



Università Politecnica della Marche  
Facoltà di Ingegneria

Corso di laurea triennale in Ingegneria Meccanica

**“Progettazione e sviluppo di un banco per  
la misura e controllo della temperatura”**

**“Design and development of a bench for  
temperature measurement and control”**

Relatore:  
Prof. Gian Marco Revel

Tesi di Laurea di:  
Fernando Alonso Pena Almidon

Correlatori:  
Dot.ssa Sara Casaccia

Dott. Riccardo Naccarelli

A.A. 2021/2022



Dedicata alla mia mamma e al mio papà,  
Con voi al mio fianco, tutto è possibile.  
E un grazie speciale a mia sorella,  
Il mio più grande orgoglio



# Indice

<b>Introduzione .....</b>	<b>1</b>
---------------------------	----------

<b>1. Sistema di controllo PID.....</b>	<b>3</b>
---	----------

1.1 Azione proporzionale .....	4
1.2 Azione integrale.....	7
1.3 Il fenomeno del wind-up .....	9
1.4 Azione derivativa .....	12
1.5 Metodi di regolazione .....	14
1.5.1 Metodo della risposta al gradino di Ziegler-Nichols .....	15
1.5.2 Metodo della risposta in frequenza Ziegler-Nichols .....	17
1.5.3 Metodo Relay (Åström-Hägglund) .....	19

<b>2. Introduzione del progetto .....</b>	<b>22</b>
---	-----------

2.1 Arduino Mega 2560 .....	23
2.2 Termocoppia – Modulo MAX6675 .....	24
2.3 Modulo relay .....	25
2.4 Display LCD 16X2 con retroilluminazione .....	27
2.5 Lampada alogena (220 W) .....	29
2.6 Primo Setup .....	30
2.7 Codice Arduino per il controllo del primo setup .....	30
2.8 Modulo Dimmer .....	32

2.9	Secondo Setup .....	33
2.10	Codice Arduino per il controllo del secondo setup.....	34

### **3. Costruzione del Banco di Misura e Controllo.....36**

3.1	Fasi costruttive del banco .....	37
3.2	Manuale d'uso .....	39

### **4. Risultati Sperimentali ..... 41**

4.1	Sistema di controllo con modulo relay .....	41
4.2	Sistema di controllo con modulo dimmer .....	43
4.2.1	Metodo Ziegler-Nichols ad anello chiuso .....	43
4.2.2	Sistema di controllo P .....	47
4.2.3	Sistema di controllo PI .....	48
4.2.4	Sistema di controllo PID .....	49

### **5. Conclusioni ..... 51**

### **Bibliografia ..... 52**

# Introduzione

L'argomento discusso in questo elaborato riguarda i sistemi di controllo, cioè dispositivi che consentono di regolare l'andamento temporale di una o più grandezze in uscita di un sistema controllato.

Tale sistema stabilisce una relazione tra una variabile di ingresso (detta di riferimento) e una di uscita, che costituisce la grandezza controllata. Le funzionalità devono essere quelle di controllo a ciclo chiuso del sistema, calcolo della variabile di set-point, analisi di eventuali anomalie ed infine una correzione, nel caso in cui siano presenti quest'ultime, che riporti ad avere il valore desiderato.

Esistono diverse tipologie di sistemi di controllo, più o meno soddisfacenti in base al tipo di sistema che si vuole controllare. Qui ci occuperemo di una tipologia che negli ultimi anni ha avuto una grande diffusione, ovvero i sistemi di controllo PID.

Tra le ragioni del vasto utilizzo dei regolatori PID, i quali vengono utilizzati soprattutto in ambito industriale, ci sono:

- Semplicità di taratura dei parametri
- Efficacia per la regolazione di un'ampia gamma di processi industriali
- Data la loro semplicità, possono essere realizzati con molteplici tecnologie
- Ridotti costi di progetto e realizzazione

L'obiettivo di questa tesi è quello di progettare e realizzare un banco per la misura ed il controllo della temperatura a scopo didattico. Per

fare ciò sono stati sviluppati due setup: il primo ha visto l'utilizzo di un modulo relay, mentre il secondo di un sistema di controllo Proporzionale-Integrativo-Derivativo (PID) e di un modulo dimmer. La variabile controllata è la temperatura misurata all'interno di un box riscaldato tramite lampada alogena.

Tale sistema deve essere in grado di mantenere la temperatura desiderata e nel caso di anomalie di tornare nel minor tempo possibile al valore impostato.



# Capitolo 1

## 1. Sistema di controllo PID

Il sistema di controllo PID, il cui nome è un acronimo che sta per Proporzionale Integrativo Derivativo (Figura 1.1) suggerisce quali siano le componenti dello stesso:

- La prima componente è proporzionale all'errore  $e(t)$  tra il riferimento  $r(t)$  e l'uscita da controllare  $y(t)$
- La seconda componente ha in uscita un segnale proporzionale all'integrale dell'errore in ingresso al sistema  $e(t)$
- La terza componente ha in uscita un segnale proporzionale alla derivata dell'errore in ingresso al sistema  $e(t)$

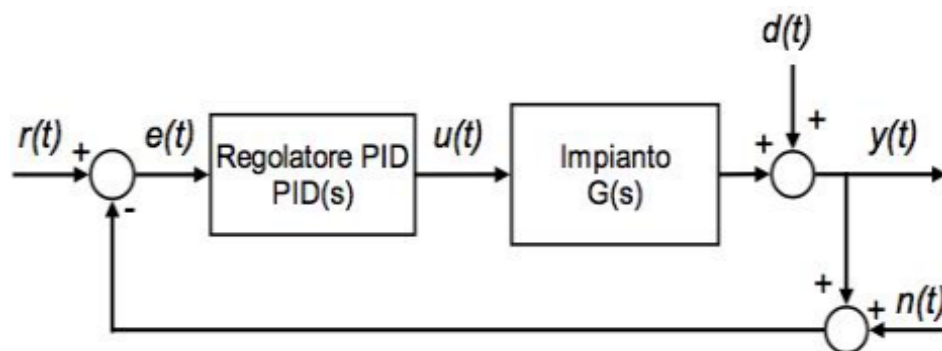


Figura 1.1: Schema generale di un controllore PID

Nella figura 1.1 possiamo notare come il regolatore PID venga posto in serie al sistema da controllare (ovviamente prima di esso perché il controllo abbia effetto sulla grandezza in ingresso al sistema  $u(t)$ ) e come il circuito di retroazione riporti in ingresso il termine  $y(t)$ , il

quale, attraverso un nodo sommatore (che in questo caso assume il ruolo di sottrattore), va a modificare il segnale vero e proprio d'ingresso del controllore.

Detto ciò, la legge di controllo del sistema di controllo PID può essere scritta come:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

dove:

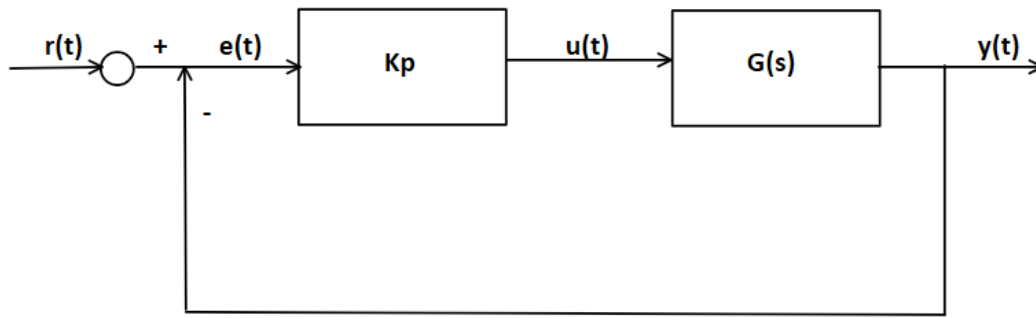
- $K_p$ : costante proporzionale
- $K_I$ : costante integrativa
- $K_D$ : costante derivativa

## 1.1 Azione proporzionale

Nell'azione proporzionale entra in gioco il coefficiente  $K_p$ , detto anche coefficiente dell'azione proporzionale, il quale lega algebricamente l'ingresso  $e(t)$  e l'uscita  $u(t)$ .

$$u(t) = K_p \cdot e(t)$$

Lo schema da prendere in considerazione (Figura 1.2) è il seguente:



*Figura 1.2: Schema controllore proporzionale*

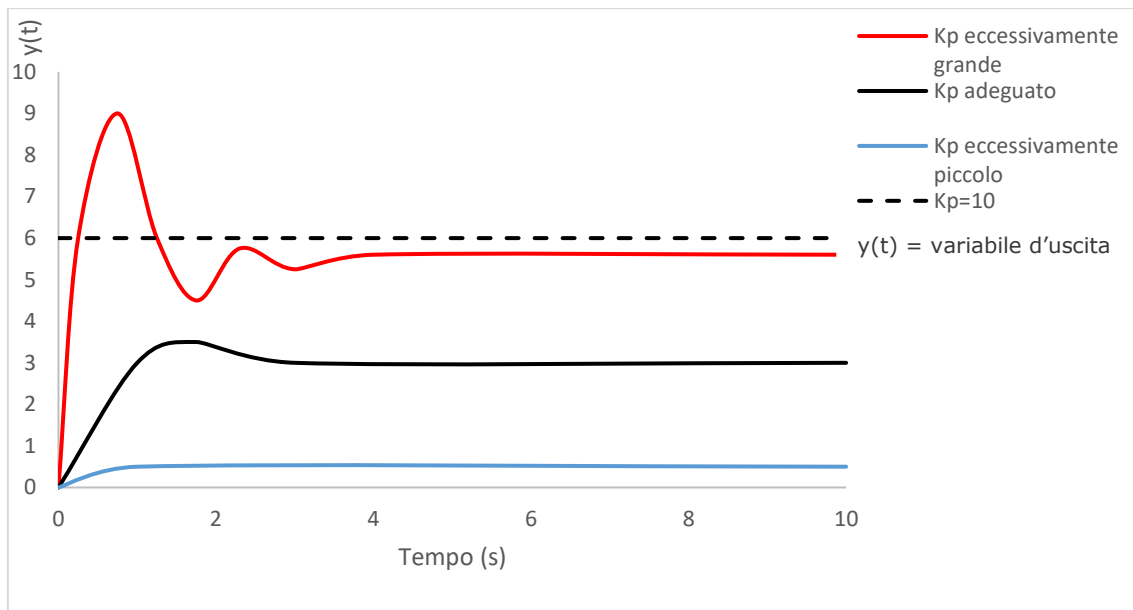
Nella realtà industriale, tuttavia, invece che il parametro  $K_p$  si prende in considerazione un'altra grandezza chiamata Banda Proporzionale e indicata con il simbolo  $B_p$ . Quest'ultima è definita come la minima variazione dell'ingresso  $e(t)$  (espressa in percentuale) che porta l'uscita  $u(t)$  dal valore minimo al suo valore di fondo scala.

$$K_p = \frac{100}{B_p}$$

Da ciò si deduce che riducendo la banda proporzionale si ottiene un maggiore guadagno e una maggiore precisione dello strumento; inoltre, ad un aumento del guadagno corrisponde anche un aumento del valore raggiunto all'uscita. Tale valore si discosterà sempre di meno dal valore originale del segnale senza però raggiungerlo a causa dell'errore a regime.

All'aumento del guadagno c'è però un limite dovuto al fatto che oltre un certo valore il sistema entra in uno stato di oscillazione, meglio detto stato di instabilità.

Tutto questo si può vedere meglio nella figura 1.3 dove è evidente lo stato di instabilità con un  $K_p$  eccessivamente piccolo.



*Figura 1.3: Grafico al variare della banda proporzionale*

In conclusione, possiamo dire:

- Per avere un errore a regime molto piccolo bisogna avere un  $K_p$  elevato, tuttavia, questo aumento può compromettere la stabilità del sistema rendendolo molto suscettibile ai rumori di sistema cioè gli errori che avvengono sull'uscita  $y(t)$ .
- Siccome l'azione proporzionale non garantisce l'annullamento dell'errore, si può pensare di annullare l'errore sommando alla variabile di controllo  $u(t)$  un valore appropriato  $U$  detto parametro di reset.

Il blocco proporzionale non introduce ritardo di fase ed aumenta la velocità di risposta del sistema rendendolo così capace di seguire più velocemente il segnale di riferimento (setpoint). Nel contempo però diminuiscono i margini di stabilità del sistema usato per processi asintoticamente o semplicemente stabili quando le prestazioni statiche non richiedano elevati guadagni e l'uso di un azione integrale.

## 1.2 Azione integrale

La seconda azione da prendere in considerazione in un controllore PID è l'azione integrale. Come detto prima quest'ultima è proporzionale all'integrale dell'errore  $e(t)$  e il coefficiente da prendere in considerazione  $K_i$  definisce la costante di tempo integrale  $T_i$  chiamata tempo di reset.

$$u(t) = K_I \cdot \int_0^t e(\tau) d\tau$$

Quest'ultima espressione deriva da quella che è la formula generale di un controllore PID, assumendo  $K_d=0$  e  $K_p=0$ .

Per quanto riguarda il tempo integrale  $T_i$ , non è altro che:

$$T_i = \frac{K_p}{K_i}$$

Inoltre, lo schema da prendere in considerazione (figura 1.4) è il seguente:

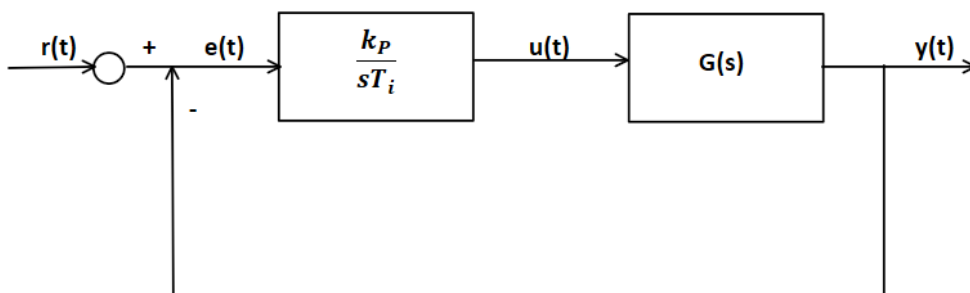


Figura 1.4: Schema controllore integrale

Prima di tutto, è importante ricordare che l'azione integrale assicura un errore nullo a regime per variazioni a gradino del riferimento  $r(t)$ .

La realizzazione dei cosiddetti controllori PI, ovvero la combinazione di azione integrale e proporzionale, permette di ottenere una maggiore precisione senza peggiorare il grado di stabilità del sistema e una maggiore velocità di risposta.

I controllori PI vengono utilizzati soprattutto nei sistemi in cui le variazioni di carico avvengono lentamente.

Per il blocco in considerazione, la variabile da tarare in base al sistema con cui si ha a che fare è la costante di tempo integrale  $T_i$ ; da essa infatti dipende l'effetto di integrazione che è tanto più importante quanto più  $T_i$  è piccola. A questo riguardo è necessario notare come riducendo il tempo di reset, l'integrale dell'errore salirà più velocemente verso il valore  $r(t)$  in ingresso al blocco pagando però questa velocità di salita con delle forti oscillazioni che richiedono tempo prima di stabilizzarsi. Viceversa, aumentando  $T_i$  e facendolo tendere ad infinito, si ha come conseguenza l'eliminazione dell'effetto dell'integrale come si vede bene nella figura 1.5.

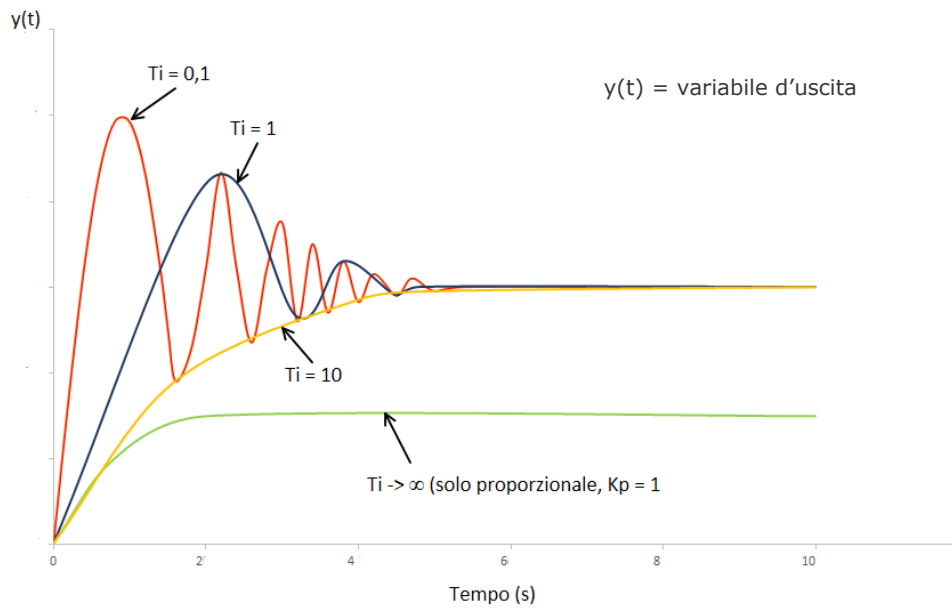


Figura 1.5: Grafico effetto  $T_i$

Questo fenomeno fa sì che il sistema dopo aver raggiunto la condizione di errore nullo, si allontani in direzione opposta, creando un effetto di sovra elongazione dalle caratteristiche non lineari.

Una situazione del genere è indesiderata e richiede un certo periodo di tempo prima che la situazione normale si ristabilisca, ma fortunatamente ci sono vari modi per poter ovviare a questo problema.

### 1.3 Il fenomeno del wind-up

La maggior parte dei sistemi di controllo viene progettata utilizzando la teoria lineare ovvero notevole semplicità e buona prestazione quando i comportamenti sono ben descritti da sistemi lineari.

Spesso però è necessario considerare anche effetti non lineari, il principale dei quali è rappresentato dalla saturazione degli attuatori.

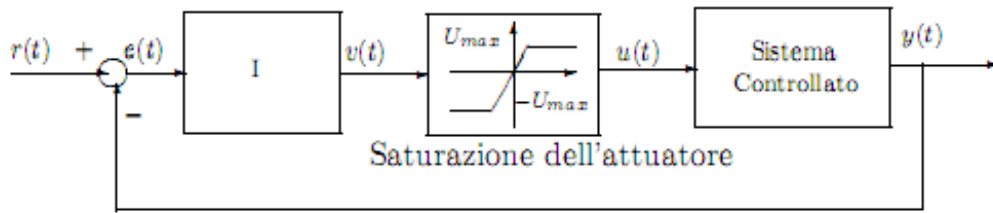


Figura 1.6: Sistema di controllo con saturazione dell'attuatore

Il fenomeno del wind-up si verifica nei sistemi di controllo in cui il controllore incorpora una azione integrale, e nel contempo si include nella analisi, come detto prima, il fenomeno della saturazione dell'attuatore, cioè il fatto che nei sistemi di controllo quest'ultimi hanno dei vincoli sull'ampiezza delle uscite, che non possono superare dei valori massimi e minimi.

In presenza di saturazione, come detto in precedenza, può verificarsi che l'uscita dell'attuatore non cresca, pur rimanendo l'errore di regolazione  $e(t)$  non nullo. Conseguentemente, il termine integrale continua a crescere, ma tale incremento non produce alcun effetto sulla variabile di comando dell'impianto.

L'uscita impiega molto tempo per portarsi su valori a regime. Il fenomeno è dovuto proprio al wind-up dell'integratore contenuto nel PID, che continua a integrare l'errore anche quando l'ingresso sta saturando.

Il sistema di regolazione può riattivarsi solo allorquando il segnale  $u(t)$  rientra nella zona di linearità della caratteristica dell'attuatore (scarica del termine integrale). Questo fenomeno si chiama comunemente integral wind-up.

L'utilizzo dell'azione integrale nei sistemi di controllo PID è diventato molto ampio e di conseguenza è cruciale importanza saper risolvere questo problema del wind-up. Sono infatti state sviluppate diverse

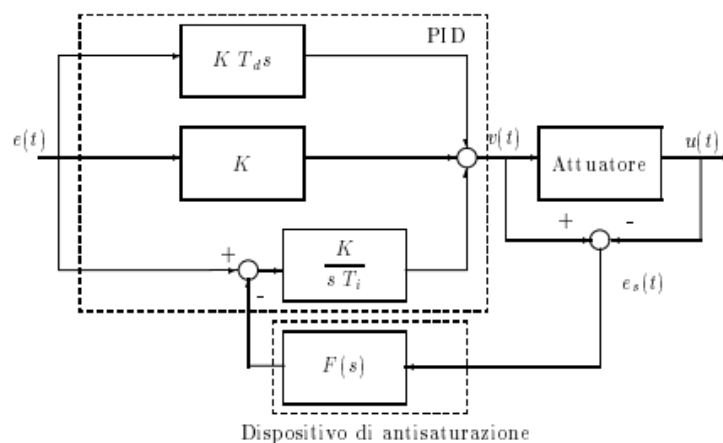


configurazioni volte ad attenuare gli effetti dannosi sulle prestazioni dei sistemi di controllo PI e PID.

Una prima configurazione è quella del ricalcolo del termine integrale, il quale introduce un termine di compensazione all'ingresso del termine integrale, con segno opposto a quello dell'errore  $e(t)$ .

$$e_s(t) = v(t) - u(t).$$

Il segnale di compensazione è nullo fino a che l'uscita del PID non è satura, mentre fornisce un'azione tendente a diminuire il termine integrale nel caso in cui l'uscita del PID si saturi



*Figura 1.7: Schema di antisaturazione mediante ricalcolo del termine integrale*

In figura 1.7 invece viene rappresentata l'integrazione condizionata dove il termine integrale viene azzerato nel caso in cui il segnale di compensazione sia nullo. Molto simile allo schema del ricalcolo del termine integrale, ma si discosta da esso per la sostituzione del dispositivo antisaturazione con un comparatore.

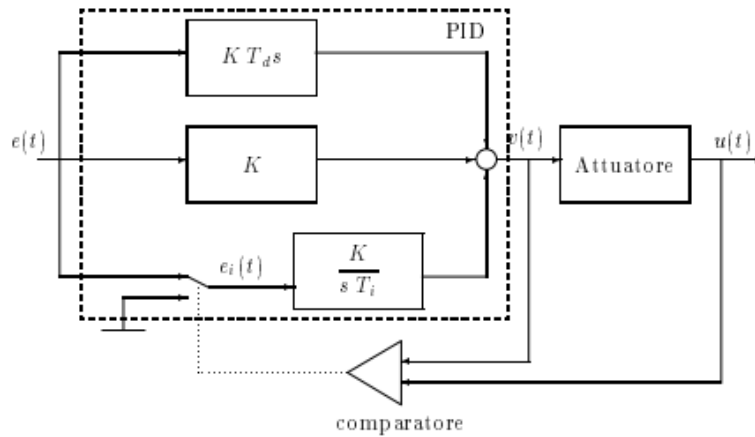


Figura 1.8: Schema di antisaturazione mediante integrazione condizionata

## 1.4 Azione derivativa

L'ultima caratteristica da studiare in un regolatore PID è l'azione derivativa. In quest'ultima entra in gioco il coefficiente dell'azione derivativa  $K_d$  che definisce la costante di tempo derivativa  $T_d$ :

$$u(t) = K_d \frac{de(t)}{dt} \qquad T_d = \frac{K_d}{K_p}$$

Le considerazioni che si andranno a fare saranno per un controllore derivativo e quindi con  $K_p=0$  e  $K_i=0$ .

Lo schema da prendere in considerazione (figura 1.9) è il seguente:

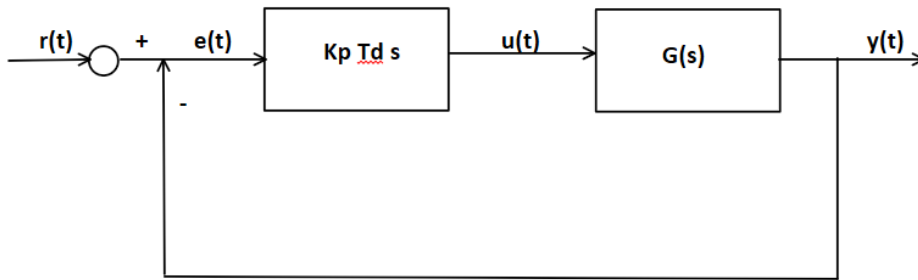


Figura 1.9: Schema controllore derivativo

Un controllore derivativo viene anche detto anticipatore o controllore di velocità. L'uscita di un controllore derivativo dipende dalla velocità con cui varia l'errore. Molto importante nell'azione derivativa è che fornisce un anticipo di fase di  $90^\circ$ .

Per comprendere questo fenomeno si prende in considerazione un sistema in cui vi è un forte ritardo di fase. A seguito di una variazione della variabile di errore si otterrà una variazione della variabile controllata con un certo ritardo. Quello che si ottiene è un ritardo del controllore rispetto alle variazioni dell'errore.

In figura 1.10 è presentato il grafico dell'esempio appena analizzato, dal quale si può apprezzare l'azione anticipatrice del blocco derivatore e la sua correzione sulle oscillazioni attorno al valore desiderato.

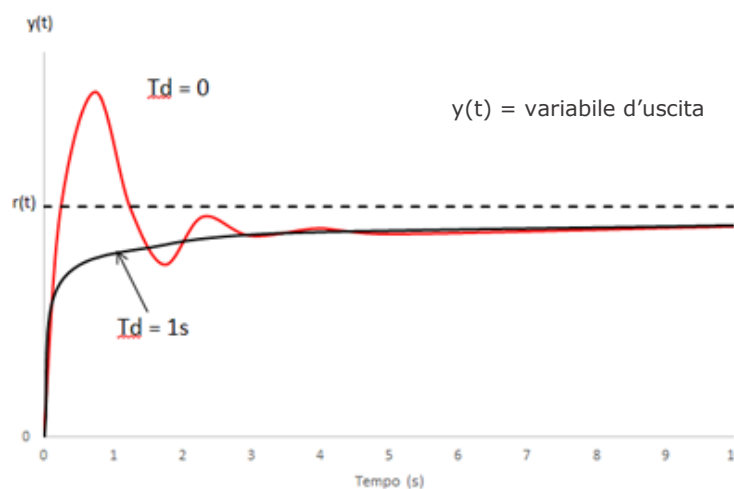


Figura 1.10: Grafico al variare di  $T_d$

Detto ciò, è molto importante ricordare anche che il termine derivativo introduce l'inconveniente di amplificare i segnali con contenuto armonico a frequenze elevate. Siccome la maggior parte delle volte questo tipo di segnale corrisponde a un rumore elettromagnetico molto fastidioso l'utilizzo del termine derivativo deve essere valutato con attenzione.

Per ovviare a questa problematica, che potrebbe condizionare l'uso del controllore, si associa un filtro passa – basso.

## **1.5 Metodi di regolazione**

Una volta che sono stati introdotti tutti i componenti presenti in un controllore PID, e quindi tutti quei parametri da modificare per regolarne il comportamento, occorrono delle tecniche per poter ottenere controllori PID il più performanti possibili.

La difficoltà nel realizzare un controllo PID è appunto trovare il miglior compromesso tra i tre parametri studiati precedentemente ovvero  $K_p$ ,  $K_i$ ,  $K_d$ .

Questo può essere fatto in modo diverso, partendo dai classici metodi manuali fino ai metodi automatici (Ziegler-Nichols, Relay Method).

Il vantaggio delle metodologie di tuning automatiche è che estraggono i parametri del PID. La presenza di metodi di taratura automatica hanno reso i PID i controllori in assoluto più utilizzati.

### 1.5.1 Metodo della risposta al gradino di Ziegler-Nichols

Si tratta di un test di risposta a gradino ad anello aperto il quale deve essere stabile. In generale la risposta al gradino è caratterizzata da due parametri:

- la costante di tempo  $T$
- il tempo di ritardo  $L$

Il metodo viene applicato solitamente a quei processi che hanno una risposta al gradino monotona, che è tipica dei processi termici.

L'obiettivo in questo caso è quello di identificare un modello del sistema e successivamente assegnare i valori al controllore. Il modello del sistema è il seguente:

$$G(s) = \frac{K}{1 + \tau s} e^{-Ts}$$

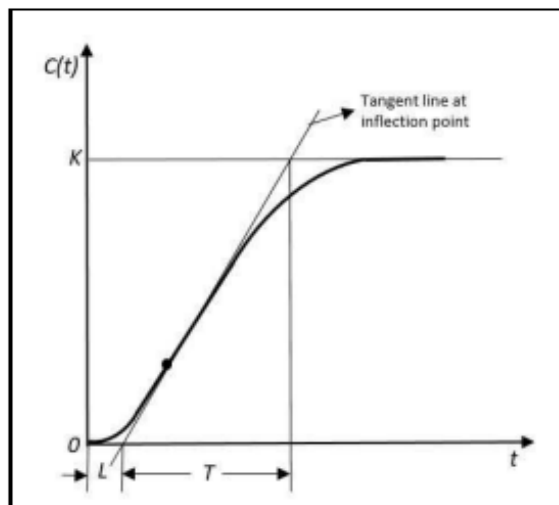


Figura 1.11: Grafico risposta a gradino

Prendendo in considerazione la figura 1.11 ovvero il grafico della risposta a gradino, si traccia una linea tangente nel punto a massima pendenza della risposta.

Fatto ciò, si identificano due parametri:

- L: intersezione tra la retta tangente e zero
- T: intersezione tra la retta tangente e il guadagno di flessione, dove la pendenza della risposta a gradino ha il suo valore massimo.

Sulla base dei parametri appena visti (L, T) troviamo, attraverso la tabella 1, i valori migliori da prendere in considerazione.

Type of Controller	$K_p$	$T_i$	$T_d$
<b>P</b>	$\frac{T}{L}$	$\infty$	0
<b>PI</b>	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
<b>PID</b>	$1.2\frac{T}{L}$	2L	0.5L

*Tabella 1: Metodo Ziegler-Nichols anello aperto.*

Quindi il principale vantaggio di questo metodo è che fornisce un punto di partenza per la determinazione dei valori che caratterizzano un controllo PID.

### 1.5.2 Metodo della risposta in frequenza Ziegler-Nichols

Il secondo metodo di Ziegler-Nichols considera un sistema in catena chiusa dove è già presente un anello di controllo in cui poter cambiare i parametri.

Il primo step in questo metodo è quello di ridurre al minimo sia l'azione integrale che quella derivativa.

Detto ciò, è intuibile che il controllore lavori con un contributo puramente proporzionale con:

- $T_i = 0$
- $T_d = 0$

Si comincia inserendo un  $K_p$  basso e si aumenta progressivamente finché  $y(t)$  diventa un'oscillazione permanente di periodo  $T$  in corrispondenza di un valore di  $K_p$  detto guadagno critico  $K_{p^*}$ .

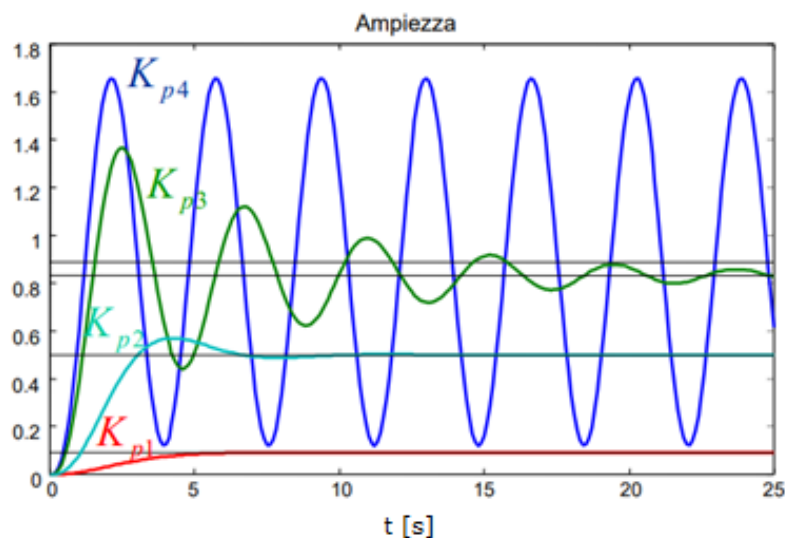


Figura 1.12: Uscita del sistema all'aumentare di  $K_p$

Come si nota in figura 1.12 all'aumentare di  $K_p$   $y(t)$  oscilla in maniera permanente. Se il sistema non entra mai in oscillazione, il metodo non è applicabile.

Aumentare  $K_p$  finché il sistema non ha oscillazioni perpetue significa ricercare il limite di stabilità.

sulla base di  $T$  e  $K_p$ , attraverso la tabella 2, si calcolano i parametri di controllo.

	$K_p$	$T_i$	$T_d$
<b>P</b>	$0.5\bar{K}_p$	-	-
<b>PI</b>	$0.45\bar{K}_p$	$0.8\bar{T}$	-
<b>PID</b>	$0.6\bar{K}_p$	$0.5\bar{T}$	$0.125\bar{T}$

*Tabella 2: Metodo Ziegler-Nichols anello chiuso*

In generale questo metodo fornisce dei risultati migliori rispetto al precedente; tuttavia, presenta la problematica di dover portare il sistema ai limiti della stabilità. Questo potrebbe essere un serio problema perché il sistema potrebbe subire danni. Tale modalità operativa infatti è fortemente rischiosa, soprattutto in impianti di grossa taglia. L'ampiezza della oscillazione permanente non può essere controllata, e può pertanto risultare eccessiva e tale da indurre situazioni di pericolo per gli organi dell'impianto o per gli operatori. Quindi, quando si utilizza questo metodo bisogna operare con la massima prudenza.

Per applicazioni critiche, si può ricorrere ad un miglioramento di questo metodo [1] dove attraverso una modifica è possibile ottenere delle nuove regole di ottimizzazione del controllore.



### 1.5.3 Metodo Relay (Åström-Hägglund)

È un metodo che cerca di superare il problema di Ziegler-Nichols in anello chiuso, il quale necessita di spingere il sistema fino al limite di stabilità.

Il metodo a relay genera un'oscillazione sostenuta della variabile di processo ma con l'ampiezza di quelle oscillazioni limitata a un intervallo di sicurezza.

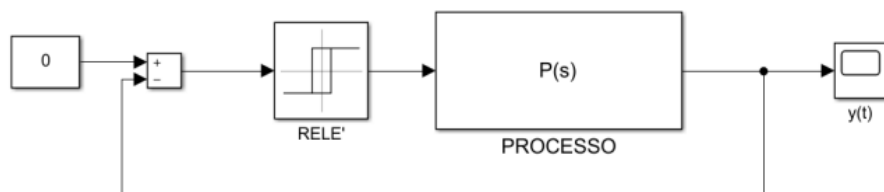


Figura 1.13: Schema metodo relay

La principale idea nel metodo Åström-Hägglund risiede nel concetto che quasi ogni processo stabile, soggetto a un relay feedback entra in un ciclo limite.

Con tutti e tre i termini PID temporaneamente disabilitati, il controller utilizza un relè on/off per applicare uno sforzo di controllo graduale al processo. Mantiene quindi costante lo sforzo di controllo e attende che la variabile di processo superi il setpoint.

A quel punto, applica un passaggio negativo e attende che la variabile di processo ritorni al di sotto del setpoint. La ripetizione di questa procedura ogni volta che la variabile di processo passa il setpoint in entrambe le direzioni costringe la variabile di processo a oscillare fuori sincrono con lo sforzo di controllo ma alla stessa frequenza.

Si instaura una condizione di regime oscillatorio periodico in cui l'uscita oscilla intorno al valore del set point. Sebbene le oscillazioni della variabile di processo non siano strettamente sinusoidali, il loro

periodo risulta essere una stretta approssimazione del periodo finale che Ziegler e Nichols hanno usato per le loro regole.

L'ampiezza  $A$  e la frequenza  $\Omega$  della armonica fondamentale di tale oscillazione sono determinabili attraverso le formule seguenti:

$$A = \frac{4h}{\pi k_{cr}} \qquad \Omega = \omega_{cr}$$

dove:

- $h$  = ampiezza del relay
- $k_{cr} = K_u$
- $T = T_u$

La procedura di questo metodo prevede che si rilevino il periodo  $T$  e l'ampiezza  $A$  della oscillazione ottenuta impiegando un relay di ampiezza  $h$ .

Valutati tali parametri, i coefficienti  $T_u$  e  $K_u$  che devono essere impiegati per il calcolo dei guadagni del controllore si determinano secondo le seguenti relazioni:

$$T_u = T \qquad k_u = \frac{4h}{\pi A}$$

A questo punto, estratte tutte le informazioni necessarie dalle oscillazioni, il controllore può essere tarato attraverso la tabella 3:

	$K_p$	$T_I$	$T_D$
P	$0.5 \cdot K_u$		
PI	$0.4 \cdot K_u$	$0.8 \cdot T_u$	
PID	$0.6 \cdot K_u$	$0.5 \cdot T_u$	$0.125 \cdot T_u$

*Tabella 3: Metodo relay*

A differenza del test a circuito chiuso originale di Ziegler-Nichols, il test del relay può essere configurato per limitare l'ampiezza delle oscillazioni della variabile di processo fissando l'ampiezza delle oscillazioni dello sforzo di controllo a un valore definito dall'utente. La variabile di processo non ha bisogno di oscillare così selvaggiamente da mettere in pericolo il processo.

Il metodo è stato migliorato per avere risultati completamente affidabili. Come si vede nell'articolo [2] infatti si possono avere varie configurazioni del relay:

- Relay ideale
- Relay con isteresi
- Relay + Delay
- Relay + integrator
- Relay con precarico

Per esempio, l'utilizzo di un relay con isteresi può evitare i problemi di rumore che nei sistemi di controllo devono essere limitati al minimo. Infine, di questo metodo si può dire che il controller può essere configurato per condurre il test senza l'intervento dell'operatore. Teoricamente, anche un operatore che non ha familiarità con questo metodo può premere un pulsante e lasciare che il controller conduca il proprio test del relè e selezioni di conseguenza i parametri necessari.

Che si tratti del metodo relè Åström-Hägglund o di una tecnica simile, tali funzioni di auto-tuning sono diventate caratteristiche standard sulla maggior parte dei controller PID commerciali.

# Capitolo 2

## 2. Introduzione del progetto

L'obiettivo di questa tesi è quello di realizzare un sistema di controllo della temperatura all'interno di un box riscaldato tramite lampada alogena.

Per il controllo della temperatura all'interno del box sono stati realizzati due setup: il primo ha visto l'utilizzo di un modulo relay come componente controllata, poi sostituito nel secondo setup da un modulo dimmer comandato da controllore PID. In entrambe le configurazioni l'elemento riscaldante utilizzato è una lampada alogena, mentre il sensore per la misura della temperatura è una termocoppia.

Utilizzando il primo setup sono emersi dei limiti sull'accuratezza del controllo del setpoint di temperatura, che sono stati invece superati dal secondo setup.

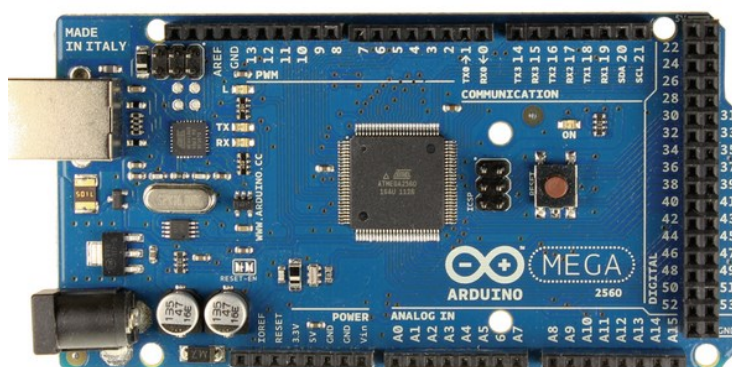
Tra le diverse caratteristiche positive, che tali sistemi garantiscono, ci sono:

- Affidabilità
- Semplicità di utilizzo e taratura
- Per sfruttare pienamente algoritmi, di controllo più complessi, è necessaria una conoscenza approfondita del processo da controllare. Nel caso dei PID questo non è necessario.
- Notevole efficacia nella regolazione di un'ampia gamma di processi industriali. Se ben
- Prestazioni soddisfacenti.
- Possibile realizzazione con diverse tecnologie (meccanica, idraulica, elettrica)
- Rapporto efficacia / costo ottima, ancora oggi difficilmente superabile.

Detto ciò, si introducono tutte le componenti utilizzate nel progetto ed i due setup utilizzati.

## 2.1 Arduino Mega 2560

L' Arduino Mega 2560 è il modulo di controllo utilizzato per la realizzazione del progetto. Questa dispone al suo interno di varie strutture hardware predispose alla comunicazione, all'adattamento e controllo delle tensioni, e altre. Inoltre, dispone di vari pin per la tensione, ingressi analogici, pin digitali di I/O e PWM e anche il riferimento digitale. Dispone di una presa femmina usb per la comunicazione e una presa femmina per il jack dell'ingresso della tensione. La scheda ha il seguente aspetto:



*Figura 2.1: Scheda Arduino Mega*

La scheda sotto esame [3] prende il nome "2560" dal tipo di microcontrollore usato ovvero l'ATMEGA2560. Come si vede in figura 2.1 i pin di tensione sono:

- IOREF: uscita della tensione al quale viene alimentato il controllore

- RESET: pin che serve per resettare il microcontrollore
- 3V, 5V
- GND e Vin: pin tensione di massa

Inoltre, essa dispone di un totale di 54 pin di input/output digitali programmabili, di cui 15 utilizzabili per fornire in uscita segnali modulati PWM (0-14) e un totale di 16 pin di input analogici.

Come riportato nell'articolo [4] "la vera novità di Arduino è permettere a chiunque fosse dotato di un minimo di creatività, passione per l'elettronica e una strumentazione di base, di trasformare in qualcosa di concreto un'idea o un progetto, semplice o complesso che fosse" ci fa capire come tramite questa scheda è possibile realizzare innumerevoli progetti.

## **2.2 Termocoppia – Modulo MAX6675**

Le termocoppie sono sensori di temperatura a resistenza utilizzati per la misurazione di temperature, trovando applicazione in moltissimi settori. Questi sensori hanno dimensioni ridotte e tempi di misurazione molto rapidi, e grazie a una grande resistenza è possibile utilizzare le termocoppie anche in condizioni critiche.

Sono diverse le tipologie di termocoppie presenti sul mercato oggi, create per adattarsi ad una vasta gamma di scenari applicativi e sfide ambientali e, soprattutto, a diversi intervalli di temperatura. Per distinguere i diversi modelli generalmente si utilizzano le lettere dell'alfabeto, tra i tipi più comuni di termocoppie ci sono le J, K, L, N e T.

In questo progetto è stata utilizzata la termocoppia di tipo K, che ha un range di temperatura misurabile da 0°C a 600°C.

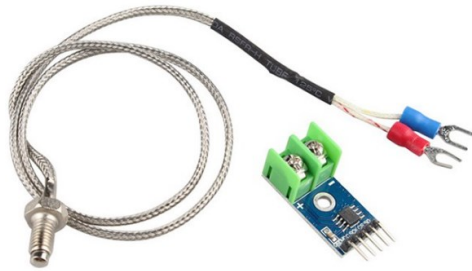


Figura 2.2: Termocoppia - MAX6675

Per la misura della temperatura, la termocoppia necessita del modulo MAX6675: un convertitore digitale che fornisce un'interfaccia seriale con Arduino.

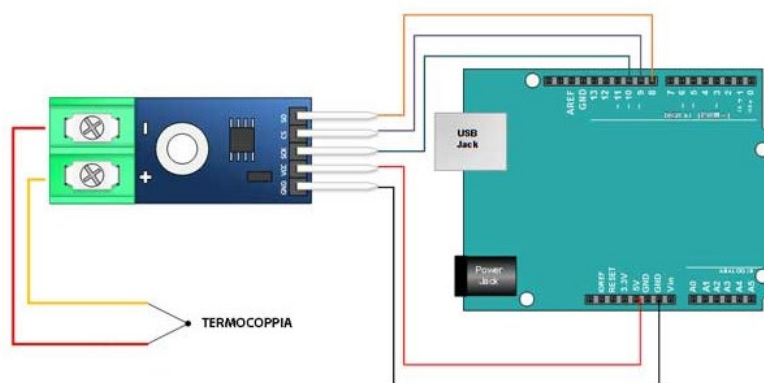


Figura 2.3: Schema di collegamento Termocoppia - MAX6675 - Arduino

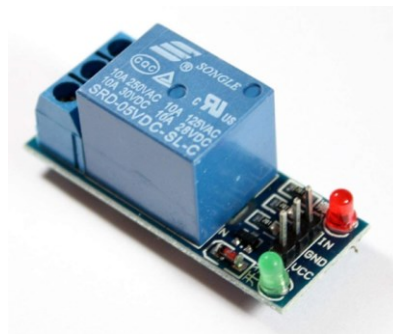
Infine, facendo riferimento al datasheet del MAX6675 [5] che ne riporta tutte le caratteristiche, questi moduli sono stati collegati tra loro.

## 2.3 Modulo relay

Il relay è un dispositivo elettromeccanico che consente di commutare più circuiti, sia separatamente sia in gruppo o in sequenza. In origine

veniva azionato da un elettromagnete che apriva o chiudeva uno o più gruppi di contatti. Questo dispositivo è, quindi, in grado di pilotare in uscita un circuito di potenza superiore rispetto al circuito di ingresso.

In sostanza il relay non è altro che un interruttore che può essere azionato, come nel nostro caso, attraverso un segnale digitale (alto, oppure basso).



*Figura 2.4: Modulo relay*

Il relay in figura 2.4 ha tre pin in ingresso, che sono collegati al modulo Arduino: il pin Vcc, il pin GND, il pin IN. Questi pin vanno collegati rispettivamente all'alimentazione a 5 V, alla messa a terra ed al pin del segnale di controllo. Dalla parte opposta c'è il circuito in media tensione dove troviamo tre connettori, due dei quali sono stati utilizzati per inserire l'interruttore in serie al circuito a media tensione (220V). Bisogna collegare il pin centrale ed il pin NO se il circuito normalmente è aperto (per esempio una lampada che normalmente è spenta aprendo il circuito), altrimenti si usa sempre il connettore centrale e quello NC per circuiti normalmente chiuso. Nel nostro caso è stato utilizzato il pin NO con la lampada come si vede in figura 2.7.



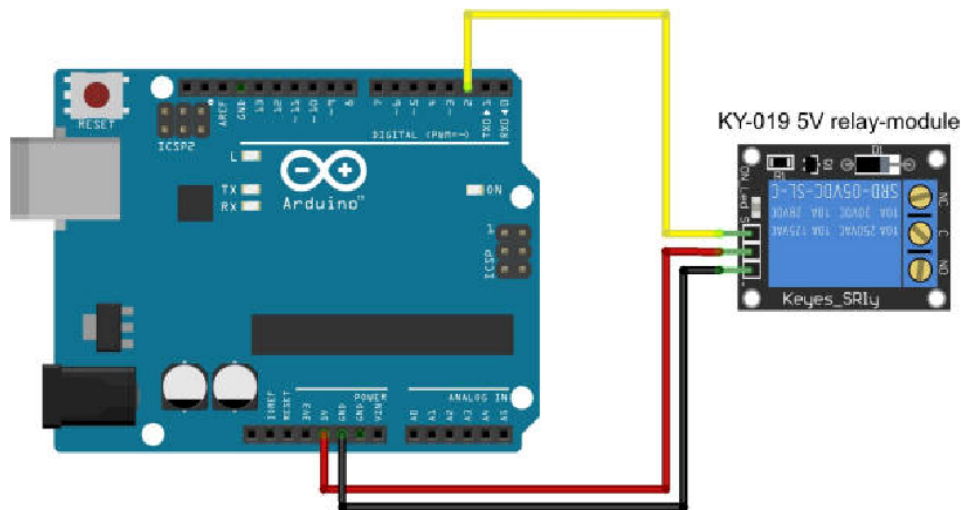


Figura 2.5: Circuito Arduino - Relay

Le caratteristiche di questo relay si trovano nel suo datasheet [6], dove sono elencate i limiti di utilizzo di tale componente. In commercio infatti c'è una grande varietà di relay, i quali vengono scelti in base al tipo di utilizzo.

## 2.4 Display LCD 16X2 con retroilluminazione

Per monitorare il cambio di temperatura è stato utilizzato il display a cristalli liquidi o LCD (Liquid Crystal Display) il quale è contraddistinto da 2 righe e 16 caratteri. Questa tipologia di display è definita intelligente perché ha all'interno un controller che ne semplifica la gestione mediante i microcontrollori. L'interfaccia utilizzata è la HD44780 della Hitachi, divenuta uno standard. Altra caratteristica importante è il colore della retroilluminazione, che come si vede in figura 2.6 è di sfondo verde con caratteri di colore nero.

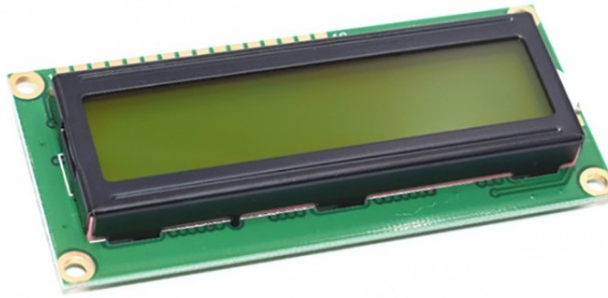


Figura 2.6: Display LCD

La piedinatura utilizzata solitamente in un display LCD è la seguente:

Pin	Denominazione	Descrizione
1	Vss o Gnd	Massa
2	Vcc	+5V
3	Vcontrasto	Regolazione contrasto ( 5 V )
4	RS	Register Select (selezione comando RS="1", dati RS="0")
5	R/W	Read/Write R/W="0" per inviare dati, R/W="1" per leggere dati
6	E	Enable (Abilitazione) linea di controllo, con E="1" si abilita il display a ricevere dati e istruzioni
7	D0	Linea Dati, bit 0
8	D1	Linea Dati, bit 1
9	D2	Linea Dati, bit 2
10	D3	Linea Dati, bit 3
11	D4	Linea Dati, bit 4
12	D5	Linea Dati, bit 5
13	D6	Linea Dati, bit 6
14	D7	Linea Dati, bit 7
15	A	Anodo Retroilluminazione (+4.2V)
16	K	Catodo Retroilluminazione (Massa)

Tabella 4: Piedinatura LCD

Nel montaggio del display sono state necessarie 2 resistenze:

- Per la regolazione del contrasto una resistenza da 3 k $\Omega$
- Per la retroilluminazione una resistenza da 220  $\Omega$

## 2.5 Lampada alogena (220 W)

L'elemento utilizzato per aumentare la temperatura all'interno del box, è una lampada alogena (220 W). Collegando quest'ultima al relay abbiamo completato tutti i collegamenti elettrici del progetto.

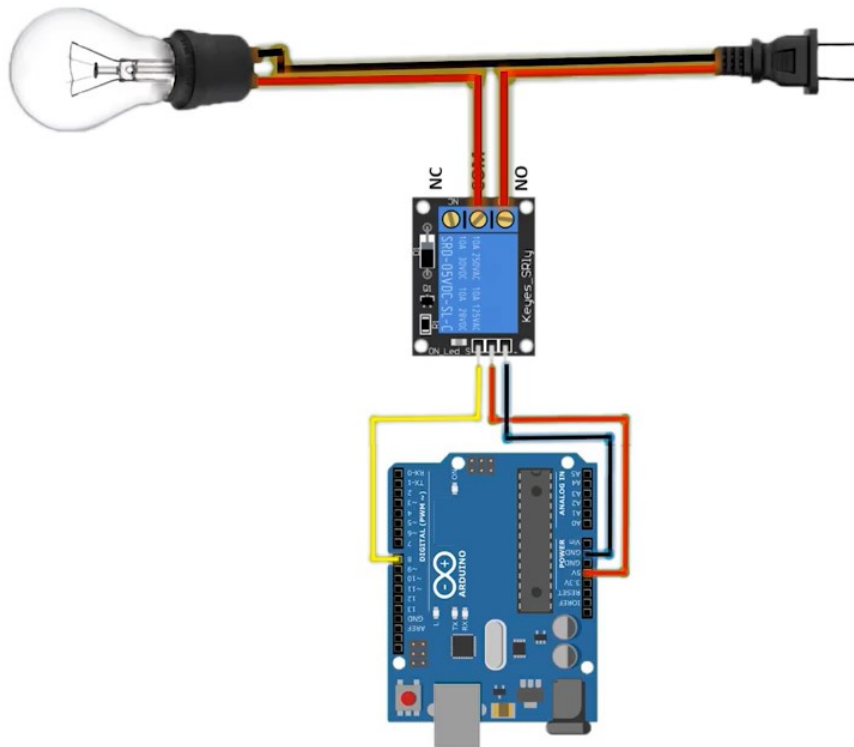
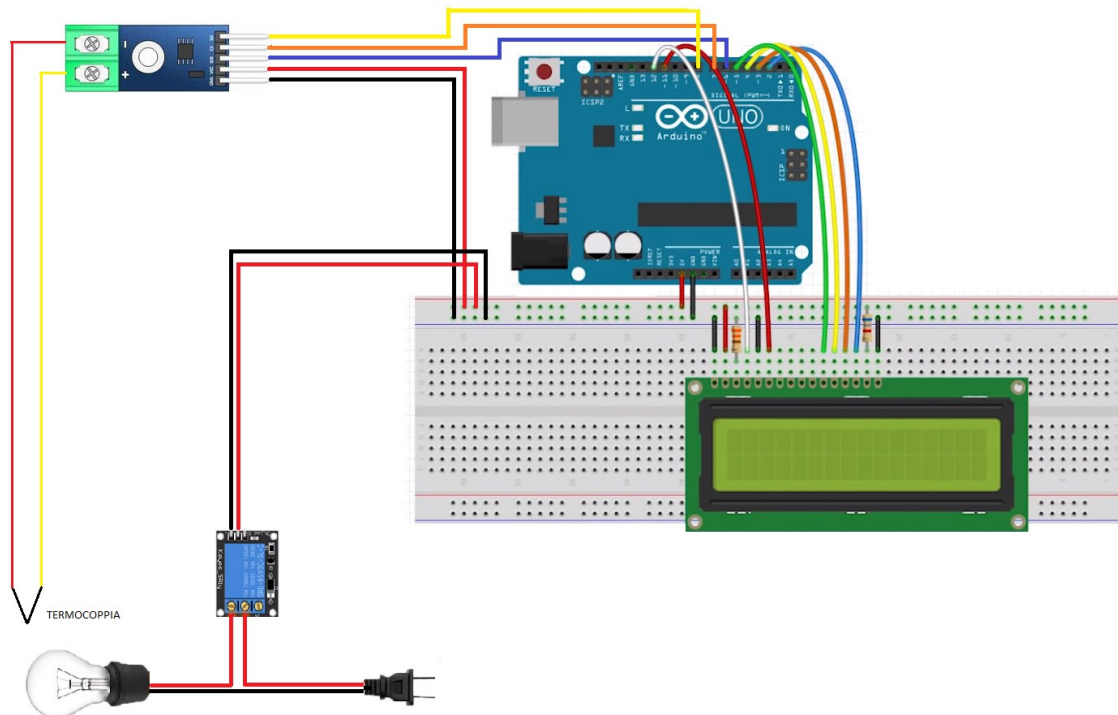


Figura 2.7: Lampada - Relay - Arduino

## 2.6 Primo Setup

Ora non resta altro che collegare tutte le varie componenti ed ottenere così il circuito finale:



*Figura 2.8: Primo setup*

## 2.7 Codice Arduino per il controllo del primo setup

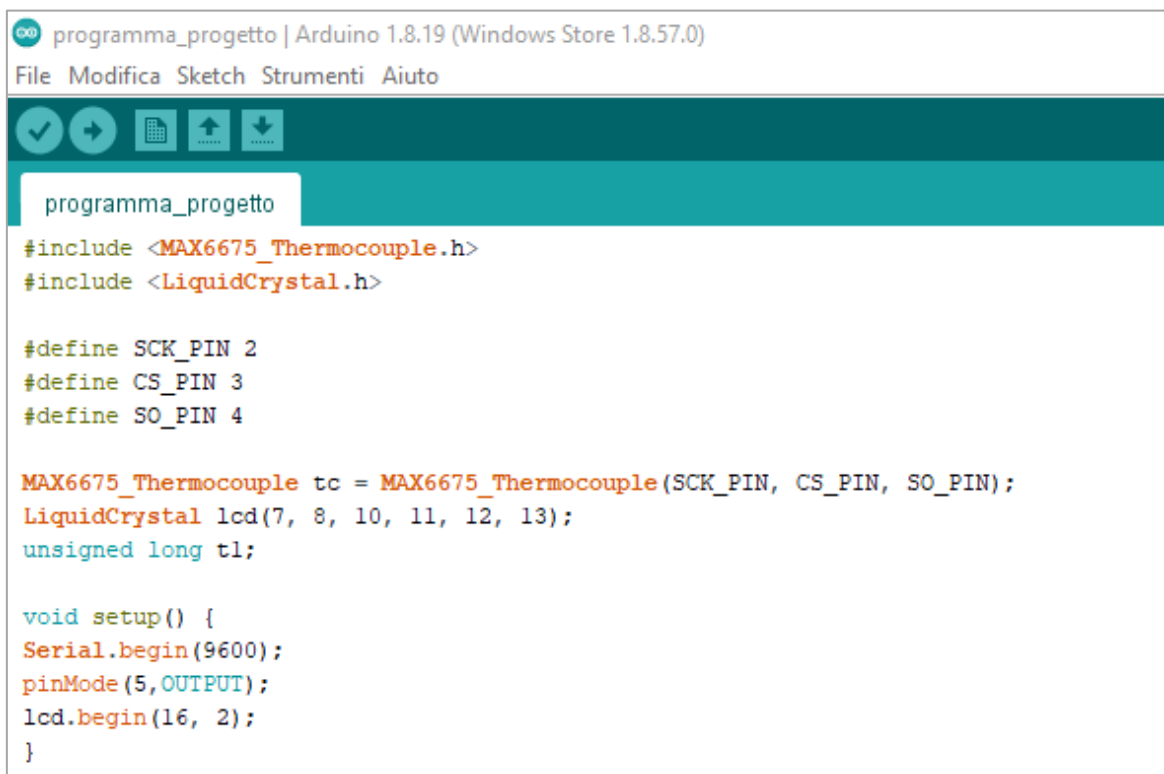
Nel mondo al giorno d'oggi esistono diversi programmi e linguaggi di programmazione. In questo progetto è stato utilizzato il programma Arduino.

Il linguaggio di programmazione di Arduino utilizza il Wiring, derivato dal C++, utilizzato per scrivere programmi installabili sul microcontrollore ATmega328 che equipaggia proprio Arduino. Il microcontrollore legge, attiva e disattiva le porte di Arduino seguendo

le istruzioni contenute in uno sketch, e cioè un programma scritto in wiring e compilato attraverso l'IDE. Quest'ultimo è un ambiente di sviluppo indispensabile per operare su Arduino.

Per quanto riguarda tutte le funzioni, variabili, costanti e tutto il necessario per scrivere un programma su Arduino, facciamo riferimento al sito ufficiale di Arduino [7].

Di seguito è riportato il programma per il controllo della temperatura tramite relay:



```
programma_progetto | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Modifica Sketch Strumenti Aiuto

programma_progetto
#include <MAX6675_Thermocouple.h>
#include <LiquidCrystal.h>

#define SCK_PIN 2
#define CS_PIN 3
#define SO_PIN 4

MAX6675_Thermocouple tc = MAX6675_Thermocouple(SCK_PIN, CS_PIN, SO_PIN);
LiquidCrystal lcd(7, 8, 10, 11, 12, 13);
unsigned long t1;

void setup() {
  Serial.begin(9600);
  pinMode(5, OUTPUT);
  lcd.begin(16, 2);
}
```

```

void loop() {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temperature");
  lcd.setCursor(0,1);
  double temp = tc.readCelsius();

  lcd.print(String(temp)+ char(0xDF) + "C");
  delay(1000);

  //tc.readKelvin() readFahrenheit()
  Serial.print("temp=");
  Serial.println(temp);
  // if temprature goes above 80.0C, turn the relay OFF
  if(tc.readCelsius()>36.00){
    digitalWrite(5, HIGH);// set pin 10 LOW
  }else{
    digitalWrite(5, LOW);// set pin 10 HIGH
  }
  delay(5);
}

```

*Figura 2.9: Codice Arduino*

## 2.8 Modulo Dimmer

Come detto prima, nel secondo setup il modulo relay è stato sostituito da un modulo dimmer. Quest'ultimo è basato sul TRIAC da 600 V ed è in grado di regolare la luminosità della lampada alogena.



*Figura 2.10: Modulo Dimmer*

Può essere collegato a microcontrollori con logica a 5V e 3,3V e quindi è possibile collegarlo alla scheda Arduino. Può essere utilizzato anche in ventilatori, pompe, filtri dell'aria, e molti altri dispositivi. Detto ciò, è facilmente intuibile come il dimmer al giorno d'oggi sia un dispositivo che trova un ampio utilizzo in diversi settori.

Infine, per quanto riguarda il codice da utilizzare, questo dispositivo può essere controllato attraverso la libreria RBDdimmer.h[8].

## 2.9 Secondo Setup

In figura 2.11 è possibile vedere come il relay è stato sostituito dal dimmer. I pin da collegare sono i seguenti:

- Vcc al pin 5V o 3,3V del microcontrollore
- GND a terra GND del microcontrollore
- Z-C
- PWM

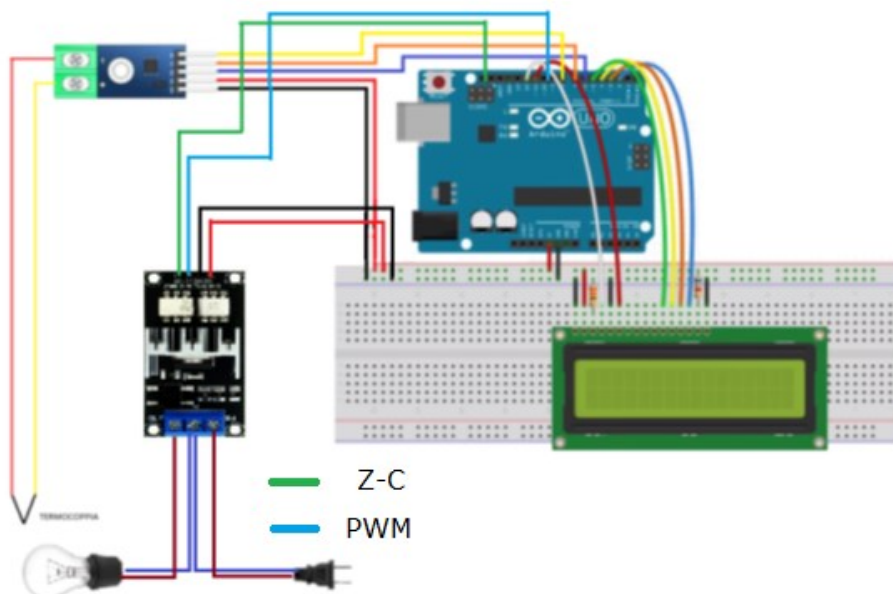


Figura 2.11: Circuito finale con Modulo Dimmer

## 2.10 Codice Arduino per il controllo del secondo setup

Come sopra, utilizziamo il programma Arduino per scrivere il codice necessario al controllo del circuito. In aggiunta al codice sopra riportato sono qui utilizzate la libreria PID\_V1.h e la libreria RBDdimmer.h.

```
programmapid
#include <RBDdimmer.h>
#include <LiquidCrystal.h>
#include <PID_v1.h>
#include <MAX6675_Thermocouple.h>

#define SCK_PIN 5
#define CS_PIN 3
#define SO_PIN 4

#define outputPin 9
#define zerocross 2
dimmerLamp dimmer(outputPin);

MAX6675_Thermocouple tc = MAX6675_Thermocouple(SCK_PIN, CS_PIN, SO_PIN);

double setpoint, input, output;
double Kp = 1.8, Ki = 7.5, Kd = 1.875;
unsigned long pidtime = 1000;
PID myPID(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);

unsigned long t1;

LiquidCrystal lcd(7, 8, 10, 11, 12, 13);

void setup() {
  Serial.begin(9600);
  dimmer.begin(NORMAL_MODE, ON);
  myPID.SetSampleTime(pidtime);
  myPID.SetOutputLimits(0, 80);
  myPID.SetMode(AUTOMATIC);
  myPID.SetTunings(Kp, Ki, Kd);

  lcd.begin(16, 2);
}
```



```

void loop() {
    float temp = tc.readCelsius();
    if((millis() - t1) > pidtime){
        input=temp;
        setpoint = 45.0;
        myPID.Compute();
        int pwm = constrain((int)output, 0, 80);
        dimmer.setPower(pwm);

        Serial.print("\nin: ");
        Serial.print(input);
        Serial.print("\tout: ");
        Serial.print(output);
        Serial.print("\tpwm: ");
        Serial.print(pwm);

        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Temperature");
        lcd.setCursor(0,1);
        temp = tc.readCelsius();

        lcd.print(String(temp)+ char(0xDF) + "C");
        delay(1000);
    }
}

```

*Figura 2.12. Codice Arduino - PID - Dimmer*

## Capitolo 3

### 3. Costruzione del Banco di Misura e Controllo

Il banco di misura è stato creato in modo tale da favorire la praticità e semplicità d'uso del sistema. Il banco è composto da una base in legno di 80 cm x 40 cm sulla quale è fissato il Box di L 55 x H 30.5 x P 38.5 cm trasparente.

Sono stati applicati due fori:

- Il primo dietro il box per facilitare l'ingresso della termocoppia al suo interno
- Il secondo sopra il box per consentire l'uscita dei fili di tensione della lampada alogena

Successivamente nella parte superiore interna del Box si è montata la lampada alogena.

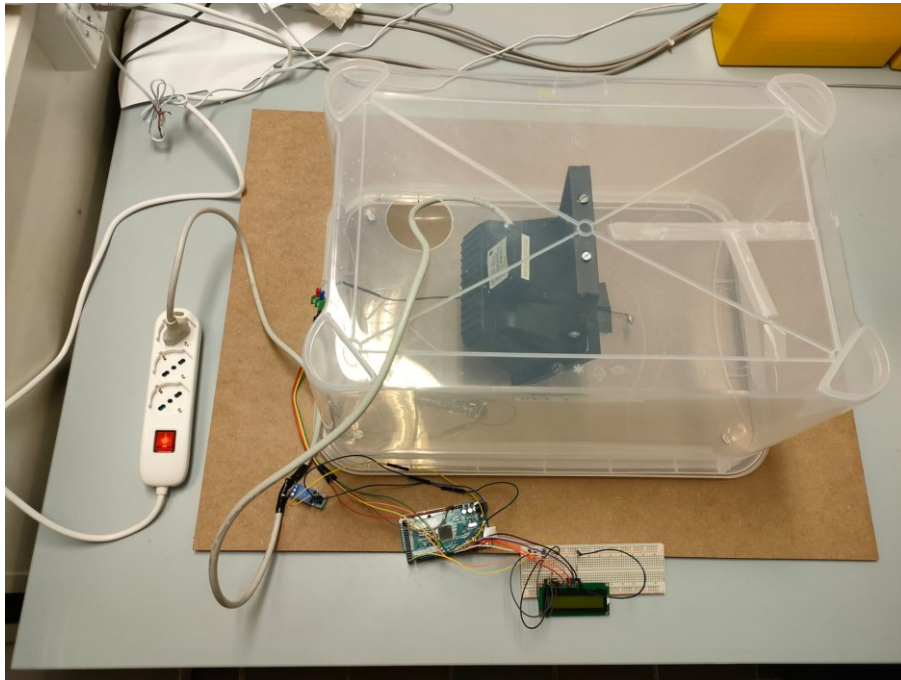
Successivamente sono collegati tutte le componenti descritte prima ovvero:

- Arduino Mega 2560
- Termocoppia – MAX6675
- Display
- Relay
- Dimmer

L'alimentazione utilizzata per l'Arduino Mega è tramite USB, utilizzabile dato che quest'ultimo è collegato al PC.

### 3.1 Fasi costruttive del banco

In figura 3.1 si vede il banco finale utilizzato per il controllo della temperatura.



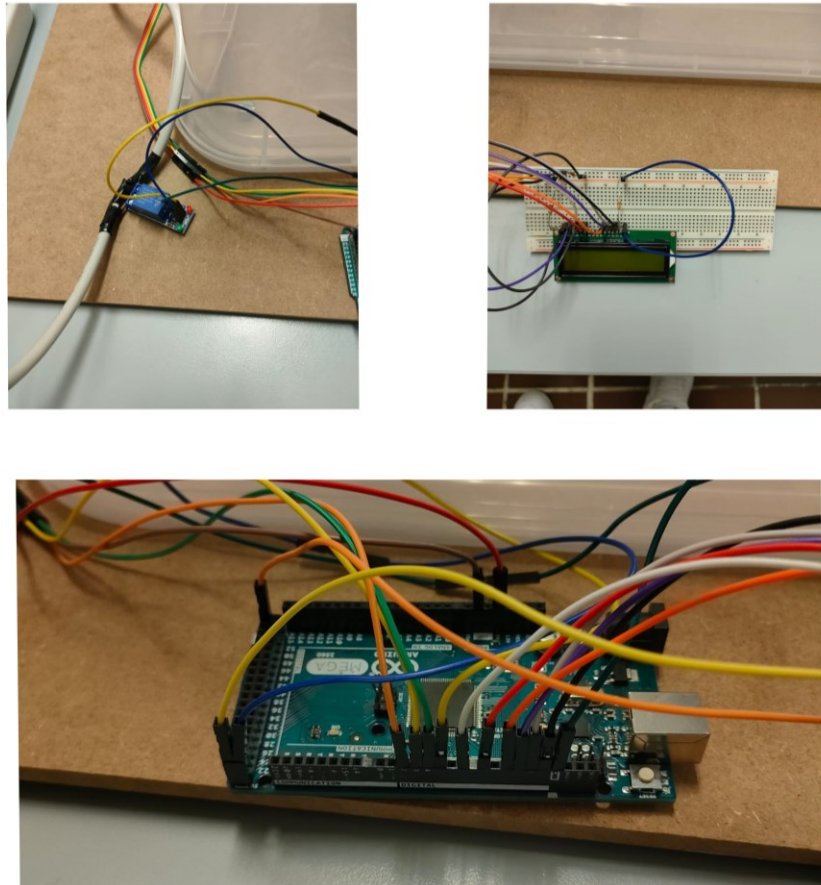
*Figura 3.1: Banco finale relè*

Per quanto riguarda il collegamento dei vari pin di Arduino con i rispettivi componenti si sono utilizzati:

- Pin Ground
- Pin Vcc
- CS – Termocoppia Pin 3
- SO – Termocoppia Pin 4
- SCK – Termocoppia Pin 5
- Display LCD pin (7, 8, 10, 11, 12, 13)

Durante il collegamento dei vari componenti si è reso necessario l'utilizzo di una breadboard attraverso la quale è stata possibile finalizzare nella maniera più corretta tutti i vari collegamenti,

soprattutto del display che necessita delle resistenze per il contrasto e per la retroilluminazione.



*Figura 3.2: Collegamenti Arduino - Relay - Display LCD*

Effettuate tutte le prove necessarie con il primo setup (relay) si è passati a collegare il modulo Dimmer utilizzando i seguenti pin:

- Output Pin 9
- ZC Pin 2

Si ottiene così il banco per la misura e il controllo della temperatura tramite secondo setup.

## 3.2 Manuale d'uso

1. Verificare che il cavo di alimentazione sia scollegato.
2. Controllare che tutti i cavi dell'Arduino siano collegati nel modo giusto e che nessun cavo sia scollegato.

PIN	
5V	5V
GND	GND
2	ZC-Dimmer
3	CS-Termocoppia
4	SO-Termocoppia
5	SCK-Termocoppia
7	LCD-RS
8	LCD-E
9	Output Dimmer
10	LCD-D4
11	LCD-D5
12	LCD-D6
13	LCD-D7

*Tabella 5: Elenco Pin*

3. Impostare nel programma scritto su Arduino IDE il valore di setpoint di temperatura desiderato.
4. Scaricare le librerie necessarie:
  - RBDdimmer.h
  - LiquidCrystal.h
  - PID\_v1.h
  - MAX6675\_Thermocouple.h

5. Nel caso i Pin utilizzati siano diversi da quelli in tabella 5 è necessario cambiarli anche nel programma.
6. Collegare la lampada alla multipresa.
7. Collegare l'Arduino tramite la porta USB del Pc.
8. Tramite la funzione "verifica" all'interno del programma Arduino IDE verificare che non ci sia nessun problema.
9. Tramite la funziona "Carica" si fa partire il programma e si controlla la temperatura attraverso il display LCD.

## Capitolo 4

### 4. Risultati Sperimentali

Come detto, l'obiettivo finale di questa tesi è quella di costruire un banco di misura e controllo della temperatura. Si è rivelato necessario quindi trovare il miglior metodo per controllare il setpoint di temperatura. Per far ciò sono stati utilizzati due setup: inizialmente un relay, per poi passare successivamente all'utilizzo di un dimmer con il sistema di controllo PID.

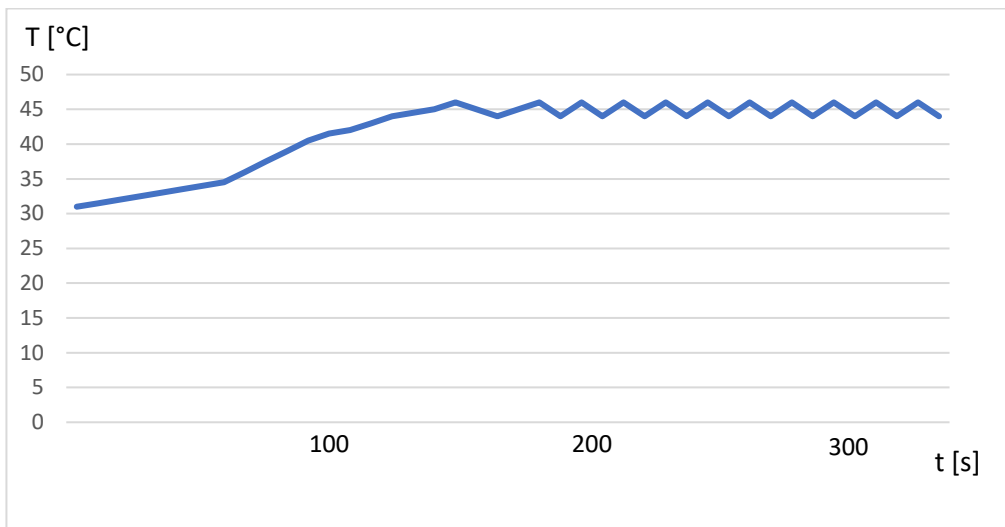
Si vedrà come passando dall'utilizzo di un semplice relay all'utilizzo di un sistema PID con dimmer ci sarà un miglioramento nel controllo della temperatura.

Per quanto riguarda il sistema PID, sono state fatte inizialmente delle prove con il controllore P, poi con il controllore PI ed infine con il controllore PID. Fatte queste, è stato valutato il sistema di controllo migliore.

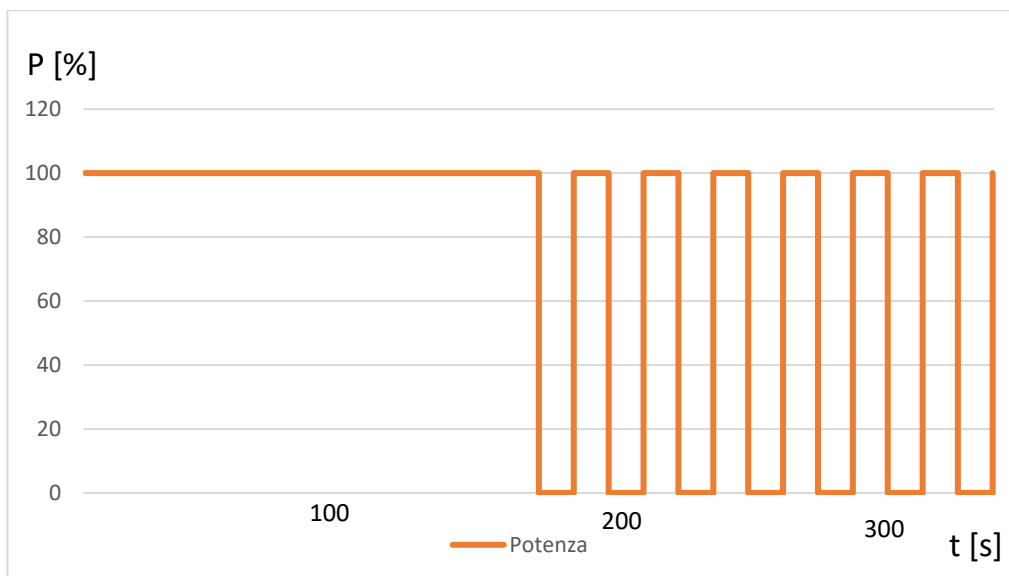
#### 4.1 Sistema di controllo con modulo relay

Il primo setup ha visto l'utilizzo del modulo relay per il controllo della temperatura, tramite la funzione ON-OFF

Come si vede in figura 4.1 la temperatura, una volta raggiunto il setpoint di 45°C preimpostato, ha un andamento discontinuo e oscilla tra un valore di 43°C e 47°C. Questo è dovuto al fatto che la lampada alogena passa da una potenza del 0% (stato OFF del relay) a 100%(stato ON del relay) non appena la temperatura scende oppure va oltre il setpoint.



*Figura 4.1: Andamento Temperatura*



*Figura 4.2: Andamento Potenza*

Per questo motivo, utilizzare come sistema di controllo un circuito con all'interno un modulo relay non è la scelta migliore. Prima di tutto perché non garantisce un controllo della temperatura accurato, ma



anche perché nel tempo la lampada potrebbe essere soggetta a problemi dovuti dal ripetuto accendersi/spegnersi dettato dal relay

## **4.2 Sistema di controllo con modulo dimmer**

Dopo aver fatto una profonda analisi dei parametri utilizzati nei controllori PID, si è deciso di applicare il metodo Ziegler-Nichols ad anello chiuso per poter trovare il miglior setup di quest'ultimi. In linea teorica il sistema deve arrivare al valore di setpoint impostato il prima possibile e non deve produrre oscillazioni. Tuttavia, se i parametri non sono corretti il sistema risulterà instabile e il comportamento del sistema di controllo non sarà ottimale.

In generale, per realizzare un sistema di controllo sarebbe sufficiente utilizzare una risposta proporzionale all'errore in ingresso ovvero utilizzare un sistema di controllo P. Tuttavia, aumentando il valore di  $k_p$  non si riesce a raggiungere un equilibrio stabile.

Proprio per questo motivo, si userà un controllore P fino ad arrivare ad un controllore PID. Si analizzeranno i vantaggi e gli svantaggi nell'utilizzo di un tipo di controllore rispetto ad un altro per poi scegliere quale tipo di controllore darà risultati migliori.

### **4.2.1 Metodo Ziegler-Nichols ad anello chiuso**

Il metodo Ziegler-Nichols ad anello chiuso viene utilizzato annullando o riducendo al minimo i contributi della parte integrale e derivativa del sistema di controllo PID. Successivamente si aumenta in maniera progressiva e prudentemente il valore della variabile di controllo  $K_p$  fino a quando si osserva che l'uscita  $y(t)$  del sistema oscilla in modo permanente.

Quindi, si parte impostando  $K_p = 5$  e si osserva in figura 4.3 la variazione della temperatura e della potenza con il passare del tempo.

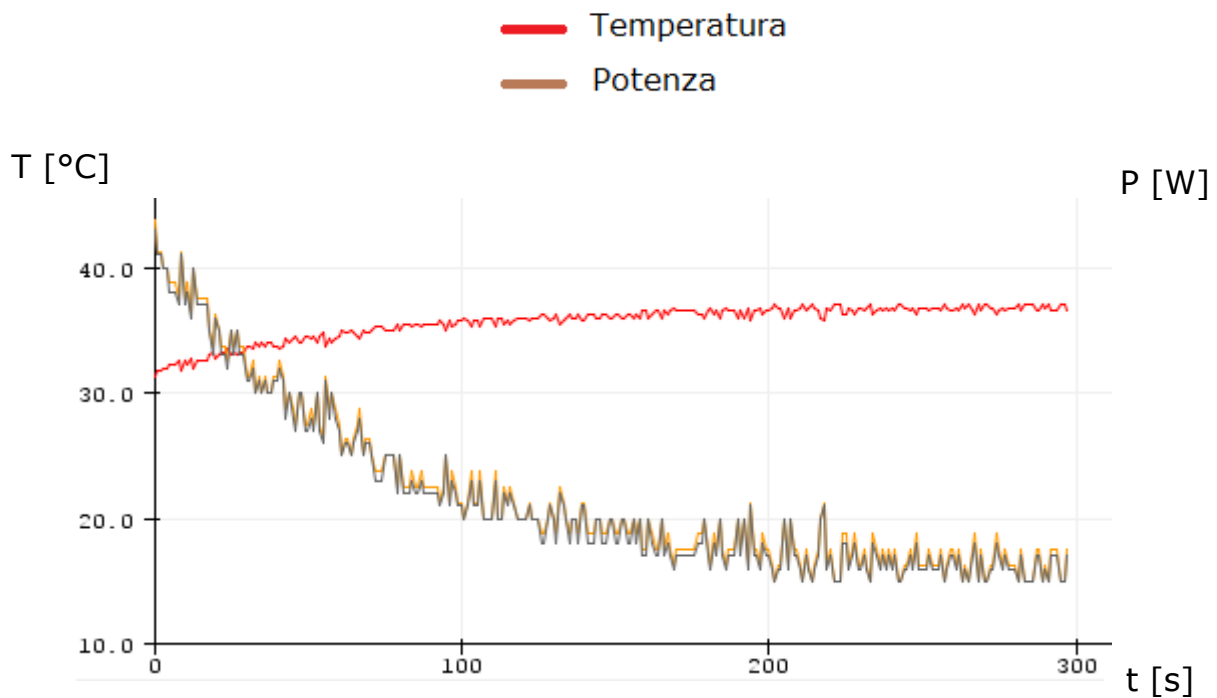


Figura 4.3:  $K_p = 5$

Si ottiene un sistema instabile che non si avvicina neanche al valore di setpoint impostato di 40 °C. Si continua dunque aumentando il valore di  $K_p$ .

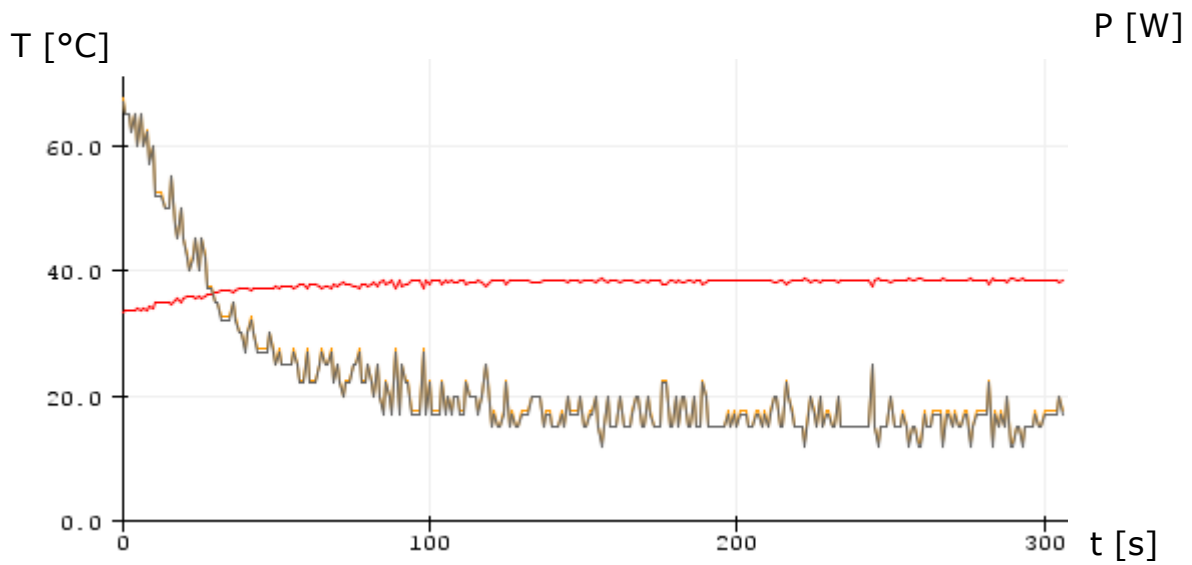


Figura 4.4.  $K_p = 10$

Come si osserva in figura 4.4 con un valore di  $K_p$  leggermente superiore ( $K_p=10$ ) il sistema comincia a tendere al valore di setpoint. Tuttavia, non è ancora possibile utilizzare il metodo Ziegler-Nichols e dunque si procede aumentando ancora il valore di  $K_p$ .

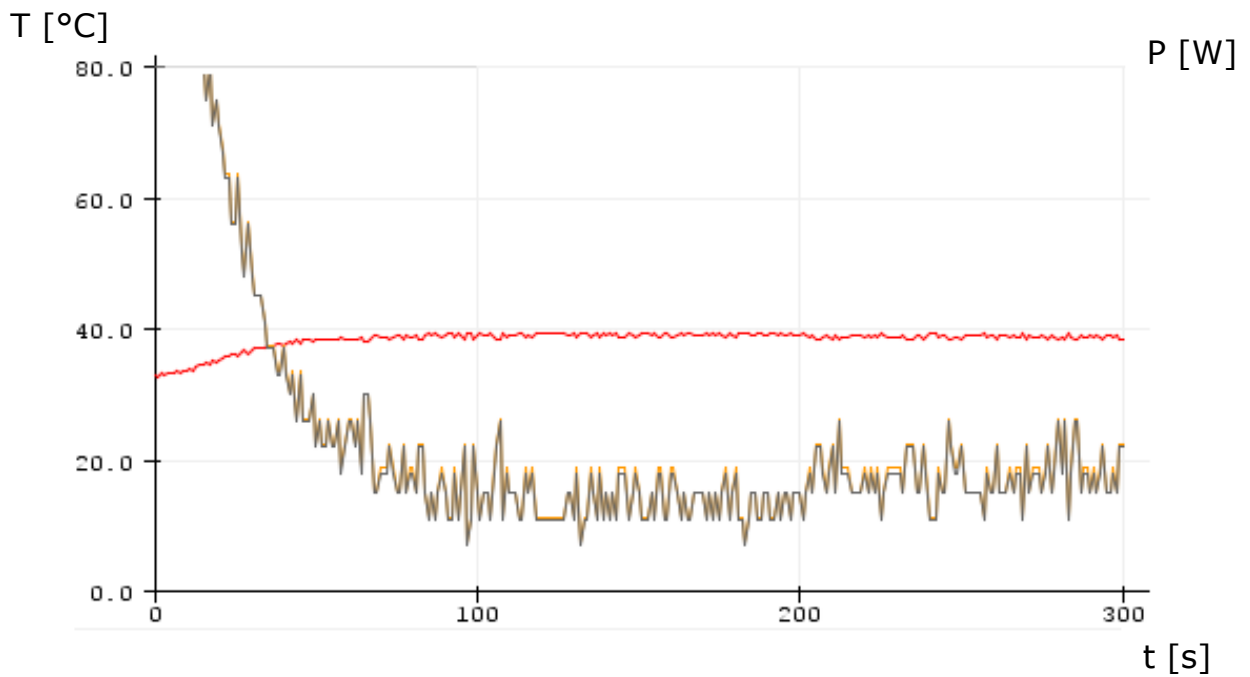
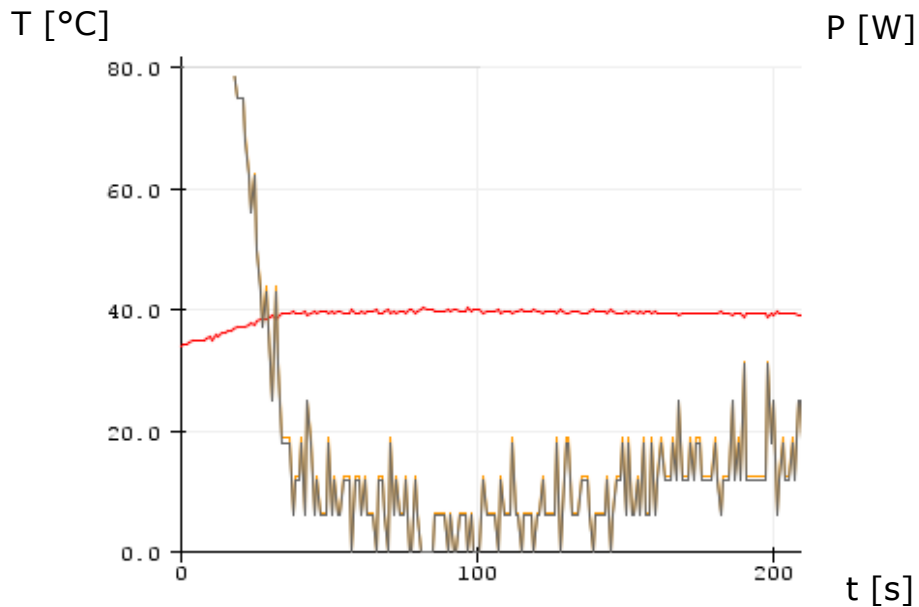


Figura 4.5:  $K_p = 25$

Si arriva quindi a un valore di  $K_p = 25$  che non è ancora sufficiente dato che non permette ancora di raggiungere il valore di setpoint.



Finalmente, con un  $K_p=35$  il valore di setpoint di  $40\text{ }^\circ\text{C}$  è raggiunto ed il sistema è in oscillazione. A questo punto dalla figura 4.6 ricaviamo i valori  $\bar{K}_p = 35$  e  $\bar{T} = 10$  e infine tramite la tabella 6 troviamo tutti i valori che ci servono.

	$K_p$	$T_i$	$T_d$
<b>P</b>	$0.5\bar{K}_p$	-	-
<b>PI</b>	$0.45\bar{K}_p$	$0.8\bar{T}$	-
<b>PID</b>	$0.6\bar{K}_p$	$0.5\bar{T}$	$0.125\bar{T}$

Tabella 6: Metodo Ziegler-Nichols

Calcolate le costanti corrette, è possibile valutare la differenza tra controllore P, PI, PID.

### 4.2.2 Sistema di controllo P

Tramite il valore di  $\overline{K_p} = 35$  trovato precedente andiamo a calcolare il valore di  $K_p$  che permette di ricavare i risultati migliori.

	$K_p$	$T_i$	$T_d$
Controllore P	17.5	0	0

Tabella 7: Sistema di controllo P

Quindi, inserisco nel programma Arduino il valore di  $K_p$  trovato nella tabella 7.

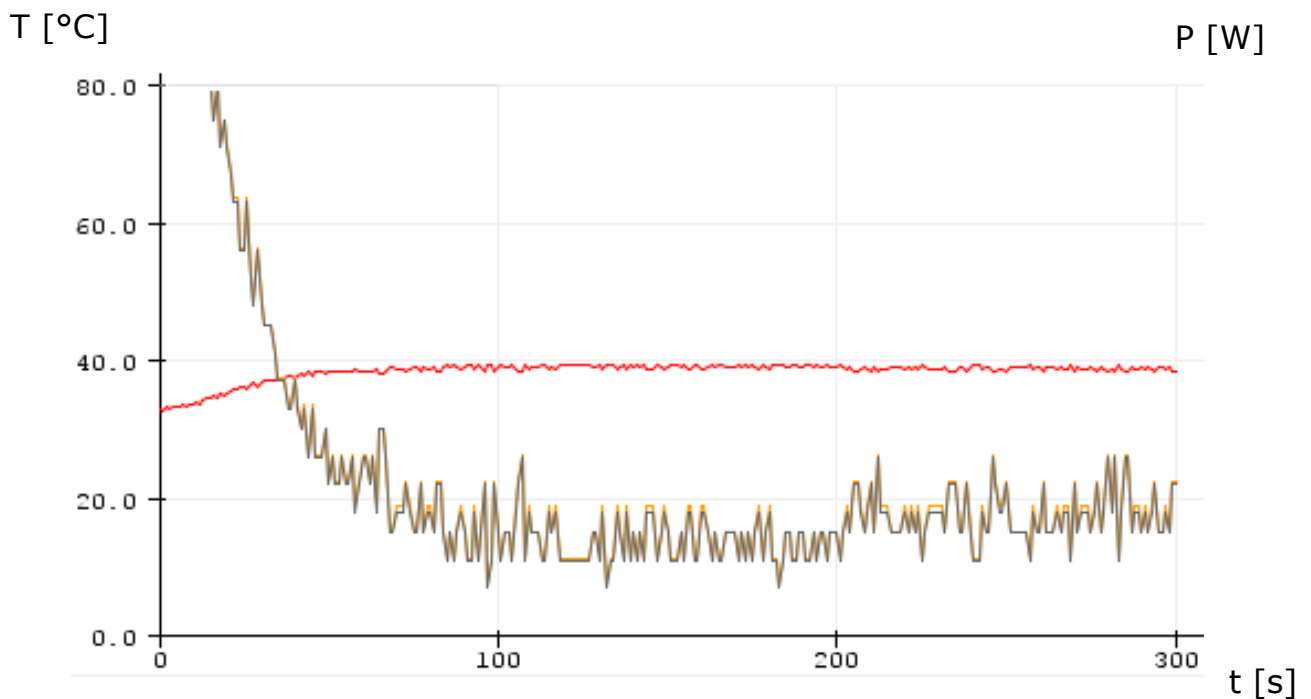


Figura 4.7: Sistema di controllo P

Quello che si ottiene è un sistema di controllo che impiega circa 200 secondi per arrivare al setpoint e, come si vede in figura 4.7, la temperatura letta dalla termocoppia varia tra 41.5 °C e 38.5 °C.

### 4.2.3 Sistema di controllo PI

Ora vediamo cosa succede quando si aggiunge il termine integrale, così da ottenere un sistema di controllo PI. Tramite il valore di  $\overline{K_p} = 35$  e  $\overline{T} = 10$  e utilizzando la tabella 8 si trovano i valori necessari in sistema di controllo PI.

	Kp	Ti	Td
Controllore P	15.75	8	0

Tabella 8: Controllore PI

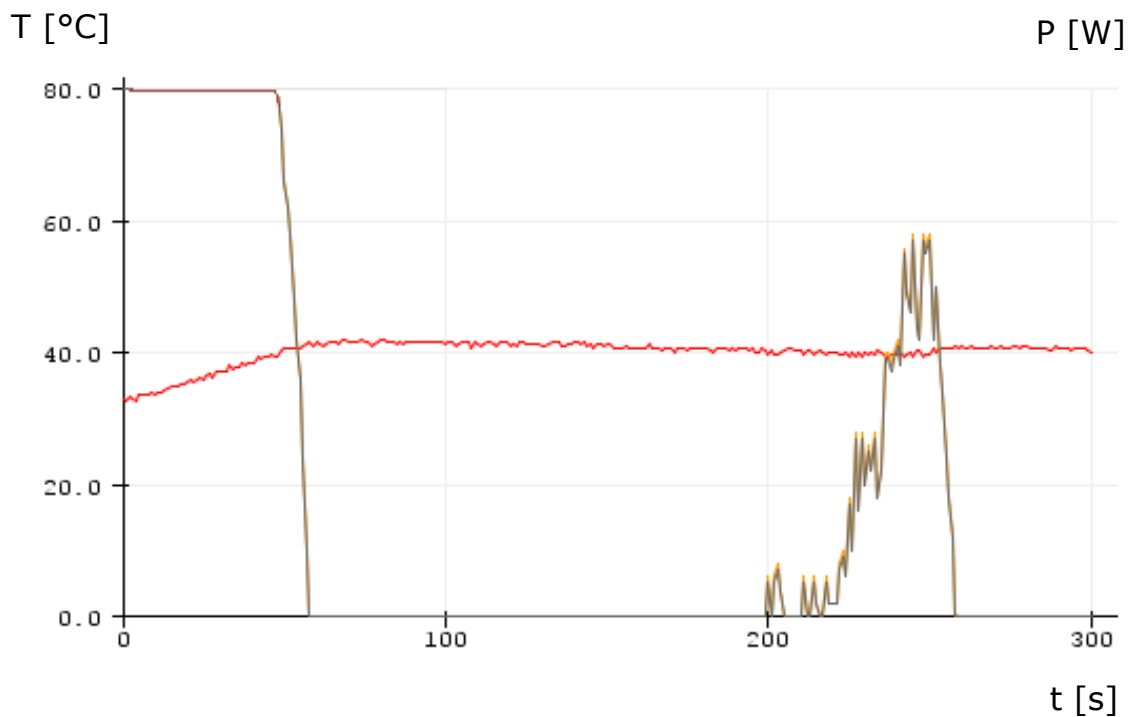


Figura 4.8: Sistema di controllo PI

Quello che si osserva fin da subito in figura 4.8 è la presenza di un overshoot che porta il sistema ad una temperatura di circa 42 ° C e che il sistema è ancora leggermente instabile.

In questo caso, superare il valore di setpoint non porta ad avere problemi, tuttavia, come si è visto nei capitoli precedenti in altri sistemi avere un overshoot può causare dei problemi al sistema in esame. Si nota inoltre che il sistema impiega meno tempo a raggiungere il setpoint rispetto al controllore P.

#### 4.2.4 Sistema di controllo PID

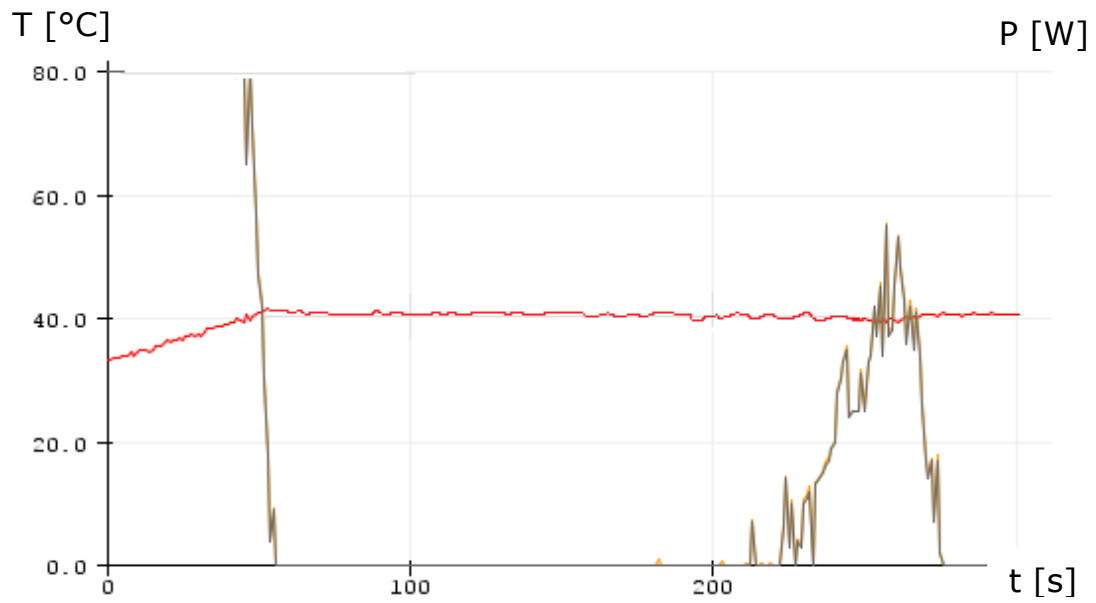
Infine, sempre tramite il metodo di Ziegler-Nichols andiamo a calcolare i valori necessari per il sistema di controllo PID della temperatura aggiungendo l'ultimo termine ovvero quello derivativo. Si utilizzano i valori trovati precedentemente, ovvero:

- $\overline{K_p} = 35$
- $\overline{T} = 10$

	Kp	Ti	Td
Controllore PID	21	5	1.25

*Tabella 9: Sistema di controllo PID*

Si ottiene un controllo della temperatura migliore rispetto ai precedenti come si vede nella figura 4.10.



*Figura 4.9: Sistema di controllo PID*

Tramite la figura 4.10 si nota che l'overshoot è minimo e il sistema è più stabile rispetto ai precedenti.



## 5. Conclusioni

L'obiettivo della tesi è la costruzione di un banco per la misura ed il controllo della temperatura a scopo didattico. Ho riscontrato che l'utilizzo del primo setup con il relay non garantisce un controllo accurato della temperatura, data l'impossibilità nel modulare la potenza della lampada, che in questo caso è solamente in grado di passare da un valore del 0 % a 100% di potenza .

Successivamente, sostituendo il relay con il dimmer nel sistema ed utilizzando quindi il sistema di controllo PID, è emerso un netto miglioramento nel controllo della temperatura. Lo si vede fin da subito nel sistema di controllo P, per poi avere un comportamento ancora più soddisfacente nel sistema di controllo PID.

In questa tesi è stato utilizzato il metodo in anello chiuso di Ziegler-Nichols che porta il sistema al limite dell'instabilità. La taratura del sistema di controllo PID con questo metodo si è rivelata efficiente, permettendo di mantenere all'interno del box una temperatura di 40°C, valore di setpoint impostato nel programma Arduino.

Complessivamente il sistema di controllo mostra un comportamento più che soddisfacente, e necessita solo di correzioni minori per raggiungere un setup migliore.

## Bibliografia

- [1] T. Häggglund e K. J. Åström, «Revisiting the Ziegler-Nichols Tuning Rules for Pi Control — Part I: the Frequency Response Method», *Asian Journal of Control*, vol. 6, n. 4, pagg. 469–482, 2004.
- [2] H. Stephen, «A Review of Relay Auto-tuning Methods for the Tuning of PID-type Controllers», *LATL Reinvention: an International Journal of Undergraduate Research*.
- [3] «Arduino Mega 2560», *Arduino Official Store*.  
<http://store.arduino.cc/products/arduino-mega-2560-rev3>
- [4] «Arduino, cos'è e perché rappresenta il futuro».  
<https://www.pirelli.com/global/it-it/life/arduino-cos-e-e-perche-rappresenta-il-futuro>
- [5] «MAX6675-Datasheet.pdf». Consultato: 30 maggio 2022. [Online].  
Disponibile su:  
<https://www.futurashop.it/image/catalog/data/Download/MAX6675-Datasheet.pdf>
- [6] «Relay-Module-Datasheet.pdf». Consultato: 5 giugno 2022. [Online].  
Disponibile su:  
[https://components101.com/sites/default/files/component\\_datasheet/Relay-Module-Datasheet.pdf](https://components101.com/sites/default/files/component_datasheet/Relay-Module-Datasheet.pdf)
- [7] «Arduino Reference - Arduino Reference».  
<https://www.arduino.cc/reference/it/>
- [8] *RobotDynOfficial/RBDDimmer*. RobotDyn, 2022. Consultato: 11 luglio 2022.  
[Online]. Disponibile su:  
<https://github.com/RobotDynOfficial/RBDDimmer>