

# Pesatura dinamica

## Stima della massa tramite algoritmi di Machine Learning

Giovanni Alghisi, Francesco Campregher, Marco Milanesi, Edoardo Mirandola, Abdelghani Msaad

### I. INTRODUZIONE

La stima della massa in maniera dinamica è un'applicazione attuabile in qualsiasi linea produttiva. Essa consente non solo di aumentare i ritmi produttivi, evitando tempi morti, ma anche avere un costante controllo dei pezzi prodotti. Scopo di questo progetto è lo sviluppo di un sistema predittivo per la valutazione della massa mediante tecniche di Machine Learning. A partire dai dati della cella di carico e tramite un regressore lineare è possibile misurare la massa in transito sul nastro trasportatore.

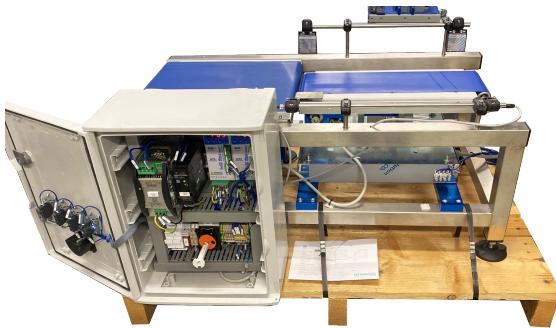


Figura 1: Nastro trasportatore visto di fronte.

### II. STRUMENTAZIONE

La strumentazione utilizzata è la seguente:

- **Nastro trasportatore** del Laboratorio *MMTLab*;
- **2 Fotocellule** Sick GL6;
- **2 Schede di acquisizione**, NI 9237 e NI 9215;
- **Cella di carico** HBM PW22C3 con fondo scala di 10 kg.

La cella è posizionata tra il telaio dell'impianto e il nastro trasportatore, mentre le fotocellule sono posizionate agli estremi di quest'ultimo (vedi Figura 1).

Il setup del sistema di misura consiste nell'utilizzo di un rack di acquisizione della National Instruments (CompactDAQ) al quale sono collegate due schede di acquisizione: una per le fotocellule e una per la cella di carico.

### III. TARATURA-ACQUISIZIONE

Per caratterizzare il comportamento del sistema di misura sono state svolte tre fasi:

#### A. Taratura Cella

La taratura della cella è stata effettuata con 100 valori istantanei di tensione per 20 campioni di massa differenti non riferiti. Essi sono stati stimati utilizzando una bilancia di risoluzione di 0.01 g, tali da coprire il range 10 ÷ 1500 g. La massa dei campioni è stata acquisita in ordine crescente fino al valore massimo e successivamente a descrescere, per includere nel modello l'errore di isteresi del sensore. Il seguente procedimento è stato ripetuto per 5 cicli.

La retta che caratterizza la cella di carico in condizioni statiche è rappresentata dalla seguente equazione

$$y = (5191550 \pm 28 \cdot 10)x - (4874.52 \pm 0.30) \quad (P = 67\%), \quad (1)$$

dove  $y [g]$  rappresenta il peso predetto e  $x [V/V]$  il segnale acquisito. Il relativo grafico è mostrato in Figura 2 con una deviazione standard sulla regressione pari a 2.7 g.

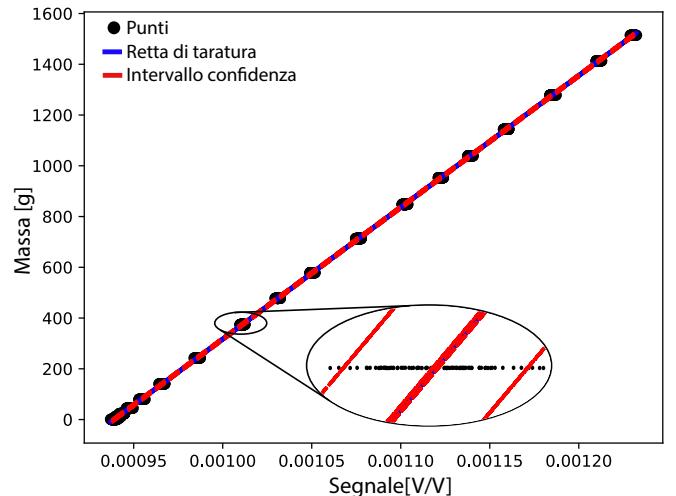


Figura 2: Retta di taratura della cella di carico.

#### B. Taratura Sistema Di Acquisizione

Inoltre è stata effettuata la taratura del sistema di acquisizione tramite media dei segnali acquisiti, in modo da ovviare al problema del rumore elettrico del sensore. In (2) è rappresentata la relativa retta di taratura.

$$y = (5191420 \pm 28 \cdot 10)x - (4874.38 \pm 0.29) \quad (P = 67\%) \quad (2)$$

Il grafico è mostrato in Figura 3 con una deviazione standard sulla regressione pari a 0.38 g.

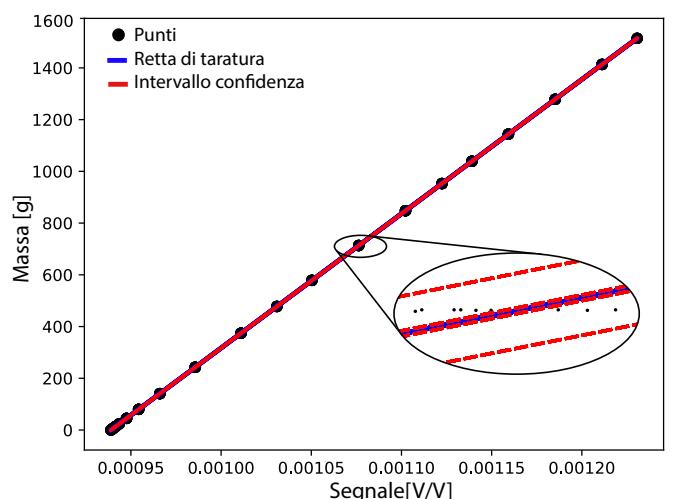


Figura 3: Retta di taratura del sistema di acquisizione.

#### C. Acquisizioni Dinamiche

Sono state considerate 4 velocità, per ciascuna si sono utilizzati 20 campioni per 35 misurazioni consecutive. Per tenere conto dei differenti fenomeni transitori generati dai vari oggetti misurati, si è scelto di utilizzare tre corpi con forme diverse.

#### IV. PRE-PROCESING ED ESTRAZIONE FEATURES

I dati acquisiti mediante LabView vengono pre-processati tramite un filtro a 100 Hz, successivamente analizzati, suddivisi in blocchi uniformi e caricati su un archivio Github.

Il dataset è costituito da:

- **label**, il valore assunto vero ricavato dalle prove di statica iniziale e finale del carico;
- **features**, consistono in N medie calcolate su N intervalli nel tratto utile del segnale e una stima della velocità nel tratto tra le due fotocellule.

Il regressore lineare è stato quindi costruito a partire dal dataset in 2 modalità: inizialmente utilizzando il dataset costituito da sole medie e successivamente effettuando una rielaborazione delle features tramite *Principal Component Analysis* (PCA).

Lo scopo è quello di omettere componenti a bassa variabilità campionaria e porre tutta l'attenzione su quelle con varianza più elevata per migliorare le prestazioni e ridurre il tempo di training e test del regressore.

#### V. STIMA DELLA MASSA

Lo sviluppo dell'algoritmo di stima della massa è stata effettuato tramite regressore lineare implementato con l'utilizzo del pacchetto *scikit-learn*.

##### A. Linear Regressor

La regressione è stata effettuata partendo dal segnale pre-processato e calcolando poi la massa in V/V (Figura 4). Questa scelta è stata eseguita per generare una regressione che sia il più possibile robusta nei confronti dei disturbi esterni.

Il regressore lineare sarà quindi del tipo

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n, \quad (3)$$

dove  $y$  [V/V] rappresenta il peso predetto,  $\beta_0$  l'intercetta,  $\beta_1, \beta_2, \dots, \beta_n$  sono i coefficienti identificati dall'algoritmo e  $x_1, x_2, \dots, x_n$  rappresentano le features del dataset.

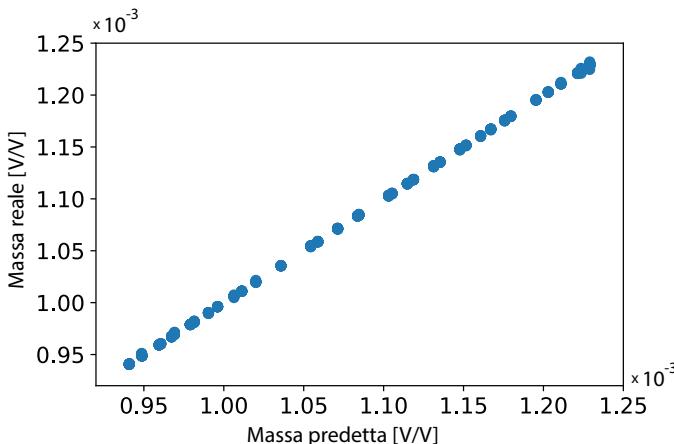


Figura 4: Relazione tra valore reale e predetto.

##### B. Post-processing

Facendo riferimento all'equazione (2) si converte il valore ottenuto dall'algoritmo in grammi. In Tabella I sono mostrati i risultati ottenuti dalla regressione. Per quantificare la bontà del regressore lineare sono stati scelti 2 parametri:

- **coefficiente di determinazione** ( $R^2$ ), indica l'adattamento tra previsione e valore effettivo nell'algoritmo di regressione lineare;
- **Root Mean Square Error** ( $RMSE$ ), valore che rappresenta quanto i risultati previsti deviano dal valore effettivo.

Numero di Features n	$R^2$	$RMSE[g]$
11	1.0	4.6
16	1.0	4.5
21	1.0	4.5
26	1.0	4.5
31	1.0	4.7

Tabella I: Confronto modelli ottenuti tramite regressore lineare.

Prendendo in considerazione il dataset con le migliori prestazioni, quello con 21 features, è stato applicato un algoritmo di PCA. In Tabella II sono mostrati i risultati ottenuti.

Numero di componenti PCA	$R^2$	$RMSE[g]$
5	1.0	6.8
10	1.0	5.6
15	1.0	4.5

Tabella II: Confronto modelli ottenuti in seguito a PCA con n componenti.

#### CONCLUSIONI

Il risultato migliore è stato ottenuto prendendo in considerazione 21 features (vedi Tabella I). Il modello ottenuto è della forma presentata nell'equazione (3), dove:

$$\begin{aligned} \beta_0 &= -0.00169 \pm 0.00059, \quad \beta_1 = 0.0110 \pm 0.0029, \\ \beta_2 &= 0.0159 \pm 0.0050, \quad \beta_3 = 0.0371 \pm 0.0057, \\ \beta_4 &= 0.0433 \pm 0.0072, \quad \beta_5 = 0.0531 \pm 0.0062, \\ \beta_6 &= 0.0672 \pm 0.0064, \quad \beta_7 = 0.0518 \pm 0.0068, \\ \beta_8 &= 0.0392 \pm 0.0074, \quad \beta_9 = 0.0426 \pm 0.0077, \\ \beta_{10} &= 0.0308 \pm 0.0077, \quad \beta_{11} = 0.0445 \pm 0.0069, \\ \beta_{12} &= 0.0471 \pm 0.0084, \quad \beta_{13} = 0.0653 \pm 0.0068, \\ \beta_{14} &= 0.0801 \pm 0.0069, \quad \beta_{15} = 0.0721 \pm 0.0076, \\ \beta_{16} &= 0.0744 \pm 0.0065, \quad \beta_{17} = 0.0868 \pm 0.0091, \\ \beta_{18} &= 0.0584 \pm 0.0092, \quad \beta_{19} = 0.0473 \pm 0.0081, \\ \beta_{20} &= 0.0340 \pm 0.0073, \quad \beta_{21} = -0,00000878 \pm 0.0000013. \end{aligned}$$

La deviazione standard raggiunta dal sistema è di 4.5 g.

L'applicazione dell'algoritmo di PCA non porta benefici. Il motivo potrebbe essere ricondotto alla poca diversità delle features e alla ridotta dimensione del dataset, non portando quindi ad avere un minore tempo di training e una minore deviazione standard.

Inoltre, si è constatato che con l'utilizzo della velocità come feature si ottiene una deviazione standard di 4.2 g. Tuttavia, allo stato attuale, non è presente un sensore per la sua misura accurata.

Si è riscontrato durante la raccolta dati per la creazione del dataset che le misure risultano particolarmente soggette a disturbi ambientali. Rimane infatti aperto il problema di ricercare una maggiore accuratezza dell'algoritmo al variare dei giorni selezionati per la fase di raccolta dati: le misure risultano particolarmente soggette a disturbi ambientali.

In particolare si è individuata una certa dipendenza tra tensione misurata e temperatura. Questo porta ad avere un maggior errore all'interno della retta di taratura. Una possibile soluzione sarebbe quella di inserire la temperatura come feature all'interno del dataset o caratterizzare la cella di carico anche in funzione della temperatura.

Il materiale del progetto è reso disponibile al seguente link: [https://github.com/EdoGitMira/Gruppo\\_H\\_pesatura\\_dinamica](https://github.com/EdoGitMira/Gruppo_H_pesatura_dinamica)