



**University of Applied Sciences
Zittau/Görlitz**
Theodor-Körner-Allee 16
D-02763 Zittau



**Technical University of
Liberec**
Studentska 2, 461 17 Liberec 1
Czech Republic

**Faculty of
Electrical Engineering
and Informatics**

**Faculty of
Mechatronics and Interdisciplinary
Engineering Studies**

Experiment Instructions
September 2019

RSCrC
**Rotation Speed Control of a DC Motor
with rigid Coupling**

Dr.-Ing. Steffen Gärtner
Doc. Ing. Osvald Modrlák



Faculty of Electrical Engineering and Informatics
Process Automation / Measurement Engineering



Contents

1	Goal of the experiment	3
2	Basic principles of identification and control	3
2.1	Analysis and parameter identification of a controlled system	3
2.1.1	Model of the controlled process	3
2.1.2	Working point and identification measuring	5
2.1.3	Model structure	7
2.1.4	Structure of an Off-line parameter estimation with LTI model	7
2.1.5	Optimization function in MATLAB	8
2.2	Tools for parametric identification	8
2.2.1	Program and functions in MATLAB	8
2.3	Feedback Control with PI controller	11
2.3.1	Closed – loop feedback structure	11
2.3.2	P controller/proportional mode: proportional only controller	12
2.3.3	PI controller: proportional-integral controller	12
2.4	Control parameter optimization	13
2.4.1	Hand tuning	13
2.5	Controller tuning based on numerical optimization	15
2.5.1	PID Control as a minimization	15
2.5.2	SIMULINK programs	15
2.5.3	MATLAB optimization program	17
3	Set-up of the experiment	19
4	Operating the equipment used in the experiment	20
5	Conducting the experiment	21
5.1	Tasks of the experiment	21
5.2	Analysing of the controlled system	21
5.2.1	Measuring of the static behaviour $U_{out} = f(U_{in})$	21
5.2.2	Recording of the step response and the influence of external disturbances (load) on the controlled system	23
5.3	Modelling	24
5.3.1	Modelling of the Controlled System	24
5.3.2	Theoretical analysis of the control loop	25
5.3.3	Experimentation on the control loop model	25
5.3.4	Numerical optimization of the cost function	25
5.3.5	Studies on the real control loop	25
6	Analysis of the experiment	27
7	Literature	27

1 Goal of the experiment

- Analysis of a motor generator system
- Identification, modelling and verification using the MATLAB
- Controller parameter optimisation
- Handling of a Programmable Logic Controller (PLC)

2 Basic principles of identification and control

Recall what we understand under a parameter identification. The model parameter identification is to estimate parameters of a LTI model by the given structure based on the observed input-output data of the system.

2.1 Analysis and parameter identification of a controlled system

2.1.1 Model of the controlled process

A general structure of a controlled process with its elements is shown in figure 2.1-1.

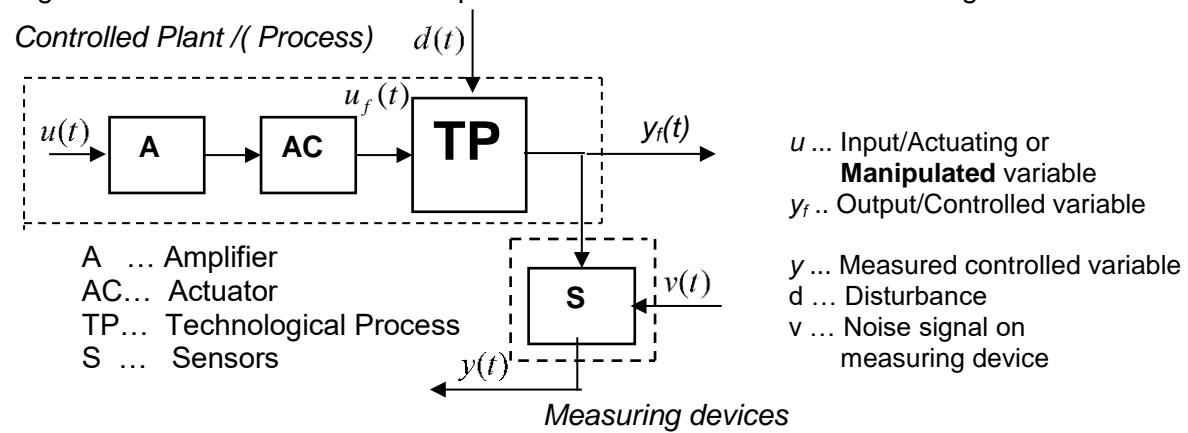


Fig. 2.1-1: Scheme of a controlled plant/process with measuring devices

Actuating/Manipulated variable $u(t)$ – is a single input variable that can be adjusted by controller or available for manual experiments.

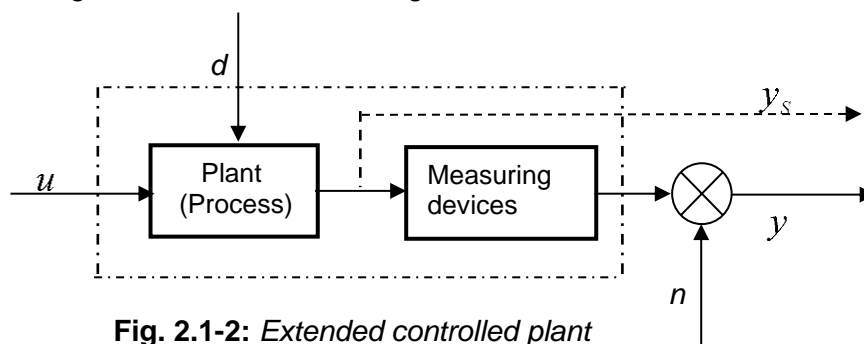
Controlled variable $y_s(t)$ – is the process selected output variable to be controlled.

Measured controlled variable $y(t)$

Disturbances $d(t)$: In addition to the manipulated variable the controlled plant has inputs, which change the measured controlled variable. These inputs are called disturbances.

Disturbance - Noise signal on measuring devices $n(t)$ – is a stochastic signal.

The extended controlled plant/process consists of plant/process and measuring devices (see figure 2.1-2) The plant contains transducers, transmitters, amplifiers and actuators. The measuring device embodies sensing elements, transducers, transmitters or sensors.



According to the Norm DIN 19221 two **linear time invariant models** can be defined for the controlled plant proposing that the disturbances $n(t), d(t) = 0$:

- 1) An extended controlled plant model determined by the transfer function (figure 2.1-3a). The transfer function **notation** is simplified by lumping all instrumentation and process dynamic into one term $F_U(s)$. It can be written

$$F_U(s) = Y_U(s) / U(s) = \frac{B(s)}{A(s)} \quad (2.1-1)$$

Where: $Y_U(s) = L\{y_U(t)\}$, $U(s) = L\{u(t)\}$, $A(s), B(s)$ are polynomials,
 $y_U(t)$ is the relating output to the input $u(t)$.

- 2) A linear model that approximates the controlled plant by two transfer functions $P(s), H(s)$ is shown in the figure 2.1-3b. The transfer function $P(s)$ approximates the dynamic behaviors of the plant (without the measuring devices) and $H(s)$ approximates the dynamic behaviors of the measuring devices.

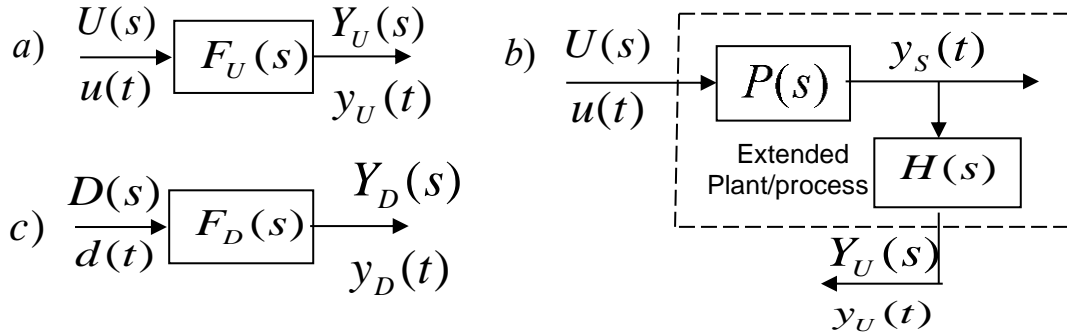


Fig 2.1-3: Model structures of extended plant/process

Fig 2.1-3b: Model structure of controlled process approximated by $P(s), H(s)$ transfer functions.

The relationship between the disturbance $d(t)$ and the measured controlled variable supposing that $n(t), u(t) = 0$ (shown in figure 2.1-3c) can be described by the transfer function ($C(s)$ is polynomial)

$$F_D(s) = Y_D(s) / D(s) = \frac{C(s)}{A(s)} \quad (2.1-2)$$

Under these conditions the model structure of an extended controlled plant with modeled disturbance d is shown in the figure 2.1-4.

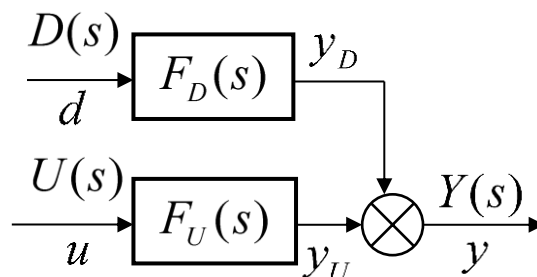


Fig. 2.1-4: Model structure of an extended plant with modeled disturbance

The output of the model is

$$y(t) = y_U(t) + y_D(t).$$

The output of the system for joint action of the manipulated variable and of the disturbance is equal to

$$Y(s) = F_{U(2-3)}(s) \cdot U(s) + F_D(s) \cdot D(s) = \frac{B(s)}{A(s)} U(s) + \frac{C(s)}{A(s)} D(s). \quad (2.1-3)$$

2.1.2 Working point and identification measuring

For an identification measurement the static characteristic has to be measured that gives the information about static behavior of the system and about the linearity of the system.

Linear time-invariant (LTI) systems have the static characteristic as a straight line which goes through the system of coordinates.

Therefore for a LTI model approximation this requirement must be fulfilled. Practically on the basis of the measured static characteristic the working point (u_0, y_0) must be selected. The chosen working point shall be in the linear region of the static characteristic. The variation of input is to be carried out around the working point where the part of the static characteristic is a linear part. That is shown in the figure 2.1-5.

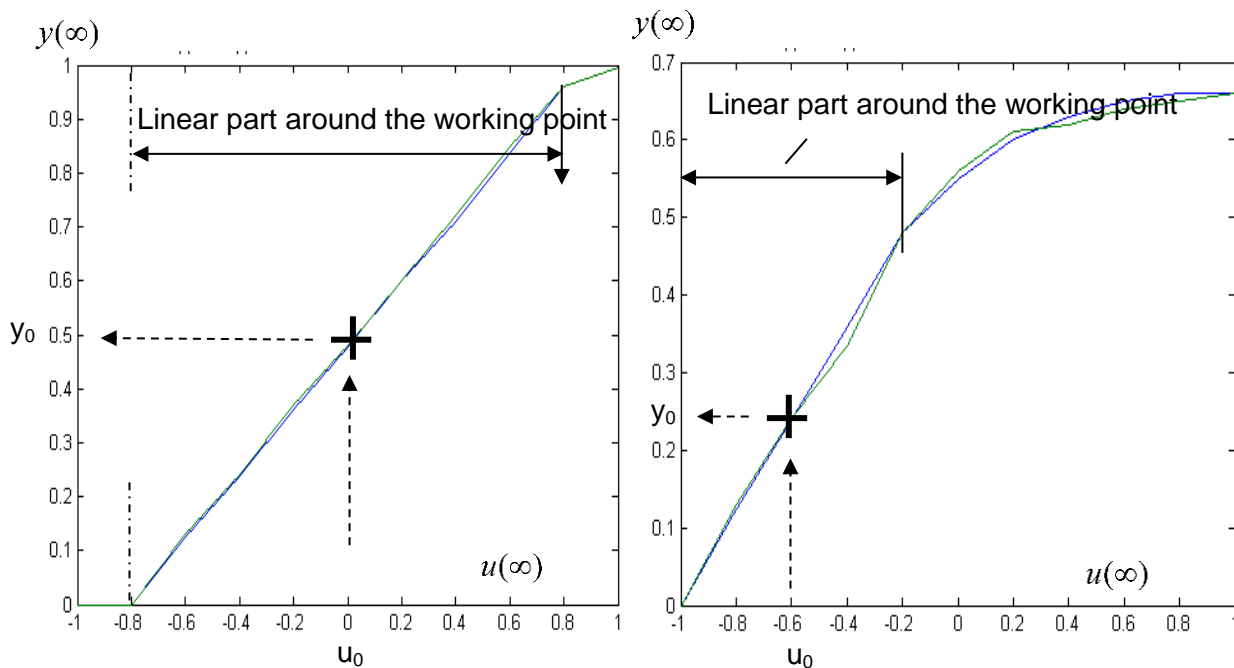


Fig. 2.1-5: Choice of working points on a measured static characteristic

As an input test signal we are using ***N* step signals**. The measuring cycles have constant time of measuring cycles TC (see figure 2.1-6). During each measured cycle the amplitudes of the manipulated variables $u_k = u_k(t)$ for $k=0 \dots N$ are constant.

It is recommended that the time period TC is so long that the steady state of the measured variable is reached. Each measuring cycle has a constant number *M* of measured samples $u(t), y(t)$ (see figure 2.1-6). The first step u_0 must excite the system to the working point (u_0, y_0) that is shown in the figure 2.1-6.

Then the measured data for identification data processing of measured input and output must be transformed to the working point. The transformation equations are

$$\Delta u_{ID}(t) = u(t) - u_0, \quad \Delta y_{ID}(t) = y(t) - y_0, \quad (2.1-4)$$

where $\Delta u_{ID}(t), \Delta y_{ID}(t)$ = Ordinates that are input to identification processing,
 $u(t), y(t)$ = Measured input and outputs,
 u_0, y_0 = Coordinates of input working point.

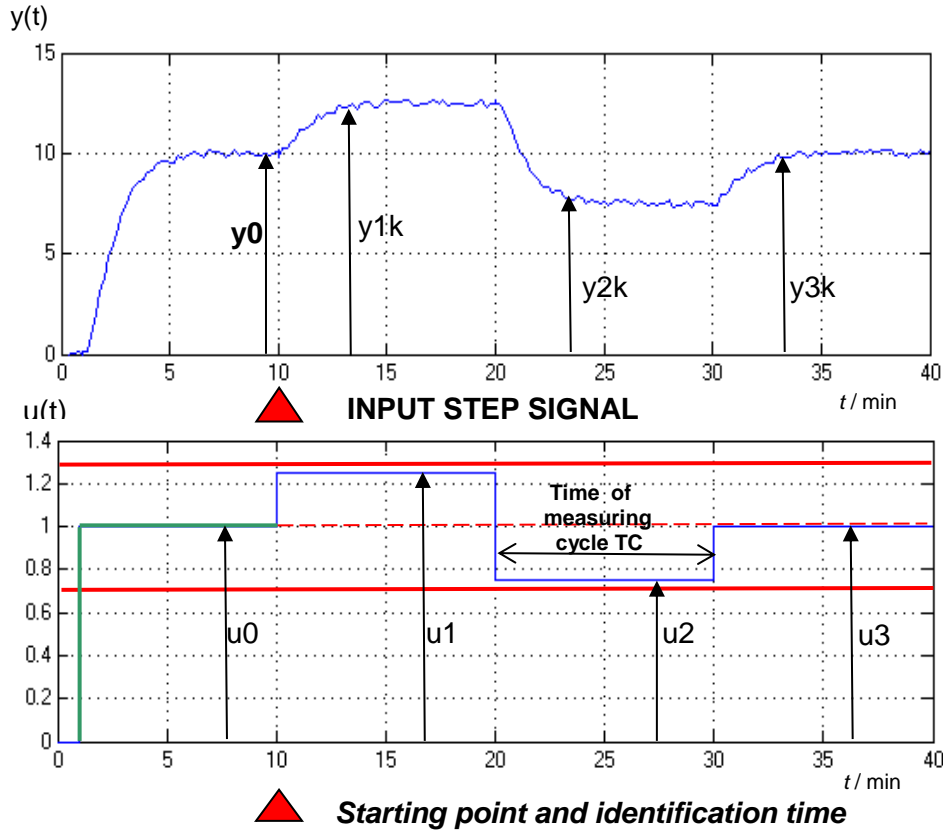


Fig. 2.1-6: Identification measurement

After the transformation the transformed data have the working point $y_0 = u_0 = 0$. The transfer function can be determined as

$$F_U(s) = \frac{\nabla Y_{ID}(s)}{\nabla U_{ID}(s)} = \frac{B(s)}{A(s)}; \quad F_D(s) = \frac{\nabla Y_{ID}(s)}{\nabla D_{ID}(s)} = \frac{C(s)}{A(s)}; \quad (2.1-5)$$

You must realized that first the transfer function $F_U(s) = Y_U(s)/U(s) = \frac{B(s)}{A(s)}$ must have been

calculated before you start to calculate the transfer function $F_D(s) = Y_D(s)/D(s) = \frac{C(s)}{A(s)}$.

That has the consequence of knowing the polynomial $A(s)$ and you calculate with your identification program only polynomial $C(s)$!

2.1.3 Model structure

Suppose that the dynamic of the controlled plant is approximated by a linear time invariant mathematical model in the form of a transfer function. The transfer function with dead time is defined

$$F_U(s) = \frac{Y_U(s)}{U(s)} = \frac{\nabla Y}{\nabla U} = \frac{b_m s^m + \dots b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots a_1 s + a_0} \cdot e^{-sT_d} = \quad (2.1-6)$$

$$= K \frac{(1 + T_{B1}s) \dots (1 + T_{Bm}s)}{(1 + T_{1s})(1 + T_{2s}) \dots (1 + T_{ns})} \cdot e^{-sT_d} = \frac{B(s)}{A(s)} \cdot e^{-sT_d}$$

where $B(s)$ and $A(s)$ describe the numerator and denominator polynomials, $n > m$, the system is called an n -th order system

$T_i = -1/s_i, T_{Bj} = -1/s_{Bj} \dots$ time constants $T_d \dots$ time delay

Some examples of available transfer function structures are described below:

- 1) $F_0(s) = \frac{K e^{-T_d s}}{\tau s + 1}$, 1th order system with time delay
- 2) $F_1(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$, 2th order system overdamped
- 3) $F_2(s) = \frac{K}{(\tau s + 1)^n}$, n-th order system critically damped
- 4) $F_3(s) = \frac{K}{(\tau_1 s + 1)[(\tau s)^2 + 2\xi\tau s + 1]}$ underdamped oscillating

2.1.4 Structure of an Off-line parameter estimation with LTI model

By Off-line parameter estimation the input and output from the measurement are saved in files. The structure is shown in the figure 2.1-7.

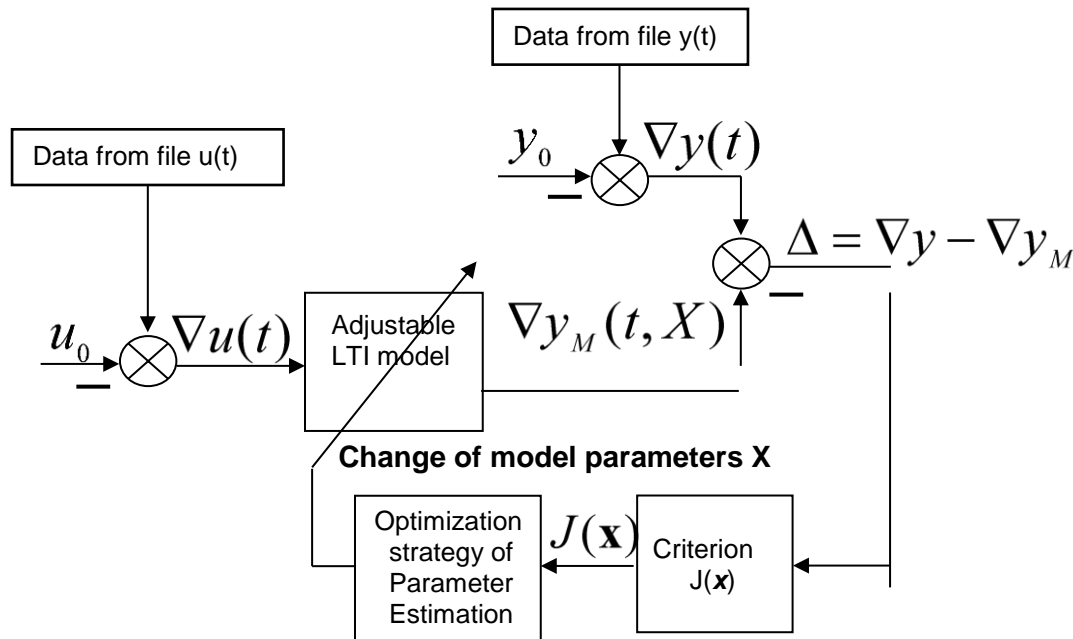


Fig.2.1-7: Structure of an Off-line parameter estimation with LTI model

The adjustable LTI model has the input $\nabla u = u(t) - u_0$ and the output $\nabla y_M(t, X)$.

The output error is determined $\Delta = \nabla y - \nabla y_M = \nabla y(t) - \nabla y_M(t, X) = \Delta(t, X)$.

2.1.5 Optimization function in MATLAB

For this optimization we are using the function “fminsearch” from the “Optimization Toolbox” which is using the not gradient Melder-Mead method. The function is completely described in the help but for our quick use we give some basic remarks.

Function **fminsearch**

Find local minimum of unconstrained multivariable function using derivative-free method

Syntax

```
x = fminsearch ('fun',x0)  
x = fminsearch ('fun',x0,OPTIONS)
```

'fun' the function to be minimized. It is a function that accepts a vector x and return a scalar f . It can be an M-file function or an anonymous function

x0 a starting vector of initial parameter estimate

x is the result vector

OPTIONS is a structure that provides control of the optimization with specific conditions
Use “**optimset**” to set the options with the following strings:

'MaxFunEvals'	Maximum number of function evaluations allowed
'MaxIter'	Maximum number of iterations allowed
'TolFun'	Termination tolerance on the function value
'TolX'	Termination tolerance on x

Example of “**optimset**”:

```
OPTIONS=optimset('MaxIter',1000, 'TolX', 1e-2)
```

2.2 Tools for parametric identification

The typical identification process can be itemized as follows:

1. Design an experiment and collect input/output data from the process to be identified.
2. Examine the data
3. Select and define a model structure (a set of candidate system descriptions)
4. Compute the best model in in the model structure according to the input/output data and given criterion for goodness of fit
5. Examine the model properties of the obtained model
6. If the model is good enough, then stop; otherwise go back to step 3 to try another model structure

2.2.1 Program and functions in MATLAB

The flow process diagram of MATLAB identification program is shown in the figure 2.2-1. This program considers the identification measurement described in the chapter 2.1.2. The file has a field variable **W** that has two columns. The first column includes the measured input (the manipulated variable **u**) and the second column includes the measured output (controlled variable **y**) . The time vector **tG** must be calculated. Then we can plot a graph of the measurement. From this graph the coordinates of the working (**u₀**,**y₀**) point can be check. The vector variables **uG**, **yG** embrace the measured input and output **u**, **y** transformed to the working point by the equations (2.1- 4). The graph of the transformed measurement is plotted and shown as figure 2.

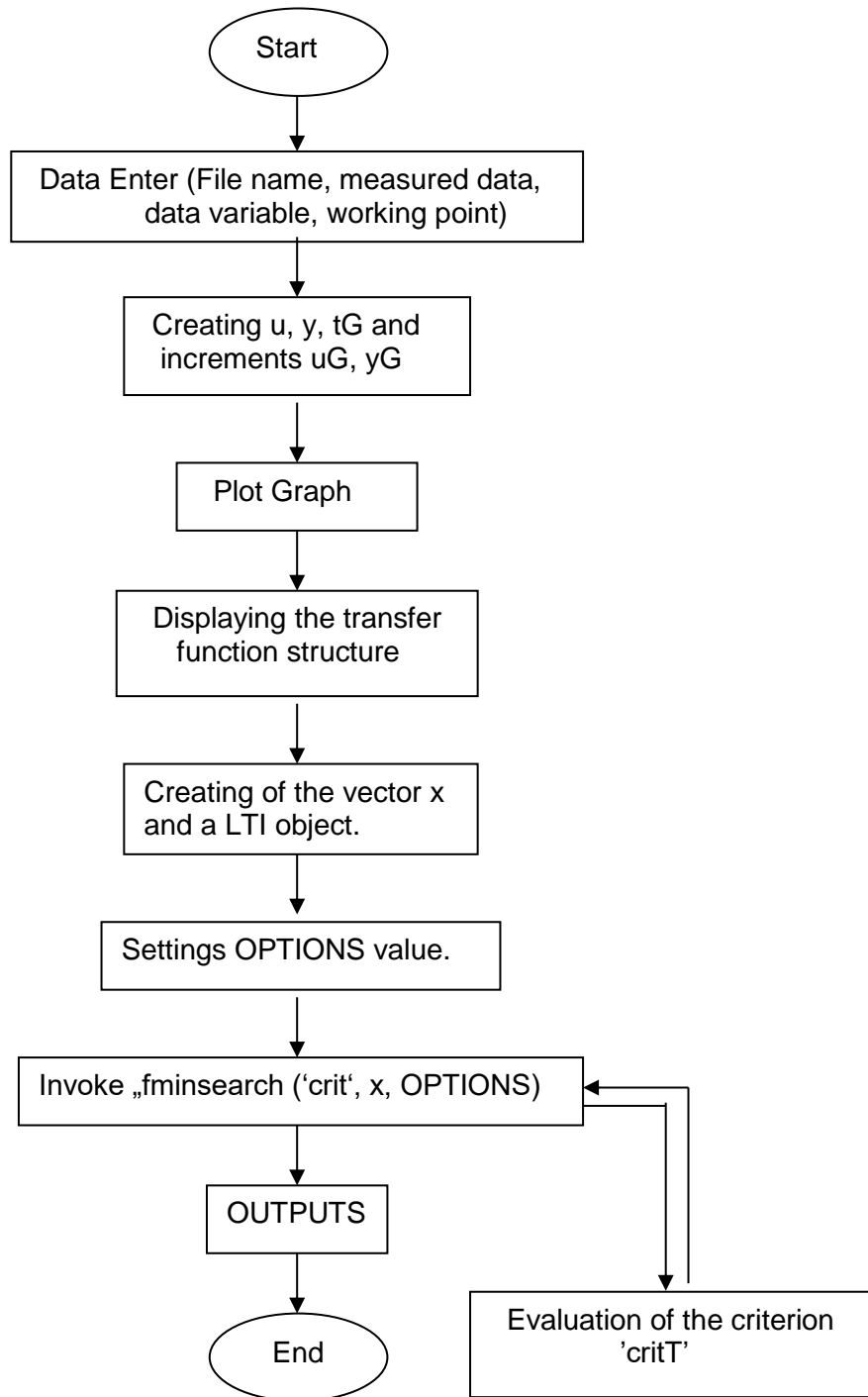


Fig. 2.2-1: Flow diagram of the MATLAB identification program

As an example we show an program in MATLAB (ident.m) that calculates parameters of the mathematical model with the structure:

$$F(s) = \frac{K}{(Ts+1)^3} = \frac{x(1)}{[x(2)s+1]^3}$$

```
%ident.m
%F=K/(Ts+1)^n, n=3,  measure with sample time=0,1sec
global tG uG yG N A
load ('SU02') %Data Enter,Datei name, measured data,
%data variable
u=W(:,1);y=W(:,2); %Creating of the vector u, y
tG=0:0.1:(length(u)-1)*0.1; %Creating of the time vector tG
figure (1)
plot(tG,y,tG,u) %Plot graph - measured data
title ('Graph of mesurement')
%working point
y0=2.5;u0=2; %working point
yG=y-y0; %increment yG
uG=u-u0; %increment uG
figure (2)
plot(tG,yG,tG,uG) %Plot graph - increments yG, uG
title ('Graph of increments yG,uG')
pause
%Displaying the transfer function structure
disp('Approximation by the transfer function')
disp('          x(1)          K ')
disp('F(s)=----- = -----')
disp('          (x(2).s + 1)^3          (Ts + 1)^3')
%Creating of the vector x and a LTI object
N=2 %N=n-1
T=0.5 %start at the Time constant=0,5
K=1 %start at the gain=1
A1=[T 1]; %polynomial A1=(Ts+1)
A=A1;
for i=1:N; %evaluation of (Ts+1)^3
    A=conv(A,A1);
end
s=tf(K,A) %LTI object s
roots(A)
x=[K T] %start point x0:x(1)=K...gain, x(2)=T... time constant
critT(x) %evaluation of the function to be minimized
disp('running...')
%setting the values of the field in the "options" structure
OPTIONS=optimset('MaxIter',50,'TolX',1e-1);
%start of the finding a minimum by the "fminsearch"
x=fminsearch('critT',x,OPTIONS);
%outputs
disp('Optimalized vector x')
x
disp('Function/criterion value J')
critT(x)
roots(A)
tf(x(1),A)
sys=tf(x(1),A);
[yi,ti]=lsim(sys,uG,tG);
figure(3) %Plot graph outpu of the LTI model and measured increments
plot(tG,yG,tG,uG,ti,yi)
```

The criterion as an M-file “criT”:

```
%criT.m
function f=critT(x)      %the function to be minimized
global tG yG uG N A
A1=[x(2) 1];
A=A1;
for i=1:N;               %evaluation of the polynomial [x(2)+1]^3=(Ts+1)^3
    A=conv(A,A1);
end
sys=tf(x(1),A);         %creating of a LTI object
[yi,ti]=lsim(sys,uG,tG); %evaluation of yi=yM(i)
%plot(ti,yi,ti,yG)
f=sum((yG-yi).*(yG-yi)); %function evaluation
```

2.3 Feedback Control with PI controller

2.3.1 Closed – loop feedback structure

Using the general output equation with a disturbance than the block diagram of the feedback control can be shown in the figure 2.3-1.

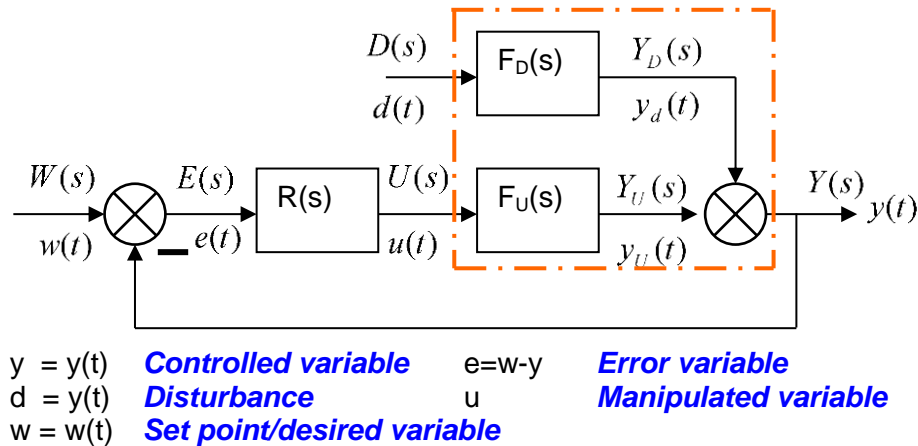


Figure 2.3-1: Block diagram of a general closed-loop feedback model control system

Simplified closed loop disturbance transfer function

$$F_{YD}(s) = \frac{Y_d(s)}{D(s)} = \frac{F_D(s)}{1 + R(s)F_U(s)} \quad (2.3-1)$$

Simplified set point transfer function

$$F_{YW}(s) = \frac{Y_w(s)}{W(s)} = \frac{R(s)F_U(s)}{1 + R(s)F_U(s)} \quad (2.3-2)$$

The general closed-loop transfer function model **can be applied to** any specific system by substituting the transfer function models for the loop elements.

2.3.2 P controller/proportional mode: proportional only controller

The output from a P controller is proportional to the error and is equal

$$u_p(t) = P \cdot e(t), \quad (2.3-3)$$

where P ... proportional gain, $e(t) = w(t) - y(t)$... error.

The Transfer function of P controller is

$$R_p(s) = \frac{U_p(s)}{E(s)} = P. \quad (2.3-4)$$

The proportional mode is simple, provides an adjustment of the manipulated variable, does not provide zero offset although it reduces the error, speeds the dynamic response, and can cause instability if tuned improperly.

2.3.3 PI controller: proportional-integral controller

A PI controller has two terms (and two tuning parameters), one proportional to the error and the other proportional to the integral of the error. The PI controller output is equal

$$u_{PI}(t) = P \cdot e(t) + I \cdot \int_0^t e(\tau) d\tau = K_R \left[e(t) + \frac{1}{T_I} \cdot \int_0^t e(\tau) d\tau \right], \quad (2.3-5)$$

where P ... proportional gain, I ... **integral gain**, K_R ... all controller gain,
 T_I ... **integral time**, $e(t) = w(t) - y(t)$... error.

The PI controller transfer function is equal

$$R_{PI}(s) = \frac{U_{PI}(s)}{E(s)} = P + \frac{I}{s} = \frac{Ps + I}{s} = K_R \frac{T_I s + 1}{T_I s} = K_R \left(1 + \frac{1}{T_I} \right). \quad (2.3-10)$$

Integral mode:

$$u_I(t) = \frac{K_R}{T_I} \cdot \int_0^t e(\tau) d\tau + e(0), \quad (2.3-11)$$

where $e(0)$... initial condition.

2.4 Control parameter optimization

The proportional-integral-derivative PID controller has three adjustable tuning constants that must be set. After parameter setting we must check the maximum value of the manipulated variable with the possibility of the real system!

The tuning constants must be derived using the same algorithm (controller structure) that is applied in the control system.

2.4.1 Hand tuning

We consider that

- 1) There is a real closed loop system with a PID controller or a model closed-loop system.
- 2) The model must be able to model the maximal movement constraints in the form of saturation limit (See figure 2.4-).

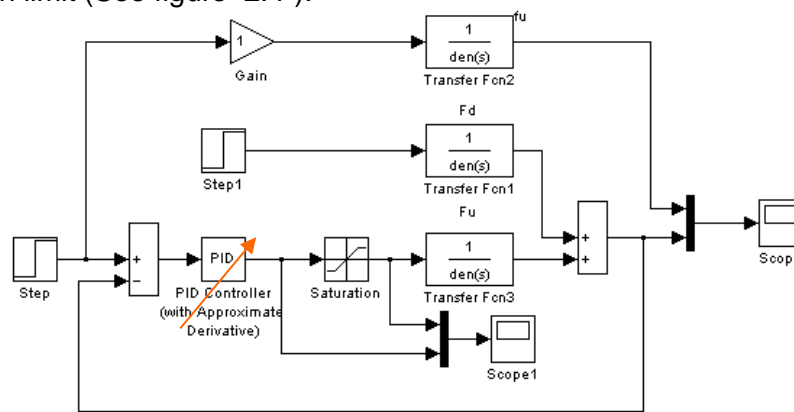


Fig. 2.4-1: Closed-loop feedback model with saturation of the actuator $\text{Gain} = (a_0 / K) \cdot w$.

Tuning procedure

- 1) Set gains P, I, D to zero
- 2) Increase gradually the proportional gain P until the transient responses of closed-loop system is fast enough (speeds of the dynamic response). See figure 2.4-2a.

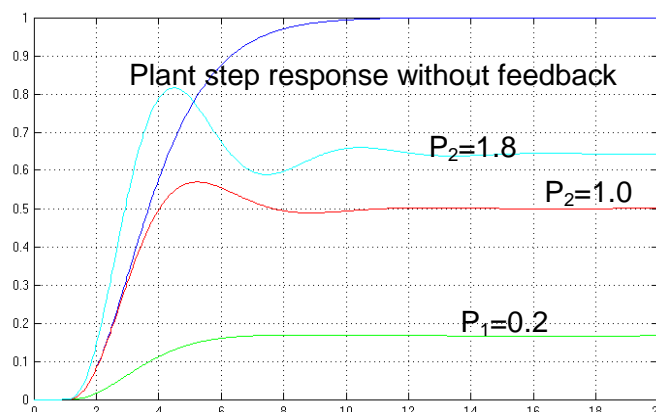


Fig. 2.4-2a: Responses by increasing gain P

- 3) Chose the appropriate proportional gain and begin to increase gradually the integral gain. The focus must be the steady error and its behavior. See figure 2.4-2b.

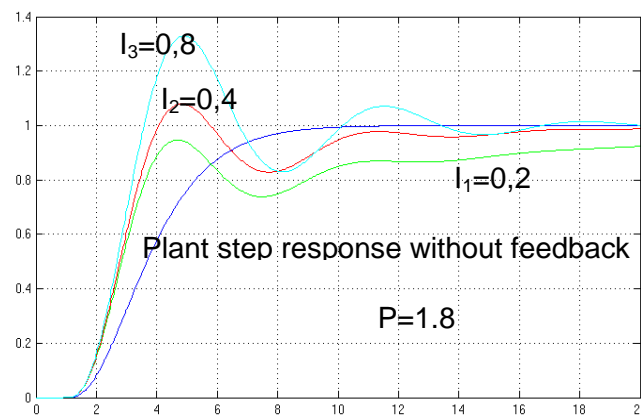


Fig. 2.4-2b: Responses by $P=1,8$ and increasing gain I

- 4) At the end you can increase the derivative gain to damp oscillations of the closed-loop system. See figure 2.4-2c.

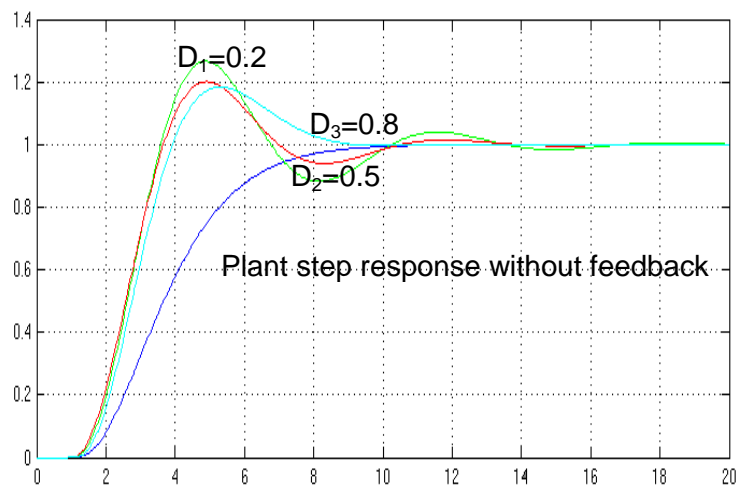


Fig. 2.4-2c: Responses by $P=1,8$, $I=0,4$ and increasing gain D

The controller output must be check. The controller output for $P=1,8$; $I=0,4$ and $D=0,5$ is shown in the figure 2.4-2d.

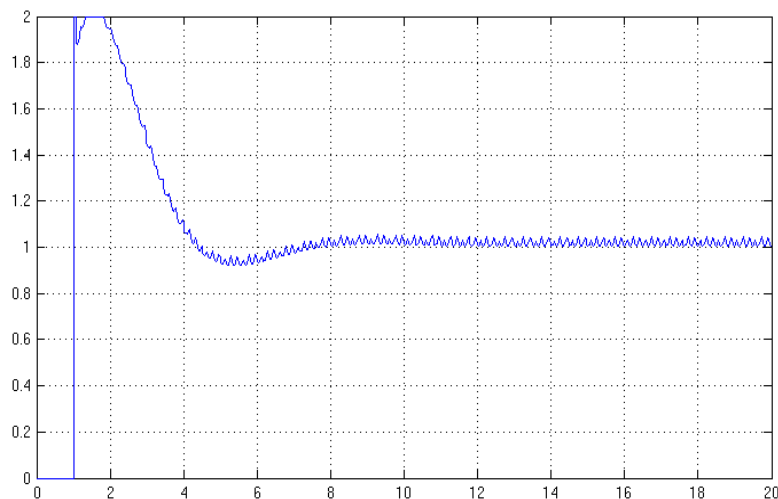


Fig. 2.4-2d: Controller output

2.5 Controller tuning based on numerical optimization

2.5.1 PID Control as a minimization

Suppose that the structure of control system is known and its model is depicted in the figure 2.3-1. The dynamic of the manipulated variable $u(t)$ is approximated by the transfer function $F_U(s)$ and the dynamic by the not measured disturbance $d(t)$ is approximated by the transfer function $F_d(s)$.

Given are scalar cost functions $J_z(P,I,D)$ which have three control parameters P , I , D . The task is to find the minimum of a chosen cost function $J_z(P,I,D)$ specified by the transfer functions $F_d(s)$, $F_U(s)$, set points $w(t)$, supposed disturbance $d(t)$ and chosen simulation time T_{SIM} .

$$\min_{P,I,D} J(P,I,D)$$

The structure of the parameter tuning by numerical minimization is shown in the figure 2.5-1. Consider that the parameter tuning can satisfied by a minimization of follow cost functions

$$J_z(P,I,D) = \int_0^{T_{sim}} \{e(t)^2 + \kappa[u(t) - u(\infty)]^2\} dt \longrightarrow \text{Min}, \quad (2.5-1)$$

2.5.2 SIMULINK programs

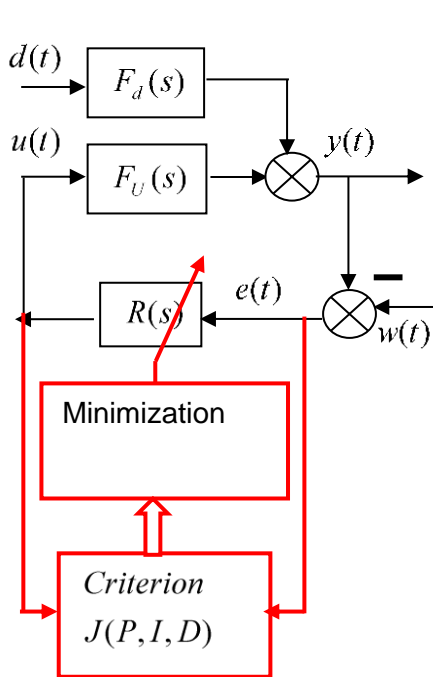


Fig. 2.5-1: Tuning P , I , D parameter by numerical minimization

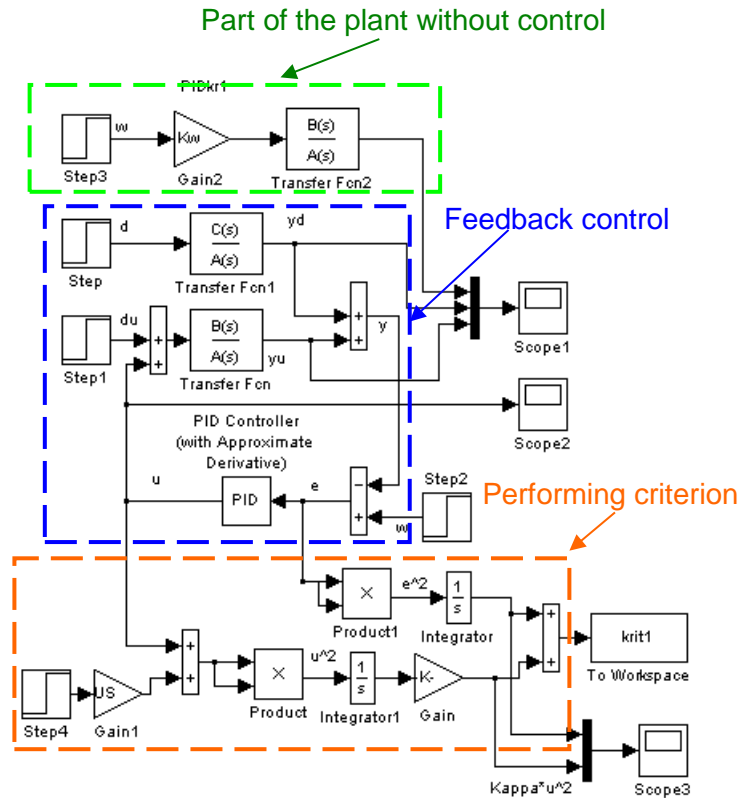


Fig. 2.5-2: "kriPID_EU", Model of closed loop system with the criterion

$$J_z(P,I,D) = \int_0^{T_{sim}} \{\bar{e}(t)^2 + \kappa[u(t) - u(\infty)]^2\} dt \longrightarrow \min$$

The kernel of the optimization is the SIMULINK program "PIDopt1" shown in figure 2.5-4. The cost function - the criterion $J_z(P,I,D)$ (2.5-1) - is calculated in the SIMULINK program "kriPID_EU" (see figure 2.5-2). The program 'kriPID_EU' can be divided in three parts: feedback control, performing criterion and a part of the plant without control.

All block parameters of the SIMULINK program "kriPID_EU" are set in the MATLAB program "PIoptEU". Inputs for set-point and disturbance are the blocks "Step1" and "Step2". The set-point $w(t)$ is generated in the block "Step1" by the program variable SetPoint. The disturbances $d_u(t)$ are generated in the block "Step2" by the program variables Disturbance_du. The Load is generated by the block "Step3". The start block for the steady state $u(\infty)$ is generated in block "Step4" by setting to 1.

The Variable κ is realized by the block "Gain". The variable K_w ($K_w = 1/K$), where K is the gain of the plant in respect to manipulated variable u , is performed by the block "Gain K_w ". The SIMULINK program is called from the MATLAB program by the function

```
" x = fminsearch('funPID_EU',x,OPTIONS);".
```

The simulation program "PIDfinalDR" is called after the optimization (see figure 2.5-3). This SIMULINK program enables to simulate responses of the control system for chosen disturbances and set-points with optimized control parameters. The set-point $w(t)$ is generated in the blocks "Step1", the disturbances $d_u(t)$ are generated by blocks "Step 2" and disturbance $d(t)$ by the "Step 3"

The results after optimization can be compared with responses without optimization. All block parameters of the SIMULINK program "PIDfinalDR" are set in the MATLAB program "PIoptEU".

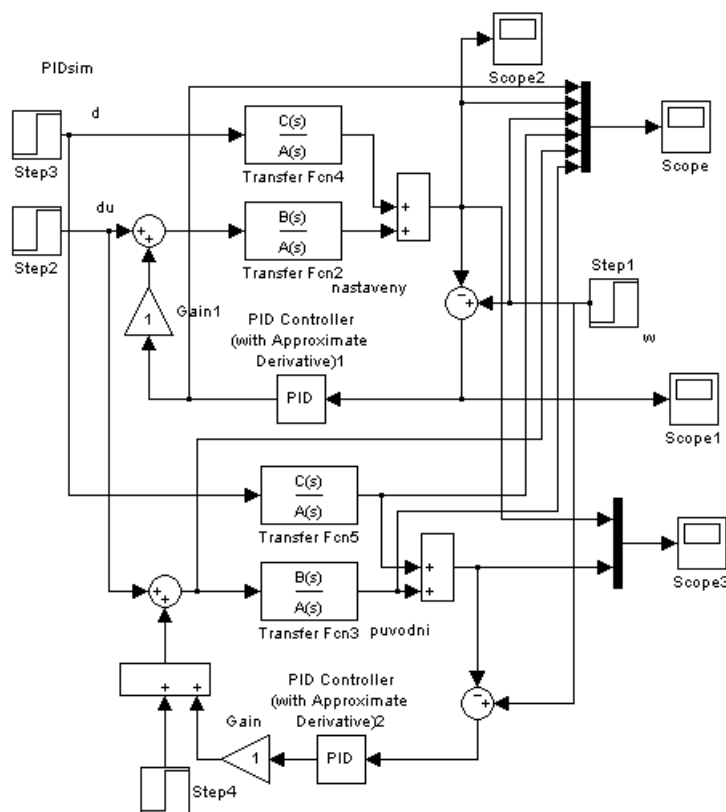


Fig. 2.5-3: Program PIDfinalDR

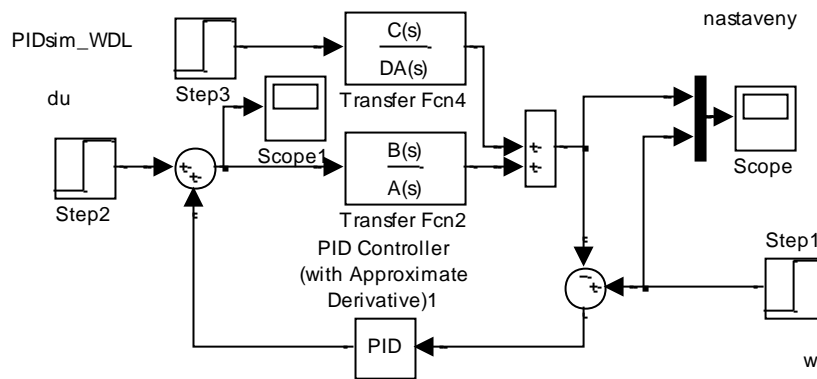


Fig. 2.5-4: SIMULINK-program "PIDopt1" with the function $f=fPIDkr1$

2.5.3 MATLAB optimization program

Program variable

P, I, D	Controller parameters	Disturbance_du	disturbance $d_v(t)$
Tsim	Simulation time	Load	Load $L(t)$
A, B, C	Polynomials $A(s), B(s), C(s)$	SetPoint	set-point $w(t)$
Ts	Sample period		
dT	Simulation step	US	Gain – $A(n)/B(n) * ZadanaW$
$P0, I0, D0$	Initial control parameters	$x=[P \ I \ D]$	vector of parameters
Kappa	Weight coefficient	Kw	Gain coefficient = $A(n)/B(n)$

MATLAB program "PIoptEU":

```
%PIoptEU Integral{e^2 + kappa * [u-u(nek)^2]}
clear all;
close all;
global P I D Tmax
%A = [ 1 49 987 3856 35693 ];
%B = [ 0 0 0 0 34365 ];
A=[0.2952 1.48 2.211 1] %Polynomial A(s)
B=[1.716] %Polynomial B(s)
C=[-0.936]; %Polynomial C(s)
DA=A;
Ts = 0.01; %Sample period
Tmax=20 %Simulation time by optimization
dT=0.01; %Simulation step
%P0=5; I0=1; D0=0.001; N=20; %Setting of initialPID parameters;
%Integral [e*e + kappa * (de/dt * de/dt)]
P0=0.5;
I0=0.1;
D0=0.001; N=20;
Kappa=0.45 %Weight coefficient
%Kappa=0.3
SetPoint=1; %Optimal control to Setpoint=1
Disturbance_du=0; %Optimal control to Disturbance_du=0
Last=0; %Optimal control to Last=0
P=P0;
I=I0;
D=D0;
x=[P I D]
disp('OPTIMIZATION OF PID-PARAMETERS:')
sim('KriPID_EU', Tmax);
```

```

disp('Criterion for initial setting of PID parameters:');krit
display('Will you continue? Hit enter')
pause
display('running...')
OPTIONS=optimset('TolFun',1e-1,'MaxFunEvals',500);
x = fminsearch('FunPID_EU',x,OPTIONS)
sim('KriPID_EU',Tmax);disp('Optimalized parameters of PID-
Controller:');krit
sim('PIDfinal_DR',Tmax);
display('Controller transfer function: R(s)=P+I/s')
display('Parameters of PID controllers')
P=x(1)
I=x(2)
D=x(3)
display('Integral Time in Seconds')
TI=1/I
%Checking of the control loop to Last and Disturbance_du
Tsim=60;
SetPoint=1; %Optimal control to Setpoint=1
Disturbance_du=1; %Optimal control to Disturbance_du=1
Last=1; %Optimal control to Last=1
sim('PIDsimWDL',Tsim);

```

MATLAB function "funPID_EU":

```

%function f=funPID_EU(x)
function f=funPID_EU(x)
global P I D Tmax
P=x(1);
I=x(2);
D=x(3);
sim('KriPID_EU',Tmax);
f=krit;

```

3 Set-up of the experiment

A picture of the equipment is shown in the figure 3-1, the structure of the experimental equipment is shown in the block diagram, figure 3-2.

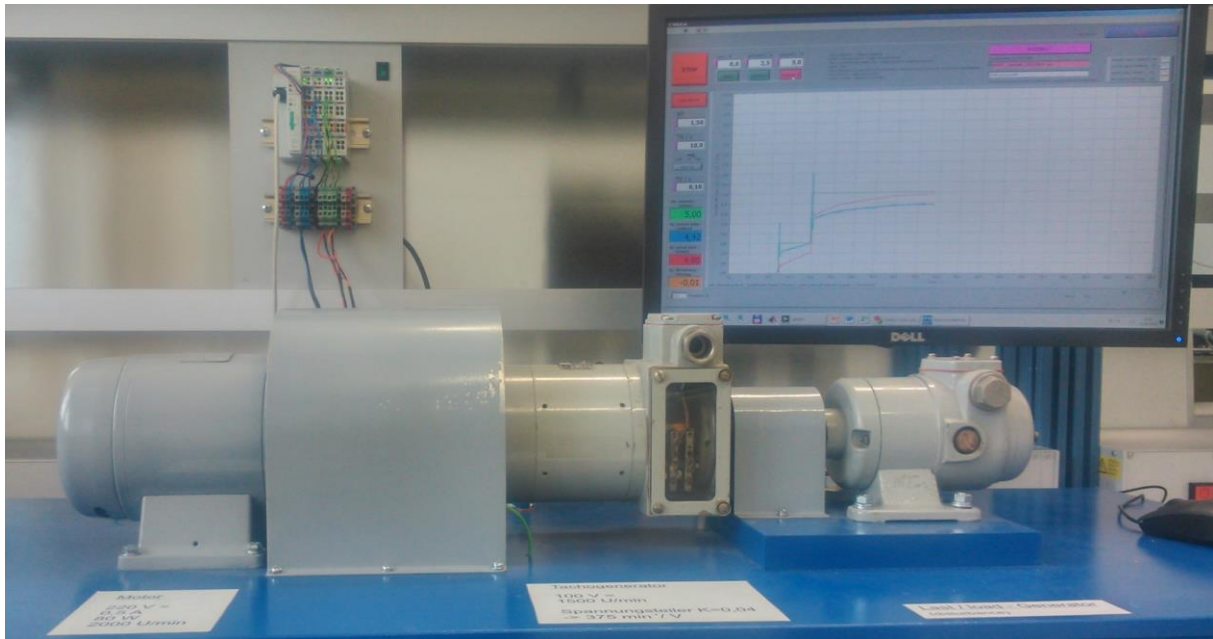


Fig. 3-1: Picture of the equipment

The experimental set-up consists of a **PLC WAGO** and the **controlled system**.

The Wago-PLC is all-purpose. It uses analog in/- and output-modules for the measuring of the process signals and the output of the actuating signals.

The PLC is connected via Ethernet (Modbus) with a PC. Over the Modbus using an PC the PLC is programmed and structured with the software Codesys. With the software LabVIEW the controller parameters are programmed into the PLC and the measured voltages are read out, displayed and saved.

The cycle time of the PLC-Controller is set to 10 ms, the cycle time of the Labview-program is set to **100 ms**.

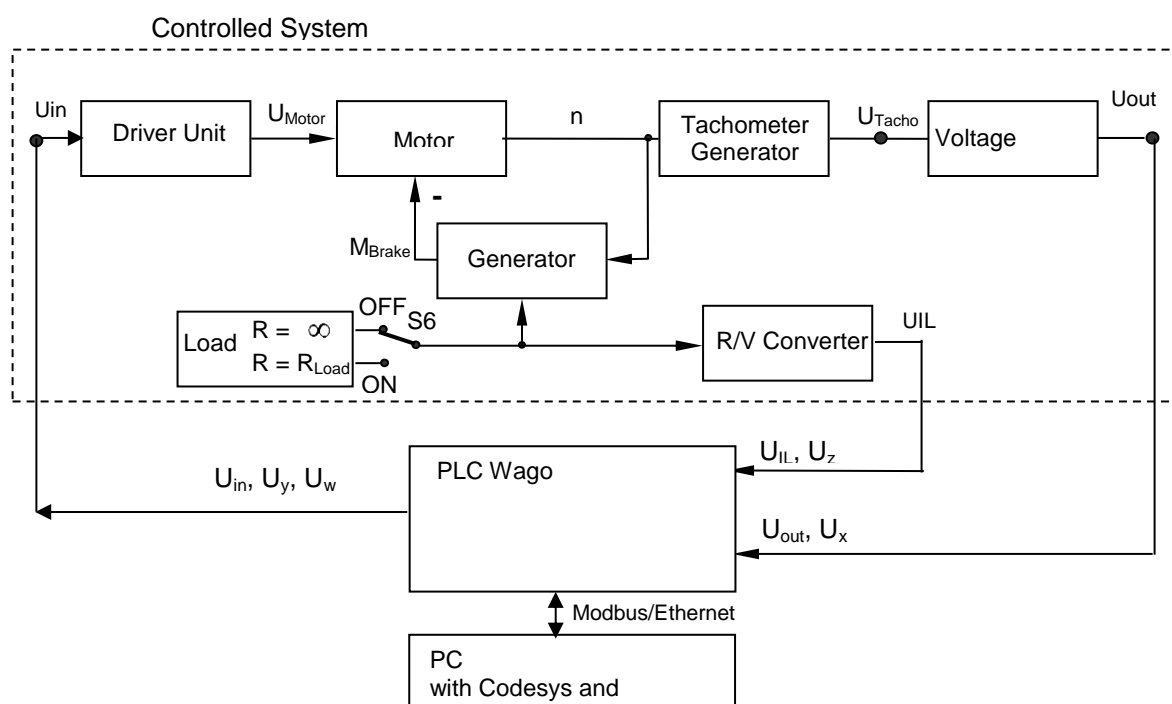


Fig. 3-2: Block diagram of the equipment

- Controlled system

The controlled system is shown in the block diagram in figure 3-2. The input and output signals are the voltages U_{in} and U_{out} , respectively. The input signal U_{in} is amplified and adapted on the input of a motor by means of the driver unit. The output-voltage U_{Motor} drives a motor. The output of the motor is the rotational speed n . The shaft of the motor is coupled rigid with the shafts of a tachometer generator and a generator. By means of switch S6 the load resistor R of the generator is switched over to $R = R_{LOAD}$ or $R = \infty$. The resistor R_{LOAD} is composed of the resistor $R_{ILOAD} = 50 \text{ Ohm}$ and the resistor of a light bulb (25 W) in series connection. The output of the R/V converter U_{IL} is the voltage over the resistor R_{ILOAD} . This means that the voltage U_{IL} is proportional the current I through the resistor R_{LOAD} .

The tachometer generator generates the voltage U_{Tacho} depending on the rotational speed n . The voltage divider transforms the voltage U_{Tacho} into the voltage U_{out} , which is adapted to the input of the controller.

The voltage U_{in} is supplied by an analog-output-modul of the Wago-PLC. The voltages U_{out} and U_{IL} are measured by an analog-input-modul of the Wago-PLC. The voltages are in the range of -10 V to +10 V.

4 Operating the equipment used in the experiment


A PC is used during this experiment:

- to enter the settings of the controller
- for data acquisition and saving
- for data computing (analyse, modelling, verification, evaluation)

Loading and preparing the SPS program:

- Start the "Wago Automation Cockpit" from the desktop of the PC.
- Load the Codesys program "RSCrC.pro" and start it (Online/Login, Online/Start).
- You can check the function blocks with the tab "POUs" and the in/output variables with the tab "Resources/Global Variables/Data_points."

Loading and preparing the LabVIEW program:

- Start the LabVIEW program "RSCrC.vi" from the desktop of the PC.
- Click on the button "Run" .
- With the tabs you can navigate through the available functions.

Data computing

This deals with MATLAB and is described in [1], [2].

Printing of windows and exporting of graphs in LabVIEW (figure 4-1)

In LabVIEW you can print out the window with „Strg+P“ (if the program is not running) or export the graph with rightclick on the graph -> “Exportieren” (export) -> “Vereinfachtes Bild exportieren” (export simple picture) -> “Bitmap.bmp” -> “In Datei speichern” (save in file) or “In Zwischenablage exportieren” (export to clipboard) -> “Speichern” (save).

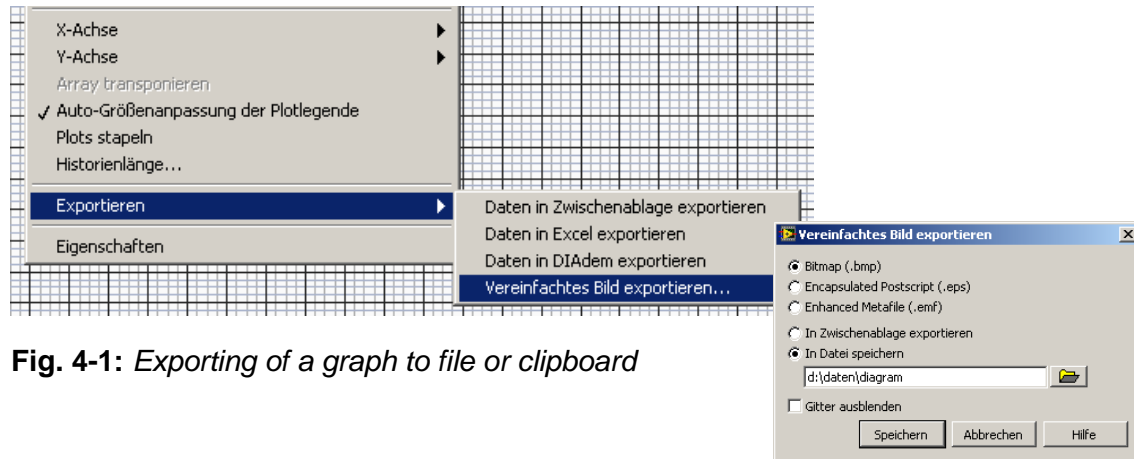


Fig. 4-1: Exporting of a graph to file or clipboard

5 Conducting the experiment

5.1 Tasks of the experiment

- 1) Make the acquaintance of the experiment and instruments, measuring and data acquisition and limitations.
- 2) Measure and plot the static characteristic.
- 3) Create the mathematical models $F_U(s)$, $F_L(s)$ by identification for the manipulated variable and the disturbance load. For the evaluation of the parameters you can use:
 - a) Modification of MATLAB programs from lecture
 - b) The MATLAB System Identification Tools
- 4) Design the optimal PI and PID controller parameters:
 - a) Manually
 - b) Numerical optimization of the cost function

$$J_z(P, I, D) = \int_0^{T_{sim}} \{e(t)^2 + \kappa[u(t) - u(\infty)]^2\} dt \longrightarrow \text{Min}$$

- 5) Simulate the feedback control and compare the responses without and with PI controller.
- 6) Check the controller by real control

5.2 Analysing of the controlled system

5.2.1 Measuring of the static behaviour $U_{out} = f(U_{in})$

- Switch on the operating voltage of the motor and adjust it to 220 V.
- Switch to the tab “**Static Behaviour**” (figure 5.2-1) in the LabVIEW-program “RSCrC.vi”.

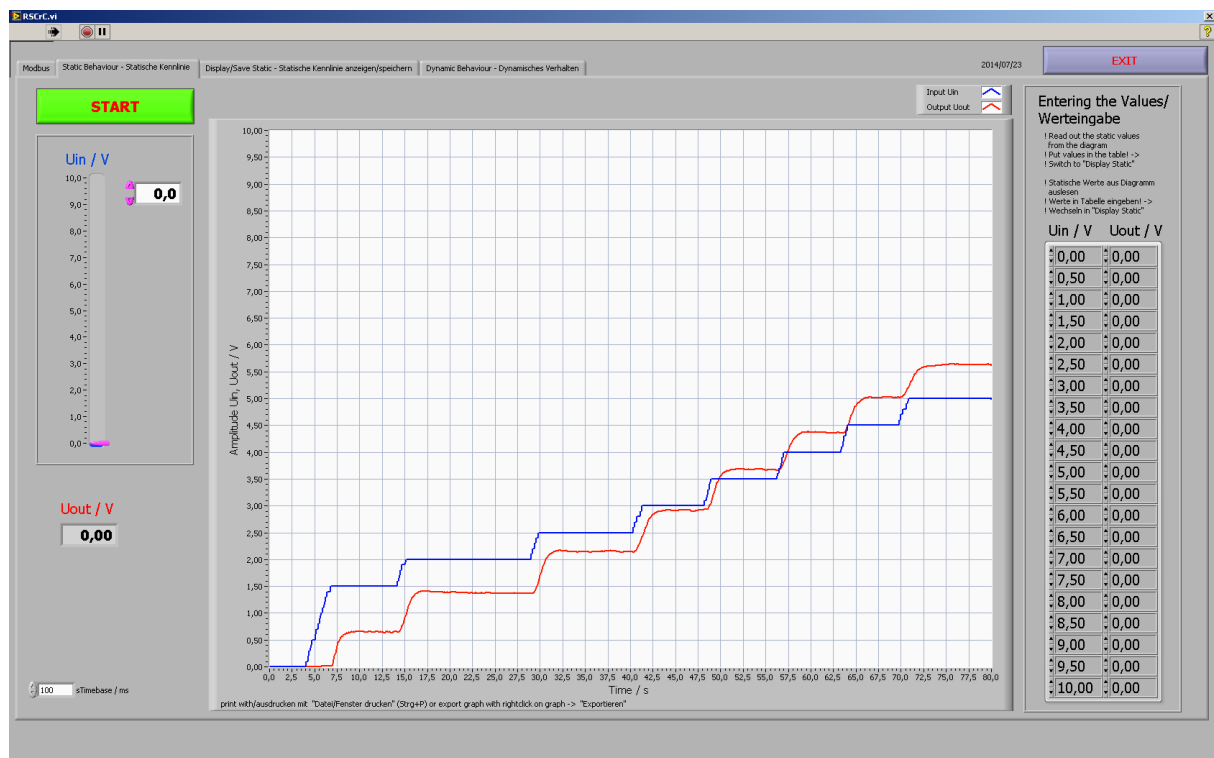


Fig. 5.2-1: Frontpanel of the „Static Behaviour“ tab

- Start the measurement with the button “START”.
- Measure the static behaviour of the controlled system between $U_{in} = 0 \text{ V}$ and 10 V in steps of $0,5 \text{ V}$:
 - a) without load (switch S6: OFF)
 - b) with load (switch S6: ON).
- You can change the input voltage U_{in} into the plant with the slider “Uin in V” or put in the value manually.
- The output voltage U_{out} of the plant is displayed in the digital display and the graph.
- Both voltages are recorded and displayed in the graph as a function $U = f(t)$.
- The measurement can be stopped with the button “STOP”.
- Print out the window with „Strg+P“, export the graph (figure 4-1) or read out and enter in the static values in the righthand table and then continue with the tab „Display Static”.
- > Switch to the tab “**Display Static**”.
- The before entered values are now displayed as a function $U_{out}=f(U_{in})$.
 - The graph display area can be changed by adjusting and entering the new final values on the co-ordinate axis.
- Print out the window with „Strg+P“ or export the graph (figure 4-1).
- You can change the entered values by switching back to the tab „Static Behaviour“.
- Save the data by entering a filename and clicking the button „Save”.
The data will be added to existing files!
- You can read in this data with MATLAB, Excel or OpenOffice (tab separated ascii values).
- The datafile has the following format:
 1. Headerline with % name, date, time
 2. Headerline with the name of the measured signals
 - ... Datalines with tab separated ascii data values
- The data values are in stored in mV without decimal separator to avoid problems with the decimal separator format!
- If you append the data to an existing file you get additional header lines!

5.2.2 Recording of the step response and the influence of external disturbances (load) on the controlled system

- Switch to the tab “**Dynamic Behaviour**” (figure 5.2-2) in the LabVIEW-program “RSCrC.vi”

According to DIN 19221 the input U_{in} into the controlled system is measured as setpoint (desired) value U_w , the output U_{out} is measured as the actual value U_x and the load U_{IL} is measured as disturbance U_z (figure 3-2 and 5.2-2).

- Switch the controller off with the button.
- Enter the setpoint values.
- Enter the filename and a comment to the experiment.

The form of the file name for this experiment is: RSCrC_u_v_w with:

u (character of analysed part) :

analysis of controlled system : **u = “csystem”**
 analysis of controlled control loop : u = “cloop”

v (character of date): year month day (for instance: “150515)

w (character of group): g + number of group (for instance: “g1”)

The data will be added to existing files!

You can put in a comment in the according field!

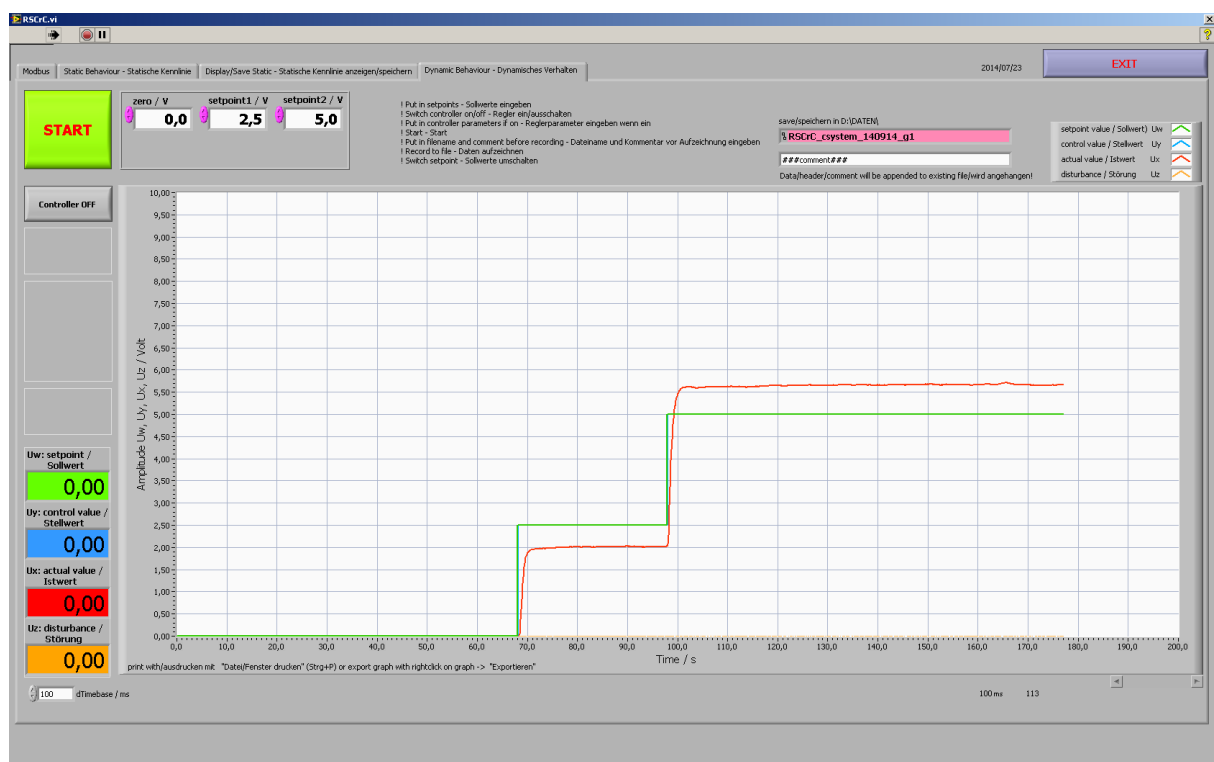


Fig. 5.2-2: Frontpanel of the „Dynamic Behaviour“ tab

- Start the measurement with the button “START”.
- You can start the recording with the now visible button “RECORD”.
- Switch the setpoints with the now visible buttons “zero”, “setpoint 1”, “setpoint 2”.
- Switch on/off the load (disturbance) with switch S6.
- Record the step response $U_x(t)$, the step signal $U_w(t)$ and the disturbance $U_z(t)$ for the following events, start the steps when the output U_x is constant!:

- a) set point step $U_w = \text{setpoint1} \Rightarrow \text{setpoint2}$; without load (switch S6: „OFF“)
 - b) load (disturbance) on and off at setpoint2
 - switch S6: „OFF“ \Rightarrow „ON“ \Rightarrow „OFF“
 - c) set point step $U_w = \text{setpoint2} \Rightarrow \text{setpoint1}$; without load
 - d) disturbance load on at setpoint1
 - switch S6: „OFF“ \Rightarrow „ON“
 - e) set point step $U_w = \text{setpoint1} \Rightarrow \text{setpoint2}$; with load
 - f) set point step $U_w = \text{setpoint2} \Rightarrow \text{setpoint1}$; with load
 - g) load off at setpoint1
 - switch S6: „ON“ \Rightarrow „OFF“.
- Switch the setpoint to zero.
 - Stop the measurement with the button “STOP”.
 - Print out the window with „Strg+P“ or export the graph (figure 4-1).
 - You can read in the recorded data with MATLAB, Excel or Open Office.
 - The datafile has the following format:
 1. Headerline with % name, date, time
 2. Headerline with the comment
 3. Headerline with the name of the measured signals
 ...Datalines with tab separated ascii data values
 - The data values are in stored in mV and ms without decimal separator to avoid problems with the decimal separator format!
 - If you append the data to an existing file you get additional header lines!

5.3 Modelling

5.3.1 Modelling of the Controlled System

- Verify the saved information.
- Use the MATLAB software package to create a linear model of the controlled system for transmission of the manipulated variable u and transmission of the disturbance d .
- Determine the transfer functions for transmission of the manipulated variable $F(s)$ and transmission of the disturbance $FD(s)$ of the controlled system.
- Give reasons for the structure of the models.
- Identify the parameters of the transfer functions using MATLAB.
- Verify the models using SIMULINK. Describe the procedure and discuss the results for the following steps:

$$\begin{array}{ll}
 u(t < 0) = 0 ; & u(t \geq 0) = 1 ; \quad d = 0 \\
 u(t < 15s) = 1 ; & u(t \geq 15s) = 0 ; d = 0 \\
 u(t < 0) = 0 ; & u(t \geq 0) = 1 ; \quad d = 1 \\
 u(t < 15s) = 1 ; & u(t \geq 15s) = 0 ; d = 1 \\
 (d=0 : R = \infty ; \quad d=1 : R = R_{\text{Load}})
 \end{array}$$
- Illustrate, in one contiguous co-ordinate plane, the response steps for both the controlled system and that of the model; these are generated by the same steps, including the input steps.

5.3.2 Theoretical analysis of the control loop

- Analyse a control loop, composed of the controlled system (result of task 5.3.1) and a PI-controller.
- Define range of stability as a function of the parameters of the PI-controller.
- Illustrate this graphically.

Studies on the model of control loop

- Use the MATLAB software package to create a model of the control loop, which is made up of the model of the controlled system (result of task 5.3.1) and a PI-controller.
- Determine the parameters of the PI-controller for the optimal behaviour of the control loop according to 5.3.3 and 5.3.4

5.3.3 Experimentation on the control loop model

- Reference analysis: $w(t < 0) = 0$, $w(t \geq 0) = 1$, $d_Y = 0$; $d_X = 0$
 $w(t < 0) = 1$, $w(t \geq 0) = 0$, $d_Y = 0$; $d_X = 0$
 $(d_X, d_Y : \text{Disturbance on } x, y; \quad d_X: R_{\text{LOAD}})$
- Disturbance analysis : $d_X(t < 0) = 0$, $d_X(t \geq 0) = 1$, $w = 0$
 $d_X(t < 0) = 1$, $d_X(t \geq 0) = 0$, $w = 0$
(one PI-controller for both analysis)
- Illustrate in one co-ordinate plane the input steps, the controlled variable x and the manipulated variable y as a result of the following steps:

$$\begin{array}{llll} w(t < 0) = 0 ; & w(t \geq 0) = 1 ; & d_Y = 0 ; & d_X = 0 \\ d_X(t < 15s) = 0 ; & d_X(t \geq 15s) = 1 ; & d_Y = 0 ; & w = 1 \\ d_Y(t < 30s) = 0 ; & d_Y(t \geq 30s) = 1 ; & d_X = 1 ; & w = 1 \end{array}$$

5.3.4 Numerical optimization of the cost function

- Use this method to determine the optimal parameters for the following input steps:
 - Reference analysis : $w(t < 0) = 0$; $w(t \geq 0) = 1$; $d_Y = 0$; $d_X = 0$
 $w(t < 0) = 1$; $w(t \geq 0) = 0$; $d_Y = 0$; $d_X = 0$
 - Disturbance analysis : $d_X(t < 0) = 0$; $d_X(t \geq 0) = 1$; $w = 0$; $d_Y = 0$
 $d_X(t < 0) = 1$; $d_X(t \geq 0) = 0$; $w = 0$; $d_Y = 0$
- The following criteria must be fulfilled:
 - the maximum overshoot of the controlled variable $< 15\%$
 - the transient time (the range of controlled variable $\pm 2,5\%$) $< 6\text{ s}$
 - the range of manipulated variable is minimal
- Illustrate in one co-ordinate plane the input steps, the controlled variable y and the manipulated variable u .

5.3.5 Studies on the real control loop

- Adjust on the controller the parameters of tasks no. 5.3.3 – 5.3.4 one after another.

According to DIN 19221 the input U_{in} into the controlled system is measured as control value U_y , the output U_{out} is measured as the actual value U_x , the setpoint is measured as stepoint (desired) value U_w and the load U_{IL} is measured as disturbance U_z (figure 3-2 and 5.3-1).

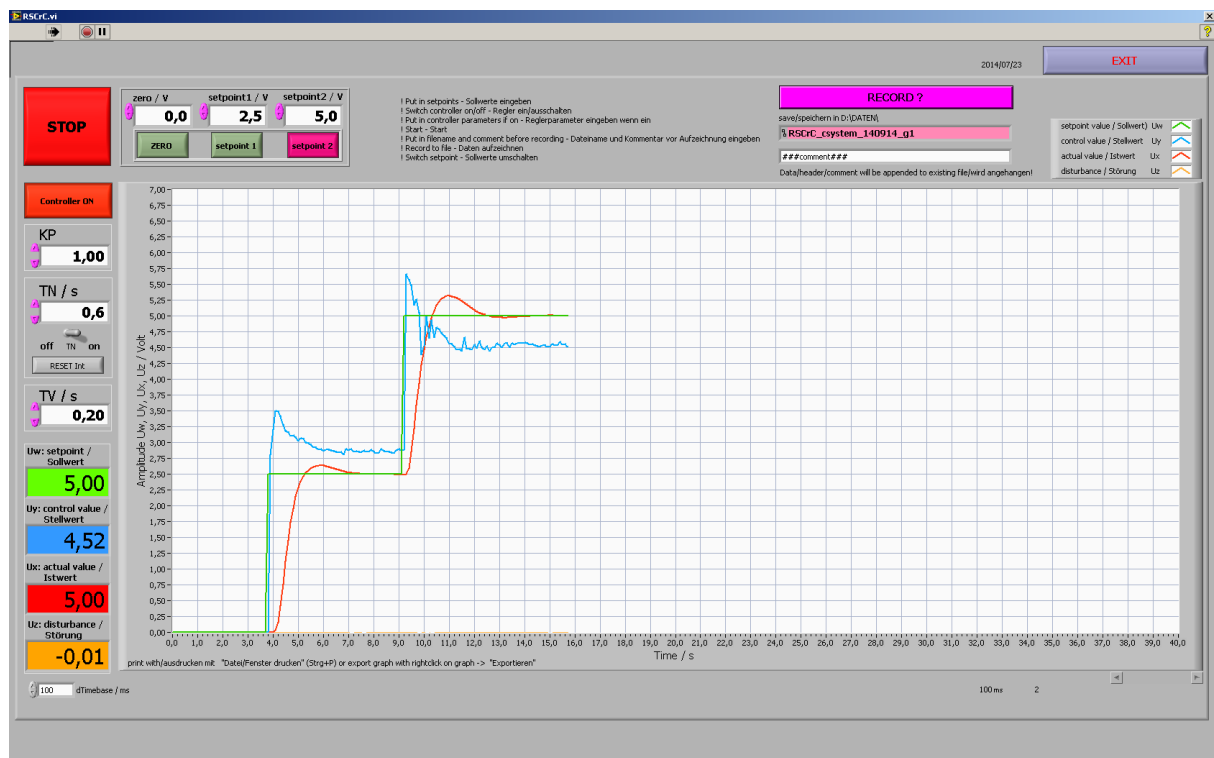


Fig. 5.3-1: Frontpanel of the „Dynamic Behaviour“ tab with controller on

- The operation of the equipment is similar to 5.2.2.
- Switch to the tab “**Dynamic Behaviour**” in the LabVIEW program “RSCrC.vi”
- Switch the controller on with the button.
- Enter the now visible values for the controller.
 - The transfer function of this controller is the following:

$$F(p) = KP * (1 + 1 / TN (p) + TV * p).$$
- You can disable the integral part of the controller with the switch "TN on/off"
- You can reset the offset of the integral part with the button "Reset Int".
- The offset of the integral part will be automatically reset with the start of measurement but not with the disabling of the integral part of the controller!
- Enter the setpoint values.
- Enter the filename and a comment to the experiment.

The form of the file name for this experiment is: RSCrC_u_v_w with:

u (character of analysed part) :

analysis of controlled system :

u = “csystem”

analysis of controlled control loop :

u = “cloop”

v (character of date): year month day (for instance: “150515)

w (character of group): g + number of group (for instance: “g1”).

The data will be added to existing files!

You can put in a comment in the according field!

- Start the measurement with the button “START”.
- You can start the recording with the now visible button “RECORD”.
- Switch the setpoints with the now visible buttons “zero”, “setpoint 1”, “setpoint 2”.
- You can switch on/off the load (disturbance) with switch S6.
- Switch setpoint to zero.
- Stop the measurement with the button “STOP”.

- Print out the window with „Strg+P“ or export the graph (figure 4-1).
- You can read in the recorded data with MATLAB, Excel or Open Office.
- The datafile has the following format:
 1. Headerline with % name, date, time
 2. Headerline with "% controller on"
 3. Headerline with % value for KP
 4. Headerline with % value for TV
 5. Headerline with % value for TN
 6. Headerline with the comment
 7. Headerline with the name of the measured signals
 - ...Datalines with tab separated ascii data values
- The data values are in stored in mV and ms without decimal separator to avoid problems with the decimal separator format!
- If you append the data to an existing file you get additional header lines!
- In this experiment the function between the rotational speed n and the voltage U is the following:

$$n = k_n \cdot U \quad \text{with} \quad k_n = 375 \text{ min}^{-1} / \text{V}.$$

- Reference analysis: n_W = reference input variable

$$\begin{aligned} n_W(t < 0) &= n_{WP} = 1406 \text{ min}^{-1}; R = \infty \\ n_W = n_{WP} &\Rightarrow n = \text{constant}, \\ n_{WP}: &\text{rotational speed of working point [min}^{-1}\text{]} \\ n: &\text{rotational speed [min}^{-1}\text{]} \\ n_W(t \geq 0) &= n_{WP} \pm \Delta n_{WS} \\ \Delta n_{WS} &= \text{step of rotational speed [min}^{-1}\text{]} \\ \Delta n_{WS} &= 469 \text{ min}^{-1} \end{aligned}$$

- Disturbance analysis:

$$\begin{aligned} n_W(t < 0) &= n_{WP} = 1406 \text{ min}^{-1}; R = \infty \\ n_W = n_{WP} &\Rightarrow n = \text{constant}, \\ 1. R(t \geq 0) &= R_{LOAD} \\ 2. R(t \geq 0) &= \infty \end{aligned}$$

- Illustrate in one co-ordinate plane the input steps and the controlled variable of both analysis.

6 Analysis of the experiment

- Discuss and evaluate the results.
- Discuss and evaluate your responses to all steps.
- Compare the responses to each step of the real control loop to those of the model of the controlled system.

7 Literature

- [1] The Student Edition of Matlab. Version 4, Users Guide. The Math Works, Inc. 1995, Prentice Hall, Englewood Cliffs. ISBN 0-13-184979-4
- [2] Simulink Dynamic System Simulation for Matlab. Using Simulink, Version 2, The Math Works, Inc. 1997