ciao!

**LINK CON TUTTO IL MATERIALE (7GB):** <https://mega.nz/folder/5y5BDTjJ#X6fJIUP4PwCD4_jV8u4Zyg>

# Codice per laboratorio

Nella cartella "*principale - codici interfaccia*" sono presenti i codici più importanti che ho scritto e utilizzato per la mia tesi. Il codice principale è **tesi\_v5**, che richiama le varie librerie quando necessario: *myInterfacciaDAQ, myOutputFile, myTdmsReader, mySignalMaker*.

- il file "**gridfit.m**" è necessario poiché utilizzato all’interno delle librerie, non ho scritto io gridfit.m e ti sconsiglio di analizzarlo a fondo, semplicemente fa una interpolazione molto simile a *ScatteredInterpolant*, è sviluppato da John D'Errico (del MATLAB Staff) tutto il resto invece è opera mia, e commentato spero a sufficienza.

- il file "**truckIcon.png**" è necessario perché l'interfaccia sfrutta l'icona di un furgone per rappresentare la distanza tra i veicoli.

- il file "**Zero.tdms**" non è strettamente necessario, ma era uno dei file di partenza che mi ha lasciato Enrico con i valori a riposo della cella di carico.

Per svolgere delle acquisizioni l’utente deve lanciare **tesi\_v5.m** dopo aver installato il driver NI ed il toolbox aggiuntivo da MATLAB come riportato nelle istruzioni “*Promemoria sulle attrezzature della Galleria Arancio.docx”.*

Tutti i parametri che l’utente è libero di impostare per una simulazione sono nelle parti iniziali di "**tesi\_v5.m**". Ad ogni simulazione vengono prodotti ben 4 file con formato *txt* nella cartella dedicata che l’utente seleziona, ognuno di questi file viene inviato in una sotto-cartella. I 4 file (con le misurazioni di *Load Cell*, *getti*, *microfoni*, e *annotazioni*) hanno lo stesso *timestamp* *(data/ora)* per facilitare il recupero e la consultazione delle informazioni.

Per recuperare velocemente tutti i file di una acquisizione già avvenuta si può lanciare questo codice da terminale:

LibO = myOutputFile carica la Libreria.

s = LibO.txt2struct('') navigare per selezionare uno qualunque tra i 4 file, MATLAB cerca gli altri. Inoltre, si può anche immettere un input specifico per automatizzare il processo.

LibO.ReviveSym(s) replay di simulazione.

LibO.plotMySym(s) plot nel dominio temporale della simulazione.

Per poter leggere le informazioni ed eventualmente plottarle ci sono delle function interessanti

dentro a **myOutputFile.m**>> ( *plotMySym / extractFromFile / ReviveSym* )

# I test svolti

Ho svolto numerosi test con diverse condizioni. Ecco una breve panoramica delle cartelle con i dati acquisiti (circa 7GB di materiale complessivo):

* **Results – Platoon** è una cartella contenente i test di Rear Platooning per 5 diverse velocità e per distanze
* **Results – FullPlat** racchiude i test svolti per Front Platooning (2 convogli) e Middle Platooning (3 convogli) sempre per 5 diverse velocità e per gli stessi parametri . Inoltre, è stato misurato anche l’effetto del forcing (getto costante aperto al ) per in entrambi i casi
* **Results – extra Platoon** è una estensione del platooning a 2 convogli sia front sia read per distanze superiori, sempre per 5 velocità del vento.
* **Results – NewtonForce** è una cartella in cui è stato misurato il segnale della cella di carico a galleria spenta azionando separatamente o simultaneamente i tre getti dalla base del convoglio. Il test serviva per ottenere una stima della forza sviluppata dal getto.
* **Results – RETRY** racchiude l’elenco completo dei test di Platooning & Forcing ottimali. Ogni Algoritmo Genetico ha prodotto una soluzione 🡪 quella soluzione è stata rieseguita e salvata in questa cartella per poter avere una panoramica di tutte le soluzioni ottimali.

# Le immagini

È presente una cartella con l’elenco completo delle immagini, molte delle quali non figurano nella tesi per non appesantire la trattazione. Molte immagini sono state suddivise in base allo scenario di appartenenza che ha caratterizzato il GA in base ai parametri: . Le immagini mostrano alcune performance sul e sullo stato dell’interfaccia.

In aggiunta, è presente una sezione dedicata all’analisi Wavelet del segnale di pressione fluttuante della scia. Tutte le immagini sono state prodotte utilizzando dei codici riportati nella cartella **ausilio – codici per fare plot.**

# Accedere alle informazioni dell’Algoritmo Genetico

Nella cartella **All GA Histories** sono salvate le variabili in formato *\*.mat* contenenti gli individui sintetizzati dal GA e la rispettiva fitness, generazione per generazione, di ogni scenario (solo 1MB).

# Cenni sul programma alternativo in python

Nel caso in cui si voglia utilizzare un metodo di acquisizione in python e plottare i risultati su MATLAB, è presente una cartella dedicata alla lettura dei file in formato *\*.tdms* che LabView produce, **ausilio – TDMS reader**, dal momento che questo formato non ha uno standard di lettura su MATLAB.

Alcune particolarità del codice Python/LabView sono state evidenziate con delle immagini o annotazioni nella cartella **secondario – Codici utili di Enrico e Domenico.** Il materiale completo in Python/Labview invece si trova in **secondario – LabView codici del Laboratorio.**