

UNIVERSITÀ DEGLI STUDI DI VERONA

---

# Bode Plot GUI (2020)

---

*Matteo Marzio*

Guida all'uso e documentazione

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Guida della parte GUI - Parte Front End</b>	<b>3</b>
<b>3</b>	<b>Codice - Parte Back End</b>	<b>5</b>
3.1	Call Backs . . . . .	5
3.2	Funzioni . . . . .	6
3.3	Properties . . . . .	7

# 1 Introduzione

L'applicazione **Bode Plot GUI Remade** é un progetto nato per studiare appieno le potenzialit  di MATLAB, basandosi su una applicazione gi  esistente, scritta da Erik Cheever, ma trascrivendo le funzionalit  base sul nuovo ambiente di sviluppo grafico *App Designer*. L'obiettivo principale   stato quello di comprendere al meglio la programmazione di un applicativo grafico. I sottoobiettivi comprendono una visualizzazione pi  confortevole e meno confusionaria, la rimozione di elementi non utili per la rappresentazione dei grafici e la ri-progettazione del core del sistema per velocizzare la rappresentazione e il calcolo dei dati. Il progetto ha aiutato molto nello studio di MATLAB e delle tecniche per la realizzazione di un applicativo grafico in questo ambiente di sviluppo. La differenza nell'utilizzare *App Designer* invece di *figure*   quella di avere sempre a disposizione il risultato finale in anteprima, la possibilit  di *scaling* delle finestre e delle componenti e di avere un ambiente di sviluppo che sar  aggiornato per molti anni. Come per l'applicazione creata da Erik Cheever, questo applicativo ha le stesse limitazioni. Per maggiori informazioni, consultate il sito Bode Plot GUI di Erik Cheever.

Si ringrazia il professore *Bogdan Mihai Maris* per l'opportunit  che mi   stata assegnata.

## 2 Guida della parte GUI - Parte Front End

**Transfer Function** La funzione di trasferimento è una funzione che caratterizza il comportamento di un sistema dinamico tempo-invariante nel dominio di frequenza. Essa viene definita in MATLAB tramite la funzione **tf**:

$$tf(num, den)$$

Essa crea la funzione di trasferimento a partire dai vettori **num** e **den**, ovvero i vettori (numeratore e denominatore) contenenti i coefficienti delle potenze di  $s$  in ordine decrescente.

Se ad esempio si avesse il vettore numeratore **num** = [3 2 1], esso equivarrebbe a

$$num = 3s^2 + 2s + 1$$

Per poter far funzionare l'applicazione correttamente, bisogna inserire una funzione di trasferimento nel formato

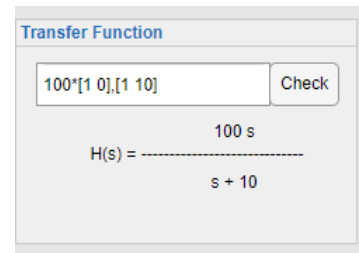
$$transfer\_function = [num, den]$$

all'interno del *TransferFunctionField*.

Il pulsante "*Check*" permette di valutare la funzione, controllando se essa è stata impostata correttamente e mostrando nell'area di testo posta al di sotto se c'è stato un errore durante il calcolo dei dati. Se la funzione sarà valutata correttamente, verrà mostrato all'utente la funzione di trasferimento in forma polinomiale, nella forma

$$H(s) = \frac{[num]}{[den]}$$

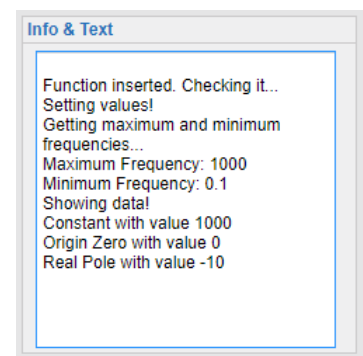
Si ricorda che la funzione di trasferimento inserita deve avere numeri uguali di valori per entrambi i vettori inseriti, altrimenti il programma darà errore. Inoltre, la funzione deve essere valida e formattata correttamente



**Info and Text** È una *TextArea* dove gli utenti possono osservare i vari dati ottenuti e alcune notifiche da parte dell'applicativo in caso di errori. Essa non è modificabile, ma il contenuto del testo può essere copiato per essere eventualmente utilizzato per scopi secondari.

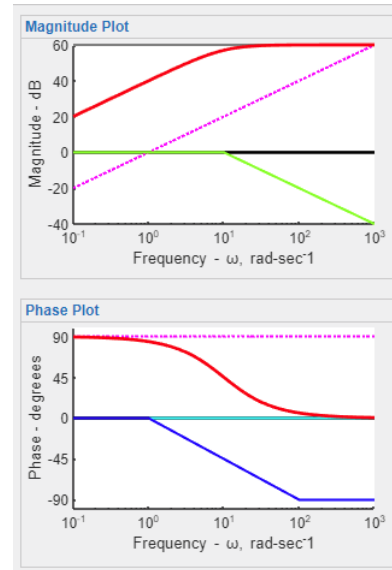
**Options** Permette di mostrare all'utente, se selezionati, la parte *asintotica* e/o la griglia nei due grafici (essa è logaritmica)

**Elements Shown in the Plot** Permette agli utenti di mostrare **tutte** le componenti calcolate (tramite *all*) oppure solo una componente per volta, selezionando rispettivamente la funzione che si vuole osservare. Ogni componente calcolata ha a lato successivamente il valore della stessa.



**Magnitude Plot e Phase Plot** I due grafici permettono di osservare le diverse componenti calcolate graficamente. Nell'asse delle ascisse (x) è presente, per entrambi i grafici, la scala logaritmica, che parte da nella maggior parte dei casi, in  $10^3$ .

Nell'asse delle ordinate (y) nel caso del *Magnitude Plot* la scala è in **decibel**, mentre nel *Phase Plot* la scala è in gradi. Rispetto al programma originale, è possibile scalare i grafici per poter visualizzare meglio le funzioni. I colori dei grafici possono essere modificati, ma solo dal codice dell'applicativo, all'interno delle funzioni *setMagnitudePlot* e *setPhasePlot*. Selezionando un elemento nella sezione *Elements shown in Plot* si può modificare la visibilità del grafico, mostrando solo la funzione dell'elemento selezionato. I **colori** dei grafici possono essere modificati solo tramite la modifica diretta del codice, spiegata successivamente. Le immagini qua a lato hanno dei colori che sono diversi dalla produzione finale del progetto.



## 3 Codice - Parte Back End

Questa sezione é destinata a coloro che si interessano del codice MATLAB scritto e utilizzato per la realizzazione del programma. Per far comprendere al meglio le metodologie utilizzate, sono stati inseriti molti commenti all'interno del codice.

### 3.1 Call Backs

Una *callback* é una funzione che viene richiamata quando avviene un particolare evento all'interno della GUI, come ad esempio il click sopra a un bottone, o la selezione di un elemento di una lista. Sono presenti **sei** callbacks create manualmente e **una** generata direttamente da MATLAB, denominata *updateAppLayout*.

**TransferFunctionCheck** La funzione piú importante, viene richiamata quando il pulsante *Check* viene premuto. Essa ha diversi step, elencati qua sotto:

1. Controlla se é presente una funzione all'interno del form di ingresso
2. Valida la funzione con *eval* e controlla che non ci siano errori di battitura
3. Setta le variabili pubbliche a un determinato valore con *setTransferFunctionValues*
4. Ottiene le frequenze massime e minime della funzione di trasferimento tramite *setMinMaxFrequencies*
5. Calcola la molteplicitá con *setMultiplicity*
6. Mostra i grafici con *setMagnitudePlot* e *setPhasePlot*

Per ogni step, la funzione *writeOnInfo* permette di mostrare i dati su *Info and Text*. In caso di errore che potrebbe bloccare il programma, é presente un **try-catch**.

**ElementsListBoxValueChanged** Quando un elemento contenuto in *ElementListBox* viene selezionato, viene mostrato solo quel elemento nei grafici. Ciò é possibile in quanto nelle funzioni della realizzazione dei grafici viene controllato se é stato inserito un valore (*app.ElementsListBox.Value*). Per questo vengono richiamate le funzioni *setMagnitudePlot* e *setPhasePlot*. Per abbellire il programma, viene mostrato nella box di testo quale elemento é stato selezionato

**GridCheckBoxValueChanged** Se é stata selezionata la checkbox denominata "Grid", ovvero *GridCheckBox*, permette di visualizzare i grafici con una griglia logaritmica

**AsymptoticCheckBoxValueChanged** Se é stato selezionato la checkbox denominata "Asymptotic", ovvero *AsymptoticCheckBox*, permette di visualizzare all'interno dei grafici anche la funzione asintotica.

**GithubMenuSelected** Se la voce *Github*, all'interno del Menu, é stata selezionata, viene aperta una pagina Github.

**DocumentationMenuSelected** Se la voce *Documentation*, all'interno del Menu, è stata selezionata, viene aperta la documentazione pdf (ovvero quella che state leggendo in questo momento).

**updateAppLayout** L'unica funzione di call-back creata automaticamente da MATLAB. Essa non è modificabile con lo strumento *App Designer* direttamente, ma solo tramite la sezione *inspector* (essa viene mostrata se viene selezionata la componente). Questa funzione si occupa di gestire tutti gli eventi che accadono all'interno della applicazione, settando i valori grafici di tutte le componenti, come ad esempio nomi, grandezza, scaling, valori modificati e così via.

### 3.2 Funzioni

Per non creare confusione nella visione del codice, il programma è diviso in più funzioni

**resetProperties** Resetta tutte le variabili pubbliche, evitando quindi dati ridondanti o non conformi alla funzione di trasferimento inserita

**writeOnInfo** Prendendo come valore di ingresso *text*, permette di mostrare una stringa all'interno della box *InfoTextArea*

**setTransferFunctionValues** Setta i valori in base alla funzione di trasferimento inserita. Comprende l'utilizzo delle funzioni *zpkdata*, che ottiene gli zeri, i poli e il gain della funzione, e *tfdata*, che ottiene il numeratore e il denominatore. I valori ottenuti sono poi memorizzati nelle variabili pubbliche

**setMinMaxFrequencies** In base agli zeri, poli e al gain trovati, ottiene la frequenza minima e massima della funzione (solitamente nel range  $[10^{-1}, 10^3]$ ).

**setMultiplicity** Ottiene la molteplicità basandosi sulla tipologia dei dati e, in corrispondenza, sui valori (esempio: la costante avrà molteplicità uguale a 1)

**showElementsOnListBox** Mostra all'utente tutti i valori che sono stati trovati e calcolati, permettendo di selezionarli per farli vedere all'interno dei grafici.

**showTransferFunctionInLabel** Questa funzione viene richiamata quando viene premuto il pulsante *Check*. Mostra all'utente la funzione di trasferimento inserita nella forma  $H(s) = \frac{num}{den}$

**hideAllLegendLabels** Questa funzione nasconde la visibilità dei *labels* contenuti nella sezione *Legend* ogni volta che viene mostrato un grafico. Questi labels saranno poi resi visibili in base ai grafici presenti. È comunque possibile modificare questo fattore commentando il codice.

**setMagnitudePlot** Mostra all'utente il grafico del magnitudo. Essa contiene il calcolo delle funzioni in base alla tipologia e ai valori ottenuti. È possibile modificare i colori delle funzioni (o plots) recandosi nella sezione **switch-case** e modificando la stringa esadecimale RGB all'interno della funzione *plot*.

**setPhasePlot** Mostra all'utente il grafico del magnitudo. Essa contiene il calcolo delle funzioni in base alla tipologia e ai valori ottenuti. È analoga alla funzione precedente, solo che si calcola rispetto all'asse delle ordinate i gradi e non i decibel.

### 3.3 Properties

Le proprietà sono variabili accessibili privatamente o meno, e fanno parte dell'oggetto **app**. In questo caso sono pubbliche, in quanto sono dati che possono essere utilizzati anche per applicativi esterni. Esse, come per le componenti grafiche, sono accessibili e modificabili specificando **app.\***