NOVA

University of Newcastle Research Online

nova.newcastle.edu.au

Betz, R. E.; Gay, D. "Local vector current control and current clamp minimisation".
Published in Proceedings of the 2016 18th European Conference on Power Electronics
and Applications (EPE'16 ECCE Europe) (Karlsruhe, Germany, 5-8 December, 2016)
(2016)

Available from: <http://dx.doi.org/10.1109/EPE.2016.7695586>

© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting /republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Accessed from: <http://hdl.handle.net/1959.13/1345802>

Local Vector Current Control and Current Clamp Minimisation

R.E. Betz¹, D. Gay²

School of Electrical Engineering and Computer Science
University of Newcastle, Australia, 2308

email:¹Robert.Betz@newcastle.edu.au

²David.Gay@newcastle.edu.au

Keywords

<<converter control>>, <<control methods for electrical systems>>,<<digital control>>,<<non-linear control>>

Abstract

This paper presents the Local Vector Control (LVC) algorithm that is applicable to power electronic systems. This controller does not differentiate between linear or non-linear systems and can operate, without change, in stationary or rotating reference frames. The performance of this current controller with particular emphasis on current clamp minimisation will be the paper's focus. A comparison between the LVC, PI and predictive dead-beat current controllers (PCC) is presented.

Introduction

The current controller is undoubtedly the most important control loop in most power electronic systems, whether they be a variable speed drive or a grid connected system. Generally speaking, if the current controller works well then the outer control loops in a hierarchical control system will also work very well.

Considerable research has been carried out over the years on power electronic converter current controllers. The references are too numerous to list here. However, most of the research has concentrated on the following types of controllers:

- Synchronous Proportional Integral (SPI) controllers.
- Proportional Resonant (PR) controllers [13].
- Predictive current controllers (PCC) [4].
- Finite state Model Predictive Control based Current Control (MPCCC) [2].

The SPI controller for AC systems is always operated in a rotating reference frame because this type of controller cannot track a sine wave with zero steady state error. The PR, PCC and MPCCC are usually operated in stationary reference frames. PR controllers are particularly useful for grid connected applications since the resonant frequency of the controller can be set at the grid frequency (which is essentially constant). It can be shown that the PR controller has a direct connection to the SPI controller [12]. The PCC is usually used in electric machine drive applications, or any application where the load can be modeled as a virtual induction machine. MPCCC can be a current control strategy. It sometimes is implemented in such a way that it bypasses the current control stage. For example, the switching of the converter can directly control the torque, and the current is only implicitly controlled.

The Local Vector Control (LVC) strategy that is the subject of this paper has been around for a long time with little practical implementation [7, 9–11]. This paper is partially motivated by curiosity as to whether LVC can offer any advantages in power electronic applications over the well known above-mentioned algorithms. Its use of derivatives superficially indicates that it would have challenges working in a power electronic environment. LVC has some similarities to sliding mode control, in that it is based on state-time trajectories, but that is where the similarities end. The potential advantages of LVC are that it can operate, without structural change, in both the synchronous and stationary reference frames,

Fig. 1: Block diagram of the LVC structure.

and is high bandwidth. It is that latter property that potentially can be exploited for current clamp minimisation. Current clamp becomes an issue when the output frequency and voltage from an inverter is low. This issue has been investigated over many years [6, 8], often involving extra hardware in the inverter. Feedback only solutions can potentially minimise current clamp, but cannot eliminate it.

Figure 1 shows the block diagram of the structure of the LVC [9]. This controller shows similarities to a classic high gain controller which works by forming a virtual inverse of the plant [3]. High gain control is essentially an open loop control strategy, whereas LVC has a state resetting function between two parts of the controller model which provides implicit feedback.

Analysis of this controller is beyond the scope of this paper. The interested reader can refer to [7, 9–11]. The term Local Vector Controller arises from the fact that the control is only concerned with the present state of the system S , its probable next state without control, and its preferred next state. The control is derived from these quantities, at an instant of time, resetting the Behaviour Reference System (BRS) system R . At the next instant of time the present state of S , the next state of S without control and the preferred state of S will be different. Therefore the control decisions are made through time local state-time vectors being fed into system R . In a traditional feedback control system the plant dynamics S are augmented by the controller dynamics. The state of the overall system is effectively a time integral of the state-time trajectories of the overall system dynamics. Therefore the systems response is a combination of previous history plus what is occurring at the current instant.

One of the key variables that will be used to compare the different algorithms in this paper is the current clamping distortion under low frequency and low amplitude conditions. The high gain characteristic of LVC indicates that it should be able to minimise the non-linear distortion produced by the dead-time of the converter.

The LVC Algorithm

Before presenting the details of the digital implementation of the LVC algorithm, we shall first understand heuristically how it works.

The plant S in Figure 1 has linear or non-linear dynamics that control the trajectories of its output derivatives. The Behaviour Reference System (BRS), i.e. block R in Figure 1, has the dynamics that we want S to have. Around the BRS model is a trajectory system T and a simple feedforward proportional controller. The purpose of this part of the controller is to make the desired dynamic system R follow the desired control reference input. System S produces a state variable (to be detailed below) which is input to the BRS that in turn produces the input to S to move its state trajectories to those of R . The next loop of the controller then samples the state trajectories of S and then updates the states of R to them. The process then repeats.

Figure 2 shows the plant S trajectory and the R system trajectory. Point 1 is at time $t(k)$ and at this point $(y_S(k), \dot{y}_S(k), t(k)) = (y_R(k), \dot{y}_R(k), t(k))$. Point 2 is on the next plane at time $t(k+1)$, this state-time point being $(y_S(k+1), \dot{y}_S(k+1), t(k+1))$. Point 3 is where R evolves to if its states are set to S 's at $t(k)$ and it is being driven by $u_R(k)$ from the trajectory system controller.

If no control is applied then S will propagate to Point 2 at $t(k+1)$ from Point 1 at $t(k)$. System R starts from the same Point 1 as it has been reset to this by the state resetting process, and propagates to

Fig. 2: Evolution of the state-time trajectories of S and R .

Point 3 at $t(k+1)$. The dashed trajectory is the desired trajectory, and the control is applied so that the trajectory of S is mapped onto that of $R \forall t$.

In order to understand how this process works a few equations must be introduced. LVC is capable of controlling n th order dynamic plant models, but we shall only consider 2nd order plants of the form:

$$\ddot{y}_S = f_S(\dot{y}_S, y_S, t) + g_S(\dot{y}_S, y_S, t)u_S \quad (1)$$

where f_S, g_S can be non-linear functions. We can write the BRS in a similar way to 1. Using $\tilde{f}_{S,R} = f_{S,R}(\dot{y}_{S,R}, y_{S,R}, t)$ and $\tilde{g}_{S,R} = g_{S,R}(\dot{y}_{S,R}, y_{S,R}, t)$ we can write the S and R models as:

$$\ddot{y}_{S,R} = \tilde{f}_{S,R} + \tilde{g}_{S,R}u_{S,R} \quad (2)$$

Let $\hat{u}_S$ be the input that generates $\hat{y}_S$ from the system. That is:

$$\hat{y}_S = \tilde{f}_S + \tilde{g}_S \hat{u}_S \quad (3)$$

Let us initially consider the algorithm in continuous time. The LVC algorithm works by making the states of the system S follow the states of the system R . This means that:

$$\ddot{y}_S = \ddot{y}_R \forall t \quad (4)$$

and consequently:

$$\ddot{y}_R = \tilde{f}_S + \tilde{g}_S u_S \quad (5)$$

Rearranging this gives:

$$u_S = \frac{\ddot{y}_R - \tilde{f}_S}{\tilde{g}_S} \quad (6)$$

Similarly (3) can be rearranged to give:

$$\hat{u}_S = \frac{\ddot{y}_S - \tilde{f}_S}{\tilde{g}_S} \quad (7)$$

One can subtract these two expressions to get the difference between the two controls. This expression can then be rearranged to give the augmented control to the S system:

$$u_S - \hat{u}_S = \frac{\ddot{y}_R - \ddot{y}_S}{\tilde{g}_S} \quad (8)$$

$$\Rightarrow u_S = \hat{u}_S + \frac{\ddot{y}_R - \ddot{y}_S}{\tilde{g}_S} \quad (9)$$

where the $\ddot{y}_R$ value required is determined from the reference system R as:

$$\ddot{y}_R = \tilde{f}_R + \tilde{g}_R u_R \quad (10)$$

If this u_S is applied to the system S then its 2nd derivative follows that of the reference system, and therefore it behaves dynamically as if it is the reference system. This can be simply seen by substituting the above control (9) into the system equation (3):

$$\ddot{y}_S = \tilde{f}_S + \tilde{g}_S \left(\hat{u}_S + \frac{\ddot{y}_R - \ddot{y}_S}{\tilde{g}_S} \right) \quad (11)$$

$$= \underbrace{\tilde{f}_S + \tilde{g}_S \hat{u}_S}_{\ddot{y}_S} + \ddot{y}_R - \ddot{y}_S \quad (12)$$

$$\therefore \ddot{y}_S = \ddot{y}_R \quad (13)$$

which implies that the dynamics of S are now the same as those of R . This means that the state trajectories of S follow those of R . Therefore if the output of R converges to the reference, then so does the output y_S of S .

Discrete Implementation

In order to implement the algorithm in a microcontroller we need to implement it in discrete time. The details of this process are presented in [9, 10]. An abridged version is presented below. As above, a 2nd order R system will assumed. Therefore the R system model is:

$$\ddot{y}_R + 2\zeta\omega_n\dot{y}_R + \omega_n^2 y_R = \omega_n^2 u_R \quad (14)$$

which can be written as two first order differential equations using the change of variable $z_1 = y_R$ and $z_2 = \dot{y}_R$ as:

$$\dot{z}_1 = z_2 \quad (15)$$

$$\dot{z}_2 + 2\zeta\omega_n z_2 + \omega_n^2 z_1 = \omega_n^2 u_R \quad (16)$$

These expressions can be discretized using a simple Euler approximation to the derivatives with Δ being the discretization interval to give:

$$\frac{z_1(k) - z_1(k-1)}{\Delta} = z_2(k-1) \quad (17)$$

$$\begin{aligned} \frac{z_2(k) - z_2(k-1)}{\Delta} &= \omega_n^2 u_R(k-1) - 2\zeta\omega_n z_2(k-1) \\ &\quad - \omega_n^2 z_1(k-1) \end{aligned} \quad (18)$$

Simple rearrangement gives expressions suitable for simulation:

$$z_1(k) = \Delta z_2(k-1) + z_1(k-1) \quad (19)$$

$$\begin{aligned} z_2(k) &= \Delta[\omega_n^2 u_R(k-1) - 2\zeta\omega_n z_2(k-1) \\ &\quad - \omega_n^2 z_1(k-1)] + z_2(k-1) \end{aligned} \quad (20)$$

These equations can be rewritten in the more familiar y_R notation as:

$$y_R(k) = \Delta \dot{y}_R(k-1) + y_R(k-1) \quad (21)$$

$$\begin{aligned} \dot{y}_R(k) &= \Delta[\omega_n^2 u_R(k-1) - 2\zeta\omega_n \dot{y}_R(k-1) \\ &\quad - \omega_n^2 y_R(k-1)] + \dot{y}_R(k-1) \end{aligned} \quad (22)$$

Fig. 3: Basic control timing offering

Remark 1 In most cases the behaviour reference model is a “nice” system such as the 2nd order system above. Instead of a Euler discretization one can use the exact discrete solution to such systems to give the y_R and $\dot{y}_R$ values at the required discrete times. ■

Notwithstanding Remark 1, Euler approximations are almost always required for the S system as one generally does not know its dynamics and the state variables are usually not measurable. Therefore the derivatives of the system have to be inferred from the output. One can see from (9) that the 2nd derivatives are key to this algorithm, therefore the discrete evaluation of this is very important. We know, that the Euler discrete approximation to this is:

$$\ddot{y}(k) \approx \frac{\dot{y}(k) - \dot{y}(k-1)}{\Delta} \quad (23)$$

In the case of the R system we already have expressions for the 1st derivatives, and in the case of the S system one can use the standard Euler approximation

$$\dot{y}_S(k) = \frac{y_S(k) - y_S(k-1)}{\Delta} \quad (24)$$

for the 1st derivatives.

The following step-by-step procedure for the algorithm will assume that the algorithm is in steady state execution – that is the initial start-up procedures are not running.

Remark 2 The steps below assume that computational delay is being handled in the algorithm. This means that the algorithm follows the pattern shown in Figure 3. Step 1 will start at time $(k-1)\Delta$ where Δ is the control interval. The sampling interval is denoted as δ . The u_{setpoint} variable is desired output of the plant S . ■

Step 1 ($k-1$)

- Apply control from previous control evaluation.
- Sample the output of the system S as $y_S(k-1)$.

Step 2 ($k-0.5$)

This is the part of the control cycle where most of the work is done in terms of computing the next control to be applied at (k) .

- Sample and save the system S output – $y_S(k-0.5)$. At this point we have the array of $Y_S = [y_S(k-1), y_S(k-0.5)]$. These values are used for the future predictions of the system S output.
- Calculate the states of the S system at time k and $k+1$:

$$\hat{y}_S(k) = 2y_S(k-0.5) - y_S(k-1) \quad (25)$$

$$\hat{\dot{y}}_S(k) = \frac{y_S(k-0.5) - y_S(k-1)}{\delta} \quad (26)$$

$$\hat{\dot{y}}_S(k+1) = \hat{\dot{y}}_S(k) \quad (27)$$

- Reset BRS R to the states of the plant S :

$$y_R(k) = \hat{y}_S(k) \quad (28)$$

$$\dot{y}_R(k) = \hat{\dot{y}}_S(k) \quad (29)$$

- Project R forward to $k + 1$ using the current control input $u_R(k - 1)$ into R :

$$\hat{y}_R(k + 0.5) = \delta \omega_n [\omega_n u_R(k - 1) - 2\zeta \dot{y}_R(k) - \omega_n y_R(k)] + \dot{y}_R(k) \quad (30)$$

$$y_R(k + 1) = \Delta \dot{y}_R(k + 0.5) + y_R(k) \quad (31)$$

- Calculate the new control to be applied to the R system so that it will track the T system output just calculated. The expression to do this is:

$$y_T(k + 1) = u_{\text{setpoint}} \quad (32)$$

$$u_R(k) = u_{\text{setpoint}} + K_R(y_T(k + 1) - y_R(k + 1)) \quad (33)$$

- Use the new $u_R(k)$ control to calculate the future $\frac{d}{dt}$ of the R system:

$$\dot{y}_R(k + 1) = \Delta \omega_n [\omega_n u_R(k) - 2\zeta \hat{y}_S(k) - \omega_n \hat{y}_S(k)] + \hat{\dot{y}}_S(k)$$

- Finally calculate the control to be applied at time k using:

$$u_S(k) = u_S(k - 1) + \frac{[\dot{y}_R(k + 1) - \hat{\dot{y}}_S(k + 1)]}{K\Delta} \quad (34)$$

where the $K >$ the S system gain.

- Limit the control magnitude to be within the limits
- Update the S and R system inputs:

$$u_S(k - 1) = u_S(k)$$

$$u_R(k - 1) = u_R(k)$$

- Go back to Step 1.

Remark 3 In the above implementation of the LVC algorithm some very crude approximations have been used, for example (27) is used to estimate $\hat{y}_S(k + 1)$. It has been found through experience that the crude approximations often work as well, and in some cases better, than the “better” approximations. ■

Simulation Results

In order to test the algorithm, a Wolfram SystemModeler® Modelica simulation was set up for an RL load. The LVC Modelica simulation block diagram appears in Figure 4. The inverter has a dead-time of $8\mu\text{sec}$ and the control rate is 5kHz , the PWM is generated using a sinusoidal triangular carrier wave, and the DC Link voltage 586V . The RL load has the parameters of $R = 0.5\Omega$ and $L = 5\text{mH}$. Included in the simulation are 12 bit A/D converters with $\pm 30\text{A}$ range. The currents have uncorrelated additive pseudo random noise of $\pm 0.05\text{A}$ with is approximately the bottom three bits of the A/D converters resolution. The noise was added to test the robustness of the derivative based algorithms (LVC and PCC). The simulations for the PI and PCC controllers have a very similar structure.

The current references are sinusoidal waveforms at 1Hz frequency and an amplitude of 5Amp to 0.5seconds , and then jumping to 10Amp . This frequency and amplitude were chosen to maximise the effects of current clamp (as a result of the dead-time) so that the comparative performance of the algorithms was clear. The step in the current reference tests the transient performance of the algorithms. The LVC R model (the behaviour reference system) is a 2nd order system with $\omega_n = 10000\text{r/s}$ and a damping factor of 1.0 , and the LVC gain factor $K = 10 \times 10^6$.

Fig. 4: SystemModeler® block diagram of the rotating frame LVC current controller with an RL load.

Fig. 5: Simulation: LVC a-phase current waveform. Right waveform is a close-up of the current clamp.

The performance of the rotating frame LVC controller is shown in Figure 5. One can see that the current overall is accurately tracking the reference, with excellent transient performance at $t = 0.5s$. The zoomed view of the current clamp shows that its duration is approximately 1.5 msec, or 7 control intervals. Notice that there is some “noise” on the current. This is to be expected from a controller that is dependent on derivatives, especially when there is 3 bits of noise on the current measurements.

The same reference waveforms were applied to the well known synchronous reference frame PI controller. It was digitally implemented, and tuned using the Zeigler-Nichols step method. The parameters were: $K_p = 11.25$ and $T_i = 0.000992$. The performance of this controller is shown in Figure 6. As can be seen there is substantial current clamp. The transient performance of the controller is similar to the LVC controller. The zoomed view of the clamp region shows that the clamp is about 14 msec or 70 control intervals. This is 1000% longer than that for the LVC controller. If this current is fed into an electrical machine substantial torque pulsations result.

The third controller investigated was the Predictive Current Controller (PCC). The algorithm implemented was the same as documented in [1, 5]. Similarly to the LVC controller this is a high bandwidth algorithm based on using derivatives. The results appear in Figure 7. The overall response is similar to LVC, but the current clamp is twice as long at 3 msec. The current clamp is measured from the point where the reference crosses through zero at 0.25 secs to when the current waveform rejoins the reference.

To make the relative differences between the algorithms performance clear comparative plots appear in Figure 8. As can be seen from this figure LVC performs significantly better than the PI controller, and

Fig. 6: Simulation: PI synchronous frame a-phase current waveform. Right waveform is a close-up of the current clamp.

Fig. 7: Simulation: PCC stationary frame a-phase current waveform. Right waveform is a close-up of the current clamp.

moderately better than PCC with respect to current clamp. With respect to transient performance, LVC compares very favourably with both PI and PCC.

Experimental Results

In order to validate the simulation results an experimental system based on a Semikron[®] Cube and a DSP microcontroller was used. The LVC algorithm implemented was identical to that used in the Modelica simulations, although the K was slightly larger than in the simulations.

Only two algorithms were programmed and tested – the LVC and the PI controller. The PI algorithm used the same tuning as in the simulation. The experimental load was identical to the simulation load. Where possible all the parameters were kept the same between simulation and experiment to allow direct comparison.

Figures 9(a) and (b) are a comparison of the LVC experimental and simulation results. As can be seen the experimental and simulation plots are very close. This shows that the simulation is a good model of the experimental system.

Figures 9(c) and (d) compare the experimental results for the LVC and PI algorithms for current clamping and transient performance. One can see that the PI controller has a clamp time of approximately 10msec, whereas the LVC controller is of the order of 2.5msec (under a worst case assessment of the figure). In fact it could be argued that the LVC has a current clamp time of approximately 1msec. Either way, the LVC algorithm is a considerable improvement on the PI controller with respect to current clamping. The transient performance shown in Figure 9(d) is approximately the same for both algorithms.

Fig. 8: Simulation comparison of LVC, PI Synchronous and PCC current clamp and transient performance.

Conclusions

This paper has presented simulation and experimental results demonstrating the performance of the LVC algorithm. Comparisons have been made with a synchronous reference frame PI controller and a Predictive Current Control dead-beat controller. It has been shown that:

- LVC minimises the current clamp issue in drive systems as compared to PI and PCC.
- LVC gives approximately the same transient current performance as PI and PCC.
- Despite the fact that derivatives of the measurements are at the heart of the algorithm, it is able to function in the presence of substantial noise on the feedback measurements.

This paper has only considered one aspect of the performance of this algorithm by tackling a normally difficult, it not esoteric problem of dead-time elimination. Although not shown in this paper, the same LVC controller can also operate in a stationary reference frame. This versatility together with the results in this paper suggest that LVC deserves more investigation to determine its usefulness in power electronic applications.

References

- [1] R.E. Betz, B.J. Cook, and S.J. Henriksen. A digital current controller for three phase voltage source inverters. *Conference Record of the IEEE-IAS Annual Meeting*, pages 722–729, New Orleans, Oct. 1997.
- [2] P. Cortes, M.P. Kazmierkowski, R.M. Kennel, D.E. Quevedo, and J. Rodriguez. Predictive control in power electronics and drives. *Industrial Electronics, IEEE Transactions on*, 55(12):4312–4324, dec. 2008.
- [3] Graham C. Goodwin, Stefan F. Grabe, and Mario E. Salgado. *Control System Design*. Prentice Hall, 2001.
- [4] Soren J. Henriksen. Digital predictive current control of induction machines. Master's thesis, Dept, Elec and Comp Eng, University of Newcastle, Australia, 2001.
- [5] Soren J. Henriksen, Robert E. Betz, and Brian J. Cook. Digital hardware implementation of a current controller for IM variable-speed drives. *IEEE Transactions on Industry Applications*, 35(5):1021–1029, Sept/Oct 1999.

Fig. 9: Simulation and experiment: Comparison of LVC and the PI controller – current clamp and transient performance.

- [6] R. P. Joshi and B. K. Bose. Base/gate drive suppression of inactive power device of a voltage fed inverter and precision synthesis of ac voltage and dc link current waves. In *IEEE IECON Conference Record*, pages 1034–1040, 1990.
- [7] I.M.Y. Mareels, H.B. Penfold, and R.J. Evans. Controlling nonlinear systems via euler approximation. *Automatica*, 28(4):681–696, July 1992.
- [8] Y. Murai, A. Riyanto, H. Nakamura, and K. Matsui. Pwm strategy for high frequency carrier inverters eliminating current clamps during switching dead-time. In *Industry Applications Society Annual Meeting, 1992., Conference Record of the 1992 IEEE*, pages 317–322 vol.1, Oct 1992.
- [9] H.B. Penfold. *Nonlinear Control: An Alternative Perspective*. PhD thesis, Electrical and Computer Engineering, University of Newcastle, Australia, 1988.
- [10] H.B Penfold and R.J. Evans. Control algorithm for unknown time-varying systems. *Int. Journal of Control*, 50:13–32, 1989.
- [11] H.B. Penfold, I.M.Y. Mareels, and R.J. Evans. Adaptively controlling non-linear systems using trajectory approximations. *Int. J. Adaptive Control and Signal Processing, Special issue on non-linear control*, 6(4):395–411, July 1992.
- [12] M. Teodorescu, R. Liserre and P. Rodriguez. *Grid Converters for Photovoltaic and Wind Power Systems, First Edition*. Wiley, 2011.
- [13] R. Teodorescu, F. Blaabjerg, M. Liserre, and P.C. Loh. Proportional-resonant controllers and filters for grid-connected voltage-source converters. *Electric Power Applications, IEE Proceedings -*, 153(5):750–762, september 2006.