A Matlab-Simulink Package for a Command Governor Experiment

Francesco Tedesco

Dipartimento di Elettronica Informatica e Sistemistica
Università della Calabria
Via P. Bucci, 42c - 87036 Rende (CS), Italy
`casavola@dsi.unifi.it`, `http://www.deis.unical.it/`

Abstract

This document describes a Simulink implementation of a basic Command Governor strategy depicted in next Fig.1 for a position electrical servomechanism .

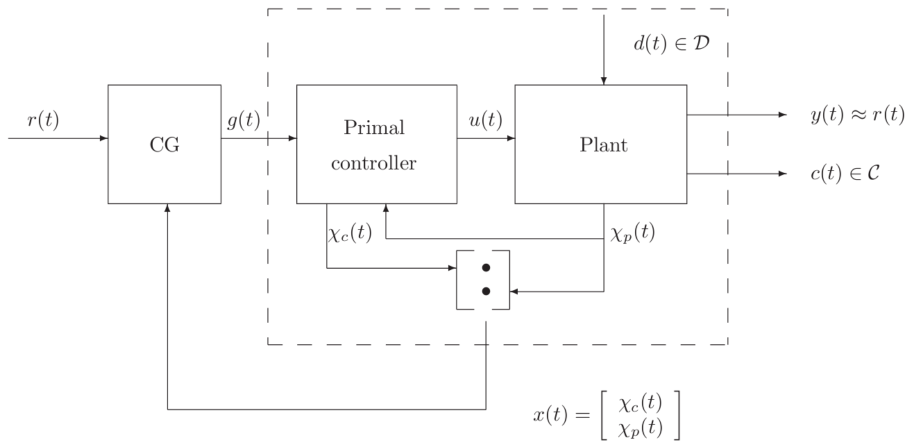

Figure 1: Control Scheme

1 Environement Usage

The CG package consists of the directory CGENV that contains two Matlab functions:

- CGOffLineSetting.m
- CG.m

two Matlab scripts:

- main.m
- makesystem.m

and a Simulink file:

- simulator.mdl

The CG implementation is based on the CG theory and implementation details described in [1]. This document is provided in the DOC directory of the package. It is strongly recommended to read this tutorial first if one wants to change the example and/or modify some parameters.

1.1 Starting the simulation

The simulation starts by running the matlab script *main.m* Matlab main window. That's all! One has to be sure that the CGENV directory is the *current directory* for Matlab.

The script *main.m* runs *makesystem.m*, that loads in the workspace the system parameters needed to call the *CGOffLineSetting.asv* function. Subsequently, the Simulink scheme *simulator.mdl*, which contains the *CG.asv* function, is opened and the simulation starts automatically. At the end of the simulation, the relevant signals are plotted.

2 Environment Description (for being used without invoking *main.m*)

2.1 Simulink schematic

The Simulink file *simulator.mdl* implements the schematic depicted in Figure 1. The entire simulink scheme is shown in Figure 2. The key block is the *CG block* (Figure 3), that executes the CG function described below. The simulation results are also stored in the workspace.

In order to make comparisons between a closed-loop system with and without a CG, in the simulator.mdl the precompensated system (the simulink part of the schematic within the green rettangle) is replicated in the yellow subsystem block

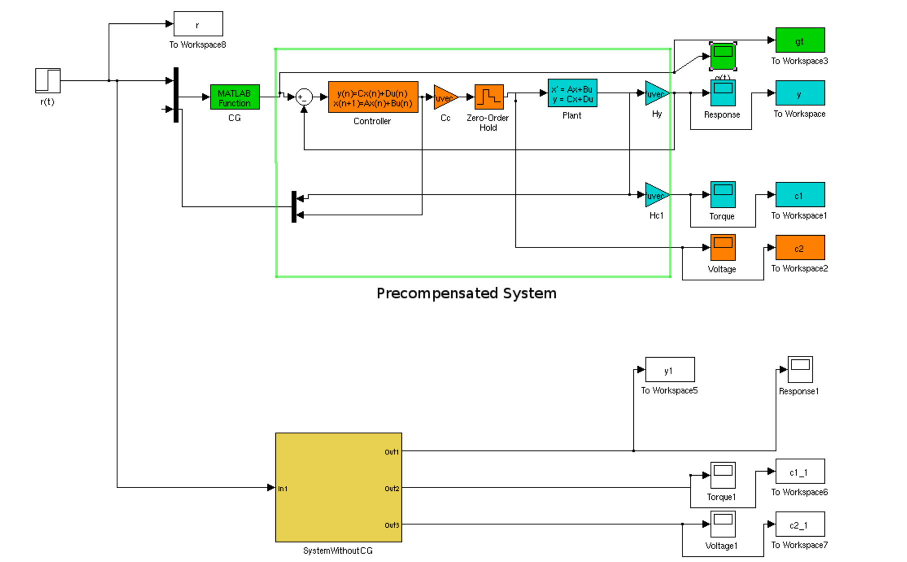

Figure 2: Simulink Schema

2.2 Function CG

The file CG.m contains the CG function with the following declaration:

function g=CG(r,x,A_s,b_s,b,T,Hc,Phi,G,L,k0,dimRef,P)

This function, starting from an initial state x, performs the CG action for current time instant, for a certain regulated system described by matrices (Phi, G, Hc, L) used to compute the virtual system evolutions up to k_0 steps (see [1]).

$$\begin{array}{c} g(t) = \min_w (w - r(t))' \Psi (w - r(t)) \\ \text{subject to} \\ TH_c \Phi^k x(t) + TR_k^c w \le g,\ \ k = 0, ..., \bar{k} \\ T\left(H_c(I - \Phi)^{-1}G + L\right) w \le g - \delta[\sqrt{T_i' T_i}] \end{array} \tag{1}$$

By denoting with

$$\begin{array}{c} \text{g:=}g(t), \\ \text{W:=}T\left(H_c(I - \Phi)^{-1}G + L\right), \\ \text{q:=}g - \delta[\sqrt{T_i' T_i}], \\ \text{A:=}TR_k^c\ \ k = 0, ..., k0, \\ \text{b:=}g - TH_c\Phi^k x(t)\ \ k = 0, ..., k0 \end{array}$$

and

$$\text{x:=}x(t)$$

then problem (1) becomes

$$g = \min_w (w - r)' Psi (w - r(t)) \tag{2}$$

$$\text{subject to} \tag{3}$$

$$Aw \le b\ \ prediction\ \ constraints \tag{4}$$

$$Ww \le q\ \ steady - state\ \ constraints \tag{5}$$

The output of CG function is the modified reference g.
The parameters of the CG function are

- r - the prescribed reference (provided in simulink scheme)
- x - the measured state at time t (provided in simulink scheme)
- W, q - needed for test inequality (5)
- b - represents the $C(\epsilon)_\infty^\delta$ region
- T - projection matrix
- Phi, G, Hc, L - matrices of the state space model of the precompensated system
- k0 - Virtual Horizon
- dimRef - reference vector dimension
- Psi - weighting matrix for objective function (2)

  Parameters b, T, Phi, G, Hc, L, k0 and dimRef are needed to build the inequalities (4)

In the simulation context, CG can be used as a Matlab Fcn block (Figure 3) as shown in the Simulink scheme (Figure 2).

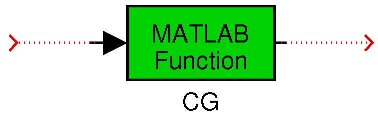

Figure 3: The CG simulink block

2.3 Function CGOffLineSetting

In this function, the CG off-line parameters are determined. This is a useful tool to compute several parameters needed to CG execution. The declaration is

function [W q b k0]=CGOffLineSettings(Phi,G,Gd,Hc,L,Ld,g,dmax,delta,epsilon,dimRef,dimCon,dimSys,T)

where the output parameters are:

- W, q - matrix used for defining the steady-state constrained region $Ww \leq q$
- b - represents the Cinf region
- k0 - Virtual Horizon

whereas the input parameters are:

- Phi - dynamic matrix of the global precompensated system
- G - input command matrix of the global precompensated system
- Gd - input disturbance matrix of the global precompensated system
- Hc - state/costrained output matrix
- L - input/costrained output matrix
- Ld - disturbace/costrained output matrix
- g - constraints definition
- dmax - if there is a disturbance signal $d \in \mathcal{D}$ perturbing the precompensated system this parameter is $dmax = \max_{d \in \mathcal{D}} \|d\|_2$
- delta - margin of Cdelta region
- epsilon - margin for $C_\infty(\epsilon)^\delta$ region
- dimRef - reference dimension
- dimCon - constraints dimension
- dimSys - system order
- T - projection matrix

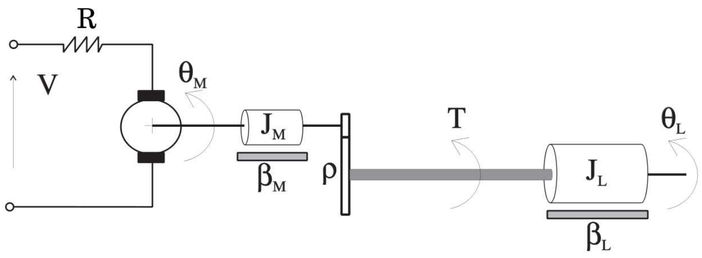

Figure 4: Servomechanism model.

2.4 Changing simulation parameters - makesystem.m

In the script *makesystem.m*, the system parameters are specified. Several variables defined are followed by the symbol %*. This indicates that you can't change the name of this variable in order to avoid errors in Simulink simulation because they appears in Simulink blocks of *simulator.mdl*.

If you want to simulate another system you have to modify the script *makesystem.m* in order to change the system parameters, *main.m* to change the plotting preferences and, of course, *simulator.mdl* to change the global system structure.

3 Simulating a position servomechanism

In this package the CG is applied to the position tracking problem of a servomechanism, schematically described in Figure 4.

3.1 System description

The system consists of a DC-motor, a gear-box, an elastic shaft and an uncertain load. Technical specifications involve bounds on the shaft torsional torque T and on the input voltage V as well. System parameters are reported in Table 1. Denoting with θ_M, θ_L respectively the motor and the load angles, and setting

Table 1: Model parameters

Symbol	Value (MKS)	Meaning	Symbol	Value (MKS)	Meaning
L_S	1.0	shaft length	d_S	0.02	shaft diameter
J_S	negligible	shaft inertia	J_M	0.5	motor inertia
β_M	0.1	motor viscous friction coefficient	R	20	resistance of armature
K_T	10	motor constant	ρ	20	gear ratio
k_θ	1280.2	torsional rigidity	J_L	$20J_M$	load inertia
β_L	25	load viscous friction coefficient	T_s	0.1	sampling time

$$x_p = \begin{bmatrix} \theta_L & \dot\theta_L & \theta_M & \dot\theta_M \end{bmatrix}'$$

the model can be described by the following state-space equations

$$\begin{cases} \dot x_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_\theta}{J_L} & -\frac{\beta_L}{J_L} & \frac{k_\theta}{\rho J_L} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_\theta}{\rho J_M} & 0 & -\frac{k_\theta}{\rho^2 J_M} & -\frac{\beta_M + k_T^2/R}{J_M} \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_T}{R J_M} \end{bmatrix} V \\ \theta_L = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x_p, \; T = \begin{bmatrix} k_\theta & 0 & -\frac{k_\theta}{\rho} & 0 \end{bmatrix} x_p \end{cases}$$

Since the steel shaft has a finite shear strength, a maximum admissible shaft $\tau_{adm} = 50N/mm^2$ impose the following constraint on the torsional torque $|T| \leq 78.5398\ Nm$. Moreover, the input DC voltage V has to be constrained within the range $|V| \leq 220\ V$. The model is transformed in discrete time by sampling every $T_s = 0.1s$ and using a zero-order holder on the input voltage. A digital controller is used having the following transfer function from $e = (r - \theta_L)$ to V

$$G_c(z) = 1000\frac{9.7929z^3 - 2.1860z^2 - 7.2663z + 2.5556}{10z^4 - 2.7282z^3 - 3.5585z^2 - 1.3029z - 0.0853}, \tag{6}$$

References

1 A. Casavola "The Command Governor Approach for Dummies", DEIS TR 7-2005, University of Calabria, 2005.