Anti-windup Schemes for Proportional Integral and Proportional Resonant Controller

Anirban Ghoshal and Vinod John
Department of Electrical Engineering, Indian Institute of Science, Bangalore
Email: anirban@ee.iisc.ernet.in, vjohn@ee.iisc.ernet.in

***Abstract*—Problems like windup or rollover arise in a PI controller working under saturation. Hence anti-windup schemes are necessary to minimize performance degradation. Similar situation may occur in a Proportional Resonant (PR) controller in the presence of a sustained error input. Several methods can be employed based on existing knowledge on PI controller to counter this problem. In this paper few such schemes are proposed and implemented in FPGA and MATLAB and from the obtained results their possible use and limitations have been studied.**

***Index Terms*—Anti-windup, Proportional resonant control, PI control, Current control.**

I. INTRODUCTION

In a PI controller windup or rollover is a widely studied problem [1–5]. Typically this problem arises if the input error to the controller is large or the input error remains nonzero for a long time. The controller output may saturate either because of large error input or because of nonzero error which makes the integrator output keep on accumulating. A controller under saturation may give delayed response to any change in the input and this delay would be more if the controller goes into deeper saturation level. Again digital implementation of controller using DSP or FPGA having fixed word length asks for careful choice of per unit values and saturation level of the controller to prevent windup phenomena. Hence it becomes necessary to employ anti-windup strategy to prevent the controller from going into deep saturation and to check windup or rollover of controller output. Anti-windup schemes for PI controller are a well researched topic though newer and innovative methods are still coming up. Several such schemes for PID controllers utilizing limited or conditional integration method and tracking anti-windup method have been discussed in [1]. A modified tracking anti-windup method for PI controllers used in variable speed motor drives is described in [2]. Design of an anti-windup scheme utilizing the benefits of both conditional integration and tracking or back-calculation method is shown in [3]. A survey on current controlled techniques [4] shows that for control of electric drives and grid connected inverter systems, popular controller strategies use PI controllers. Anti-windup strategies for PI controllers used in such applications were under study for long. In recent times current control strategies such as Proportional Resonant (PR) Controller are gaining popularity for their advantages over PI controller particularly for control of grid connected inverters [6, 7]. Here the controller operates on AC signal whereas in popular d-q control or the synchronous reference frame based control the PI controller works on DC signal. It can be shown that under certain condition a PR controller can become prone to windup phenomena. In this paper studies on few ‘Anti-windup’ schemes for PI controller and their applicability in solving windup problem in PR controller are discussed. These methods were simulated in MATLAB and then implemented on a FPGA based digital controller. This digital controller board uses Altera’s ‘Cyclone’ series FPGA. A word length of 16 bit, $1pu = 3FFFh$ which is equivalent to $5V$ when viewed through DAC and sampling rate of 10kHz were used for implementation.

II. PI CONTROLLER WITH ANTI-WINDUP

A. PI controller

Transfer function of a PI controller is expressed as

$$G_{PI}(s) = K_p + \frac{K_i}{s} \tag{1}$$

Here K_p is the proportional constant and K_i is the integral constant. The output from controller y can be expressed as

$$y = K_p \times e + K_i \int edt \tag{2}$$

Here e is the input error. The block diagram for implementation is given in figure: 1. Nowadays digital control platforms like DSP or FPGA with fixed word length are more in use for implementation of all control logics.

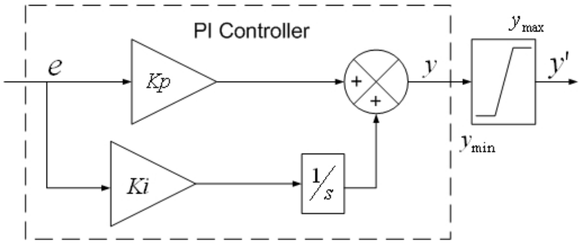

Fig. 1. Block diagram representation of a PI controller with saturation.

So the variables can attain values between certain finite range and any value out of that range will lead to roll over and output of controller (y) would show it as a sudden jump from positive to negative or vice versa. Such undesirable condition can arise if the error (e) is too large or it remains non-zero for long duration during which the integrator causes the roll over. To limit the output a saturation block can be used at the output terminal as shown in Fig. 1. Here output $(y^{'})$ can be expressed as

$$y^{'} = \begin{cases} y & \text{for } Y_{max} > y > Y_{min} \\ Y_{max} & \text{for } y \geq Y_{max} \\ Y_{min} & \text{for } y \leq Y_{min} \end{cases} \tag{3}$$

But the above approach would lead to problem if the input 'e' remains non-zero for long and during that period the integrator output keeps on accumulating. This may either introduce delay in response when the input error changes or even lead to roll over of output 'y' as can be seen in Fig. 2. To avoid this it is necessary to check the integration process during such situations which is in general known as anti-windup. In this paper study has been carried out on two schemes using conditional integration and tracking anti-windup. Discussions on these commonly adopted anti-windup schemes for PI controllers are in the next subsections.

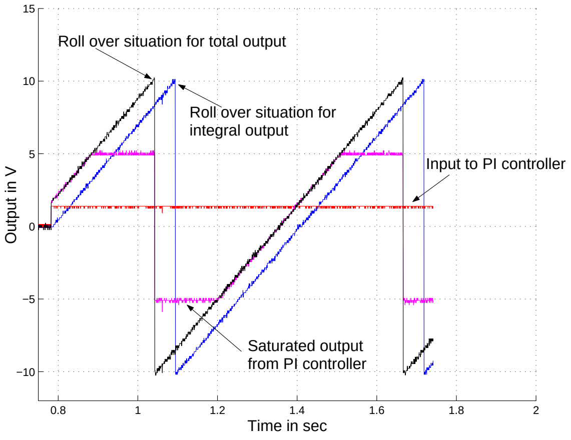

Fig. 2. Output of PI controller with saturation implemented in FPGA for error input $e = 1.25$. Here $Kp = 1.33$ and $Ki = 20.7$. Upper saturation limit $Y_{max} = 5V$ and lower saturation limit $Y_{min} = -5V$.

B. Conditional Integration

The first method, shown in Fig. 3 is by stopping the integration process when the output y has reached the saturation limit. This method can also be considered as a variable structure PID control method discussed in [5]. This ensures that while the controller is experiencing saturation there is no further increase in the value of output 'y'. If the error reduces below certain level for which output comes out of saturation, the integrator starts working again. The results are shown in Fig. 4. In this method the PI controller output never goes beyond the saturation limit. Upon reaching saturation the value at which the integrator output is held, depends on input error magnitude and the proportional constant. A variation of this method can be by forcing the integrator output to some predetermined value whenever the controller output gets saturated.

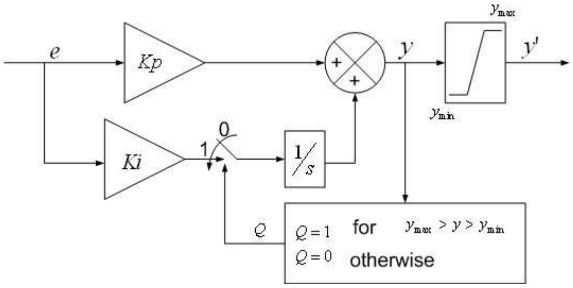

Fig. 3. Block diagram representation of a PI controller with anti-windup scheme described in Conditional Integration.

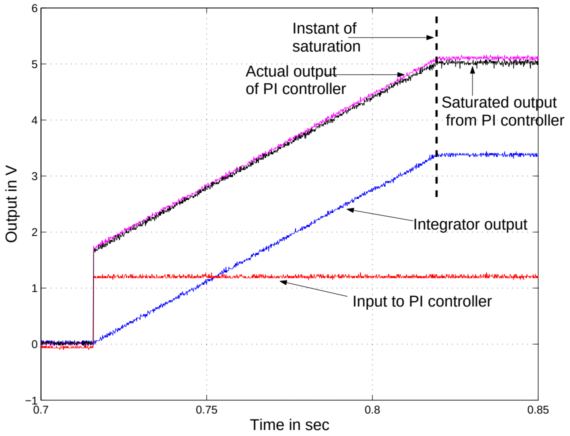

Fig. 4. Output of PI controller with anti-windup scheme of Conditional Integration implemented in FPGA for an error input $e = 1.25$. Here $Kp = 1.33$ and $Ki = 20.7$. Upper saturation limit $Y_{max} = 5V$ and lower saturation limit $Y_{min} = -5V$.

C. Tracking Anti-windup

Fig. 5 shows another way of controlling the extent of saturation. Here the difference of actual output (y) and saturated output $(y^{'})$ is fed back through a gain (K_{lim}) to reduce the amount of error input e going into integrator.

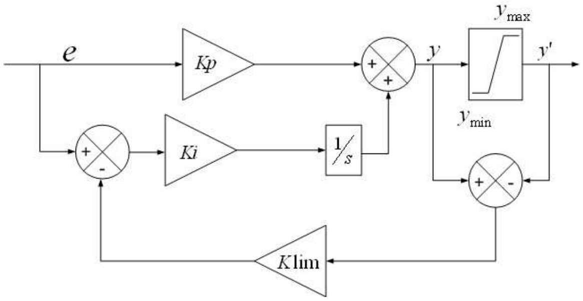

Fig. 5. Block diagram representation of a PI controller where the difference between actual output (y) and saturated output $(y^{'})$ is used as a feedback signal through a gain (K_{lim}) to reduced the input to integrator.

Thus rate at which the output of integrator would have increased is reduced. Under normal situation the anti-windup path does not contribute in any way as $y = y^{'}$. The results are shown in Fig. 6. Choice of limiting gain (K_{lim}) depends on acceptable restriction on integrator output. Higher value of K_{lim} keeps the actual output (y) close to the saturated output $(y^{'})$ which in turn enables the controller to come out of saturation quickly when the error reverses.

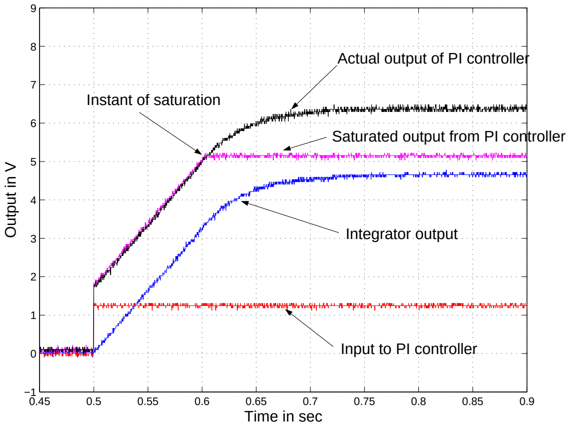

Fig. 6. Output of PI controller with anti-windup scheme of Tracking Anti-windup implemented in FPGA for an error input $e = 1.25$. Here $K_p = 1.33$, $K_i = 20.7$ and $K_{lim} = 1$. Upper saturation limit $Y_{max} = 5V$ and lower saturation limit $Y_{min} = -5V$.

The steady state value of output 'y' is of interest for choosing K_{lim}. Considering a situation where $y > Y_{max}$ following relationships can be derived to get an indication of steady state and dynamic behaviour of the PI controller under saturation with tracking anti-windup control. If the output of integrator is x, then

$$\frac{dx}{dt} = K_i e - K_{lim}(y - Y_{max}) \tag{4}$$

$$y = x + K_p \times e \tag{5}$$

Hence,

$$\frac{dx}{dt} = -K_i K_{lim} x + K_i(1 - K_{lim}K_p)e + K_i K_{lim} Y_{max} \tag{6}$$

Solution to the above differential equation for a constant error input $e(t) = E$ gives

$$x(t) = (X_0 - \frac{E}{K_{lim}} - Y_{max} + K_p \times E)\exp^{-K_i K_{lim} t} + (\frac{E}{Klim} + Y_{max} - K_p \times E) \tag{7}$$

Here X_0 is the initial value of state 'x'. Now output 'y' can be expressed as

$$y(t) = (X_0 - \frac{E}{K_{lim}} - Y_{max} + K_p \times E)\exp^{-K_i K_{lim} t} + (\frac{E}{Klim} + Y_{max}) \tag{8}$$

If steady state value of 'y' is Y_{ss}, then

$$Y_{ss} = Y_{max} + \frac{E}{K_{lim}} \tag{9}$$

It can be observed from the dynamic and steady state relationships that

1) the value of K_{lim} has to be high so that $Y_{ss} \simeq Y_{max}$ and hence controller will come out of saturation quickly when the error input 'e'reverses.
2) to reach steady state quickly under saturation condition K_{lim} has to be very high.

III. PR CONTROLLER WITH ANTI-WINDUP

A. PR Controller

A PR controller has a transfer function of the form

$$G_{PR}(s) = K_p + \frac{sK_i}{s^2 + \omega^2} \tag{10}$$

Here $\frac{sK_i}{s^2 + \omega^2}$ is the resonant part whose resonant frequency is at ω and K_p is the proportional part. Discussions on PR controller can be found in [6, 7].

The resonant part is able to provide a very high gain at and around the resonant frequency which forces the error to become zero at steady state. Frequency response of the resonant part of the PR controller, generated using MATLAB and the frequency response of the actual controller implemented in FPGA are shown in Fig. 7. Fig. 8 shows the block diagram form of realisation of PR controller. The discrete time equations (11), (12) and

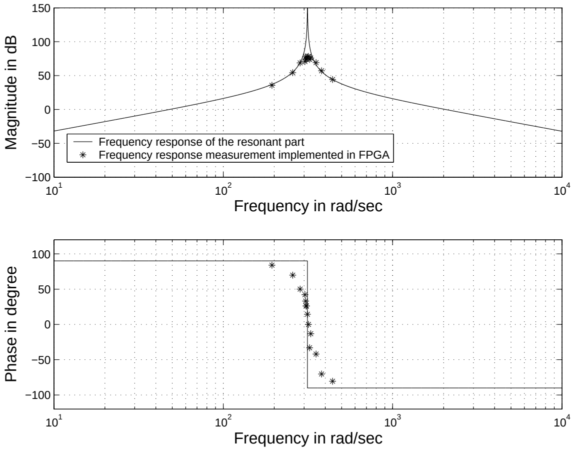

Fig. 7. Frequency response of the resonant part of a PR controller. Here $K_i = 2000$ and $\omega = 314 rad/sec$.

(13) for a stable realisation are obtained using Eulers modified method. Here ΔT is the integration step size.

$$p[n+1] = p[n] + K_i \times \Delta T \times e[n] + \omega \times \Delta T \times q[n] \quad (11)$$

$$q[n+1] = q[n] - \omega \times \Delta T \times p[n+1] \quad (12)$$

$$y[n+1] = K_p \times e[n] + p[n+1] \quad (13)$$

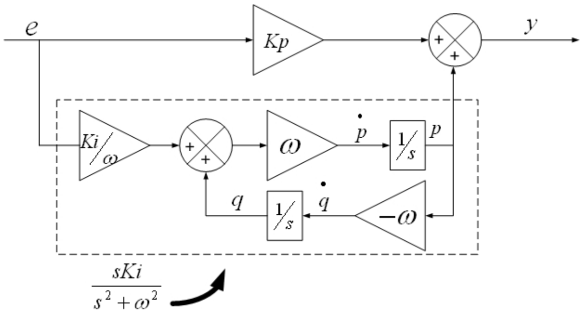

Fig. 8. Block diagram representation of PR controller.

Present literatures on PR controller shows several other ways of realisation. In the most commonly used form the resonant part is realized by a second order transfer function with a non-zero damping factor. In this realisation the resonant controller output is bounded for any sinusoidal input but error input is never actually zero. Here choice of damping factor would decide on error magnitude. Also, design of controller should take account of maximum allowable error input for a given damping factor to prevent roll over. The way of implementation where damping factor is zero, shown in Fig. 8 and implemented by equations (11), (12) and (13), works fine as the error 'e' goes to zero at steady state. But if error 'e', typically a sinusoidal signal, persists for long because of some operating situations then output of resonant part can show unstable behaviour. The time domain expression for output of resonant part i.e. 'p', can be expressed as follows

$$p(t) = A_0 \sin(\omega t + \phi_0) + \frac{A \times K_i \times t}{2} \sin(\omega t - \phi) - \frac{A \times K_i}{2\omega} \sin(\omega t) \sin(\phi) \quad (14)$$

where,

$$e(t) = A \times \sin(\omega t - \phi) \quad (15)$$

$$A_0 = \sqrt{P_0^2 + Q_0^2} \quad (16)$$

$$\phi_0 = \tan^{-1}(\frac{P_0}{Q_0}) \quad (17)$$

P_0 and Q_0 are the initial conditions of the states p and q. The above expression shows that if the error is present continuously then output of resonant part would keep on increasing and after sometime the entire controller output y would experience roll over. So it becomes necessary for the controller to limit the output under such situations. Therefore use of anti-windup logic need to be looked into. In this regard the schemes that find use in case of a PI controller can be used with a PR controller and their suitability and limitations of the proposed methods can be studied.

B. Method 1

The scheme of Fig. 3 would not directly work for a PR controller as stopping of integrator will introduce large DC offset into controller output. Fig. 9 represents an anti-windup scheme where the resonant part is withdrawn from the control loop and the integrator outputs are forced to zero if the output of controller goes out of desired range. This can be considered reasonable in situations where feed-forward terms and proportional terms are responsible for a large portion of the control action and the integral terms are responsible only to zero out long term output error.

The results are shown in Fig. 10. It can be observed that due to the proportional part the output from the controller will not be zero. Reappointing the controller has to be done according to the application, from a higher level master control system, or a manual reset can be a solution for this method.

In general to minimize the burden on controller the grid voltage is added as a feed-forward term. The desired range of controller output can be derived from the steady state drop of voltage across the filter between inverter and grid, in case of a grid connected application. Now if

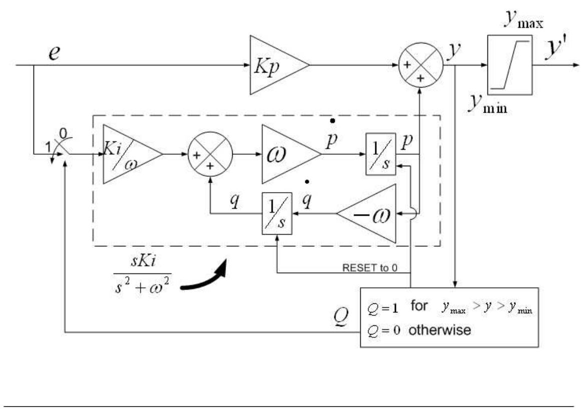

Fig. 9. PR controller with anti-windup scheme of ‘Method 1’.

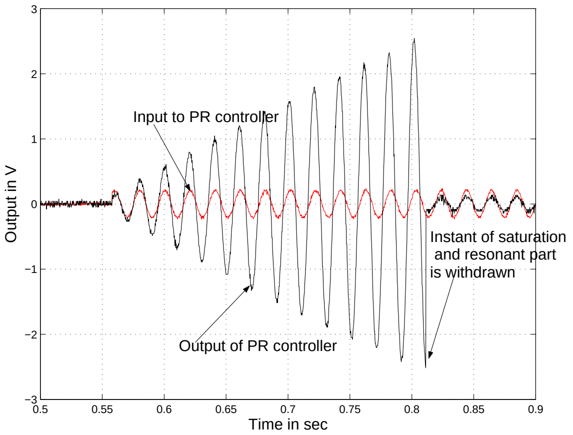

Fig. 10. Output of a PR controller implemented in FPGA with anti-windup scheme using ‘ Method 1’. Here $K_p = 0.8$, $K_i = 125$, and $\omega = 314$ rad/sec. Here $Y_{max} = 2.5V$ and $Y_{min} = -2.5V$.

the error is very large and hence the contribution from proportional part becomes significantly high then entire controller output should be forced to zero.

C. Method 2

The second scheme shown in Fig. 11 uses the difference of actual output y and saturated output $y^{'}$ to be used as feed back signal through a gain K_{lim}. As the saturated output $y^{'}$ would be a clipped sinusoidal wave so feed back through K_{lim} would contain higher order harmonics. But the resonant part of the controller provides low gain other than the resonant frequency. Therefore it can be expected that the output y would not be distorted. However higher value of K_{lim} ensures that output y is small. So, value of K_{lim} can be selected based upon desired range of y for maximum error input e.

If value of K_{lim} is high then the magnitudes of harmonics also get increased which in turn distorts the output y. Again for a lower magnitude error input e the saturated output $y^{'}$ follows the output y for longer time.

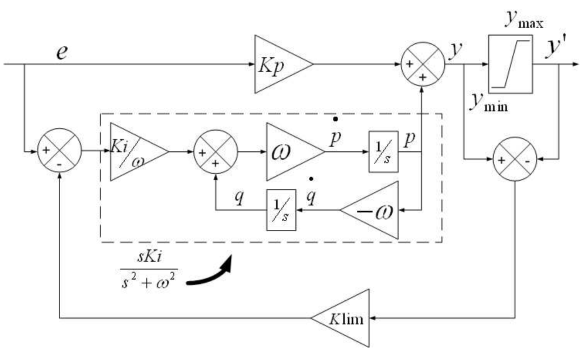

Fig. 11. PR controller with anti-windup scheme of ‘Method 2’.

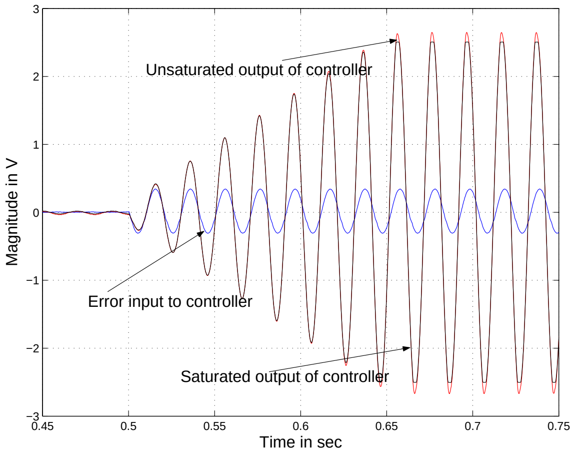

Fig. 12. Output of a PR controller implemented in FPGA with anti-windup scheme using ‘ Method 2’. Here $K_p = 0.8$, $K_i = 125$, $K_{lim} = 10$ and $\omega = 314$ rad/sec. Here $Y_{max} = 2.5V$ and $Y_{min} = -2.5V$.

The result shown in Fig. 12 shows that the saturated output is almost following the unsaturated output. Here the input value taken is $0.1pu$. So for lower input value the output y can be used as final output but for higher input value output y becomes significantly distorted. Also if the saturation level chosen are smaller such as 10% of rated voltage then level of distortion would increase.

IV. Experimental Result

To verify the operation of PR controller a 3 phase 4 wire test setup of inverter with resistive-inductive load was chosen. The neutral point of the load was connected to the dc bus mid point. A current command of 4A r.m.s. was given as the input to the current controller. The overall control strategy contains three individual current controller for each phase. The result in Fig. 13 shows that the controller is capable of tracking the reference current. During the experiment DC bus was held constant at 600V and the load was series connection of of 35Ω resistance and $84mH$ inductance in each phase .

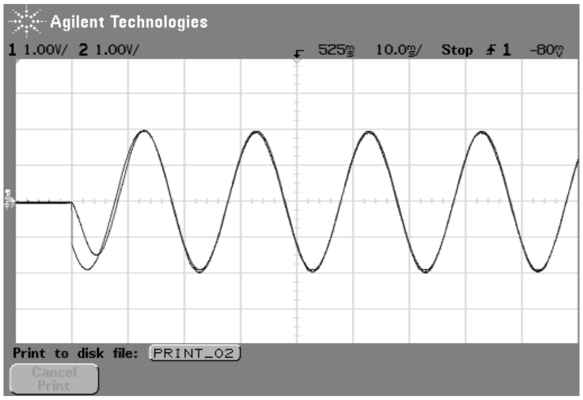

Fig. 13. Phase current tracking using the proposed PR controller. Here $K_p = 1.0$, $K_i = 1000$, $K_{lim} = 10$ and $\omega = 314$ rad/sec. Here $Y_{max} = 5.0V$ and $Y_{min} = -5.0V$.

V. Conclusion

The present paper tries to address the windup or rollover problem that can occur in a PR controller. Two anti-windup schemes have been proposed for PR controller on the basis of the understanding on anti-windup schemes for PI controller. It is shown that a direct extension of the anti-windup mechanism is not feasible. The proposed methods for PR anti-windup implementation indicate that it is possible to obtain an approximate anti-windup performance. The controller logic has been implemented in hardware and results are shown.

References

[1] C. Bohn and D.P. Atherton, *"An analysis package comparing pid anti-windup strategies,"* IEEE systems magazine, 15(2):34 - 40, April 1995.

[2] H. B. Shin, *"New anti-windup PI controller for variable speed motor drives,"* IEEE Transactions on Industrial Electronics, Vol. 45, No. 3, Page(s) 445 - 450, June 1998.

[3] A. Visioli, *"Modified anti-windup schemes for pid controllers,"* IEE Control Theory and Application, 150(1): 49 - 54, January 2003.

[4] M. P. Kazmierkowski and L. Malesani, *"Current control techniques for three-phase voltage-source PWM converters: a survey,"* in Industrial Electronics, IEEE Transactions on, Volume 45, Issue 5, Oct 1998 Page(s) 691 - 703.

[5] A. Scottedward Hodel and C. E. Hall, *"Variable-structure pid control to prevent integrator windup,"* IEEE Transactions on Industrial Electronics, 48(2):442 - 451, April 2001.

[6] D. N. Zmood and D. G. Holmes, *"Stationary frame current regulation of PWM inverters with zero steady-state error,"* Power Electronics, IEEE Transactions on Volume 18, Issue 3, May 2003 Page(s):814 - 822.

[7] Y. Xiaoming and W. Merk, *"The non-ideal generalised amplitude integrator (NGAI): interpretation, implementation and applications,"* Power Electronics Specialist Conference, 2001. PESC. 2001 IEEE 32nd Annual Volume 4, 17-21 June 2001 Pages(s): 1857 - 1861.

Anirban Ghoshal received B.E. degree in Electrical Engineering from the Bengal Engineering College,Shibpur and M.Tech degree from the Indian Institute of Technology, Delhi. Currently he is working towards his Phd at the Indian Institute of Science, Bangalore. His areas of interests include power electronics, distributed generation and drives.

Vinod John received B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Madras, M.S.E.E. degree from the University of Minnesota, Minneapolis, and Ph.D. from the University of Wisconsin, Madison. He has worked in research and development positions at GE Global Research, Niskayuna, NY and Northern Power, VT. He is currently working as an Assistant Professor at the Indian Institute of Science, Bangalore. His primary areas of interests are in power electronics and distributed generation, power quality, high power converters and motor drives.