Scandinavian Control Library - A Subroutine Library in the Field of Automatic Control

Wieslander, Johan

1977

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

Link to publication

Citation for published version (APA):
Wieslander, J. (1977). *Scandinavian Control Library - A Subroutine Library in the Field of Automatic Control*. (Research Reports TFRT-3146). Department of Automatic Control, Lund Institute of Technology (LTH).

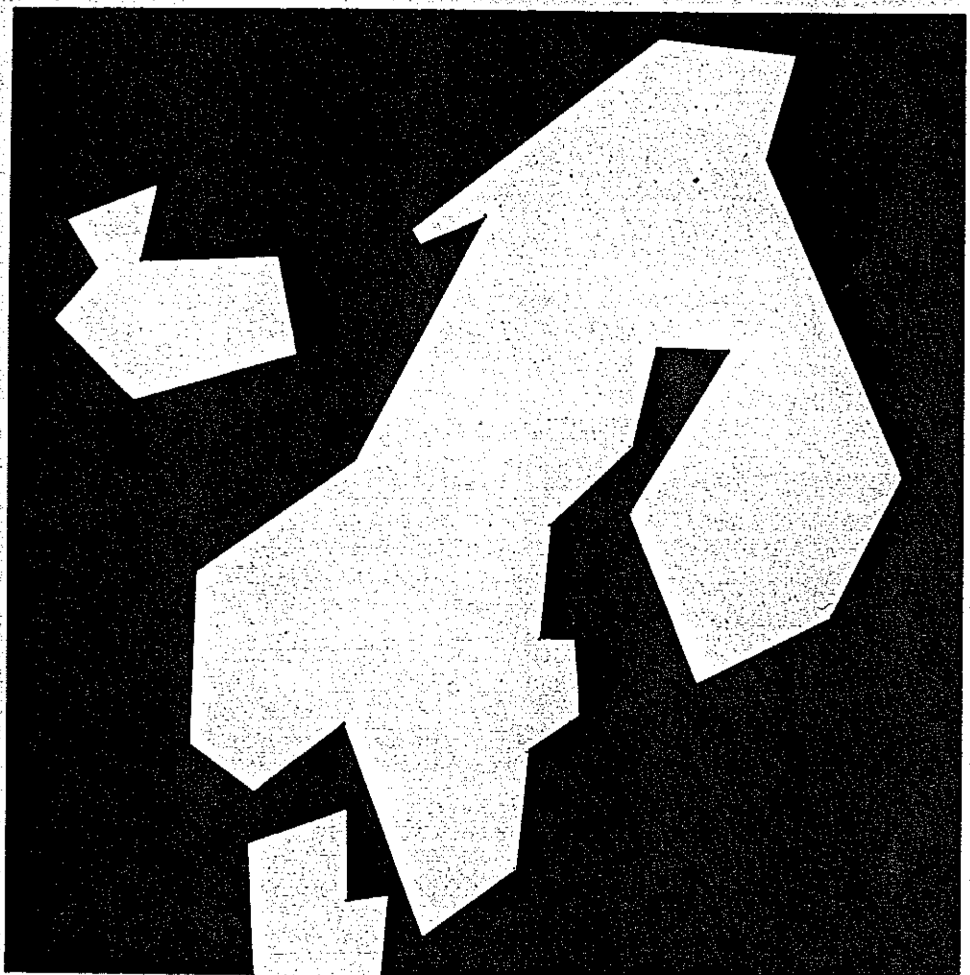
Total number of authors:
1

CODEN: LUTFD2/(TFRT-3146)/1-038/(1977)

SCL
Scandinavian Control Library

A SUBROUTINE LIBRARY IN THE FIELD OF AUTOMATIC CONTROL

EDITED BY J WIESLANDER



NORDFORSK
The Scandinavian Council for Applied Research

A SUBROUTINE LIBRARY IN THE FIELD OF AUTOMATIC CONTROL

The NORDFORSK project-group on "Computer Aided Design of Dynamical Systems"
January 1977

CONTENTS

§0. Introduction

The Scandinavian Council of Applied Research, 'NORDFORSK', has sponsored a project 'Computer Aided Design of Dynamical Systems'. Members in the project are the following institutes who have signed an agreement to cooperate in keeping this library up to date.

Starkstrømsafdelingen
Danmarks Tekniska Højskole
Lyngby, DANMARK (Denmark)

Säätötekniika
Teknillinen Korkea Koulu
Espoo, SUOMI (Finland)

(Laboratoriet för Reglerteknik
Tekniska Högskolan i Helsingfors
Esbo, FINLAND)

Institutt for Teknisk Kybernetikk
Norges Tekniske Høgskole
Trondheim, NORGE (Norway)

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Lund, SVERIGE (Sweden)

The aim of the project is to develop interactive programs for computer aided design and analysis of dynamical systems. It was decided that a prerequisite of the cooperation was a basis for exchange of subprograms (subroutines, functions, procedures etc). The result is reported in: Elmqvist, H; Tyssø, A; and Wieslander, J: 'Programming and documentation rules for subroutine libraries - Designed for the SCL', NORDFORSK. It was also decided to form a subroutine library to give the project members equal computational tools.

§1. General policy

A great problem is that of commercial interests. Therefore this library will contain routines of a general algorithmic nature, hopefully being more neutral in this respect.

Complete program packages and other routines with commercial or copyright restrictions are not included. This will facilitate access for outside bodies, cf §§ 5 & 6.

The project members and all other bodies that in the future will benefit from this library, should feel a moral obligation to submit new routines to the library as they are developed, and also to revise old ones so they fit into the library to make it complete.

The list of members given in §0 is not at all definite, those who wish to take a more active part in the development of the library may be elected to achieve full membership.

It is also possible to be included on the send list (§4) and thus to recieve regular information of new (or deleted) routines.

§2. Acceptance rules

The main rule for acceptance of a subprogram is that it conforms to the rules given in: Elmqvist et al: 'Programming and documentation...', see §0.

This should apply both to coding and documentation. The subprogram should also deal with problems found in the Systematic Contents List, §8, and should not duplicate a routine already in the library.

A routine submitted for acceptance should be followed by some test problems to demonstrate its use and properties. This is an aid to the members in their acceptance tests, § 3.

§3. Acceptance & deletion procedures

Acceptance of a subprogram into the library is based on a review procedure:

When someone has constructed a routine to be included in the library, he should send the program code and documentation together with some examples to the coordinator, either directly or through a member, §§0,1. The coordinator will then send the routine to each of the members. They are then free to criticize code and/or documentation, and to try to contrive examples where the proposed routine fails or behave erroneously.

If no complaints have been filed within two month's time, the routine has been accepted. On the other hand, in the interest of the quality of the library, no routines should be allowed with unsolved objections.

A subprogram may be deleted from the library, following a renewed review procedure. Such procedure is initiated by the coordinator after recieving a performance report with substantial complaints on a library routine.

The result of the review should be either to delete the routine, to correct it, to change it to a better one, or at least to include in the (revised) documentation a warning of the problem.

§4. Administration

A coordinator is elected from one of the member institutes for a one-year term. The coordinator has several duties:

a) It is the coordinator's responsibility to initiate and supervise the acceptance procedure for new routines;

to assign them a number within the library; to see to it that documentation is complete and duplicated so that all library binders can be properly updated.

b) The alphabetical and systematical contents list should be continually updated.

c) The coordinator maintains the send-list.

d) The coordinator recieves and acts upon performance reports.

e) He should also have a function as a 'clearing-house' to prevent people from working on the same problems.

§5. Juridical and economical complications

Routines within the library are, to the best of the implementor's knowledge, free of claims in any form. The program listing and documentation is public. There is no responsibility for errors that may be found in the programs or may be the result of running them.

§6. Access

Routines in the library are available to anyone. The program listing is public and the implementor's institute, indicated in the documentation, is willing to produce a copy on paper or a computer readable medium (paper-tape, magnetic tape, cards etc) at nominal cost.

However, test examples, test programs, and related material are the implementor's property and is available from him only according to his own desires.

§7. Organization of the library binder

In §8, a systematic table of contents is described. Each library routine should fit into this table. This would give the routine a section number. Appended to this is an ordinal number within that section. In this way a number is constructed that will give the documentation of a routine a unique place within the library binder. The pages should have a corresponding enumeration.

The table of contents in §8 will in due time be filled with routine names and subtitles.

There will also be an alphabetical table of contents, §9.

Revised pages will have a date in their lower left corner to distinguish them from earlier revisions.

The library binder will also include some forms for the performance reports.

§8. Systematic contents list

This list of contents includes all library routines, ordered systematically in the following chapters.

1. Utility routines

This chapter contains routines peripheral to the main topic of the library but essential to many applications of it. Presently it contains four sections.

1.1 Text handling

1.2 File handling

1.3 Plotting

2. Basic mathematical routines

Routines of a mathematical nature, of basic interest in Automatic Control and similar applications.

2.1 Matrix handling

2.2 Polynomials & rational functions

2.3 Functions of several variables

2.4 Differential equations

2.5 Statistical

3. Analysis of linear dynamic systems

This chapter divides the linear dynamic systems into four different representation methods. It also gives references to routines that may transform one representation into another. Note especially that the second section, although headed 'Transfer Functions' actually contains more complex forms as well.

3.1 State space representation

3.2 Transfer function representation, polynomials in the differentiation operator and Rosenbrock's system matrix

3.3 Frequency response representation

3.4 Time response representation

3.5 Transformations between system representations

4. Synthesis

Methods to design control systems, not based on linear quadratic theory (treated in chapter 5) are found in this chapter.

4.1 Bode-Nichols-Nyquist

4.2 Root-Locus method

4.3 Pole assignment - state space

4.4 Pole assignment - transfer function

4.5 Minimal variance control

4.6 Multivariable frequency response methods

4.7 Observers

5. Linear quadratic control

This chapter contains routines used to design optimal control for linear systems with a quadratic performance criterion.

5.1 Covariances and quadratic loss functions

5.2 Transformation from continuous time to discrete time form

5.3 Time variable solution based on Riccati equation

5.4 Stationary solution based on Riccati equation

5.5 Solution with spectral factorization

6. Data analysis

This chapter contains routines used e.g. in analysis of measured data.

6.1 Scaling, interpolation

6.2 Simple statistical properties, statistical tests

6.3 Estimation (removal) of trends, periodic signal components etc

6.4 General transformations

6.5 Auto- & cross-covariances

6.6 Auto- & cross-spectra

6.7 Laplace- & Fourier-transforms, FFT

6.8 Input & output

7. Identification

Routines to estimate models from measured data are contained in this chapter.

7.1 Time response to polynomial representation. No noise model.

7.2 Time response to polynomial representation. With noise model.

7.3 Pulse- or step-response to state space model. Realization.

7.4 Time response to state space model

7.5 Frequency response to transfer function representation

7.6 Frequency reponse to state space representation

7.7 Optimal test signals

8. Simulation

With simulation is understood generation of an output as a function of time when a specified input is given from a system represented e.g. as in chapter 3.

8.1 Generation of time series, random numbers

8.2 Simulation by frequency response methods

8.3 Simulation of linear systems on polynomial form

8.4 Simulation of linear systems on state space form

8.5 Simulation of non-linear systems

9. On-line control

This chapter contains some modules of a more general nature for implementation of on-line control.

9.1 Analog input & output (Purdue standard)

9.2 Classical regulators, e.g. PID-regulators

9.3 General linear regulators

9.4 Kalman filter

9.5 Self-tuning regulators

9.6 Adaptive regulators

1. Utility routines

1.1 Text handling

1.1.1 Character packing & unpacking

1.1.2 Line input & output

1.1.3 Text string manipulation

1.1.4 Formula manipulation

1.2 File handling

1.2.1 File management

1.2.2 File input & output

1.3 Plotting

1.3.1 Setup, plotting parameters

1.3.2 Scaling

1.3.3 Display formatting

1.3.4 Data output

2. Basic mathematical routines

2.1 Matrix handling

2.1.1 Elementary matrix operations

2.1.2 Linear equation systems

2.1.3 Linear algebra, ranks, bases

2.1.4 Pseudoinverses, singular values

2.1.5 Eigenvalues, eigenvectors. Diagonalization, Jordan form

2.1.6 Matrix functions

2.1.7 Matrix input & output

2.1.8 Other

2.2 Polynomials & rational functions

2.2.1 Elementary polynomial operations

2.2.2 Linear polynomial equations

2.2.3 Roots, stability

2.2.4 Smith-form, Smith-McMillan-form

2.2.5 Spectral factorization

2.2.6 Input & output

2.2.7 Other

2.3 Functions of several variables

2.3.1 Function minimization with special structure (Linear programming, Integer programming etc.)

2.3.2 Global function minimization

2.3.3 Local function minimization without constraints

2.3.4 Local function minimization with constraints

2.3.5 Level-curves. Projections

2.3.6 Linearization. Higher derivatives

2.3.7 Non-linear equation systems

2.3.8 Other

2.4 Differential equations

2.4.1 Integration of ordinary 1st-order differential equations

2.4.2 Integration of stiff differential equations

2.4.3 Integration of ordinary higher-order differential equations

2.4.4 Integration of partial differential equations

2.4.5 Other

2.5 Statistical (cf. Data analysis, Chapter 6)

2.5.1 Elementary random numbers

2.5.2 Random numbers with given distribution

2.5.3 Distribution functions

2.5.4 Other

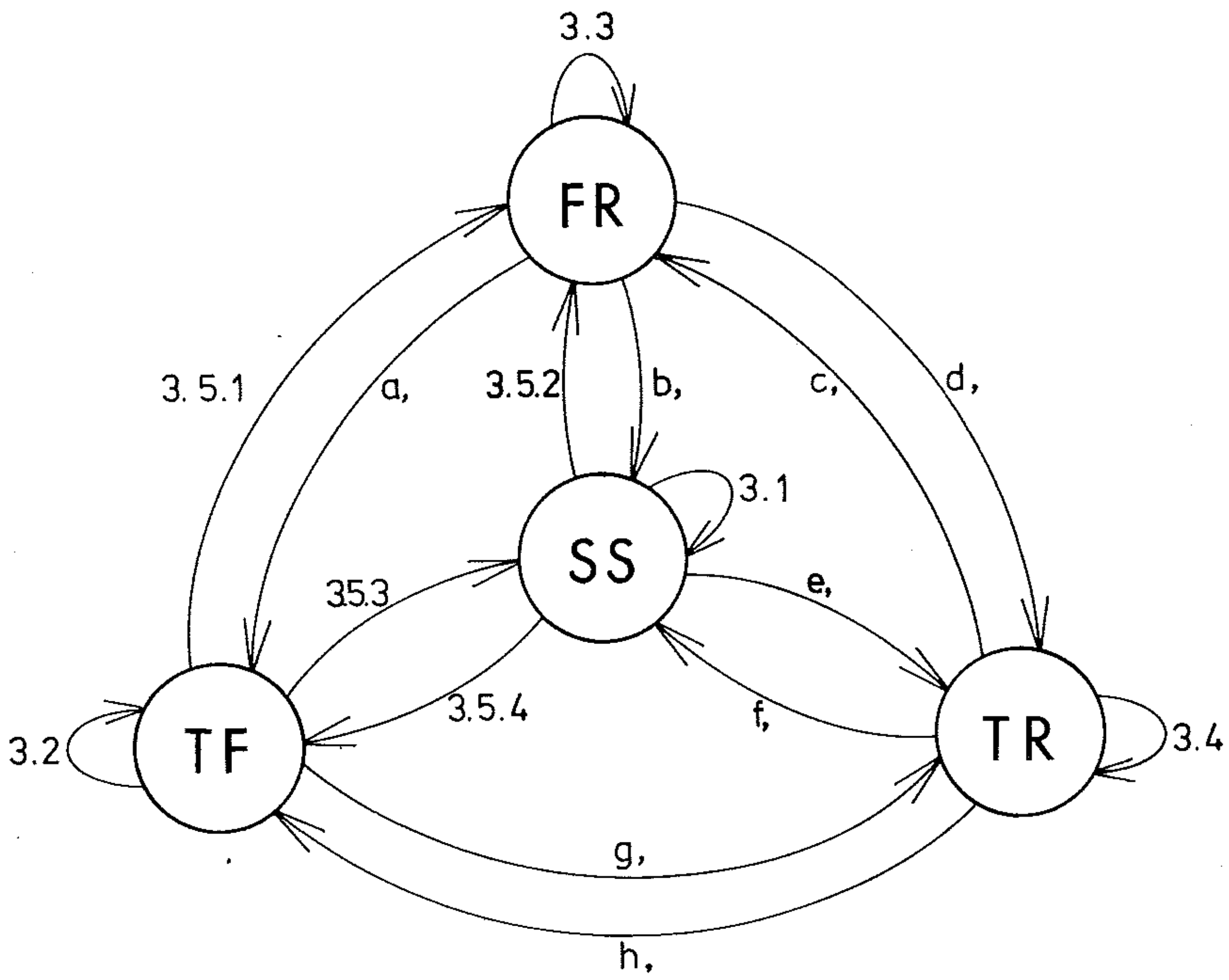
3. Analysis of linear dynamic systems

Linear dynamic systems can be represented in four different ways:

State Space representation	- SS
Transfer Function representation	- TF
Frequency Response representation	- FR
Time Response representation	- TR

NB. Transfer Function section also contains the more general representation with operator polynomials.

The following figure indicates all possible computations that can be done, whether or not the result is in the same representation as the input, and where to find the required routine. The numbers refer to sections and subsections in this chapter, the letters refer to other chapters, see explanation below.



Computations found in other chapters

a. Fitting a transfer function to a frequency response; see Identification, 7.5.

b. Fitting a state space representation to a frequency response; see Identification, 7.6.

c. Fourier transform & FFT; see Data Analysis, 6.7.

d. (Inverse) Fourier transform & FFT; see Data Analysis, 6.7.

e. Simulation of system on State Space form; see Simulation, 8.4.

f. Time response to State Space representation; see Identification, 7.4.

g. Simulation of system on transfer function form; see Simulation 8.3 (8.2).

h. Transfer function from time response; see Identification, 7.1 or 7.2.

3.1 State space representation

3.1.1 Change of coordinates. Kanonical forms

3.1.2 Controllability & observability.
Kalman-decomposition, observability indices

3.1.3 Inverse systems. Poles & zeroes

3.1.4 Conversion continuous time-discrete time

3.1.5 Interconnection of sub-systems

3.1.6 Model reduction. Linear dependence

3.1.7 Input & output

3.2 Transfer function representation and systems represented with polynomials in the differentiation operator or with Rosenbrock's system matrix.

3.2.1 Strict system equivalence transformations

3.2.2 Controllability, observability, Kalman decomposition, common factors

3.2.3 Inverse systems, poles & zeroes

3.2.4 Conversion continuous time - discrete time

3.2.5 Interconnection of subsystems

3.2.6 Model reduction

3.2.7 Input & output

3.2.8 Non-rational transfer functions

3.3 Frequency response representation.

(Systems represented via a table of complex numbers (the value of the transfer function) as a function of ω.)

3.3.1 Polar & rectangular coordinates, logarithmic representation

3.3.2 Interpolation

3.3.3 Inverse systems, Minimum-phase tests

3.3.4 Transformation continuous time - discrete time

3.3.5 Interconnection of subsystems

3.3.6 Input & output

3.3.7 Other

3.4 Time response, special case: impulse response.

(Systems represented via a table of input--output values as a function of time.)

3.4.1 Scaling; see 6.1

3.4.2 Interpolation; see 6.1

3.4.3 Causality; see 6.5 or 6.6

3.4.4 Convolution, deconvolution

3.4.5 Interconnection; see 3.4.4

3.4.6 Input & output; see 6.8

3.4.7 Other

3.5 Transformations between system representations.

NB. Many such transformations are filed under other headings, cf beginning of this chapter.

3.5.1 Transfer function to Frequency response

3.5.2 State space to Frequency response

3.5.3 Transfer function to State space. Realization algorithms.

3.5.4 State space to Transfer function

4. Synthesis

(See also 5. Linear quadratic control.)

4.1 Bode - Nichols - Nyquist

4.1.1 Synthesis by rule-of-thumb

4.1.2 Phase-amplitude margin, error coefficients etc.

4.1.3 Special purpose plotting routines

4.2 Root - Locus methods

4.2.1 Transformation to standard formulation P + KQ = 0

4.2.2 Root-finding algorithms

4.2.3 Curve following algorithms

4.2.4 Special purpose plotting routines

4.3 Pole assignment - state space

4.3.1 Single input - single output

4.3.2 Multivariable

4.3.3 Modal feedback

4.4 Pole assignment - transfer function

4.4.1 Single input - single output

4.4.2 Multivariable

4.5 Minimal variance

4.5.1 Optimal case

4.5.2 Sub-optimal case

4.6 Multivariable frequency response

4.6.1 Characteristic Loci computation

4.6.2 Inverse Nyquist array

4.6.3 Gershgorin & Ostrowsky circles

4.6.4 Special purpose plotting

4.6.5 Service routines, dominance, elementary controllers

4.7 Observers
(see also 5.3 and 5.4)

5. Linear quadratic control

5.1 Covariances and quadratic loss functions

5.1.1 State space representation

5.1.2 Transfer function representation

5.2 Transformation from continuous time form to discrete time form

5.2.1 State space representation

5.2.2 Transfer function representation

5.3 Riccati equation, solution of the time-variable case

5.3.1 Continuous time

5.3.2 Discrete time

5.4 Riccati equation, stationary solution

5.4.1 Continuous time

5.4.2 Discrete time

5.5 Solution by spectral factorization

6. Data Analysis

6.1 Scaling, interpolation

6.2 Simple statistical properties, statistical tests

6.3 Estimation (removal) of trends, periodic components etc

6.4 General transformations

6.5 Auto- & cross-covariances

6.6 Auto- & cross-spectra

6.7 Laplace- & Fourier-transforms, FFT

6.8 Input & output

7. Identification

7.1 Time response to polynomial representation. No noise model

7.1.1 Least squares

7.1.2 Recursive & real time least squares

7.1.3 Instrumental variable & tally principle

7.1.4 Other

7.2 Time response to polynomial representation. With noise model

7.2.1 Maximum Likelihood method, Generalized Least Squares

7.2.2 Recursive Maximum Likelihood

7.3 Pulse- or step-response to state space model. Realization

7.4 Time response to state space model

7.4.1 Extended Kalman

7.4.2 LISPID

7.5 Frequency response to transfer function representation

7.6 Frequency response to state space representation

7.7 Optimal test signals

8. Simulation

8.1 Generation of time series, random numbers (see also 2.5)

8.2 Simulation by frequency response methods

8.3 Simulation of linear systems on polynomial form

8.3.1 Single input - single output

8.3.2 Multivariable

8.4 Simulation of linear systems on state space form

8.4.1 Single input - single output

8.4.2 Multivariable

8.5 Simulation of non-linear systems (Integration methods: see 2.4)

9. On-line control

9.1 Analog input & output (Purdue standard)

9.2 Classical regulators. PID-regulators

9.3 General linear regulator

9.4 Kalman filter

9.5 Self-tuning regulators

9.6 Adaptive regulators