

# Feedback loop con un PID

Mattia Ragaglia<sup>1</sup> and Alessandro Serafini<sup>2</sup>

Dipartimento di Fisica, Sapienza Università di Roma, Piazzale Aldo Moro, 5, 00185, Roma, Italia  
e-mail: serafini.1916973@studenti.uniroma1.it  
e-mail: ragaglia.1920104@studenti.uniroma1.it



December 9, 2021

## ABSTRACT

**Aims.** Realizzazione di un feedback loop

**Methods.** Regolando l'accensione di un LED tramite scheda Arduino, in base al segnale letto da un fotodiodo

**Results.** Stima dei parametri  $K_P$  e  $K_I$  per un feedback loop ottimale

## 1. Apparato Sperimentale

### 1.1. Strumentazione

- Oscilloscopio
- Scheda Arduino con cavo USB
- Basetta elettronica
- LED
- Fotodiodo
- Cavi BNC
- Generatore power supply

### 1.2. Introduzione

Tramite la basetta elettronica, il LED e il fotodiodo, alla quale vengono connessi con cavi BNC, sono collegati rispettivamente al pin di uscita digitale N=5 e al pin di ingresso analogico A0 della scheda Arduino, opportunamente messa alla stessa massa del sistema.

La porta in uscita della scheda invierà in questo modo un segnale onda quadra a 1kHz a duty cycle variabile. A seconda del valore assegnato a quest'ultimo il fotodiodo invierà un segnale diverso alla porta in entrata al quale è connesso.

Tramite l'analisi del segnale programmando la scheda è possibile costruire un semplice feedback loop che porti il sistema a misurare un valore scelto con il fotodiodo.

Mediando su vari periodi, infatti, il segnale inviato al led, così come quello dovuto al fotodiodo, è proporzionale al duty cycle impostato: variando opportunamente questo valore si può ottenere un arbitrario valore desiderato nella porta di entrata della scheda.

### 1.3. Codice per il PID

Al fine di creare un efficace feedback loop, che sia in grado di portare il sistema nella configurazione desiderata in modo rapido e preciso, si programma la scheda Arduino opportunamente:

- Viene stabilito un valore (*set point*) a circa metà del range di valori rivelabili dal fotodiodo
- Si calcola l'*errore*, la differenza tra il valore rilevato in entrata e il *set point*
- Il segnale in uscita viene modificato variando il valore assegnato al duty cycle a seconda dell'entità della *correzione* definita di seguito

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(t') dt' \quad (1)$$

dove  $K_P$  e  $K_I$  sono coefficienti costanti definiti di seguito.

```
int led=5;
int fotodiodo=A0;
int valorefoto=0;
int duty=0;
int prop;
int i;
long sum;
float valoremedio;
int dil=100;
float setpoint=1.5;
float error;
float errorint=0;
int delta=0;
int kp=104;
float ki=0.67;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(fotodiodo, INPUT);
  Serial.begin(9600);
}

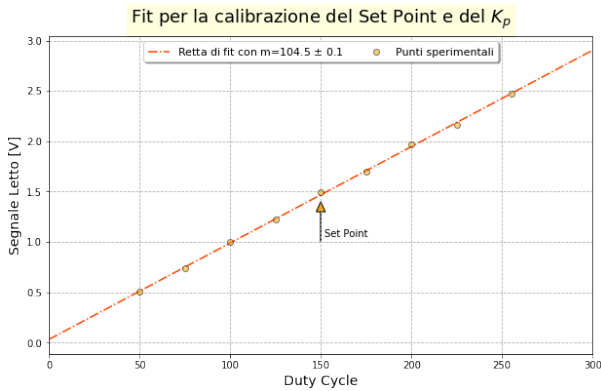
void loop() {
  duty=duty+delta;
  analogWrite(led, duty);
  sum=0;
  for (i=0; i<200; i++) {
    valorefoto=analogRead(fotodiodo);
    sum= sum + valorefoto;
  }
  valoremedio= sum*5./ (1024.*200.);
  error=-(valoremedio-setpoint);
  errorint=errorint + error;
  delta=kp*error +ki*errorint;

  Serial.print(setpoint);
  Serial.print(" ");
  Serial.println(valoremedio);
  delay(dil);
}
```

## 2. Calibrazione e stima $K_P$

Adattando la distanza tra LED e fotodiodo in modo che il segnale massimo letto non superi la dinamica dell'ADC, si effettuano

varie misure del segnale in entrata al variare del duty cycle impostato in modo da costruire una retta di calibrazione con un fit lineare.

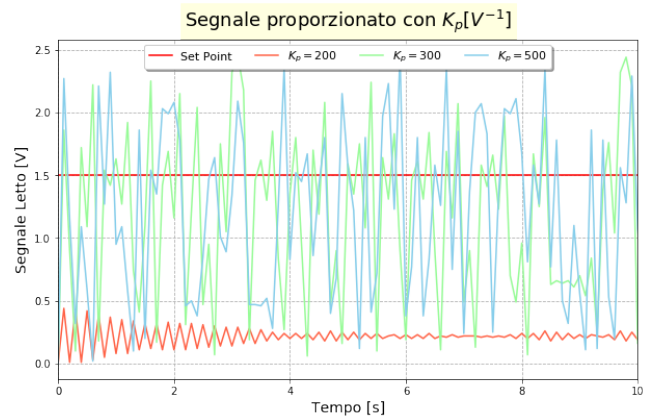
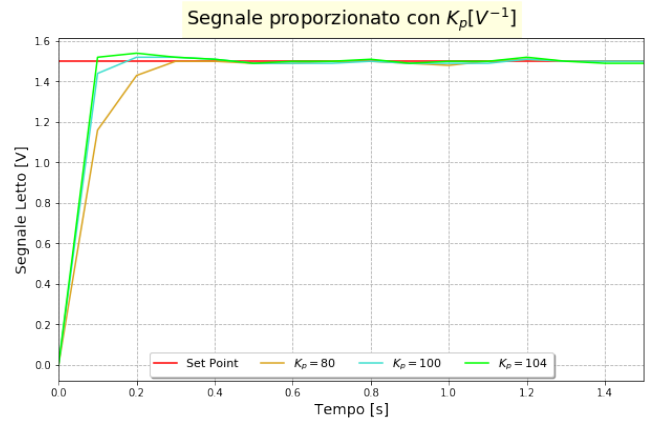
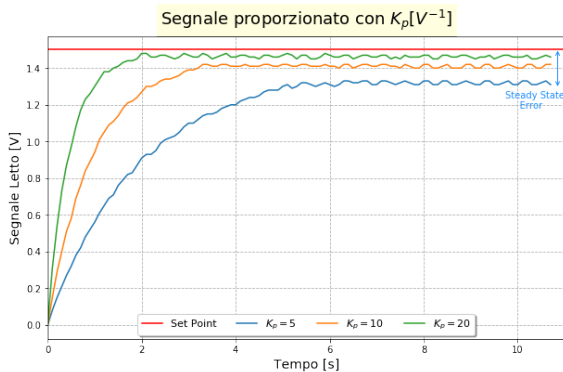


Segnale in Entrata[V]	Duty
$0.50 \pm 0.02$	50
$0.73 \pm 0.03$	75
$1.00 \pm 0.03$	100
$1.23 \pm 0.02$	125
$1.49 \pm 0.03$	150
$1.70 \pm 0.02$	175
$1.97 \pm 0.04$	200
$2.16 \pm 0.02$	225
$2.48 \pm 0.03$	255

**Table 1.** Il segnale in entrata e la sua incertezza sono ottenuti mediando su più valori

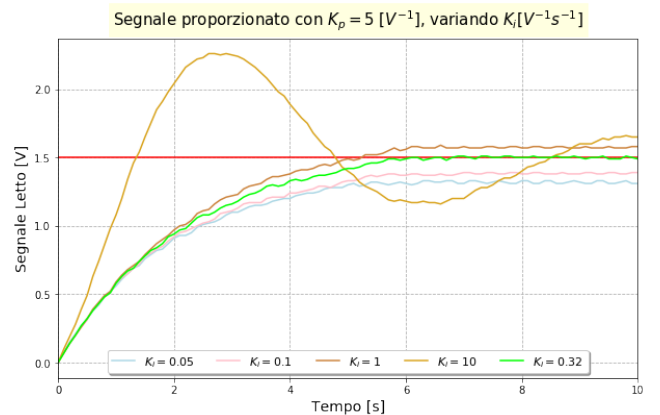
L'inverso del coefficiente angolare della retta risultante dal fit coincide con il valore ideale del coefficiente che risulta essere  $K_P = (104.5 \pm 0.1)V^{-1}$

Partendo da un  $K_P$  molto piccolo e incrementando il suo valore si ottiene infatti che il segnale raggiunge in modo accurato il valore del *set point* solo a valori prossimi a quello sopra specificato: per i valori minori si può notare il fenomeno dello *steady state error*, dovuto alla quantizzazione della DAC, che porta il segnale a non raggiungere mai quello desiderato, mentre per valori troppo alti si rivelano oscillazioni molto grandi, indice della non stabilità del sistema. Nei grafici seguenti si consideri che il sistema varia con un *delay* di 100ms e con lo stesso ritardo vengono fatte le misure.



### 3. Valore del coefficiente $K_I$

Per stimare il valore di  $K_I$  si registrano gli andamenti del sistema utilizzando valori diversi con  $K_P$  fissato, e si sceglie il valore per il quale la stabilizzazione avviene rapidamente e il più possibile vicino al valore del *set point*. Nel grafico seguente si riportano gli andamenti del segnale in entrata a per diversi valori di  $K_I$  e con  $K_P = 5$ .

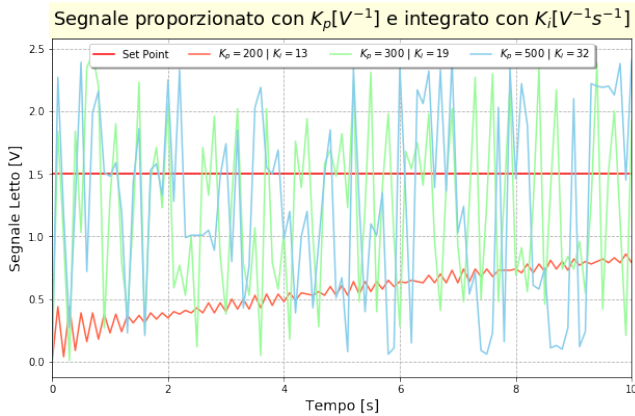
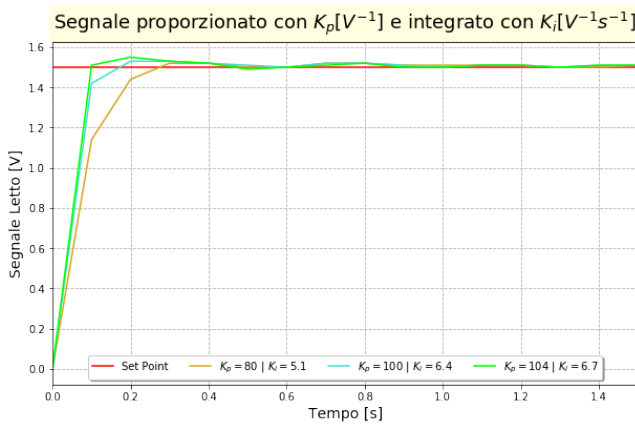
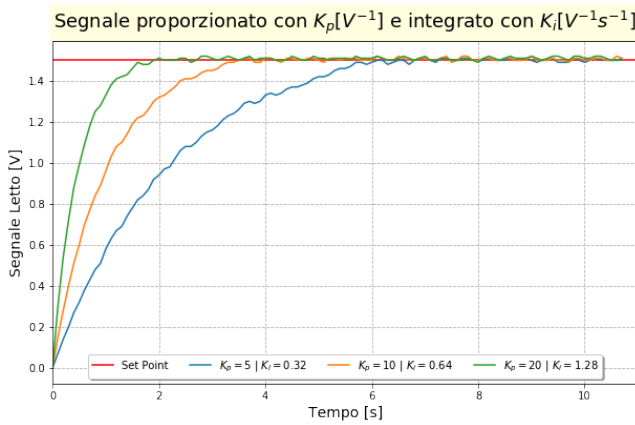


Sapendo che vale la seguente relazione:

$$K_I = \frac{K_P}{\tau} \quad (2)$$

si può stimare il valore di  $K_i$  scegliendo un valore per  $\tau$ , tempo di convergenza del sistema. Scegliendo il valore di  $\tau = 15.6s$ , stabilito a partire dal valore qualitativamente ottimale di  $K_I = 0.32(s \cdot V)^{-1}$  per  $K_P = 5V^{-1}$ , si ripetono le misurazioni ai valori di  $K_P$  già analizzati senza il contributo integrale, variando il

$K_I$  nell'intorno di quello stabilito con il fissato valore di  $\tau$ . Si ottengono i seguenti grafici:



#### 4. Conclusioni

Analizzando i grafici sopra esposti risultano evidenti alcuni risultati importanti:

- Per valori di  $K_P$  molto piccoli si verifica il problema dello *Steady State Error* dovuto alla quantizzazione della DAC della scheda; ciò si risolve evidentemente con l'aggiunta del termine integrale che riesce a far convergere il segnale al *set point* in un tempo diverso a seconda del  $K_P$

- Per valori di  $K_P \approx 100$ , vicini al valore ottenuto dalla calibrazione, si nota che il segnale converge rapidamente al *set point* e che il contributo integrale è pressoché ininfluente.
- Per valori di  $K_P$  molto grandi il sistema risulta sempre più instabile all'aumentare del coefficiente: non abbiamo più convergenza, neanche con l'aggiunta del  $K_I$ . Il segnale anzi oscilla attorno al *Set Point* a causa del fatto che si va a sommare un apporto proporzionale troppo elevato rispetto al range duty della scheda Arduino.

*Acknowledgements.* This work is supported by the Physics department of Sapienza, University of Rome