UNIVERSITA' DEGLI STUDI DI ROMA

TOR VERGATA

LAUREA MAGISTRALE IN

INGEGNERIA DELL'AUTOMAZIONE

Corso di

Meccanica delle vibrazioni (6 CFU)

STUDIO DELLE VIBRAZIONI DI UN SISTEMA MOTORIZZATO LUNGO L'ASSE VERTICALE SU DIVERSI TIPI DI TERRENO

STUDENTI

Alessandra Carmanini

Ilaria Costa

PROFESSORI

Ing. Massimo Cavacece

A.A. 2024/2025

Sommario

Indice delle figure

1 Introduzione

Il presente progetto ha l'obiettivo di studiare il comportamento dinamico di un sistema motorizzato soggetto a vibrazioni lungo l'asse verticale quando posto in movimento su diversi tipi di terreno. L'analisi si basa su un approccio sperimentale e teorico integrato, con lo scopo di confrontare i dati reali con un modello fisico ideale.

Per la parte sperimentale è stato realizzato un piccolo veicolo controllato da una scheda Arduino Mega, alimentata esternamente e interfacciata con un driver L298N per il controllo di due motori DC. Un accelerometro MPU6050 rileva le vibrazioni verticali durante il moto, mentre un modulo Bluetooth HC-05 trasmette i dati in tempo reale a un computer.

I segnali acquisiti vengono elaborati in MATLAB, dove si esegue un'analisi di Fourier per identificare le componenti in frequenza associate alle vibrazioni su differenti superfici (asfalto, sabbia, erba, ecc.). A partire dai dati ottenuti, si procede alla stima dei parametri c (smorzamento) e k (rigidezza) di un modello meccanico equivalente del sistema, rappresentato come massa-molla-smorzatore.

L'obiettivo finale è verificare la coerenza tra modello e realtà, validando la capacità del modello teorico di rappresentare fedelmente il comportamento dinamico del sistema.

2 Descrizione del progetto

Per modellare il comportamento dinamico del sistema in esame, è stato adottato un modello fisico semplificato costituito da una massa soggetta a forze elastiche e dissipative: sistema massa-molla-smorzatore. Tale modello rappresenta in modo efficace la risposta alle vibrazioni indotte dal moto su differenti tipi di terreno.

Nel modello considerato:

- le proprietà inerziali del sistema sono rappresentate dalla massa m;
- le proprietà elastiche da una molla con rigidezza k;
- le proprietà dissipative da uno smorzatore viscoso con coefficiente c.

2.1 Equazione del moto (Modello matematico)

Il sistema può essere descritto dalla seguente equazione differenziale del secondo ordine:

$$m\,\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t)$$

Dove:

- m è la massa del veicolo;
- c è il coefficiente di smorzamento viscoso;
- k è la costante elastica della molla equivalente;
- x(t) è lo spostamento verticale;
- $\ddot{x}(t)$ e $\dot{x}(t)$ appresentano rispettivamente l'accelerazione e la velocità (verticali);
- $F(t)$ è la forza esterna generata dalle irregolarità del terreno.

L'accelerazione $\ddot{x}(t)$ è misurata direttamente tramite l'accelerometro MPU6050 montato sul sistema. Il segnale grezzo viene trasmesso via Bluetooth (modulo HC-05) e analizzato in MATLAB.

2.2 Analisi nel dominio della frequenza

Per caratterizzare le vibrazioni indotte dal terreno, i segnali vengono elaborati tramite trasformata di Fourier, ottenendo lo spettro di frequenza della risposta del sistema. Questa analisi consente di identificare la frequenza dominante del sistema e confrontarla con quella prevista dal modello teorico. La funzione di trasferimento del sistema nel dominio della frequenza è data da:

$$H(j\omega) = \frac{X(j\omega)}{F(j\omega)} = \frac{1}{m(j\omega)^2 + c(j\omega) + k}$$

Questa espressione permette di analizzare la risposta del sistema in frequenza e di identificare fenomeni come la risonanza.

2.3 Stima dei parametri c e k

Parallelamente all'analisi in frequenza, è stata adottata una procedura di stima diretta dei parametri del modello nel dominio del termpo. A partire dai segnali di accelerazione, sono stati ricavati i segnali di velocità e posizione, atttraverso l'integrazione, e l'equazione del moto:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0$$

È stata riorganizzata nella forma:

$$c\dot{x}(t) + kx(t) = -m\ddot{x}(t)$$

E risolta tramite regressione lineare. Questa procedura ha consentito di stimare direttamete i valori del coefficiente di smorzamento c e della rigidezza k, fornendo una descrizione quantitativa del comportamento del sistema. I valori stimati di c e k vengono poi utilizzati per costruire il modello teorico del sistema, la cui risposta (in termini di accelerazione) viene confrontata direttamente con quella misurata sperimentalmente.

L'obiettivo della simulazione è verificare la coerenza tra la risposta teorica del modello e quella misurata sperimentalmente, confermando la validità della modellazione adottata.

3 Hardware

Per la realizzazione del sistema di misura delle vibrazioni, è stata progettata e assemblata una piattaforma hardware in grado di muoversi autonomamente e acquisire dati in tempo reale. Il cuore del sistema è costituito da una scheda di controllo programmabile, alla quale sono stati collegati diversi moduli elettronici dedicati alla movimentazione, alla rilevazione delle accelerazioni e alla trasmissione dei dati.

L'obiettivo principale dell'hardware è permettere il rilevamento delle vibrazioni verticali a cui è sottoposto il sistema durante il moto su superfici differenti, così da consentire un'analisi comparativa delle risposte meccaniche.

3.1 Arduino Mega

La scheda Arduino Mega 2560 rappresenta il centro di controllo del sistema. È una piattaforma di prototipazione elettronica basata sul microcontrollore ATmega2560, progettata per applicazioni che richiedono un numero elevato di connessioni e maggiori risorse rispetto ai modelli base. Dispone di 54 pin digitali di input/output (di cui 15 possono essere utilizzati come uscite PWM), 16 ingressi analogici, 4 porte seriali hardware (UART) e una memoria maggiore rispetto ad altre schede Arduino, con 256 KB di memoria flash per il programma e 8 KB di SRAM.

Nel progetto in esame, Arduino Mega è stato utilizzato per:

- acquisire i dati dall'accelerometro MPU6050 tramite interfaccia I2C,
- controllare la direzione e la velocità dei motori tramite il driver L298N,
- gestire la comunicazione Bluetooth con il modulo HC-05 per l'invio dei dati al PC.

La sua semplicità d'uso, unita alla compatibilità con numerosi sensori e moduli, ha reso questa scheda una scelta ideale per integrare in modo efficiente tutte le funzionalità necessarie al sistema di misura.

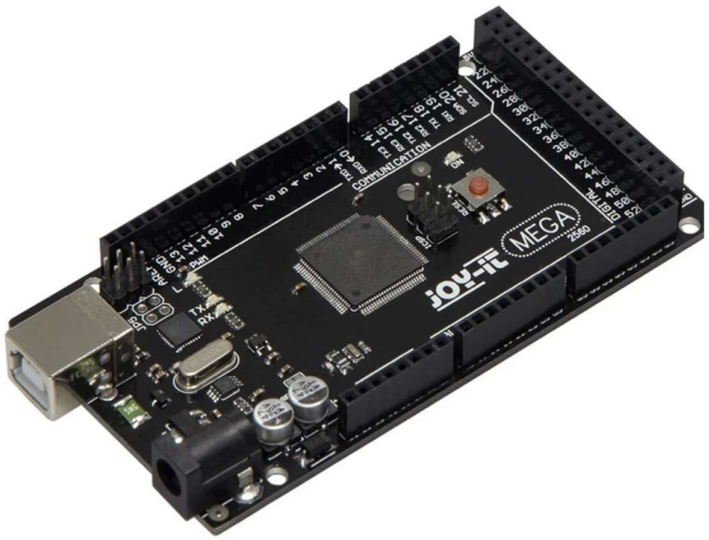

Figura 3.1 - Arduino Mega

3.2 Driver L928N

Il driver L298N è un modulo ponte H a doppio canale, utilizzato per controllare motori in corrente continua (DC) in entrambe le direzioni di rotazione. È basato sull'integrato L298N, un doppio ponte H capace di pilotare carichi come motori, relè e solenoidi. Nel progetto è stato impiegato per pilotare due motori DC, fornendo sia il controllo della direzione di rotazione sia la regolazione della velocità tramite segnale PWM (modulazione di larghezza di impulso) generato da Arduino.

Il modulo dispone di morsetti a vite per il collegamento dei motori e dell'alimentazione esterna, oltre a pin di controllo per l'interfaccia con la scheda Arduino. È stato alimentato separatamente dalla logica di controllo, così da garantire una potenza adeguata ai motori senza sovraccaricare la scheda.

L'utilizzo del driver L298N ha permesso una gestione stabile e flessibile del movimento del veicolo su differenti superfici, condizione necessaria per una corretta acquisizione delle vibrazioni durante la marcia.

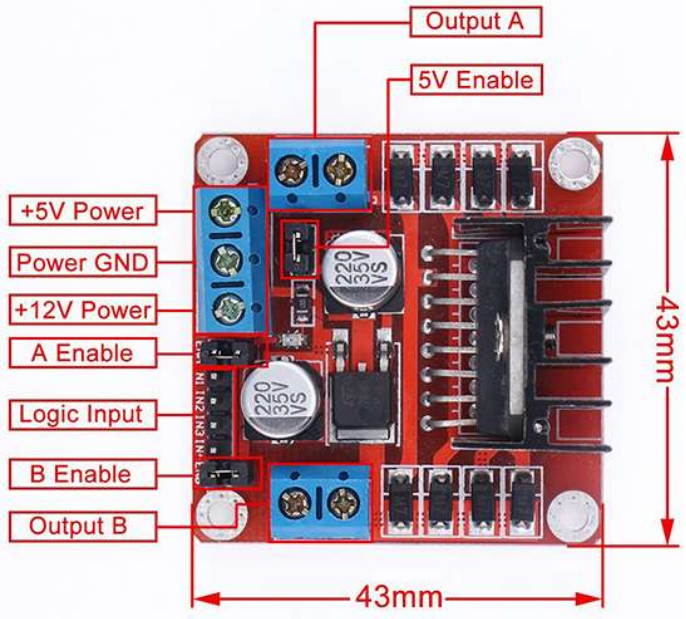

Figura 3.2 Driver L928N

3.3 Accelerometro MPU6050

Il MPU6050 è un sensore MEMS (Micro-Electro-Mechanical System) che integra in un unico chip un accelerometro triassiale e un giroscopio triassiale, permettendo di rilevare accelerazioni lineari e velocità angolari lungo i tre assi spaziali (X, Y, Z). Nel progetto è stato utilizzato principalmente per misurare le accelerazioni verticali, ovvero lungo l'asse Z, così da monitorare le vibrazioni trasmesse al sistema durante il movimento su diverse superfici.

Il modulo comunica con Arduino tramite il protocollo I²C, che consente una trasmissione dati semplice e veloce utilizzando solo due fili (SDA e SCL). I dati grezzi acquisiti sono espressi in unità di gravità (gg) e vengono successivamente convertiti in accelerazioni (m/s2m/s2) durante l'elaborazione su PC.

Grazie alla sua elevata sensibilità e compattezza, l'MPU6050 ha rappresentato una soluzione efficace per acquisire in tempo reale le sollecitazioni dinamiche del sistema, risultando fondamentale per le analisi vibrazionali condotte nella fase sperimentale.

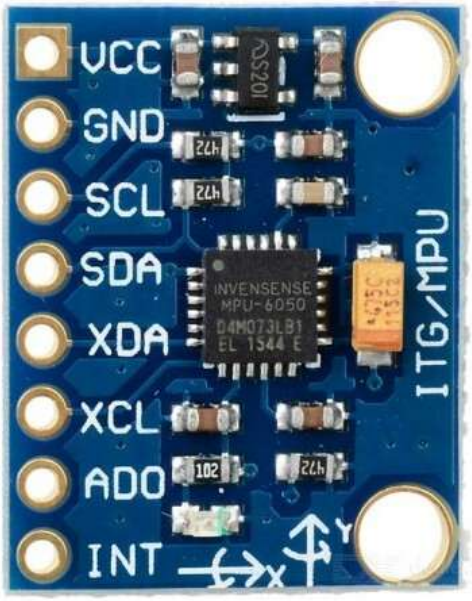

Figura 3.3 Acceleratore MPU6050

3.4 Modulo Bluetooth HC05

Il HC-05 è un modulo Bluetooth a bassa potenza progettato per la comunicazione seriale wireless, utilizzato nel progetto per trasmettere i dati raccolti dal sensore MPU6050 e dal sistema di controllo di Arduino al computer. Questo modulo supporta il protocollo Serial Port Profile (SPP), che consente di inviare e ricevere dati tra dispositivi Bluetooth in modo semplice ed efficiente.

Il modulo HC-05 è stato configurato in modalità slave, per ricevere i dati da Arduino e trasmetterli al PC via Bluetooth, eliminando la necessità di cavi fisici. La connessione tra il modulo e la scheda Arduino avviene tramite la porta seriale, utilizzando i pin RX e TX per la trasmissione e ricezione dei dati.

Grazie alla sua compatibilità con la maggior parte dei dispositivi Bluetooth e alla facilità di configurazione, il modulo HC-05 ha permesso di implementare una comunicazione wireless stabile, fondamentale per il trasferimento in tempo reale delle misurazioni delle vibrazioni e per l'analisi successiva su PC.

Figura 3.4 Modulo HC05

3.5 Componenti aggiuntivi

Oltre ai componenti principali già descritti, il sistema comprende due motori DC montati su una base mobile con ruote, che consentono al veicolo di muoversi su diverse superfici. Il movimento dei motori è gestito tramite un driver L298N, che controlla sia la direzione sia la velocità in risposta ai comandi inviati da Arduino. Questo controllo avviene mediante modulazione PWM, permettendo variazioni progressive della velocità e garantendo un movimento fluido e reattivo.

Tutti i componenti elettronici e meccanici sono montati su una struttura realizzata in legno, scelta per il suo buon compromesso tra leggerezza e robustezza. Questa soluzione costruttiva fornisce una base stabile al sistema, mantenendo al tempo stesso una buona manovrabilità complessiva del veicolo.

Il modulo Bluetooth HC-05 è collegato ad Arduino attraverso un breadboard, che ospita anche un partitore di tensione. Quest'ultimo è necessario per adattare i livelli logici del segnale seriale, evitando danni al modulo e garantendo una comunicazione affidabile tra i dispositivi.

A supporto dell'interazione con l'utente, è stato integrato un display LCD 16x2, utilizzato per visualizzare in tempo reale informazioni operative del robot, come lo stato di avvio, arresto e i valori acquisiti dall'accelerometro. La comunicazione con Arduino avviene in modalità parallela, e il contenuto del display viene aggiornato dinamicamente in base alle condizioni del sistema. Ciò permette un monitoraggio immediato anche in assenza di connessione al PC.

Per migliorare la leggibilità del display in diverse condizioni di luce ambientale, è stato aggiunto un potenziometro connesso al pin V0 dell'LCD. Questo semplice accorgimento consente la regolazione manuale del contrasto, aumentando l'usabilità complessiva del sistema.

L'alimentazione dei motori avviene tramite uno o più porta batterie dedicati, separati dall'alimentazione del microcontrollore. Questa separazione tra la parte logica (Arduino) e quella di potenza (motori) migliora la stabilità del sistema, evitando cali di tensione durante i momenti di maggiore assorbimento di corrente.

Figura 3.5 - Schermo LCD

Figura 3.6 - Potenziometro

Figura 3.7 - Pulsante

In figura mostriamo la macchina completa

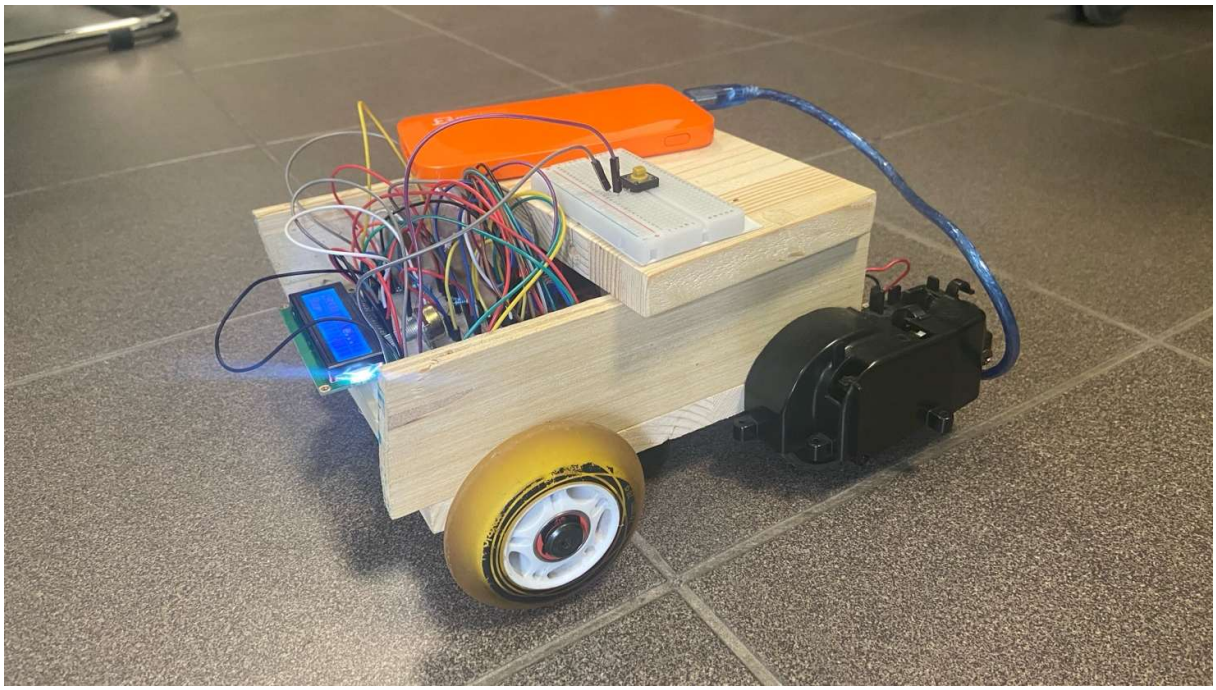

Figura 3.8 - Configurazione macchina completa

4 Software

Il codice sviluppato per questo progetto si compone di due parti principali:

1. Codice Arduino per il controllo della macchina e l'acquisizione dei dati.
2. Script Matlab per l'elaborazione, l'analisi e la modellazione dei segnali accelerometrici.

4.1 Arduino

Il codice implementato permette il controllo del movimento di una macchinetta tramite l'interazione con un pulsante fisico, un modulo Bluetooth e un sensore di accelerazione MPU6050. Il microcontrollore gestisce vari componenti hardware: due motori DC, un display LCD 16x2, il modulo Bluetooth (tramite la porta seriale Serial1) e il sensore MPU6050, coordinandoli per garantire un funzionamento fluido e affidabile.

In una fase preliminare è stato eseguito uno script di calibrazione per l'accelerometro, volto a determinare e correggere gli eventuali offset sui singoli assi. Questo passaggio è stato fondamentale per garantire che le misure di accelerazione fossero accurate e prive di errori sistematici dovuti a imperfezioni del sensore.

Nella funzione `setup()`, avviene la fase di inizializzazione in cui vengono configurati i pin per il controllo dei motori e del pulsante con resistenza di pull-up interna. Viene avviata la comunicazione seriale utilizzata per il monitoraggio via USB (debug) e per la comunicazione Bluetooth. Il display LCD viene inizializzato per mostrare messaggi di stato. Il sensore MPU6050 viene attivato tramite protocollo I2C, uscendo dalla modalità sleep, e configurato per misurare l'accelerazione con un intervallo di ±4g, bilanciando sensibilità e precisione.

```
void setup() {
  // Imposta i pin dei motori come output
  pinMode(enA, OUTPUT); pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT); pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT); pinMode(in4, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);  // Pulsante con pull-up interno

  // Inizializza comunicazione seriale e Bluetooth
  Serial.begin(9600);
  BTSerial.begin(9600);

  // Inizializza LCD
  lcd.begin(16, 2);
  lcd.print("Connessione...");

  // Inizializza MPU6050
  Wire.begin();
  Wire.beginTransmission(MPU_I2C_ADDR);
  Wire.write(0x6B);       // Registro di "power management"
  Wire.write(0);          // Wake-up MPU6050
  Wire.endTransmission(true);

  Wire.beginTransmission(MPU_I2C_ADDR);
  Wire.write(0x1C);       // Registro di configurazione accelerometro
  Wire.write(0x08);       // ±4g (sensibilità intermedia)
  Wire.endTransmission(true);

  // Display pronto
  lcd.setCursor(0, 1);
  lcd.print("Pronto         ");
}
```

Figura 4.1 - funzione setup()

Nel ciclo principale `loop()`, la lettura dei dati dall'accelerometro avviene ogni 2 ms (500 Hz), applicando una media mobile sui valori dell'asse Z per filtrare il rumore. Il valore filtrato viene inviato via Bluetooth e visualizzato in tempo reale sul display LCD, permettendo un monitoraggio locale e remoto.

```
// Legge accelerometro ogni 2ms
if (currentTime - lastReadTime >= interval) {
  lastReadTime = currentTime;
  readAccel();                    // Acquisisce nuovo valore da MPU
  float filteredZ = filteredAcZ(); // Calcola media mobile
```

Figura 4.2 - chiamata della funzione per filtrare i dati

```
// Calcola la media mobile dei valori Z per ridurre il rumore
float filteredAcZ() {
  float sum = 0;
  for (int i = 0; i < WINDOW_SIZE; i++) {
    sum += AcZ_window[i];
  }
  return sum / WINDOW_SIZE;
}
```

Figura 4.3 - definizione funzione per filtrare i dati

```
// Invia valore filtrato via Bluetooth e mostra su LCD
String out = String(filteredZ, 3);   // 3 cifre decimali
BTSerial.println(out);               // Invia via Bluetooth
lcd.setCursor(0, 0);
lcd.print("Z: ");
lcd.print(out);
lcd.print(" g   ");                  // Mostra su display
```

Figura 4.4 - invio dei dati tramite Seriale Bluetooth

Il movimento della macchinetta viene attivato tramite un pulsante fisico: alla pressione del pulsante, se la macchina è ferma, i motori si avviano immediatamente alla massima velocità (PWM pari a 255). Dopo 5 secondi di funzionamento a velocità costante, si avvia una fase di decelerazione lineare della durata di ulteriori 5 secondi, al termine della quale i motori si fermano completamente. Durante l'intero ciclo operativo, lo stato dei motori viene comunicato in tempo reale sul display tramite i messaggi "Motori ON" e "Motori OFF".

```
// Attiva i motori in avanti con PWM corrente
void startMotors() {
  analogWrite(enA, pwmValue);
  analogWrite(enB, pwmValue);
  digitalWrite(in1, HIGH); digitalWrite(in2, LOW); // direzione avanti A
  digitalWrite(in3, HIGH); digitalWrite(in4, LOW); // direzione avanti B
}
```

Figura 4.5 - funzione per attivare i motori

```
// Ferma completamente i motori
void stopMotors() {
  analogWrite(enA, 0);
  analogWrite(enB, 0);
  digitalWrite(in1, LOW); digitalWrite(in2, LOW);
  digitalWrite(in3, LOW); digitalWrite(in4, LOW);
}
```

Figura 4.6 - funzione per disattivare i motori

Il codice gestisce in modo efficace la temporizzazione del movimento e della decelerazione, assicurando transizioni graduali per evitare scatti o arresti bruschi. Inoltre, la gestione del pulsante prevede un debounce software semplice per evitare più attivazioni involontarie.
In sintesi, il codice integra la lettura e il filtraggio dei dati dell'accelerometro MPU6050, il controllo preciso dei motori DC tramite PWM e la gestione dell'interfaccia utente con pulsante e display LCD, realizzando un sistema di movimento controllato e monitorabile in tempo reale.

4.2 Matlab

Al termine della fase di acquisizione, i dati di accelerazione raccolti tramite accelerometro sono salvati in un file di testo per una successiva elaborazione in MATLAB.

```
% filename= 'Piastrelle.txt';                % Nome del file dati con le misure
% filename = 'Asfalto.txt';
filename = 'Brecciolato.txt';
Measures = importdata(filename);           % Importa i dati come cell array di stringhe
```

Figura 4.7 - importazione dei dati

Lo scopo principale di questa analisi è quello di caratterizzare la risposta dinamica del sistema vibrante, stimando parametri fisici fondamentali quali il coefficiente di smorzamento e la rigidezza elastica.

L'analisi ha inizio con l'importazione del file contenente le misure, dove ogni riga riporta il tempo di campionamento e il valore grezzo di accelerazione lungo l'asse z. I dati sono stati convertiti in unità fisiche corrette (m/s²), moltiplicando i valori originariamente espressi in g per l'accelerazione di gravità (9.81 m/s²). Successivamente si è eliminato l'offset medio per rimuovere la componente statica gravitazionale, ottenendo così il segnale di accelerazione dinamica pulito e si è applicato un filtro di smoothing basato su media mobile, per ridurre il rumore ad alta frequenza.

```
z = z_values* 9.81;
L = length(z);                             % Numero di campioni
t = (0:L-1)*Ts;                            % Vettore tempi corrispondenti

% Rimuovo offset DC (la media), per togliere la componente gravitazionale       %
z_real = z - mean(z);
z = smoothdata(z_real, 'movmean', 15);       % applico uno smoothing ai dati    %
```

Figura 4.8 - eliminazione del rumore nei dati

Il segnale risultante viene poi associato a un asse temporale costruito assumendo un intervallo di campionamento costante, scelto in base alle informazioni acquisite durante la misura.

Si procede quindi all'integrazione numerica del segnale di accelerazione per ricavare la velocità e, successivamente, la posizione del sistema vibrante nel tempo. Per minimizzare effetti di deriva dovuti all'integrazione, è stata applicata una correzione ai segnali integrati, sia per la velocità che per la posizione, eliminando eventuali trend lineari indesiderati.

```
%% Integrazione numerica per ottenere velocità e posizione
v = cumtrapz(t, a_moto);   % Velocità tramite integrazione numerica della accelerazione
v = detrend(v);            % Rimuove eventuali trend lineari dovuti a errori cumulativi nell'integrazione
x = cumtrapz(t, v);        % Posizione tramite integrazione numerica della velocità
x = detrend(x);            % Rimuove eventuali trend lineari da posizione
```

Figura 4.9 - integrazione dei dati

La dinamica del sistema viene modellata attraverso un modello massa-molla-smorzatore, rappresentato dall'equazione differenziale:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t).$$

Che, isolando i termini in c e k, si riscrive come:

$$c\dot{x}(t) + kx(t) = -m\ddot{x}(t)$$

Questa forma consente di effettuare una stima dei coefficienti di smorzamento viscoso c e di rigidezza elastica k tramite regressione lineare, risolvendo il problema dei minimi quadrati con vincolo di non negatività, utilizzando come regressori la velocità $\dot{x}(t)$ e la posizione $x(t)$, e come termine noto $-m\ddot{x}(t)$.

```
%% Preparazione dati per regressione
% Modello dinamico: m*a + c*v + k*x = 0
% Lo riorganizzo per la regressione: c*v + k*x = -m*a
Y = -m * a_moto;           % Vettore "uscita" (termine noto)
X = [v, x];                % Matrice regressori (velocità e posizione)

%% Regressione con vincolo di non negatività (per evitare valori fisicamente non plausibili)
params = lsqnonneg(X, Y);  % Risolve il problema di minimo quadrati con c,k >= 0

c = params(1);             % Coefficiente di smorzamento stimato (Ns/m)
k = params(2);             % Coefficiente di rigidezza stimato (N/m)

fprintf('Risultati regressione:\n');
fprintf('Smorzamento c = %.4f Ns/m\n', c);
fprintf('Rigidezza k = %.4f N/m\n', k);
```

Figura 4.10 - Regressione lineare

Infine, per analizzare le componenti in frequenza del segnale di accelerazione, si esegue la Trasformata di Fourier veloce (FFT), che consente di individuare le frequenze caratteristiche di risonanza e di indentificare eventuali disturbi o fenomeni periodici del sistema. Tale analisi è fondamentale per comprendere a fondo il comportamento dinamico del sistema.

```
len_d = length(a_moto);
Z_f = fft(a_moto); % analisi frequenze
P2 = abs(Z_f/len_d);
P1 = P2(1:len_d/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(len_d/2))/len_d;
plot(f,P1)
```

Figura 4.11 - applicazione della FFT

I risultati, comprensivi di valori stimati di smorzamento e rigidezza sono visualizzati graficamente e salvati per consentire i confronti delle simulazioni in diversi ambienti.

5 Analisi dei dati

I dati raccolti sono stati analizzati in due domini distinti. Tramite il software MATLAB, sono stati generati i grafici nel dominio della frequenza, al fine di confrontare il contenuto spettrale delle vibrazioni registrate su diversi tipi di terreno, e nel dominio del tempo per una valutazione qualitativa dell'andamento temporale del segnale di accelerazione.

Come ci aspettavamo, nei terreni che presentano poche irregolarità si osservano vibrazioni di intensità inferiore e con picchi meno marcati, mentre nei terreni più dissestati si registrano maggiori sollecitazioni, con valori di accelerazione più elevati e un contenuto spettrale più ricco. Questo conferma la sensibilità del sistema nella rilevazione delle condizioni del suolo e la sua efficacia nell'analisi vibrazionale.

5.1 Pavimento liscio

Il terreno in piastrelle rappresenta il caso più regolare tra quelli analizzati. L'analisi spettrale mostra un contenuto in frequenza contenuto e relativamente pulito: le ampiezze spettrali rimangono molto basse, con picchi limitati a circa 0.012 e frequenze concentrate prevalentemente al di sotto dei 25–30 Hz. Questo indica che la macchinetta si muove in modo stabile, con vibrazioni leggere e ben distribuite.

Nel dominio del tempo, il segnale sperimentale mostra una serie di picchi ben definiti ma non particolarmente intensi. Queste vibrazioni sono probabilmente dovute a discontinuità localizzate nella pavimentazione (come le fughe tra le piastrelle). Il confronto con il modello evidenzia una buona coerenza: la risposta simulata riesce a seguire l'andamento generale dell'accelerazione, riproducendo sia la forma che l'ampiezza dei principali eventi dinamici. Ciò dimostra che il modello è in grado di rappresentare fedelmente il comportamento su una superficie regolare, anche in presenza di piccole disomogeneità.

```
Risultati regressione:
Smorzamento c = 0.5313 Ns/m
Rigidezza k = 8.5380 N/m
```

Figura 5.1 - Risultati regressione lineare, Piastrelle

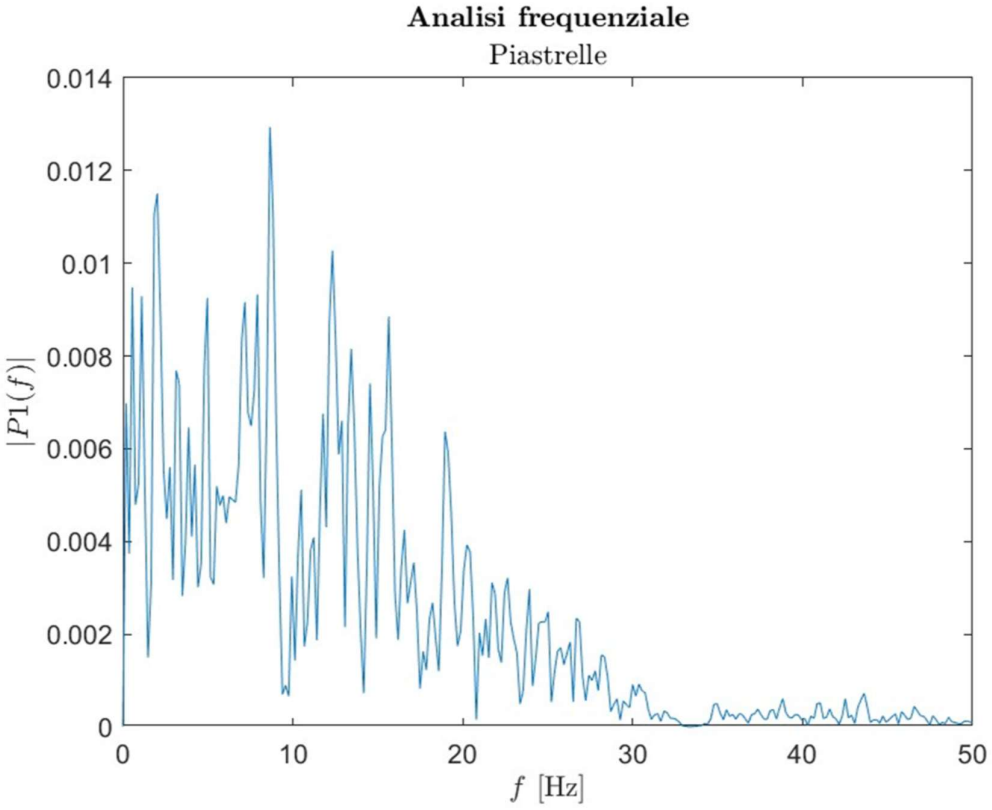

Figura 5.2 - Analisi frequenziale, Piastrelle

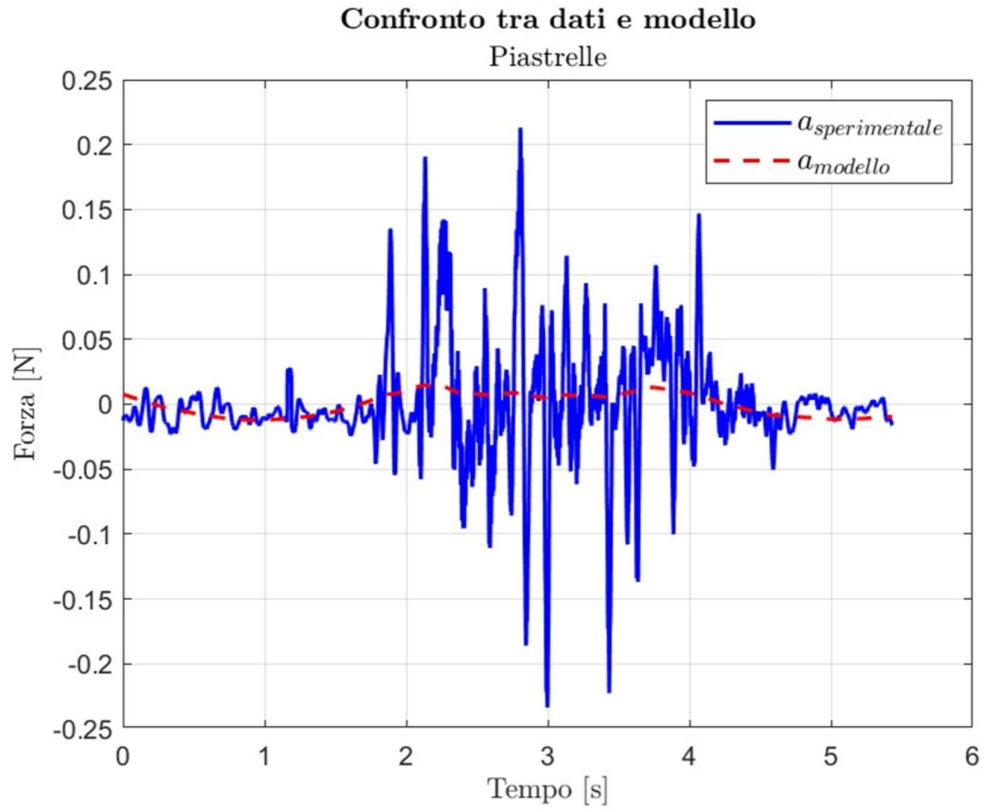

Figura 5.3 - Confronto accelerazione sperimentale e teorica, Piastrelle

5.2 Asfalto

L’asfalto costituisce un’altra superficie regolare, ma con caratteristiche diverse rispetto alle piastrelle. L’analisi frequenziale mostra vibrazioni a bassa intensità, con ampiezze spettrali inferiori a 0.05. Le frequenze significative si concentrano sotto i 25 Hz. Questo conferma che l’asfalto induce un comportamento stabile e prevedibile nella macchinetta.

Il segnale nel dominio del tempo presenta oscillazioni leggere e più smorzate rispetto agli altri due casi. Il confronto tra segnale sperimentale e simulato evidenzia un’eccellente corrispondenza: il modello segue con precisione l’andamento del segnale reale sia in termini di ampiezza che di fase. Le discrepanze sono minime e limitate a brevi tratti. Questo risultato suggerisce che il modello è altamente affidabile per descrivere il comportamento su terreni uniformi come l’asfalto, dove le vibrazioni sono contenute e poco variabili.

```
Risultati regressione:
Smorzamento c = 0.2218 Ns/m
Rigidezza k = 25.7451 N/m
```

Figura 5.4 - Risultati regressione lineare, Asfalto

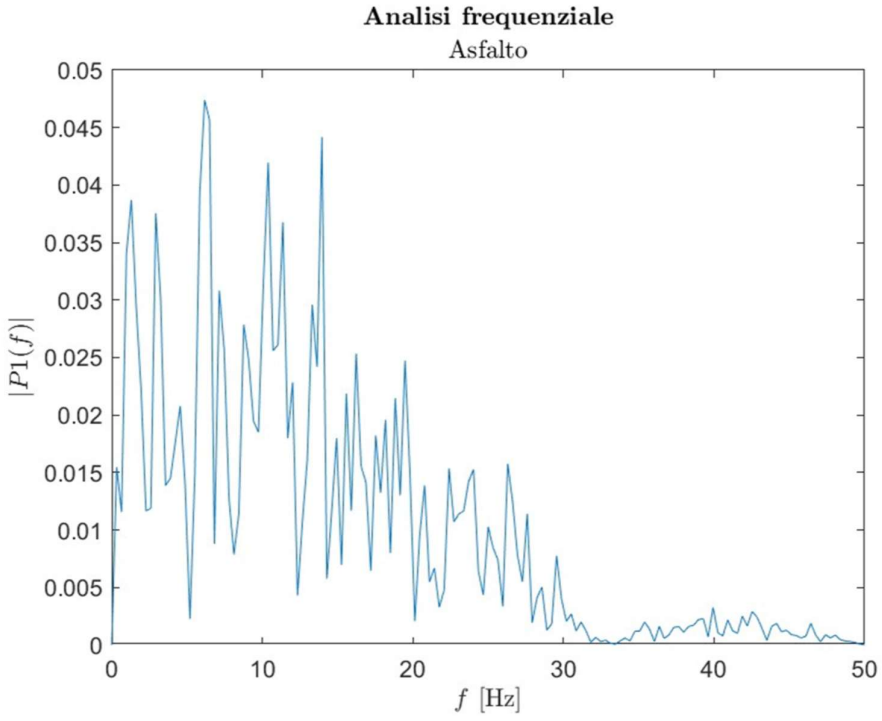

Figura 5.5 - Analisi frequenziale, Asfalto

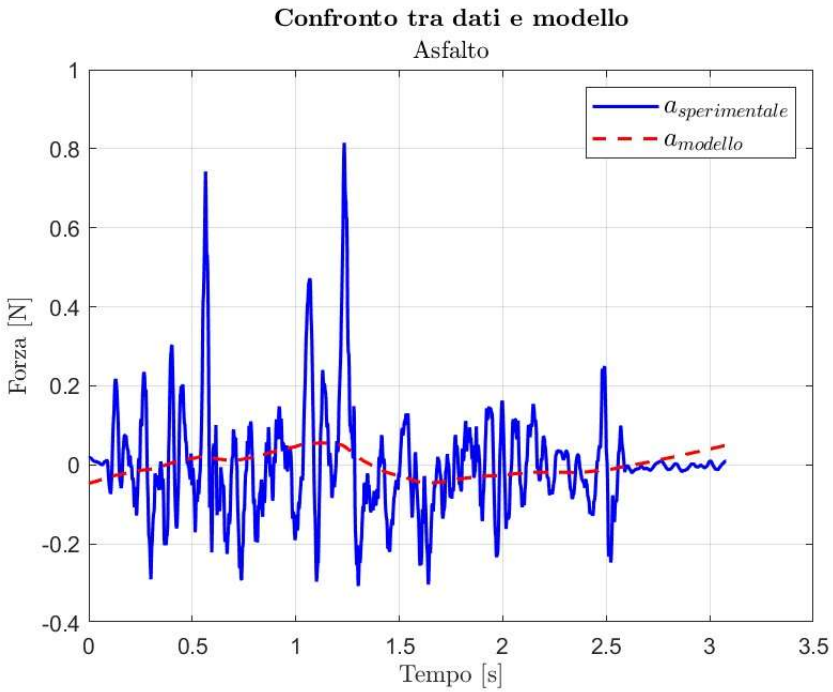

Figura 5.6 - Confronto accelerazione sperimentale e teorica, Asfalto

5.3 Brecciolato

Il brecciolato rappresenta la superficie più irregolare tra quelle analizzate e si comporta come un caso limite per la robustezza del modello. L'analisi frequenziale rivela un contenuto energetico molto più elevato rispetto agli altri due terreni: si osservano ampie ampiezze spettrali, con picchi superiori a 0.1 e un'estensione in frequenza che arriva fino a 35–40 Hz. Questi valori indicano un moto altamente irregolare, generato da urti continui e disomogenei tra le ruote e i frammenti di pietrisco.

Nel dominio del tempo, il segnale sperimentale mostra vibrazioni molto ampie, con variazioni improvvise e poco regolari. Tuttavia, nonostante l'elevata complessità del terreno, il modello è comunque in grado di riprodurre l'andamento complessivo del fenomeno. La risposta simulata segue la forma generale dell'accelerazione, pur non riuscendo a catturare tutte le fluttuazioni locali più rapide. Questo comportamento è comprensibile, in ogni caso, la capacità del modello di mantenere una coerenza di fondo anche su un terreno così sconnesso dimostra la sua buona adattabilità a condizioni reali e complesse.

```
Risultati regressione:
Smorzamento c = 0.1149 Ns/m
Rigidezza k = 10.9896 N/m
```

Figura 5.7 - Risultati regresisone lineare, Brecciolato

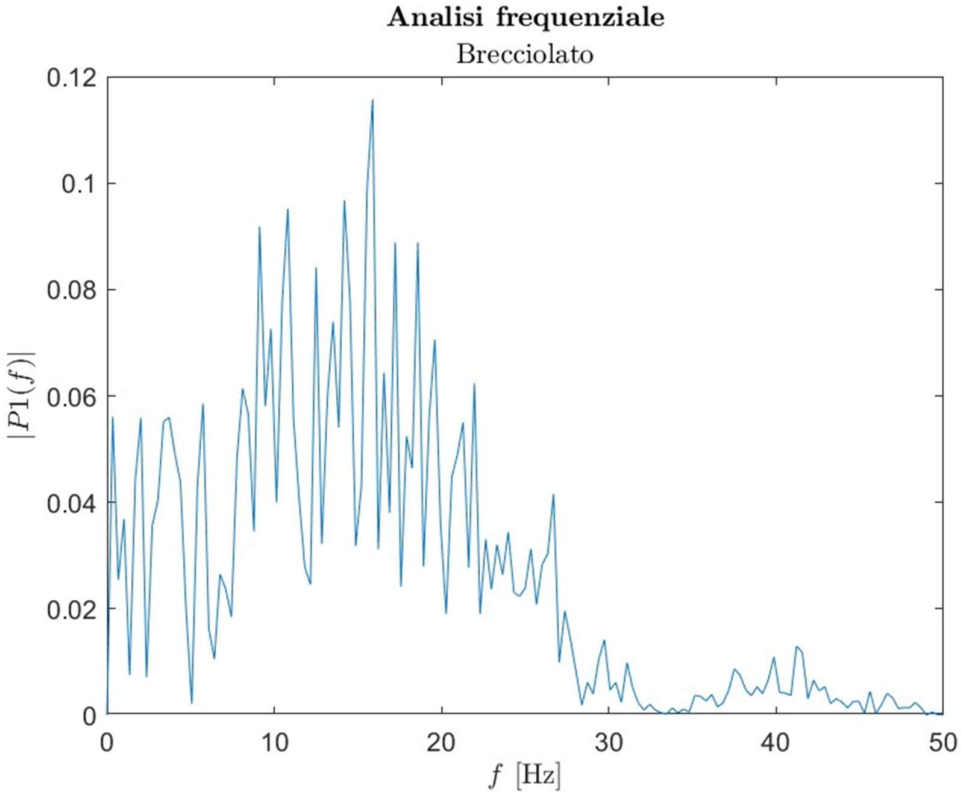

Figura 5.8 - Analisi frequenziale, Brecciolato

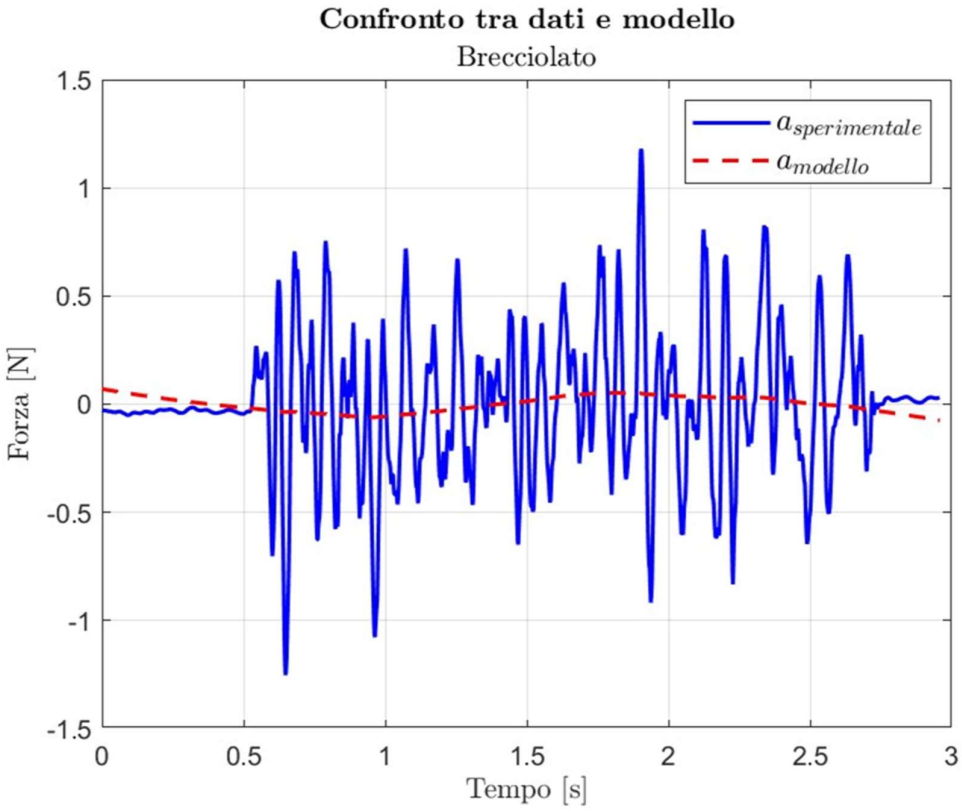

Figura 5.9 - Confronto accelerazione sperimentale e teorica, Brecciolato

6 Simulink

L'equazione del moto che descrive il comportamento del sistema meccanico è

$$m\,\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t)$$

dove m rappresenta la massa, c il coefficiente di smorzamento, k la costante elastica e F(t) la forza esterna applicata. Tale modello è stato implementato in Simulink, un ambiente integrato per la modellazione, l'analisi e la simulazione di sistemi dinamici. Attraverso questa simulazione è stata studiata la stabilità del sistema in risposta a sollecitazioni esterne. I risultati ottenuti hanno dimostrato che il modello si comporta in modo stabile e coerente su tutti i tipi di terreno testati, confermandone l'affidabilità nelle diverse condizioni operative.

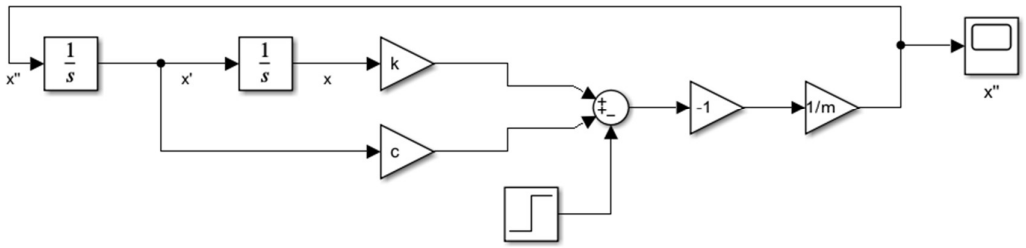

Figura 6.1 - Modello su Simulink

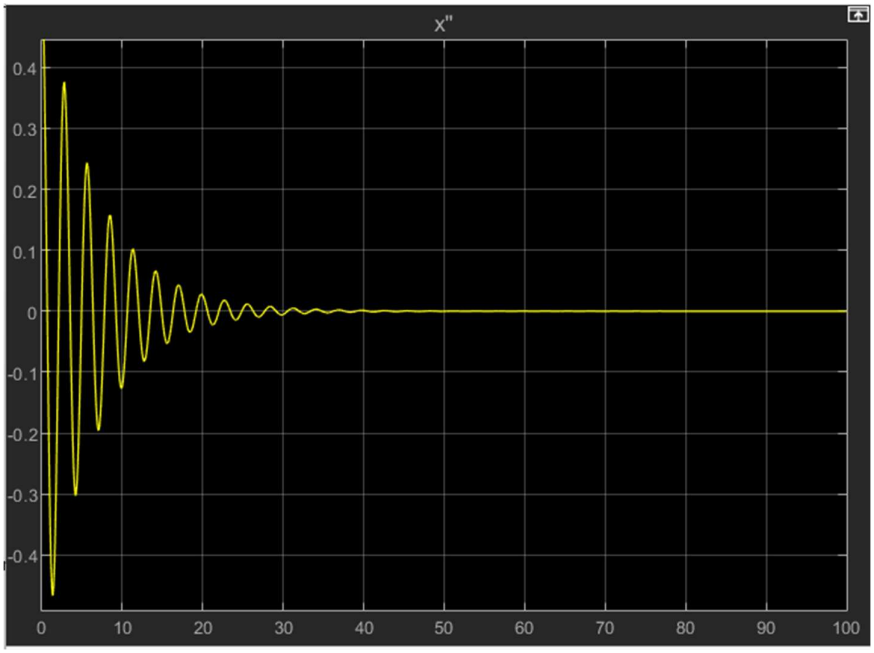

Figura 6.2 - Risposta al gradino, Piastrelle

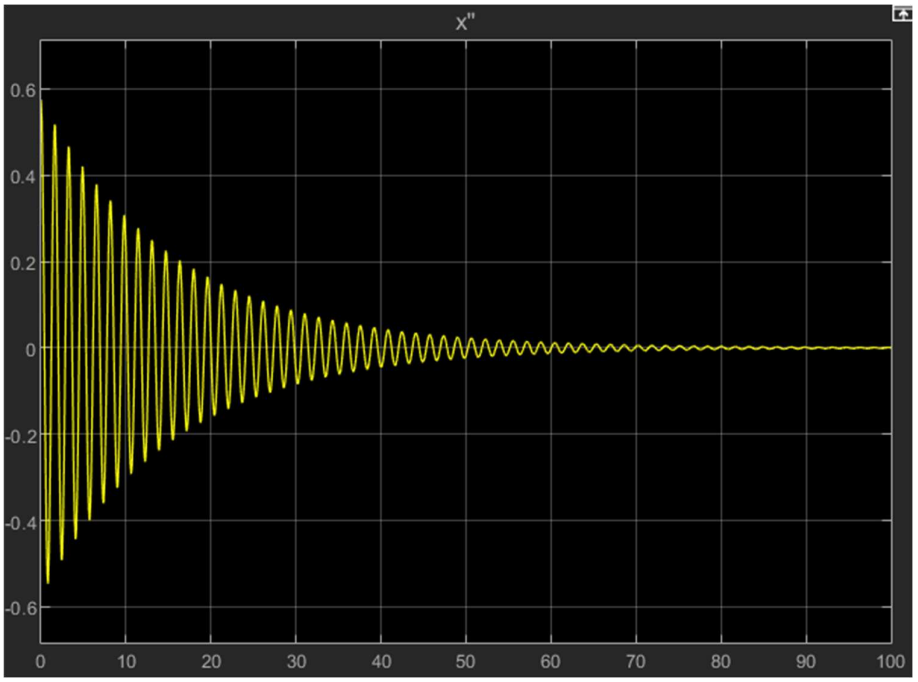

Figura 6.3 - Risposta al gradino, Asfalto

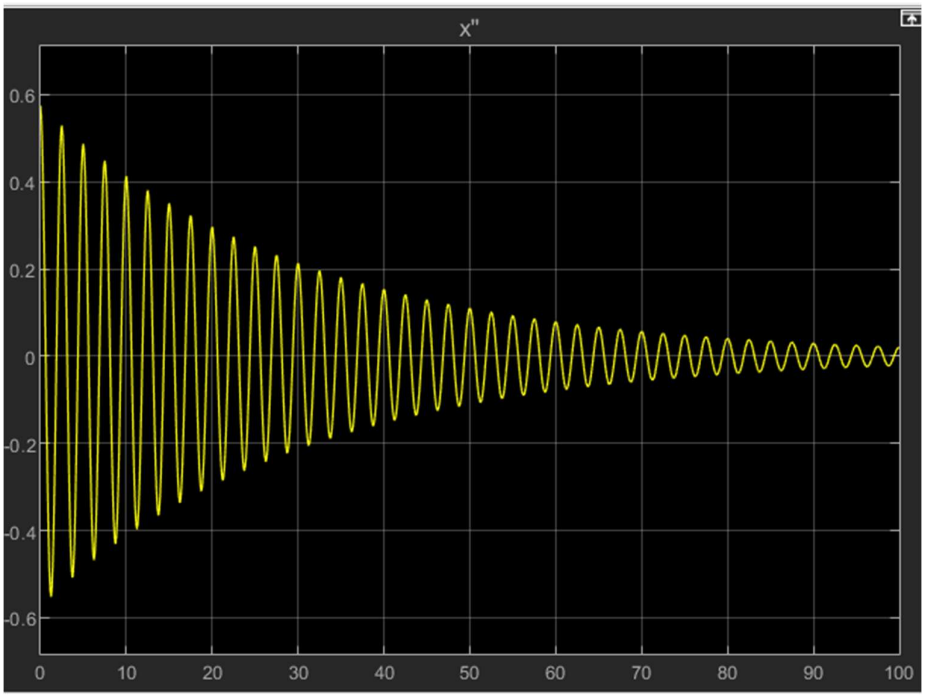

Figura 6.4 - Risposta al gradino, Bracciolato

7 Conclusioni

Il sistema progettato ha dimostrato un funzionamento completo e affidabile in tutte le sue componenti, sia hardware che software. I motori rispondono correttamente ai comandi inviati tramite pulsante o connessione Bluetooth, garantendo un movimento fluido, progressivo e controllato.

Il microcontrollore Arduino ha svolto un ruolo centrale nella gestione dell'intero sistema: il codice sviluppato permette un controllo preciso dei motori, una corretta lettura dei dati dal sensore MPU6050 e una comunicazione efficace con il display LCD e con il modulo Bluetooth. L'elaborazione dei dati avviene in tempo reale, consentendo un'analisi accurata e continua delle vibrazioni.

L'acquisizione dei dati è risultata stabile e affidabile, permettendo una caratterizzazione dinamica dettagliata tramite strumenti come MATLAB ed Excel. Le simulazioni condotte in Simulink hanno inoltre confermato la stabilità del sistema fisico modellato, riproducendo fedelmente il comportamento osservato sperimentalmente.

Il sistema funziona correttamente su diversi tipi di terreno, e ha soddisfatto tutti gli obiettivi progettuali, risultando efficace sia per l'analisi delle vibrazioni che per lo studio del comportamento dinamico di strutture mobili.

8 Bibliografia

Accelerometro:

- slide professore-codici professore;
- https://gzuliani.github.io/arduino/arduino-mpu6050.html;
- howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050 -accelerometer-and-gyroscope-tutorial/

Bluetooth:

- https://youtu.be/zGjD1TbTZmE
- https://techatronic.com/how-to-make-arduino-bluetooth-rc-car/

Matlab:

- Documentazione da MathWorks.com;