UNIVERSITÀ
DEGLI STUDI
FIRENZE

Lego EV3 Segway

Stabilizzazione in C

Laboratorio di controlli automatici

Bernardo Tiezzi, Edoardo Re

UNIVERSITÀ DEGLI STUDI FIRENZE

Overview

- Introduzione al problema
- Configurazione EV3 Segway
- Sistema di riferimento
- Angoli importanti
- Schema Simulink
- Relazioni tra gli angoli
- Legge di controllo ottima
- RobotC
- Funzioni di libreria
- Implementazione in C
- Valori osservati
- Esecuzione del processo
- Link utili e bibliografia

UNIVERSITÀ DEGLI STUDI FIRENZE

Introduzione al problema

La configurazione del LEGO EV3 Segway ha due posizioni di equilibrio:

1. Orizzontale appoggiato su un piano
2. Verticale -> Equilibrio instabile

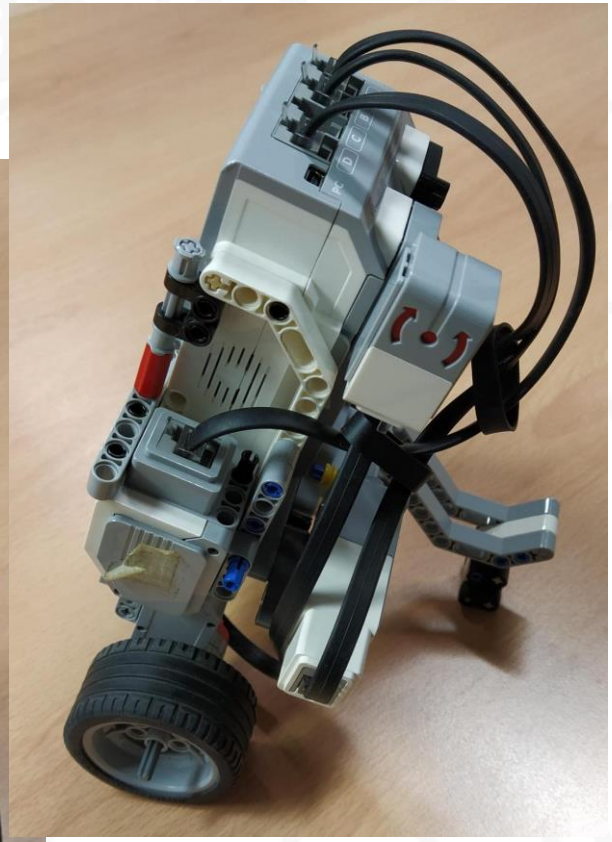
L'intento di questo progetto è di stabilizzare il robot sulla verticale stimando lo stato e tramite una legge di controllo in retroazione ad anello chiuso tenerlo in equilibrio.

Il modello Simulink è stato riadattato e modificato per poter essere implementato in un linguaggio di programmazione, si è scelto in sequenza:

1. Python con libreria EV3Dev2
2. Java con LeJOS
3. **C con ambiente di sviluppo RobotC**

Configurazione EV3 Segway

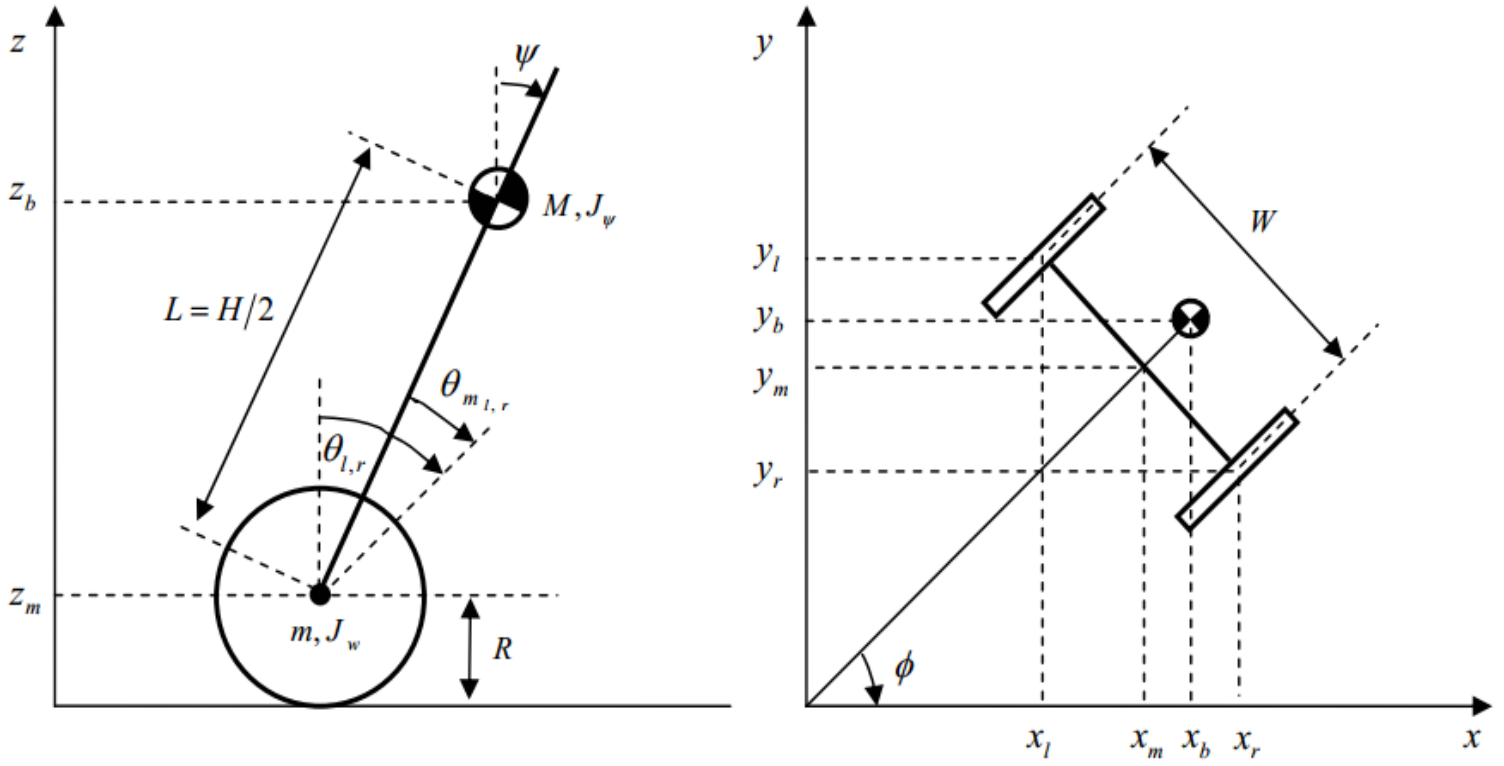
Porta A -> Motore DX
Porta B -> Motore del cavalletto
Porta D -> Motore SX



Porta S4 -> Gyro sensor

Sistema di riferimento

Si considera come posizione di equilibrio Psi = 0
ovvero quando il centro di massa del robot è posizionato sulla verticale.



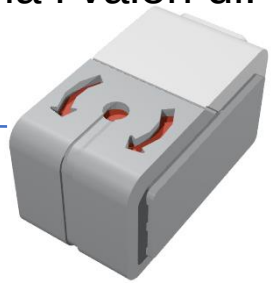
Sistema di riferimento adottato nel progetto

Angoli importanti

Definito il sistema di riferimento è necessario individuare uno stato del robot tramite misurazioni di angoli da parte dei sensori.
In particolare risultano essenziali alla legge di controllo ottima i valori di:

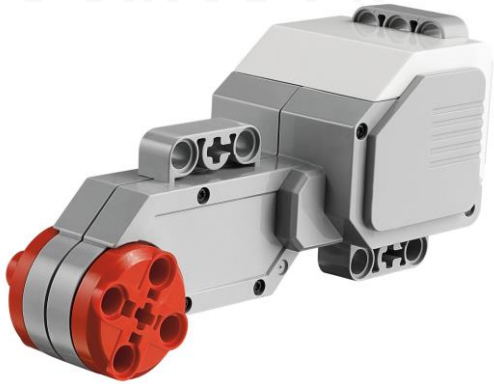
Psi_dot
Psi

Gyro Sensor



Theta
Theta_dot
Theta_int

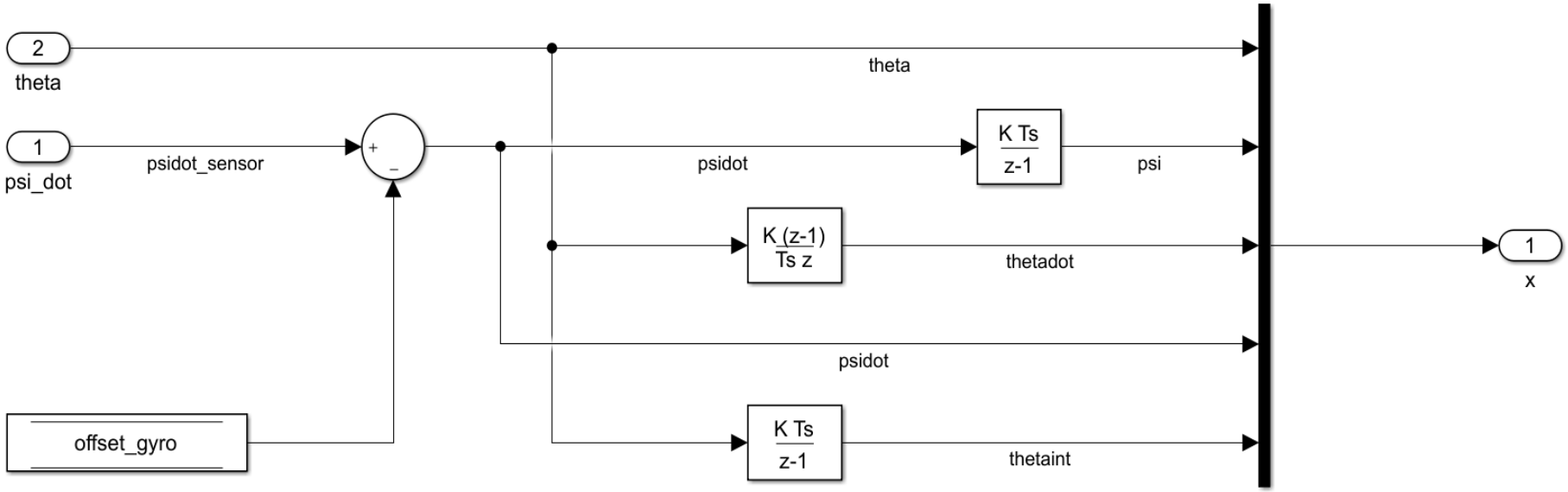
Motor Encoders



UNIVERSITÀ DEGLI STUDI FIRENZE

Schema Simulink

Per ottenere la stima dello stato si prelevano solamente i valori di Theta: dagli encoders e Psi_dot: dal Gyro sensor. I valori in gradi dovranno successivamente essere convertiti in radianti per la legge di controllo ottima che permette di calcolare la velocità da dare ai motori.



UNIVERSITÀ DEGLI STUDI FIRENZE

Relazioni tra gli angoli

1. **Psi_dot** è la velocità angolare in deg/s misurata direttamente dal gyro sensor
2. **Psi** è l'integrale discreto di Psi_dot in dt con Ts=0.01s
3. **Theta** si ottiene direttamente eseguendo una media tra i due valori dati dagli encoders dei motori e sommando l'angolo Psi ovvero: ½(ThetaR+ThetaL) + Psi
4. **Theta_dot** si ottiene derivando Theta in dt
5. **Theta_int** si ottiene integrando Theta in dt

UNIVERSITÀ DEGLI STUDI FIRENZE

Legge di controllo ottima

Si è implementata una legge di controllo in retroazione ad anello chiuso del tipo:
u = -Kx dove K è un vettore di guadagni di dimensione cinque e x è il vettore di stato.

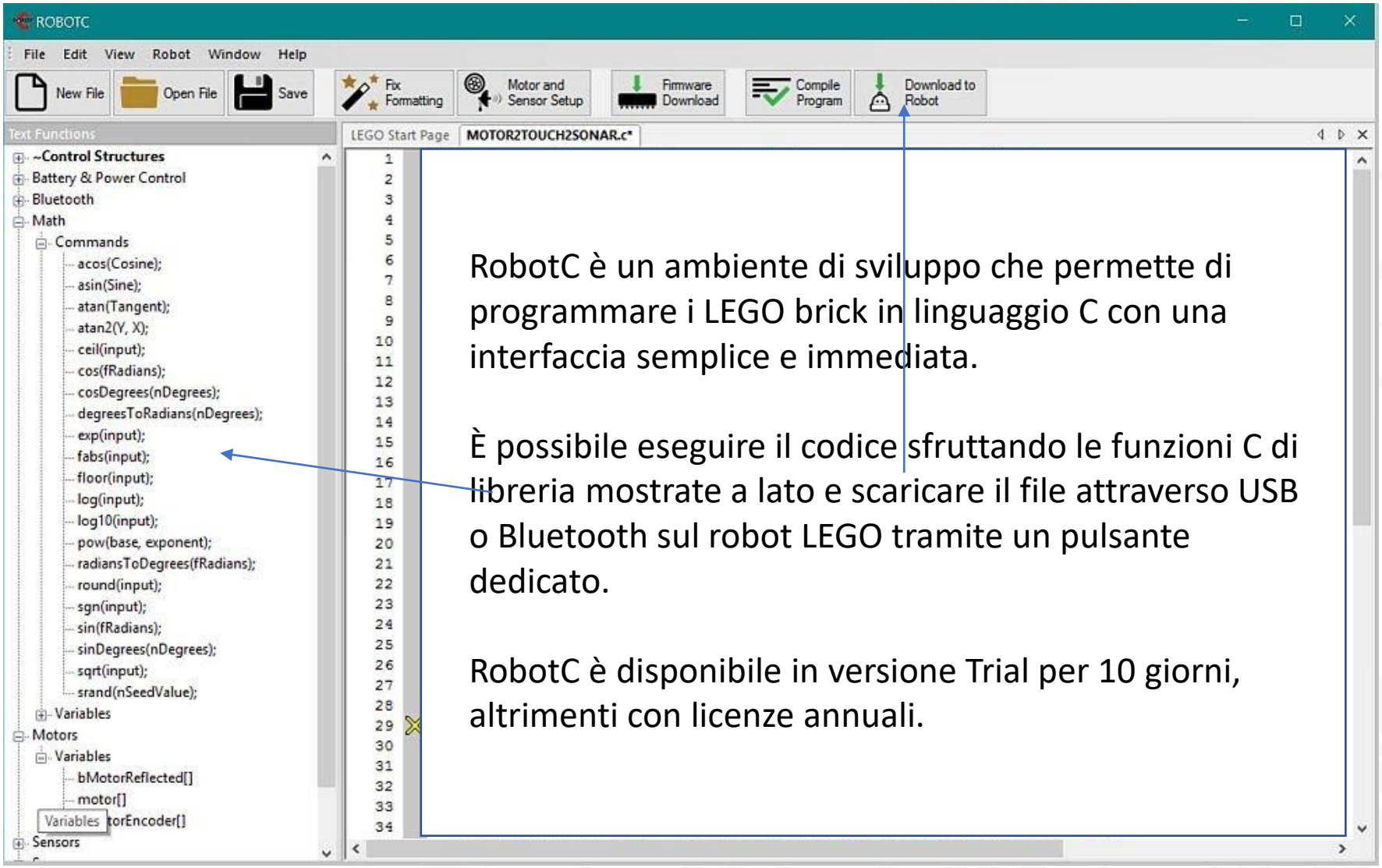
K = (-0,8559 -44,7896 -0,9936 -4.6061 -0.5000)
x' = (Theta Psi Theta_dot Psi_dot Theta_int)

Eseguendo l'aggiornamento dello stato grazie ai sensori è possibile ricavare lo scalare u che indica in un range di [-100, +100] il valore in pwm da dare ai motori.

I valori di K sono stati ricavati minimizzando la funzione di costo: [1] (Q e P matrici di pesi)

$$J = \int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Pu(t)\right)\, dt$$

RobotC



RobotC è un ambiente di sviluppo che permette di programmare i LEGO brick in linguaggio C con una interfaccia semplice e immediata.

È possibile eseguire il codice sfruttando le funzioni C di libreria mostrate a lato e scaricare il file attraverso USB o Bluetooth sul robot LEGO tramite un pulsante dedicato.

RobotC è disponibile in versione Trial per 10 giorni, altrimenti con licenze annuali.

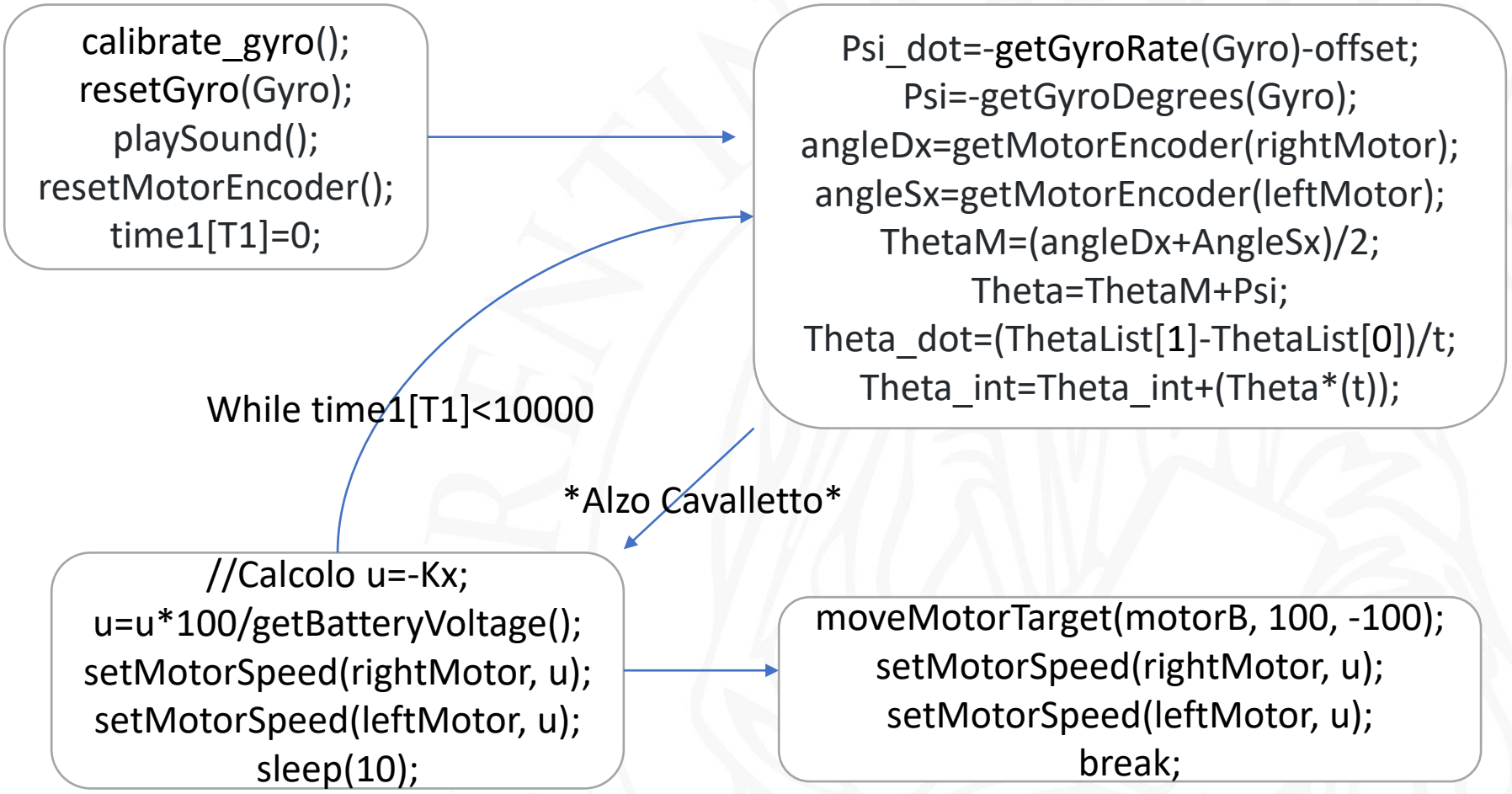
UNIVERSITÀ DEGLI STUDI FIRENZE

Funzioni di Libreria

- playSound();
- eraseDisplay();
- displayCenteredBigTextLine(size, String);
- restetGyro(Gyro);
- resetMotorEncoder(Motor);
- Time1[T1];
- getGyroRate(Gyro);
- getGyroDegrees(Gyro);
- getMotorEncoder(Motor);
- getBatteryVoltage();
- setMotorSpeed(Motor, speed);
- moveMotorTarget(Motor, TargetAngle, speed);

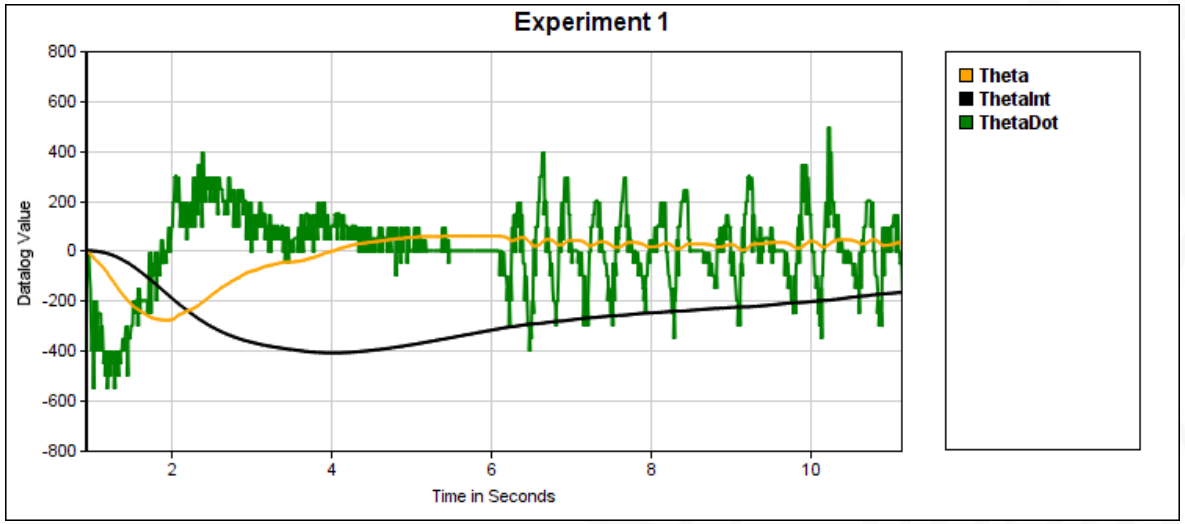
UNIVERSITÀ DEGLI STUDI FIRENZE

Implementazione in C



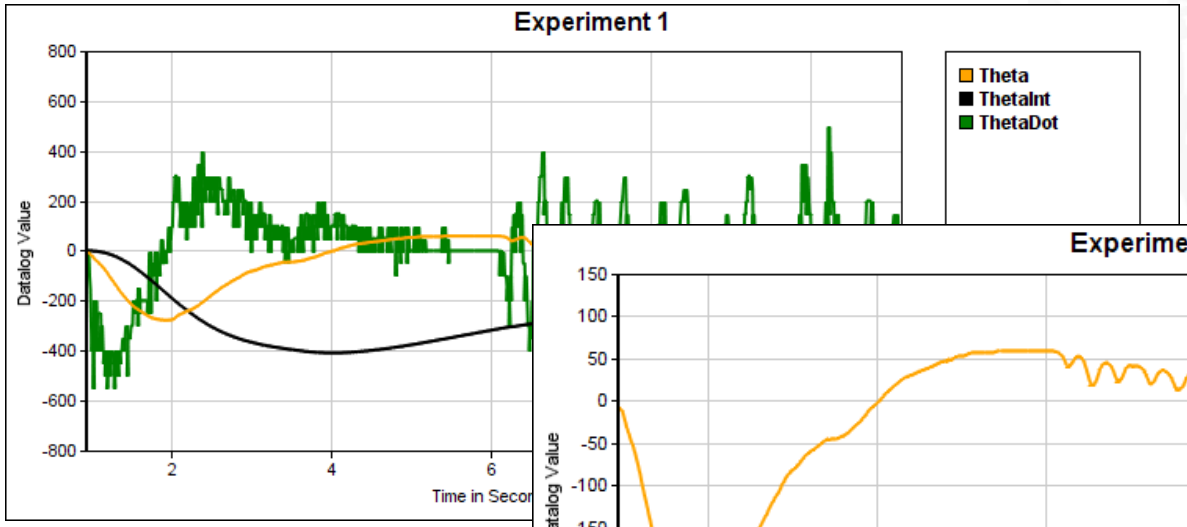
UNIVERSITÀ DEGLI STUDI FIRENZE

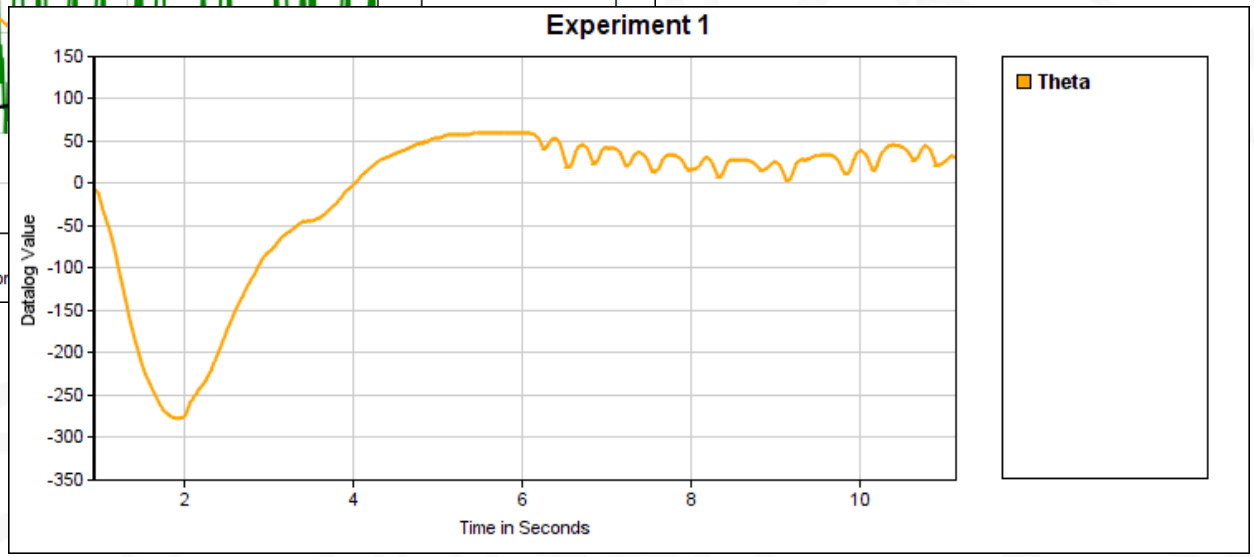
Valori osservati:



UNIVERSITÀ DEGLI STUDI FIRENZE

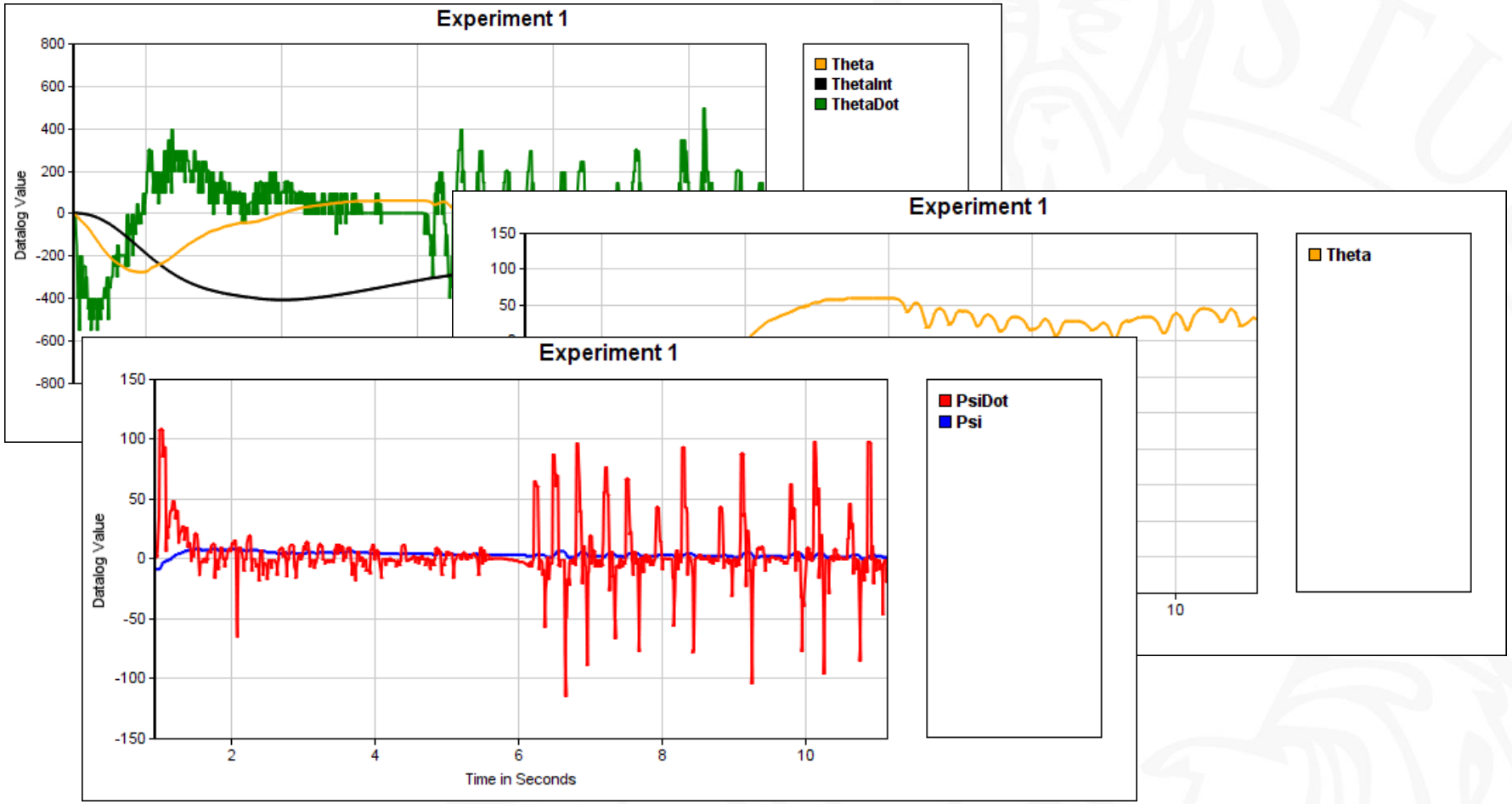
Valori osservati:





UNIVERSITÀ DEGLI STUDI FIRENZE

Valori osservati:



UNIVERSITÀ DEGLI STUDI FIRENZE

Esecuzione del processo

UNIVERSITÀ DEGLI STUDI FIRENZE

Link utili e Bibliografia

Codice Sorgente: https://github.com/edoardore/LegoEv3SegwayC
Video Esecuzione: https://www.youtube.com/watch?v=ddvBCt-Wbh0

Codice GitHub

Video YouTube

[1] Davide Martini, Sergio Carleo - «Control of a LEGO Mindstorms EV/3 two-wheeled robot»

[2] Yorihisa Yamamoto. Nxtway-gs (self-balancing two-wheeled robot) controller design

UNIVERSITÀ DEGLI STUDI FIRENZE

Grazie per l’attenzione!

Bernardo Tiezzi, Edoardo Re