



Zeno Martini (admin)

IMPARIAMO A SIMULARE

4 July 2007

E' BENE SIMULARE

No, non è l'insegnamento da impartire ad un attaccante per tuffarsi in area di rigore ingannando l'arbitro. Intendiamo simulare un sistema fisico, il che significa costruire un modello, nel nostro caso mediante computer, quindi, come si dice, virtuale, che abbia un comportamento simile al sistema fisico reale.

Lo scopo della simulazione non è dunque l'inganno, ma l'implementazione di modelli in grado di farci vedere come funziona un sistema reale.

Essa ci permette, ad esempio, di capire se ciò che abbiamo progettato eseguirà effettivamente in modo corretto la funzione per cui è stato pensato. La risposta definitiva l'avremo dal sistema reale, ma se il modello l'avrà ben simulato le differenze saranno contenute entro i limiti di precisione delle apparecchiature utilizzate e di cui il modello può tenere conto.

E' dunque fondamentale sapere impostare i modelli per la simulazione.

Questo articolo illustra come si può fare utilizzando un simulatore eccellente: il computer.

DESCRIVERE E TRASFORMARE

Ogni linguaggio è un codice immateriale che può descrivere ciò che ci circonda. Il linguaggio della scienza e della tecnica è la matematica. E' traducendo i fenomeni fisici in oggetti matematici che è possibile creare modelli della realtà.

Analizzare matematicamente la dinamica di un sistema è in genere molto complicato. Le relazioni tra le grandezze in gioco, variabili nel tempo, non sono esprimibili con maneggevoli equazioni algebriche, ma con equazioni che coinvolgono le derivate rispetto al tempo delle grandezze anche di ordine elevato, cioè equazioni differenziali. Risolverle direttamente per trovare il vero andamento delle grandezze nel tempo, non è compito agevole, nemmeno per i matematici. I quali però hanno inventato metodi per bypassare il problema, se così si può dire, o più tradizionalmente, trasformarlo, dare ad esso una forma matematica che consenta una trattazione più semplice.

La trasformata di Laplace, ad esempio, è uno strumento particolarmente efficace: le equazioni differenziali diventano equazioni algebriche. Poiché in generale come variabile algebrica si usa s , un numero complesso, si dice che il problema è studiato nel dominio di s .

Non è che in questo modo analisi e sintesi diventino improvvisamente banali. Ciò che occorre conoscere davvero è, come già detto, l'andamento delle grandezze nel tempo. Quindi trovate le soluzioni nel dominio di s , si deve antitrasformare, cioè trovare le grandezze nel dominio del tempo. Cosa magari agevole per semplici sistemi ma che può diventare proibitiva per sistemi appena un po' complessi. Ecco allora che ci soccorre la potenza di elaborazione del computer che può fornire i risultati sotto forma di grafici. E' necessario "solo" costruire un modello del sistema che rappresenti in qualche modo grandezze e relazioni tra le stesse in modo che possano essere calcolate istante per istante.

L'insieme dei valori che le grandezze assumono nel tempo, può essere visto come una tabella in cui ogni riga è individuata dall'istante ed ogni colonna rappresenta una data grandezza. Naturale quindi ricorrere ad un software la cui struttura fondamentale è la matrice.

Diagrammi a blocchi con SCICOS

Il modello completo di un sistema è bene suddividerlo in parti tra loro interconnesse. Tali parti sono dette blocchi. Un diagramma a blocchi è una rappresentazione grafica in cui ogni componente è rappresentato come una scatola chiusa, il blocco appunto, che svolge una precisa funzione.

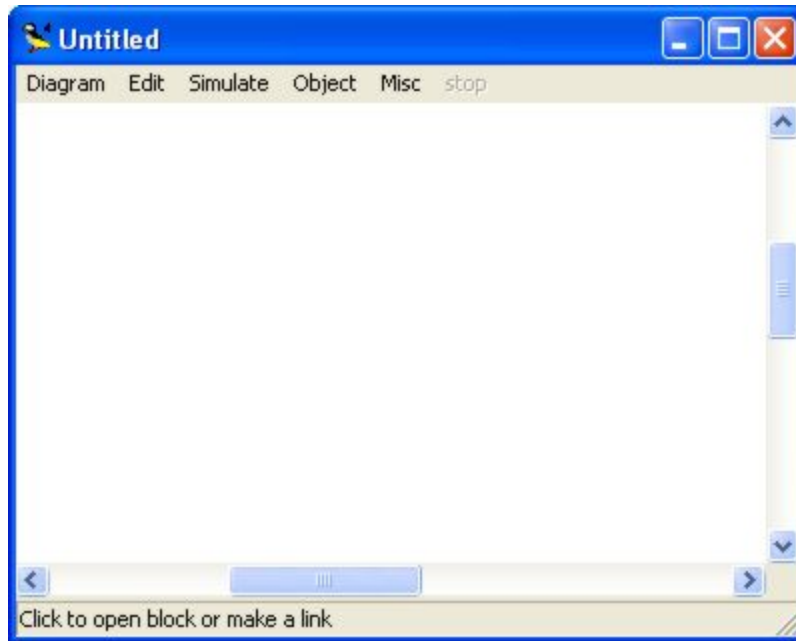
Può generare autonomamente un segnale, visualizzarlo, o fornire un'uscita che dipende dall'ingresso secondo una relazione algebrica chiamata funzione di trasferimento.

I blocchi sono collegati tra loro da segmenti la cui orientazione indica il verso del segnale di cui essi sono la canalizzazione.

Quanto detto può essere realizzato utilizzando l'applicazione SCICOS interna al programma di matematica SCILAB 4.1. (Come già detto in un [precedente articolo](#), **Scilab** è un programma gratuito scaricabile da www.scilab.org). Costruito il diagramma utilizzando l'editor di scicos, un semplice comando dà il via alla simulazione, cioè ai calcoli da parte del computer, che rappresenterà nei grafici richiesti i risultati.

Scicos si lancia dall'interno di Scilab selezionando (o scrivendo) il comando **scicos** dal menù **Applications**.

Si apre la seguente finestra vuota dove si può tracciare il diagramma a blocchi.



Per capire come, costruiamo un diagramma per tracciare una sinusoide.

Il *sinusoid generator*, che si sceglie cliccando sul blocco che vi corrisponde nella finestra che si apre dal menù *edit->palettes->source*, è dotato di un'uscita nella quale emette la forma d'onda di una sinusoide, di cui si possono stabilire ampiezza, pulsazione e fase in una finestra di dialogo che si attiva cliccando sul blocco stesso. Esempio:

- magnitudo: 325;
- Frequency: 314;
- Phase:0.

Per vedere il segnale occorre un blocco di visualizzazione (scope), scelto come il precedente blocco nella finestra che si apre dal menu' *edit->palettes->sinks*.

Questo blocco lo si può immaginare come una macchina fotografica che scatta una foto del grafico della grandezza che riceve ad intervalli di tempo regolari. La loro durata è denominata *Refresh period*, ed è stabilita in una finestra di dialogo, sempre attivata cliccando sul blocco. In essa è anche possibile definire altre caratteristiche per il grafico, tra cui i valori massimi, positivi e negativi, previsti sull'asse delle ordinate che, nel caso specifico, saranno maggiori dell'ampiezza del segnale sinusoidale che si desidera visualizzare. Per l'esempio sarà:

- $Y_{\max}$:350
- $Y_{\min}$:-350
- Refresh period: 0.1

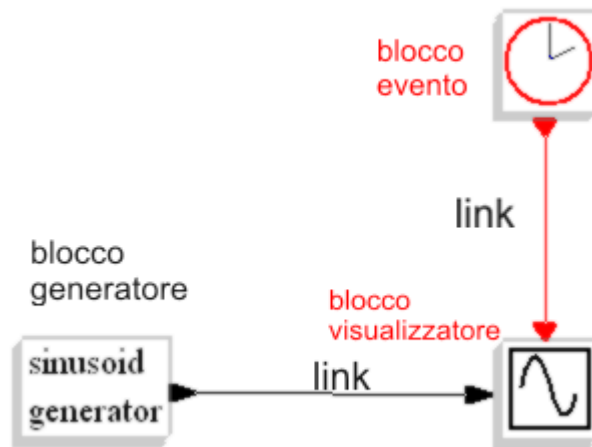
Il segnale sinusoidale va campionato per la visualizzazione. Per questo motivo il blocco di visualizzazione, oltre all'ingresso cui sarà collegata l'uscita del blocco generatore, è dotato di un ulteriore ingresso che riceve la temporizzazione per il campionamento. Questo segnale è generato dall'orologio, il blocco evento scelto dal menu *edit->palettes->events*. La temporizzazione è stabilita nella finestra di dialogo del blocco. Occorre ovviamente rispettare il teorema del campionamento di Shannon e, nel caso specifico, si sceglierà un periodo che sia un molto inferiore a quello del segnale per averne una rappresentazione fedele. Nell'esempio:

- Period: 0,0002;
- Init time:0

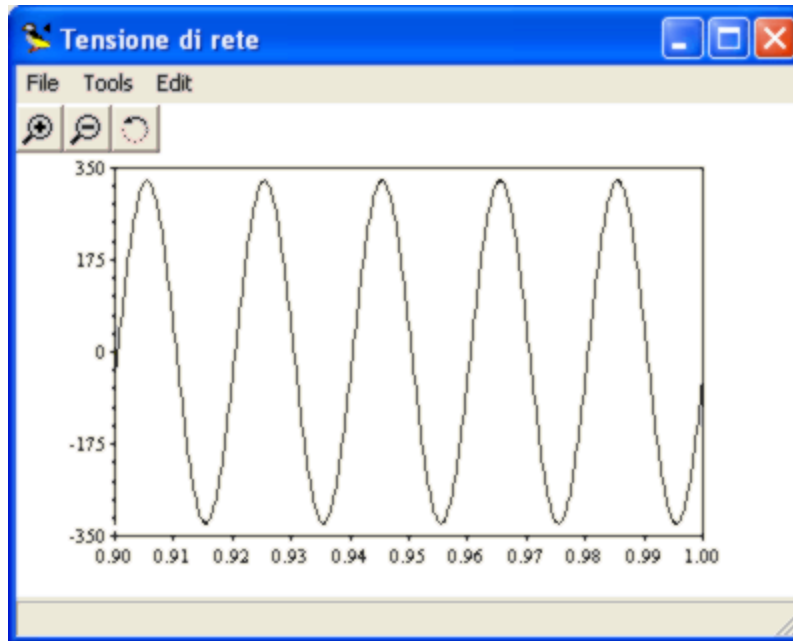
A questo punto si effettuano i *links* scegliendo la funzione dal menu *edit->link*. Quindi si clicca sull'uscita di un blocco e l'ingresso dell'altro che vi deve essere connesso. Ora il diagramma è completato e si può lanciare la simulazione selezionando il comando *Run* dal menu *Simulate*.

Occorrerà però stabilire dapprima le modalità della simulazione con la finestra di dialogo scelta dal menu *editor->Setup*. Tra queste la durata:

- Integration time: 1.



Il risultato è il seguente diagramma



In pratica è mostrato il decimo fotogramma. La durata della simulazione è stata stabilita infatti di un secondo ed ogni decimo di secondo il blocco *Scope* scatta una nuova istantanea. Il nome al grafico(nel caso specifico: Tensione di rete) è stabilito nella finestra di dialogo del blocco *Scope*, già menzionata.

Il nocciolo delle possibilità di simulazione che si può conseguire con Scilab 4.1 è tutto qui. I blocchi sono comunque numerosissimi e si possono anche costruire ex-novo. Ovviamente le cose si complicano ma, d'altra parte, non potrebbe essere altrimenti visto che si devono rappresentare sistemi comunque complessi.

Sistemi del Primo Ordine

Quando in un sistema che elabora un segnale per attivare un attuatore sono presenti elementi che accumulano energia, l'uscita segue obbligatoriamente con un certo ritardo l'ingresso. Se l'entità del ritardo può essere misurata da un unico parametro denominato costante di tempo, **t**, il sistema si dice del primo ordine. Esso può in tal caso essere rappresentato da un blocco descritto matematicamente nel dominio di *s* da una frazione del tipo:

$$F(s) = \text{Num}(s) / \text{Den}(s) = K / (1 + s \cdot t)$$

K è detto *guadagno statico*: è il valore per cui bisogna moltiplicare l'ingresso per ottenere l'uscita a regime.

Immaginiamo di applicare ad un tale sistema un segnale a gradino, nullo fino all'istante zero e diverso da zero e costante subito dopo.

Supponiamo che la costante di tempo sia **t=3** sec ed il guadagno sia **K=10**.

Costruiamo il diagramma in Scicos scegliendo nel modo in precedenza illustrato i

blocchi. Il blocco della $F(s)$ va scelto in *edit->linear* scrivendo la $F(s)$ nella finestra di dialogo. Quindi, per l'esempio:

- Numerator (s) = 10;
- Denominator (s) = $1+3s$

Il blocco gradino, scelto in *palettes->sources* lo consideriamo di ampiezza unitaria. Quindi nella finestra di dialogo:

- Initial value:0;
- Final Value:1

Poiché desideriamo visualizzare sia il gradino che l'uscita dal blocco $F(s)$ sceglieremo in *palettes->sinks*, il blocco multiscopo (*Mscope*) settando i valori massimi previsti che dovranno essere maggiori di 1 per il gradino e di $k=10$ per l'uscita. I valori minimi sono entrambi nulli. Nella finestra di dialogo tali valori vanno scritti in sequenza rispettando l'ordine degli ingressi. Il numero di questi va scritto nella casella Input port size, ad esempio, nel nostro caso, inserendo due 1 consecutivi:

- Input port size: 1 1
- Ymax= 2 15
- Ymin: 0 0

Si sceglie quindi il blocco per la temporizzazione, stabilendo per il periodo di campionamento un tempo che sia, poniamo, un decimo della costante di tempo e facendola partire dall'istante zero: quindi

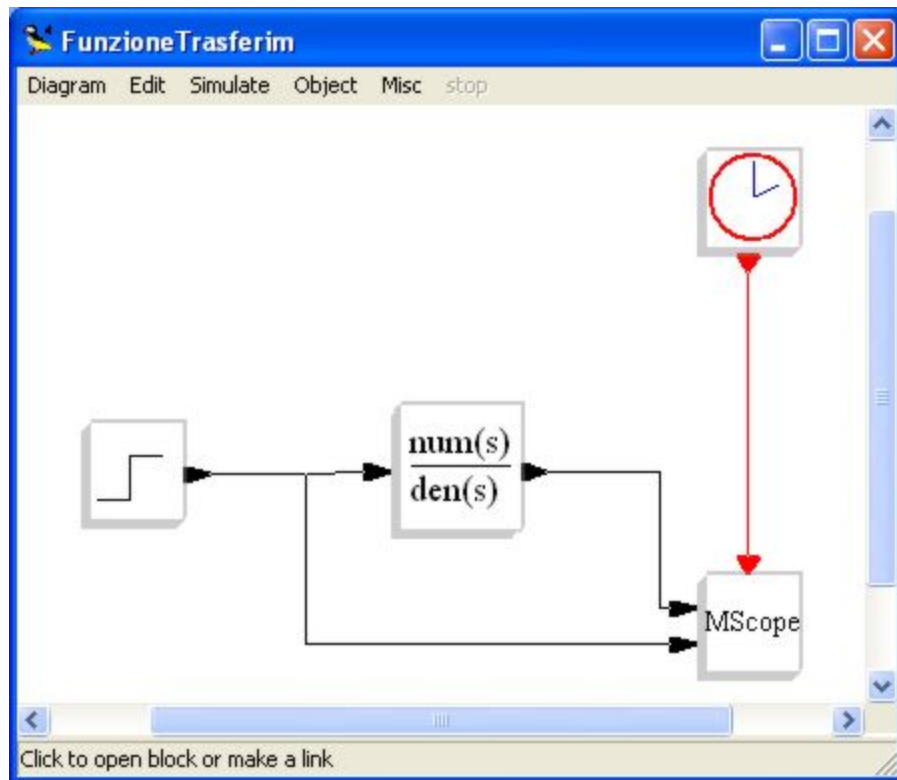
- Period=0,3;
- init time:0.

Per il Setup della simulazione sceglieremo un tempo maggiore di 5 volte la costante di tempo, durata pratica del transitorio. Quindi:

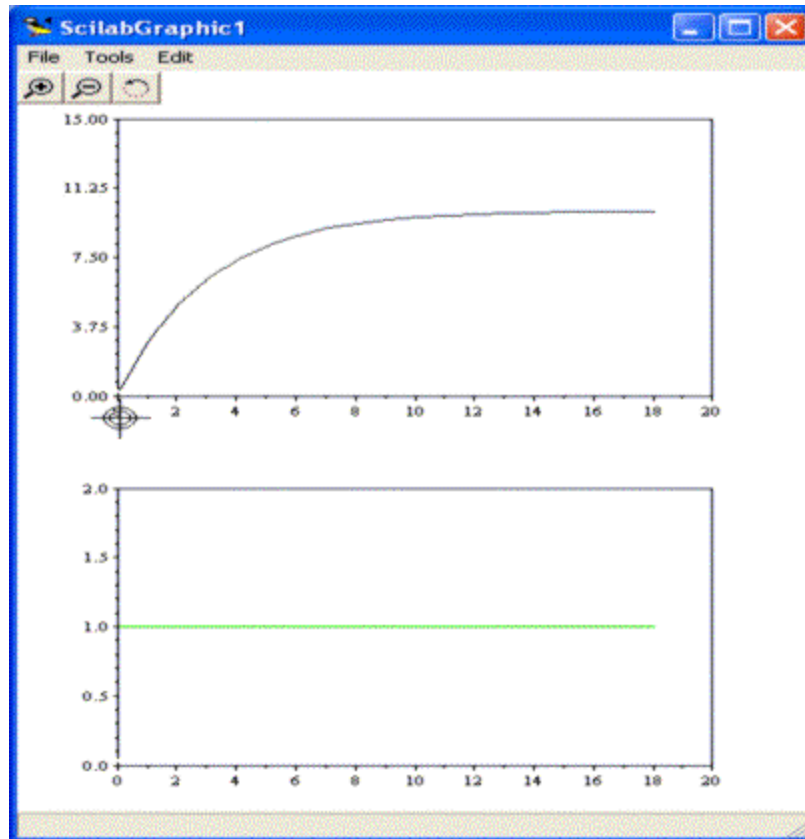
- Final Integration Time: 20.

Lo stesso tempo lo sceglieremo per il Refresh di *Mscope*: nel caso specifico esso eseguirà un'unica istantanea.

Eseguiamo ora i link come fatto nel precedente esempio ed ecco il diagramma a blocchi ottenuto



Infine si fa partire la simulazione con il comando *Run* del menu *Simulate*.
Questo il risultato che si ottiene.



L'uscita ha il noto andamento esponenziale che, come elettrici conosciamo bene, della tensione di carica di un condensatore C alimentato da una tensione costante attraverso una resistenza R per il quale la costante di tempo è $t=R*C$, o della corrente in un circuito R-L, alimentato da un generatore di tensione costante, dove la costante di tempo è L/R .

Sistemi del secondo ordine

Concludiamo questa breve introduzione ai metodi di simulazione dei sistemi analizzando un sistema del secondo ordine. Ciò che cambia rispetto al paragrafo precedente è semplicemente la funzione di trasferimento che ora può essere scritta nella forma:

$$F(s) = K' / (w^2 + 2*x*w*s + s^2)$$

E' caratterizzata da tre parametri

- **w**: pulsazione naturale = $2\pi*f$ con f: frequenza in Hz
- **x**: smorzamento

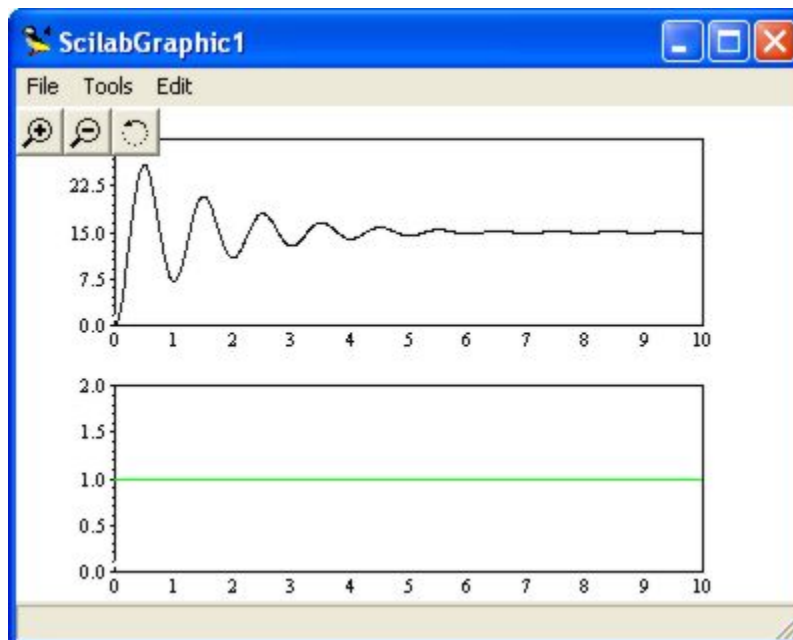
- **K'**: costante che, divisa per ω^2 fornisce il guadagno statico **K**=**K'**/ ω^2

I parametri sono legati a loro volta ai parametri del sistema direttamente collegati ai fenomeni energetici che in esso avvengono. In particolare la pulsazione naturale dipende dai componenti che immagazzinano energia (induttanze, capacità, masse), lo smorzamento dai fenomeni dissipativi (resistenze elettriche o di attrito meccanico). Quando lo smorzamento è maggiore di 1 ($\alpha > 1$) l'uscita non presenta oscillazioni ed il sistema può essere scisso in due sistemi del primo ordine caratterizzati da due costanti di tempo distinte. Si può cioè scrivere

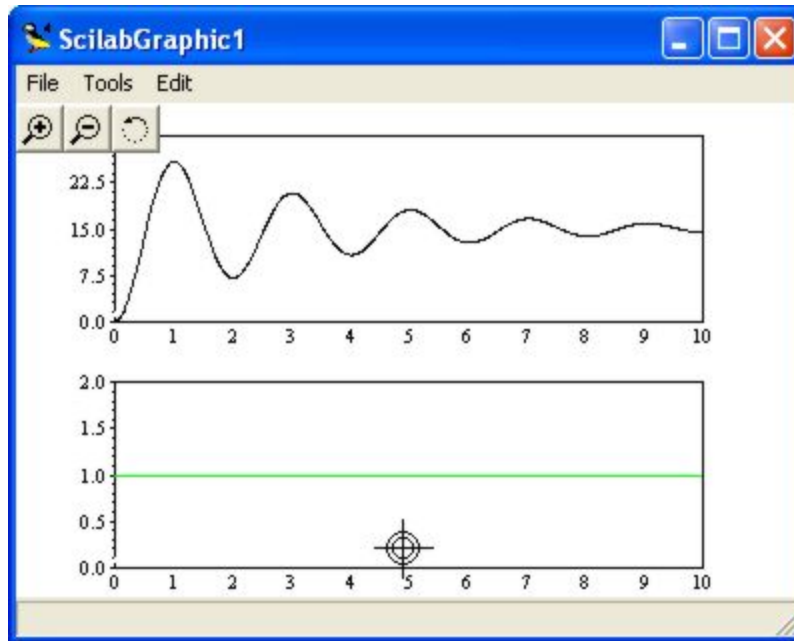
$$F(s) = k' / (1 + s \cdot t_1) \cdot (1 + s \cdot t_2)$$

L'andamento dell'uscita è simile a quello di un sistema del primo ordine. La costante di tempo più influente è quella di valore maggiore e, al limite, se sono molto diverse, la più piccola può essere trascurata. Frequentemente lo smorzamento è compreso tra 0 ed 1 (Nb: nei sistemi fisici reali lo smorzamento è sempre maggiore di zero). In questo caso l'uscita, prima di assestarsi sul valore di regime costante, presenta delle oscillazioni, tanto più pronunciate e prolungate quanto minore è lo smorzamento.

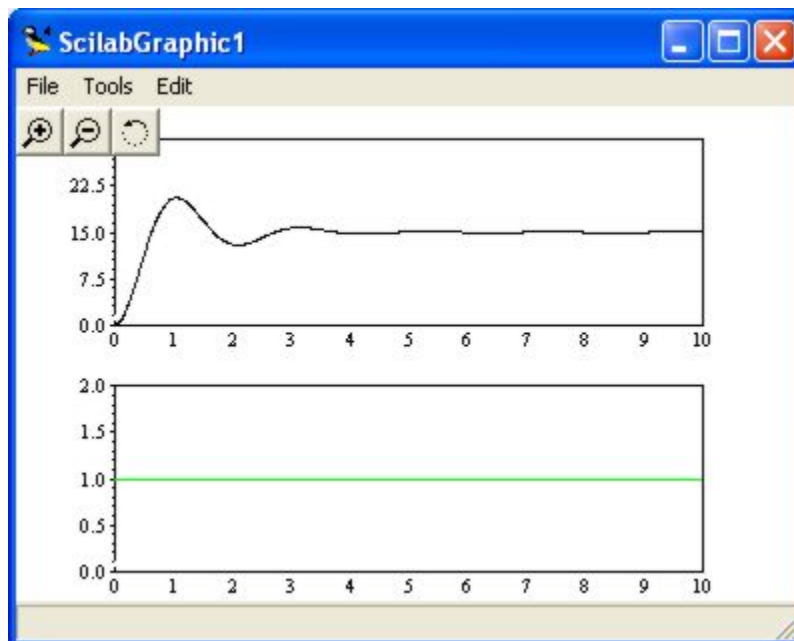
Le figure seguenti mostrano i grafici ottenuti variando i parametri. E proprio in questa semplicità di variazione dei parametri sta la potenza della simulazione che ci permette di capire su che cosa e come intervenire per ottenere le risposte desiderate.



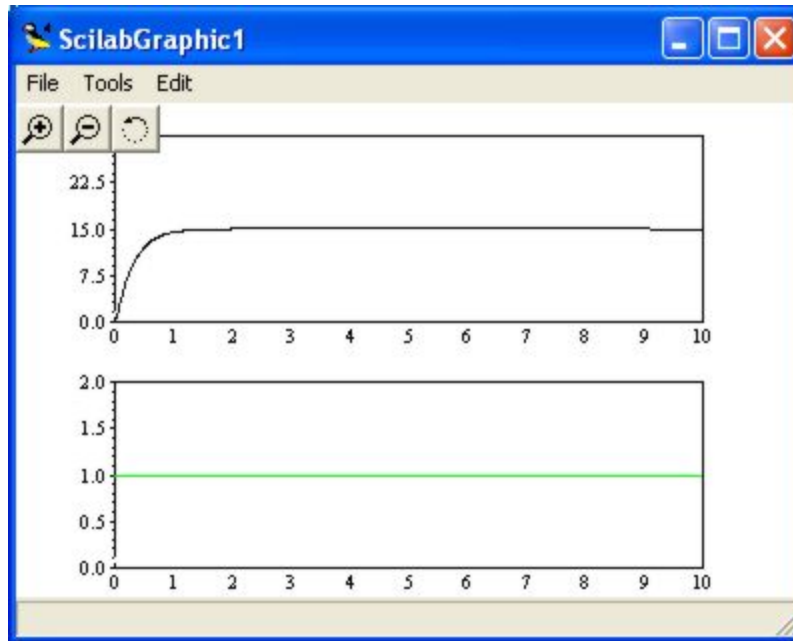
$$K=15; f=1 \text{ Hz}; \alpha=0.1$$



$K=15$; $f=0.5$ Hz; $x=0.1$



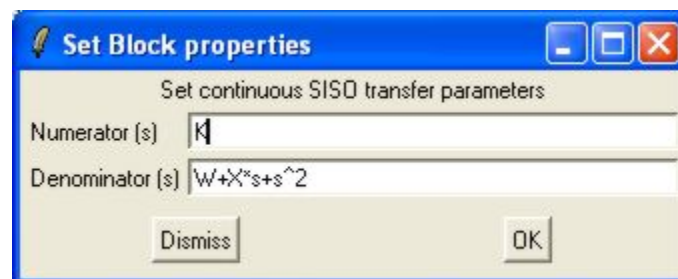
$K=15$; $f=0.5$ Hz; $x=0.3$



K=15; f=2 Hz; x=2.

Note

I valori dei parametri si modificano prima di dare il comando *Run*, scegliendo il comando *Eval* dal menu simulate. La $F(s)$ scritta nel blocco è così



Invece dei valori numerici come coefficienti e termini noti ci sono dei parametri che vengono calcolati con Eval, utilizzando lo script esistente nel Context. La finestra relativa è attivabile con *Edit -> context*

- $K = \text{evstr}(\text{x_dialog}(\text{'Guadagno statico?'}, '15'))$;
- $W = \text{evstr}(\text{x_dialog}(\text{'Frequenza naturale?'}, '0.8'))$;
- $W = 2 * \%pi * W$;
- $X = \text{evstr}(\text{x_dialog}(\text{'Smorzamento ?'}, '0.1'))$;
- $X = 2 * W * X$;
- $W = W^2$;
- $K = K * W$;

Le applicazioni Scicos si possono scaricare [qui](#).

Sono i file

- Sinus.cos
- Fdt_sec_ord.cos

Dopo averli scaricati li si attiva in Scicos dal menu *Diagram->Load* (ovviamente avendo installato Scilab 4.1 e lanciata l'applicazione Scicos)