Predictive Direct Torque Control Algorithm for Induction Motors and its Digital Implementation

Mutschler, Peter Flach, Erich
Institute of Power Electronics and Control of Drives, Darmstadt University of Technology,
Landgraf-Georg-Str. 4, D-64283 Darmstadt, Germany
Phone +49-6151-16-2166 Fax +49-6151-16-2613

ABSTRACT – To achieve fast control action, direct control methods should be used. "Direct Mean Torque Control" (DMTC) combines the good dynamic performance of Direct Torque Control (DTC) with the advantages of inherently constant switching frequency and time equidistant control for implementation in a digital signal processor. Since DMTC is a predictive control algorithm, the model and its correction deserves special investigations. This paper proposes a steady-state Kalman filter which is well suited for fast computation.

INTRODUCTION

In the last years, methods to control *directly* torque and flux of induction motors gained increasing interest, since they promise an excellent dynamic performance. Several types of DTC evolved ([1], [2]). Time-continuous (analogue) implementations of the control algorithm have been realized first. For many reasons, a digital realization of direct torque control methods is desirable [3].

This paper deals with the Direct Mean Torque Control method and its digital implementation [5]. The method uses a sophisticated algorithm to predict the next two switching actions of the inverter. To do this, the state equations (2) of the machine are numerically integrated in a digital signal processor (DSP). This integration is done a small amount of time (e.g. 30...100µs) ahead of real time. This means, that the computed torque (6) is earlier available for the control algorithm than it will be produced by the real machine. The method uses the derivation dT/dt of (6) to determine the next switching actions in the inverter.

In contrast to [3], this method has the advantage of an inherently constant switching frequency, but a predictive machine model is required. To keep the states of the model as close as possible to the real states, a feedback, i. e. an observer, should be implemented. A simple design method is manual pole placement. This method leads to unsatisfactory results because of the unbalanced root locuses of induction motors. A Kalman filter seems to be better suited, but computation extensive. This paper presents a steady-state approach. Its real time implementation is as simple and fast as pole placement. Experimental results show the validity of the control scheme.

MODELING

The classical DTC scheme uses a simple machine model based on the measured stator currents and known stator voltages. It consists on the integration

$$\underline{\psi}_1 = \int (\underline{u}_1 - R_1 \cdot \underline{i}_1)\,dt\,. \tag{1}$$

To avoid a significant delay while determining switching state, the control scheme calls for the prediction of the system states. The currents used in (1) are not available from the future. Thus, a complete machine model in the stator frame of reference (α,β) with the fluxes as state variables, and the stator voltages as inputs is used according to:

$$\frac{d}{dt}\begin{bmatrix}\psi_{1\alpha}\\ \psi_{1\beta}\\ \psi_{2\alpha}\\ \psi_{2\beta}\end{bmatrix} = \begin{bmatrix} -\frac{R_1}{\sigma L_1} & 0 & \frac{R_1 L_h}{\sigma L_1 L_2} & 0 \\ 0 & -\frac{R_1}{\sigma L_1} & 0 & \frac{R_1 L_h}{\sigma L_1 L_2} \\ \frac{R_2 L_h}{\sigma L_1 L_2} & 0 & -\frac{R_2}{\sigma L_2} & -n_p\cdot\omega_m \\ 0 & \frac{R_2 L_h}{\sigma L_1 L_2} & n_p\cdot\omega_m & -\frac{R_2}{\sigma L_2} \end{bmatrix} \cdot \begin{bmatrix}\psi_{1\alpha}\\ \psi_{1\beta}\\ \psi_{2\alpha}\\ \psi_{2\beta}\end{bmatrix} + \begin{bmatrix}u_\alpha\\ u_\beta\\ 0\\ 0\end{bmatrix} \tag{2}$$

Equation (2) is numerically integrated using Runge-Kutta second order integration. The stator voltages as only inputs in (2) are predicted quite easily from the nearly constant dc-link voltage and the desired switching state. Measured currents are taken into account afterwards to correct the estimated states. For the elaboration of model correction, thus the observer, eq. (2) has already the commonly used structure:

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}(\omega_m)\cdot\tilde{\mathbf{x}} + \mathbf{u} \tag{3}$$

The measured outputs are the currents of the machine:

$$\begin{bmatrix} i_{1\alpha} \\ i_{1\beta} \end{bmatrix} = \frac{1}{\sigma L_1 L_2}\begin{bmatrix} L_2 & 0 & -L_h & 0 \\ 0 & L_2 & 0 & -L_h \end{bmatrix}\cdot\left[\psi_{1\alpha}\ \psi_{1\beta}\ \psi_{2\alpha}\ \psi_{2\beta}\right]^T \tag{4}$$

with $\sigma = 1 - L_h^2/(L_1\cdot L_2)$. Similarly to (3), the outputs from the model can be expressed as:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{C}}\cdot\tilde{\mathbf{x}} \tag{5}$$

Using this model, the torque is given by:

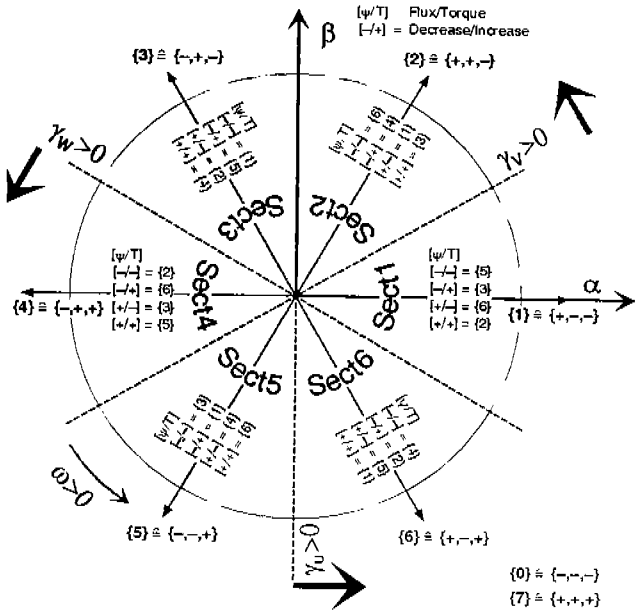

Fig. 1: Sectors , Voltage and Zero Vectors

$$T = \frac{3}{2} n_p \frac{L_h}{(L_1 L_2 - L_h^2)} (\psi_{1\beta} \cdot \psi_{2\alpha} - \psi_{1\alpha} \cdot \psi_{2\beta}) \qquad (6)$$

Likewise to [2], we determine the sector of the flux by comparing its components $\psi_{1\alpha}$, $\psi_{1\beta}$ with three digital comparators γ_u, γ_v, γ_w. The comparator outputs allow to classify the voltage vectors (VV) of the converter to flux and torque increasing and decreasing vectors, illustrated in fig. 1. The VVm±0 according to sector m (i. e. {1} is according to Sect1) is to be used as a mainly flux increasing VV. The VVm+1 is flux and torque increasing, VVm+2 decreases the flux and increases the torque. Similarly, VVm−1,2 decrease the torque.

CONTROL STRATEGY

The main application area of [1] are high power motors with a considerably large leakage inductance. Due to this, currents and torque change relatively slow. This eases the digital implementation of the control algorithm using a DSP. With a large leakage inductance in the motor, a quasi-analogue implementation of direct torque control is straight forward.

On the other hand, up to now the main application area of [2] are medium and low power drives, having smaller leakage inductance. Especially servo drives show a small leakage inductance in order to achieve high dynamics. Thus, currents and torque can change very quickly. To keep the torque within the desired hysteresis band, the switching states of the inverter must be changed in short intervals. This means, for a quasi-analogue implementation, a very high repetition rate of the control algorithm is necessary. For a servo drive, the torque may travel through the hysteresis band in less than 5µs. In this case, a quasi-analogue implementation is not satisfactory because the computational effort would be too high. Using a predictive algorithm, fast switching sequences can be calculated in advance. The computation time may be equally distributed within two successive switching intervals of the inverter.

DMTC inherently uses constant switching frequency and determines two switching events in advance for a fixed cycle t_S. The aim is to place the switching events directly in a way that the mean torque over the cycle is equal to the desired value. In most cases, alternate switching of a voltage vector (VV) and a zero voltage vector (ZV) satisfies the demand. Fig. 2 shows the control structure of DMTC.

Fig. 3 shows a typical cycle of operation. Applying a VV first, the torque increases at the beginning. Then, applying a ZV, the torque decreases. In [4] the timing of the switching events was determined by equalizing the different hatched areas in fig. 3. In that case steady state will not be reached at the end of the cycle, if $T(t_n) \neq T(t_{n+1})$ and $T(t_{n+1}) \neq T(t_{n+2})$. In this paper we mainly use another algorithm [5], which reaches steady state at the end of the cycle.

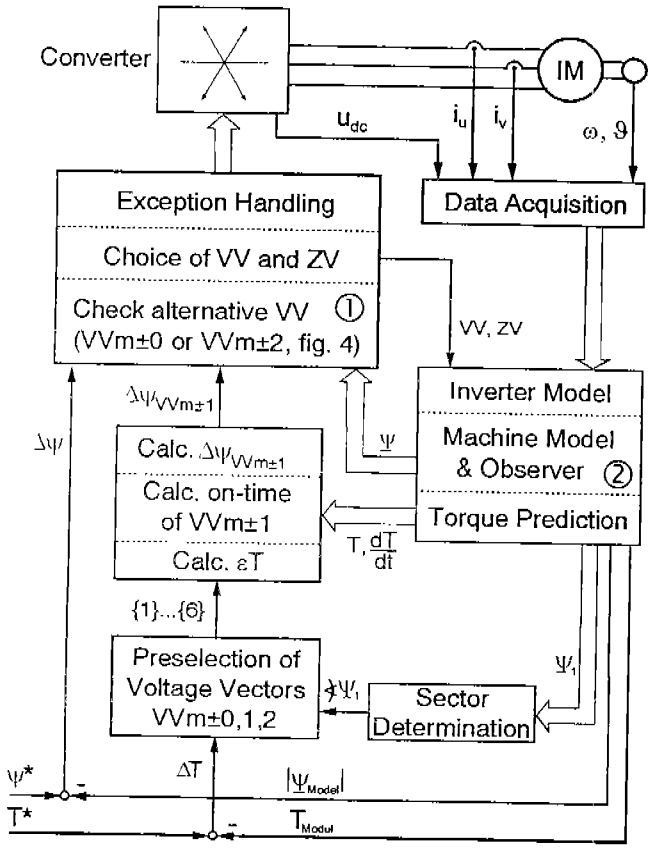

Fig. 2: Control structure of DMTC

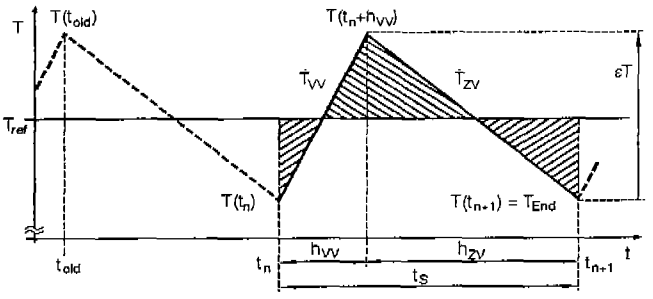

Figure 3: Typical operation cycle of DMTC

TORQUE CONTROL

In steady state, the torque $T(t_n)$ at the beginning of a cycle should be equal to its value $T(t_{n+1})$ at the end. Instead of simply equalizing the different hatched areas for the cycle n, it should also be possible to switch the VV to reach directly the value of $T(t_{n+1})$ according to steady state, thus

$$T(t_{n+1}) = T_{End} = T_{ref} - \tfrac{1}{2} \cdot \varepsilon T. \qquad (7)$$

where εT is the "virtual hysteresis width". Sustaining flux and torque as well, VVm±1 is normally the most taken VV. Hence we assume εT as the "virtual hysteresis width" while using VVm±1. εT can be expressed as:

$$\varepsilon T = \dot{T}_{VV} \cdot h_{VV} = -\dot{T}_{ZV} \cdot h_{ZV} = -\dot{T}_{ZV} \cdot (t_S - h_{VV}) \qquad (8)$$

$\dot{T}_{VV}$ can be calculated by deriving (6). Since the state of the machine at t_n+h_{VV} is not known yet, it is the easiest way to get $\dot{T}_{ZV}$ from the last cycle:

$$\dot{T}_{ZV} \cong \frac{T(t_n) - T(t_{old})}{t_n - t_{old}} \qquad (9)$$

We eliminate h_{VV} in (8) to get:

$$\varepsilon T = -\frac{\dot{T}_{VV} \cdot \dot{T}_{ZV}}{\dot{T}_{VV} - \dot{T}_{ZV}} \cdot t_S \qquad (10)$$

With $\dot{T}_{VV} = \dot{T}_{VV1}$, this equation is used in (7) to determine the instantaneous torque to be reached at the end of the cycle. To avoid torque offsets, εT must be limited to $\varepsilon T \geq \dot{T}_{VV1} \cdot h_{min}$, where h_{min} is the shortest switching interval of the inverter.

Based on fig. 3, a preselection of the VV is done. If the aspired value of $T(t_{n+1})$ (i.e. $T_{ref} - \tfrac{1}{2} \cdot \varepsilon T$) is greater than the value which would be reached when applying the ZV all the cycle time (i.e. $T(t_n) + \dot{T}_{ZV} \cdot t_S$), then VVm+0,1,2 are preselected. Thus, the counter clockwise rotation (i.e. VVm+0,1,2) is chosen if

$$T_{ref} - \tfrac{1}{2} \cdot \varepsilon T > T(t_n) + \dot{T}_{ZV} \cdot t_S. \qquad (11)$$

Otherwise, we use VVm-0,1,2.

For given $\dot{T}_{VV} = \dot{T}_{VV0,1,2}$ of VVm±0,1,2, we search its on-time h_{VV} to attain exactly T_{End}. The torque $T(t_{n+1})$ at the end of the cycle can be expressed as:

$$T(t_{n+1}) = T(t_n) + \dot{T}_{VV} \cdot h_{VV} + \dot{T}_{ZV} \cdot (t_S - h_{VV}) \qquad (12)$$

Solving this equation for h_{VV} leads with (7) finally to:

$$h_{VV} = \frac{T_{ref} - T(t_n) - \tfrac{1}{2} \cdot \varepsilon T - t_S \cdot \dot{T}_{ZV}}{\dot{T}_{VV} - \dot{T}_{ZV}} \qquad (13)$$

Ordinarily, a VV should only be applied, if the result of (13) is within $h_{min} \leq h_{VV} \leq t_s - h_{min}$. In this case, the torque will stay in the "virtual hysteresis band". Since εT has been calculated for $\dot{T}_{VV1}$, VVm±1 ($\Rightarrow h_{VV1}$) satisfies the demand. For lower angular speed of the flux vector, VVm±2 is well suited, too, whereas VVm±0 has to be checked out. Depending on the state of the machine, VVm±0 may increase, or even decrease the torque.

FLUX CONTROL

Since high dynamic torque control is our main goal, the flux controller should not interfere with it. Similarly to [2], the stator flux is kept as close as possible to a circular trajectory. After the preselection of the VVm±0,1,2, the flux controller uses the flux propagation and some supplementary rules to choose the appropriate VV. The length of the flux vector and its derivative are given by

$$\psi_1 = \left|\underline{\psi}_1\right| = \sqrt{\psi_{1\alpha}^2 + \psi_{1\beta}^2} \qquad (14)$$

$$\dot{\psi}_1 = (\dot{\psi}_{1\alpha} \cdot \psi_{1\alpha} + \dot{\psi}_{1\beta} \cdot \psi_{1\beta}) \,/\, \psi_1 \qquad (15)$$

For the VV, $\dot{\psi}_{1,VV}$ results from (2), whereas for the ZV, similar to (9), $\dot{\psi}_{1,ZV}$ can simply be taken from the last cycle:

$$\dot{\psi}_{1,ZV} = (\psi(t_{old}) - \psi(t_n)) \,/\, (t_n - t_{old}) \qquad (16)$$

Starting from (13) of the torque controller, and $\dot{\psi}_1$, we can predict the evolution of the flux vector over the next cycle for each VV. At higher speed, the flux controller chooses between VVm±1 and VVm±2 to influence the flux. A common problem of DTC methods is the flux maintenance at lower speed, because the on-times of the VV decrease. First the controller checks if the VVm±0 is able to control both flux and torque. Its on-time h_{VV0} is limited to $h_{VV0,max}$ by the maximum flux increase which seems to be tolerable (using $\Delta\psi$ from (19a)):

$$h_{VV0,max} = C_1 \cdot \frac{\Delta\psi + C_2 \cdot \psi_{ref}}{\dot{\psi}_{1,VV0}} \qquad (17)$$

Empirical tests showed acceptable results with $C_1 = 1.5$ and $C_2 = 2\,\%$. If (13) leads to

$$\tfrac{1}{2} \cdot h_{min} \leq h_{VV0} \leq h_{VV0,max}, \qquad (18)$$

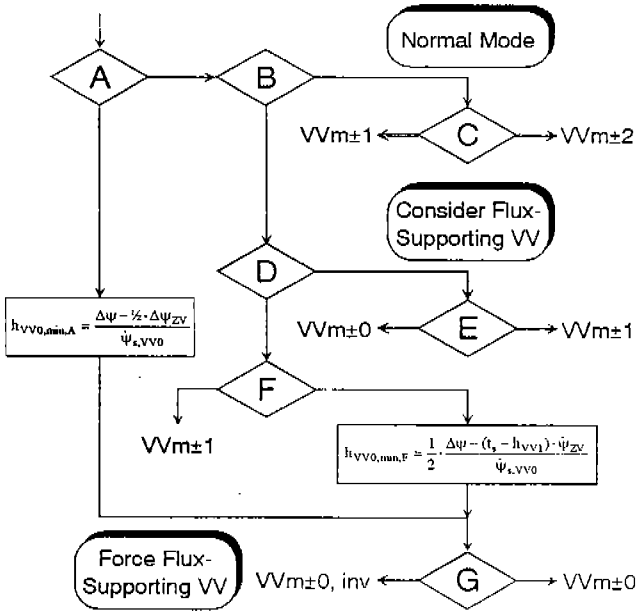

Fig. 4: Flow chart of flux controller

then VVm±0 is adequate to control the torque. If this is not the case, and the flux cannot be preserved over several cycles, it is necessary to apply a flux-supporting VV (i. e. VVm±0), although the torque control may be affected. If the flux-supporting VV is switched too late, flux decrease has become significant, hence it requires a longer on-time. The idea is to detect the necessity of flux-supporting VV as soon as possible to diminish its influence over the torque. A basic intention was to find expressions which check out relationships between the influences of the switching states over the flux. Avoiding to compare with absolute quantities has the advantage of uncomplicated parameterization for different machines. The algorithm consists of a set of decision rules. We define its inputs as:

$$\Delta\psi = \Delta\psi(t_n) = \psi_{ref}(t_n) - \psi(t_n) \tag{19a}$$

$$\Delta\psi_{VV0} = \Delta\psi - \tfrac{1}{2} \cdot h_{VV0} \cdot \dot{\psi}_{1,VV0} \tag{19b}$$

$$\Delta\psi_{VV1} = \Delta\psi - \tfrac{1}{2} \cdot h_{VV1} \cdot \dot{\psi}_{1,VV1} \tag{19c}$$

$$\Delta\psi_{VV2} = \Delta\psi - h_{VV2} \cdot \dot{\psi}_{1,VV2} - (t_S - h_{VV2}) \cdot \dot{\psi}_{1,ZV} \tag{19d}$$

$$\Delta\psi_{ZV} = \psi(t_n) - \psi(t_{old}) \tag{19e}$$

Fig. 4 shows the flow chart of the flux controller, which is performed in fig. 2, block ①. There are three principle operation modes to enter:

- **"Normal Mode"** where to chose between flux increasing VVm±1 or flux decreasing VVm±2.
- **"Consider Flux-Supporting VV"** where VVm±0 may only be switched without torque disruption, VVm±1 otherwise.
- **"Force Flux-Supporting VV"** where VVm±0 must be switched, although $T_{End,VV0} \neq T_{ref} - \tfrac{1}{2} \cdot \varepsilon T$.

In "Flux-Supporting VV" mode, decision G, if $\dot{T}_{VV0}$ decreases the torque error (i. e. $T_{ref} - T(t_{n+1}) - \tfrac{1}{2} \cdot \varepsilon T$), VVm±0 is switched for $h_{VV0,max}$ to keep the torque as good as possible. Otherwise, h_{VV0} is reduced. The resulting torque error over such a cycle can be taken into account by adding it to T_{ref} just for the next cycle.

The decision rules A-G in fig. 4 are mainly
(→ condition is satisfied, ↛ otherwise):

A: $\Delta\psi \geq C_3 \cdot \psi_{ref}$ (i. e. $C_3 = 3\%$)
AND $\psi_1(t_n) - \psi_1(t_{n-1}) \leq 0$ → G ↛ B

B: $\Delta\psi_{VV1} \leq 0$ → C ↛ D

C: $|\Delta\psi_{VV1}| \leq |\Delta\psi_{VV2}|$ → VVm±1 ↛ VVm±2

D: $h_{min} \leq h_{VV0} \leq h_{VV0,max}$ → E ↛ F

E: $|\Delta\psi_{VV0}| \leq |\Delta\psi_{VV1}|$ → VVm±0 ↛ VVm±1

F: $\Delta\psi > -0.75 \cdot t_S \cdot \dot{\psi}_{1,ZV}$
AND $h_{VV1} \cdot \dot{\psi}_{1,VV1} + (t_S - h_{VV1}) \cdot \dot{\psi}_{1,ZV} \leq 0$
→ G ↛ VVm±1

G: $\text{sign}(\dot{T}_{VV0}) = \text{sign}(\dot{T}_{VV1})$
→ VVm±0 first, then ZV, $h_{VV0} = h_{VV0,max}$
↛ VVm±0, inverted switching order, $h_{VV0} = h_{VV0,min}$

For the branches A → G, resp. F → G, $h_{VV0,min}$ is set to

$$h_{VV0,min,A} = \frac{\Delta\psi - \tfrac{1}{2} \cdot \Delta\psi_{ZV}}{\dot{\psi}_{1,VV0}}, \tag{20a}$$

$$h_{VV0,min,F} = C_4 \cdot \frac{\Delta\psi - (t_S - h_{VV1}) \cdot \dot{\psi}_{ZV}}{\dot{\psi}_{1,VV0}} \tag{20b}$$

Since branch condition A (fig. 4) checks for a somewhat critical flux decrease, (20a) results in longer on-times. Setting $C_4 = \tfrac{1}{2}$, (20b) assumes that, in the succeeding cycle, VVm±1 will provide a certain flux increase, too.

Some additional constraints are generally included for the branch conditions to avoid suspicious switching patterns:

- When entering "Force Flux-Supporting VV" mode, a flag is set. After this cycle, we never use flux decreasing VVm±2.
- "Force Flux-Supporting VV" mode is never entered after a cycle with VVm±2.

MACHINE OBSERVER

Fig. 2 contains a block ② "Machine Model & Observer". Its purpose is to deliver (predicted) values of the fluxes, the torque and its derivatives. A Luenberger observer is used, which consists of the machine's equations (2), (3), (5), and the feedback matrix $\mathbf{K}_b(\omega_m)$ as shown in fig. 5.

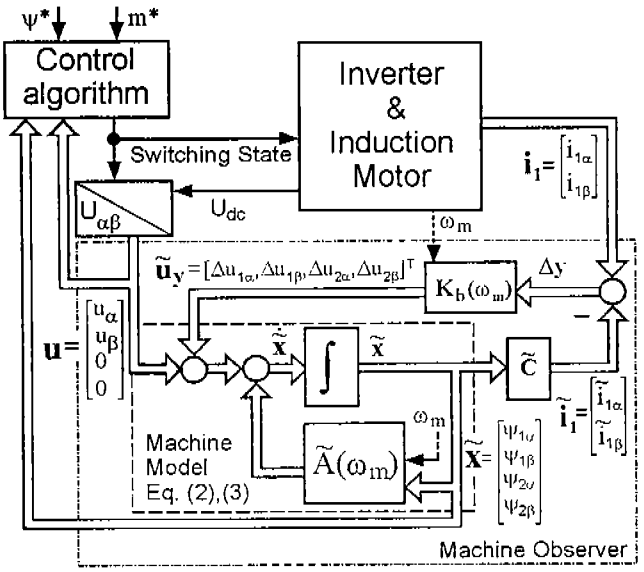

Fig. 5: Structure of control with observer

The control algorithm needs the observed states $\tilde{\mathbf{x}}$ not only at the equidistant points $\{..t_n,\ t_{n+1},...\}$ but at all switching events of the inverter $\{..t_n,\ t_n+h_{vv},\ t_{n+1},\ t_{n+1}+h_{vv1},...\}$. The states for these non-equidistant points are obtained by integrating the model numerically using Runge-Kutta second order method. The step size is chosen so that the boundaries of the integration intervals coincide with the switching events of the inverter. The speed ω_m is assumed to be constant over a cycle t_S.

The main task in designing the observer is to set the 2×4 feedback-matrix $\mathbf{K_b}(\omega_m)$. Here, a Kalman filter has been chosen. Its principle is to calculate the optimal gain as a function of covariance matrices of the process noise $\mathbf{Q}$, the measurement noise $\mathbf{R}$, and the state error $\mathbf{P}$, thus $\mathbf{K_b} = f(\omega_m, \mathbf{P}, \mathbf{Q}, \mathbf{R})$. In [6], the matrices $\mathbf{Q}$ and $\mathbf{R}$ are assumed to be constant, whereas $\mathbf{P}$ is updated each step using the actual state prediction error $(\tilde{\mathbf{x}} - \mathbf{x})$. In our case, the real time implementation of the complete algorithm, especially the update of $\mathbf{P}$ and $\mathbf{K_b}(\omega_m, \mathbf{P})$, would exceed the computational recources in the desired cycle time. On the other hand, after numerous iterations at constant speed, $\mathbf{P}$ varies less and less. $\mathbf{K_b}(\omega_m, \mathbf{P})$ reaches a steady state, and may be stored in a lookup table $\mathbf{K_b} = f(\omega_m)$ [7]. Its implementation is quite simple and fast. In this paper, to get $\mathbf{P}$ in steady state, we use the function `lqe` [8] for solving the Riccati-Equation (21) and getting directly $\mathbf{K_b}$:

$$\mathbf{P}\,\tilde{\mathbf{C}}^T\mathbf{R}^{-1}\tilde{\mathbf{C}}\mathbf{P} - \mathbf{P}\,\tilde{\mathbf{A}}^T - \tilde{\mathbf{A}}\mathbf{P} - \mathbf{Q} = 0 \qquad (21)$$

$$\mathbf{K_b} = (\mathbf{R}^{-1}\mathbf{C}\mathbf{P})^T \qquad (22)$$

The determination of $\mathbf{Q}$ and $\mathbf{R}$ is a delicate task. The switching delay of the inverter cause some process noise, slot harmonics too. Due to the switching delay and very high current slopes (i. e. 0.3 A/μs), the apparent measurement noise may become important. We have chosen:

$$\mathbf{Q} = C_{QR}\cdot\begin{bmatrix}64&0&0&0\\0&64&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\mathrm{V}^2; \qquad \mathbf{R} = C_{QR}\cdot\begin{bmatrix}1&0\\0&1\end{bmatrix}\mathrm{A}^2 \qquad (23)$$

Numerical restrictions of `lqe` obliged us to define $C_{QR} = 100$. It is to be mentioned that only the relationship between the elements of $\mathbf{Q}$ and $\mathbf{R}$ affecs the result. Using diagonal matrices for $\mathbf{Q}$ and $\mathbf{R}$ results in a symmetrical structure of $\mathbf{K_b}$:

$$\mathbf{K_b} = \begin{bmatrix}k_{b11}&k_{b12}\\k_{b21}&k_{b22}\\k_{b31}&k_{b32}\\k_{b41}&k_{b42}\end{bmatrix} = \begin{bmatrix}k_{b11}&-k_{b21}\\k_{b21}&k_{b11}\\k_{b31}&-k_{b41}\\k_{b41}&k_{b31}\end{bmatrix} = \begin{bmatrix}k_{b1}&-k_{b2}\\k_{b2}&k_{b1}\\k_{b3}&-k_{b4}\\k_{b4}&k_{b3}\end{bmatrix} \qquad (24)$$

For this example, fig. 6 shows the elements of $\mathbf{K_b}(\omega_m)$ for a speed range from −3300 rpm to +3300 rpm.

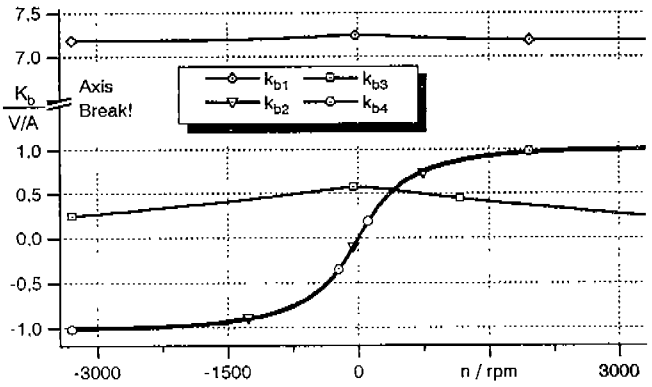

Fig. 6: Elements of the feedback matrix

As two elements the system matrix $\tilde{\mathbf{A}}$ depend on the speed, the numerical solution is repeated for different values $\omega_{m,v}$ over the speed range. To achieve a smooth behaviour, the resulting $\mathbf{K_b}(\omega_{m,v})$ are used for cubic spline approximations in ten overlapping intervals. Since ω_m varies slowly, the on-line adjustment is not time-critical.

IMPLEMENTATION ON DSP

The control algorithm induces some constraints. In contrast to quasi-analogue implementations, it is not possible to estimate the state of the machine based on actual measured quantities, because the currents of the machine change too fast. The delay between measurement and selection of the appropriate VV would be too large.

A quasi-analogue implementation observes always the state of the machine in order to match the switching criterion (i. e. the hysteresis boundary) as good as possible. A simple machine model allows to increase the *sampling* frequency, resulting in a considerable lower *switching* frequency. In our case, a fixed switching frequency of about 6.67 kHz is used. The average switching frequency

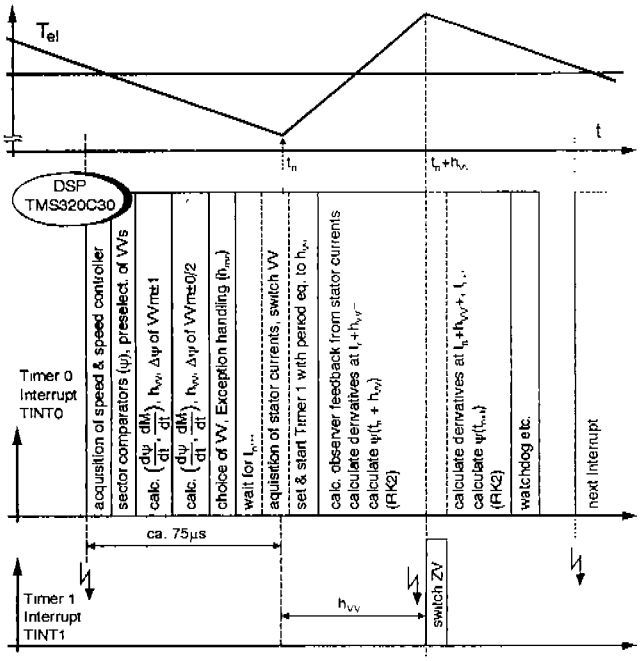

Fig. 7: Implementation of DMTC in DSP TMS320C30

per transistor is with DMTC usually half of the controllers switching frequency, thus 3.33 kHz. The cycle time of 150 µs is long enough to accomplish all tasks (fig. 2, 4, 5).

Fig. 7 illustrates the consecution of the control algorithm. A well defined time ahead of the switching instant t_n, a timer interrupt (TINT0) is generated. The control algorithm compares flux and torque of the model predicted for t_{n+1} with its setpoint values. Then the control scheme described above is processed. The second timer (TINT1) is programmed for the on-time of the VV. At t_n the VV becomes active and the counter for TINT1 starts. The different branches of the flux controller may cause varying delays for the computation whereas the model requires exact consistency to t_n. This problem can be solved by polling the timer of TINT0, waiting for t_n. After switching the VV, the states at the beginning of the next cycle (t_{n+1}) can be predicted. Two Runge-Kutta steps are calculated, one with the step-width h_{VV} applying the VV to the model and one with h_{ZV} and zero voltage.

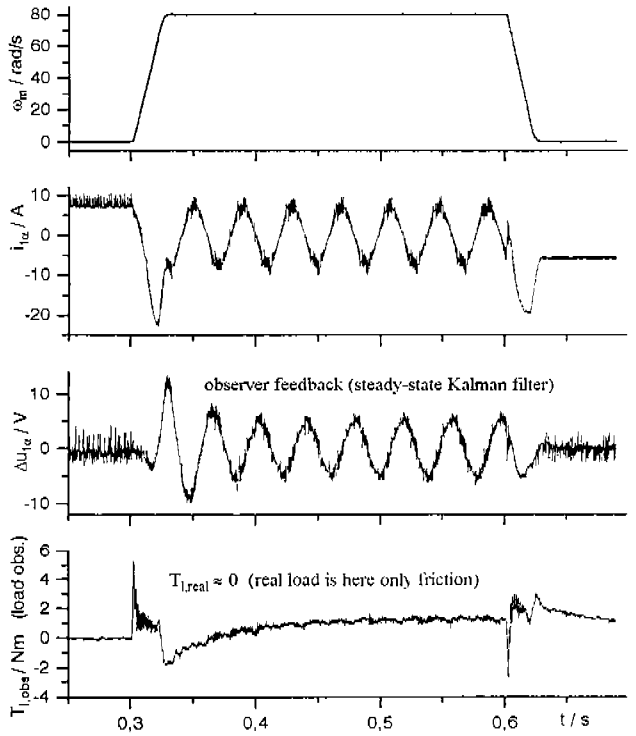

Fig.8: Experimental start-up and stop, T_N = 33Nm

EXPERIMENTAL RESULTS

The control scheme has been implemented on a DSP TMS320C30 at 40 MHz, mainly programmed in high level programming language C. To show the performance of the observer, fig. 8 presents a brief start-up to ω_m = 80rad/s, followed by a deceleration to zero. A fast change of the measured current can be noticed. The observer feedback is in the 2%-range of the intermediate circuit tension $u_{dc} \approx 600V$. A load observer with two poles at $\lambda_{Tl,obs} = -1600$rad/s has been implemented. Its results depend strongly on the knowledge of the electrical torque, thus the quality of the machine's observer. Only short error transients of about 15% nominal torque (T_N) occur.

REFERENCES

[1] Depenbrock, M.: Direkte Selbstregelung (DSR) für hochdyn. Drehfeldantriebe mit Stromrichterspeisung, etzArchiv, Bd. 7, 1985, Heft 7, p. 211-218

[2] Takahashi, I., Noguchi, T.: A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor, IEEE Transactions on Industry Applications., Vol. 22, No. 5, Sept./Oct. 1986

[3] Aaltonen, M., Tiitinen, P., Lalu, J., Heikkilä S.: Direkte Drehmomentregelung von Drehstromantrieben, ABB Technik 3/1995, p. 19-24

[4] Flach, E., Hoffmann, R., Mutschler, P.: Direct Mean Torque Control of an Induction Motor, EPE Trondheim/Norway 1997, Vol. 3, p. 672-677

[5] Flach, E.: Improved Algorithm for Direct Mean Torque Control of an Induction Motor, PCIM'98 Nürnberg/Germany 1998, IM p. 261-267

[6] Ben Ammar, F.;Pietrzak-David, M., de Fornel, B., Mirzaian, A.: Flux-Oriented Control of High-Power Induction Machines by Kalman Filter Observation, EPE Florenz/Italien 1991, Vol. 2, p. 182–187

[7] Atkinson, D. J.; Acarnlay, P. P.; Finch, J. W.: Observers for Induction Motor State and Parameter Estimation, IEEE Transactions on Industry Applications, Vol. 27, No. 6, Nov./Dec. 1991

[8] Grace, A.; Laub, A. J.; Little, J. N.; Thompson, C. M.: Control System Toolbox For Use with Matlab (version 3.0b), The Mathworks Inc. 1994