experimental FS-MBPC with analytical and ANN prediction models respectively. The stator flux reference is set to the nominal value $|\Psi_s|_{nom} = 1$Wb and the torque setpoint is fixed to 3.5 Nm (26.5% of T_{nom}), the motor speed during the tests was 500 rpm. The results validate those obtained in simulations. The control quality with the ANN is lower than with the analytical model, the already mentioned ondulations are clearly visible. Better control is achieved if the number of hidden neurons would be increased.

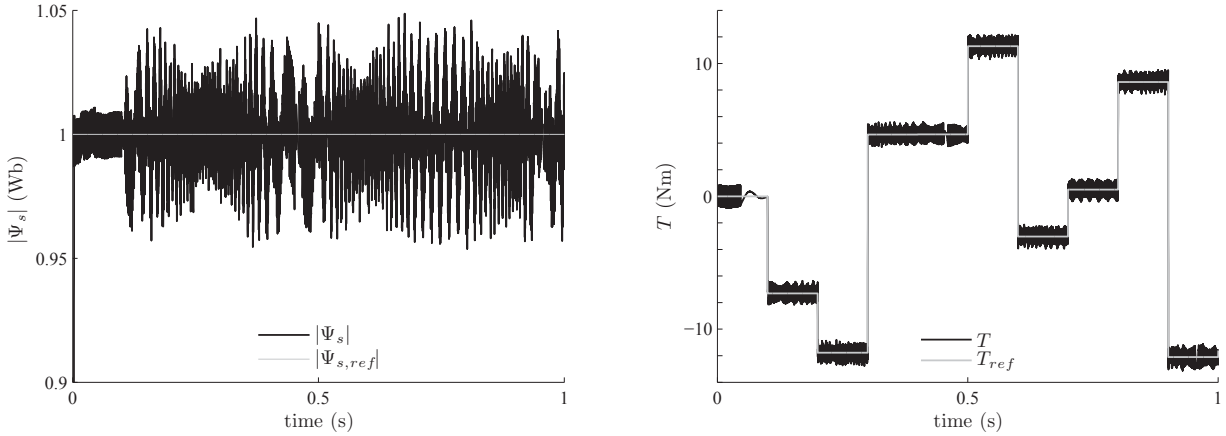


Figure 10: Predictive flux and torque control with an analytical prediction model (simulation)

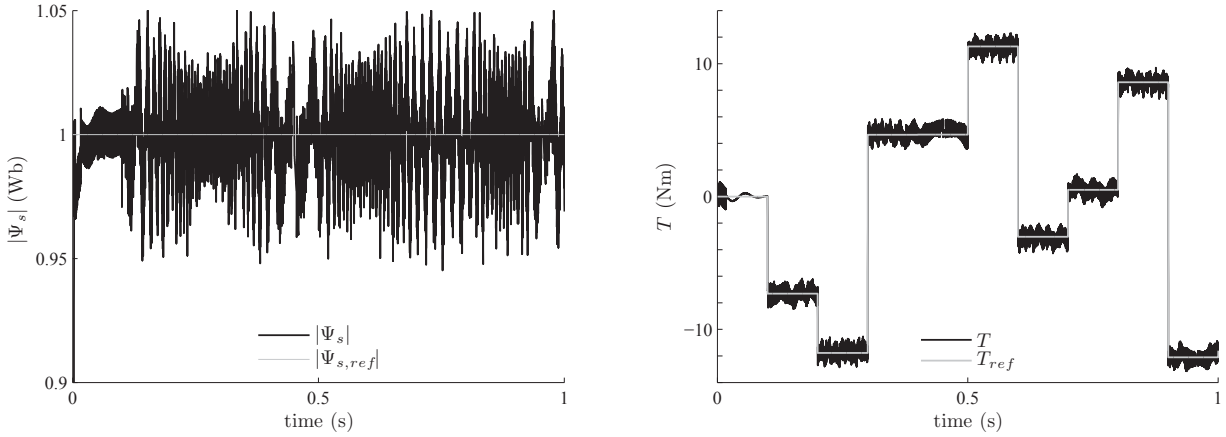


Figure 11: Predictive flux and torque control with an artificial neural network prediction model (1 and 2 neurons in the hidden layer for flux and current estimation respectively) (simulation)

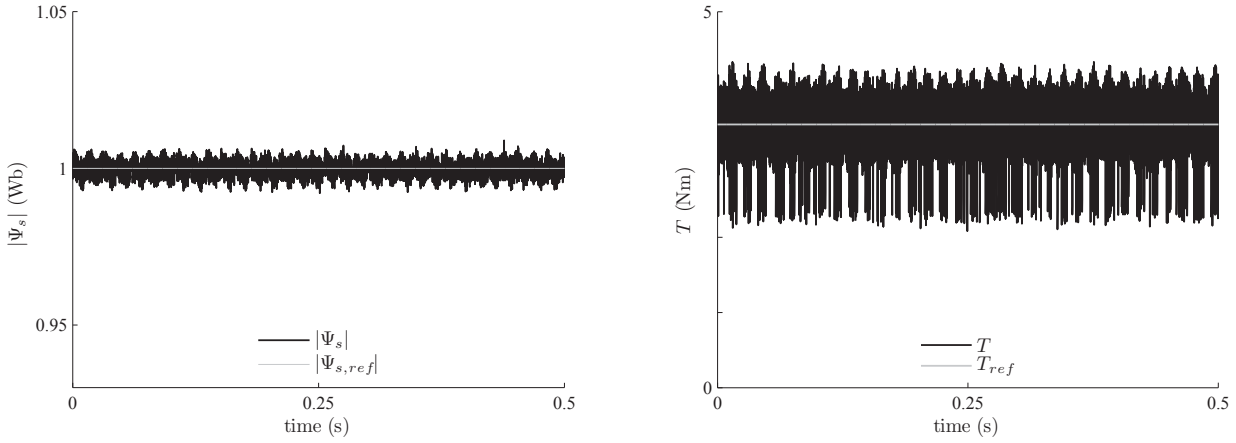


Figure 12: Predictive flux and torque control with an analytical prediction model (experimental)

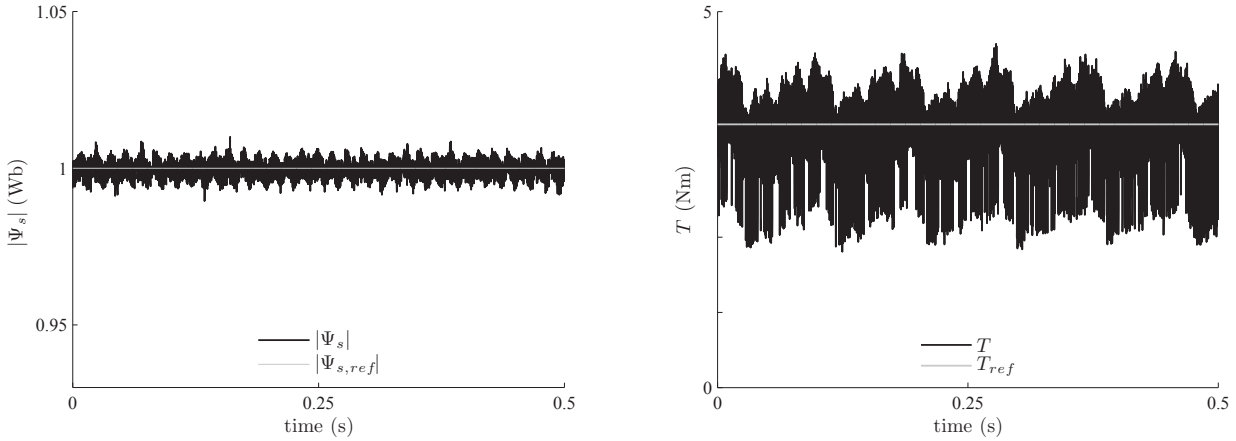


Figure 13: Predictive flux and torque control with an artificial neural network prediction model (1 and 2 neurons in the hidden layer for flux and current estimation respectively) (experimental)

V. Conclusions

In this paper several options for the formulation of the cost function in the optimization step and for the prediction model in the prediction step of FS-MBPC have been explored. This was done with two specific applications: current control of FC converters and torque control of induction motors. It has been shown that quadratic and absolute-value cost functions are interchangeable and that reducing the tracking term on the capacitor voltage of an FCC is a favorable change. Furthermore the feasibility to replace the analytical prediction model for torque control with an ANN is demonstrated.
The practical feasibility to implement all these versions of FS-MBPC in an FPGA has been shown.

Acknowledgement

This work was supported by the the Research Foundation Flanders (FWO) under Project G.0665.06 and by the Interuniversity Attraction Pole under Project (IUAP) P6/21. The work of T.J. Vyncke was supported by a Ph.D. fellowship from the Research Foundation Flanders (FWO).

References

[1] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, "Model predictive control-a simple and powerful method to control power converters," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1826–1838, Jun. 2009.

[2] P. Lezana, R. Aguilera, and D. E. Quevedo, "Model predictive control of an asymmetric flying capacitor converter," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1839–1846, Jun. 2009.

[3] T. Geyer, G. Papafotiou, and M. Morari, "Model predictive direct torque control – Part I: concept, algorithm, and analysis," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.

[4] A. Linder, R. Kanchan, R. Kennel, and P. Stolze, *Model-Based Predictive Control of Electric Drives*. Göttingen: Cuvillier Verlag, 2010.

[5] H. Miranda, P. Cortés, J. I. Yuz, and J. Rodríguez, "Predictive torque control of induction machines based on state-space models," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1916–1924, Jun. 2009.

[6] T. J. Vyncke, S. Thielemans, M. Jacxsens, and J. A. A. Melkebeek, "Analysis of design choices in model based predictive control of flying-capacitor inverters," *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, accepted for publication.

[7] G. Papafotiou, J. Kley, K. G. Papadopoulos, P. Bohren, and M. Morari, "Model predictive direct torque control – Part II: implementation and experimental evaluation," *IEEE Trans. Ind. Electron.*, vol. 56, Issue 6, pp. 1906–1915, Jun. 2009.

[8] E. F. Camacho and C. Bordons, *Model Predictive Control*. London: Springer-Verlag, 1998.

[9] Rossiter, *Model-Based Predicitive Control, a Practical Approach*. Boca Raton: CRC Press, 2003.

[10] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto, "Design and implementation of model predictive control for electrical motor drives," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1925–1936, Jun. 2009.

[11] R. Rojas, *Neural networks, a systematic introduction*. London: Springer-Verlag, 1996.