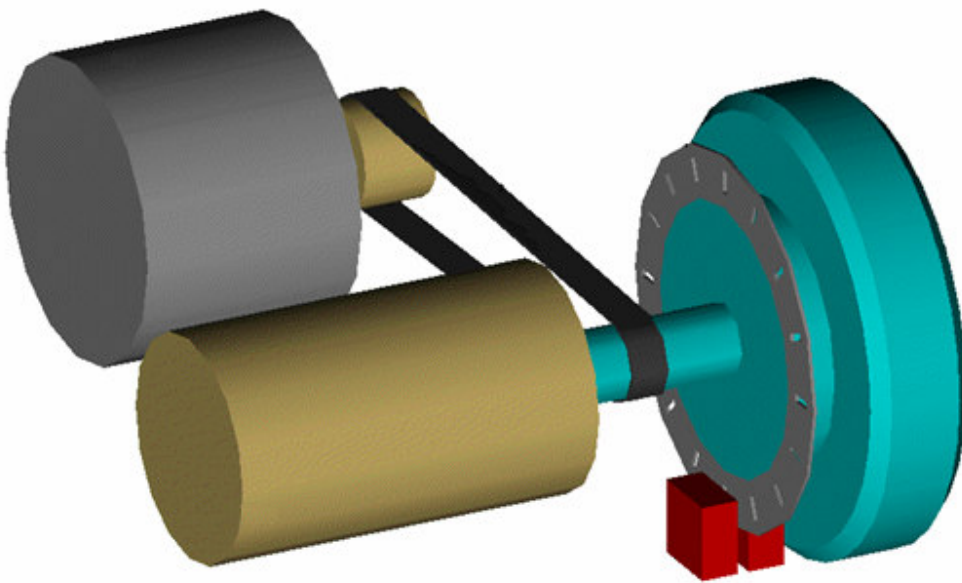


# **Relazione sul progetto di "Laboratorio di misure elettroniche"**



Redatta dagli studenti  
Pangrazzi Michele (105572) - Iacomino Antonio (101276)

Anno Accademico 2005/2006 corso di Laurea in Ingegneria delle  
Telecomunicazioni – Anno 3°

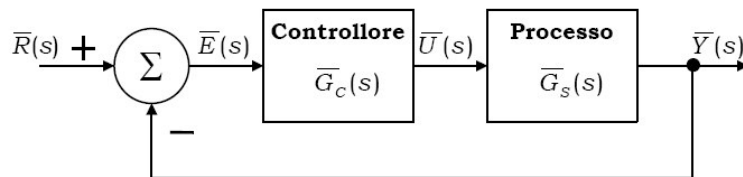
# SOMMARIO

<b>Note generali sui controllori di tipo PID</b> .....	<b>2</b>
Principi di funzionamento.....	2
<b>Il nostro controllore PID</b> .....	<b>10</b>
Schema generale e descrizione del sistema.....	10
La scheda hardware.....	11
Il motore.....	12
L'encoder ottico.....	14
Misura della corrente circolante nel motore.....	16
<b>Descrizione del software</b> .....	<b>18</b>
Descrizione del VI "Simple PID".....	19
Descrizione dei VI "STC Period".....	26
Descrizione dei VI per la generazione dell'onda.....	29
<b>Tuning del controllore PID</b> .....	<b>18</b>
Qualche considerazione sui nostri tuning.....	18
<b>Conclusione</b> .....	<b>33</b>
Tabella esplicativa.....	34
<b>Esempi grafici di funzionamento</b> .....	<b>35</b>
Steady-state con errore elevato.....	36
Steady-state ottimale.....	37
Variazione del set point che genera "salita" significativa.....	38
Variazione del set point che genera "discesa" significativa .....	39

# NOTE GENERALI SUI CONTROLLORI DI TIPO PID

## Principi di funzionamento

Il termine controllo definisce l'azione svolta per portare e mantenere ad un valore prefissato un parametro fisico di un impianto o di un processo (ad esempio, la temperatura di un forno, il livello di un fluido in un serbatoio, la posizione del braccio di un robot, la velocità di rotazione di un motore, ecc.). Indicando con  $r(t)$  il valore che si vuole far assumere alla variabile controllata e con  $y(t)$  il valore effettivamente assunto da tale grandezza, possiamo introdurre una funzione d'errore definita come:  $e(t) = r(t) - y(t)$ . Lo scopo dell'azione di controllo è quello di applicare la migliore scelta possibile della funzione  $u(t)$  (detta variabile di controllo) che i) renda il sistema asintoticamente stabile, ii) minimizzi il valor medio di  $e^2(t)$  oppure di  $|e(t)|$  e iii), riduca al livello minore possibile il tempo di risposta e le fluttuazioni intorno al valore asintotico in concomitanza di transitori di  $r(t)$ . Passando alla trasformata di Laplace, possiamo schematizzare il problema nel seguente modo:



**Fig. 1:** schema a blocchi semplificato di un sistema contenente il controllo attivo di un processo. Per semplicità, non sono mostrate le sorgenti di rumore interne e le perturbazioni esterne che si sommano ai segnali indicati in figura.

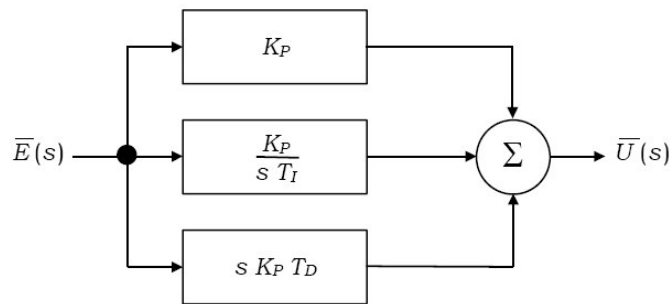
Le principali componenti del sistema sono: il rivelatore di errore  $E(s) = R(s) - Y(s)$ , il controllore che ha il compito di trasformare il segnale d'errore in un segnale  $U(s)$  che agisce sul processo sottoposto a controllo, un sensore posto all'interno del processo che misura la grandezza fisica da controllare fornendo il segnale  $Y(s)$ . Supponiamo che sia il controllore sia il processo possano essere schematizzati come sistemi lineari e stazionari caratterizzati da una funzione di trasferimento che varrà, rispettivamente,  $G(s)_C$  e  $G(s)_P$ . Va detto che lo schema mostrato in Fig.1 è semplificato perché non mostra le sorgenti di rumore interne al

sistema e le altre perturbazioni esterne che si sommano ai vari segnali. Una caratteristica essenziale di un buon controllore è quella di possedere una elevata “robustezza” rispetto alle fluttuazioni del sistema, mantenendo la grandezza controllata al valore desiderato anche in presenza di eventuali piccole variazioni della funzione di trasferimento che caratterizza il processo.

Utilizzando la ben nota relazione dei sistemi reazionati, scriviamo la funzione di trasferimento del sistema ad anello chiuso:

$$\bar{T}(s) \equiv \frac{\bar{Y}(s)}{\bar{R}(s)} = \frac{\bar{G}_C(s) \cdot \bar{G}_P(s)}{1 + \bar{G}_C(s) \cdot \bar{G}_P(s)}$$

Il problema generale del controllo si riduce quindi a determinare, per una certa funzione di trasferimento del processo  $G(s)P$ , la migliore funzione di trasferimento del controllore  $G(s)C$  che ottimizza la  $T(s)$ . Uno schema ampiamente utilizzato è quello PID, acronimo che indica l'utilizzo combinato di tre funzioni di controllo di tipo Proporzionale, Integrale e Differenziale. Lo schema generale di un controllore di tipo PID è il seguente:



**Fig. 2:** schema a blocchi di un controllore di tipo PID (vedere il testo per dettagli)

In pratica l'uscita di un controllore PID è costituita dalla somma di tre termini:

$$\bar{U}(s) = K_P \cdot \left( 1 + \frac{1}{sT_I} + sT_D \right) \cdot \bar{E}(s)$$

dove  $K_P$ ,  $T_I$  e  $T_D$  sono tre costanti positive.

Il primo termine,  $K_P$ , è detto coefficiente proporzionale. Ponendo  $T_D = 0$  e  $T_I = \infty$  l'equazione precedente si riduce a:  $U(s) = K_P E(s)$  e, in tal caso, si parla di controllore proporzionale o di tipo P. Il contributo dovuto a  $K_P$  è appunto proporzionale all'errore e diminuisce man mano che l'errore si avvicina a zero. A titolo d'esempio, consideriamo il comportamento di un

processo caratterizzato da una funzione di trasferimento contenente un polo singolo (con costante tempo  $T$ ) quando sia gestito tramite un controllore di tipo proporzionale. Sia:

$$\bar{G}_C(s) = K_P \quad \bar{G}_P(s) = \frac{A_0}{1 + \tau s}$$

Ricaviamo

$$\bar{T}(s) = \frac{\frac{K_P A_0}{1 + \tau s}}{1 + \frac{K_P A_0}{1 + \tau s}} = \frac{\frac{K_P A_0}{\tau}}{s + \frac{1 + K_P A_0}{\tau}}$$

Consideriamo il comportamento del sistema quando, al tempo  $t = 0$ , viene applicata in entrata una funzione gradino di ampiezza unitaria:

$$\bar{R}(s) = \frac{1}{s}$$

il segnale d'uscita vale:

$$\bar{Y}(s) = \frac{1}{s} \cdot \frac{\frac{K_P A_0}{\tau}}{s + \frac{1 + K_P A_0}{\tau}}$$

e la funzione d'errore vale:

$$\bar{E}(s) = \bar{R}(s) - \bar{Y}(s) = \bar{R}(s) - \bar{E}(s)\bar{G}_C(s)\bar{G}_P(s) \Rightarrow \bar{E}(s) = \frac{1}{s} \cdot \frac{1 + \tau s}{1 + \tau s + K_P A_0}$$

Utilizzando il teorema dei residui si ricava:

$$y(t) = \left( \frac{K_P \cdot A_0}{1 + K_P \cdot A_0} \right) \cdot \left[ 1 - \exp\left( -\frac{1 + K_P \cdot A_0}{\tau} t \right) \right]$$

Osserviamo che al crescere di  $K_P$  il guadagno in continua del sistema si avvicina all'unità e la costante tempo viene ridotta. In altre parole, aumentando  $K_P$ , il valore asintotico dell'uscita sarà sempre più vicino al valore richiesto (1 per il nostro segnale d'entrata corrispondente ad un gradino di ampiezza unitaria) e si osserva una contemporanea riduzione del tempo necessario per arrivare a tale livello asintotico. Notiamo tuttavia che l'uscita non arriverà mai esattamente al valore richiesto, a meno di non far tendere  $K_P$  all'infinito. Il valore asintotico dell'errore viene detto offset. L'offset può essere calcolato facilmente utilizzando il teorema del valore finale della trasformata di Laplace:  $\lim_{t \rightarrow \infty} x(t) = \lim_{s \rightarrow 0} s \cdot \bar{X}(s)$ . Applicando tale teorema alla espressione della funzione d'errore si ottiene:

$$\lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s \cdot \bar{E}(s) = \frac{1}{1 + K_P \cdot A_0}$$

da cui verifichiamo che l'offset si riduce a zero solo quando il coefficiente proporzionale tende ad infinito. Consideriamo ora un altro esempio. Supponiamo che il processo sia caratterizzato da una funzione di trasferimento del secondo ordine:

$$\bar{G}_P(s) = A_0 \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

dove  $\omega_0$  e  $\zeta$  sono due costanti positive. Con un procedimento analogo a quello sviluppato per il polo del primo ordine si ricava:

$$\bar{T}(s) \equiv \frac{\bar{Y}(s)}{\bar{R}(s)} = A_0 \frac{K_P \omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2(1 + K_P A_0)}$$

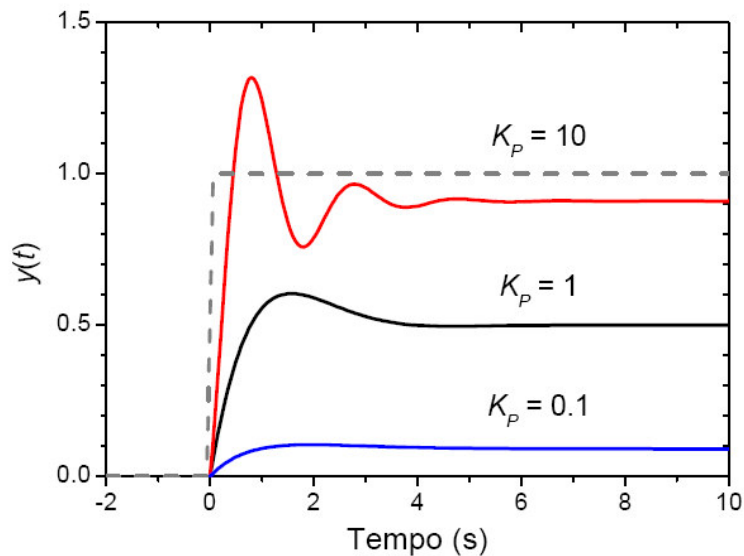
Quando si applica un gradino  $R(s)=1/s$  in entrata, si ottiene la seguente espressione per la trasformata di Laplace dell'errore:

$$\bar{E}(s) = \frac{1}{s} \cdot \frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2(1 + K_P A_0)}$$

da cui si ricava l'errore asintotico:

$$\lim_{t \rightarrow \infty} e(t) = \frac{1}{1 + K_P \cdot A_0}$$

Analogamente a quanto visto per un processo con un polo del primo ordine, anche in questo caso si osserva la presenza di un offset che diminuisce all'aumentare di  $K_P$ . Tuttavia, analizzando l'espressione della funzione di trasferimento del sistema ad anello chiuso, notiamo che la frequenza caratteristica di oscillazione vale  $\omega_f = \omega_0^2(1 + K_P \cdot A_0)^{1/2}$  ed il fattore di smorzamento è pari a  $\zeta = \omega_0 / \omega_f$ . Questo significa che, all'aumentare di  $K_P$ , il transitorio che segue rapide variazioni di  $r(t)$  è caratterizzato da oscillazioni di frequenza sempre più elevata e meno smorzate. Questa situazione è illustrata in Fig.2. Riassumendo, utilizzando un controllore esclusivamente proporzionale (P) si produce una differenza (offset) tra il valore richiesto e quello effettivamente ottenuto. Tale differenza può essere ridotta aumentando il guadagno del controllore. Tuttavia, se il processo da controllare possiede coppie di poli c.c., l'aumento del coefficiente proporzionale è accompagnato da un corrispondente aumento delle oscillazioni generate a seguito di rapidi transitori.



**Fig. 3:** Risposta ad un gradino di ampiezza unitaria, applicato al tempo  $t = 0$ , ad un processo caratterizzato da una coppia di poli c.c. ( $A_0 = 1$ ,  $\omega_0 = 1$  Hz,  $\zeta = 1$ ) gestito tramite un controllore di tipo proporzionale (P). Il calcolo è effettuato per diversi valori di  $K_P$ . Si noti che per  $K_P = 10$  si riduce l'effetto dell'offset, ma la risposta al transitorio è caratterizzata da forti oscillazioni.

Per porre rimedio a questo problema è necessario aggiungere al termine proporzionale un termine aggiuntivo che elimini a priori la presenza dell'offset. In un controllore PID tale funzione è svolta dal termine inversamente proporzionale a  $s \cdot T_I$ , detto anche contributo integrale. È evidente che in presenza di un offset costante il contributo integrale è destinato a crescere indefinitamente nel tempo e questo ci permette di attivare una efficace azione correttiva. La costante  $T_I$  è detta tempo di reset.

L'effetto dell'integrazione è tanto più importante, quanto più  $T_I$  è piccolo. Consideriamo ora un esempio: il controllo di un processo caratterizzato da un polo singolo per mezzo di un controllore PD, ovvero una combinazione di due termini, uno proporzionale ed uno integrale.

Supponiamo di inviare in entrata un gradino di ampiezza unitaria. Potremo scrivere:

$$\bar{G}_C(s) = K_P \left( 1 + \frac{1}{T_I s} \right) \quad \bar{G}_P(s) = \frac{A_0}{1 + \tau s} \quad \bar{R}(s) = \frac{1}{s}$$

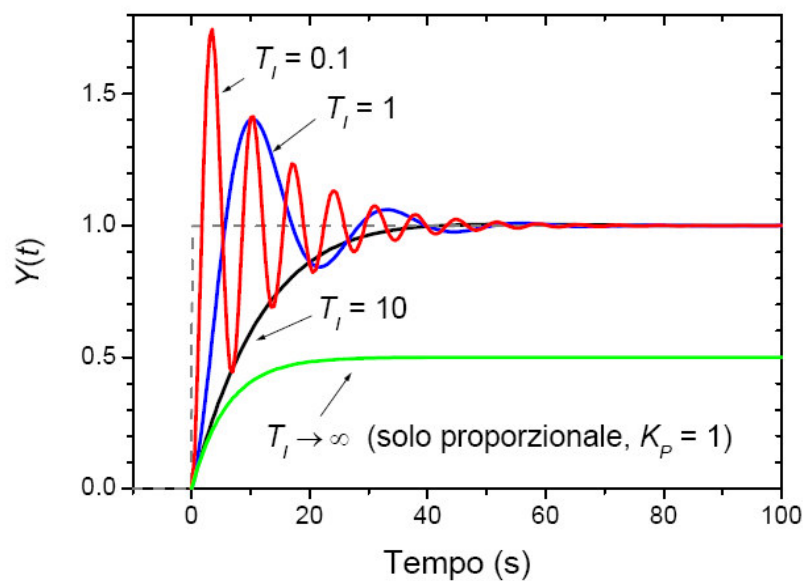
Da cui si ricava

$$\bar{T}(s) = \frac{1}{1 + \frac{s T_I (1 + \tau s)}{A_0 K_P (1 + T_I s)}} \quad \bar{E}(s) = \frac{1}{s} \cdot \frac{s T_I (1 + \tau s)}{s T_I (1 + \tau s + A_0 K_P) + A_0 K_P}$$

L'errore asintotico vale

$$\lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s \bar{E}(s) = 0$$

Notiamo che facendo tendere  $T_I$  all'infinito, ovvero eliminando il termine integrale, le espressioni precedenti si riducono a quelle trovate precedentemente per l'analogo esempio sviluppato considerando un controllore esclusivamente proporzionale. L'effetto della scelta di  $T_I$  è mostrato in Fig. 4. Si nota che l'offset presente quando si opera solo con il controllore proporzionale ( $T_I \rightarrow \infty$ ) sparisce attivando il termine integrale. Riducendo  $T_I$  il sistema risponde più velocemente al transitorio, ma si osservano anche delle forti oscillazioni.



**Fig. 4:** risposta ad un gradino di ampiezza unitaria (linea tratteggiata) di un processo caratterizzato da un polo singolo (con costante tempo  $T=12$  s e guadagno in continua  $A_0=1$ ), gestito tramite un controllore PI, calcolata per quattro diversi valori del tempo di reset  $T_I$ . Il valore di  $K_P$  è posto sempre uguale ad 1.

Passiamo ora al terzo termine di un controllore PID, quello derivativo. Il contributo derivativo al controllo tiene conto delle rapide variazioni dell'errore e cerca, in qualche modo, di anticipare



la futura azione correttiva tenendo conto delle variazioni dell'errore nei tempi più recenti. In pratica questo si concretizza in una riduzione delle oscillazioni. Per capire meglio l'effetto indotto dal contributo derivativo studiamo il comportamento di un controllore PD, ovvero un controllore in cui siano presenti sia il contributo proporzionale che quello derivativo.

$$\bar{G}_C(s) = K_P(1 + sT_D)$$

dove TD è detta costante tempo di derivazione. Supponiamo che il processo da controllare possieda due poli c.c. e scriviamo:

$$\bar{G}_P(s) = A_0 \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

Consideriamo, come già visto negli esempi precedenti, il caso in cui si applichi in entrata una funzione gradino di ampiezza unitaria. L'uscita del sistema diventa:

$$\bar{Y}(s) = \frac{A_0}{s} \cdot \frac{\omega_0^2 K_P (1 + sT_D)}{s^2 + s(2\zeta\omega_0 + A_0\omega_0^2 K_P T_D) + \omega_0^2(1 + A_0 K_P)}$$

Si nota che la presenza del termine derivativo introduce uno zero ed aumenta il coefficiente di s nel polinomio di II grado posto al denominatore. Ambedue questi effetti producono una riduzione delle oscillazioni che si verificano in occasione dei transitori contribuendo a stabilizzare il sistema. L'offset è lo stesso che si ha con il solo controllo proporzionale, ma come abbiamo visto precedentemente, questo effetto può essere eliminato utilizzando un opportuno contributo integrale. Un esempio di andamento tipico è mostrato in Fig. 5. Riassumendo, il contributo derivativo permette di ridurre le oscillazioni spurie del sistema, evitando che questo oscilli intorno al valore asintotico. D'altro canto, in presenza di un forte rumore esterno, il contributo derivativo tende ad amplificare l'effetto del rumore producendo una instabilità addizionale del sistema. In conclusione, il problema del progetto di un controllore PID si riduce alla scelta dei valori più opportuni per i parametri KP, TI e TD. Tale scelta non è banale perché richiede la conoscenza dettagliata delle proprietà del processo che si vuole controllare. Esistono vari metodi per effettuare la scelta (tuning) dei coefficienti più opportuni. Nei prossimi paragrafi vedremo brevemente due metodi semi-empirici che trovano grande utilizzo pratico.

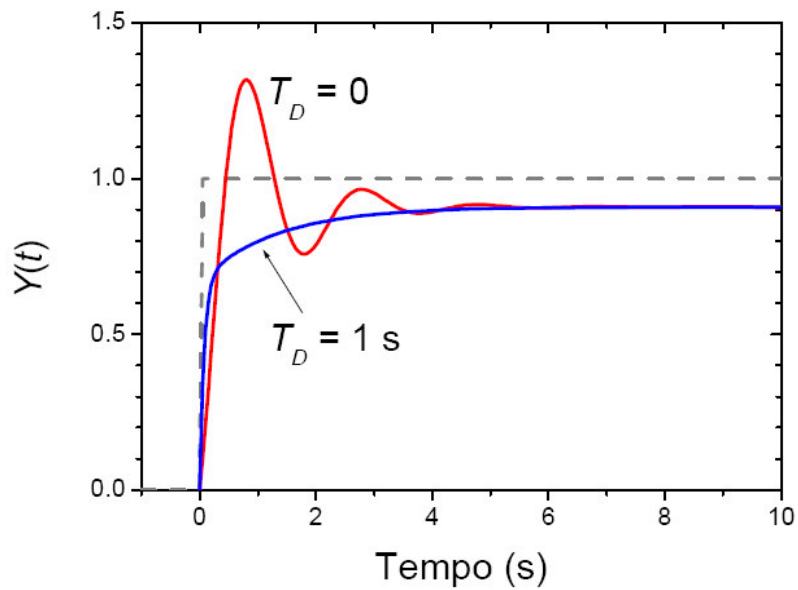


Fig. 5: risposta ad un gradino di ampiezza unitaria di un processo contenente due poli c.c. (stessi parametri usati in Fig. 3,  $K_P = 10$ ), controllato tramite il metodo PD. Quando la costante tempo di derivazione è nulla la risposta è ovviamente la stessa di Fig. 3 (controllo proporzionale). Viceversa, l'aggiunta del termine derivativo elimina le oscillazioni, pur non avendo alcun effetto sull'offset che è lo stesso del caso proporzionale.

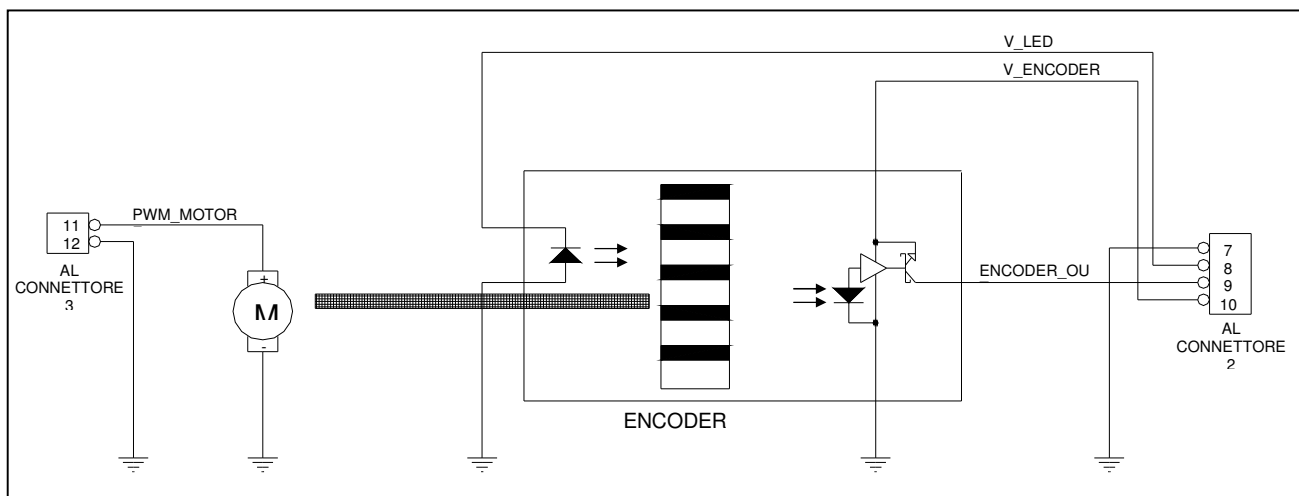
# IL NOSTRO CONTROLLORE PID

Il nostro progetto di controllo PID è stato creato per poter funzionare con un motore elettrico alimentato in continua da un generatore di tensione che può fornire da 0 Volt a 15 Volt. Il motore è poi stato opportunamente collegato ad una scheda di acquisizione; qui di seguito vengono proposti lo schema dei collegamenti e le specifiche tecniche di tutto il dispositivo.

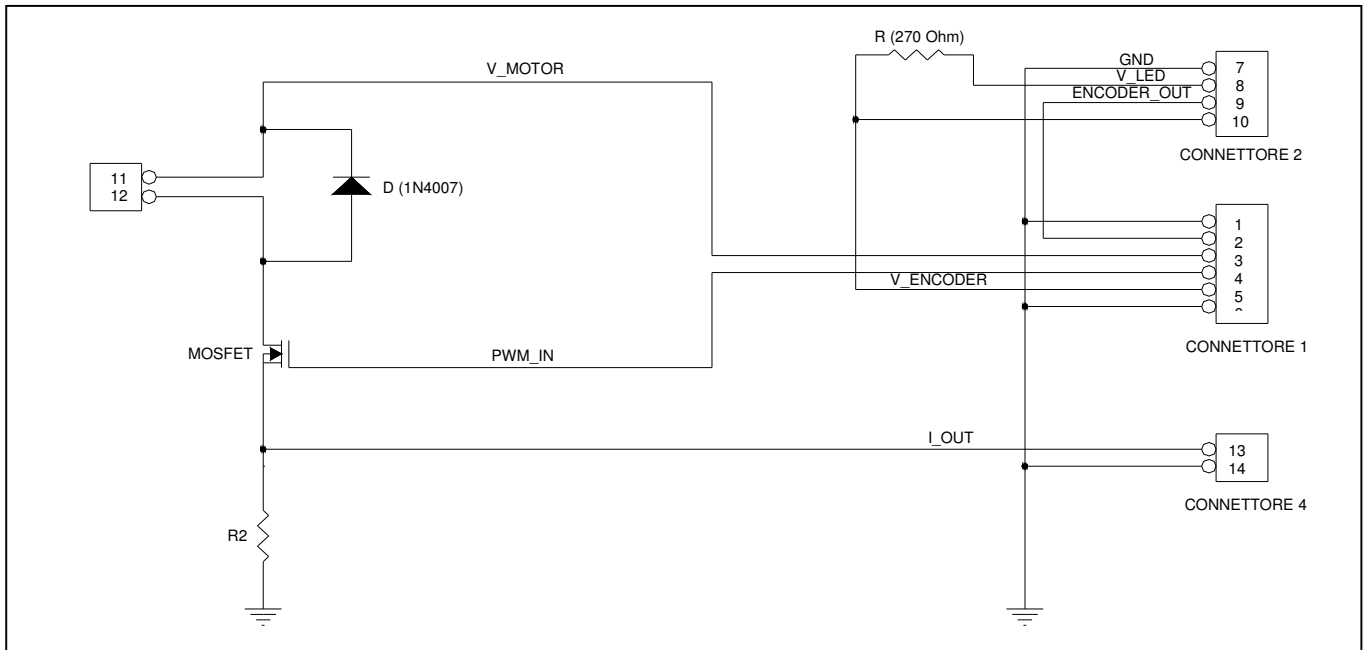
## SCHEMA GENERALE E DESCRIZIONE DEL SISTEMA

Di seguito riportiamo gli schemi della scheda e del gruppo motore:

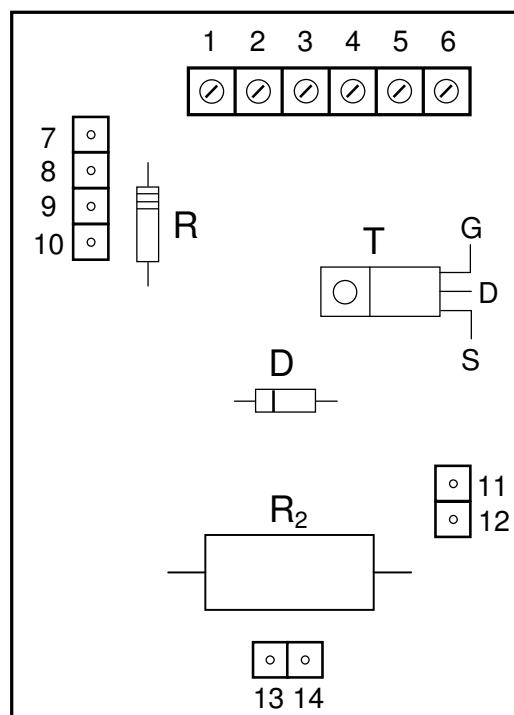
### Gruppo motore



## Scheda hardware



La scheda hardware collegata alla scheda PCI 6024E della National Instruments attraverso i connettori 1 e 4, un opportuno adattatore (CB-68LP) e un cavo parallelo e si presenta nel seguente modo.



I dati tecnici della scheda hardware sono:

Dispositivo	Simbolo	Codice
Resistenza	R	270 □
Resistenza 2	R <sub>2</sub>	1 □
Diodo	D	1N4007
Transistor	T	IRF730

N° morsetto	Descrizione
1	Massa potenza
2	Uscita encoder
3	+18 Volt
4	Ingresso Onda Quadra
5	+5 Volt
6	Massa controllo
7	Massa fotodiodo-rilevatore
8	+5 Volt fotodiodo
9	Uscita fotorilevatore
10	+5 Volt fotorilevatore
11	+18 Volt motore
12	Massa motore
13	Massa misura corrente
14	Uscita misura corrente

## Il motore

É stato utilizzato un motore in corrente continua con tensione nominale di funzionamento  $V_n$  da 12V a 18V. Il motore aziona un volano direttamente montato sull'asse e una dinamo, che alimenta una piccola lampadina, mediante una trasmissione a cinghia.

Questi carichi consentono una maggiore stabilità del motore in rotazione.

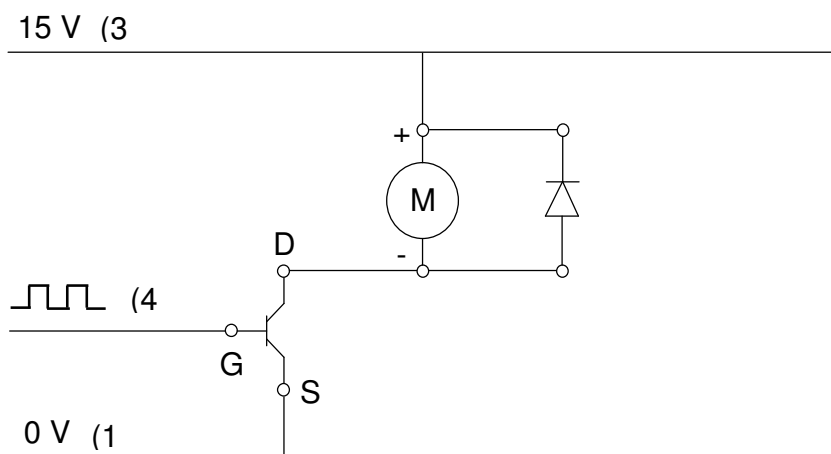
Il funzionamento del motore si basa sul principio descritto nel paragrafo precedente, dove si è visto come variando la tensione efficace ai morsetti del motore in c.c., cambiando il duty cycle dell'onda quadra può essere regolato la velocità di rotazione del motore.

L'onda quadra generata dalla scheda PCI 6024E (uscita contatore 1) comanda un transistor MOSFET che ha la funzione di aprire e chiudere il circuito di potenza, come un pulsante on-off. Il segnale in uscita dalla scheda varia tra 0 e 5V (tensione di attivazione del transistor).

Così il motore viene alimentato a 15V (tensione nominale effettivamente utilizzata ed erogata da un generatore esterno) per i tempi che dipendono dal duty cycle dell'onda quadra generata via software mediante i contatori della scheda PCI 6024E.

Per come è stato configurato l'intero sistema, la velocità di rotazione varia da 0 a circa 4000 rpm ( duty cycle=100%,cioè segnale continuo). Occorre però dire che a basso numero di giri il funzionamento del motore non è stabile e serve un duty cycle di almeno 20% per avviare il motore.

In seguito viene riportato lo schema elettrico di regolazione e controllo del motore, cioè il circuito elettrico di potenza che comanda la velocità di rotazione:



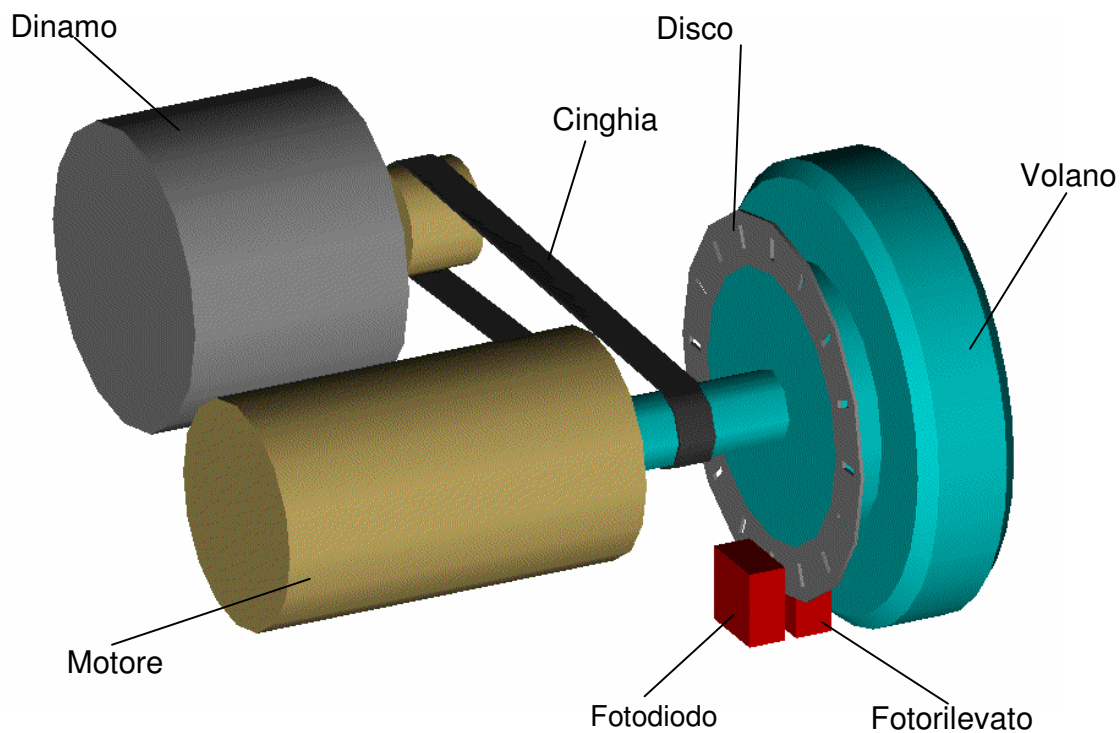
I numeri tra parentesi indicano a quale morsetti vengono collegati i dispositivi riportati nello schema.

Il diodo (1N4007) in parallelo al motore ha lo scopo di proteggere il sistema dalle extracorrenti che nascono a causa dell'induttanza dell'avvolgimento rotorico ogni qual volta il transistor apre il circuito di potenza.

## L'encoder ottico

Per il rilevamento della velocità di rotazione è stato utilizzato un sensore (RS304560). Si tratta di un interruttore ottico biforcuto realizzato con un LED che funge da fonte di luce infrarossa ed un fotorilevatore integrato. Quest'ultimo è costituito da un fotodiodo, un amplificatore, un regolatore di tensione, un trigger di Smith e da uno stadio di uscita.

Il regolatore di tensione integrato assicura un'elevata immunità ai disturbi. La parte meccanica dell'encoder è costituita da un disco con 16 fenditure fissato all'asse di rotazione. (vedi schema)

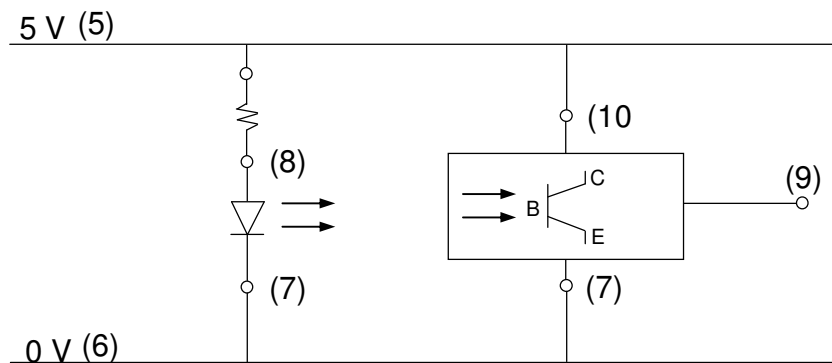


Le specifiche tecniche dell'encoder sono:

Diodo	
Max $V_F$ ( $I_F=20\text{mA}$ )	1,5 V
Max $V_R$	3 V
Max $I_F$ (cont.)	50 mA

Fotorilevatore	
Tensione di alimentazione	da +4,5 a 16 V
Dissipazione massima	250 mW a 25°C
Corrente di uscita assorbita	max. 50 mA
Corrente di esercizio	max. 15 mA

Il principio di funzionamento dell'encoder viene illustrato nel seguente schema elettrico.



Il fotodiodo , collegato alla tensione di 5 V, continua a emettere luce infrarossa; quando una fessura del disco passa davanti al fotodiodo, la luce emessa va a colpire il fotorilevatore che entra in conduzione. In questo modo al terminale del fotorilevatore si ha un impulso di tensione e dunque un segnale che per le caratteristiche dell'encoder risulta molto chiaro e privo di disturbi.



Attraverso il canale analogico ACH1 della scheda PCI 6024E questo segnale viene utilizzato per ottenere una misura della velocità di rotazione in rpm.

### **Misura della corrente circolante nel motore**

I morsetti 13 e 14 sono collegati ai capi della resistenza R2 (1  $\square$ ) in serie con il motore e quindi percorsa dalla medesima corrente. Il morsetto 13 è poi collegato all'ingresso analogico ACH2 della scheda PCI 6024E ed il 14 con la massa AIGND cosicché una misura di tensione tra ACH2 e AIGND ci fornisce direttamente il valore della corrente circolante nel motore ad ogni istante.

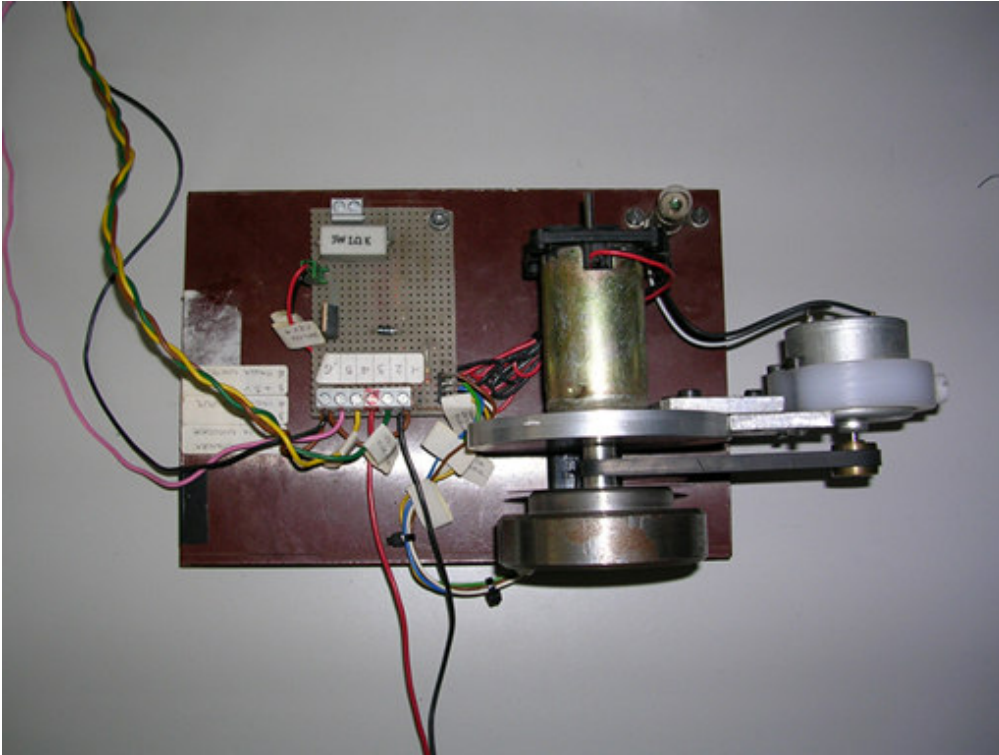
### **Tabella collegamenti schede hardware-PCI 6024E**

Scheda Hardware		Scheda PCI 6024E	
Descrizione	N° morsetto	N° connettore	Descrizione
Uscita encoder	2	33	ACH1
Alimentazione encoder +5V	5	8	+5V
Ingresso Onda Quadra	4	40	GPCTR1 OUT
Massa controllo	6	50	DGND
Massa misura corrente	13	32	AIGND
Uscita misura corrente	14	65	ACH2

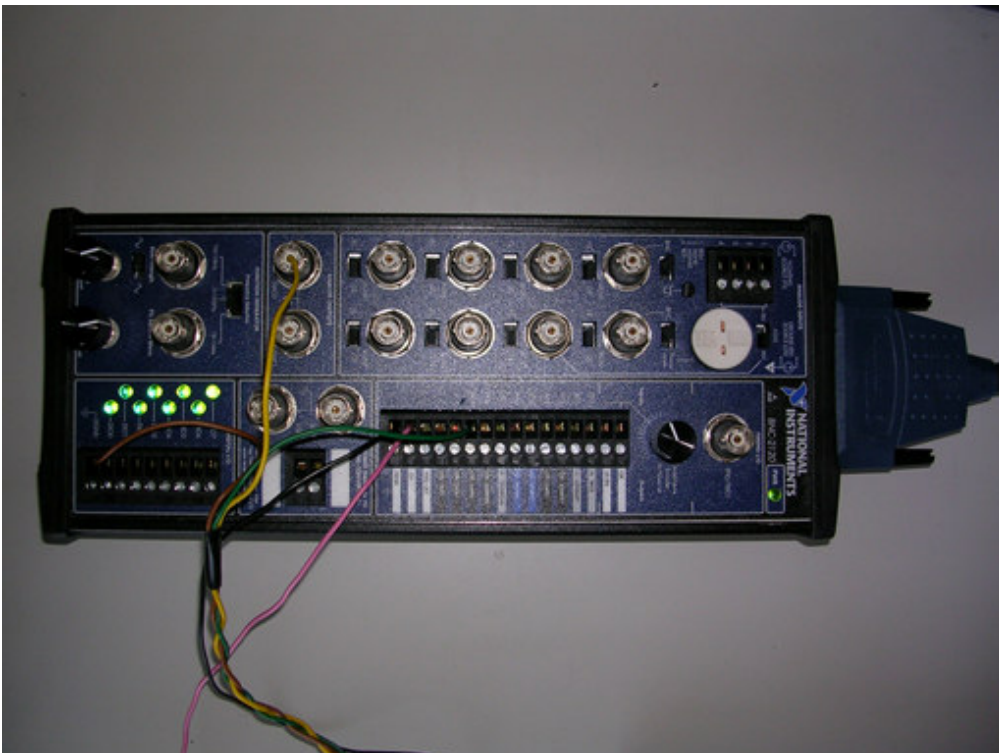
### **FOTO DEL SISTEMA**

Alcune foto esplicative del sistema, riguardanti il motore, la scheda di acquisizione e il sistema nella sua totalità collegato al PC.

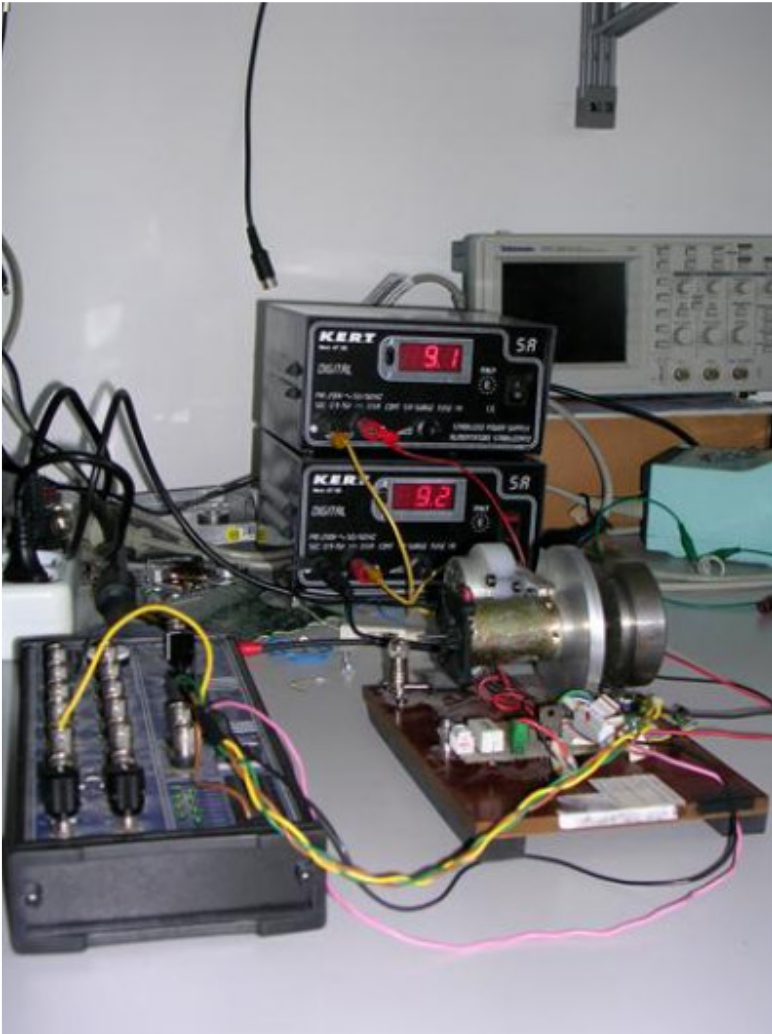
**Il motore**



**La scheda di acquisizione**



**L'intero sistema:**

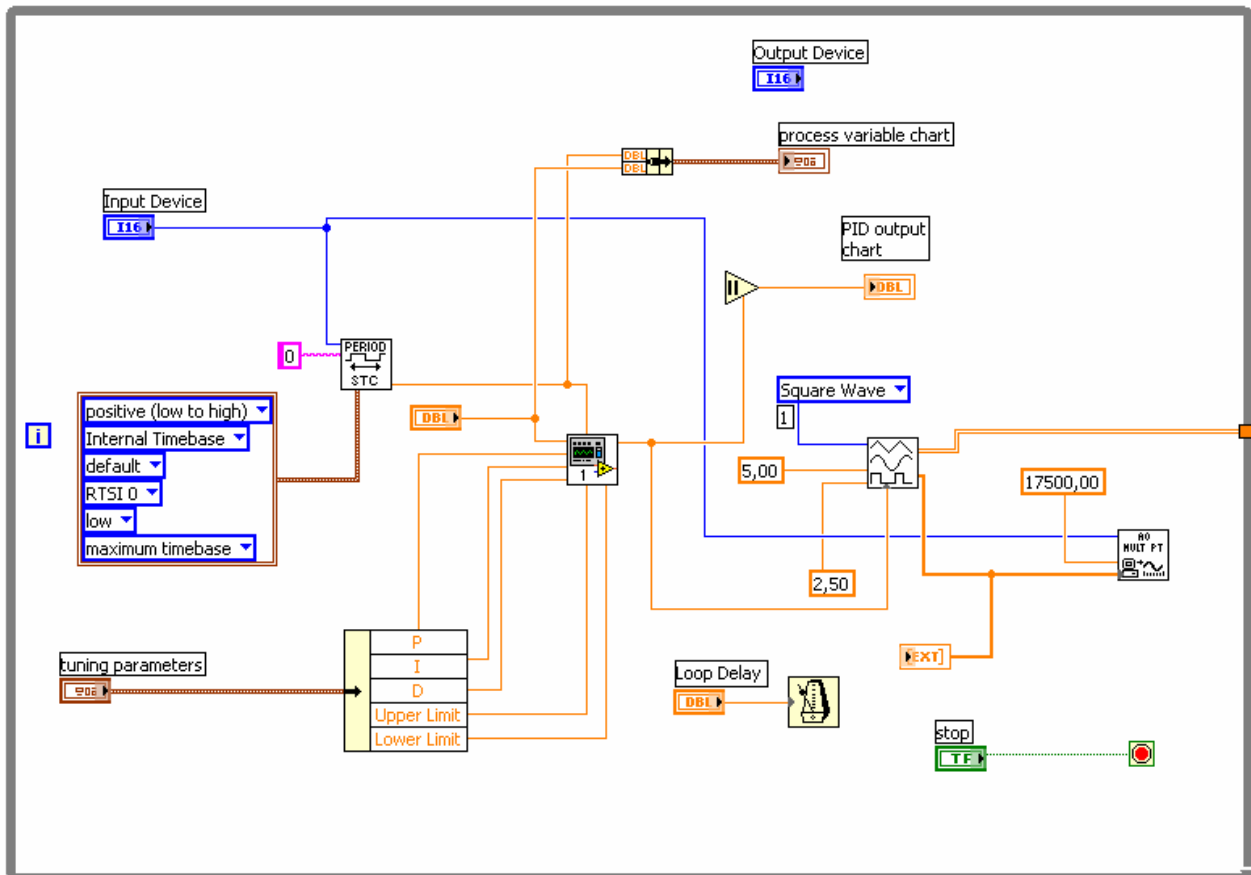


## **DESCRIZIONE DEL SOFTWARE**

Nel software da noi implementato sono presenti numerose funzioni, tutte indispensabili per un corretto funzionamento del programma. L'idea di base è stata quella di utilizzare, come forma d'onda in ingressi al motore, un'onda quadra di periodo variabile. La tensione dell'onda quadra, come si vede anche dallo schema dei collegamenti citato prima, costituisce l'ingresso del gate del MOS che gestisce l'accensione e lo spegnimento del motore. Quando si regola il comando di SetPoint si agisce sul numero di giri che il motore deve fare al secondo; questa regolazione da un comando al Simple PID che come spiegato precedentemente regola e gestisce l'onda quadra, il cui valore di tensione che può essere o 7 Volt o -2 Volt influenza il

gate del MOS rispettivamente lasciandolo attivato o disattivato e questo mi permette quindi di far aumentare il numero di giri del motore o farli diminuire.

Passiamo ora ad una descrizione delle funzioni (Vi) che entrano in gioco nel programma.



## DESCRIZIONE DEL VI SIMPLE PID

Cominciamo con la descrizione del file Vi più importante, ovvero quello che gestisce il controllo del PID. Troviamo una struttura "case" che comprende 6 casi:

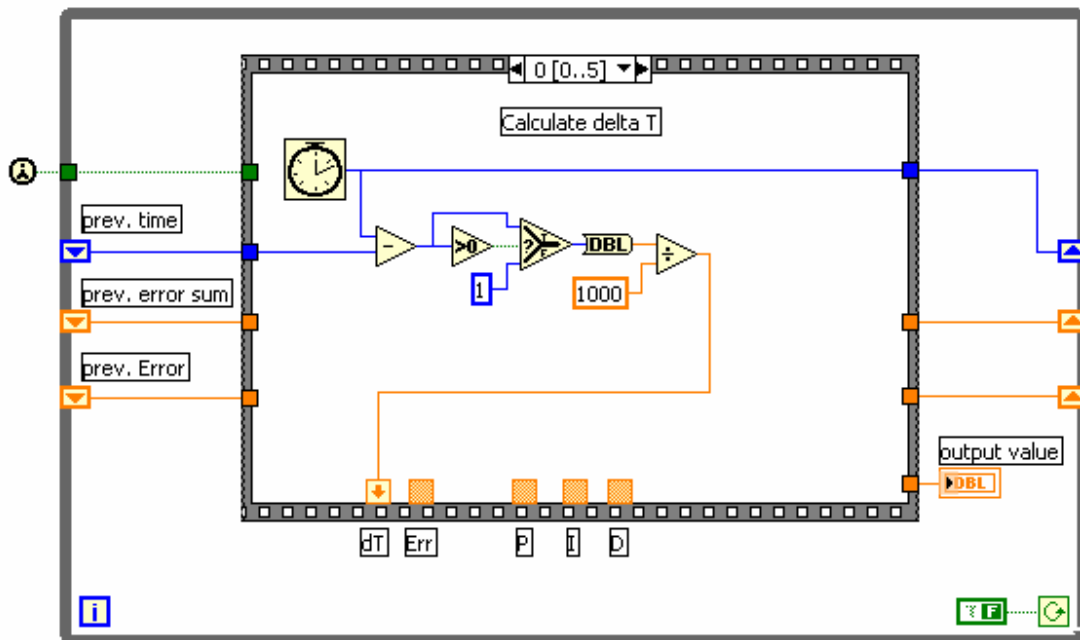
- Calcolo del delta T
- Calcolo dell'errore
- Calcolo della componente proporzionale

- Calcolo della componente integrativa
- Calcolo della componente derivativa
- Calcolo dell'output del PID

Vediamo nel dettaglio la descrizione di ognuno di questi casi:

### Calcolo del delta T

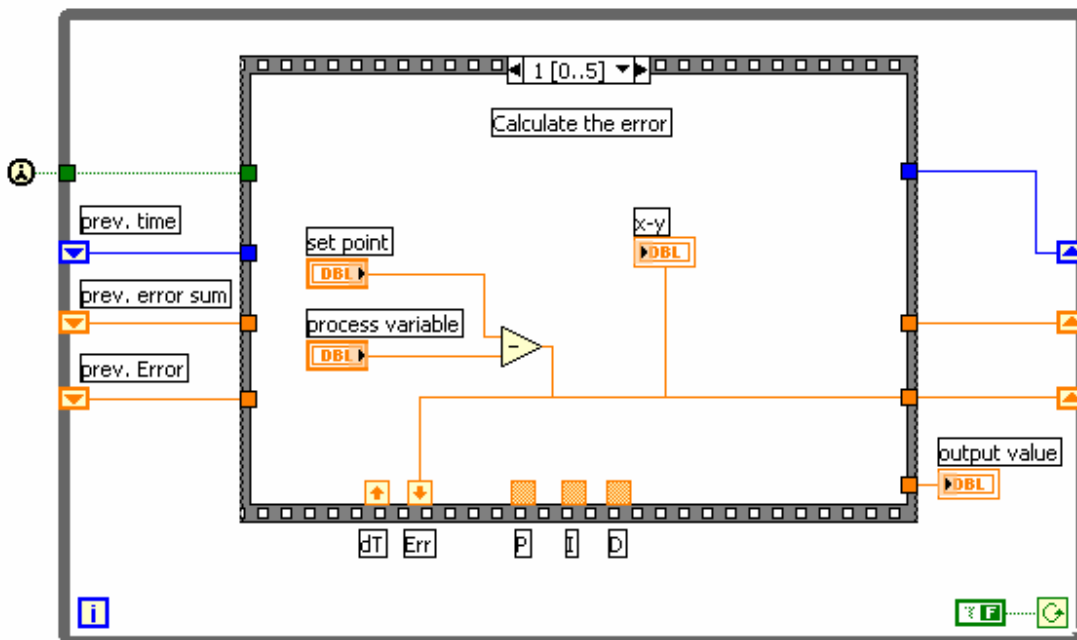
Il diagramma in Labview è in seguente:



In questo blocco viene calcolato un intervallo di tempo che poi verrà usato nel calcolo delle componenti integrativa e derivativa, come variabile di dominio. Non viene usato negli altri casi.

## Calcolo dell'errore

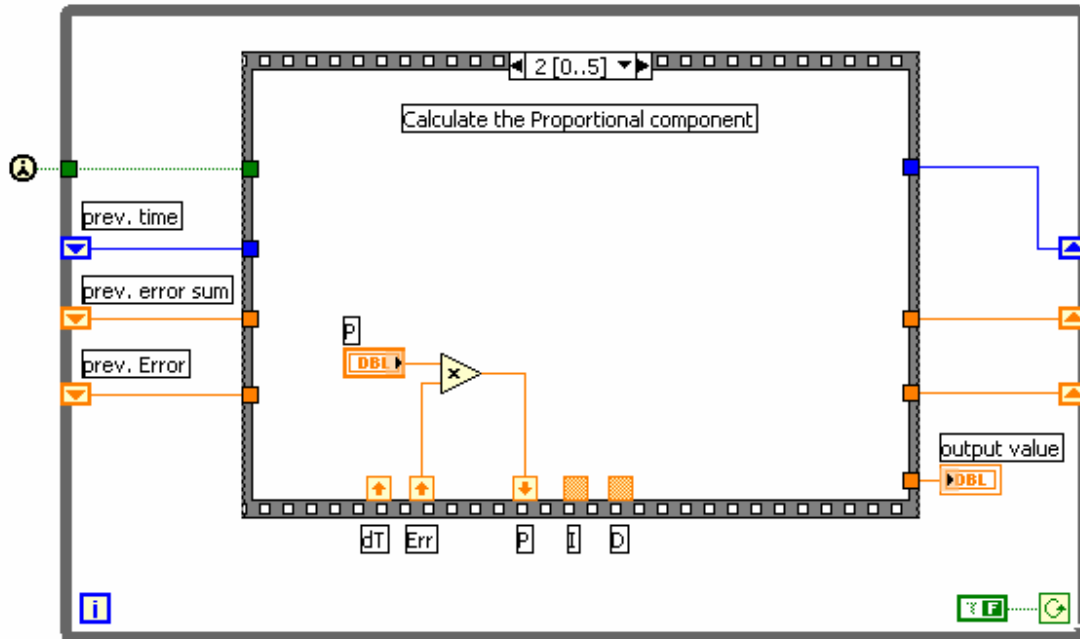
Il diagramma in Labview è in seguente:



Questo diagramma rappresenta la stima dell'errore, presente in ogni loop del ciclo while. Viene trovato mediante una banale differenza tra il set point impostato e l'output del PID, che è chiamato "process variable". Per verificarne il corretto funzionamento, abbiamo inserito un piccolo display digitale nel pannello della Vi.

## Calcolo della componente proporzionale

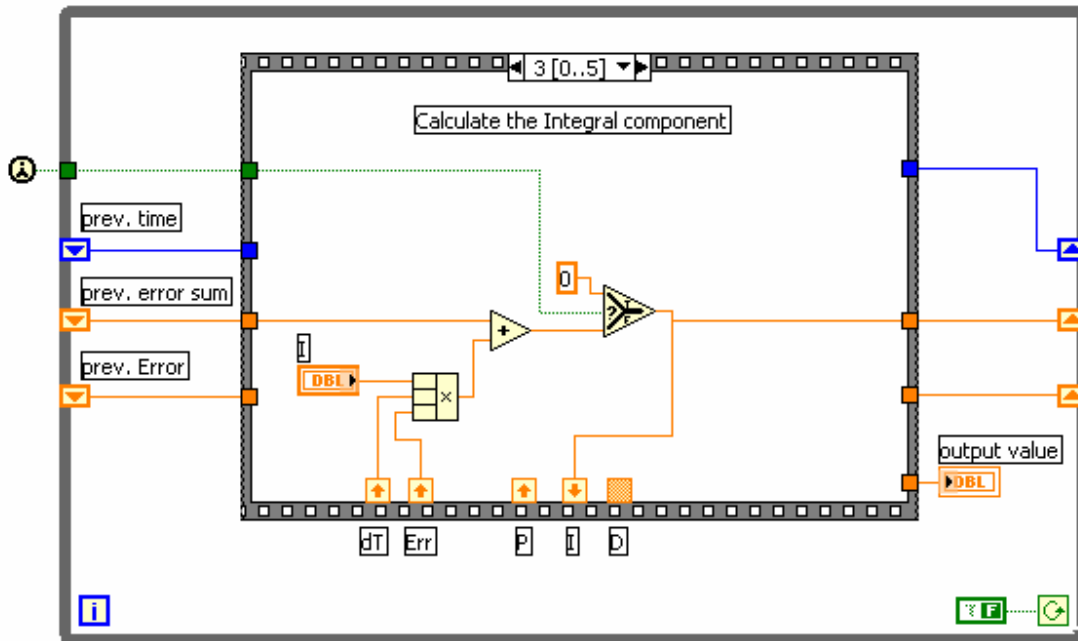
Il diagramma in Labview è in seguente:



Questo diagramma mostra il calcolo della componente proporzionale, presente in qualsiasi tipo di PID, anche il più elementare (ovvero senza componenti derivative/integrative). Il valore della componente è, come vediamo, legato al precedente calcolo dell'errore: proprio per questo motivo è da considerarsi di basso livello prestazionale se usato senza accoppiarlo a componenti di integrative e derivative, in quanto l'errore non è mai nullo a meno di non trovarsi in uno "steady state" (ovvero per mantenere un determinato set point). Questo può comunque non essere vero; infatti se diamo valori molto alti alla costante P, ci avvicineremo ancora meglio al set point, ma con un errore più grande, che si riflette in continue oscillazioni della "process variable".

## Calcolo della componente integrativa

Il diagramma in Labview è in seguente:

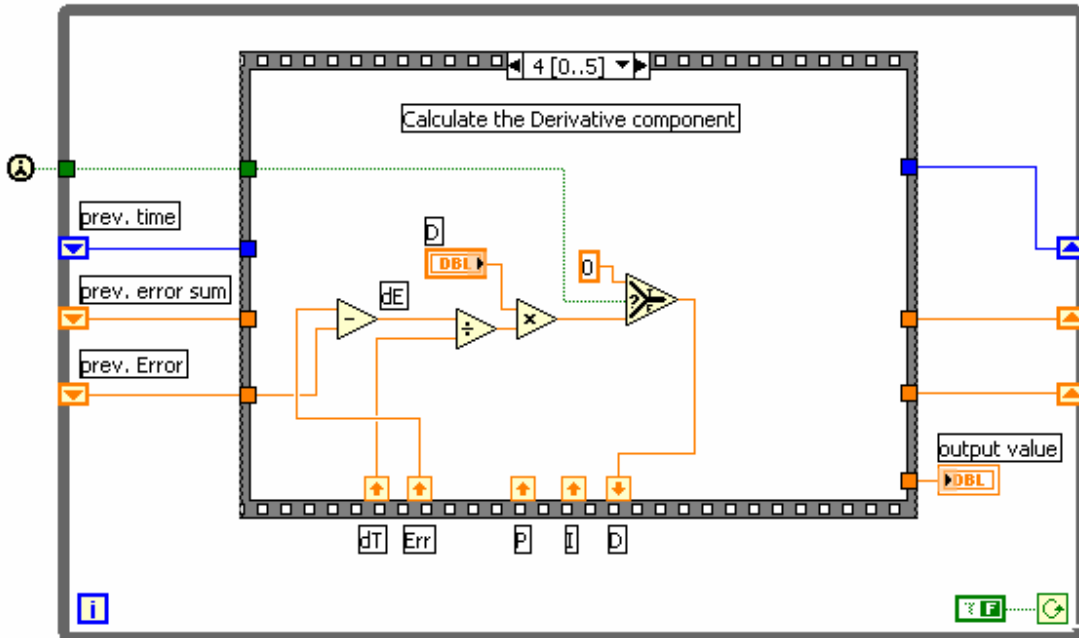


Questo blocco descrive il calcolo della componente integrativa. Anche in questo caso notiamo che è legata all'errore, ma in altro modo: infatti la funzione è proprio quella di ridurre l'errore a valori prossimi allo zero quando ci troviamo in uno "steady-state". E' intuitivo quindi l'utilizzo in combinazione con la componente proporzionale, visto che in quest'ultima l'errore in "steady-state" è sempre presente. Presenta però uno svantaggio, ovvero diminuisce la "reattività" del PID, e quindi la velocità di reazione dell'output del PID al cambiamento del set point. Questo avviene perché graficamente l'integrazione non fa altro che limitare le oscillazioni del PID.

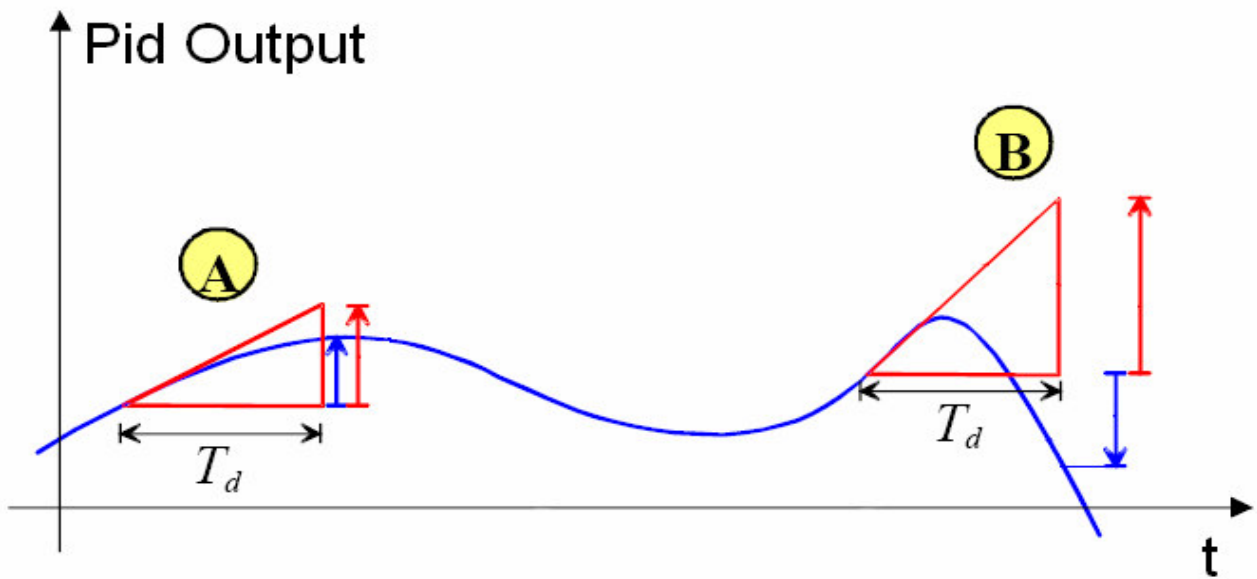


## Calcolo della componente derivativa

Il diagramma in Labview è in seguente:



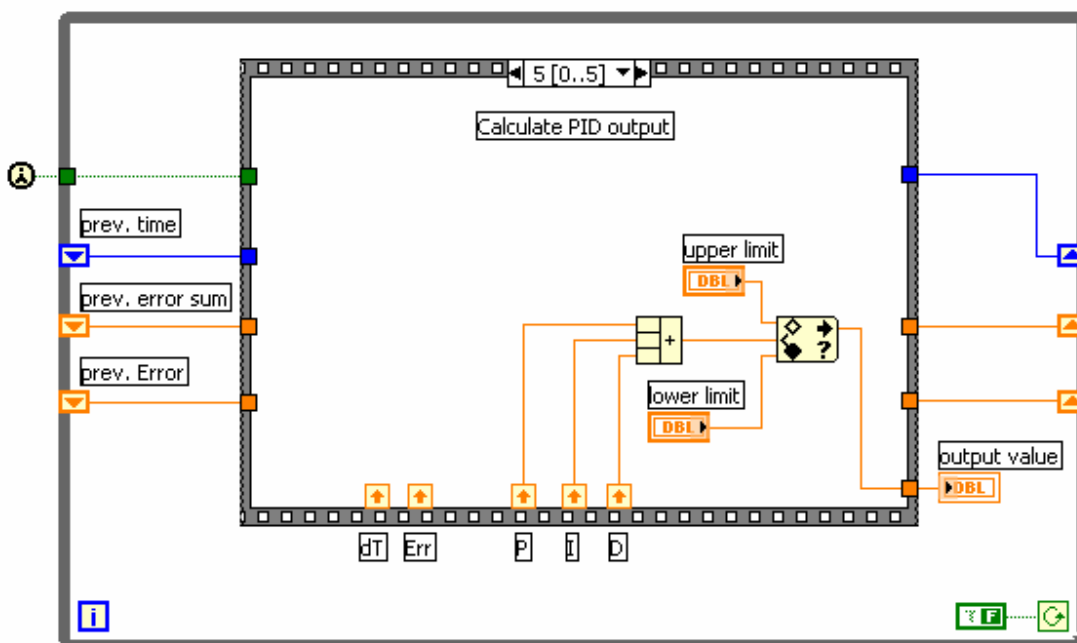
Questo blocco descrive il calcolo della componente derivativa. Notiamo il legame con il tempo (di dominio) e con l'errore. Riguardo quest'ultimo, è importante specificare che l'azione derivativa è legata alla quantità di variazione dell'errore nel tempo. Ciò significa che, in un determinato intervallo di tempo, se l'errore varia di molto allora l'azione derivativa sarà significativa, mentre se l'errore rimane costante la componente derivativa non dà alcun contributo. L'azione derivativa è, al contrario di quella integrativa, molto veloce, e quindi aumenta la "reattività" del PID. Purtroppo però, presenta uno svantaggio: il fatto che sia legata all'incremento/decremento dell'errore. Infatti può succedere che, quando il set point viene variato in modo significativo (sempre in un determinato intervallo di tempo), la derivata può non funzionare in modo corretto.



Di conseguenza bisogna capire sperimentalmente il range ottimale di funzionamento della derivata, oppure inserire una costante addizionale per quantificare l'intervallo di tempo sopraccitato, in modo che per grandi variazioni di set point sia il più piccolo possibile.

### Calcolo dell'output del PID

Il diagramma in Labview:



In questo blocco viene calcolato l'output del PID, precisamente il valore della "process variable". Essenzialmente viene fatta una somma dei tre contributi, la quale viene fatta passare per un controllo del range, che restituisce il valore in output solo se è compreso tra un determinato limite superiore e inferiore.

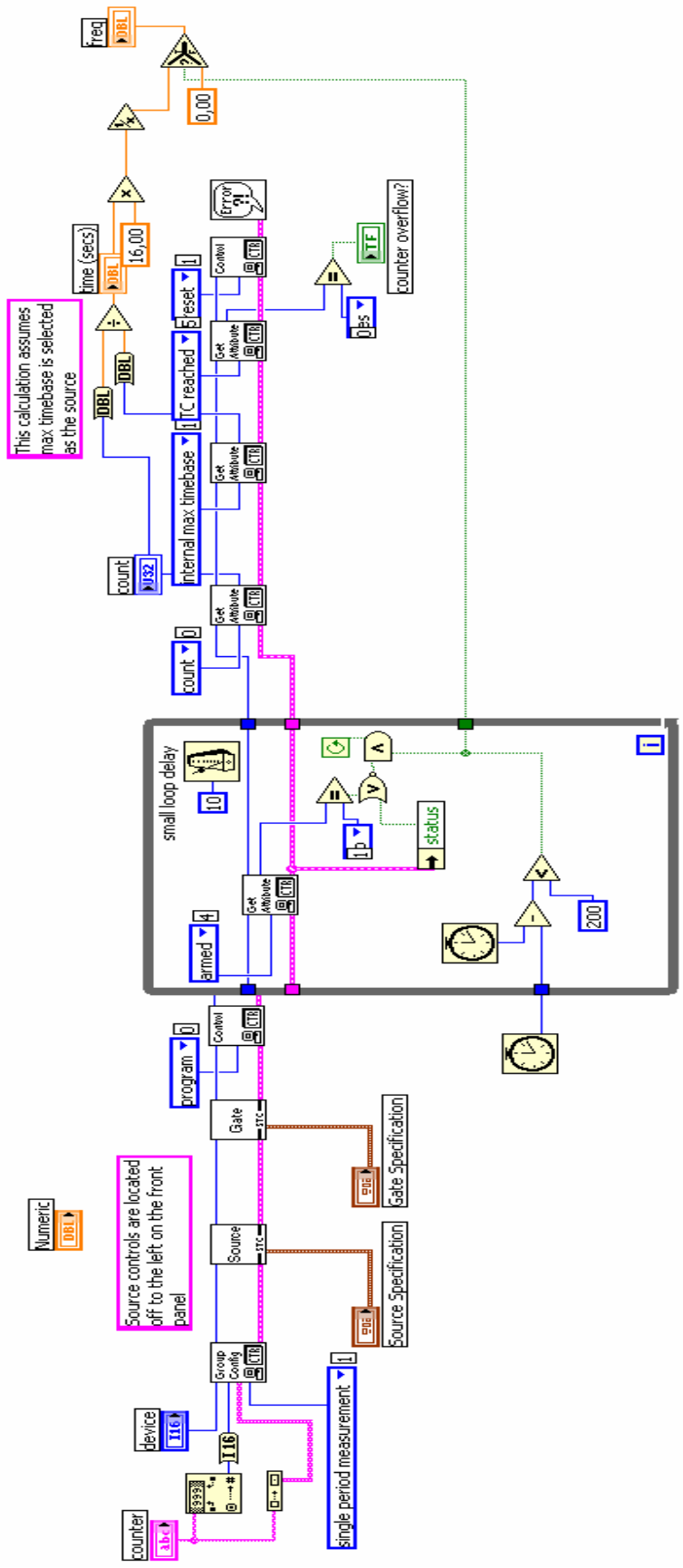
Questo VI non è stato da noi modificato significativamente. Abbiamo solamente aggiunto, nei casi che a prima vista non ci erano sembrati chiari, alcune visualizzazioni a schermo di determinati valori, per monitorarli e quindi capirne di più riguardo il tuning delle costanti (descritto in seguito).

## **DESCRIZIONE DEL VI "STC PERIOD"**

Nella pagina seguente riportiamo il diagramma in Labview.

Questo VI è molto importante per il controllore PID. Ci serviva un modulo in grado di misurare esattamente i giri/s del motore, i quali sono essenziali per il sistema in quanto rappresentano l'unità di misura del set point.

Siamo partiti da un vi già incluso in Labview, che misurava il periodo di rotazione di un generico device esterno connesso ad una scheda di acquisizione NI-DAQ.



### *LE NOSTRE MODIFICHE:*

Una volta configurato il Vi con il nostro sistema si sono presentati due problemi:

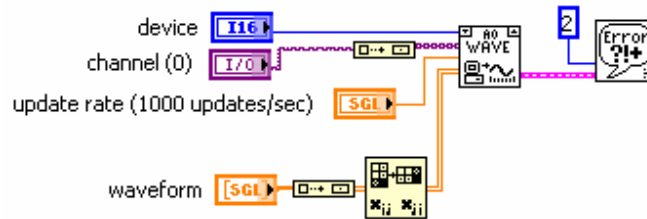
1. Misurava il periodo, mentre a noi serviva la frequenza (giri/s)
2. All'avvio del programma l'uscita era costantemente nulla e di conseguenza il controllore non entrava in funzione. In sostanza, era necessaria una piccola spinta manuale per azionare il sistema.

I problemi sono stati così risolti:

1. Abbiamo inserito una moltiplicazione per 16 (il numero delle fessure del disco fissato al volano) del valore del periodo in modo da avere tempo impiegato dal motore per percorrere un giro, dopodichè abbiamo eseguito il reciproco di tale valore in modo da avere la conversione in frequenza (giri/s).
2. Questo problema è stato di più difficile soluzione: dovevamo fare in modo che all'avvio del programma, motore si portasse al set point impostato anziché rimanere fermo ad aspettare la sopraccitata spinta manuale. Per fare questo, inizialmente abbiamo inserito un ciclo WHILE con tempo di loop impostato a 10ms. Abbiamo poi inserito due timer, uno all'esterno del ciclo ed uno all'interno: se la differenza tra i tempi è minore di 200 millisecondi, abbiamo in uscita un valore booleano. Questo valore funge da trigger nel selettore dell'output posto in fondo al VI: in sostanza se questo è vero fa passare il valore di frequenza ( $1/T$ ), mentre se è falso fa passare il valore 0 (motore fermo). Nel primo ciclo del while avremo necessariamente un valore true in entrata al selettore, e di conseguenza l'uscita sarà  $1/T$  (la frequenza appunto). Essendo in questo caso  $T$  tendente a zero, avremo un valore di frequenza in uscita che invece tenderà ad infinito, tuttavia numericamente quantificabile (un double). Questo sarà interpretato dal motore come un set point molto alto, che sarà causa del successivo e immediato incremento dell'output del PID (tutto questo sempre nei primi 10 millisecondi) e quindi farà sì che il motore possa partire; nel successivo ciclo invece l'output sarà regolato dal controllore, e quindi tendente al valore di set point impostato dall'utente.

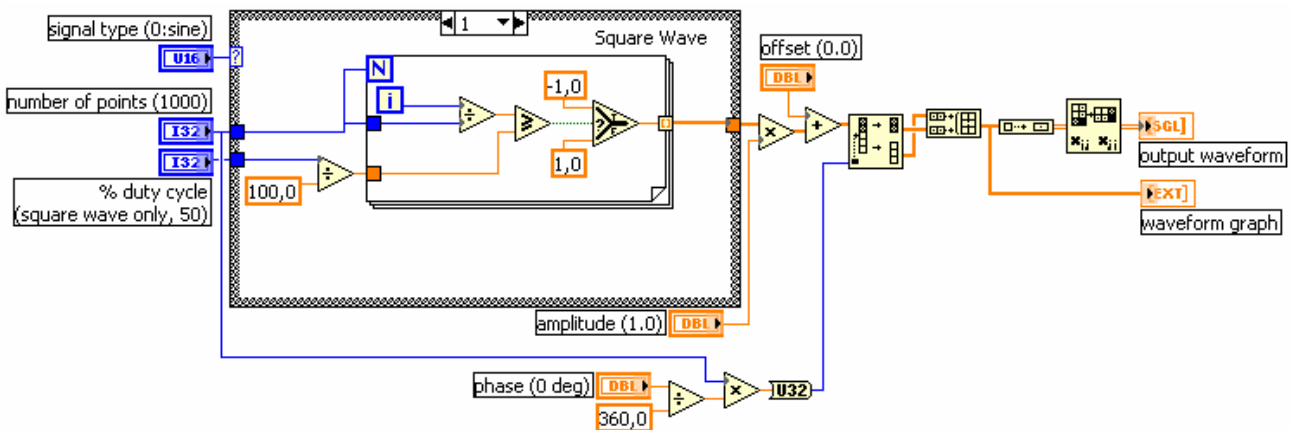
## VI RIGUARDANTI LA GENERAZIONE DELL'ONDA

Generate Waveform:



Questa VI è associata alla successiva Compute Waveform riguardo la generazione dell'onda quadra e la gestione dei campioni per secondo. Un quantitativo di 17500 sample/s permette una migliore visualizzazione a schermo in quanto modifica il tempo di aggiornamento (più rapido). Un valore eccessivo (intorno ai 20000 sample/s), aumenterebbe solamente la possibilità di un buffer overflow.

Compute (Square) Waveform:



Questa VI è stata da noi impiegata nella generazione della forma d'onda in uscita dal PID. Essendo l'output digitale, dobbiamo utilizzare un'onda quadra di ampiezza 5V e offset 2,5. La VI sarebbe in grado di generare anche altri tipi di onde (si noti infatti la struttura "case"), ma ci limitiamo a rappresentare quella da noi usata. In entrata quindi utilizziamo la percentuale di duty-cycle uscente dalla VI principale simple PID. Inoltre utilizziamo, oltre all'uscita numerica,

quella grafica, in modo da visionare sul pannello in tempo reale sia l'output analogico uscente dal simple PID che l'output in forma d'onda creato con questa VI.

## **TUNING DEL CONTROLLORE PID**

Il metodo più tradizionale per regolare le costanti P, I e D è quello empirico, ovvero basato su un metodo sperimentale. Esistono oggi tanti metodi di autotuning per controllori PID industriali, ma non ne abbiamo fatto uso poiché l'hardware da noi utilizzato non è destinato a tal tipo di impiego.

Illustriamo brevemente il metodo che abbiamo usato per regolare le costanti: inizialmente abbiamo azzerato le costanti I e D, e ci siamo basati unicamente sulla costante P. Questo perché dovevamo trovare un valore di proporzionalità tale da poter mantenere l'output abbastanza vicino al set point anche in caso di incrementi e decrementi del set point significativi. Abbiamo notato però che se si aumenta in modo eccessivo il valore di P, l'output comincia ad oscillare in maniera spropositata nel tentativo di mantenersi vicino al set point. Tal vicinanza però è inutile: abbiamo ugual (o migliore) rendimento a valori più bassi (inserire disegno).

Dopo aver trovato un valore ragionevole di P, passiamo alla regolazione di I e D. Notiamo tuttavia che regolando la costante I varia notevolmente la reattività del PID, anche se assegniamo valori molto bassi. Questo per noi è sicuramente svantaggioso poiché il nostro motore non dispone di freno, e quando deve rallentare lo fa per inerzia. Di conseguenza abbiamo deciso di utilizzare un valore molto basso per garantire una reattività maggiore.

Infine passiamo alla regolazione di D: deve essere regolata in maniera da non obbligarci ad usare una quarta costante che regoli l'intervallo di tempo, come già detto sopra. Sperimentalmente siamo riusciti a trovare un valore che minimizza l'errore, e soprattutto che non presenta l'inconveniente spiegato nel paragrafo "calcolo della componente derivativa".

### **QUALCHE CONSIDERAZIONE SUI NOSTRI TUNING**

Basandoci sul procedimento sopra descritto, passiamo ad analizzare in dettaglio i valori da noi trovati e soprattutto a spiegare il perché li consideriamo ottimali nel nostro caso.

Abbiamo sperimentato diversi valori facendo variare il set point di piccole e grandi quantità, e abbiamo testato la stabilità generale del sistema. Abbiamo fissato inoltre un range utile del set point di 55 unità, precisamente dalla velocità di 5 giri/s alla velocità di 60 giri/s. Superata quest'ultima, vista la limitazione dell'alimentazione a 18v, risulta difficoltoso ottenere un comportamento del PID omogeneo su tutto il range. Il limite massimo ottenibile sarebbe leggermente superiore a 60 giri/s, però non potrebbe essere considerato utile: infatti bisognerebbe usare in modo improprio alcune costanti per garantire il raggiungimento del limite di circa 65 giri/s alterando tantissimo il funzionamento del PID a valori bassi di set point.

I primi valori che hanno dato un risultato apprezzabile sono:

P: 35

I: 1

D: 15

Con questi valori il mantenimento del set point é molto stabile e ci sono poche oscillazioni. Questo però è vero solo per un range limitato di valori: infatti per valori vicini al limite superiore abbiamo un notevole scarto medio (2-3 unità) tra il set point e l'output del PID. Lo stesso problema si verifica vicino al limite inferiore. Tutto sommato abbiamo visto che il rendimento è molto buono se si fa variare il set point di quantità molto piccole (non più di 10/20 unità per volta, nel range 10-45). Questo è dovuto al fatto che, come già detto sopra, il contributo derivativo è utile solamente se il "salto" è relativamente piccolo; oltretutto sia la proporzionalità, probabilmente troppo bassa, che il contributo integrativo troppo elevato per le caratteristiche del nostro dispositivo davano origine ad una scarsa reattività generale del dispositivo.

Per correggere il problema causato dalla derivata, abbiamo inizialmente deciso di provare a raggiungere in modo graduale il limite superiore. Abbiamo aumentato il valore di proporzionalità a 200, in modo che, con il set point impostato a 60 giri/s, l'output sia abbastanza stabile. Dopodiché abbiamo leggermente abbassato il valore di D per limitare l'errore dovuto a variazioni consistenti del set point (ad esempio da 60 giri/s a 10 giri/s). Abbiamo infine leggermente abbassato anche il valore della costante integrativa per aumentare ulteriormente la reattività. Otteniamo quindi:



P: 200

I: 0,5

D: 8

Giunti a questo punto, il PID funzionava in modo corretto, e l'effetto dovuto all'errore della componente derivativa era nullo. Tuttavia ci siamo accorti che, almeno nel nostro caso, al diminuire del contributo integrativo aumentava la precisione e la reattività generale del dispositivo.

Abbiamo quindi provato a rendere nullo tale contributo e ci siamo in seguito accorti che non era necessario un valore così elevato di proporzionalità: in seguito a ciò abbiamo settato a 100 il valore di P, e leggermente aumentato il valore di D per sopperire alle lievi oscillazioni. In questo modo abbiamo ottenuto la condizione ideale per il nostro dispositivo:

P: 100

I: 0

D: 10

Con questi valori il PID è stabile e reattivo in tutto il range, ed è possibile anche variare significativamente il set point a differenza di quanto accadeva nei primi settaggi.

## **CONCLUSIONE**

Dopo aver effettuato diversi test abbiamo maturato un'idea abbastanza precisa degli effetti che produciamo sul nostro sistema quando andiamo a variare i valori delle costanti P, I e D. Vogliamo precisare che questa tabella dice il vero per quanto riguarda il nostro sistema le regolazioni da noi sperimentate, ma non è da ritenersi universalmente efficace per qualsiasi controllore PID.

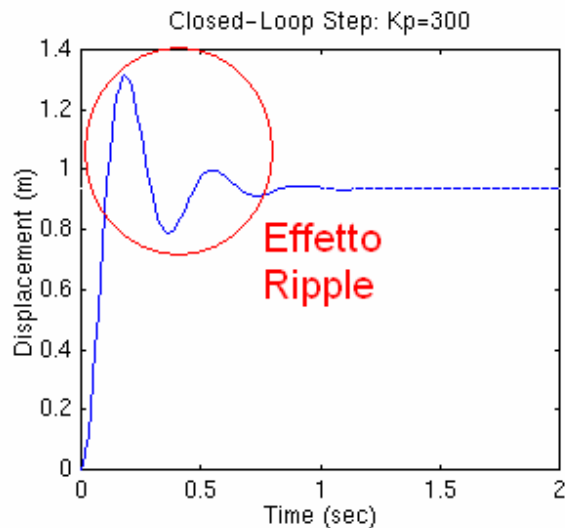
Nella pagina seguente riportiamo una breve tabella che funge da sunto di quanto appena detto:

Contributo	Reattività (nel salire)	Effetto ripple al raggiungimento del set point	Tempo di stabilizzazione	Errore
<b>Proporzionale</b>	Diminuisce	Aumenta	Cambiamento poco significativo	Diminuisce
<b>Integrativo</b>	Diminuisce	Aumenta	Aumenta	Diminuisce notevolmente
<b>Derivativo</b>	Cambiamento poco significativo	Diminuisce	Diminuisce	Cambiamento poco significativo

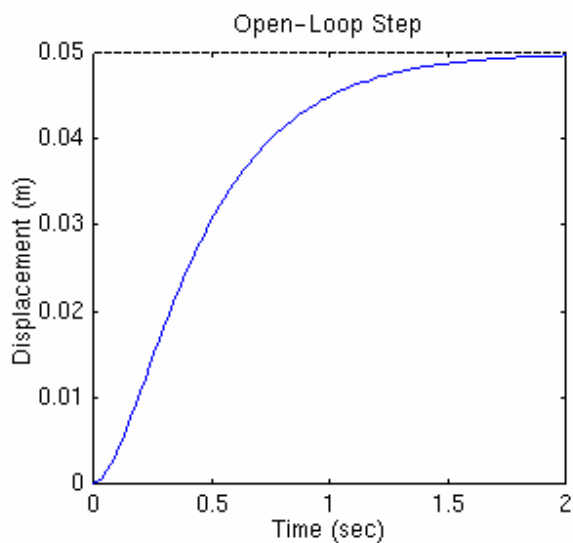
### QUALCHE NOTA SULLA TABELLA:

**Reattività (nel salire):** intendiamo il tempo di salita dell'output, ovvero la "ripidità" con cui graficamente sale.

**Effetto ripple al raggiungimento del set point (overshoot):** intendiamo quanto spiegato con la seguente immagine, riconducibile ad un comune effetto ripple:



**Tempo di stabilizzazione:** intendiamo il tempo che intercorre tra due steady-state, come esplicitato dall'immagine:



In questo caso abbiamo un tempo di stabilizzazione di circa 1.7s.

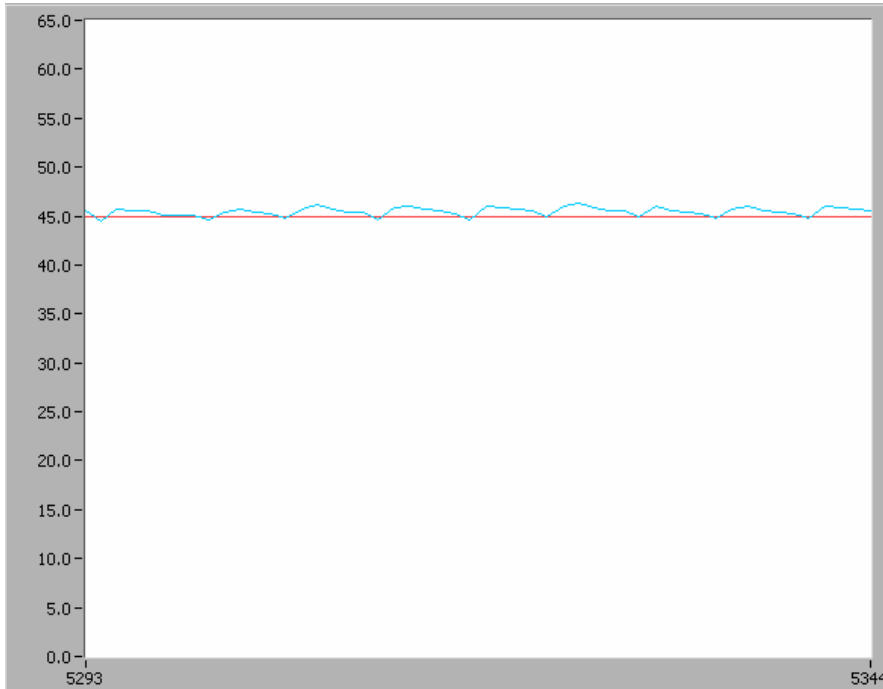
**Errore:** si intende l'errore calcolato dalla simple PID  $v_i$ , che sostanzialmente è causa delle continue oscillazioni dell'output nel cercare di mantenere un determinato set point.

## ESEMPI GRAFICI DI FUNZIONAMENTO

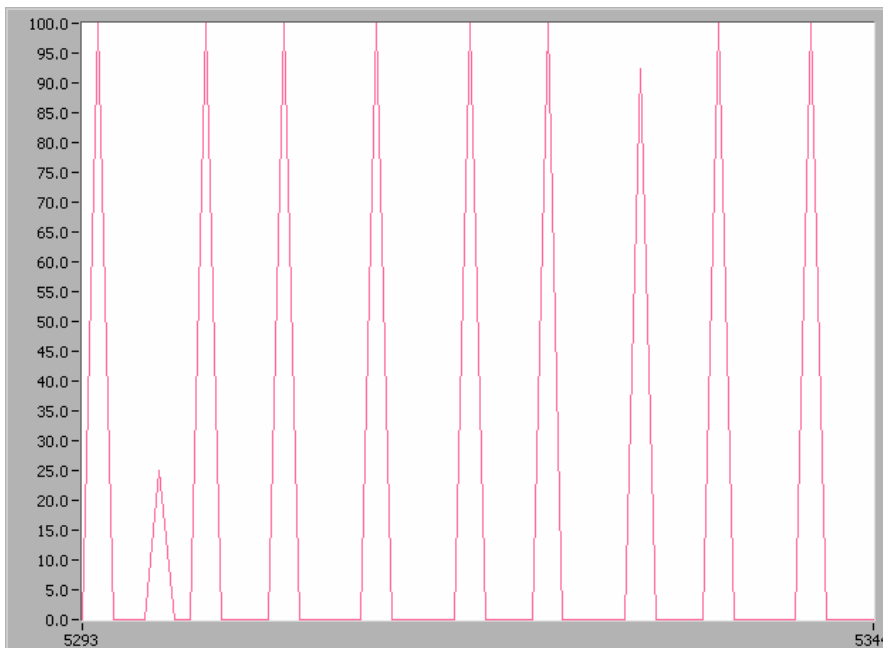
Nelle pagine seguenti riportiamo alcuni esempi grafici di funzionamento del nostro controllore PID. Mostriamo due situazioni di funzionamento in steady-state (una ottimale e una a basse prestazioni, per notare l'effetto positivo dei nostri tuning), un esempio di forte "salita" del set point e un esempio di brusca "discesa".

Esempio di funzionamento steady-state (con set point a 45 giri/s) con contributo proporzionale troppo alto (quindi elevato errore):

*Set point (rosso), giri effettivi (azzurro)*

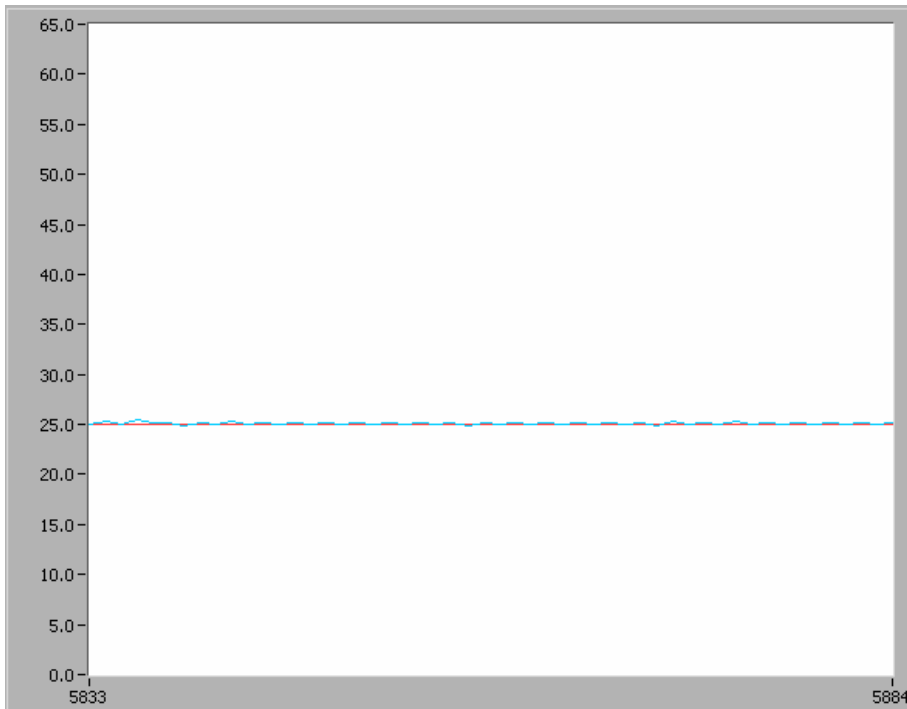


*Andamento del PID*

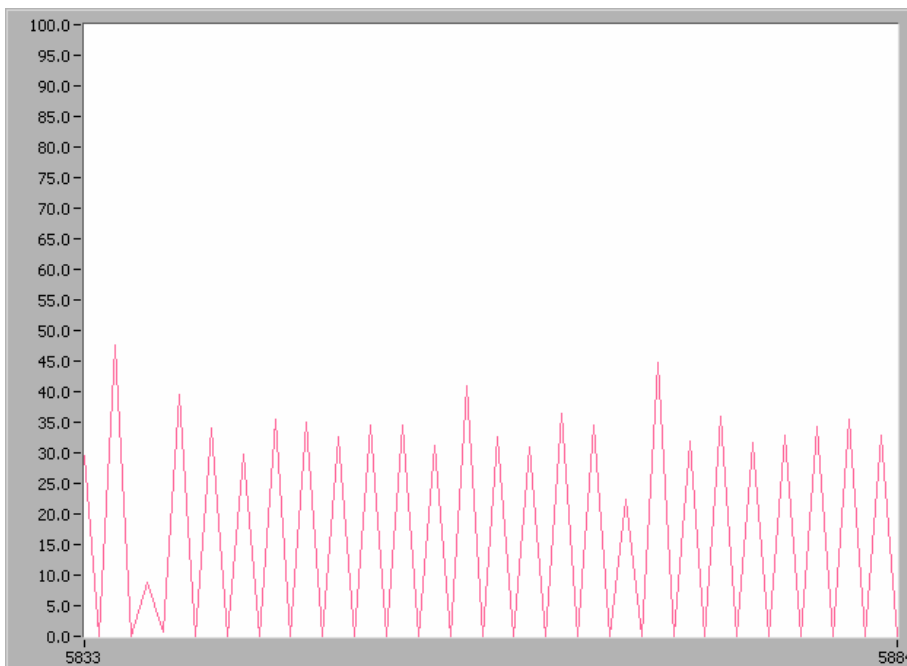


Esempio di funzionamento ottimale steady-state a 25 giri/s:

*Set point (rosso), giri effettivi (azzurro)*

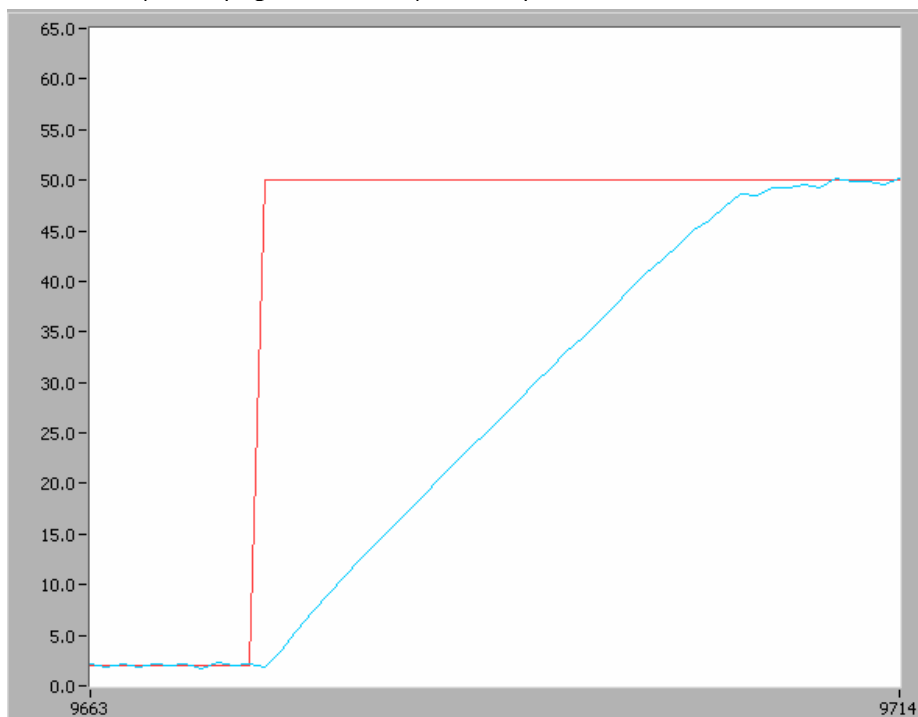


*Andamento del PID*

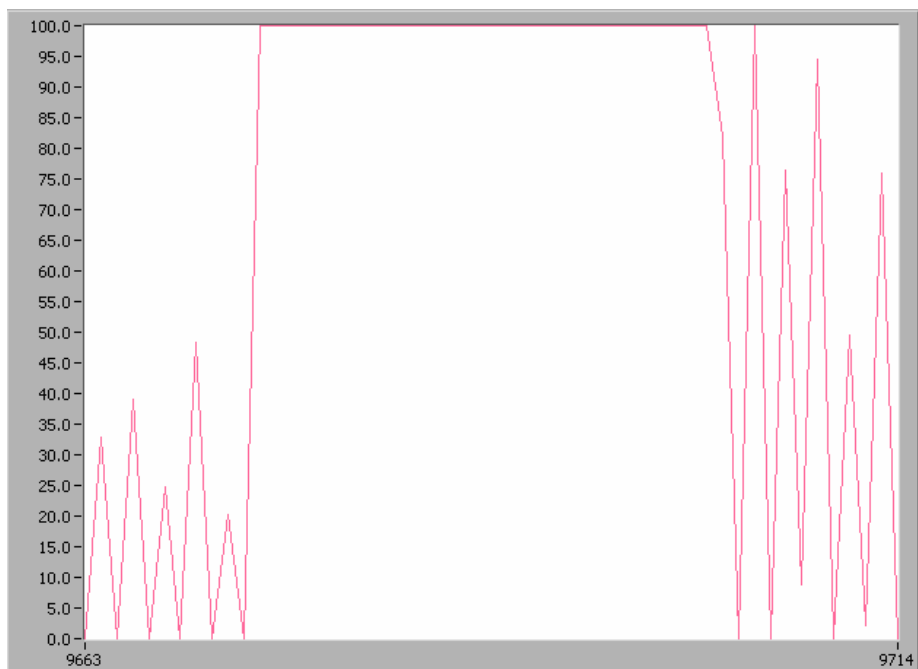


Esempio di salita abbastanza “ripida”, con set point che passa da 2 a 50 giri/s:

*Set point (rosso), giri effettivi (azzurro)*

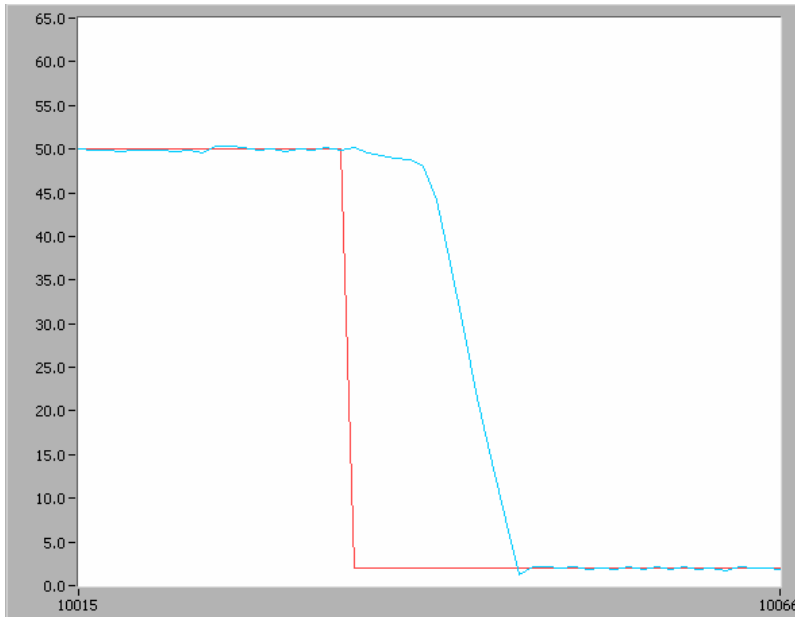


*Andamento del PID*



Esempio di brusca discesa con set point passante da 50 a 2 giri/s. Non potendo il nostro motore frenare, la discesa sarebbe molto più lunga di quanto vediamo in figura. Abbiamo molto empiricamente frenato il motore con una mano per dimostrare che comunque una volta raggiunti, i giri impostati vengono mantenuti in modo molto regolare:

*Set point (rosso), giri effettivi (azzurro)*



*Notiamo l'accenno a scendere per inerzia, la brusca discesa causata dalla nostra "frenata", e il successivo mantenimento del set point.*

*Andamento del PID*

