8th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
Suceava, Romania, May 25 – 27, 2006

A COMPARISON OF PI CURRENT CONTROLLERS IN FIELD ORIENTED INDUCTION MOTOR DRIVE

Dejan RELJIĆ[1], **Veran VASIĆ**[2], **Darko OSTOJIĆ**[3], **Boris DUMNIĆ**[4]
[1,2,3,4)] "Faculty of Technical Sciences" University of Novi Sad
Trg Dositeja Obradovića 6, 21000 Novi Sad
reljic@uns.ns.ac.yu

Abstract. *The PID controller is by far the most common algorithm. Most feedback loops are controlled by this algorithm or minor variations of it. Ttraditionally the PI controller has been widely used, while some other types of PID controller have been applied in some applications. In this article, the modified PI controller which has the feedforward term is introduced. In order to achieve desired performance criteria such as quick response, reference tracking, steady state error, system bandwidth and disturbance rejection capability, an optimization procedure for classical PI and modified PI controller is introduced with experimental results. The characteristic of current control loop of vector controlled induction machine with classical PI and modified PI controllers is also discussed along with experimental results.*

Keywords: *ac drives, vector control, current control, PI controller*

I. Introduction

PI(D) controllers are found in large numbers in all industries. They come in many different forms. PI(D) controllers are also embedded in all kinds of special purpose control system. They have survived many changes in technology ranging from pneumatics to integrated circuits and microprocessors [1].

These controllers have several important functions: they provide feedback, they have the ability to eliminate steady state error through the integral action, they can anticipate the future through the derivative action, and they can cope with the actuator saturation [1]. PI(D) controllers are also sufficient for many control problems, particularly in cases where modest performance requirements are needed. They are thus important components in the control engineer's toolbox [1].

The microprocessor has had a dramatic effect on PI(D) controllers as on other types of industrial electronics, and a large number of those manufactured today are based on microprocessors.

Although PI(D) controllers are common and well known, they are often poorly tuned. Evidence for this can be found in the control rooms of any industry. For example, 20 % of control loops are improperly designed, 15 % are not installed correctly, 30 % of control loops have out of sense tuning parameters, 85 % are not tuned well [2].

This article deals with tuning current controller in field oriented vector controlled induction machine drive. In motion control applications a classical multi loop structure has been widely used and each loop is controlled by appropriate controller. The multi loop structure is still favored because of simplicity in control and the natural hierarchy of bandwidth requirements [3].

The first step is to select correct algorithm. Each algorithm uses a different set of mathematical equations to determine controller action. PID controller has been very popular due to its simple structure and relatively easy tuning procedures with acceptable performance [3]. But, there are few different PID algorithms. Selecting the right one algorithm requires understanding of the process to be controlled. Although not as popular as PID, some modified PID controllers have been proposed and used in some applications. In this article the characteristic of current control loop of vector controlled induction machine with classical PI and modified PID controllers is discussed along with experimental results.

The next step that is performed is to choose correct tuning algorithm for current controller. Traditional control techniques based on modeling and design are revealed, but there are also special methods for direct tuning based on simple process experiments. Trial and error methods of tuning controllers are the simplest methods and still remain in widespread use today. But, trial and error methods are not enough in today's world. Several tuning techniques can provide a good intuitive concept to tune the drive without knowing the accurate model of a electric drive. Sometimes, these tuning methods can not provide satisfactory responses and adequate bandwidth of the current loop. Additional tuning would be done. In this article an optimization procedure for classical PI and modified PID controllers using Matlab's Nonlinear Control Design Blockset is described along with experimental results.

II. A Short Description of Vector Control of Induction Motor Drive

Application such as robotics and factory automation require accurate control of speed and position. This can be accomplished by vector control of induction machines, which emulate the performance of dc motor and brushless dc motor servo drives. Compared to dc and brushless dc motors, induction motors have a lower cost and a more rugged construction [4].

In any speed and position control application, torque is the fundamental variable that needs to be controlled. The ability to produce a step change in torque on command represents total control over the drives for high performance speed control [5].

There are many ways in which an induction motor drive can emulate the performance of dc and brushless dc motor drives. One of the vector control methods is called indirect vector control in the rotor flux reference frame. For many other possible methods and their pros and cons, readers are urged to look at several books on vector control. Rotor flux oriented control has emerged as one of the most frequently used techniques. The application of this technique yields fast dynamic response and most of the requirements such as smooth speed response without cogging or torque pulsations at low speed, smooth speed reversal under any torque condition, operation of the drive with constant full torque below base speed and above base speed with reduced flux, can also be satisfied [6]. A block diagram of a indirect vector controlled induction motor in the rotor flux reference frame drive is shown in Figure 1. In vector control of induction motor drives, the stator phase currents $i_a(t), i_b(t)$ and $i_c(t)$ are controlled in such manner that i_{qs} delivers the desired electromagnetic torque while i_{ds} maintains the peak rotor flux density at its rated value [4]. The reference values i_{qs}^* and i_{ds}^* are generated by the speed control loops. The current reference i_{qs}^* is generated by speed controller and the current reference i_{ds}^* is kept constant in order to keep the rotor flux along the d axis constant. For operation in an extended speed range beyond the rated speed, the flux weakening should be implemented as a function of rotor speed, and i_{ds}^* is also generated by speed controller. The space angle of the rotor flux linkage space phasor is obtained as the sum of the monitored rotor angle (θ_r) and the computed reference value of the slip angle (θ_{sl}), where the slip angle gives the position of the rotor flux linkage space phasor relative to the rotor (or more precisely relative to the d axis of the reference frame fixed to the rotor). As shown in the block diagram of Figure 1, we can generate references v_{qs} and v_{ds} from given i_{qs}^* and i_{ds}^*. Voltage references $v_{\alpha s}$ and $v_{\beta s}$ are calculated from given v_{qs}, v_{ds} and calculated value of θ_s. The actual stator voltages are supplied by the power electronic converter, using the stator voltage space vector modulation technique (SVPWM). The two stator phase currents $i_a(t)$ and $i_b(t)$ are measured and the third one, $i_c(t)$, is calculated. A step change in the stator q winding current suddenly induces currents in the rotor equivalent q axis winding while keeping its flux linkage zero. Therefore, the rotor flux linkage remains unchanged in amplitude.

Sudden appearance of current along the rotor q axis, in the presence of d axis flux, results in a step change in torque. To maintain the induced rotor q winding current from decaying, the dq winding set must be rotated at an appropriate slip speed with respect to the rotor [5].

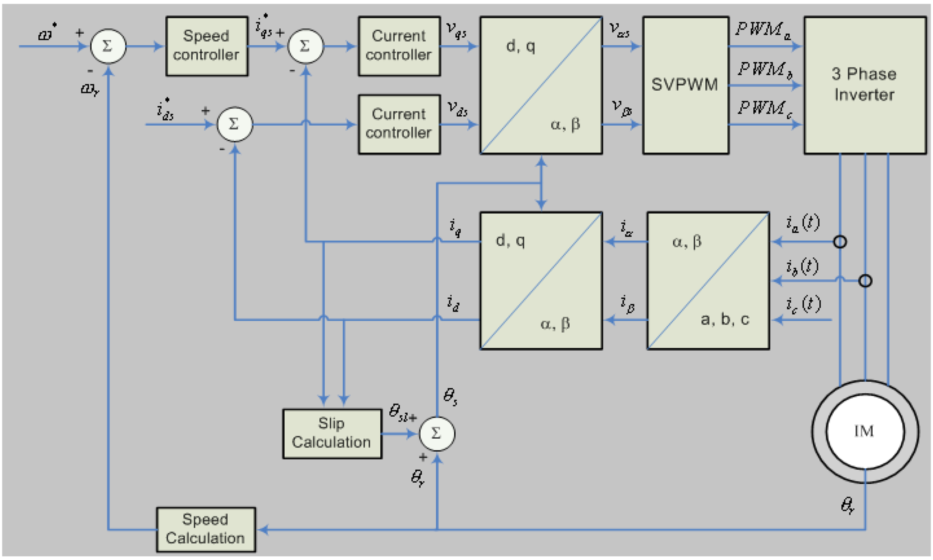

Figure 1. Schematic of the Indirect Implementation of the Rotor Flux Oriented Control of an Induction Motor Drive.

III. Design of the Current Controller

Current regulation is performed in a dq domain (B domain). A block diagram representation of the current control loop is illustrated in Figure 2.

The current references i_{qs}^* and i_{ds}^* (inputs in the block diagram of Figure 2) are generated by the cascaded speed control loop shown in the block diagram of Figure 1, where ω^* is speed reference input. For operation below the rated speed i_{ds}^* is kept constant in order to keep the rotor flux along the d axis constant (Figure 1). A space vector pulse width modulated inverter (SVPWM) is used to supply motor voltages that result in the desired currents calculated by the controller. Currents are measured and transformed into digital signals by AD converter. The error between the reference and the measured current serves as a input to the current controller.

The first step is to establish current controller structure. Each algorithm uses a different set of mathematical equations to determine current controller action. The PID controller is by far the most common algorithm.

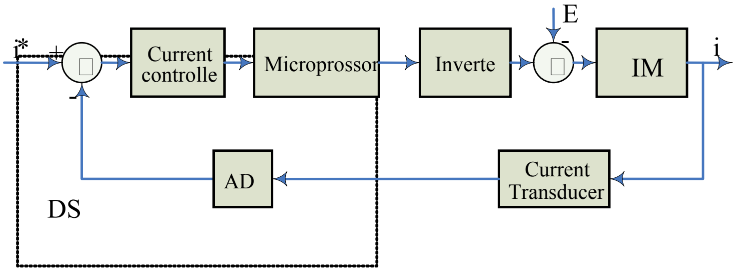

Figure 2. A block diagram representation of the current control loop in the Indirect Implementation of the Rotor Flux Oriented Control of an Induction Motor Drive.

A very common textbook version of the PID algorithm [1] has the form shown in equation 1. The system is characterized by forming an error (e) that is the difference between the set point and measured value. The control variable (v) is thus a sum of three terms: P term, I term and the D term [1]. The controller parameters are proportional gain K, integral time Ti and derivative time Td.

$$v = K[e(t) + \frac{1}{T_i}\int_0^t e(s)ds + T_d\frac{de(t)}{dt}] \tag{1}$$

The loop gain should be high in order to ensure that process output is close to set point. The main function of the integral action is to make sure that the process of output agrees with the set point in steady state. The purpose of the derivative actions is to improve the closed loop stability and to predict future error. But, differentiation is always sensitive to noise. This is clearly seen from the transfer function $G(s)=s$ of a differentiator which goes to infinity for large s. For this reason, D action is often switched off. In a practical controller with derivative action it is necessary to limit the high frequency gain of the derivative term. This can be done by suitable implementation of the derivative term. Note that derivative action does not help if the prediction time Td is too large. However, in this practical example of the current control loop in field oriented vector controlled electric drive, a classical PI controller has been introduced. PI control is adequate for all processes where the dynamics are essentially of the first order [1]. It was found out by measuring the step response or the frequency response of the current control loop. The step response looks like that of the first order system. According to [1], PI control is sufficient. The textbook version of the PI algorithm is described by:

$$v = K[e(t) + \frac{1}{T_i}\int_0^t e(s)ds] \tag{2}$$

In general, a current control system has many different requirements. It should have good transient response to a set point changes, and it should reject disturbances and measurement noise. For a system with error feedback only, an attempt is maid is to satisfy all demands with the same mechanism [1]. Such systems are also called one degree of freedom systems.

There is much more to the PI controller than is revealed by equation 2. A faithful implementation of the equation 2 will actually not result in a good current controller. To obtain a good PI controller it is also necessary to consider set point weighting, tuning and computer implementation. When using the control law given by equation 2 it follows that a step change in the reference signal will result in an impulse in the control signal. This is often highly undesirable. This problem can be avoided by filtering the reference value before feeding it to the controller. Another possibility, that is presented in this article, is to let proportional action acts only on part of the reference signal. This is called set point weighting [1]. The PI controller given by equation 2 then becomes:

$$v = K[e_p(t) + \frac{1}{T_i}\int_0^t e(s)ds] \tag{3}$$

Here the error in the proportional part (ep) is:

$$e_p = b \cdot i^* - i \tag{4}$$

where b is additional parameter. The integral term must be based on error feedback to ensure the desired steady state. The controller given by equation 3 has a structure with two degrees of freedom because the signal path from i to v is different from that from i* to v. The transfer function from i* to v is:

$$G_{i^*}(s) = K \cdot (b + \frac{1}{s \cdot T_i}) \tag{5}$$

and the transfer function from i to v is:

$$G_i(s) = K \cdot (1 + \frac{1}{s \cdot T_i}) \tag{6}$$

By having different signal paths for the set point and the process output (two degrees of freedom systems), there is more flexibility to satisfy the design compromise. The controllers obtained for different values of b will respond to load disturbances and measurement noise in the same way as the controller defined by equation 2. The response to set point changes will, however, depend on the value of b [1]. The overshoot for set point changes is smallest for b = 0, which is the case where the reference is only introduced in the integral term, and increases with increasing b. The reason for this is that modified controller has a zero at [1]:

$$s = -\frac{1}{b \cdot T_i} \tag{7}$$

which can be positioned properly by choosing the parameter b suitably. In contrast to this controller, the controller defined by equation 2 introduces a closed loop zero at:

$$s = -\frac{1}{T_i} \qquad (8)$$

which depends only on integral time Ti.
In order to avoid an excessive overshoot, in [1] is suggested that parameter b should be chosen so that the zero is two to three times larger than the magnitude of the dominant poles.
This is carried much further in more sophisticated control system.

IV. Implementation and Tuning Method of the Current Controller

One method of tuning a controller is to first determine a model for the process dynamics and than calculate the controller parameters using some design method. The model of the current control loop in the indirect implementation of the rotor flux oriented control of an induction motor drive (Figure 1 and Figure 2) is illustrated in Figure 3.

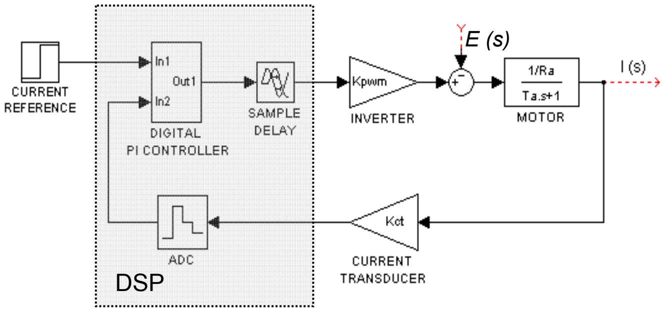

Figure 3. Model of the current control loop.

A block diagram representation of the current control loop shown in Figure 3 has the following meanings:
Digital PI controller – software implemented PI controller,
Sample delay – delay of the computed control variable,
Inverter – power electronic converter that for given control variable generates motor voltages,
Motor – equivalent electrical circuit of one phase winding (equivalent d or q winding)
Current transducer – used for current measuring,
ADC – analog to digital converter,
E (s) – back emf.

Current controllers are implemented in Digital Signal Processor (DSP). When the controller is implemented in a DSP, the analog inputs are read and the outputs are set with a certain sampling period. This is a drawback compared to the analog implementations, since the sampling introduces dead time in the control loop. Sequences of operations are the following: wait for clock interrupt, read analog input, compute control signal, set analog output, update controller variables, go to 1.

The methods of tuning the current controller differ with respect to the knowledge of the process dynamics they require. Since the model of the current control loop can be determined, numerous methods, such as Trial and error (by far the most utilized), Ziegler-Nichols (quarter turn amplitude, critical oscillation), Lambda method and others, are available. Sometimes, these tuning methods can not provide satisfactory responses and adequate bandwidth of the current control loop. Additional tuning should be done. Another tuning procedure is based on control specifications [7]. In that case factors such as set point tracking (ITAE, ITE, ITSE), rise time, settling time, overshoot ratio, steady state error, attenuation of load disturbances (IAE, IE, ISE) and others are taken into account.

In this article an optimization procedure for classical PI and modified PI controllers using Matlab's Nonlinear Control Design Blockset is presented.

The Nonlinear Control Design Blockset provides a graphical user interface (GUI) to assist in time-domain-based control design. With this blockset, it is possible to tune parameters within a nonlinear Simulink model to meet time domain performance requirements by graphically placing constraints within a time domain window [8]. Any number of Simulink variables can be declared tunable by entering the variable name into the appropriate dialog box. A classical PI controller is described by two parameters (K and Ti) and modified PI controller is described by three parameters (b, K and Ti). Uncertainty bounds can be placed on other variables in the model for robust control design [8]. The Nonlinear Control Design Blockset makes attaining performance objectives and optimizing tunable parameters an intuitive and easy process [8]. After adjusting the constraint bounds in the Nonlinear Control Design Blockset constraint figure and declaring the tunable variables using the Optimization Parameters dialog box. When the optimization is started, the Nonlinear Control Design Blockset automatically converts the constraint bound data and tunable variable information into a constrained optimization problem. It then invokes the Optimization Toolbox routine. The routine adjusts the tunable variables in an attempt to better achieve the constraints on system signals defined by the Nonlinear Control Design Blockset main interface. The routine solves constrained optimization problems using a sequential quadratic programming (SQP) algorithm and quasi-Newton gradient search techniques [8]. In short, the optimization problem formulated by the Nonlinear Control Design Blockset minimizes the maximum constraint violation.

V. Results of Computer Simulations

Matlab software is very useful tool for analyzing dynamic processes. The whole current control loop (Figure 3) is modeled using Matlab - Simulink software. The model has been developed using available data and it is a little bit simplified for the fake of simplicity. Thanks to this, it is possible to perform a computer simulation and get usable results in a brief period of time.

Because of magnetic core symmetry, the current time responses of both currents (i_{qs}^{*} and i_{ds}^{*}) are identical. It means that parameters for q and d current controllers are the same. Sample time (Ts) is 100 μs, Kpwm= 360 V, Ra= 10 Ω, Ta=6,46 ms, Kct= 0,0952 A-1.

We want to design PI controller for the drive so that the current closed loop system meets the following tracking specifications:

1 kHz bandwidth,

less than 25 % overshoot,

Maximum 6tr settling time,

Further, we want the closed loop response to have good disturbance rejection capability.

Step signal $i_{qs}^{*} = 1{,}6$ A is used as reference input. With the Nonlinear Control Design, parameters are tuned within the Simulink model to meet performance requirements. Responses to step change in the set point are shown in Figure 4 (classical PI controller) and Figure 5 (modified PI controller). The simulations support the results of the analysis. Parameters that are obtained for classical PI controller are: K=6,25, Ti=0,0021 (Ki=3000) and for modified PI controller: K=5,93, Ti=0,0020 (Ki=3000), b=0,91. Although performance tracking requirements were accomplished with both classical and modified PI controllers, modified PI controller has achieved even better tracking specifications. Overshot is reduced to 15 % and settling time is about 4,7tr. These are very important features for current control loop because torque pulsations are reduced.

The error integral due to a disturbance is inversely proportional to Ki (Ki= K/Ti) [1]. Since both classical and modified PI controllers have the same integral gain Ki, both controllers will respond to disturbances in the same way. For most practical systems, both reference tracking and disturbance rejection capability are important.

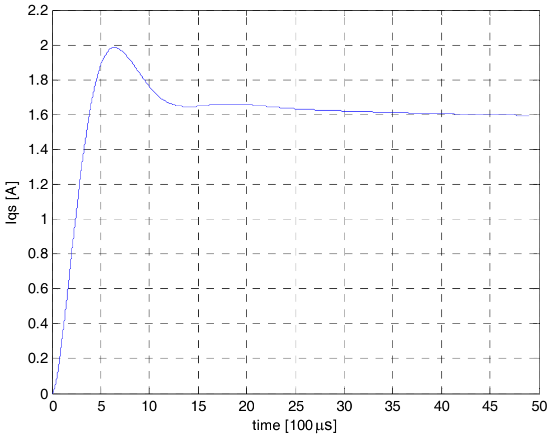

Figure 4. Step response of iqs current with classical PI controller.

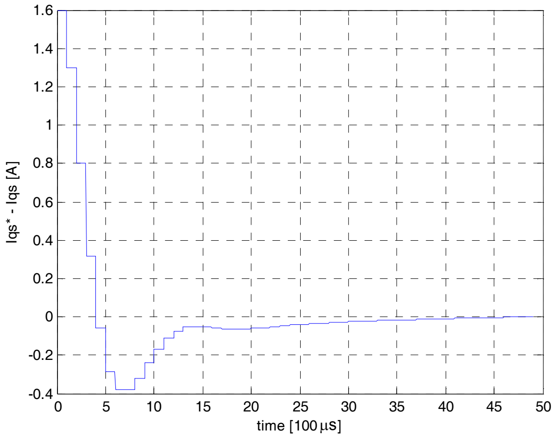

Figure 5. Error signal e for the step response shown in Figure 4.

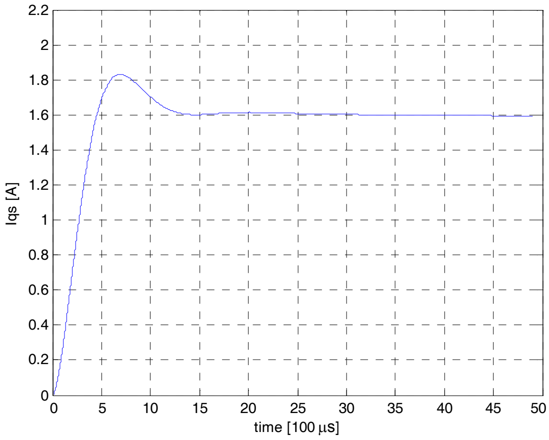

Figure 6. Step response of the iqs current with modified PI controller.

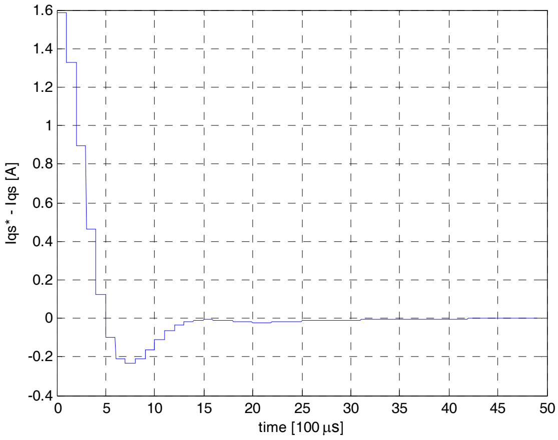

Figure 7. Error signal e for the step response shown in Figure 6.

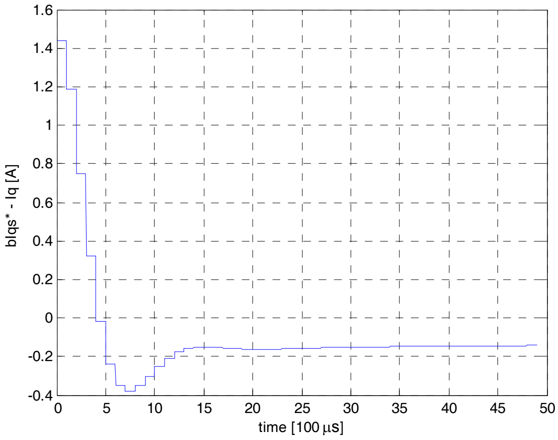

Figure 8. Error signal ep for the step response shown in Figure 6.

VI. Experimental Results

The above computer simulations are verified by a laboratory prototype of vector controlled induction machine drive. The drive consists of induction machine ZK 80 B4, laboratory prototype of inverter and ADMC401 Development Kit. Development Kit ADMC401 is flexible DSP system which is designed for motor control applications. Current controllers are digitally implemented and each algorithm (PI and modified PI) uses a different set of mathematical equations to determine current controller action. Those equations are implemented in DSP.

Figure 9 and Figure 10 show reference step response of iqs current with classical and modified PI controller which are tuned according to parameters that were obtained from above computer simulations. Note that actual values shown in the figures (y axis) are in hexadecimal but can be compared to simulation results because the same step signal as in simulations (0x1300hex = = 1,6 A) is used as reference input. The results illustrated in Figure 9 and Figure 10 show the great level of matching Figure 4 and Figure 6 This indicates the developed Simulink current control loop model is good enough to obtain controller's parameters.

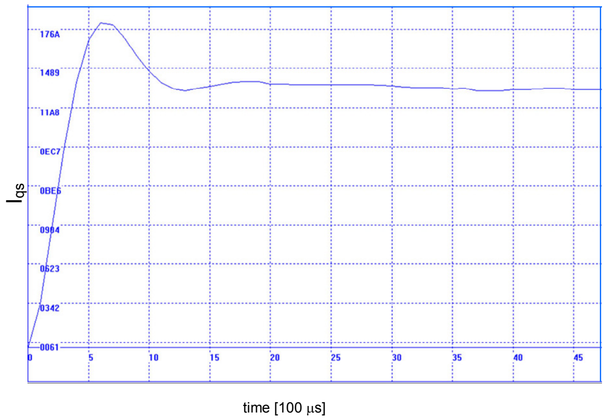

Figure 9. Step response of the iqs current with classical PI controller.

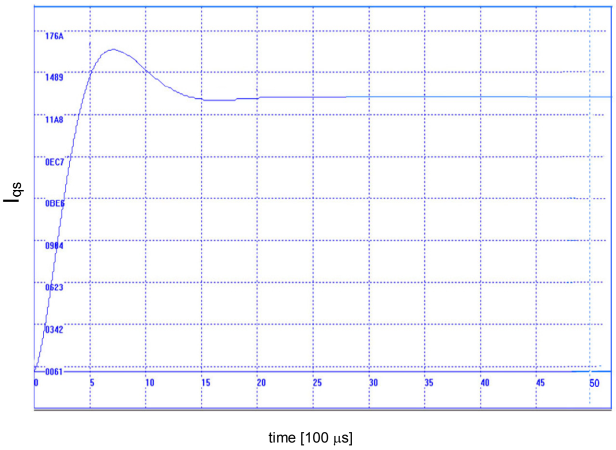

Figure 10. Step response of the iqs current with modified PI controller.

As it can be seen, modified PI controller has achieved better tracking specifications. Overshot and settling time are reduced and as a consequence no torque pulsations are expected. Note that this experiment was done at standstill.

VII. Conclusion

This paper has presented design procedure of the current controllers in field oriented vector controlled electric drive. For most practical systems, both reference tracking and disturbance rejection capability are important, and modified PI controller gives flexibility in weighing relative importance of these two performance requirements. The only disadvantage of here presented modified PI controller is that it has 3 parameters to tune. However, using Matlab's Nonlinear Control Design Blockset it is quite easy tuning procedure. For that reason a model of a current control loop should be first determined and than calculate the controller parameters using Nonlinear Control Design Blockset.

The effectiveness of this method is demonstrated by experiment. Excellent tracking specifications are achieved.

References

[1] Karl Johan Astrom, Tore Hagglund (1988) *Automatic Tuning of PID Controllers*, Instrument Society of America.

[2] ***, www.expertune.com, *On – Line Process Control Tutorials*

[3] Dal Y. Ohm (1994) *PID and PDFF Compensators for Motion Control*, IEEE IAS Annual Meeting.

[4] Ned Mohan (2000) *Electric Drives, an Integrative Approach*, MNPERE, Minneapolis.

[5] Ned Mohan (2001) *Advanced Electric Drives, Analysis, Control and Modelling using Simulink*, MNPERE, Minneapolis.

[6] Peter Vas (1998) *Sensorless Vector and Direct Torque Control*, Oxford University Press.

[7] Guy A. Dumont (2005) *PID Control-2-, EECE 460 Control Systems Lecture Notes –* Guy A. Dumont.

[8] ***, *MATLAB documentation and related papers*