

Modellazione e Simulazione di Sistemi Fisiologici – Esercitazioni d'esame

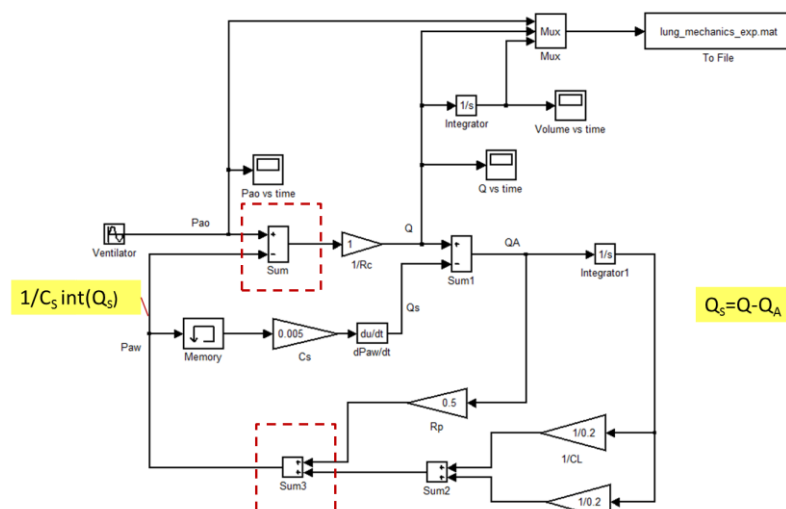
A.A. 2024/2025

Scegliere 4/5 esercitazioni tra quelle proposte

Le esercitazioni vanno consegnate (es via WeTransfer) almeno 48 ore prima della prova orale, accompagnate da un breve report (pdf) che indichi le esercitazioni svolte e segnali eventuali loro parti facoltative/problematiche. (Note: salvare alcune immagini con i principali risultati di simulazione, per evitare di dover far rigirare i codici durante la prova orale).

Esercitazione 1: Meccanica respiratoria - modellazione Simulink

- Implementare in **Simulink** il modello della **meccanica respiratoria** presentato a lezione (modello R_c, R_p, C_s, C_l, C_w) e riportato in figura. Considerare come input di pressione P_{aO} sia il caso sinusoidale sia il caso di un'onda quadra (scegliendo opportunamente i loro parametri). Facoltativamente: arricchire il modello o implementare altri input pressori prendendo ispirazione, per esempio, dal corso di STMT.
- Simulare alcune condizioni patologiche (e.g., asma, enfisema, fibrosi, bradipnea, tachipnea) ed analizzare le differenze con il caso fisiologico. Considerare in particolare l'andamento del flusso Q_A che raggiunge gli alveoli rispetto al flusso totale Q , ed i rispettivi volumi. Predisporre un breve report per il confronto dei risultati.



Esercitazione 2: Meccanica respiratoria – identificazione parametrica

- Modificare i codici Matlab relativi all'**identificazione parametrica** del **modello RLC** della meccanica respiratoria tramite **fminsearch.m**, considerando i **due parametri LC ed RC** anziché i tre parametri R, L e C .
- Completare il codice per l'**identificazione parametrica** con il **metodo di Gauss-Newton** (i.e., scrivere le routines `rlc_fun_two_parameters.m` e `jacobian_fun.m`). Comprendere i dettagli del codice.

Esercitazione 3: Segnali di citometria ad impedenza – curve fitting e compensazione

Integrando quanto fatto in classe con quanto riportato nell'articolo **Errico_Caselli_SAB_2017**, scrivere un codice Matlab che:

- Implementare il **curve fitting** dei dati di impedenza presenti nel file `mystery.mat`, usando come template una Gaussiana bipolare e sfruttando se possibile il `parfor`.
- Implementare la procedura di **compensazione** per rimuovere il *position blurring* dalla stima del diametro elettrico. Visualizzare l'istogramma del diametro elettrico, lo scatter plot "shape parameter vs diametro elettrico", e lo scatter plot "velocità vs diametro elettrico" (in tutti i casi, fare i grafici sia col diametro non corretto sia col diametro corretto).
- Facoltativamente: Usando i comandi `getline` e `inpolygon` e lo scatter plot "shape parameter vs diametro elettrico", individuare i datapoints corrispondenti a ciascuna delle tre popolazioni di beads. Riprodurre nuovamente le figure richieste al punto b) usando stavolta colori diversi per ciascuna popolazione (come in e Fig. 5(a) e (d) di **Errico_Caselli_SAB_2017**). Fittare una Gaussiana sull'istogramma del diametro elettrico corretto di ciascuna popolazione e riportare in una tabella i corrispondenti valori di media, deviazione standard e coefficiente di variazione (cf. Fig. 5(e)).

Esercitazione 4: Riflesso del fuso neuromuscolare e riflesso pupillare

- Ricavare analiticamente la funzione di trasferimento del **modello del riflesso neuromuscolare** ed implementare il corrispondente modello Simulink. Simulare il test clinico applicando una sollecitazione a gradino di ampiezza 5 e diagrammare l'andamento dell'angolo theta. Assumere $J=0.1 \text{ Kg}\cdot\text{m}^2$, $k=50 \text{ Nm}$, $B=2 \text{ Nms}$, $T_d=0.02 \text{ s}$, $\tau=1/300 \text{ s}$, $\eta=5$ e considerare i seguenti tre casi per il valore del guadagno beta: 50, 100, 150, 250 e provare a variare T_d . (Impostare i seguenti parametri per la simulazione: tempo iniziale 0; tempo finale 1; opzioni del solutore: fixed step, ode3, fixed-step size 0.001; tasking mode: SingleTasking). Analizzare la stabilità del modello in funzione del parametro beta (ad esempio col luogo delle radici).
- Implementare in Simulink il modello di Stark del **riflesso pupillare**. Analizzare la stabilità del sistema e provare ad ottenere un "hyppus". Fare riferimento al libro di Khoo per i valori dei parametri.

Esercitazione 5: Sferette a rilascio di farmaco

Seguendo le indicazioni fornite a lezione, realizzare un mini-Toolbox per la simulazione del rilascio di farmaco da un sistema costituito da sferette inizialmente cariche poste in un cilindro. Visualizzare l'andamento temporale della massa di farmaco nel cilindro e il profilo radiale di concentrazione in una delle sferette. La selezione automatica di domini e superfici è molto apprezzata, ma non obbligatoria. Analogamente è molto apprezzata (ma non obbligatoria) la modularità del codice (i.e., la suddivisione in routines: main, datainput, geometry, physics, solver, postprocessing).

Facoltativamente: modellare la presenza di un sottile rivestimento su alcune sferette.

```
R_cyl=3e-2; % [m]
H_cyl=2e-2; % [m]
R_sph=0.3e-2; % [m]
D_liq=1e-6; % [m^2/s]
D_sph=2e-6; % [m^2/s]

% initial concentration in the spheres
C0=3; % [Kg/m^3]

% coating admittance
Y_ctg=0.1*1e-3; % Y=D_ctg/t_ctg
% coating thickness
t_ctg=R_sph/1e3; % [m]
% coating diffusion coefficient
D_ctg=h_ctg*t_ctg; % [m^2/s]
```

Esercitazione 6: Modello minimo del glucosio

- Scrivere un codice Matlab per la soluzione numerica del modello minimo del glucosio (predizione). Implementare inoltre uno schema Simulink equivalente.
- Facoltativamente: Scrivere un codice Matlab per l'identificazione parametrica tramite test IVGTT (utilizzare il comando `lsqnonlin`).

Esercitazione 7: Modellazione citometro ad impedenza (MMT, shell-models, Clausius-Mossotti)

- Modellare i segnali di citometria ad impedenza (i.e., diametro e fase) ottenuti con cellule vitali, necrotiche e apoptotiche (cf. file `dettagli_esercitazione_7`).
- Implementare il modello “double-shell”, graficare `eps_star_mix` e facoltativamente studiare l'effetto delle variazioni dei singoli parametri (cf. file `dettagli_esercitazione_7`).

Esercitazione 8: Modello preda-predatore

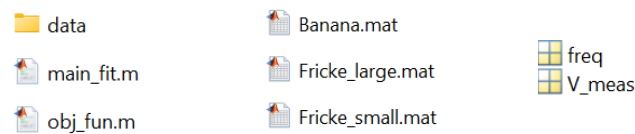
Assumendo $\alpha=1$, $\beta=0.05$, $\gamma=1$, $\delta=0.01$, $x_0=20:40:100$, $y_0=10:10:30$, $t_f=300$:

- Calcolare il punto di equilibrio del sistema (x^*, y^*)
- Calcolare e visualizzare l'andamento delle popolazioni di prede e predatori nel tempo per ogni coppia x_0, y_0 (si consiglia l'utilizzo di `ode45`).
- Generare orbite nello spazio delle fasi (x, y) per ogni coppia x_0, y_0 . Se le orbite non dovessero essere costanti, usare le opzioni di ODE “RelativeTolerance” e “AbsoluteTolerance” per ridurre gli errori numerici fino ad un valore accettabile.
- Calcolare il potenziale ecologico ϕ nel tempo per ogni orbita.
- Calcolare la durata temporale di un ciclo per ogni orbita (usando `findpeaks` per trovare massimi adiacenti).

- f) Calcolare valori medi di $\langle x \rangle$, $\langle y \rangle$ per un ciclo di ogni orbita, e confrontarli con x^*, y^* (si consiglia di far funzionare ODE45 a timestep costante).
- g) Facoltativamente: Per una griglia di valori iniziali x_0, y_0 più fitta ed estesa, generare la superficie del potenziale ecologico ϕ vs x, y e disegnare le orbite sul piano delle fasi x, y come curve di livello del potenziale.

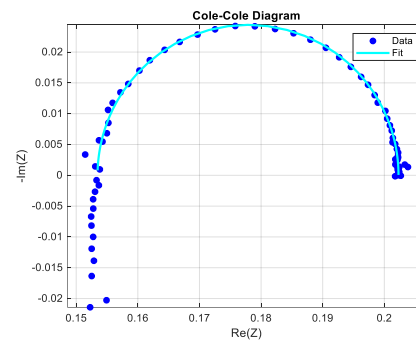
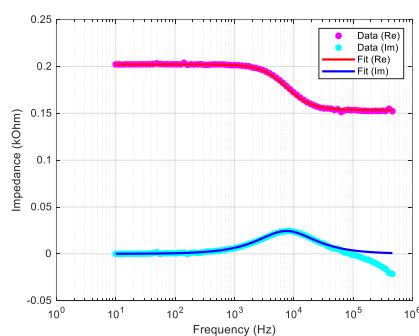
Esercitazione 9: FIT di dati di bioimpedenza

All'interno della cartella **Codici** è presente la cartella **data**, il file **main_fit** e la **funzione obiettivo**. **data** contiene tre file: *Banana*, *Fricke_large* e *Fricke_small*. Ciascun file contiene le **frequenze** e i valori di **Vmeas**.



Dato il file **main_fit** e la **funzione obiettivo**:

- A. **1) Costruire** la **model fun**, utilizzando il **modello di Fricke**, per i dataset *Fricke_large* e *Fricke_small*.
2) Generare i **grafici** che confrontano il **modello fittato** con i **dati sperimentali** sia per la **parte reale** e **immaginaria dell'impedenza**, sia per il **diagramma di Cole-Cole**.
- B. **Modificare** la **model fun** utilizzando il **modello di Cole-Cole** e **fittare** i **dati** relativi al dataset *Banana*, come fatto nel punto **1**.



Esercitazione 10: modello effetto piezoelettrico ed interazione fluido-struttura

A partire dal codice Matlab implementato a lezione per la soluzione numerica del modello dell'attuatore piezoelettrico e dell'interazione fluido-struttura:

- 1) È molto apprezzato, ma non obbligatorio, effettuare la selezione automatica di domini e superfici;
- 2) Simulare uno studio transitorio con istante iniziale 0, un time step di 0.01 s, ed un istante finale dipendente dalla frequenza di attuazione (almeno 1 periodo);
- 3) Generare una routine di post-processing in Matlab, dalla quale sia possibile visualizzare i seguenti risultati: andamento nel tempo dello spostamento verticale del punto centrale (origine del sistema di riferimento) e verifica dell'equazione di continuità:

$Q_{fsi} = Q_{out}$, dove con Q_{fsi} è intesa la portata in corrispondenza dell'interfaccia fluido-struttura e Q_{out} è la portata in uscita.

Facoltativamente generare dei grafici di superficie che rappresentino i valori di pressione nel tempo e le linee di flusso nel dominio fluido (camera di attuazione e outlet).