# Lesson 3

# Digital Control of Three-Phase DC/AC Converters: Current Control Techniques

## Linear Current Control



## Current Sensing

- Current sensing is normally not a troublesome part of the control design. Several solutions can be adopted:
  - resistive shunts;
  - current transformers;
  - LEM (Hall effect based) sensors.
- In DC/DC converter applications, resistive shunts or current transformers, if isolated sensing is required, are the preferred, cost-driven choices.

## Current Sensing

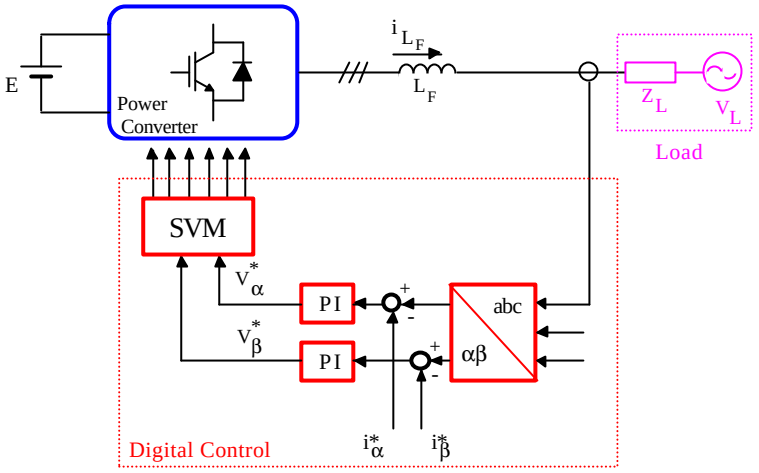
- In VSI's applications (such as drives, rectifiers, UPS's), the cost of three (or two if the system has only three wires) Hall effect sensors is normally reasonable.
- This solution greatly simplifies the design of the current control system, providing isolated and relatively large bandwidth (» 100 kHz) current sensing.
- Only simple filtering (high frequency noise) and level adjusting circuitry (to get A/D converter compatible voltage ranges) are normally required.



## Linear Current Control

- The simplest way of achieving current regulation in a DC/AC or AC/DC power converter is to employ linear regulators (usually of PI type).
- The design and digital implementation of such regulators is quite straightforward and normally effective, unless demanding static and dynamic performance specifications are given for the current loop.
- Some basic guidelines can be given for a succesfull implementation.

## Linear Current Control



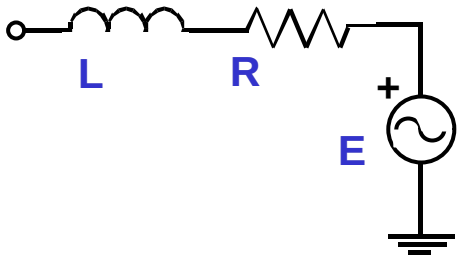

**Typical control set-up**



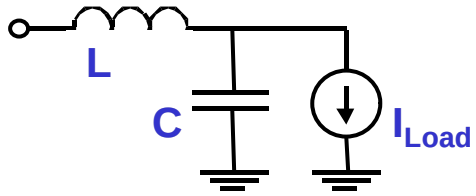
## Control Design

- The choice of the controller (and its design) in general depends on the particular converter load.
- It is very common that the dynamic behaviour of the load, in the typical frequency range of interest for current control design, is almost purely inductive.
- Possible capacitive filters (e. g. in UPS applications) normally exhibit a negligible impedance in the vicinity of the switching frequency.

## Typical Inverter Loads



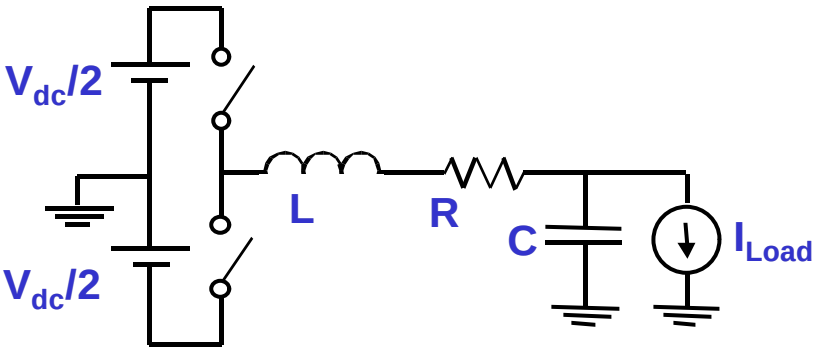

Can represent both a drive or an active filter application or rectifier (with a suitable choice of the model parameters).




Represents the typical UPS application.



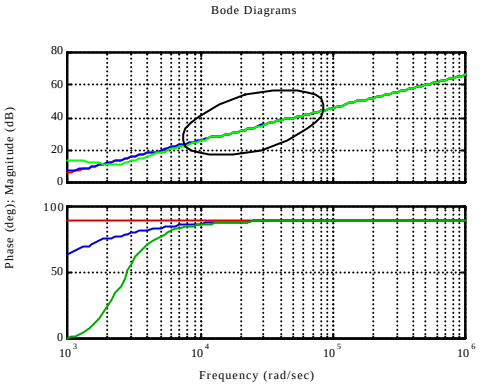
## Design Example




C = 100 mF, R = 0.1 W, L = 2 mH

$V_{dc}$ = 400 V $f_{sw}$ = 10 kHz

## Typical Inverter Loads




In the area around the possible current loop cross-over frequency the inverter load is 'seen' as almost purely inductive.
The controller design can be very simple.

- L - C load (UPS)
- R - L load (drives)
- Purely inductive load



## Control Design

- The typically adopted controller structure is the proportional-integral (PI).
- The design of the controller parameters (proportional and integral gains) is straightforward in the continous time domain.
- The discretization of the controller can be easily performed as the final step, according to one of the various integration methods (e.g. Euler or trapezoidal integration).
- There is normally no reason to consider more sofisticated approaches.

## Control Design

A simple procedure to design a PI regulator can be the following (continuous time domain):

- Set the proportional gain $k_p$ to:

  $$k_p = 2 \cdot \omega_{cr} \cdot L / V_{dc}$$

  to get the desired cross-over frequency $\omega_{cr}$;
- Set the integral gain $k_i$ to:

  $$k_i = k_p \cdot \omega_{cr} / \tan(m_f)$$

  so as to get the desired phase margin $m_f$.

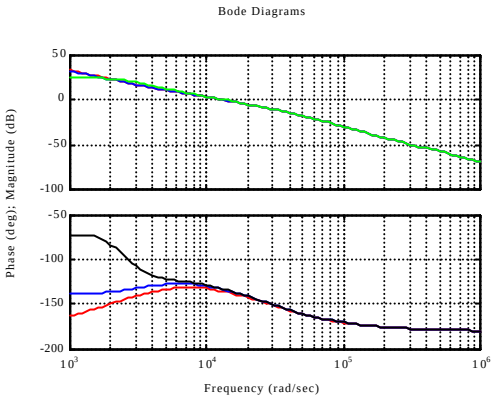
The gain of the power converter is assumed to be $V_{dc}/2$ (no PWM delay effect) and the load to be purely inductive. Transducer gains are not taken into account.



## Control Design

- It is important to take into account, even at this early stage of the design, that the final controller is going to be digitally implemented.
- In particular, the delay of the sample and hold (@ T/2, where T is the sampling period) must be considered designing the continuous time controller.
- The required phase margin has to be oversized to cope with such delay. A rule of thumb is to increase $m_f$ by at least 20° (for a control cross-over frequency about a decade lower than the sampling one).

# Control Design




**Open loop gain for the considered example with the suggested design.**
**Almost the same crossover frequency and phase margin is achieved.**

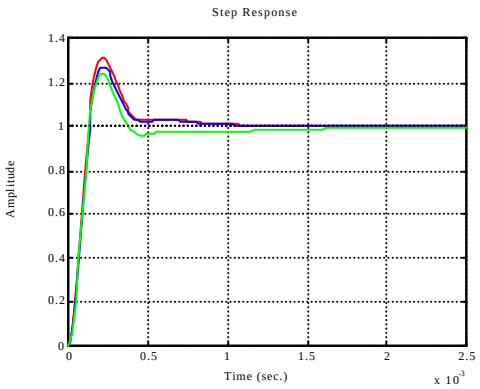
**L - C load (UPS)**

**R - L load (drives)**

**Purely inductive load**



# Control Design




**Closed loop step response for the considered example with the suggested design.**
**Almost the same performance is achieved in the three cases.**

**L - C load (UPS)**

**R - L load (drives)**

**Purely inductive load**

## Control Design

- The continuous time controller must be turned into an equivalent dicrete time controller;
- This can be easily done by substituting continuous integrators with discrete ones, according to any discrete time integration method;
- Using Euler's integration method, this is equivalent to set:

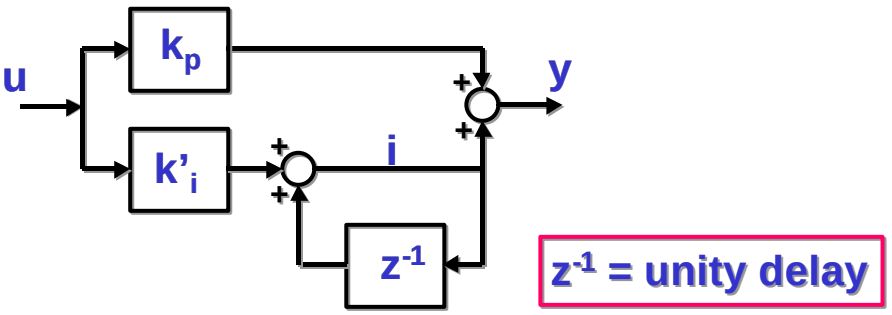
$s = (1-z^{-1})/T$ T = sampling period,

in the original controller's Laplace transform.



## Control Design

- In the case of a PI controller, this is equivalent to the following block diagram:




where $k'_I = k_i \cdot T$ (sampling period).

## Control Design

- The block diagram directly leads to the following implementation of the digital PI regulator:

$$y(k) = k_p \cdot u(k)$$
$$i(k) = i(k-1) + k'_i \cdot u(k)$$
$$y(k) = y(k) + i(k)$$

which normally requires only a few clock cycles to be calculated (2 products and 2 sums).



## Control Design

- It is important to notice that this procedure is approximated. The discretization process 'warps' the frequency response of the controller, especially for frequencies which are close to the sampling frequency.
- As a rule of thumb, at least a decade must be taken between the sampling frequency and the highest frequency of interest (normally the loop cross-over frequency).
- If the specs require a less conservative design, different synthesis strategies ought to be considered.

## Switching Frequency / Sampling Frequency

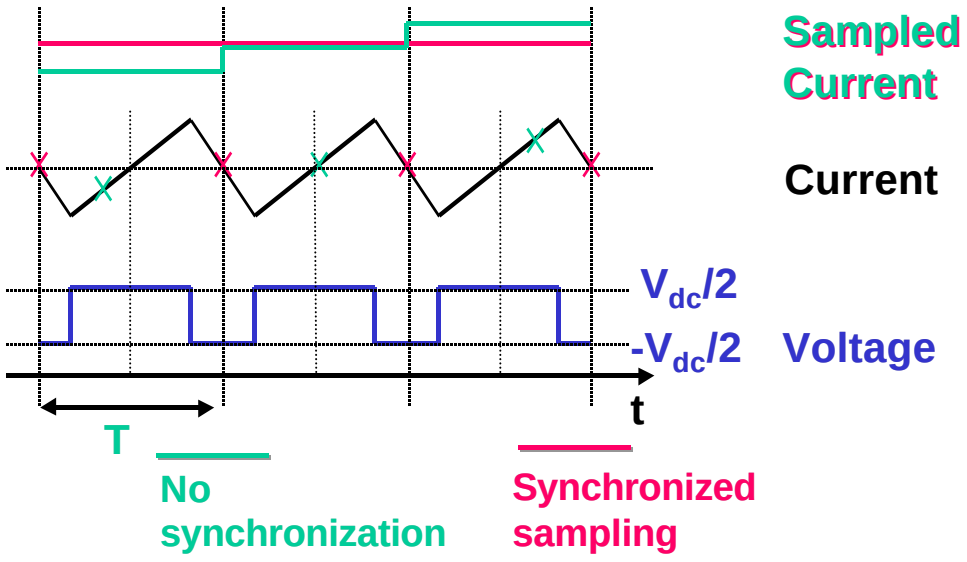
- The current waveforms in VSI's exhibit a significant high frequency harmonic content due to the PWM modulation (ripple).
- The current control loop is normally needed to control the average value of the converter's currents.
- Sampling at very high frequencies to reconstruct the complete current waveform (including the ripple) is therefore not necessary.



## Switching Frequency / Sampling Frequency

- On the other hand, sampling at high frequencies increases the achievable current loop bandwidth.
- Therefore, a simple or double sampling per modulation period is normally adopted.
- The basic idea is that sampling and switching must be synchronized to avoid errors.
- Currents are typically sampled at the beginning and in the middle of the modulation period.

## Switching Frequency / Sampling Frequency



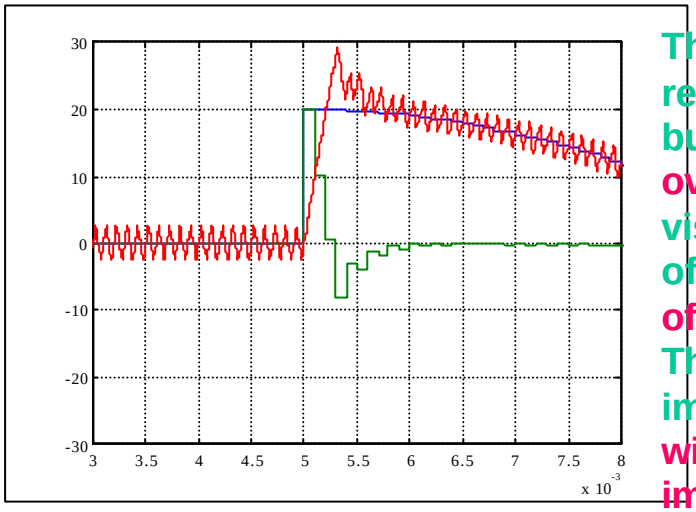



## Control Design

- Continuos time synthesis and successive discretization is a simple, but approximate, procedure, requiring particular caution in the design of the closed loop bandwidth.
- In general, the design is more conservative than direct digital design.
- Sampling frequency does not need to be much higher than the switching frequency: practically the former is set equal to or twice the latter.
- It is fundamental that sampling and switching are synchronized to avoid aliasing errors.

## Control Design

### Time domain simulation with PWM modulation



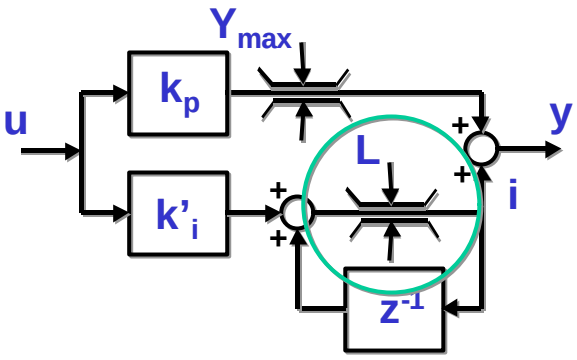

The controller's response is good but some overshoot is visible at the end of saturated mode of operation. This can be improved by anti wind-up implementation.



## Control Design

- The overshoot is due to the integral part of the controller, which gets "overcharged" during the saturated mode of operation. This has no relation with the open loop phase margin.
- As a consequence, a negative error will be needed to remove the accumulated positive error. This generates the overshoot.
- A good solution to this problem is the so-called anti wind-up action.
- This basically consists in the dynamic saturation of the integral part of the controller.

## Control Design




Modified PI controller with anti wind-up action. The control now is non-linear. Note the position of the saturation function.

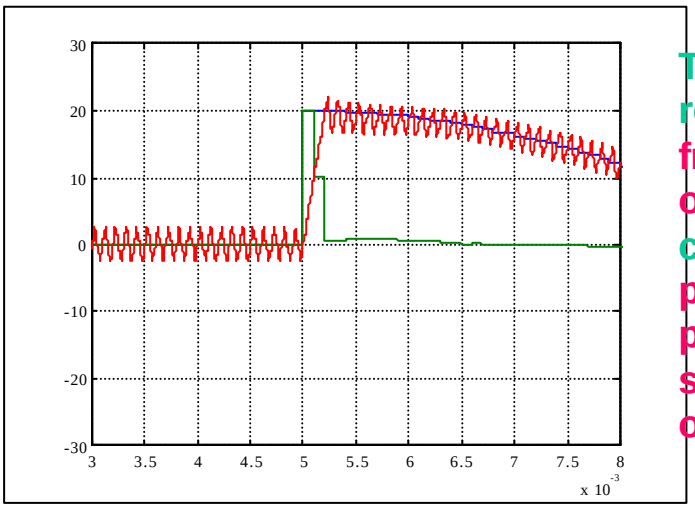
$$|L(k)| = Y_{max} - |k_p \cdot u(k)|$$

In any control cycle, the dynamic saturation limit L is calculated, so as to keep the control's output y within the limit $Y_{max}$.



## Control Design

### Time domain simulation with PWM modulation




The controller's response is now free of overshoot. The control action is purely proportional until saturation is over.

## Three Phase Implementation

- The control of three-phase VSI's requires, in general, three indipendent, but normally identical, current controllers.
- If the neutral is not connected, currents are not indipendent, and therefore it is possible to control only two currents out of three.
- It is therefore also possible to:
  - ✓ use Park's transformation to move from the a, b, c to the $\alpha$, $\beta$ fixed reference frame;
  - ✓ use d, q rotating reference frame.



## Three Phase Implementation

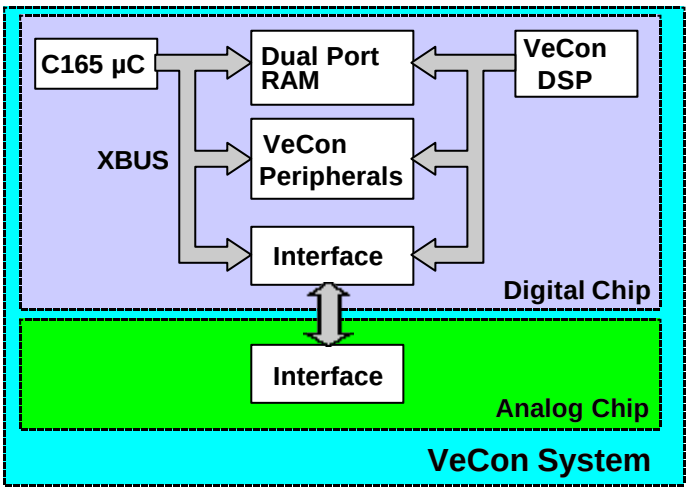
- The use of d, q rotating reference frame is quite common in adjustable speed drive applications to simplify the dynamics of the electrical machine.
- It can also be used in rectifier, UPS or active filter applications of VSI's. The advantage is the possibility of controlling very precisely the fundamental component of any variable.
- In any case, the d, q transformation modifies the dynamic equations of the system. This must be considered in the controller's design.

## Implementation Example

- In the following, an implementation example will be given with a fixed point DSP.
- The employed system is the so-called VeCon system manufactured, but not sold on the market, by a consortium of european companies, such as Siemens and ABB.
- In the early 90's, when this system first became available, it represented the first DSP system specifically designed for power converters' control tasks.



## Implementation Example




Structure of the VeCon System

## Implementation Example

**C165 microcontroller:**

- 16-bit CPU with extended instruction set (compared to 80C166)
- 16 Mbyte address space
- 2 kbyte RAM
- 32 interrupt sources (8 @ 50ns sampling rate)
- 8 peripheral event controllers
- asynchronous\synchronous interfaces



## Implementation Example

**VeCon DSP:**

- 20 MHz clock
- ALU with three 32-bit registers for arithmetic operations
- 16*16 bit multiplier for signed fractional multiplication (MACL: 100ns)
- 1024 24-bit locations internal program memory
- two data memories each with 128 16-bit locations

## Implementation Example

**Peripheral Units:**

- Position sensor evaluation
- Inverter triggering (PWM)
- Interface to Analog Chip
- Synchronous serial interface

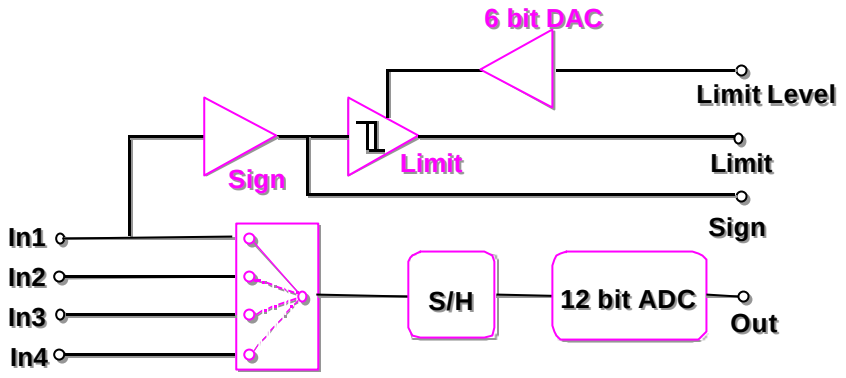


## Implementation Example

**The analog ASIC includes:**

- 3 12-bit 10µs A/D converters each with 4 multiplexed inputs
- 7 comparators to check input signals sign
- 4 hysteresis comparators to check overcurrent condition
- 6-bit D/A converters to set overcurrent limit
- 4 additional sample & hold circuits

## Implementation Example




**Analog to Digital Conversion Unit**



## Implementation Example

- The program calculates the current references in the **a**, **b** reference frame. This reduces the number of accesses to the sin(x) look-up table.
- The current control is performed on the a, b, c reference frame and, consequently, **a**, **b** - a, b, c transformation is used.
- Two PI controllers are used to implement the current regulation (3 wires).
- Post processing of the duty-cycles with third harmonic injection is used to implement SVM.

## Implementation Example

```
;       Alfa-Beta Current Reference Generation
;
        MOVE        X1,INCREM  ; Load increment
        MOVE        A1,THETA   ; Load current result in the
                               ; accumulator
        ADD   A1,X1            ; Sum THETA + INCREM and
        MOVE        THETA,A1   ; move the result in THETA
;
SIN
        MOVE        X0,THETA   ; Calculate sin
        TAB   SINTAB,7         ;
        MOVE        IA_REF,A1  ; Store it in IA_REF=Sin(THETA)
        MOVE        A1,THETA   ;
        MOVE        X1,HALF    ; THETA + 90 degrees
        ADD         A1,X1      ;
        MOVE  X0,A1            ;
        TAB   SINTAB,7         ;
        MOVE        IB_REF,A1  ; IB_REF=Sin(THETA+90)
```



## Implementation Example

```
;       Alfa-Beta -> U,V,Z Transform
;
TRSF23
        MOVE            X1,TWO_THIRDS   ;
        MOVE            Y1,IA_REF       ;
        MUL    X1,Y1    X0,SIN_60       ;
        MOVE            IU_REF,A1       ;  IU_REF = 2/3*IA_REF
        MOVE            Y0,IB_REF       ;
        MUL    X0,Y0                    ;
        MOVE            X0,A1           ;
        MUL    X0,X1    Y0,SIN_30       ;
        MOVE            X0,A1           ;
        MOVE            Y1,IU_REF       ;
        MUL    Y0,Y1                    ;
        ADD    -A1,X0                   ;
        MOVE            IV_REF,A1       ;  IV_REF = 1/SQRT(3)*IB_REF
                                                    -1/3*IA_REF
```

## Implementation Example

```
PI_U                                    ; PI phase U
     MOVE            X1,IU_ANA     ;
     MOVE            A1,IU_RF      ;
     SUBL            A1,X1         ;
     MOVE            ERROR,A1      ; Control input variable
     MOVE            Y1,KPU        ; Proportional gain
     MOVE            X1,A1         ;
     MUL   X1,Y1     X0,UCOMP      ;  Proportional part OK
     ADDL  A1,X0     X0,LIMPOS     ;  Feed-forward compensation
     LIMS  A1,X0     Y0,LIMNEG     ;
     LIMI  A1,X0     PROPU,A1      ;  Anti wind-up for the
     MOVE            A,INTEGU      ;  integral part
     MOVE            Y1,KIU        ;
     MACL  X1,Y1                   ;
     LIM   X0,Y0     X1,PROPU      ;
     ADDL  A1,X1     INTEGU,A      ;  N.B.  The move is executed
                                   ;  before the add
     MOVE            UREF,A1       ;  end PI phase U
```



## Implementation Example

```
MODULA                          ; SVM part of the program
     MOVE       Y0,UREF     ;
     MOVE       Y1,VREF     ;
     HAR3
     MOVE       Y1,A1       ;
     MOVE       THIRD,Y1    ;
;
;    Switching instants
;
     MOVE       X0,MODE_U ; Load PWM mode
     MOVE       Y1,THIRD    ; Calculate 'third harmonic'
     MOVE       Y0,TMOD_1 ; Limit to TMOD - 1 (PWM)
     MOVE       A1,UREF     ; Load voltage reference
     MOVE       X1,TMOD_2 ;
     ADD        A1,Y1       ; Load third harmonic
     PWM        A1,X1       ; Calculate instant TU1
     LIMU       A1,Y0       ; Limit TU1 to TMOD_1
     MOVE       TU1,A1      ; Store the result
```

# References


[1] L. Rossetto, L. Malesani, P. Tenti, S. Buso, A. Pollmann: "Fully Digital Control of a Three Phase UPS by VeCon Integrated Controller" IEEE Industry Applications Society (IAS) Annual Meeting, Orlando, October 8-12, 1995, pp. 2663-2669.

[2] E. Kiel and W. Schumacher, "VeCon: A High-Performance Single-Chip-Servocontroller for AC Drives", IPEC Conf. Proc., 1995, pp. 1284-1289.