



# **VAB - Veicolo auto bilanciato**

## ***Relazione di progetto***

Laboratorio di Sistemi Meccatronici II  
Università degli Studi di Bergamo

*Kilometro rosso*

A.A. 2019/2020

CALEGARI ANDREA - 1041183  
PIFFARI MICHELE - 1040658

May 15, 2020



# Contents

|           |   |           |
|-----------|---|-----------|
| <b>1</b>  | <b>Introduzione</b>   | <b>1</b>  |
| <b>I</b>  | <b>Dinamica</b>   | <b>3</b>  |
| <b>2</b>  | <b>Scomposizione del VAB</b>  | <b>5</b>  |
| 2.1       | Considerazioni iniziali . . . . .   | 5         |
| 2.2       | Grandezze di supporto . . . . .   | 5         |
| 2.3       | Calcolo componenti dinamiche e potenziali per ogni corpo rigido del sistema . . . . . | 6         |
| 2.3.1     | Asta . . . . .  | 7         |
| 2.3.2     | Chassis (base) . . . . .  | 7         |
| 2.3.3     | Utente . . . . .  | 8         |
| 2.3.4     | Ruota . . . . .   | 8         |
| 2.3.5     | Veicolo completo . . . . .  | 8         |
| 2.3.6     | Note sul calcolo delle componenti dinamiche . . . . .                                 | 8         |
| <b>3</b>  | <b>Equazioni del moto</b>   | <b>11</b> |
| <b>II</b> | <b>Controllo</b>  | <b>13</b> |
| <b>4</b>  | <b>OPC - UA</b>   | <b>15</b> |
| 4.1       | Idea base OPC-UA . . . . .  | 15        |
| 4.2       | OPC-UA e V.A.B. . . . .   | 16        |
| <b>5</b>  | <b>Rumore</b>   | <b>19</b> |



# List of Figures

|     |   |    |
|-----|---|----|
| 2.1 | Baricentri dei singoli corpi rigidi . . . . .                               | 6  |
| 2.2 | Lunghezze di supporto . . . . .   | 6  |
| 2.3 | Ragionamento per il calcolo dei contributi dinamici dello chassis . . . . . | 9  |
| 4.1 | Schema di massima dell'utilizzo di Raspberry Pi 3 . . . . .                 | 15 |
| 4.2 | Diagramma rappresentante le funzionalità del server . . . . .               | 17 |



# 1

## Introduzione

L'approccio seguito per la stesura del modello dinamico del veicolo autobilanciato ha da subito preso una via meno *tradizionale* rispetto al classico metodo risolutivo: abbiamo infatti preferito, data il nostro *background* informatico, approcciare il problema direttamente in ambiente Matlab, sfruttando sin da subito le potenzialità di calcolo offerte dal software di *Mathworks*.

Nello specifico, per la parte di stesura e definizione della dinamica, abbiamo inizialmente seguito una via risolutiva duale, portando avanti sia un'analisi letterale, sfruttando le potenzialità del **calcolo simbolico** messe a disposizione delle funzionalità di **live scripting**, sia uno studio numerico (considerando quindi le varie grandezze fisiche con i valori definiti delle specifiche di progetto).

In linea di massima lo sviluppo del progetto ha seguito un andamento a step gradualmente, cadenzati da incontri settimanali in cui poter confrontare e consolidare lo *stato di avanzamento dei lavori*: nello specifico, il lavoro ha seguito uno sviluppo in questa direzione step by step, rappresentabile in linea di massima da queste *pietre miliari*:

- **Dinamica di ogni singolo corpo rigido**: abbiamo impostato il problema della dinamica andando a considerare il veicolo auto bilanciato come un insieme di corpi rigidi di cui poterne studiare la dinamica in maniera separata;
- **Dinamica completa del VAB**: siamo andati poi a considerare il sistema nella sua completezza, andando ad unire i contributi dei corpi rigidi considerati in prima battuta singolarmente.  
Questo ci ha permesso di ottenere le equazioni del moto, in forma non lineare, le quali hanno permesso una completa descrizione del sistema che abbiamo modellizzato;
- **Linearizzazione**: siamo poi andati a linearizzare queste equazioni dinamiche (appunto non lineari) nell'intorno dell'equilibrio;
- **Definizione del controllo**: definizione del controllore tramite tecnica di *pole placement*;
- **Modellizzazione del motore**: introduzione del modello del motore sulla base delle caratteristiche contenute nella specifica di progetto;
- **Discretizzazione**: passaggio a tempo discreto per i segnali derivanti dal mondo analogico, ovvero per tutto ciò che concerne la parte di sensoristica;
- **Non idealità**: siamo andati a modellizzare anche la presenza di disturbi, di natura stocastica e dovuti alla quantizzazione;
- **Trasformazioni di blocchi in *interpreted function***: conversione delle funzionalità di controllo del motore e filtraggio dei segnali provenienti dai sensori in blocchi di codice *Matlab* per favorirne poi la successiva conversione ed implementazione a bordo del controllore;

TODO: note varie





Part I

**Dinamica**



## 2

# Scomposizione del VAB

## 2.1 Considerazioni iniziali

Per il calcolo delle equazioni dinamiche del sistema siamo andati a considerare ogni singolo corpo rigido componente il sistema, calcolandone le grandezze fisiche di posizione e velocità, seguendo un approccio cartesiano. Nello specifico abbiamo considerato il sistema composto da:

- Asta
- Utente a bordo dello chassis
- Chassis (nel corso della trattazione sarà chiamata talvolta anche base)
- Ruota (che poi sarà considerata con un contributo, essendo il VAB composto da due ruote)

Ognuno di questi corpi rigidi separati è individuato da un punto, che ne rappresenta il centro di massa (o baricentro del corpo stesso): avremo quindi questo insieme di punti caratterizzanti il sistema (figura 2.1)

- $P_a$
- $P_b$
- $P_c$
- $P_r$

## 2.2 Grandezze di supporto

Prima di andare a definire le componenti di energia potenziale e cinetica di ogni singolo corpo, siamo andati ad introdurre alcune grandezze geometriche di supporto che definiremo qui di seguito.

Nello specifico abbiamo introdotto i seguenti parametri, specificati anche in figura 2.2:

- $l_a$ : rappresenta la congiungente tra il centro del sistema di riferimento e il centro dell'asta, utilizzata appunto come manubrio, che abbiamo individuato come

$$\sqrt{\left(\frac{h_a}{2} + \frac{h_b}{2}\right)^2 + \left(\frac{w_b}{2}\right)^2}$$

- $l_c$ : (TODO: check) questa grandezza invece rappresenta per noi l'altezza del baricentro del corpo dell'utente, la quale ovviamente andrà a dipendere dal valore di inclinazione del corpo stesso. Considerando il corpo inizialmente in posizione verticale, avremo che questa grandezza corrisponde alla congiungente dal centro del sistema di riferimento al punto  $P_c$ , che equivale a dire che

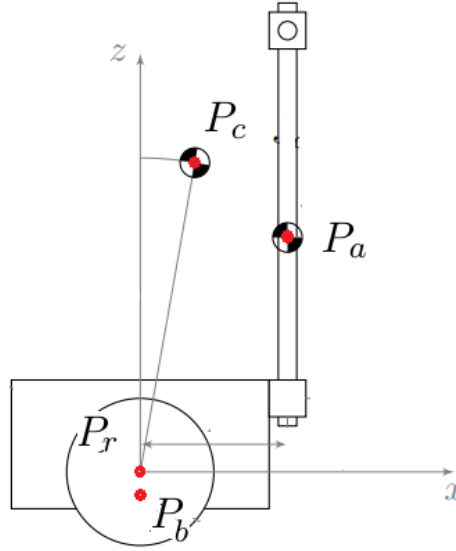


Figure 2.1: Baricentri dei singoli corpi rigidi

$$l_c = 0.55 \bullet h_c + \frac{h_b}{2}$$

- $l_b$ : spostamento verso il basso, lungo l'asse  $z$ , del baricentro dello chassis. Da specifiche del progetto sappiamo che questa grandezza ha valore (con segno negativo) di:

$$l_b = 0.1m$$

- $\beta$ : angolo formato con la verticale dalla congiungente tra il centro del sistema di riferimento e il punto  $P_a$ . Si ricava, con un semplice approccio trigonometrico, che l'angolo in questione ha questa forma

$$\arctan\left(\frac{\frac{w_b}{2}}{\frac{h_a}{2} + \frac{h_b}{2}}\right)$$

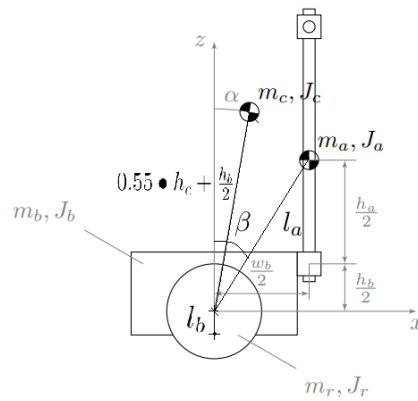


Figure 2.2: Lunghezze di supporto

## 2.3 Calcolo componenti dinamiche e potenziali per ogni corpo rigido del sistema

Per ognuno dei corpi rigidi definiti in precedenza siamo andati appunto a calcolare:

- **Coordinate spaziali  $\mathbf{P}$**  espresse nel sistema di riferimento XZ. A queste due coordinate cartesiane ne va aggiunta una terza, relativa alle coordinate angolari (per poter tener così conto dei contributi inerziali);
- **Vettore delle velocità  $\mathbf{V}$**   $\rightarrow$  vettore  $3 \times 1$
- **Matrice delle masse  $\mathbf{M}$**   $\rightarrow$  matrice  $3 \times 3$
- **Energia cinetica  $\mathbf{T}$**   $\rightarrow \frac{1}{2} \bullet \mathbf{V}^T \bullet \mathbf{M} \bullet \mathbf{V}$
- **Energia potenziale  $\mathbf{U}$**
- **Lagrangiana *parziale*  $\mathbf{L}$**

### 2.3.1 Asta

- $$\mathbf{P}_a = \begin{pmatrix} r \phi(t) + l_a \sin(\beta + \theta(t)) \\ l_a \cos(\beta + \theta(t)) \\ \theta(t) \end{pmatrix}$$
- $$\mathbf{V}_a = \begin{pmatrix} r \dot{\phi}(t) + l_a \cos(\beta + \theta(t)) \dot{\theta}(t) \\ -l_a \sin(\beta + \theta(t)) \dot{\theta}(t) \\ \dot{\theta}(t) \end{pmatrix}$$
- $$\mathbf{M}_a = \begin{pmatrix} m_a & 0 & 0 \\ 0 & m_a & 0 \\ 0 & 0 & m_a l_a^2 + J_a \end{pmatrix}$$
- $$T_a = m_a l_a^2 (\dot{\theta}(t))^2 + \frac{m_a r^2 (\dot{\phi}(t))^2}{2} + \frac{J_a (\dot{\theta}(t))^2}{2} + m_a \cos(\beta + \theta(t)) l_a r \dot{\theta}(t) \dot{\phi}(t)$$
- $$U_a = g l_a m_a \cos(\beta + \theta(t))$$
- $$L_a = m_a l_a^2 (\dot{\theta}(t))^2 + \frac{m_a r^2 (\dot{\phi}(t))^2}{2} + \frac{J_a (\dot{\theta}(t))^2}{2} + m_a \cos(\beta + \theta(t)) l_a r \dot{\theta}(t) \dot{\phi}(t) - g m_a \cos(\beta + \theta(t)) l_a$$

### 2.3.2 Chassis (base)

- $$\mathbf{P}_b = \begin{pmatrix} r \phi(t) - l_b \sin(\theta(t)) \\ -l_b \cos(\theta(t)) \\ \theta(t) \end{pmatrix}$$
- $$\mathbf{V}_b = \begin{pmatrix} r \dot{\phi}(t) - l_b \cos(\theta(t)) \dot{\theta}(t) \\ l_b \sin(\theta(t)) \dot{\theta}(t) \\ \dot{\theta}(t) \end{pmatrix}$$
- $$\mathbf{M}_b = \begin{pmatrix} m_b & 0 & 0 \\ 0 & m_b & 0 \\ 0 & 0 & m_b l_b^2 + J_b \end{pmatrix}$$
- $$T_b = m_b l_b^2 (\dot{\theta}(t))^2 + \frac{m_b r^2 (\dot{\phi}(t))^2}{2} + \frac{J_b (\dot{\theta}(t))^2}{2} - m_b \cos(\theta(t)) l_b r \dot{\theta}(t) \dot{\phi}(t)$$
- $$U_b = -g l_b m_b \cos(\theta(t))$$
- $$L_b = m_b l_b^2 (\dot{\theta}(t))^2 + \frac{m_b r^2 (\dot{\phi}(t))^2}{2} + \frac{J_b (\dot{\theta}(t))^2}{2} - m_b \cos(\theta(t)) l_b r \dot{\theta}(t) \dot{\phi}(t) + g m_b \cos(\theta(t)) l_b$$

### 2.3.3 Utente

- $P_c = \begin{pmatrix} r \phi(t) + l_c \sin(\alpha + \theta(t)) \\ l_c \cos(\alpha + \theta(t)) \\ \alpha + \theta(t) \end{pmatrix}$
- $V_c = \begin{pmatrix} r \dot{\phi}(t) + l_c \cos(\alpha + \theta(t)) \dot{\theta}(t) \\ -l_c \sin(\alpha + \theta(t)) \dot{\theta}(t) \\ \dot{\theta}(t) \end{pmatrix}$
- $M_c = \begin{pmatrix} m_c & 0 & 0 \\ 0 & m_c & 0 \\ 0 & 0 & m_c l_c^2 + J_c \end{pmatrix}$
- $T_c = m_c l_c^2 (\dot{\theta}(t))^2 + \frac{m_c r^2 (\dot{\phi}(t))^2}{2} + \frac{J_c (\dot{\theta}(t))^2}{2} + m_c \cos(\alpha + \theta(t)) l_c r \dot{\theta}(t) \dot{\phi}(t)$
- $U_c = g l_c m_c \cos(\alpha + \theta(t))$
- $L_c = m_c l_c^2 (\dot{\theta}(t))^2 + \frac{m_c r^2 (\dot{\phi}(t))^2}{2} + \frac{J_c (\dot{\theta}(t))^2}{2} + m_c \cos(\alpha + \theta(t)) l_c r \dot{\theta}(t) \dot{\phi}(t) - g m_c \cos(\alpha + \theta(t)) l_c$

### 2.3.4 Ruota

- $P_r = \begin{pmatrix} r \phi(t) \\ 0 \\ \phi(t) \end{pmatrix}$
- $V_r = \begin{pmatrix} r \dot{\phi}(t) \\ 0 \\ \dot{\phi}(t) \end{pmatrix}$
- $M_r = \begin{pmatrix} m_r & 0 & 0 \\ 0 & m_r & 0 \\ 0 & 0 & J_r \end{pmatrix}$
- $T_r = \frac{(m_r r^2 + J_r) (\dot{\phi}(t))^2}{2}$
- $U_r = 0$
- $L_r = \frac{(m_r r^2 + J_r) (\dot{\phi}(t))^2}{2}$

### 2.3.5 Veicolo completo

Una volta trovati le componenti dinamiche dei singoli corpi rigidi, possiamo andare a definire l'energia cinetica e potenziale totale del sistema, per poter poi andare a calcolare l'equazione di Lagrange per l'intero sistema. In sostanza quindi avremo:

$$L = L_a + L_b + L_c + 2 \bullet L_r$$

### 2.3.6 Note sul calcolo delle componenti dinamiche

- Nel calcolo della matrice di massa abbiamo considerato tre componenti:
  - Componente di massa lungo x;
  - Componente di massa lungo z;

– Componente di massa rotazionale: dalla meccanica è noto che, un corpo con una certa massa che si trova in uno stato di rotazione, avrà un contributo inerziale che dipende dal braccio rispetto al quale avviene la rotazione. Nello specifico, per ogni singolo corpo rigido che è stato preso in esame, abbiamo considerato, secondo il teorema di *Huygens-Steiner*, il quale permette di definire l'inerzia di un corpo come la somma di due diverse componenti:

- \* Momento d'inerzia definito rispetto all'asse passante per il centro di massa: questo parametro rappresenta il valore che è fornito dalle specifiche del progetto;
- \* prodotto tra la massa  $m$  del corpo preso in considerazione e la distanza tra l'asse in esame e quello passante per il centro di massa;

Questa terza componente è visibile nelle matrici di massa in posizione (3,1): si sottolinea come invece, per la matrice 2.3.4, non sia presente la componente esplicitata dal teorema di *Huygens-Steiner* per il fatto che il centro di massa della ruota coincide con quello del sistema di riferimento XZ;

- Nel calcolo simbolico in ambiente *Matlab* siamo andati ad utilizzare le funzionalità di *collect* e *simplify*, per permettere di ridurre e semplificare le equazioni. Nello specifico, con il comando *simplify*, tramite l'opzione *Steps*, siamo andati a settare il numero di step che l'algoritmo di calcolo simbolico andrà a seguire per poter ridurre e semplificare il maggior numero di termini ( $\uparrow$  *Steps*,  $\uparrow$  *Compattezza eq symb*);
- Le ruote, nel calcolo della dinamica completa del VAB, sono state considerate con un contributo doppio;
- Nello studio dello chassis (base), abbiamo seguito questo approccio. Il baricentro della base stessa sappiamo essere posizionato ad una quota differente rispetto al centro del sistema di coordinate XZ preso come riferimento.

Per questo motivo il suo contributo in termini cinetici e potenziali dipende dal valore dell'inclinazione dello chassis stesso, ovvero dal valore dell'angolo *caratterizzante* il sistema  $\theta$ : questo concetto è evidenziato in figura 2.3.

L'aggiunta di  $\pi$  al valore di  $\theta$  è necessaria per poter rendere sensibile i valori di energia cinetica e potenziale al *quadrante* in cui si trova ad essere posizionato il centro di massa della base stessa ( $P_b$ ).

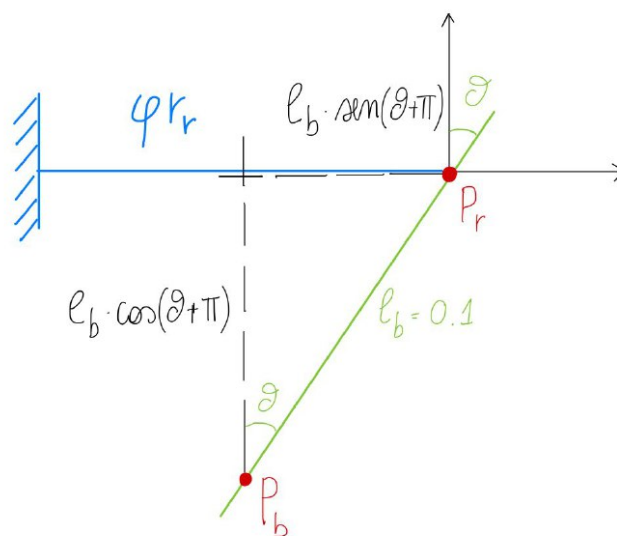


Figure 2.3: Ragionamento per il calcolo dei contributi dinamici dello chassis





### 3

## Equazioni del moto

Una volta definita la dinamica del sistema ed ottenuta quindi l'equazione di Lagrange che ne caratterizza il comportamento, andiamo a ricavare le equazioni del moto, le quali consentono di definire l'andamento delle *coordinate libere* (scelte in fase iniziale di progetto) che sappiamo essere

- $\phi = q_1 = \text{angolo di rotazione delle ruote}$
- $\theta = q_2 = \text{angolo di inclinazione dello chassis}$

In particolare possiamo definire le cosiddette *equazioni di Eulero-Lagrange*, ovvero un insieme di  $n$  equazioni differenziali (con  $n$  pari al numero di coordinate libere del sistema), la cui risoluzione fornisce le equazioni del moto del sistema.

$$\frac{d}{dt} \left( \frac{\partial}{\partial \dot{q}} L \right) - \frac{\partial}{\partial q} L = \text{torque}$$

Sono da effettuare alcune annotazioni sugli addenti presenti nell'equazione 3:

- $q$  rappresenta la coordinata libera (nel nostro caso sarà  $\theta$  e  $\phi$ );
- al secondo membro della lagrangiana TODO



# **Part II**

## **Controllo**



## 4

# OPC - UA

Per quanto riguarda la parte di controllo, il sistema presenta un articolato insieme di controllori inter-operanti tra di loro: nello specifico ad ogni singolo controllore sono affidate delle mansioni ben specifiche, tutte ovviamente volte al controllo e alla stabilizzazione del *veicolo auto bilanciato*.

In questa fase dello sviluppo del progetto, siamo andati ad implementare parte del codice che verrà installato, in un secondo momento, a bordo del raspberry: esso infatti svolge, all'interno del sistema (come si vede in figura 4.1) una comunicazione a due direzioni, che ne determinano due comportamenti differenti:

- Come **server** per la parte di comunicazione *OPC-UA* (per il settaggio dei guadagni);
- Come **master** nella comunicazione seriale verso Arduino (per quanto riguarda invece la gestione dell'algoritmo di controllo);

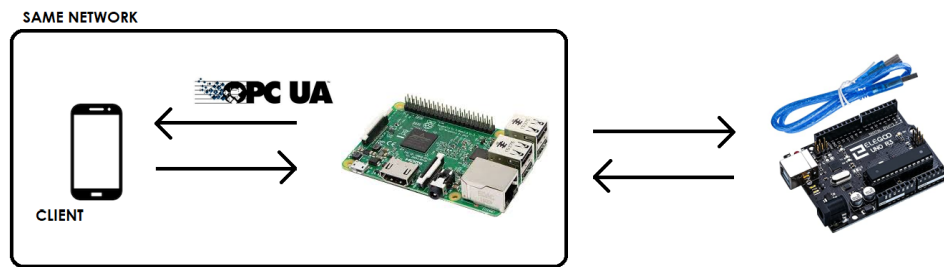


Figure 4.1: Schema di massima dell'utilizzo di Raspberry Pi 3

In questa fase abbiamo quindi sviluppato la parte relativa all'utilizzo di *Raspberry Pi 3* come *server OPCUA*.

## 4.1 Idea base OPC-UA

L'*Open Platform Communications Unified Architecture* (OPC UA) è un protocollo di comunicazione automatico per l'automazione industriale.

OPC UA sostituisce il protocollo *OPC Classic*, conservando tutte le funzionalità del predecessore. Poiché OPC Classic era stato costruito su una tecnologia Microsoft detta modello a oggetti per componenti distribuiti, era vincolato a Microsoft, ma questa caratteristica è diventata sempre più limitante.

OPC UA risulta completamente interoperabile tra i diversi sistemi operativi usati, aggiungendosi a Windows e alle tecnologie industriali come i PLC, inoltre comprende Linux, iOS e anche sistemi operativi per dispositivi mobili come Android. Consentire al maggior numero di dispositivi possibile di comunicare contribuisce al progresso dell'IoT.

Queste caratteristiche di **interoperabilità** sono state sfruttate al massimo in questo contesto, potendo così creare, in maniera semplice e veloce, una comunicazione tra differenti tipologie e famiglie di dispositivi.

## 4.2 OPC-UA e V.A.B.

Nel contesto del progetto del *veicolo auto bilanciato*, siamo andati ad utilizzare il protocollo di comunicazione *OPC-UA* come supporto per il tuning dei parametri relativi al **gain** del controllore, ovvero ai parametri del vettore  $K$ , che abbiamo chiamato (all'interno dello script di Python):

- **K\_phi**
- **K\_phi\_p**
- **K\_theta**
- **K\_theta\_p**

Abbiamo racchiuso in figura 4.2 il funzionamento di massima del codice lato server: codice che siamo andati a testare utilizzando un'apposita app per smartphone Android (OPC-UA Android client).

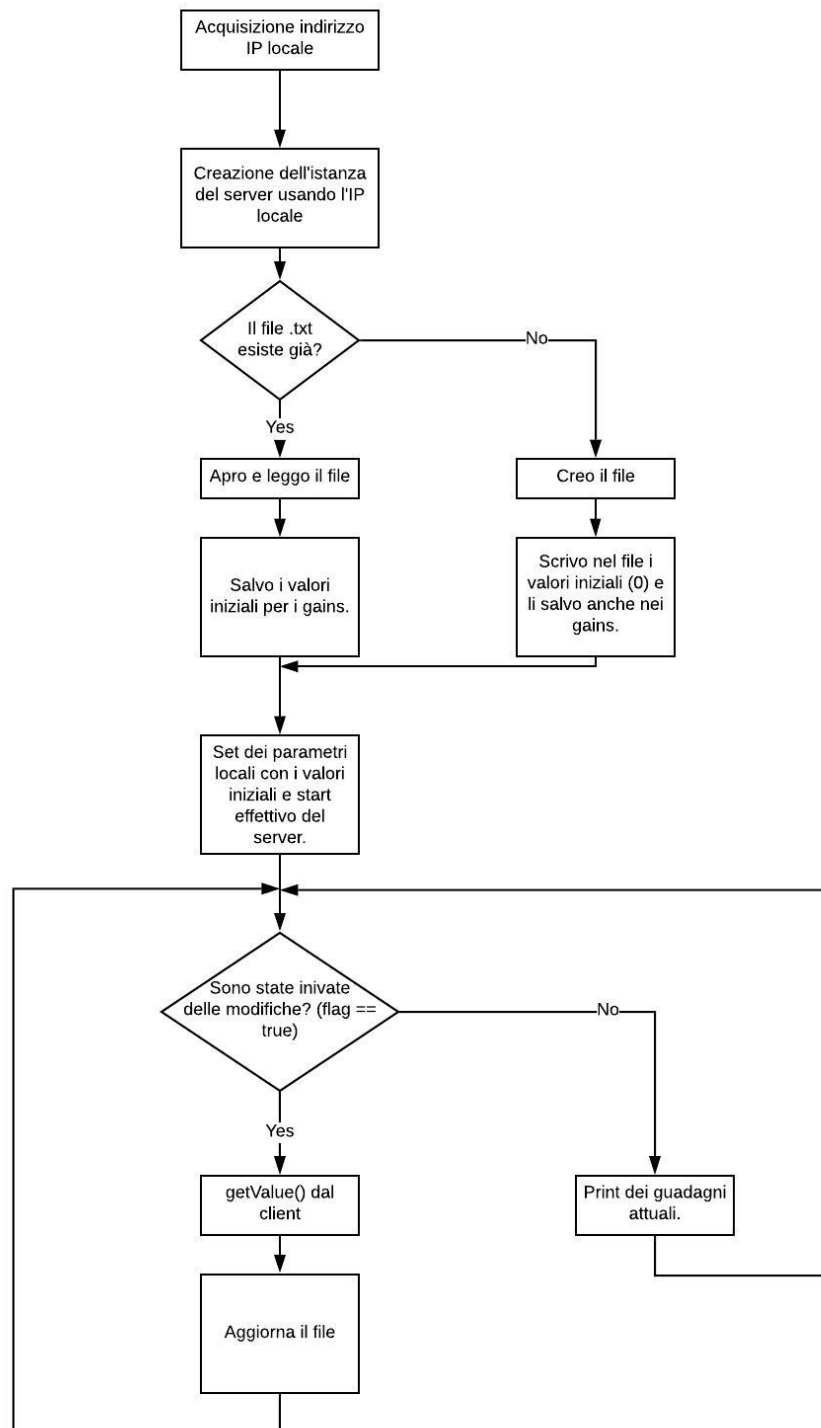


Figure 4.2: Diagramma rappresentante le funzionalità del server





5

## Rumore

Piattaforma inerziale  $\rightarrow$  PSD Questione anti windup  $\rightarrow$  coppia già limitata quindi nessun vincolo di saturazione



# Bibliography

[1] *bla bla bla* blablabla