



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Meccanica

A.a. 2025/2026

Sessione di Laurea Magistrale Marzo/Aprile 2026

Simulazione multibody di una stampante FDM

Relatore:
Prof. Federico Colombo

Candidato:
Enrico Bortone
Matricola 318710

Abstract

Il presente lavoro di tesi ha l'obiettivo di studiare, implementare in MATLAB/Simulink/Simscape e analizzare un sistema meccatronico per il controllo degli assi X-Y di una stampante FDM cartesiana. Il sistema sviluppato comprende la generazione della traiettoria percorsa dall'ugello nel piano di stampa tramite MATLAB, la sua traduzione in G-code e la successiva lettura del codice, da cui viene generata la sequenza di attivazione degli avvolgimenti dei motori passo-passo che gestiscono gli assi X e Y della stampante. Il lavoro include inoltre, mediante l'utilizzo di Simulink/Simscape, la modellazione del controllo elettrico dei motori passo-passo, dei motori stessi, delle pulegge, delle cinghie, dei carrelli e dell'attrito sulle guide, al fine di ottenere un modello completo della dinamica del sistema. Una volta sviluppati gli script MATLAB, implementati i modelli in Simulink/Simscape, e definiti i parametri fisici, sono state svolte diverse simulazioni finalizzate alla validazione dei modelli. In particolare, la simulazione conclusiva verifica che, a partire dalla traiettoria desiderata dell'ugello, i carrelli degli assi X e Y permettano all'ugello di seguire il percorso con elevata accuratezza. I risultati ottenuti confermano la validità dei modelli sviluppati e l'efficacia del sistema di controllo, evidenziando come questi strumenti di calcolo numerico e simulazione possano essere impiegati per la progettazione, l'analisi e l'ottimizzazione di una stampante FDM.

Keywords: produzione additiva, stampante FDM cartesiana, curve di Bézier, motore passo-passo ibrido, controllo del moto, simulazione in MATLAB/Simulink/Simscape

Indice

Indice	II
Indice delle figure	V
Indice delle tabelle.....	X
Introduzione.....	1
1 Additive Manufacturing	2
1.1 Confronto tra produzione additiva (AM) e lavorazione CNC	3
1.1.1 Materiali	3
1.1.2 Velocità del processo.....	3
1.1.3 Complessità del componente	4
1.1.4 Precisione	4
1.1.5 Geometria del componente.....	4
1.1.6 Programmazione.....	4
1.2 Fasi del processo di produzione additiva.....	4
1.2.1 Modello CAD	5
1.2.2 Conversione in formato STL/AMF	5
1.2.3 Trasferimento e manipolazione del file STL sulla macchina AM.....	6
1.2.4 Setup della macchina	6
1.2.5 Costruzione della parte	6
1.2.6 Rimozione e pulizia del prodotto	6
1.2.7 Post-processo del prodotto	7
1.2.8 Applicazione.....	7
1.3 Generazione del G-code a partire dal modello CAD.....	7
1.4 Generazione dei percorsi utensile nella produzione additiva	9
1.4.1 Contour-parallel tool-path	9
1.4.2 Direction-parallel tool-path	9
1.4.3 Traiettoria di deposizione mista	11
1.4.4 Pianificazione dei percorsi ottimali	11
1.5 Tecnologia Fused Deposition Modelling (FDM)	13
1.5.1 Parametri di processo nella tecnica FDM.....	14
1.5.2 Qualità del componente finito tramite stampa FDM.....	15
1.5.3 Stampante FDM cartesiana.....	15
2 Modellazione di un sistema cinghia-carrello-guida in cui si presenta il fenomeno dello stick-slip	18
2.1 Modello di attrito	18
2.1.1 Regione di velocità prossima allo zero.....	19
2.1.2 Regione di velocità intermedie	19
2.1.3 Regione di velocità elevata.....	20
2.1.4 Velocità negative	20

2.2	Fenomeno dello stick-slip	20
2.3	Descrizione del sistema	21
2.4	Implementazione modello in Simulink/Simscape	21
2.5	Solver	25
2.6	Definizione dei parametri del sistema	26
2.7	Risultati della simulazione	27
2.7.1	Moto uniforme a velocità costante	28
2.7.2	Moto sinusoidale di posizione	31
2.8	Variazione dei parametri del modello, caso velocità costante	33
2.8.1	Variazione della rigidezza della cinghia.....	33
2.8.2	Variazione dello smorzamento della cinghia.....	37
2.8.3	Variazione della massa del carrello	39
2.8.4	Variazione del coefficiente di attrito dinamico	41
2.8.5	Variazione del coefficiente di attrito statico.....	43
2.9	Variazione dei parametri del modello, caso moto sinusoidale	45
2.9.1	Variazione della rigidezza della cinghia.....	45
2.9.2	Variazione dello smorzamento della cinghia.....	46
2.9.3	Variazione della massa del carrello	48
2.9.4	Variazione del coefficiente di attrito dinamico	49
2.9.5	Variazione del coefficiente di attrito statico.....	50
3	Generazione del G-code a partire da una traiettoria	52
3.1	Curve di Bézier	52
3.1.1	Curve di Bézier lineari.....	53
3.1.2	Curve di Bézier quadratiche	54
3.2	Calcolo delle coordinate dei punti lungo una traiettoria.....	56
4	Motori passo-passo	71
4.1	Motore passo-passo a riluttanza variabile (VR).....	71
4.1.1	Funzionamento del motore passo-passo a riluttanza variabile (VR).....	72
4.2	Motore passo-passo a magnete permanente (PM)	73
4.2.1	Funzionamento del motore passo-passo a magnete permanente (PM)	73
4.3	Motore passo-passo ibrido	76
4.3.1	Funzionamento del motore passo-passo ibrido	77
5	Modellazione di un motore passo-passo ibrido in Simulink/Simscape.....	80
5.1	Modello Simulink/Simscape	80
5.1.1	Generatori di impulsi	82
5.1.2	Ponti H.....	83
5.1.3	Blocco Stepper Motor.....	85
5.1.4	Solver.....	89

5.2	Simulazione in modalità full-step	89
5.3	Simulazione in modalità half-step	91
6	Modellazione della trasmissione a cinghia.....	94
6.1	Trasmissione a cinghia.....	94
6.2	Parametri del modello	96
6.3	Implementazione del modello.....	98
6.4	Risultati della simulazione.....	100
7	Controllo del movimento dell'ugello nel piano XY di una stampante FDM cartesiana	102
7.1	Script MATLAB per la generazione delle sequenze di attivazione.....	104
7.2	Script completo	113
7.3	Risultati della simulazione.....	120
7.3.1	Solver.....	120
7.3.2	Confronto degli spostamenti lungo l'asse x	120
7.3.3	Confronto degli spostamenti lungo l'asse y	121
7.3.4	Confronto delle velocità lungo l'asse x	122
7.3.5	Confronto delle velocità lungo l'asse y	123
7.3.6	Confronto degli spostamenti nel piano XY	123
7.3.7	Confronto dei moduli delle velocità nel piano XY	124
8	Conclusione	126
	Bibliografia.....	128

Indice delle figure

Figura 1.1: (a) Assemblaggio finale di un collettore di aspirazione realizzato tramite stampante FDM, (b) Sistema di aspirazione completato dopo processi di finitura e l'assemblaggio finale di sensori e supporti [5].....	2
Figura 1.2: Modello CAD di una tazza con immagini aggiuntive che illustrano l'influenza di layer di diverso spessore sulla costruzione [6].....	3
Figura 1.3: Le otto fasi della produzione additiva [6].....	5
Figura 1.4: Conversione del modello CAD nel formato STL [6].....	5
Figura 1.5: Esempio di una piramide triangolare descritta dal file STL. Si osservano 3 coordinate per ciascun nodo e 3 coordinate per il versore normale [6].....	6
Figura 1.6: A sinistra un modello in formato STL e a destra lo stesso modello suddiviso in layer tramite il processo di slicing [7].....	8
Figura 1.7: Confronto tra le due strategie di riempimento [9].....	9
Figura 1.8: Cambi di direzione nella traiettoria DP [9].....	10
Figura 1.9: Traiettoria DP e velocità di deposizione lungo la curva [9].....	10
Figura 1.10: Fenomeni del sovrariempimento e sottoriempimento lungo una traiettoria di stampa DP [9].....	10
Figura 1.11: Esempio di una traiettoria di deposizione mista [9].....	11
Figura 1.12: Traiettorie di deposizione e curve di velocità per differenti angoli di inclinazione [9].....	11
Figura 1.13: Lunghezza della traiettoria e tempo di lavorazione per diverse inclinazioni [9].....	12
Figura 1.14: Prima partizione in sotto-regioni [9].....	12
Figura 1.15: Seconda partizione in sotto-regioni [9].....	12
Figura 1.16: Partizione finale e sotto-percorsi di stampa [9].....	12
Figura 1.17: Tecnologia FDM [10].....	13
Figura 1.18: Parametri che influenzano il processo FDM [10].....	14
Figura 1.19: Tipologia di architettura delle stampanti FDM cartesiane [11].....	15
Figura 1.20: Rappresentazioni di schemi cinematici tipici delle stampanti 3D cartesiane. I rettangoli indicano giunti prismatici, i cerchi giunti sferici, il triangolo nero corrisponde alla testina di stampa e il parallelepipedo nero al piano di stampa. Gli schemi (a-c) illustrano schemi cinematici funzionali, mentre gli schemi (d-f) includono anche i giunti del telaio e delle guide, i quali non possono essere considerati completamente rigidi. [11].....	16
Figura 2.1: Andamento della forza di attrito in funzione della velocità relativa [13].....	18
Figura 2.2: Modello cinghia-carrello-attrito.....	24
Figura 2.3: Blocco Solver Configuration.....	25
Figura 2.4: Sezione Solver Information.....	25
Figura 2.5: Impostazioni del solver.....	26
Figura 2.6: Velocità di distacco v_{brk}	27
Figura 2.7: Impostazioni del blocco Translational Friction.....	27
Figura 2.8: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$	29
Figura 2.9: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$	29

Figura 2.10: Confronto tra spostamento $x(t)$, velocità $\dot{x}(t)$ e forza applicata dal motore $F(t)$	30
Figura 2.11: Analisi del fenomeno dello stick-slip	30
Figura 2.12: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$	31
Figura 2.13: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$	32
Figura 2.14: Confronto tra spostamento $x(t)$, velocità $\dot{x}(t)$ e forza applicata dal motore $F(t)$	32
Figura 2.15: Analisi del fenomeno dello stick-slip	32
Figura 2.16: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della rigidità della cinghia.....	35
Figura 2.17: Ingrandimento della Figura 2.16.....	36
Figura 2.18: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare della rigidità della cinghia	36
Figura 2.19: Andamento della forza trasmessa al carrello al variare della rigidità della cinghia	36
Figura 2.20: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare dello smorzamento della cinghia.....	37
Figura 2.21: Ingrandimento Figura 2.20	38
Figura 2.22: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare dello smorzamento della cinghia	38
Figura 2.23: Andamento della forza trasmessa al carrello al variare dello smorzamento della cinghia	39
Figura 2.24: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della massa del carrello	40
Figura 2.25: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare della massa del carrello.....	40
Figura 2.26: Andamento della forza trasmessa al carrello al variare della massa del carrello	40
Figura 2.27: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito dinamico	41
Figura 2.28: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare del coefficiente di attrito dinamico	42
Figura 2.29: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito dinamico	42
Figura 2.30: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito statico.....	43
Figura 2.31: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare del coefficiente di attrito statico	44
Figura 2.32: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito statico.....	44

Figura 2.33: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della rigidezza della cinghia.....	45
Figura 2.34: Confronto tra velocità del carrello $x(t)$ e velocità imposta dal motore $x_m(t)$ al variare della rigidezza della cinghia	46
Figura 2.35: Andamento della forza trasmessa al carrello al variare del coefficiente della rigidezza della cinghia	46
Figura 2.36: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare dello smorzamento della cinghia.....	47
Figura 2.37: Confronto tra velocità del carrello $x(t)$ e velocità imposta dal motore $x_m(t)$ al variare dello smorzamento della cinghia	47
Figura 2.38: Andamento della forza trasmessa al carrello al variare dello smorzamento della cinghia	47
Figura 2.39: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della massa del carrello	48
Figura 2.40: Confronto tra velocità del carrello $x(t)$ e velocità imposta dal motore $x_m(t)$ al variare della massa del carrello.....	48
Figura 2.41: Andamento della forza trasmessa al carrello al variare della massa del carrello	49
Figura 2.42: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito dinamico	49
Figura 2.43: Confronto tra velocità del carrello $x(t)$ e velocità imposta dal motore $x_m(t)$ al variare del coefficiente di attrito dinamico	50
Figura 2.44: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito statico.....	50
Figura 2.45: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito statico.....	51
Figura 2.46: Confronto tra velocità del carrello $x(t)$ e velocità imposta dal motore $x_m(t)$ al variare del coefficiente di attrito statico	51
Figura 2.47: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito dinamico	51
Figura 3.1: Traiettoria composta da segmenti rettilinei e raccordo parabolico [16].....	52
Figura 3.2: Curva di Bézier lineare e relativi punti di controllo P_0 e P_1 , identificati dai vettori di posizione p_0 e p_1	53
Figura 3.3: Andamento temporale del parametro u per una curva di Bézier lineare.....	54
Figura 3.4: Curva di Bézier quadratica e relativi punti di controllo P_0, P_1, P_2	55
Figura 3.5: Traiettoria nel piano X-Y costituita da due segmenti rettilinei, P_0P_1 e P_1P_2 con un raccordo di raggio δ	57
Figura 3.6: Passaggio dal primo tratto rettilineo al raccordo	59
Figura 3.7: Passaggio dal raccordo al secondo tratto rettilineo	59
Figura 3.8: Spostamento nel piano X-Y	63
Figura 3.9: Spostamento lungo l'asse X e spostamento lungo l'asse Y in funzione del tempo	64
Figura 3.10: Velocità lungo l'asse X e la velocità lungo l'asse Y in funzione del tempo .	65

Figura 3.11: Velocità risultante nel piano XY	65
Figura 3.12: File G-code generato per la realizzazione traiettoria pianificata dell'ugello di stampa.....	66
Figura 4.1: Componenti principali di un motore passo-passo [18].....	71
Figura 4.2: Schema di un motore passo-passo a riluttanza variabile con $ns = 6$ e $nr = 4$ [19]	72
Figura 4.3: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a riluttanza variabile con $ns = 6$ e $nr = 4$ in modalità di funzionamento full step [20].....	72
Figura 4.4: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a riluttanza variabile con $ns = 6$ e $nr = 4$ in modalità di funzionamento half step [20].....	73
Figura 4.5: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $ns = 6$ e $nr = 4$ in modalità di funzionamento full step [20].....	74
Figura 4.6: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $ns = 6$ e $nr = 4$ in modalità di funzionamento half step [20]	75
Figura 4.7: Motore passo-passo ibrido [21]	76
Figura 4.8: Schema di un motore passo-passo ibrido da 200 step/giro [17]	77
Figura 4.9: Struttura interna di un motore ibrido [22].....	77
Figura 4.10: Sequenza di attivazione degli avvolgimenti di un motore passo-passo ibrido con $ns = 6$ e $nr = 4$ in modalità di funzionamento full step [20].....	78
Figura 4.11: Sequenza di attivazione degli avvolgimenti di un motore passo-passo ibrido con $ns = 6$ e $nr = 4$ in modalità di funzionamento half step [20]	78
Figura 5.1: Motore passo-passo 17HM19-2004S Full Datasheet [23]	80
Figura 5.2: Modello Simulink/Simscape del motore passo-passo e del relativo sistema di controllo.....	81
Figura 5.3: Descrizione dei blocchi che compongono il modello riportato in Figura 5.2 .	81
Figura 5.4: Descrizione dei blocchi che compongono il modello riportato in Figura 5.2 .	81
Figura 5.5: Segnale a onda quadra [24].....	82
Figura 5.6: Configurazione del blocco Pulse Generator per ottenere il segnale riportato in Figura 5.7.....	82
Figura 5.7: Esempio di un segnale a onda quadra.....	82
Figura 5.8: Dettaglio del primo ponte H riportato in Figura 5.2.....	83
Figura 5.9: Dettaglio del secondo ponte H riportato in Figura 5.2	83
Figura 5.10: Modello di uno degli interruttori dei ponti H	84
Figura 5.11: Configurazione del blocco Switch.....	84
Figura 5.12: Configurazione del blocco Cast.....	85
Figura 5.13: Configurazione del blocco Stepper Motor.....	87
Figura 5.14: Modello del motore inizialmente sviluppato in Simscape 2024.....	88
Figura 5.15: Configurazione blocco Stepper Motor (2024).....	88
Figura 5.16: Configurazione dei blocchi Pulse Generator in funzionamento full step	89
Figura 5.17: Stato dei comandi in funzione del tempo in funzionamento full step	90
Figura 5.18: Spostamento angolare del rotore in funzione del tempo in funzionamento full step.....	90
Figura 5.19: Ingrandimento Figura 5.18	91
Figura 5.20: Configurazione dei blocchi Pulse Generator in funzionamento half step	91

Figura 5.21: Stato dei comandi in funzione del tempo in funzionamento half step.....	92
Figura 5.22: Spostamento angolare del rotore in funzione del tempo, in funzionamento half step.....	92
Figura 5.23: Ingrandimento Figura 5.22	93
Figura 6.1: Schema geometrico della trasmissione a cinghia	94
Figura 6.2: Schema cinematico della trasmissione a cinghia.....	95
Figura 6.3: Puleggia dentata 16 denti in alluminio, diametro esterno 13 mm, diametro foro 5 mm, larghezza cinghia max. 7 mm, larghezza totale 14 mm [28].....	96
Figura 6.4: Cinghia dentata con profilo GT [29]	97
Figura 6.5: Modello Simulink/Simscape del motore passo-passo, del sistema di controllo, della trasmissione a cinghia e dell'attrito	98
Figura 6.6: Dettaglio della trasmissione a cinghia e dell'attrito riportato in Figura 6.5	99
Figura 6.7: Stato dei comandi in funzione del tempo in funzionamento full step	100
Figura 6.8: Confronto tra lo spostamento imposto sulla puleggia e lo spostamento del carrello.....	100
Figura 6.9: Ingrandimento della Figura 6.8.....	101
Figura 7.1: Estratto del file G-code riportato in Figura 3.12	102
Figura 7.2: Modello Simulink/Simscape aggiornato con l'aggiunta di un secondo motore passo-passo per la gestione dell'asse Y della stampante.....	103
Figura 7.3: Dettaglio del sistema motore e trasmissione che gestisce l'asse x, riportato in Figura 7.2.....	103
Figura 7.4: Dettaglio del blocco From Workspace, riportato in Figura 7.2.....	104
Figura 7.5: Estratto della matrice "input.x"	112
Figura 7.6: Dettaglio dei sensori riportati in Figura 7.2.....	120
Figura 7.7: Confronto degli spostamenti lungo l'asse x.	121
Figura 7.8: Confronto degli spostamenti lungo l'asse y.	121
Figura 7.9: Confronto delle velocità lungo l'asse x.	122
Figura 7.10: Ingrandimento Figura 7.9	122
Figura 7.11: Confronto delle velocità lungo l'asse y	123
Figura 7.12: Ingrandimento Figura 7.11	123
Figura 7.13: Confronto degli spostamenti nel piano XY	124
Figura 7.14: Ingrandimento Figura 7.13	124
Figura 7.15: Confronto dei moduli delle velocità nel piano XY.....	125

Indice delle tabelle

Tabella 1.1: Principali variabili CNC per generare il file G-code [8]	8
Tabella 2.1: Parametri del modello	21
Tabella 2.2: Blocchi Simscape utilizzati nel modello [14]	21
Tabella 2.3: Blocchi Simulink utilizzati nel modello [15]	24
Tabella 4.1: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $ns = 6$ e $nr = 4$ in modalità di funzionamento full-step	75
Tabella 4.2: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $ns = 6$ e $nr = 4$ in modalità di funzionamento half-step	76
Tabella 6.1: Parametri utilizzati per la modellazione di cinghia e carrello	97
Tabella 6.2: Parametri utilizzati per la modellazione dell'attrito	98
Tabella 7.1: Esempio movimento in avanti con $step_mm = 0.08$ mm, $x_reale = 0$, $step_dx = 0.02$ mm	109
Tabella 7.2: Esempio movimento negativo con $step_mm = 0.08$ mm, $x_reale = 0.16$ mm, $step_dx = -0.02$ mm	109
Tabella 7.3: Esempio istruzione $mod(fase_idx_x, Nseq) + 1$	110

Introduzione

Negli ultimi anni, la crescente complessità dei sistemi ingegneristici ha reso necessario l'utilizzo di software di modellazione e simulazione sempre più avanzati.

Un sistema mecatronico, come una stampante FDM, è costituito da componenti meccanici, elettrici ed elettronici che interagiscono tra di loro, quindi è un insieme di sottosistemi di natura fisica differente che operano in modo interconnesso tra loro.

I sottosistemi meccanici sono costituiti da corpi, rigidi o deformabili, collegati tra loro mediante dei giunti che permettono il moto relativo in determinate direzioni e lo impediscono in altre.

L'energia necessaria all'azionamento degli attuatori e all'alimentazione di tutti gli elementi elettrici è descritta dai sottosistemi elettrici, i cui componenti consentono la conversione tra energia elettrica e meccanica.

Componenti come microcontrollori e sensori, necessari per l'elaborazione dei segnali, sono descritti dalla parte elettronica e di controllo, la quale ha l'obiettivo di garantire le prestazioni richieste nel tempo.

Per progettare e simulare questi sistemi complessi, servirebbero fin dalle prime fasi della progettazione, prototipi fisici costosi. Da qui nasce lo sviluppo di software di modellazione e simulazione, i quali permettono di realizzare un banco prova virtuale in cui è possibile studiare il sistema in modo approfondito. È possibile quindi implementare e interconnettere i vari sottosistemi della macchina da progettare, e in seguito valutare il comportamento del sistema al variare di uno o più parametri fisici fino a scegliere i parametri di progetto attraverso tecniche di ottimizzazione. Oltre al vantaggio legato alla riduzione dei costi, vi è anche una notevole riduzione dei tempi di progettazione e un ambiente più sicuro, con rischi minori per le persone.

Il lavoro di tesi è stato svolto utilizzando il software MATLAB, che offre un ambiente di calcolo numerico e modellazione. Inoltre, sono stati anche utilizzati due strumenti interni a Matlab: Simulink e Simscape.

Simulink è un ambiente di simulazione in cui i modelli sono descritti tramite blocchi collegati tra loro da segnali. All'interno di ogni blocco sono presenti una o più relazioni matematiche, mentre i segnali possono essere numeri, vettori o matrici.

Simscape, invece, non richiede la scrittura di equazioni, poiché il modello viene creato implementando blocchi che rappresentano i componenti fisici del sistema. Questi blocchi sono collegati da segnali fisici.

Integrando MATLAB, Simulink e Simscape è possibile studiare e progettare sistemi mecatronici, e più in generale, sistemi ingegneristici. [1] [2] [3] [4]

1 Additive Manufacturing

Produzione additiva (Additive Manufacturing, o AM) è il termine tecnico utilizzato per indicare ciò che un tempo veniva chiamato prototipazione rapida (Rapid Prototyping, o RP) e che comunemente è noto come stampa 3D. Con questo termine si indicava un processo industriale per produrre rapidamente un prototipo di un sistema o componente a partire da un modello digitale, utile per le successive fasi di sviluppo fino al prodotto finale. Tuttavia, con l'evoluzione di queste tecnologie, il termine RP non è più sufficiente a descriverne tutte le potenzialità, poiché molte parti realizzate con questo metodo possono essere già impiegate come prodotti finiti di ottima qualità. Inoltre, il termine prototipazione rapida non evidenzia il fatto che i componenti vengono costruiti aggiungendo progressivamente materiale.

Per questo oggi si utilizza il termine produzione additiva. Questa tecnologia si basa sul principio secondo cui un modello creato tramite CAD può essere trasformato direttamente in un componente fisico complesso, senza una pianificazione articolata delle fasi produttive.



Figura 1.1: (a) Assemblaggio finale di un collettore di aspirazione realizzato tramite stampante FDM, (b) Sistema di aspirazione completato dopo processi di finitura e l'assemblaggio finale di sensori e supporti [5]

A differenza dei metodi tradizionali, in cui è necessario analizzare attentamente la geometria del componente per stabilire l'ordine delle operazioni e gli strumenti da utilizzare, con la produzione additiva è sufficiente una conoscenza della modellazione CAD, del funzionamento della macchina e dei materiali impiegati.

Tutte le macchine di fabbricazione additiva disponibili sul mercato utilizzano un approccio basato sulla costruzione dei componenti strato dopo strato, aggiungendo progressivamente materiale. Ogni strato, o layer, rappresenta una sezione molto sottile del modello tridimensionale ricavato dal CAD e più i layer sono sottili, più il prodotto finale sarà fedele al progetto iniziale (Figura 1.2).

Le diverse tecnologie di fabbricazione additiva adottano quest'approccio di slicing, cioè la suddivisione del modello CAD in una sequenza di layer, e si differenziano principalmente per architettura della macchina, materiali utilizzabili e modalità di realizzazione degli strati. Questi fattori influenzano proprietà meccaniche, velocità di produzione, lavorazioni successive, dimensioni della macchina e costo complessivo del processo. [6]



Figura 1.2: Modello CAD di una tazza con immagini aggiuntive che illustrano l'influenza di layer di diverso spessore sulla costruzione [6]

1.1 Confronto tra produzione additiva (AM) e lavorazione CNC

La produzione additiva condivide alcune caratteristiche con la lavorazione CNC, pur presentando differenze significative. Entrambi i processi sono controllati da computer, ma la principale distinzione risiede nel fatto che la produzione additiva è un processo additivo, mentre la lavorazione CNC è un processo sottrattivo, in cui, a partire da un blocco di materiale più grande, si rimuove la parte in eccesso fino ad ottenere la forma desiderata. Non esiste un processo intrinsecamente migliore, la scelta dipende da diversi fattori, quali la tipologia del componente da realizzare, la precisione, i materiali impiegati e i costi, per cui ciascuna tecnologia può risultare più adatta in contesti specifici. [6]

1.1.1 Materiali

In origine, i processi di produzione additiva (AM) erano basati principalmente su polimeri e cere. Con il tempo, la tecnologia si è evoluta, permettendo di lavorare anche con materiali compositi, metalli e ceramiche. La lavorazione CNC, invece, è particolarmente efficace con materiali duri e relativamente fragili, come acciai e altre leghe metalliche, consentendo di ottenere pezzi uniformi e prevedibili dal punto di vista strutturale. Questo si differenzia dai componenti realizzati tramite AM, che possono presentare piccole irregolarità interne, come vuoti o variazioni delle proprietà meccaniche, dovute all'orientamento e ai parametri di stampa. Per quanto riguarda polimeri e cere, la lavorazione CNC può essere applicata a questi materiali in casi specifici, come la realizzazione di stampi.

1.1.2 Velocità del processo

Generalmente, le macchine CNC rimuovono materiale molto più rapidamente di quanto una macchina AM riesca ad aggiungerne nello stesso intervallo di tempo. Tuttavia, la produzione AM realizza il pezzo in un'unica fase, mentre la lavorazione CNC richiede operazioni preliminari, come la preparazione della macchina, la pianificazione delle lavorazioni e la scelta degli utensili, oltre al necessario riposizionamento del pezzo o all'utilizzo di macchine diverse. Di conseguenza, la velocità va valutata sull'intero processo produttivo e non solo nella fase di lavorazione del pezzo. Complessivamente, l'AM può costruire il componente in poche ore, mentre con la lavorazione CNC sono necessari tempi molto più lunghi.

1.1.3 Complessità del componente

Uno dei principali vantaggi della produzione additiva emerge quando si devono realizzare componenti con geometrie complesse. Infatti, nelle tecnologie AM, elementi come sottosquadri o cavità interne possono essere realizzati senza particolari difficoltà, mentre nel caso delle lavorazioni CNC la forma del pezzo può limitare la capacità dell'utensile, montato sul mandrino, di raggiungere tutte le superfici da lavorare. Di conseguenza, alcuni pezzi particolarmente complessi non potrebbero essere prodotti con macchine CNC se non suddividendoli in più parti e assemblandoli successivamente.

1.1.4 Precisione

La risoluzione delle macchine AM è nell'ordine delle decine di micron e dipende dal sistema di posizionamento della macchina, che include motori e sistemi di trasmissione. Inoltre, il diametro dell'ugello influenza lo spessore delle pareti.

Spesso la risoluzione non è la stessa lungo tutti gli assi. In particolare, l'asse verticale, che corrisponde allo spessore degli strati, presenta generalmente una risoluzione inferiore rispetto a quella nel piano di stampa. Nelle macchine CNC la precisione dipende, oltre che dal sistema di movimento dei tre assi della macchina, anche dalle dimensioni degli utensili di taglio, come il raggio degli angoli interni. Tuttavia, essendo un processo sottrattivo, è possibile ottenere pareti anche più sottili rispetto alle dimensioni dell'utensile.

1.1.5 Geometria del componente

Nella produzione additiva, gli oggetti tridimensionali sono suddivisi in sezioni bidimensionali sovrapposte, ognuna con uno spessore definito. In questo modo, anche un oggetto tridimensionale complesso viene trasformato in strati più semplici da realizzare. Nelle lavorazioni CNC, invece, la lavorazione delle superfici avviene direttamente nello spazio tridimensionale. Forme semplici, come cilindri o parallelepipedi, sono facilmente lavorabili, mentre forme complesse richiedono un notevole aumento dei punti che l'utensile deve seguire, oltre alla variazione dell'orientamento dello stesso, e in alcuni casi possono risultare impossibili da produrre.

1.1.6 Programmazione

La programmazione di una macchina CNC è più complessa rispetto a quella di una macchina AM, poiché richiede la definizione di numerosi parametri. Inoltre, mentre una programmazione non corretta di una macchina AM può portare semplicemente a un pezzo difettoso, un errore nella programmazione di una lavorazione CNC può causare danni alla macchina e comportare rischi per la sicurezza dell'operatore.

1.2 Fasi del processo di produzione additiva

La maggior parte dei processi di produzione additiva comprende le seguenti otto fasi [6]:

1. Modello CAD
2. Conversione nel formato STL/AMF
3. Trasferimento e manipolazione del file STL/AMF sulla macchina AM
4. Setup della macchina
5. Costruzione della parte

6. Rimozione e pulizia del prodotto
7. Post-processo del prodotto
8. Applicazione

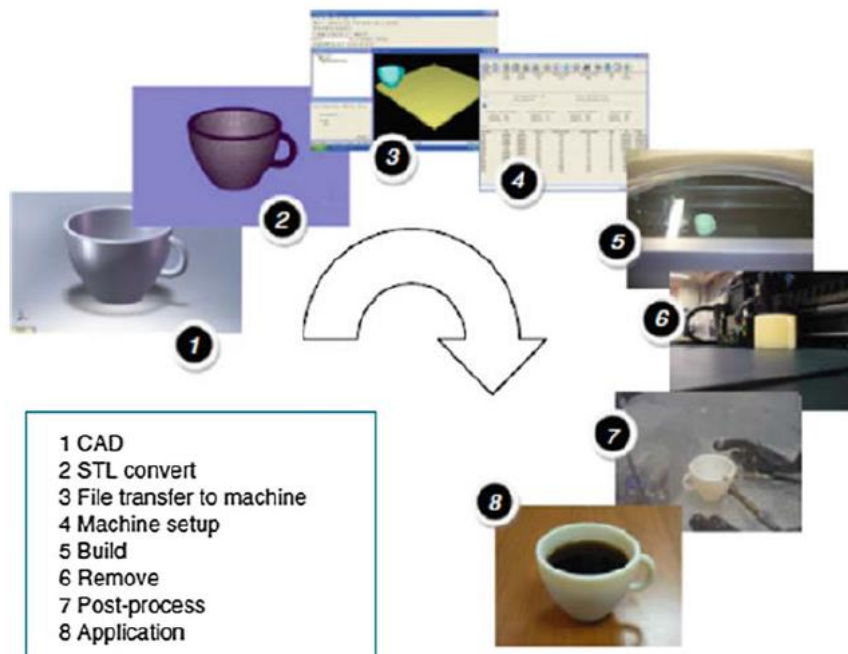


Figura 1.3: Le otto fasi della produzione additiva [6]

1.2.1 Modello CAD

La tecnologia AM non esisterebbe senza la modellazione 3D CAD. Il modello può essere generato da un progettista tramite l'interfaccia utente di un software CAD oppure tramite la scansione 3D di un pezzo già esistente.

1.2.2 Conversione in formato STL/AMF

Il formato STL è considerato di fatto uno standard per la maggior parte delle tecnologie AM. Tale formato rimuove i dati di costruzione e la cronologia di modellazione CAD, approssimando le superfici mediante una serie di triangoli. La dimensione dei triangoli può essere impostata dal software CAD e deve essere sufficientemente piccola da rappresentare accuratamente la geometria del modello.



Figura 1.4: Conversione del modello CAD nel formato STL [6]

I file STL sono un insieme di vertici e vettori normali associati ai triangoli, e non contengono unità di misura, colore o altre informazioni sui materiali (Figura 1.5). Per questo motivo si

adotta anche il formato AMF (Additive Manufacturing File), che estende il formato STL includendo dimensioni, colore e proprietà dei materiali.

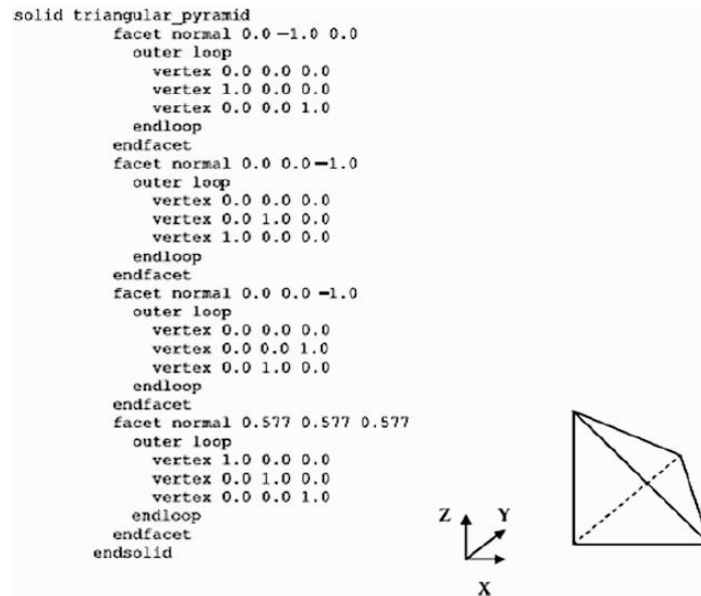


Figura 1.5: Esempio di una piramide triangolare descritta dal file STL. Si osservano 3 coordinate per ciascun nodo e 3 coordinate per il vettore normale [6]

1.2.3 Trasferimento e manipolazione del file STL sulla macchina AM

Una volta creato e riparato il file STL, esso può essere inviato alla macchina AM. Tuttavia, prima di procedere con la stampa del componente, è necessario verificare il modello e modificare, se necessario, la posizione e l'orientamento del componente tramite l'interfaccia grafica del software della macchina AM.

1.2.4 Setup della macchina

Le macchine AM presentano specifici parametri di configurazione. Alcune macchine permettono l'uso di diversi materiali e consentono di ottimizzare i parametri per migliorare velocità e risoluzione. Oltre alle impostazioni del software di slicing, è necessario effettuare anche alcune preparazioni della macchina, come la verifica della presenza sufficiente di materiale e il livellamento del piano di stampa rispetto agli assi della macchina. Un'errata configurazione potrebbe produrre un pezzo con bassa qualità finale.

1.2.5 Costruzione della parte

Completate le fasi precedenti, il processo passa alla fase di costruzione, controllata interamente dal computer, durante la quale avviene la deposizione del materiale strato dopo strato in base alla tecnica di stampa e alla macchina utilizzata.

1.2.6 Rimozione e pulizia del prodotto

Idealmente il pezzo prodotto dovrebbe essere finito con interventi minimi, ma in realtà spesso sono necessari trattamenti successivi per raggiungere la qualità desiderata. Il pezzo deve innanzitutto essere separato dal piano di stampa e deve essere rimosso il materiale di supporto, necessario per evitare deformazioni durante il processo di stampa. Anche i supporti

devono essere progettati per essere rimovibili in modo semplice, ma spesso è comunque necessario del lavoro manuale.

1.2.7 Post-processo del prodotto

La fase di post-elaborazione comprende le operazioni di finitura del prodotto, come levigatura, lucidatura e applicazione di rivestimenti. Inoltre, per migliorare le proprietà del materiale, possono essere eseguiti trattamenti chimici o termici.

1.2.8 Applicazione

Dopo la fase di post-elaborazione il prodotto può essere utilizzato. Tuttavia si deve tenere conto che, anche se simili a quelli prodotti con tecnologie tradizionali, i pezzi realizzati mediante AM possono presentare proprietà meccaniche anisotrope dovute al processo di costruzione strato dopo strato e alla direzione di fabbricazione. Ad esempio, nei pezzi metallici prodotti mediante AM il rapido raffreddamento può generare microstrutture differenti rispetto allo stesso componente realizzato con tecnologie tradizionali. Di conseguenza, il comportamento durante l'utilizzo può risultare diverso e i progettisti devono tenerne conto durante la fase di progettazione.

1.3 Generazione del G-code a partire dal modello CAD

Per realizzare un oggetto a partire da un modello digitale, il processo di Additive Manufacturing (AM) richiede algoritmi in grado di tradurre il modello CAD in un file G-code, ossia un file testuale contenente istruzioni per il controllo dei motori e degli altri componenti della stampante. Dalla creazione del modello CAD alla generazione del G-code, il processo prevede diversi passaggi:

1. Generazione del file STL a partire dal modello CAD
2. Algoritmo per la lettura e riparazione dei file STL, per correggere eventuali errori di mesh
3. Determinazione della posizione e dell'orientamento del pezzo da parte dell'operatore tramite l'interfaccia software della macchina AM
4. Algoritmo di slicing, che suddivide il modello in strati (layer) e definisce parametri di stampa come spessore degli strati, pattern di riempimento e supporti
5. Algoritmo per la generazione del percorso di deposizione della testina di stampa
6. Algoritmo per la creazione del file G-code, che raccoglie tutte le informazioni provenienti dallo slicing e dal percorso utensile

Il modello viene progettato in un software CAD e salvato come file STL. Questo file viene successivamente elaborato tramite il processo di slicing, che suddivide il modello in una serie di layer e definisce parametri di stampa.

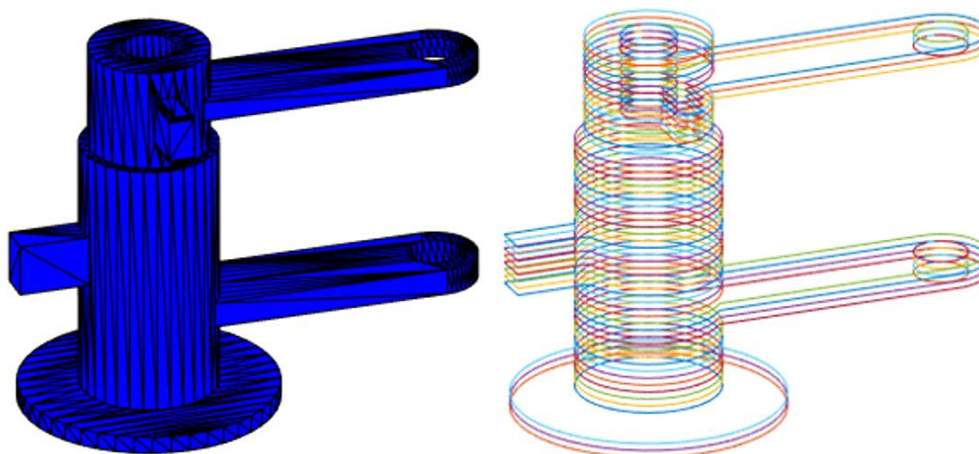


Figura 1.6: A sinistra un modello in formato STL e a destra lo stesso modello suddiviso in layer tramite il processo di slicing [7]

Dopo lo slicing, viene generato il percorso dell'utensile, che traduce ogni layer in movimenti precisi dell'ugello o della testa di stampa per depositare il materiale in modo efficiente e accurato. Infine, viene creato il file G-code, contenente tutte le istruzioni necessarie per la produzione dell'oggetto.

Tabella 1.1: Principali variabili CNC per generare il file G-code [8]

Variabili CNC	Descrizione
G0	Posizionamento rapido
G1	Interpolazione lineare
G21	Impostazione unità in mm
G28	Ritorno alla posizione (0,0,0)
G90	Coordinate assolute
X	Posizione assoluta o incrementale dell'asse X
Y	Posizione assoluta o incrementale dell'asse Y
Z	Posizione assoluta o incrementale dell'asse Z
F	Impostazione della velocità della testina di stampa
M101	Attivazione estrusore
M103	Disattivazione estrusore
M104S	Impostazione della temperatura dell'estrusore

La fase di slicing è cruciale nel processo di AM ed esiste una grande varietà di software specializzati. Numerosi studi si sono concentrati sullo sviluppo di algoritmi capaci di variare lo spessore degli strati in base alla geometria dell'oggetto, con l'obiettivo di migliorare la finitura superficiale. In particolare, vengono adottati strati più spessi nelle zone in cui il profilo del componente rimane pressoché costante, mentre strati più sottili sono utilizzati nelle aree in cui è necessario seguire con maggiore precisione la geometria del componente. Altri studi hanno sviluppato algoritmi adattivi per ottimizzare la generazione del percorso utensile, riducendo tempi di stampa e difetti superficiali. [8]

1.4 Generazione dei percorsi utensile nella produzione additiva

Il percorso utensile (tool-path) consiste in una traiettoria controllata lungo la quale l'ugello si muove per depositare il materiale e costruire il componente strato dopo strato. La qualità finale del pezzo, in termini di rugosità superficiale, precisione dimensionale e resistenza meccanica, è fortemente legata al percorso seguito dall'ugello, insieme ad altri parametri di processo. Per questo motivo sono stati svolti diversi studi per migliorare le strategie di pianificazione del percorso utensile. Le principali strategie di riempimento adottate nella Additive Manufacturing (AM) sono: la contour-parallel tool-path e la direction-parallel tool-path. [9]

1.4.1 Contour-parallel tool-path

Nella tecnica contour-parallel (CP), il percorso utensile è costituito da una serie di linee che seguono il profilo dei bordi della sezione bidimensionale ottenuta dal processo di slicing (Figura 1.7b). Questa strategia di deposizione consente di ottenere una maggiore precisione geometrica e superfici più regolari rispetto alla strategia direction-parallel. Tuttavia, lo svantaggio principale è rappresentato dall'elevata complessità computazionale necessaria per generare l'offset dei contorni, soprattutto in presenza di geometrie complesse o con molteplici cavità. In alcuni casi, tale tecnica può inoltre generare percorsi non ottimali, costringendo l'ugello a spostamenti a vuoto tra linee non connesse, interrompendo temporaneamente il flusso di materiale.

1.4.2 Direction-parallel tool-path

La tecnica direction-parallel tool-path (DP) è una delle più utilizzate nella Additive Manufacturing (AM). Questo metodo consiste nel riempire la superficie di uno strato mediante linee parallele orientate secondo una direzione predefinita (Figura 1.7a). Questa modalità è più semplice a livello computazionale rispetto alla tecnica CP, ma offre una precisione inferiore. Infatti, quando si realizzano modelli complessi con cavità interne, può risultare difficile ottenere una regione continua senza interrompere frequentemente la deposizione del materiale.

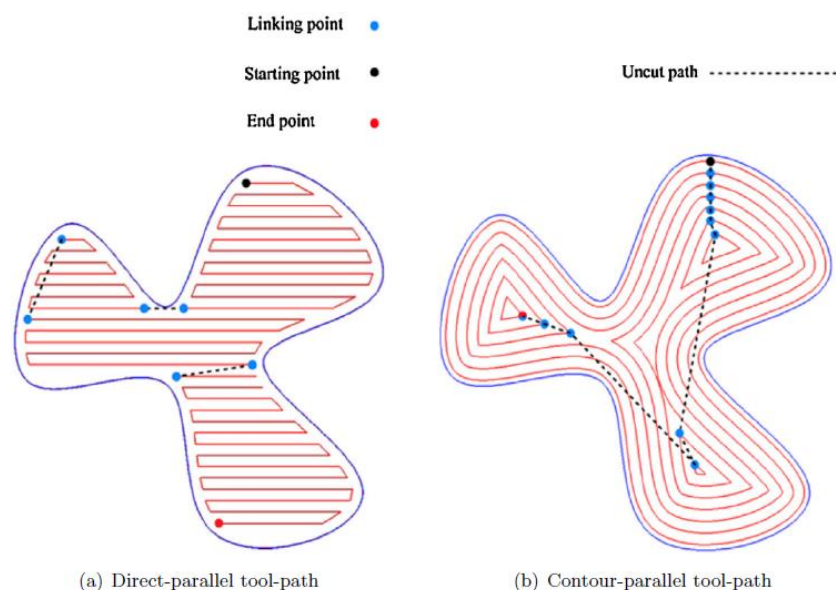


Figura 1.7: Confronto tra le due strategie di riempimento [9]

Inoltre, la presenza di numerosi cambi di direzione richiede frequenti riduzioni di velocità (Figura 1.9), provocando continui cicli di accelerazione e decelerazione che riducono la velocità media di avanzamento dell'ugello, rallentando quindi l'intero processo di stampa e potenzialmente compromettendo la precisione della deposizione. Un fattore determinante è dunque il numero di angoli presenti nei punti di giunzione tra i segmenti del percorso. Per evitare bruschi cambi di direzione, il percorso reale deve essere raccordato in modo fluido tra un segmento e l'altro, in base ai parametri di accelerazione e decelerazione impostati.

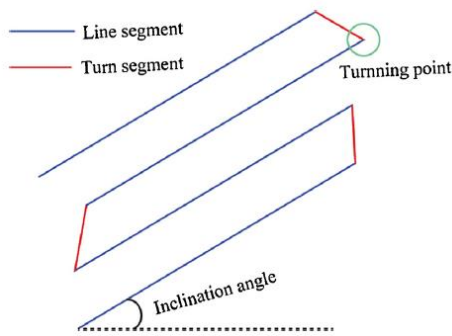


Figura 1.8: Cambi di direzione nella traiettoria DP [9]

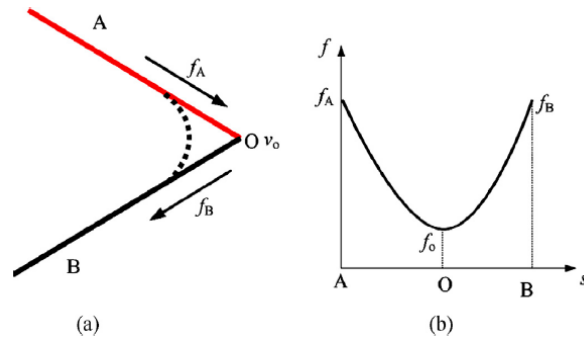


Figura 1.9: Traiettoria DP e velocità di deposizione lungo la curva [9]

In corrispondenza dei cambi di direzione possono verificarsi due fenomeni opposti:

- sovrariempimento (overfilling), quando una zona riceve materiale più di una volta
- sottoriempimento (underfilling), quando una zona non viene riempita completamente

La probabilità che questi due fenomeni si verifichino aumenta quanto più piccolo è l'angolo tra due segmenti successivi e quanto maggiore è il numero di cambi di direzione. Di conseguenza, la pianificazione dei percorsi deve mirare a ridurre il numero di angoli, al fine di garantire una distribuzione del materiale il più uniforme possibile.

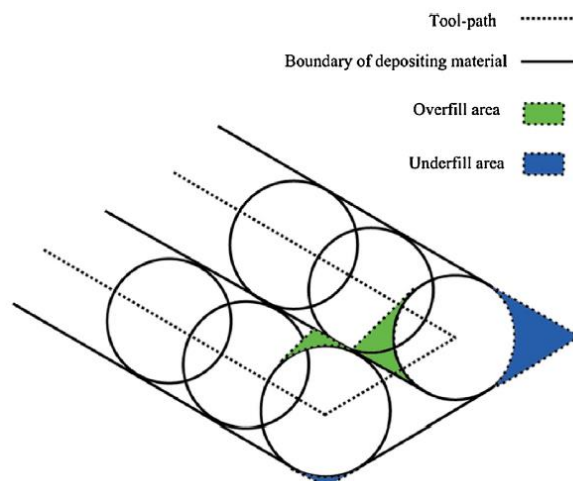


Figura 1.10: Fenomeni del sovrariempimento e sottoriempimento lungo una traiettoria di stampa DP [9]

1.4.3 Traiettoria di deposizione mista

Una strategia comune per aumentare la precisione consiste nell'utilizzare il contour-parallel per le regioni di bordo, dove è richiesta un'elevata qualità superficiale, e il direction-parallel per il riempimento interno, dove risultano più importanti la resistenza del pezzo e l'efficienza del processo di stampa.

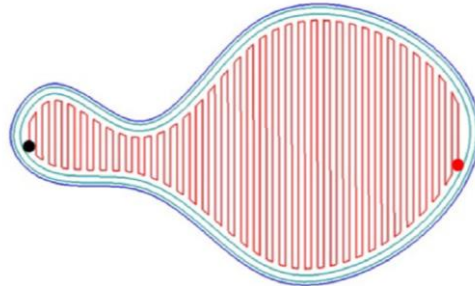


Figura 1.11: Esempio di una traiettoria di deposizione mista [9]

1.4.4 Pianificazione dei percorsi ottimali

Nella pianificazione dei percorsi di stampa, sono particolarmente importanti due fattori: l'angolo di inclinazione delle linee di riferimento e la strategia di collegamento tra i sottopercorsi. L'angolo di inclinazione delle linee di riferimento, ovvero l'angolo formato con l'asse x, può variare tra 0° e 180° . La scelta di questo valore influisce sulla lunghezza totale del percorso, sul numero di segmenti e sul tempo complessivo di costruzione. In Figura 1.12 sono riportate le linee di deposizione e la velocità lungo la traiettoria al variare dell'angolo di inclinazione.

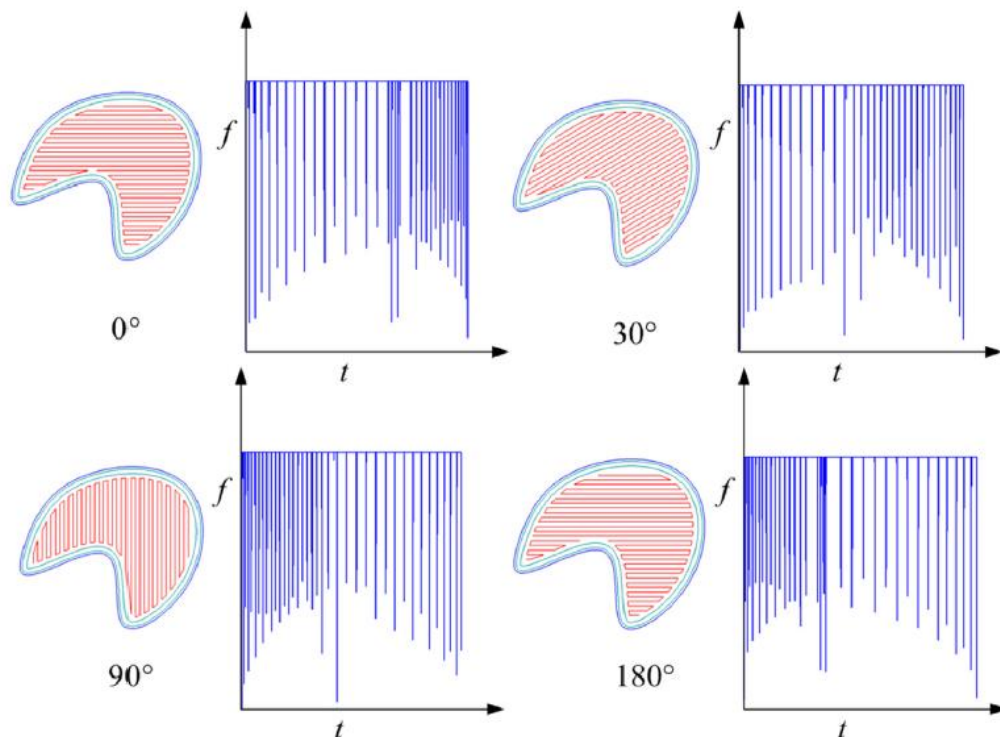


Figura 1.12: Traiettorie di deposizione e curve di velocità per differenti angoli di inclinazione [9]

Per individuare l'inclinazione più efficiente, si riportano su un diagramma la lunghezza della traiettoria e il tempo di lavorazione per diversi valori dell'angolo di inclinazione.

Nell'esempio mostrato in Figura 1.13, l'inclinazione più vantaggiosa corrisponde a un angolo di circa 60°.

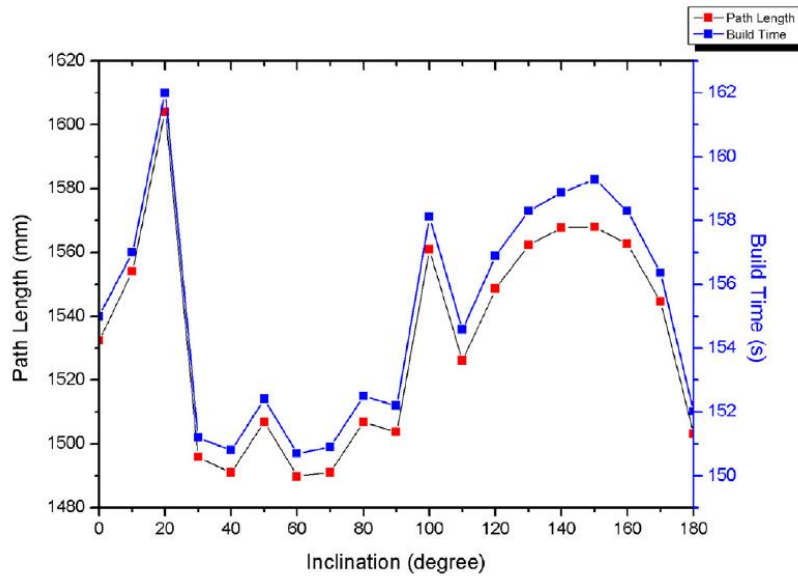


Figura 1.13: Lunghezza della traiettoria e tempo di lavorazione per diverse inclinazioni [9]

Dopo aver stabilito l'angolo di inclinazione delle linee di riempimento, è necessario suddividere il dominio da riempire in sotto-regioni. La strategia di collegamento tra i sottopercorsi diventa fondamentale quando la geometria del pezzo è complessa e non consente di riempire uno strato con un unico percorso continuo. È importante ridurre al minimo il tempo trascorso dall'ugello nei movimenti senza deposizione. Come illustrato nella Figura 1.14, una singola regione può essere divisa in più parti. Nell'esempio, sono state identificate 7 sotto-regioni basate sulle intersezioni tra rette parallele alla direzione di deposizione e i contorni delle cavità interne dello strato. Ciascuna di queste sotto-regioni può essere ulteriormente suddivisa in regioni più piccole. In Figura 1.15 è mostrata la suddivisione in 14 sotto-regioni. Tuttavia, generare un percorso utensile separato per ciascuna sotto-regione non è la strategia ottimale, perché molte di esse possono essere collegate tra loro senza compromettere la qualità della fabbricazione. Questo approccio riduce il numero di attivazioni e disattivazioni dell'estrusore, migliorando sia l'efficienza del processo sia la qualità della deposizione. Come mostrato in Figura 1.16, a partire dalle 14 sotto-regioni si arriva a 4 sottopercorsi dell'ugello. [9]

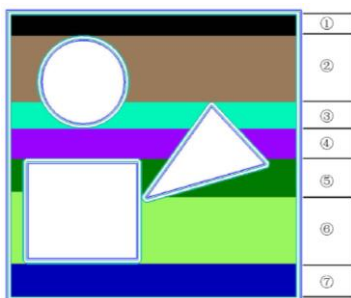


Figura 1.14: Prima partizione in sotto-regioni [9]

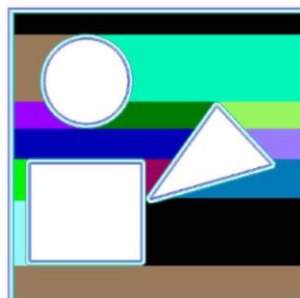


Figura 1.15: Seconda partizione in sotto-regioni [9]

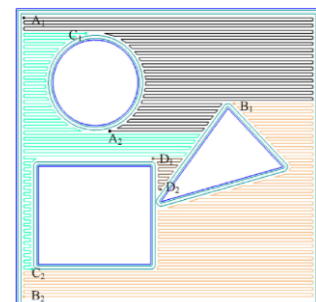


Figura 1.16: Partizione finale e sottopercorsi di stampa [9]

1.5 Tecnologia Fused Deposition Modelling (FDM)

La produzione additiva comprende diverse tecnologie di fabbricazione, che possono essere classificate in base al tipo di materiale impiegato e alla tecnologia della macchina utilizzata per realizzare i componenti. Secondo la classificazione proposta dall'American Society for Testing and Materials (ASTM F42) nel 2010, i processi di produzione additiva possono essere suddivisi in sette categorie principali:

1. Material extrusion techniques;
2. Powder bed fusion techniques;
3. VAT photopolymerization methods;
4. Material jetting techniques;
5. Binder jetting techniques;
6. Sheet lamination techniques;
7. Direct energy deposition techniques.

In questo lavoro di tesi si considera il processo di stampa Fused Deposition Modelling (FDM), classificato come tecnica di estrusione del materiale. La scelta di concentrarsi sull'FDM è dovuta alla sua ampia gamma di applicazioni e alla sua diffusione sia in ambito industriale sia tra gli utenti privati. Inoltre, questa tecnologia di stampa è la più economica e facilmente accessibile per la lavorazione di materiali polimerici.

La tecnica FDM utilizza polimeri sotto forma di filamento. Durante il processo, il filamento viene riscaldato fino allo stato fuso e successivamente spinto attraverso l'ugello della macchina. Seguendo le istruzioni del G-code, la testa dell'ugello può muoversi con tre gradi di libertà per depositare in modo controllato il polimero estruso sul piano di stampa. Come mostrato in Figura 1.17, il filamento viene trascinato all'interno dell'estrusore tramite due rulli controrotanti, che lo spingono attraverso la zona di fusione fino all'ugello. Il materiale fuso viene quindi depositato sul piano di stampa strato dopo strato, fino al completo completamento della geometria del componente. [10]

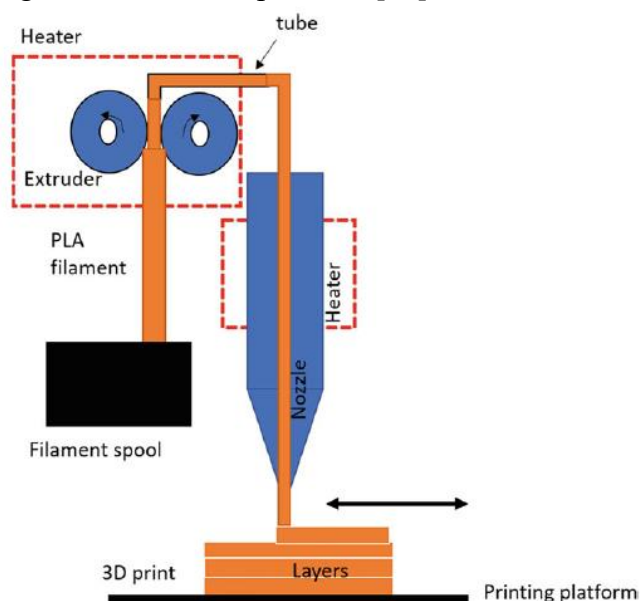


Figura 1.17: Tecnologia FDM [10]

La precisione e l'efficienza del processo di estrusione dipendono, oltre che dai sistemi di trasmissione che muovono l'ugello, soprattutto dalle caratteristiche del filamento termoplastico utilizzato. Per questo motivo, molte stampanti FDM sono progettate per lavorare con specifiche tipologie di materiale.

Generalmente, le stampanti di fascia economica operano principalmente con acido polilattico (PLA). Oltre al PLA, il processo FDM può utilizzare diversi altri materiali, tra cui policaprolattone (PCL), polipropilene (PP), polietilene (PE), polibutilene tereftalato (PBT), ABS, nylon, materiali compositi a base di fibra di carbonio e filamenti contenenti polveri metalliche. La tecnologia FDM trova applicazione in numerosi ambiti industriali e di ricerca. Le applicazioni principali sono: prototipazione rapida, settore biomedicale, componenti automobilistici, industria dei giocattoli e produzione di beni di consumo. [10]

1.5.1 Parametri di processo nella tecnica FDM

La produzione additiva FDM dipende da numerosi parametri, illustrati nella Figura 1.18. Tali parametri sono suddivisi in due grandi categorie: parametri della macchina e parametri del materiale. I primi sono quelli che l'utente imposta nel software di slicing durante la generazione del G-code, mentre i secondi riguardano le caratteristiche del filamento estruso attraverso l'ugello.

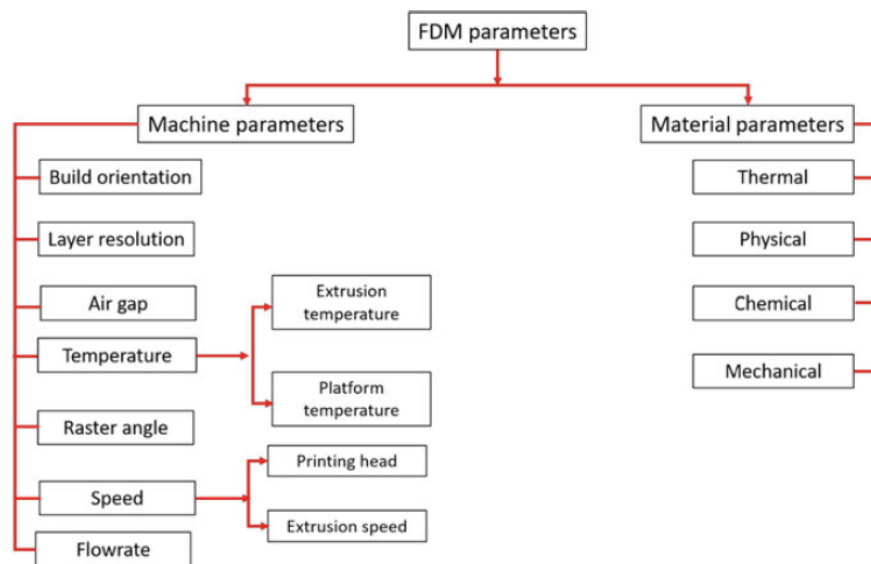


Figura 1.18: Parametri che influenzano il processo FDM [10]

Tra i principali parametri della macchina si possono citare la velocità di stampa, la portata del materiale fuso attraverso l'ugello, l'air gap, l'angolo di raster, la temperatura, l'orientamento di costruzione e la risoluzione del layer.

La temperatura di estrusione è la temperatura necessaria a fondere il filamento e consentirne l'estrusione, mentre la temperatura della piattaforma di stampa è quella applicata al piano per migliorare l'adesione del materiale. L'orientamento di costruzione indica l'angolo con cui la dimensione maggiore del componente è inclinata rispetto al piano di stampa. La risoluzione del layer corrisponde allo spessore minimo di ciascun layer, che può variare da pochi micrometri a diversi millimetri, a seconda della precisione richiesta.

Parallelamente, le proprietà del materiale, in particolare quelle termiche, meccaniche e chimiche, influenzano il processo di estrusione e la qualità finale del pezzo stampato.

Durante la stampa è molto importante impostare la temperatura di estrusione all'interno dell'intervallo di fusione del filamento utilizzato. Infatti, le proprietà termiche del materiale influenzano sia il processo di fusione sia il flusso del materiale attraverso l'ugello della stampante. Le caratteristiche chimiche del filamento determinano la zona di transizione vetrosa, mentre le proprietà meccaniche, come la resistenza e l'attrito del materiale allo stato fuso, condizionano la rigidità del pezzo stampato e la capacità del materiale di fluire correttamente durante l'estrusione. Le proprietà meccaniche sono inoltre legate a fenomeni che possono provocare l'intasamento dell'ugello. [10]

1.5.2 Qualità del componente finito tramite stampa FDM

La qualità di stampa dei componenti prodotti tramite processo FDM dipende principalmente dall'adesione e fusione degli strati di filamento fuso. Condizioni inadeguate di estrusione possono compromettere la precisione geometrica, la qualità superficiale e le prestazioni meccaniche del pezzo. Un limite della stampa FDM è la rugosità superficiale dovuta alle tracce del movimento dell'ugello. Questo fenomeno, chiamato stair-stepping, genera terrazze superficiali e valori elevati di rugosità, che possono limitare l'uso dei pezzi in ambito biomedico e sensoristico. Inoltre, le superfici irregolari causano proprietà meccaniche inferiori e favoriscono la penetrazione di umidità, deteriorando il materiale. Inoltre, difetti del processo di estrusione possono provocare microfratture, porosità e deformazioni.

La rugosità può essere migliorata con trattamenti successivi, sia meccanici, come lucidatura, carteggiatura e finitura a tamburo, sia chimici, come verniciatura, rivestimenti o trattamenti termici. Anche il processo di slicing e lo spessore dei layer influenzano la qualità: layer sottili riducono l'effetto a gradino, ma aumentano i tempi di stampa. Per garantire stampe accurate, è quindi necessario monitorare attentamente il processo. [10]

1.5.3 Stampante FDM cartesiana

Le stampanti FDM cartesiane sono costituite da due elementi principali indipendenti: la struttura su cui è montato l'ugello di stampa e il piano di stampa. Questi componenti si muovono reciprocamente e, per far funzionare il sistema di stampa, è necessaria la possibilità di movimento lungo tre assi ortogonali tra loro. In Figura 1.19 sono riportate le principali architetture di questa tipologia di stampanti. [11] [12]

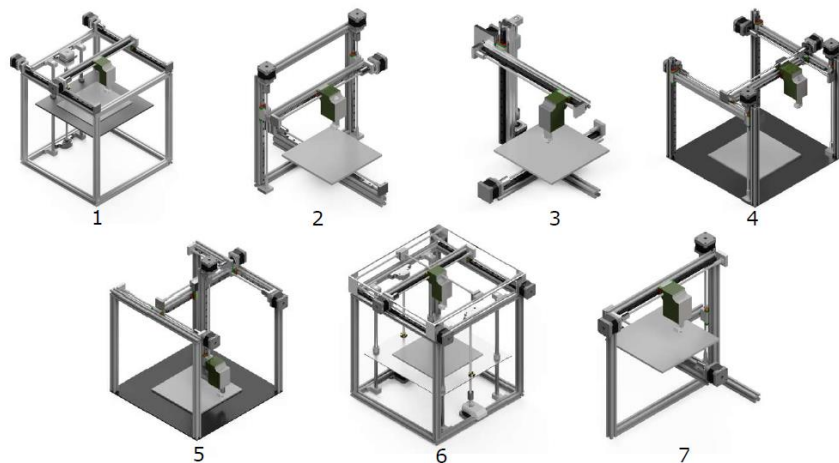


Figura 1.19: Tipologia di architettura delle stampanti FDM cartesiane [11]

Adottando la notazione grafica standard degli schemi cinematici, è possibile rappresentare la cinematica di queste macchine come un modello dotato di tre gradi di libertà traslazionali, forniti da tre gruppi di giunti prismatici ortogonali tra loro. Esempi di schemi delle principali configurazioni delle stampanti cartesiane sono rappresentati in Figura 1.20. Lo schema 1.8a mostra la struttura più semplice, lo schema 1.8b rappresenta la struttura cinematica a ponte più diffusa, e lo schema 1.8c è una variante più rigida.

Gli schemi della colonna sinistra mostrano esclusivamente il modo in cui la struttura può muoversi, tuttavia, occorre considerare anche la rigidità finita dei giunti tra travi, guide lineari e altri componenti strutturali. I giunti del telaio, cioè i punti in cui elementi come travi e supporti sono collegati tra loro, non sono perfettamente rigidi, ma presentano una certa elasticità dovuta ai materiali, oltre a giochi meccanici in corrispondenza di viti e serraggi. Anche i giunti delle guide, ossia i collegamenti tra le guide lineari (binari su cui scorrono i carrelli) e le parti mobili, possono mostrare piccoli discostamenti dovuti a giochi meccanici ed elasticità. Come prima approssimazione, tali giunti possono essere considerati giunti sferici.

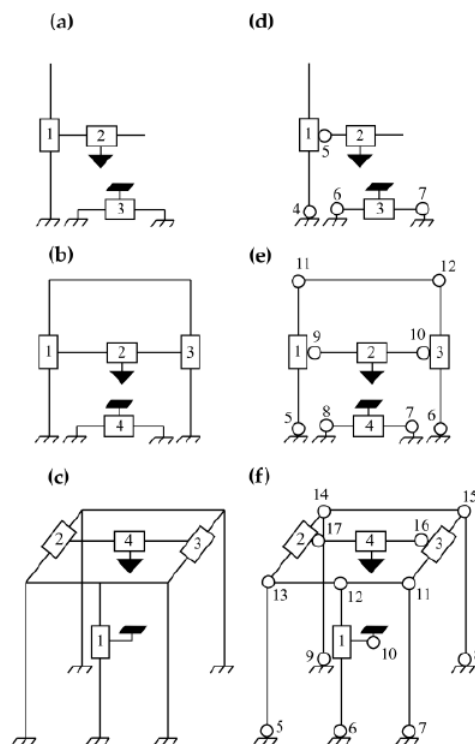


Figura 1.20: Rappresentazioni di schemi cinematici tipici delle stampanti 3D cartesiane. I rettangoli indicano giunti prismatici, i cerchi giunti sferici, il triangolo nero corrisponde alla testina di stampa e il parallelepipedo nero al piano di stampa. Gli schemi (a-c) illustrano schemi cinematici funzionali, mentre gli schemi (d-f) includono anche i giunti del telaio e delle guide, i quali non possono essere considerati completamente rigidi. [11]

Le stampanti cartesiane sono generalmente dotate di almeno tre motori, ciascuno responsabile del movimento lungo uno degli assi del sistema cartesiano: X, Y e Z.

Gli assi X e Y determinano le coordinate di stampa all'interno di ciascun layer, mentre l'asse Z interviene per spostare l'ugello di un'altezza pari allo spessore del layer successivo, muovendo il piano di stampa o l'ugello. Se il modello presenta parti sporgenti, si inseriscono supporti temporanei sotto di esse.

La maggior parte delle stampanti ad uso domestico adotta questa struttura cartesiana grazie alla sua diffusione e al basso costo di produzione. Il principale vantaggio di queste stampanti è la loro ampia diffusione, che facilita la risoluzione di problemi comuni e la reperibilità di componenti di ricambio. Tuttavia, l'architettura della stampante cartesiana presenta il limite che, a velocità elevate, la qualità di stampa diminuisce a causa del peso delle strutture mobili. Inoltre, il volume complessivo della macchina è decisamente più grande del volume effettivo di costruzione, poiché la macchina deve consentire il movimento completo di tutti gli assi. Nel presente lavoro di tesi si analizzerà una traiettoria di stampa nel piano XY per una stampante FDM cartesiana. Di conseguenza, l'asse Z non sarà preso in considerazione. Inoltre, nei modelli sviluppati nei prossimi capitoli, il movimento dell'ugello lungo gli assi X e Y è realizzato tramite carrelli scorrevoli su guide, azionati da motori passo-passo ibridi mediante trasmissioni a cinghia dentata.

2 Modellazione di un sistema cinghia-carrello-guida in cui si presenta il fenomeno dello stick-slip

In questo capitolo si analizza il comportamento di un carrello di una stampante FDM cartesiana che si sposta lungo una guida orizzontale tramite una trasmissione a cinghia.

Il sistema sarà modellato in Simscape come un insieme massa-molla-smorzatore con attrito al fine di osservare il fenomeno dello stick-slip e l'influenza dei vari parametri.

2.1 Modello di attrito

La forza di attrito è funzione della velocità relativa ed è data dalla somma dei contributi di Stribeck, di Coulomb e del termine viscoso. (Figura 2.1)

L'attrito di Stribeck, F_{St} , si manifesta a basse velocità e presenta una pendenza negativa, si oppone quindi al moto, ma il suo valore diminuisce all'aumentare della velocità relativa. L'attrito di Coulomb, F_{Coul} , si oppone al moto con una forza costante, indipendentemente dalla velocità. L'attrito viscoso, F_v , si oppone al moto con una forza direttamente proporzionale alla velocità relativa.

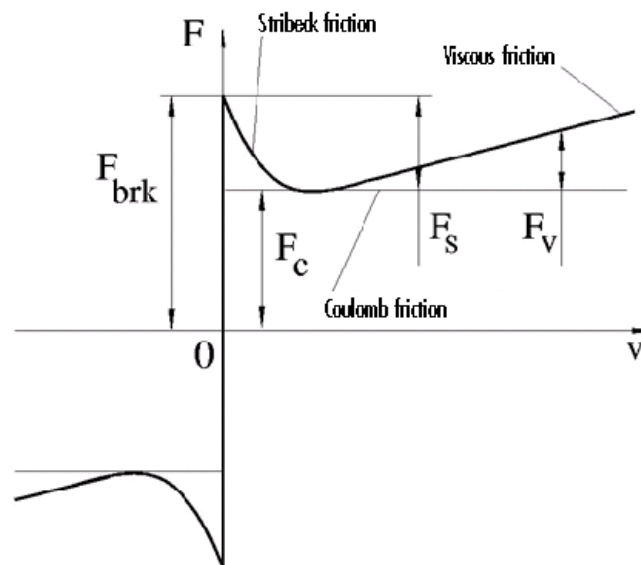


Figura 2.1: Andamento della forza di attrito in funzione della velocità relativa [13]

La somma degli attriti di Coulomb e di Stribeck in prossimità della velocità nulla è definita attrito di distacco (breakaway friction). La curva del grafico F - v è approssimata mediante la seguente relazione: [13]

$$F = \sqrt{2e} \cdot (F_{brk} - F_c) \cdot \exp\left[-\left(\frac{v}{v_{St}}\right)^2\right] \cdot \frac{v}{v_{St}} + F_c \cdot \tanh\left(-\frac{v}{v_{Coul}}\right) + c \cdot v$$

dove:

- F è la forza di attrito;

- F_c è la forza di attrito di Coulomb;
- F_{brk} è la forza di attrito di distacco;
- v_{brk} è la velocità di distacco;
- v_{St} è la velocità del contributo di Stribeck;
- v_{Coul} è la velocità del contributo di Coulomb;
- v è la velocità relativa $v = v_R - v_C$;
- c è il coefficiente di attrito viscoso.

L'andamento della curva, ottenuto come somma dei tre contributi di attrito, è continuo su tutto il dominio della velocità relativa e consente di descrivere il fenomeno meccanico nel modo più realistico possibile, sia in prossimità dello stato di quiete sia a velocità elevate. In ogni intervallo di velocità prevale quindi un determinato contributo.

2.1.1 Regione di velocità prossima allo zero

In prossimità della velocità relativa nulla, la forza di attrito raggiunge il valore massimo, pari alla forza di attrito di distacco F_{brk} . Ciò è dovuto al fatto che le superfici a contatto non sono perfettamente lisce, ma presentano microscopiche asperità che si incastrano tra loro formando microlegami nei punti di contatto.

Per porre il corpo in movimento è necessario rompere tali microlegami e, per farlo, è richiesta una forza maggiore rispetto a quella necessaria quando il moto è già iniziato. Quando la forza esterna supera la forza di attrito massima, i microlegami si rompono e, una volta avviato il movimento, non hanno più il tempo di riformarsi, di conseguenza, la forza di attrito diminuisce.

Subito dopo il distacco prevale il contributo dell'attrito di Stribeck, modellato mediante la seguente funzione esponenziale:

$$F_{St} = \sqrt{2e} \cdot (F_{brk} - F_c) \cdot \exp\left[-\left(\frac{v}{v_{St}}\right)^2\right] \cdot \frac{v}{v_{St}}$$

Questa funzione presenta le seguenti caratteristiche:

- vale 0 N per $v = 0$ m/s;
- cresce rapidamente per velocità molto basse;
- successivamente decresce fino a diventare trascurabile;
- è simmetrica rispetto all'origine.

La velocità $v_{St} = v_{brk}\sqrt{2}$ determina l'estensione dell'effetto Stribeck: quanto più v_{St} è piccolo, tanto più brusco risulta il calo dell'attrito.

2.1.2 Regione di velocità intermedie

All'aumentare della velocità relativa, il contributo di Stribeck diventa trascurabile e, per velocità intermedie, domina l'attrito di Coulomb, modellato dalla seguente funzione:

$$F_{Coul} = F_c \cdot \tanh\left(-\frac{v}{v_{Coul}}\right)$$

Questa funzione presenta le seguenti caratteristiche:

- vale 0 N per $v = 0$ m/s;
- cresce rapidamente per velocità molto basse;
- si satura ai valori $+F_c$ o $-F_c$.

Il parametro $v_{\text{Coul}} = \frac{v_{\text{brk}}}{10}$ determina la ripidità della transizione. Idealmente dovrebbe essere molto piccolo, ma in simulazione non può assumere valori troppo ridotti, al fine di evitare problemi numerici dovuti al passaggio brusco tra $+F_c$ e $-F_c$.

In questa regione il contributo viscoso non è ancora rilevante, poiché le velocità non sono sufficientemente elevate.

2.1.3 Regione di velocità elevata

In questa regione diventa significativo il contributo dell'attrito viscoso, associato a fenomeni dissipativi quali la presenza di lubrificanti o smorzatori:

$$F_v = c \cdot v$$

Questo termine introduce un andamento lineare con pendenza positiva che si somma al contributo costante dell'attrito di Coulomb. Risulta quindi una retta traslata verticalmente di $+F_c$ o $-F_c$, a seconda che v sia rispettivamente maggiore o minore di zero. Pertanto, a velocità elevate, la forza di attrito cresce linearmente in modulo

2.1.4 Velocità negative

Come già accennato, l'andamento della forza di attrito per velocità negative è simmetrico rispetto all'origine. Tutti e tre i contributi cambiano segno in modo da opporsi sempre al moto relativo.

2.2 Fenomeno dello stick-slip

Il modello di attrito precedentemente descritto è particolarmente rilevante, soprattutto a basse velocità, nell'analisi del moto di un carrello lungo una guida lineare di una stampante FDM, dove è richiesta un'elevata precisione di posizionamento. Il fenomeno di stick-slip alterna fasi di adesione (stick), in cui il carrello resta fermo rispetto alla guida, a fasi di scorrimento (slip), in cui il carrello si muove improvvisamente. Tale fenomeno è responsabile di perdite di precisione nella deposizione del materiale.

Durante la fase di stick, il carrello è fermo e la forza applicata dal motore tramite la cinghia è inferiore alla forza di attrito di distacco F_{brk} (attrito statico). In queste condizioni, la forza motrice non riesce a muovere il corpo, ma provoca una deformazione elastica nei componenti del sistema, con conseguente accumulo di energia elastica.

Quando la forza elastica supera la forza di attrito di distacco, il carrello entra in movimento e inizia la fase di slip. Tuttavia, appena si instaura il moto relativo tra le superfici, la forza di attrito non rimane costante a F_{brk} , ma diminuisce rapidamente a causa dell'effetto Stribeck fino al valore dell'attrito di Coulomb F_c (attrito dinamico). Questa brusca diminuzione della forza resistente genera un'accelerazione impulsiva e non uniforme del carrello, liberando

l'energia elastica accumulata. Durante l'accelerazione, la forza elastica si riduce e gli elementi elastici, come la cinghia, si rilassano. Quando la forza elastica diventa minore della forza di attrito, la risultante delle forze si oppone al moto e la velocità diminuisce. Se la velocità del carrello diminuisce fino a diventare nulla, si innesca un nuovo ciclo di stick-slip.

2.3 Descrizione del sistema

Il sistema in esame è costituito da un carrello di massa m che scorre lungo una guida lineare e viene movimentato da un motore tramite una cinghia dentata. La cinghia è modellata come una molla con smorzatore, collegata tra motore e carrello. I valori dei parametri del modello sono riportati in Tabella 2.1.

Tabella 2.1: Parametri del modello

	Parametro	Valore	u.d.m
Massa del carrello	m	0.5	kg
Coefficiente di attrito statico	μ_s	0.3	-
Coefficiente di attrito dinamico	μ_d	0.15	-
Rigidezza della cinghia	k	$1 \cdot 10^4$	$\frac{N}{m}$
Smorzamento della cinghia	c	10	$\frac{Ns}{m}$
Smorzamento nel contatto carrello-guida (lubrificazione)	c'	$1 \cdot 10^{-4}$	$\frac{Ns}{m}$
Carico normale agente sul carrello	N	10	N

L'inerzia del motore non viene considerata.

L'obiettivo consiste nel simulare il comportamento del carrello considerando due tipi di moto imposto dal motore:

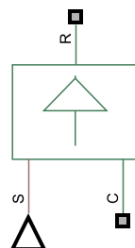



- Moto uniforme a velocità costante $v_m = 10 \frac{mm}{s}$
- Moto sinusoidale con ampiezza $x_{m0} = 1 mm$ e frequenza $f = 2 Hz$

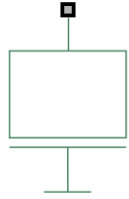
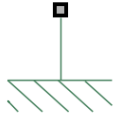
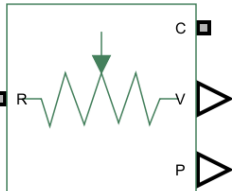
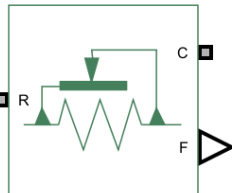
2.4 Implementazione modello in Simulink/Simscape

I blocchi Simscape utilizzati nel modello sono riportati in Tabella 2.2.

Tabella 2.2: Blocchi Simscape utilizzati nel modello [14]



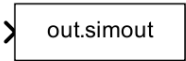


Ideal Translational Velocity Source	
Il blocco rappresenta una sorgente ideale di velocità che genera una differenza di velocità ai suoi terminali proporzionale al segnale fisico in ingresso. La sorgente è ideale, cioè è in grado di mantenere la velocità specificata indipendentemente dalla forza esercitata sul sistema.	

<p>Le connessioni R e C sono porte meccaniche in traslazione, mentre la porta S consente l'ingresso del segnale fisico, ovvero del segnale di controllo che pilota la sorgente.</p> <p>La velocità relativa è determinata come $v = v_R - v_C$, dove v_R e v_C sono rispettivamente le velocità assolute delle porte R e C.</p>	
<p style="text-align: center;">Translational Spring</p> <p>Rappresenta una molla lineare ideale, descritta dalle seguenti relazioni:</p> $F = k \cdot x ; x = x_{in} + x_R - x_C ; v = \frac{dx}{dt}$ <p>Dove:</p> <ul style="list-style-type: none"> - F è la forza trasmessa attraverso la molla; - k è il coefficiente di rigidità della molla; - x è la posizione relativa, corrispondente alla deformazione della molla; - x_{in} è la posizione iniziale, corrispondente alla deformazione iniziale della molla; - x_R e x_C sono le posizioni assolute delle porte R e C; - v è la velocità relativa. <p>La forza è positiva se agisce da R a C.</p>	
<p style="text-align: center;">Translational Damper</p> <p>Rappresenta uno smorzatore viscoso lineare ideale, descritto dalle seguenti relazioni:</p> $F = c \cdot v ; v = v_R - v_C$ <p>Dove:</p> <ul style="list-style-type: none"> - F è la forza trasmessa attraverso lo smorzatore; - c è il coefficiente di smorzamento; - v è la velocità relativa; - v_R e v_C sono le velocità assolute delle porte R e C. <p>La forza è positiva se agisce da R a C.</p>	
<p style="text-align: center;">Translational Friction</p> <p>Questo blocco rappresenta l'attrito nel contatto tra corpi in movimento. La forza di attrito è funzione della velocità relativa e, come già descritto, è dato dalla somma dei contributi di Stribeck, Coulomb e viscoso:</p> <p>[6]</p> $F = \sqrt{2e} \cdot (F_{brk} - F_c) \cdot \exp\left[-\left(\frac{v}{v_{St}}\right)^2\right] \cdot \frac{v}{v_{St}} + F_c \cdot \tanh\left(-\frac{v}{v_{Coul}}\right) + c \cdot v$	

<p>La porta R va connessa al corpo in movimento traslatorio (carrello), mentre la porta C va collegata al corpo fisso (guida). I parametri da inserire nel blocco sono i seguenti:</p> <ul style="list-style-type: none"> - breakaway friction force (F_{brk}) - breakaway friction velocity (v_{brk}) - Coulomb friction force (F_c) - viscous friction coefficient (c) 	
<p style="text-align: center;">Mass</p> <p>Rappresenta una massa ideale in moto traslatorio, descritto dalla seguente equazione:</p> $F = m \frac{dv}{dt}$	
<p style="text-align: center;">Mechanical Translational Reference</p> <p>Per fissare rigidamente al telaio le porte meccaniche in traslazione, è necessario il blocco Mechanical Translational Reference.</p>	
<p style="text-align: center;">Ideal Translational Motion Sensor</p> <p>Rappresenta un sensore ideale che converte una variabile misurata tra due nodi meccanici in traslazione in un segnale di controllo proporzionale all'accelerazione, alla velocità o alla posizione.</p> <p>Le connessioni R e C sono porte meccaniche in traslazione, che collegano il blocco ai nodi in cui viene misurato il moto.</p> <p>La velocità viene misurata come $v = v_R - v_C$, dove v_R e v_C sono rispettivamente le velocità assolute delle porte R e C.</p> <p>Le porte A, V e P forniscono in uscita i segnali corrispondenti all'accelerazione, alla velocità e alla posizione.</p>	
<p style="text-align: center;">Ideal Force Sensor</p> <p>Rappresenta un sensore ideale che converte una variabile che passa attraverso il sensore, in un segnale di controllo proporzionale alla forza.</p> <p>Le connessioni R e C sono porte meccaniche in traslazione.</p> <p>Il blocco deve essere inserito in serie al blocco di cui si vuole misurare la forza in uscita e mai in parallelo, poiché in questo caso influirebbe sui risultati della simulazione. La direzione positiva del blocco va dalla porta R alla porta C, quindi una forza positiva applicata alla porta R genera un segnale di uscita positivo. La porta F fornisce l'uscita del segnale fisico di forza.</p>	

Per generare i segnali e acquisirli per successive analisi, è necessario collegare ai blocchi Simscape i blocchi Simulink riportati in Tabella 2.3

Tabella 2.3: Blocchi Simulink utilizzati nel modello [15]

<p>Constant Genera un segnale costante nel tempo. È utile per fornire il segnale di velocità costante v_m.</p>	
<p>Sine Wave Genera un segnale sinusoidale continuo nel tempo. È utile per fornire il segnale sinusoidale di posizione x_m, con ampiezza x_{m0} e frequenza f</p>	
<p>To Workspace Serve per registrare i dati dei segnali di uscita (posizione, velocità e forza) e inviarli al Workspace di MATLAB. I dati possono essere salvati come Array, Structure o Timeseries. Questo blocco è fondamentale per analizzare i segnali e generare i grafici.</p>	
<p>Simulink - PS Converter Converte un segnale Simulink in un segnale fisico compatibile con Simscape. È necessario impostare l'unità di misura del segnale.</p>	
<p>PS- Simulink Converter Converte un segnale fisico Simscape in un segnale Simulink. È necessario impostare l'unità di misura del segnale</p>	

Collegando i blocchi Simulink e Simscape, si crea il modello del carrello di massa m che scorre lungo una guida e viene movimentato da un motore tramite una cinghia dentata.

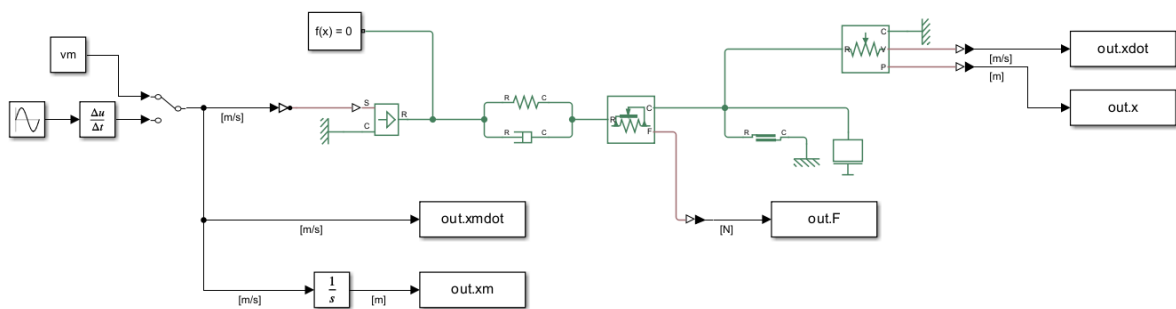


Figura 2.2: Modello cinghia-carrello-attrito

Si noti che la cinghia è stata modellata come una rigidità e uno smorzatore in parallelo.

Tramite un blocco Manual Switch è possibile selezionare la tipologia di moto da inviare in input al blocco Ideal Translational Velocity Source. Poiché tale sorgente richiede in ingresso un segnale di velocità, il segnale sinusoidale di posizione deve essere derivato.

Durante la simulazione, tramite i blocchi To Workspace, vengono salvati i dati relativi ai segnali:

- spostamento e velocità imposti dal motore $x_m(t)$ e $\dot{x}_m(t)$
- spostamento e velocità del carrello $x(t)$ e $\dot{x}(t)$
- forza trasmessa al carrello $F(t)$

2.5 Solver

Per avviare la simulazione è necessario collegare il blocco Solver Configuration al modello Simscape.

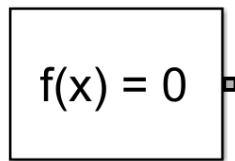


Figura 2.3: Blocco Solver Configuration

Questo blocco fornisce al risolutore numerico le informazioni necessarie per gestire le equazioni differenziali e algebriche presenti nel modello e, in assenza di esso, la simulazione non può essere eseguita. Il blocco Solver Configuration è inoltre collegato in modo funzionale alle impostazioni del solver di Simulink, accessibili nella sezione Solver Information, visibile in basso a destra nell'ambiente di simulazione.

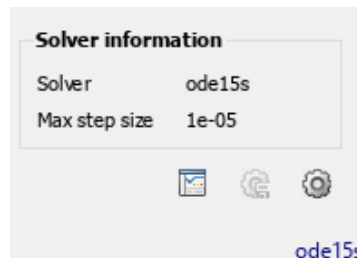


Figura 2.4: Sezione Solver Information

Le impostazioni del solver di Simulink possono essere modificate consentendo di scegliere il tipo di risolutore, il passo di integrazione e le tolleranze numeriche (Figura 2.5). È molto importante selezionare correttamente questi parametri, poiché influenzano significativamente la precisione e la stabilità della simulazione del sistema fisico.

Quindi, il blocco Solver Configuration fornisce le informazioni necessarie per gestire il sistema Simscape, mentre Simulink utilizza tali informazioni insieme alle impostazioni globali definite nella sezione Solver Information per eseguire la simulazione.

Per evitare errori numerici e garantire una corretta rappresentazione del fenomeno di stick-slip, sono state adottate le seguenti impostazioni del solver:

- Type: Variable-step

- Solver: ode15s (stiff/NDF)
- Max step size: 1e-5
- Relative tolerance: 1e-6
- Absolute tolerance: 1e-3

Figura 2.5: Impostazioni del solver

2.6 Definizione dei parametri del sistema

Una volta implementato il modello in Simulink/Simscape, è necessario definire i parametri dei vari blocchi tramite uno script in MATLAB.

```

%% Parametri

m = 0.5;           % massa [kg]
mus = 0.3;        % coefficiente di attrito statico
mud = 0.15;       % coefficiente di attrito dinamico
c = 10;           % coefficiente di attrito viscoso cinghia [Ns/m]
k = 1e4;          % rigidezza [N/m]
N = 10;           % carico normale [N]
F_brk = mus*N;    % forza di attrito di distacco [N]
F_Coul = mud*N;   % forza di Coulomb [N]
v_brk = 1e-5;     % velocità di distacco [m/s]
c1 = 1e-4;        % coefficiente di attrito viscoso guida [Ns/m]

vm = 10e-3;       % velocità costante imposta dal motore [m/s]
xm0 = 1e-3;       % ampiezza posizione forma sinusoidale [m]
f = 2;            % frequenza forma sinusoidale [Hz]

```

I coefficienti di attrito, il carico normale e la velocità di distacco sono necessari al blocco Translational Friction, poiché questo blocco richiede (Figura 2.7):

- la forza di attrito di distacco $F_{brk} = \mu_s N$;
- la forza di attrito di Coulomb $F_c = \mu_d N$;
- la velocità di distacco v_{brk} ;

- lo smorzamento del contatto carrello-guida c' ;

Si ricorda che v_{brk} è la velocità per cui la somma dell'attrito di Stribeck e di Coulomb raggiunge il valore della forza di attrito di distacco.

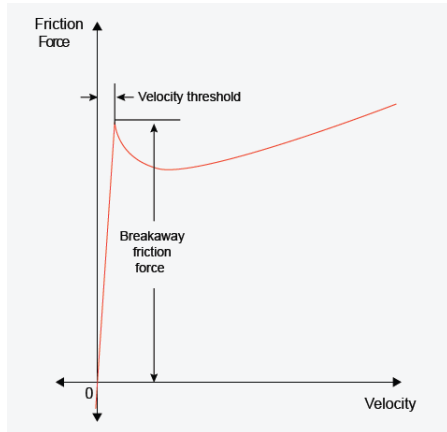


Figura 2.6: Velocità di distacco v_{brk}

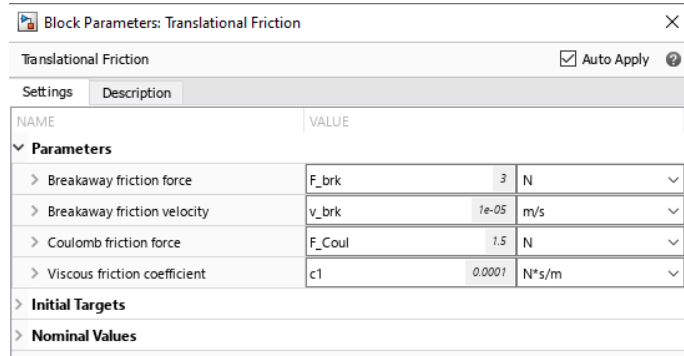


Figura 2.7: Impostazioni del blocco Translational Friction

Per evitare instabilità numeriche, si utilizza un valore di Breakaway Friction Velocity pari a $1 \cdot 10^{-5} m/s$.

2.7 Risultati della simulazione

Il modello è stato creato e i parametri sono stati definiti, a questo punto è possibile avviare la simulazione per un intervallo di tempo di 1 s e rappresentare graficamente i risultati.

Di seguito si riporta lo script MATLAB completo.

```

clc
clear all
close all

%% Parametri

m = 0.5;           % massa [kg]
mus = 0.3;        % coefficiente di attrito statico
mud = 0.15;       % coefficiente di attrito dinamico
c = 10;           % coefficiente di attrito viscoso cinghia[Ns/m]
k = 1e4;          % rigidezza [N/m]
N = 10;           % carico normale [N]
F_brk = mus*N;    % forza di attrito di distacco [N]
F_Coul = mud*N;   % forza di Coulomb [N]
v_brk = 1e-5;     % velocità di distacco [m/s]
c1 = 1e-4;        % coefficiente di attrito viscoso guide [Ns/m]

vm = 10e-3;       % velocità costante imposta dal motore [m/s]
xm0 = 1e-3;       % ampiezza posizione forma sinusoidale [m]
f = 2;            % frequenza forma sinusoidale [Hz]

%% Simulazione modello Simscape

simOut = sim('Es_attrito')

```

```

%% Estrazione dati

t = simOut.tout;           % [s]
x = simOut.x.Data * 1e3;  % [mm]
xm = simOut.xm.Data * 1e3; % [mm]
xdot = simOut.xdot.Data * 1e3; % [mm/s]
xmdot = simOut.xmdot.Data * 1e3; % [mm/s]
F = simOut.F.Data;       % [N]

%% Plot

figure(1)
plot(t, xm, t, x, 'LineWidth', 1.5)
xlabel ('Tempo [s]')
ylabel ('Spostamento [mm]')
legend ('Spostamento imposto dal motore', 'Spostamento carrello')
title('Spostamento imposto vs effettivo')
grid on

figure(2)
plot(t, xmdot, t, xdot, 'LineWidth', 1.5)
xlabel ('Tempo [s]')
ylabel ('Velocità [mm/s]')
legend ('Velocità imposta dal motore', 'Velocità carrello')
title('Velocità imposta vs effettiva')
grid on

figure(3)
plot(t, x, t, xdot, t, F, 'LineWidth', 1.5)
xlabel ('Tempo [s]')
ylabel ('Spostamento [mm], Velocità [mm/s], Forza [N]')
legend('Spostamento carrello', 'Velocità carrello', 'Forza trasmessa al carrello')
title ('Analisi fenomeno stick e slip')
grid on

```

2.7.1 Moto uniforme a velocità costante

Si avvia la simulazione per il moto uniforme a velocità costante $v_m = 10 \frac{mm}{s}$.

Per valutare come il carrello risponde al moto imposto dal motore tramite la cinghia, è utile confrontare lo spostamento del carrello $x(t)$ con lo spostamento imposto dal motore $x_m(t)$ (Figura 2.8), e successivamente la velocità del carrello $\dot{x}(t)$ con la velocità imposta dal motore $\dot{x}_m(t)$ (Figura 2.9).

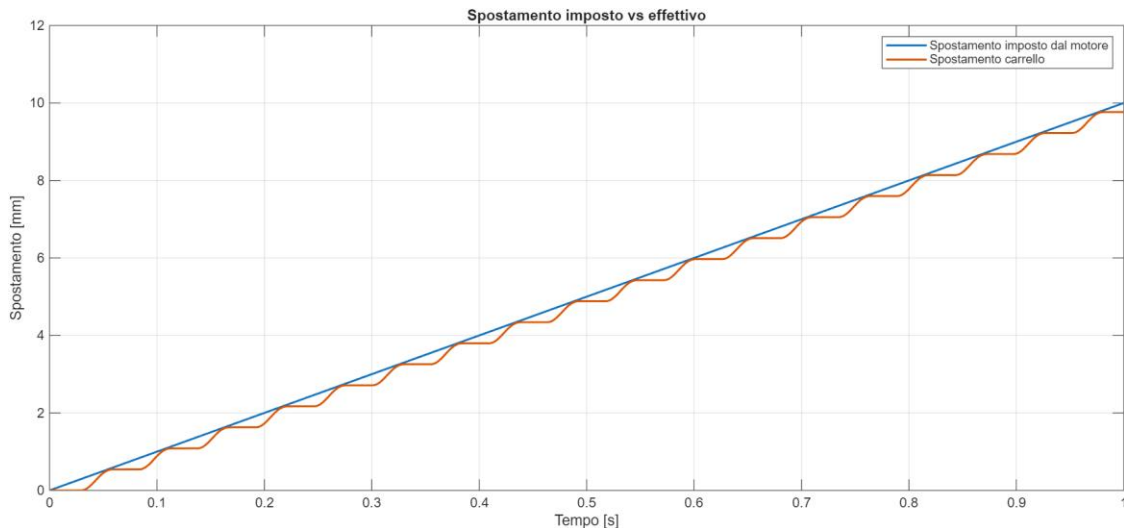


Figura 2.8: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$

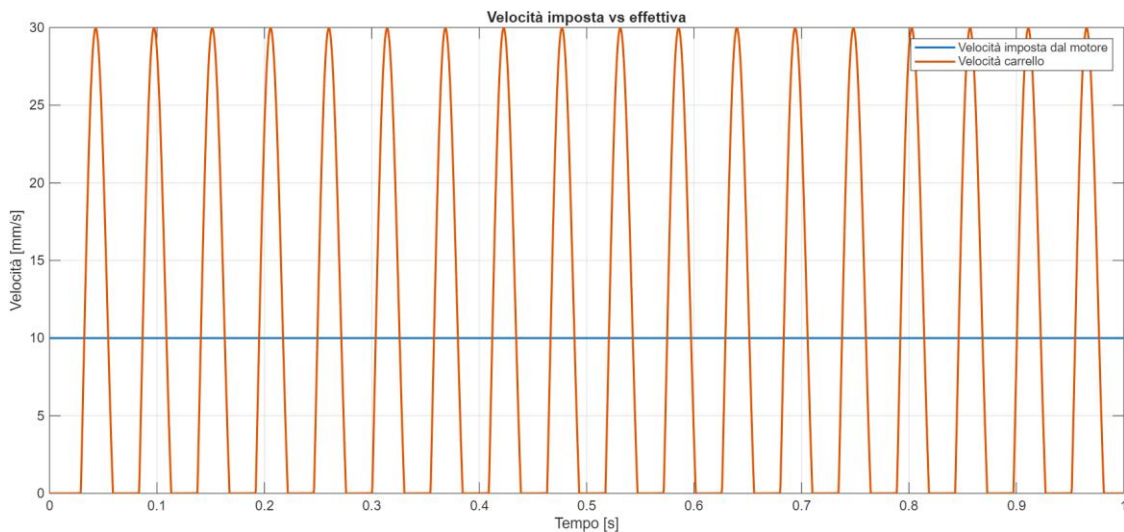


Figura 2.9: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$

Si osserva che lo spostamento del carrello nel tempo è una curva che rincorre lo spostamento imposto dal motore. La curva è caratterizzata da un andamento a gradini che alterna tratti in cui il carrello resta fermo e tratti in cui il carrello avanza per poi fermarsi di nuovo.

Per quanto riguarda la velocità del carrello, questa non è costante, ma alterna picchi di velocità, a tratti in cui la velocità è nulla.

Questi andamenti sono causati dal fenomeno dello stick-slip. Al fine di analizzare questo fenomeno si confrontano lo spostamento $x(t)$, la velocità $\dot{x}(t)$ e la forza applicata dal motore $F(t)$. (Figura 2.10)

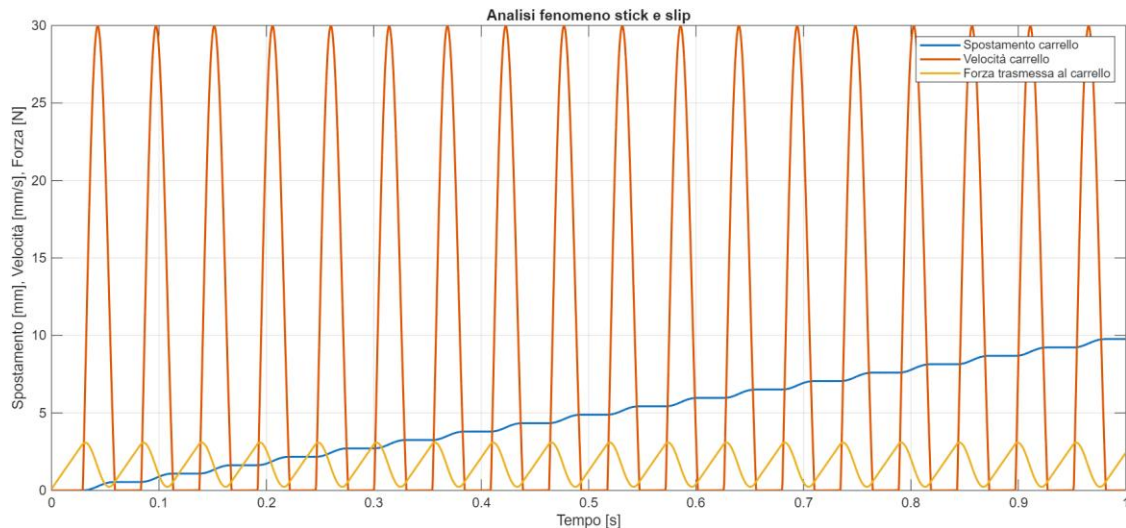


Figura 2.10: Confronto tra spostamento $x(t)$, velocità $\dot{x}(t)$ e forza applicata dal motore $F(t)$

Le condizioni di transizione da stick a slip sono le seguenti:

- Transizione da stick a slip

$$|F| > F_{brk}$$

- Transizione da slip a stick

$$[\dot{x}(t) = 0] \wedge [|F| < F_{brk}]$$

Nota la forza di distacco $F_{brk} = 3 N$, è possibile verificare se i gradini nello spostamento siano effettivamente dovuti al fenomeno di stick-slip.

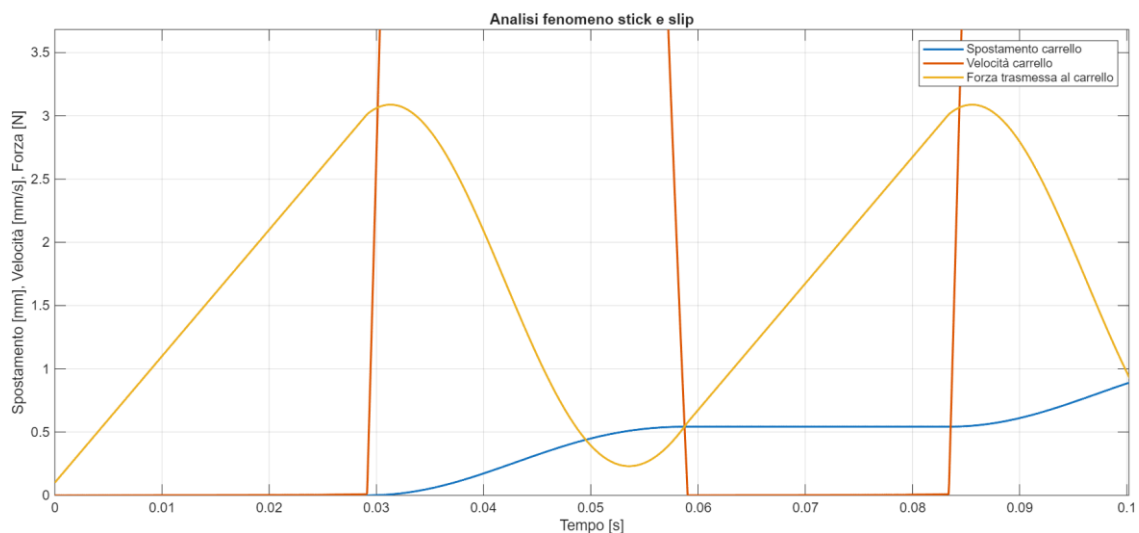


Figura 2.11: Analisi del fenomeno dello stick-slip

Si osservano diverse fasi temporali:

- I. La forza trasmessa al carrello aumenta, ma resta inferiore a $F_{brk} = 3N$, di conseguenza, il carrello rimane fermo e l'energia elastica della cinghia aumenta progressivamente;
- II. La forza trasmessa al carrello supera F_{brk} , quindi si entra nella condizione di slip. Il carrello inizia a muoversi, ma per l'effetto Stribeck la forza resistente di attrito

- diminuisce rapidamente a basse velocità relative. La forza elastica della cinghia è allora molto maggiore della forza resistente, generando un'accelerazione impulsiva e un rapido aumento della velocità del carrello. In questa fase l'energia elastica accumulata viene parzialmente convertita in energia cinetica del carrello;
- III. Man mano che il carrello accelera, la deformazione della cinghia diminuisce e quindi la forza elastica si riduce fino a quando $|F| = F_{Coul}$. In questo istante l'accelerazione si annulla e la velocità raggiunge il suo valore massimo. Successivamente, poiché $|F| < F_{Coul}$, la risultante delle forze si oppone al moto, quindi l'accelerazione cambia segno e la velocità diminuisce fino a tornare a zero;
 - IV. Quando la velocità del carrello è nulla e la forza applicata è minore di F_{brk} , il carrello entra nuovamente nella condizione di stick;
 - V. Successivamente, la forza applicata aumenta di nuovo e, quando supera F_{brk} , si rientra nella condizione di slip, dando inizio a un nuovo ciclo di stick-slip.

2.7.2 Moto sinusoidale di posizione

Si avvia ora la simulazione per il moto sinusoidale di posizione con ampiezza $x_{m0} = 1 \text{ mm}$ e frequenza $f = 2 \text{ Hz}$.

Anche in questo caso il carrello segue lo spostamento desiderato con un andamento a gradino, dovuto al fenomeno dello stick e slip.

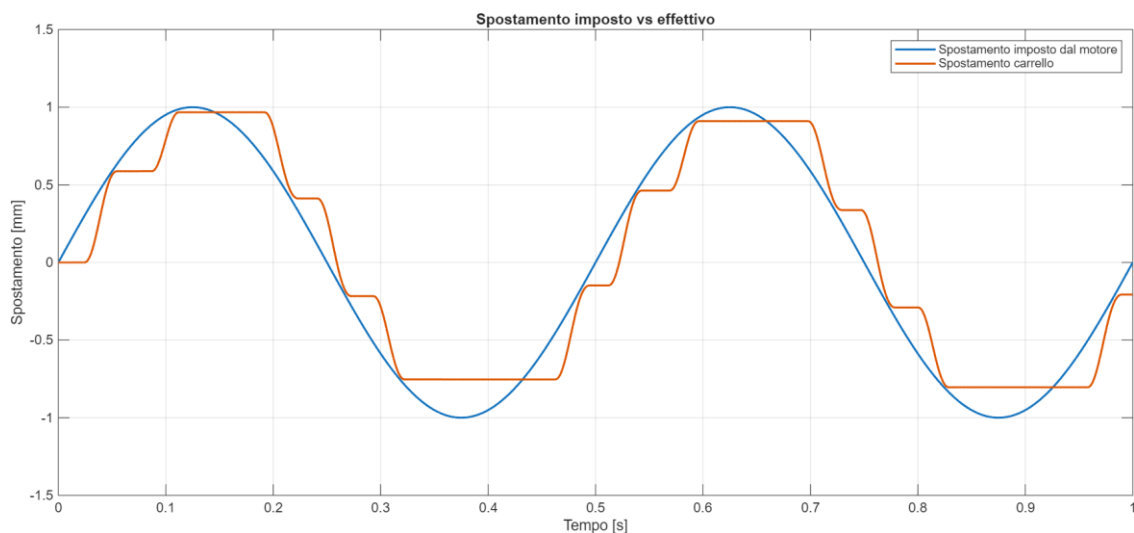


Figura 2.12: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$

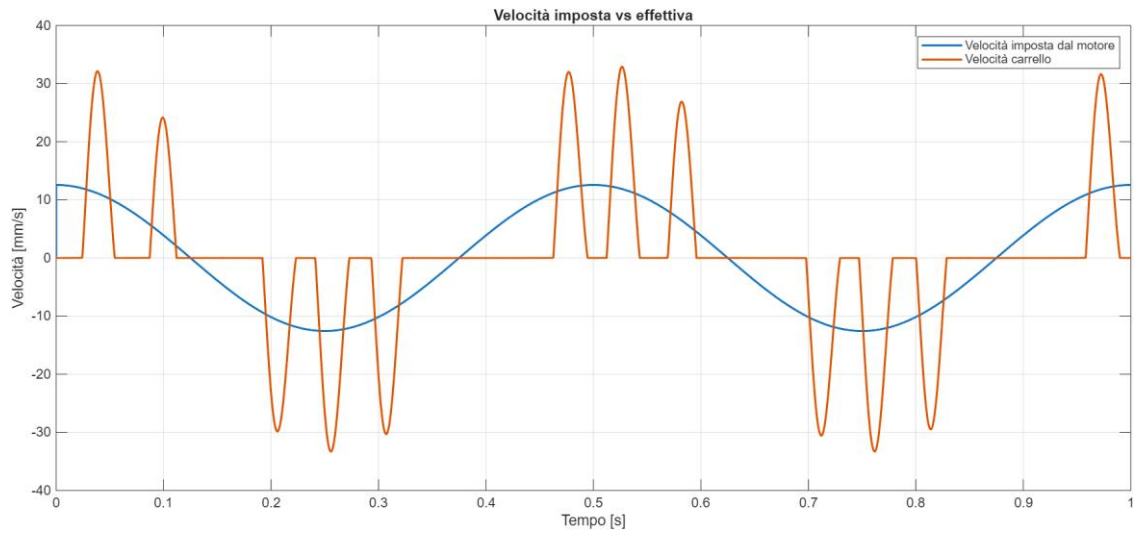


Figura 2.13: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$

Nella Figura 2.14 è riportato il confronto tra lo spostamento e la velocità del carrello, insieme alla forza trasmessa al carrello.

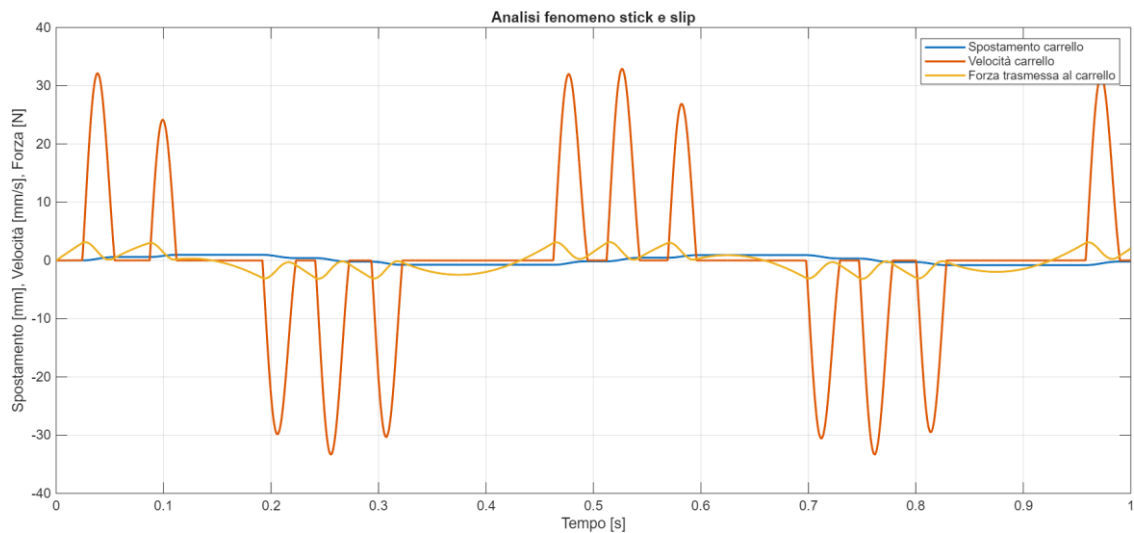


Figura 2.14: Confronto tra spostamento $x(t)$, velocità $\dot{x}(t)$ e forza applicata dal motore $F(t)$

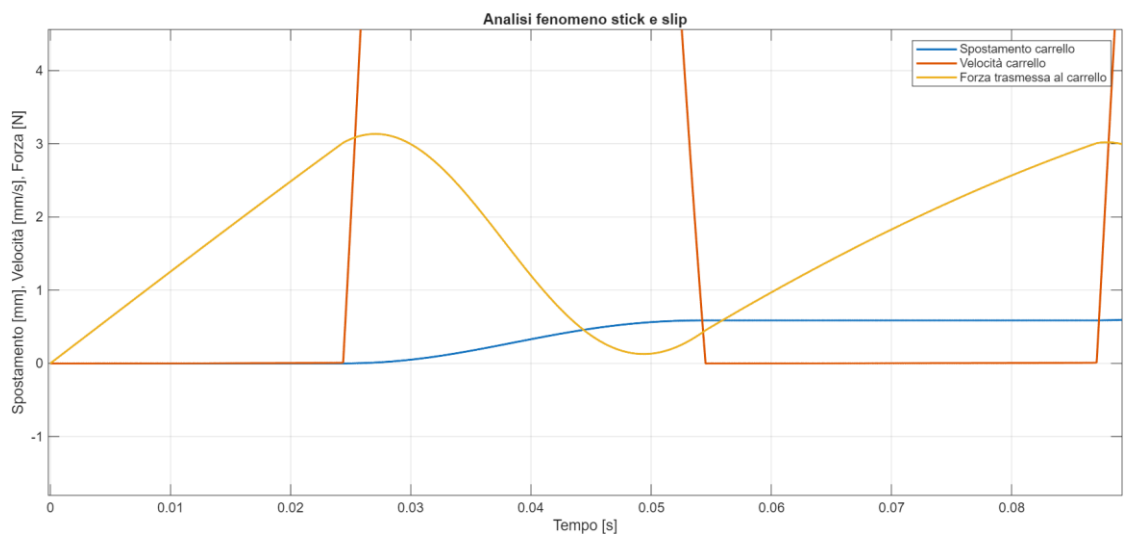


Figura 2.15: Analisi del fenomeno dello stick-slip

Come nel caso del moto a velocità costante, anche qui (Figura 2.15) si osserva la presenza dello stick-slip e valgono le stesse considerazioni.

2.8 Variazione dei parametri del modello, caso velocità costante

Poiché i risultati ottenuti sono fortemente influenzati dal fenomeno dello stick-slip, è necessario valutare come i parametri del sistema incidano sulla precisione del moto del carrello. L'obiettivo consiste quindi nel ridurre al minimo il fenomeno dello stick-slip, studiando l'influenza dei singoli parametri variandoli uno alla volta: rigidità k , smorzamento c , massa m , coefficiente di attrito statico μ_s e dinamico μ_d .

Per ciascuna variazione si generano i grafici che mostrano:

- Spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$
- Velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$
- Forza trasmessa al carrello

2.8.1 Variazione della rigidità della cinghia

Lo script è stato modificato in modo da variare esclusivamente il parametro di rigidità k , mantenendo costanti tutti gli altri parametri del modello. I valori di rigidità considerati sono $k = [0.5 \cdot 10^4 \text{ N/m}; 1 \cdot 10^4 \text{ N/m}; 1 \cdot 10^5 \text{ N/m}]$. Tali valori sono stati scelti in modo da coprire un intervallo sufficientemente ampio, così da poter valutare in maniera significativa le differenze negli andamenti del sistema al variare della rigidità, sia in aumento sia in diminuzione.

```
clc
clear all
close all

%% Parametri costanti

m = 0.5;           % massa [kg]
mus = 0.3;        % coefficiente di attrito statico
mud = 0.15;       % coefficiente di attrito dinamico
c = 10;           % coefficiente di attrito viscoso [Ns/m]
N = 10;           % carico normale [N]
F_brk = mus*N;    % forza di attrito di distacco [N]
F_Coul = mud*N;   % forza di Coulomb [N]
v_brk = 1e-5;     % velocità di distacco [m/s]
c1 = 1e-4;        % coefficiente di attrito viscoso guide [Ns/m]

vm = 10e-3;       % velocità costante imposta dal motore [m/s]
xm0 = 1e-3;       % ampiezza posizione forma sinusoidale [m]
f = 2;            % frequenza forma sinusoidale [Hz]

%% Parametro variabile

valori_k = [0.5e4 1e4 1e5]; % valori di rigidità da confrontare [N/m]
colors = lines(length(valori_k));

%% Preparazione figure
```

```

figure(1)
title('Spostamento imposto vs effettivo (variazione k)')
xlabel('Tempo [s]')
ylabel('Spostamento [mm]')
grid on
hold on

figure(2)
title('Velocità imposta vs effettiva (variazione k)')
xlabel('Tempo [s]')
ylabel('Velocità [mm/s]')
grid on
hold on

figure(3)
title('Forza trasmessa al carrello (variazione k)')
xlabel('Tempo [s]')
ylabel('Forza [N]')
grid on
hold on

%% Ciclo sui valori di k e plot

for i = 1:length(valori_k)

    k = valori_k(i); % aggiorna rigidezza nel workspace

    % Simulazione modello Simscape
    simOut = sim('Es_attrito');

    % Estrazione dati
    t = simOut.tout; % [s]
    x = simOut.x.Data * 1e3; % [mm]
    xm = simOut.xm.Data * 1e3; % [mm]
    xdot = simOut.xdot.Data * 1e3; % [mm/s]
    xmdot = simOut.xmdot.Data * 1e3; % [mm/s]
    F = simOut.F.Data; % [N]

    % Grafico 1: Spostamento
    figure(1)
    if i == 1
        plot(t, xm, 'k--', 'LineWidth', 1.5, 'DisplayName', 'spostamento imposto');
    end
    plot(t, x, '-', 'Color', colors(i,:), 'LineWidth', 1.5, 'DisplayName', sprintf('x
effettivo (k = %.1e N/m)', k));

    % Grafico 2: Velocità
    figure(2)
    if i == 1
        plot(t, xmdot, 'k--', 'LineWidth', 1.5, 'DisplayName', 'velocità imposta');
    end
    plot(t, xdot, '-', 'Color', colors(i,:), 'LineWidth', 1.5, 'DisplayName',
sprintf('$\dot{x}$ effettiva (k = %.1e N/m)', k));

    % Grafico 3: Forza
    figure(3)

```

```

    plot(t, F, '-', 'Color', colors(i,:), 'LineWidth', 1.5, 'DisplayName', sprintf('k =
    %.1e N/m', k));
end

%% Legende

figure(1)
legend('Location', 'best', 'Interpreter', 'latex')

figure(2)
legend('Location', 'best', 'Interpreter', 'latex')

figure(3)
legend('Location', 'best', 'Interpreter', 'latex')

```

Di seguito, sono riportati i risultati della simulazione al variare della rigidezza k . Si osserva che, all'aumentare della rigidezza della cinghia, lo spostamento del carrello (Figura 2.16) tende sempre di più a seguire lo spostamento imposto dal motore poiché le deformazioni elastiche della cinghia diventano progressivamente minori. La distanza massima tra le due curve diminuisce e, allo stesso tempo, aumenta il numero di gradini (Figura 2.17). Di conseguenza, le fasi di stick e di slip hanno durata minore. Il fenomeno dello stick-slip rimane presente, ma influisce meno sulla precisione del sistema, poiché l'energia elastica accumulata durante la fase di stick è inferiore. Di conseguenza, aumentare la rigidezza della cinghia è vantaggioso per ridurre sensibilmente l'effetto negativo dello stick-slip sulla precisione.

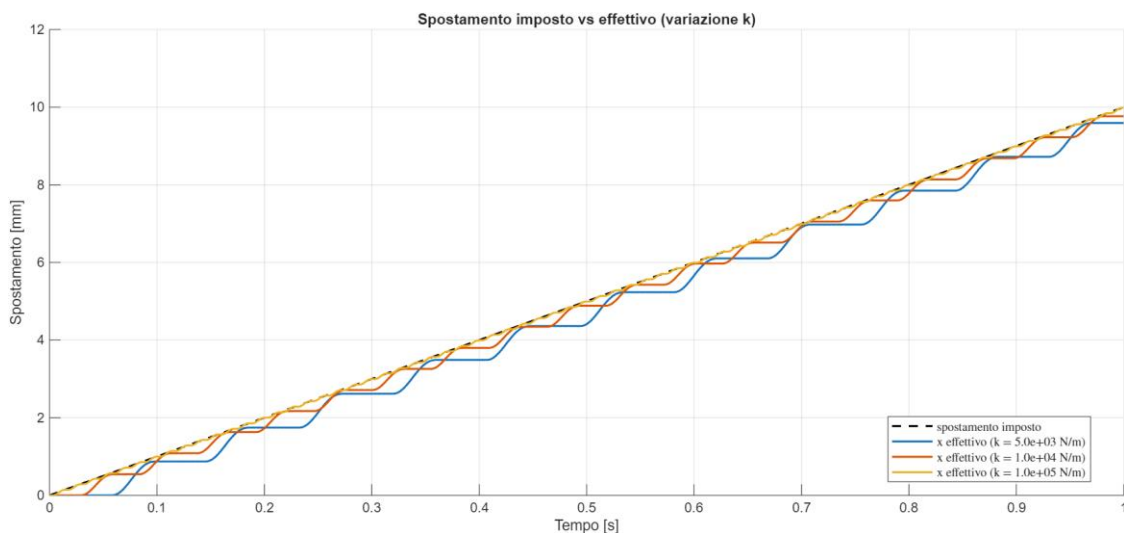


Figura 2.16: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della rigidezza della cinghia

Dai grafici di velocità e forza si osserva che, all'aumentare della rigidezza della cinghia, i picchi di velocità diventano più bassi ma più frequenti (Figura 2.18), mentre i picchi di forza sono più alti e anch'essi più frequenti (Figura 2.19). Ciò si spiega con il fatto che, aumentando k , le deformazioni elastiche della cinghia diminuiscono e, di conseguenza, la forza generata dal motore viene trasmessa in maniera più diretta al carrello.

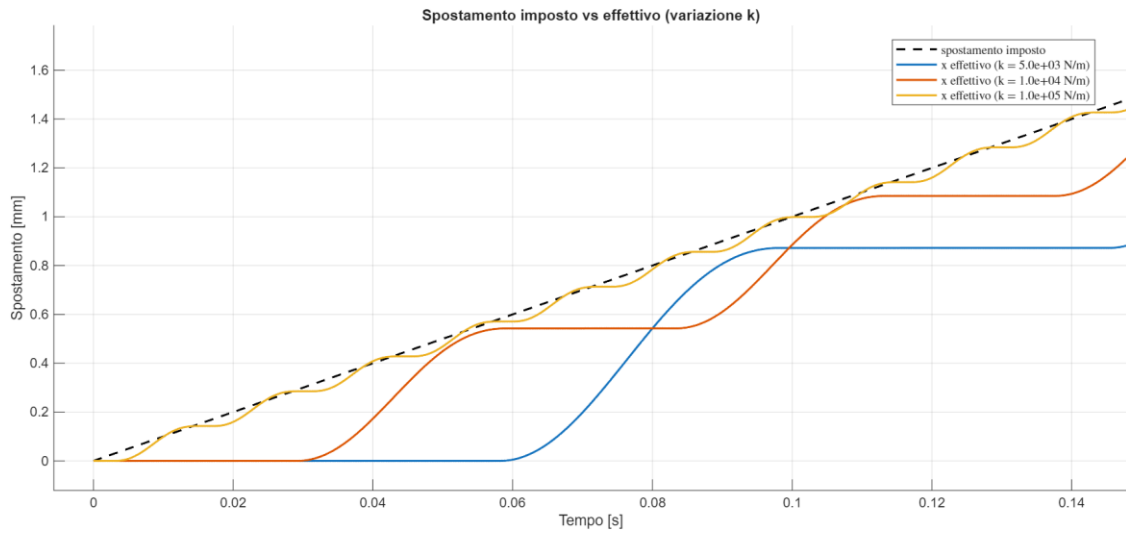


Figura 2.17: Ingrandimento della Figura 2.16

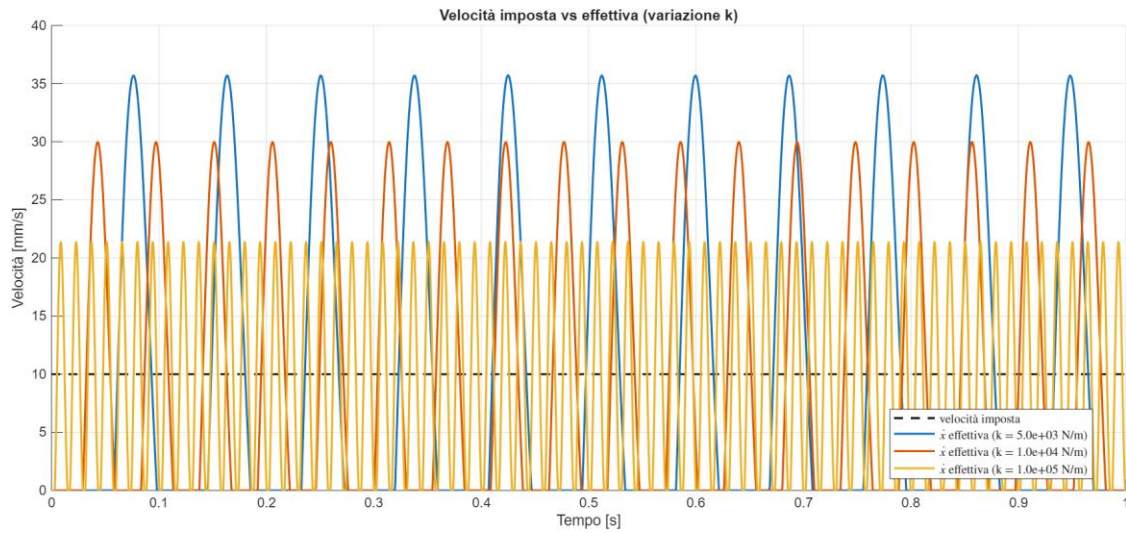


Figura 2.18: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare della rigidità della cinghia

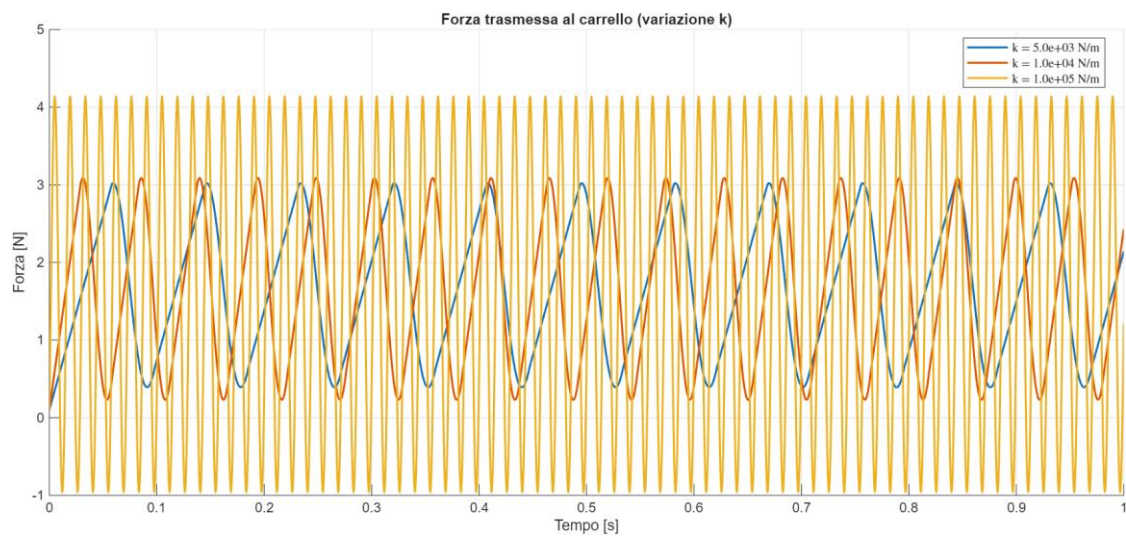


Figura 2.19: Andamento della forza trasmessa al carrello al variare della rigidità della cinghia

I picchi di forza raggiungono quindi ampiezze maggiori e una frequenza superiore, superando con maggiore facilità la forza di attrito di distacco. Questo accade perché il carrello, per inerzia, si muove con un ritardo rispetto alla cinghia e, nel tempo di ritardo, il motore continua a imporre spostamento, aumentando ulteriormente la forza.

Di conseguenza, all'aumentare della rigidità, l'energia elastica accumulata durante la fase di stick è minore, ma viene rilasciata molto più rapidamente. L'accelerazione, essendo proporzionale alla forza, assume quindi un andamento sempre più impulsivo, con picchi elevati ma concentrati in tempi molto brevi.

La velocità del carrello, essendo l'integrale dell'accelerazione, risente meno dei picchi impulsivi, derivano oscillazioni di ampiezza ridotta ma ad alta frequenza.

Quindi, aumentando la rigidità della cinghia, lo stick-slip non scompare, ma si manifesta come cicli molto piccoli e ad alta frequenza. Ciò migliora la precisione della posizione del carrello e riduce gli effetti macroscopici dello stick-slip.

Tuttavia, un aumento eccessivo della rigidità può incrementare troppo i picchi di forza e accelerazione, con conseguenti maggiori sollecitazioni sui componenti del sistema e problemi di vibrazione e rumore. È quindi necessario trovare un compromesso tra il miglioramento della precisione del moto e il contenimento delle sollecitazioni dinamiche. Una cinghia ad elevata rigidità può essere utilizzata solo se il sistema è in grado di sopportare i picchi elevati di forza e accelerazione.

2.8.2 Variazione dello smorzamento della cinghia

Lo script è stato modificato in modo da variare esclusivamente il parametro di smorzamento c , mantenendo costanti tutti gli altri parametri del modello. I valori di smorzamento considerati sono $c = [1 \text{ Ns/m}; 10 \text{ Ns/m}; 100 \text{ Ns/m}]$. Di seguito sono riportati i risultati della simulazione al variare dello smorzamento c .

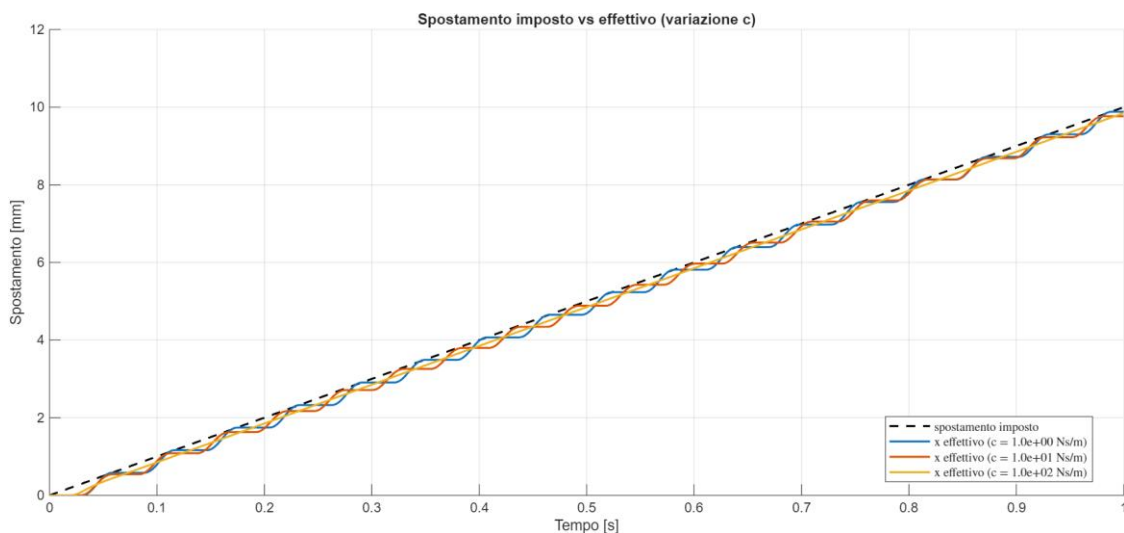


Figura 2.20: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare dello smorzamento della cinghia

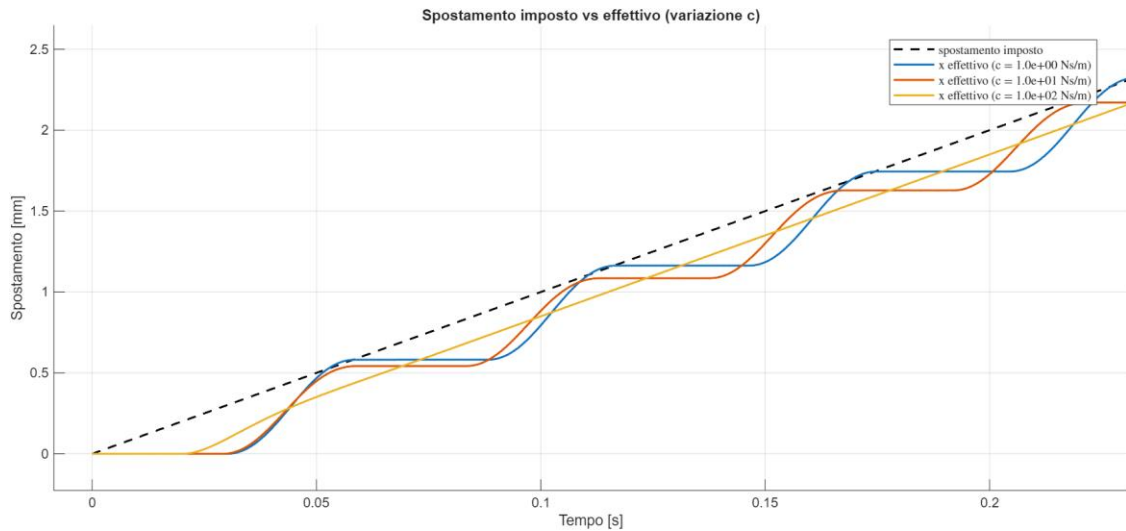


Figura 2.21: Ingrandimento Figura 2.20

Lo smorzamento viscoso introduce una forza dissipativa proporzionale alla velocità di deformazione della cinghia. Tale forza non immagazzina energia, ma la dissipa sotto forma di calore. Osservando lo spostamento del carrello (Figura 2.20) si nota che, all'aumentare dello smorzamento viscoso, l'ampiezza delle fasi di stick-slip si riduce e i gradini diventano sempre più piccoli e ravvicinati. Superato un certo valore di smorzamento, il fenomeno di stick-slip scompare e la posizione del carrello, dopo un breve ritardo, assume un andamento lineare a pendenza costante, tuttavia, la curva risulta traslata rispetto alla posizione desiderata. Ciò è dovuto al fatto che la forza viscosa richiede una deformazione permanente della cinghia per mantenere il moto.

Per quanto riguarda la velocità (Figura 2.22), all'aumentare dello smorzamento, l'ampiezza dei picchi diminuisce e le oscillazioni si attenuano progressivamente fino a ridursi, per valori elevati di smorzamento, a un singolo picco. Successivamente, la velocità converge rapidamente verso il valore costante di 10 mm/s imposto dal motore. Questo comportamento è molto positivo, poiché il sistema entra in un regime privo di instabilità dinamiche

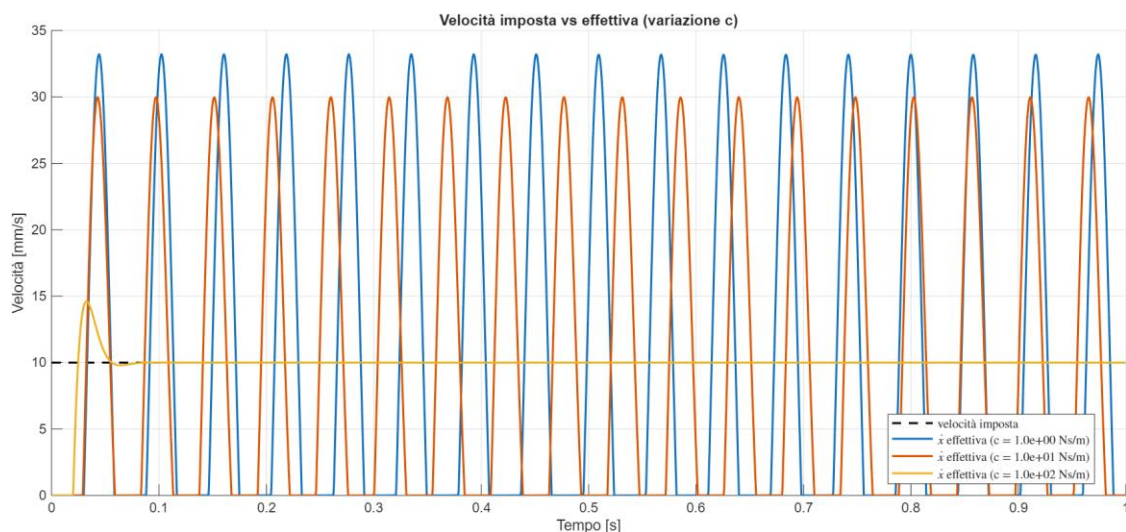


Figura 2.22: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare dello smorzamento della cinghia

Infine, per quanto riguarda la forza, all'aumentare dello smorzamento l'overshoot rispetto alla forza di distacco si riduce. Questo avviene perché lo smorzamento comporta una maggiore dissipazione dell'energia fornita dal motore durante la fase di adesione, limitando la crescita della deformazione elastica della cinghia.

Di conseguenza, l'energia elastica accumulata prima del distacco risulta inferiore rispetto al caso di basso smorzamento. Per valori elevati di smorzamento, la forza presenta un unico picco iniziale e successivamente diventa costante nel tempo, pari alla forza di attrito di Coulomb (il contributo dell'attrito viscoso lungo le guide è molto basso).

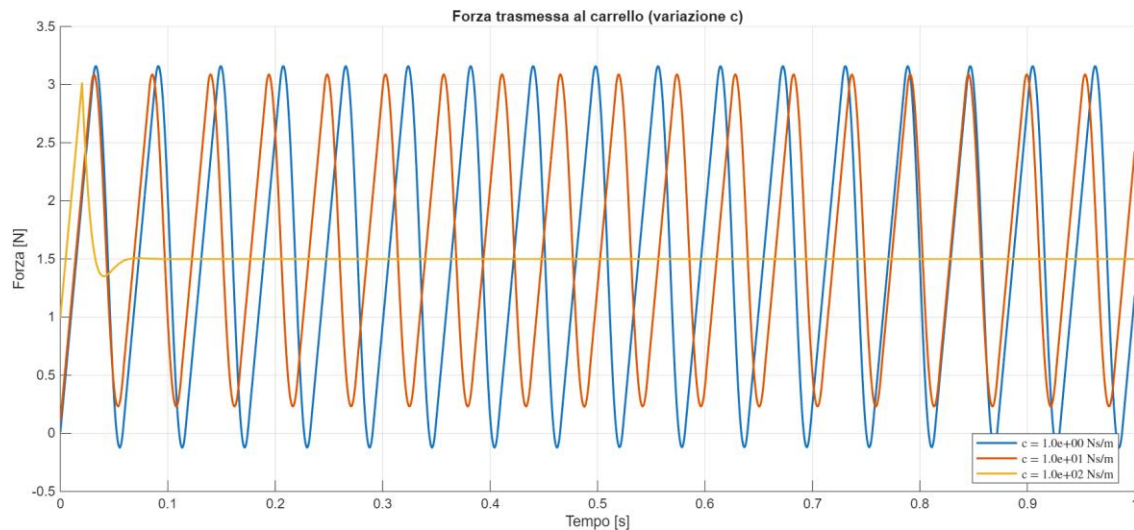


Figura 2.23: Andamento della forza trasmessa al carrello al variare dello smorzamento della cinghia

Quindi l'aumento dello smorzamento comporta una riduzione fino all'eliminazione dello stick-slip, una diminuzione dei picchi di forza e accelerazione, e una conseguente riduzione delle instabilità dinamiche.

Tuttavia, comporta anche effetti negativi come la comparsa di un offset permanente nella posizione del carrello, perdita di reattività della risposta e maggiori perdite energetiche. Di conseguenza, valori di smorzamento troppo elevati possono rappresentare un problema in applicazioni a elevate velocità. È quindi necessario trovare un compromesso tra la riduzione delle instabilità dinamiche e la precisione del moto.

2.8.3 Variazione della massa del carrello

Lo script è stato modificato in modo da variare esclusivamente il parametro di massa m , mantenendo costanti tutti gli altri parametri del modello. I valori di massa considerati sono $m = [0.1 \text{ kg}; 0.5 \text{ kg}; 1 \text{ kg}]$. Di seguito sono riportati i risultati della simulazione al variare della massa m .

Diminuendo la massa del carrello, la sua inerzia si riduce, rendendolo più reattivo alla forza trasmessa dal motore. Per quanto riguarda la curva della posizione (Figura 2.24), i gradini dovuti al fenomeno dello stick-slip diventano più piccoli e più frequenti, poiché l'energia necessaria a superare l'attrito statico è inferiore.

I picchi di velocità, al diminuire della massa, diventano più alti e ravvicinati (Figura 2.25), poiché la minore inerzia implica accelerazioni più immediate. Di conseguenza, si osservano oscillazioni più frequenti ma di breve durata.

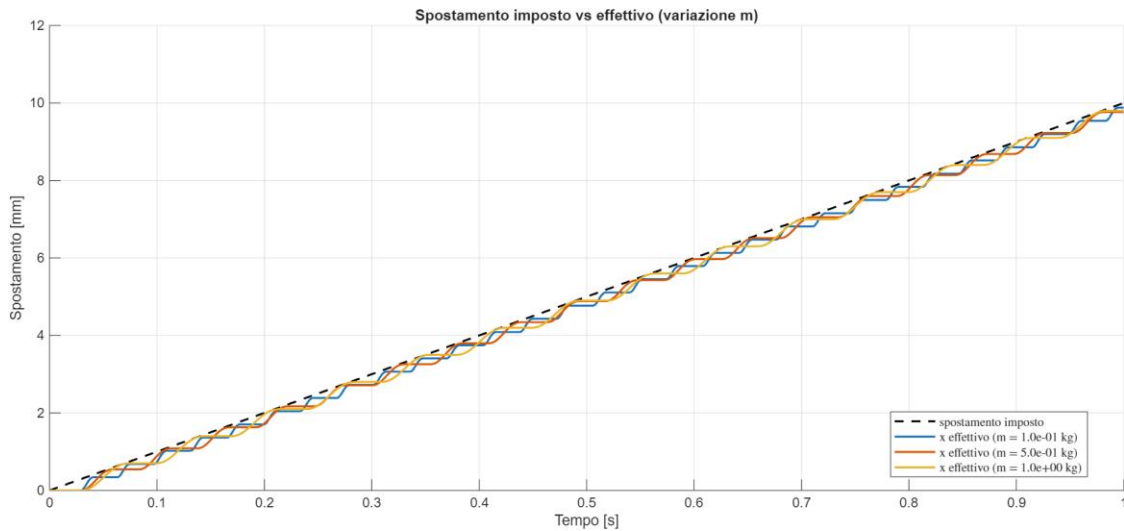


Figura 2.24: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della massa del carrello

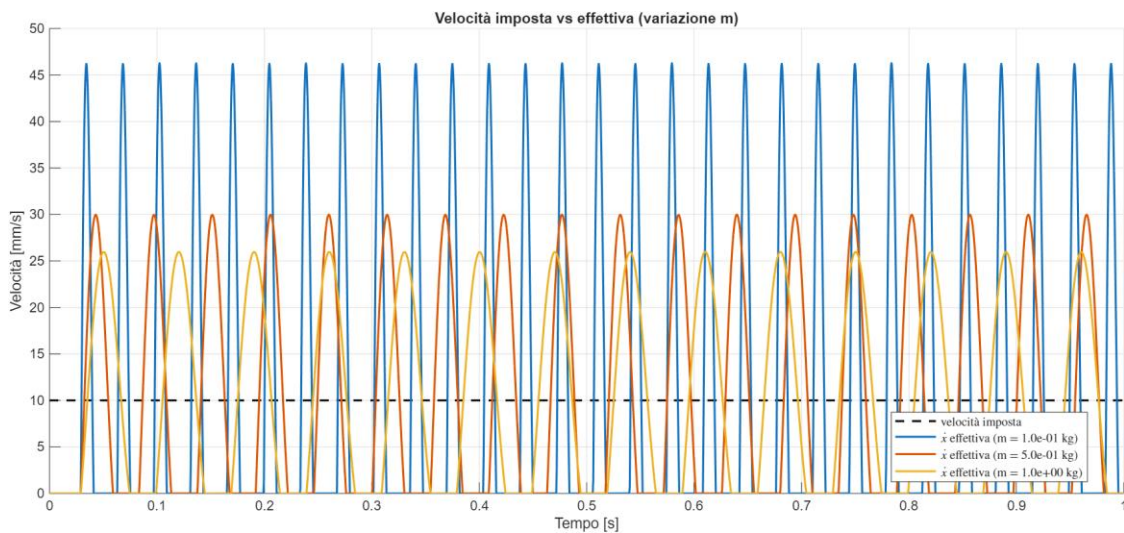


Figura 2.25: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare della massa del carrello

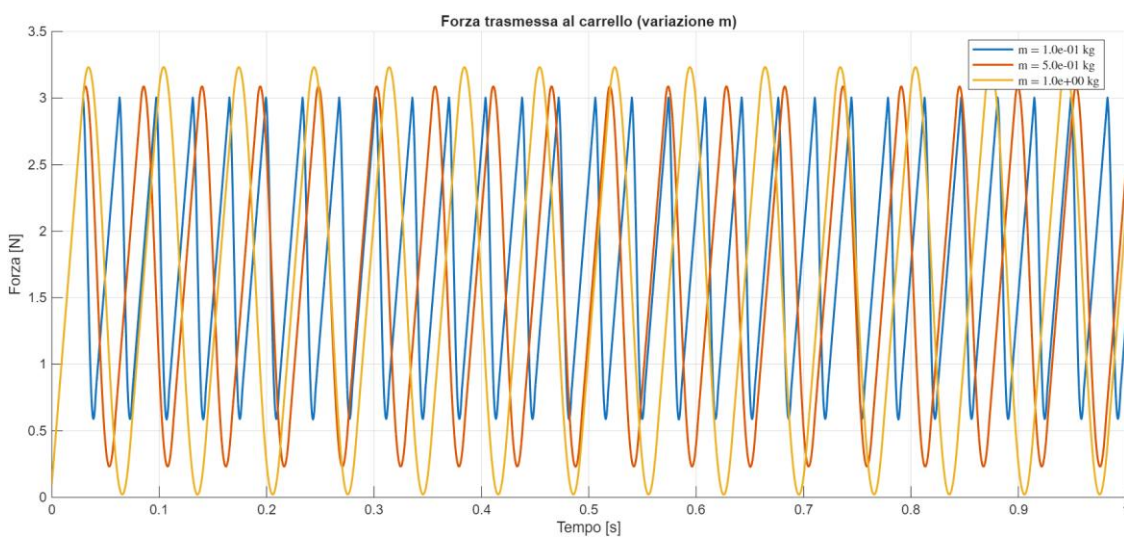


Figura 2.26: Andamento della forza trasmessa al carrello al variare della massa del carrello

I picchi di forza, invece, diventano più bassi rispetto a masse maggiori (Figura 2.26), perché il sistema più reattivo permette di vincere l'inerzia del carrello con minore energia.

I principali vantaggi di una massa ridotta sono quindi: una maggiore reattività del carrello e minor ritardo rispetto alla posizione desiderata, fasi di stick-slip più brevi e precisione del movimento migliorata, e minore forza richiesta per vincere l'inerzia, con conseguente riduzione delle sollecitazioni sui componenti meccanici.

Gli svantaggi sono rappresentati da un aumento delle vibrazioni, dovuto a picchi di velocità più elevati.

2.8.4 Variazione del coefficiente di attrito dinamico

Lo script è stato modificato in modo da variare esclusivamente il parametro μ_d , mantenendo costanti tutti gli altri parametri del modello. I valori di coefficiente di attrito dinamico considerati sono $\mu_d = [0.05; 0.15; 0.25]$. Di seguito sono riportati i risultati della simulazione al variare di μ_d .

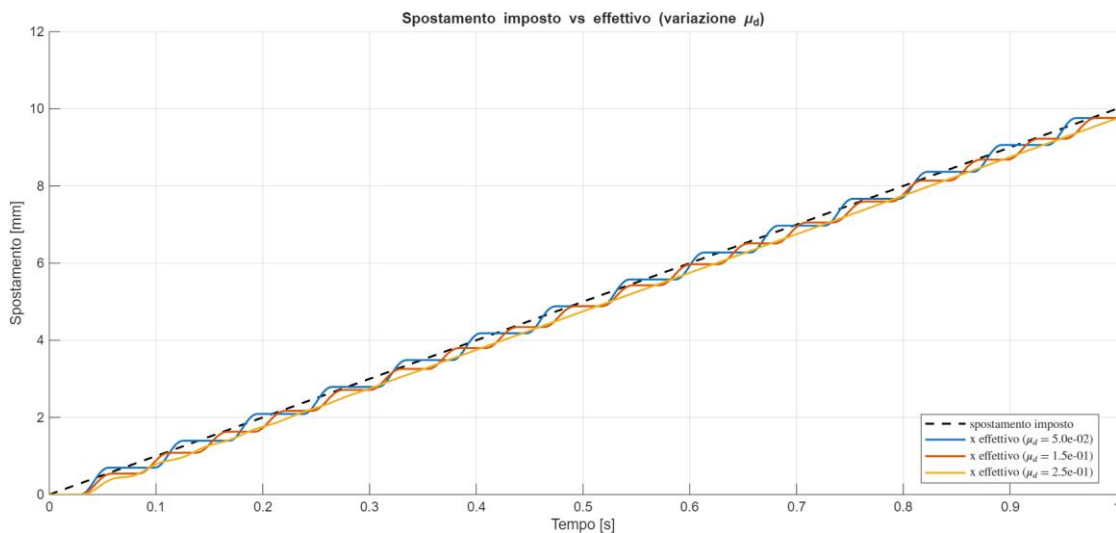


Figura 2.27: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito dinamico

Per bassi valori del coefficiente di attrito dinamico μ_d , le fasi di stick-slip risultano molto marcate. La posizione del carrello (Figura 2.27) presenta gradini ampi e poco frequenti, pur seguendo mediamente la traiettoria imposta dal motore. Al contrario, per valori elevati del coefficiente di attrito dinamico, il fenomeno dello stick-slip si attenua. Quindi i gradini diventano progressivamente più piccoli e, superato un certo valore di μ_d , la posizione ha solo poche oscillazioni smorzate. Successivamente, la posizione evolve come una retta a pendenza costante, traslata di un offset costante nel tempo, dovuto alla maggiore forza resistente dovuta all'elevato attrito dinamico. In questo caso, la resistenza durante la fase di slip aumenta, e l'energia rilasciata viene rapidamente dissipata, riducendo ulteriormente gli effetti dello stick-slip.

I picchi di velocità (Figura 2.28), per bassi valori del coefficiente di attrito dinamico, sono più alti a causa della minore resistenza allo slip che comporta accelerazioni più elevate. Al contrario, per valori elevati del coefficiente di attrito, la velocità presenta oscillazioni

smorzate e, dopo pochi istanti, la velocità converge rapidamente al valore imposto dal motore.

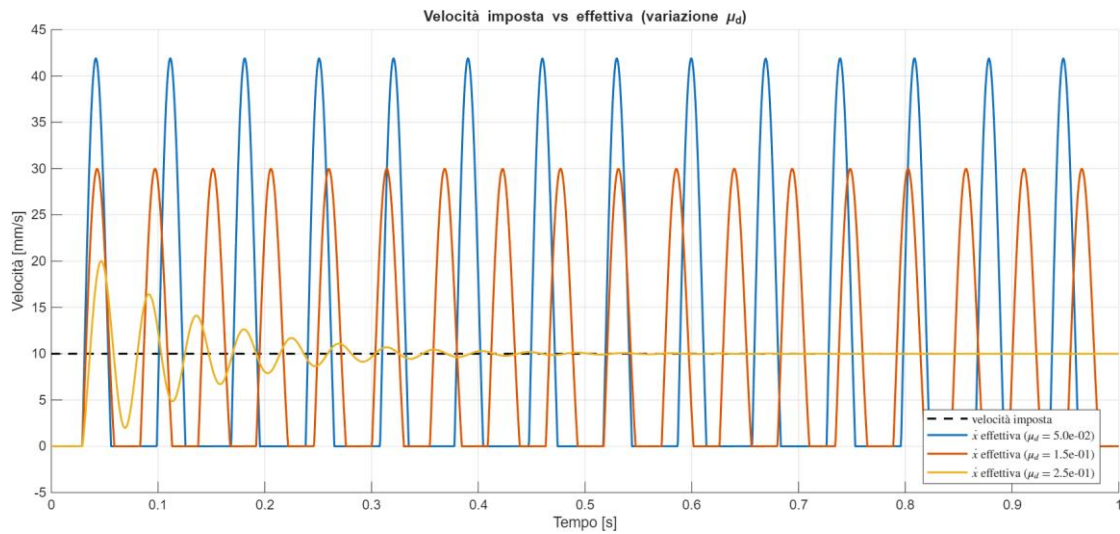


Figura 2.28: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare del coefficiente di attrito dinamico

Per quanto riguarda la forza trasmessa al carrello (Figura 2.29), con bassi valori di μ_d si osservano oscillazioni di ampiezza molto elevata che possono assumere anche valori negativi a causa del grande rilascio di energia elastica durante le fasi di slip. Questo indica che la deformazione della cinghia viene invertita, lavorando sia in trazione sia in compressione. Con alti valori di μ_d , la forza trasmessa presenta oscillazioni smorzate e successivamente converge a un valore costante pari alla forza di attrito di Coulomb.

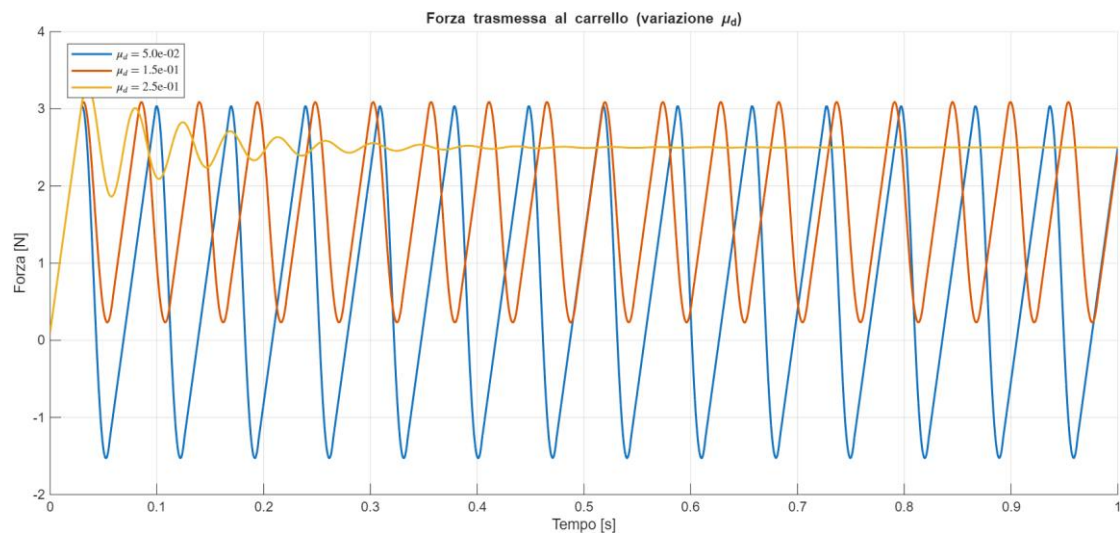


Figura 2.29: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito dinamico

Quindi, un coefficiente elevato stabilizza il moto e riduce lo stick-slip, ma introduce una forza resistente costante nel tempo che determina un offset di posizione. Viceversa, un coefficiente basso causa grandi rilasci di energia elastica, amplificando il fenomeno dello stick-slip e i picchi di velocità.

2.8.5 Variazione del coefficiente di attrito statico

Lo script è stato modificato in modo da variare esclusivamente il parametro μ_s , mantenendo costanti tutti gli altri parametri del modello. I valori di coefficiente di attrito statico considerati sono $\mu_s = [0.2; 0.3; 0.6]$.

In seguito, vengono riportati i risultati della simulazione al variare di μ_s .

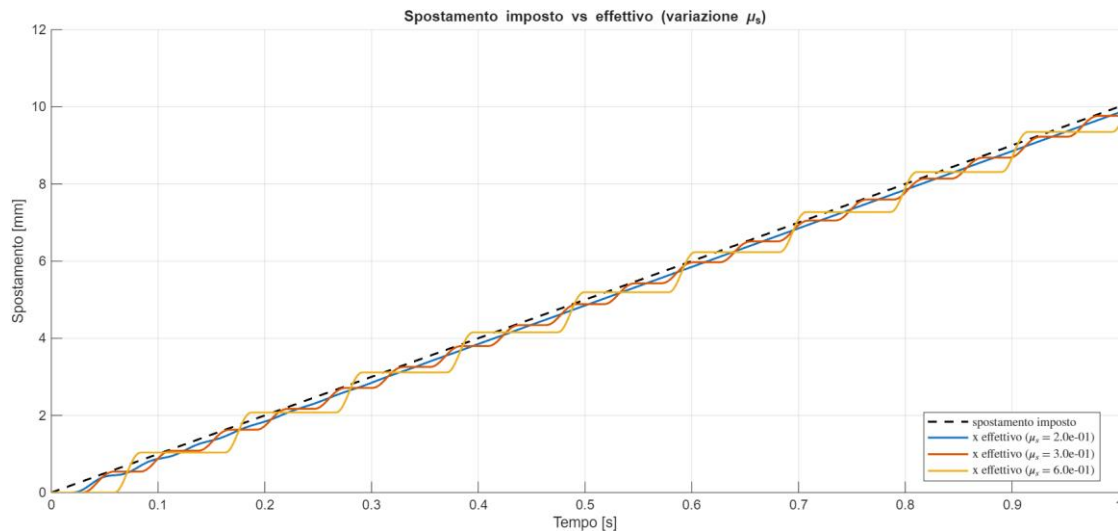


Figura 2.30: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito statico

Al diminuire del coefficiente di attrito statico, il carrello richiede una forza minore per innescare il moto. Di conseguenza, la durata delle fasi di stick si riduce e i gradini della posizione diventano più piccoli e più frequenti (Figura 2.30). Per valori sufficientemente bassi di attrito, il moto, dopo alcune piccole oscillazioni smorzate, diventa una retta a pendenza costante traslata di un offset costante nel tempo. In questo caso, la posizione del carrello risulta molto stabile, senza il fenomeno dello stick-slip, ma con un leggero ritardo rispetto alla posizione desiderata. Al contrario, aumentando il coefficiente di attrito statico, la forza necessaria per innescare il moto del carrello cresce. Di conseguenza, aumenta la durata delle fasi di stick e i gradini di posizione diventano più ampi, poiché durante la fase di slip viene rilasciata una maggiore quantità di energia elastica accumulata.

Per quanto riguarda la velocità, al diminuire del coefficiente di attrito statico μ_s , i picchi diventano progressivamente più bassi, mentre la frequenza delle oscillazioni aumenta (Figura 2.31). Al di sotto di un certo valore di μ_s , si osserva un numero limitato di oscillazioni iniziali, fortemente smorzate, e dopo pochi istanti la velocità converge rapidamente al valore imposto dal motore. Al contrario, per valori elevati di μ_s , i picchi di velocità diventano più alti. Questo avviene perché l'aumento della differenza tra la forza di distacco e la forza di Coulomb genera fasi di slip più intense, con un maggiore rilascio di energia elastica accumulata.

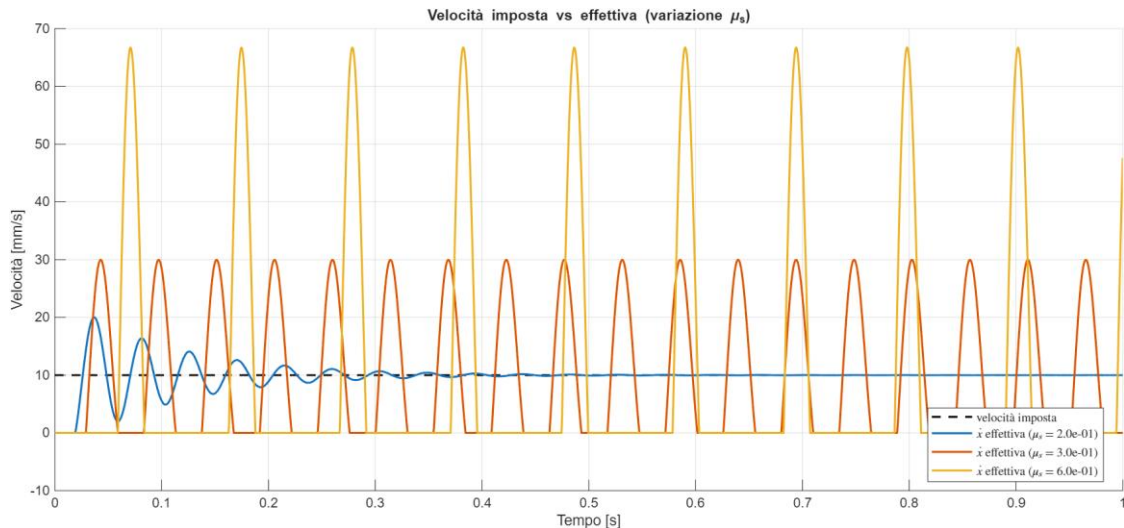


Figura 2.31: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare del coefficiente di attrito statico

Riducendo il coefficiente di attrito statico, la soglia di attrito di distacco si abbassa e, di conseguenza, la forza trasmessa al carrello necessaria per avviare la fase di slip diminuisce (Figura 2.32). Al di sotto di un certo valore di μ_s , si osservano solo poche oscillazioni smorzate e, successivamente, la forza trasmessa converge al valore dell'attrito di Coulomb. Per valori elevati di μ_s , invece, la forza elastica della cinghia deve crescere fino a valori più alti prima del distacco, poiché i picchi di attrito statico sono maggiori e più distanziati nel tempo. Inoltre, la forza trasmessa al carrello può assumere valori negativi, indicando che la deformazione della cinghia viene invertita e lavora sia in trazione sia in compressione, a causa del grande rilascio di energia elastica durante fasi di slip molto intense.

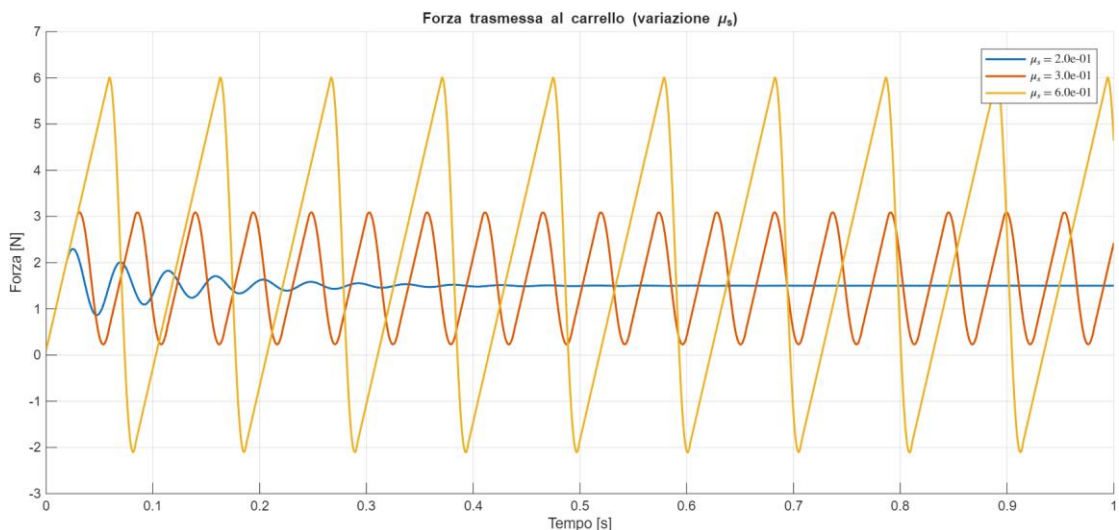


Figura 2.32: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito statico

Quindi, un coefficiente di attrito statico basso presenta vantaggi come la riduzione, fino alla soppressione, del fenomeno dello stick-slip, picchi di forza e velocità più contenuti e un moto più stabile. Tuttavia, comporta uno svantaggio: la posizione del carrello a regime presenta un offset, quindi vi è un ritardo rispetto alla posizione desiderata.

Viceversa, un coefficiente di attrito elevato genera una posizione che segue mediamente la traiettoria desiderata, ma con evidenti gradini dovuti allo stick-slip, e picchi di forza e velocità più alti. Perciò, nella progettazione del sistema è necessario trovare un compromesso tra stabilità, precisione e ampiezza dei picchi.

2.9 Variazione dei parametri del modello, caso moto sinusoidale

In modo analogo a quanto descritto nella sezione 2.8, si è effettuato lo studio dell'influenza dei parametri del sistema per il moto di input di posizione sinusoidale.

Nelle sottosezioni successive vengono presentati i risultati delle simulazioni. Si riportano esclusivamente i grafici, poiché le considerazioni sono analoghe a quelle del moto a velocità costante.

2.9.1 Variazione della rigidità della cinghia

Lo script è stato modificato in modo da variare esclusivamente il parametro di rigidità k , mantenendo costanti tutti gli altri parametri del modello. I valori di rigidità considerati sono $k = [0.5 \cdot 10^4 \text{ N/m}; 1 \cdot 10^4 \text{ N/m}; 1 \cdot 10^5 \text{ N/m}]$.

In seguito, vengono riportati i risultati della simulazione al variare della rigidità k .

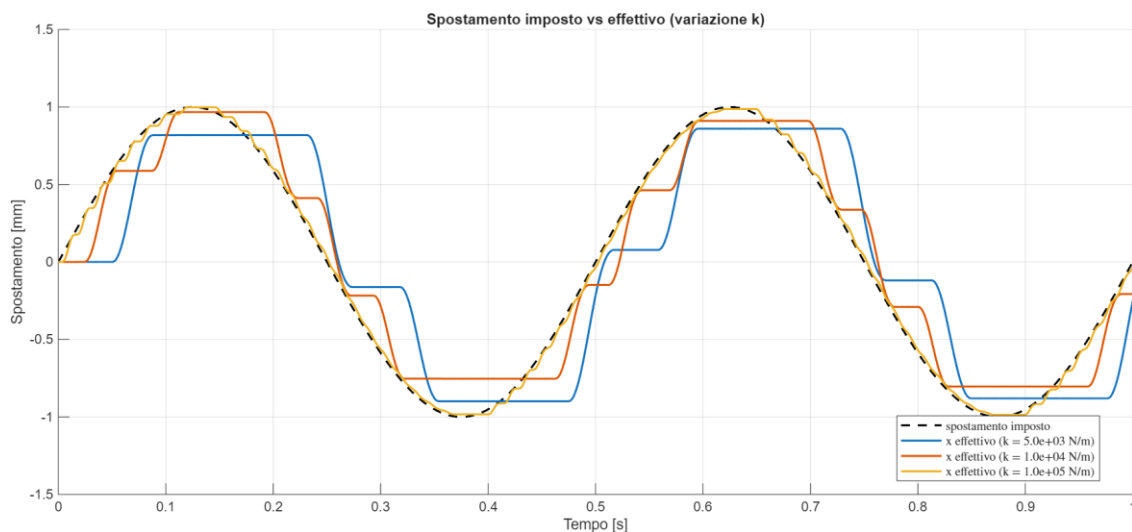


Figura 2.33: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della rigidità della cinghia

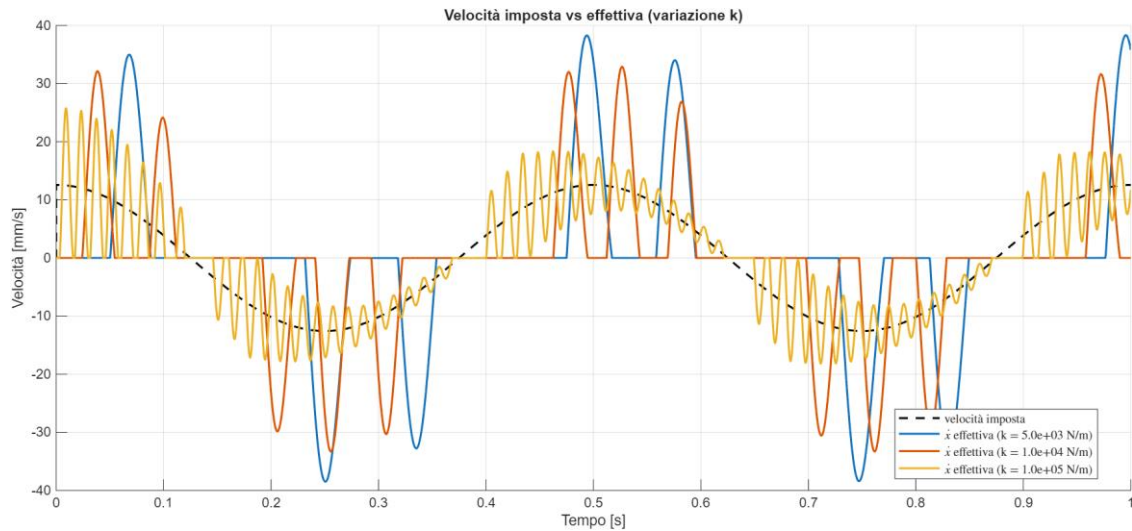


Figura 2.34: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare della rigidità della cinghia

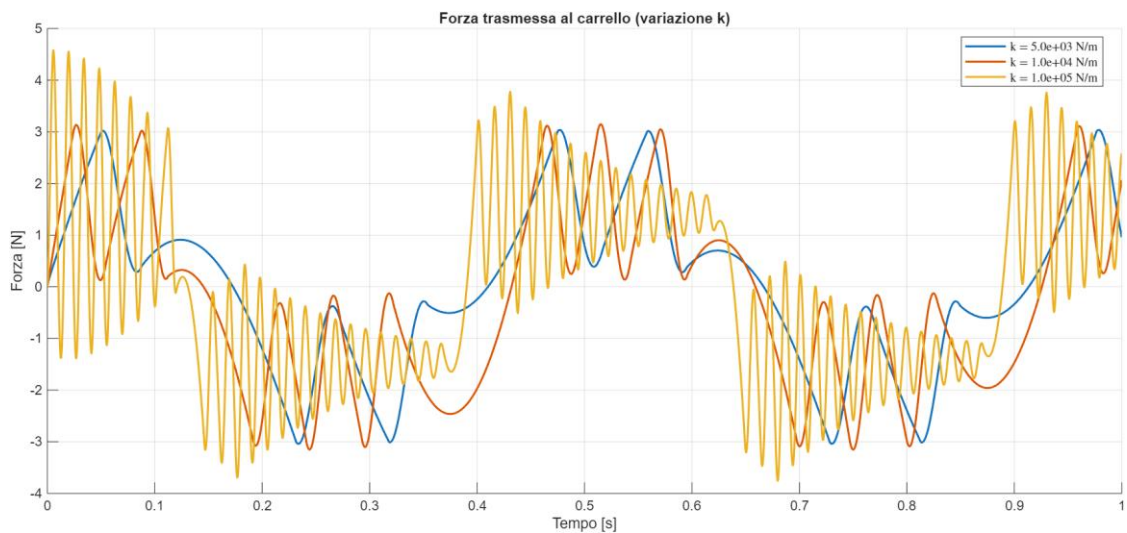


Figura 2.35: Andamento della forza trasmessa al carrello al variare del coefficiente della rigidità della cinghia

2.9.2 Variazione dello smorzamento della cinghia

Lo script è stato modificato in modo da variare esclusivamente il parametro di smorzamento c , mantenendo costanti tutti gli altri parametri del modello. I valori di smorzamento considerati sono $c = [1 \text{ Ns/m}; 10 \text{ Ns/m}; 100 \text{ Ns/m}]$.

In seguito, vengono riportati i risultati della simulazione al variare dello smorzamento c .

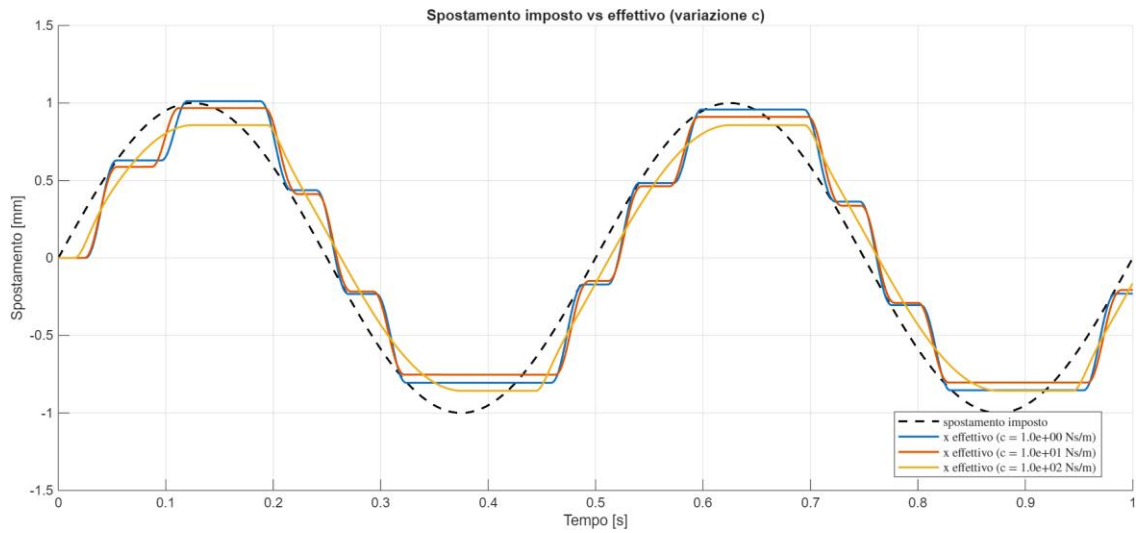


Figura 2.36: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare dello smorzamento della cinghia

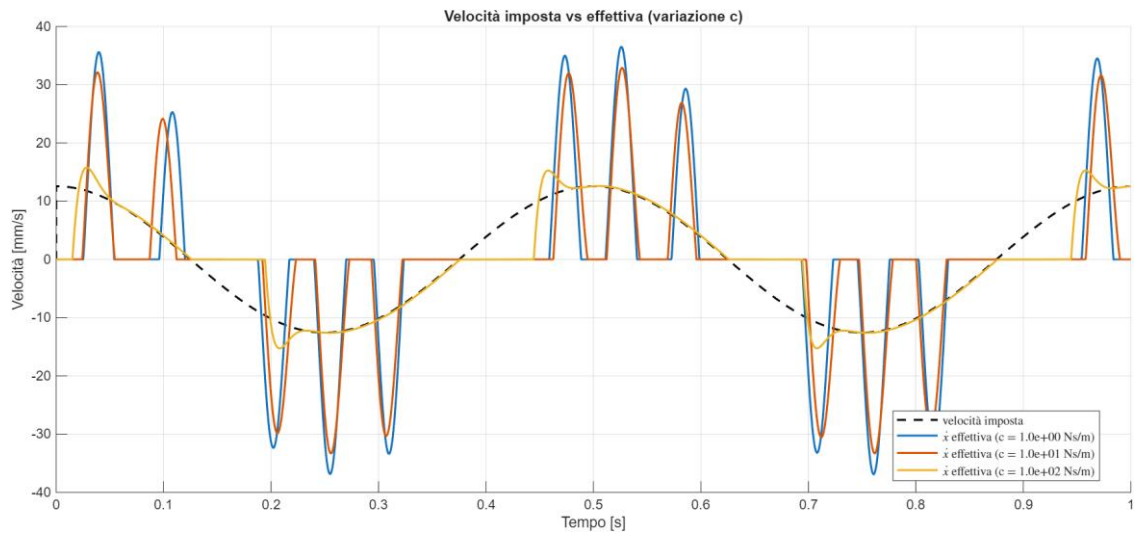


Figura 2.37: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare dello smorzamento della cinghia

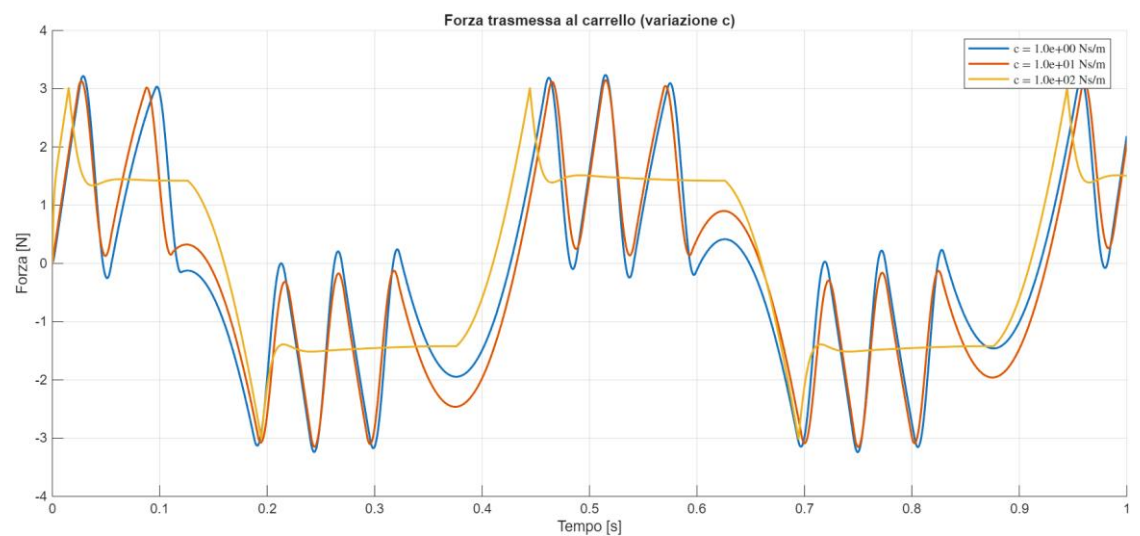


Figura 2.38: Andamento della forza trasmessa al carrello al variare dello smorzamento della cinghia

2.9.3 Variazione della massa del carrello

Lo script è stato modificato in modo da variare esclusivamente il parametro di massa m , mantenendo costanti tutti gli altri parametri del modello. I valori di massa considerati sono $m = [0.1 \text{ kg}; 0.5 \text{ kg}; 1 \text{ kg}]$.

In seguito, vengono riportati i risultati della simulazione al variare della massa m .

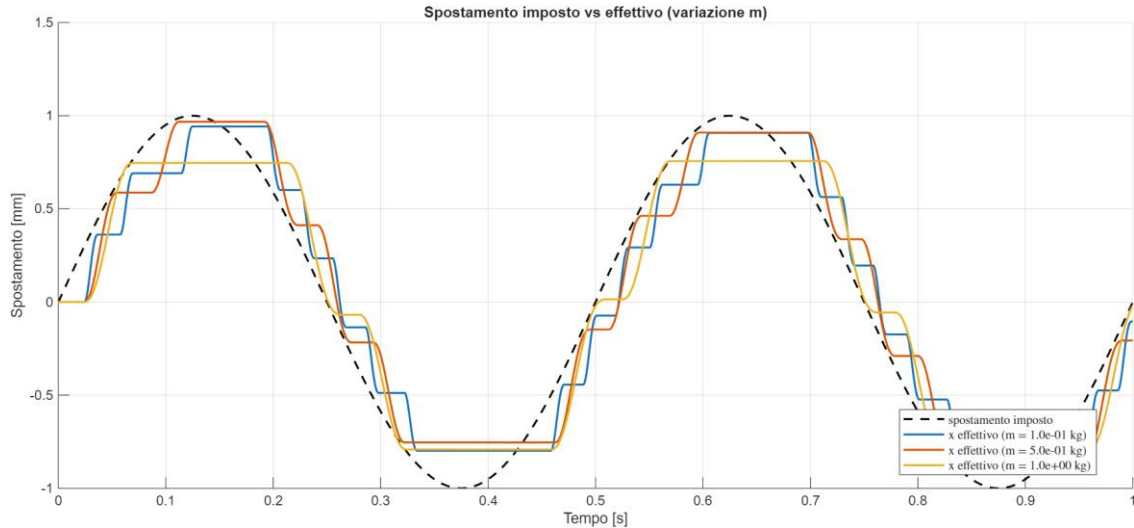


Figura 2.39: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare della massa del carrello

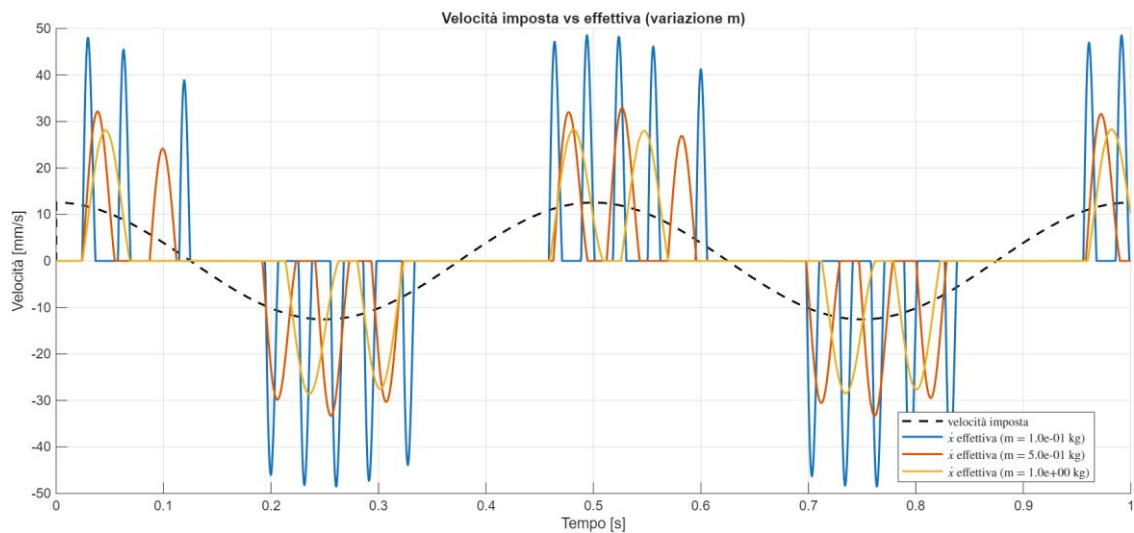


Figura 2.40: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare della massa del carrello

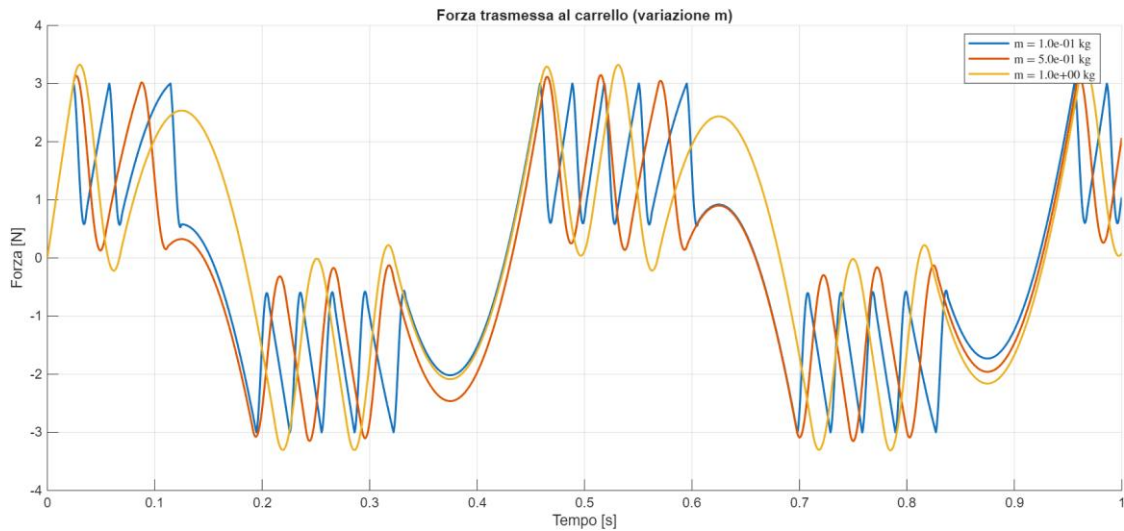


Figura 2.41: Andamento della forza trasmessa al carrello al variare della massa del carrello

2.9.4 Variazione del coefficiente di attrito dinamico

Lo script è stato modificato in modo da variare esclusivamente il parametro μ_d , mantenendo costanti tutti gli altri parametri del modello. I valori di coefficiente di attrito dinamico considerati sono $\mu_d = [0.05; 0.15; 0.25]$.

In seguito, vengono riportati i risultati della simulazione al variare di μ_d .

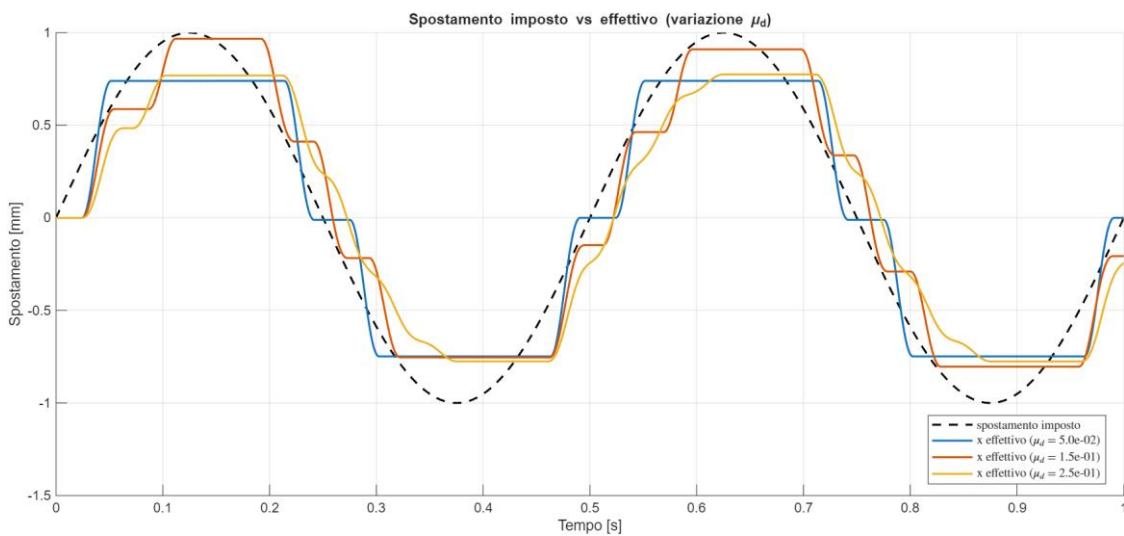


Figura 2.42: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito dinamico

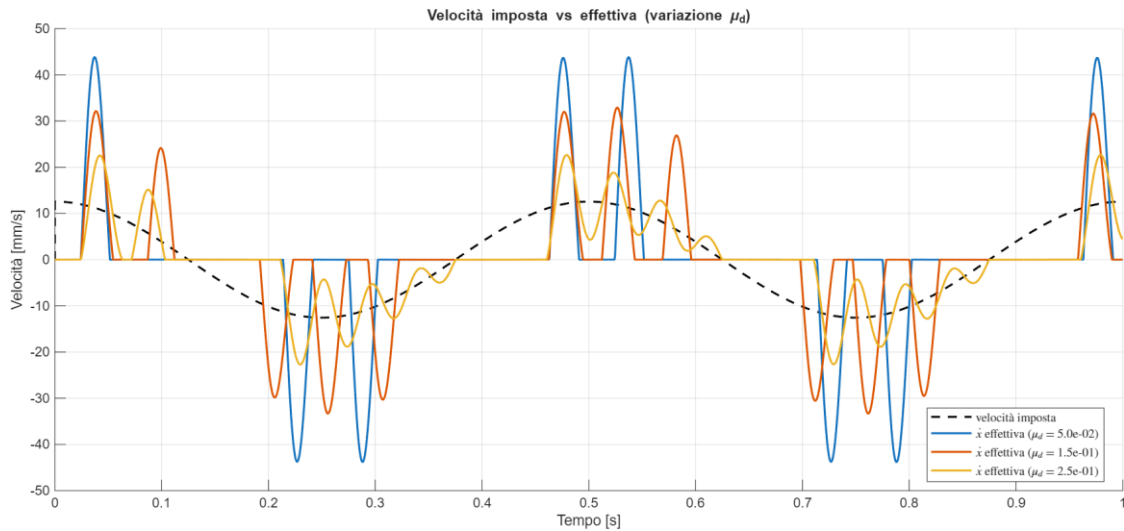


Figura 2.43: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare del coefficiente di attrito dinamico

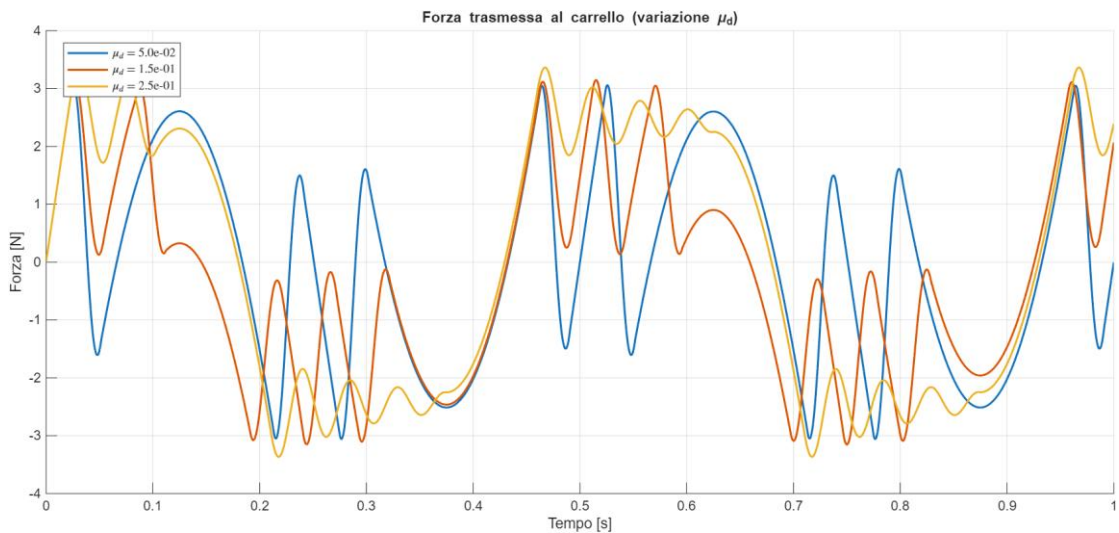


Figura 2.44: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito statico

2.9.5 Variazione del coefficiente di attrito statico

Lo script è stato modificato in modo da variare esclusivamente il parametro μ_s , mantenendo costanti tutti gli altri parametri del modello. I valori di coefficiente di attrito statico considerati sono $\mu_s = [0.2; 0.3; 0.6]$.

In seguito, vengono riportati i risultati della simulazione al variare di μ_s .

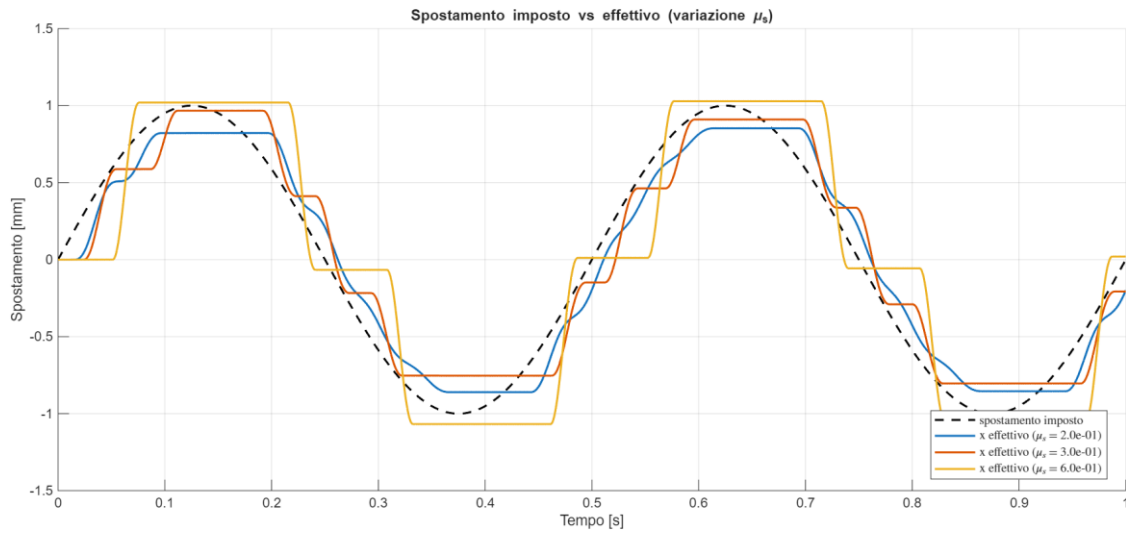


Figura 2.45: Confronto tra spostamento del carrello $x(t)$ e spostamento imposto dal motore $x_m(t)$ al variare del coefficiente di attrito statico

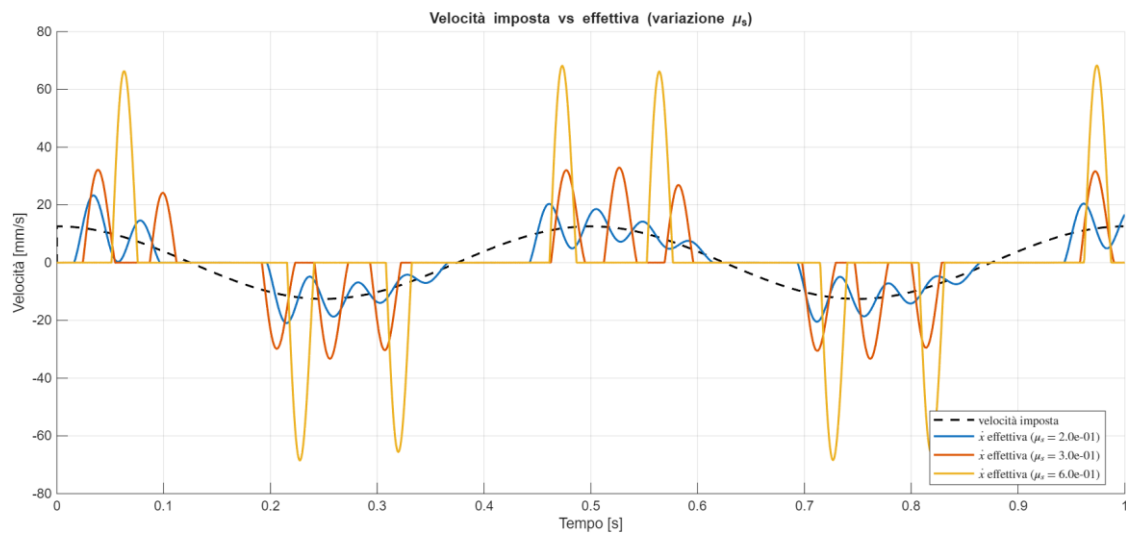


Figura 2.46: Confronto tra velocità del carrello $\dot{x}(t)$ e velocità imposta dal motore $\dot{x}_m(t)$ al variare del coefficiente di attrito statico

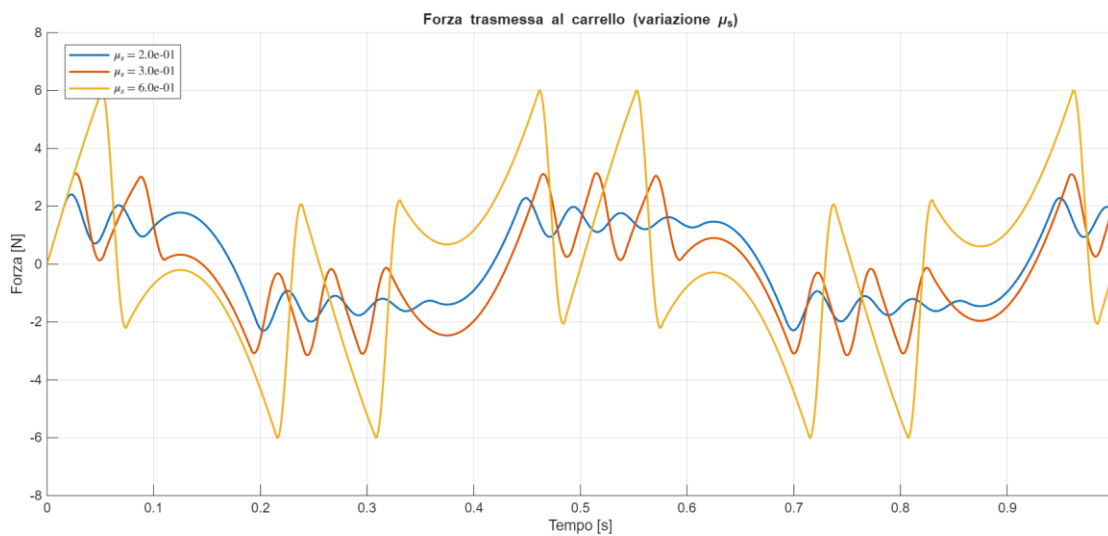


Figura 2.47: Andamento della forza trasmessa al carrello al variare del coefficiente di attrito dinamico

3 Generazione del G-code a partire da una traiettoria

Questo capitolo presenta innanzitutto la descrizione matematica delle curve di Bézier, utili per definire le traiettorie lungo le quali l'estrusore di una stampante FDM si sposta nello spazio di lavoro. Successivamente, si implementerà in MATLAB una curva di Bézier che descrive una traiettoria costituita da due segmenti rettilinei, calcolando, per ogni istante di tempo, le coordinate del punto che l'estrusore deve raggiungere.

Infine, si verifica che la velocità dell'ugello nel piano XY risulti effettivamente costante lungo tutta la traiettoria e si genera il file G-code necessario per i capitoli successivi, nei quali, a partire dal G-code, sarà controllato il movimento dei carrelli di una stampante FDM cartesiana.

3.1 Curve di Bézier

Le curve di Bézier sono state introdotte nel 1962 dall'ingegnere Pierre Bézier. Queste curve permettono di rappresentare forme in vettoriale, cioè immagini o percorsi che possono essere ingranditi senza perdita di definizione. Grazie alla possibilità di descrivere forme complesse utilizzando pochi punti di controllo, le curve di Bézier sono facilmente modificabili e per questo motivo vengono utilizzate in software come AutoCAD e Illustrator. [16]

Dal punto di vista matematico, una curva di Bézier è una funzione $\mathbf{p}(u)$ che restituisce un vettore posizione nello spazio, dove il parametro u varia tra 0 e 1.

Nelle stampanti FDM cartesiane, questo vettore viene espresso rispetto a un sistema di riferimento cartesiano, corrispondente al piano di stampa XY.

Le curve di Bézier considerate nella presente tesi descrivono traiettorie nel piano XY della stampante, quindi i punti di controllo hanno coordinate (x,y) . Di conseguenza, la curva $\mathbf{p}(u)$ restituisce un vettore $[x(u), y(u)]$, che rappresenta la posizione dell'estrusore lungo la traiettoria. Le curve di Bézier sono particolarmente utili nelle stampanti FDM perché consentono di descrivere sia segmenti rettilinei sia raccordi parabolici tra gli stessi, evitando spigoli e cambi di direzione improvvisi.

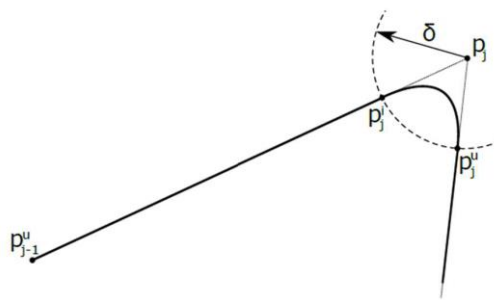


Figura 3.1: Traiettoria composta da segmenti rettilinei e raccordo parabolico [16]

Le traiettorie dell'estrusore di una stampante FDM cartesiana sono parametrizzate rispetto a un'unica ascissa curvilinea $s(t)$, che rappresenta la distanza percorsa lungo la traiettoria, definita tramite curve di Bézier di primo e secondo grado. Nella stampa 3D è importante minimizzare le accelerazioni, poiché queste possono causare instabilità dinamiche e imprecisioni nella deposizione del materiale, dovute a forze centrifughe che deviano la posizione effettiva del filamento rispetto a quella teorica. Per questo motivo, nel presente

lavoro di tesi ci si concentra su una legge oraria a velocità costante, in modo da garantire un moto uniforme dell'estrusore lungo la traiettoria.

3.1.1 Curve di Bézier lineari

Le curve di Bézier lineari sono le più semplici e sono definite da due punti di controllo P_0 e P_1 , identificati dai vettori di posizione \mathbf{p}_0 e \mathbf{p}_1

L'equazione vettoriale della curva è:

$$\mathbf{p}(u) = (1 - u)\mathbf{p}_0 + u\mathbf{p}_1 \quad (3.1)$$

dove: $0 \leq u \leq 1$.

Variando u da 0 a 1, la curva restituisce vettori posizione lungo il segmento definito dai vettori \mathbf{p}_0 e \mathbf{p}_1 .

Quando $u = 0$, il vettore coincide con \mathbf{p}_0 , corrispondente al primo punto di controllo P_0 , mentre quando $u = 1$, il vettore coincide con \mathbf{p}_1 , che individua l'ultimo punto di controllo P_1 .

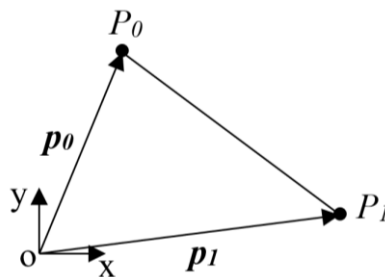


Figura 3.2: Curva di Bézier lineare e relativi punti di controllo P_0 e P_1 , identificati dai vettori di posizione \mathbf{p}_0 e \mathbf{p}_1

Derivando la curva lineare di Bézier rispetto al parametro u , si ottiene:

$$\frac{d\mathbf{p}}{du} = \mathbf{p}_1 - \mathbf{p}_0 \quad (3.2)$$

Si nota che la derivata è costante, quindi aumentando u di una quantità Δu costante, la variazione di posizione lungo la curva sarà anch'essa costante e parallela al segmento da P_0 a P_1 .

Infatti, partendo dalla posizione iniziale, il vettore posizione dopo un incremento Δu sarà:

$$\mathbf{p}(\Delta u) - \mathbf{p}(0) = \Delta u(\mathbf{p}_f - \mathbf{p}_i)$$

Allo stesso modo, dopo due incrementi successivi di Δu :

$$\mathbf{p}(2\Delta u) - \mathbf{p}(\Delta u) = \Delta u(\mathbf{p}_f - \mathbf{p}_i)$$

Questo mostra chiaramente che incrementi costanti di u corrispondono a spostamenti $\Delta u(\mathbf{p}_f - \mathbf{p}_i)$ costanti lungo la curva.

La velocità lungo la curva è data da:

$$\frac{d\mathbf{p}}{dt} = \frac{d\mathbf{p}}{du} \frac{du}{dt} \quad (3.3)$$

Il cui modulo è:

$$\left| \frac{d\mathbf{p}}{dt} \right| = v = \sqrt{v_x^2 + v_y^2} = \frac{du}{dt} \left| \frac{d\mathbf{p}}{du} \right| \quad (3.4)$$

Poiché $\frac{d\mathbf{p}}{du}$ è costante, per avere una velocità $\left| \frac{d\mathbf{p}}{dt} \right|$ costante, è necessario scegliere $\frac{du}{dt}$ costante, quindi far crescere u linearmente nel tempo.

Di conseguenza, la relazione che lega il parametro u allo spazio percorso lungo la curva è:

$$u = \frac{t}{t_f} = \frac{s(t)}{s_f} = \frac{vt}{s_f} \quad (3.5)$$

Dove v è la velocità costante desiderata, $s(t) = vt$ è l'ascissa curvilinea percorsa al tempo t , s_f è la lunghezza totale della curva e $t_f = \frac{s_f}{v}$ è il tempo necessario per percorrere l'intera traiettoria.

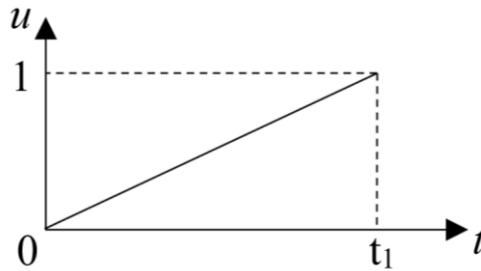


Figura 3.3: Andamento temporale del parametro u per una curva di Bézier lineare

Quindi, discretizzare la curva mediante incrementi costanti di Δu ad ogni passo temporale Δt permette di calcolare le posizioni dell'estrusore nei diversi istanti di tempo, consentendo di percorrere la curva di Bézier lineare con incrementi spaziali costanti, quindi a velocità costante. [16]

3.1.2 Curve di Bézier quadratiche

Le curve di Bézier quadratiche, o di secondo grado, sono definite da tre punti di controllo P_0 , P_1 e P_2 , identificati dai vettori posizione \mathbf{p}_0 , \mathbf{p}_1 e \mathbf{p}_2 .

L'equazione vettoriale della curva è:

$$\mathbf{p}(u) = (1 - u)^2 \mathbf{p}_0 + 2u(1 - u) \mathbf{p}_1 + u^2 \mathbf{p}_2 \quad (3.6)$$

dove $0 \leq u \leq 1$

Il termine $(1 - u)^2 \mathbf{p}_0$ fa sì che, quando $u = 0$, il vettore posizione coincida con \mathbf{p}_0 , corrispondente al primo punto di controllo P_0 .

Il termine $u^2 \mathbf{p}_2$ fa sì che, quando $u = 1$, il vettore coincida con \mathbf{p}_2 , corrispondente all'ultimo punto di controllo P_2 .

Il termine intermedio $2u(1-u)\mathbf{p}_1$ spinge la curva verso il punto di controllo P_1 , determinandone la curvatura.

Derivando rispetto a u si ottiene:

$$\frac{d\mathbf{p}}{du} = -2(1-u)\mathbf{p}_0 + 2[1-2u]\mathbf{p}_1 + 2u\mathbf{p}_2 \quad (3.7)$$

A differenza della curva di Bézier lineare, la derivata dipende u . Di conseguenza, incrementi costanti di Δu non producono spostamenti costanti lungo la curva.

Inoltre, la direzione della variazione di posizione cambia lungo il percorso, in funzione di u .

La derivata $\frac{d\mathbf{p}}{du}$ è costante solo agli estremi della curva:

$$\frac{d\mathbf{p}}{du}(u=0) = 2(\mathbf{p}_1 - \mathbf{p}_0) \quad (3.8)$$

$$\frac{d\mathbf{p}}{du}(u=1) = 2(\mathbf{p}_2 - \mathbf{p}_1) \quad (3.9)$$

Quindi la curva di Bézier quadratica è una curva di secondo grado, avente la tangente alla curva nei punti iniziale finale parallela ai lati della poligonale di controllo costituita dai punti P_0, P_1 e P_2 , ovvero parallele ai vettori P_0P_1 (nel caso $u=0$) e P_1P_2 (nel caso $u=1$).

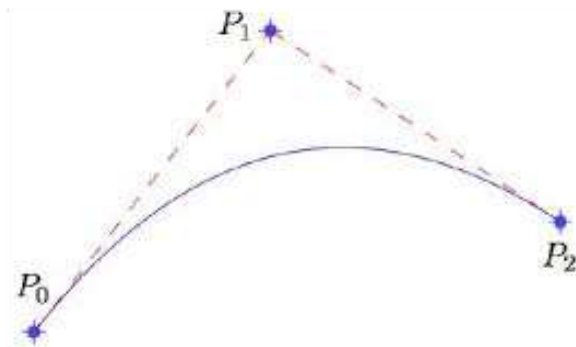


Figura 3.4: Curva di Bézier quadratica e relativi punti di controllo P_0, P_1, P_2

A differenza del caso della curva di Bézier lineare, la relazione $u = \frac{vt}{s_f}$, che lega il parametro u allo spazio percorso lungo la curva, non garantisce velocità costante poiché in questo caso se si incrementa u di Δu costante, lo spostamento lungo la curva non è costante lungo la traiettoria e di conseguenza il parametro u deve essere calcolato iterativamente.

Infatti:

Il modulo della velocità lungo la curva è:

$$\left| \frac{d\mathbf{p}}{dt} \right| = v = \sqrt{v_x^2 + v_y^2} = \frac{du}{dt} \left| \frac{d\mathbf{p}}{du} \right| \quad (3.10)$$

Se si imponesse che $u = \frac{vt}{s_f}$, si avrebbe:

$$\left| \frac{d\mathbf{p}}{dt} \right| = v = \sqrt{v_x^2 + v_y^2} = \frac{v}{s_f} \left| \frac{d\mathbf{p}}{du} \right|$$

Poiché $\frac{d\mathbf{p}}{du}$ varia lungo la curva, anche la velocità $v = \left| \frac{d\mathbf{p}}{dt} \right|$ varierebbe. Quindi scegliere $\frac{du}{dt}$ costante comporterebbe accelerazioni e decelerazioni dell'estrusore lungo la traiettoria, con conseguente deposizione non uniforme del materiale.

Perciò bisogna calcolare $u(t)$ in modo iterativo in modo che il modulo della velocità sia costante lungo la curva e pari al valore desiderato v .

Considerando un tempo di campionamento T_s , per calcolare il valore di u per ogni intervallo di campionamento, si sviluppa in serie la funzione $u(t)$, e si calcolano i valori di u all'istante $k+1$ in funzione del valore di u all'istante k . In questo modo si procede per iterazioni successive. [16]

Arrestando al primo ordine:

$$u_{k+1} = u_k + T_s \dot{u}_k + o(T_s) \quad (3.11)$$

Dove \dot{u} varia lungo la curva secondo la legge:

$$\dot{u} = \frac{v}{\left| \frac{d\mathbf{p}}{du} \right|} \quad (3.12)$$

Se serve maggior precisione si può arrestare al secondo ordine:

$$u_{k+1} = u_k + T_s \dot{u}_k + \frac{T_s^2}{2} \ddot{u}_k + o(T_s^2) \quad (3.13)$$

Dove \ddot{u} varia lungo la curva secondo la legge:

$$\ddot{u} = -v^2 \frac{\frac{d\mathbf{p}^T}{du} \cdot \frac{d^2\mathbf{p}}{du^2}}{\left| \frac{d\mathbf{p}}{du} \right|^4} \quad (3.14)$$

3.2 Calcolo delle coordinate dei punti lungo una traiettoria

L'obiettivo del presente paragrafo consiste nel calcolare, mediante MATLAB, le coordinate dei punti di una traiettoria percorsa a velocità costante dall'ugello di una stampante FDM.

La traiettoria è definita nel piano X-Y ed è costituita da due segmenti rettilinei, P_0P_1 e P_1P_2 con un raccordo di raggio $\delta = 2 \text{ mm}$, in modo da evitare una discontinuità nella direzione del moto. [16]

Le coordinate dei segmenti sono:

$$\mathbf{P}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} 15 \\ 10 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 15 \\ 5 \end{bmatrix}$$

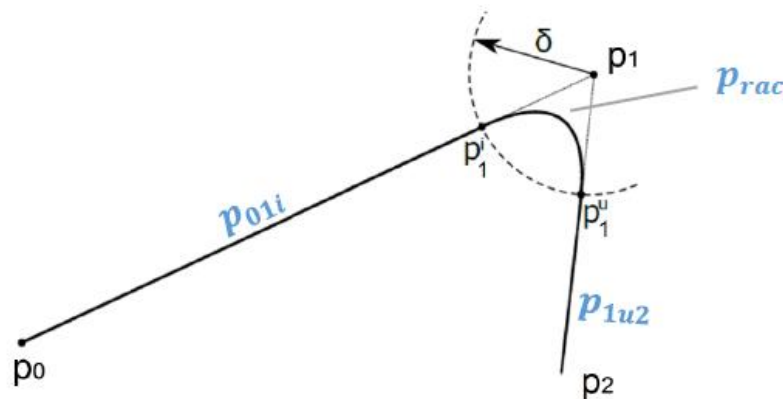


Figura 3.5: Traiettorie nel piano X-Y costituita da due segmenti rettilinei, P_0P_1 e P_1P_2 con un raccordo di raggio δ

I tratti rettilinei della traiettoria sono stati rappresentati da curve di Bézier di primo grado, mentre il raccordo da una curva di Bézier di secondo grado.

Si desidera che l'ugello si muova a velocità costante $v = 40 \text{ mm/s}$ e si sceglie un tempo di campionamento pari a $\Delta t = 0,01 \text{ s}$. Le curve di Bézier sono state calcolate adottando un arresto al primo ordine.

Lo script inizia con l'inizializzazione dei punti di controllo e dei parametri della traiettoria. Successivamente si determinano le coordinate dei punti p_1^i e p_1^u necessari per definire il poligono di controllo dei tre tratti. A questo punto è possibile plottare i punti di inizio e fine dei tre tratti, così da verificare graficamente la correttezza della geometria della traiettoria prima di generare i punti intermedi lungo le curve.

```

%% Coordinate punti poligono di controllo
P0= [0 0]';
P1= [15 10]';
P2= [15 5]';
delta = 2;          % raggio raccordo [mm]
dt = 0.01;
v = 40;            % [mm/s]

%% Calcolo intersezioni con circonferenza di raggio delta
%%con relazioni trigonometriche
alfa = atan((P1(2)-P0(2))/(P1(1)-P0(1)));
p1i = P1 - delta*[cos(alfa) sin(alfa)]';
p1u = P1 - [0 delta]';

%%Plottaggio punti di inizio e fine dei 3 tratti
figure(1)
plot(P0(1),P0(2),'ro',P1(1),P1(2),'ko',P2(1),P2(2),'ro',...
      p1i(1),p1i(2),'r*',p1u(1),p1u(2),'r*', 'linewidth',2)
axis equal
xlabel('asse x (mm)')
ylabel('asse y (mm)')
grid on
hold on

```

Ora si passa al primo tratto, ovvero il tratto Bezièr lineare che va da P_0 a p_1^i . Di seguito si riporta lo script.

```

%% Tratto di ingresso rettilineo
k = 1;
u(k) = 0;

%ciclo su k per u<1
while u(k)<1
    l_1 = norm(p1i-P0); % lunghezza primo tratto rettilineo
    du=v/l_1*dt; % incremento du
    u(k+1) = u(k)+du;
    k=k+1;
end

u = u(u <=1) ; % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_01i = P0*(1-u) + p1i*u; %Bezier primo tratto lineare

```

Essendo di primo grado, la derivata della posizione rispetto al parametro u è costante, come mostrato nell'Eq.(3.2), e il suo modulo coincide con la lunghezza del tratto rettilineo l_1 :

$$\frac{d\mathbf{p}}{du} = \mathbf{p}_1^i - \mathbf{P}_0$$

$$l_1 = \left| \frac{d\mathbf{p}}{du} \right| = |\mathbf{p}_1^i - \mathbf{P}_0|$$

Definito il passo temporale dt e la velocità desiderata lungo la traiettoria, l'incremento del parametro u si ottiene sostituendo $l_1 = \left| \frac{d\mathbf{p}}{du} \right|$ nella relazione (3.4):

$$du = \frac{v}{l_1} dt$$

Nello script MATLAB, il parametro u viene inizializzato a zero e poi aggiornato all'interno di un ciclo while fino a raggiungere il valore unitario, corrispondente al punto finale del tratto rettilineo. Una volta determinato il vettore dei valori del parametro u , si devono rimuovere eventuali valori maggiori di 1. A questo punto si possono calcolare le coordinate dei punti della traiettoria mediante la curva lineare di Bezier definita dall'Eq (3.1):

$$\mathbf{p}(u) = (1 - u)\mathbf{p}_0 + u\mathbf{p}_1^i$$

Prima di passare al tratto parabolico, è necessario fare alcune osservazioni. Nel passaggio dal primo tratto rettilineo al raccordo e dal raccordo al secondo tratto rettilineo, non è detto

che il primo punto del raccordo e il primo punto del secondo tratto rettilineo coincidano esattamente con i punti di intersezione p_1^i e p_1^u .
 L'estrusore, passando dal tratto rettilineo al raccordo, all'interno dell'ultimo intervallo di campionamento dt si troverà per una frazione di tempo $\frac{|p_1^i - A|}{v}$ sul tratto rettilineo e per il restante tempo $-\frac{|p_1^i - A|}{v}$ sul tratto di raccordo parabolico.

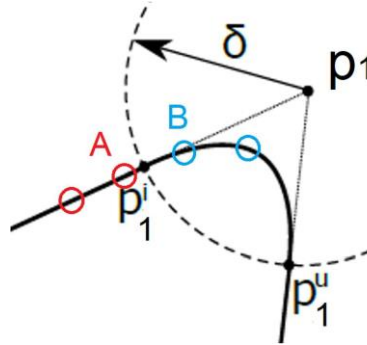


Figura 3.6: Passaggio dal primo tratto rettilineo al raccordo

Di conseguenza, il primo punto B del raccordo non corrisponderà a $u = 0$, ma sarà dato da:

$$u = \left(dt - \frac{|p_1^i - A|}{v} \right) \dot{u}_k \quad (3.15)$$

dove \dot{u}_k , come mostrato nell' Eq.(3.12), si ottiene come:

$$\dot{u}_k = \frac{v}{\left| \frac{dp}{du} \right|_{p_1^i}}$$

Allo stesso modo, passando dal raccordo al tratto rettilineo, l'estrusore si troverà, nell'ultimo intervallo dt , per una frazione di tempo $\frac{|p_1^u - C|}{v}$ sul tratto parabolico e per il restante tempo $-\frac{|p_1^u - C|}{v}$ sul tratto rettilineo.

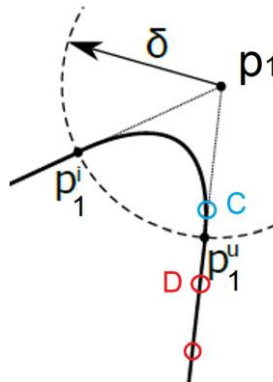


Figura 3.7: Passaggio dal raccordo al secondo tratto rettilineo

Di conseguenza, il primo punto D sul tratto rettilineo non corrisponderà a $u = 0$, ma sarà dato da:

$$u = \left(dt - \frac{|p_1^u - C|}{v} \right) \dot{u}_k \quad (3.16)$$

dove:

$$\dot{u}_k = \frac{v}{\left| \frac{d\mathbf{p}}{du} \right|_{p_1^u}}$$

Fatte queste precisazioni, è possibile passare allo script relativo al tratto di raccordo parabolico.

```
% Raccordo parabolico
% Implementare un ciclo iterativo per il calcolo di u(k+1) a partire da u(k)
u=[];
dpsudu = []; % dp/du

k = 1; % reinizializzo contatore

dpsudu0 = 2*(P1-p1i) ; % reinizializzo dp/du
u(k)=(dt-norm(p1i-p_01i(:,end))/v)*(v/norm(dpsudu0)); % reinizializzo parametro u

%ciclo su k per u<1
while u(k)<1
    dpsudu = -2*(1-u(k))*p1i+2*(1-2*u(k))*P1+2*u(k)*p1u;
    du = v/norm(dpsudu)*dt;
    u(k+1) = u(k) + du;
    k=k+1;
end

u = u(u <=1) ; % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_rac = (1-u).^2.*p1i + 2*(1-u).*u.*P1+u.^2.*p1u; %Bezier di raccordo
```

Si reinizializza il contatore e, come mostrato nell'Eq.(3.15), il primo valore di u non è zero, ma si calcola come:

$$u(1) = \left(dt - \frac{|p_1^i - A|}{v} \right) \frac{v}{\left| \frac{d\mathbf{p}}{du} \right|_{p_1^i}}$$

dove:

$$\left| \frac{d\mathbf{p}}{du} \right|_{p_1^i} = |2(P_1 - p_1^i)|$$

A questo punto si apre un ciclo while in cui il parametro u viene aggiornato fino a raggiungere il valore unitario, corrispondente al punto finale del tratto curvilineo. È importante ricordare che, come si osserva dall'Eq.(3.7), per una Bèzier quadratica la derivata della posizione rispetto al parametro u non è costante:

$$\frac{d\mathbf{p}}{du} = -2(1-u)\mathbf{p}_1^i + 2[1-2u]\mathbf{P}_1 + 2u\mathbf{p}_1^u$$

Definito il passo temporale dt e la velocità desiderata lungo la traiettoria, l'incremento del parametro u si ricava dalla relazione (3.10):

$$\left| \frac{d\mathbf{p}}{dt} \right| = v = \frac{du}{dt} \left| \frac{d\mathbf{p}}{du} \right|$$

Quindi:

$$du = \frac{v}{\left| \frac{d\mathbf{p}}{du} \right|} dt$$

Una volta determinato il vettore dei valori del parametro u , si devono rimuovere eventuali valori maggiori di 1. A questo punto si possono calcolare le coordinate dei punti della traiettoria mediante l'equazione della curva quadratica di Bézier definita dall'Eq.(3.6):

$$\mathbf{p}(u) = (1-u)^2\mathbf{p}_1^i + 2u(1-u)\mathbf{P}_1 + u^2\mathbf{p}_1^u$$

Infine, si riporta lo script relativo al secondo tratto rettilineo.

```

%% Tratto di uscita rettilineo
%Implementare un ciclo iterativo per il calcolo di u(k+1) a partire da u(k)

u = [];          % reinizializzo parametro u
dpsudu = [];    % reinizializzo dp/du

k = 1; % reinizializzo contatore

dpsudu0 = 2*(p1u-P1) ; % reinizializzo dp/du
u(k)=(dt-norm(p1u-p_rac(:,end))/v)*(v/norm(dpsudu0)); % reinizializzo parametro u

while u(k)<1
    l_2 = norm(P2-p1u); % lunghezza secondo tratto rettilineo
    du=v/l_2*dt;      % incremento du
    u(k+1) = u(k)+du;
    k=k+1;
end

u = u(u <=1) ; % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_1u2 = p1u*(1-u) + P2*u; %Bezier secondo tratto lineare

```

In modo analogo al raccordo, si reinizializza il contatore e, come mostrato nell'Eq.(3.16), il primo valore di u non è zero, ma si calcola come:

$$u(1) = \left(dt - \frac{|\mathbf{p}_1^u - \mathbf{C}|}{v} \right) \frac{v}{\left| \frac{d\mathbf{p}}{du} \right|_{\mathbf{p}_1^u}}$$

dove:

$$\left| \frac{d\mathbf{p}}{du} \right|_{\mathbf{p}_1^u} = |2(\mathbf{P}_2 - \mathbf{p}_1^u)|$$

Essendo di primo grado, la derivata della posizione rispetto al parametro u è costante, come mostrato nell'Eq.(3.2), e il suo modulo coincide con la lunghezza del tratto rettilineo l_2 .

$$\frac{d\mathbf{p}}{du} = \mathbf{P}_2 - \mathbf{p}_1^u$$

$$l_2 = \left| \frac{d\mathbf{p}}{du} \right| = \mathbf{P}_2 - \mathbf{p}_1^u$$

Definito il passo temporale dt e la velocità desiderata lungo la traiettoria, l'incremento del parametro u si ottiene sostituendo $l_2 = \left| \frac{d\mathbf{p}}{du} \right|$ nella relazione (3.4):

$$du = \frac{v}{l_2} dt$$

Nello script MATLAB, il parametro u viene inizializzato a zero e poi aggiornato all'interno di un ciclo while fino a raggiungere il valore unitario, corrispondente al punto finale del tratto rettilineo. Una volta determinato il vettore dei valori del parametro u , si devono rimuovere eventuali valori maggiori di 1. Fatto ciò, le coordinate dei punti della traiettoria si calcolano mediante l'equazione della curva lineare di Bézier, definita dall'Eq (3.1):

$$\mathbf{p}(u) = (1 - u) \mathbf{p}_1^u + u\mathbf{P}_2$$

A questo punto, è possibile unire le tre curve di Bézier ottenendo una traiettoria continua percorsa a velocità costante.

```
% Assemblo i 3 tratti

p = [p_01i p_rac p_1u2];
px = p(1,:); %prima riga, componenti x
py = p(2,:); %seconda riga, componenti y

figure(1)
plot(px,py,'bo','linewidth',1.5)
xlabel('x [mm]')
ylabel('y [mm]')
title('Spostamento nel piano XY')
grid on
```

Estraendo le coordinate dei punti lungo l'asse X e lungo l'asse Y, è possibile riportare lo spostamento nel piano X-Y.

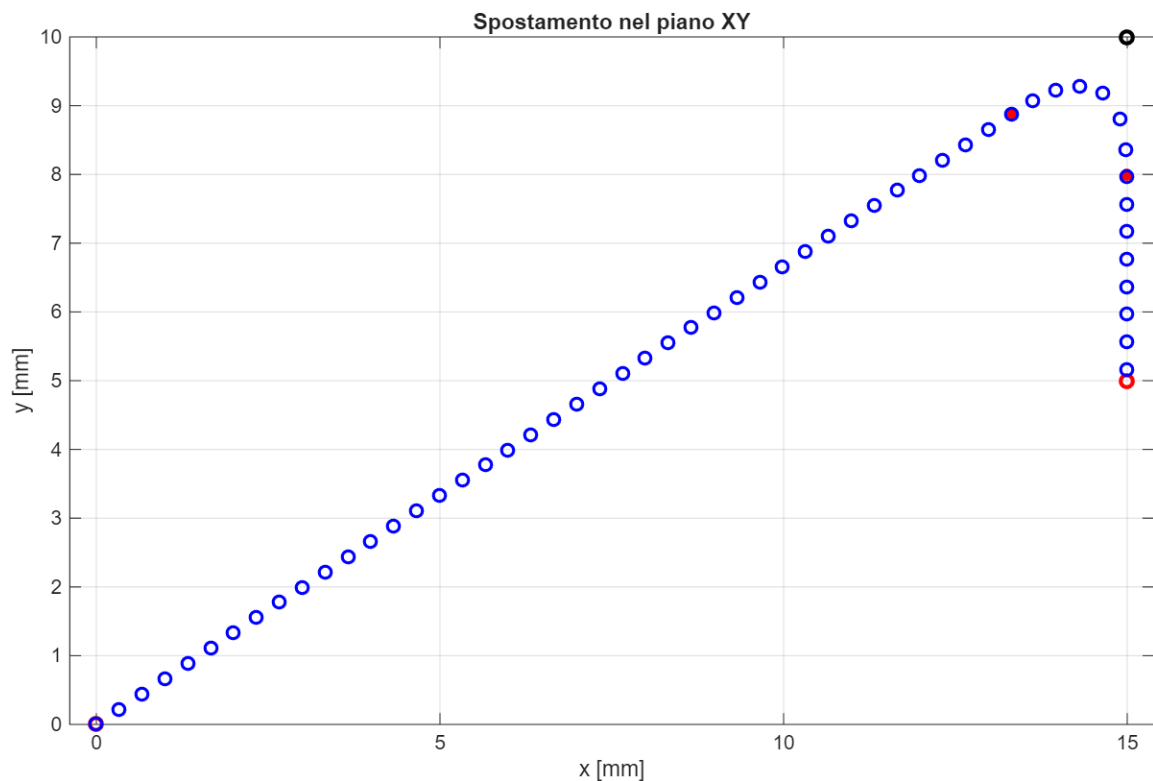


Figura 3.8: Spostamento nel piano X-Y

Per le successive analisi, è utile rappresentare graficamente lo spostamento lungo X e lo spostamento lungo Y in funzione del tempo, e successivamente la velocità lungo X e la velocità lungo Y in funzione del tempo.

Lo script è il seguente, e i grafici ottenuti sono riportati di seguito.

```

%% Vettore tempo
N = size(p,2);
t = zeros(1,N);

for k = 2:N
    %ds = norm(p(:,k) - p(:,k-1));
    ds = sqrt((p(1,k) - p(1,k-1))^2 + (p(2,k) - p(2,k-1))^2); % distanza tra punti
    consecutivi
    t(k) = t(k-1) + ds/v;
end
%% Plot x(t) e y(t)
figure (2)
subplot(2,1,1)
plot(t, px, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('x [mm]')
title('Spostamento x(t)')
grid on

subplot(2,1,2)

```

```

plot(t, py, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('y [mm]')
title('Spostamento y(t)')
grid on

%% Velocità di stampa
dx = diff(px);      % differenze di posizione
dt = diff(t);      % differenze di tempo
vx = dx./dt;       % velocità media tra punti
t_v = t(1:end-1);  % istanti associati alla velocità

figure (3)
subplot(2,1,1)
plot(t_v, vx, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('v_x [mm/s]')
title('Velocità v_x(t)')
grid on

dy = diff(py);      % differenze di posizione
dt = diff(t);      % differenze di tempo
vy = dy./dt;       % velocità media tra punti
t_v = t(1:end-1);  % istanti associati alla velocità

figure (3)
subplot(2,1,2)
plot(t_v, vy, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('v_y [mm/s]')
title('Velocità v_y(t)')
grid on

```

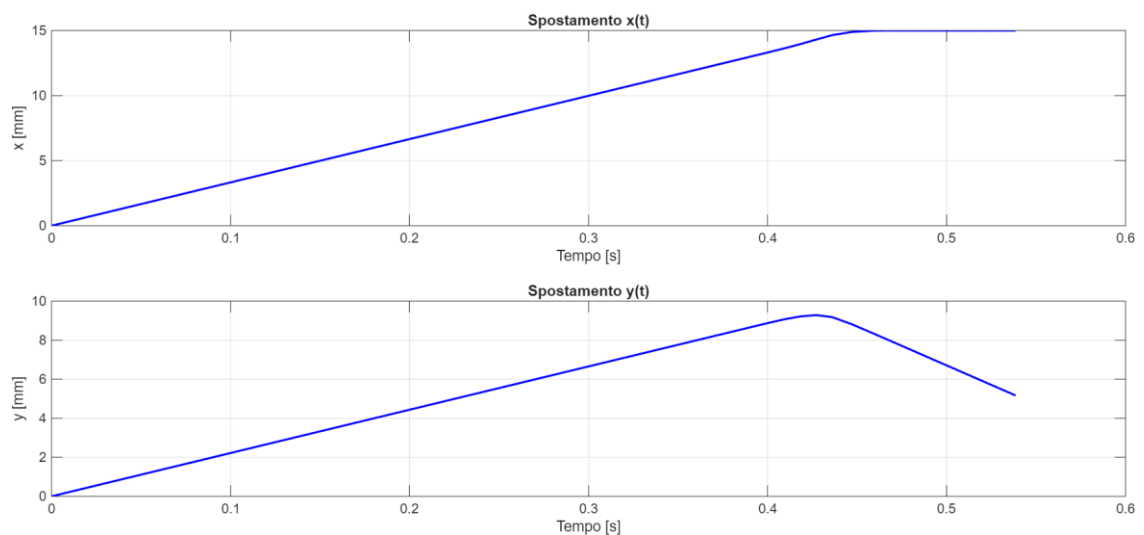


Figura 3.9: Spostamento lungo l'asse X e spostamento lungo l'asse Y in funzione del tempo

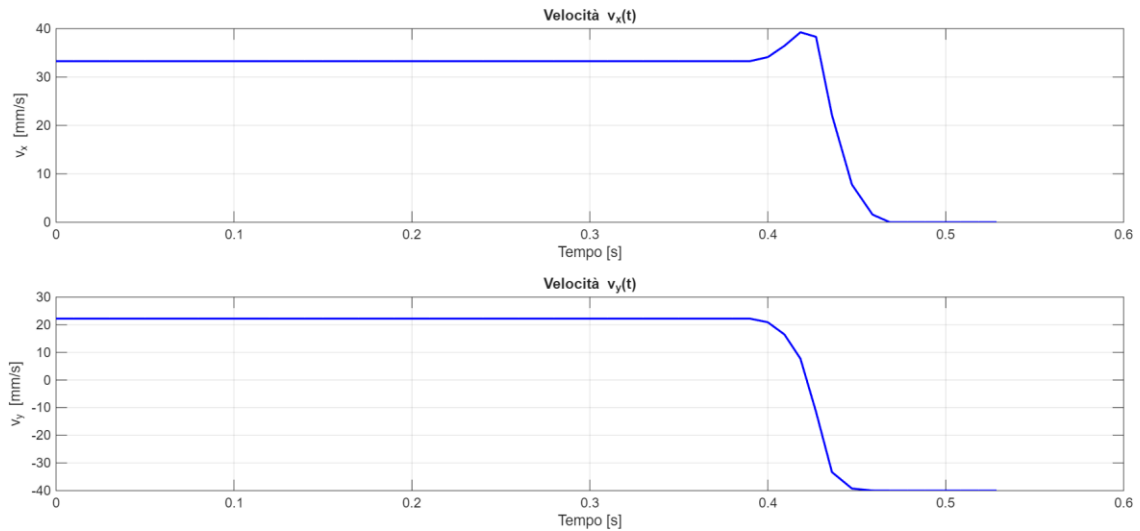


Figura 3.10: Velocità lungo l'asse X e la velocità lungo l'asse Y in funzione del tempo

A questo punto si può calcolare la velocità risultante dell'ugello nel piano XY e verificare graficamente che essa sia costante e pari a 40 mm/s lungo la traiettoria.

```
% Calcolo del modulo della velocità
v = sqrt(vx.^2 + vy.^2);

figure (4)
plot(t_v, v, 'b-', 'LineWidth', 2)
xlabel('Tempo [s]')
ylabel('Velocità nel piano [mm/s]')
title('Modulo della velocità nel piano XY')
grid on
```

Dalla Figura 3.11 si osserva che la velocità lungo la traiettoria rimane costante nel tempo, confermando che il modello è stato costruito correttamente.

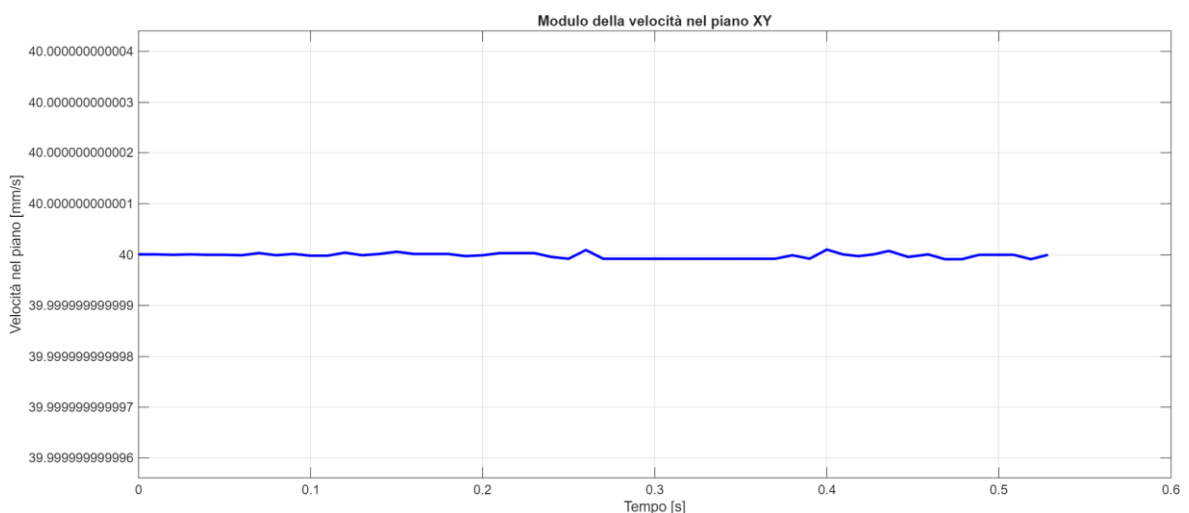


Figura 3.11: Velocità risultante nel piano XY

Con la traiettoria modellata e le coordinate dei punti calcolate, si dispongono ora di tutti i dati necessari per generare il file G-code, contenente i comandi necessari a realizzare il

movimento dell'ugello secondo il percorso definito nel piano XY. Di seguito si riporta lo script.

```

%% Generazione Gcode
gcodeFile = fopen('gcode', 'w');
fprintf(gcodeFile, 'G21 \n');      % usa mm
fprintf(gcodeFile, 'G90 \n');      % coordinate assolute
fprintf(gcodeFile, 'G1 F2400 \n'); % velocità 40 mm/s (in mm/min)
fprintf(gcodeFile, 'G0 X%.3f Y%.3f\n', px(1), py(1)); % movimento rapido

for i = 2:length(px)
    fprintf(gcodeFile, 'G1 X%.3f Y%.3f\n', px(i), py(i));
end

fclose(gcodeFile);

```

Il comando “fopen” apre o crea un file denominato “gcode” in modalità scrittura (w) e restituisce un identificatore (gcodeFile) da utilizzare per le successive operazioni di scrittura del codice. Se il file “gcode” è già esistente, il contenuto viene sovrascritto. A questo punto è possibile scrivere i vari comandi all'interno del file.

Il comando “G21” imposta le unità di misura in millimetri, mentre “G90” attiva la modalità a coordinate assolute, in cui ogni posizione è riferita all'origine e non al punto precedente. Successivamente, il comando “G1 F2400” imposta una velocità di avanzamento di 2400 mm/min, corrispondente a 40 mm/s, che resterà la stessa per tutti i movimenti finché non viene modificata. Il comando “G1 X%.3f Y%.3f\n', px(i), py(i)” genera un movimento rapido verso il primo punto della traiettoria, avente le coordinate X e Y corrispondenti a px(1) e py(1) ed espresse con tre cifre decimali.

Mediante un ciclo for si scorre dal secondo fino all'ultimo punto dei vettori px e py, scrivendo per ciascun punto un comando “G1” che esegue un movimento lineare a velocità costante verso le coordinate corrispondenti. Infine, il file viene chiuso con il comando “fclose(gcodeFile)”.

1	G21	30	G1 X8.653 Y5.769
2	G90	31	G1 X8.986 Y5.991
3	G1 F2400	32	G1 X9.319 Y6.213
4	G0 X0.000 Y0.000	33	G1 X9.652 Y6.435
5	G1 X0.333 Y0.222	34	G1 X9.985 Y6.656
6	G1 X0.666 Y0.444	35	G1 X10.317 Y6.878
7	G1 X0.998 Y0.666	36	G1 X10.650 Y7.100
8	G1 X1.331 Y0.888	37	G1 X10.983 Y7.322
9	G1 X1.664 Y1.109	38	G1 X11.316 Y7.544
10	G1 X1.997 Y1.331	39	G1 X11.649 Y7.766
11	G1 X2.330 Y1.553	40	G1 X11.982 Y7.988
12	G1 X2.663 Y1.775	41	G1 X12.314 Y8.210
13	G1 X2.995 Y1.997	42	G1 X12.647 Y8.431
14	G1 X3.328 Y2.219	43	G1 X12.980 Y8.653
15	G1 X3.661 Y2.441	44	G1 X13.313 Y8.875
16	G1 X3.994 Y2.663	45	G1 X13.631 Y9.070
17	G1 X4.327 Y2.884	46	G1 X13.960 Y9.219
18	G1 X4.659 Y3.106	47	G1 X14.304 Y9.286
19	G1 X4.992 Y3.328	48	G1 X14.646 Y9.183
20	G1 X5.325 Y3.550	49	G1 X14.895 Y8.809
21	G1 X5.658 Y3.772	50	G1 X14.985 Y8.356
22	G1 X5.991 Y3.994	51	G1 X15.000 Y7.967
23	G1 X6.324 Y4.216	52	G1 X15.000 Y7.567
24	G1 X6.656 Y4.438	53	G1 X15.000 Y7.167
25	G1 X6.989 Y4.659	54	G1 X15.000 Y6.767
26	G1 X7.322 Y4.881	55	G1 X15.000 Y6.367
27	G1 X7.655 Y5.103	56	G1 X15.000 Y5.967
28	G1 X7.988 Y5.325	57	G1 X15.000 Y5.567
29	G1 X8.321 Y5.547	58	G1 X15.000 Y5.167

Figura 3.12: File G-code generato per la realizzazione traiettoria pianificata dell'ugello di stampa

Di seguito viene riportato lo script completo.

```
% Raccordo di Bézier parabolico fra due segmenti
%Inizializzazione
clc
clear all
close all

% Coordinate punti poligono di controllo
P0= [0 0]';
P1= [15 10]';
P2= [15 5]';
delta = 2;           % raggio raccordo [mm]
dt = 0.01;
v = 40;             % [mm/s]

% Calcolo intersezioni con circonferenza di raggio delta
%con relazioni trigonometriche
alfa = atan((P1(2)-P0(2))/(P1(1)-P0(1)));
p1i = P1 - delta*[cos(alfa) sin(alfa)];
p1u = P1 - [0 delta]';

%Plottaggio punti di inizio e fine dei 3 tratti
figure(1)
plot(P0(1),P0(2),'ro',P1(1),P1(2),'ko',P2(1),P2(2),'ro',...
     p1i(1),p1i(2),'r*',p1u(1),p1u(2),'r*', 'linewidth',2)
axis equal
xlabel('asse x (mm)')
ylabel('asse y (mm)')
grid on
hold on

% Tratto di ingresso rettilineo
k = 1;
u(k) = 0;

%ciclo su k per u<1
while u(k)<1
    l_1 = norm(p1i-P0); % lunghezza primo tratto rettilineo
    du=v/l_1*dt;      % incremento du
    u(k+1) = u(k)+du;
    k=k+1;
end

u = u(u <=1) ;      % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_01i = P0*(1-u) + p1i*u; %Bezier primo tratto lineare

% Raccordo parabolico
% Implementare un ciclo iterativo per il calcolo di u(k+1) a partire da u(k)
u=[]; % reinizializzo parametro u
dpsudu = []; % reinizializzo dp/du

k = 1; % reinizializzo contatore
```

```

dpsudu0 = 2*(P1-p1i) ; % reinizializzo dp/du
u(k)=(dt-norm(p1i-p_01i(:,end))/v)*(v/norm(dpsudu0)); % reinizializzo parametro u

%ciclo su k per u<1
while u(k)<1
    dpsudu = -2*(1-u(k))*p1i+2*(1-2*u(k))*P1+2*u(k)*p1u;
    du = v/norm(dpsudu)*dt;
    u(k+1) = u(k) + du;
    k=k+1;
end

u = u(u <=1) ; % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_rac = (1-u).^2.*p1i + 2*(1-u).*u.*P1+u.^2.*p1u; %Bezier di raccordo

%% Tratto di uscita rettilineo
%%Implementare un ciclo iterativo per il calcolo di u(k+1) a partire da u(k)

u = []; % reinizializzo parametro u
dpsudu = []; % reinizializzo dp/du

k = 1; % reinizializzo contatore

dpsudu0 = 2*(p1u-P1) ; % reinizializzo dp/du
u(k)=(dt-norm(p1u-p_rac(:,end))/v)*(v/norm(dpsudu0)); % reinizializzo parametro u

while u(k)<1
    l_2 = norm(P2-p1u); % lunghezza secondo tratto rettilineo
    du=v/l_2*dt; % incremento du
    u(k+1) = u(k)+du;
    k=k+1;
end

u = u(u <=1) ; % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_1u2 = p1u*(1-u) + P2*u; %Bezier secondo tratto lineare

%% Assemblo i 3 tratti

p = [p_01i p_rac p_1u2];
px = p(1,:); %prima riga, componenti x
py = p(2,:); %seconda riga, componenti y

figure(1)
plot(px,py,'bo','linewidth',1.5)
xlabel('x [mm]')
ylabel('y [mm]')
title('Spostamento nel piano XY')
grid on

%% Vettore tempo
N = size(p,2);
t = zeros(1,N);

for k = 2:N

```

```

    %ds = norm(p(:,k) - p(:,k-1));
    ds = sqrt((p(1,k) - p(1,k-1))^2 + (p(2,k) - p(2,k-1))^2); % distanza tra punti
consecutivi
    t(k) = t(k-1) + ds/v;
end
%% Plot x(t) e y(t)
figure (2)
subplot(2,1,1)
plot(t, px, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('x [mm]')
title('Spostamento x(t)')
grid on

subplot(2,1,2)
plot(t, py, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('y [mm]')
title('Spostamento y(t)')
grid on

%% Velocità di stampa
dx = diff(px); % differenze di posizione
dt = diff(t); % differenze di tempo
vx = dx./dt; % velocità media tra punti
t_v = t(1:end-1); % istanti associati alla velocità

figure (3)
subplot(2,1,1)
plot(t_v, vx, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('v_x [mm/s]')
title('Velocità v_x(t)')
grid on

dy = diff(py); % differenze di posizione
dt = diff(t); % differenze di tempo
vy = dy./dt; % velocità media tra punti
t_v = t(1:end-1); % istanti associati alla velocità

figure (3)
subplot(2,1,2)
plot(t_v, vy, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('v_y [mm/s]')
title('Velocità v_y(t)')
grid on

% Calcolo del modulo della velocità
v = sqrt(vx.^2 + vy.^2);

figure (4)
plot(t_v, v, 'b-', 'LineWidth', 2)
xlabel('Tempo [s]')
ylabel('Velocità nel piano [mm/s]')
title('Modulo della velocità nel piano XY')
grid on

```

```
%% Generazione Gcode
gcodeFile = fopen('gcode', 'w');

fprintf(gcodeFile, 'G21 \n');      % usa mm
fprintf(gcodeFile, 'G90 \n');      % coordinate assolute
fprintf(gcodeFile, 'G1 F2400 \n'); % velocità 40 mm/s (in mm/min)
fprintf(gcodeFile, 'G0 X%.3f Y%.3f\n', px(1), py(1)); % movimento rapido

for i = 2:length(px)
    fprintf(gcodeFile, 'G1 X%.3f Y%.3f\n', px(i), py(i));
end

fclose(gcodeFile);
```

4 Motori passo-passo

I motori passo-passo sono dispositivi elettromeccanici in grado di trasformare impulsi elettrici in movimenti rotatori discreti dell'albero motore.

A differenza dei motori tradizionali, il loro funzionamento si basa su step precisi: ogni impulso fa avanzare l'albero motore di un angolo determinato, e terminato il passo si arresta mantenendo la posizione consentendo un controllo accurato della posizione senza necessità di sensori di retroazione. [17]

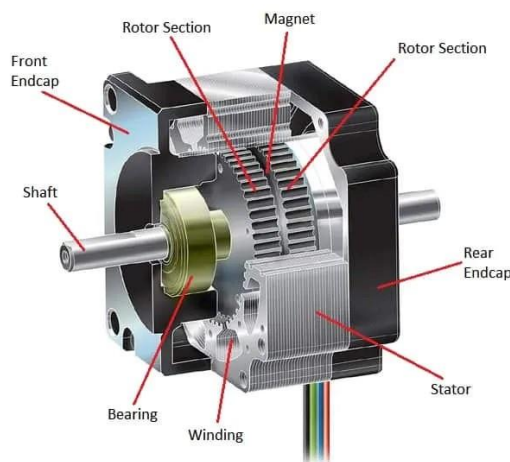


Figura 4.1: Componenti principali di un motore passo-passo [18]

Per questo motivo, i motori passo-passo operano prevalentemente in configurazione ad anello aperto, affidandosi esclusivamente alla sequenza di comandi ricevuti dal sistema di controllo. Il controllo del motore passo-passo si basa sulla sequenza e sulla frequenza degli impulsi elettrici. Un generatore di impulsi invia i segnali alle bobine attraverso transistor che ne regolano la corrente e la velocità, mentre un contatore calcola il numero di impulsi necessari per raggiungere la posizione desiderata.

Oggi i sistemi di controllo possono essere implementati con microprocessori, riducendo costi e complessità. Questi dispositivi consentono diverse modalità operative, come full-step, half-step e micro-step, in base alla precisione richiesta. Due configurazioni di collegamento influenzano la coppia e la velocità del motore: il collegamento bipolare a quattro derivazioni massimizza la coppia a basse velocità, mentre il collegamento unipolare a sei derivazioni offre velocità maggiori a scapito della coppia.

Sono in commercio diverse tipologie di motore passo-passo: motore a riluttanza variabile (VR), Motore a magnete permanente (PM), Motore passo-passo ibrido.

4.1 Motore passo-passo a riluttanza variabile (VR)

I motori a riluttanza variabile sfruttano la tendenza dei materiali ferromagnetici a posizionarsi lungo il percorso di minima riluttanza magnetica. Il rotore è in ferro dolce, quindi non è magnetizzato in modo permanente, ma si magnetizza solo in presenza del campo magnetico dello statore, perdendo poi la polarità quando le bobine statoriche vengono diseccitate.

Lo statore è dotato di denti con avvolgimenti che, quando eccitati in sequenza, attraggono i denti del rotore, producendo così un movimento rotatorio a step. Per definire il senso di rotazione sono necessarie almeno tre fasi sullo statore. L'angolo di passo dipende dal numero di denti del rotore n_r e dello statore n_s . [17] [19]

4.1.1 Funzionamento del motore passo-passo a riluttanza variabile (VR)

Si consideri ad esempio un motore passo-passo con $n_s = 6$ e $n_r = 4$.

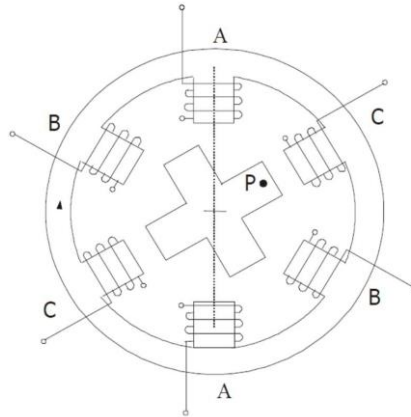


Figura 4.2: Schema di un motore passo-passo a riluttanza variabile con $n_s = 6$ e $n_r = 4$ [19]

Il numero di fasi si calcola come $\frac{n_s}{2} = 3$, si avranno quindi tre fasi, indicate con A, B, C.

Per ciascuna fase, un capo dell'avvolgimento è collegato all'anodo mentre l'altro a catodo. Il motore a riluttanza variabile è unipolare, pertanto non è possibile invertire la polarità elettrica ai capi dell'avvolgimento.

Nella Figura 4.3 sono riportate le quattro sequenze di attivazione (o eccitazione) degli avvolgimenti, che determinano la rotazione del rotore per passi successivi.

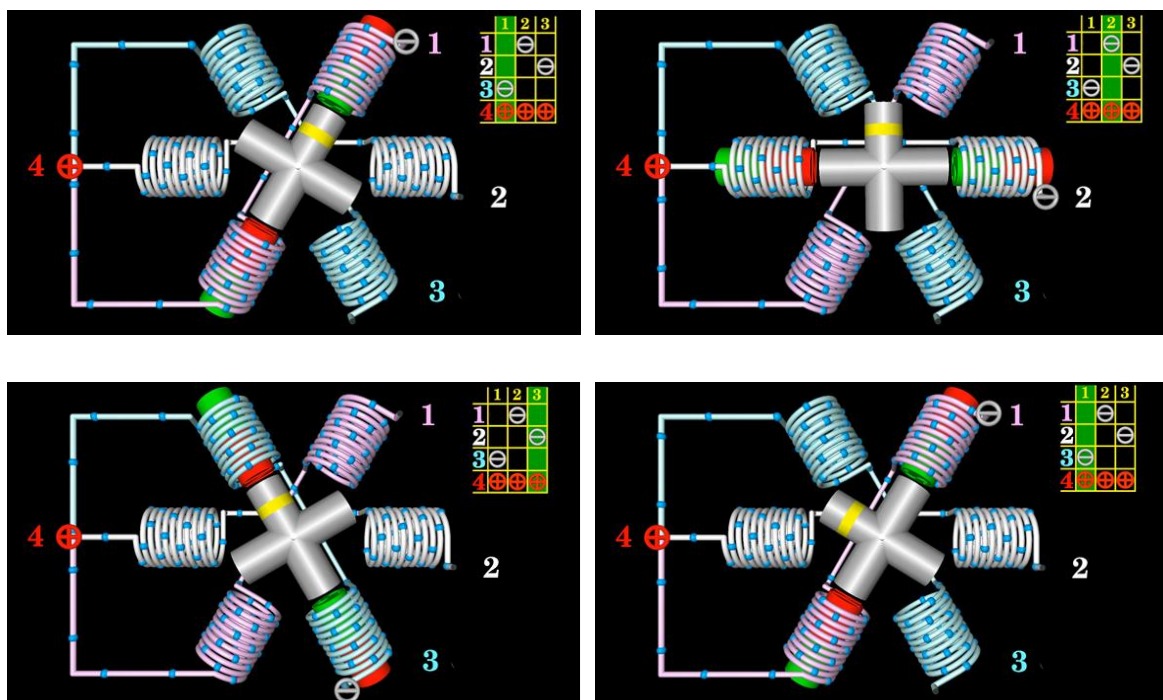


Figura 4.3: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a riluttanza variabile con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento full step [20]

Il passo tra i denti dello statore risulta pari a $\theta_s = \frac{360^\circ}{6} = 60^\circ$, mentre il passo tra i denti di rotore è $\theta_r = \frac{360^\circ}{4} = 90^\circ$. Dopo 3 passi il rotore compie $\frac{1}{4}$ di giro, in quanto un dente di rotore si viene a trovare nella posizione precedentemente occupata dal dente successivo. L'angolo di passo risulta quindi pari a $\gamma = \frac{1}{3} \cdot \left(\frac{360^\circ}{4}\right) = 30^\circ$, per un totale di 12 passi/giro. Aumentando il numero di denti del rotore si riduce il passo. Si desidera ora dimezzare l'ampiezza del passo, portandola a 15° (ovvero raddoppiare la risoluzione angolare) senza modificare il numero di coppie polari del motore. Per ottenere questo risultato si utilizza la tecnica half-step. Tale tecnica consiste nell'alternare l'eccitazione di una singola fase con l'eccitazione simultanea di due fasi, così da generare posizioni intermedie del rotore rispetto a quelle del funzionamento in full-step. La sequenza di attivazione degli avvolgimenti in modalità half-step è illustrata nello schema riportato in Figura 4.4.

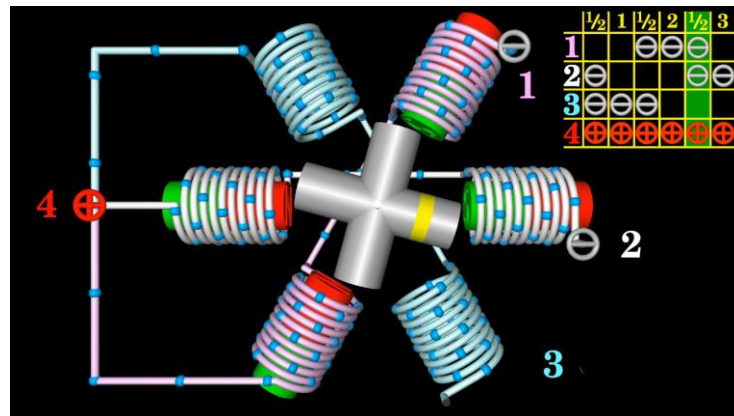


Figura 4.4: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a riluttanza variabile con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento half step [20]

Questi motori sono generalmente i più semplici dal punto di vista costruttivo, tuttavia, la coppia erogata e la precisione risultano inferiori rispetto ad altre tipologie di motori passo-passo.

4.2 Motore passo-passo a magnete permanente (PM)

I motori passo-passo a magnete permanente sono caratterizzati da un rotore dotato di magneti permanenti alle estremità e da uno statore con avvolgimenti elettrici. L'eccitazione sequenziale delle bobine statoriche genera un campo magnetico variabile che interagisce con il rotore, producendo un movimento rotatorio a passi. Questi motori, tipicamente a due fasi e sono indicati per applicazioni in cui non è richiesta un'elevata precisione. L'angolo di passo più utilizzato per questo motore è 7.5° per cui sono sufficienti pochi impulsi per raggiungere la posizione desiderata. [17] [19]

4.2.1 Funzionamento del motore passo-passo a magnete permanente (PM)

Per questa tipologia di motore è necessario approfondire il funzionamento in modo più dettagliato rispetto al motore a riluttanza variabile. La logica delle sequenze di attivazione di un motore a magnete permanente è infatti molto simile a quella di un motore ibrido, che

rappresenta il motore più usato nelle applicazioni delle macchine FDM. Comprendere il funzionamento del motore PM rende quindi più semplice modellizzare, tramite MATLAB, la logica del motore ibrido.

Si consideri un motore passo-passo bipolare a magneti permanenti, dove il termine bipolare indica la possibilità di invertire la polarità elettrica degli avvolgimenti.

Si assume un rotore con tre coppie polari (sei magneti permanenti) e uno statore con due coppie di avvolgimenti, indicati come fase A e fase B. L'avvolgimento della fase A presenta i terminali 1 e 2, mentre quello della fase B i terminali 3 e 4.

Nella Figura 4.5 sono riportate le quattro sequenze di attivazione (o eccitazione) degli avvolgimenti, che determinano la rotazione del rotore per passi successivi.

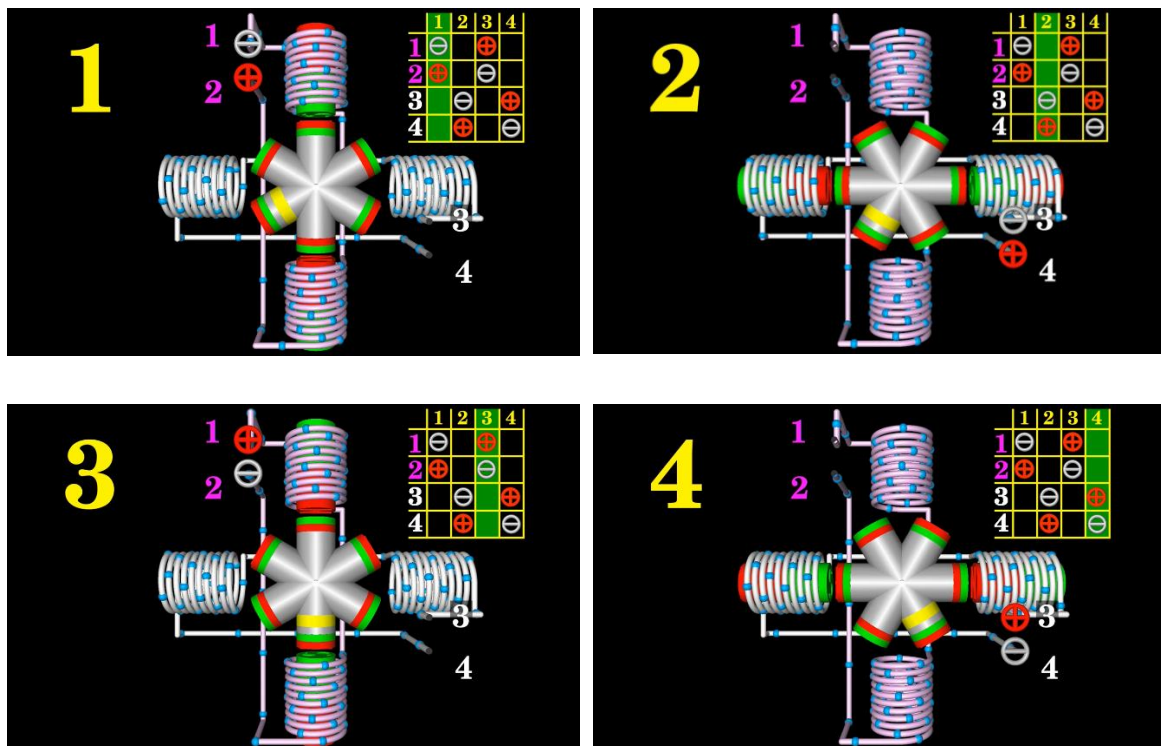


Figura 4.5: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento full step [20]

Per determinare il verso del campo magnetico in un avvolgimento, si può usare la regola della mano destra, che consiste nell'avvolgere le dita della mano destra nella direzione della corrente che percorre le spire dell'avvolgimento e il verso del campo magnetico all'interno della bobina, corrispondente al polo nord, sarà indicato dal verso del pollice.

Per stabilire la direzione della corrente si deve ricordare che gli elettroni si muovono sempre dal polo negativo al polo positivo, tuttavia, per convenzione la corrente elettrica ha verso opposto a quello del flusso di elettroni e quindi va dal polo positivo al polo negativo dell'avvolgimento.

Nella Tabella 4.1 sono riportate le informazioni contenute nella Figura 4.5. Per ciascun passo si indicano:

- la fase attiva (A, B, \bar{A} o \bar{B} , dove le ultime due rappresentano le stesse fasi con polarità invertita);
- il verso della corrente nell'avvolgimento attivo;

- i corrispondenti poli positivo e negativo.

Tabella 4.1: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento full-step

Numero di passi	Fase attiva	Verso della corrente	Polo positivo	Polo negativo
1	A	$2 \rightarrow 1$	2	1
2	B	$4 \rightarrow 3$	4	3
3	\bar{A}	$1 \rightarrow 2$	1	2
4	\bar{B}	$3 \rightarrow 4$	3	4

Dopo 2 passi il rotore compie $\frac{1}{6}$ di giro, in quanto un dente di rotore si viene a trovare nella posizione precedentemente occupata dal dente successivo. L'angolo di passo risulta quindi pari a $\gamma = \frac{1}{2} \cdot \left(\frac{360^\circ}{6}\right) = 30^\circ$, per un totale di 12 passi/giro.

Si desidera ora dimezzare l'ampiezza del passo, portandola a 15° (ovvero raddoppiare la risoluzione angolare), senza modificare il numero di coppie polari del motore.

Per ottenere questo risultato si utilizza la tecnica half-step, le cui sequenze di attivazione sono illustrate nello schema riportato in Figura 4.6.

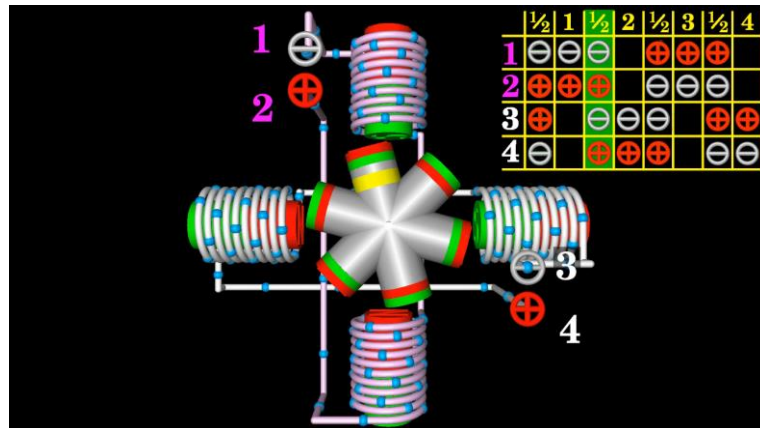


Figura 4.6: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento half step [20]

Nella Tabella 4.2 si riporta la nuova sequenza di attivazione in modalità half step, indicando per ciascun passo:

- la fase o le fasi attive;
- il verso della corrente in ogni avvolgimento attivo;
- i corrispondenti poli positivo e negativo.

Tabella 4.2: Sequenza di attivazione degli avvolgimenti di un motore passo-passo a magnete permanente con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento half-step

Numero di passi	Fase attiva	Verso della corrente	Polo positivo	Polo negativo
0.5	A	2→1	2	1
	\bar{B}	3→4	3	4
1	A	2→1	2	1
1.5	A	2→1	2	1
	B	4→3	4	3
2	B	4→3	4	3
2.5	\bar{A}	1→2	1	2
	B	4→3	4	3
3	\bar{A}	1→2	1	3
3.5	\bar{A}	1→2	1	2
	\bar{B}	3→4	3	4
4	\bar{B}	3→4	3	4

4.3 Motore passo-passo ibrido

I motori più utilizzati nelle stampanti FDM sono i motori passo-passo ibridi bipolari, che combinano le caratteristiche dei motori a magnete permanente e di quelli a riluttanza, offrendo la miglior precisione e coppia continua tra tutti i motori passo-passo. Il rotore di questi motori possiede sia un magnete permanente sia una serie di denti.

Sono disponibili in configurazioni a due, tre o cinque fasi e offrono angoli di passo ridotti, tipicamente $1,8^\circ$. [17] [19]



Figura 4.7: Motore passo-passo ibrido [21]

In Figura 4.8, viene mostrato lo schema di un motore passo-passo ibrido da 200 step/giro, evidenziando in particolare il disallineamento tra rotore e statore da cui deriva l'angolo di passo di $1,8^\circ$.

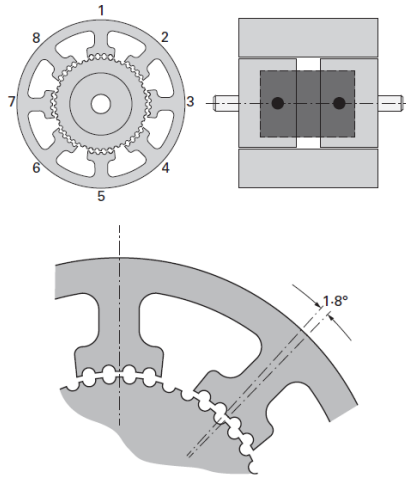


Figura 4.8: Schema di un motore passo-passo ibrido da 200 step/giro [17]

4.3.1 Funzionamento del motore passo-passo ibrido

La struttura interna di un motore ibrido (Figura 4.9) è più complessa: il rotore è costituito da due parti in ferro dolce con piccoli denti, collegate da un magnete orientato assialmente e sfalsate di mezzo dente. Lo statore presenta denti principali e denti minori, ciascuno con bobine di rame avvolte su ogni dente in configurazione bipolare.

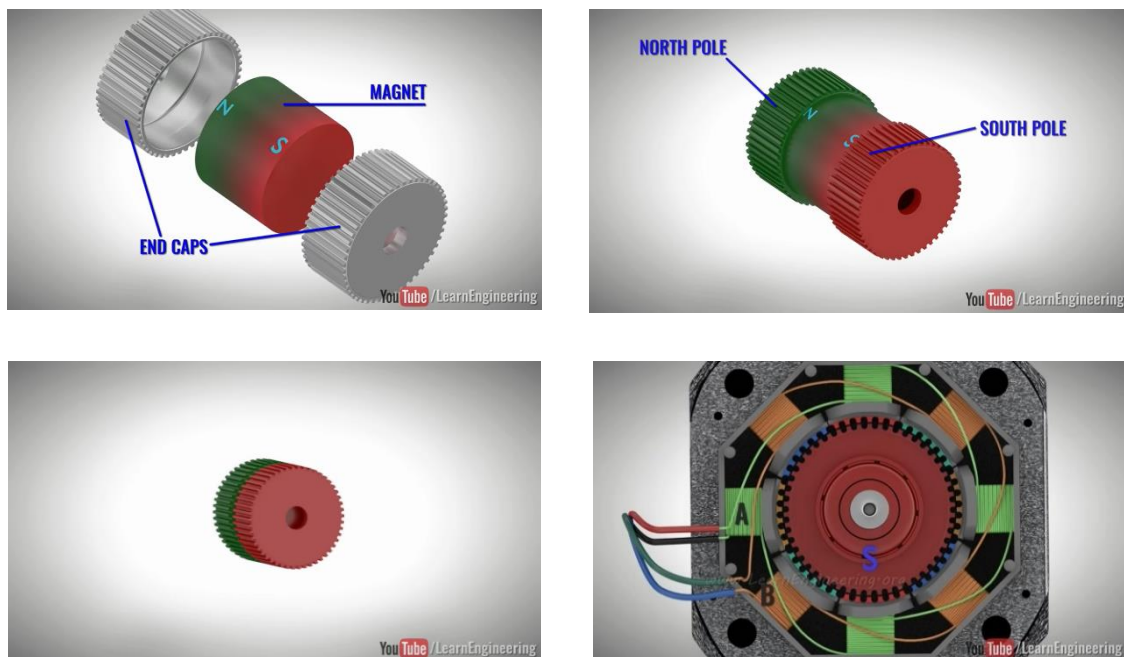


Figura 4.9: Struttura interna di un motore ibrido [22]

Questo schema garantisce movimento uniforme, coppia elevata e alta precisione.

Nel esempio successivo si considera un rotore con 13 denti per ciascuna parte, quindi complessivamente 26 denti, mentre sullo statore sono presenti 12 denti collegati a due fasi bipolari.

Nella Figura 4.10 sono riportate le quattro sequenze di attivazione (o eccitazione) degli avvolgimenti, che determinano la rotazione del rotore per passi successivi.

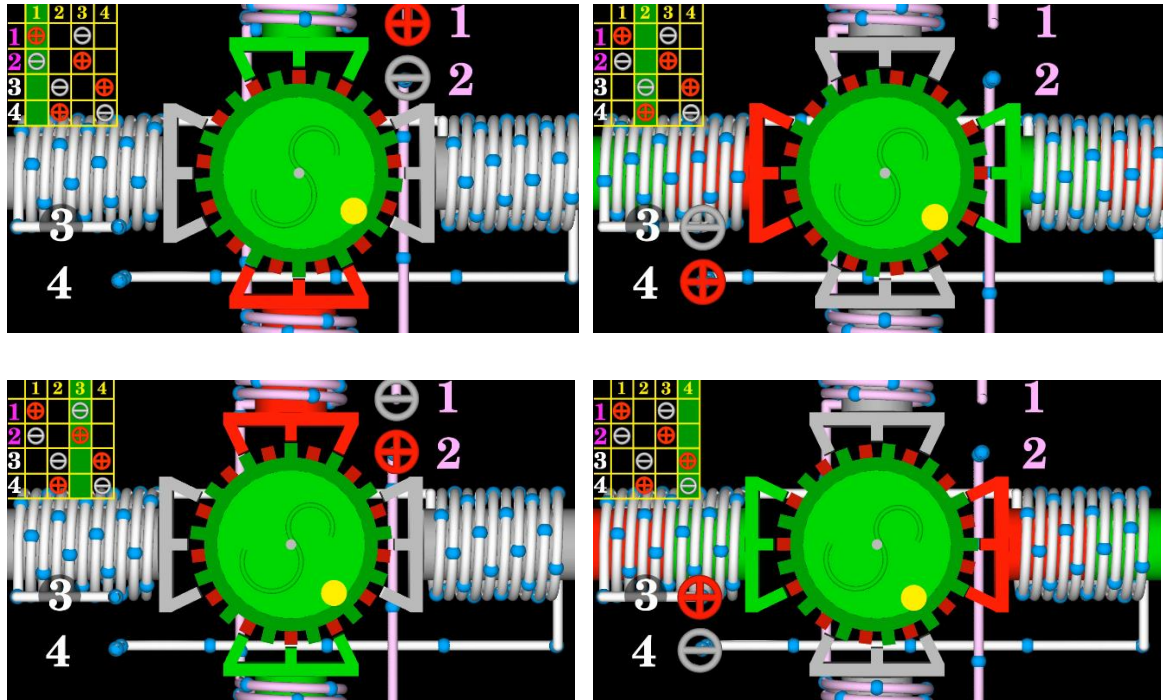


Figura 4.10: Sequenza di attivazione degli avvolgimenti di un motore passo-passo ibrido con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento full step [20]

Si osserva che in ogni passo della sequenza, i denti di una parte e dell'altra si affacciano a poli opposti dello statore.

Dopo 4 passi, il rotore compie $\frac{1}{13}$ di giro, in quanto un dente di rotore (di una singola parte) si viene a trovare nella posizione precedentemente occupata dal dente successivo. L'angolo di passo risulta quindi $\gamma = \frac{1}{4} \cdot \left(\frac{360^\circ}{13}\right) = 6.92^\circ$, per un totale di 52 passi/giro.

Si desidera ora dimezzare l'ampiezza del passo, portandola a 3.46° (ovvero raddoppiare la risoluzione angolare), senza modificare il numero di coppie polari del motore.

Per ottenere questo risultato si utilizza la tecnica half-step, le cui sequenze di attivazione sono illustrate nello schema riportato in Figura 4.11.

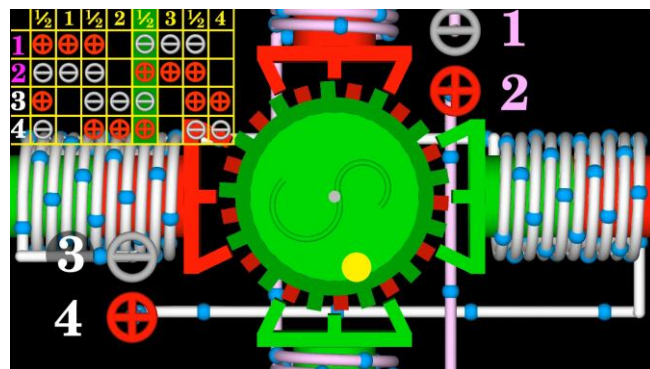


Figura 4.11: Sequenza di attivazione degli avvolgimenti di un motore passo-passo ibrido con $n_s = 6$ e $n_r = 4$ in modalità di funzionamento half step [20]

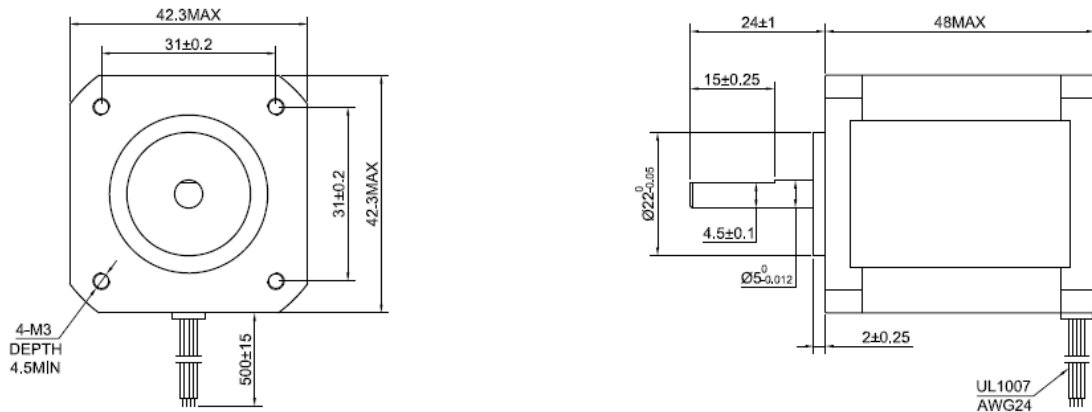
Dalle sequenze presenti nelle figure, si osserva che la logica delle sequenze di attivazione degli avvolgimenti di un motore passo-passo ibrido bipolare è identica a quella di un motore

passo-passo a magneti permanenti bipolare. Pertanto, è possibile utilizzare le stesse sequenze di attivazione già riportate nelle tabelle 4.1 e 4.2.

5 Modellazione di un motore passo-passo ibrido in Simulink/Simscape

L'obiettivo di questo capitolo consiste nel modellare un motore passo-passo ibrido e implementare un semplice controllo utilizzando Simulink/Simscape, al fine di osservare l'andamento della posizione angolare dell'albero in uscita.

I parametri del motore sono stati presi da catalogo e corrispondono al modello 17HM19-2004S, che ha un angolo di passo di 0.9°.



SPECIFICATION	CONNECTION	BIPOLAR	TYPE OF CONNECTION (EXTERN)				MOTOR																																		
AMPS/PHASE		2.00	PIN NO		BIPOLAR	LEADS	WINDING																																		
RESISTANCE/PHASE(Ohms)@25°C		1.45±10%	1	A+ —	BLK	A+ —																																			
INDUCTANCE/PHASE(mH)@1KHz		4.00±20%	2	A- —	GRN	A- —																																			
HOLDING TORQUE(Nm) [lb-in]		0.46[4.07]	3	B+ —	RED	B+ —																																			
STEP ANGLE(°)		0.90	4	B- —	BLU	B- —																																			
STEP ACCURACY(NON-ACCUM)		±5.00%	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>						STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+							A-	B-	CCW																														
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
ROTOR INERTIA(g-cm²)		68,00	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>						STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+	A-	B-	CCW																																				
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
WEIGHT(Kg) [lb]		—	<table border="1"> <thead> <tr> <th>TYPE OF CONNECTION (EXTERN)</th> <th colspan="2">MOTOR</th> </tr> <tr> <th>PIN NO</th> <th>BIPOLAR</th> <th>LEADS</th> <th>WINDING</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>A+ —</td> <td>BLK</td> <td>A+ —</td> </tr> <tr> <td>2</td> <td>A- —</td> <td>GRN</td> <td>A- —</td> </tr> <tr> <td>3</td> <td>B+ —</td> <td>RED</td> <td>B+ —</td> </tr> <tr> <td>4</td> <td>B- —</td> <td>BLU</td> <td>B- —</td> </tr> </tbody> </table>						TYPE OF CONNECTION (EXTERN)	MOTOR		PIN NO	BIPOLAR	LEADS	WINDING	1	A+ —	BLK	A+ —	2	A- —	GRN	A- —	3	B+ —	RED	B+ —	4	B- —	BLU	B- —										
TYPE OF CONNECTION (EXTERN)	MOTOR																																								
PIN NO	BIPOLAR	LEADS	WINDING																																						
1	A+ —	BLK	A+ —																																						
2	A- —	GRN	A- —																																						
3	B+ —	RED	B+ —																																						
4	B- —	BLU	B- —																																						
TEMPERATURE RISE;MAX.80°C (MOTOR STANDSTILL;FOR 2PHASE ENERGIZED)	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>								STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+	A-	B-	CCW																																				
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
AMBIENT TEMPERATURE -10°C-50°C [14°F-122°F]	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>								STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+	A-	B-	CCW																																				
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
INSULATION RESISTANCE 100 Mohm (UNDER NORMAL TEMPERATURE AND HUMIDITY)	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>								STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+	A-	B-	CCW																																				
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
INSULATION CLASS B 130°C [266°F]	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>								STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+	A-	B-	CCW																																				
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
DIELECTRIC STRENGTH 500VAC FOR 1MIN.(BETWEEN THE MOTOR COILS AND THE MOTOR CASE)	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>								STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+	A-	B-	CCW																																				
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
AMBIENT HUMIDITY MAX.85%(NO CONDENSATION)	<table border="1"> <thead> <tr> <th>STER</th> <th>A+</th> <th>B+</th> <th>A-</th> <th>B-</th> <th colspan="2">CCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>2</td> <td>-</td> <td>+</td> <td>+</td> <td>-</td> <td colspan="2" rowspan="2">CW</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>4</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td colspan="2"></td> </tr> </tbody> </table>								STER	A+	B+	A-	B-	CCW		1	+	+	-	-	↓	↑	2	-	+	+	-	CW		3	-	-	+	+	4	+	-	-	+		
STER	A+	B+	A-	B-	CCW																																				
1	+	+	-	-	↓	↑																																			
2	-	+	+	-	CW																																				
3	-	-	+	+																																					
4	+	-	-	+																																					
		APVD		10.30.2020	STEPPER MOTOR 17HM19-2004S																																				
		CHKD																																							
		1:1	DRN																																						
		SCALE	SIGNATURE	DATE																																					

Figura 5.1: Motore passo-passo 17HM19-2004S Full Datasheet [23]

5.1 Modello Simulink/Simscape

Il modello del controllo e del motore passo-passo implementato è riportato in Figura 5.2.

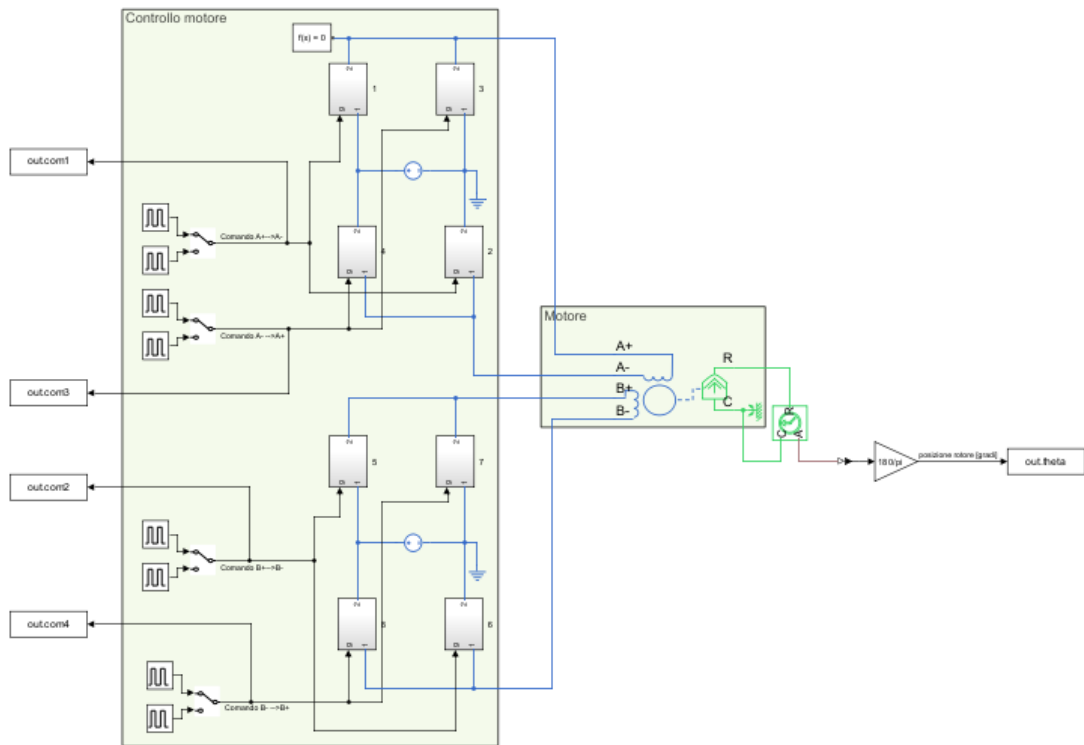


Figura 5.2: Modello Simulink/Simscape del motore passo-passo e del relativo sistema di controllo

Di seguito, i blocchi utilizzati.

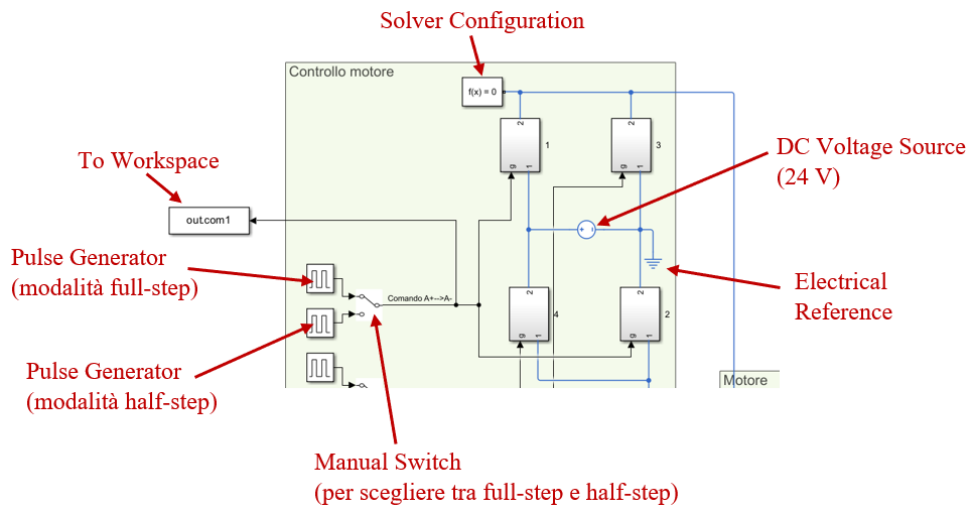


Figura 5.3: Descrizione dei blocchi che compongono il modello riportato in Figura 5.2

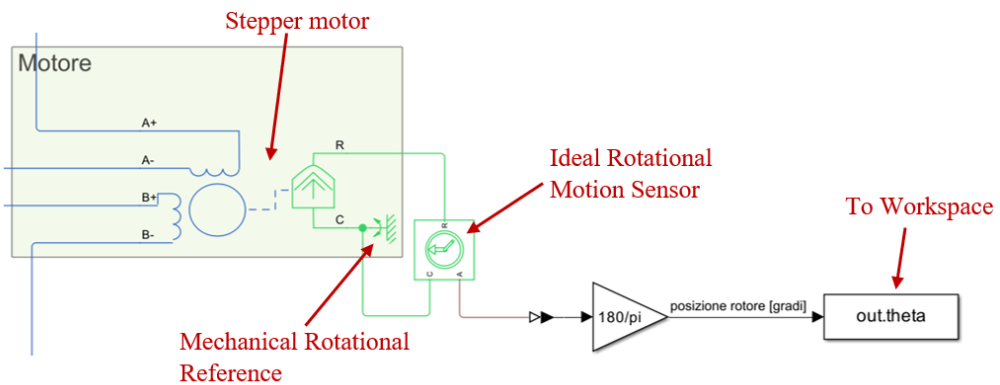


Figura 5.4: Descrizione dei blocchi che compongono il modello riportato in Figura 5.2

5.1.1 Generatori di impulsi

Il blocco Pulse Generator consente di generare un segnale a onda quadra caratterizzato da impulsi periodici. La forma dell'uscita è determinata da un insieme di parametri: Amplitude, Pulse Width, Period e Phase delay.

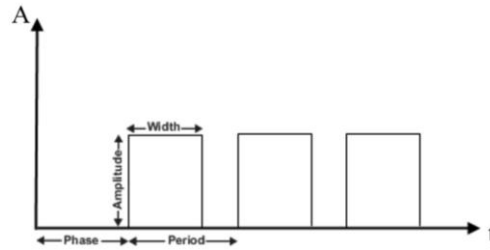


Figura 5.5: Segnale a onda quadra [24]

Il parametro Pulse type, impostato di default in modalità Time based, stabilisce che la generazione del segnale avvenga in funzione del tempo. In questa configurazione si sceglie di utilizzare il tempo di simulazione (Use simulation time).

L'Amplitude, con valore predefinito pari a 1, è uno scalare che definisce l'ampiezza dell'onda durante la fase attiva dell'impulso.

Il Period (secs), anch'esso scalare, rappresenta la durata di un ciclo completo del segnale ed è espresso in secondi. Questo parametro determina l'intervallo temporale al termine del quale l'impulso si ripete, risultando quindi direttamente collegato alla frequenza del segnale generato.

La Pulse width, compresa tra 0 e 100, indica la percentuale del periodo durante la quale il segnale assume valore attivo. Questo parametro definisce il duty cycle dell'onda quadra, cioè il rapporto tra il tempo di attivazione e la durata complessiva del periodo.

Infine, il Phase delay (secs), introduce un ritardo iniziale, espresso in secondi, prima della generazione del primo impulso. Esso consente di traslare temporalmente il segnale rispetto all'istante iniziale della simulazione, permettendo un controllo più accurato della sincronizzazione con eventuali altri segnali presenti nel modello.

In Figura 5.6, viene mostrato un esempio di configurazione del blocco per ottenere il segnale riportato in Figura 5.7.

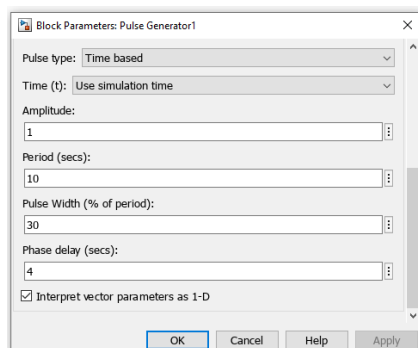


Figura 5.6: Configurazione del blocco Pulse Generator per ottenere il segnale riportato in Figura 5.7

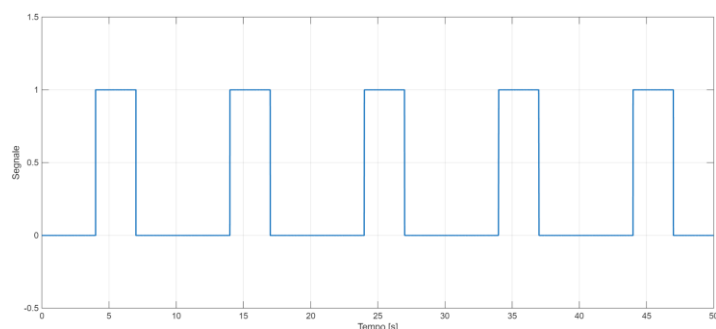


Figura 5.7: Esempio di un segnale a onda quadra

5.1.2 Ponti H

Per controllare elettricamente il motore si utilizzano quattro generatori di impulsi e due ponti H, in modo da poter attivare e disattivare gli avvolgimenti e controllarne anche il verso della corrente secondo la sequenza di attivazione delle fasi desiderata.

Il primo ponte H è collegato all'avvolgimento della fase A e, a monte del ponte, sono collegati i due blocchi Pulse Generator relativi al comando A (verso corrente: $A^+ \rightarrow A^-$) e al comando \bar{A} (verso corrente: $A^- \rightarrow A^+$).

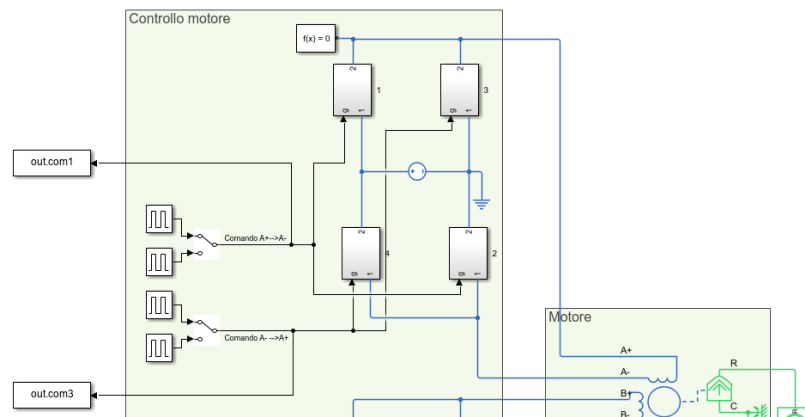


Figura 5.8: Dettaglio del primo ponte H riportato in Figura 5.2

Il secondo ponte H è collegato all'avvolgimento della fase B e, a monte del ponte, sono collegati i due blocchi Pulse Generator relativi al comando B (verso corrente: $B^+ \rightarrow B^-$) e al comando \bar{B} (verso corrente: $B^- \rightarrow B^+$).

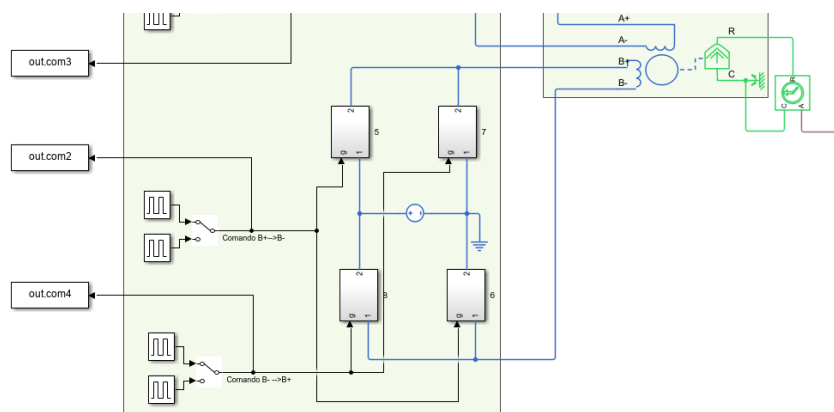


Figura 5.9: Dettaglio del secondo ponte H riportato in Figura 5.2

Per ciascun comando è presente un blocco Manual Switch che permette di scegliere tra due generatori di impulsi. Quando tutti e quattro i Manual Switch sono impostati verso l'alto, si selezionano i generatori di impulsi per il funzionamento in full step, quando tutti sono verso il basso, il funzionamento avviene in half-step.

Ciascun ponte H è costituito da quattro interruttori modellati nel seguente modo:

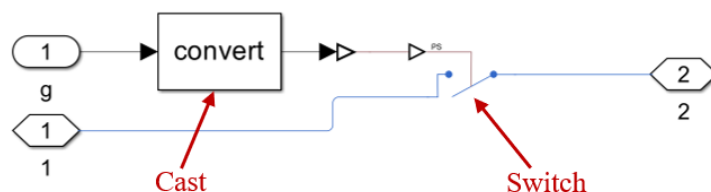


Figura 5.10: Modello di uno degli interruttori dei ponti H

Il blocco Switch rappresenta un interruttore controllato da un segnale fisico esterno privo di unità di misura, inviato tramite la porta PS. [25]

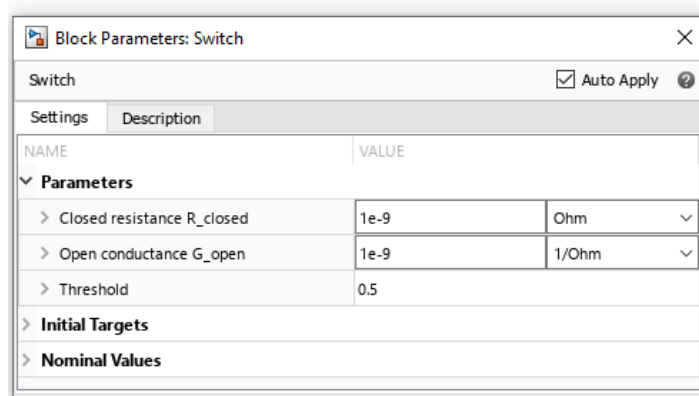


Figura 5.11: Configurazione del blocco Switch

Il segnale di controllo determina lo stato dell'interruttore: se il valore del segnale applicato alla porta PS supera la soglia definita dal parametro Threshold, lo switch si chiude, in caso contrario, rimane aperto. Di conseguenza, un segnale pari a 0 mantiene l'interruttore aperto, mentre un segnale pari a 1 ne provoca la chiusura.

La resistenza a interruttore chiuso (R_{closed}) descrive la resistenza elettrica dello switch quando si trova nello stato di conduzione. Questo parametro deve essere strettamente maggiore di zero. Nel modello è stato impostato pari a $10^{-9} \Omega$, in modo da rendere il suo l'effetto praticamente trascurabile durante la simulazione.

La conduttanza a interruttore aperto (G_{open}) rappresenta invece la conduttanza dello switch quando è aperto e deve anch'essa essere positiva. Nel modello è stata impostata a $10^{-9} \frac{1}{\Omega}$, ottenendo così una resistenza equivalente molto elevata quando l'interruttore è aperto, bloccando il passaggio di corrente.

Il parametro Threshold, impostato a 0,5 e adimensionale, stabilisce il limite che discrimina tra stato aperto e chiuso dello switch. Valori del segnale PS superiori alla soglia determinano la chiusura dell'interruttore, mentre valori inferiori ne causano l'apertura.

A monte del blocco Switch è stato inserito un blocco Cast per garantire la corretta conversione del tipo di dato del segnale di controllo.

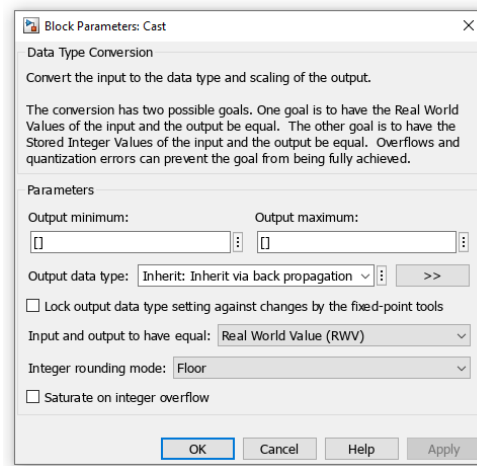


Figura 5.12: Configurazione del blocco Cast

Questo passaggio è necessario per assicurare la compatibilità tra il segnale in ingresso e il tipo di dato richiesto dal blocco Switch, evitando possibili problemi di arrotondamenti o saturazione numerica.

Il funzionamento dei due ponti H è il seguente:

Quando è attivo il comando A , si chiudono gli interruttori 1 e 2, e la corrente entra dal terminale A^+ ed esce dal terminale A^- dell'avvolgimento statorico.

Quando è attivo il comando \bar{A} , si chiudono gli interruttori 3 e 4, e la corrente entra dal terminale A^- ed esce da A^+ .

In funzione del comando attivo, la polarità dell'avvolgimento può quindi essere invertita.

Lo stesso ragionamento vale per i comandi B e \bar{B} .

5.1.3 Blocco Stepper Motor

Per la simulazione del motore passo-passo ibrido in Simscape è stato utilizzato il blocco Stepper Motor riferito ai motori a magnete permanente, poiché non è disponibile un modello specifico per motori ibridi. Il blocco non considera l'effetto della riluttanza variabile, tipico dei motori ibridi, e quindi non riproduce in modo esatto la coppia durante transitori rapidi né alcune dinamiche più complesse legate alla diversa geometria del motore. Tuttavia, queste limitazioni non influiscono sugli obiettivi trattati nei paragrafi successivi. Come verrà illustrato, l'obiettivo finale della tesi è implementare un codice MATLAB che, a partire dal G-code della traiettoria precedentemente studiata, generi le sequenze di attivazione delle bobine. In questo contesto si analizza esclusivamente il comportamento a gradini e la risposta cinematica del rotore, valutando la precisione del passo e la capacità del rotore di seguire la traiettoria desiderata. Di conseguenza, il blocco del motore a magneti permanenti risulta sufficiente, poiché la logica di attivazione delle fasi bipolari e il comportamento cinematico del rotore sono sostanzialmente equivalenti nei motori ibridi e in quelli a magnete permanente.

Sulla base di queste considerazioni, si procede con la descrizione del blocco adottato.

Il modello di motore passo-passo a magneti permanenti (Stepper Motor) descrive un dispositivo in grado di trasformare impulsi elettrici discreti in un movimento meccanico preciso. Il motore è costituito da un rotore a magnete permanente e da due avvolgimenti nello statore, indicati come fase A e fase B.

L'applicazione di una tensione alle bobine genera una corrente che produce il campo magnetico statorico che, interagendo con il campo magnetico del rotore, genera una coppia elettrica che accelera il rotore.

In particolare, la variazione della corrente nelle bobine è descritta dalle Eq. (5.1) e (5.2):

$$\frac{di_A}{dt} = \frac{(v_A - Ri_A - e_A)}{L} \quad (5.3)$$

$$\frac{di_B}{dt} = \frac{(v_B - Ri_B - e_B)}{L} \quad (5.4)$$

dove v_A e v_B sono le tensioni applicate, R e L rappresentano rispettivamente la resistenza e l'induttanza degli avvolgimenti, mentre e_A ed e_B sono le forze contro-elettromotrici indotte dal movimento del rotore stesso, espresse dalle Eq. (5.5) e (5.6).

$$e_A = -K_t \omega \sin(N_r \theta) \quad (5.5)$$

$$e_B = -K_t \omega \cos(N_r \theta) \quad (5.6)$$

Le forze contro-elettromotrici rappresentano il fenomeno per cui il movimento del rotore induce una tensione che si oppone a quella applicata agli avvolgimenti, riducendo la corrente circolante nell'avvolgimento.

La corrente nelle bobine produce la coppia elettrica T_e , espressa come:

$$T_e = -K_t \left(i_A - \frac{e_A}{R_m} \right) \sin(N_r \theta) + K_t \left(i_B - \frac{e_B}{R_m} \right) \cos(N_r \theta) - T_d \sin(4N_r \theta) \quad (5.7)$$

In questa espressione, R_m è la resistenza magnetizzante, N_r il numero di denti del rotore, K_t la costante di coppia, e T_d la detent torque, ossia la coppia residua dovuta a una parziale magnetizzazione dello statore anche in assenza di alimentazione delle fasi.

L'equilibrio meccanico al rotore è descritto dalla seguente equazione:

$$J \frac{d\omega}{dt} + B\omega = T_e \quad (5.8)$$

dove J rappresenta l'inerzia del rotore e B lo smorzamento meccanico dovuto agli attriti. La velocità angolare ω così generata determina la variazione dell'angolo del rotore.

Il blocco presenta quattro porte elettriche associate agli avvolgimenti delle due fasi: A+ e A- costituiscono rispettivamente il terminale positivo e negativo della fase A, mentre B+ e B- rappresentano il terminale positivo e negativo della fase B.

In uscita sono presenti due porte meccaniche: la porta C, collegata alla carcassa del motore (machine case), e la porta R, collegata al rotore.

La coppia elettromagnetica generata dal motore agisce tra queste due porte, descrivendo l'interazione dinamica tra parte fissa e parte mobile della macchina.

Tra le porte R e C è stato inserito un sensore di rotazione per misurare lo spostamento angolare nel rotore. [26]

Il blocco Stepper Motor è stato parametrizzato con i valori reali del motore ibrido modello 17HM19-2004S, presi da catalogo, così da garantire una rappresentazione realistica del comportamento cinematico.

%% Parametri Motore

```

step_deg = 0.9;      % Step angle (°)
R = 1.45;           % Phase winding resistance [ohm]
L = 4e-3;           % Phase winding inductance [H]
kt= 4;              % Motor torque constant [N*m/A]
C = 0.46;           % Detent torque [N*m]
I_r = 6.8e-6;       % Rotor inertia [kg*m^2]
c_r = 0.001;        % Rotor damping [N*m*s/rad]
V = 24;             % DC voltage source [V]

```

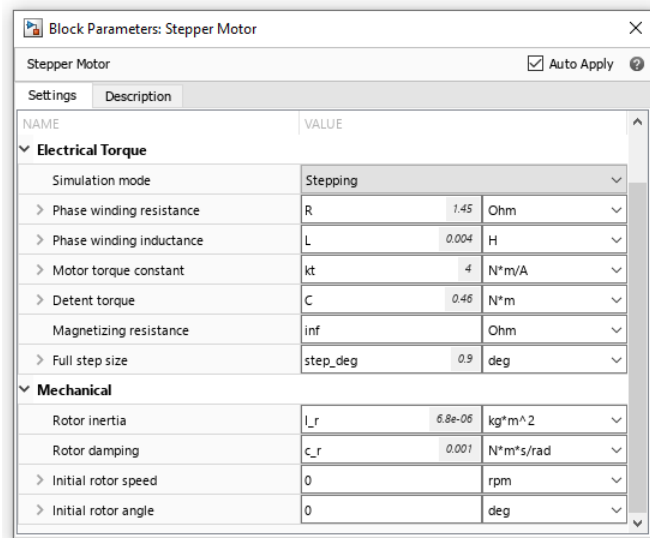


Figura 5.13: Configurazione del blocco Stepper Motor

Si osserva che nella configurazione del blocco Stepper Motor (Figura 5.13) sono stati inseriti anche i parametri Motor Torque Constant e Rotor Damping, non riportati nel catalogo. Di seguito viene descritta la procedura adottata per determinarli.

Il modello del motore è stato inizialmente sviluppato in Simscape R2024b (Figura 5.14), utilizzando il blocco Stepper Motor (STM) appartenente alla libreria Simscape / Electrical / Specialized Power Systems / Electrical Machines. Tale blocco consentiva di modellare sia motori a riluttanza variabile sia motori a magneti permanenti o ibridi.

Nella versione R2024b era richiesto esclusivamente l'inserimento dei parametri forniti dal catalogo (Figura 5.15), poiché la costante di coppia veniva calcolata automaticamente attraverso le equazioni fisiche interne già implementate nel modello.

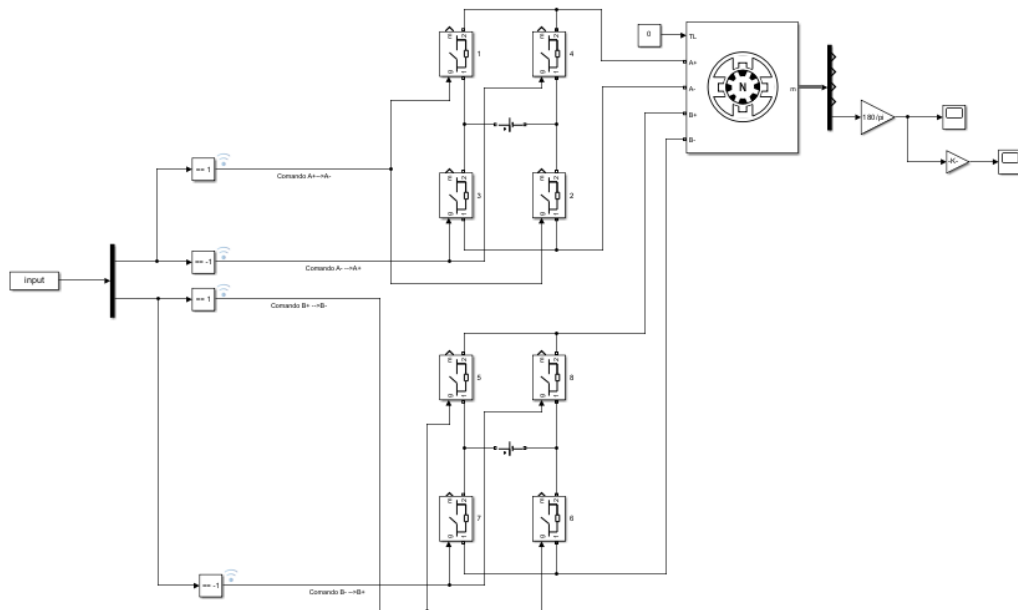


Figura 5.14: Modello del motore inizialmente sviluppato in Simscape 2024

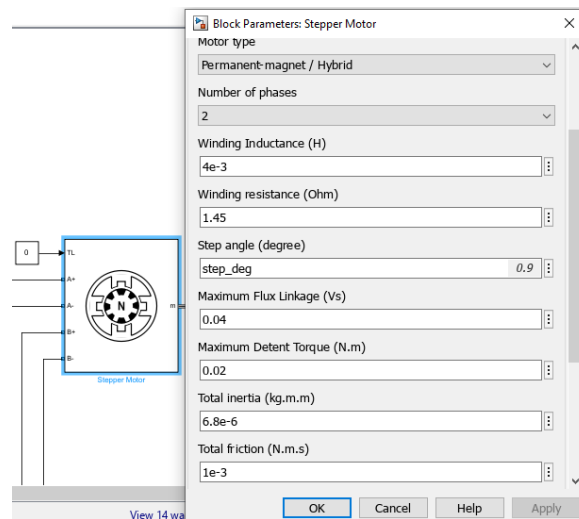


Figura 5.15: Configurazione blocco Stepper Motor (2024)

Con il passaggio a Simscape R2025b, la libreria Specialized Power Systems è stata rimossa ed è stato necessario ricostruire il modello utilizzando i blocchi della nuova libreria Simscape Electrical.

Con il nuovo blocco Stepper Motor, tuttavia, la costante di coppia non viene più determinata automaticamente e deve essere inserita manualmente. La determinazione analitica della costante di coppia per un motore passo-passo è più complessa rispetto a quella di un motore in corrente continua. Essa dipende infatti da diversi fattori costruttivi ed elettromagnetici, tra cui il flusso magnetico, il numero di denti di rotore e statore, la geometria della macchina, l'ampiezza del traferro e le proprietà dei materiali impiegati.

Una stima rigorosa richiederebbe quindi un'analisi elettromeccanica dettagliata del motore, non direttamente realizzabile nell'ambito del presente lavoro.

Sono state comunque applicate diverse relazioni teoriche presenti in letteratura per stimare il valore di K_t , ma i risultati ottenuti non hanno permesso di riprodurre una risposta dinamica

stabile come avveniva nella versione del 2024, evidenziando instabilità o scostamenti rispetto alla risposta attesa.

Per garantire continuità con il modello precedentemente validato, è stato quindi analizzato il blocco Stepper Motor utilizzato in Simscape R2024b, dal quale è stato possibile ricavare il valore della costante di coppia calcolato automaticamente attraverso le sue equazioni interne:

$$K_t = 4 \frac{Nm}{A} \quad (5.9)$$

In modo analogo, anche il coefficiente di smorzamento viscoso del rotore è stato estratto dal modello precedente:

$$B = 0.001 \frac{Nms}{rad} \quad (5.10)$$

5.1.4 Solver

Per evitare errori numerici sono state utilizzate le seguenti impostazioni del solver:

- Type: Variable-step
- Solver: auto (Automatic solver selection)
- Max step size: 1e-3

5.2 Simulazione in modalità full-step

Implementato il modello e impostati i parametri del motore, è ora possibile configurare i blocchi Pulse Generator per avviare la simulazione e rappresentare graficamente lo spostamento angolare del rotore.

In questa sezione si simula lo spostamento in avanti del rotore in modalità full-step, facendo muovere il rotore a 0.9°/s.

I blocchi Pulse Generator sono configurati come segue:

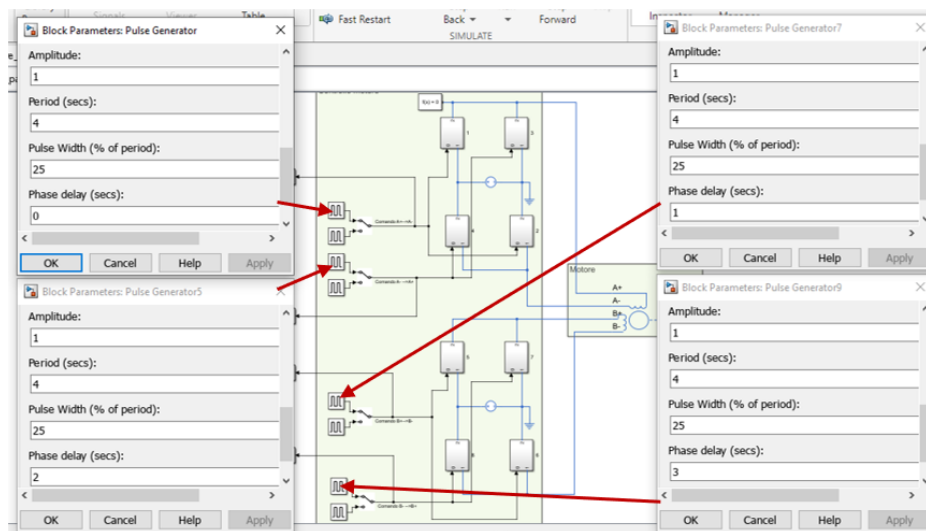


Figura 5.16: Configurazione dei blocchi Pulse Generator in funzionamento full step

I blocchi considerati per la modalità full step sono quelli per cui il Manual Switch è rivolto verso l'alto.

Ogni comando ha un periodo di 4 secondi, con una percentuale del periodo durante la quale il segnale assume valore attivo pari al 25%.

I blocchi sono configurati in modo tale che inizialmente si attivi il comando A , dopo un ritardo di 1 secondo si attivi il comando B , dopo 2 secondi il comando \bar{A} , dopo 3 secondi il comando \bar{B} , e a 4 secondi la sequenza ricomincia.

Per verificare la correttezza della sequenza, si rappresenta graficamente lo stato dei comandi in funzione del tempo.

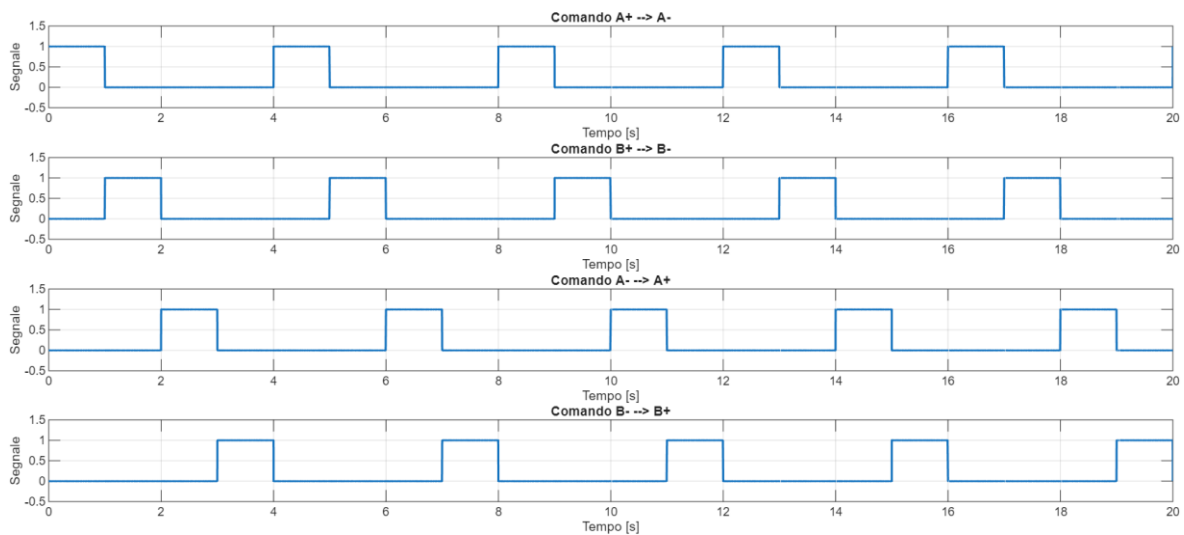


Figura 5.17: Stato dei comandi in funzione del tempo in funzionamento full step

I blocchi Pulse Generator risultano quindi configurati correttamente, e ora è possibile rappresentare graficamente lo spostamento angolare del rotore in funzione del tempo.

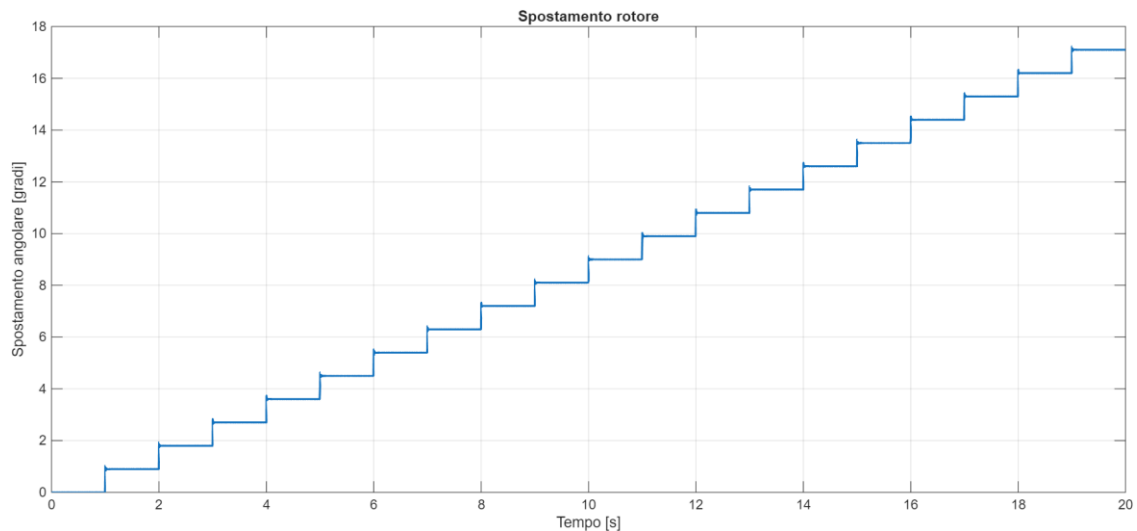


Figura 5.18: Spostamento angolare del rotore in funzione del tempo in funzionamento full step

Il grafico mostra uno spostamento a gradini crescente nel tempo. Ogni gradino corrisponde a un passo angolare del motore, quindi come previsto, il motore si muove in modo discreto e non continuo. Fino al tempo di 1 secondo, il rotore rimane fermo poiché, nella posizione iniziale, è già allineato alla fase A.

Quando viene generata la coppia per spostare il rotore, l'inerzia del rotore fa sì che il movimento non si fermi esattamente alla posizione del gradino, ma il rotore continua leggermente oltre, generando un overshoot. Successivamente, grazie alla coppia di ritorno, il rotore oscilla avanti e indietro con oscillazioni che si smorzano molto rapidamente fino a raggiungere la posizione stabile. Quindi, coppia e inerzia influenzano precisione e tempi di assestamento del rotore.

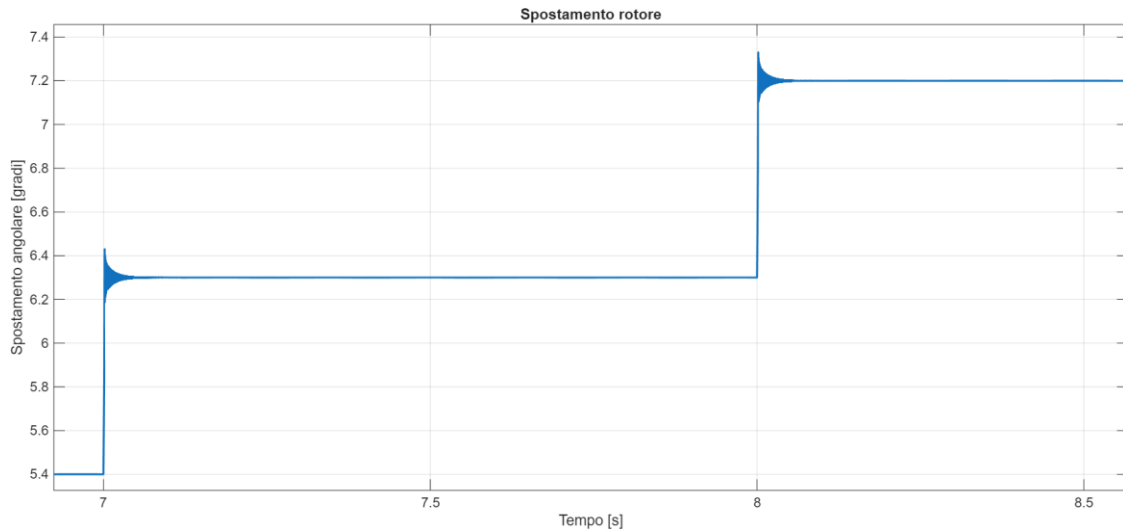


Figura 5.19: Ingrandimento Figura 5.18

5.3 Simulazione in modalità half-step

In questa sezione, si configurano i blocchi Pulse Generator per simulare lo spostamento in avanti del rotore in modalità half-step, facendo muovere il rotore a $0.45^\circ/\text{s}$:

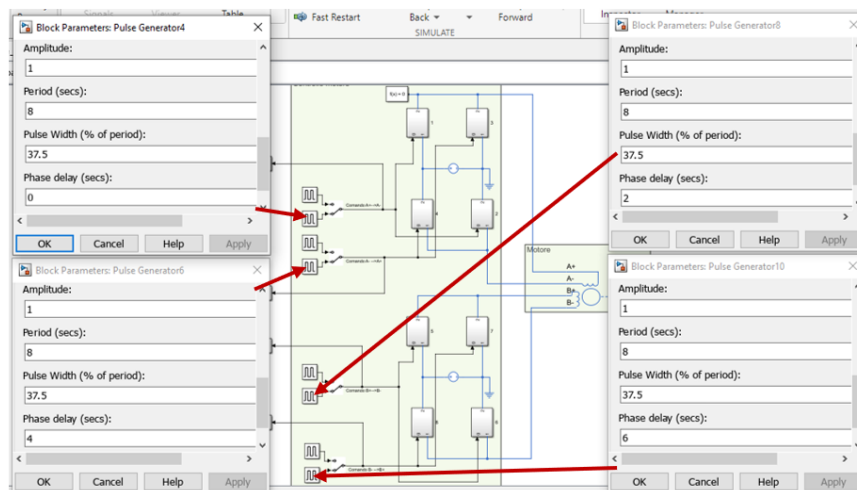


Figura 5.20: Configurazione dei blocchi Pulse Generator in funzionamento half step

I blocchi considerati per la modalità half step sono quelli per cui il Manual Switch è rivolto verso il basso.

Ogni comando ha un periodo di 8 secondi, con percentuale del periodo durante la quale il segnale assume valore attivo del 37.5%.

I blocchi sono configurati in modo tale che inizialmente si attivi il comando A , dopo un ritardo di 2 secondi si attivi il comando B , dopo 4 secondi il comando \bar{A} , dopo 6 secondi il comando \bar{B} , e a 8 secondi la sequenza ricomincia.

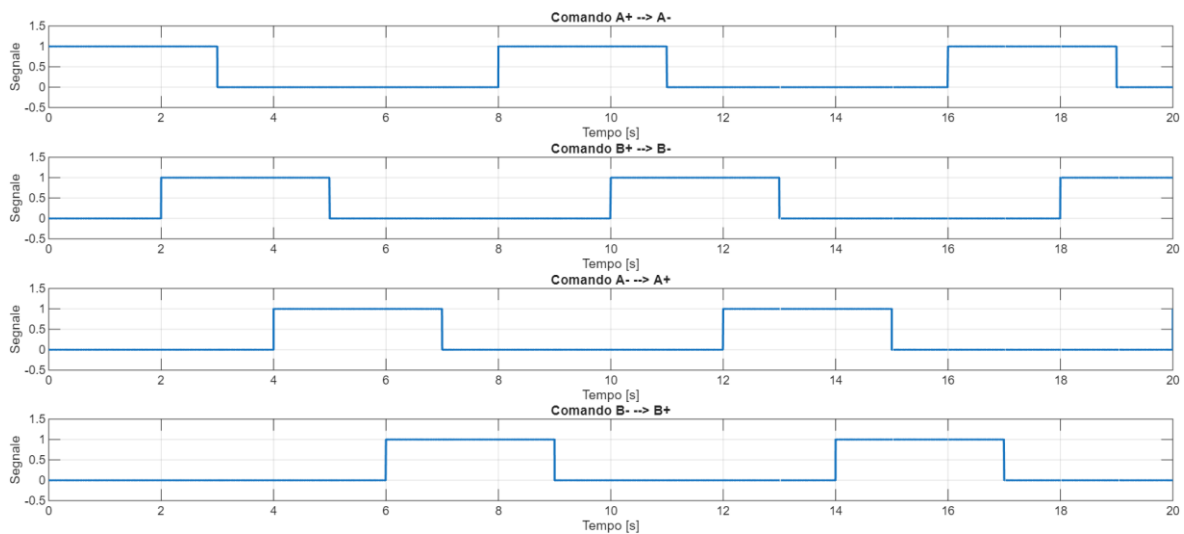


Figura 5.21: Stato dei comandi in funzione del tempo in funzionamento half step

Dalla Figura 5.21 si osserva che i blocchi Pulse Generator sono stati impostati correttamente: all'inizio è attivo solo il comando A ; da 2 a 3 secondi sono attivi contemporaneamente i comandi A e B ; da 3 a 4 secondi è attivo solo B ; da 4 a 5 secondi sono attivi contemporaneamente i comandi B e \bar{A} ; da 5 a 6 secondi è attivo solo \bar{A} ; da 6 a 7 secondi sono attivi contemporaneamente i comandi \bar{A} e \bar{B} ; da 7 a 8 secondi è attivo solo \bar{B} ; da 8 a 9 secondi sono attivi contemporaneamente i comandi \bar{B} e A .

A questo punto è possibile rappresentare graficamente lo spostamento angolare del rotore in funzione del tempo.

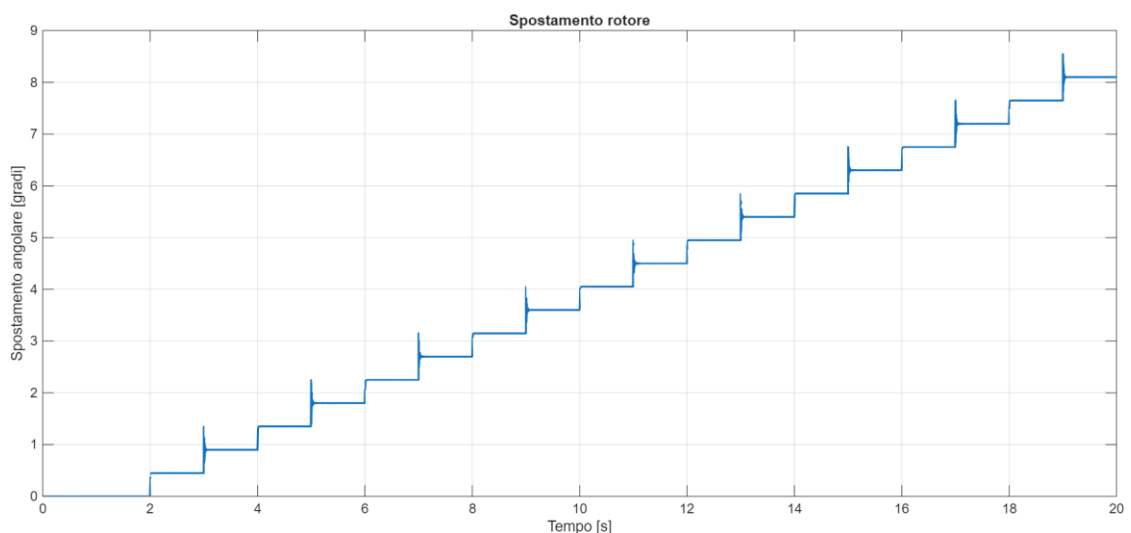


Figura 5.22: Spostamento angolare del rotore in funzione del tempo, in funzionamento half step

Nel funzionamento half step, il motore ibrido alterna l'eccitazione di un solo avvolgimento con quella simultanea di due avvolgimenti. Questa alternanza influisce sulla coppia risultante e sul comportamento dinamico del rotore.

Nei gradini in cui sono eccitati due avvolgimenti, la coppia stabilizzante è maggiore, e il rotore si assesta rapidamente senza overshoot né oscillazioni visibili.

Nei gradini in cui è eccitato un solo avvolgimento, la coppia stabilizzante è inferiore, quindi l'inerzia del rotore produce overshoot e oscillazioni transitorie, che si smorzano velocemente fino al gradino successivo. Questa dinamica spiega l'alternanza osservata nei grafici, con gradini stabili seguiti da gradini in cui il rotore oscilla prima di stabilizzarsi completamente.

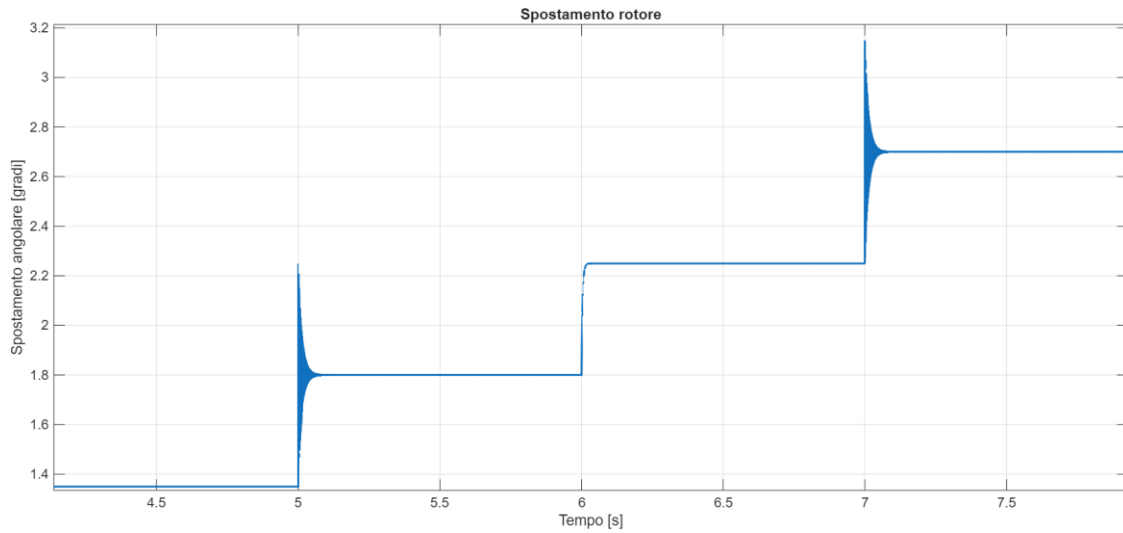


Figura 5.23: Ingrandimento Figura 5.22

6 Modellazione della trasmissione a cinghia

Nel capitolo 2 era stato introdotto un modello semplificato in cui il carrello veniva messo in movimento da una cinghia rappresentata tramite una rigidità e uno smorzatore in parallelo, mentre il moto era imposto direttamente da una sorgente di velocità. Si trattava, chiaramente, di un'idealizzazione utile per isolare alcuni effetti, ma non del tutto rappresentativa del comportamento reale del sistema.

A questo punto si vuole rendere la modellazione della trasmissione a cinghia più realistica, avvicinandola maggiormente a quella effettivamente presente in una stampante FDM. Il sistema considerato è quindi composto da puleggia motrice, cinghia, puleggia di rinvio e carrello.

Successivamente, questo modello della trasmissione verrà posto a valle del modello del motore passo-passo implementato nel paragrafo precedente. In questo modo sarà possibile analizzare l'intero sistema in maniera accoppiata, calcolando lo spostamento del carrello in funzione dello spostamento del motore, tenendo conto della cedevolezza della cinghia, delle forze d'inerzia e dell'attrito.

6.1 Trasmissione a cinghia

Si prende in esame l'asse di una macchina FDM, formato da un carrello che si muove lungo una guida ed è azionato da un motore tramite una trasmissione a cinghia dentata [27]. La cinghia è avvolta su due pulegge di raggio r , poste a una distanza reciproca pari a $2l$. Il carrello possiede massa M , mentre la cinghia è caratterizzata da una sezione A e da un modulo di Young E . Si considera anche il momento di inerzia rotazionale I_m del motore, mentre si trascura l'inerzia della puleggia di rinvio. Per questa descrizione non si considera lo smorzamento della cinghia, che però verrà introdotto nel modello Simscape.

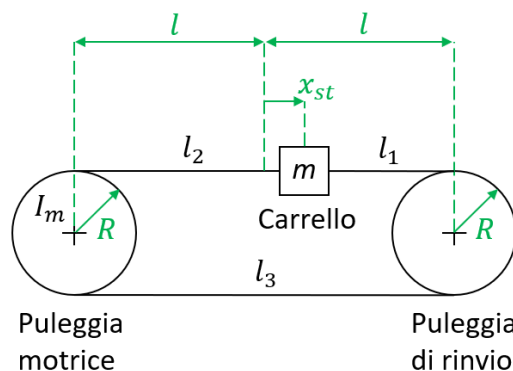


Figura 6.1: Schema geometrico della trasmissione a cinghia

La cinghia può essere divisa in tre rami di lunghezze l_1, l_2, l_3 aventi rigidità k_1, k_2, k_3 , dove k_1 e k_2 non sono costanti ma variano al variare della posizione del carrello poiché cambiano l_1 e l_2 . Nella posizione iniziale il carrello si trova a x_{st} dalla mezzeria. Per semplicità si considerano costanti le rigidità dei tre tratti di cinghia al variare della posizione del carrello, assumendole pari al valore corrispondente alla posizione iniziale x_{st} . Le lunghezze dei vari tratti nella condizione iniziale statica sono:

$$l_1 = l - x_{st} \quad (6.1)$$

$$l_2 = l + x_{st} \quad (6.2)$$

$$l_3 = 2l \quad (6.3)$$

Allontanando gli assi delle pulegge si impone ai tre rami di cinghia una stessa tensione iniziale pari a F_i :

$$T_1 = T_2 = T_3 = F_i \quad (6.4)$$

Noti il modulo di Young, la sezione della cinghia e le lunghezze dei vari tratti, si calcolano le rispettive rigidità mediante l'Eq.(6.5):

$$k_i = \frac{T_i}{\Delta l_i} = \frac{\sigma_i A}{\varepsilon_i l_i} = \frac{EA}{l_i} \quad (6.6)$$

Quindi:

$$k_1 = \frac{EA}{l - x_{st}} \quad (6.7)$$

$$k_2 = \frac{EA}{l + x_{st}} \quad (6.8)$$

$$k_3 = \frac{EA}{2l} \quad (6.9)$$

Siano θ_a, θ_b, x rispettivamente l'angolo di rotazione della puleggia motrice, della puleggia di rinvio e lo spostamento del carrello a partire dalla condizione statica.

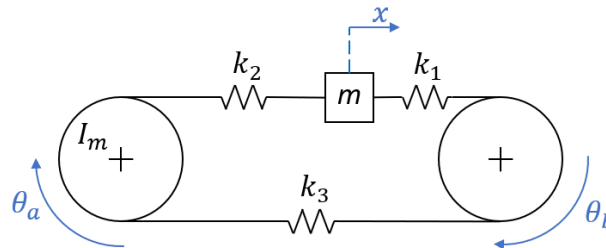


Figura 6.2: Schema cinematico della trasmissione a cinghia

Le tensioni nei tre rami di cinghia sono quindi determinate mediante le seguenti equazioni di equilibrio:

$$T_1 = F_i + k_1(\theta_b r - x) \quad (6.10)$$

$$T_2 = F_i + k_2(x - \theta_a r) \quad (6.11)$$

$$T_3 = F_i + k_3(\theta_a - \theta_b)r \quad (6.12)$$

Di seguito sono riportate le equazioni di equilibrio della puleggia motrice, del carrello e della puleggia di rinvio.

Equilibrio puleggia motrice:

$$C_m = I_m \ddot{\theta}_a + (T_3 - T_2)r = I_m \ddot{\theta}_a + (k_3 + k_2)r^2\theta_a - k_3r^2\theta_b - k_2rx \quad (6.13)$$

Equilibrio carrello:

$$m\ddot{x} = T_1 - T_2 = m\ddot{x} = k_2r\theta_a + K_1\theta_b r - (k_1 + k_2)x \quad (6.14)$$

Equilibrio puleggia di rinvio:

$$T_1 = T_3 \Rightarrow k_3\theta_a r - \theta_b r(k_3 + k_1) + k_1x = 0 \quad (6.15)$$

Sostituendo le tensioni (6.10), (6.11) e (6.12) nelle equazioni di equilibrio (6.13), (6.14) e (6.15), si ottiene un sistema di equazioni nelle incognite θ_a , θ_b , x .

Questa analisi analitica del sistema di trasmissione a cinghia è stata necessaria per comprenderne il funzionamento fisico. A questo punto si può passare alla modellazione della trasmissione in ambiente Simulink/Simscape.

6.2 Parametri del modello

A partire dal motore modello 17HM19-2004S, è stata in seguito scelta una puleggia a 16 denti GT2, con foro da 5 mm compatibile con il diametro dell'albero del motore (Figura 6.3), insieme alla relativa cinghia dentata con passo di 2 mm e larghezza di 6 mm, compatibile con la puleggia (Figura 6.3).



Figura 6.3: Puleggia dentata 16 denti in alluminio, diametro esterno 13 mm, diametro foro 5 mm, larghezza cinghia max. 7 mm, larghezza totale 14 mm [28]

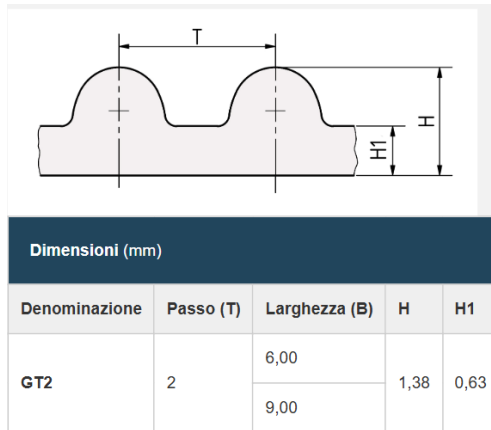


Figura 6.4: Cinghia dentata con profilo GT [29]

Il diametro primitivo della puleggia, la cui circonferenza rappresenta i punti in cui la velocità tangenziale della cinghia coincide con quella della puleggia, è quindi pari a:

$$d_p = \frac{p \cdot z}{\pi} = 10,19 \text{ mm}$$

dove p è il passo dei denti della puleggia e z il numero di denti.

I parametri utilizzati per la modellazione di cinghia e carrello sono i seguenti:

Tabella 6.1: Parametri utilizzati per la modellazione di cinghia e carrello

	Parametro	Valore	u.d.m
Massa del carrello	m	1	kg
Posizione iniziale del carrello	x_0	0	mm
Semi interasse tra le due pulegge	l	0.3	mm
Modulo di Young della cinghia	E	822	Mpa
Larghezza della cinghia	b	6	mm
Altezza della cinghia	h	1.38	mm
Frequenza della sollecitazione dinamica	f	5	Hz
Coefficiente di perdita dovuto alla deformazione della cinghia	η	0.1	-

Per definire lo smorzamento viscoso equivalente dei tre tratti di cinghia si adotta un modello isteretico del materiale, assumendo un coefficiente di perdita (loss factor) η :

$$c_i = \frac{k_i \eta}{\omega}$$

dove $\omega = 2\pi f$ rappresenta la pulsazione della sollecitazione dinamica alla quale viene valutato lo smorzamento viscoso equivalente, con $f = 5 \text{ Hz}$.

Al carrello si aggiunge anche il modello di attrito già discusso, i cui parametri utilizzati sono i seguenti:

Tabella 6.2: Parametri utilizzati per la modellazione dell'attrito

	Parametro	Valore	u.d.m
Coefficiente di attrito statico	μ_s	0.3	-
Coefficiente di attrito dinamico	μ_d	0.15	-
Coefficiente di attrito viscoso guide	c'	10^{-4}	$\frac{Ns}{m}$
Carico normale	N	10	N
Velocità di distacco	v_{brk}	0.005	$\frac{m}{s}$

6.3 Implementazione del modello

Al modello del motore passo passo sviluppato nel paragrafo precedente si aggiunge una puleggia (blocco Wheel and Axle) collegata all'uscita del rotore. Successivamente, all'uscita della puleggia si aggiunge la cinghia con la massa del carrello, modellando ciascun tratto di cinghia come una molla e uno smorzatore in parallelo.

Infine, la massa del carrello viene connessa alla modellazione dell'attrito sulle guide, come già fatto nel paragrafo dedicato al fenomeno di stick-slip.

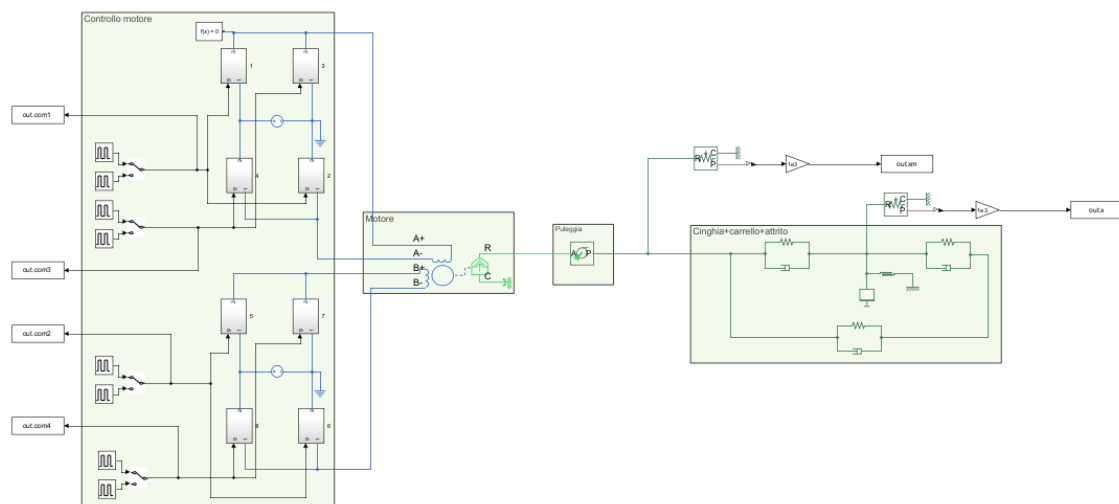


Figura 6.5: Modello Simulink/Simscape del motore passo-passo, del sistema di controllo, della trasmissione a cinghia e dell'attrito

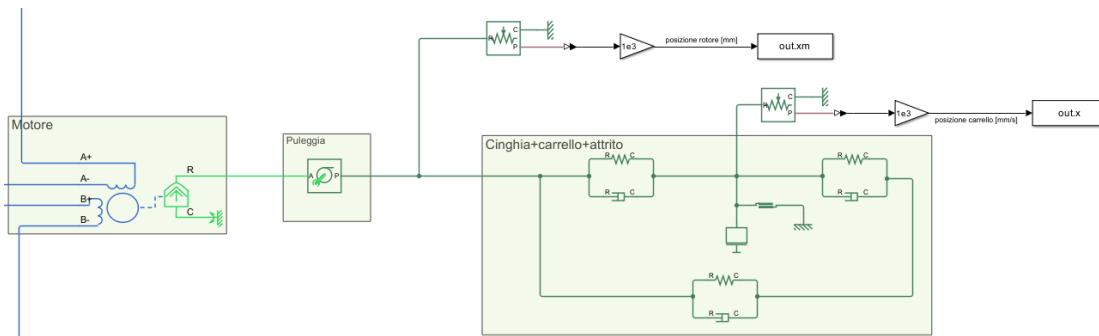


Figura 6.6: Dettaglio della trasmissione a cinghia e dell'attrito riportato in Figura 6.5

Di seguito si riporta lo script MATLAB che definisce i parametri del modello.

```

%% Motore
step_deg = 0.9;           % Step angle (°)
R = 1.45;                 % Phase winding resistance [ohm]
L = 4e-3;                 % Phase winding inductance [H]
kt = 4;                   % Motor torque constant [N*m/A]
C = 0.46;                 % Detent torque [N*m]
I_r = 6.8e-6;             % Rotor inertia [kg*m^2]
c_r = 0.001;              % Rotor damping [N*m*s/rad]
V = 24;                   % DC voltage source [V]

%% Puleggia
p = 2;                    % passo (GT2-16) [mm]
z = 16;                   % denti (GT2-16)
d_p = p * z / pi;         % diametro primitivo (GT2-16) [mm]

%% Cinghia
x0 = 0;                   % posizione iniziale carrello [mm]
l = 0.6/2;                % semi interasse tra le due pulegge [mm]
E = 822;                  % modulo di Young cinghia [MPa]
b = 6;                    % larghezza cinghia [mm]
h = 1.38;                 % altezza cinghia [mm]
A = b*h;                  % sezione cinghia [mm^2]
eta = 0.1;                % coefficiente di perdita dovuto a deformazione
f = 5;                    % frequenza [Hz]
w = 2*pi*f;               % pulsazione [rad/s]

k1 = E*A/(1+x0);          % rigidezza tratto 1
k2 = E*A/(1-x0);          % rigidezza tratto 2
k3 = E*A/(2*l);           % rigidezza tratto 3

c1 = k1*eta/w;            % smorzamento tratto 1
c2 = k2*eta/w;            % smorzamento tratto 2
c3 = k3*eta/w;            % smorzamento tratto 3

m = 1;                    % massa carrello
%% Attrito
mus = 0.3;                % coefficiente di attrito statico
mud = 0.15;               % coefficiente di attrito dinamico
N = 10;                   % carico normale [N]
F_brk = mus*N;            % forza di attrito di distacco [N]
F_Coul = mud*N;           % forza di Coulomb [N]
v_brk = 0.005;            % velocità di distacco [m/s]

```

```
c_1 = 1e-4; % coefficiente di attrito viscoso guide [Ns/m]
```

6.4 Risultati della simulazione

Nel modello sono stati inseriti due sensori di posizione per misurare lo spostamento imposto sulla puleggia motrice, $x_m(t)$, e lo spostamento $x(t)$ del carrello. Per confrontare le differenze tra lo spostamento imposto sulla puleggia motrice e quello effettivo del carrello, vengono rappresentati entrambi nello stesso grafico (Figura 6.8).

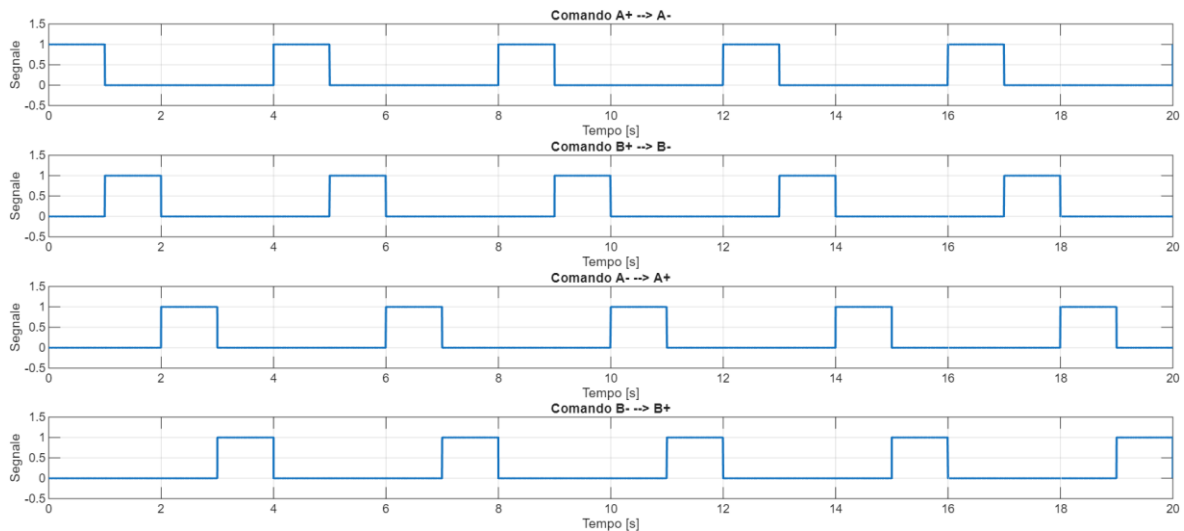


Figura 6.7: Stato dei comandi in funzione del tempo in funzionamento full step

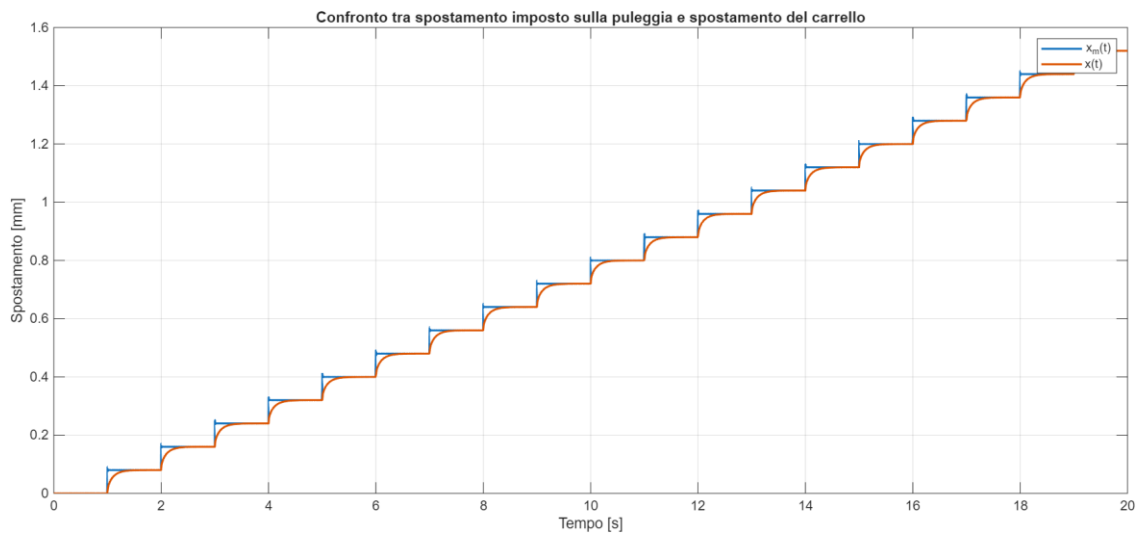


Figura 6.8: Confronto tra lo spostamento imposto sulla puleggia e lo spostamento del carrello

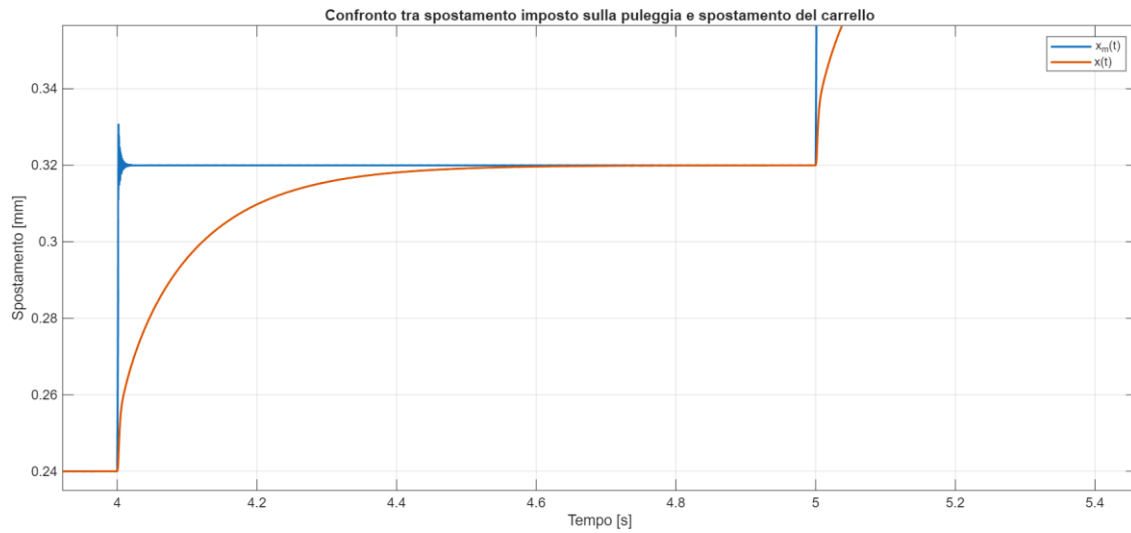


Figura 6.9: Ingrandimento della Figura 6.8

Come previsto, lo spostamento $x(t)$ del carrello risulta più smorzato rispetto a quello imposto sulla puleggia motrice, a causa dell'effetto smorzante introdotto dalla trasmissione a cinghia.

7 Controllo del movimento dell'ugello nel piano XY di una stampante FDM cartesiana

Tutte le analisi e simulazioni svolte nei paragrafi precedenti vengono qui riunite per controllare il movimento dell'ugello nel piano XY di una stampante FDM cartesiana.

Il sistema comprende:

- 1) La generazione della traiettoria in MATLAB;
- 2) La traduzione della traiettoria in G-code;
- 3) La lettura del G-code in MATLAB e la generazione della sequenza delle fasi di attivazione degli avvolgimenti per il motore passo-passo che gestisce l'asse x e per quello che gestisce l'asse y.

Include inoltre, mediante l'utilizzo di Simulink/Simscape:

- 4) Il controllo elettrico dei motori passo-passo e i motori stessi;
- 5) Le pulegge, le cinghie, i carrelli e la modellazione dell'attrito.

Nella sezione 3.2 è stato generato il file G-code (Figura 7.1) relativo a una traiettoria nel piano XY, in cui ogni riga contiene sia la coordinata X sia la coordinata Y del punto da raggiungere.

```
1 G21
2 G90
3 G1 F2400
4 G0 X0.000 Y0.000
5 G1 X0.333 Y0.222
6 G1 X0.666 Y0.444
7 G1 X0.998 Y0.666
8 G1 X1.331 Y0.888
9 G1 X1.664 Y1.109
10 G1 X1.997 Y1.331
11 G1 X2.330 Y1.553
12 G1 X2.663 Y1.775
13 G1 X2.995 Y1.997
14 G1 X3.328 Y2.219
15 G1 X3.661 Y2.441
```

Figura 7.1: Estratto del file G-code riportato in Figura 3.12

Il modello Simscape del motore passo-passo, trasmissione, carrello e attrito, sviluppato nel capitolo 6, è stato modificato introducendo un secondo motore passo-passo per la gestione dell'asse Y della stampante (Figura 7.2). Inoltre, i blocchi Pulse Generator sono stati rimossi e sostituiti con un blocco From Workspace, che permette di importare i segnali di comando generati dallo script MATLAB. (Figura 7.4)

Nel blocco From Workspace denominato "input" (Figura 7.4) è presente il codice MATLAB che, a partire dalla lettura della traiettoria in formato G-code, consente di generare un file .mat contenente i segnali di attivazione degli avvolgimenti da fornire al motore per eseguire i movimenti richiesti.

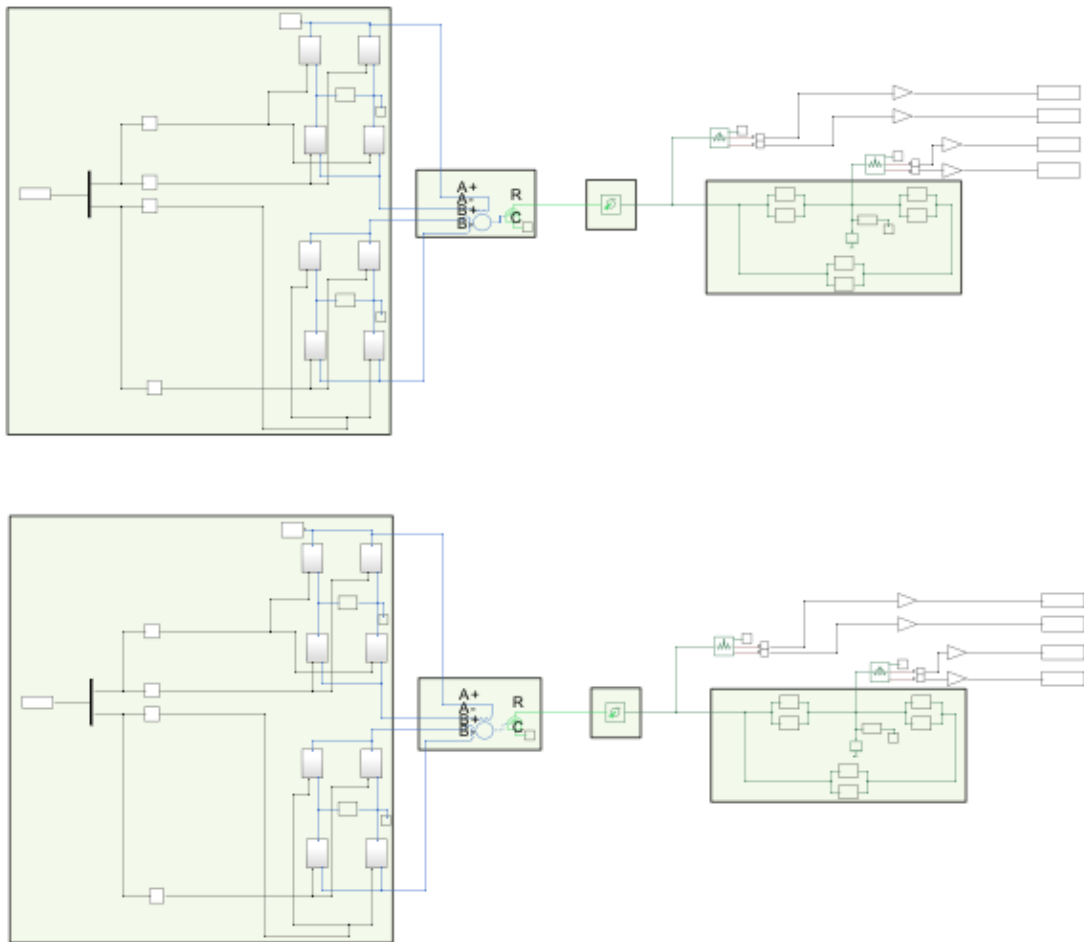


Figura 7.2: Modello Simulink/Simscape aggiornato con l'aggiunta di un secondo motore passo-passo per la gestione dell'asse Y della stampante

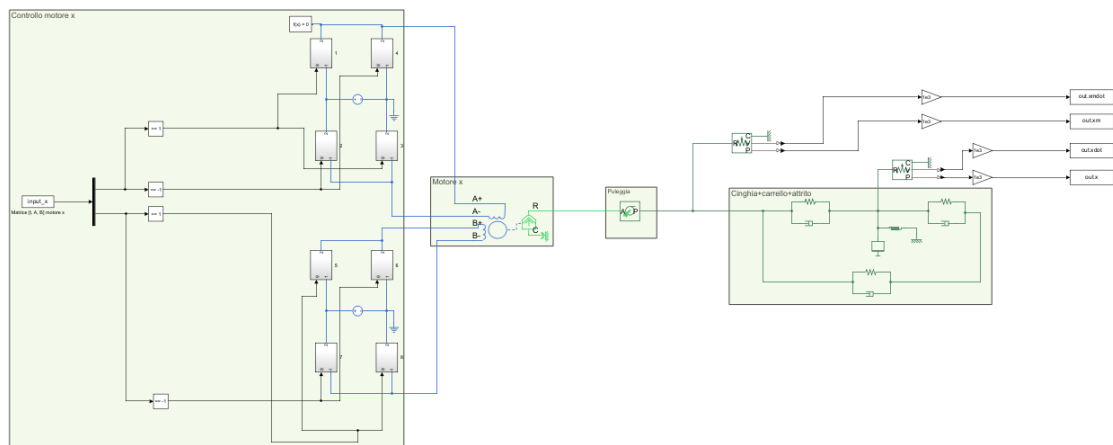


Figura 7.3: Dettaglio del sistema motore e trasmissione che gestisce l'asse x, riportato in Figura 7.2

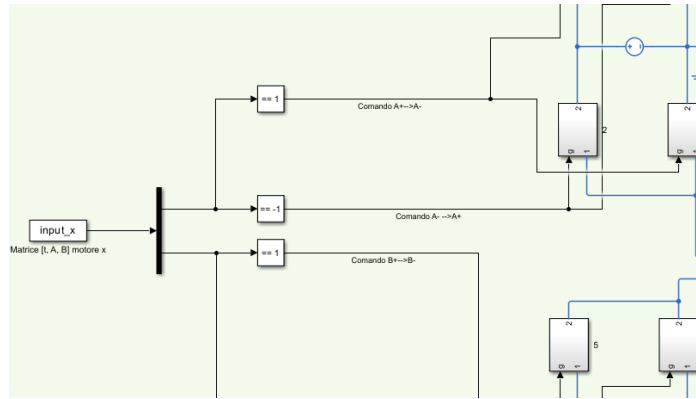


Figura 7.4: Dettaglio del blocco From Workspace, riportato in Figura 7.2

Il codice MATLAB per la generazione delle sequenze di attivazione è stato il lavoro principale e più impegnativo della tesi, perché ha richiesto tempo sia per elaborare la logica e la struttura dello script, sia per approfondire e sperimentare diversi comandi MATLAB necessari a raggiungere l'obiettivo. Per questo motivo, nel paragrafo seguente, il codice viene spiegato nel dettaglio riga per riga.

7.1 Script MATLAB per la generazione delle sequenze di attivazione

Innanzitutto, si definiscono i parametri del motore passo-passo a due fasi, convertendo il passo angolare in spostamento lineare attraverso le caratteristiche della cinghia GT2. Viene calcolato il diametro primitivo della puleggia, la circonferenza e i millimetri spostati per grado, così da determinare lo spostamento lineare per passo del motore, che risulta essere circa 0.08 mm.

```

%% Parametri motore 2 fasi
step_deg = 0.9;           % gradi per passo
p = 2;                   % passo (GT2-16)
z = 16;                  % denti (GT2-16)
d_p = p * z / pi;        % diametro primitivo (GT2-16)
circ = pi * d_p;         % circonferenza primitiva
mm_per_deg = circ / 360; % mm spostati per grado
step_mm = step_deg * mm_per_deg; % mm per passo (0.08 mm)

```

La sequenza bipolare full-step del motore è definita in una matrice, dove ogni riga rappresenta uno stato (comando) delle due fasi A e B.

Il numero di stati della sequenza viene memorizzato nella variabile Nseq.

```

% Sequenza bipolare full-step
fasi = [ 1  0;           % Comando A+ A-
         0  1;           % Comando B+ B-
        -1  0;           % Comando A- A+
         0 -1 ];         % Comando B- B+
Nseq = size(fasi, 1);

```

Successivamente, lo script legge il file G-code: la funzione readlines importa tutte le righe del file in un vettore di stringhe, e subito dopo, con strtrim, vengono rimossi eventuali spazi superflui all'inizio e alla fine di ogni riga.

Infine, si eliminano tutte le righe vuote, in modo che il programma lavori solo con i comandi effettivi.

```
% Lettura G-code da file
righe = readlines('gcode'); % legge tutte le righe
righe = strtrim(righe);     % rimuove spazi iniziali e finali
righe = righe(righe ~= ""); % elimina righe vuote
```

Si imposta lo stato iniziale della simulazione:

- le posizioni reali dei motori sono inizializzate a 0 mm;
- il tempo della simulazione parte da 0 secondi;
- viene definito un indice per tenere traccia della fase attuale del motore, che permette di sapere quale segnale inviare al passo successivo;
- vengono preparati vettori vuoti per salvare l'evoluzione temporale delle fasi e dei tempi corrispondenti.

```
% Inizializzazioni
x_reale = 0;           % posizione effettiva motore x
y_reale = 0;           % posizione effettiva motore y
t = 0;                % tempo simulazione
fase_idx_x = 0;       % indice iniziale di fase motore x
fase_idx_y = 0;       % indice iniziale di fase motore y
tempo = [];           % memoria dei tempi
phases_x = [];        % memoria delle fasi (A, B) motore x
phases_y = [];        % memoria delle fasi (A, B) motore y
```

Viene definita una velocità rapida per i comandi G0 e viene memorizzata l'ultima velocità valida per i comandi G1. La prima serve come velocità fissa quando il G-code contiene un comando G0, che corrisponde a un movimento rapido senza specifica della velocità.

La seconda, `ultima_velocita`, serve a ricordare l'ultima velocità impostata con un comando G1, utile perché nel linguaggio G-code non è sempre necessario riscrivere la velocità a ogni riga: se non viene ripetuta, il programma recupera quella già usata.

```
velocita_rapida = 100; % mm/s per G0
ultima_velocita = [];  % variabile per memorizzare l'ultima velocità G1 valida
```

Per ogni riga del G-code che rappresenta un movimento lineare con velocità definita G1 o spostamento rapido G0, lo script estrae le coordinate X e Y e la velocità, calcola la distanza dal target e la suddivide nel numero di passi necessari.

La funzione `regexp` serve a cercare un testo all'interno di una stringa. In questo caso l'obiettivo è trovare il valore della velocità indicata con la lettera F nel comando G-code.

La parte tra virgolette `'F([\d\.\-]+)'` è la regola di ricerca: significa "cerca una F seguita da una sequenza di numeri, che può contenere anche punti decimali e segni meno".

Le parentesi tonde indicano che quella parte deve essere memorizzata come risultato, mentre l'opzione `'tokens'` fa sì che la funzione restituisca proprio il valore numerico trovato, senza la lettera F.

Se viene individuato un numero, questo viene trasformato da stringa a numero con `str2double` e diviso per 60, passando così da millimetri al minuto a millimetri al secondo. Inoltre, il valore viene salvato nella variabile `ultima_velocita` per poterlo riutilizzare.

Se invece la velocità non è scritta e non ne era mai stata salvata una prima, il codice mostra un avviso e salta quel comando.

Se invece una velocità era già stata memorizzata, il codice la riutilizza senza problemi.

Infine, se il comando è `G0`, il movimento viene eseguito a velocità rapida: la variabile `velocita` viene quindi impostata al valore predefinito `velocita_rapida`.

In pratica significa che il carrello si sposta velocemente da un punto all'altro senza preoccuparsi della precisione di lavorazione.

```
% Elaborazione G-code
for i = 1:length(righe)
    riga = righe(i);

    if startsWith(riga, 'G1') || startsWith(riga, 'G0')
        % Estrazione velocità
        if startsWith(riga, 'G1')
            v_estratta = regexp(riga, 'F([\d\.\-]+)', 'tokens');
            if ~isempty(v_estratta)
                velocita = str2double(v_estratta{1}) / 60; % mm/s
                ultima_velocita = velocita; % aggiorna la memoria
            else
                if isempty(ultima_velocita)
                    warning('Comando G1 senza velocità F e nessuna velocità memorizzata:
%s', riga);
                    continue; % salta il comando se nessuna velocità precedente
                else
                    velocita = ultima_velocita; % usa ultima velocità valida
                end
            end
        else
            % G0 movimento rapido
            velocita = velocita_rapida;
        end
    end
end
```

Qui viene fatto un lavoro simile a quello della velocità, ma per le coordinate. In particolare, la prima riga cerca se il comando contiene un valore per X, la seconda se contiene un valore per Y.

Anche qui si usa `regexp`: la regola `'X([\d\.\-]+)'` indica di trovare la lettera X seguita da una sequenza di numeri che può contenere punto e segno.

In pratica, questa funzione estrae numeri anche con decimali e segno negativo, ad esempio:

- F300.5 estrae 300.5
- X-12.3 estrae -12.3
- Y15 estrae 15

Successivamente, il programma determina le coordinate di destinazione per i motori X e Y a partire dalle istruzioni del G-code.

Per prima cosa verifica se nella riga corrente del G-code è presente un valore per la coordinata x usando `if ~isempty(x_estratta)`. La funzione `isempty` controlla se una variabile è vuota, e il simbolo `~` nega il risultato, quindi questa condizione significa “se `x_estratta` non è vuota”. Se c’è un valore, `str2double(x_estratta{1})` lo trasforma da stringa a numero. Le parentesi graffe `{1}` servono a estrarre il contenuto del primo elemento di una cella, perché `regexp`, usato in precedenza per cercare il numero nella riga, restituisce sempre i risultati come celle.

Se invece non c’è nessun valore X, il motore mantiene la sua posizione attuale, quindi `x_target` viene impostato uguale a `x_reale`.

Lo stesso procedimento viene eseguito per la coordinata Y: se `y_estratta` contiene un valore, viene convertito in numero con `str2double`, altrimenti `y_target` assume il valore attuale `y_reale`.

```
% Estrazione coordinate X e Y
x_estratta = regexp(riga, 'X([\d\.\-]+)', 'tokens');
y_estratta = regexp(riga, 'Y([\d\.\-]+)', 'tokens');

if ~isempty(x_estratta)
    x_target = str2double(x_estratta{1});
else
    x_target = x_reale;
end
if ~isempty(y_estratta)
    y_target = str2double(y_estratta{1});
else
    y_target = y_reale;
end
```

Una volta note le coordinate target per X e Y, il programma calcola quanto ciascun motore deve muoversi per raggiungere la nuova posizione.

La differenza $dx = x_target - x_reale$ indica lo spostamento necessario lungo l’asse X, mentre $dy = y_target - y_reale$ indica lo spostamento lungo l’asse Y.

Questi valori vengono poi utilizzati per determinare la direzione del movimento (positiva o negativa) sia il numero di passi necessari per ciascun motore. A questo punto il codice non considera più un movimento lungo un singolo asse, ma calcola la distanza reale da percorrere nel piano, cioè la lunghezza del segmento che unisce il punto di partenza a quello di arrivo nel piano X-Y.

Se la lunghezza `L` risulta praticamente nulla, significa che la posizione attuale coincide già con quella richiesta, quindi il ciclo viene saltato e non viene eseguito alcun movimento.

```
% Calcolo distanze dal target e direzioni
dx = x_target - x_reale;
dy = y_target - y_reale;
L = sqrt(dx^2 + dy^2); % lunghezza spostamento

if L < 1e-9
    continue; % niente movimento
end
```

Successivamente si determina il numero di “passi virtuali” lungo la traiettoria L.

Il numero di step Nsteps viene calcolato come la lunghezza della traiettoria divisa per lo spostamento che il motore può fare in un passo fisico (step_mm).

Il comando max(...,1) garantisce che venga eseguito almeno un passo virtuale, anche nel caso in cui la distanza sia molto piccola.

È importante chiarire che questo numero serve esclusivamente a suddividere la traiettoria L in segmenti virtuali per calcolare la sequenza dei passi dei motori, ma non significa che i motori si muoveranno di frazioni di passo. I motori avanzano solo quando la posizione accumulata supera step_mm, garantendo che ogni passo sia sempre fisicamente corretto.

A questo punto si calcola quanto avanzare lungo X e Y in ciascun passo virtuale: step_dx = dx / Nsteps e step_dy = dy / Nsteps.

Questi valori servono a creare posizioni intermedie lungo la traiettoria, permettendo al codice di verificare quando i motori devono realmente fare un passo fisico. Viene inoltre calcolato il tempo associato a ciascun passo virtuale ($T_s = L / (\text{velocita} * \text{Nsteps})$), ossia il tempo che deve trascorrere per rispettare la velocità impostata.

Quindi, il codice trasforma una traiettoria continua in una serie di passi discreti, rispettando sia la distanza minima fisica dei motori (step_mm) sia la velocità impostata, senza mai muovere i motori di frazioni di passo.

La suddivisione della traiettoria in segmenti serve solo a calcolare quando ciascun motore deve fare effettivamente un passo, garantendo precisione e continuità nel movimento.

```
Nsteps = max(round(L / step_mm),1);
step_dx = dx / Nsteps;
step_dy = dy / Nsteps;
Ts = L / (velocita * Nsteps);
```

Il ciclo for k = 1:Nsteps serve a ripetere il calcolo dei passi dei motori per tutta la traiettoria, da una posizione iniziale a una finale.

La variabile k indica semplicemente quante iterazioni fare, cioè quante "frazioni di movimento" calcolare, anche se non viene utilizzata all'interno del ciclo. Ogni ciclo rappresenta un avanzamento lungo la traiettoria.

In questo ciclo vengono calcolate le nuove posizioni continue x_new e y_new. Queste posizioni non sono ancora passi fisici del motore, ma valori continui lungo X e Y, ottenuti sommando alla posizione reale corrente x_reale e y_reale i piccoli incrementi step_dx e step_dy calcolati in precedenza.

Questo permette di suddividere la distanza totale in tanti piccoli avanzamenti.

```
for k = 1:Nsteps
    %nuova posizione
    x_new = x_reale + step_dx;
    y_new = y_reale + step_dy;
```

Confrontando round(x_new/step_mm) con round(x_reale/step_mm) il codice verifica se la nuova posizione ha superato almeno un passo completo rispetto a quella corrente. Se il valore arrotondato cambia, significa che il motore X deve effettivamente muoversi.

In particolare:

- $step_mm$ è la distanza fisica percorsa con un singolo passo;
- $x_new/step_mm$ rappresenta il numero teorico di passi necessari per raggiungere x_new ;
- $round(...)$ arrotonda questo numero al numero intero più vicino, perché il motore può muoversi solo in passi completi.

Questo significa che, anche se la posizione continua aumenta di piccole frazioni, il motore avanza solo quando si raggiunge un passo fisico completo.

Di seguito sono riportati alcuni esempi utili a chiarire questa parte di codice.

Esempio 1: movimento in avanti

Si suppone $step_mm = 0.08$ mm, $x_reale = 0$, $step_dx = 0.02$ mm.

Tabella 7.1: Esempio movimento in avanti con $step_mm = 0.08$ mm, $x_reale = 0$, $step_dx = 0.02$ mm

x_new	$x_new/step_mm$	$round(x_new/step_mm)$	Avanza passo?
0.02	0.25	0	No
0.04	0.5	1	Sì
0.06	0.75	1	No
0.08	1	1	No
0.1	1.25	1	No
0.12	1.5	2	Sì

La posizione x_new aumenta progressivamente, ma il rotore rimane fermo finché la somma degli avanzamenti non supera un passo completo pari a 0.08 mm. Solo nel momento in cui il valore intero $round(x_new/step_mm)$ varia, il rotore compie un nuovo passo.

Esempio 2: movimento negativo

Si suppone $step_mm = 0.08$ mm, $x_reale = 0.16$ mm, $step_dx = -0.02$ mm.

Tabella 7.2: Esempio movimento negativo con $step_mm = 0.08$ mm, $x_reale = 0.16$ mm, $step_dx = -0.02$ mm

x_new	$x_new/step_mm$	$round(x_new/step_mm)$	Avanza passo?
0.14	1.75	2	No
0.12	1.5	2	No
0.10	1.25	1	Sì
0.08	1	1	No
0.06	0.75	1	No
0.04	0.5	1	No
0.02	0.25	0	Sì

La posizione x_{new} diminuisce progressivamente, ma, analogamente al caso del moto positivo, il rotore rimane fermo finché la somma del modulo degli avanzamenti non supera un passo completo pari a 0.08 mm. Solo nel momento in cui il valore intero $\text{round}(x_{new}/\text{step_mm})$ varia, il rotore compie un nuovo passo.

```

    % Avanza X se serve
    if round(x_new/step_mm) ~= round(x_reale/step_mm)

```

Se il motore deve muoversi, $\text{dir_x} = \text{sign}(\text{step_dx})$ determina la direzione del passo, mentre fase_idx_x aggiorna l'indice della fase corrente.

L'indice della fase da attivare viene aggiornato incrementandolo o decrementandolo di uno in funzione della direzione del movimento: la variabile dir vale +1 se il motore deve avanzare e -1 se deve indietreggiare. In questo modo è possibile scorrere in avanti o all'indietro nella sequenza delle fasi del motore.

L'istruzione $\text{mod}(\text{fase_idx_x}, \text{Nseq}) + 1$ serve invece a scorrere tra le fasi disponibili.

In generale, $b = \text{mod}(a,m)$ restituisce il resto della divisione di a per m , dove a è il dividendo e m è il divisore.

Questa operazione, chiamata modulo, può essere espressa come $b = a - m \cdot \text{floor}(a./m)$.

La funzione mod segue la convenzione per cui $\text{mod}(a,0)$ restituisce a .

La funzione $Y = \text{floor}(X)$ arrotonda ciascun elemento di X all'intero minore o uguale più vicino.

Si calcola quindi l'indice effettivo da utilizzare per accedere alla matrice delle fasi mediante la funzione mod , che restituisce il resto della divisione tra fase_idx e Nseq , assicurando che il risultato sia sempre compreso tra 0 e $\text{Nseq}-1$. Aggiungendo 1, si ottiene un valore tra 1 e Nseq , coerente con l'indicizzazione di MATLAB, che parte da 1.

Esempio:

Tabella 7.3: Esempio istruzione $\text{mod}(\text{fase_idx_x}, \text{Nseq}) + 1$

Passo	dir	fase_idx	mod(fase_idx, 4)	mod + 1	Riga scelta	fasi(...)
1	1	0	0	1	fasi(1,:)	[1 0]
2	1	1	1	2	fasi(2,:)	[0 1]
3	-1	0	0	1	fasi(1,:)	[1 0] indietro
4	1	1	1	2	fasi(2,:)	[0 1]
5	1	2	2	3	fasi(3,:)	[-1 0]
6	1	3	3	4	fasi(4,:)	[0 -1]
7	1	4	0	1	fasi(1,:)	[1 0]
8	1	5	1	2	fasi(2,:)	[0 1]
9	1	6	2	3	fasi(3,:)	[-1 0]
10	-1	5	1	2	fasi(2,:)	[0 1] indietro

L'istruzione $\text{fase_x} = \text{fasi}(\text{idx_mod_x}, :)$ seleziona quindi la fase da inviare alle bobine del motore.

Se invece non serve muovere l'asse X, il motore mantiene la fase precedente per non cambiare la configurazione.

```
dir_x = sign(step_dx);
fase_idx_x = fase_idx_x + dir_x;
idx_mod_x = mod(fase_idx_x, Nseq) + 1;
fase_x = fasi(idx_mod_x, :);
else
if isempty(phases_x)
fase_x = [0 0];
else
fase_x = phases_x(end,:);
end
end
```

La parte successiva dello script è analoga a quella precedente, ma gestisce il motore Y. La logica di aggiornamento della fase, il calcolo dell'indice tramite operazione modulo e la selezione della configurazione da inviare alle bobine sono analoghi a quanto descritto per il motore X.

```
% Avanza Y se serve
if round(y_new/step_mm) ~= round(y_reale/step_mm)
dir_y = sign(step_dy);
fase_idx_y = fase_idx_y + dir_y;
idx_mod_y = mod(fase_idx_y, Nseq) + 1;
fase_y = fasi(idx_mod_y, :);
else
if isempty(phases_y)
fase_y = [0 0];
else
fase_y = phases_y(end,:);
end
end
```

Dopo aver verificato e aggiornato eventuali passi dei motori, le posizioni reali vengono aggiornate con i valori continui appena calcolati. Queste posizioni costituiscono la base per il passo successivo del ciclo, permettendo al motore di avanzare lungo la traiettoria corretta. Successivamente, viene aggiornato il tempo totale t sommando l'intervallo T_s calcolato per rispettare la velocità desiderata.

Vengono quindi registrati il tempo corrente e le fasi dei motori X e Y nelle variabili tempo, phases_x e phases_y. In questo modo si costruisce la sequenza temporizzata completa dei segnali da inviare ai motori.

```
% Aggiorna posizioni reali
x_reale = x_new;
y_reale = y_new;

% Aggiorna tempo e fasi
t = t + Ts;
tempo(end+1,1) = t;
phases_x(end+1,:) = fase_x;
phases_y(end+1,:) = fase_y;
```

```
end
```

Lo script verifica anche se la riga contiene un comando G21 (unità in millimetri) o G90 (coordinate assolute).

Poiché il programma lavora già assumendo millimetri e coordinate assolute, questi comandi vengono riconosciuti ma ignorati. mentre eventuali comandi non riconosciuti generano un avviso.

```
elseif startswith(riga, 'G21') || startswith(riga, 'G90')
    % Ignora questi comandi
else
    warning('Comando G-code non riconosciuto: %s', riga);
end
end
```

Alla fine dello script vengono create due matrici di output, `input_x` e `input_y`, che contengono il tempo e lo stato delle fasi dei motori per ciascun passo della simulazione.

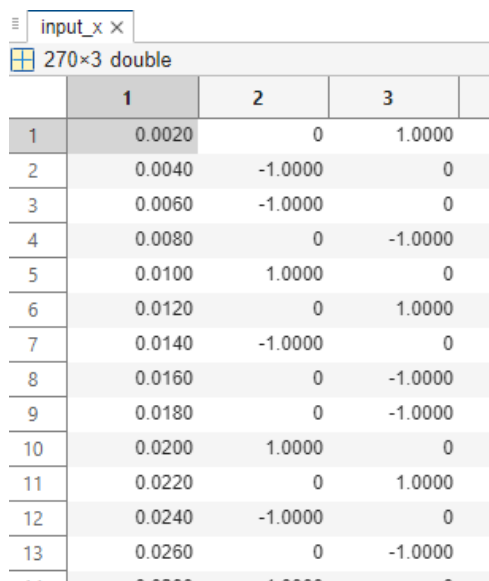
In queste matrici, ogni riga rappresenta un passo temporale e le colonne indicano rispettivamente: il tempo, lo stato della fase A e lo stato della fase B.

```
% Output per Simulink
input_x = [tempo, phases_x];
input_y = [tempo, phases_y];
```

Infine, queste matrici vengono salvate in file `.mat` e sono pronte per essere utilizzate come input in Simulink, consentendo di simulare correttamente il comportamento dei motori.

```
save('stepper_input_x.mat', 'input_x');
save('stepper_input_y.mat', 'input_y');
```

In Figura 7.5 si riporta una parte della matrice “input.x”



	1	2	3
1	0.0020	0	1.0000
2	0.0040	-1.0000	0
3	0.0060	-1.0000	0
4	0.0080	0	-1.0000
5	0.0100	1.0000	0
6	0.0120	0	1.0000
7	0.0140	-1.0000	0
8	0.0160	0	-1.0000
9	0.0180	0	-1.0000
10	0.0200	1.0000	0
11	0.0220	0	1.0000
12	0.0240	-1.0000	0
13	0.0260	0	-1.0000

Figura 7.5: Estratto della matrice “input.x”

Ogni riga può avere il seguente significato:

- [t 1 0] → Comando fase A
- [t 0 1] → Comando fase B
- [t -1 0] → Comando fase \bar{A}
- [t 0 -1] → Comando fase \bar{B}

7.2 Script completo

Al fine di poter svolgere la simulazione, è necessario unire in un unico script MATLAB i seguenti codici, nel seguente ordine:

- Lo script per generare il G-code della stessa traiettoria modellata nella sezione 3.2;
- Lo script per generare le sequenze di attivazione degli avvolgimenti, precedentemente discusso;
- Lo script che definisce i parametri del motore, della cinghia, del carrello e dell'attrito necessari ai vari blocchi di Simscape, come già descritto nella sezione 6.3.

```
%% Raccordo di Bézier parabolico fra due segmenti
%Inizializzazione
clc
clear all
close all

%% Coordinate punti poligono di controllo
P0= [0 0]';
P1= [15 10]';
P2= [15 5]';
delta = 2;      % raggio raccordo, mm
dt = 0.01;
v = 40;        % mm/s

%% Calcolo intersezioni con circonferenza di raggio delta
%con relazioni trigonometriche
alfa = atan((P1(2)-P0(2))/(P1(1)-P0(1)));
p1i = P1 - delta*[cos(alfa) sin(alfa)]';
p1u = P1 - [0 delta]';

%%Plottaggio punti di inizio e fine dei 3 tratti
figure(1)
plot(P0(1),P0(2), 'ro',P1(1),P1(2), 'ko',P2(1),P2(2), 'ro',...
      p1i(1),p1i(2), 'r*',p1u(1),p1u(2), 'r*', 'linewidth',2)
axis equal
xlabel('asse x (mm)')
ylabel('asse y (mm)')
grid on
hold on

%% Tratto di ingresso rettilineo
k = 1;
u(k) = 0;

%ciclo su k per u<1
while u(k)<1
    l_1 = norm(p1i-P0); % lunghezza primo tratto rettilineo
    du=v/l_1*dt;      % incremento du
    u(k+1) = u(k)+du;
```

```

        k=k+1;
end

u = u(u <=1) ;          % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_01i = P0*(1-u) + p1i*u; %Bezier primo tratto lineare

%% Raccordo parabolico
% Implementare un ciclo iterativo per il calcolo di u(k+1) a partire da u(k)
u=[]; % reinizializzo parametro u
dpsudu = []; % reinizializzo dp/du

k = 1; % reinizializzo contatore

dpsudu0 = 2*(P1-p1i) ; % reinizializzo dp/du
u(k)=(dt-norm(p1i-p_01i(:,end))/v)*(v/norm(dpsudu0)); % reinizializzo parametro u

%ciclo su k per u<1
while u(k)<1
    dpsudu = -2*(1-u(k))*p1i+2*(1-2*u(k))*P1+2*u(k)*p1u;
    du = v/norm(dpsudu)*dt;
    u(k+1) = u(k) + du;
    k=k+1;
end

u = u(u <=1) ; % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_rac = (1-u).^2.*p1i + 2* (1-u).*u.*P1+u.^2.*p1u; %Bezier di raccordo

%% Tratto di uscita rettilineo
%Implementare un ciclo iterativo per il calcolo di u(k+1) a partire da u(k)

u = []; % reinizializzo parametro u
dpsudu = []; % reinizializzo dp/du

k = 1; % reinizializzo contatore

dpsudu0 = 2*(p1u-P1) ; % reinizializzo dp/du
u(k)=(dt-norm(p1u-p_rac(:,end))/v)*(v/norm(dpsudu0)); % reinizializzo parametro u

while u(k)<1
    l_2 = norm(P2-p1u); % lunghezza secondo tratto rettilineo
    du=v/l_2*dt; % incremento du
    u(k+1) = u(k)+du;
    k=k+1;
end

u = u(u <=1) ;          % QUALI VALORI NON MI SERVONO?

%Calcolo coordinate dei punti in funzione del vettore u
p_1u2 = p1u*(1-u) + P2*u; %Bezier secondo tratto lineare

%% Assemlo i 3 tratti

p = [p_01i p_rac p_1u2];
px = p(1,:); %prima riga, componenti x
py = p(2,:); %seconda riga, componenti y

figure(1)
plot(px,py,'bo','linewidth',1.5)
xlabel('x [mm]')
ylabel('y [mm]')
title('Spostamento nel piano XY')

```

```

grid on

%% Vettore tempo
N = size(p,2);
t = zeros(1,N);

for k = 2:N
    %ds = norm(p(:,k) - p(:,k-1));
    ds = sqrt((p(1,k) - p(1,k-1))^2 + (p(2,k) - p(2,k-1))^2); % distanza tra punti
    consecutivi
    t(k) = t(k-1) + ds/v;
end
time=t;

%% Plot x(t) e y(t)
figure (2)
subplot(2,1,1)
plot(time, px, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('x [mm]')
title('Spostamento x(t)')
grid on

subplot(2,1,2)
plot(time, py, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('y [mm]')
title('Spostamento y(t)')
grid on

%% Velocità di stampa

dx = diff(px);      % differenze di posizione
dt = diff(time);   % differenze di tempo
vx = dx./dt;       % velocità media tra punti
t_v = time(1:end-1); % istanti associati alla velocità

figure (3)
subplot(2,1,1)
plot(t_v, vx, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('v_x [mm/s]')
title('Velocità v_x(t)')
grid on

dy = diff(py);      % differenze di posizione
dt = diff(t);       % differenze di tempo
vy = dy./dt;       % velocità media tra punti
t_v = t(1:end-1);  % istanti associati alla velocità

figure (3)
subplot(2,1,2)
plot(t_v, vy, 'b-', 'LineWidth', 1.5)
xlabel('Tempo [s]')
ylabel('v_y [mm/s]')
title('Velocità v_y(t)')
grid on

% Calcolo del modulo della velocità
v = sqrt(vx.^2 + vy.^2);

figure (4)
plot(t_v, v, 'b-', 'LineWidth', 2)
xlabel('Tempo [s]')
ylabel('Velocità nel piano [mm/s]')
title('Modulo della velocità nel piano XY')

```

```

grid on

%% Generazione Gcode
gcodeFile = fopen('gcode', 'w');

fprintf(gcodeFile, 'G21 \n');      % usa mm
fprintf(gcodeFile, 'G90 \n');      % coordinate assolute
fprintf(gcodeFile, 'G1 F2400 \n'); % velocità 40 mm/s (in mm/min)
fprintf(gcodeFile, 'G0 X%.3f Y%.3f\n', px(1), py(1)); % movimento rapido

for i = 2:length(px)
    fprintf(gcodeFile, 'G1 X%.3f Y%.3f\n', px(i), py(i));
end

fclose(gcodeFile);

%% Parametri motore 2 fasi
step_deg = 0.9;                    % gradi per passo
p = 2;                              % passo (GT2-16)
z = 16;                              % denti (GT2-16)
d_p = p * z / pi;                  % diametro primitivo (GT2-16)
circ = pi * d_p;                   % circonferenza primitiva
mm_per_deg = circ / 360;           % mm spostati per grado
step_mm = step_deg * mm_per_deg;   % mm per passo (0.08 mm)

% Sequenza bipolare full-step
fasi = [ 1  0;
        0  1;
       -1  0;
        0 -1 ];
Nseq = size(fasi, 1);

% Lettura G-code da file
righe = readlines('gcode'); % legge tutte le righe
righe = strtrim(righe);      % rimuove spazi iniziali e finali
righe = righe(righe ~= ""); % elimina righe vuote

% Inizializzazioni
x_reale = 0;                    % posizione effettiva motore x
y_reale = 0;                    % posizione effettiva motore y
t = 0;                          % tempo simulazione
fase_idx_x = 0;                 % indice iniziale di fase motore x
fase_idx_y = 0;                 % indice iniziale di fase motore y
tempo = [];                     % memoria dei tempi
phases_x = [];                 % memoria delle fasi (A, B) motore x
phases_y = [];                 % memoria delle fasi (A, B) motore y

velocita_rapida = 100;          % mm/s per G0
ultima_velocita = [];          % variabile per memorizzare l'ultima velocità G1 valida

% Elaborazione G-code
for i = 1:length(righe)
    riga = righe(i);

    if startsWith(riga, 'G1') || startsWith(riga, 'G0')
        % Estrazione velocità
        if startsWith(riga, 'G1')
            v_estratta = regexp(riga, 'F([\d\.\\-]+)', 'tokens');
            if ~isempty(v_estratta)
                velocita = str2double(v_estratta{1}) / 60; % mm/s
                ultima_velocita = velocita;                 % aggiorna la memoria
            else
                if isempty(ultima_velocita)
                    warning('Comando G1 senza velocità F e nessuna velocità
memorizzata: %s', riga);
                continue; % salta il comando se nessuna velocità precedente
            end
        end
    end
end

```

```

                else
                    velocita = ultima_velocita;                % usa ultima velocità
valida
                end
            end
        else
            % G0 movimento rapido
            velocita = velocita_rapida;
        end

% Estrazione coordinate X e Y
x_estratta = regexp(riga, 'X([\d\.\-]+)', 'tokens');
y_estratta = regexp(riga, 'Y([\d\.\-]+)', 'tokens');

if ~isempty(x_estratta)
    x_target = str2double(x_estratta{1});
else
    x_target = x_reale;
end
if ~isempty(y_estratta)
    y_target = str2double(y_estratta{1});
else
    y_target = y_reale;
end

% Calcolo distanze dal target e direzioni
dx = x_target - x_reale;
dy = y_target - y_reale;
L = sqrt(dx^2 + dy^2); % lunghezza spostamento

if L < 1e-9
    continue; % niente movimento
end

Nsteps = max(round(L / step_mm),1);
step_dx = dx / Nsteps;
step_dy = dy / Nsteps;
Ts = L / (velocita * Nsteps);

for k = 1:Nsteps
    %nuova posizione
    x_new = x_reale + step_dx;
    y_new = y_reale + step_dy;

    % Avanza X se serve
    if round(x_new/step_mm) ~= round(x_reale/step_mm)
        dir_x = sign(step_dx);
        fase_idx_x = fase_idx_x + dir_x;
        idx_mod_x = mod(fase_idx_x, Nseq) + 1;
        fase_x = fasi(idx_mod_x, :);
    else
        if isempty(phases_x)
            fase_x = [0 0];
        else
            fase_x = phases_x(end,:);
        end
    end

    % Avanza Y se serve
    if round(y_new/step_mm) ~= round(y_reale/step_mm)
        dir_y = sign(step_dy);
        fase_idx_y = fase_idx_y + dir_y;
        idx_mod_y = mod(fase_idx_y, Nseq) + 1;
        fase_y = fasi(idx_mod_y, :);
    else
        if isempty(phases_y)

```

```

        fase_y = [0 0];
    else
        fase_y = phases_y(end,:);
    end
end

% Aggiorna posizioni reali
x_reale = x_new;
y_reale = y_new;

% Aggiorna tempo e fasi
t = t + Ts;
tempo(end+1,1) = t;
phases_x(end+1,:) = fase_x;
phases_y(end+1,:) = fase_y;
end
elseif startswith(riga, 'G21') || startswith(riga, 'G90')
    % Ignora questi comandi
else
    warning('Comando G-code non riconosciuto: %s', riga);
end
end

% Output per Simulink
input_x = [tempo, phases_x];
input_y = [tempo, phases_y];

save('stepper_input_x.mat', 'input_x');
save('stepper_input_y.mat', 'input_y');

%% Carrello
m = 1;          % massa carrello [kg]

%% Cinghia
x0 = 0;        % posizione iniziale carrello
l = 0.6/2;     % semi interasse tra le due pulegge
E = 822;       % modulo di Young cinghia [MPa]
b = 6;         % larghezza cinghia
h = 1.38;     % altezza cinghia
A = b*h;      % sezione cinghia [mm^2]
eta = 0.1;    % coefficiente di perdita dovuto a deformazione
f = 5;        % frequenza Hz
w = 2*pi*f;   % pilsazione [rad/s]

k1 = E*A/(l+x0); % rigidezza tratto 1
k2 = E*A/(l-x0); % rigidezza tratto 2
k3 = E*A/(2*l);  % rigidezza tratto 3

c1 = k1*eta/w;   % smorzamento tratto 1
c2 = k2*eta/w;   % smorzamento tratto 2
c3 = k3*eta/w;   % smorzamento tratto 3

%% Attrito
mus = 0.3;      % coefficiente di attrito statico
mud = 0.15;    % coefficiente di attrito dinamico
N = 10;        % carico normale [N]

F_brk = mus*N;  % forza di attrito di distacco [N]
F_Coul = mud*N; % forza di Coulomb [N]
v_brk = 0.001; % velocità di distacco [m/s]
%v_brk = 0.005; % velocità di distacco [m/s]
c_1 = 1e-4;    % coefficiente di attrito viscoso guide [Ns/m]
%% Avvio simulazione
sim Modello_completo

```

```

%% Plot asse x
figure(5)
plot(time, px, ans.xm.Time, ans.xm.Data, ans.x.Time, ans.x.Data, 'LineWidth', 1.5)
grid on
xlabel ('Tempo [s]')
ylabel ('Spostamento [mm]')
legend ('Spostamento teorico', 'Spostamento imposto sulla puleggia', 'Spostamento del
carrello')
title ('Spostamento x(t)')

figure(6)
plot(t_v, vx, ans.xmdot.Time, ans.xmdot.Data, ans.xdot.Time, ans.xdot.Data,
'LineWidth', 1.5)
grid on
xlabel ('Tempo [s]')
ylabel ('Velocità [mm/s]')
legend ('Velocità teorica', 'Velocità imposta sulla puleggia', 'Spostamento del
carrello')
title ('Velocità v_x(t)')

%% Plot asse y
figure(7)
plot(time, py, ans.ym.Time, ans.ym.Data, ans.y.Time, ans.y.Data, 'LineWidth', 1.5)
grid on
xlabel ('Tempo [s]')
ylabel ('Spostamento [mm]')
legend ('Spostamento teorico', 'Spostamento imposto sulla puleggia', 'Spostamento del
carrello')
title ('Spostamento y(t)')

figure(8)
plot(t_v, vy, ans.ymdot.Time, ans.ymdot.Data, ans.ydot.Time, ans.ydot.Data,
'LineWidth', 1.5)
grid on
xlabel ('Tempo [s]')
ylabel ('Velocità [mm/s]')
legend ('Velocità teorica', 'Velocità imposta sulla puleggia', 'Spostamento del
carrello')
title ('Velocità v_y(t)')

%% Plot spostamento asse xy
figure(9)
plot(px, py, ans.xm.Data, ans.ym.Data, ans.x.Data, ans.y.Data, 'LineWidth', 1.5)
grid on
xlabel ('Spostamento lungo x [mm]')
ylabel ('Spostamento lungo y [mm]')
legend ('Spostamento teorico', 'Spostamento imposto sulle pulegge', 'Spostamento
ugello')
title('Spostamento nel piano XY')

%% Plot velocità asse xy
vf = sqrt((ans.xdot.Data).^2 + (ans.ydot.Data).^2);

figure(10)
plot(t_v, v, ans.xdot.Time, vf, 'LineWidth', 1.5)
grid on
xlabel ('Tempo [s]')
ylabel ('Velocità ugello [mm/s]')
legend ('Velocità teorica', 'Velocità ugello')
title ('Modulo della velocità nel piano XY')

```

7.3 Risultati della simulazione

Nel modello sono stati inseriti sensori di posizione per misurare:

- lo spostamento $x_m(t)$ trasmesso alla puleggia motrice dell'asse x, e la velocità $\dot{x}_m(t)$
- lo spostamento $y_m(t)$ trasmesso alla puleggia motrice dell'asse y, e la velocità $\dot{y}_m(t)$
- lo spostamento $x(t)$ del carrello e la sua velocità $\dot{x}(t)$
- lo spostamento $y(t)$ del carrello e la sua velocità $\dot{y}(t)$

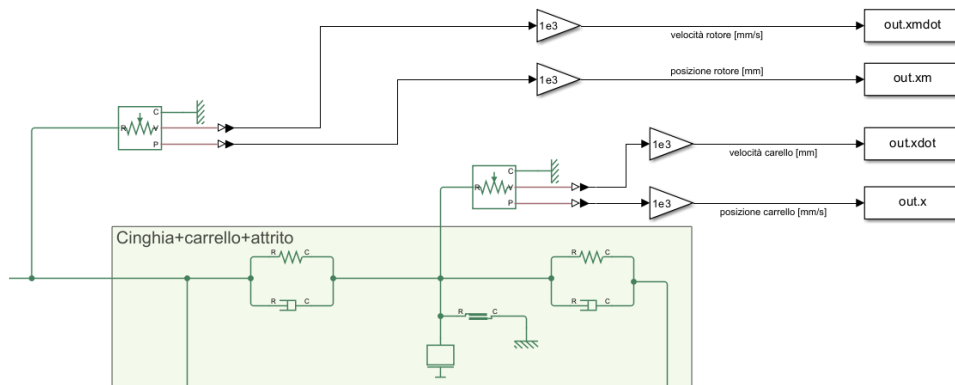


Figura 7.6: Dettaglio dei sensori riportati in Figura 7.2

A questo punto è possibile avviare la simulazione e rappresentare graficamente gli andamenti delle grandezze misurate, confrontandole con i corrispettivi andamenti teorici ottenuti nella sezione 3.2.

7.3.1 Solver

Per evitare errori numerici, si utilizzano le seguenti impostazioni del solver:

- Type: Variable-step
- Solver: ode23t (mod. stiff/Trapezoidal)
- Max step size: 1e-3
- Relative tolerance: 1e-4

7.3.2 Confronto degli spostamenti lungo l'asse x

In Figura 7.7 è riportato il confronto tra lo spostamento teorico del carrello, lo spostamento del carrello $x_m(t)$ ottenuto dal modello del motore passo-passo e della puleggia, e lo spostamento del carrello $x(t)$ ottenuto dal modello completo, che comprende anche la trasmissione a cinghia e l'attrito.

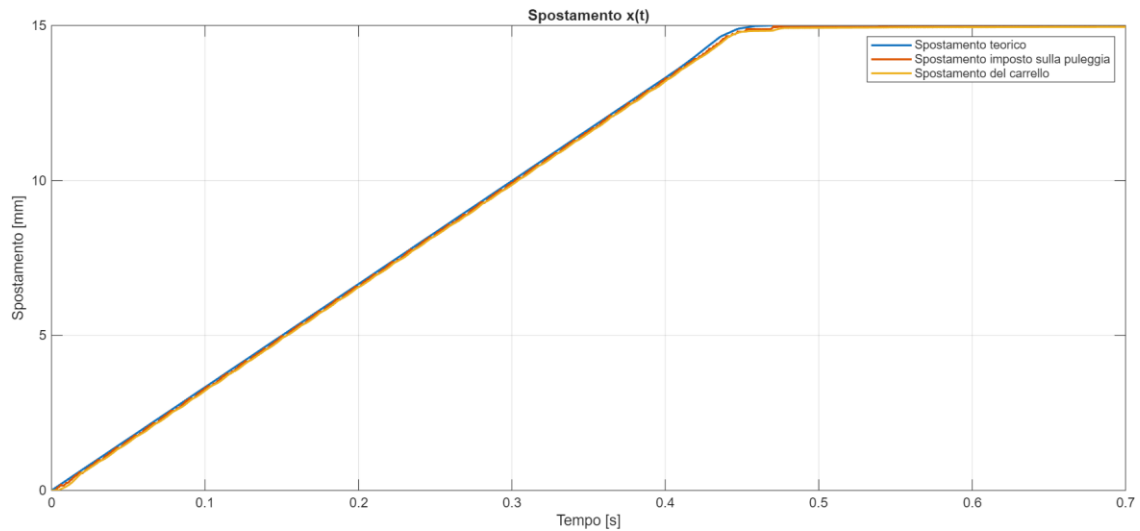


Figura 7.7: Confronto degli spostamenti lungo l'asse x.

Lungo i tratti rettilinei, gli spostamenti lungo l'asse x seguono con elevata accuratezza lo spostamento teorico. Nel tratto parabolico si osserva una lieve riduzione della precisione, che tuttavia rimane ampiamente adeguata agli obiettivi della tesi.

7.3.3 Confronto degli spostamenti lungo l'asse y

In Figura 7.8 è riportato il confronto tra lo spostamento teorico del carrello, lo spostamento del carrello $y_m(t)$ ottenuto dal modello del motore passo-passo e della puleggia, e lo spostamento del carrello $y(t)$ ottenuto dal modello completo, che comprende anche la trasmissione a cinghia e l'attrito.

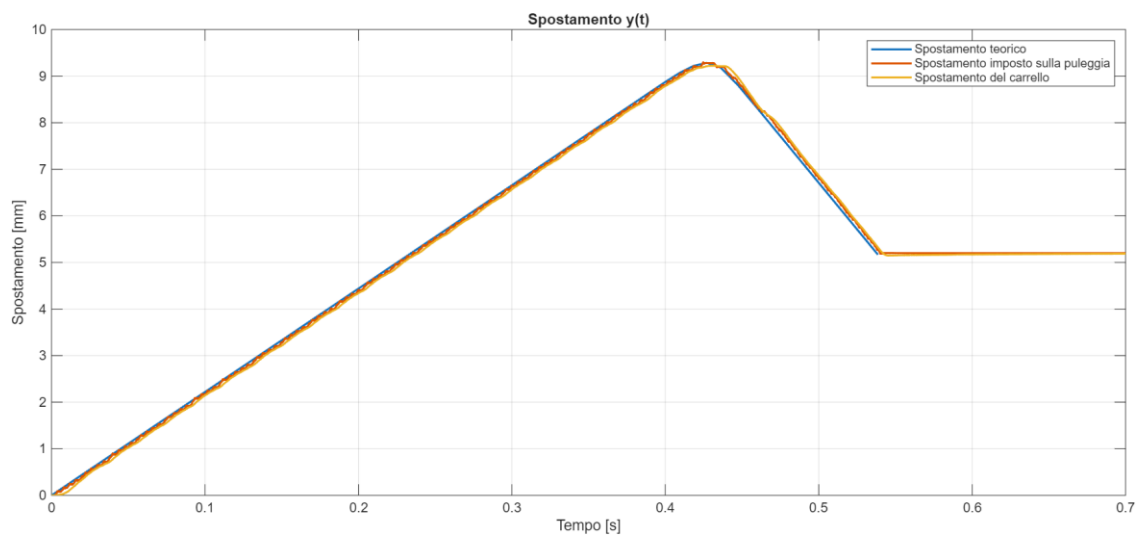


Figura 7.8: Confronto degli spostamenti lungo l'asse y.

Lungo il primo tratto rettilineo, lo spostamento lungo l'asse y segue con elevata accuratezza lo spostamento teorico. Nel tratto parabolico e nel secondo tratto rettilineo, si osserva una lieve riduzione della precisione, che tuttavia rimane ampiamente adeguata agli obiettivi della tesi.

7.3.4 Confronto delle velocità lungo l'asse x

In Figura 7.9 è riportato il confronto tra la velocità teorica del carrello, la velocità del carrello $\dot{x}_m(t)$ ottenuta dal modello del motore passo-passo e della puleggia, e la velocità del carrello $\dot{x}(t)$ ottenuta dal modello completo, che comprende anche la trasmissione a cinghia e l'attrito.

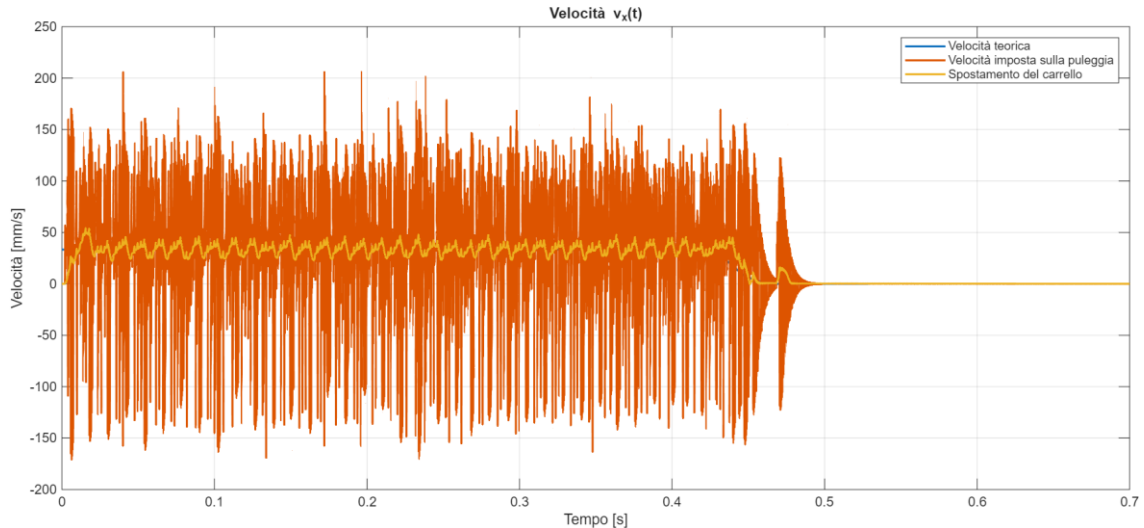


Figura 7.9: Confronto delle velocità lungo l'asse x.

L'andamento delle velocità lungo l'asse x è caratterizzato da oscillazioni. In particolare, la velocità trasmessa dal motore passo-passo alla puleggia presenta oscillazioni molto elevate, dovute alla dinamica del motore (si veda Sezione 5.2). Nella Figura 7.10 è riportato un ingrandimento, dal quale si osserva che il valore medio delle oscillazioni risulta coerente con la velocità teorica.

Per quanto riguarda la velocità del carrello, si osservano oscillazioni di ampiezza inferiore rispetto alla precedente e con frequenza notevolmente più bassa, a causa dell'effetto smorzante introdotto dalla trasmissione a cinghia, inoltre, il valore medio segue molto bene il valore teorico. Pertanto, i risultati della simulazione per la velocità lungo l'asse x risultano validi e coerenti.

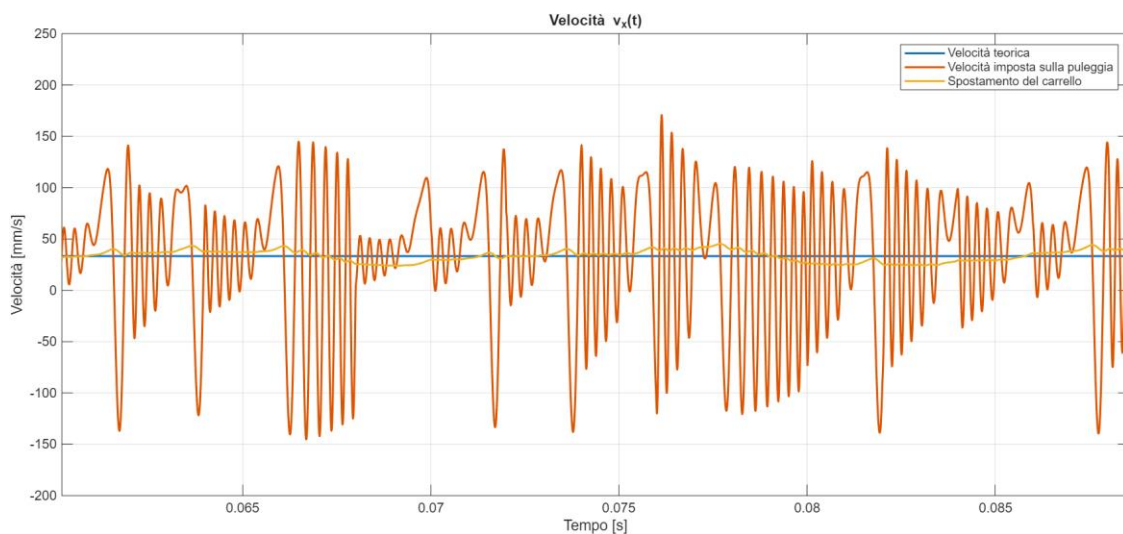


Figura 7.10: Ingrandimento Figura 7.9

7.3.5 Confronto delle velocità lungo l'asse y

In Figura 7.11 è riportato il confronto tra la velocità teorica del carrello, la velocità del carrello $\dot{y}_m(t)$ ottenuta dal modello del motore passo-passo e della puleggia, e la velocità del carrello $\dot{y}(t)$ ottenuta dal modello completo, che comprende anche la trasmissione a cinghia e l'attrito.

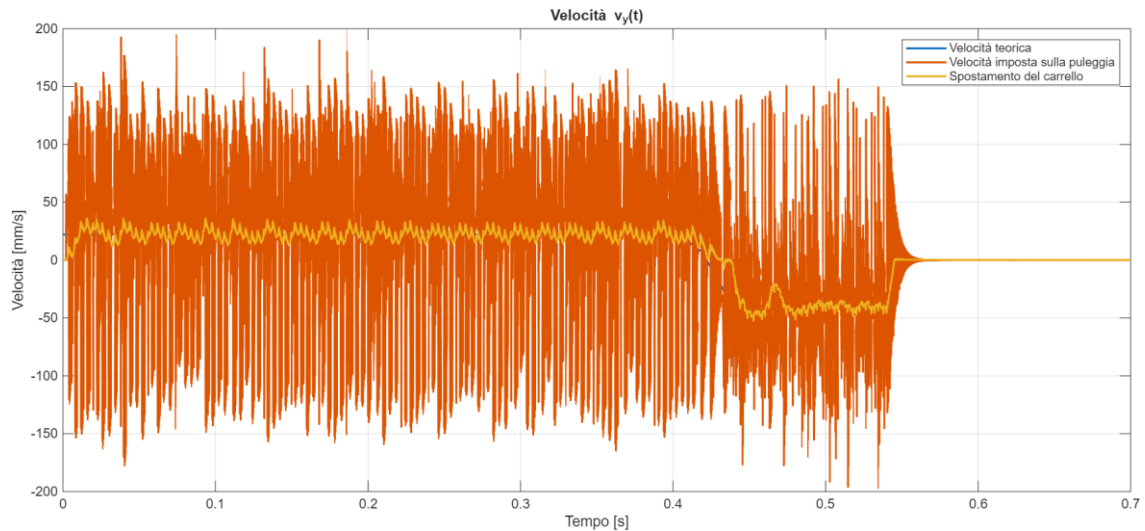


Figura 7.11: Confronto delle velocità lungo l'asse y

Per quanto riguarda l'andamento delle velocità lungo l'asse y, valgono le stesse considerazioni già discusse per l'asse x. Dalla Figura 7.12 si osserva che i risultati della simulazione per la velocità lungo l'asse y risultano validi e coerenti.

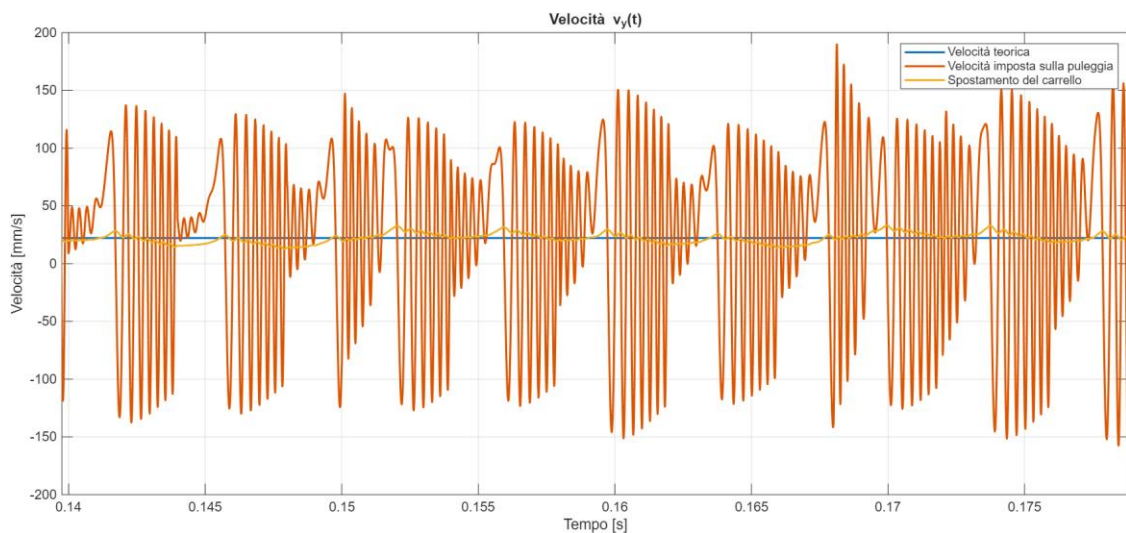


Figura 7.12: Ingrandimento Figura 7.11

7.3.6 Confronto degli spostamenti nel piano XY

In Figura 7.13 è riportato il confronto tra lo spostamento teorico dell'ugello, lo spostamento dell'ugello $y_m(x_m)$ ottenuto dal modello del motore passo-passo e puleggia, e lo spostamento dell'ugello $y(x)$ ottenuto dal modello completo che considera anche la trasmissione a cinghia e l'attrito.

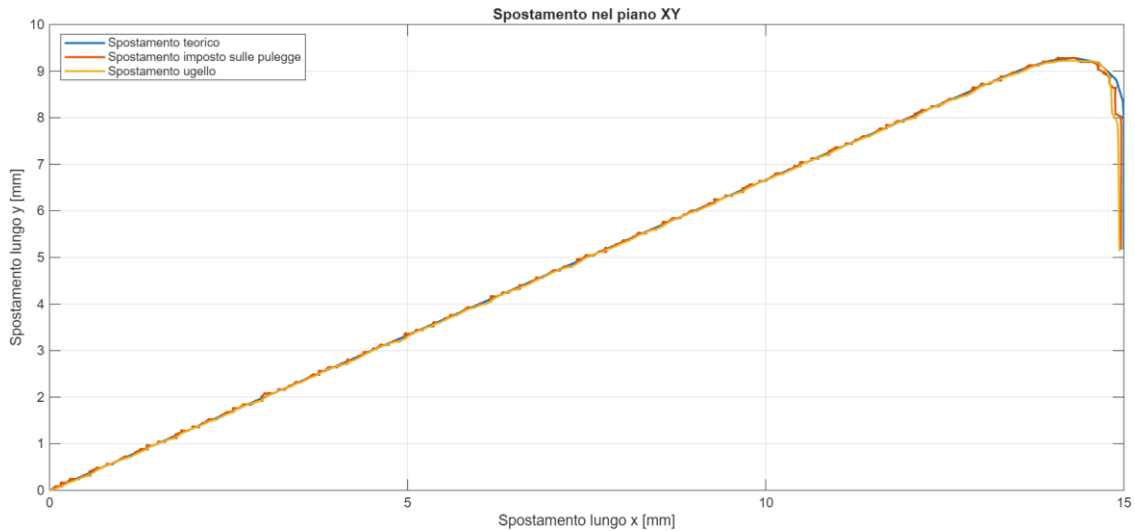


Figura 7.13: Confronto degli spostamenti nel piano XY

Lungo il primo tratto rettilineo, lo spostamento dell'ugello nel piano XY segue con elevata accuratezza lo spostamento teorico. Dall'ingrandimento riportato in Figura 7.14 si osserva, infatti, che in questo tratto lo spostamento dell'ugello risulta molto prossimo a quello teorico. Nel tratto parabolico e nel secondo tratto rettilineo si rileva una lieve riduzione della precisione, che tuttavia rimane ampiamente adeguata agli obiettivi della tesi. Lo scopo di questo capitolo è infatti quello di sviluppare ex novo un codice MATLAB per il controllo dei motori, quindi il risultato ottenuto può ritenersi pienamente soddisfacente.

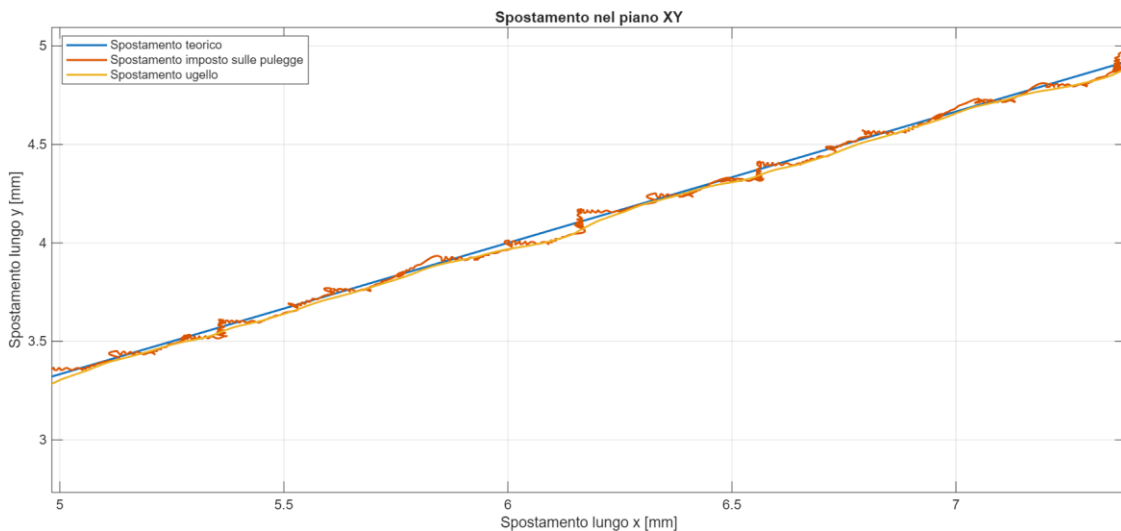


Figura 7.14: Ingrandimento Figura 7.13

7.3.7 Confronto dei moduli delle velocità nel piano XY

In Figura 7.15 è riportato il confronto tra i moduli della velocità teorica dell'ugello e della velocità dell'ugello $v = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}$ calcolata dai componenti di velocità lungo X e Y ottenuti dal modello completo che considera anche la trasmissione a cinghia e l'attrito.

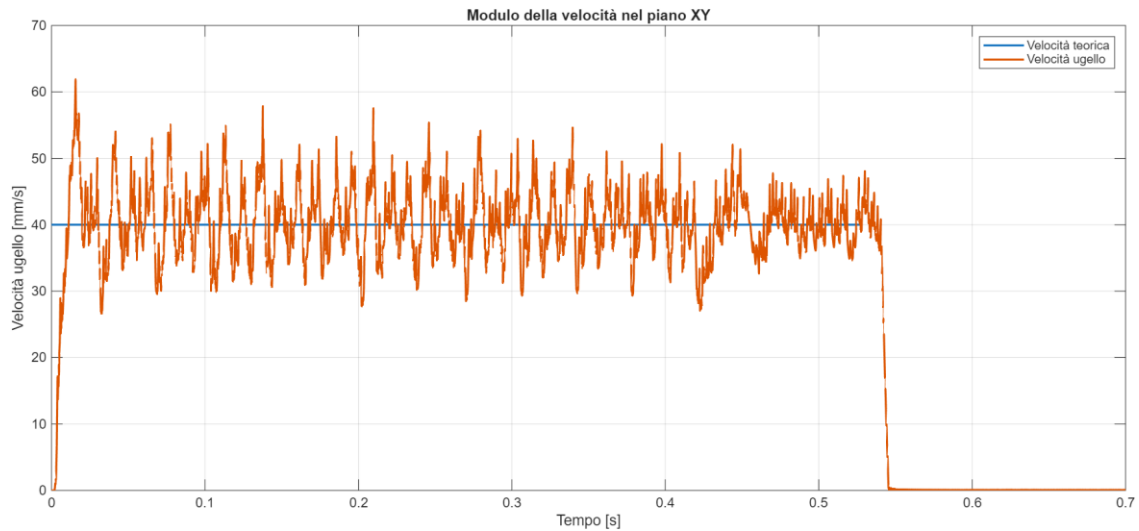


Figura 7.15: Confronto dei moduli delle velocità nel piano XY

L'andamento del modulo delle velocità dell'ugello nel piano XY è caratterizzato da oscillazioni, attribuibili alla dinamica del motore. Nonostante ciò, il valore medio segue il valore teorico di 40 mm/s. Pertanto, i risultati ottenuti confermano la validità dei modelli sviluppati e l'efficacia del sistema di controllo, evidenziando come questi strumenti di calcolo numerico e di simulazione possano essere impiegati per la progettazione, l'analisi e l'ottimizzazione di una stampante FDM.

8 Conclusione

In questo lavoro di tesi è stato sviluppato un sistema per generare e controllare il movimento dell'ugello di una stampante FDM cartesiana, a partire da un modello matematico della traiettoria di stampa fino alla gestione delle sequenze di attivazione degli avvolgimenti dei motori passo-passo che controllano gli assi X e Y della stampante. È stata inoltre realizzata la simulazione dinamica dell'intero sistema comprendente la modellazione del controllo elettrico dei motori passo-passo, dei motori stessi, delle pulegge, delle cinghie, dei carrelli e dell'attrito sulle guide. L'obiettivo principale è stato quindi quello di trasformare le traiettorie continue in comandi discreti per i motori passo-passo, al fine di ottenere un movimento dell'ugello con elevata precisione di posizionamento e il più possibile fluido.

Il primo studio condotto in questo lavoro riguarda il fenomeno dello stick slip nella trasmissione a cinghia dentata. I risultati hanno mostrato che questo fenomeno è particolarmente evidente alle basse velocità e dipende fortemente dai parametri del sistema. Come osservato, l'accumulo e il rilascio di energia elastica nella cinghia, dovuti alla forza resistente di attrito, fanno sì che il carrello segua lo spostamento desiderato con un andamento a gradini, generando oscillazioni sia nella velocità sia nelle forze trasmesse al carrello. Da questo studio è emerso che, per migliorare la stabilità del movimento riducendo al minimo il fenomeno dello stick slip e aumentando la precisione di posizionamento del carrello, è necessario scegliere in modo adeguato i parametri di rigidità e smorzamento della cinghia, la massa del carrello e il valore dell'attrito.

Successivamente è stata studiata una traiettoria costituita da due tratti rettilinei raccordati tramite curve di Bézier, le quali sono descritte da una funzione $\mathbf{p}(u)$ che restituisce un vettore posizione nello spazio, dove il parametro u varia tra 0 e 1. Nelle stampanti FDM cartesiane questo vettore viene espresso rispetto a un sistema di riferimento cartesiano corrispondente al piano XY del piano di stampa. La traiettoria di stampa è stata generata in modo tale da essere percorsa a velocità costante, al fine di limitare accelerazioni troppo brusche, ridurre le vibrazioni meccaniche e migliorare la qualità della stampa. Una volta generate in MATLAB le curve di Bézier e verificato che le coordinate dei punti nel piano seguissero la traiettoria desiderata, si è proceduto con la generazione del file G-code necessario per il controllo dei motori passo-passo della macchina FDM.

Lo studio successivo delle diverse tipologie di motori passo-passo ha confermato l'efficacia e l'ampio utilizzo, in particolare del motore passo-passo ibrido, nelle applicazioni delle stampanti FDM. La simulazione del motore ha mostrato che il movimento dell'albero motore presenta un andamento a gradini, con piccole oscillazioni dovute all'inerzia del rotore. Sono state analizzate le modalità di controllo full-step e half-step, quest'ultima aumenta la risoluzione angolare rendendo il moto più continuo e risultando quindi più adatta quando è richiesta una maggiore precisione. Successivamente, la modellazione della trasmissione meccanica, che include il motore passo-passo che gestisce l'asse X e il relativo sistema di controllo, insieme agli elementi meccanici quali cinghia, puleggia e carrello, considerando anche l'attrito, ha permesso di ottenere un modello più realistico del sistema, evidenziando l'effetto smorzante della cinghia.

In seguito, è stato aggiunto nel modello un secondo motore passo-passo per gestire l'asse Y. La parte centrale del lavoro di tesi è stata quindi la creazione di una logica di controllo in

grado di fornire la corretta sequenza di attivazione degli avvolgimenti dei motori passo-passo, in modo che i due motori, quello che gestisce l'asse X e quello che gestisce l'asse Y, lavorino in maniera coordinata affinché i rispettivi carrelli permettano all'ugello di stampa di muoversi secondo la traiettoria definita nel file G-code precedentemente generato.

Questo codice MATLAB, combinato con gli altri codici sviluppati per la modellazione dell'attrito, del motore e della trasmissione, e integrato in Simulink e Simscape con i componenti modellati, ha permesso di simulare l'intero sistema di stampa. I risultati hanno mostrato una buona corrispondenza tra le traiettorie teoriche e quelle effettivamente percorse dall'ugello, soprattutto nei tratti rettilinei, e hanno confermato che la velocità dell'ugello lungo la traiettoria rimane sostanzialmente costante, come previsto dalla descrizione matematica delle curve di Bézier. Nel tratto curvilineo della traiettoria si osservano lievi deviazioni, dovute principalmente alla dinamica e alla precisione dei motori e probabilmente anche all'algoritmo utilizzato per la generazione delle sequenze di attivazione degli avvolgimenti, la cui logica è stata sviluppata interamente in questo lavoro di tesi, costituendo quindi una prima implementazione del codice. Tali deviazioni risultano comunque molto contenute e non compromettono l'obiettivo del lavoro, per cui i risultati ottenuti possono essere considerati altamente soddisfacenti rispetto agli obiettivi prefissati.

Nel complesso, i risultati confermano la validità dei modelli sviluppati e l'efficacia del sistema di controllo proposto, evidenziando come strumenti di calcolo numerico e simulazione possano essere efficacemente impiegati nella progettazione, nell'analisi e nell'ottimizzazione del funzionamento di una stampante FDM.

In futuro, dato che il codice MATLAB sviluppato per la generazione delle sequenze di attivazione è stato progettato per far lavorare i motori in modalità full step, a partire da questo lavoro di tesi si potrebbe modificare l'algoritmo per operare in modalità half-step, al fine di migliorare ulteriormente la precisione di posizionamento dell'ugello. Inoltre, si potrebbe lavorare sull'ottimizzazione del codice sviluppato per renderlo ancora più efficiente. Infine, a partire dai modelli già sviluppati nella tesi, un ulteriore sviluppo potrebbe riguardare l'estensione dello studio mediante la realizzazione di modelli ancora più dettagliati del sistema di stampa, aggiungendo la modellazione di altri componenti della stampante per simulare in maniera ancora più realistica il comportamento della macchina.

Bibliografia

- [1] Mathworks, «MATLAB,» [Online]. Available: <https://it.mathworks.com/products/matlab.html>.
- [2] MathWorks, «Simulink,» [Online]. Available: <https://it.mathworks.com/products/simulink.html>.
- [3] MathWorks, «Simscape,» [Online]. Available: <https://it.mathworks.com/products/simscape.html>.
- [4] Federico Colombo, Sistemi multibody, slide del corso, Progettazione per la fabbricazione additiva, Politecnico di Torino.
- [5] Nannan GUO e Ming C.LEU, «Additive manufacturing: technology, applications and research needs,» *Frontiers of Mechanical Engineering* 2013. [Online]. Available: [257690902_Additive_manufacturing_Technology_applications_and_research_needs](https://doi.org/10.1007/978-98-99-10-257-6_25).
- [6] Ian Gibson, David Rosen e Brent Stucker, «Additive Manufacturing Technologies - 3D Printing, Rapid Prototyping and Direct Digital Manufacturing,» Second Edition, Springer. [Online]. Available: https://repo.darmajaya.ac.id/3831/1/Additive%20Manufacturing%20Technologies_%203D%20Printing%2C%20Rapid%20Prototyping%2C%20and%20Direct%20Digital%20Manufacturing%20%28%20PDFDrive%20%29.pdf.
- [7] Donghong Ding, Zengxi Pan, Dominic Cuiuri, Huijun Li e Stephen van Duin, «Advanced Design for Additive Manufacturing: 3D Slicing and 2D Path Planning,» 2016. [Online]. Available: <https://www.intechopen.com/chapters/50453>.
- [8] A.Brown, D. De Beer e P. Conradie, «DEVELOPMENT OF A STEREO LITHOGRAPHY (STL) INPUT AND COMPUTER NUMERICAL CONTROL (CNC) OUTPUT ALGORITHM FOR AN ENTRY-LEVEL 3-D PRINTER,» *S AFR J IND ENG*, vol. 25, no. 2, pp. 39–47, Aug. 2014. [Online]. Available: <https://sajie.journals.ac.za/pub/article/view/675/541>.
- [9] Yu-an Jin, Yong He, Jian-zhong Fu, Wen-feng Gan e Zhi-wei Lin, «Optimization of tool-path generation for material extrusion-based additive manufacturing technology,» *Additive Manufacturing Volumes 1–4*, October 2014, Pages 32-47. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S2214860414000098>.
- [10] Fredrick Madaraka Mwema e Esther Titilayo Akinlabi, «Basics of Fused Deposition Modelling (FDM),» *Fused Deposition Modeling. SpringerBriefs in Applied Sciences and Technology()*. Springer, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-48259-6_1.
- [11] Ekaterina Kopets, Artur Karimov, Lorenzo Scalera e Denis Butusov, «Estimating Natural Frequencies of Cartesian 3D Printer Based on Kinematic Scheme,» *Applied*

- Sciences. 2022; 12(9):4514. [Online]. Available: <https://doi.org/10.3390/app12094514>.
- [12] Prusa Knowledge Base, «Tipi di stampanti e le loro differenze,» [Online]. Available: https://help.prusa3d.com/it/article/tipi-di-stampanti-e-le-loro-differenze_112464.
- [13] MathWorks, «Translational Friction,» [Online]. Available: <https://it.mathworks.com/help/simscape/ref/translationalfriction.html>.
- [14] MathWorks, «Simscape - Blocchi,» [Online]. Available: https://it.mathworks.com/help/simscape/referencelist.html?type=block&s_tid=CRUX_topnav.
- [15] MathWorks, «Simulink - Blocchi,» [Online]. Available: https://it.mathworks.com/help/simulink/referencelist.html?type=block&s_tid=CRUX_topnav.
- [16] Federico Colombo, Curve parametriche e traiettorie di stampa, slide del corso, Progettazione per la fabbricazione additiva, Politecnico di Torino.
- [17] Franco Concli, «Il motore passo passo,» Organi di Trasmissione 2014 (10). [Online]. Available: https://www.researchgate.net/publication/269407472_Il_motore_passo_passo.
- [18] ISL Products, «Stepper Motor Fundamentals,» [Online]. Available: <https://islproducts.com/design-note/stepper-motor-fundamentals/?srsltid=AfmBOoq70hUYKdC4YHuwQ1hjIl6b0Z1-jWikpy5zMtaUN6iRrMFmhVeq>.
- [19] Federico Colombo, I motori elettrici a passo, slide del corso, Progettazione per la fabbricazione additiva, Politecnico di Torino.
- [20] HomoFaciens, «Working principles of several stepper motors (bipolar, unipolar, reluctance and can-stack),» [Online]. Available: <https://www.youtube.com/watch?v=Qc8z cst2blU>.
- [21] FARE ELETTRONICA, «I Motori Passo-Passo,» [Online]. Available: <https://farelettronica.it/i-motori-passo-passo/>.
- [22] S. C. Engineering, «How does a Stepper Motor work?,» [Online]. Available: <https://www.youtube.com/watch?v=eyqwLiowZiU>.
- [23] StepperOnline, «Nema 17 Bipolare 0.9deg 46Ncm(65.1 oz.in) 2A 42x42x48mm 4 Fili,» [Online]. Available: <https://www.omc-stepperonline.com/it/nema-17-bipolare-0-9deg-46ncm-65-1-oz-in-2a-2-8v-42x42x48mm-4-fili-17hm19-2004s>.
- [24] MathWorks, «Pulse Generator,» [Online]. Available: <https://www.mathworks.com/help/simulink/slref/pulsegenerator.html>.
- [25] MathWorks, «Switch,» [Online]. Available: <https://it.mathworks.com/help/simscape/ref/switch.html>.
- [26] MathWorks, «Stepper Motor,» [Online]. Available: <https://it.mathworks.com/help/sps/ref/steppermotor.html>.

- [27] Federico Colombo, Modello trasmissione a cinghia, slide del corso, Progettazione per la fabbricazione additiva, Politecnico di Torino.
- [28] Pinto Elettronica, «PULEGGIA 16 DENTI GT2 (FORO 5 MM),» [Online]. Available: <https://www.pintoelettronica.com/varie-puleggia-16-denti-gt2-foro-5-mm-.1.2.180.gp.12278.uw>.
- [29] Maschinentechnik, «Cinghia dentata Profilo GT,» [Online]. Available: <https://hc-maschinentechnik.de/Cinghia-dentata-Profilo-GT?srsltid=AfmBOopuyEIB-uvAU-ozorz0zkJwocXZjRunJUTXKcTrcL7bdigIiEQ>.