

Università Politecnica delle Marche  
Facoltà di Ingegneria

Corso di Laurea triennale in Ingegneria Informatica e dell'Automazione



# **Progettazione e realizzazione di un sistema gemello digitale di propulsori subacquei**

*Design and realization of a digital twin system for  
DPVs*

Relatore:

Prof. David Scaradozzi

Correlatore:

Dott. Nicolò Ciuccoli

Candidato:

Enrico Tarsi

Anno Accademico 2019/2020

## *Indice*

1	Introduzione.....	1
2	Studio della dinamica .....	3
2.1	Massa aggiunta .....	4
2.2	Forza di attrito viscoso.....	5
2.3	Spinta dell'elica .....	6
2.4	Modello dinamico Scooter.....	6
2.5	Modello dinamico Scooter+Diver.....	7
3	Infrastruttura di simulazione.....	9
3.1	Adams MSC.....	9
3.1.1	Disegno del modello CAD 3D.....	10
3.1.2	Creazione delle misure .....	11
3.1.3	Applicazione delle forze.....	12
3.1.4	Creazione delle variabili.....	12
3.1.5	Esportazione del modello.....	14
3.2	Matlab e Simulink.....	16
3.2.1	Importazione in Simulink del modello Adams .....	17
4	Schemi e simulazioni .....	19
4.1	Sistema Scooter+Diver .....	19
4.2	Sistema Scooter+Diver+MotoreBLDC.....	21
4.3	Sistema Scooter+Diver+MotoreDC.....	24
5	Identificazione.....	26
5.1	Raccolta dei dati.....	26

5.2	Individuazione della famiglia di modelli.....	28
5.3	Scelta del miglior modello della famiglia.....	32
5.4	Validazione .....	35
6	Identificazione dei Sistemi .....	36
6.1	Identificazione Sistema Scooter+Diver .....	36
6.1.1	Raccolta dei dati.....	36
6.1.2	Determinazione dell'ordine a priori .....	40
6.1.3	Stima dei modelli.....	42
6.1.4	Validazione.....	45
6.2	Identificazione Sistema Scooter+Diver+MotoreDC .....	49
6.3	Simulazioni .....	53
6.3.1	Simulazione modello arx210.....	53
6.3.2	Simulazione modello armax3140 .....	54
7	Conclusioni.....	56

# 1 Introduzione

I propulsori subacquei anche noti con l'acronimo DPV (Diver Propulsion Vehicle) sono dei dispositivi di supporto usati durante le immersioni che hanno lo scopo di facilitare gli spostamenti dei sub e permettere loro di coprire aree maggiori. Essi sono provvisti di una o più eliche azionate da dei motori elettrici che ne permettono il movimento, il tutto alimentato da una batteria.

Esistono diverse tipologie di DPV, le quali si differenziano per forma e per posizione di guida, in questo elaborato si farà riferimento allo scooter che è caratterizzato da una forma cilindrica con un'elica posta nella parte posteriore in cui il diver viene trascinato.



*Figura 1.1: tre sub in immersione con l'ausilio di un DPV Suex  
(<http://www.stylediving.com/SUEX/CORSI%20DPV/>)*

Questa tesi si pone l'obiettivo di creare un gemello digitale di un sistema formato da un subacqueo e un DPV scooter e di ricavarne un modello parametrico attraverso un'identificazione black box che ne descriva il movimento nell'asse longitudinale. Nell'elaborato verranno messi a confronto più sistemi che si differenziano per l'uso del blocco propulsivo, ovvero del motore che aziona l'elica, considerando: un motore ideale, un motore BLDC e un motore DC.

Per gemello digitale si intende una rappresentazione virtuale del sistema nel suo complesso in grado di simularne e monitorarne il comportamento. Per la sua realizzazione in primo luogo si sono studiate le equazioni che governano il sistema nell'asse longitudinale considerando le reazioni idrodinamiche e facendo le opportune

approssimazioni del caso. In seguito, sfruttando il software CAD Adams si è potuto realizzare la struttura meccanica a cui si sono applicate le forze che ne regolano la dinamica. Integrando il modello realizzato in Adams con i software Matlab e Simulink è stato possibile effettuare delle simulazioni e, grazie alla libreria Simscape Electrical, aggiungere i vari blocchi propulsivi al sistema. Inoltre, si è voluto effettuare l'identificazione mediante la tecnica black box di due dei sistemi creati: quello con motore ideale e quello con motore DC, così da ottenere dei modelli parametrici, indipendenti dai software usati per la creazione del gemello digitale, che descrivono i due sistemi.

## 2 Studio della dinamica

In questo capitolo si vuole studiare la dinamica del DPV scooter nella sola direzione longitudinale 'x' trovando l'equazione che ne governa il movimento. Questa fase è fondamentale per determinare le forze che agiscono sul sistema e per essere inserite poi nel modello Adams. Lo scooter preso in esame fa riferimento al DPV SUEX XJT per dimensioni, peso e caratteristiche e viene approssimato con un cilindro di raggio R e lunghezza L (vedi Tabella 2.2). Una volta determinata l'equazione dello scooter vengono aggiunti i contributi delle forze dovute alla presenza del diver (sub) e della bombola dell'ossigeno, entrambi vengono approssimati a due cilindri di opportune dimensioni come fatto per lo scooter (vedi Tabella 2.3 e Tabella 2.4).



Figura 2.1: DPV Suex modello XJT (<https://www.suex.it/prodotti.php>)

Il sistema di riferimento utilizzato è descritto dalle seguenti terne:

$$\gamma_1 = [x, y, z]^T \quad (2.1)$$

$$v_1 = [u, v, w]^T \quad (2.2)$$

$$\gamma_2 = [\phi, \theta, \varphi]^T \quad (2.3)$$

$$v_2 = [p, q, r]^T \quad (2.4)$$

Dove  $\gamma_1$  descrive la posizione nello spazio e  $v_1$  le velocità lungo i tre assi, mentre  $\gamma_2$  rappresenta l'insieme degli angoli di rotazione nei tre assi (roll, pitch, yaw) e  $v_2$  le rispettive velocità angolari.

L'equazione della dinamica dello scooter in accordo con la formulazione di Newton-Eulero è descritta da:

$$m(\dot{u} - vr + wq) = \sum X_{ext} \quad (2.5)$$

Considerando però i contributi di  $v$ ,  $w$ ,  $r$  e  $q$  nulli si ottiene semplicemente:

$$m\dot{u} = \sum X_{ext} \quad (2.6)$$

Dove  $m$  è la massa dello scooter mentre  $X_{ext}$  rappresenta le forze esterne applicate allo scooter nella direzione 'x', che sono:

- Forza della massa aggiunta
- Forza di attrito
- Spinta dell'elica

## 2.1 Massa aggiunta

La massa aggiunta corrisponde alla massa dell'acqua che viene messa in movimento quando avviene una accelerazione del veicolo. Per stimare la massa aggiunta si approssima la forma dello scafo del veicolo con un ellissoide, dove l'asse maggiore ha come valore metà della lunghezza  $L$  del veicolo, e l'asse minore equivale a metà del diametro  $D$  del veicolo. La massa aggiunta per un ellissoide è definita da:

$$X_u = \frac{4\alpha\rho\pi}{3} \left(\frac{L}{2}\right) \left(\frac{D}{2}\right)^2 \quad (2.7)$$

Dove  $\rho$  è la densità dell'acqua e  $\alpha$  è un parametro determinato dal rapporto tra la lunghezza e il diametro del veicolo (vedi Tabella 2.1). La massa aggiunta, quando avviene un'accelerazione, causa una forza in opposizione al movimento e avrà quindi segno negativo:

$$X = -X_u|\dot{u}| \quad (2.8)$$

Tabella 2.1: parametri  $\alpha$  della massa aggiunta dati dal rapporto tra lunghezza e diametro dell'oggetto

$L/D$	$\alpha$
0.01	-
0.1	6.148
0.2	3.008
0.4	1.428
0.6	0.9078
0.8	0.6514
1.0	0.5000
1.5	0.3038
2.0	0.2100
2.5	0.1563
3.0	0.1220
5.0	0.05912
7.0	0.03585
10.0	0.02071

## 2.2 Forza di attrito viscoso

La forza di attrito viscoso viene espressa dalla seguente relazione:

$$X = -\left(\frac{1}{2}\rho c_d A_f\right)u|u| \quad (2.9)$$

Il coefficiente di attrito viscoso può quindi essere espresso in questo modo:

$$X_{|u|u} = -\frac{\rho A_f c_d}{2} \quad (2.10)$$

dove  $A_f = \pi R^2$  è l'area frontale del veicolo, nel nostro caso quindi l'area del cerchio di raggio  $R$ , mentre il coefficiente di attrito  $c_d$  è dato da:

$$c_d = \frac{c_{ss}\pi A_p}{A_f} \left[ 1 + 60 \left(\frac{D}{L}\right)^3 + 0.0025 \left(\frac{L}{D}\right) \right] \quad (2.11)$$

in cui  $A_p = LD$  è l'area planare del veicolo e  $c_{ss}$  è il valore di Schoenherr per l'attrito della lastra piana.



## 2.3 Spinta dell'elica

La spinta generata dalla rotazione dell'elica può essere espressa come segue

$$T = T_{|n|n}|n| - T_{|n|u_a}|n|u_a \quad (2.12)$$

Dove  $n$  è la velocità di rotazione dell'elica espressa in rad/s mentre  $u_a$  è l'*ambient water velocity* data dall'equazione:

$$u_a = (1 - \omega)u \quad (2.13)$$

in cui  $\omega$  è il *wake fraction number* (tipicamente compreso tra 0.1 e 0.4).

I due coefficienti di spinta vengono determinati da:

$$T_{|n|n} = \rho d^4 \alpha_2 \quad (2.14)$$

$$T_{|n|u_a} = \rho d^4 \alpha_1 \quad (2.15)$$

dove  $d$  è il diametro dell'elica, mentre  $\alpha_1$  e  $\alpha_2$  sono due coefficienti che si possono ottenere effettuando specifici test nei tunnel di cavitazione.

Non avendo a disposizione lo scooter e non potendo effettuare i test per trovare i coefficienti di spinta  $T_{|n|n}$  e  $T_{|n|u_a}$  si è deciso di stimare i due valori in modo tale che la spinta dell'elica fosse adeguata al tipo di veicolo preso in considerazione.

Tenendo presente che il DPV Suex XJT raggiunge una velocità massima di 1.5m/s e considerando la velocità massima dell'elica di 800rpm si sono stimati i valori dei due coefficienti effettuando delle prove attraverso il modello Adams (vedi Tabella 2.2).

## 2.4 Modello dinamico Scooter

Dopo aver analizzato tutte le forze che lo caratterizzano si può esprimere il modello dinamico dello scooter nella direzione 'x' come:

$$\sum X_{ext} = -X_{\dot{u}}|\dot{u}| - X_{|u|u}u|u| + T \quad (2.16)$$

In cui si distinguono la forza di massa aggiunta e la forza di attrito viscoso in opposizione alla spinta dell'elica.

## 2.5 Modello dinamico Scooter+Diver

Per ottenere l'equazione che descrive un sistema formato da uno scooter e un diver dotato di bombola dell'ossigeno, si aggiungono all'equazione dello scooter (2.16) i contributi del diver e della bombola relativi alla forza di attrito viscoso e alla forza della massa aggiunta, come mostrato nella (2.17).

$$\sum X_{ext} = -(X_{\dot{u}} + X_{D,\dot{u}} + X_{B,\dot{u}})|\dot{u}| - (X_{|u|u} + X_{D,|u|u} + X_{B,|u|u})u|u| + T \quad (2.17)$$

dove,  $X_{D,\dot{u}}$  e  $X_{D,|u|u}$  sono rispettivamente la massa aggiunta e il coefficiente di attrito del diver, mentre  $X_{B,\dot{u}}$  e  $X_{B,|u|u}$  sono la massa aggiunta e il coefficiente di attrito della bombola. Entrambi sono stati approssimati a un cilindro di opportune dimensioni e i coefficienti sono stati calcolati nello stesso modo visto per lo scooter (vedi Tabella 2.3 e Tabella 2.4).

Tabella 2.2: Parametri Scooter

Parametro	Valore	Unità di misura	Descrizione
$L$	0.814	$m$	Lunghezza Scooter
$D$	0.197	$m$	Diametro Scooter
$R$	0.0985	$m$	Raggio Scooter
$m$	20	$Kg$	Massa Scooter
$A_f = \pi R^2$	0.0305	$m^2$	Area frontale Scooter
$A_p = LD$	0.1604	$m^2$	Area planare Scooter
$\rho$	1025	$Kg/m^3$	Densità acqua
$c_{ss}$	$3.397 \times 10^{-3}$		Valore di Schoenherr
$c_d$	0.1221		Coefficiente di attrito Scooter
$T_{ n n }$	0.5		Coefficiente di spinta
$T_{ n u_a}$	32		Coefficiente di spinta
$\omega$	0.2		Wake fraction number
$X_{\dot{u}}$	1.5354	$kg$	Massa aggiunta Scooter
$X_{ u u}$	1.9074	$Kg/m$	Coefficiente di attrito Scooter

Tabella 2.3: Parametri Diver

Parametro	Valore	Unità di misura	Descrizione
$L_D$	1.8	$m$	Lunghezza Diver
$D_D$	0.4	$m$	Diametro Diver
$R_D$	0.2	$m$	Raggio Diver
$m_D$	80	$Kg$	Massa Diver
$A_{D,f} = \pi R_D^2$	0.1257	$m^2$	Area frontale Diver
$A_{D,p} = L_D D_D$	0.72	$m^2$	Area planare Diver
$c_{ss}$	$3.397 \times 10^{-3}$		Valore di Schoenherr
$c_{D,d}$	0.1021		Coefficiente di attrito Diver
$X_{D,\dot{u}}$	11.5677	$kg$	Massa aggiunta Diver
$X_{D, u u}$	6.5752	$Kg/m$	Coefficiente di attrito Diver

Tabella 2.4: Parametri Bombola

Parametro	Valore	Unità di misura	Descrizione
$L_B$	0.65	$m$	Lunghezza Bombola
$D_B$	0.2	$m$	Diametro Bombola
$R_B$	0.1	$m$	Raggio Bombola
$m_B$	16	$Kg$	Massa Bombola
$A_{B,f} = \pi R_B^2$	0.0314	$m^2$	Area frontale Bombola
$A_{B,p} = L_B D_B$	0.13	$m^2$	Area planare Bombola
$c_{ss}$	$3.397 \times 10^{-3}$		Valore di Schoenherr
$c_{B,d}$	0.1217		Coefficiente di attrito Bombola
$X_{B,\dot{u}}$	1.5927	$kg$	Massa aggiunta Bombola
$X_{B, u u}$	1.9595	$Kg/m$	Coefficiente di attrito Bombola

### 3 Infrastruttura di simulazione

Per la realizzazione del gemello digitale è stato necessario l'utilizzo di diversi software in grado di interagire tra loro. Di seguito verrà illustrato come creare il modello CAD 3D dello scooter utilizzando il software Adams MSC e come effettuare il collegamento con i software Matlab e Simulink per permetterne poi l'esecuzione. Al termine di questa fase si avrà a disposizione un blocco in Simulink che fa riferimento al modello creato in Adams (che chiameremo blocco Adams) e sarà caratterizzato da un ingresso in cui va inserita la velocità dell'elica espressa in rad/s, mentre in uscita verranno riportati i valori della posizione, velocità e accelerazione oltre ai valori delle forze che agiscono su di esso.

#### 3.1 Adams MSC

Adams è un software sviluppato da MSC che viene usato per la simulazione dinamica multibody. Supporta gli ingegneri nello studio della dinamica delle parti in movimento e nell'analisi della distribuzione di forze e carichi attraverso sistemi meccanici<sup>1</sup>. L'intero sistema costituito dallo scooter, dal diver e dalla bombola è stato raffigurato dal solo disegno 3D dello scooter sul quale però sono state applicate tutte le forze viste nel capitolo precedente, considerando quindi anche le forze di attrito e di massa aggiunta del diver e della bombola. Per far sì che tale approssimazione non comprometta il corretto funzionamento si deve associare al solido la massa totale del sistema data dalla somma delle tre masse. Questo perché l'accelerazione del solido dovuta alle forze che agiscono su di esso dipende ovviamente dalla massa.

L'ambiente di lavoro di Adams si presenta come in Figura 3.1 dove nella parte superiore è presente la barra degli strumenti divisa per categorie, mentre sulla sinistra si trova il browser del modello in cui viene riportata una lista di tutti gli elementi creati. La realizzazione del modello viene suddivisa in cinque fasi:

---

<sup>1</sup> <https://www.mscsoftware.com/it/product/adams>

1. Disegno del modello CAD 3D
2. Creazione delle misure
3. Applicazione delle forze
4. Creazione delle variabili
5. Esportazione del modello

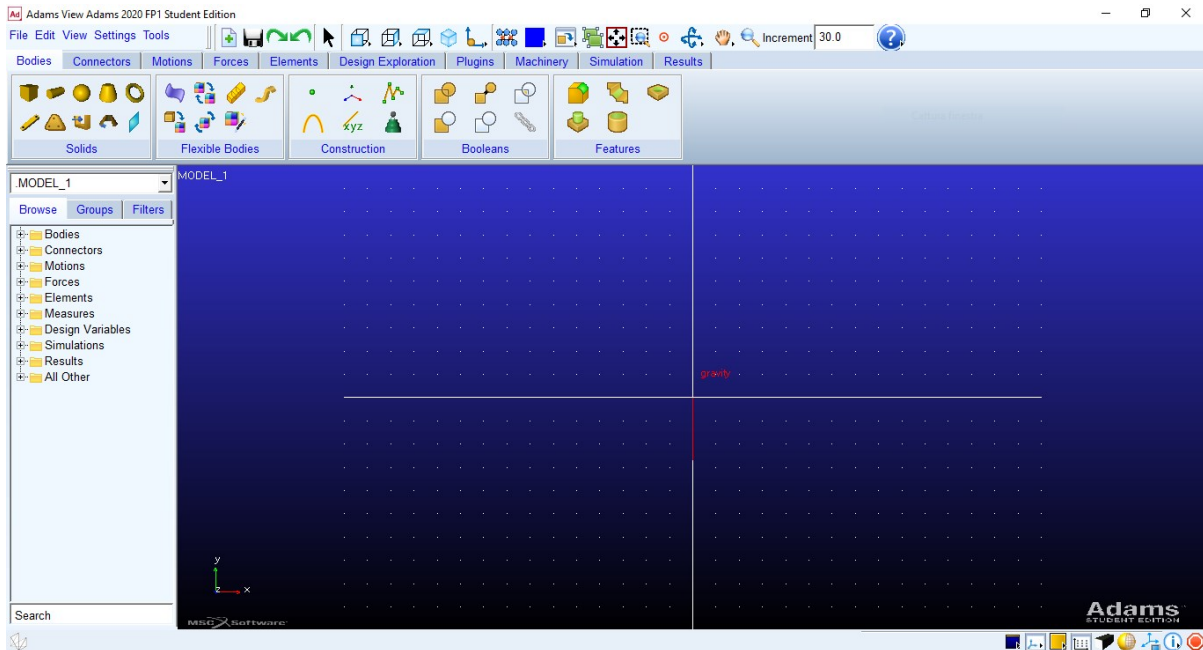


Figura 3.1: ambiente di lavoro del software Adams

### 3.1.1 Disegno del modello CAD 3D

Per prima cosa è stato necessario effettuare il disegno 3D della struttura che in questo caso è stata semplificata ad un cilindro di lunghezza  $L$  e raggio  $R$ . Dalla sezione 'Bodies' si seleziona l'icona del cilindro e portando il cursore sull'ambiente di lavoro si posiziona il solido; a questo punto si apre una finestra in cui vanno inseriti lunghezza e raggio del cilindro. Oltre al solido vengono creati automaticamente dei MARKER, in particolare si vedrà l'utilizzo del MARKER 'cm' che corrisponde al centro di massa del solido, quest'ultimo infatti verrà usato come punto di riferimento per le misure di posizione, velocità e accelerazione. Inoltre, è necessario impostare il valore della massa per cui, premendo con il tasto dx sul cilindro e selezionando 'modify', si aprirà una finestra come quella mostrata in Figura 3.2 nella quale è sufficiente inserire il valore in Kg.

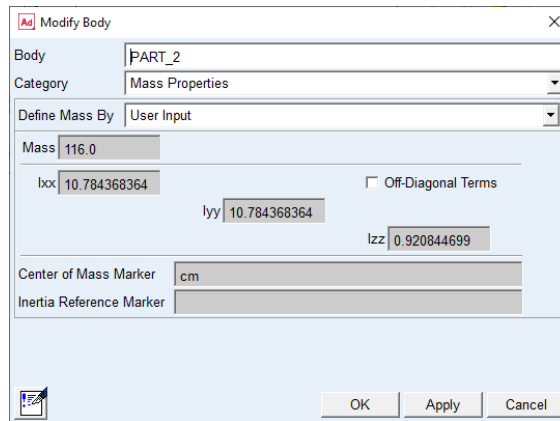


Figura 3.2: finestra di dialogo del software Adams per modificare il valore della massa di un corpo

### 3.1.2 Creazione delle misure

Per monitorare e studiare il comportamento del solido è necessario creare delle misure in grado di determinare posizione, velocità e accelerazione del centro di massa. Queste verranno poi utilizzate anche nelle equazioni che regolano le forze applicate al solido. Premendo con il tasto dx sul Marker 'cm' e selezionando 'Measure' si apre una finestra come in Figura 3.3 in cui si può scegliere il tipo di misura da effettuare. Nella Tabella 3.1 vengono riportate tutte le misure effettuate sullo scooter.

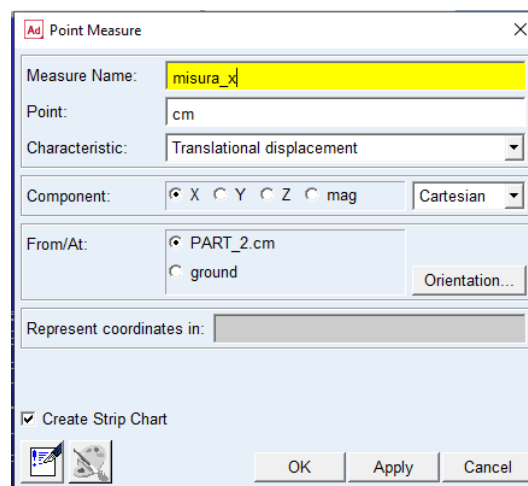


Figura 3.3: finestra di dialogo del software Adams per la creazione di una misura

### 3.1.3 Applicazione delle forze

Per applicare le forze sul solido è sufficiente selezionare 'Forces' nella barra dei comandi, premere su 'create a force' e di nuovo nel punto in cui si vuole applicare la forza; in questo caso tutte le forze sono state applicate nella parte posteriore del solido. L'unica forza nella direzione positiva dell'asse x è la forza chiamata 'Forza\_motori' che corrisponde alla spinta dell'elica, mentre le forze di attrito e le forze di massa aggiunta sono state applicate in direzione opposta. Alla creazione si apre una finestra come quella mostrata in Figura 3.4 dove nella sezione 'Function' va inserita, per ogni forza, l'equazione che la regola. Nella Tabella 3.2 vengono riportate tutte le forze applicate al modello con le relative equazioni che fanno riferimento a quelle studiate nel capitolo 2. Inoltre, per poter riportare in uscita al modello l'andamento delle forze è necessario effettuarne una misura.

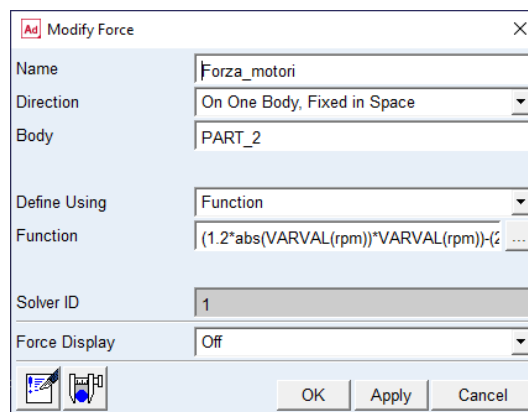


Figura 3.4: finestra di dialogo del software Adams per la creazione di una forza

### 3.1.4 Creazione delle variabili

Le variabili vengono utilizzate per definire l'ingresso e l'uscita del blocco Adams che verrà creato in Simulink.

Per la creazione di una nuova variabile è necessario selezionare 'Elements' nella barra di controllo e premere il comando **x**. A questo punto si apre una finestra come quella in Figura 3.5 in cui nella casella 'F(time, ...)' si può associare una misura alla variabile. Si andrà quindi a creare una variabile per ogni misura effettuata oltre ad una variabile incognita chiamata 'Vel\_elica'. Le variabili associate alle misure verranno usate come

variabili di output, mentre la variabile incognita costituirà la variabile di controllo del modello. Infatti, quest'ultima è la variabile che agisce sull'equazione della spinta dell'elica (forza\_motori) che permette il movimento del modello.

In Figura 3.6 viene riportato il modello Adams creato, si può notare la presenza di due forze che agiscono nell'asse verticale che sono rispettivamente la forza peso e la spinta di Archimede. La spinta di Archimede è stata approssimata in modo tale da contrastare la forza peso così da non avere movimenti sull'asse verticale.

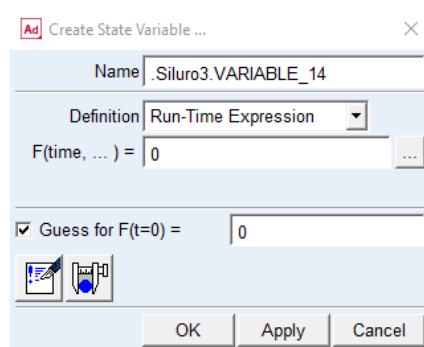


Figura 3.5: finestra di dialogo del software Adams per la creazione di una variabile

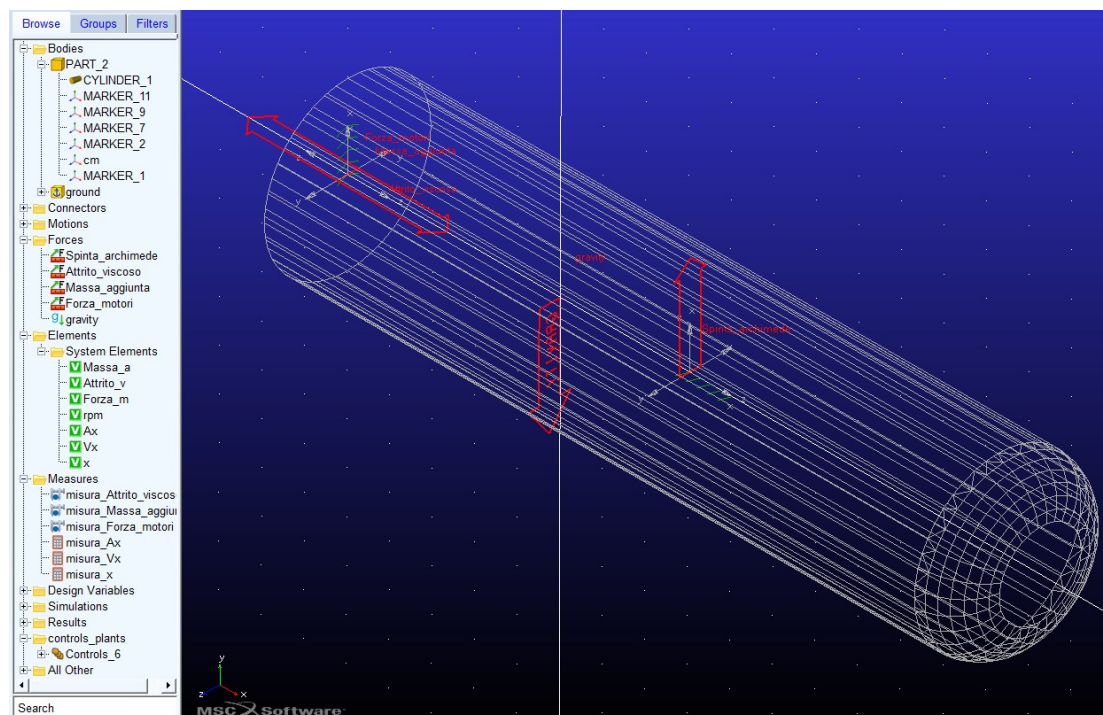


Figura 3.6: modello completo realizzato in Adams. le frecce in rosso rappresentano le forze applicate al sistema. Nel browser a sinistra sono riportati tutti gli oggetti e strumenti utilizzati per la creazione del modello



### 3.1.5 Esportazione del modello

In questa fase vengono generati due file di cui uno con estensione .m (estensione Matlab) che verranno usati in seguito per generare il blocco Adams in Simulink e permettere l'interazione tra i due software. È necessario selezionare 'Plugins' nella barra di controllo, premere su 'Controls' e di nuovo su 'Plant Export'. Nella finestra di dialogo mostrata in Figura 3.7 si dovranno inserire le variabili d'ingresso nella sezione 'Input Signal' e quelle di uscita nella sezione 'Output Signal' (vedi Tabella 3.3), mentre in 'Target Software' va selezionata la voce 'MATLAB'.

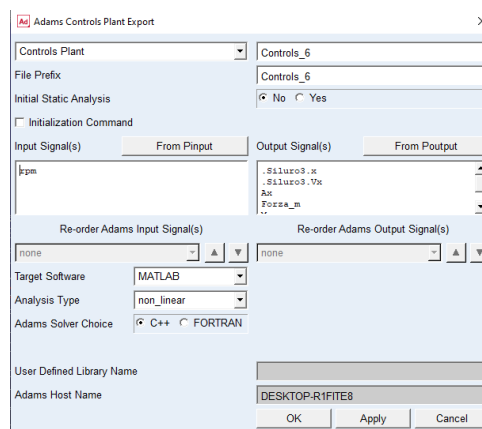


Figura 3.7: finestra di dialogo del software Adams per l'esportazione del modello in Simulink

Tabella 3.1: misure Adams

Nome	Characteristic	Descrizione
<i>misura_x</i>	<i>Translational Displacement</i>	Posizione del c.m. rispetto all'origine
<i>misura_Vx</i>	<i>Translational Velocity</i>	Velocità del c.m.
<i>misura_Ax</i>	<i>Translational Acceleration</i>	Accelerazione del c.m.
<i>misura_Forza_motori</i>	<i>Force</i>	Forza_motore
<i>misura_Attrito_viscoso</i>	<i>Force</i>	Attrito_viscoso
<i>misura_Massa_aggiunta</i>	<i>Force</i>	Massa_aggiunta
<i>misura_Diver_AV</i>	<i>Force</i>	Diver_AV
<i>misura_Diver_MA</i>	<i>Force</i>	Diver_MA
<i>misura_Bomb_AV</i>	<i>Force</i>	Bomb_AV
<i>misura_Bomb_MA</i>	<i>Force</i>	Bomb_MA

Tabella 3.2: Forze Adams

Nome	Function	Descrizione
<i>Forza_motore</i>	$(0.5*abs(VARVAL(Vel\_elica))*VARVAL(Vel\_elica))-(32*abs(VARVAL(Vel\_elica))*(1-0.2)*misura\_Vx)$	Spinta dell'elica
<i>Massa_aggiunta</i>	$1.5354*misura\_Ax$	Forza massa aggiunta scooter
<i>Attrito_viscoso</i>	$1.9074*abs(misura\_Vx)*misura\_Vx$	Forza attrito viscoso scooter
<i>Diver_AV</i>	$6,5752*abs(misura\_Vx)*misura\_Vx$	Forza attrito viscoso diver
<i>Diver_MA</i>	$11,5677*misura\_Ax$	Forza massa aggiunta diver
<i>Bomb_AV</i>	$1,9595*abs(misura\_Vx)*misura\_Vx$	Forza attrito viscoso bombola
<i>Bomb_MA</i>	$1,5927*misura\_Ax$	Forza massa aggiunta bombola

Tabella 3.3: Variabili Adams

Variabili	F(time, ...) =	Descrizione
<i>x</i>	<i>misura_x</i>	Output_Signal
<i>Vx</i>	<i>misura_Vx</i>	Output_Signal
<i>Ax</i>	<i>misura_Ax</i>	Output_Signal
<i>Forza_m</i>	<i>misura_Forza_motori</i>	Output_Signal
<i>Attrito_V</i>	<i>misura_Attrito_viscoso</i>	Output_Signal
<i>Massa_a</i>	<i>misura_Massa_aggiunta</i>	Output_Signal
<i>AV_Diver</i>	<i>misura_Diver_AV</i>	Output_Signal
<i>MA_Diver</i>	<i>misura_Diver_MA</i>	Output_Signal
<i>AV_Bomb</i>	<i>misura_Bomb_AV</i>	Output_Signal
<i>MA_Bomb</i>	<i>misura_Bomb_MA</i>	Output_Signal
<i>Vel_elica</i>	0	Input_Signal

## 3.2 Matlab e Simulink

Il Software Matlab, abbreviazione di Matrix Laboratory, è un ambiente per il calcolo numerico e l'analisi statistica scritto in C e che comprende anche l'omonimo linguaggio di programmazione creato dalla MathWorks<sup>2</sup>. Come si può intuire dal nome, questo software è specializzato nella manipolazione delle matrici e consente: la visualizzazione di funzioni e dati, l'implementazione di algoritmi, la creazione di interfacce utente e l'integrazione con altri programmi. Grazie alla sua versatilità è uno dei software più utilizzati in molteplici campi di studio, soprattutto per la quantità di funzioni base che possiede e la possibilità di espanderle attraverso dei toolbox. In particolare, per quanto riguarda il campo dell'automazione esistono molti toolbox utili per l'analisi e la progettazione di sistemi di controllo come ad esempio il 'Control System' o il 'SystemIdentification'.

Simulink è un altro software creato dalla MathWorks ed è strettamente legato a Matlab, viene usato per la modellazione, simulazione multidominio e analisi di sistemi dinamici.

L'interfaccia di Matlab si presenta come in Figura 3.8 ed è costituita principalmente da cinque finestre:

- La Command window in cui vengono eseguiti direttamente i comandi Matlab.
- Il Worksapece in cui vengono salvate tutte le variabili utilizzate.
- La Current Directory che consiste nella lista di file inclusi nella cartella corrente.
- La Command History in cui vengono visualizzati gli ultimi comandi eseguiti.
- Il Live Editor che è l'ambiente in cui è possibile creare script che combinano codice, output e testo formattato.

L'interfaccia di Simulink invece risulta essere completamente grafica e utilizza strumenti e blocchi logici precostruiti come si può vedere dalla Figura 3.9. Grazie all'integrazione di Simscape, inoltre, è possibile ampliare gli strumenti già presenti attraverso i quali si possono modellare sistemi come motori elettrici, raddrizzatori,

---

<sup>2</sup> <https://it.wikipedia.org/wiki/MATLAB>

attuatori idraulici e molto altro. Esistono inoltre diverse librerie come ‘Simscape Electrical’ che mettono già a disposizione questi sistemi.

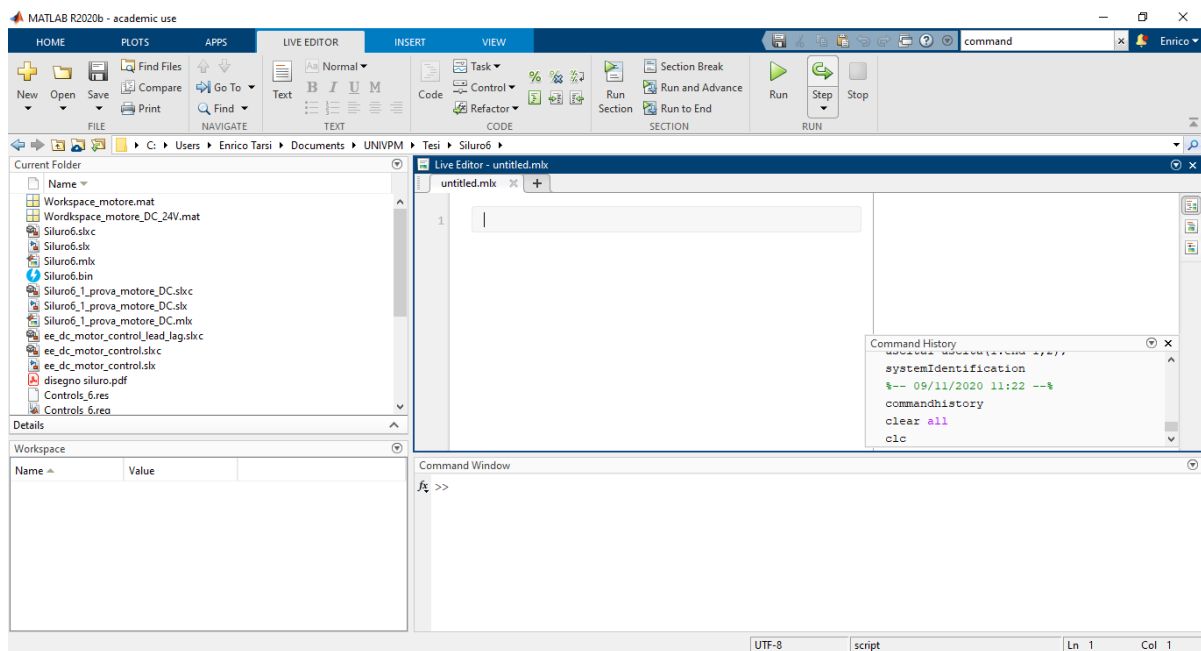


Figura 3.8: interfaccia del software Matlab

### 3.2.1 Importazione in Simulink del modello Adams

Come già spiegato, è stato necessario importare in Matlab il modello creato in Adams. Per prima cosa, accertandosi di operare nella stessa cartella di lavoro in cui è stata fatta l’esportazione da Adams, si esegue da Matlab il file con estensione .m creato precedentemente. A questo punto eseguendo il comando ‘adams\_sys’ si apre un progetto Simulink come quello mostrato in Figura 3.9.

Il blocco di interesse è quello arancione denominato ‘adams\_sub’ (che chiameremo blocco Adams) attraverso il quale sarà possibile effettuare le simulazioni del modello. Il blocco Adams prende come variabile di ingresso la velocità dell’elica e fornisce in uscita: posizione, velocità e accelerazione del centro di massa, oltre alla misura di tutte le forze agenti sul sistema.

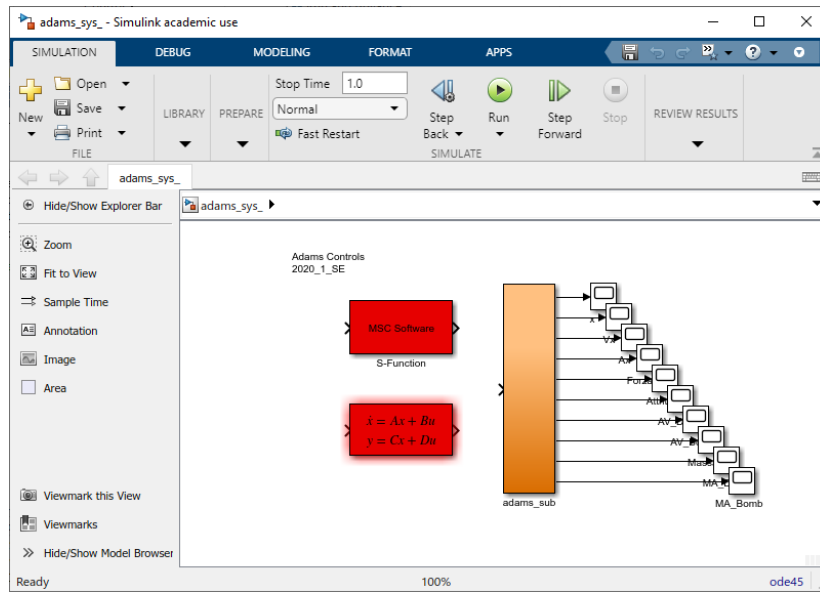


Figura 3.9: progetto Simulink aperto tramite il comando 'adams\_sys', il blocco arancione corrisponde al blocco Adams

## 4 Schemi e simulazioni

In questo capitolo vengono introdotti tre diversi schemi funzionali del sistema formato dallo scooter e dal diver che si differenziano nel blocco propulsivo ovvero dalla tipologia di motore che muove l'elica dello scooter.

Si distinguono quindi tre differenti sistemi:

- Scooter+Diver: costituito dal solo blocco Adams in cui non viene considerata la dinamica del motore, ovvero il sistema che prende in ingresso la velocità di rotazione dell'elica risponde come se il motore fosse in grado di assicurare immediatamente la velocità richiesta.
- Scooter+Diver+MotoreBLDC: costituito dal blocco Adams a cui viene aggiunto il modello di un motore BLDC.
- Scooter+Diver+MotoreDC: costituito dal blocco Adams a cui viene aggiunto il modello di un motore DC.

Per ognuno di essi viene quindi realizzato un progetto Simulink grazie al quale si possono effettuare delle simulazioni e monitorare l'andamento di tutte le uscite del blocco Adams.

Si deve inoltre precisare che per tutti e tre i sistemi non è stato modellato il comportamento del fluido sull'elica, ovvero non si è tenuto conto della coppia resistente che agisce su di essa e di come si trasferisce sull'albero motore. Pertanto, anche modellando il comportamento del motore, non si ottiene il transitorio reale della velocità di rotazione dell'elica e quindi neanche il reale comportamento dello scooter; questo sarà ben visibile dai risultati delle simulazioni.

### 4.1 Sistema Scooter+Diver

In Figura 4.1 viene mostrato il progetto Simulink realizzato per la simulazione del sistema Scooter+Diver. In questo caso, come già specificato, non viene modellata la dinamica del motore, infatti lo schema funzionale è costituito dal solo blocco Adams

oltre che da un blocco 'Signal builder' per generare il segnale di ingresso e uno 'Scope' per mostrare in grafico l'andamento nel tempo delle uscite.

In ingresso allo scope vengono riportate: la posizione ( $x$ ), la velocità ( $V_x$ ), l'accelerazione ( $A_x$ ), la forza totale di attrito (data dalla somma di Attrito\_V, AV\_Diver e AV\_Bomb) e la forza totale di massa aggiunta (data dalla somma di Massa\_a, MA\_Diver, e MA\_Bomb) del blocco Adams, oltre alla somma di tutte le forze.

Si vuole effettuare una simulazione per verificare la risposta al gradino del sistema, ricordando che in ingresso al blocco Adams si deve applicare la velocità di rotazione dell'elica espressa in rad/s. Quindi attraverso il 'Signal builder' si genera un gradino di ampiezza 83 (velocità massima di rotazione dell'elica in rad/s) e si effettua una simulazione di 2s con un tempo di campionamento pari a 0.005s. Terminata la simulazione, dallo Scope si ottengono i grafici riportati in Figura 4.2.

Si pone particolare attenzione all'andamento della spinta dell'elica, che si ricorda essere la forza che mette in movimento l'intero sistema. Essa in corrispondenza del gradino presenta un picco di 3250N che non può corrispondere ad un valore reale in quanto troppo elevato. Tale comportamento è riconducibile oltre dall'assenza di modellazione del motore anche dall'assenza di modellazione del comportamento del fluido sull'elica. Si osserva infatti che una spinta così elevata causa a sua volta un'accelerazione molto elevata di circa  $25 \text{ m/s}^2$  e di conseguenza un tempo di salita della velocità ' $V_x$ ' (tempo che impiega a passare dal 10% al 90% del valore finale) molto breve di 128ms.

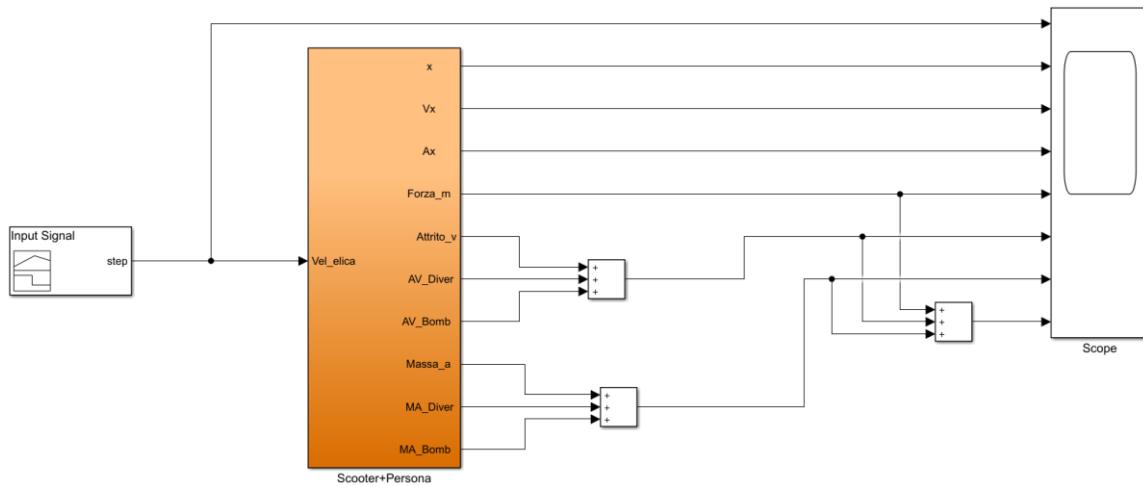


Figura 4.1: schema funzionale del sistema Scooter+Diver

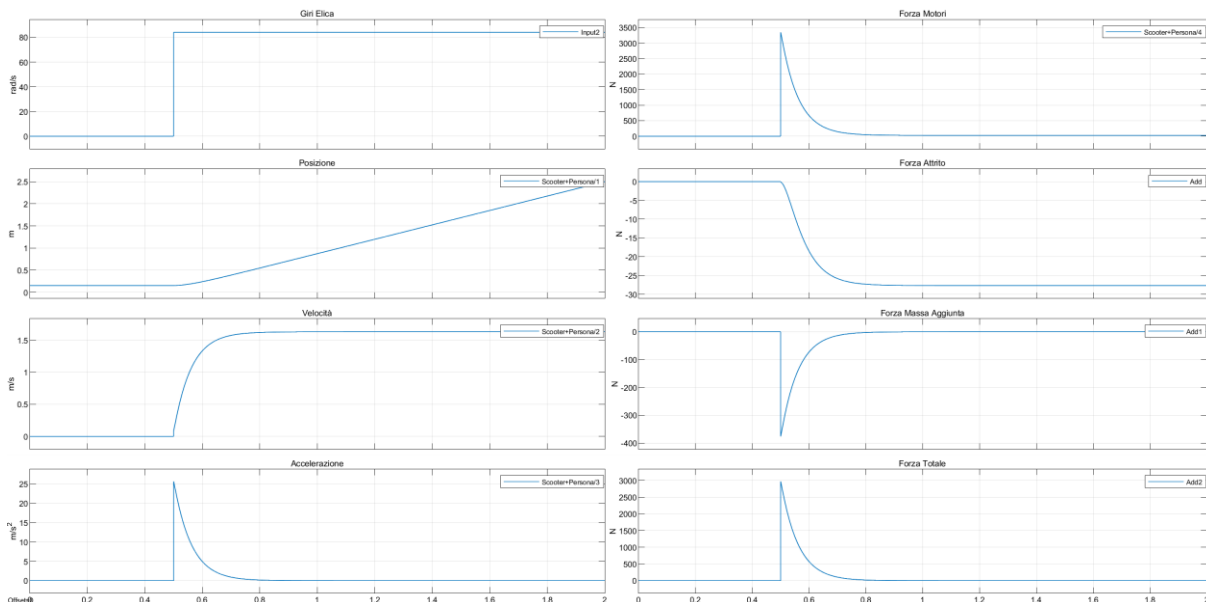


Figura 4.2: risultato della simulazione eseguita sul sistema raffigurato in Figura 4.1, dove si è applicato in ingresso un gradino di ampiezza 83

## 4.2 Sistema Scooter+Diver+MotoreBLDC

Per il sistema Scooter+Diver+MotoreBLDC, il cui schema è mostrato in Figura 4.3, viene aggiunto il modello Simulink di un motore BLDC controllato sulla velocità. Quest'ultimo è reso disponibile dagli esempi di Matlab utilizzando il comando "opensystem('scdbldcspedcontrol')", ed è costituito da un motore BLDC trifase accoppiato con un convertitore buck e alimentato tramite un inverter trifase. Il



convertitore buck e l'inverter sono modellati rispettivamente con transistor MOSFET e IGBT anziché con interruttori ideali in modo che le resistenze e le caratteristiche del dispositivo siano rappresentate correttamente. Il sottosistema 'control' utilizza una struttura in cascata basata su PI con un anello esterno di controllo della velocità e un anello interno di controllo della tensione<sup>3</sup>. Il motore controllato prende in ingresso la velocità di rotazione di riferimento espressa in rpm e con il tag 'omega' fornisce la velocità reale espressa in rad/s. In questo sistema quindi la velocità di rotazione desiderata generata dal 'Signal builder' viene posta in ingresso al motore e non più direttamente al blocco Adams, il quale a sua volta prende in ingresso la velocità reale fornita dal modello del motore. In questo modo viene considerata nella velocità di rotazione dell'elica la dinamica aggiunta dalla risposta del motore al valore di riferimento.

Per verificare la risposta al gradino del sistema in questo caso è stato generato un gradino di ampiezza 800 (velocità massima di rotazione in rpm) ed è stata eseguita una simulazione della durata di 2s con un tempo di campionamento di 1e-05s. A simulazione completata dallo scope vengono visualizzati i grafici mostrati in Figura 4.4.

Nel primo grafico viene riportato in rosso l'andamento reale della velocità di rotazione del motore e in blu il segnale di riferimento. Si può notare che la risposta del motore BLDC è molto rapida e raggiunge il valore di riferimento quasi istantaneamente. Infatti, anche in questo si ha un comportamento simile a quello precedente dove la spinta dell'elica (Forza motori) raggiunge un valore di circa 3000N; si osserva comunque una piccola riduzione dovuta all'aggiunta della dinamica del motore. Il tempo di risposta della velocità 'Vx' risulta essere di 135ms. Si può affermare che la dinamica del motore BLDC preso in questione permette delle prestazioni prossime a quelle ideali, tenendo in considerazione che per entrambi i sistemi è stata considerata la coppia resistente del fluido sull'elica.

---

<sup>3</sup><https://it.mathworks.com/help/slcontrol/ug/bldc-motor-speed-control-with-cascade-pi-controllers.html>

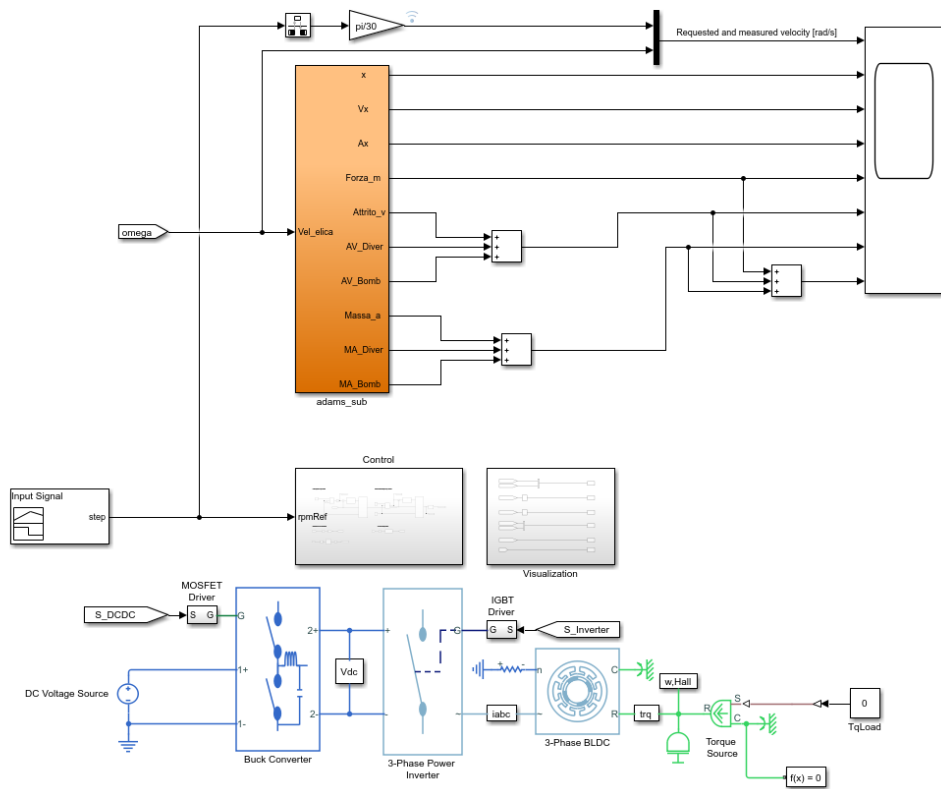


Figura 4.3: schema funzionale del sistema Scooter+Diver+MotoreBLDC

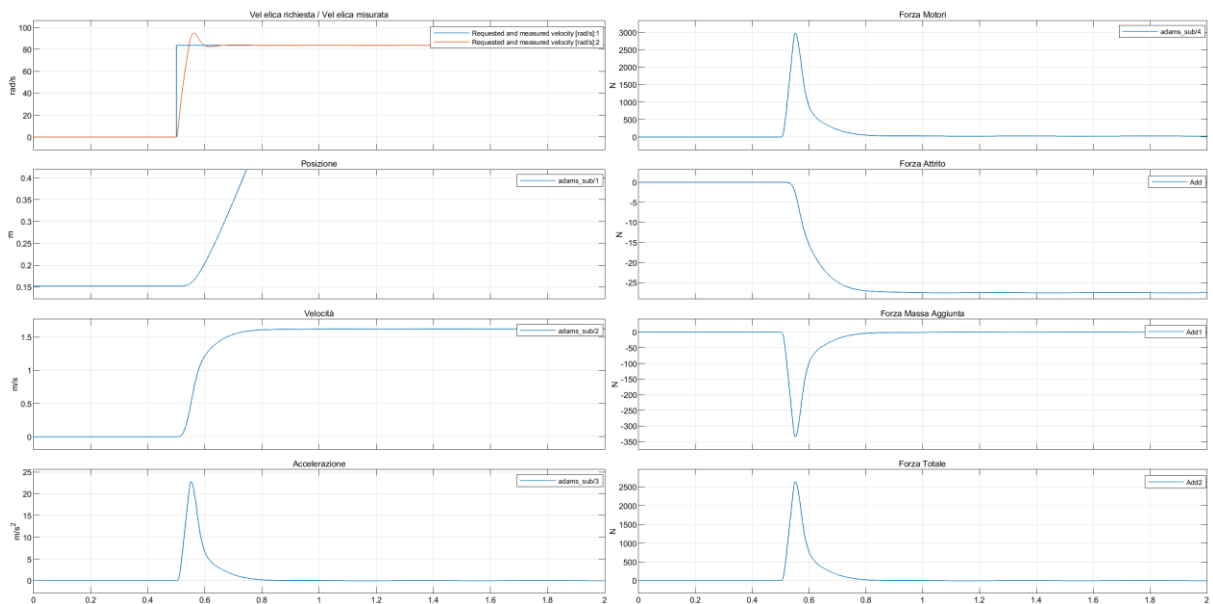


Figura 4.4 risultato della simulazione eseguita sul sistema raffigurato in Figura 4.3, dove si è applicato in ingresso un gradino di ampiezza 800

### 4.3 Sistema Scooter+Diver+MotoreDC

Per il sistema Scooter+Diver+MotoreDC viene aggiunto un blocco propulsivo costituito da un motore DC controllato sulla velocità mostrato in Figura 4.5. Anch'esso è reso disponibile dagli esempi di Matlab usando il comando "opensystem('ee\_dc\_motor\_control')" ed è costituito da un motore DC della libreria Simscape Electrical. Il motore è controllato in cascata da due controllori PI: uno per il loop esterno di controllo della velocità e uno per il loop interno di controllo della corrente. A valle dei due PI è presente il generatore PWM che comanda il chopper a quattro quadranti realizzato mediante transistor IGBT, il quale regola a sua volta la corrente di armatura sul motore<sup>4</sup>. In questo caso il motore controllato prende in ingresso la velocità di rotazione di riferimento e fornisce, tramite il tag 'w', una misura della velocità di rotazione reale espressa in rpm. Quest'ultima è stata applicata in ingresso al blocco Adams effettuandone prima la conversione in rad/s.

È stata realizzata una simulazione di 5s con uno tempo di campionamento di 1e-05s in cui in ingresso tramite il 'Signal builder' è stato applicato un gradino di ampiezza 800 (velocità di rotazione massima in rpm). Terminata la simulazione si ottengono i grafici mostrati in Figura 4.6.

Come ci si poteva aspettare si osserva nel primo grafico che la risposta del motore DC è molto più lenta rispetto a quella del motore BLDC; infatti la spinta dell'elica raggiunge il suo valore massimo a 460N che risulta essere di gran lunga inferiore rispetto ai 3000N del caso precedente. Si misura inoltre un tempo di salita della velocità 'Vx' pari a 850ms. Pur assumendo valori conformi a quelli reali si deve ricordare che anche in questo caso il contributo della coppia resistente del fluido sull'elica non viene considerata, quindi in condizioni reali si avrebbe avuto un comportamento ancora più lento.

---

<sup>4</sup> <https://it.mathworks.com/help/physmod/sps/ug/dc-motor-control.html>

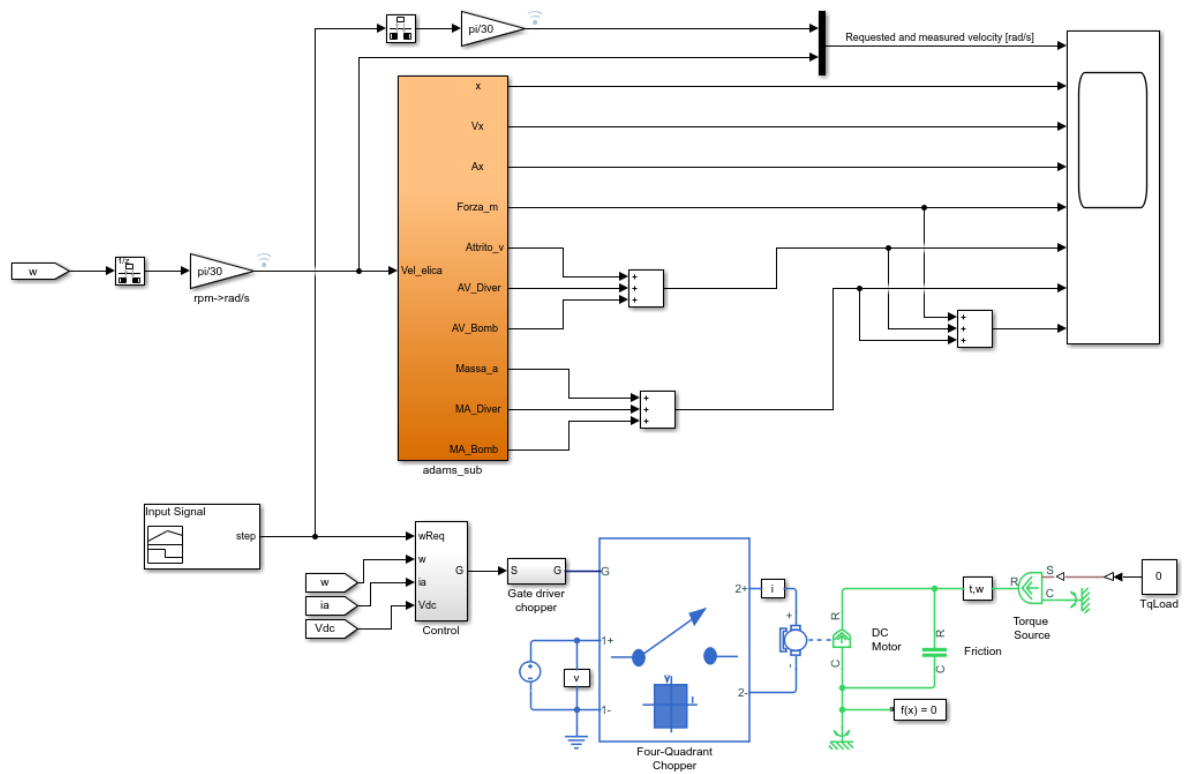


Figura 4.5 schema funzionale del sistema Scooter+Diver+MotoreDC

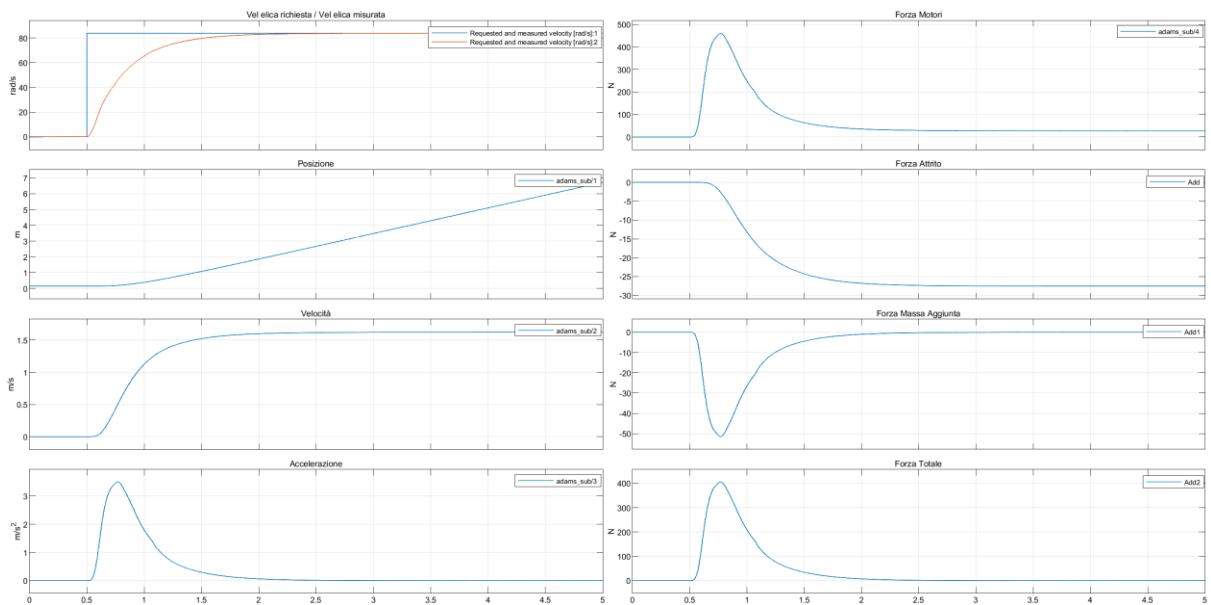


Figura 4.6: risultato della simulazione eseguita sul sistema raffigurato in Figura 4.5, dove si è applicato in ingresso un gradino di ampiezza 800

## 5 Identificazione

In questo capitolo vengono introdotti i concetti principali dell'identificazione parametrica e i suoi aspetti teorici. In particolare, verrà affrontato il tema dell'identificazione attraverso un approccio *black box*, ovvero ipotizzando di non conoscere il sistema che si vuole modellare. Questa tecnica verrà usata in seguito per trovare un modello parametrico per il sistema Scooter+Diver e per il sistema Scooter+Diver+MotoreDC.

Il problema dell'identificazione consiste nel determinare un modello che spieghi adeguatamente l'andamento osservato dei dati. Questi dati si riferiscono a valori assunti da variabili del sistema negli istanti di tempo  $[0, T]$ .

L'approccio *black box* quindi consiste nel dedurre il modello direttamente dai dati raccolti, ovvero dalle coppie ingresso-uscita, astraendo completamente dal contesto del problema fisico. Si può suddividere la fase dell'identificazione in quattro passaggi fondamentali:

1. Raccolta dei dati
2. Individuazione della famiglia di modelli da utilizzare
3. Scelta del miglior modello nella famiglia
4. Validazione del modello

### 5.1 Raccolta dei dati

La raccolta dei dati risulta essere un passaggio fondamentale e particolarmente delicato per effettuare una buona modellazione. Considerando infatti il sistema una scatola nera, i dati che si raccolgono sono l'unica conoscenza a disposizione per la determinazione del modello. Per ottenere delle coppie ingresso-uscita è necessario porre un segnale in ingresso al sistema (ingresso) e osservarne la risposta (uscita), si cerca quindi di stimolare tutta la dinamica del sistema applicando un segnale che sia persistentemente eccitante e in grado di raggiungere tutti i modi naturali che sono raggiungibili. Viene usato un rumore bianco che consiste in un segnale caratterizzato

dall'assenza di periodicità nel tempo e da ampiezza costante su tutto lo spettro di frequenze, quindi si tratta di un segnale totalmente casuale e imprevedibile ed avendo una banda infinita garantisce l'eccitazione di tutti gli autovalori. Si intuisce però che nella realtà è impossibile creare un segnale con queste caratteristiche in quanto nessun segnale può generare uno spettro uniforme per tutte le frequenze. Per tutti i casi reali tuttavia è sufficiente l'applicazione di rumori bianchi limitati, ovvero di segnali che hanno le caratteristiche di un rumore bianco solo per un determinato range di frequenze.

È necessario quindi trovare un modo per determinare quando un segnale si può considerare rumore bianco partendo dalla sua definizione: un segnale si definisce rumore bianco se il suo vettore medio è nullo e se la sua matrice di autocorrelazione è un multiplo della matrice d'identità

$$\mu_w = E\{w\} = 0 \quad (5.1)$$

$$R_{ww} = E\{ww^T\} = \sigma^2 I \quad (5.2)$$

La matrice di autocorrelazione  $R_{ww}$  viene usata per fornire una misura della regolarità del segnale, infatti la funzione di autocorrelazione mette a confronto un segnale con sé stesso ma ritardato di un certo tempo. La larghezza di banda del segnale dipende direttamente dalla funzione di autocorrelazione essendo che la densità spettrale di un segnale corrisponde alla trasformata di Fourier della funzione di autocorrelazione del segnale stesso.

A fronte di queste definizioni si possono implementare diversi test per verificare se un segnale può considerarsi un rumore bianco, in particolare verrà analizzato e sviluppato tramite Matlab il test di Anderson. Esso sfrutta il fatto che se si considera  $w(t)$  un rumore bianco allora la sua covarianza campionaria  $\rho(\tau)$  sarà nulla per valori di  $\tau$  diversi da zero. Si costruisce allora uno stimatore della covarianza campionaria normalizzata:

$$\hat{r}(\tau) = \frac{\hat{R}(\tau)}{\hat{R}(0)} \quad (5.3)$$

Dove la stima della covarianza è data da:

$$\hat{R}(\tau) = \frac{1}{N} \sum_{t=1}^{N-t} w(t+\tau) * w(t) \quad (5.4)$$

in cui  $N$  corrisponde al ritardo massimo tra i campioni nel calcolo della stima; quindi  $\hat{R}(\tau)$  sarà calcolato per  $\tau \in [0, N]$ . Si può dimostrare che se  $w(t)$  è un rumore bianco allora la quantità  $\sqrt{N}\hat{r}(\tau)$  è distribuita come una gaussiana standard. Quindi fissando un valore di confidenza  $\alpha$  che rappresenta l'area della gaussiana di validazione (del test di bianchezza) e ricavando il valore  $\beta$  corrispondente, si verifica che la probabilità per cui la quantità  $\sqrt{N}\hat{r}(\tau)$  ricade al di fuori dall'intervallo  $[-\beta, \beta]$  sia minore di  $\alpha$ . A seguito di quanto affermato, indicando con 'c' il numero dei valori per cui la quantità  $\sqrt{N}\hat{r}(\tau)$  è maggiore in modulo di  $\beta$  si avrà che se la quantità  $\frac{c}{N}$  è minore del valore di confidenza  $\alpha$  allora  $w(t)$  può essere considerato un rumore bianco.

## 5.2 Individuazione della famiglia di modelli

Per individuare le famiglie dei modelli viene prima introdotto il sistema di controllo rappresentato come in Figura 5.1. Si può notare che oltre alla parte deterministica descritta dalla funzione di trasferimento  $G(z)$ , che ha come ingresso un segnale esogeno di controllo  $u(t)$ , è presente in sovrapposizione anche una parte aleatoria descritta dalla funzione di trasferimento  $W(z)$  che ha come ingresso un rumore bianco. Quest'ultima fornisce in uscita un segnale di disturbo  $y_s(t)$  detta componente stocastica della risposta che si va a sommare alla componente deterministica  $y_d(t)$ .

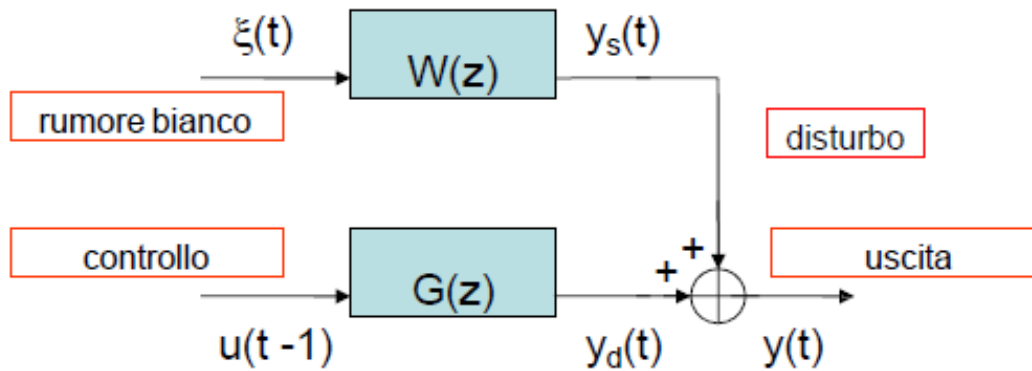


Figura 5.1 schema di un sistema di controllo

Si può descrivere l'uscita del sistema di controllo, data dalla somma della componente deterministica e quella aleatoria, come segue:

$$y(t) = G(z)u(t - 1) + W(z)\xi(t) \quad (5.5)$$

Si nota che l'uscita dipende dall'ingresso ad un istante di tempo precedente, in quanto viene tenuto in considerazione il tempo di elaborazione dei dati.

Introducendo l'operatore di traslazione  $z^{-1}$  definito da:

$$z^{-1}(f(t)) = f(t - 1) \quad (5.6)$$

e ipotizzando che:

- $G(z) \times z^{-1}$  sia strettamente propria e  $W(z)$  sia propria.
- $G(z)$  e  $W(z)$  abbiano denominatori monici e  $W(z)$  abbia numeratore monico.
- $W(z)$  abbia poli con modulo minore di uno.
- $W(z)$  abbia zeri con modulo minore o uguale a uno.

allora la quantità deterministica  $y_d(t)$  può essere riscritta come equazione alle differenze ponendo

$$G(z) = \frac{B(z)}{A(z)} \quad (5.7)$$

dove il numeratore e il denominatore sono per ipotesi:



$$B(z) = b_1 z^n + b_2 z^{n-1} + \dots + b_{n+1} \quad (5.8)$$

$$A(z) = z^n - a_1 z^{n-1} - \dots - a_n \quad (5.9)$$

Moltiplicando ora la (5.8) e la (5.9) per  $z^{-1}$  e razionalizzando si può riscrivere  $y_d(t) = G(z)u(t-1)$  come:

$$(1 - a_1 z^{-1} - \dots - a_n z^{-n})y_d(t) = (b_1 + b_2 z^{-1} + \dots + b_{n+1} z^{-n})u(t-1) \quad (5.10)$$

dando luogo all'equazione alle differenze:

$$y_d(t) = a_1 y(t-1) + \dots + a_n y(t-n) + b_1 u(t-1) + \dots + b_{n+1} u(t-n-1) \quad (5.11)$$

da cui si osserva che essa dipende da valori ad istanti precedenti di  $y$  e  $u$ .

Le famiglie più interessanti e significative da tenere in considerazione sono quelle dei modelli ad errore di equazione, dove l'uscita  $y(t)$  consiste della componente deterministica  $y_d(t)$  sommata ad un residuo di equazione:

$$\begin{aligned} y(t) = & a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) + \\ & + b_1 u(t-1) + b_2 u(t-2) + \dots + b_{nb} u(t-nb) + \\ & + w(t) \end{aligned} \quad (5.12)$$

L'uscita è determinata da  $na$  parametri che modellano il comportamento autoregressivo e  $nb$  parametri che modellano il comportamento della variabile esogena, inoltre è presente un residuo di equazione  $w(t)$  che rappresenta il disturbo non modellato agente sul sistema.

Le varie famiglie si differenziano a seconda della presenza della parte esogena e dalla caratterizzazione del residuo di equazione. In questo elaborato verranno prese in considerazione la famiglia ARX e la famiglia ARMAX.

Nei modelli della famiglia ARX l'uscita  $y(t)$  è esprimibile come:

$$\begin{aligned} y(t) = & a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) + \\ & + b_1 u(t-1) + b_2 u(t-2) + \dots + b_{nb} u(t-nb) + \\ & + \xi(t) \end{aligned} \quad (5.13)$$

Dove il residuo d'equazione  $w(t)$  corrisponde ad un rumore bianco.

Per ricavare l'equazione del modello ARX si considerano i polinomi costituenti del modello:

$$A(z) = 1 - a_1z^{-1} - a_2z^{-2} - \dots - a_{na}z^{-na} \quad (5.14)$$

$$B(z) = b_1 + b_2z^{-1} - \dots - b_{nb}z^{-nb+1} \quad (5.15)$$

ed effettuando operazioni di traslazione come visto in precedenza si ottiene l'equazione alle differenze riscritta in forma matriciale:

$$y(t) = \frac{B(z)}{A(z)}u(t-1) + \frac{1}{A(z)}\xi(t). \quad (5.16)$$

I vari modelli della famiglia ARX si differenziano tra loro per gli ordini della parte autoregressiva e della parte esogena e tali ordini corrispondono alla memoria del sistema.

In conclusione, un modello della famiglia ARX viene descritto completamente dal vettore dei parametri:

$$\theta = [a_1, a_2, \dots, a_{na}, b_1, b_2, \dots, b_{nb}] \quad (5.17)$$

in cui sono presenti  $na$  parametri per la parte autoregressiva e  $nb$  parametri per la parte esogena, per cui il modello viene indicato come ARX( $na, nb$ ).

Per quanto riguarda i modelli della famiglia ARMAX invece, l'uscita  $y(t)$  è esprimibile come segue:

$$\begin{aligned} y(t) = & a_1y(t-1) + a_2y(t-2) + \dots + a_{na}y(t-na) + \\ & + b_1u(t-1) + b_2u(t-2) + \dots + b_{nb}u(t-nb) + \\ & + \xi(t) + c_1\xi(t-1) + \dots + c_{nc}\xi(t-nc) \end{aligned} \quad (5.18)$$

Si nota che in questo caso il residuo di equazione corrisponde alla combinazione lineare dei valori di un rumore bianco  $\xi(t)$  campionati sull'intervallo  $[t-nc, t]$  in cui l'azione di una dinamica su  $\xi(t)$  lo rende un rumore a media mobile.

Si considerano in questo caso i polinomi:

$$A(z) = 1 - a_1z^{-1} - a_2z^{-2} - \dots - a_{na}z^{-na} \quad (5.19)$$

$$B(z) = b_1 + b_2z^{-1} - \dots - b_{nb}z^{-nb+1} \quad (5.20)$$

$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{nc} z^{-nc} \quad (5.21)$$

Da cui si ottiene l'equazione alle differenze in forma matriciale:

$$y(t) = \frac{B(z)}{A(z)} u(t-1) + \frac{C(z)}{A(z)} \xi(t) \quad (5.22)$$

Un modello della famiglia ARMAX viene descritto completamente dal vettore dei parametri:

$$\theta = [a_1, a_2, \dots, a_{na}, b_1, b_2, \dots, b_{nb}, c_1, \dots, c_{nc}] \quad (5.23)$$

A differenza della famiglia ARX, oltre agli  $na$  parametri della parte autoregressiva e agli  $nb$  parametri della parte esogena, sono presenti anche  $nc$  parametri della parte a media mobile.

In entrambe le famiglie è possibile inoltre definire il ritardo del modello: se  $b_k$  è il primo coefficiente della parte esogena diverso da zero si dice infatti che il modello ha ritardo  $k$ . In Matlab infatti, quando si definisce un modello ARX o un modello ARMAX oltre agli ordini  $na$ ,  $nb$  e  $nc$ , viene richiesto un ordine  $nk$  che corrisponde proprio al ritardo del modello.

### 5.3 Scelta del miglior modello della famiglia

Per poter determinare se un modello di una determinata famiglia risulta essere un buon modello è necessario riuscire a mettere a confronto il suo comportamento con quello del sistema reale. Verrà indicato con  $M$  la famiglia scelta e con  $M(\theta)$  il modello che si vuole confrontare. È importante capire che il modello  $M(\theta)$  tenendo conto anche dei rumori sovrapposti fornisce un'uscita  $y_\theta(t)$  di tipo stocastica, mentre l'uscita misurata del sistema reale  $y(t)$  è di tipo deterministico; risulta chiaro quindi che il confronto tra le due grandezze non è realizzabile. Perciò ad ogni modello  $M(\theta)$  si deve associare un sistema  $\hat{M}(\theta)$  detto predittore che, elaborando i dati di I/O di  $M(\theta)$  sull'intervallo  $[0, t-1]$  fornisce una predizione dell'uscita  $y_\theta(t)$  all'istante  $t$ . l'uscita del

predittore viene indicata con  $\hat{y}(t)$  ed è confrontabile con  $y(t)$  essendo anch'essa di tipo deterministico.

Per determinare l'equazione del predittore si considera il modello  $M(\theta)$  descritto dalla (5.5) e andando a sommare e sottrarre la quantità  $W(z)y(t)$  si ottiene:

$$W(z)y(t) = -y(t) + W(z)y(t) + G(z)u(t-1) + W(z)\xi(t) \quad (5.24)$$

Da cui l'equazione del sistema può essere riscritta come:

$$y(t) = \left[1 - \frac{1}{W(z)}\right]y(t) + \frac{G(z)}{W(z)}u(t-1) + \xi(t) \quad (5.25)$$

Essendo inoltre  $W(z)$  per ipotesi un rapporto di polinomi monici di ugual grado, può essere espressa come:

$$\left[1 - \frac{1}{W(z)}\right]y(t) = -\alpha_1y(t-1) - \alpha_2y(t-2) - \dots \quad (5.26)$$

e risulta quindi che  $y(t)$  è una combinazione di valori di  $y$  e  $u$  ad istanti precedenti a  $t$  ed addizionata ad un rumore bianco  $\xi(t)$ , il cui valore atteso è nullo. Perciò, noti tali valori la miglior predizione di  $y(t)$  sarà sicuramente data da:

$$\hat{y}(t) = \left[1 - \frac{1}{W(z)}\right]y(t) + \frac{G(z)}{W(z)}u(t-1) \quad (5.27)$$

A questo punto l'errore di predizione stabilito dalla differenza dell'uscita del sistema con quella del predittore:

$$\varepsilon_\theta(t) = y(t) - \hat{y}_\theta(t) \quad (5.28)$$

deve risultare essere un rumore bianco. Questo perché si sta ipotizzando che il modello sia giusto rispetto al comportamento del sistema e si suppone inoltre che ci sia un rumore bianco sovrapposto ad esso, quindi l'uscita del predittore non deve essere uguale a quella del sistema ma deve differire di un segnale imprevedibile, perché se

così non fosse (quindi se non corrispondesse ad un rumore bianco) sarebbe entrato a far parte dell'equazione del predittore anch'esso.

Calcolando inoltre l'errore quadratico medio dato da:

$$J_N(\theta) = \frac{1}{N} \sum_{\tau=0}^N \varepsilon_{\theta}^2(\tau) \quad (5.29)$$

dove  $N$  è il numero di campioni prelevati dal sistema reale, si avrà che il miglior modello per un dato sistema  $S$ , tra tutti quelli della famiglia  $M$ , è quello per il quale  $J_N(\theta)$  risulta essere minimo. Come si può intuire  $J_N(\theta)$  non fornisce una misura della bontà del modello, ma solamente di come  $M(\theta)$  aderisce ai dati, risultando quindi un metodo non oggettivo. Per di più al crescere della complessità del modello,  $J_N(\theta)$  tenderà sempre a diminuire rischiando di incorrere in quello che viene chiamato sovrapparametrizzazione; in tal caso modellando il sistema con una complessità eccessiva si rischia di rappresentare anche parte del disturbo.

Uno strumento utile per capire la complessità del sistema è la tecnica di valutazione a priori dell'ordine: se si considera la matrice tecnica, costruita con i dati sperimentali mostrata in (5.30) e ipotizzando che il sistema sia dimensionato come  $n_a=n_b=n$

$$M(n) = \begin{bmatrix} y(n+1) & y(n) & y(n-1) & \dots & y(1) & u(n) & u(n-1) & \dots & u(1) \\ y(n+2) & y(n+1) & y(n) & \dots & y(2) & u(n+1) & u(n) & \dots & u(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y(N) & y(N-1) & y(N-2) & \dots & y(N-n) & u(N-1) & u(N-2) & \dots & u(N-n) \end{bmatrix} \quad (5.30)$$

allora  $n$  sarà il più piccolo intero per il quale le colonne di  $M(n)$  sono linearmente dipendenti, quindi per valori di  $k < n$  le colonne risulteranno invece linearmente indipendenti. Costruendo la matrice tecnica  $M(k)$  partendo da  $k=1$  e calcolando il rango della matrice quadrata data da  $M(k)^T M(k)$  si verifica la singolarità e quindi anche la lineare dipendenza delle colonne. Si aumenta  $k$  fino ad ottenere una matrice  $M(k)^T M(k)$  che risulti non singolare e a questo punto è chiaro che il primo valore di  $k$  per cui  $M(k)^T M(k)$  è non singolare corrisponde all'ordine  $n$  del sistema. Poiché il modello non verifica mai in maniera esatta il sistema ma viene sempre introdotto un

errore di equazione  $w(t)$ , la singolarità della matrice non si potrà mai verificare. È possibile però definire un'approssimazione di singolarità valutando gli autovalori di  $M(k)^T M(k)$  e si dirà che: la matrice è prossima alla condizione di singolarità quanto più il suo autovalore minimo  $|\lambda_{min,k}|$  è in valore assoluto prossimo allo zero. È ovvio che anche in questo caso al crescere di  $k$  e quindi della complessità,  $|\lambda_{min,k}|$  decresce e tende a zero; quindi per poter mediare tra un'elevata aderenza e bassa complessità ha senso scegliere il  $k$  per cui si verifica il massimo salto tra  $|\lambda_{min,k}|$  e  $|\lambda_{min,k+1}|$ .

Si vedrà nel capitolo successivo come implementare tale algoritmo.

## 5.4 Validazione

Si è già specificato che valutare il modello dal solo valore di  $J_N(\theta)$  non fornisce una misura assoluta della bontà, soprattutto a causa della sovrapparametrizzazione. Per ovviare a questo problema, oltre al calcolo dell'ordine a priori, viene utilizzata la tecnica della cross validazione. Questa tecnica consiste nel dividere in due parti uguali il set di dati che si ha a disposizione: una delle due parti viene usata per identificare il modello, mentre l'altra viene fornita in ingresso al modello per valutare la bianchezza dell'errore di predizione. Questa tecnica sfrutta il fatto che su due set di dati differenti il disturbo risulta essere sicuramente scorrelato, quindi fornendo in ingresso al modello un set di dati diverso da quello utilizzato per l'identificazione e si verifica che l'errore di predizione continua ad essere un rumore bianco, significa che si è modellato solo il sistema e non il disturbo. Per fare una validazione completa esistono altri criteri oggettivi forniti da alcuni indici che dipendono da  $J_N(\theta)$  e che sono calcolati in modo da penalizzare l'aumento della complessità. Ne vengono mostrati qui di seguito alcuni:

- $FPE = \frac{N+n}{N-n} J_N(\theta)$
- $AIC = \frac{2}{N} n + \log(J_N(\theta))$
- $MDL = \frac{n}{N} \log(N) + \log(J_N(\theta))$

## 6 Identificazione dei Sistemi

In questo capitolo vengono applicati i concetti visti nel capitolo precedente. Si vuole infatti trovare un modello parametrico che descrive il sistema Scooter+Diver e uno che descrive il sistema Scooter+Diver+MotoreDC attraverso un approccio black box, in cui i dati vengono prelevati direttamente dal modello creato in Adams attraverso una simulazione in Simulink.

### 6.1 Identificazione Sistema Scooter+Diver

Vengono qui descritti i procedimenti e gli strumenti utilizzati per effettuare l'identificazione del sistema Scooter+Diver. Prima di tutto si procede con la raccolta dei dati usando Matlab per la generazione del rumore bianco e Simulink per la simulazione del sistema. Ottenute le coppie ingresso-uscita viene calcolato l'ordine a priori per fare una prima valutazione sulla complessità del sistema e in seguito, sfruttando il toolbox 'systemIdentification', si stimano prima i modelli della famiglia ARX e poi quelli della famiglia ARMAX. A questo punto valutando i modelli ottenuti dalle singole stime si sceglie il miglior modello per il sistema.

#### 6.1.1 Raccolta dei dati

Per la raccolta dei dati del sistema Scooter+Diver è stato creato un nuovo blocco Adams in cui viene riportato in uscita solo il valore della velocità 'Vx'. Le coppie ingresso-uscita necessarie per l'identificazione sono infatti costituite dalla velocità dell'elica (Vel\_elica) e dalla velocità dello scooter (Vx). La simulazione in Simulink avrà una durata di 100s in cui i dati vengono acquisiti ogni 0.01s così da ottenere un totale di coppie ingresso-uscite pari a 10000.

Per la generazione del rumore bianco è necessario assicurarsi che esso abbia un tempo di variazione coerente al sistema in esame. Questo perché se venisse applicato un ingresso che varia molto velocemente ad un sistema con tempo di risposta molto lento, il sistema non sarebbe in grado di reagire alle variazioni dell'ingresso; così facendo le

informazioni inerenti alla frequenza di lavoro del sistema non verrebbero raccolte causando quindi un effetto controproducente. A tal proposito si è deciso di applicare un tempo di variazione del rumore bianco inferiore di una decade al tempo di risposta del sistema. Si è osservato che il tempo di risposta per la velocità 'Vx' del sistema Scooter+Diver è di 128ms, quindi il rumore bianco verrà fatto variare ogni 0.01s. Da questo ne deriva anche il tempo di campionamento del segnale di uscita, assicurando il rispetto del teorema del campionamento di Nyquist-Shannon.

In Figura 6.1 viene mostrata la parte di codice necessaria per generare il rumore bianco: si utilizza il comando 'rand(1,n)' per generare un vettore di n valori casuali nell'intervallo [0, 1] chiamato 'rumore\_bianco'. Essendo per definizione con media nulla, viene calcolata con il comando 'mean' la media del vettore che viene sottratta al vettore stesso, ottenendo così valori casuali nell'intervallo [-0.5, 0.5]. Tale procedimento non assicura però la bianchezza del segnale, per questo la generazione del vettore viene posta all'interno di un ciclo while dal quale si può uscire solo una volta superato il test di Anderson che ne stabilisce la bianchezza.

```
clear all;
n=10000;
N=10000;
alfa=0.05;
test=0;
while(test~=1)
    rumore_bianco=rand(1,n);
    rumore_bianco=rumore_bianco-mean(rumore_bianco);
    test=anderson2(rumore_bianco,alfa,N)
end
```

Figura 6.1: codice Matlab per generare il rumore bianco

L'implementazione del test di Anderson è riportata in Figura 6.2 per la quale è stata definita una funzione che prende in ingresso tre quantità: il segnale in esame, il valore di confidenza  $\alpha$  e il ritardo massimo tra i campioni nel calcolo della stima della covarianza. La funzione, tramite la variabile 'test', fornisce in uscita il valore '1' se il test è stato superato permettendo così anche l'uscita dal ciclo while, altrimenti fornisce



il valore '0' causando invece un nuovo ciclo e quindi una nuova generazione del vettore.

Nella prima parte di codice con la funzione 'alfa2beta' viene calcolato il  $\beta$  corrispondente al valore di confidenza 'alfa' e viene salvato nella variabile denominata 'beta'. Viene inizializzato poi un vettore 'R' usato per ospitare i valori della stima della covarianza. Mediante un ciclo for viene calcolato 'R' e viene salvato nella variabile 'rho' il suo valore normalizzato. All'interno dello stesso ciclo for quando la quantità ' $\sqrt{N}\rho(\tau + 1)$ ' risulta maggiore in modulo a beta viene incrementata una variabile chiamata 'c'. Infine, nella parte di verifica, la variabile test viene posta ad '1' se la quantità 'c/N' è minore di 'alfa', altrimenti viene posta a '0'.

#### Test di Anderson

```
function test=anderson2(in,alfa,tau_max)
in=in(:)';
R=zeros(N+1,1);
N=tau_max;
beta=alfa2bet(alfa,0.5e-9);
c=0;

for tau=0:N
R(tau+1)=(1/N)*in(1:N-tau)*in(1+tau:N)';
rho(tau+1)=R(tau+1)/R(1);

if abs(sqrt(N)*rho(tau+1))>beta
c=c+1;
end
end
```

#### Verifica

```
if c/N<(alfa)
test=1;
else
test=0;
end
```

Figura 6.2: codice Matlab della funzione 'anderson2'

A questo punto si ha a disposizione un vettore corrispondente ad un rumore bianco; per poter stimolare il sistema però è necessario applicare in ingresso un segnale che abbia un'ampiezza massima adeguata alla velocità dell'elica. Risulta chiaro che valori compresi tra [-0.5 0.5] non sono sufficienti, infatti nella seconda parte di codice mostrata in Figura 6.3 il vettore 'rumore\_bianco' viene moltiplicato per una quantità pari a 80 così da ottenere dei valori compresi tra [-40, 40].

Infine, per fornire in ingresso al blocco Adams il rumore bianco, oltre al valore dell'ingresso, è necessario indicare l'istante di tempo in cui tale valore deve essere applicato. Quindi viene creata una matrice chiamata 'vel\_elica' costituita nella prima colonna da un vettore dei tempi 't' e nella seconda colonna dal vettore 'rumore\_bianco'. Essendo il tempo di simulazione pari a 100s e il tempo di campionamento 0.01s, il vettore dei tempi 't' dovrà essere costituito da 10000 elementi che vanno da 0 a 99.99 con un momento progressivo di 0.01.

#### Creazione ingresso per modello Adams

```
rumore_bianco=rumore_bianco'*80; %rumore con valori compresi tra [40,-40]
t=(0:0.01:99.99)'; %vettore dei tempi
vel_elica=[t rumore_bianco]; %matrice per l'ingresso al blocco Adams
Controls_Scooter_Persona %codice .m generato da Adams
Scooter_Persona %progetto Simulink
```

Figura 6.3: codice Matlab per creare il segnale di ingresso 'vel\_elica' per il blocco Adams e per aprire il progetto Simulink

A questo punto nel progetto Simulink, come mostrato in Figura 6.4, viene usato un blocco 'From Workspace' per applicare la matrice 'vel\_elica' in ingresso al blocco Adams, mentre con un blocco 'To Workspace' viene salvato il segnale di uscita in una variabile. Terminata la simulazione saranno disponibili nel workspace due vettori di dimensione 10000 che costituiscono le coppie di dati ingresso-uscita che verranno usati poi per calcolare l'ordine a priori ed effettuare l'identificazione.

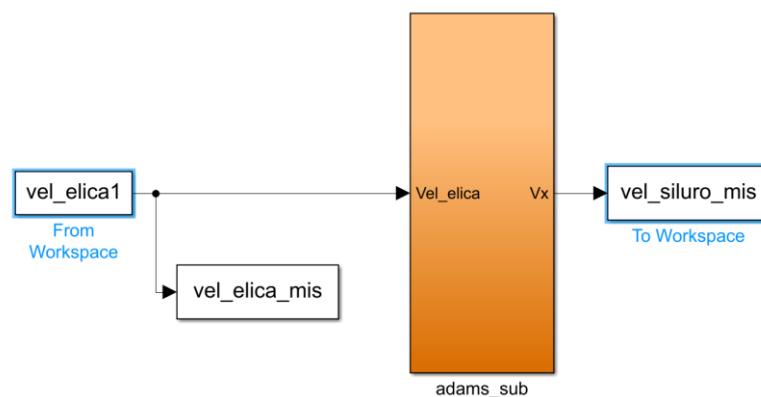


Figura 6.4: progetto Simulink usato per l'acquisizione dei dati del sistema Scooter+Diver

### 6.1.2 Determinazione dell'ordine a priori

Di seguito viene riportato il codice utilizzato per calcolare l'ordine a priori del sistema. Come già visto, per definire l'ordine del sistema si deve costruire la matrice tecnica  $M(n)$  definita dalla (5.3) e si valuta il massimo salto tra la differenza degli autovalori minimi per trovare l'ordine  $n$  per cui tale matrice risulta essere singolare. È stata definita una funzione chiamata 'DeterminazioneOrdine' che prende in ingresso i due vettori dell'ingresso e dell'uscita. All'interno della funzione viene definita la matrice tecnica 'M1' calcolata considerando  $n=1$  e viene salvato l'autovalore minimo della matrice  $M1^*M1$  nel vettore 'eigsMin'. Si definisce poi una variabile 'differenzaAttuale' usata per salvare la differenza tra l'autovalore minimo della matrice  $M(n)$  e quello della matrice  $M(n-1)$ , mentre la variabile 'differenzaPrecedente' viene usata per la differenza tra l'autovalore della matrice  $M(n-1)$  ed  $M(n-2)$ . Sarà quindi necessario calcolare almeno tre matrici tecniche, per poter verificare il primo salto tra la differenza degli autovalori minimi. All'interno del ciclo while, ad ogni nuovo ciclo, viene definita la matrice tecnica successiva e si calcola la differenza con l'autovalore della matrice precedente. Si esce dal ciclo while quando sono state valutate le prime cinque matrici tecniche e la 'differenzaPrecedente' risulta essere maggiore della 'differenzaAttuale'. Al termine viene mostrato, usando la funzione plot, un grafico con il valore dell'autovalore minimo nell'ordinata e il valore di  $n$  nell'ascissa. Si osserva del grafico riportato in Figura 6.6 che il maggior salto è in corrispondenza di  $n=2$ , fornendo così una prima indicazione sulla complessità del modello.

### Ordine a priori

```
function DeterminazioneOrdine(ingresso1,uscital)
ingresso=ingresso1;
uscita=uscital;
sizeIN=size(ingresso,1);
sizeOUT=size(uscita,1);
```

### M(1)

```
n=1;
M1=[uscita(2:end),uscita(1:(end-1)),ingresso(1:(end-1))];
```

### Inizializzo algoritmo

```
eigsMin=zeros(sizeIN,1); %vettore degli autovettori
eigsMin(1)=min(eig((M1')*M1));
differenzaAttuale=-1;
differenzaPrecedente=-1;
Mn=M1;
n=n+1;
```

### Algoritmo

```
while(n<=5) || ((differenzaPrecedente>differenzaAttuale))
%RIDUCO IL NUMERO DI RIGHE DI Mn DI 1
Mridotta=Mn(1:end-1,:);
%Predispongo la nuova 1 colonna
nuovoY=uscita(n+1:end);
%Predispongo la nuova n+2 -esima colonna
nuovoU=ingresso(n:end-1);
%Concateno le varie parti per realizzare la nuova Mn
Mn=[nuovoY,Mridotta(:,(1:n)),nuovoU,Mridotta(:,(n+1):end)];
%Calcolo l'autovalore minimo della nuova Mn
eigsMin(n)=min(eig((Mn')*Mn));
%Calcolo la nuova differenza
differenzaAttuale=eigsMin(n)-eigsMin(n-1);
if(n>2)
differenzaPrecedente=eigsMin(n-1)-eigsMin(n-2);
end
n=n+1;
end
```

### Grafici

```
plot([1:n-1],eigsMin(1:n-1));
semilogy([1:n-1],eigsMin(1:n-1))
```

Figura 6.5: codice Matlab della funzione *DeterminazioneOrdine*

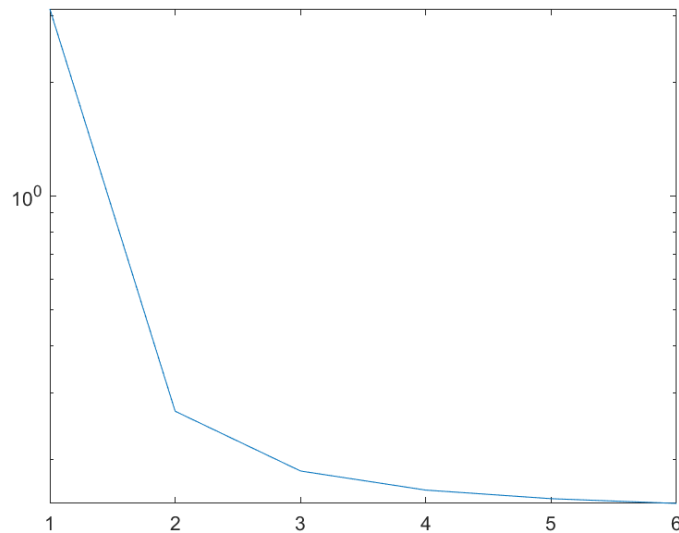


Figura 6.6: risultato della determinazione dell'ordine a priori usando i vettori di ingresso e uscita raccolti dal sistema *Scooter+Diver*

### 6.1.3 Stima dei modelli

Per la l'identificazione, Matlab mette a disposizione un toolbox chiamato 'systemIdentification' che permette di effettuare la stima sia dei modelli ARX che dei modelli ARMAX, oltre ad offrire diversi strumenti per effettuare una buona validazione. L'interfaccia del toolbox si presenta come in Figura 6.7.

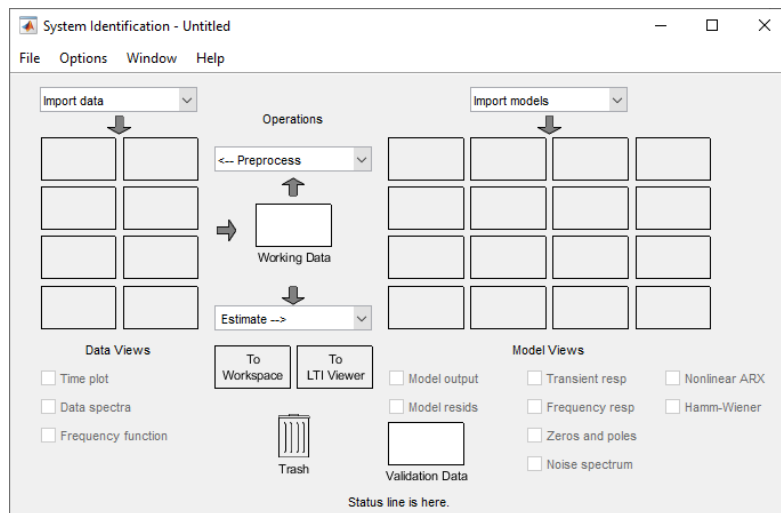


Figura 6.7: interfaccia del toolbox systemIdentification

Per prima cosa è necessario importare nel toolbox i dati raccolti selezionando 'Import data' e fornendo il vettore degli ingressi e il vettore delle uscite oltre al tempo di campionamento. A questo punto nella sezione 'Data Views' si viene a creare un blocco denominato 'data' che rappresenta il set dei dati composto dal segnale di ingresso e il segnale di uscita del sistema. Per una migliore identificazione viene rimossa la media e il trend dei segnali importati ottenendo un nuovo set di dati che verrà chiamato 'datad'. Per eseguire la cross-validazione inoltre è necessario dividere il set dei dati in due parti uguali, la prima metà verrà usata per la stima dei modelli, mentre la seconda verrà usata per la fase di validazione, si definiscono quindi due nuovi set chiamati 'datade' e 'datadv'. È inoltre possibile visualizzare l'andamento sia nel tempo che in frequenza dei dati raccolti ed elaborati, in modo da effettuare un'analisi pre-identificativa. In Figura 6.8 si osservano in colorazioni diverse l'andamento nel tempo dell'ingresso e dell'uscita dei set per l'identificazione e per la validazione, inoltre si può notare un piccolo discostamento nel segnale di uscita inerente al set dei dati 'data'

in cui non è stata applicata l'operazione di detrend. In Figura 6.9, invece viene mostrato il diagramma di bode generato dai dati importati e attraverso il quale è possibile fare una valutazione sul comportamento del sistema al variare della frequenza. Si osserva che oltre la frequenza di 300 rad/s sono presenti dei disturbi, quindi nella fase di validazione sarà sufficiente assicurarsi che il diagramma di bode del modello rispecchi quello dei dati almeno fino a tale frequenza.

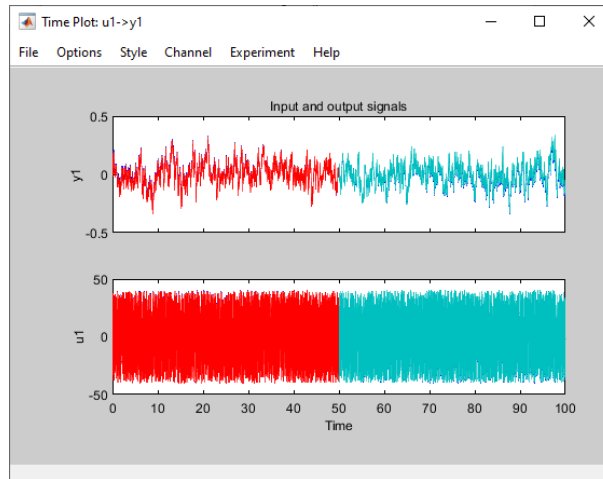


Figura 6.8: Time plot dei set di dati: 'data' in blu, 'datade' in rosso e 'datadv' in azzurro. Si nota uno scostamento impercettibile tra il set 'data' e i set 'datade' e 'datadv'

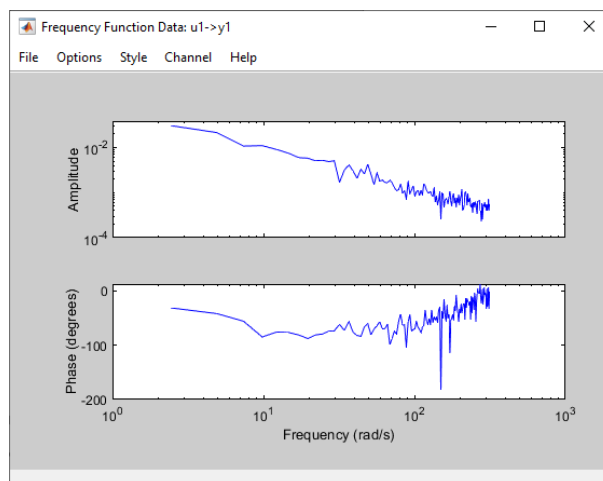


Figura 6.9: Diagramma di bode del set 'data'

A questo punto si trascina 'datade' nella casella 'Working Data' e 'datadv' nella casella 'Validation Data' per procedere con la stima dei modelli. Per prima cosa vengono stimati i modelli della famiglia ARX, si seleziona quindi dalla tendina 'Estimate' la voce 'Polynomial Models' e viene aperta una finestra come quella mostrata in Figura

6.10 in cui, selezionando 'Order Selection', sarà possibile indicare l'intervallo dei valori dei coefficienti  $n_a$ ,  $n_b$  e  $n_k$  per i quali si vuole stimare il modello. Come già specificato nel capitolo 5, oltre agli ordini  $n_a$  e  $n_b$ , viene richiesto anche l'ordine  $n_k$  che corrisponde al ritardo del modello. In questo modo verranno stimati tutti i modelli realizzabili dati dalle combinazioni dei parametri  $(n_a, n_b, n_k)$  nell'intervallo selezionato. Procedendo con il tasto 'estimate' viene mostrato un istogramma come in Figura 6.11, in cui ogni barra indica il valore dell'errore quadratico medio del modello corrispondente, inoltre con le colorazioni verde e blu vengono indicati rispettivamente i modelli che presentano i migliori indici AIC e MDL, mentre in rosso viene segnalato il modello che presenta una migliore aderenza ai dati (Best-fit). Verrà quindi fatta una prima selezione dei modelli scegliendo quelli in cui si verifica il maggior 'salto' oltre a quelli che mostrano i migliori indici. I modelli candidati vengono quindi importati e visualizzati nella sezione 'Model Views' come mostrato nella Figura 6.12.

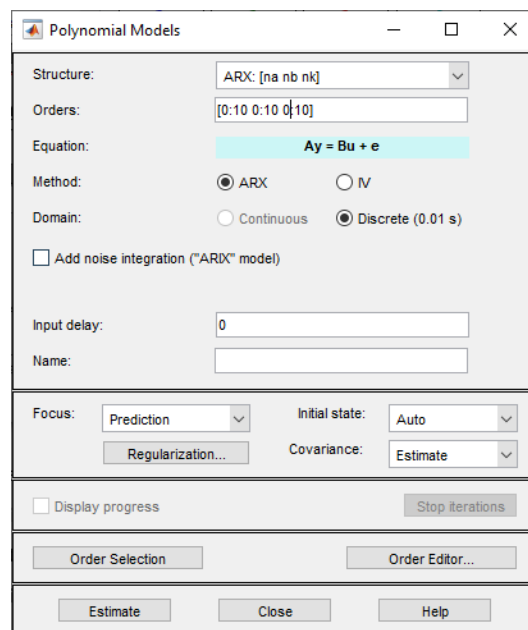


Figura 6.10: finestra di dialogo del toolbox systemIdentification per la stima dei modelli ARX

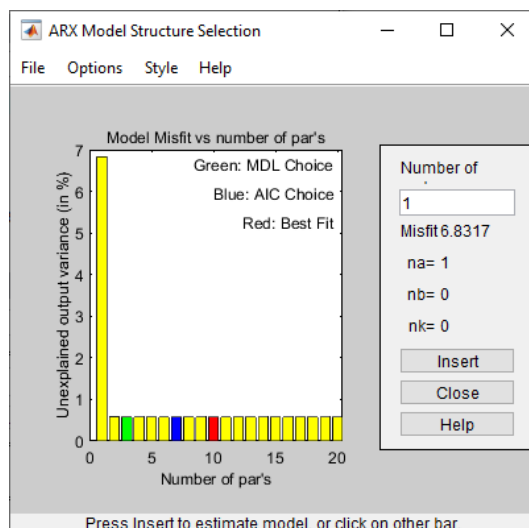


Figura 6.11: istogramma dei possibili modelli ARX

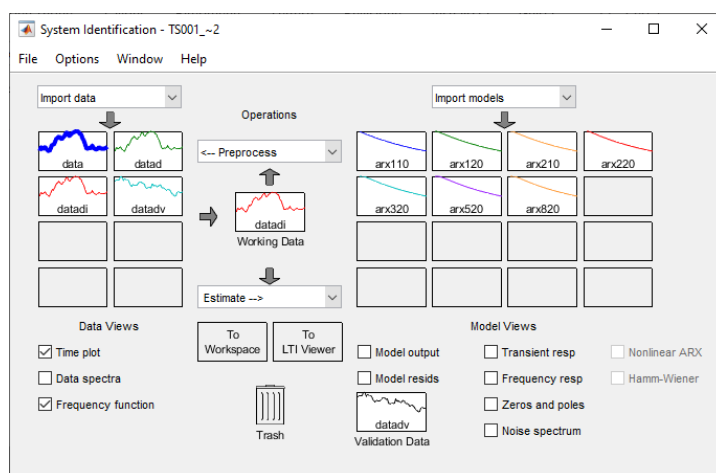


Figura 6.12: interfaccia toolbox 'systemIdentification' dopo la stima dei modelli arx

### 6.1.4 Validazione

Nella fase di validazione ogni modello viene valutato con le modalità descritte di seguito e successivamente, mettendoli a confronto, viene scelto il miglior modello che descrive il sistema. Si deve tenere conto che per evitare il problema della sovrapparametrizzazione si tende a privilegiare modelli più semplici rispetto a modelli complessi ma che aderiscono meglio ai dati.

Per valutare l'adesione ai dati di uno o più modelli, si seleziona 'Model output', dal quale si apre una finestra come quella mostrata in Figura 6.13. Nel grafico vengono messe a confronto l'uscita simulata del modello e quella misurata del sistema, inoltre



viene riportato il Best-fit che indica in percentuale quanto l'uscita del modello segue quella del sistema.

Per valutare la complessità invece, oltre a tenere in considerazione il risultato del calcolo dell'ordine a priori, è possibile analizzare i poli e gli zeri del sistema. Selezionando 'Zeros and Poles' si apre una finestra come in Figura 6.14 in cui vengono indicati con un cerchio gli zeri e con una croce i poli nel piano complesso. È possibile in aggiunta visualizzare gli intervalli di confidenza al 99% dei singoli poli e zeri e, se vi è una sovrapposizione degli intervalli tra un polo e uno zero, significa che c'è la possibilità che si annullino a vicenda e che quindi si potrebbe considerare un sistema di ordine inferiore. Vengono perciò scartati tutti quei modelli considerati troppo complessi, oltre a quelli che presentano un polo all'esterno della circonferenza unitaria in quanto implica l'instabilità del sistema.

Per valutare l'andamento in frequenza del modello, selezionando 'Frequency resp', si apre una finestra in cui viene visualizzato il diagramma di bode; si deve quindi verificare che il comportamento sia paragonabile al diagramma di bode ottenuto dai dati importati come mostrato in Figura 6.15.

Come riportato dalla teoria è molto importante verificare che il modello generi come residuo un rumore bianco. Matlab effettua di default il test di Anderson dell'errore di predizione considerando  $N=20$  ed  $\alpha=0.01$ , in cui quindi nel nostro caso, per soddisfare il test, si dovranno avere zero campioni uscenti dall'intervallo di confidenza escludendo il valore in zero. Selezionando 'Model Resid' Si apre una finestra come in Figura 6.16 in cui, nel grafico superiore, si può valutare il test di Anderson, mentre nel grafico sottostante viene riportata la correlazione tra i dati in ingresso e quelli in uscita per poter valutare se il ritardo  $n_k$  del modello è corretto.

Infine, selezionando 'present', vengono indicati nella command window i valori dei parametri del modello con i corrispondenti indici di incertezza, oltre ai valori degli indici MSE e AIC. Un'ulteriore verifica sulla complessità del modello viene fatta considerando l'ordine di grandezza degli indici di incertezza dei parametri, infatti se questi fossero dello stesso ordine di grandezza del parametro potrebbero causarne

l'annullamento e la conseguente variazione di grado del polinomio, indicando quindi una complessità eccessiva.

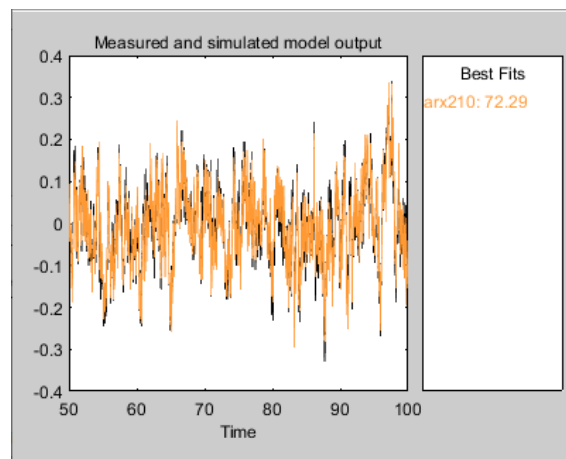
Una volta definiti i migliori modelli ARX vengono stimati i modelli ARMAX di ordini simili. Per la stima di quest'ultimi non è possibile impostare un range di variazione dei coefficienti come fatto per i modelli ARX, è quindi necessario per ogni stima stabilire gli ordini  $n_a$ ,  $n_b$ ,  $n_c$  e  $n_k$ . Per la fase di validazione invece si procede come visto per i modelli ARX.

Dopo aver eseguito la validazione per tutti i modelli ARX e ARMAX stimati si è giunti alla conclusione che il modello che aderisce meglio ai dati del sistema pur mantenendo una complessità adeguata risulta essere l'ARX(210) descritto da:

$$A(z) = 1 - 0.9976 z^{-1} + 0.03169 z^{-2} \quad (6.1)$$

$$B(z) = 0.001022 \quad (6.2)$$

Si è scelto un modello della famiglia ARX piuttosto che della famiglia ARMAX perché in fase di validazione si è riscontrato nei migliori modelli di entrambe le famiglie un comportamento quasi analogo, quindi la scelta è ricaduta nella famiglia più semplice. Di seguito vengono riportati i grafici usati per la validazione.



*Figura 6.13: confronto tra uscita simulata del modello ARX(210) e l'uscita del sistema misurata*

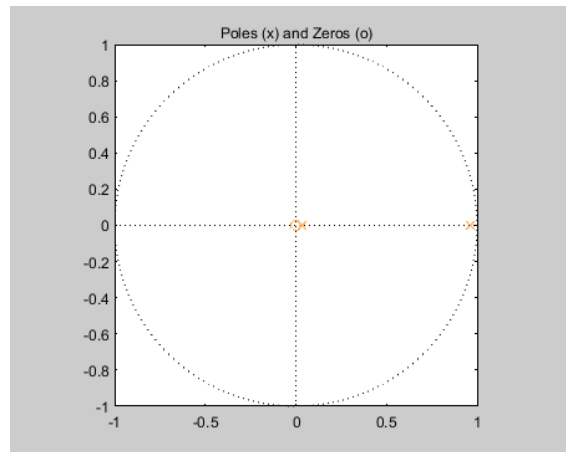


Figura 6.14: poli e zeri del modello ARX(210) rappresentati nel piano complesso, si precisa che con un'operazione di zoom si è verificato che il polo e lo zero non fossero sovrapposti.

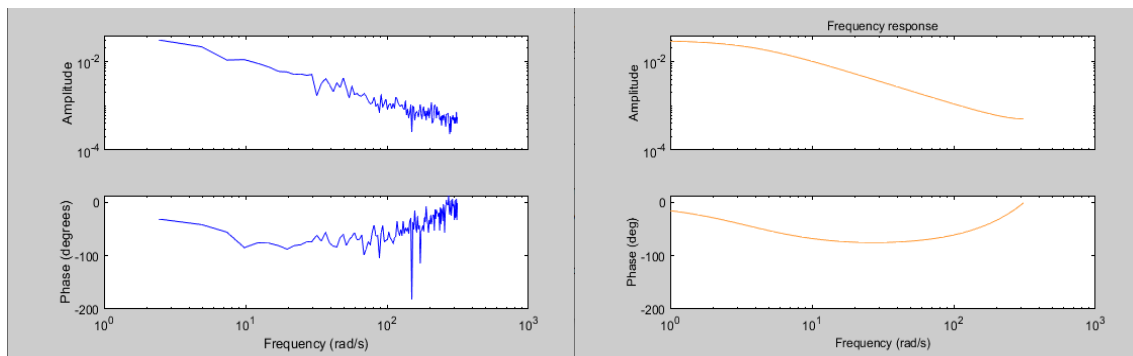


Figura 6.15: confronto diagramma di bode dei dati con il diagramma di bode del modello ARX(210)

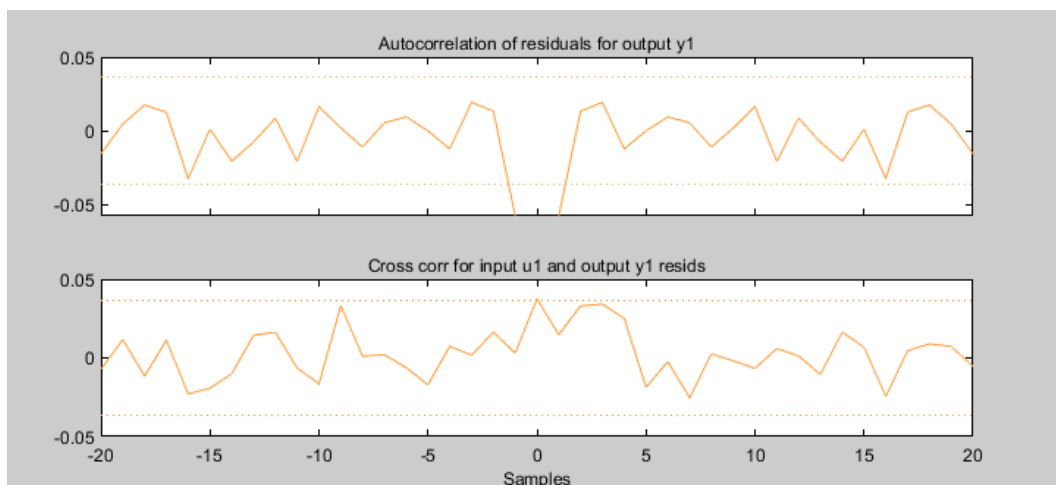


Figura 6.16: Risultato del test di Anderson sui residui e della cross correlazione tra ingresso e uscita del modello ARX(210)

Si può osservare per prima cosa dalla Figura 6.13 che il modello ARX(210) ha una discreta aderenza ai dati, infatti il Best-fit risulta essere del 72%. Dalla Figura 6.14 emerge la presenza di due poli e uno zero tutti all'interno della circonferenza unitaria, tra i quali non sono presenti sovrapposizioni. Il contributo dei due poli si può osservare anche dall'andamento del diagramma di bode visibile in Figura 6.15, il quale risulta essere fedele al diagramma di bode dei dati. Il test di Anderson del residuo risulta superato in quanto, come si può osservare in Figura 6.16, non sono presenti campioni all'esterno dell'intervallo di confidenza e si nota inoltre che nella correlazione tra ingresso e uscita non vi sono picchi al di fuori dell'intervallo, i quali indicherebbero un errore nella stima del ritardo  $nk$ . Infine, si è verificato che gli indici di incertezza fossero almeno di un ordine di grandezza inferiore a quello dei parametri e che gli indici FPE e MSE fossero i migliori tra i modelli stimati.

## **6.2 Identificazione Sistema Scooter+Diver+MotoreDC**

Per poter modellare il sistema con il motore DC si procede allo stesso modo descritto per il sistema Scooter+Diver, con l'unica differenza nella raccolta dei dati. In questo caso infatti, il rumore bianco viene posto in ingresso al motore, il quale riceve la velocità di rotazione espressa in rpm. Considerando anche in questo caso, metà della velocità massima, l'ampiezza del rumore bianco dovrà essere di 400 e sarà quindi sufficiente moltiplicare il vettore per 800. Inoltre, il sistema in esame presenta una risposta al gradino di 850ms, per questo si è scelto 0.1s come tempo di variazione del rumore bianco. È stata eseguita poi una simulazione di 100s e i dati sono stati raccolti ogni 0.1s, ottenendo 1000 campioni di coppie ingresso-uscita.

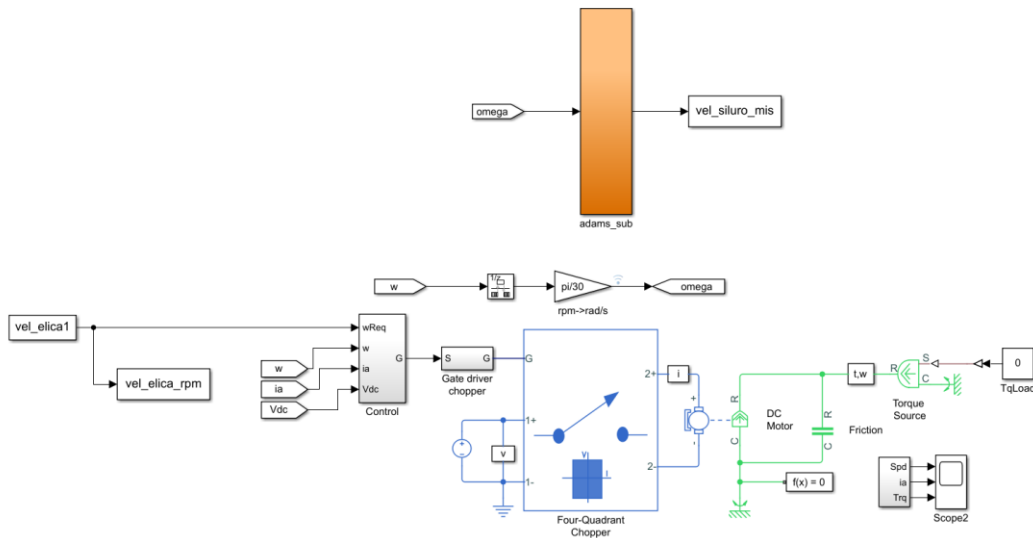


Figura 6.17: progetto Simulink per la raccolta dei dati del sistema Scooter+Diver+MotoreDC

La prima valutazione della complessità del sistema è stata fatta con il calcolo dell'ordine a priori. Dalla Figura 6.18 emerge che i maggiori 'salti' si hanno per valori di  $n=2$  e  $n=3$ .

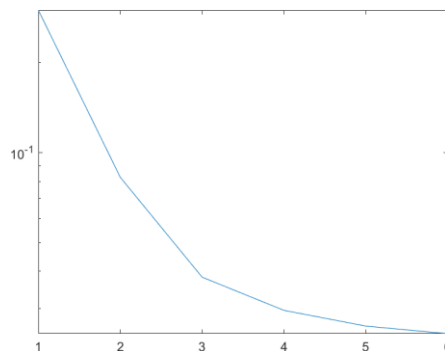


Figura 6.18: risultato della determinazione dell'ordine a priori usando i vettori di ingresso e uscita raccolti dal sistema Scooter+Diver+MotoreDC

Le procedure per la stima dei modelli e la validazione non vengono riportate essendo identiche a quelle viste nel caso precedente.

Viene quindi subito illustrato il modello scelto per il sistema Scooter+Diver+MotoreDC che corrisponde a l'ARMAX(3140) definito da:

$$A(z) = 1 - 1.667 z^{-1} + 1.077 z^{-2} - 0.3354 z^{-3} \quad (6.3)$$

$$B(z) = 0.0002424 \quad (6.4)$$

$$C(z) = 1 - 1.082 z^{-1} + 0.8595 z^{-2} - 0.2175 z^{-3} + 0.06892 z^{-4} \quad (6.5)$$

Quest'ultimo è stato preferito ad un modello ARMAX(1260) che pur avendo caratteristiche migliori presentava uno zero fuori dal cerchio unitario, indice del fatto che la fase non è minima, inoltre essendo l'ordine della parte esogena maggiore di quello della parte autoregressiva si potrebbero osservare problemi di stabilità del modello.

Di seguito vengono riportati i grafici usati per la validazione del modello scelto.

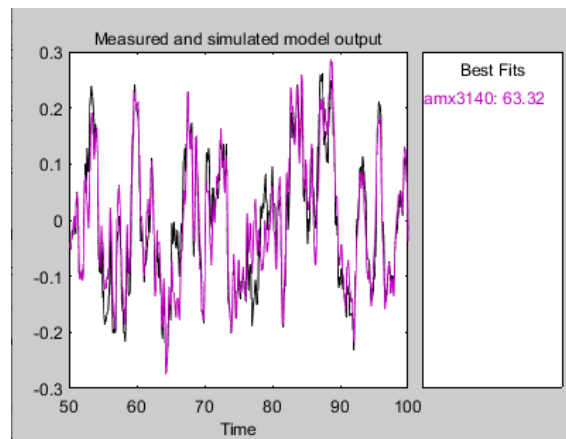


Figura 6.19: confronto tra uscita simulata del modello ARMAX(3140) e l'uscita del sistema misurata

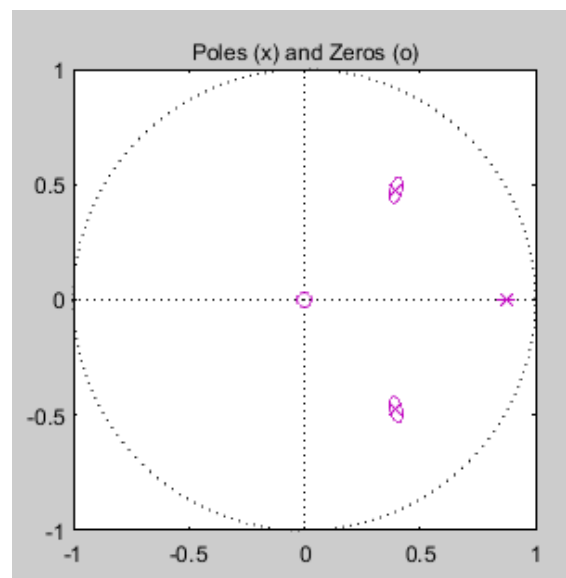


Figura 6.20: : Poli e zeri del modello ARMAX(3140) nel piano complesso.

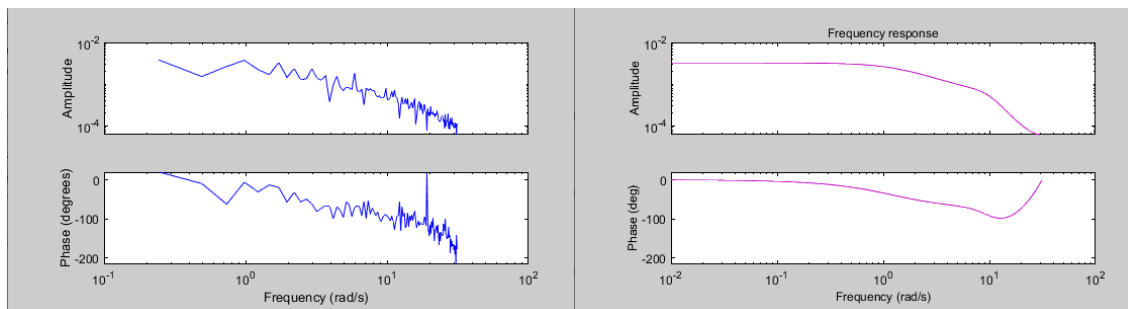


Figura 6.21: confronto diagramma di bode dei dati con il diagramma di bode del modello ARMAX(3140)

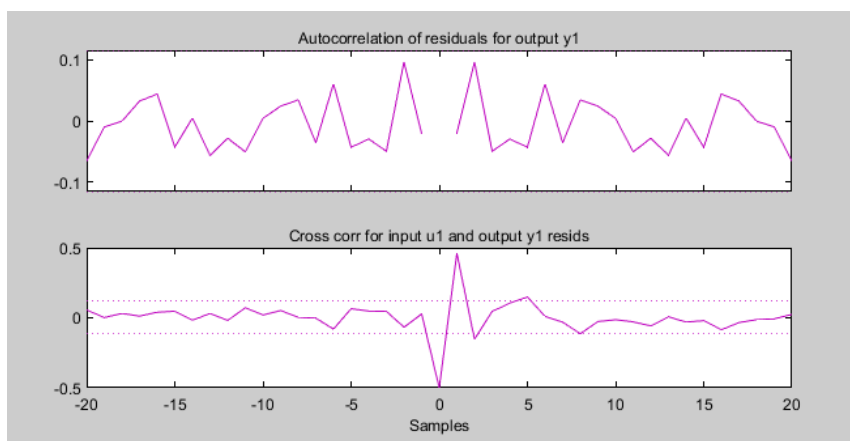


Figura 6.22: Risultato del test di Anderson sui residui e della cross correlazione tra ingresso e uscita del modello ARMAX(3140)

Dal confronto tra l'uscita simulata del modello e quella misurata dell'uscita si ottiene un Best-fit del 63.42% visibile in Figura 6.19. Tutti i poli sono all'interno del cerchio unitario e non sono presenti sovrapposizioni tra poli e zeri, assicurando la stabilità del modello. Dal confronto dei diagrammi di bode si osserva una buona aderenza per le frequenze che non sono soggette a rumori, mentre si rileva una variazione nell'andamento della fase per le frequenze maggiori. Si può affermare inoltre che l'errore di predizione del modello è un rumore bianco, infatti non sono presenti campioni al di fuori dell'intervallo di confidenza. Si osserva invece un picco al di fuori dell'intervallo nella correlazione tra ingresso e uscita che potrebbe quindi indicare un errore nella stima dell'ordine  $n_k$ . Si è provato a risolvere il problema stimando altri modelli ARMAX con valori di  $n_k$  maggiori di zero osservando però un decadimento quasi totale dell'aderenza ai dati e un aumento non indifferente degli indici FPE e

MSE; per tale ragione la scelta è comunque ricaduta sul modello ARMAX3140. Infine, si è verificato che tutti gli indici di incertezza fossero di almeno un ordine di grandezza inferiore ai parametri e che gli indici FPE e MSE fossero tra i migliori tra i modelli stimati.

## **6.3 Simulazioni**

Vengono di seguito riportate le simulazioni in cui vengono messe a confronto la risposta dei modelli con quella dei sistemi modellati.

È stato necessario per prima cosa salvare i modelli nel workspace e nel progetto Simulink aggiungere un blocco 'idmodel' che permette l'esecuzione sia di modelli arx che modelli armax.

### **6.3.1 Simulazione modello arx210**

In Figura 6.23 viene mostrato il progetto Simulink usato per la simulazione del modello ARX210 e del sistema Scooter+Diver. Si è voluto esaminare anche in questo caso la risposta al gradino, perciò è stato posto in ingresso sia al modello che al blocco Adams un gradino di ampiezza 83 (visibile nel grafico superiore della Figura 6.24). Al termine della simulazione si ottengono i grafici riportati in Figura 6.24 dove, nel grafico inferiore si osserva il confronto tra l'uscita del modello in blu e l'uscita del sistema in rosso. La risposta del modello risulta fedele a quella del sistema osservando però un errore tra le due uscite che potrebbe essere causato, come specificato anche nel capitolo 4, dal non aver modellato in maniera esatta il comportamento del fluido sull'elica oltre in parte dalle non linearità del sistema.



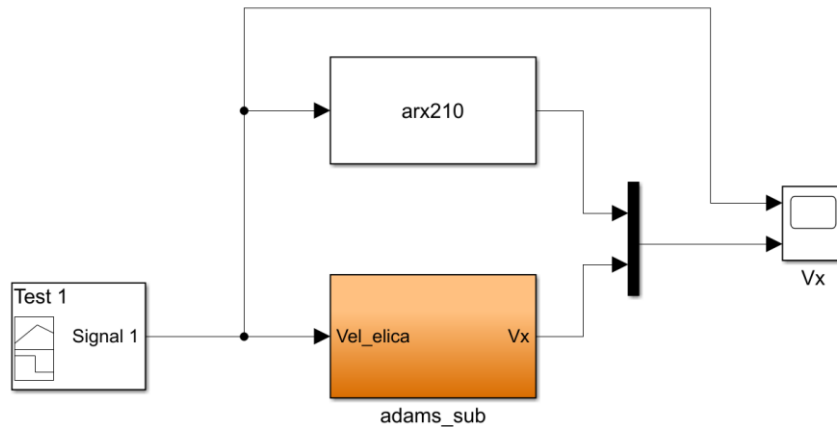


Figura 6.23: progetto Simulink per la simulazione del modello ARX(210) e il sistema Scooter+Diver

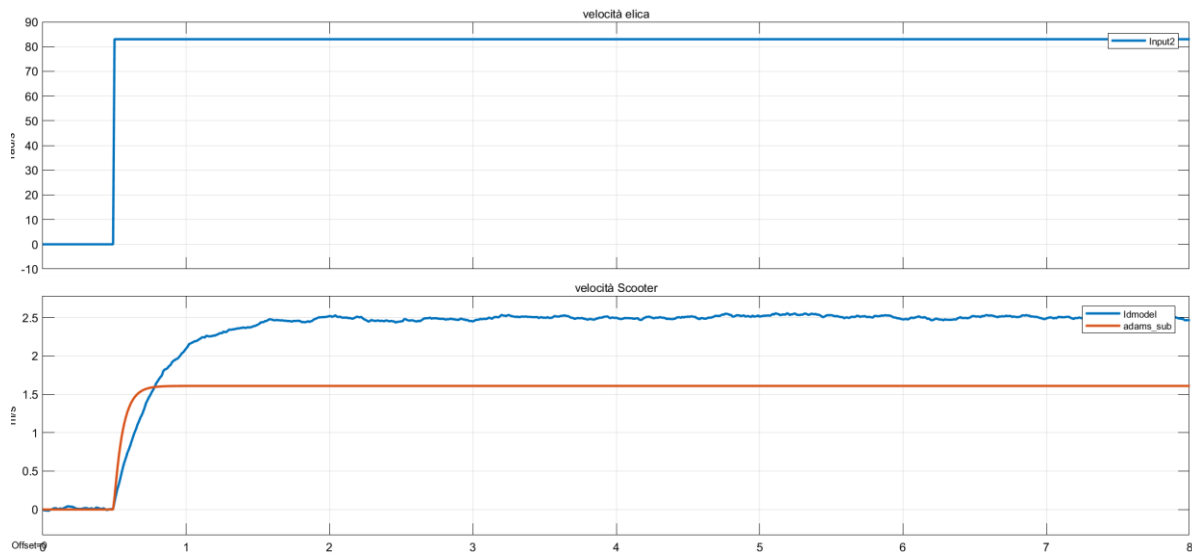


Figura 6.24: risposta al gradino del modello ARX(210) in blu e del sistema Scooter+Diver in rosso

### 6.3.2 Simulazione modello armax3140

Il progetto Simulink per la simulazione del modello ARMAX3140 e del sistema Scooter+Diver+MotoreDC viene mostrato in Figura 6.25. In questo caso è stato applicato in ingresso ai due blocchi un gradino di ampiezza 800 visibile nel grafico superiore della Figura 6.26, mentre nel grafico inferiore sono riportate l'uscita del modello in blu e quella del sistema in rosso.

Anche in questo caso l'uscita del modello segue quella del sistema con l'aggiunta di un errore, per il quale si possono fare le stesse considerazioni fatte in precedenza.

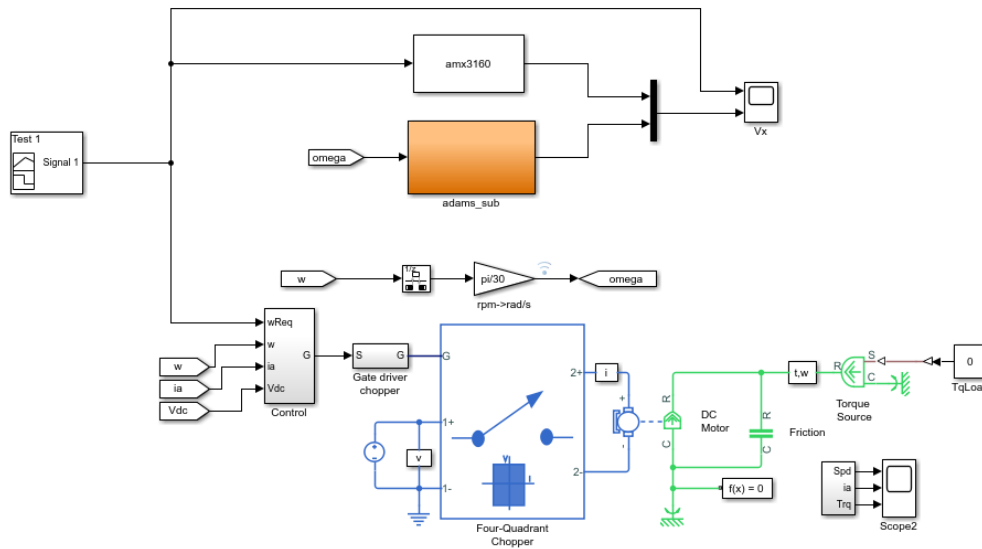


Figura 6.25: progetto Simulink per la simulazione del modello ARMAX(3140) e il sistema Scooter+Diver+MotoreDC

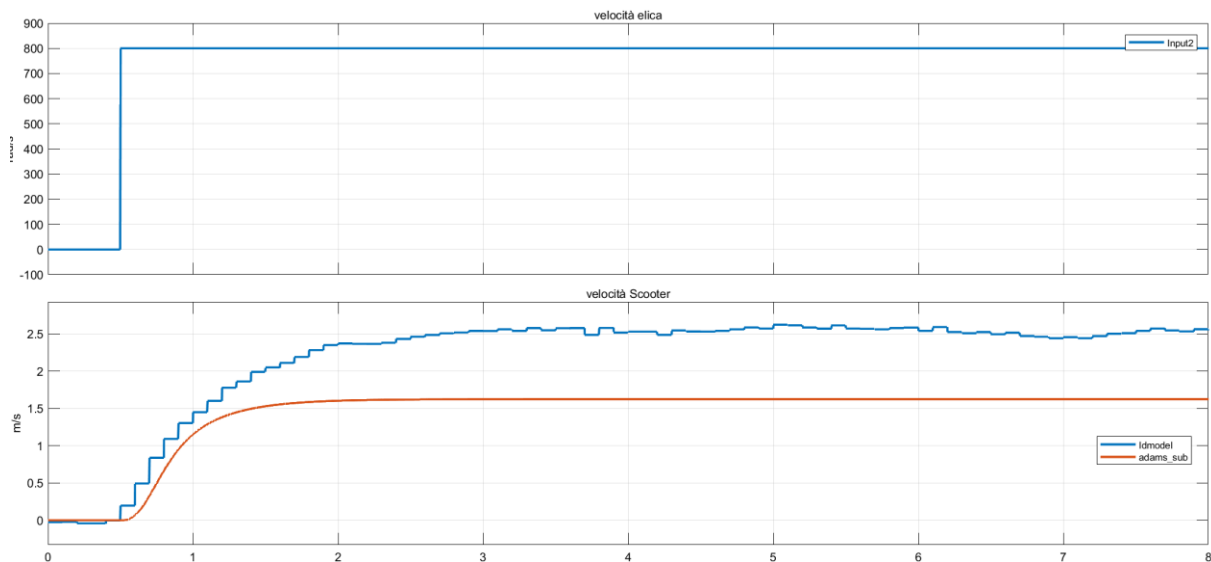


Figura 6.26: risposta al gradino del modello ARMAX(3140) in blu e del sistema Scooter+Diver+MotoreDC in rosso

## 7 Conclusioni

Si è creato un gemello digitale per il movimento nella direzione longitudinale di un sistema costituito da uno scooter DPV e un diver attraverso il quale è stato possibile effettuare simulazioni e verificare il comportamento del sistema. Per la sua realizzazione sono stati impiegati: il software Adams per la struttura meccanica e l'applicazione della dinamica e i software Matlab e Simulink per l'esecuzione delle simulazioni. Si è confrontato poi il comportamento del sistema a seguito dell'impiego di diversi blocchi propulsivi sfruttando la libreria Simscape Electrical, inizialmente considerando un motore ideale e in un secondo momento un motore DC e un motore BLDC.

Inoltre, sono stati ricavati due modelli parametrici, di cui uno per il sistema con il motore ideale e uno per il sistema con il motore DC, tramite un'identificazione con approccio black box, in particolare, valutando le famiglie ARX e ARMAX. Le coppie ingresso-uscita necessarie all'identificazione sono state ricavate dalla simulazione dei due sistemi realizzati in Simulink, mentre per la stima dei modelli si è utilizzato il toolbox 'systemIdentification'. In conclusione, si è trovato un modello ARX(210) per il sistema con motore ideale e un modello ARMAX(3140) per il sistema con motore DC. Si deve comunque specificare che il comportamento del modello digitale non rispecchia correttamente il comportamento reale del sistema, in quanto non è stata considerata la dinamica data dalla presenza del fluido sull'elica dello scooter. Sarebbe quindi interessante per degli sviluppi futuri studiare l'intera dinamica dell'elica, oltre ad ampliare il modello per tutte le direzioni del movimento.

Si potrebbe inoltre sviluppare un controllo sulla velocità dello scooter sfruttando i modelli parametrici ottenuti dall'identificazione.

## Bibliografia

- Presterro T. T. J. (2001). *Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle.*
- Fossen T. I. (2002). *Marine Control Systems–Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles.*
- Scaradozzi D. *Dispense corso Modellistica e Identificazione dei Processi Dinamici.* Università Politecnica delle Marche.
- Mathworks.com. (2020). *Mathworks - Makers Of MATLAB And Simulink.* [online] Available at: <<https://www.mathworks.com/>>