

Research Project ICAROS 2020/2021 Rehabilitation Robotics

Piccoli Danilo, Alessandro Santopaolo

10 December 2020



This report is submitted in fulfilment of the requirements for the research project "Reh. Kuka Med" by Prof. Fanny Ficuciello

1 Introduction

I robot sono sempre stati i perfetti candidati per svolgere compiti di tipo industriale che richiedessero elevata precisione e grandi sforzi. Tuttavia, a partire dagli anni Novanta la robotica si è sviluppata anche in altri ambiti, come quello della medicina. In tale ambito si sono andati a creare svariati tipi di manipolatori atti a migliorare il lavoro dei medici e le condizioni dei pazienti. Una parte di essi è stata indirizzata verso compiti di tipo riabilitativo, andando a creare la branca della robotica riabilitativa. Ogni anno ictus e traumi cerebrali producono milioni di vittime e la maggior parte dei sopravvissuti riporta gravi danni. I deficit cerebrali causati dalla malattia causano perdite di controllo motorio relativi agli arti superiori o inferiori. Per questo motivo questi soggetti devono iniziare una fase riabilitativa per recuperare tale controllo. Durante la riabilitazione ogni paziente deve svolgere specifici esercizi migliaia di volte e deve essere coadiuvato da un terapeuta che fisicamente lo supporta e lo guida nel farli. Tuttavia, data la presenza di migliaia di pazienti che necessitano di tali attenzioni, il lavoro dei fisioterapisti può diventare eccessivo ed estenuante. Per risolvere questo problema sono stati sviluppati robot riabilitativi con lo scopo di diminuire il carico lavorativo dei terapeuti e velocizzare il tempo di degenza dei soggetti. Un allenamento assistito roboticamente ha un grande potenziale per recuperare le capacità motorie e per la riabilitazione. Tale allenamento può essere altamente preciso, può essere sostenuto per lunghi periodi di tempo, può misurare automaticamente i progressi nel tempo e memorizzarli e infine può riprodurre una vasta gamma di movimenti e forze.

Numerosi robot sono stati progettati per la riabilitazione degli arti superiori e degli arti inferiori. I due più famosi robot riabilitativi delle braccia sono: Il MIT-MANUS, pioniere di tale branca, ora venduto come InMotion ARM (Interactive Motion Technologies, USA) e il ARMin, ora venduto come ArmeoPower (Hocoma AG, Svizzera). Questi e molti altri robot sono stati sviluppati per supportare e guidare gli arti del paziente tuttavia, il solo hardware, anche se appropriato, non è abbastanza. Infatti, insieme alla componente hardware è necessario anche un controllore adatto al manipolatore e al compito che esso deve svolgere. I primi controllori per i robot riabilitativi non adattavano il loro supporto al paziente. Essi erano rigidi ed essenzialmente guidati dall'arto del paziente lungo una traiettoria predefinita con una noncuranza di cosa il paziente stesse facendo o volesse fare. Test clinici hanno trovato che i pazienti mettevano significativamente meno impegno negli esercizi con l'assistente robotico che con quelli con il terapeuta, e frequentemente lasciavano che il robot muovesse il loro arto senza una partecipazione attiva nel movimento. Questo ovviamente portava ad un recupero più lento del

paziente. Per questo motivo i robot necessitano di una strategia di controllo che assista il paziente solo quando ce ne bisogno, ovvero una strategia di controllo collaborativa o condivisa tra umano e macchina.

In questo documento andremo a descrivere il lavoro svolto sul controllore basato sulle regioni per il controllo del moto di un manipolatore atto alla riabilitazione di pazienti affetti da difficoltà motorie relative agli arti superiori. Verrà presentata una lista di funzioni utilizzate all'interno del controllore, con descrizione del funzionamento ed esempio di implementazione.

1.1 Outline

La seguente parte del documento è strutturata come segue:

Nel capitolo 2, è sintetizzato il paper dal quale il lavoro ha preso origine [1].

Nel capitolo 3 invece sono introdotte in dettaglio le funzioni utilizzate per l'implementazione dei controllori in esame.

Nel Capitolo 4 sono presenti i risultati della riproduzione del paper.

2 Controllo di interazione uomo-robot per la robotica riabilitativa di arti superiori

2.1 Interazione uomo-robot comprendente il controllo:

Si identifica il controllo in tre modalità operative basate sul task da effettuare. Il controllore si suddivide in due parti: una relativa al movimento e una relativa all'interazione con l'umano. Entrambe implementate in ogni modalità. Le modalità operative sono tre e sono identificabili con:

- La modalità di dominio umano (H-DR) dove i movimenti del paziente sono affidati a lui e incoraggiati, questi movimenti sono solo leggermente interferiti dal sistema robotico. Questa modalità si riferisce alla situazione dove il controllo del moto umano viene parzialmente ristabilito; il movimento libero del braccio è permesso, mentre la velocità è moderata tra un minimo e un massimo.
- La modalità di dominio robotico (R-DR) nella quale il controllo del movimento non è affidato al paziente e i movimenti desiderati sono compiuti per la maggior parte dal sistema robotico. Questo è lo scenario in cui l'umano effettua un movimento non sicuro e/o inefficiente. Allora, l'assistente robotico fa terminare il movimento e lo fa tendere verso il percorso corretto. In questo caso l'inseguimento della traiettoria avviene con un regolatore sia di posizione che di velocità.
- la modalità di sosta di sicurezza (S-SR) nella quale il sistema robotico viene fermato a causa della possibile esistenza di movimenti uomo-macchina dannosi per l'umano.

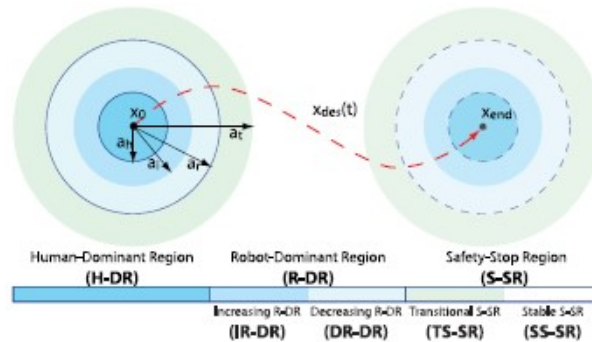


Figure 1: Regioni operative

Quando il sistema subisce un'estensione tale che l'errore di posizione diventi troppo elevato e di conseguenza il controllo della coppia possa diventare dannoso per il corpo umano, l'assistente robotico deve fermarsi. Inoltre, per adattarsi alle differenti condizioni dei pazienti e del loro variegato stato di salute, il peso dell'HDR e del RDR deve essere adattato di caso in caso. Allo stato iniziale della terapia la HDR è ridotta al minimo o resa inaffidabile (nessun controllo all'umano) e l'assistente robotico diventa costantemente presente. Man mano che si avanza con la terapia, la zona dell'HDR diventa sempre più larga. Le dimensioni delle restanti regioni possono essere definire durante gli allenamenti iniziali.

Come scritto sopra, il controllore è diviso in due parti:

$$\tau = \tau_m + \tau_i$$

dove τ_m è controllore del moto e τ_i il controllore relativo all'interazione. La τ_m realizza le tre principali modalità cercando di compensare l'incertezza del modello, mentre la τ_i lavora sui movimenti umano-robot conflittuali.

2.2 Controllore relativo al movimento:

Il controllore include un termine per controllo di rigidezza(proporzionale) (a), un termine per il controllo sliding (b) e un termine adattivo (c). I primi due insieme realizzano le tre operazioni regionali presenti in figura1, mentre il terzo lavora per compensare le incertezze del sistema. Tale controllore è espresso come:

$$\tau_m = \underbrace{-\mathcal{M}^T(x)J^T(q)k_p(\delta x)\delta x}_{\text{term a}} - \underbrace{K_s s}_{\text{term b}} + \underbrace{Y_d(q, \dot{q}, \ddot{q}_r, \ddot{q}_r)\hat{\theta}}_{\text{term c}}$$

Con x la posizione dell'end-effector nello spazio del task, δx errore di posizione pesato, M una matrice modificatrice, k_p rigideza definita positiva, K_s guadagno positivo del controllo sliding, q_r vettore dei giunti di riferimento, Y_d matrice di regressione, $\hat{\theta}$ parametro dinamico stimato del vettore teta. k_p parametro dipendente dalla posizione δx con:

$$\delta x = x - x_d$$

dove x_d traiettoria pesata espressa come:

$$x_d = x_i + w(\Delta x)(x_{\text{des}} - x_i) = x_i + w(\Delta x)x_v$$

con x_i posizione di riferimento tempo invariante, x_v è la parte tempo variante di x_d e $w(\Delta x)$ è un vettore pesato su Δx (errore di posizione nello spazio operativo). La matrice M è aggiunta per eliminare l'uso di \ddot{x} nel controllore.

$$\mathcal{M}(x) = \mathbb{I}_n - J^+(q)A(x)J(q)$$

$$A(x) = \begin{bmatrix} \frac{\partial w}{\partial x_1} x_{\text{des},1} & \dots & \frac{\partial w}{\partial x_m} x_{\text{des},1} \\ \vdots & \ddots & \vdots \\ \frac{\partial w}{\partial x_1} x_{\text{des},m} & \dots & \frac{\partial w}{\partial x_m} x_{\text{des},m} \end{bmatrix}$$

A è la matrice di transizione.

Riguardo la componente di sliding:

$$\begin{aligned} s &= \dot{q} - \dot{q}_r \\ &= \dot{q} - \mathcal{M}^{-1}(x)[J^+(q)\dot{x}_f - \alpha J^+(q)k_p(\delta x)\delta x] \end{aligned}$$

α costante e x_f posizione di riferimento.

Per il termine adattivo invece:

$$\dot{\hat{\theta}} = -LY_d^\top(q, \dot{q}, \dot{q}_r, \ddot{q}_r)s$$

con L matrice di guadagno quadrata non singolare positiva.

2.3 Controllo relativo all'interazione:

Il controllore ha due principali obiettivi, assicurare la stabilità a ciclo chiuso insieme al controllore di movimento, e di ridurre il conflitto tra movimenti del robot e dell'umano. Esso comprende un controllore dipendente dalla forza applicata dall'umano, dall'errore di posizione e dal vettore di sliding:

$$\tau_i = J^T(q)C_x(f_H, \Delta x, s)$$

con C_x controllore di iterazione dello spazio del task. I controllori di tale interazione uomo-robot risultano essere stabili, come dimostrato in [1].

3 Dettagli dell'implementazione

Nella tabella seguente 1 vi è una panoramica delle funzioni Matlab implementate per la creazione dell'algoritmo di controllo:

Nome del File
<i>derivate_A.m</i>
<i>derivate_Iplus.m</i>
<i>derivate_delta_x.m</i>
<i>derivate_dqr.m</i>
<i>derivate_dxf.m</i>
<i>derivate_inv_M.m</i>
<i>derivate_jacobian.m</i>
<i>derivate_kp.m</i>
<i>derivate_w.m</i>
<i>dominant_region.m</i>
<i>force_diff.m</i>
<i>func_forwardKinematics.m</i>
<i>func_getJacobian.m</i>
<i>func_getTransformations.m</i>
<i>func_make_rotation.m</i>
<i>func_make_translation.m</i>
<i>interaction_controller.m</i>
<i>interaction_tem.m</i>
<i>pos_stiffness.m</i>
<i>region_controller.m</i>
<i>saturation_func.m</i>
<i>theta_estimate.m</i>
<i>transition_matrix_A.m</i>
<i>transition_smooth_func.m</i>
<i>unit_vector_sx.m</i>
<i>regress_matrix.m</i>
<i>weight_vector.m</i>

Table 1: Elenco delle funzioni Matlab per controllore

3.1 Region Controller

E' la funzione che elabora la coppia da applicare ai giunti in base alla regione di appartenenza. Esso si suddivide in due controllori: Un controllore relativo al moto (Equazioni 3a/3b/3c) e un controllore relativo all'interazione (Equazione 19) andando a definire il controllore totale nell'equazione (2).

3.1.1 Controllore del moto

L'espressione matematica del controllore del moto:

```
tau_m_a=-M'*J'*k_p*delta_x;           % stiffness controller (3a)
tau_m_b=-K_s*s;                         % sliding controller (3b)
tau_m_c=Y_d*theta_cap;                  % adaptive controller (3c)
```

Figure 2: Controllore del moto

Dove, all'interno del controllore di rigidità (3a) sono presenti:

1. δx Errore di posizione pesato. (Equazione 4)
2. k_p Termine di rigidità dipendente dalla posizione. (Equazione 17)
Calcolata tramite funzione *pos_stiffness.m*
3. J Jacobiano del manipolatore.
Calcolato tramite funzione *func_getJacobian.m*
4. M Matrice modificatrice. (Equazione 6)

Il termine 1 δx dipende dai termini x_e e x_d che rappresentano rispettivamente il vettore della posizione corrente dell'end effector e il vettore di una traiettoria pesata.

La traiettoria pesata x_d (Equazione 5) è così calcolata :

```
x_d=x_i+w*(xdes-x_i);                  % weighted trajectory (5)
delta_x=x_e-x_d;                        % weighted position error (4)
```

Dove x_i è un riferimento di posizione tempo invariante; w (Equazione 10) termine pesato sull'errore di posizione Δx , **calcolato tramite funzione *weight_vector.m***; x_{des} riferimento di posizione dell'end-effector.

Il termine 4 M è così calcolato:


```
M=eye(7)-pinv(J)*A*J; % Modifier Matrix (6)
```

e dipende dalla matrice di transizione A (Equazione 7), **calcolata tramite funzione *transition_matrix_A.m***.

Per il controllore di sliding (3b) sono presenti:

1. K_s : Matrice dei guadagni di sliding.
2. s : Vettore di sliding (Equazione 8).

Il termine 2:

```
s=dq-dqr; % sliding vector (8)
```

Dipende dal termine \dot{q} vettore delle velocità dei giunti e \dot{q}_r vettore delle velocità di riferimento virtuali, calcolato così (Equazione 8b):

```
dqr=M\ (pinv(J)*dx_f-alpha*pinv(J)*k_p*delta_x); % derivate of virtual joint reference vector (8b)
```

Dove α è un termine scalare costante e \dot{x}_f è il vettore di riferimento delle velocità virtuali (Equazione 8a) calcolato così:

```
dx_f=w*dxdes-A*dxdes; % virtual position reference vector (8a)
```

che dipende dalla matrice di transizione A e dalla velocità di riferimento \dot{x}_{des} .

Nel controllore adattativo (3c) invece sono presenti:

```
Y_d=regres_matrix(q, dq, dqr, ddqr); %da fare (trovare modello dinamico) %regressor matrix
theta_cap=theta_estimate(theta, Y_d, L) % estimated dynamic parameter vector (9)
```

1. Y_d : Matrice di regressione del modello.
2. $\hat{\theta}$: Vettore dei parametri dinamici, calcolato in linea (Equazione 9).

Il termine 1 dipende dai termini q , \dot{q} , \dot{q}_r , \ddot{q}_r , **calcolata tramite funzione *regress_matrix***. Dove \ddot{q}_r **calcolato tramite funzioni *derivate_dqr*, *derivate_A*, *derivate_Jplus*, *derivate_delta_x*, *derivate_dqr*, *derivate_dxf*, *derivate_inv_M*, *derivate_jacobian***.

```
ddqr=derivate_dqr();
```

Il termine 2 invece viene **calcolata tramite funzione *theta_estimate***, dove L è una matrice definita positiva e non-singolare.

3.2 Interaction Controller

Il controllo di iterazione (Equazione 19) è così composto:

$$\tau_i = J' * C_x$$

Dove J' è lo jacobiano trasposto and C_x è un controllo di interazione nello spazio del task (Equazione 21). **calcolato tramite funzione** *interaction_controller.m*

```
C_x=interaction_controller(mu_s, mu_x, c_x, f_h, region); % term of interaction controller (21)
```

Dipendente dai termini:

1. μ_x è una funzione di transizione liscia dalla regione Human Dominant a quella Robot Dominant (Equazione 23), **calcolata tramite funzione** *transition_smooth_func.m*.
2. μ_s è una funzione di saturazione (Equazione 23a) , **calcolata tramite funzione** *saturation_func.m*.
3. c_x termine di interazione (Equazione 22), **calcolata tramite funzione** *interaction_tem.m*.

Il termine 1 dipende dall'errore di posizione nello spazio operativo e dalla regione di appartenenza.

Il termine 2 dipende da s_x variabile composita (Equazione 20)

$$s_x = J * s$$

e da w_s costante di saturazione.

Il termine 3 dipende dalla regione di interazione, calcolata in base alla distanza angolare γ tra il vettore $-s_x$ e f_h vettore delle forze di interazione uomo-robot; In base al valore di gamma si definisce la regione di interazione (Retail Region RR, Projecting Region PR, Cancelling Region CR). E in base al versione $-s'_x$ proiezione del vettore $-s_x$ ad una distanza angolare β . **Calcolate tramite le funzioni** *unit_vector_sx.m*, *force_diff.m*.

```
interact_region=force_diff(f_h, s_x, beta, sigma); % interaction region selector
mu_s=saturation_func(s_x, w_s); % saturation function (23a)
mu_x=transition_smooth_func(region, a_r, a_h, Delta_x); % smooth transition function from HD and RD (23)
c_x=interaction_tem(interact_region, beta, sigma, f_h, s_x); %interaction term (22)
C_x=interaction_controller(mu_s, mu_x, c_x, f_h, region); % term of interaction controller (21)
```

References

- [1] Juanjuan Zhang and Chien Chern Cheah. Passivity and stability of human–robot interaction control for upper-limb rehabilitation robots. *IEEE Transactions on Robotics*, 31(2):233–245, 2015.