Logger : A Program for Data Logging

Gustafsson, Kjell

1990

Document Version:
Publisher's PDF, also known as Version of record

CODEN: LUTFD2/(TFRT-7457)/1–4/(1989)

`logger` – a program for data logging

Kjell Gustafsson

Department of Automatic Control
Lund Institute of Technology
July 1990

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Internal Report

Date of issue
July 1990

Document Number
CODEN: LUTFD2/(TFRT-7457)/1–4/(1990)

Author(s)
Kjell Gustafsson

Supervisor

Sponsoring organisation

Title and subtitle
`logger` – a program for data logging

Abstract

`logger` is a small program for data logging on an IBM PC/AT. Its intended use is data collection during identification experiments. This report provides a short user manual.

Key words
data logging, identification

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
4

Recipient's notes

Security classification

A Manual for logger

Introduction

Logger is a program for collecting data during an identification experiment. The program contains a signal generator that can be used as input signal to the process or reference value to the controller. The excitation signal and two external signals (typically the process input and output) are stored and saved on a disk file. The data is saved on a format allowing later processing in MATLAB.

The program is started with the command `logger`. It will print a welcome message, a list of available commands, and the current parameter values on the screen. Then a prompt appears and the program waits for the user's action. Using the information on the screen it should be rather obvious how to proceed. If in doubt use the help command (`help`).

The excitation signal

The excitation signal is a PRBS (pseudo random binary sequence) produced from an 8-bit shift register. A 1 from the shift register is interpreted as a positive signal and a 0 as a negative. The amplitude of the signal can be set using the command `amplitude`, and its mean value is set using `mean`. Changing the mean value while having the amplitude equal to zero is equivalent to a step signal, and can be used for step response experiments.

It is up to the user to check that the mean value of the signal plus its amplitude is within the output signal limitations, i.e. ± 10 V. If this is not fulfilled the signal will saturate causing the actual signal fed to the process to differ from the one stored by the program.

The shift register does not have to be updated every sampling period. Through the command `period` one decides how many sampling periods there should be between each shift register update.

Input and output

The excitation signal is fed out through the DA-converter at channel 0. Similarly, the external signal to be logged is fetched from the AD-converter at channel 0.

The excitation signal is fed out as long as the program executes, and is not effected by the `log`-command.

Sampling interval and synchronism

The sampling interval is changed using the command `tsamp`. The shortest interval possible on the Tandon is 1 ms, while on the IBM one has to settle with 2 ms due to slower AD and DA-converters. If changing sampling interval while logging data the current data will be discarded and the logging stopped.

It is not possible to make the AD and DA synchronous. First the external signal is sampled, then the new excitation signal is fed out. On the Tandon the delay between these two operations is approximately 0.1 ms while it is 0.5 ms on the IBM. Hence, if using the logged data to identify a continuous time process, the process will appear to have an extra time delay of approximately 0.1 ms or 0.5 ms, respectively.

Logging data

Using the command `n` one decides how many data points to collect. The maximum value is 10000. PC-MATLAB is only able to handle matrices with less than 8000 entries, and hence one can only use something like 2600 data points (three columns). PRO-MATLAB on VAX and SUN does not have this limitation.

The data logging is started using the command `log`. The logging will continue until all the data points are collected. If the `log`-command is given while already logging, the current data will be discarded and the logging restarted.

The command `show` tells if the logging is ready.

Data file

When all data points are collected the program is stopped using the command `quit`. All data points are written to a file, `data.mat`, on the current directory. The file contains three columns: the first two with the measured signals (column from AD-channel 0 and column 2 from AD-channel 1), and the third with the excitation signal. If the `quit`-command is given while logging the data collected so far will be saved.

If the file `data.mat` already exists it will be deleted. The only exception is when no data points have been collected, then the file `data.mat` is left unchanged.

Command listing

All commands may be abbreviated provided they do not become ambiguous. There is a help command which will give a listing of all available commands together with a short explanation of their meaning.

The available commands are

`quit`	Stop program and write collected data on file `data.mat`.
`log`	Start logging data. If already logging then discard current data and restart logging.
`tsamp <val>`	Set the sampling interval to `<val>` ms. If the command is given while logging data the current data is discarded and the logging stopped.
`period <val>`	Set the number of sampling intervals between PRBS shift register updates to `<val>`.
`n <val>`	Set the number of data points to be collected to `<val>`. The maximum number is 10000.
`mean <val>`	Set the mean value of the excitation signal to `<val>` V. The value should be in the interval [-10.0,10.0] V.
`amplitude <val>`	Set the amplitude of the excitation signal to `<val>`. The value should be in the interval [0.0,10.0] V.
`show`	Print current parameter values on screen, including information telling if the program is logging, if the logging is off, or if the logging is ready.
`help`	Print the available commands and a short explanation on the screen.