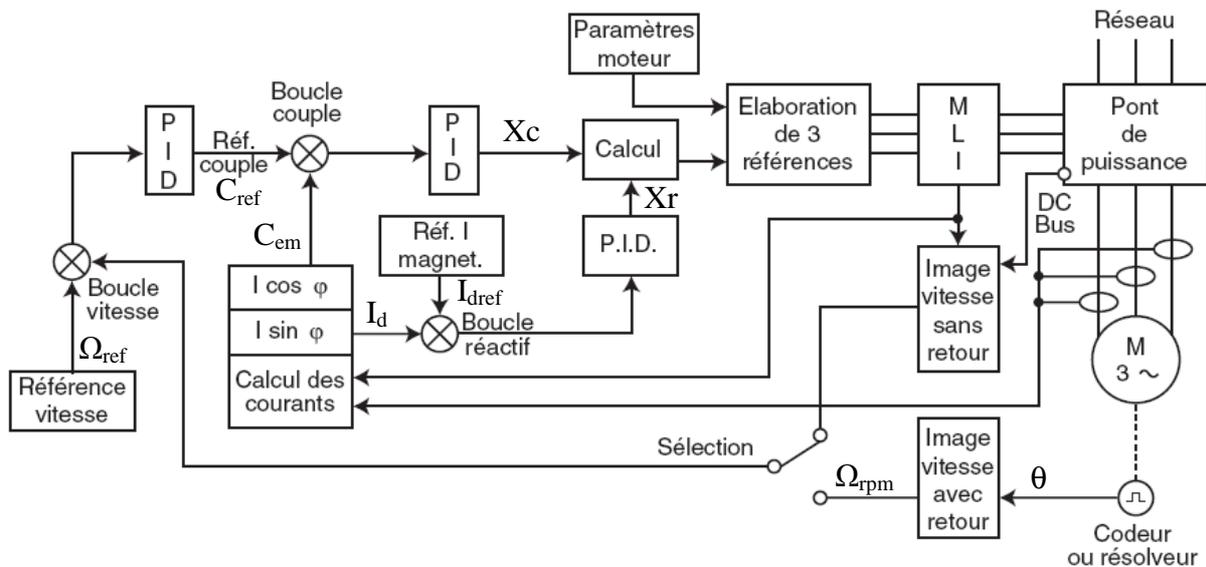


Médian TR57 : commande en temps réel

Variateur de vitesse universelle Unidrive SP

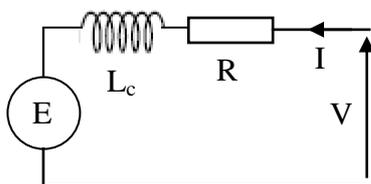
On étudie la gestion logicielle des fonctionnalités du variateur de vitesse Unidrive SP pour moteur alternatif. Le principe de contrôle illustré dans la figure ci-dessous sera limité dans cette étude au contrôle d'une machine synchronisée triphasée à aimants permanents.



Les principaux éléments physiques considérés pour cette étude sont les suivants :

- le pont de puissance alimentant le moteur est un onduleur triphasé à IGBT. La tension continue (DC Bus) en entrée de l'onduleur est $V_{bus} = 540V$ et provient d'un redresseur alimenté par le réseau 230/400V triphasé
- la vitesse est mesurée par un codeur incrémental de 1000 pas/tour. La connectique entre le codeur et le variateur est au standard RS422
- le pilotage du convertisseur est assuré par un microcontrôleur cadencé à 40 MHz. La mise à l'échelle des variables en virgule fixe n'est pas à traiter ici.
- les paramètres et consignes sont transmis au variateur par une liaison série SCI utilisant un support physique au standard RS485. La référence de vitesse peut être analogique ou transmise par la liaison série

Le modèle d'une phase de la machine synchronisée est illustré ci-dessous : l'onduleur permet d'imposer la tension V aux bornes de la machine connectée en étoile.



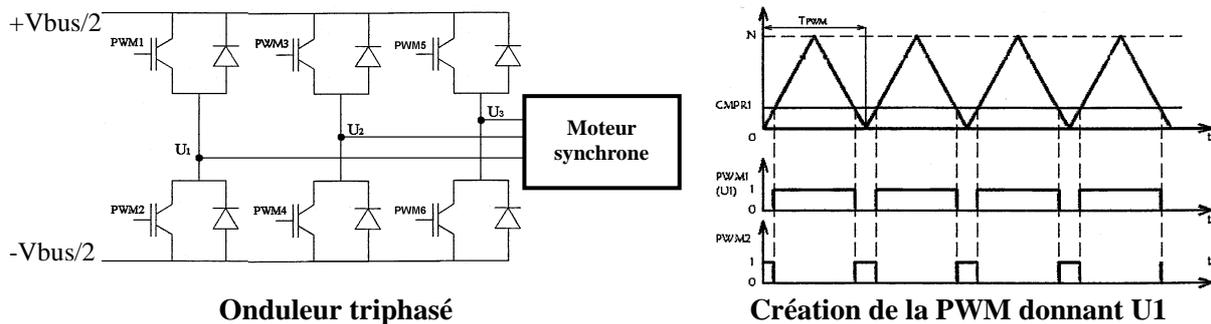
- avec
- R : résistance d'une phase
 - L_c : inductance cyclique d'une phase
 - V : tension d'alimentation (aux bornes d'une phase)
 - E : f.e.m. d'une phase
 - I : courant dans une phase : $\underline{I} = I_q + j.I_d = I \cdot \cos(\Psi) + j.I \cdot \sin(\Psi)$
 - Φ_{eff} : flux efficace créé par les aimants
 - C_{em} : couple électromagnétique : $C_{em} = 3p\Phi_{eff}I_q$
(ou encore $C_{em} = 3p\Phi_{eff}I \cos(\Psi)$) avec $p=1$ (paires de pôles)

La f.e.m. E est prise comme référence pour le contrôle du courant : plutôt que de contrôler les 3 courants alternatifs, on exprime le courant d'une phase dans le repère (d,q) tournant avec la f.e.m. E (I_d et I_q deviennent constants en régime permanent). On exprime séparément I_d et I_q qui définissent les composantes réactive et active du courant. Le déphasage Ψ des courants est exprimé par rapport à E . Les tensions à appliquer à chaque phase sont reconstituées à partir des équations électriques.

Ainsi le courant I_q permet simplement de contrôler le couple C_{em} de la machine, tandis que I_d (courant réactif) permet de déphaser le courant par rapport à E (le plus souvent $I_d = 0$ lorsque c'est possible).

Les asservissements sont cadencés à la fréquence d'échantillonnage $F_e = 2$ kHz, les réceptions sur le port série sont gérées par interruption avec mise en buffer circulaire. Les données stockées dans le buffer de réception sont interprétées en tâche de fond.

Le schéma de l'onduleur et sa commande sont illustrés ci-dessous. La fréquence PWM est fixée à 10kHz.



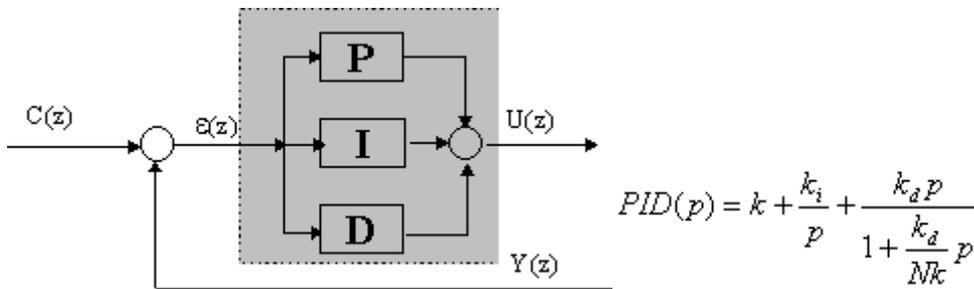
Onduleur triphasé

Création de la PWM donnant U1

- 1 Comment définit-on la fréquence d'échantillonnage minimale d'un asservissement.
- 2 Quel est l'avantage de contrôler I_d et I_q plutôt que les courants de phase directement.
- 3 Quelle est la valeur maximale possible de la fréquence d'échantillonnage dans notre cas.
- 4 On peut montrer que la tension $V1$ appliquée à une phase du moteur est l'image (en grandeur fondamentale) de la tension $U1$ en sortie du premier bras de l'onduleur. En fixant les potentiels en entrée du pont onduleur à $-V_{bus}/2$ et $+V_{bus}/2$ exprimer le rapport cyclique $\alpha 1(t)$ de la commande PWM1 en fonction de la tension instantanée $V1(t)$ et de V_{bus} .
- 5 Boucle de réactif : déterminer l'équation récurrente exprimant la variable X_r en fonction de la référence de courant magnétisant I_{dref} et de la mesure de courant $I_d = I_{sin}(\Psi)$. On utilisera le correcteur PID décrit en annexe en utilisant l'indice 1 pour les paramètres.
- 6 Boucle de couple : de la même manière déterminer l'équation récurrente exprimant la variable X_c en fonction de la référence de couple C_{ref} et de la mesure de courant $I_q = I_{cos}(\Psi)$ en utilisant l'indice 2 pour les paramètres du correcteur.
- 7 Exprimer la position θ du rotor en radians à partir de la valeur du compteur CNT associé au QEP connecté au codeur incrémental ($\theta = 0$ pour $CNT = 0$).
- 8 Déterminer l'équation récurrente donnant Ω_{rpm} la vitesse du rotor en tour/mn en fonction de CNT.
- 9 Boucle de vitesse : déterminer l'équation récurrente exprimant la référence de couple C_{ref} en fonction de la consigne de vitesse Ω_{ref} et de la mesure de vitesse Ω_{rpm} exprimées en tour/mn en utilisant l'indice 3 pour les paramètres du correcteur.
- 10 Parmi les paramètres du variateur, on s'intéresse à la limitation de couple du moteur C_{max} permettant d'éviter les surcharges. Indiquer comment compléter le schéma automatique du contrôle pour prendre en compte ce paramètre.
- 11 Indiquer comment traduire en langage C la limitation de couple.

- 12 Est-il nécessaire dans notre cas de réceptionner les données du port série par interruption et buffer circulaire ? Expliquer pourquoi.
- 13 La gestion logicielle est effectuée avec le noyau temps réel PICOS. Proposer une solution de mise en œuvre en schématisant les différentes tâches ou interruptions décrites ci-dessus et en faisant apparaître leur fonction et les éventuels évènements et synchronisation nécessaires.
- 14 Indiquer quels sont les éléments à définir pour créer une tâche sous PICOS (sans détailler la syntaxe).
- 15 Ecrire en langage C sous PICOS la fonction d'interruption `void __attribute__((__interrupt__)) _U2RXinterrupt(void)` qui gère le stockage en buffer circulaire des données reçues sur la liaison série.
- 16 Qu'appelle-t-on gigue maximale et à quel type de tâche s'applique-t-elle ?
- 17 Indiquer comment réduire au minimum la gigue de tâches d'asservissement de fréquence élevée.

Correcteur PID parallèle



UART du microcontrôleur

TABLE 18-2: UART2 REGISTER MAP (NOT AVAILABLE ON dsPIC30F4012)⁽¹⁾

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
U2MODE	0216	UARTEN	—	USIDL	—	—	—	—	—	WAKE	LPBACK	ABAUD	—	—	PDSEL1	PDSEL0	STSEL	0000 0000 0000 0000
U2STA	0218	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDL	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U2TXREG	021A	—	—	—	—	—	—	—	—	UTX8	Transmit Register							0000 0000 0000 0000
U2RXREG	021C	—	—	—	—	—	—	—	—	URX8	Receive Register							0000 0000 0000 0000
U2BRG	021E	Baud Rate Generator Prescaler															0000 0000 0000 0000	

Legend: — = uninitialized bit, — = unimplemented bit, read as '0'

Note 1: Refer to the "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

TABLE 5-2: INTERRUPT CONTROLLER REGISTER MAP⁽¹⁾

SFR Name	ADR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
INTCON1	0080	NSTDIS	—	—	—	—	OVATE	OVATE	OVTE	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000 0000 0000 0000
INTCON2	0082	ALTIPT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0IF	0000 0000 0000 0000
IFS1	0086	—	—	—	—	C1IF	—	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	IC8IF	IC7IF	INT1IF	0000 0000 0000 0000
IFS2	0088	—	—	—	—	FLTAIF	—	—	QE1IF	PWM1IF	—	—	—	—	—	—	—	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	008E	—	—	—	—	C1IE	—	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	IC8IE	IC7IE	INT1IE	0000 0000 0000 0000
IEC2	0090	—	—	—	—	FLTAIE	—	—	QE1IE	PWM1IE	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC0	0094	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			0100 0100 0100 0100
IPC1	0096	—	T3IP<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			—	IC2IP<2:0>			0100 0100 0100 0100
IPC2	0098	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100
IPC3	009A	—	CNIP<2:0>			—	MI2CIP<2:0>			—	SI2CIP<2:0>			—	NVMIP<2:0>			0100 0100 0100 0100
IPC4	009C	—	OC3IP<2:0>			—	IC8IP<2:0>			—	IC7IP<2:0>			—	INT1IP<2:0>			0100 0100 0100 0100
IPC5	009E	—	INT2IP<2:0>			—	T5IP<2:0>			—	T4IP<2:0>			—	OC4IP<2:0>			0100 0100 0100 0100
IPC6	00A0	—	C1IP<2:0>			—	—	—	—	—	U2TXIP<2:0>			—	U2RXIP<2:0>			0100 0000 0100 0100
IPC7	00A2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC8	00A4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC9	00A6	—	PWM1IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	0100 0000 0100 0100
IPC10	00A8	—	FLTAIP<2:0>			—	—	—	—	—	—	—	—	—	QE1IP<2:0>			0100 0000 0000 0100
IPC11	00AA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000

Legend: — = unimplemented bit, read as '0'

Note 1: Refer to the "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.