# Adaptor—A System for Process Identification and Adaptive Control

## An Implementation Report

Johansson, Rolf

1987

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

Link to publication

*Citation for published version (APA):*
Johansson, R. (1987). *Adaptor—A System for Process Identification and Adaptive Control: An Implementation Report*. (Research Reports TFRT-3190). Department of Automatic Control, Lund Institute of Technology (LTH).

*Total number of authors:*
1

CODEN: LUTFD2/(TFRT-3190)/1-24/(1987)

# Adaptor
# A System for Process Identification and Adaptive Control

## An Implementation Report

Rolf Johansson

Department of Automatic Control
Lund Institute of Technology
September 1987

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

*Document name*
Report

*Date of issue*
September 1987

*Document Number*
CODEN: LUTFD2/(TFRT-3190)/1-24/(1987)

*Author(s)*
Rolf Johansson

*Supervisor*

*Sponsoring organisation*

*Title and subtitle*
Adaptor. A System for Process Identification and Adaptive Control. An Implementation Report.

*Abstract*

An implementation of an adaptive digital regulator is described in this report. The controller is developed for general purpose adaptive digital control with many different algorithm features such as self-tuning regulators and model reference adaptive control. Several options with respect to regulator complexity are available to the operator.

The implementation is made on an IBM-AT in Modula-2. The operating system MS-DOS is used together with a kernel for real time operations and multitasking. A chapter on the interface to the operator describes the commands available to the operator during execution of the program. A simulation support facility for the regulator is also described. The program is intended for experiments with adaptive control for engineering practice, research and education.

*Key words*

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

*ISSN and key title*

*ISBN*

*Language*
English

*Number of pages*
24

*Security classification*

*Recipient's notes*

*The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.*

# ADAPTOR

—

# A SYSTEM FOR PROCESS IDENTIFICATION AND ADAPTIVE CONTROL

—

# AN IMPLEMENTATION REPORT

Rolf JOHANSSON

Department of Automatic Control,
Lund Institute of Technology,
Box 118, S-221 00 LUND,
Sweden

## ABSTRACT

An implementation of an adaptive digital regulator is described in this report. The controller is developed for general purpose adaptive digital control with many different algorithm features such as self-tuning regulators and model reference adaptive control. Several options with respect to regulator complexity are available to the operator.

The implementation is made on an IBM-AT in Modula-2. The operating system MS-DOS is used together with a kernel for real time operations and multitasking. A chapter on the interface to the operator describes the commands available to the operator during execution of the program. A simulation support facility for the regulator is also described. The program is intended for experiments with adaptive control for engineering practice, research, and education.

# CONTENTS

# 1. INTRODUCTION

An implementation of an adaptive digital regulator is described in this report. The controller is developed for general purpose adaptive digital control with many different algorithm features such as self-tuning regulators and model reference adaptive control. Several options with respect to regulator complexity are available to the operator.

The implementation is made on an IBM-AT in Modula-2. The operating system MS-DOS is used together with a kernel for real time operations and multitasking [1], [2] developed at the department.

The report begins with derivations of the implemented algorithms and explanations on the options of the regulator.

A chapter on the interface to the operator describes the commands available to the operator during execution of the program. The user's guide helps the beginner to get started. There are also some hints how to improve the adaptation results in some typical situations.

A simulation support facility for the regulator is described in chapter 4. The simulation package Simnon [3], [4] has been extended with a 'standard system' which executes the code of the adaptive regulator in a simulation environment.

The regulator has been used extensively for education purposes in regular courses (*Digital control, Process identification*) and in the extension program on adaptive control as well as system identification at the Department of Automatic Control. The program has also been used for system identification and feasibility tests of adaptive control in several master thesis projects.

# 2. THE ALGORITHMS

A short description of some regulator algorithms is given below. Derivations of the algorithms are not given here. The reader is requested to consult the following work for a theoretical motivation [5], [6], [7].

The program *Adaptor* contains a number of algorithms for adaptive control. A rich parameter set to each algorithm provides a very general setting for experiments with discrete time adaptive control. There are separate algorithms, parameter sets, and screen setups for the following algorithms:

- Model reference direct adaptive control [5]
- Self-tuning control *ad modum* Clark-Gawthrop [8]
- Indirect adaptive control based on process identification

The regulator complexity may be changed during operation without compilation. Several different parametrisations are included. Output error and equation error identification are implemented. A particular parametrisation for integral action is used in the case of model reference direct adaptive control.

## Mathematical notation

The following notations are used below:

| | |
|---|---|
| $u$ : | Control input |
| $y$ : | Measured control object signal |
| $u_c$ : | Reference signal |
| $y_m$ : | Output of the reference model |
| $v$ : | Measurable disturbance; Feed forward signal |
| $d$ : | Steplike disturbance; Low frequency disturbance |
| $\nu$ : | Disturbance |

| | |
|---|---|
| $A$ | Control object pole polynomial |
| $B$ | Control object zero polynomial |
| $b_0$ | Control object gain factor |
| $k$ | Control object time delay |
| $R$ | Regulator pole polynomial |
| $S$ | Regulator numerator polynomial w.r.t. $y$ |
| $T$ | Regulator numerator polynom w.r.t. $u_c$ |
| $\Delta$ | Difference operator; $1 - q^{-1}$ |
| $A_0$ | Observer pole polynomial $1 - tq^{-1}$ |
| $A_M$ | Reference model pole polynomial $1 - pq^{-1}$ |
| $\theta$ | Parameter vector to estimate |
| $\widehat{\theta}$ | Parameter estimate vector |
| $\varphi$ | Regression vector for estimation |
| $\varepsilon$ | Prediction error |

A mathematical description is now given for the control object assumptions and the adaptive control algorithm for model reference control without integral action. The arrows indicate the result of each expression. The algorithms are developed under the assumption that the input-output behavior of the process is well described by the following process model

– **Process model:**

$$A(q^{-1})y(t) = b_0 q^{-k} B(q^{-1})u(t) + d(t) \tag{1}$$
$$\uparrow$$

Let the regulator be a linear discrete time controller of the type

– **Regulator model:**

$$R(q^{-1})u(t) = -S(q^{-1})y(t) + T(q^{-1})u_c(t) \tag{2}$$
$$\uparrow$$

It is under these model assumption possible to find a linear regulator so that the closed-loop system corresponds to the input-output behavior of the the following reference model whose parameters should contain closed-loop system specifications in a condensed form.

– **Reference model:**

$$y_m(t) = q^{-k}\frac{A_M(1)}{A_M(q^{-1})}u_c(t); \qquad A_M(q^{-1}) = 1 - a_{m_1}q^{-1} \tag{3}$$
$$\uparrow$$

The desired model matching is achieved if $R$, $S$, and $T$ are chosen as solutions to the following equation

– **Regulator equation:**

$$R(q^{-1})A(q^{-1}) + \quad S(q^{-1})b_0 q^{-k}B(q^{-1}) = b_0 A_0(q^{-1})A_M(q^{-1})B(q^{-1}) \tag{4}$$
$$\uparrow \qquad\qquad \uparrow$$

$$T(q^{-1}) = A_0(q^{-1})A_M(1); \qquad A_0(q^{-1}) = 1 - a_{0_1}q^{-1}$$
$$\uparrow$$

There are good reasons to avoid solving the regulator equation on-line. An alternative is to solve the problem is via direct identification of parameters from the following equivalent process modell

– **Parameter estimation model:**

$$y(t) = R(q^{-1})\left(\frac{q^{-k}}{A_0(q^{-1})A_M(q^{-1})}u(t)\right) + \quad S(q^{-1})\left(\frac{q^{-k}}{A_0(q^{-1})A_M(q^{-1})}y(t)\right) = \theta^T\varphi(t)$$
$$\uparrow \qquad\qquad\qquad \uparrow$$
$$(5)$$

Parameter estimation is conveniently done with an identification method such as recursive least squares estimation

– **Identification method:**

$$\varepsilon(t) = y(t) - \widehat{\theta}^T(t-1)\varphi(t)$$
$$\uparrow$$

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$$
$$\uparrow$$

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$
$$\uparrow \tag{6}$$

## System Identification

The problem is here to find the values of $A$ and $b_0 B$ of (1) in the presence of an unknown disturbance $d$, see [9], [10]. The problem is solved via the formulation

$$y(t) = -a_1 y(t-h) - \ldots - a_{n_A} + b_0 u(t-kh) + \ldots = \theta^{\varphi}(t)$$
$$\theta^T = \begin{pmatrix} -a_1 & \ldots & -a_{n_A} & b_0 & b_0 b_1 & \ldots & b_0 b_{n_B} \end{pmatrix} \tag{7}$$

Identification of the parameters $\theta$ gives the desired result. The implementation includes a number of methods to avoid bias from the influence of $d$.

- Subtraction of the equilibrium level $u_0$, $y_0$ from signals.
- Taking differences between subsequent data points.
- Band pass filtering of data

## Adaptive Control Algorithms

### Explicit model matching

This algorithm starts the cycle with estimation of $A$ and $b_0 B$ of (1) and proceeds by solving the regulator equation for $R$, $S$ and $T$. These values are then used in the regulator (2).

### Direct adaptive regulator (implicit)

The alternative parameter model (5) is used with the parameter vector

$$\theta^T = \begin{pmatrix} r_0 & r_1 & \ldots & r_{n_R} & s_0 & s_1 & \ldots & s_{n_S} \end{pmatrix} \tag{8}$$

and the direct control law

$$u(t) = -\frac{1}{\widehat{r}_0}\Big(\widehat{r}_1(t)u(t-h) + \ldots + \widehat{r}_{n_R}(t)u(t-n_R h) +$$
$$+\widehat{s}_0(t)y(t) + \ldots + \widehat{s}_{n_S}(t)y(t-n_S h) + \widehat{t}_0 u_c(t) + \ldots + \widehat{t}_{n_T} u_c(t-n_T h)\Big) \tag{9}$$

### Clark-Gawthrop algorithm

This is similar to the direct adaptive regulator above but avoids immediate division by $\widehat{r}_0$ which may introduce errors when it is very small. The design parameter $\varrho$ is chosen by the user.

$$u(t) = -\frac{\widehat{r}_0(t)}{\varrho^2 + \widehat{r}_0^2(t)}\Big(\widehat{r}_1(t)u(t-h) + \ldots + \widehat{r}_{n_R}(t)u(t-n_R h) +$$
$$+\widehat{s}_0(t)y(t) + \ldots + \widehat{s}_{n_S}(t)y(t-n_S h) + \widehat{t}_0 u_c(t) + \ldots + \widehat{t}_{n_T} u_c(t-n_T h)\Big) \tag{10}$$

### Direct adaptive regulator with integral action

The following regulator includes an integral action and is based on differentiation of subsequent data points so that low frequency disturbances do not interfere with the identification. The regulator model is then an incremental model that automatically provides some integral action. The regulator model (4) is modified to

– **Regulator model:**

$$\underset{\uparrow}{R(q^{-1})}\Delta u(t) = -S(q^{-1})\Delta y(t) + T(q^{-1})\Delta u_c(t) + \alpha(u_c(t) - y(t)) \tag{11}$$

with the coefficient $\alpha = A_0(1)A_M(1)$ for the integral action.

The associated parameter estimation model is then

– **Parameter estimation model:**

$$y - \frac{\alpha q^{-k}}{A_0 A_M} y = \underset{\uparrow}{R}\left(\frac{\Delta q^{-k}}{A_0 A_M} u(t)\right) + \underset{\uparrow}{S}\left(\frac{\Delta q^{-k}}{A_0 A_M} y(t)\right) = \underset{\uparrow}{\theta^T} \varphi(t) \tag{12}$$

The standard practice would be to implement this as an incremental regulator. In this implementation, however, it is modified so that the integral action is explicit with $i$ as a separate state to update. The integration windup problems may then be treated by standard methods.

$$\begin{aligned} R(q^{-1})u(t) &= -S(q^{-1})y(t) + T(q^{-1})u_c(t) + i(t) \\ i(t) &= i(t-h) + \alpha(u_c(t) - y(t)) \end{aligned} \tag{13}$$

The full algorithm is given in appendix 1.

## Signal Processing Aspects

All the above algorithms are implemented are implemented with two different parametric models. Let $F(q^{-1})$ be some filtering polynomial. The two models are then of the equation error type (FILTER F)

$$F(q^{-1})y(t) = R(q^{-1})(q^{-k}u(t)) + S(q^{-1})(q^{-k}y(t)) \tag{14}$$

and the output error type (FILTER T)

$$y(t) = R(q^{-1})(\frac{q^{-k}}{F(q^{-1})}u(t)) + S(q^{-1})(\frac{q^{-k}}{F(q^{-1})}y(t)) \tag{15}$$

The first model implies high pass filtering of the regressors while the other means low pass filtering. These models are algebraicly equivalent but exhibit very different properties. The weighting of signals in the frequency domain becomes very different.

Another property is included in the system identification signal processing where an alternative model is given by

$$A(q^{-1})(\frac{\Delta}{F(q^{-1})}y(t)) = b_0 B(q^{-1})(\frac{\Delta q^{-k}}{F(q^{-1})}u(t)) + (\frac{\Delta}{F(q^{-1})}d(t)) \tag{16}$$

This signal processing means band pass filtering of regressors which is often beneficial for good closed-loop performance.

# 3. IMPLEMENTATION AND HARDWARE

The implementation is made on an IBM-AT in Modula-2 from Logitech [11]. The operating system MS-DOS is used together with a kernel for real time operations and multitasking [1], [2] developed at the department. The input-output board is a 12 bit IBM or an Analog Devices RTL-800, see [12].

### Multitasking capabilities

The context switch of the real-time kernel allows a fastest sampling rate of 100 $[Hz]$. The lowest possible practical sampling rate with a graphics interface and a control algorithm with adaptation is 20 $[Hz]$.

The external communication takes place exclusively via the input-output card and via the keybord. The 'mouse' was not utilized because initial experiments showed that performance was degraded significantly due to the necessary process changes when polling the mouse.

### Graphics capabilities

The set-up utilizes a EGA color graphics display that allows presentation of signal histories of inputs and outputs as well as parameter monitoring. Different keyboard commands causes switching between the screen modes.

### Program organization

The program is made in Modula-2 which favours a good program structure. The organization of the program is divided into the following larger modules which also run concurrently as separate processes on the computer.

- **Regulator:**
  - The regulator module handles all basic signal processing. The process runs with high priority and starts a work cycle by reading signals from the A/D-converter. Then follows filtering operations and computation of control signals with D/A-conversion.
- **Estimator:**
  - The estimator module makes parameter estimation and updating. It keeps an own data base of inputs and outputs that is updated on request. This process runs as a background process and may be interrupted by the regulator process.
- **Graphics handler:**
  - The graphics handler manages the screen which is an interface to the user. There are four different screen modes with different sets of information. Two screen modes shows parameter status of the process identification and the adaptive control algorithms. One mode shows a matrix of the identification algorithm. The fourth mode shows the process graphs of control and parameter estimation. There is also one permanent display with signal processing information.
- **Operator's interface:**
  - This module handles the operator's communication with other program modules. It contains a command interpreter with error messages for caution and safety in application. It also handles a display on the screen with information on the command history.

## Early versions of the program

An earlier version of the program with only direct adaptive control has been implemented on an LSI-11 with the operating system RT-11 and a real-time kernel that allowing the use of sequential Pascal routines in a multitasking environment. The properties were similar but with lower emphasis on the interaction.

# 4. SIMULATION SUPPORT

It is in some cases desirable to predict or verify the performance of adaptive regulator before it is checked out in the real application. Another desirable feature is to have a tool for investigation of unexpected behavior. Simulation is then of good help to support check-out of algorithms and is a complement to other design calculations. A simulation support facility around *Adaptor* has therefore been developed. The simulation package *Simnon* has been extended with a 'standard system' which executes the code of the adaptive regulator in a simulation environment.

The adaptive control program is available as a 'standard system' ADAPT with inputs and outputs as in *Adaptor*. The simulated control object and ADAPT communicate via an interface defined by a 'connecting system', see [3], [4]. A possible set-up is given by

```
Connecting system con
u[object]=u[adapt]
y[adapt]=y[object]
uc[adapt]=1.0
end
```

The variables of adaptor are referenced via *Simnon's* syntax. The command of changing the regulator complexity

```
->nS 1
```

is thus replaced in the simulation interaction by the *Simnon* expression

```
>par nS:1
```

This facility has proved very useful and valuable for testing of correctness of the coding of various adaptive control algorithms.

# 5. OPERATOR'S INTERFACE

The operator may interact with the program via the keyboard and the color graphics screen. Most information is displayed on the screen as text and numbers or as graphs. A command interpreter also gives necessary error messages for caution and safety in application. It also handles a display on the screen with information on the command history. The central part of the screen is as follows.

```
=========================================================
RECURSIVE LEAST SQUARES IDENTIFICATION
---------------------------------------------------------
Regulatormode: Stop                    tsamp = 1.000
Estim = false       Diff = false       ulow  = 0.000
Tune  = false       Ff   = false       uhigh = 1.000

Regulator
 nR = 0      R =     1.0000
 nS = 0      S =     1.0000
 nT = 1      T =     0.0000    0.0000
 nV = -1      V =


Estimator
 nA = 1      A =     1.0000    0.0000
 nB = 0      B =     0.0000
 nC = -1      C =

 nF = 0      F =     1.0000
 k = 1       Lambda =     0.9900

---------------------------------------------------------
```

## Input and Output Limitations

All signals need to be limited to the range -10 — +10 V.

## Command Structure

The system prompts with '→' and is then ready to accept commands. The commands should have the following structure after the prompting arrow:

```
-> <command> <value>
```

The 'commands' are described below. The 'values' for the commands are of five different types namely :

**r** A real number e.g. 0.99

**i** An integer e.g. 3

**p** Polynomial coefficients e.g. 1.0 -0.20 0.01

| | |
|---|---|
| l | A logical value i.e. T (true) or F (false) |
| t | A character string: 'A', 'B', 'C', 'R', 'S', 'V', 'SIG' |

A command interpreter also gives necessary error messages for caution and safety in application. It also handles a display on the screen with information on the command history.

## Screen Modes

The set-up utilizes a EGA color graphics display that allows presentation of signal histories of inputs and outputs as well as parameter monitoring. There are separate algorithms, parameter sets, and screen setups for algorithms concerning model reference direct adaptive control, self-tuning control *ad modum* Clark-Gawthrop, and indirect adaptive control based on process identification.

| | |
|---|---|
| **INDIRE l** | Request of recursive least squares identification or indirect adaptive control. |
| **DIRECT l** | Request of direct adaptive control. |
| **CLARKG l** | Clark-Gawthrop regulator (toggles T/F). |
| **BACKUP** | Show and edit the backup menu. |
| **SHOW SIG** | Show process graphs with $y$, $u$, $u_c$, and $v$. |
| **SHOW A** | History of the $A-$polynomial of the estimated system parameters. |
| **SHOW B** | History of the $B-$polynomial of the estimated system parameters. |
| **SHOW C** | History of the $C-$polynomial of the estimated system parameters. |
| **SHOW R** | History of the $R-$parameters of the regulator. |
| **SHOW S** | History of the $S-$parameters of the regulator. |
| **SHOW V** | History of the $V-$parameters of the regulator. |
| **P** | Show the P-matrix of (6). |
| **SHOW** | Show the status of the active regulator. |

## Regulator Modes

The activation and suspension of regulators is made by the following commands.

| | |
|---|---|
| **MANUAL** | Request of manual control. |
| **RUN** | Request of automatic control. |
| **STOP** | Stop all regulator activity. |
| **EXIT** | Exit to MS-DOS. |

## Function Modes

Switching between separate regulators is made with the commands.

| | |
|---|---|
| **DIRECT l** | Request of direct adaptive control. |
| **INDIRE l** | Request of recursive least squares estimation or indirect adaptive control. |

The following screen is associated with direct adaptive control.

```
=========================================================
ADAPTIVE REGULATOR
---------------------------------------------------------
Regulator Mode: STOP
                                             tsamp = 1.000
Estim = FALSE          Integ= TRUE           ulow  = 0.000
Tune = FALSE           FF   = FALSE          uhigh = 1.000

Regulator
 nR = 0     R =    1.0000
 nS = 0     S =    1.0000
 nT = 1     T =    0.0000   0.0000
 nV = -1     V =


Estimator
 nR = 0     R =    1.0000
 nS = 0     S =    1.0000
 nV = -1    V =
 nF = 2     F =    1.0000   -1.0000 0.2100
 nA0= 1     A0=    1.0000   -0.7000
 nAm= 1     Am=    1.0000   -0.3000

 k = 1   Lambda =    0.9900

---------------------------------------------------------
```

## Regulator Complexity

The regulator complexity may be changed during operation by the following commands.

| | |
|---|---|
| **ESTIM l** | Start/stop of parameter estimation. |
| **TUNE l** | Start/stop of regulator tuning. |
| **INTEG l** | Integral action (toggles T/F). |
| **FF l** | Feedforward control from $v$ (toggles T/F). |
| **CLARKG l** | Clark-Gawthrop regulator (toggles T/F). |
| **R0 r** | Set parameter value of $\varrho$ in Clark-Gawthrop regulator. |
| **NR i** | Polynomial degree of $R$. |
| **NS i** | Polynomial degree of $S$. |
| **NT i** | Polynomial degree of $T$. |
| **NV i** | Polynomial degree of $V$. |
| **NA0 i** | Polynomial degree of $A_0$. |
| **NAm i** | Polynomial degree of $A_m$. |
| **R p** | Assignment of regulator polynomial coefficients to $R$. |
| **S p** | Assignment of regulator polynomial coefficients to $S$. |
| **T p** | Assignment of regulator polynomial coefficients to $T$. |
| **V p** | Assignment of regulator polynomial coefficients to $V$. |

| | |
|---|---|
| **A0 p** | Assignment of regulator polynomial coefficients to $A_0$. |
| **Am p** | Assignment of regulator polynomial coefficients to $A_m$. |

## Regulator Limitations

The operating ranges and the limitations of the regulator may be set with the following commands.

| | |
|---|---|
| **TSAMP r** | Assignment of sampling period. |
| **ULOW r** | Set lower limitation of $u$. |
| **UHIGH r** | Set upper limitation of $y$. |
| **R0 r** | Control signal weighting of the Clark-Gawthrop regulator. |

## System Identification

| | |
|---|---|
| **ESTIM r** | Parameter estimation is toggled T/F. |
| **DIFF l** | Differentiation of input-output data in estimation algorithms. |
| **FILTER l** | High pass filtering (F) or low pass filtering (T) of regressors. |
| **NA i** | Polynomial degree of the $A-$polynomial. |
| **NB i** | Polynomial degree of the $B-$polynomial. |
| **NC i** | Polynomial degree of the $C-$polynomial. |
| **NF i** | Polynomial degree of the filter polynomial $F$. |
| **A p** | Assignment of initial values to the $A-$polynomial. |
| **B p** | Assignment of initial values to the $B-$polynomial. |
| **C p** | Assignment of initial values to the $C-$polynomial. |
| **F p** | Assignment of value to the filter polynomial $F$. |
| **K i** | Set the prediction horizon. |
| **LAMBDA r** | Set the forgetting factor of (6). |
| **P** | Show the P-matrix. |
| **P0 r** | Set initialization value of the diagonal elements of the P-matrix. |
| **EQUIL** | Read equilibrium input-output values. The result is subtracted from regressors. |
| **EQUIL0** | Set the equilibrium to zero. |

## Signal Processing

The signal processing is a facility for convenient generation of reproducable reference signals. This is helpful for design of standard experiments.

| | |
|---|---|
| **EXTREF l** | Choose external reference value $u_c$ from AD0. |
| **INTREF l** | Choose reference value $u_c$ from internal square wave signal generator. |
| **AMP r** | Choose the amplitude of the internal square wave generator. |
| **PER r** | Choose the period $[s]$ of the internal square wave generator. |
| **MEAN r** | Choose the mean value of the internal square wave generator. |
| **DELAY i** | Make a delay of the control signal $u$ for $i$ sampling intervals. |

## Graphical Output

| | |
|---|---|
| **SHOW SIG** | Show process graphs. |
| **SHOW A** | History of the $A-$polynomial of the estimated system parameters. |
| **SHOW B** | History of the $B-$polynomial of the estimated system parameters. |
| **SHOW C** | History of the $C-$polynomial of the estimated system parameters. |
| **SHOW R** | History of the $R-$parameters of the regulator. |
| **SHOW S** | History of the $S-$parameters of the regulator. |
| **SHOW V** | History of the $V-$parameters of the regulator. |
| **P** | Show the P-matrix. |
| **SHOW** | Show the active regulator. |

There are four different screen modes with different sets of information. Two screen modes shows parameter status of the process identification and the adaptive control algorithms. One mode shows the P-matrix of the identification algorithm. The fourth mode shows the process graphs of control and parameter estimation.

## Backup Facility

The backup possiblity is valuable in exeperimental work. A safe and known regulator stored as backup may be called upon in some situation that develops unfavourably during an experiment. The following commands affect the operation.

| | |
|---|---|
| **BACKUP** | Show backup regulator. |
| **LOAD** | Let the backup regulator resume active status. |
| **SAVE** | Save current active regulator as a backup regulator. |
| **SHOW** | Show the active regulator. |

# 6. USER'S GUIDE

There are analog inputs (AD) and outputs (DA) of the IBM-AT as follows:

| | | |
|---|---|---|
| AD 0: | $u_c$ | Reference signal |
| AD 1: | $y$ | Measured controlled variable |
| AD 2: | $v$ | Feed forward signal; Measurable disturbance |
| DA 0: | $u$ | Control variable |

Make sure that the Modula-2 system and the real time kernel are available with correct path specifications. The program *Adaptor* may now be started on the IBM-PC with the command

C:\ >m2 Adaptor

The program starts and prompts with an arrow '→' and the following screen with the regulator status appears

```
=========================================================
ADAPTIVE REGULATOR
---------------------------------------------------------
Regulator Mode: STOP
                                          tsamp = 1.000
Estim = FALSE          Integ= TRUE        ulow  = 0.000
Tune = FALSE           FF    = FALSE      uhigh = 1.000

Regulator
 nR = 0      R =    1.0000
 nS = 0      S =    1.0000
 nT = 1      T =    0.0000    0.0000
 nV = -1      V =


Estimator
 nR = 0      R =    1.0000
 nS = 0      S =    1.0000
 nV = -1     V =
 nF = 2      F =    1.0000  -1.0000 0.2100
 nA0= 1      A0=    1.0000  -0.7000
 nAm= 1      Am=    1.0000  -0.3000

 k = 1    Lambda =    0.9900

---------------------------------------------------------
```

All program parameters may now be changed via keyboard commands found in the appendix 2. The following example shows a possible start-up.

## Example 1 - A Session of System Identification

Consider the following command sequence.

```
->INDIRE
```

```
->MANUAL
->nA 1
->S 1.0 -0.5
->k 2
->TSAMP 3
->ESTIM T
```

The sequence is started with a request of the identification mode and manual control and proceeds with command to set the polynomial degree of $S(q^{-1})$ equal to 1. The polynomial coefficients of $A(q^{-1})$ then obtain their new initial values 1.0 and -0.5. The prediction horizon $k$ is then assigned the value 2 and the sampling interval is set to 3 $[s]$. Finally, the system identification is started with the command "ESTIM T". The parameter convergence may now be demonstrated by showing graphs of the parameter histories.

```
->SHOW A
->SHOW B
->P
```

The graph may be updated by hitting 'return'. The program returns to the alfanumerical mode with the command

```
->SHOW
```

The programs stops execution with the command

```
->EXIT
```

## Example 2 - Starting up Adaptive Control

Let the control object be a water tank with the inflow as the control variable $u$ and the water level as the measured output $y$. The control objective is supposed to be level control of $y$. The process is roughly described by the nominal transfer function

$$G(s) = \frac{b}{s+a} \approx \frac{3.4}{1+s70.0}$$

around a certain water level $y_0$. There is however a considerable variation between individual copies and adaptive control is motivated. A discrete time model with the sampling interval $h$ $[s]$ then obtains the form

$$y(t+h) = -a_1 y(t) + b_0 u(t)$$

with

$$a_1 = -\exp(-ah) \qquad b_0 = \frac{b}{a}(1-\exp(-ah))$$

From this background knowledge we conclude that direct adaptive control may be feasible with the polynomial degrees assigned the values $nR = nS = 1$, $nV = -1$ (no feed forward) with prediction horizon $k = 2$. This information enters the program via the following commands

```
->direct
```

```
->nR 0
->nS 1
->nV -1
->k 2
```

Adaptive control may now start. It is reasonable to start in manual mode when no regulator is available. The transient properties of an adaptation may however also be studied. Choose e.g. the following command sequence

```
->TSAMP 3
->AO 1 -0.7
->AM 1 -0.3
->INTEG T
```

The samling period is set to 3 [$s$]. Then follows specifications on the desired observer pole polynomial and the reference model pole polynomial which are assigned pole locations in 0.3 and 0.7, respectively. Adaptation is now started e.g. via the command sequence

```
->MANUAL
->ESTIM T
->TUNE T
->RUN
```

The resulting parameters may be saved via the command "SAVE" for later use. The session may be finished by the command

```
->EXIT
```

which returns the control to MS-DOS.

# 7. EXPERIENCES

### Implementation experiences

The check-out time proved to be very time consuming due to the 'combinatoric explosion' that appears also in a program of moderate size when there are many features.

Exploitation of the 'mouse' as a convenient tool of the operator's interaction with the program was a disappointment. There is a clear interference between the demands for efficient real-time operation and the necessary interrupt handling for interaction with the 'mouse'. The sampling frequency decreases significantly and inacceptably for real time use.

The programming language Modula-2 has proved to be a good and versatile language for implementation of these algorithms. The slow maximal sampling rate of 100 $[Hz]$ is to blame on the relatively slow context switches in the real time kernel. A faster program would be possible to construct by using a simpler foreground -background scheduler with the algorithm coding in assembler language.

The simulation facility has proved very valuable for testing correctness during the implementation work.

### Adaptive control experiences

The program *Adaptor* has been used frequently and in large scale for educational purposes for technology students and also in extension courses for practising engineers. It has become a good educational tool to demonstrate properties of adaptive control.

The simulation facility has proved very valuable for testing correctness during the implementation work. It is naturally of good help to distinguish between algorithm errors and, say, errors due to faults in the real time operation.

Nevertheless, there are significant differences between conclusions drawn from simulations and conclusions from practice. Two striking differences between the simulation world and the program in real operation are the following:

- The sampling frequency needs to be higher in real time operation than what may be expected from simulation experiences. It is very desirable to counteract and monitor disturbances more often than what is suggested by many theoretically motivated rules-of-thumb.

  A significant reason behind this discrepancy is probably the fact that disturbance models are often the least sophisticated model component of a simulation. A common mistake in simulation is to start a disturbance in synchronism with a sampled model. The intersampling behaviour is then neglected.

- The output error oriented adaptive algorithms with filtered regressors are superior to the equation error type algorithms.
  - Better low frequency properties
  - Better parameter convergence properties
  - Better robustness w.r.t. choice of sampling interval and model errors.

  The improved behaviour is reasonable because the regressors are weighted with respect to the desired closed loop system bandwidth irrespectible of the sampling interval.

# 8. REFERENCES


[1.] T.Essebo, R.Johansson, M.Lenells, and L.Nielsen, "A facility for executing concurrent processes in Pascal", Report 7194, Department of Automatic Control, Lund Instititute of Technology, Sweden, 1980.

[2.] Elmqvist, H., Mattson, S.E., Olsson, G., *Datorer i reglersystem - Realtidsprogrammering*, Department of Automatic Control, Lund Instititute of Technology, Sweden, 1981, ch.10 ff.

[3.] Elmqvist, H. "Simnon - An interactive simulation program for nonlinear systems", Report 3091, Department of Automatic Control, Lund Instititute of Technology, Sweden, 1975.

[4.] Åström, K.J., "A simnon tutorial", Report 3176, Department of Automatic Control, Lund Instititute of Technology, Sweden, 1981.

[5.] Landau, Y.D., *Adaptive control - The model reference approach*, Marcel Dekker, New York, 1979.

[6.] Goodwin, G.C. and Sin K.S., *Adaptive filtering, prediction and control*, Prentice-Hall, 1984

[7.] Åström, K.J. and Wittenmark, B. , *Computer controlled systems*, Prentice Hall, 1984, 343ff.

[8.] Clark, D.W. and Gawthrop, P.J., "Self-tuning controllers', Proceedings IEE 122, pp.929-934.

[9.] Goodwin, G.C. and Payne, R.L., *Dynamic systems identification - Experiment design and data analysis*, Academic Press, 1977

[10.] Ljung, L. and Söderström, T., *Theory and practice of recursive identification*, MIT Press, Cambridge, 1983.

[11.] *Modula-2/86 User's Manual*, Document LU-GU101-3, Logitech, Inc., Apples, Switzerland, 1985.

[12.] *RTL-800/815 User's manual*, Analog Devices and International Business Machines Corporation, 1986.


## Acknowledgements


The structure for a Pascal interface to Simnon has been developed by Leif Andersson. Karl Johan Åström has provided an algorithm of model reference adaptive control with integral action.

# APPENDIX 1 - Adaptive control algorithms

## Adaptive regulator without integral action

- Process model:

$$A(q^{-1})\underset{\uparrow}{y(t)} = b_0 q^{-k} B(q^{-1}) u(t) + d(t)$$

- Regulator model:

$$R(q^{-1})\underset{\uparrow}{u(t)} = -S(q^{-1})y(t) + T(q^{-1})u_c(t)$$

- Reference model:

$$\underset{\uparrow}{y_m(t)} = q^{-k}\frac{A_M(1)}{A_M(q^{-1})}u_c(t); \qquad A_M(q^{-1}) = 1 - a_{m_1}q^{-1}$$

- Regulator equation:

$$\underset{\uparrow}{R}(q^{-1})A(q^{-1}) + \underset{\uparrow}{S}(q^{-1})b_0 q^{-k}B(q^{-1}) = b_0 A_0(q^{-1})A_M(q^{-1})B(q^{-1})$$

$$\underset{\uparrow}{T}(q^{-1}) = A_0(q^{-1})A_M(1); \qquad A_0(q^{-1}) = 1 - a_{0_1}q^{-1}$$

- Parameter estimation model:

$$y(t) = \underset{\uparrow}{R}(q^{-1})\left(\frac{q^{-k}}{A_0(q^{-1})A_M(q^{-1})}u(t)\right) + \underset{\uparrow}{S}(q^{-1})\left(\frac{q^{-k}}{A_0(q^{-1})A_M(q^{-1})}y(t)\right) = \theta^T\varphi(t)$$

- Identification method:

$$\underset{\uparrow}{\varepsilon(t)} = y(t) - \widehat{\theta}^T(t-1)\varphi(t)$$

$$\underset{\uparrow}{P(t)} = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$$

$$\underset{\uparrow}{\widehat{\theta}(t)} = \widehat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$

**Adaptive regulator with integral action**

– Process model:

$$A(q^{-1})y(t) = b_0 q^{-k} B(q^{-1})u(t) + d(t)$$

$$\uparrow$$

$$A(q^{-1})\Delta y(t) = b_0 q^{-k} B(q^{-1})\Delta u(t) + \nu(t)$$

$$\uparrow$$

– Regulator model:

$$R(q^{-1})\Delta u(t) = -S(q^{-1})\Delta y(t) + T(q^{-1})\Delta u_c(t) + \alpha(u_c(t) - y(t))$$

$$\uparrow$$

$$\alpha = A_0(1)A_M(1)$$

$$\uparrow$$

– Reference model:

$$y_m(t) = q^{-k}\frac{A_M(1)}{A_M(q^{-1})}u_c(t); \qquad A_M(q^{-1}) = 1 - a_{m_1}q^{-1}$$

$$\uparrow$$

– Regulator equation:

$$\Big(R(q^{-1})\Delta\Big)A(q^{-1}) + \Big(\quad S(q^{-1})\Delta + \alpha\Big)b_0 q^{-k}B(q^{-1}) = b_0 A_0(q^{-1})A_M(q^{-1})B(q^{-1})$$

$$\uparrow \qquad\qquad\qquad \uparrow$$

$$T(q^{-1})\Delta + \alpha = A_0(q^{-1})A_M(1)$$

$$\uparrow$$

– Parameter estimation model:

$$y - \frac{\alpha q^{-k}}{A_0 A_M}y = R\left(\frac{\Delta q^{-k}}{A_0 A_M}u(t)\right) + \quad S\left(\frac{\Delta q^{-k}}{A_0 A_M}y(t)\right) = \theta^T\varphi(t)$$

$$\uparrow \qquad\qquad\qquad \uparrow \qquad\qquad\qquad \uparrow$$

– Identification method:

$$\varepsilon(t) = y(t) - \frac{\alpha}{A_0 A_M}y(t) - \widehat{\theta}^T(t-1)\varphi(t)$$

$$\uparrow$$

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$$

$$\uparrow$$

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$

$$\uparrow$$

# APPENDIX 2 - Summary of commands

The system prompts with '→' and is then ready to accept commands. The commands should have the following structure after the prompting arrow:

```
-> <command> <value>
```

The 'values' for the commands are of five different types namely :

| | |
|---|---|
| r | A real number e.g. 0.99 |
| i | An integer e.g. 3 |
| p | Polynomial coefficients e.g. 1.0 -0.20 0.01 |
| l | A logical value i.e. T (true) or F (false) |
| t | A character string: 'A', 'B', 'C', 'R', 'S', 'V', 'SIG' |

The following lists of commands are made in alfabetical order.

## On/Off

| | |
|---|---|
| **CLARKG l** | Clark-Gawthrop regulator (toggles T/F). |
| **DIFF l** | Differentiation of input-output data in estimation algorithms. |
| **DIRECT l** | Request of direct adaptive control. |
| **ESTIM l** | Start/stop of parameter estimation. |
| **EXIT** | Exit to MS-DOS. |
| **EXTREF l** | Choose external reference value from AD0. |
| **FF l** | Feedforward control from $v$ (toggles T/F). |
| **INDIRE l** | Request of recursive least squares identification or indirect adaptive control. |
| **INTEG l** | Integral action (toggles T/F). |
| **INTREF l** | Choose reference value $u_c$ from internal square wave signal generator. |
| **LOAD** | Let the backup regulator resume active status. |
| **MANUAL** | Request of manual control. |
| **RUN** | Request of automatic control. |
| **SAVE** | Save current active regulator as a backup regulator. |
| **STOP** | Stop all regulator activity. |
| **TUNE l** | Start/stop of regulator tuning. |

## Polynomial degrees

| | |
|---|---|
| **NA i** | Polynomial degree of the $A-$polynomial. |
| **NA0 i** | Polynomial degree of observer polynomial $A_0$. |
| **NAm i** | Polynomial degree of reference model polynomial $A_m$. |
| **NB i** | Polynomial degree of the $B-$polynomial. |
| **NC i** | Polynomial degree of the $C-$polynomial. |
| **NF i** | Polynomial degree of the filter polynomial $F$. |

| | |
|---|---|
| **NR i** | Polynomial degree of $R$. |
| **NS i** | Polynomial degree of $S$. |
| **NT i** | Polynomial degree of $T$. |
| **NV i** | Polynomial degree of $V$. |

## Polynomials

| | |
|---|---|
| **A p** | Assignment of initial values to the $A-$polynomial. |
| **A0 p** | Assignment of regulator polynomial coefficients to $A_0$. |
| **Am p** | Assignment of regulator polynomial coefficients to $A_m$. |
| **B p** | Assignment of initial values to the $B-$polynomial. |
| **C p** | Assignment of initial values to the $C-$polynomial. |
| **F p** | Assignment of value to the filter polynomial $F$. |
| **R p** | Assignment of regulator polynomial coefficients to $R$. |
| **S p** | Assignment of regulator polynomial coefficients to $S$. |
| **T p** | Assignment of regulator polynomial coefficients to $T$. |
| **V p** | Assignment of regulator polynomial coefficients to $V$. |

## Screen management

| | |
|---|---|
| **BACKUP** | Show the backup regulator. |
| **P** | Show the P-matrix of the identification algorithm (6). |
| **SHOW** | Show the status of the active regulator. |
| **SHOW A** | History of the $A-$polynomial of the estimated system parameters. |
| **SHOW B** | History of the $B-$polynomial of the estimated system parameters. |
| **SHOW C** | History of the $C-$polynomial of the estimated system parameters. |
| **SHOW R** | History of the $R-$parameters of the regulator. |
| **SHOW S** | History of the $S-$parameters of the regulator. |
| **SHOW V** | History of the $V-$parameters of the regulator. |
| **SHOW SIG** | Show process graphs. |

## Value assignments

| | |
|---|---|
| **AMP r** | Choose the amplitude of the internal square wave generator. |
| **DELAY i** | Make a delay of the control signal $u$ for $i$ sampling intervals. |
| **EQUIL** | Read equilibrium input-output values. The result is subtracted from regressors. |
| **EQUIL0** | Set the equilibrium to zero. |
| **K i** | Set the prediction horizon to $i$ sampling periods. |
| **LAMBDA r** | Set the forgetting factor of the recursive least squares identification (6). |
| **MEAN r** | Choose the mean value of the internal square wave generator. |
| **PER r** | Choose the period [$s$] of the internal square wave generator. |

| | |
|---|---|
| **P0 r** | Set initialization value of the diagonal elements of the P-matrix. |
| **R0 r** | Set the control signal weighting $\varrho$ of the Clark-Gawthrop regulator. |
| **TSAMP r** | Assignment of sampling period. |
| **UHIGH r** | Set upper limitation of $y$. |
| **ULOW r** | Set lower limitation of $u$. |