A TOOLBOX for Discrete Time Design and On-Line Control

Lundh, Michael

1988

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

Link to publication

Citation for published version (APA):
Lundh, M. (1988). *A TOOLBOX for Discrete Time Design and On-Line Control*. (Technical Reports TFRT-7382). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7382)/1-12/(1988)

A TOOLBOX for Discrete Time Design and On-Line Control

Michael Lundh

Department of Automatic Control
Lund Institute of Technology
March 1988

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

Document name
Report

Date of issue
March 1988

Document Number
CODEN: LUTFD2/(TFRT-7382)/1-12/(1988)

Author(s)
Michael Lundh

Supervisor

Sponsoring organisation

Title and subtitle
A TOOLBOX for Discrete Time Design and On-Line Control

Abstract

This report describes a package for on-line control using an IBM AT with EGA graphics. The package contains tools for control design and real time control. It is provided with a mouse-based man-machine interface. The package is intended for teaching where the students can quickly move from design to implementation of control systems. Therefore it is easy and fast to investigate what different specifications mean for the performance. Some features of the package are: Pole placement design, Generation of discrete specification from continuous specification, Recursive parameter estimation, Adaptive control and Graphics display. The toolbox has been used for laboratory experiments in courses in digital and adaptive control.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
12

Recipient's notes

Security classification

Graphics Display

The program has plotting facilities. Plotting of signals is done in real time. It is always possible to see the reference signal and the plant output y_1. The other signals y_2, y_3 and u can together with the control error $r - y_1$ also be shown in real time.
The user choses the time scale on the horizontal axis and how often signals are plotted. The program can plot the signals every, every second and every fourth sample. For long sampling intervals it is possible to plot four times per sampling interval. Plotting can also be suppressed.

Design

Different specifications for the closed loop system can be explored by using the built in pole placement design procedure for SISO systems. For the process

$$H(q) = \frac{B(q)}{A(q)} \tag{3}$$

given by the user or estimated by the internal identification routine, we want to achieve a closed loop system with the pulse transfer operator

$$H_m(q) = \frac{B_m(q)}{A_m(q)} \tag{4}$$

The design procedure given by Algorithm 10.1 in Åström and Wittenmark is implemented. The controller polynomials $R(q)$, $S_1(q)$ and $T(q)$ are calculated from the model (3) and the specifications. In this design we specify $B^+(q)$, $B^-(q)$, $A(q)$, $B'_m(q)$, $A_m(q)$ and $A_o(q)$ and if the controller should have integral action. The polynomial degrees are less than or equal to five. The characteristic polynomials $A_m(q)$ and $A_o(q)$ can be specified in terms of corresponding continuous time polynomials.

Process Identification

A recursive Least Squares parameter estimator based on dyadic reduction [Peterka] is implemented and can estimate the parameters of the processes. It is a numerically well behaved square root algorithm where the covariance matrix is factorized as $P(k) = L(k)D(k)L^T(k)$ where

$$L(k) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{pmatrix} \tag{5}$$

and

$$D(k) = \begin{pmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d_n \end{pmatrix} \tag{6}$$

are updated separately. The routine can estimate up to ten parameters. The estimator properties are given by the forgetting factor λ, the initial covariance

Commands from Mouse

SAVE	Copy the polynomials of the ready regulator to the running regulator.
INIT	Initialisation of states and polynomials in the regulator.
REPRIN	Show updated menu.

REFSIG-menu

Commands from Keyboard

MEAN r	Choice of mean for internal reference-signal. $(-1.0 \le r \le 1.0)$
AMPLITUDE r	Choice of amplitude for internal reference-signal. $(-1.0 \le r \le 1.0)$
PERIOD r	Choice of period-time for internal reference-signal. $(0.0 < r \le 1000\ s)$

Commands from Mouse

SAVE	Store data shown on the screen.
EXTERN	Choice of external reference-signal.
SQUARE	Choice of square-form of reference-signal.
TRIANGLE	Choice of triangle-form of reference-signal.
SINE	Choice of sinus-form of reference-signal.
STEP	Choice of step as reference-signal. (Single shot)
RAMP	Choice of ramp as reference-signal. (Single shot)

PLOT-menu

Commands from Mouse

RY1	Plot signals r and y_1.
RY2	Plot signals r and y_2.
RY3	Plot signals r and y_3.
RY12	Plot signals r, y_1 and y_2.
RY13	Plot signals r, y_1 and y_3.
RY23	Plot signals r, y_2 and y_3.
RY123	Plot signals r, y_1, y_2 and y_3.

ESTIM-menu

Commands from Keyboard

LAMBDA r	Choice of forgetting factor. $(0.1 \le r \le 1.0)$
D0 r	Choice of initial covariance-matrix $D(0) = I \cdot D_0$. $(r > 0.0)$
BF p	Choice of numerator for filter $B_f(q)/A_f(q)$. $(\deg B_f \le 2)$
AF p	Choice of denominator for filter $B_f(q)/A_f(q)$. $(\deg A_f \le 2)$
CSAF r_1 (r_2)	Like CSAM above but $A_f(q)$ is defined.
HIST r	Time history to be shown for estimated parameters.

Commands from Mouse

SAVE	Store data shown on the screen.
DMATR	Plot diagonal of D-matrix.
REPRIN	Show main table of this menu.

SHOWAB	Plot estimated parameters of A and B.
BODE	Display the bode-diagram for the model B/A.
REPRIN	Show main table of this menu.
RESET	Set parameters of A and B in model to default values.

DESIGN-menu

Commands from Keyboard

BPLUS p	Choice of $B^+(q)$.
BMINUS p	Choice of $B^-(q)$.
BMPRIM p	Choice of $B'_m(q)$.
BPLUS B	Cancel all process zeros: $B^+(q) := B(q)$
BMINUS B	Don't cancel any process zero: $B^-(q) := B(q)$
FACTORB	Factorize $B(q)$ in stable $B^+(q)$ and unstable $B^-(q)$ part. Note that only stability is regarded. Poorly damped zeros will appear in $B^+(q)$.
AM p	Choice of $A_m(q)$.
AO p	Choice of $A_o(q)$.
CSAM r_1 (r_2 (r_3))	Choice of $A_m(q)$ from continuous time specifications. Degree of polynomial is determined by the number of arguments. One argument $r_1 = \omega$ gives sampled counterpart to $(s+\omega)$. Two parameters $r_1 = \omega$ and $r_2 = \zeta$ gives the counterpart to $(s^2 + 2\zeta\omega s + \omega^2)$ and three parameters $r_1 = \omega$, $r_2 = \zeta$ and $r_3 = \alpha$ specifies the corresponding characteristic polynomial to $(s + \alpha\omega)(s^2 + 2\zeta\omega s + \omega^2)$.
CSAO r_1 (r_2 (r_3))	Like CSAM but $A_o(q)$ is defined instead.
FIXED	Choice of non-adaptive regulator.
ISTR	Choice of general adaptive pole-placement algorithm.

Commands from Mouse

SAVE	Store data shown on the screen.
COMPUTE	Do the design calculation. New polynomials in ready regulator. ONLY for fixed regulator!
TOREG	Do the design calculation and enter the new polynomials into the running regulator. ONLY for fixed regulator!
INTON	Choice of integrating regulator.
INTOFF	Choice of not integrating regulator.

REGUL-menu

Commands from Keyboard

R p	Choice of $R(q)$.
S1 p	Choice of $S_1(q)$.
S2 p	Choice of $S_2(q)$.
S3 p	Choice of $S_3(q)$.
T p	Choice of $T(q)$.
AO p	Choice of $A_o(q)$. This is the observer polynomial for the anti-windup function.
PID r_1 (r_2 (r_3 (r_4)))	Choice of PID-regulator. $r_1 = K$, $r_2 = T_i = (-1 = \text{off})$, $r_3 = T_d = (0)$, $r_4 = N = (30)$

matrix $D(0)$ and by the second order regression filter $\frac{B_f(q)}{A_f(q)}$. The user specifies these parameters. The estimated process model can either be used in the design procedure above or as a part of the adaptive controller. Only the A and B polynomials of single input single output systems can be estimated.

Adaptive Controller

Applying the design algorithm to a new estimate of the process model every sample gives an indirect adaptive controller based on pole placement where no process zeros are canceled. The design parameters are the characteristic polynomials $A_m(q)$ and $A_o(q)$ and if the controller should have integral action. Only single input single output systems can be controlled by the adaptive controller.

When the adaptive controller is running the following is done every sample:

- The control signal u is given by (2).
- Estimate $A(q)$ and $B(q)$ in (3).
- Calculate new $R(q)$, $S_1(q)$ and $T(q)$ in (2).

3. Man-Machine Interface

The program is interactive and the user communicates with it using mouse and keyboard. The mouse acts on three rows, each with different areas on the screen. Activating a function connected to such an area is done by first moving the mouse cursor to the desired area and then pressing one of the buttons on the mouse. The upper row depends on the chosen menu. Menu is chosen from the row in the middle. The bottom row affects the global modes of the program.

Global Modes

The controller works in three different modes.

RUN	This is the operating mode. The control signal is computed to make the process output follow the reference signal. If the parameter estimator or the adaptive controller are activated these algorithms are executed.
ZERO	The control signal is computed to make the process output zero. No execution of parameter estimator or adaptive controller.
STOP	Idle mode. The control signal is zero.

The parameter estimator and the adaptive controller can be activated when mode *RUN* is active. They can however be pre-activated when *ZERO* or *STOP* is active. Pushing *STOP* puts the controller in mode *STOP* and deactivates the parameter estimator and the adaptive controller. Signals are plotted only when mode *RUN* is active and plotting is on.

Menus

The man-machine interface is based on seven different menus. These are

SETUP	Setting of different parameters affecting basic functions of the program. These parameters are sampling period, I/O-channels, control signal limits, horizontal time scale for plots and how often

the program plots. Selecting this menu sets the regulator in mode *STOP*.

MODEL Specification and analysis of a model of the controlled process. The A and B polynomials, additional poles in the origin are given by the user. When the parameter estimator is running it updates the A and B polynomials. Structure of the estimator and and initial estimates are specified by assigning the A and B polynomials before the estimator is activated. The history of the parameter-estimates could be plotted. Finally the bode plot for the actual model can be drawn. If the estimator is active also the frequency response for the regression filter appears.

DESIGN Menu for designing a controller from given specifications and the model from the previous menu. The *DESIGN*-menu is different depending if a fixed or an adaptive controller is chosen. It is possible to give the characteristic polynomials $A_m(q)$ and $A_o(q)$ in continuous time specifications. The corresponding discrete time characteristic polynomial is calculated from the sampling interval and ω, ζ and α in one of

- $(s+\omega)$,
- $(s^2+2\zeta\omega s+\omega^2)$ or
- $(s+\alpha\omega)(s^2+2\zeta\omega s+\omega^2)$

The continuous time parameters are not stored so the correspondence between the discrete and continuous polynomials is lost if the sampling interval is changed.

REGUL Menu where controller polynomials directly are specified. Here is the only possibility to set the polynomials S_2 and S_3, i.e. to use a multi input single output regulator. The observer polynomial A_o for the anti-windup is specified here.

REFSIG Chosing the reference signal either by internal reference generator or by external signal. The form of the internally generated signals are of two types.

- One type for repeated runs. When the screen is plotted full, it is erased and plotting continued.
- The other type for single shots. For the forms STEP and RAMP the controller mode is set to *ZERO* when the screen is full. A new run is started by *RUN*.

PLOT Use whole screen for plotting. Signals plotted are the reference r, one or more of the y_i signals, the difference $r-y_1$ and the control signal u.

ESTIM Setting and displaying variables of the recursive parameter estimator.

A complete list of commands available on the different menus is found in the appendix. The changes made on a menu are *not* stored before the *SAVE* function is invoked! This means that previous data can be recalled by selecting the menu again if no *SAVE* has been done.

Example

This is an example which can be run through by an unexperienced user. Commands are given in *italics*. Commands given with the mouse are marked *(mouse)*. Comments are given to the right.

APPENDIX - Summary of Commands

Here is a complete list of all commands available in the in the program. The arguments for the keyboard commands are classified as r (real number), i (integer) or p (polynomial coefficients). The number of entered coefficients determines the degree of the polynomial. For real numbers and integers the allowed range is given. Polynomial degrees are limited to 5. A bracket means that the argument in it can be omitted. Slash means choices.

SETUP-MENU

Commands from Keyboard

CHANREF i	Choice of channel for reference input r. $(0 \le i \le 3)$
CHANY1 i	Choice of channel for process output y_1. $(0 \le i \le 3)$
CHANY2 i	Choice of channel for process output y_2. $(0 \le i \le 3)$
CHANY3 i	Choice of channel for process output y_3. $(0 \le i \le 3)$
CHANU i	Choice of channel for control signal u. $(0 \le i \le 1)$
TSAMP r	Choice of sampling-interval. $(0.01 \le r \le 10.0$ s$)$
HPT r	Choice of horisontal time scale for plotting. $(2.0 \le r \le 100.0$ s$)$
ULIM r_1 r_2	Choice of control signal limits. $(-1.0 \le r_1 < r_2 \le 1.0)$
NUMINP i	Choice of number of process output signals to be read from AD-converters. $(1 \le i \le 3)$

Commands from Mouse

SAVE	Store data shown on the screen.
BETWEEN	Choice of plotting between samples.
EVERY	Choice of plotting every sample.
EVERY2	Choice of plotting every second sample.
EVERY4	Choice of plotting every fourth sample.

MODEL-MENU

Commands from Keyboard

A p	Choice of $A(q)$.
B p	Choice of $B(q)$.
C p	Choice of $C(q)$.
DEG A/B/C i	Choice of degree for $A(q)$, $B(q)$ or $C(q)$. $(0 \le i \le 5)$
ADDPOL i	Additional poles in origin.
HIST r	Time history to be shown for estimated parameters.
BODEFREQ r_1 r_2	Frequency interval for bode-diagram. $(0 < r_1 < r_2 \le \frac{\pi}{t_{samp}})$
BODEFREQ r PI	Frequency interval for bode-diagram. Upper limit equal to Nyqvist-frequency. $(0 < r < \frac{\pi}{t_{samp}})$

Commands from Mouse

SAVE	Store data shown on the screen.
SAMPLE	Fast way to generate a sampled process model $H(q)$ for the servo in the laboratory experiments in Digital Control. The sampling interval is thereby used.
SHOWB	Plot estimated parameters of B.
SHOWA	Plot estimated parameters of A.

6. References

ÅSTRÖM, K. J. and B. WITTENMARK (1984): *Computer Controlled Systems*, Prentice-Hall, Inc., Englewood Cliffs, N.J..

PETERKA, V. (1986): "Algorithms for LQG self-tuning control based on input-output delta models," *2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing*, Lund, Sweden.

ANDERSSON, L. (1988): "Real Time Kernel and Graphics Library," Personal Conference with Leif Andersson.

ÅSTRÖM, K. J. and T. SCHÖNTHAL (1983): "PCALC A Polynomial Calculator – A Users Manual," Internal Report, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden..

Starting the program enter You into the *SETUP*-menu.

TSAMP 0.1	Set the sampling interval to 0.10 s
SAVE (mouse)	Save the changes
MODEL (mouse)	Select *MODEL*-menu
B 0.13 0.11	Set $B(q) = 0.13q + 0.11$
A 1 -1.98 0.98	Set $A(q) = q^2 - 1.98q + 0.98$
SAVE (mouse)	Save the changes
DESIGN (mouse)	Select *DESIGN*-menu
BMINUS B	Set $B^-(q) = B(q)$
CSAM 8 0.7	Sample $s^2 + 2 \cdot 8 \cdot 0.7s + 8^2$ to set $A_m(q)$
CSAO 12 0.9	Sample $s^2 + 2 \cdot 12 \cdot 0.9s + 12^2$ to set $A_o(q)$
INTON (mouse)	Select integrating controller
SAVE (mouse)	Save the changes
TOREG (mouse)	Do the design
REFSIG (mouse)	Select *REFSIG*-menu
AMPLITUDE 0.5	Set the amplitude of the reference signal to 5V
SAVE (mouse)	Save the changes
RUN (mouse)	Start the controller
PLOT (mouse)	Select *PLOT*-menu

Specifications can be changed and new controller polynomials can be calculated while the controller is running.

4. Software Architecture

The program is written in *MODULA 2*. It is about 5000 lines of code. It is implemented on an IBM AT (or compatible machine) with an Enchanged Graphics Adapter (EGA) and DA- and AD-converters. The program uses the real time kernel and the graphics library written at The Department of Automatic Control [Andersson].

Concurrent Processes

The program itself has six concurrent processes. One is waiting for the real time clock and the others are waiting for different events. Here are the processes and their main tasks.

Execute — This process is executed periodicly. It determines the internal reference signal, calculates the control signal and if activated, executes the parameter estimator or the adaptive controller. If plotting not is done between the sampling instants the controller states are updated. Plotting information are sent to process Plotter.

Update — Updates the controller states if plotting is done between samples. Since process Execute in this case waits for the clock four times per sample the time between two such AD-conversion instants is too short to update the regulator state. The updating procedure has to be split. Here it is easy handled by process changes.

Plotter — Plotting process that is waiting for data from process Execute. This process executes as long as data is available. Depending on menu a small or a big area is used.

MouseHigh — Waiting for the mouse. It performs high-priority functions invoked by the mouse, e.g. mode changes.

MouseLow Waiting for process MouseHigh. Executes low-priority mouse functions.

Keyboard Waiting for input from keyboard.

The priority order for execution is MouseHigh (10), Execute (20), Update (30), Plotter (40), MouseLow (50) and Keyboard (60).

Data Structure and Internal Communication

The program has several different data structures. Each is associated with a function of the program.

Controller

The parameters and states of the controller are stored in a monitor. Its data is is only accessible through monitor procedures, and thus only accessible by one process at a time.

Global Modes

Global modes are stored in a special kind of monitor. Data is only accessed through procedures. Setting data in the monitor is protected with semaphores as in a normal monitor. Reading data from the monitor is not protected. A certain structure of the program is required to avoid problems. Each process may only read from the monitor at one point in their code. This is the point where an execution cycle starts. The reason for using this type of monitor is to minimize execution time.

Plotting

Data for plotting is produced in the process Execute which puts recorded data for one sample in a ring buffer. This can contain 400 samples. Process Plotter is waiting for data to appear in the buffer. Data is processed for one sample at a time. This is somewhat inefficient. When the buffer is full the regulator mode is set to *ZERO*, implying that data is no more put in the buffer. Process Plotter continues to process data until buffer is empty. When plotting is off no data are put in the ring buffer.

Parameter Estimation

The recursive estimator has a record with its parameters and states. These are set from man-machine data when activating the estimator and displayed on the ESTIM- and the MODEL-menu.

The history of the estimated parameters are stored in a ring buffer. It contains data for 710 samples. Maximum number of data to be read is 700. The reason for having more data than the number allowed to read is that there is no protection for the buffer. Instead, there is a distance between the last written and the last read element. Mutual exclusion can not be guaranteed, but data is read such that problems will not appear unless the execution time of the controller exceeds the available time.

Man-Machine Interface

Data entered by the user are stored in a huge global record, with one sub-record for each menu. Each menu has its own copy of the sub-record. The *SAVE* function transfers data from the own to the global sub-record. When functions are activated, data from the global record is processed, if necessary, and transfered to the function-connected data structure.

Real Time Problems

The available execution time is always too short. Minimizing execution time has been a key issue throughout the whole program. Some sloppy real-time programming style has been used to make the program faster. This has been mentioned above. Another thing is the in-line code used in the controller instead of FOR-loops.
In spite of the somewhat sloppy style it has not been possible on an IBM AT to execute a second order controller while plotting in real time faster than

0.02*s* when a fixed controller is used,

0.04*s* when a fixed controller is used and four parameters are estimated and

0.14*s* when an indirect adaptive controller estimates four parameters and determines a new controller every sample.

A TANDOON AT is somewhat faster. The corresponding minimum sample intervals are 0.02, 0.03 and 0.10*s*. It is preferable for adaptive control.

Polynomial Handling

The module *PCALC* is a translation of some parts of the PASCAL program PCALC [Åström and Schönthal]. It contains routines for polynomial handling. There are routines for addition (Poladd), subtraction (Polsub), multiplication, (Polmul), division (Poldiv), greatest common divisor (Gcd), solution of diophantine equation (Diophantine) and spectral factorization (Sfactorize).

5. Concluding Discussion

A software package like this TOOLBOX is a good tool to visualize discrete time control theory in practice. Having the design algorithms in the same package that implements the control algorithm leads to short time from a specification to a running controller.
The advanced features of the package are the powerful pole placement design algorithm, the design based on continuous specifications although the regulator is discrete, the colour graphics display, the general least squares recursive parameter estimator and the general indirect adaptive regulator.
The TOOLBOX package has until now been used for laboratory experiments in courses in digital and adaptive control given at The Department of Automatic Control in Lund.

Future Development

Making a list of all interesting ideas that appeared during the development of this program would be too long to present here. However some nice functions are listed here.

- Possibility to read a file with controller polynomials. A controller could then be designed e.g. using PC-MATLAB.
- A direct adaptive controller.
- Logging of signals and parameters on file for off-line processing.