

Indian Institute of Technology Bombay

Hardware Interfacing & Control

Prof. P. S. V. Nataraj

PC Based Interface

Using Custom Hardware interface









DC Motor Control







Training Modules



Modules for Speed & Position control

System Identification

Traditional Control using P PI PID

Neural Networks

Deep Learning



Robust Control



USB Connectivity – Plug & Play



Creates real world plant experience on your desktop

Interactive frontend Software modules

Complete Lab Development Package for Academics

Used for class room teaching over 4 years in IIT

Open communication command set





Indian Institute of Technology Bombay

DC Motor control using MATLAB/SIMULINK



Outline

- Experiment No- 1
 - Validation of motor model for speed control
- Experiment No-2
 - PI Control Gains for Motor speed control



Experiment No-1 Validation of motor model for speed control









Procedure

- Set ts=0.015
- Run the matlab simulink model
- To stop the motor press the reset button on the DC motor kit .



FOPTD

• The First-order Plus Time Delay (FOPTD) model is given by $G(s) = \frac{\Delta Y(S)}{\Delta U(s)} = \frac{K e^{-tdS}}{\tau s + 1}$

gain K ,time constant $\tau\,$ and dead time t_d



Apply two-point method for system

- t_{63.2} = Time required for the output to reach 63.2 % of the steadystate value
 - t_{28.3} = Time required for the output to reach 28.3 % of the steadystate value.
- $K = \frac{Difference in two steady states of output}{Difference in two steady states of input}$
- $\tau = 1.5(t_{63.2} t_{28.3})$
- $t_d = t_{63.2} \tau$



Sample values

- t_{63.2} = 0.45 sec
 - t_{28.3} = 0.33 sec
 - $\Delta u(t)$ = 20 PWM units
 - $\Delta y(t) = 359 RPM$
 - Using the two-point method

K= 17.95 τ= 0.18 sec L= 0.12sec



Transfer function

• G(s) =
$$\frac{\Delta Y(S)}{\Delta U(s)} = \frac{17.95 \ e^{-0.12s}}{0.18s+1}$$



Output response

- In the open loop, the plant is brought to equilibrium by applying a step of 150 PWM units.
- The corresponding speed is around 2000 RPM
- After the motor speed settles, the PWM input is instantaneously changed to 170.
- As a result, the speed increases to around 2400 RPM.



Output vs Input



IIT Bombay

Experiment No-2 PI Control Gains for Motor speed control





400 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3500 800 1200 1/10/0 2000 2400 2800

Set Foint Value

Ziegler-Nichols Rule for Tuning PID Controllers

Type of controller	K _c	T _i	Τ _d
Ρ	1/RL	∞	0
PI	0.9/RL	3L	0
PID	1.2/RL	2L	0.5L



Calculations

• Compute the controller parameters as follows:

$$K_{c} = \frac{0.9}{RL}$$
$$T_{i} = 3L$$
$$R = k/\tau$$

Sample values

 $K_c = 0.04102$ $T_i = 0.825 \text{ sec}$



Procedure

- Double click on the controller block .
- Double click on PID Controller block.
- Enter the P and I values calculated using the Ziegler-Nichols Rule .



PID Block





PID block configuration

🔚 Block Parameters	a PLD Controller	. X.
PID Controller		
This block implement ant-windup, exten (requires Simulink	ents continuous - and discrete time PID control algorithms and includes a mai reset, and signal tracking. You can tune the PID gains automatically (c Control Design).	idvanced features such as using the "lune" button
Controller: PI	•] Form : [Nice]	-
Time domain: Continuous tim Discrete-time	ne	
Main PID Adva	etens	
Source:	internal 👻	E <u>Compensator formula</u>
Proportional (P):	0.075208913049025	2003 000
Integral (I):	0.36 	$P_{(1+I-\overline{\epsilon})}^{(1+I-\overline{\epsilon})}$
Initial conditions		
Source: Intern	ral	~
Integrator: U		
0	OK Cancel] [Help Apply



Output

- Kc = 0.075 and Ti = 0.36 sec.
- After the speed settles at 2000 RPM, a step of 400 RPM is applied.
- It is seen that the output follows the set point and the speed settles at 2400 RPM.
- Next, a negative step of 400 RPM is applied.
- It is clearly observed that motor speed decreases and settles at 2000 RPM.



Output response





Neural Network Model And Neural Network Controller for DC Motor

Prof.P.S.V.Nataraj Systems and Control Engineeering IIT Bombay



1.How to use Neural Network tools

Neural Network Start (rinstart)				
Welcome to Neural Network Start Learn how to solve problems with neural networks.				
Getting Started Wizards More Inform	ation			
Each of these wizards helps you solve each wizard generates a MATLAB scri Example datasets are provided if you o	a different kind of problem. The last panel of pt for solving the same or similar problems. do not have data of your own.			
Each of these wizards helps you solve each wizard generates a MATLAB scri Example datasets are provided if your Input-output and curve fitting.	a different kind of problem. The last panel of pt for solving the same or similar problems. do not have data of your own. Fitting app (nftool)			
Each of these wizards helps you solve each wizard generates a MATLAB scri Example datasets are provided if your Input-output and curve fitting. Pattern recognition and classification.	a different kind of problem. The last panel of pt for solving the same or similar problems. do not have data of your own.			
Each of these wizards helps you solve each wizard generates a MATLAB scri Example datasets are provided if you Input-output and curve fitting. Pattern recognition and classification. Clustering.	a different kind of problem. The last panel of pt for solving the same or similar problems. do not have data of your own.			



2.Network Fitting GUI

Neural Fitting (nftool)

	å	10	8	λ.
the second s	6	23	ε.	φ

Welcome to the Neural Fitting app.

Solve an input-output fitting problem with a two-layer feed-forward neural network.

					- 10		
	 	~	-			-	
-	 _	-	-	-		-	

In fitting problems, you want a neural network to map between a data set of numeric inputs and a set of numeric targets.

Examples of this type of problem include estimating house prices from such input variables as tax rate, pupil/teacher ratio in local schools and crime rate (house dataset); estimating engine emission levels based on measurements of fuel consumption and speed (engine_dataset); or predicting a patient's bodyfat level based on body measurements (bodyfat_dataset).

The Neural Fitting app will help you select data, create and train a network, and evaluate its performance using mean square error and regression analysis.



A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons (fitnet), can fit multi-dimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer.

The network will be trained with Levenberg-Marquardt backpropagation algorithm (trainfin), unless there is not enough memory, in which case scaled conjugate gradient backpropagation (trainsig) will be used.

To continue, click [Next].				
👄 Neural Network Start 🔰 🕅 🛛	Velcome	🖶 Back	Next	Cancel



3.Data set selection

📣 Neural Fitting (nftool)	
Select Data What inputs and targets define your fitting problem?	
Get Data from Workspace	Summary
Input data to present to the network.	No inputs selected.
Inputz:	
Target data defining desired network output.	No targets selected.
Targets: (none) •	
Samples are: 🛞 🔲 Matrix columns 🔿 🗐 Matrix rows	
Wester to out this tool - 24 or example data and	
want to try out this tool with an example data set	
Load Example Data Set	
Select inputs and targets, then dick (Next).	
Reural Network Start H4 Welcome	Sack Next Cancel



4.Network Architecture





5.Train Network

🚸 Neural Fitting Inftaol				
Train Network Train the network to fit the inputs and targets.				
Train Network	Results			
Choose a training algorithms		💑 Samples	😑 MSE	E
Levenberg Morgwordt T	Training:	346		-
This almost them thereing the manuface second sector at the time time. Training	👽 Validations	75		
automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.	🤨 Teating.	75		-
Train using Levenberg-Marquardt. (Insidier)	[Plot Fit Pitz	t Error Histogram]
Train		PlotReg	reation	
Netes				
Training multiple times will generate different results due to tiffesent initial conditions and sampling.	Mean Squared B between output means no error.	mor is the average so s and tangets. Lower	wared difference values are better. Zo	era
	Regression R Val outputs and tary relationship, 0 a	tees measure the corr pets. An R value of 1 r random relationship	relation between means a close	
Itain network, then click [Next].				
👄 Nasaral National Start		🌲 Bar	k Net	Carrol



6.Training Results

Neural Network Training (notraintool))			
Neural Network				
Algorithms Data Division: Random (dividerend Training: Lavankerg Marquardt Performance: Mean Squared Error (Calculations: MEX) i (trainlm) (mae)			
Progress Epoch: 0 Lime: 208 Gradient: 3.07e+03 Mu: 0.00100 Validation Checks: 0	19 iterations 0:00:00 2.24 9.55 0.00000 5	1000 0.00 1.00e-07 1.00e-10 6		
Nuts Performance (plotperform) Training State (plottrainstate) Enver Histogram (plotter hist) Regression (plottegression) Fit (plottit)				
Piot Interval:	1	perche		
Validation stop.	Stop Training	g Carnel		



7. Deploy Solution

📣 Neural Fitting (nftool)	
Deploy Solution Generate deployable versions of your trained neural network.	
Application Deployment Prepare neural network for deployment with MATLAB Compiler and Builder tools.	
Generate a MATLAD function with matrix and cell array argument support:	(genFunction) MATLAB Function
Code Generation	
Prepare neural network for deployment with MATLAB Coder tools.	
Generate a MATLAB function with matrix-only arguments (no cell array support):	(genFunction) MATLAB Matrix-Only Function
Simulink Deployment	
Simulate neural network in Simulink or deploy with Simulink Coder tools.	
Generate a Simulinic diagram:	(genzim) Simulink Diagram
Graphics	
Generate a graphical diagram of the neural network:	(network/view) 🖉 Meural Network Oisgram
Deploy a neural network or click [Next].	
Rearel Network Start Hill Welcome	Pack Next Octool



Comparison between DC Motor Model and Neural Network Model

Steady state output





Random Reference Signal





Neural Network Controller Replacing PID



Steady state and random reference signal tracking





Neural Network Controller with DC Motor





Contact

Prof. P. S. V. Nataraj Systems & Control Engineering IIT Bombay nataraj@sc.iitb.ac.in

Pramod Mhaske Sr. Project Manager pramod.mhk@gmail.com Tel 022-25764884 , 9320201648

