

ALMA MATER STUDIORUM · UNIVERSITÀ DI  
BOLOGNA

---

Scuola di Scienze  
Dipartimento di Fisica e Astronomia  
Corso di Laurea in Fisica

## Sviluppo di sistema integrato per rampe isocrone di temperatura

**Relatore:**  
Prof. Luca Pasquini

**Presentata da:**  
Riccardo Russo

**Correlatore:**  
Dott. Filippo Maria Giorgi

Anno Accademico 2015/2016



*Felicità non è fare tutto ciò che si vuole, ma volere tutto ciò che si fa.*  
Friedrich Nietzsche



# Sommario

Lo scopo del presente lavoro è la realizzazione e l'ottimizzazione di un software che, tramite l'utilizzo di un controllo automatico Proporzionale-Integrativo-Derivativo: PID, gestisca la temperatura di un fornetto in camera a vuoto. È necessario che il sistema sia in grado di eseguire rampe regolari di temperatura di diversa pendenza scelta dall'utente, in modo che possa essere utilizzato in futuro per esperimenti di Desorbimento Termico da parte di vari materiali.

La tesi è così suddivisa, nel primo capitolo sono illustrati i concetti teorici di base utilizzati nello sviluppo dei controlli automatici. Nel secondo capitolo è descritta la parte hardware: sono mostrate le diverse sezioni che compongono il fornetto e la camera a vuoto, è inoltre illustrato il cablaggio che permette l'interfaccia del forno alla scheda Arduino ed al software LabVIEW. La terza sezione è dedicata agli studi svolti per la realizzazione del sistema di controllo PID e per la sua ottimizzazione. Il quarto capitolo è invece dedicato alla descrizione del software creato per la gestione del fornetto. Nel quinto capitolo sono infine mostrati i metodi utilizzati per il calcolo delle costanti operative del PID ed i risultati sperimentali ottenuti.



# Indice

<b>1</b>	<b>Teoria dei controlli PID</b>	<b>1</b>
1.1	Controlli a Retroazione . . . . .	1
1.2	La Trasformata di Laplace . . . . .	3
1.2.1	Proprietà della Trasformata di Laplace . . . . .	3
1.2.2	Antitrasformata di Laplace . . . . .	4
1.3	Funzioni di Trasferimento . . . . .	4
1.3.1	Diagrammi a Blocchi . . . . .	5
1.3.2	Studio delle Funzioni di Trasferimento: Classificazione e Zeri & Poli . . . . .	8
1.4	I Controlli PID . . . . .	10
<b>2</b>	<b>L'Apparato Sperimentale</b>	<b>13</b>
2.1	Sistema Fisico . . . . .	13
2.1.1	Il Fornetto . . . . .	16
2.1.2	Camera a Vuoto e Sistema di Pompaggio . . . . .	16
2.2	Apparato di Controllo . . . . .	16
2.2.1	Termocoppia . . . . .	17
2.2.2	Scheda Arduino Uno . . . . .	17
2.2.3	Le Uscite PWM . . . . .	18
2.2.4	Filtri Passa Basso ed Amplificatori Operazionali . . . . .	19
2.2.5	Generatore di Tensione/Corrente . . . . .	19
<b>3</b>	<b>Studio Del Sistema</b>	<b>21</b>
3.1	Studio a Corrente Costante . . . . .	21
3.1.1	Cenni sulla classificazione dei sistemi lineari . . . . .	21
3.1.2	Fornetto a Corrente Costante . . . . .	24
3.2	La Simulazione Massa/Molla . . . . .	25
3.3	Il Rapporto Corrente/Potenza . . . . .	27
3.3.1	Il Windup della Componente Integrativa . . . . .	27
3.3.2	Potenza dissipata . . . . .	28

<b>4</b>	<b>Il Software</b>	<b>31</b>
4.1	Il Sistema di Interfaccia LINX . . . . .	31
4.2	La sub-VI PID . . . . .	32
4.3	Software di Controllo . . . . .	32
4.3.1	Isoterma . . . . .	33
4.3.2	Rampa . . . . .	34
<b>5</b>	<b>Risultati Sperimentali e Conclusioni</b>	<b>37</b>
5.1	Risultati in Isoterma . . . . .	37
5.2	Metodo di Ziegler-Nichols . . . . .	40
5.3	Risultati in Rampa . . . . .	42
5.3.1	Costanti Adattive . . . . .	47
5.4	Conclusioni . . . . .	56
5.4.1	Possibili Sviluppi Futuri . . . . .	56
<b>A</b>	<b>Il PID</b>	<b>57</b>
<b>B</b>	<b>Il Software</b>	<b>59</b>
B.1	Isoterma . . . . .	60
B.2	Rampa . . . . .	61

# Capitolo 1

## Teoria dei controlli PID

### 1.1 Controlli a Retroazione

Un *sistema* è un ente il cui stato o comportamento cambia nel tempo, solitamente in risposta a stimoli esterni; è caratterizzato da un'uscita, che è la variabile di interesse del sistema, e da un ingresso, che è ciò che mette in moto il sistema ed è *causa* dell'uscita. Il sistema è definito *dinamico* se l'uscita dipende da tutti i valori assunti dall'ingresso fino ad un dato istante di tempo.

Si parla di *Controllo* quando due sistemi dinamici sono tra loro connessi, in modo che ogni sistema influenzi lo stato dell'altro: il sistema a monte è detto di Controllo (vedi Figura 1.1). I Controlli sono suddivisi in due macro categorie: ad *anello aperto* e ad *anello chiuso o a retroazione*. Nella prima categoria un segnale in ingresso viene applicato al sistema, dove è amplificato o ridotto: il segnale in uscita è solitamente predeterminato da una calibrazione avvenuta in precedenza. In questo tipo di controlli la bontà del segnale di uscita dipende in gran parte dalla accuratezza con cui è stata eseguita la calibrazione, ma può anche essere soggetta a variazioni dovute alle condizioni ambientali: non è raro quindi che il segnale di uscita differisca dal valore atteso. I controlli ad anello chiuso hanno le medesime componenti della categoria precedente, con ulteriori caratteristiche: in questo caso il segnale in uscita è misurato e trasmesso "indietro" all'ingresso, dove viene comparato con l'*input*; il segnale che giunge al Controllo non è quindi quello di ingresso, ma la differenza (*errore*) tra l'ingresso e l'uscita. È quindi chiaro come i controlli a retroazione abbiano la possibilità di correggere automaticamente le discrepanze tra l'output richiesto e quello reale, rendendosi in questo modo virtualmente indipendenti da calibrazioni e variazioni ambientali: in questa trattazione sarà esaminato quest'ultimo tipo di controlli.

Sebbene esistano diverse tipologie di controlli, applicabili ad altrettanti

sistemi (elettrici, pneumatici, ottici, idraulici ecc), le equazioni che li descrivono hanno la stessa forma, e della medesima forma saranno quindi anche le soluzioni.

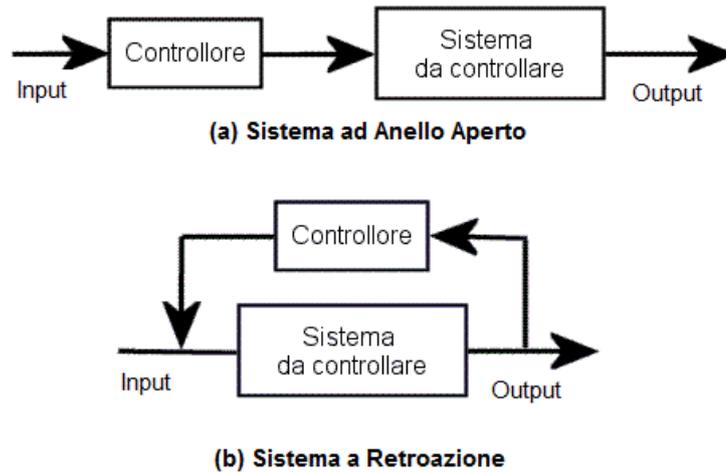


Figura 1.1: Schemi dei controlli

In molti casi è possibile modellizzare il sistema tramite un'equazione differenziale lineare del tipo seguente:

$$A_n \frac{d^n \theta_{out}}{dt^n} + \dots + A_2 \frac{d^2 \theta_{out}}{dt^2} + A_1 \frac{d\theta_{out}}{dt} + A_0 \theta_{out} + A + A_{-1} \int \theta_{out} dt = f(E) \quad (1.1)$$

dove

$\theta_{out}$  = quantità in uscita

$E$  = errore =  $\theta_{in} - \theta_{out}$

$\theta_{in}$  = quantità in ingresso

$A_n, \dots, A$  = coefficienti che sono funzioni dei parametri del sistema

nell'ipotesi che il sistema sia lineare i coefficienti sono costanti nel tempo.

Poiché la soluzione diretta delle equazioni differenziali che descrivono il sistema presenta diversi problemi, esiste un metodo molto efficace per lo studio dei sistemi dinamici, che sfrutta la *Trasformata di Laplace* per ricavare la *Funzione di Trasferimento* del sistema.

## 1.2 La Trasformata di Laplace

La Trasformata di Laplace è un operatore funzionale lineare che, ad una funzione a dominio Reale associa una funzione di variabile complessa: nel presente caso la Trasformata agisce su funzioni nel dominio dei tempi, associandovi funzioni nel dominio delle frequenze tramite il cambio di variabile:  $t \rightarrow s = \sigma + i\omega$  dove  $\omega$ , costante reale, è la frequenza di oscillazione del sistema.

È quindi possibile definire la *Funzione Trasformata* di  $f(t)$ , funzione del tempo qualsiasi, nel seguente modo:

$$F(s) := \mathcal{L}[f(t)] \quad (1.2)$$

dove  $\mathcal{L}$  è l' *Integrale di Laplace*, definito come:

$$\mathcal{L} := \int_0^{\infty} e^{-st} dt \quad (1.3)$$

Si ha quindi:

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} e^{-st} dt [f(t)] = \int_0^{\infty} f(t) e^{-st} dt \quad (1.4)$$

### 1.2.1 Proprietà della Trasformata di Laplace

La Trasformata gode delle seguenti proprietà:

- **Linearità:**

$$\mathcal{L}[af(t) + bg(t)] = a\mathcal{L}[f(t)] + b\mathcal{L}[g(t)] \quad (1.5)$$

- **Derivazione (I Ordine):**

$$\mathcal{L}[f'] = s\mathcal{L}[f] - f(0^+) \quad (1.6)$$

- **Derivazione (Ordine n):**

$$\mathcal{L}[f^{(n)}] = s^n \mathcal{L}[f] - \sum_{k=1}^n s^{n-k} \frac{d^{k-1} f(0)}{dt^{k-1}} \quad (1.7)$$

- **Integrazione:**

$$\mathcal{L}\left[\int_0^t f(\tau) dt\right] = \frac{\mathcal{L}[f(t)]}{s} \quad (1.8)$$

- **Prodotto di convoluzione:**

$$\mathcal{L}[f * g] = \mathcal{L}[f] \cdot \mathcal{L}[g] \quad (1.9)$$

Di seguito sono inoltre riportate le Trasformate di Laplace di alcune funzioni notevoli:

- $f(t) = A = \text{const}$

$$\mathcal{L}[A] = \frac{A}{s}$$

- $f(t) = e^{-\alpha t}$

$$\mathcal{L}[e^{-\alpha t}] = \frac{1}{\alpha + s}$$

- $f(t) = \sin(\beta t)$

$$\mathcal{L}[\sin(\beta t)] = \frac{\beta}{s^2 + \beta^2}$$

- $f(t) = \cos(\beta t)$

$$\mathcal{L}[\cos(\beta t)] = \frac{s}{s^2 + \beta^2}$$

- $f(t) = t^n$

$$\mathcal{L}[t^n] = \frac{n!}{s^{n+1}}$$

### 1.2.2 Antitrasformata di Laplace

può essere necessario passare dal dominio Complesso a quello Reale, o nel nostro caso dal dominio dei tempi a quello delle frequenze. Questa operazione è detta *Antitrasformata di Laplace*: è l'operazione inversa della Trasformata e si indica come:

$$\mathcal{L}^{-1}F(s) = f(t) \tag{1.10}$$

L'Antitrasformata di Laplace gode delle stesse proprietà della Trasformata (Linearità, Derivazione, etc.). Vale inoltre la seguente proprietà:

*Per ogni funzione  $f(t)$  data, esiste una e una sola Trasformata di Laplace diretta  $F(s)$ , e per ogni  $F(s)$  data esiste una ed una sola Trasformata di Laplace inversa  $f(t)$*

## 1.3 Funzioni di Trasferimento

Come visto in precedenza, la maggior parte dei sistemi dinamici può essere descritta per mezzo di un'equazione differenziale lineare del tipo (1.1). La soluzione diretta di queste equazioni è però spesso abbastanza complessa, specialmente nel caso di sistemi del terzo ordine o superiori. La Trasformata di

Laplace permette di ovviare a questo problema, trasformando le equazioni dei sistemi in equazioni algebriche di maggiore facilità risolutiva. Se il sistema ha un segnale in ingresso ed uno in uscita, l'equazione algebrica ottenuta può essere manipolata in modo da formare il rapporto tra il segnale di output e quello di input: la funzione così ottenuta è detta *Funzione di Trasferimento* del sistema ed è la più utilizzata dello studio dei sistemi dinamici.

La Funzione di Trasferimento è quindi definita come *rapporto tra le Trasformate di Laplace dei segnali di uscita e ingresso*, e si scrive:

$$G(s) := \frac{\theta_{out}(s)}{\theta_{in}(s)} \quad (1.11)$$

Nel caso in cui il segnale in input non sia in regime transitorio, la variabile complessa  $s$  si riduce alla sola parte immaginaria:  $s = i\omega$ , e si ha quindi un segnale oscillatorio ad ampiezza costante. Uno dei vantaggi che si hanno nell'utilizzo delle Funzioni di Trasferimento è la possibilità di semplificare gli schemi dei sistemi da studiare: una volta trovata la Funzione di Trasferimento di un componente, infatti, la sua natura fisica perde di importanza e può essere sostituito da un *Diagramma a Blocchi*.

### 1.3.1 Diagrammi a Blocchi

A titolo di esempio è mostrato il calcolo della Funzione di Trasferimento di un circuito RC, e del suo diagramma a blocchi:

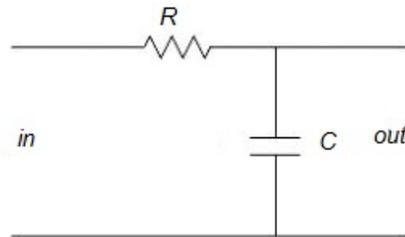


Figura 1.2: Circuito RC

**Esempio 1.** Per il circuito in figura 1.2 le equazioni di input e output all'equilibrio sono:

$$v_{in}(t) = iR + \frac{1}{C} \int i dt = R \frac{dq}{dt} + \frac{q}{C} \quad (1.12)$$

$$v_{out}(t) = \frac{q}{C} \quad (1.13)$$

Le cui Trasformate di Laplace si scrivono:

$$V_{in}(s) = (sR + \frac{1}{C})Q(s) \quad (1.14)$$

$$V_{out}(s) = \frac{1}{C}Q(s) \quad (1.15)$$

Definendo  $\tau := RC$  costante di tempo del circuito è quindi possibile scrivere la Funzione di Trasferimento come:

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{sRC + 1} = \frac{1}{s\tau + 1} \quad (1.16)$$

Si suppone di trascurare il regime transitorio e di avere quindi in ingresso un segnale sinusoidale ad ampiezza e frequenza costanti; in questo caso si ha  $s = j\omega$  e si può scrivere:

$$\frac{V_{out}(j\omega)}{V_{in}(j\omega)} = \frac{1}{j\omega\tau + 1} \quad (1.17)$$

Una volta ottenuta la Funzione di Trasferimento è possibile sostituire l'intero blocco circuitale come in figura 1.3.

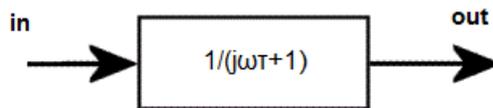


Figura 1.3: Diagramma a Blocchi di un Circuito RC

Qualsiasi sistema può essere rappresentato nel modo visto nell'esempio, ed un insieme di sistemi *in serie* può essere rappresentata tramite blocchi in cascata come mostrato in figura 1.4. Dalle Funzioni di Trasferimento dei singoli sistemi è possibile ricavare la funzione *equivalente* in questo modo:

$$G(j\omega) = \frac{\theta_{out}}{E}(j\omega) = \frac{A}{E}(j\omega) \frac{B}{A}(j\omega) \frac{\theta_{out}}{B}(j\omega) \quad (1.18)$$

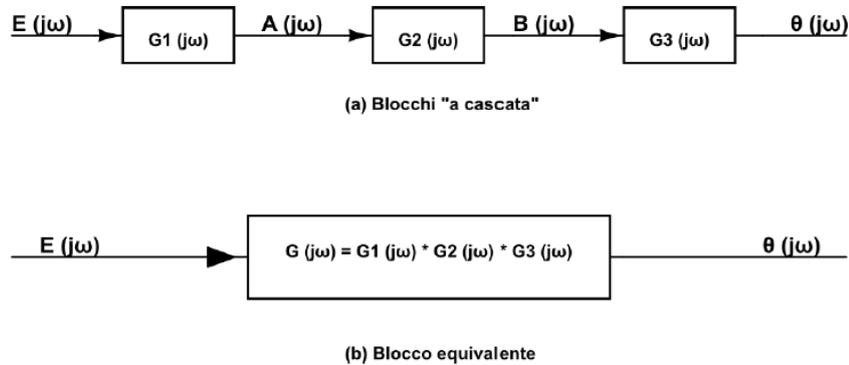


Figura 1.4: Blocchi in Serie

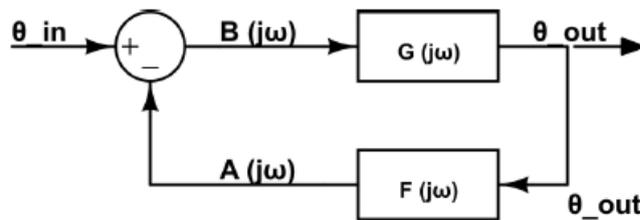


Figura 1.5: Schema di un Controllo a Retroazione

Poichè:

$$\begin{cases} \frac{A}{E}(j\omega) = G_1(j\omega) \\ \frac{B}{A}(j\omega) = G_2(j\omega) \\ \frac{\theta_{out}}{B}(j\omega) = G_3(j\omega) \end{cases}$$

Si ottiene:

$$\begin{aligned} G(j\omega) &= G_1(j\omega)G_2(j\omega)G_3(j\omega) \\ &= G_1G_2G_3(j\omega) \end{aligned} \quad (1.19)$$

Nel caso invece di un controllo retroazionato, illustrato in figura 1.5, è necessario aggiungere la funzione di Trasferimento dell'elemento di Feedback, che in questo caso è denotata con  $F(i\omega)$ ; inoltre il segnale di output  $\theta_{out}$  è confrontato con un valore di set denominato *setpoint* fornito in input,  $\theta_{in}$ , che è il valore a cui si vuole portare il sistema. In questo caso il segnale in ingresso al sistema da controllare è differenza tra il primo e il secondo (l'*errore*). Nel caso in cui il sistema di Feedback e il sistema che misura l'errore siano ideali e non contengano componenti aggiuntive si può assumere:  $F(i\omega) = 1$ . Osservando la figura

è chiaro che:

$$\frac{\theta_{out}}{B}(j\omega) = G(j\omega) \quad (1.20)$$

$$\frac{A}{\theta_{out}} = F(j\omega) \quad (1.21)$$

$$B(j\omega) = \theta_{in}(j\omega) - A(j\omega) \quad (1.22)$$

Sostituendo l'equazione 1.22 nella 1.20 si ottiene:

$$\begin{aligned} \theta_{out}(j\omega) &= [\theta_{in}(j\omega) - A(j\omega)]G(j\omega) \\ &= [\theta_{in}(j\omega) - \theta_{in}(j\omega)F(j\omega)]G(j\omega) \\ &= \theta_{in}(j\omega)G(j\omega) - \theta_{out}(j\omega)F(j\omega)G(j\omega) \end{aligned} \quad (1.23)$$

quindi:

$$\theta_{out}(j\omega)[1 + F(j\omega)G(j\omega)] = \theta_{in}(j\omega)G(j\omega) \quad (1.24)$$

Infine la Funzione di Trasferimento complessiva è data da:

$$\frac{\theta_{out}}{\theta_{in}}(j\omega) = \frac{G(j\omega)}{1 + F(j\omega)G(j\omega)} \quad (1.25)$$

### 1.3.2 Studio delle Funzioni di Trasferimento: Classificazione e Zeri & Poli

Una Volta ricavata la funzione di trasferimento è utile analizzarla, in particolare sono di forte rilevanza la classificazione della funzione e lo studio dei suoi Zeri e Poli. Si consideri una funzione di trasferimento del tipo:

$$G(s) = \frac{b(s)}{a(s)}$$

Ricordando che i termini  $a$  e  $b$  sono polinomiali, poiché sono le trasformate di Laplace dei segnali di input e output, è possibile classificare la funzione in base al *grado* dei due polinomi: detti  $n$  il grado di  $a$  ed  $m$  il grado di  $b$  si definisce *Grado Relativo* della funzione di trasferimento la quantità

$$r := n - m$$

ovviamente vale  $r \in \mathbb{Z}$ .

I sistemi lineari si suddividono in tre macro categorie in base al grado della funzione di trasferimento:

- Per  $r \geq 1$  la funzione si dice *Strettamente Propria*: il sistema è fisicamente realizzabile e la funzione  $G(s)$  si può scomporre in una somma di fratti semplici come illustrato più avanti.
- Per  $r = 0$  la funzione è *Propria*: dividendo i polinomi si ottiene la somma di una costante e di una funzione strettamente propria, che può essere scomposta in fratti semplici.
- Per  $r \leq -1$  la funzione si dice *Impropria*: si perde il rapporto causa effetto tra l'input e l'output in quanto l'uscita "anticipa" temporalmente l'azione dell'ingresso, il sistema non è quindi fisicamente realizzabile.

Tornando ad osservare la funzione  $G(s)$  si ha che le radici del polinomio  $a(s)$  sono dette *poli* del sistema, mentre le radici di  $b(s)$  sono gli *zeri*.

Una volta ricavati i poli, è possibile fattorizzare la funzione in fratti semplici nel modo seguente. Data la Funzione di Trasferimento  $G(s)$  sopra:

$$G(s) = \frac{b(s)}{(s - a_1)(s - a_2) \dots (s - a_n)} = \sum_{i=1}^n \frac{K_i}{s - a_i} \quad (1.26)$$

dove:

- $n$  è il grado del polinomio
- $a_i$  sono le radici del polinomio  $a(s)$
- $K_i$  sono i *residui* relativi ai vari poli

Eseguendo l'antitrasformata di Laplace di  $G(s)$  nella forma sopra si ottiene:

$$\mathcal{L}[G(s)] = g(t) = \sum_{i=1}^n K_i e^{a_i t} \quad (1.27)$$

Ogni polo corrisponde quindi ad un *modo* del sistema e l'evoluzione libera del sistema dopo un impulso arbitrario è data da una somma pesata di modi come sopra.

Gli zeri invece indicano le componenti della frequenza che risultano bloccate a regime: calcolando gli zeri è possibile tenere conto delle frequenze che non lasciano passare alcun segnale.

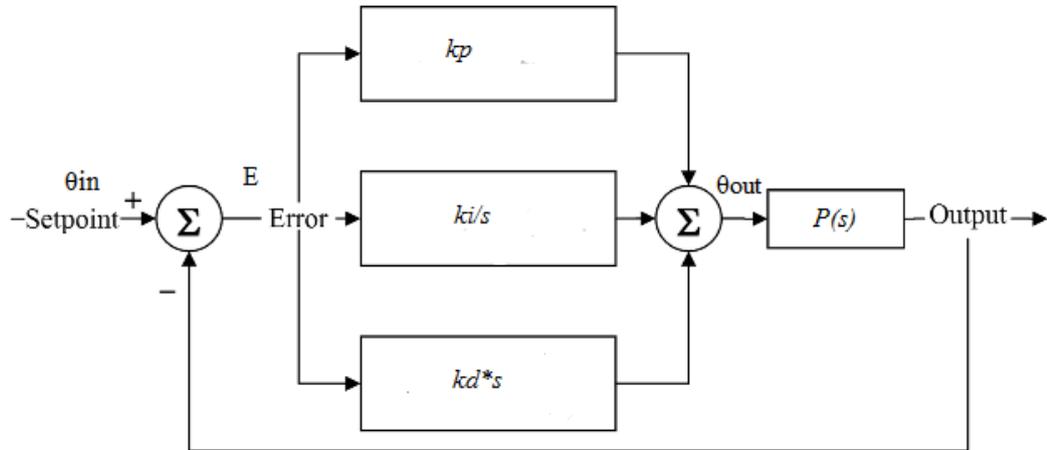


Figura 1.6: Diagramma a Blocchi di un Controllo PID

## 1.4 I Controlli PID

Il sistema di Controllo a retroazione più utilizzato nella progettazione di sistemi ingegneristici è il Controllo *Proporzionale Derivativo Integrativo*, o *PID*; tale popolarità è dovuta principalmente alla loro semplicità ed alla possibilità di ottenere buone prestazioni in numerose condizioni di lavoro. Come suggerito dal nome, un algoritmo PID consiste di tre coefficienti: proporzionale, integrativo e derivativo che verranno esaminate a breve; queste agiscono sull'errore calcolato tra l'uscita del sistema, misurata tramite un sensore, ed un *setpoint* fornito in input che rappresenta il valore dell'uscita desiderato. Lo schema a blocchi di un controllo di questo tipo è mostrato in Fig. 1.6.

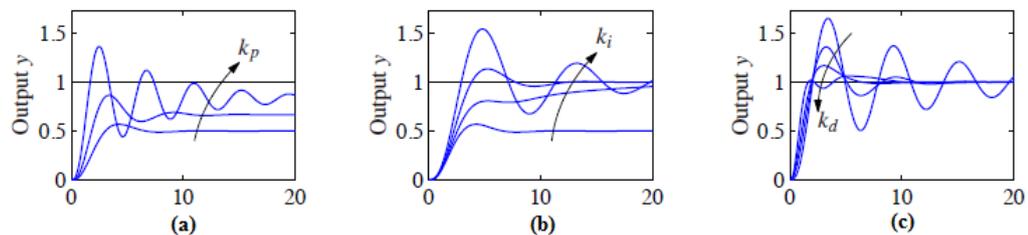


Figura 1.7: Comportamento di un controllo: (a)P; (b)PI; (c)PID. Le figure illustrano il comportamento del sistema con l'introduzione di un segnale di ingresso a gradino unitario e le diverse curve mostrano l'andamento al crescere di  $k_p$ ,  $k_i$  e  $k_d$

- **Componente Proporzionale**

La componente proporzionale consiste semplicemente nell'errore in ingresso, moltiplicato per una costante  $k_p$ : l'equazione che definisce l'uscita è quindi

$$\theta_{out} = k_p E$$

Un controllo di questo tipo è detto semplicemente P. La costante serve ad impostare il guadagno della risposta all'errore: aumentare  $k_p$  rende la risposta, e quindi il controllo, più veloci, ma un valore troppo alto può causare un'instabilità del sistema. In linea teorica un controllo può essere costituito semplicemente dalla componente proporzionale, tuttavia in questo modo il sistema tende ad assestarsi ad un valore stabile inferiore al setpoint, come si può vedere in fig 1.7. Si può aggirare questo problema semplicemente aggiungendo una costante alla relazione 1.4, detta costante di *reset*:

$$\theta_{out} = k_p E + r$$

Tale costante, però, è arbitraria e dovrebbe essere ricalibrata ogni volta che venisse variato il setpoint. Per questo motivo viene introdotta la componente integrativa.

- **Componente Integrativa** Come già accennato questa componente fa la funzione di *reset variabile*, integrando l'errore ed aumentando la risposta finché l'output non raggiunge il setpoint desiderato. L'uscita di un controllo di questo tipo, che possiamo chiamare *controllo PI*, è del tipo:

$$\theta_{out} = k_p E + k_i \int_0^t E(\tau) d\tau$$

Una componente integrativa troppo alta in sistemi non ideali può portare ad un fenomeno detto di *windup* che generalmente crea problemi alla stabilità del sistema e che verrà descritto in seguito (Vedi sez. 3.3.1).

- **Componente Derivativa** La componente derivativa si rende necessaria per contrastare il fenomeno dell'*overshoot*, ovvero la tendenza del sistema a superare il setpoint alla prima oscillazione, come illustrato il figura 1.7. Questa è proporzionale alla derivata dell'errore e fa in modo che l'uscita del controllo diminuisca quando l'errore ha variazioni troppo repentine. L'introduzione di questa componente completa la relazione ingresso/uscita del PID, che prende quindi la forma:

$$\theta_{out} = k_p E + k_i \int_0^t E(\tau) d\tau + k_d \frac{dE(t)}{dt} \quad (1.28)$$

Dalla equazione 1.28 dell'uscita del controllo è possibile calcolare la funzione di trasferimento come:

$$\begin{aligned} G_{PID}(s) &= G_p(s) + G_i(s) + G_d(s) \\ &= k_p + \frac{k_i}{s} + k_d s \end{aligned} \tag{1.29}$$

Una volta realizzato il sistema di controllo PID è necessario trovare una stima (o *tuning*) per le tre costanti: queste infatti sono fortemente dipendenti dalla natura del sistema da controllare. Per farlo esistono diversi metodi empirici: nella presente trattazione è stato preso in considerazione il metodo di *Ziegler-Nichols*; poiché però, date le caratteristiche del sistema in esame che vedremo più avanti, è stato necessario sfruttare un algoritmo modificato rispetto questa versione, il *tuning* delle costanti verrà mostrato in seguito.

# Capitolo 2

## L'Apparato Sperimentale

L'apparato sperimentale è costituito essenzialmente di due sezioni: il sistema fisico ed il sistema di controllo, ovvero il cablaggio che consente il dialogo del primo con il generatore di corrente, la scheda Arduino ed il software LabVIEW. Le due sezioni sono esaminate nel dettaglio di seguito.

### 2.1 Sistema Fisico

Questa sezione è composta da una camera a vuoto, al cui interno sono posti: il *fornetto*, la cui temperatura si vuole controllare, e la termocoppia per la misurazione della temperatura. Il vuoto è creato per mezzo di due pompe che vedremo più avanti, di seguito sono incluse alcune foto dell'apparato.



Figura 2.1: L'apparato nel suo insieme

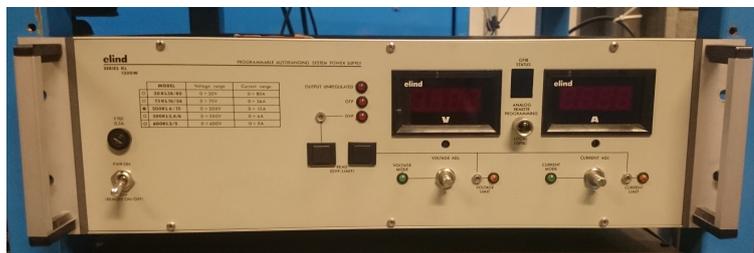


Figura 2.2: Il generatore di corrente/tensione

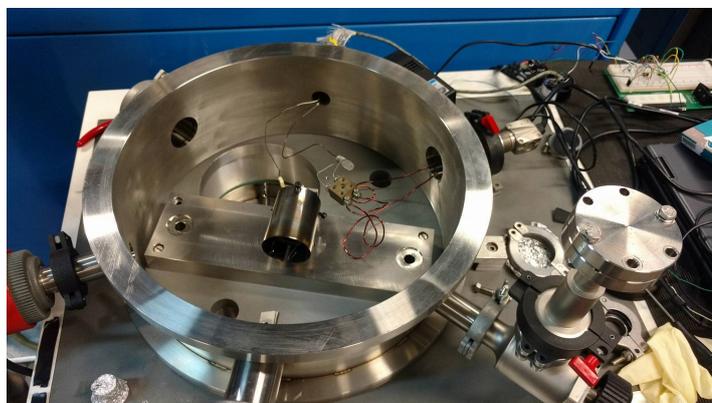


Figura 2.3: La camera a vuoto aperta

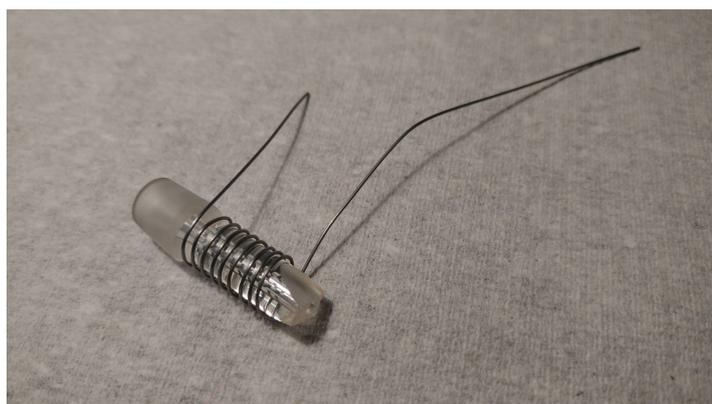


Figura 2.4: Particolare del fornetto: è possibile notare il foro sul fondo della provetta



Figura 2.5: La pompa rotativa



Figura 2.6: Il misuratore di pressione Pirani, sullo sfondo si può intravedere la valvola a tre vie in posizione di roughing, e sulla sinistra una parte della pompa diffusiva

### 2.1.1 Il Fornetto

Il fornello è essenzialmente una provetta in vetro con il fondo forato per permettere l'ingresso della termocoppia e la misurazione della temperatura all'interno; attorno alla provetta è avvolto un filo costituito di *molibdeno* al 99,95%, di sezione  $0,5\text{mm}$ , che svolge la funzione di resistenza e scalda il campione da analizzare. Il fornello è stato costruito di dimensioni ridotte per minimizzare l'*inerzia termica*: poiché infatti la provetta è posta in vuoto, lo scambio di calore non può avvenire per *convezione*, ma solamente tramite *conduzione* e *irraggiamento*. Si è visto in realtà, la dispersione di calore tramite conduzione sul filo di rame che porta corrente, domina sull'irraggiamento e quest'ultimo è quindi trascurabile.

### 2.1.2 Camera a Vuoto e Sistema di Pompaggio

La camera a vuoto è dotata di diversi passanti per permettere l'inserimento di cavi elettrici, come quelli utilizzati per riscaldare il fornello, e di uno spettrometro. Il vuoto all'interno è creato per mezzo di una coppia di pompe: una *rotativa* ed una *diffusiva*.

La pompa rotativa permette di portare il sistema ad un vuoto medio ( $\approx 10^{-2} \text{ mBar}$ ): è collegata alla camera tramite una valvola a tre vie con cui è possibile decidere dove la pompa deve aspirare aria. La valvola permette alla pompa di lavorare in: *modalità roughing*, aspirando aria direttamente dalla camera o *modalità backing*, aspirando dalla pompa diffusiva; in alternativa può chiudere il condotto. La pressione è quindi misurata tramite un misuratore di pressione di tipo *Pirani* posto nel tubo di aspirazione, prima della valvola. Il posizionamento in questo punto ha vantaggi e svantaggi: consente infatti di misurare la pressione indipendentemente dalla posizione della valvola, ma il valore letto non corrisponde esattamente alla pressione nella camera. I valori misurati sono riportati dall'Active Gauge Controller.

La pompa a diffusione, lavorando in coppia con la pompa rotativa, riesce a portare la camera in vuoto spinto, a circa  $10^{-6} \text{ mBar}$ . È necessario utilizzarla per gli esperimenti di spettroscopia, ma non è stata sfruttata per lo sviluppo del software di controllo.

## 2.2 Apparato di Controllo

L'acquisizione dei dati sulla temperatura avviene tramite una termocoppia collegata ad una scheda *Arduino Uno* e posta all'interno della provetta, tramite il foro sul fondo. Il controllo della temperatura viene gestito invece tramite

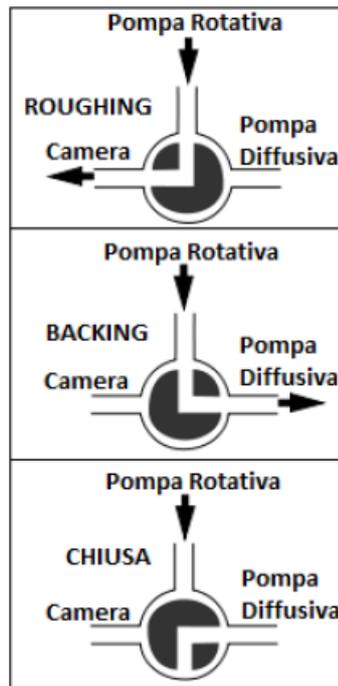


Figura 2.7: Funzionamento della valvola a tre vie

software LabVIEW (descritto nel Capitolo 4), la scheda Arduino ed un circuito su breadboard permettono infine il dialogo del software con il generatore di tensione e corrente *ELIND 200KL 6/15*.

### 2.2.1 Termocoppia

La termocoppia è un sensore di temperatura abbastanza semplice: consiste di due fili di diverso materiale, che per effetto Seebeck si portano a potenziale elettrico diverso. In questo modo, tramite una relazione non lineare, è possibile misurare la temperatura. La termocoppia utilizzata è di *tipo K (Chromel (Ni-Cr)(+) / Alumel (Ni-Al)(-))*, il segnale in uscita è processato da un *linearizzatore e amplificatore* modello *Minco TT190* e letto da uno dei canali di input analogico della scheda Arduino.

### 2.2.2 Scheda Arduino Uno

Per l'interfaccia tra il software e le componenti hardware per il controllo del fornello si è scelto di utilizzare una scheda Arduino Uno. Il modello Uno è il primo di una serie di piattaforme sviluppate dall'*Interaction Design Insti-*

*tute* di Ivrea, particolarmente apprezzate da privati ed hobbisti per il basso costo e la facilità di programmazione, garantita da un ambiente di sviluppo integrato e multiplatforma. Generalmente una scheda Arduino comprende: un microcontrollore *ATmega*, un regolatore di tensione e ingresso USB per la comunicazione con un computer e diversi pin di I/O. In particolare il modello Uno si basa su un controllore *ATmega328P*, dispone di pin di *ground* e *vcc* a 5 e 3,3 Volts e possiede pin di analog input e digital I/O, con possibilità di funzionare in modalità *PWM*. Per la comunicazione con il software LabVIEW è stato utilizzato il sistema di interfaccia LINX descritto nel Cap. 4. Poiché le uscite PWM hanno caratteristiche fondamentali per la gestione dell'apparato, sono descritte in dettaglio di seguito.

### 2.2.3 Le Uscite PWM

PWM è un acronimo che sta per *Pulse-Width Modulation*: è un tipo di modulazione digitale che serve per simulare un segnale analogico attraverso l'utilizzo del *duty cycle*. Nel sistema di controllo in esame le uscite PWM sono state utilizzate per permettere al software di pilotare la corrente e la tensione erogate dal generatore di tensione/corrente.

Il duty cycle di un dispositivo elettronico digitale, che ha quindi solo due stati *attivo* e *non attivo*, è definito come il *rapporto tra il tempo che il dispositivo passa nello stato attivo ed il tempo totale di lavoro*; un'uscita PWM subisce una variazione continua del duty cycle in modo da ottenere in uscita una tensione media variabile dipendente dalla variazione di quest'ultimo. si definisce quindi il duty cycle come:

$$d = \frac{\tau}{T} \quad (2.1)$$

dove  $\tau$  è il lasso di tempo passato nello stato "attivo" e  $T$  il periodo totale: poiché la frequenza di default delle uscite PWM di Arduino Uno è  $\approx 980Hz$ , un periodo vale  $T \approx 1ms$ .

Data la definizione 2.1 è chiaro che: ad un valore  $d = 0$  corrisponde una tensione nulla e a  $d = 1$  la massima tensione continua consentita dal dispositivo, che nel caso della scheda Arduino è di 5V. La risoluzione delle uscite digitali di Arduino è di 8 bit, quindi si hanno  $2^8 = 256$  livelli possibili di tensione, ovvero  $\frac{5}{256} \approx 0,02$  Volts per step.

Nel caso in cui il sistema debba disporre di molta potenza, utilizzare un output PWM con risoluzione troppo bassa potrebbe portare ad un'uscita rumorosa e ad una conseguente instabilità del sistema di controllo. Come mostrato in seguito (Sez. 3.3.2), si è calcolato che la potenza media dissipata dal fornetto è di  $\approx 15Watt$ , si hanno quindi  $\frac{15}{256} \approx 0,06$  Watt per step PWM: la risoluzione della scheda è quindi accettabile per il sistema in esame.

### 2.2.4 Filtri Passa Basso ed Amplificatori Operazionali

I Filtri passa basso sono utilizzati per permettere in passaggio di segnali con frequenza inferiore ad una *frequenza di taglio* predeterminata. In questo caso sono realizzati con un circuito RC in serie, come quello in fig. 1.2, e la frequenza di taglio è data da  $\frac{1}{RC} = \frac{1}{\tau} := \omega_0$ . Poiché l'utilità è unicamente quella di limitare eventuali "sbalzi" nelle uscite PWM, si sono scelte R e C in modo che  $\omega_0 = 1kHz$ , di poco superiore ai  $980Hz$  di default di Arduino.

L'uscita dai filtri è quindi passata attraverso degli amplificatori differenziali, che hanno funzione di adattatori di impedenza: poiché infatti le entrate del controllo remoto del generatore ELIND sono a bassa impedenza, il collegamento diretto al filtro passa basso passivo vanificherebbe il funzionamento di quest'ultimo. L'amplificatore ha impedenza teorica in ingresso infinita e in uscita virtualmente nulla, in questo modo evita i disturbi sui segnali in arrivo al generatore. Il modello utilizzato è il  $\mu A741CP$ .

### 2.2.5 Generatore di Tensione/Corrente

Il generatore di corrente utilizzato è il modello *ELIND 200KL 6/15*, di cui è riportata una foto fig. 2.2, può fornire fino a  $1,2 kW$  di potenza ed ha la possibilità di essere pilotato manualmente o in remoto, tramite ingressi posti sul retro. In entrambe le configurazioni il generatore può lavorare in *current* o *voltage mode*, ovvero può lavorare come generatore di corrente o generatore di tensione.

Per fare un esempio di funzionamento supponiamo di voler utilizzare il generatore in *current mode*, poiché questa è la modalità che si è scelta per il pilotaggio da parte del software: si fissa un certo voltaggio, quindi è possibile variare l'erogazione di corrente finché la tensione richiesta è inferiore al valore fissato, se questa supera tale valore l'erogazione di corrente smette di crescere e il generatore passa in *voltage mode*. Quindi mantenendo la tensione al valore massimo possibile è possibile utilizzare il generatore sempre in *current mode*. Il generatore eroga un massimo di  $200V$  di tensione o  $15A$  di corrente.

Il generatore è pilotabile dagli ingressi di controllo remoto secondo le formule:

$$\begin{cases} V_{out} = k_V \cdot V_{V,in} & \wedge & k_V = 20 \\ C_{out} = k_C \cdot V_{C,in} & \wedge & k_C = 1,5 \end{cases} \quad (2.2)$$

Ma dato che le uscite PWM di Arduino erogano un massimo di  $5V$ , i valori massimi ottenibili sono:

$$\begin{cases} V_{max} = 100 V \\ C_{max} = 7,5 A \end{cases} \quad (2.3)$$

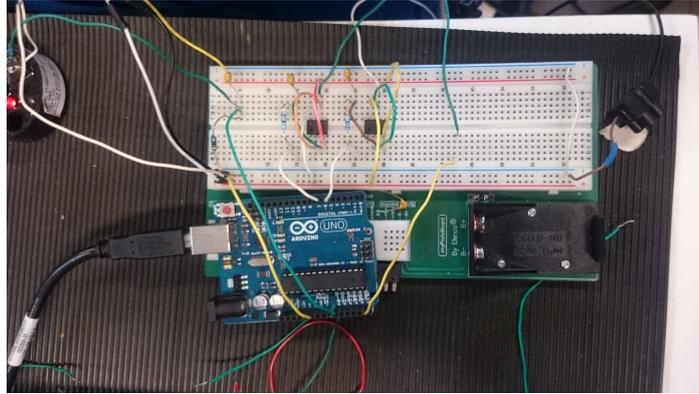


Figura 2.8: Scheda Arduino Uno e circuito di misura e controllo

In realtà, viste le numerose componenti passive presenti nel circuito, è stato necessario ricalibrare la costante di corrente: sono state effettuate diverse misure richiedendo un valore di corrente dal software ed osservando tramite il display quella effettivamente erogata, ottenendo quindi il risultato:

$$k_C = 1,6125$$

# Capitolo 3

## Studio Del Sistema

In questo capitolo sono descritte le operazioni svolte per studiare il comportamento del fornetto in diverse situazioni. Queste sono state necessarie per lo sviluppo di un sistema di controllo automatico che tenesse conto delle caratteristiche specifiche: del fornetto, del generatore di corrente e della termocoppia utilizzati.

### 3.1 Studio a Corrente Costante

Per prima cosa si è studiata la risposta del sistema ad un input a gradino, ovvero il comportamento con un input di corrente costante: da questo semplice studio sono emerse diverse caratteristiche peculiari del sistema. È stato così possibile osservare, come vedremo, che il fornetto appartiene alla categoria dei sistemi di II Ordine: questo ha permesso di sviluppare preliminarmente un controllo PID che lavorasse su un sistema di questo tipo simulato da LabVIEW, descritto nella sez.3.2. Inoltre è stato possibile determinare un "range operativo" di correnti per il funzionamento del sistema.

#### 3.1.1 Cenni sulla classificazione dei sistemi lineari

Si è già visto nel Cap. 1 come ad un dato sistema possa essere associata una funzione di trasferimento che lo descrive, tramite l'utilizzo della trasformata di Laplace, definita dall'equazione 1.11.

Si suppone ora di avere un sistema con segnali di input ed output rispettivamente  $u(t)$  e  $y(t)$ , e funzione di trasferimento

$$G(s) = \frac{Y(s)}{U(s)}$$

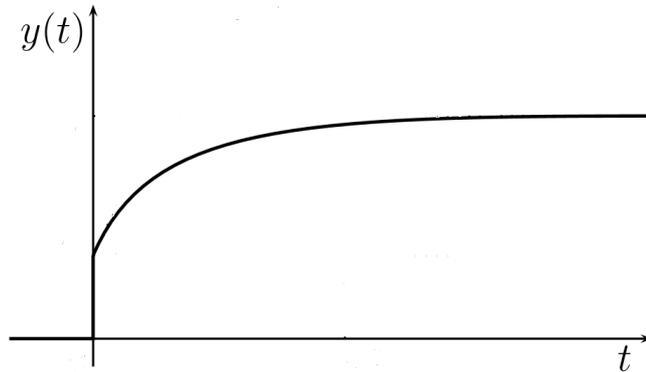


Figura 3.1: Risposta al gradino di un sistema del I Ordine

dove  $U(s)$  e  $Y(s)$  sono le trasformate di Laplace di  $u(t)$  e  $y(t)$ . Si suppone inoltre di applicare al sistema un segnale di input unitario a gradino del tipo:

$$u(t) = \begin{cases} 0 & \text{per } t < 0 \\ 1 & \text{per } t > 0 \end{cases}$$

- **Sistemi del I Ordine**

Sono detti del I Ordine i sistemi lineari che possono essere rappresentati da equazioni differenziali lineari del I ordine a coefficienti costanti del tipo:

$$a_1 \dot{y} + a_0 y(t) = b_1 \dot{u} + b_0 u(t) \quad (3.1)$$

Applicando la trasformata di Laplace all'equazione si ottiene:

$$(a_1 s + a_0) Y(s) = (b_1 s + b_0) U(s) \quad (3.2)$$

e la funzione di trasferimento diventa quindi:

$$G(s) = \frac{b_1 s + b_0}{a_1 s + a_0} \quad (3.3)$$

Facendo la trasformata inversa si ottiene l'equazione per l'output:

$$y(t) = \frac{b_1}{a_1} + \left( \frac{b_0}{a_0} - \frac{b_1}{a_1} \right) (1 - e^{-\frac{a_0}{a_1} t}) \quad (3.4)$$

La risposta ad un gradino unitario di input per sistemi di questo tipo è quindi data dal termine esponenziale ed ha una forma come quella mostrata in figura 3.1

Si nota come l'assenza di un termine massivo, che sarebbe dato da un termine derivativo di secondo grado, comporta che il sistema non abbia inerzia, e risponda immediatamente ad uno stimolo. Fanno parte dei sistemi di questo tipo, tra gli altri, i circuiti RC.

### • Sistemi del II Ordine

Sono detti invece del II Ordine i sistemi lineari che possono essere rappresentati da equazioni differenziali lineari del II ordine a coefficienti costanti del tipo:

$$a_2\ddot{y} + a_1\dot{y} + a_0y(t) = b_2\ddot{u} + b_1\dot{u} + b_0u(t) \quad (3.5)$$

Prendendo a titolo di esempio un sistema dove i coefficienti  $b_2$  e  $b_1$  sono nulli, e  $b_0 = 1$  la funzione di trasferimento diventa:

$$G(s) = \frac{1}{a_0} \frac{\omega^2}{s^2 + 2\delta\omega s + \omega^2} \quad (3.6)$$

Dove  $\omega = \frac{a_0}{a_2}$  è la *pulsazione* del sistema e  $\delta = \frac{a_1}{2a_0}$  il *coefficiente di smorzamento*.

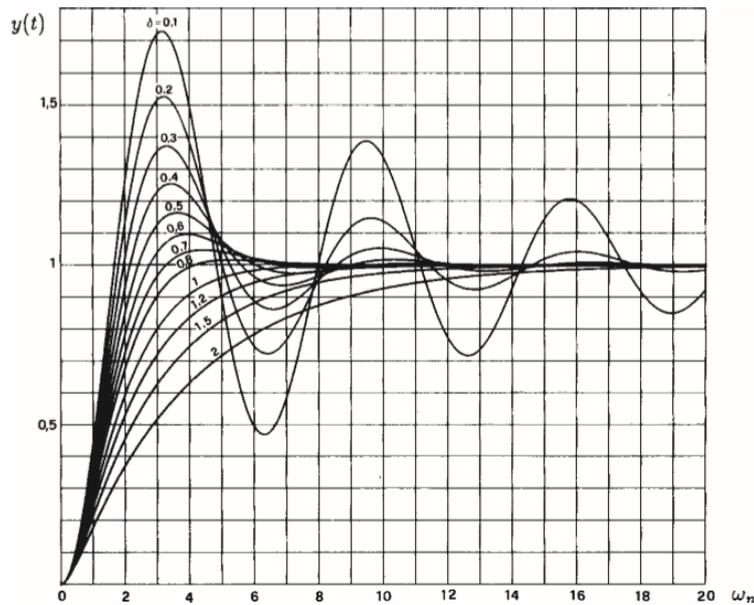


Figura 3.2: Risposta al gradino di un sistema del II Ordine in funzione della pulsazione e del coefficiente di smorzamento

Il grafico della risposta ad un input a gradino è riportato in figura 3.2, l'equazione si trova con l'antitrasformata di Laplace ed è data da una

sinusoide modulata da un esponenziale: in figura è possibile notare la dipendenza dal coefficiente di smorzamento ed il tipico comportamento "ad s" dato dalla presenza di un termine di inerzia (come una massa o un'induttanza) che si oppone alle variazioni richieste. Fanno parte di questo tipo sistemi come: i circuiti RLC, i sistemi massa/molla, i sistemi termici e molti apparati di controllo.

### 3.1.2 Fornetto a Corrente Costante

Questo studio è consistito nel fornire diversi valori di corrente costante al fornello, osservandone il comportamento dopo pochi minuti per vedere la risposta al gradino di input; si è quindi atteso un certo lasso di tempo per lasciare che il sistema si stabilizzasse e si è osservata la temperatura di arrivo. Questo accade poiché la potenza che il sistema dissipa dipende dalla temperatura: quando la potenza fornita dal generatore uguaglia quella dissipata il fornello si stabilizza ad una temperatura costante. I risultati ottenuti sono riportati in Fig 3.3

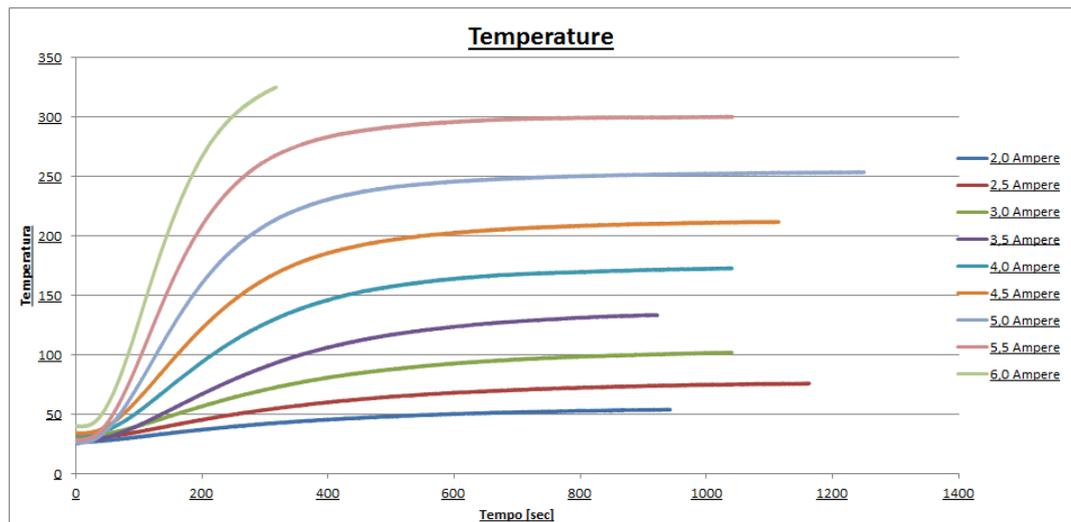


Figura 3.3: Comportamento con i diversi valori di corrente: è possibile notare il comportamento tipico dei sistemi del II Ordine e le temperature relative alle diverse correnti. La curva a 6A è interrotta poiché il sistema di sicurezza del software è progettato per interrompere l'erogazione di corrente se la temperatura supera i 350°C

Come risultato immediato di questo studio si è trovata una corrispondenza corrente/temperatura che è tornata utile nello sviluppo del controllo PID. Per fare un esempio: poiché le rampe di temperatura richieste per gli esperimenti non avrebbero superato i 20°C/min, dal grafico si è notato che le correnti

superiori a 6A difficilmente sarebbero state utili al sistema di controllo, nonostante il generatore fosse in grado, come già visto, di erogare fino a 7,5A; questo è stato confermato anche da un calcolo delle derivate di queste curve, per estrapolare la pendenza. Anche per questo motivo è stato quindi effettuato lo studio descritto nella sezione 3.3. Osservando la forma delle curve è inoltre chiara l'analogia con i sistemi del secondo ordine visti nella sezione precedente.

A titolo conoscitivo si fa notare che tramite un'analisi di questo tipo, eventualmente più dettagliata, ed il calcolo delle derivate di queste curve, sarebbe stato possibile arrivare allo sviluppo di un controllo ad anello aperto che, tramite una tabella codificata con le relazioni Pendenza/Temperatura/Corrente, potesse effettuare rampe di temperatura senza il bisogno di un controllo in retroazione; un sistema di questo tipo sarebbe stato però fortemente dipendente dalle condizioni ambientali e dalle qualità degli strumenti di misura, motivo per cui si è scelto di non percorrere questa strada.

## 3.2 La Simulazione Massa/Molla

Per prendere pratica con i controlli PID si è simulato con LabVIEW un sistema massa/molla smorzato: come già visto questo è un sistema classificato tra quelli di secondo ordine, che ben si adatta quindi alla simulazione di un sistema termico. L'equazione che lo descrive, risolta numericamente tramite software è:

$$m\ddot{x} = -kx - \beta\dot{x} + E \quad (3.7)$$

dove:

$m$  è la massa

$E$  è il segnale corretto in uscita dal PID

$k$  è il coefficiente della molla

$\beta$  è il coefficiente di smorzamento

Il PID riceve in input il tempo trascorso, il setpoint scelto dall'utente e la posizione istantanea della massa virtuale; trova l'errore e di questo calcola le tre componenti proporzionale, derivativa ed integrativa, le moltiplica per le rispettive costanti scelte dall'utente e restituisce quindi la forza da applicare alla massa. I risultati sono mostrati nelle figure 3.4, 3.5 e 3.6 e sono relativi ad un setpoint richiesto di 50. Il software PID realizzato per questa simulazione è sostanzialmente quello che si è sfruttato per il controllo del fornetto, descritto nella sezione 4.2.

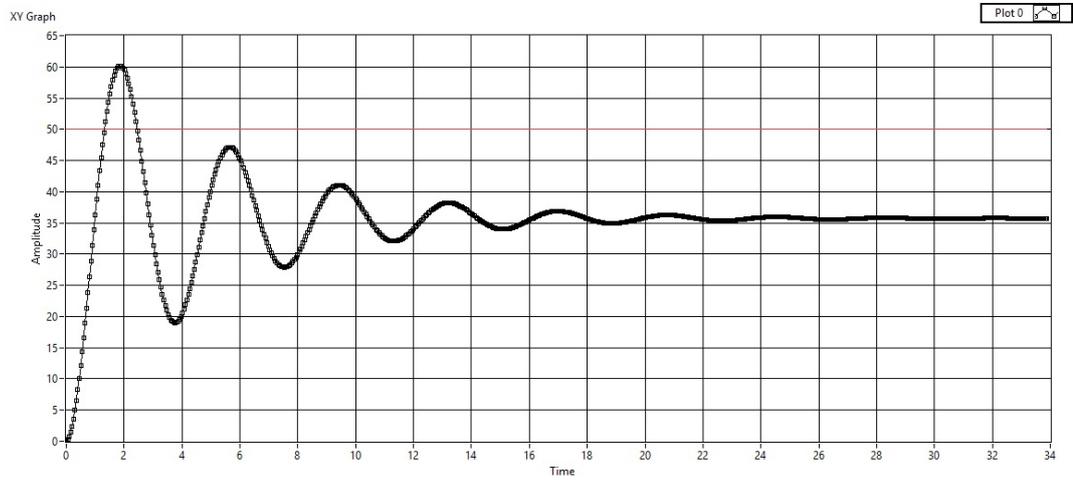


Figura 3.4: Controllo con la sola componente proporzionale: si nota che dopo qualche oscillazione il sistema si assesta ad una posizione inferiore a quella richiesta. In rosso è rappresentato il setpoint

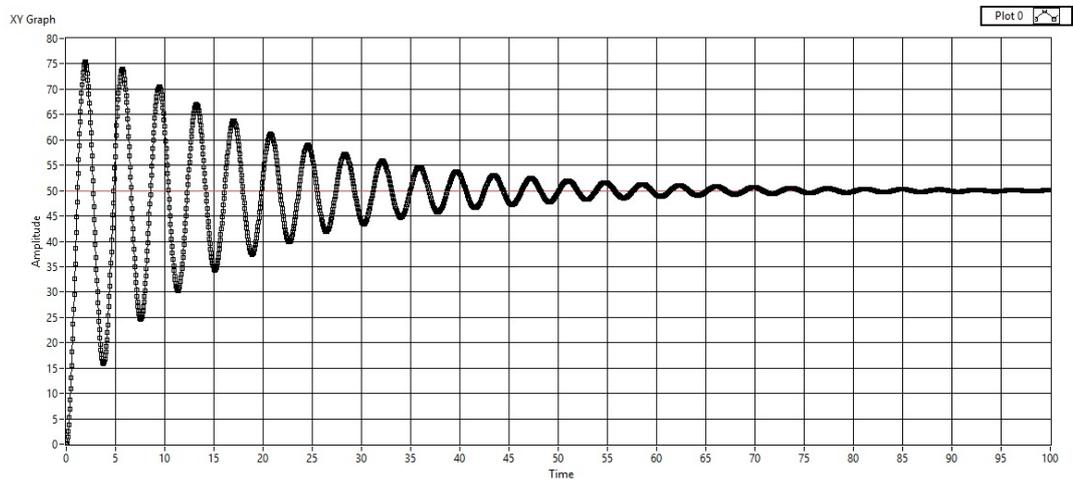


Figura 3.5: Aggiunta della componente integrativa: il sistema raggiunge il setpoint ma si può notare il tipico "overshoot" iniziale, ossia il superamento del setpoint

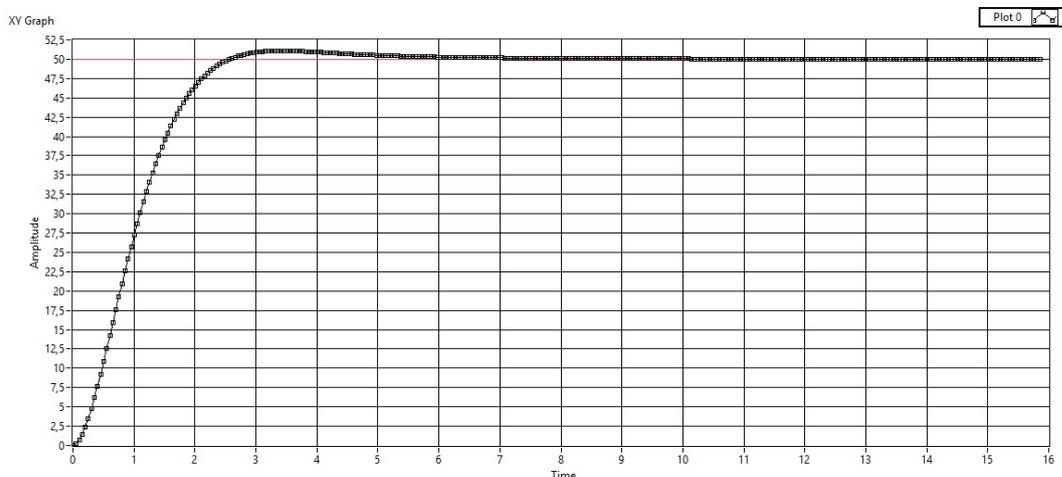


Figura 3.6: Con la componente derivativa vengono smorzate le eccessive oscillazioni iniziali e l'overshoot

## 3.3 Il Rapporto Corrente/Potenza

### 3.3.1 Il Windup della Componente Integrativa

Il *Windup* integrale è un fenomeno che si verifica nei sistemi di controllo non ideali: in particolare insorge quando il comando su cui il controllo PID agisce è vincolato da un valore massimo o minimo. In pratica, poiché l'elaborazione dell'errore nel PID è puramente matematica, può accadere che il segnale in uscita richieda all'attuatore (che può essere un generatore di corrente, una pressa, etc.) di erogare una potenza maggiore di quella possibile fisicamente: in questo caso la forza non è sufficiente per causare al sistema la variazione richiesta dal PID e questo impiega più tempo di quello previsto per raggiungere il setpoint. In questo lasso di tempo però la componente integrale continua ad accumularsi integrando l'errore, quindi una volta raggiunto il setpoint dovrà essere "scaricata" trascorrendo un simile lasso di tempo con un errore opposto. Tipicamente questo fenomeno aggiunge instabilità al sistema: nel migliore dei casi aumenta il tempo necessario per la stabilizzazione, mentre se il sistema ha già elementi di instabilità potrebbe portare ad "un'esplosione" delle componenti del PID.

Poiché inizialmente la richiesta di corrente del PID eccedeva spesso il limite del generatore di 7,5A andando in saturazione, si è deciso di studiare il rapporto corrente/potenza dissipata per fare in modo di evitare il windup.

### 3.3.2 Potenza dissipata

Per il calcolo della potenza dissipata si sono ancora una volta forniti al sistema diversi valori di corrente costanti, misurando questa volta il voltaggio corrispondente. Dai valori trovati è stato possibile calcolare la potenza dissipata tramite la relazione:

$$P = VI \quad (3.8)$$

L'idea di fondo alla base dello studio era di dare modo al PID di gestire la *potenza* dissipata dal forno e non, come accadeva in precedenza, direttamente la corrente erogata: la potenza sarebbe quindi stata convertita nella corrente necessaria una volta trovata una relazione che legasse le due quantità.

Per trovare suddetta relazione si è quindi realizzato un grafico che rappresenta la corrente in funzione della potenza. Dei punti trovati è stato poi effettuato un fit ottenendo la relazione logaritmica:

$$I = 1,4888 \log_2(P + 1) \quad (3.9)$$

I punti acquisiti e la curva di fit sono riportati in Fig. 3.7

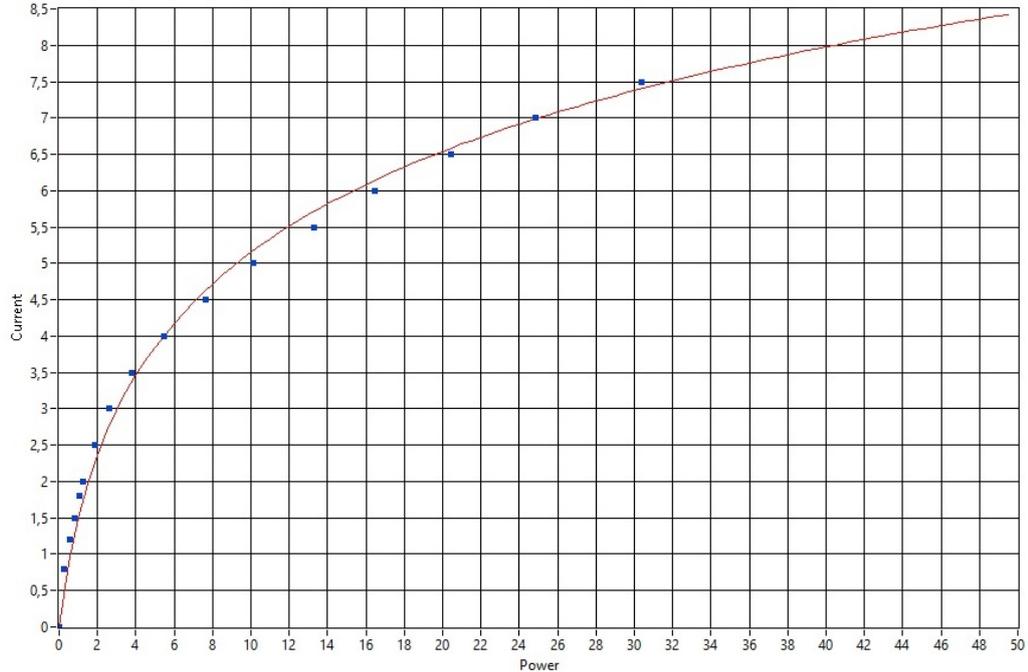


Figura 3.7: Grafico corrente/potenza: in blu sono rappresentati i punti acquisiti, in rosso invece la funzione di fit.

Trovata la giusta funzione si è fatto in modo di processare l'uscita del PID tramite quest'ultima, ottenendo il risultato desiderato. Osservando il grafico 3.7, infatti, appare chiaro come questo procedimento vincoli la corrente erogata in valori intermedi: solamente una grande richiesta di potenza comporterebbe una saturazione dell'erogazione di corrente e difficilmente il fornetto necessiterà di tale quantità di potenza. In definitiva questo studio si è rivelato molto utile ai fini di evitare il windup: infatti osservando la corrente erogata dal sistema in diverse condizioni di lavoro, non accade mai che la richiesta del sistema superi il limite di  $7,5A$  del generatore.



# Capitolo 4

## Il Software

Questo capitolo è dedicato al software realizzato per il controllo del fornello: saranno mostrate le varie sezioni del programma come il sistema di interfaccia LINX, il PID, i software di controllo veri e propri ed i vari accorgimenti per giungere ai risultati mostrati infine nel capitolo 5.

### 4.1 Il Sistema di Interfaccia LINX

Per l'interfaccia della scheda Arduino con il programma LabVIEW si è scelto di utilizzare la libreria *LINX* del *LabVIEW Maker Hub*: questa contiene le *VI* necessarie per la comunicazione in entrata e uscita con: diversi sensori compatibili con Arduino attualmente in commercio, oppure direttamente con tutte le periferiche di cui dispone la scheda. Per il sistema in esame erano state precedentemente realizzate due *sub-VI* ad hoc per la lettura dai canali analogici e la scrittura in digitale dalle uscite PWM, quest'ultima in particolare sfrutta la *VI* di *LINX* per l'impostazione del duty cycle della scheda. Nelle *VI* di lettura scrittura erano inoltre già state implementate le costanti menzionate nella sezione 2.2.5 per il controllo remoto del generatore: la costante di corrente è stata anche qui opportunamente modificata dopo la calibrazione svolta. Inoltre la *sub-VI* di lettura analogica incorpora un filtro di media mobile per far fronte a problemi di rumore del segnale della termocoppia.

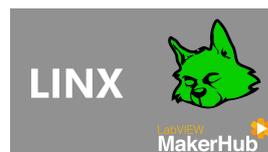


Figura 4.1: Logo di *LINX & LabVIEW Maker Hub*

## 4.2 La sub-VI PID

Come già ampiamente riportato in precedenza, il controllo automatico del fornetto è affidato ad un controllo di tipo proporzionale integrativo derivativo PID, che è stato realizzato in una sub-VI di LabVIEW ed è mostrato in Appendice A. Il funzionamento riprende i concetti teorici esposti nel primo capitolo: riceve in input l'errore precalcolato, le tre costanti  $k_p$ ,  $k_i$ ,  $k_d$  e l'intervallo di tempo di lavoro del programma e restituisce la correzione del segnale; inoltre è stata fornita la possibilità di ricevere in output in tempo reale le tre componenti prima che vengano sommate, utile principalmente per il debug. È stato anche implementato un sistema di *reset* della memoria integrale: il programma è fatto in modo da mantenere il controllo di reset attivo finché si è in modalità manuale, altrimenti rimarrebbero in memoria i valori delle prove precedenti, falsando il funzionamento del PID. Il controllo è molto utile anche nel caso in cui, per condizioni di partenza molto sfavorevoli, si verifichi il windup integrale: resettando la memoria si impedirà un accumulo eccessivo sulla componente integrativa evitando il fenomeno dell'overshoot.

Nonostante il principio di funzionamento sia identico a quello teorico, per la realizzazione pratica e l'adattamento agli strumenti di misura utilizzati sono stati necessari degli accorgimenti in più: in particolare si è dovuto far fronte al consistente rumore di cui sono affetti, nonostante il filtro di media mobile citato sopra, i valori di temperature riportati dalla termocoppia. Per questo la derivata dell'errore è calcolata effettuando un *fit lineare* su 100 punti ottenuti in ingresso, tramite una funzione preesistente in LabVIEW, tale procedimento ha permesso un calcolo più accurato della derivata, a discapito di un maggiore ritardo nell'esecuzione del software. L'integrale dell'errore è invece calcolato con il *metodo dei trapezi* secondo la formula:

$$\int_{t_1}^{t_n} f(t)dt \approx \frac{\Delta t}{2}[f(t_1) + f(t_n) + 2(f(t_2) + \dots + f(t_n))] \quad (4.1)$$

## 4.3 Software di Controllo

In questa sezione è mostrato il funzionamento del software di controllo vero e proprio, realizzato con LabVIEW: questo sfrutta sia la libreria LINX che la sub-VI PID mostrate in precedenza. Sono stati realizzati due software: il primo, chiamato *Isoterma*, è volto al raggiungimento ed al mantenimento di una temperatura richiesta, il secondo invece, chiamato *Rampa*, consente il mantenimento di una rampa costante di temperatura.

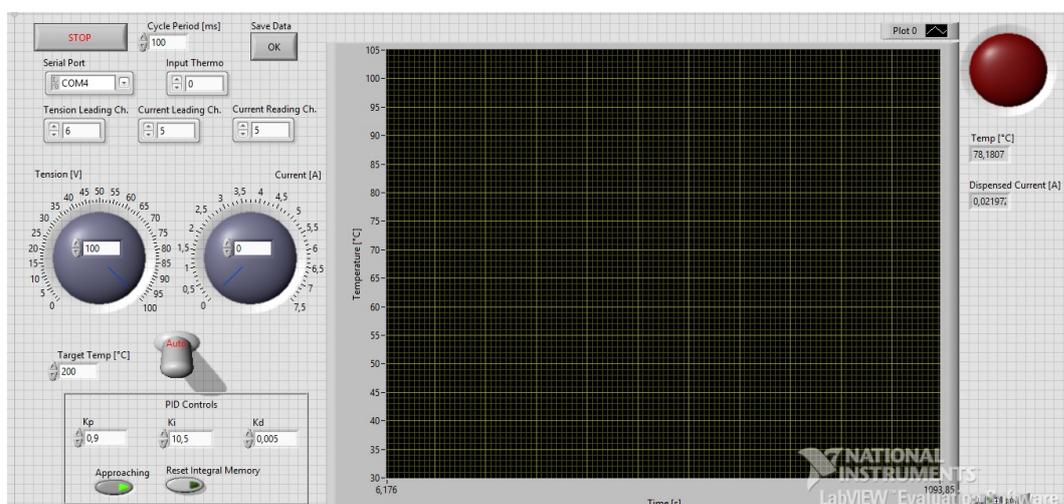


Figura 4.2: Front Panel del software "Isotherma"

### 4.3.1 Isotherma

Come già accennato, lo scopo di questo software è di stabilizzare la temperatura del fornetto a quella richiesta: in questo caso il setpoint è la temperatura richiesta e l'errore è calcolato basandosi su questa. Il pannello posteriore è riportato in Appendice B.1, in Fig. 4.2 è invece illustrato il *front panel*. Il programma permette di lavorare in modalità Manuale e Automatica. La prima è sostanzialmente una replica a livello software del pannello frontale del generatore di tensione/corrente: tramite le due manopole è possibile regolare direttamente la tensione o la corrente da erogare; passando in modalità Automatica invece entra in azione il controllo PID ed il programma si porta automaticamente alla temperatura desiderata. I valori di temperatura del fornetto sono quindi osservabili nel grafico, inoltre è presente un LED di allerta che si attiva quando la temperatura eccede un valore di sicurezza preimpostato, oltre il quale il programma stacca l'erogazione di corrente.

L'estetica del programma è rimasta sostanzialmente invariata rispetto alle versioni precedenti poichè le modifiche sono avvenute principalmente a livello del pannello posteriore; è però stato inserito il pulsante "*Approaching*" che mantiene a zero la memoria integrativa e la costante  $k_i$ : questo controllo è volto sostanzialmente ad evitare il windup dell'integrale, ed è consigliabile mantenerlo attivo durante la fase di avvicinamento al setpoint. Infatti se il setpoint è molto alto l'errore iniziale sarà molto grande, la componente integrativa diventerà immediatamente eccessiva, saturando il controllo di corrente e causando il windup come descritto in Sez. 3.3.1; mantenendola invece disattivata fino al raggiungimento del setpoint questa si stabilizzerà subito su di un valore accet-

tabile, evitando l'overshoot.

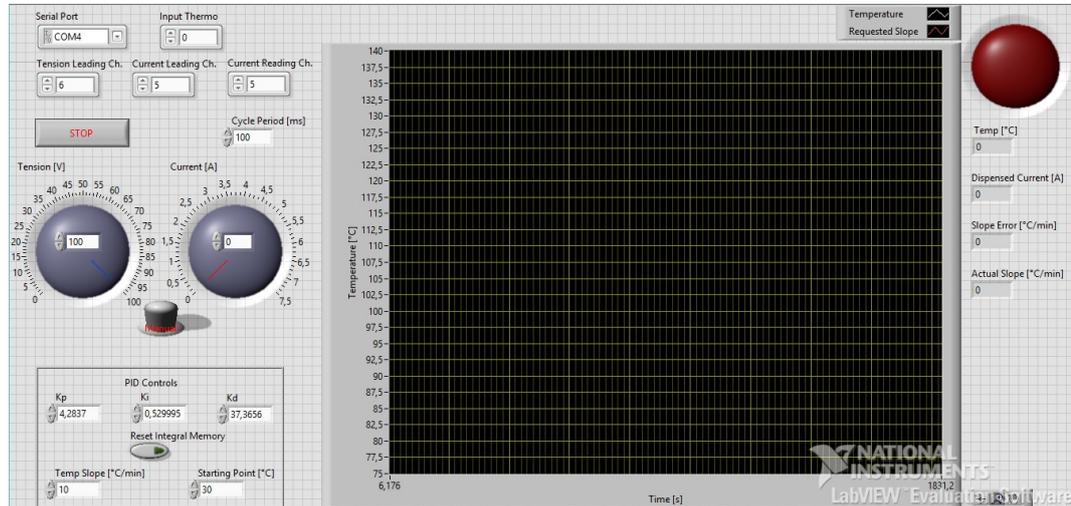


Figura 4.3: Front Panel del software "Rampa"

### 4.3.2 Rampa

Come suggerito dal nome questo software, la cui realizzazione ed ottimizzazione era lo scopo della tesi, serve per mantenere rampe costanti di temperatura, di pendenza costante scelta dall'utente. Il pannello posteriore è riportato in Appendice B.2, mentre il front panel in Fig. 4.3. Il software dispone delle stesse modalità di funzionamento Manuale e Automatica del precedente ma, se la prima lavora in modo identico, la seconda è stata profondamente variata. La modifica più sostanziale è stata l'introduzione del setpoint come pendenza e non più come temperatura: precedentemente infatti il programma lavorava con un setpoint di temperatura che veniva gradualmente alzato secondo la pendenza scelta in ingresso. Questo introduceva forti instabilità poiché il sistema non aveva il tempo di raggiungere il punto di arrivo e stabilizzarsi, ma rimaneva costantemente in fase di "avvicinamento", di conseguenza le componenti del PID non avevano il tempo di giungere a condizioni di lavoro stabili. Questa modifica invece fa in modo che il sistema riceva comunque un setpoint fisso, nonostante la temperatura continui a crescere. La pendenza della temperatura è calcolata come nella sub-VI PID, tramite un fit lineare di 100 punti raccolti da un buffer. È inoltre presente il tasto *Reset Integral Memory*, utile se si necessita di attivare la modalità Automatica in condizioni iniziali molto sfavorevoli, ad esempio durante il raffreddamento quando la pendenza è negativa; è offerta

anche la possibilità di disegnare sul grafico una linea "*guida*" che rappresenti la pendenza richiesta. Si fa notare come con questo software non sia possibile ottenere una temperatura costante, dal momento che il setpoint non è una temperatura.



# Capitolo 5

## Risultati Sperimentali e Conclusioni

L'obiettivo finale della tesi era la realizzazione di un software che permettesse di effettuare rampe regolari di temperatura con pendenza a scelta. Per giungere ai risultati richiesti sono stati utilizzati diversi metodi, sia teorici che puramente empirici, per ottenere una stima delle migliori costanti di controllo da applicare al PID. Di seguito sono riportati i diversi step che hanno portato ai risultati mostrati alla fine del capitolo.

### 5.1 Risultati in Isoterma

Prima di procedere allo studio delle rampe di temperatura sono state effettuate delle prove con il software per la creazione di un'isoterma, in modo da testare il buon funzionamento del PID e l'efficacia del controllo *Approaching* implementato. Dato che la generazione di un'isoterma non ha utilità ai fini degli esperimenti di desorbimento termico, ed è quindi stata realizzata prevalentemente per impraticarsi con i controlli del sistema, le costanti del PID sono state ricavate empiricamente. In Fig. 5.1 sono riportati i risultati. È possibile notare come l'utilizzo del controllo di *Approaching* consenta al sistema di stabilizzarsi in tempi molto più brevi, evitando in modo significativo il windup integrale e, di conseguenza, l'overshoot iniziale.

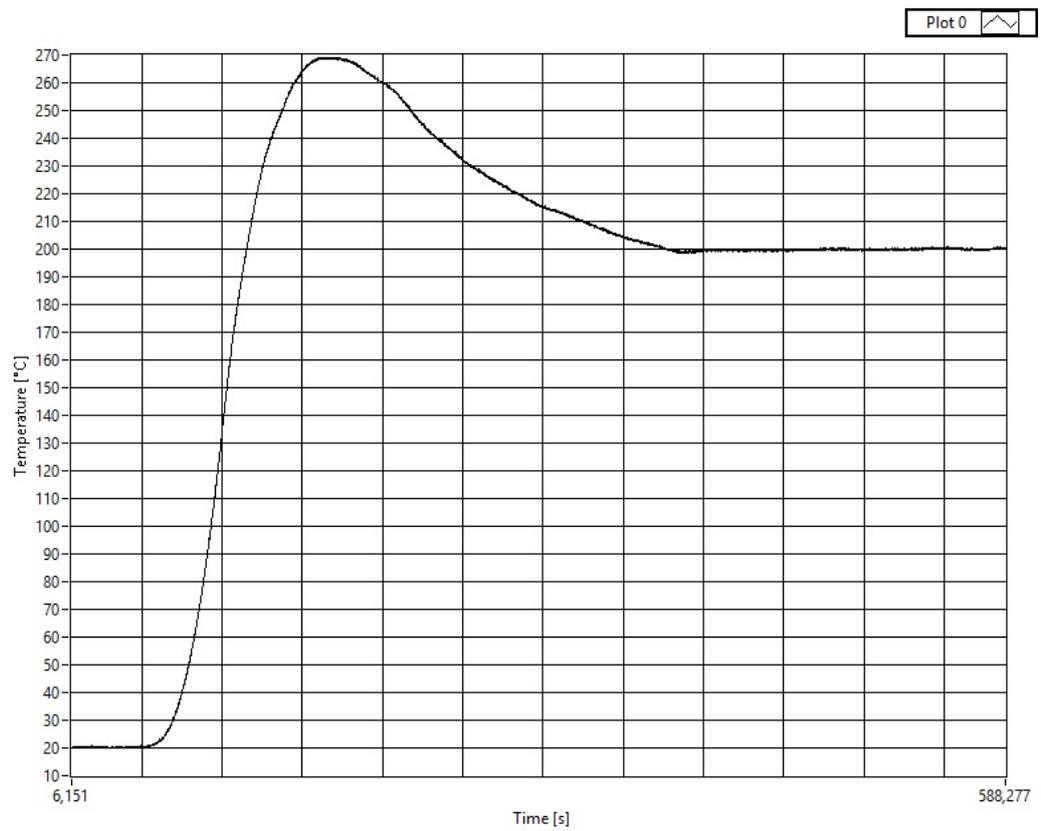


Figura 5.1: Controllo in isoterma con setpoint a  $200^{\circ}\text{C}$  e controllo *Approaching* disattivato: si nota il grande overshoot iniziale dovuto al windup della componente integrativa.

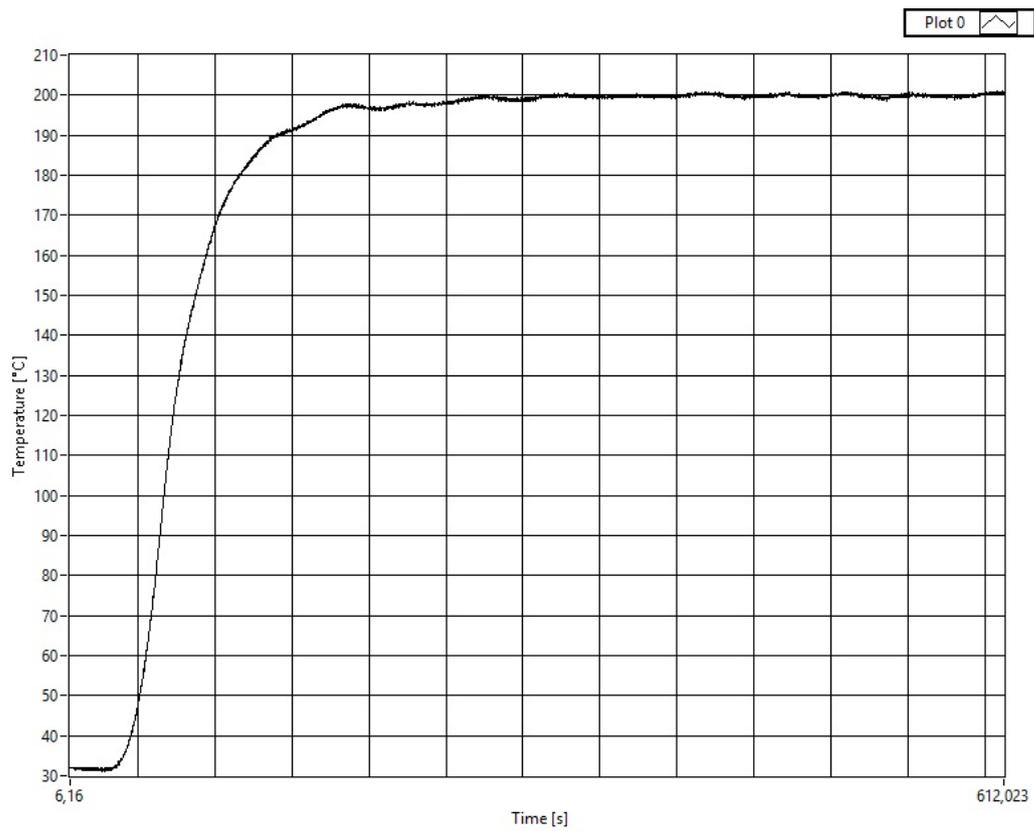


Figura 5.2: Controllo in isoterma con setpoint a  $200^{\circ}\text{C}$  e controllo *Approaching* attivato: in questo caso la componente integrativa è stata attivata intorno ai  $197^{\circ}\text{C}$  ed è evidente come l'utilizzo di questo controllo elimini totalmente l'overshoot iniziale.

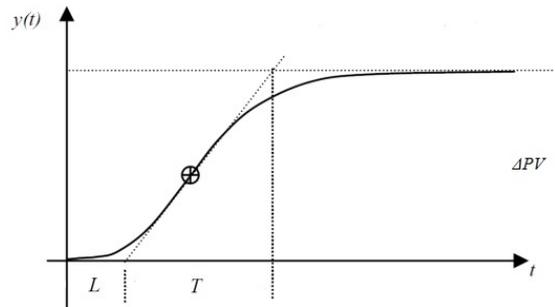


Figura 5.3: Schema per il metodo di Ziegler-Nichols: nell'origine viene applicata la variazione di input per portare il sistema dal 10% al 20% di output.  $L$  è il ritardo del sistema,  $\Delta PV$  la variazione di *Process Variabile*, ovvero la pendenza

## 5.2 Metodo di Ziegler-Nichols

Per fare una prima stima teorica delle costanti di controllo ci si è affidati all'algoritmo di Ziegler-Nichols, che tiene conto anche del *ritardo* del sistema. Il ritardo  $L$  di un sistema è definito come *il tempo che si deve attendere affinché una variazione dell'ingresso si manifesti in una corrispondente variazione dell'uscita*, ed è stato calcolato per il sistema in esame effettuando diverse misure, ed ottenendo quindi il risultato  $L \approx 10s$ . L'algoritmo di Ziegler-Nichols è da effettuarsi ad anello aperto, ovvero controllando il sistema in modalità Manuale, e comprende i seguenti step.

- Si setta l'output al 20% del massimo ottenibile: poiché l'output nel nostro caso era una pendenza sono stati svolti i seguenti passaggi. Si è settata la corrente al massimo possibile, ovvero 7,5A, e si è osservato che la pendenza ottenuta era di  $4^{\circ}C/sec$ , quindi si è calcolato che il 20% del massimo corrisponde ad una pendenza di  $0,75^{\circ}C/sec$  e, tramite un calcolo delle derivate delle curve in Fig. 3.3, si è trovato che tale pendenza è attuabile con una corrente di 4,7A.
- Si introduce una variazione di output del 10%, portando quindi l'uscita al 30%: ripetendo i calcoli svolti in precedenza si è osservato questo significava ottenere una pendenza di  $1,2^{\circ}C/sec$ , attuabile con una corrente di 5,3A.
- Una volta fatti questi passaggi si è analizzato il grafico ottenuto tramite le formule riportate in seguito:

$$K_p = \frac{\Delta PV}{\Delta Output} \quad (5.1)$$

Dove  $\Delta Output$  è la variazione in percentuale del segnale di output da 0 a 100%. Il metodo grafico è invece illustrato in Fig 5.3. Una volta calcolata questa costante, le costanti del PID sono state ottenute tramite le relazioni:

Control Type	$k_p$	$k_i$	$k_d$
P	$\frac{T}{K_p L}$	/	/
PI	$0,9 \frac{T}{K_p L}$	$3L$	/
PID	$1,2 \frac{T}{K_p L}$	$2L$	$0,5L$

Tabella 5.1: Calcolo delle costanti PID con il metodo di Ziegler-Nichols

È da notare che l'algoritmo è pensato per un sistema che possa lavorare in condizioni di isoterma, con la temperatura come setpoint: uno dei primi problemi che sono sopraggiunti è stata l'impossibilità di mantenere una pendenza fissata con una corrente costante, era infatti inevitabile che dopo un certo di tempo il sistema tendesse a stabilizzarsi ad una temperatura, e la pendenza calasse di conseguenza.

Eseguendo le operazioni sopra citate si è giunti ai seguenti risultati:

Control Type	$k_p$	$k_i$	$k_d$
P	59,238	/	/
PI	53,314	25,926	/
PID	71,086	17,284	4,231

Tabella 5.2: Costanti PID Ricavate

Purtroppo le costanti ricavate con questo metodo non si sono rivelate quelle corrette, infatti nelle prove effettuate il sistema perdeva stabilità oltre i  $60^\circ C$  e cominciava ad oscillare repentinamente. Nonostante ciò, l'utilizzo di questo algoritmo ha fornito informazioni utili: infatti dalle numerose prove effettuate, prima con la simulazione descritta in sezione 3.2 ed in seguito con il software vero e proprio, si era compreso che questo comportamento è tipicamente dovuto a costanti proporzionale e integrativa *troppo alte*, e si è quindi intuito che per stabilizzare il sistema queste andassero ridotte in modo consistente.

Il fatto che il metodo teorico abbia presentato molti problemi è principalmente dovuto a due fattori: il primo, già accennato poco sopra, è il fatto che questo algoritmo è pensato per un setpoint di temperatura, e non di pendenza. Inoltre un controllo PID prevede che l'attuatore possa lavorare sia aumentando, sia diminuendo la forza da applicare; ma nel nostro caso non sono presenti sistemi di refrigerazione ed il generatore ha unicamente la possibilità di scaldare il fornello, mentre per il raffreddamento è necessario affidarsi alla naturale conduzione. Questi problemi hanno fatto sì che il sistema si discostasse molto da quelli *ideali* trattati dal metodo teorico. Una volta tenuto conto di questi fattori, si è provveduto a calare opportunamente le costanti.

### 5.3 Risultati in Rampa

Le costanti del PID sono state modificate empiricamente grazie all'ausilio di un grafico su LabVIEW che riportava il contributo di ogni componente in tempo reale, si è quindi fatto in modo che i diversi contributi assumessero valori comparabili. Con questo metodo è stato possibile ottenere la seguente terna di costanti:

$$k_p = 8,026$$

$$k_i = 1,20028$$

$$k_d = 100$$

I risultati sono visibili nelle figure 5.4 e 5.5. Da questi grafici emerge il problema che ha portato all'implementazione di costanti adattive dipendenti dalla temperatura, descritto in sez. 5.3.1. Infatti si osserva come questa terna di costanti permetta di ottenere rampe stabili soltanto fino al raggiungimento dei  $120^{\circ}\text{C}$ .

Come ulteriore prova si è tentato di ridurre ulteriormente le costanti, ottenendo la terna:

$$k_p = 2,026$$

$$k_i = 0,30028$$

$$k_d = 25$$

I risultati sono riportati nelle figure 5.6 e 5.7. Si nota come questi valori riescano a stabilizzare in modo molto più significativo il sistema, nonostante la pendenza reale rimanga leggermente inferiore a quella richiesta: questo è probabilmente dovuto ad una componente integrativa non sufficientemente alta.

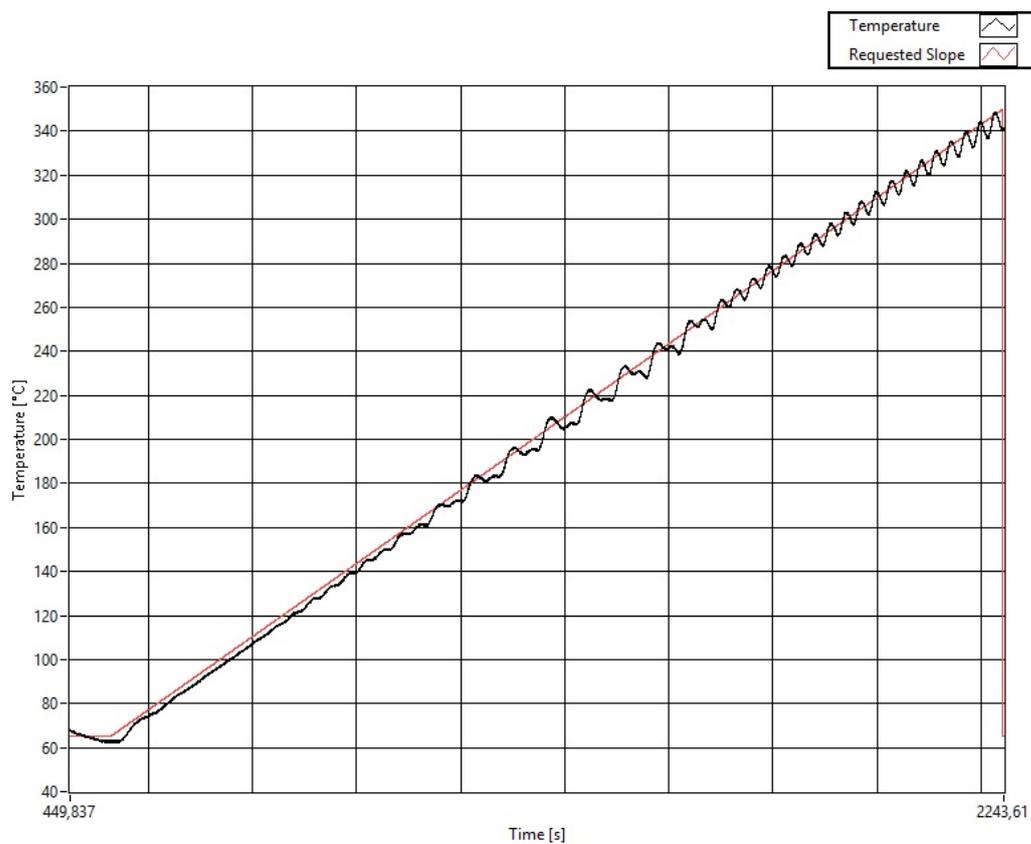


Figura 5.4: Risultati della prima terna di costanti: grafico della temperatura. L'immagine è relativa ad una pendenza richiesta di  $10^{\circ}\text{C}/\text{min}$ .

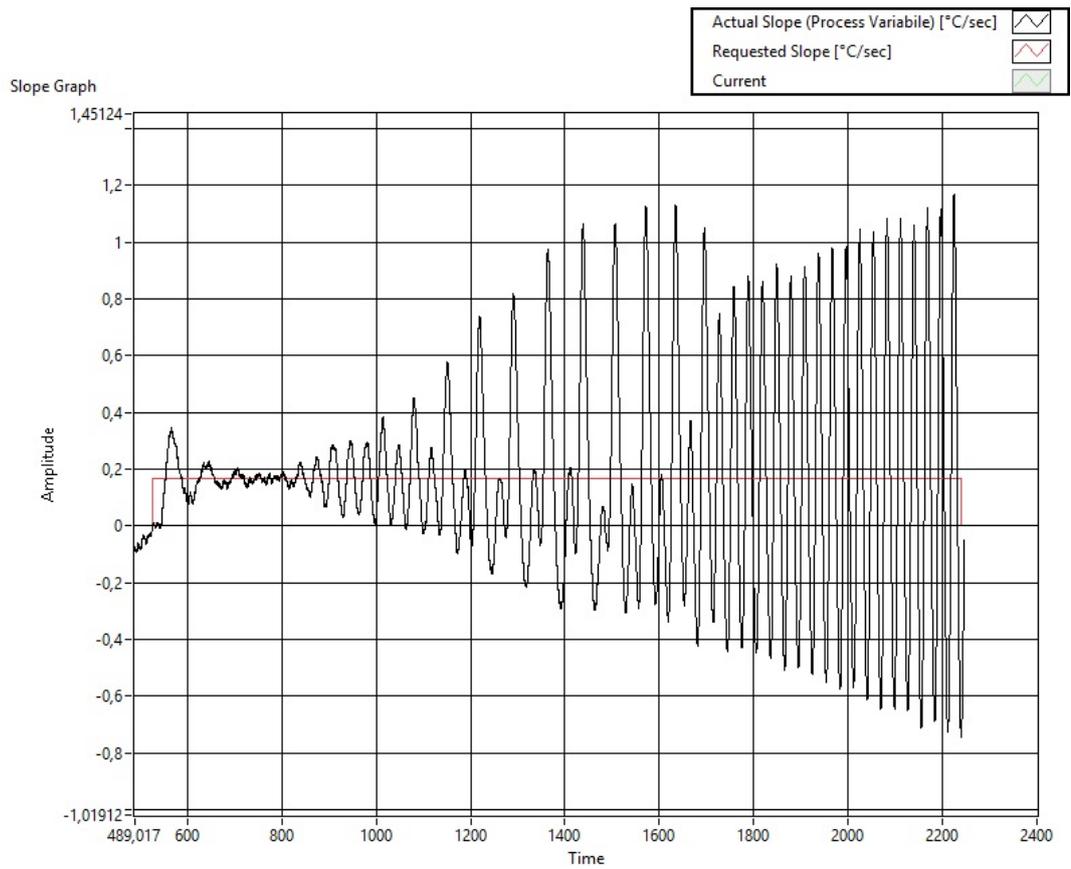


Figura 5.5: Risultati della prima terna di costanti: grafico del setpoint. Come si può vedere il setpoint è fisso, nonostante la temperatura cresca. La pendenza è espressa in  $^{\circ}C/sec$ .

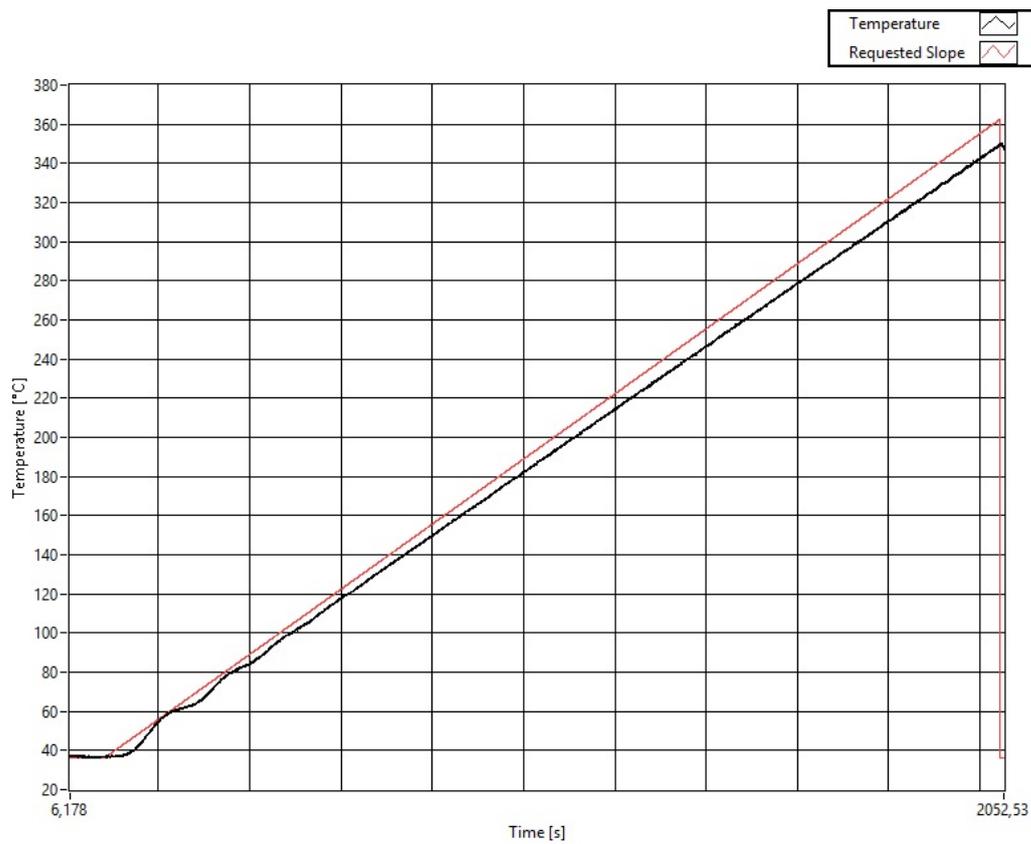


Figura 5.6: Risultati della terna di costanti ridotte: grafico della temperatura. L'immagine è relativa ad una pendenza richiesta di  $10^{\circ}\text{C}/\text{min}$ .

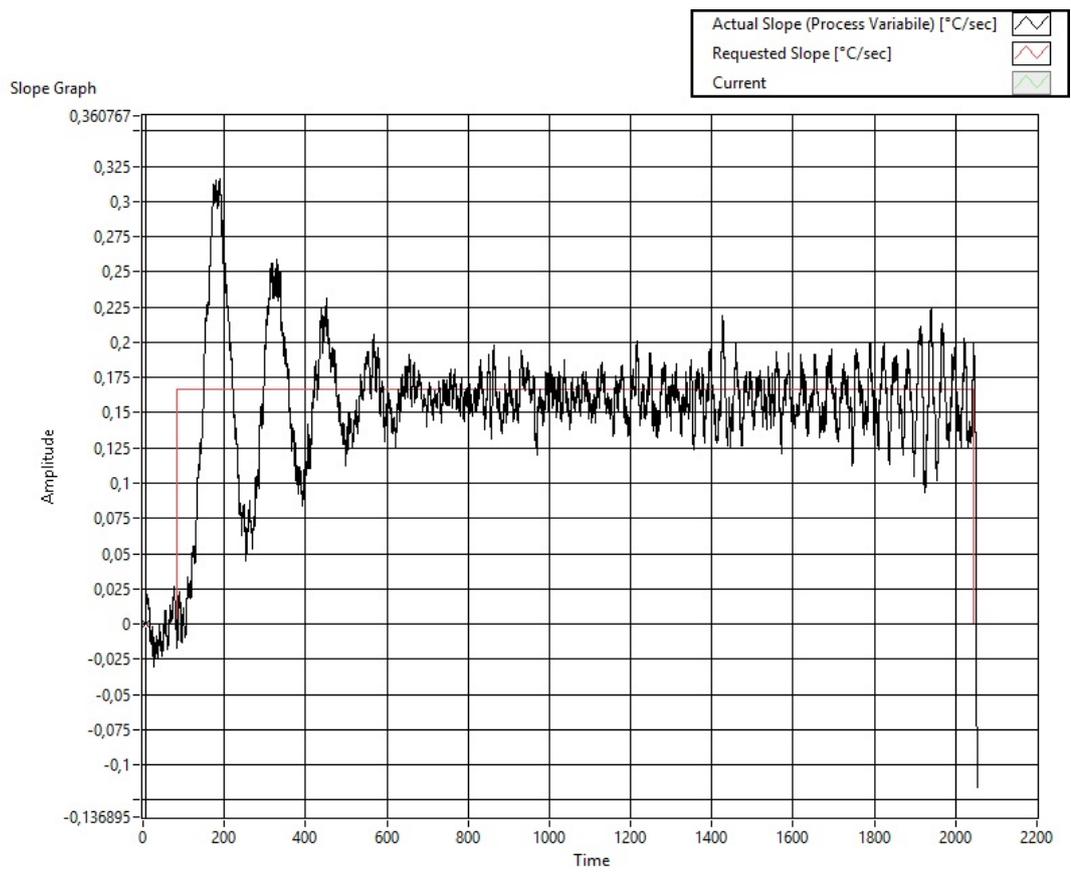


Figura 5.7: Risultati della seconda terna di costanti ridotte: grafico del setpoint. La pendenza è espressa in  $^{\circ}\text{C}/\text{sec}$ .

### 5.3.1 Costanti Adattive

Nonostante con l'ultima terna di costanti il sistema appaia più stabile, è possibile notare come verso le alte temperature si inneschi comunque un fenomeno di oscillazione dovuto a costanti troppo alte. Questo accade poiché la velocità di risposta del fornetto cresce con la temperatura, dal momento che dissipa più velocemente. Per questo motivo si è deciso di adattare automaticamente le costanti alla temperatura tramite una relazione lineare del tipo:

$$k(T) = k_0(1 + \alpha T) \quad (5.2)$$

Dove  $k_0$  è la costante a  $0^\circ C$  e  $\alpha$  il coefficiente di proporzionalità rispetto alla temperatura. Per trovare questi valori è stato svolto il seguente procedimento: sono state individuate due terne che si adattassero rispettivamente a  $25^\circ C$  e  $340^\circ C$ , quindi si sono trovate pendenza e intercetta della retta che interpolava i due punti.

In questo modo si è giunti a formulare le costanti definitive:

$$k_p(T) = 4,2837(1 - 0.001867T)$$

$$k_i(T) = 0,529995(1 - 0.001465T)$$

$$k_d(T) = 37,3656(1 - 0.001465T)$$

Di seguito sono riportati i risultati ottenuti con questi valori, sono state effettuate rampe di diversa pendenza per verificare che il sistema si adattasse a tutte le condizioni imposte dall'utente. Si è inoltre calcolato che le pendenze reali sono inferiori rispetto a quelle richieste di circa il 4%.

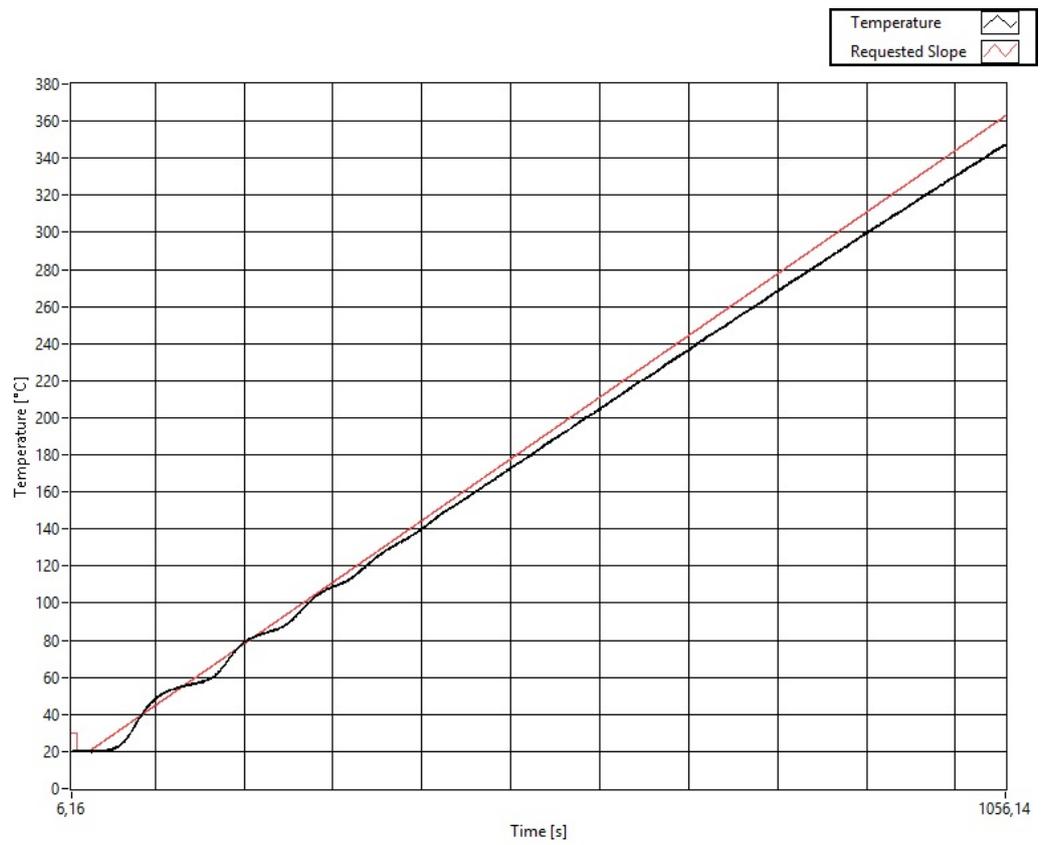
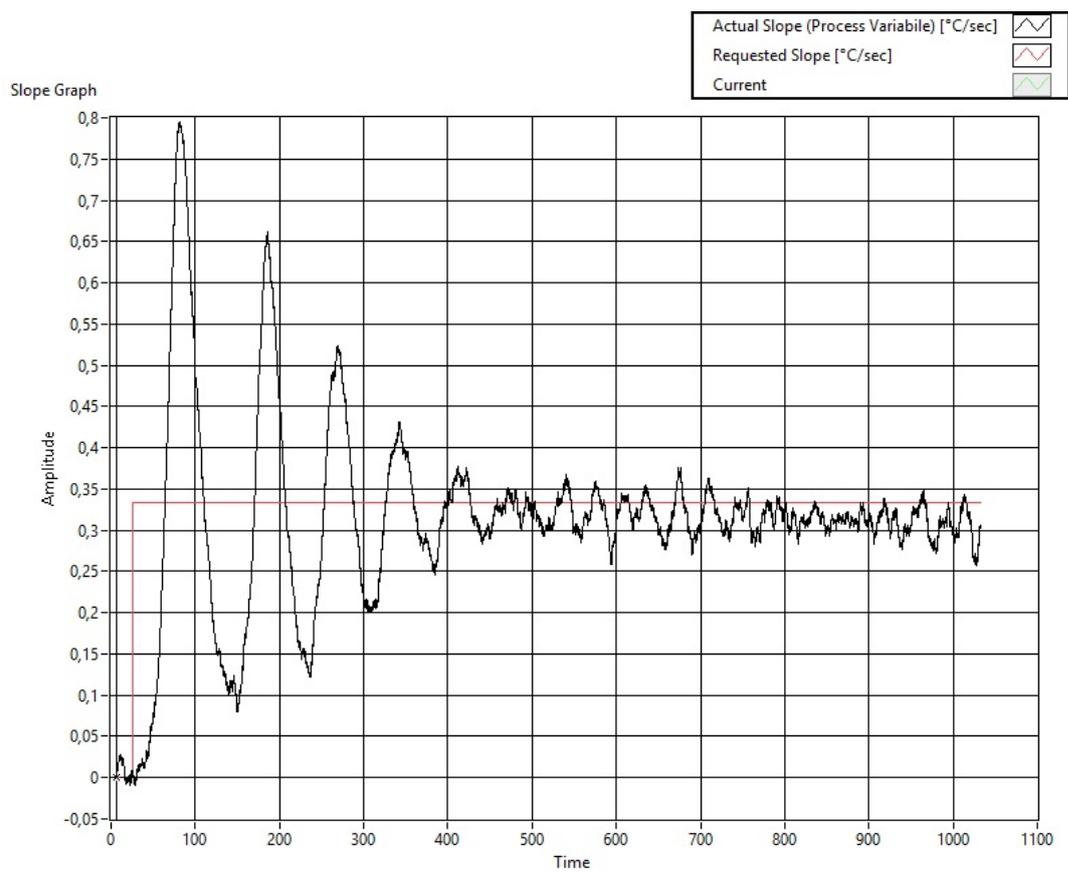


Figura 5.8: Rampa a  $20^{\circ}\text{C}/\text{min}$ . In rosso è evidenziata la pendenza richiesta.

Figura 5.9: Rampa a  $20^{\circ}\text{C}/\text{min}$ : grafico del setpoint

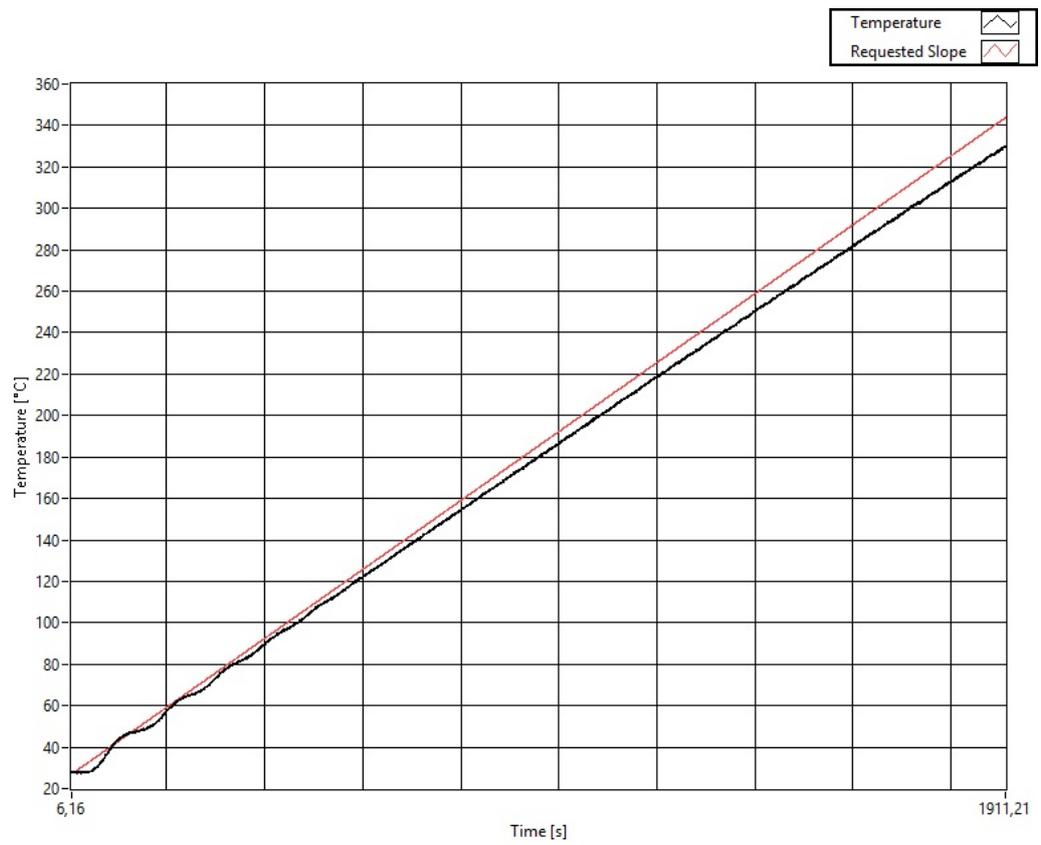
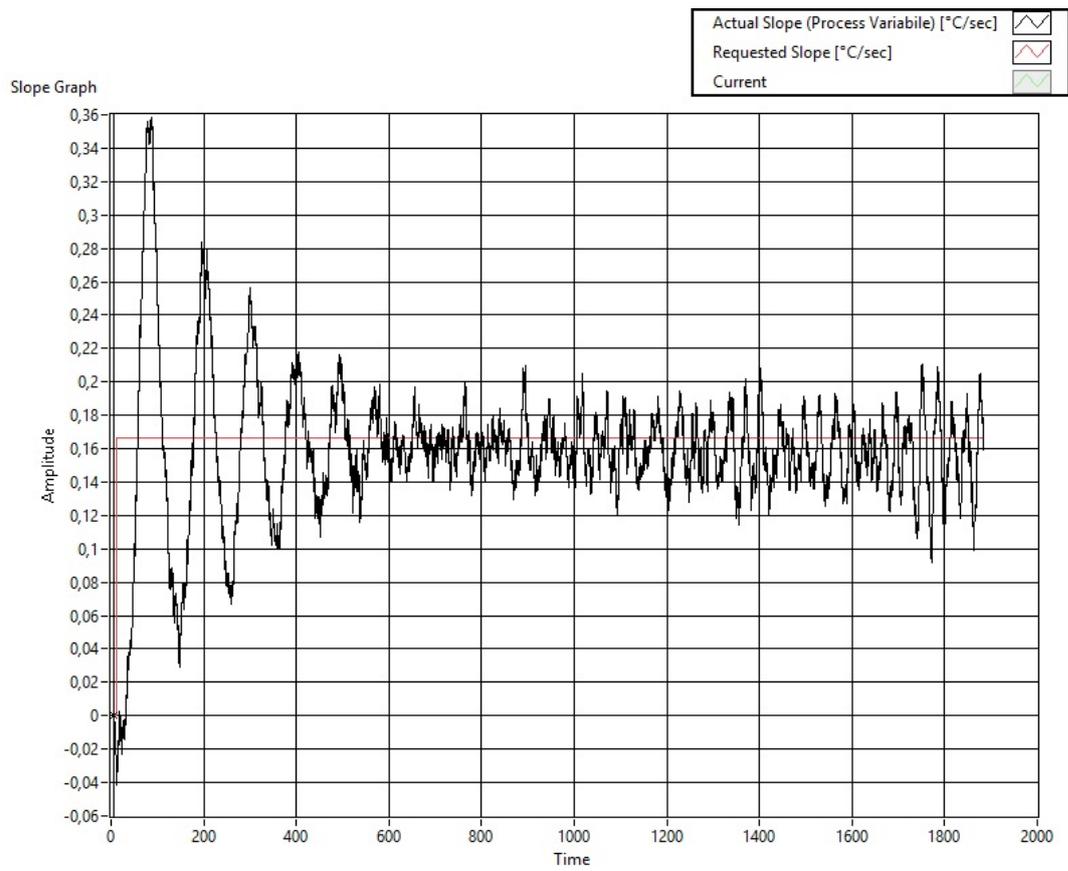


Figura 5.10: Rampa a  $10^{\circ}\text{C}/\text{min}$ . In rosso è evidenziata la pendenza richiesta.

Figura 5.11: Rampa a  $10^{\circ}\text{C}/\text{min}$ : grafico del setpoint

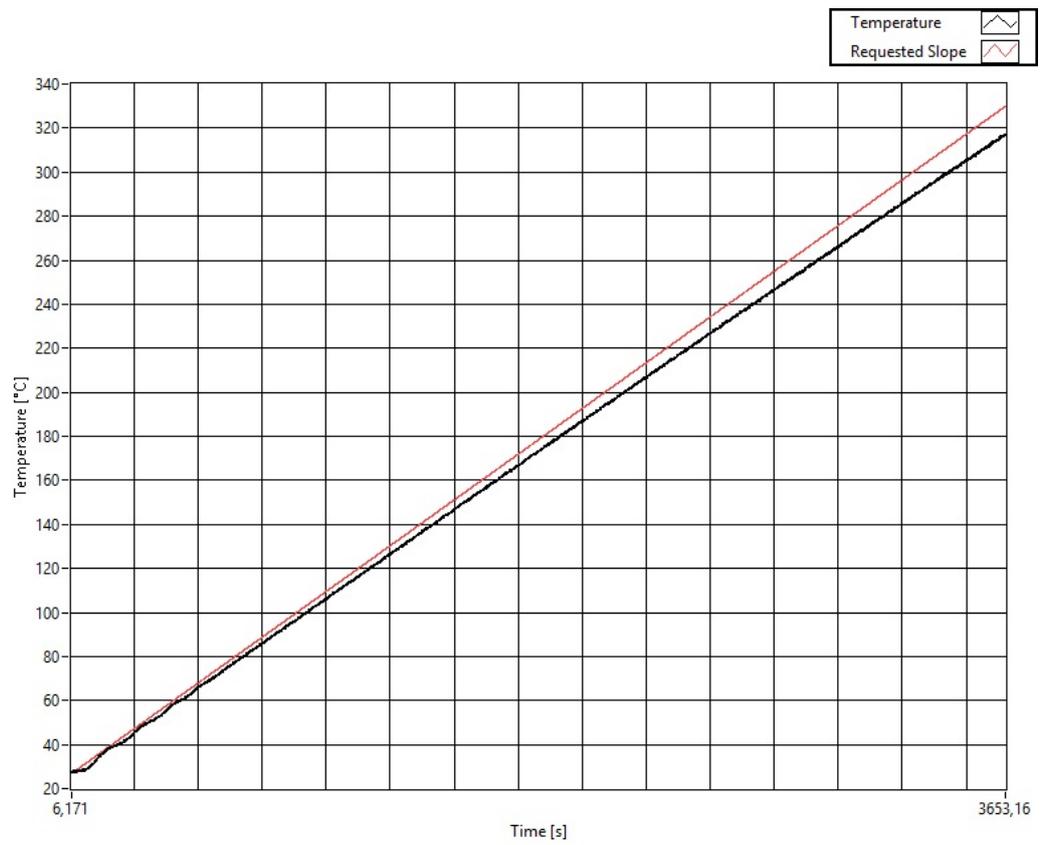
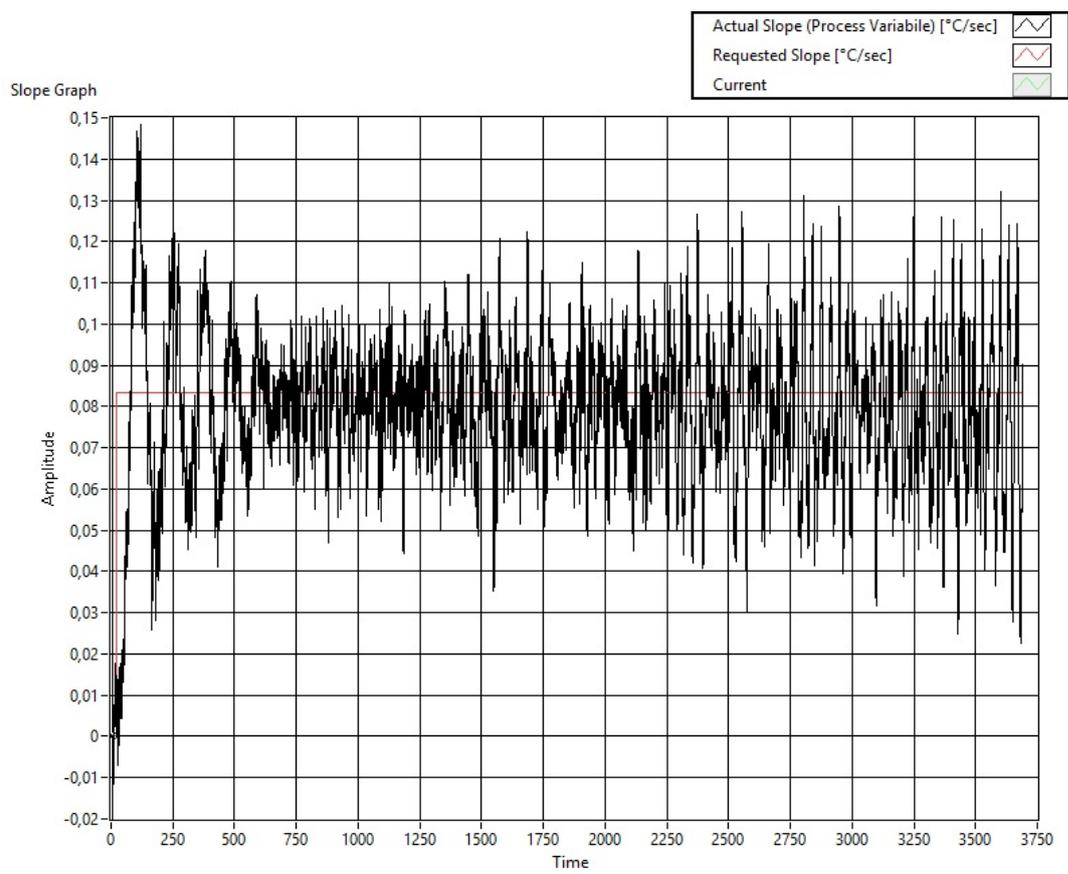


Figura 5.12: Rampa a  $5^{\circ}\text{C}/\text{min}$ . In rosso è evidenziata la pendenza richiesta.

Figura 5.13: Rampa a  $5^{\circ}\text{C}/\text{min}$ : grafico del setpoint

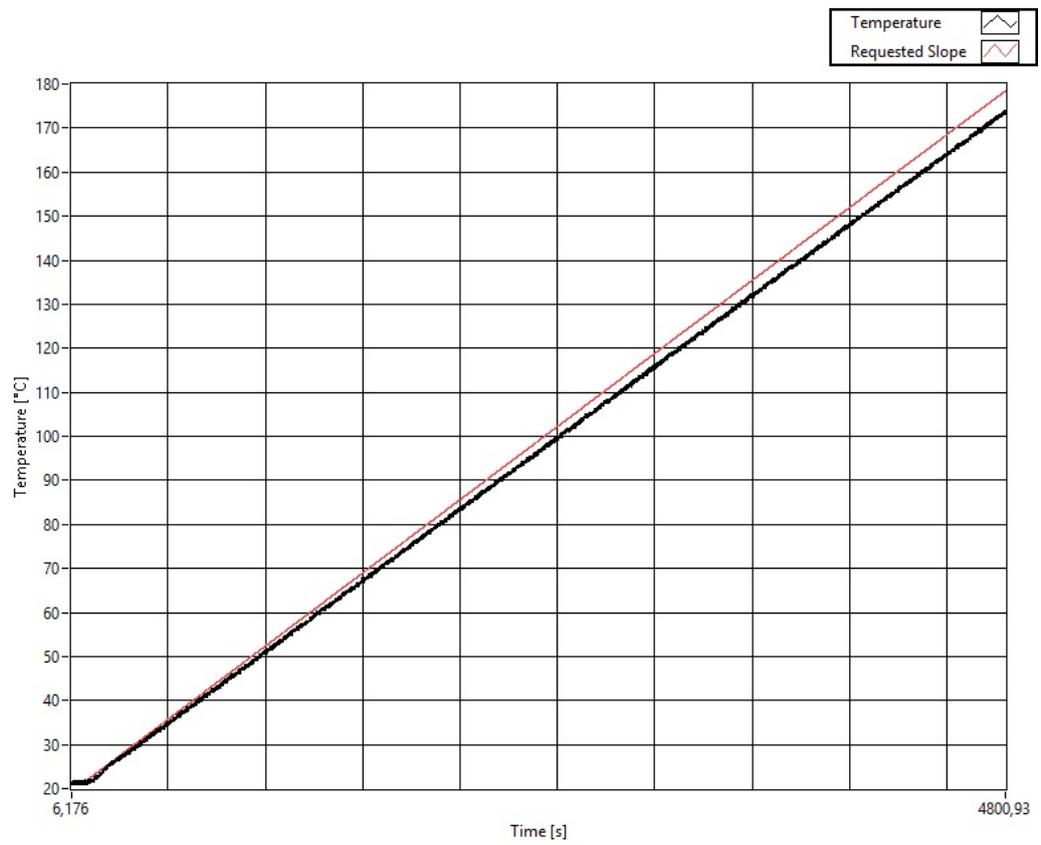
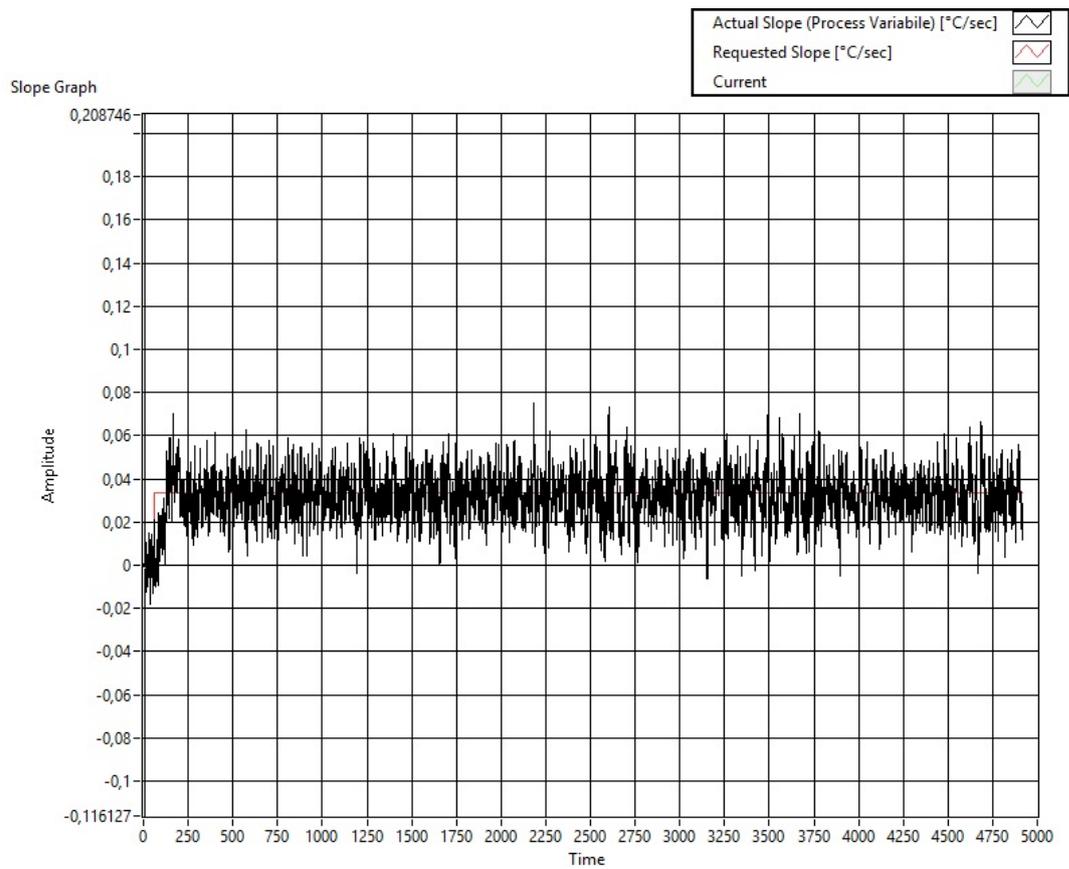


Figura 5.14: Rampa a  $2^{\circ}\text{C}/\text{min}$ . In rosso è evidenziata la pendenza richiesta.

Figura 5.15: Rampa a  $2^{\circ}\text{C}/\text{min}$ : grafico del setpoint

## 5.4 Conclusioni

Lo sviluppo del software ha presentato diverse difficoltà: in particolare si è dovuto far fronte ad un forte *ritardo* del sistema, dovuto ad un posizionamento non ottimale della termocoppia ed all'inerzia termica tipica del fornetto, e al grande rumore di fondo di cui erano affette le misure di temperatura. Quest'ultimo problema è stato parzialmente risolto tramite l'utilizzo in abbondanza di algoritmi di fit e di media mobile, ma se da una parte questo ha migliorato la sensibilità del PID, dall'altra ha introdotto ulteriore ritardo nella risposta del sistema. Per far fronte al ritardo invece, è stato necessario mantenere piccole le costanti operative del PID, per questo motivo la pendenza reale ottenuta è di qualche punto percentuale inferiore rispetto a quella richiesta. Nonostante ciò, dagli ultimi grafici si evince come il sistema si comporti in modo molto soddisfacente per qualsiasi tipo di rampa richiesta, mantenendo una pendenza costante e senza particolari oscillazioni a qualsiasi temperatura.

### 5.4.1 Possibili Sviluppi Futuri

È già stato accennato come le rampe reali differiscano lievemente da quelle teoriche: questo problema può essere risolto tramite una più attenta calibrazione delle costanti PID e della loro dipendenza dalla temperatura. Inoltre, dal momento che la buona riuscita di una rampa dipende in gran parte dalla bontà delle condizioni iniziali, potrebbe essere implementato un controllo di partenza ad anello aperto che, tramite una *Lookup Table* con corrispondenze di Pendenza/Corrente/Temperatura, porti il sistema ad una pendenza simile a quella richiesta, attivando soltanto a quel punto il controllo automatico. A livello hardware si potrebbe invece intervenire inserendo un'altra termocoppia che misuri la temperatura molto vicino al filamento: utilizzando questa per il controllo del fornetto si avrebbe una risposta molto più rapida del sistema ed un ritardo minore, la termocoppia all'interno del fornetto rimarrebbe solamente per controllare la temperatura del campione da analizzare.

# Appendice A

## II PID

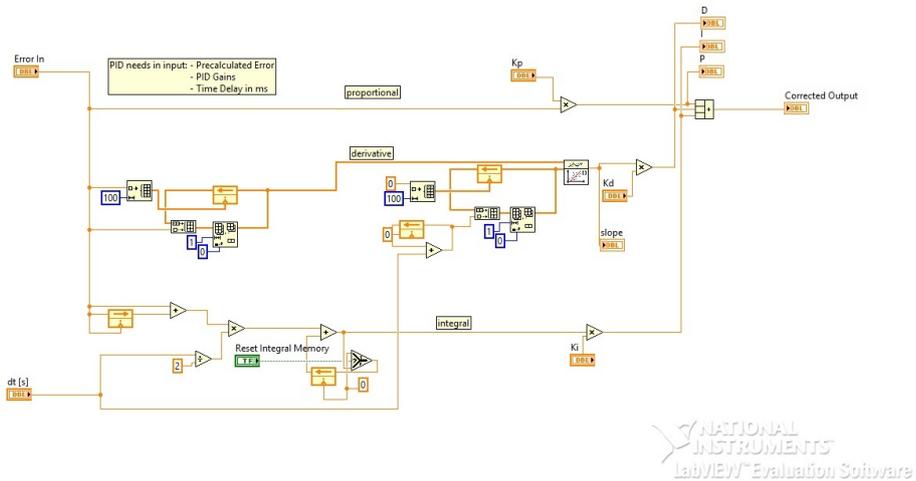


Figura A.1: Pannello posteriore della sub-VI PID



# Appendice B

## Il Software



## B.2 Rampa

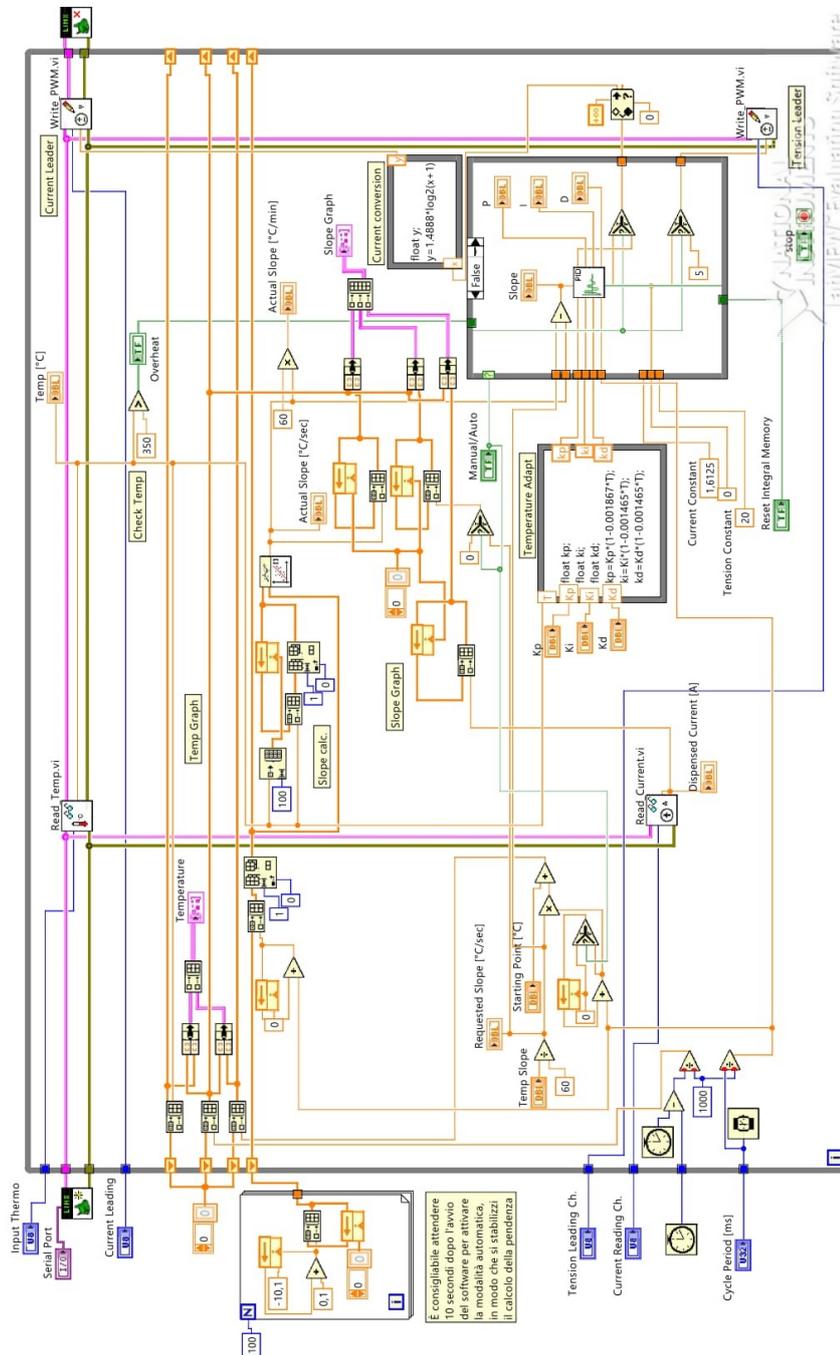


Figura B.2: Pannello posteriore della VI Rampa



# Bibliografia

1. Sviluppo del Sistema di Controllo della Temperatura per Spettroscopia di Desorbimento Termico. *Tommaso Neri, tesi di laurea in Fisica, Università di Bologna. 2014.*
2. Sistema di Controllo PID di Temperatura con LabVIEW e Scheda Arduino. *Matteo Marchesini, tesi di laurea in Fisica, Università di Bologna. 2016*
3. Analysis & Design of Feedback Control Systems. George J. Thaler e Robert G. Brown. McGraw-Hill, 1960.
4. Design of Feedback Systems. George J. Thaler Dowden, Hutchinson & Ross Inc. , 1973.
5. PID Theory Explained. National Instruments.
6. Closed-Loop Temperature Control of an Electric Heater. Deniz Yildirim, Yeditepe University, 2011.
7. Antitrasformate di Laplace, Modi di un Sistema. Luigi Biagiotti, Università di Bologna.
8. Sistemi Dinamici Lineari. Roberto Zanasi, Università di Modena e Reggio Emilia.