

**Modeling, Identification and Control of Periodic Disturbances
in
Drive-Load Angular Velocity Servo Control Systems
with
Self-Excited Oscillations**

Dissertation
submitted to the
Faculty of Mathematics/Computer Sciences
and Mechanical Engineering,
Clausthal University of Technology

for the degree of
Doktor-Ingenieur

by
M.Sc. Izziddien Ahmed Abdella Alsogkier
Tripoli, Libya

Dissertation submitted on 14 July 2014

Oral examination held on 6 January 2015

Committee chair: Prof. Dr.-Ing. Armin Lohrengel

First referee: Prof. Dr.-Ing. Christian Bohn (dissertation supervisor)

Second referee: Prof. Dr.-Ing. Ferdinand Svaricek

ABSTRACT:

In this work, the control system problems, the set point tracking and the periodic disturbance rejection, are considered in general and in particular for an angular velocity servo control drive-load system. The sources of the periodic disturbances are that either the drive side, the load side or both of them have angle dependent parameters, this could take place because of the working principle of the drive as in internal combustion engines, electrical (induction or permanent magnet synchronous) motors, etc., as well as in the load side e.g. eccentricity as in crankshaft or camshaft mechanisms. These types of periodic disturbances are also called self-excited periodic disturbances generated by state dependent (periodic) parameters and have been classified and modeled in this work as internal periodic disturbances. On the other hand, periodic disturbances that do not depend on any of the system states or parameters are classified and modeled as external periodic disturbances.

Therefore, for analysis and design objectives, a mathematical model has been built for a rigid and flexible drive-load system with angle dependent spring, damper and moment of inertia load elements plus an external periodic disturbance source. This has led to build an identification model representing the drive-load system that consists of input to output dynamic part plus an internal and external input periodic disturbance part. Thus, an identification algorithm is used to identify this identification model. Then, by using the identified periodic disturbance parameters of the identification model, a feed-forward controller has been designed to compensate the drive-load system periodic disturbances as an add-on to an already existing set point tracking feedback controller. This algorithm has turned out to be an indirect adaptive feed-forward periodic disturbance controller. The algorithm has been tested in simulation as well as in real-time control implementation intensively with very good results. Therefore as a next step, a test platform for a drive-load system has been built up, where the load side has a crankshaft mechanism that generates state (angle) dependent oscillations. And finally, the algorithm has been implemented in a real-time controller and successfully applied on the test platform to model, identify and consequently to compensate the periodic disturbances.

TABLE OF CONTENTS

TABLE OF CONTENTS.....	IV
NOMENCLATURE	VIII
LIST OF FIGURES	XX
1 INTRODUCTION.....	1
1.1 Problem Description.....	1
1.1.1 Set Point Tracking and Disturbance Rejection.....	1
1.1.2 Disturbances in General.....	1
1.1.3 Oscillation Sources and their Classification	2
1.2 Periodic Disturbance Rejection Methods	3
1.3 Main Distinctions of the Developed Algorithm	5
1.4 Targets of this Work.....	5
1.5 Contributions and Outline of this Work	6
2 MODELING AND SIMULATION	9
2.1 Periodically Disturbed Drive-load Systems	9
2.2 Rigid Drive-load System	10
2.2.1 Externally Excited Periodic Disturbances	11
2.2.2 Internally (Self) Excited Periodic Disturbances	12
2.2.3 The Disturbance Format of the Rigid Drive-load System.....	20
2.2.4 Simulation Examples	22
2.3 Flexible Drive-load System.....	25
2.3.1 State Space Model of the Linear Flexible Drive-load System	29
2.3.2 Externally and Internally Disturbed Flexible Drive-load System	29
2.4 The Decision of How Many Degrees of Freedom.....	33
3 PERIODIC DISTURBANCE REJECTION	36
3.1 Feedback Disturbance Rejection	36
3.1.1 Periodic Disturbance Rejection Filter.....	37
3.2 Feed-forward Disturbance Compensation.....	42
3.2.1 Introduction to the Feed-forward Disturbance Compensation	42
3.2.2 Introduction to Disturbance Observers	43

3.2.3	State and Disturbance Observer in State Space	48
3.2.4	Adaptive Periodic Disturbance Cancelation	50
3.3	Feed-forward Disturbance Compensation of the Drive-load System.....	51
3.3.1	Rigid Drive-load System	51
3.3.2	Flexible Drive-load System	52
4	MODELING, IDENTIFICATION AND CONTROL	54
4.1	Introduction	54
4.2	Externally and Internally Disturbed Identification Modell	56
4.2.1	Building up the Identification Model	57
4.2.2	Parameter Identification of the Identification Model	60
4.3	Discrete Periodic Disturbance Compensation Using RLS Method.....	62
4.3.1	Discrete First Order Identification Model	64
4.3.2	Discrete Second Order Identification Model	65
4.4	Identification and Control Strategies	66
5	SIMULATION EXAMPLES	68
5.1	Rigid Drive-load System	68
5.1.1	Feedback Control.....	68
5.1.2	Feedback Feed-forward Control	72
5.1.3	Comments on the Rigid Drive-load System	81
5.2	Flexible Drive-load System.....	85
5.2.1	Externally Excited Periodic Disturbance Compensation.....	85
5.2.2	Internally Excited Periodic Disturbance Compensation.....	97
5.2.3	Discrete Periodic Disturbance Compensation	102
5.2.4	Comments on the Flexible Drive-load System.....	105
6	REAL-TIME EXPERIMENTS	106
6.1	Torsional Test Platform.....	106
6.1.1	Test Platform Description.....	106
6.1.2	External Periodic Disturbance	106
6.1.3	Internal Periodic Disturbance	113
6.1.4	Comments on the Torsional Platform Experiments.....	119
6.2	Self-excited Machine Test Platform.....	120
6.2.1	Test Platform Description.....	120

6.2.2	Frequency Spectrum Analysis	121
6.2.3	Identification and Control of Periodic Disturbances	123
6.2.4	Comments on the Self-excited Machine Experiments.....	124
7	CONCLUSIONS AND FUTURE WORK	131
7.1	Conclusions	131
7.2	Future Work	135
7.2.1	Theoretical	135
7.2.2	Practical	135
Appendix A	Eccentric Mechanisms	136
A.1	Scotch Yoke Mechanism.....	136
A.1.1	Scotch Yoke Kinematics.....	136
A.1.2	Scotch Yoke Dynamics.....	136
A.1.3	Scotch Yoke Dynamic Model.....	139
A.1.4	Frequency Spectrum Analysis	139
A.2	Crank Mechanism.....	142
A.2.1	Crank Mechanism Kinematics.....	142
A.2.2	Frequency Spectrum Analysis	144
Appendix B	System Identification	147
B.1	Introduction	147
B.2	Linear Least Squares Parametric Identification Methods	150
B.2.1	Simple Least Squares of Prediction Error.....	151
B.2.2	Least Squares of Output Error	153
B.2.3	The Recursive Least Squares Algorithm with Forgetting Factor	155
B.2.4	Starting the RLS for Online Identification.....	155
B.2.5	Extra Applications of Linear Least Squares Algorithm.....	156
B.3	Nonlinear Least Squares Parametric Methods	156
B.3.1	Output Error Method.....	156
B.3.2	Recursive Gauss-Newton Method	162
B.3.3	Prediction Error Method	164
B.4	Practical Aspects of Parametric System Identification	165
B.4.1	Identification in Closed Loop	165

References 166

NOMENCLATURE

Abbreviations

1DOF	1 (one) Degree Of Freedom
2DOF	2 (two) Degree Of Freedom
AFFC	Adaptive Feed-Forward Control
FFC	Feed-Forward Control
ILC	Iterative Learning Control
IMP	Internal Model Principle
LPV	Linear Parameter Varying
LTI-DE	Linear Time-Invariant Differential Equation
LTI-IMP	Linear Time-Invariant Internal Model Principle
PDRF	Periodic Disturbance Rejection Filter
PID	Proportional-Integral-Derivative
PI	Proportional-Integral
ARX	AutoRegressive model with eXogenous input
SLS	Simple Least Squares
RLS	Recursive Least Squares
LMS	Least Mean Squares
ANVC	Active Noise and Vibration Control
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
PC	Personal Computer
PCI	Peripheral Component Interconnect
I/O	Input/Output
RTWT	Real-Time Windows Target
VFD	Variable-Frequency-Drive
AC	Alternating Current
RTCB	Real-Time Controller Board

Global Constants, Variables, Operators and Functions

t	Time
s	Complex variable of Laplace transform
z	Complex variable of z -transform
i	Imaginary number $\sqrt{-1}$ (in chapters 2, 3 and 5)
i, j, k	Index integer numbers (0, 1, 2, ...)
$\mathbf{0}$	Zero matrix
\mathbf{I}	Identity matrix
\sin, \cos	Sine and cosine trigonometric functions
\tan^{-1}	Inverse of the trigonometric tangent function \tan
\mathbf{A}^T	Transpose of matrix (vector) \mathbf{A}
\mathbf{A}^{-1}	Inverse of matrix \mathbf{A}
$f^{-1}(\cdot)$	Function inverse of function $f(\cdot)$
$\mathcal{E}\{x\}$	Expected value of x
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	System matrix, input (matrix) vector and output (matrix) row vector of a state space dynamic system
$\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t)$	State vector, output and input of a state space dynamic system
\otimes	Kronecker product

Chapter 2 Variables

φ	Angular position [rad]
$\vartheta = \dot{\varphi}$	Angular velocity [rad/s]
J	Moment of inertia [kgm^2]
D	Damping factor [Nms/rad]
T_{in}	Total torque acting on the load [Nm]
T	Drive (input) torque [Nm]
a, b	Drive-load system mathematical parameters
T_{dis}	Disturbance torque [Nm]
N_x	Number of external disturbance harmonics
α_{xi}, β_{xi}	Sine and cosine coefficients of external disturbance harmonics
ω	Disturbance principal angular frequency [rad/s]
$K(\varphi)$	Angle dependent spring (torque) load element [Nm]

$D(\varphi)$	Angle dependent damper load element [Nms/rad]
$J(\varphi)$	Angle dependent moment of inertia load element [kgm ²]
$dJ(\varphi)/dt$	Derivative of angle dependent moment of inertia load element [kgm ² /s]
N_K, N_D, N_J	Number of harmonics of the angle dependent spring, damper and moment of inertia load elements
k_i, d_i, j_i	Amplitude of the i -th. Harmonic of the angle dependent spring, damper and moment of inertia load elements
$\phi_{Ki}, \phi_{Di}, \phi_{Ji}$	Phase-shift of the i -th. Harmonic of the angle dependent spring, damper and moment of inertia load elements [rad]
$a(\varphi), \alpha(\varphi),$ $b(\varphi), c(\varphi)$	Angle dependent mathematical parameters
N_a, N_α, N_b, N_c	Number of harmonics of $a(\varphi), \alpha(\varphi), b(\varphi), c(\varphi)$
a_i, α_i, b_i, c_i	Amplitude of the i -th. Harmonic of $a(\varphi), \alpha(\varphi), b(\varphi), c(\varphi)$
$\phi_{ai}, \phi_{\alpha i}, \phi_{bi}, \phi_{ci}$	Phase-shift of the i -th. Harmonic of $a(\varphi), \alpha(\varphi), b(\varphi), c(\varphi)$
$T_{dis0}(\varphi)$	Angle dependent disturbance torque [Nm]
$T_{dis1}(\varphi, T)$	Angle and input torque dependent disturbance torque [Nm]
$T_{dis2}(\varphi, \dot{\varphi})$	Angle and angular velocity dependent disturbance torque [Nm]
$T_{dis3}(\varphi, \dot{\varphi})$	Angle and angular velocity (nonlinear) dependent disturbance torque [Nm]
$T_{dis}(\varphi, \dot{\varphi}, T)$	Total internal periodic disturbance torque [Nm]
$\dot{c}(\varphi), \dot{b}(\varphi),$ $\dot{a}(\varphi), \dot{\alpha}(\varphi)$	Angle dependent mathematical parameters
$\mathbf{A}(x)$	(State dependent) system matrix of state space system
$\mathbf{B}(x)$	(State dependent) input (matrix) vector of state space system
$T_{external}(t)$	External disturbance torque [Nm]
$T_{internal}(\varphi)$	Internal disturbance torque [Nm]
φ_D & φ_L	Angular positions of the drive and the load sides respectively [rad]
φ_Δ	Angular position difference between the drive and the load sides [rad]
T_D	Motor torque acts on the drive-side [Nm]
T_L	Disturbance (load) torque acts on the load-side [Nm]
\mathbf{T}	Vector of drive-side and load-side torques [Nm]
J_D & J_L	Moments of inertia of the drive and load sides respectively [kgm ²]
d_D & d_L	Damping factors of the drive and the load sides respectively [Nms/rad]

NOMENCLATURE

K_R & D_R	Spring [Nm/rad] and Damping [Nms/rad] factors of the mechanical link
g	Gear box ratio between the drive and load sides
$a_{DL0}, a_{DL1}, a_{DL2}$	Drive-load denominator polynomial coefficients
b_{D0}, b_{D1}	Numerator drive polynomial coefficients
b_{L0}, b_{L1}, b_{L2}	Numerator load polynomial coefficients
$P_{ND}(s)$	Drive-side numerator polynomial
$P_{NL}(s)$	Load-side numerator polynomial
$P_{DL}(s)$	Drive-load system denominator polynomial
$K_L(\varphi_L)$	Load angle dependent spring (torque) [Nm]
$D_L(\varphi_L)$	Load angle dependent damper [Nms/rad]
$J_L(\varphi_L)$	Load angle dependent moment of inertia [kgm^2]
k_{L0}, d_{L0}, J_{L0}	Load spring, damper and moment of inertia constants
T_{disx}	External disturbance torque acting on the load side [Nm]
β_{xi}	Amplitude of the i -th. Harmonic of external periodic disturbance
ϕ_{xi}	Phase-shift of the i -th. Harmonic of external periodic disturbance
$T_L(t, \varphi_L, \dot{\varphi}_L)$	Total external and internal periodic disturbance torque acting on the load-side [Nm]

Chapter 3 Variables

y	Process (system) output
d	Process disturbance
d_i	Process input disturbance
d_o	Process output disturbance
u	Process manipulated (control) input variable
e	Error between set point input and process output
r	Set point
$G_P(s)$	System (Process) input-output (dynamics) transfer function
$G_C(s)$	Feedback controller transfer function
$G_N(s)$	Periodic disturbance rejection filter (PDRF) transfer function
ω_d	PDRF disturbance frequency
ξ_d	PDRF damping ratio

K_d	PDRF gain factor
y_A, y_B	Process outputs for PDRF variations A and B respectively
$f(u(t), d(t))$	System functional of manipulate and disturbance inputs
$f_u(u(t))$	System manipulated input functional
$f_d(d(t))$	System disturbance functional
$G_u(s)$	System manipulated input transfer function
$G_d(s)$	System disturbance transfer function
$G_c(s)$	Controller transfer function
$\hat{G}_P(s)$	Estimated Process input-output transfer function
\hat{y}	Estimated process (system) output
y_u	Process output response to the manipulated input u
\hat{y}_u	Estimated process output response to the manipulated input u
y_m	Process measured output
e_m	Measurement error
e_{OB}	Observer error
\hat{d}_o, \hat{d}_i	Estimated output and input disturbance respectively
d_{ME}	Modelling error disturbance
$G_{CL}(s)$	Closed loop transfer function
x_d	Extended disturbance state
$\alpha, \beta,$	Amplitudes (factors) of sine and cosine functions respectively
ω_d	Disturbance frequency
a_d, ϕ_d	Amplitude and phase-shift of the disturbance function
G_{FF}	Feed-forward controller transfer function

Chapter 4 Variables

$G_{IM}(s)$	Identification Model (IM) input-output dynamics
n	Order of identification model input-output dynamics
a_0, a_1, \dots, a_{n-1}	Denominator polynomial coefficients of IM input-output dynamics
b_0, b_1, \dots, b_n	Numerator polynomial coefficients of IM input-output dynamics
y	Identification model output
u_{in}	Total input to identification model input-output dynamics

u	Manipulated (control) input
u_{dis}	Disturbance input (external plus internal)
u_{ext}	External disturbance
u_{int}	Internal disturbance
ω	(Measured) external disturbance angular frequency (synchronization signal) [rad/s]
φ_m	Measured process angular position (internal disturbance synchronization signal) [rad]
$\alpha_{0,i}, \beta_{0,i}$	Sine and cosine factors of the external disturbance i -th. Harmonic
N_0	Number of external disturbance harmonics
$\alpha_{1,i}, \beta_{1,i}$	Sine and cosine factors of the internal disturbance i -th. Harmonic (first component)
N_1	Number of internal disturbance harmonics (first component)
$\alpha_{2,i}, \beta_{2,i}$	Sine and cosine factors of the internal disturbance i -th. Harmonic (second component)
N_2	Number of internal disturbance harmonics (second component)
$\alpha_{3,i}, \beta_{3,i}$	Sine and cosine factors of the internal disturbance i -th. Harmonic (third component)
N_3	Number of internal disturbance harmonics (third component)
α_i, β_i	Sine and cosine factors in vector format
$\sin(\omega t), \cos(\omega t)$	External disturbance sine and cosine functions in vector format
$\sin(\varphi_m), \cos(\varphi_m)$	Internal disturbance sine and cosine functions in vector format
$u_{sc}(t)$	Sine cosine vector of external and internal disturbance components
θ	Parameter vector
x_m	(Measured) internal disturbance synchronization state
T_s	Sampling (period) time
$b_i \alpha_j, b_i \beta_j$	Numerator discrete polynomial factors and sine cosine factors multiplications in compact vector format
Z	Data (row) vector

Chapter 5 & 6 Variables

$\vartheta, \dot{\vartheta}$	Angular velocity [rad/s] and its derivative [rad/s ²]
J	Moment of inertia [kgm ²]

D	Damping factor [Nms/rad]
T	Drive (input) torque [Nm]
T_{dis}	Disturbance torque [Nm]
K_P, K_I	Proportional-Integral (PI-) controller gains
G_N	Periodic Disturbance Rejection Filter (PDRF)
ω_d	PDRF disturbance frequency
ξ_d	PDRF damping ratio
K_d	PDRF gain factor
y, \dot{y}	Identification model output and its derivative
a_i, b_i	Identification model dynamic parameters
u	Identification model manipulated input
u_{dis}	Identification model external and internal disturbance
θ	Identification model parameter vector
$K(\varphi)$	Angle dependent spring (torque) [Nm]
$D(\varphi)$	Angle dependent damper [Nms/rad]
$J(\varphi)$	Angle dependent moment of inertia [kgm ²]
$\alpha_{0,i}, \beta_{0,i}$	Identification model external disturbance parameters ($i = 1, 2, \dots, N_0$)
N_0	Number of external disturbance harmonics
$\alpha_{1,i}, \beta_{1,i}$	Identification model internal disturbance parameters ($i = 1, 2, \dots, N_1$)
N_1	Number of internal disturbance harmonics
$\alpha_{2,i}, \beta_{2,i}$	Identification model internal disturbance parameters multiplied by the output ($i = 1, 2, \dots, N_2$)
N_2	Number of internal disturbance harmonics
$\alpha_{3,i}, \beta_{3,i}$	Identification model internal disturbance parameters multiplied by the squared output ($i = 1, 2, \dots, N_3$)
N_3	Number of internal disturbance harmonics multiplied by the squared output
$\alpha_{4,i}, \beta_{4,i}$	Identification model internal disturbance parameters multiplied by the input ($i = 1, 2, \dots, N_4$)
N_4	Number of internal disturbance harmonics multiplied by the input
ω	External disturbance frequency [rad/s]
φ_m	Measured process angular position (internal disturbance synchronization)

	signal) [rad]
φ_D & φ_L	Angular positions of the drive and the load sides respectively [rad]
T_D	Motor torque acts on the drive-side [Nm]
T_L	Disturbance (load) torque acts on the load-side [Nm]
J_D & J_L	Moments of inertia of the drive and load sides respectively [kgm^2]
d_D & d_L	Damping factors of the drive and the load sides respectively [Nms/rad]
K_R & D_R	Spring [Nm/rad] and Damping [Nms/rad] factors of the mechanical link
g	Gear box ratio between the drive and load sides
$K_L(\varphi_L)$	Load angle dependent spring (torque) function load element [Nm]
$D_L(\varphi_L)$	Load angle dependent damper function load element [Nms/rad]
$J_L(\varphi_L)$	Load angle dependent moment of inertia function load element [kgm^2]
$d(t)$	Internal disturbance (chapter 7)

Appendix A Variables

x	Reciprocating linear motion in [m]
\dot{x}	Velocity of the reciprocating linear motion in [m/s]
\ddot{x}	Acceleration of the reciprocating linear motion in [m/s^2]
φ	Rotational motion in [rad]
$\dot{\varphi}$	Velocity of the rotational motion in [rad/s]
$\ddot{\varphi}$	Acceleration of the rotational motion in [rad/s^2]
r	Radius of the rotational motion φ in [m]
T_D	Drive (source) torque (Nm)
k	Linear motion spring constant [N/m]
F_k	Force generated by the linear motion spring load element [N]
T_k	Reaction torque of the linear motion spring load element [Nm]
$K(\varphi)$	Converted rotational angle dependent spring (torque) function [Nm]
K_r	Constant of Converted rotational angle dependent spring function [Nm]
d	Linear motion damping constant [Ns/m]
F_d	Force generated by the linear motion damper load element [N]
T_d	Reaction torque of the linear motion damper load element [Nm]
$D(\varphi)$	Converted rotational angle dependent damper function [Nms]

D_r	Constant of Converted rotational angle dependent damper function [Nms]
D_0	Damping coefficient of the rotational system [Nms]
$D_{total}(\varphi)$	Total damping coefficient function of the rotational system [Nms]
m	Linear motion mass [kg]
F_m	Force generated by the linear motion mass load element [N]
T_m	Reaction torque of the linear motion mass load element [Nm]
$J(\varphi)$	Converted rotational angle dependent moment of inertia function [Kgm ²]
J_r	Constant of Converted rotational angle dependent moment of inertia function [Kgm ²]
J_0	Moment of inertia of the rotational system [Kgm ²]
$J_{total}(\varphi)$	Total moment of inertia function of the rotational system [Kgm ²]
K_p, K_i	Proportional and integral actions of PI-controller
l	Linking rod between the rotational and reciprocating linear motion of the crank mechanism [m]
$n = l/r$	Ratio between the linking rod to the rotational motion radius of the crank mechanism
α	Angle of the linking rod l of the crank mechanism [rad]
\acute{x}	Angle dependent part of the reciprocating linear motion of the crank mechanism [m]

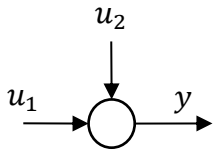
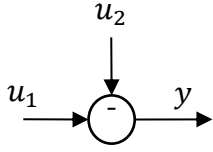
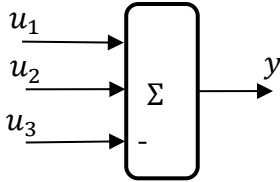
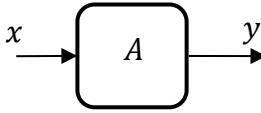
Appendix B Variables

y	Process output due to measured (manipulated, control) input and disturbance input
y_m	Process measured output
y_u	Process output due to measured (manipulated, control) input only
u	Measured (manipulated, control) input
d_o	Output disturbance
e_m	Output measurement error
t	Sampling time sequence index ($t = 0, 1, 2, \dots$)
$G_P(s), G_P(z^{-1})$	Process continuous and discrete transfer functions
n	Oder of the process dynamics
$A(s), B(s)$	Denominator and numerator of the process continuous transfer function

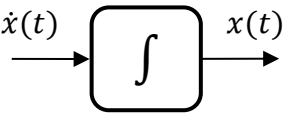
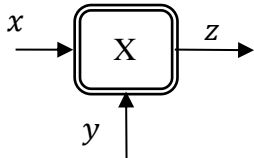
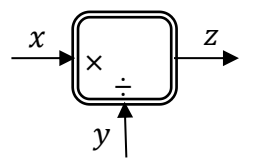
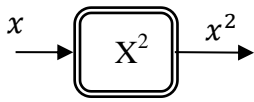
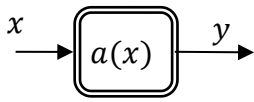
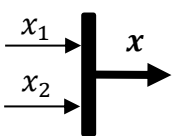
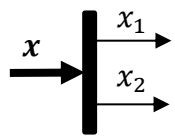
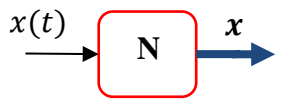
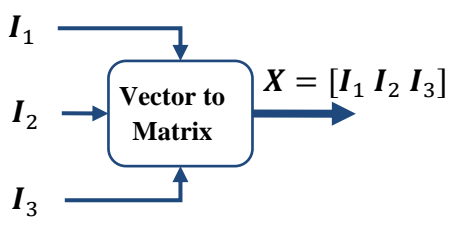
$\beta_0, \dots, \beta_{n-1}$	Numerator polynomial coefficients of the process continuous transfer function
$\alpha_0, \dots, \alpha_{n-1}$	Denominator polynomial coefficients of the process continuous transfer function
$A(z^{-1}), B(z^{-1})$	Denominator and numerator of the process discrete transfer function
b_1, \dots, b_n	Numerator polynomial coefficients of the process discrete transfer function
$\alpha_1, \dots, \alpha_n$	Denominator polynomial coefficients of the process discrete transfer function
ϕ_u	Data vector contains past samples of y_u and u .
θ	Process Parameter vector
ϕ_m	Data vector contains past samples of y_m and u .
e_{EE}	Equation Error (filtered measurement error)
\hat{y}_m	Estimation (Prediction) of the process measured output
$\hat{\theta}$	Estimation (Prediction) of the process parameter vector
$\hat{a}_1, \dots, \hat{a}_n; \hat{b}_1, \dots, \hat{b}_n$	Estimated discrete process parameters
N	Number of sampled input-output pairs
e_{PE}	Prediction error
J_N	Performance index
Φ_m	Data matrix for N sampled input-output data pairs
Y_m	Measured output array for N sampled input-output data pairs
E_{PE}	Prediction error array for N sampled input-output data pairs
E_{EE}	Equation Error array for N sampled input-output data pairs
ζ	White noise with zero mean
\hat{y}_u	Estimation of Process output due to measured (manipulated, control) input only
$\hat{\phi}_u$	Data vector contains past samples of \hat{y}_u and u .
$\hat{\Phi}_u$	Data matrix for N sampled input-(estimated) output data pairs
e_{OE}	Output error
E_{OE}	Output error array for N sampled input-output data pairs
λ	Forgetting factor
P	Covariance matrix

\mathbf{K}	Gain vector
γ	Scalar for covariance matrix initialization
$\mathcal{M}(\boldsymbol{\theta}, u)$	Mathematical model
$J'_N(\boldsymbol{\theta}(k))$	Jacobian of the functional $J_N(\boldsymbol{\theta})$ at $\boldsymbol{\theta}(k)$
$J''_N(\boldsymbol{\theta}(k))$	Hessian matrix of the functional $J_N(\boldsymbol{\theta})$ at $\boldsymbol{\theta}(k)$
N_p	Number of the parameters
\mathbf{E}_N	Output error vector with N samples
$\boldsymbol{\Psi}_N$	Gradient matrix of the error vector with respect to the parameters
\mathbf{H}_N	Hessian matrix approximation
$\hat{\mathbf{x}}$	State estimate
$\mathbf{A}(\boldsymbol{\theta}), \mathbf{B}(\boldsymbol{\theta}), \mathbf{C}(\boldsymbol{\theta})$	Parameterized state space matrices/vectors of a dynamic system
\mathbf{X}_i	Innovation model states
$\boldsymbol{\psi}$	Gradient vector
\mathbf{K}_0	Observer gain

Block Diagram Symbols

	<p>Sum block</p> $y = u_1 + u_2$
	<p>Sum (subtraction) block</p> $y = u_1 - u_2$
	<p>Sum (subtraction) block</p> $y = u_1 + u_2 - u_3$
	<p>Amplifier block</p> $y = Ax$

NOMENCLATURE

	<p>Integrator block</p> $x(t) = \int \dot{x}(t) dt$
	<p>Multiplication block</p> $z = xy$
	<p>Division block</p> $z = x/y$
	<p>Square block</p>
	<p>Nonlinear function block</p> $y = a(x)$
	<p>Scalars to vector block</p>
	<p>Vector to scalars block</p>
	<p>Samples accumulator: It accumulates N samples of $x(t)$ into a vector x.</p>
	<p>Vector to matrix block: It combines the input vectors I_i in an output matrix X.</p>

LIST OF FIGURES

Figure 1.1: Control system with feedback and feed-forward controllers.....	1
Figure 2.1: Drive-load system.....	10
Figure 2.2: Drive-load system using the physical parameters.....	11
Figure 2.3: Drive-load system using the mathematical parameters.....	11
Figure 2.4: Externally disturbed system.....	12
Figure 2.5: Frequency response.....	13
Figure 2.6: Unit step response with different external periodic disturbance frequencies.....	13
Figure 2.7: Eccentric mechanisms.....	16
Figure 2.8: Drive-load system with angle dependent mathematical parameters.....	16
Figure 2.9: Drive-load system with internal periodic disturbances.....	18
Figure 2.10: Nonlinear state space representation of the drive-load system.....	18
Figure 2.11: Separated linear nonlinear state space representation of the drive-load system.....	19
Figure 2.12: Drive-load system with nonlinear spring, damper and approximate moment of inertia load elements.....	21
Figure 2.13: Externally and internally disturbed rigid drive-load system.....	21
Figure 2.14: Stair case step response and the variable nonlinear spring torque value.....	23
Figure 2.15: Stair case step response and the variable state dependent damper load element.....	23
Figure 2.16: Stair case step response and the variable angle dependent moment of inertia for exact $1/J$ and constant approximation of $1/J$ cases.....	24
Figure 2.17: The drive-load system with gear.....	25
Figure 2.18: The physical parameters of the flexibly linked drive-load system with gear.....	25
Figure 2.19: Block diagram of the flexibly linked drive-load system.....	27
Figure 2.20: Compact block diagram of the flexibly linked drive-load system.....	27
Figure 2.21: Block diagram of the flexible drive-load system.....	28
Figure 2.22: Block diagram of the flexibly linked drive-load system with angle dependent physical parameters.....	31
Figure 2.23: Flexible drive-load system with a separate external and internal periodic disturbance torque function.....	34
Figure 2.24: Block diagram of externally and internally disturbed flexibel drive-load system.....	35
Figure 2.25: Externally and internally disturbed flexible drive-load system in state space representation.....	35
Figure 3.1: Feedback control of disturbed process.....	36
Figure 3.2: Addition of PDRF in two variations A and B.....	39
Figure 3.3: The PDRF poles in s -plane for $\omega d = \gamma$ and $\xi d = 0$	39
Figure 3.4: Disturbance frequency inside the (demanded) system bandwidth.....	40
Figure 3.5: Disturbance frequency outside (demanded) system bandwidth.....	41

LIST OF FIGURES

Figure 3.6: Disturbances and their feed-forward control.	44
Figure 3.7: Output-input disturbance observer.	45
Figure 3.8: Computing the input disturbance without using the inverse of estimated dynamics.	46
Figure 3.9: Model following control concept.	47
Figure 3.10: The adaptive periodic disturbance canceler.	50
Figure 3.11: Feed-forward control of internally and externally disturbed system.	52
Figure 3.12: Phasor diagram.	53
Figure 3.13: Load-side disturbance feed-forward control of the flexible drive-load system.	53
Figure 3.14: Input disturbance feed-forward control of the flexible drive-load system.	53
Figure 4.1: General control strategy of the adaptive set point feedback controller and periodic disturbance feed-forward controller.	55
Figure 4.2: Externally and internally disturbed identification model.	56
Figure 4.3: Identification model for externally and internally disturbed system.	62
Figure 5.1: Example 1.1: Closed loop frequency response to the set point and the disturbance for the parameter sets A and B of the PI feedback controller.	70
Figure 5.2: Example 1.1: Closed loop set point step response for parameter sets A and B of the PI feedback controller.	70
Figure 5.3: Example 1.2: Set point and disturbance frequency response of closed loop system for only PI controller (parameter set A), with PDRF variant A and with PDRF variant B.	71
Figure 5.4: Example 1.2: Set point step response of closed loop system for only PI (parameter set A), with PDRF variant A and with PDRF variant B.	71
Figure 5.5: Example 1.3: Open-loop identification run.	73
Figure 5.6: Example 1.3: Compensation in open loop.	74
Figure 5.7: Example 1.3: Compensation in closed loop under PI controller.	74
Figure 5.8: Example 1.3: Closed loop adaptive run.	75
Figure 5.9: Example 1.4: 5-harmonics open loop identification run.	77
Figure 5.10: Example 1.4: 5-Harmonics open loop compensation.	78
Figure 5.11: Example 1.4: 5-Harmonics closed loop compensation.	78
Figure 5.12: Example 1.4: Adaptive run.	79
Figure 5.13: Example 1.5: Open loop identification.	82
Figure 5.14: Example 1.5: Periodic disturbance compensation in open loop.	83
Figure 5.15: Example 1.5: Periodic disturbance compensation in closed loop.	83
Figure 5.16: Example 1.5: Adaptive run.	84
Figure 5.17: Example 2.1: Frequency response of the drive and disturbance load torques.	87
Figure 5.18: Example 2.1: Comparison between real and identified dynamics.	87
Figure 5.19: Example 2.1: Open loop identification run.	88
Figure 5.20: Example 2.1: Periodic disturbance compensation in open loop.	89
Figure 5.21: Example 2.1: Periodic disturbance compensation in closed loop.	89
Figure 5.22: Example 2.1: Adaptive run.	90

Figure 5.23: Example 2.2: Frequency response of the flexible drive-load system.	92
Figure 5.24: Example 2.2: The frequency response of the real flexible and the identified first order system dynamics.	92
Figure 5.25: Example 2.2: Closed loop adaptive run.	93
Figure 5.26: Example 2.3: Closed loop adaptive run.	94
Figure 5.27: Example 2.4: Frequency response of the real and the identified system dynamics.	95
Figure 5.28: Example 2.4: Closed loop adaptive run.	96
Figure 5.29: Example 2.5: Closed loop adaptive run.	99
Figure 5.30: Example 2.6: Closed loop adaptive run.	100
Figure 5.31: Example 2.6: Periodic disturbance compensation in open loop.	101
Figure 5.32: Example 2.6: Periodic disturbance compensation in closed loop.	101
Figure 5.33: Example 2.7: Closed loop adaptive with 5-harmonics.	103
Figure 5.34: Example 2.8: Closed loop adaptive with 3-harmonics.	104
Figure 6.1: Overview of the torsional plant real-time control system from Educational Control Products Company.	107
Figure 6.2: Matlab-Simulink diagram of experiment 1.1.	108
Figure 6.3: The system time response of experiment 1.1.	108
Figure 6.4: The Matlab-Simulink diagram of experiment 1.2.	109
Figure 6.5: Experiment 1.2 run.	110
Figure 6.6: Experiment 1.3 run.	112
Figure 6.7: MatLab-Simulink diagram of experiment 1.4.	115
Figure 6.8: MatLab-Simulink diagram of experiment 1.5.	115
Figure 6.9: Experiment 1.4 run.	116
Figure 6.10: Experiment 1.5 run.	117
Figure 6.11: Experiment 1.6 run.	118
Figure 6.12: The schematic diagram of the drive-load self-excited machine.	121
Figure 6.13: The drive-load self-excited rotational machine.	122
Figure 6.14: The crankshaft mechanism moving a mass, spring and damper load.	122
Figure 6.15: Time and finite DFT frequency spectrum estimate without compensation.	125
Figure 6.16: Time and finite DFT frequency spectrum estimate without compensation (gray line) and with 4-harmonics compensation (black line).	126
Figure 6.17: Identification and compensation of 5-harmonics.	127
Figure 6.18: System output of uncompensated; compensated with 4-harmonics and compensated with 5-harmonics.	128
Figure 6.19: The finite DFT frequency spectrum estimates of uncompensated; compensated with 4-harmonics and compensated with 5-harmonics.	128
Figure 6.20: Stepwise variable set point and periodic disturbance compensation.	129
Figure 6.21: Consistently varying set point tracking and periodic disturbance compensation.	130

Appendix A

Figure A.1: Scotch yoke mechanism..... 137

Figure A.2: Block diagram of the Scotch yoke dynamic model. 140

Figure A.3: Angular velocity finite DFT frequency spectrum estimate of Scotch yoke mechanism with linear motion damper and spring load elements. 141

Figure A.4: Angular velocity finite DFT frequency spectrum estimate of Scotch yoke mechanism with linear motion mass load element..... 141

Figure A.5: Crank mechanism moving a mass, spring and damper load elements..... 142

Figure A.6: Time plot (top) and frequency spectrum plot (bottom) of the angle dependent linear motion part at 50 [rad/s] rotational velocity. 145

Figure A.7: Time plot (top) and frequency spectrum plot (bottom) of angle dependent linear motion velocity part at 50 [rad/s] rotational velocity..... 146

Figure A.8: Time plot (top) and frequency spectrum plot (bottom) of angle dependent linear motion acceleration part at 50 [rad/s] rotational velocity..... 146

Appendix B

Figure B.1: Disturbed process with output measurement error. 150

Figure B.2: Prediction error system identification. 153

Figure B.3: Output error system identification. 154

Figure B.4: Output error parameter identification strategy..... 157

Figure B.5: Batch (offline) iterative (successive) Gauss-Newton output error method..... 161

Figure B.6: Online recursive Gauss-Newton output error method. 163

Figure B.7: Prediction error parameter identification strategy. 164

1 INTRODUCTION

1.1 Problem Description

1.1.1 Set Point Tracking and Disturbance Rejection

Set-point tracking and disturbance rejection are the main goals to achieve in control system theory and its applications. The main job of a controller is first to guarantee the set point tracking and second is to suppress the effect of any kind of disturbances acting on the system. The set point tracking and disturbance rejection in control applications can be done by using a negative feedback controller, a feed-forward controller or a combination of them. Figure 1.1 shows a block diagram of a typical control system that has a feedback controller, a set point feed-forward controller and a disturbance feed-forward controller. To realize the feedback controller, the process output must be measured. Also, to realize the disturbance feed-forward controller, the disturbance must be available either by direct measurement, observation, estimation or prediction.

1.1.2 Disturbances in General

Generally, the disturbance models a real physical effect on the process either that comes from inside (internal) or outside (external) of the process, see Figure 1.1. On the other hand, noise is also assumed to be some sort of disturbance that usually models, for example, the errors that take place during the measurement of the controlled variable which is the output. Therefore, the influence of the disturbance on the process output should be regulated, while the measurement error (noise) needs to be canceled from the measurements, or it should be as small as possible to be ignored. Moreover, disturbance can be classified as deterministic signal (for example constant, ramp, sine or square wave, etc.) or stochastic (random) signal, stationary signal (its parameter does not change with time) or non-stationary (for time-varying parameters). However, the periodic disturbances (oscillations) are exclusively considered in this work.

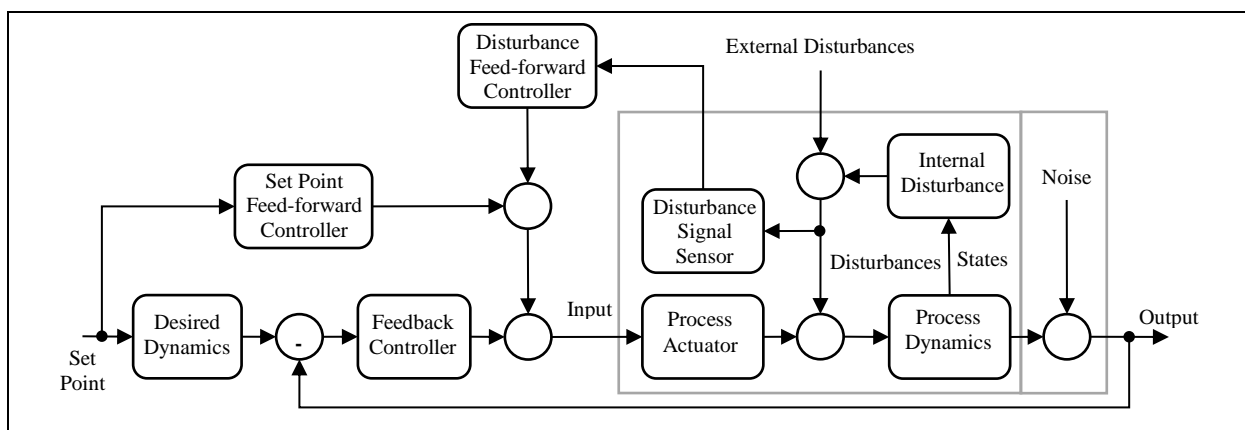


Figure 1.1: Control system with feedback and feed-forward controllers.

1.1.3 Oscillation Sources and their Classification

Undesirable oscillations are in particular assumed to be periodic disturbances, which could be generated by external sources, called forced oscillations, or internally by linear system dynamics, called free response oscillations. If one or more of the system modes are excited, this could lead to a constant oscillation (pure sinusoidal oscillation when a system has a pair of complex conjugate imaginary poles or more at the imaginary-axis in the s -plane), or a transient oscillation when the system has positive or negative damping coefficient. Moreover, sustained oscillations could also take place because of nonlinear system dynamics (Harris and Piersol 2002). For example, limit cycle oscillations induced from hysteresis elements, or oscillations generated because of state dependent (periodic) parameters like in rotational drive-load systems, which could cause the rotor angular position or velocity to oscillate, and this is mainly because of angle dependent drive or load parameters. Moreover, these types of oscillations can also be called or classified as self-excited oscillations.

In practice, such oscillations could actually be induced in the angular velocity servo control drive-load systems either from the drive side and/or from the load side. The drive side systems are, for example, reciprocating engines, where according to their construction and working principles, they generate angle dependent pulses of torque that cause oscillation in the rotational velocity. Therefore, a lot of work has been done in order to reduce these oscillations, for example, Zaremba, Burkov and Stuntz (1998) have developed a control algorithm to reduce the oscillations of the engine idle speed, Gusev, Johnson and Miller (1997) have developed an active flywheel algorithm to reduce the engine speed oscillation and Njeh, Cauet and Coirault (2010) have developed a new control strategy to reduce the torque ripples of the combustion engine in hybrid electric vehicles. Moreover, electrical drives also have torque ripples that need to be reduced, for example, in induction motor (Basu, et al. 2010; Itoh and Kubota 2005), as well as the cogging torque in permanent magnet synchronous-motors (Li and Slemon 1988; Jahns and Soong 1996; Holtz and Springob 1996; Petrovic, et al. 2000; Gan and Qui 2004; Wang, Gan and Qiu 2007; Maier, Bals and Bodson 2011), etc.

On the other hand, the angle dependent load machines can also be the main source of oscillations in the drive-load systems, as in load machines with reciprocating motion like in crankshaft or camshaft machines. As examples for these types of machines are reciprocating air compressor machines, rectilinear or reciprocating saw machines, weaving machines (Smolders, et al. 2005), etc.; also machines with undesirable eccentricity like disc drive systems (Sacks, Bodson and Khosla 1993; Nagashima, Usui and Kobayashi 2006); or in noncircular roll machines (Garimella and Srinivasan 1994; Kugi, et al. 2000; Shin, et al. 2003).

Therefore, it is very important for the performance of the drive-load system to prevent (reject) the oscillation generated at the drive side to go to the load side, or the oscillation generated at some load to affect the other coupled loads, especially, when drive-load system has more than one load linked together. Actually, this can be done by using passive control methods, by using passive elements (flywheels, dampers), or semi-active control by using passive elements with tunable (adaptive) parameters. Nevertheless, this work is mainly interested in active control methods that generate an anti-action (force, torque ...) actively against the periodic disturbances by utilizing the already existing computer control system.

1.2 Periodic Disturbance Rejection Methods

Disturbance rejection is a classical problem of control system theory and its applications. Periodic (sinusoidal) disturbance is a special case that has its distinction discretely in the frequency domain. The integral part of the classical Proportional-Integral-Derivative (PID) controller deals explicitly only with a constant disturbance that has a zero frequency component. Therefore, to compensate a sinusoidal disturbance, an internal model representing this disturbance should be added to a controller according to the Internal Model Principle (IMP) (Francis and Wonham 1976; Bodson 2005) by means of complex conjugate poles at the disturbance frequency. The added internal model will alternatively be called in this context as Periodic Disturbance Rejection Filter (PDRF). The filter will create a notch in the resulted closed loop disturbance frequency response. Moreover, the disturbance frequency is assumed to be either known as the case of internal disturbance in rotational machines or measurable in the case of external periodic disturbance.

However, most of the use of the notch filters in control applications is to shape the frequency response of the feedback control system in order to suppress the resonance modes of flexible structure characteristics, which are actually the cause of the oscillations (see, for example, Schmidt and Rehm 1999; Yoo, et al. 2011). On the other hand, the “inverse” notch filter can also be used to suppress the external periodic disturbances (see, for example, Zaremba and Davis 2000). But the challenge here is to compromise between a perfect disturbance rejection and the set point tracking characteristics mostly in terms of the closed loop system relative stability (for more details of periodic disturbance rejection, see chapter 3). Generally, for multi-harmonic disturbances, a number of PDRFs can be designed and used to reject every harmonic distinctively or one PDRF for a band of disturbance frequencies. For the case of infinite harmonics, a repetitive control filter can be applied to reject them, for example, as presented by Kobayashi, Hara and Tanaka (1990); Steinbuch (2002); Quan and Cai (2010).

Also, Iterative Learning Control (ILC) can be used to reject periodic disturbances, although it is originally designed to optimize the repeated (set-point trajectory) task tracking of robots. However, the ILC can be used to iteratively learn (estimate or adapt) a proper (feed-forward) control signal to reject a periodic disturbance given in the repeated task period, which is at the end acts as repetitive control algorithm, for example, see (Zaremba, Burkov and Stuntz 1998; Kim, et al. 1998). Moreover, the ILC control learns (estimates or adapts) the control signal from the past iteration to optimize a repetitive task, rather than estimating the controller parameters as in the case of adaptive control, for more general information about ILC refer, for example, to (Bristow, Tharayil and Alleyne 2006; Ahn, Chen and Moore 2007).

Furthermore, the disturbance can be cancelled, if it is available, by using a feed-forward controller. However, in practice, the disturbance is not always directly available (measurable). Alternatively, it can be computed by using a disturbance observer. But, the implementation of a disturbance observer in estimation and cancellation of the disturbance simultaneously, makes the method to function as a pure feedback control technique, particularly when the disturbance observer uses the output signal in estimation of the disturbance signal. The disturbance observer can be constructed, for example, as transfer function based disturbance observer (Ellis 2002; Du, et al. 2010; Ohishi 1987; Ohishi 2010), or as state space

based disturbance observers extended from state space observers (Kalman and Bucy 1961; Luenberger 1964). Moreover, these methods are realized, for example, by designing a set of linear time-invariant observers for a set of disturbance frequencies in some operating regions, so the resulted design is a gain scheduled control (see, for example, Bohn, et al 2004). Additionally, the controller design can also be carried out by using robust control Linear Parameter Varying (LPV) techniques, for example, Du and Shi (2002), Du, et al. (2003) and Kinney and Callafon (2006) have developed H_∞ and LPV methods respectively for active vibration control application; or Ballesteros and Bohn (2011) have developed and applied the algorithms in discrete-time format.

Alternatively, the Adaptive Feed-Forward Control (AFFC) algorithms can also be used to estimate and compensate the unavailable periodic disturbance signal in a form of its finite Fourier series. But, most of the AFFC strategies, for example, Least Mean Squares (LMS) family methods (Widrow and Stearns 1985; Hansen 2001), are originally developed for pure active noise and vibration control applications to suppress the vibrations generated in flexible structures, which are globally stable in nature. Moreover, these methods are usually implemented by actuators that are acting locally on the system without constant set point tracking and constant disturbance rejection objectives. Therefore on the contrary, these strategies do not suit in particular this work control problem because of (global) stability issue, parameter convergence and the online direct adaptation (tuning) problems (Elliott 2001; Bodson 2005). This means that as long as the algorithm does not converge, the system operates with very poor performance and that will be until and only if the parameters converge.

Generally, most of AFFC algorithms incorporate the output signal in the estimation of the periodic disturbance which makes them also function as a pure feedback control algorithm. However, exception could be made as the direct adaptation is deactivated, when the parameters converge. Notwithstanding this, for a rapid variation of disturbance frequencies case like in the current work main problem (the velocity servo control of rotational machines with angle dependent load elements), the AFFC algorithms are more attractive and suitable for this case, since their theory and applications do not depend on the assumption that the disturbance amplitude and frequency are constants, slowly varying or linear time-varying (Bodson 2004, Section 5).

Moreover, for practical applications, Wit and Parly (2000) have developed an adaptive eccentricity compensation algorithm, which is an AFFC, also Maier, Bals and Bodson (2011) have developed an AFFC to cancel the ripples of a permanent magnet synchronous-motor, but both of them are with a direct controller parameter (tuning) optimization algorithm. While in this work the problem of Feed-Forward Control (FFC) adaptation will be transferred into a system identification (parameter optimization) problem, where the identification model is constructed by two parts, dynamic part in terms of linear or nonlinear differential or difference equations and a disturbance part in terms of sine cosine sum functions (Bodson 2005) that are related linearly or nonlinearly to external and internal system variables. Utilizing the identification model, this will allow the identification to gather information online (offline as a priori) about the process and to pass the converged parameters to use in the FFC law, even when the FFC is deactivated, particularly at the starting phase when the dynamic and disturbance parameters are far from the optimal ones.

1.3 Main Distinctions of the Developed Algorithm

First of all, it is a general strategy that can be interpreted to and implemented in many variations, as stated in section 4.4. It classifies the disturbances as external and internal disturbances and put them in a more general format, while most of the developed algorithms model the disturbance simply as external disturbances even if they are really correlated to the system variables or parameters. Where in this work developed algorithm allows and introduces this possibility to achieve more global modelling, which of course increases the system performance in a wider region about the operating point.

Moreover, it (approximates) models and identifies the periodically disturbed system as input-output dynamics, and input external and/or internal disturbances, which is presented as identification model. This makes the realization of both adaptive feedback control and adaptive feed-forward control strategies possible. Besides, using the input disturbance format makes its implementation very easy in computation without extra transformations.

Furthermore, It uses system identification algorithms to identify the identification model which makes it as an indirect adaptive algorithm, while most of the other algorithms use direct adaptive methods to tune the (disturbance) feed-forward controller parameters or use indirect adaptive method but by using (fixed) pre-identified process model. For example, Wit and Parly (2000) have developed an adaptive eccentricity compensation algorithm, which is an adaptive feed-forward control, Maier, Bals and Bodson (2011) have also developed an adaptive feed-forward to cancel the ripples of a permanent magnet synchronous motor, but both of them are with a direct controller parameter (tuning) optimization algorithm. Moreover, Na and Park (1997) have developed an adaptive feed-forward control that works like the typical LMS-algorithm but in indirect way, by using pre-estimated process and closed loop transfer functions, also Wang and Ren (1994) have developed an indirect adaptive feed-forward control for and to be applied exclusively in active noise and vibration control systems.

1.4 Targets of this Work

The main objective here is to develop an active control method to reject/compensate the continual oscillations that are generated in an angular velocity servo control of a rotary drive-load system. These oscillations are an undesirable (nonlinear) behavior and considered as an internal (self-excited) periodic disturbance generated mainly because of that the load machine has an angle (state) dependent (nonlinear) spring, damper or moment of inertia load element or their combinations.

Moreover, the problem is also extended to oscillations that have no relation to the system dynamics and considered as an external periodic disturbance. Using the developed strategy, a feed-forward controller will be designed mainly to identify and compensate the internal and the external periodic disturbances and it should work as an add-on feed-forward controller to an already existing set-point tracking feedback controller also with a minimum interference between them. This has to be done without using extra drive. The method will be applied and tested in simulation as well as in real-time control experiments.

Furthermore, a drive-load test platform machine will also be built up as an angular velocity servo control system, where the load machine is constructed to have a crankshaft mechanism that moves a mass in a translated reciprocating linear motion up and down against spring, damper and mass forces that generate the undesirable nonlinear angle (state) dependent oscillations. Finally, the developed algorithm will be applied in real-time control on this test platform to identify and to cancel these unwanted oscillations.

1.5 Contributions and Outline of this Work

One of the main contributions of this work, presented in **chapter 2**, is the modeling, analysis and simulation of the angular velocity servo control drive-load system with angle (state) dependent (periodic) spring, damper and moment of inertia load elements. Where, models for the angular velocity drive-load system are developed as rigid as well as flexible systems with external and internal (self) excited periodic disturbances. First, the models are developed by using the physical parameters explicitly, then by using the mathematical parameters. These models have state (angle) dependent (periodic) parameters.

Moreover, since the state dependent (periodic) parameters are the main cause of the oscillations in the system, which is an undesirable (nonlinear) behavior that needs to be rejected or compensated. Therefore, the cause of this behavior is separated and redefined in variables space as internal periodic disturbance acting on the load side. This transformation of the state dependent periodic parameters to an (input) internal periodic disturbance and the extension to external periodic disturbance, are the main inspiration for developing the control strategy of periodic disturbances later in chapter 4.

In **chapter 3**, a brief introduction and a brief literature review are presented in general to the methods of periodic disturbance rejection by using only feedback control techniques to achieve the demands of set point tracking and periodic disturbance rejection by implementing the internal model principle in the feedback controller, for example, by using the periodic disturbance rejection filter (“inverse” notch filter) for a single harmonic or a repetitive control for infinite harmonics of a periodic disturbance.

Furthermore, the methods of feed-forward control are also briefly introduced in general when the disturbance is directly available. Alternatively, the methods of (pseudo) feed-forward control when the disturbance is not directly available, and therefore, the disturbance is computed (estimated) by observing its effect on the system output, for example, by using the transfer function based observer and the state space (Luenberger) observer by implementing the disturbance internal model. Moreover, the adaptive direct feed-forward periodic disturbance estimation and compensation algorithm (for example, LMS family) that was originally developed for active control of noise and vibration applications is briefly introduced.

In **chapter 4**, the main contribution of this work is the development and presentation of a new feed-forward control of periodic disturbances (Alsogkier and Bohn 2012), that is originally inspired and stemmed from the adaptive feed-forward active noise and vibration control applications, but the direct adaptation problem of disturbance parameters is solved by transforming it into system identification optimization problem in form of parametric identification algorithms. This algorithm has turned out to be an indirect adaptive method. The feed-forward

controller is exclusively implemented to model, identify and to compensate the (external and the internal) periodic disturbances as an add-on to an already existing feedback controller.

The general strategy, feedback control for set point tracking and feed-forward for periodic disturbance rejection, is achieved by constructing an identification model representing the input to output dynamics of the periodically disturbed process together with the external and internal input periodic disturbance. Therefore, by identifying the identification model, the adaptation can be practically done to both the feedback controller and the feed-forward controller. Furthermore, the identification model is constructed in general for a drive-load system with external and internal periodic disturbances for continuous and discrete nonlinear in parameters by using nonlinear least squares (Gauss-Newton method), also linear in parameter discrete identification model is developed to be used with linear least squares.

Chapter 5 contribution, the drive-load system is simulated by using its rigid and flexible models with external and internal (self-excited) periodic disturbances that have been developed in chapter 2. In these simulation examples, the system is constructed as an angular velocity servo control system to follow the given set point tracks. And moreover, because of the external and/or internal periodic disturbances, oscillations appear on the system output, the methods introduced in chapters 3 and 4 are applied in order to reject these periodic disturbances or to compensate their effect on the system output. Therefore, the problem of the set point tracking and periodic disturbance rejection is tried to be done first by using feedback control (Simulation Example 1.1) only and second by incorporating a periodic disturbance internal model according to IMP (Francis and Wonham 1976) (Simulation Example 1.2).

Then, the problem is done by using the developed strategy of feedback control for set point tracking and an add-on feed-forward for periodic disturbance compensation introduced in chapter 4. Where in simulation example 1.3, the feedback for set point tracking and feed-forward for periodic disturbance compensation is done for single harmonic disturbance, while in simulation example 1.4, the multi-harmonic external periodic disturbance is considered. Simulation example 1.5 considers the externally and internally (self-excited oscillations) periodically disturbed system case.

Moreover, the externally excited periodic disturbance compensation in flexible drive-load system is considered for rigid link case (simulation example 2.1), and simulation examples 2.2-4 for the flexible link cases. Simulation examples 2.5 and 2.6 present the internally excited periodic disturbance compensation for local and global identification models respectively. Furthermore, simulation examples 2.7 and 2.8 present the periodic disturbance compensation using discrete local and global identification models respectively, where the identification models are identified by using the method of recursive least squares.

Chapter 6 contribution, the developed algorithms, modeling, identification and control of periodic disturbance compensation, after their successful implementation and evaluation in the simulation experiments, are implemented now in real-time controllers and tested on real mechanical plants. The torsional plant is presented in the first section, while the self-excited machine is presented in the second section. Moreover, a brief description to the torsional test plat form is given first, then the external periodic disturbance rejection using periodic disturbance rejection filter is presented in experiment 1.1. While, continuous and discrete

modeling, identification and feed-forward control as an add-on to a feedback controller mainly to compensate the external periodic disturbance are presented in examples 1.2 and 1.3 respectively, and also in examples 1.4 and 1.5 correspondingly for the internal periodic disturbance. Furthermore, experiment 1.6 is the same as in 1.5 but with a variable set point signal.

Also in the second section 6.2, a short description to the self-excited machine test platform is given. Then in subsection 6.2.2 the frequency spectrum analysis is done at different system output (angular velocity) levels under only PI feedback controller first without periodic disturbance compensation then with (4-harmonic) periodic disturbance compensation. Then according to the frequency spectrum analysis results, a first sub-harmonic component is also modeled, identified and controlled together with the principal and three super-harmonics (subsection 6.2.3), this is tested on constant, stepwise variable and linear time variable set point signals.

Finally, **chapter 7** will give some summary and conclusion as well as theoretical and practical future work suggestions as extensions to this work.

Extra more, in **Appendix A**, simple eccentric mechanisms, Scotch yoke and crank mechanisms, are introduced. Where, in the first section, the kinematics and the dynamics are presented when the Scotch yoke mechanism has spring, damper and mass linear motion load elements. Moreover, a dynamic model for this system is constructed. Consequently, the frequency spectrum analysis for the angular velocity oscillation is presented for different linear motion load parameters. While, in the second section, the kinematic analysis of the crank mechanism is presented, as well as the frequency spectrum analysis of its transformed reciprocating linear motion, velocity and acceleration.

Furthermore, in **Appendix B**, a brief introduction is given to the system identification in general and, in particular, to the parametric identification methods for continuous and discrete models with linear and nonlinear in parameters format. The contribution of this appendix is that, these methods are represented in graphical block diagram format, so that they can be easily programmed in Simulink-Matlab from MathWorks Company, which is a graphical-oriented programming environment. This will make them easy to compile and to implement online and in real-time control by using, for example, the dSPACE Company real-time controller utilities.

2 MODELING AND SIMULATION

Modeling is the art of describing an observed physical process either by analogy in terms of small sizing of a very large process or large sizing of a very small process, or by using the analogy between different fields of physical processes, for example, the analogy between mechanical and electrical fields. This type of modeling is generally called analog computing, which it was and still is the ultimate way of modeling and simulation of physical processes in all engineering fields. However, in the recent decades, the rapid development of digital computers in terms of their computing power, cost and availability has made the way open for digital simulation, where it computes the system behavior by using its mathematical model that was built based on the principles of physical laws. Therefore in this chapter, the digital modeling and simulation are mainly presented for a drive-load system with angle dependent (spring, damper and moment of inertia) rotational load elements that actually cause the self-excited oscillations in the system.

2.1 Periodically Disturbed Drive-load Systems

Oscillation is a special disturbance case that appears frequently in rotational machines, particularly when some of the load parameters are angle dependent, in other words, it can also be called state dependent (periodic) parameters. This can be because of an eccentricity, non-uniform machine parts or the internal design of the load machine, for example, camshaft or crankshaft mechanisms (piston compressor, textile machine, etc.) (Vinogradov 2000; Östman and Toivonen 2008; Holm, et al. 2012). This type of effect can be modeled as internal periodic disturbance which is also called self-excited oscillation. But, when the disturbance parameters (amplitude, frequency) are not dependent on the angular position or on any other internal variable or parameter of the machine in particular or the process in general, then this disturbance type is modeled as an external periodic disturbance.

So the challenge here is whether to model these variations in parameter space, which is used normally for constant or at most slowly varying events, or in state variable space, which is usually used to model the relatively very fast changes.

In this chapter, some different forms of the drive-load machines are presented, and correspondingly modeled, starting with rigid body system, section 2.2 which matches the form of a 1 Degree Of Freedom (1DOF) system and can be modeled, using the physical laws to relate the system physical variables mathematically, by Linear Time-Invariant Differential Equation (LTI-DE). Where, LTI-DEs are particularly used to describe systems that are to some extent assumed to have linear dynamics with constant parameters. Then in subsection 2.2.1, the system is considered to have some sort of forced oscillation classified as an external periodic disturbance. Moreover, the angle dependent spring, damper and moment of inertia load elements are added to the drive-load system in subsection 2.2.2, where subsection 2.2.3 discusses the disturbance format of the system. Subsection 2.2.4 presents the effects of the angle dependent load elements separately on the rigid drive-load system.

Furthermore, in section 2.3, the flexible body is considered and in subsection 2.3.1 modeled as a 2 Degree Of Freedom (2DOF) system which described by two LTI-DEs. Subsection 2.3.2 considers the nonlinear angle dependent spring, damper and moment of inertia load elements and their combinations correspondingly. At the end, a general discussion is presented in section 2.4 about how many degrees of freedoms should be considered.

The target of the constructed models is the analysis and the simulation of drive-load system and the impact of the external and the self-excited periodic disturbances on the system. Consequently, they will be used to analyze the angular velocity servo control of the drive-load system and to develop periodic disturbance rejection algorithms later on in chapters 3 and 4.

2.2 Rigid Drive-load System

This is a simple rotational mechanical system, which is simply a motor that drives an inertial load through a rigid mechanical link (shaft) against a damping torque plus a disturbance torque, see Figure 2.1. Only the load side is considered and for simplicity the gear between the motor and the load is eliminated from the figure.

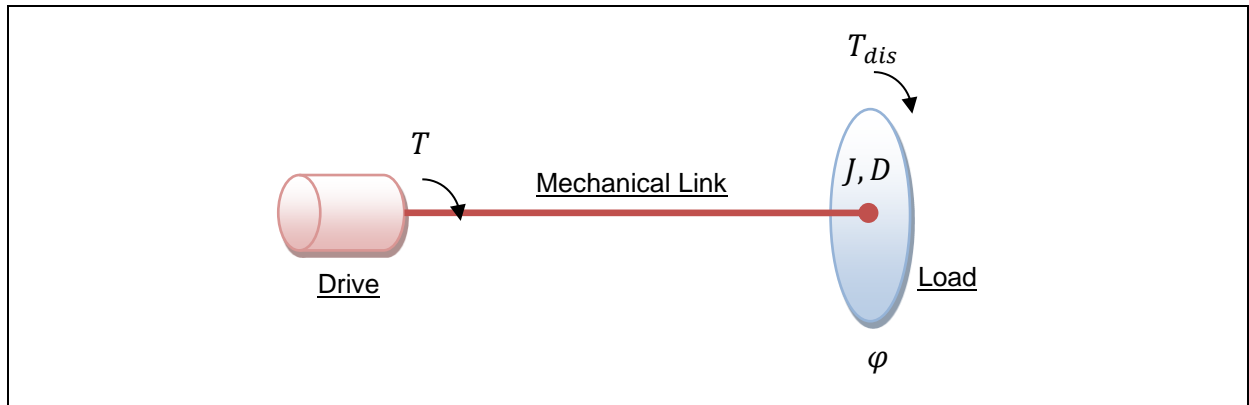


Figure 2.1: Drive-load system.

The linear differential equation that describes the drive-load system, in terms of angular position derivatives, is given as

$$J\ddot{\varphi}(t) + D\dot{\varphi}(t) = T_{in}(t), \quad (2.1)$$

$$T_{in}(t) = T(t) + T_{dis}(t). \quad (2.2)$$

Now, if the angular velocity is defined as $\vartheta(t) = \dot{\varphi}(t)$, then the same differential equation (2.1) but in terms of angular velocity and its derivative can be rewritten as

$$J\dot{\vartheta}(t) + D\vartheta(t) = T(t) + T_{dis}(t), \quad (2.3)$$

$$\dot{\vartheta}(t) = -\frac{D}{J}\vartheta(t) + \frac{1}{J}[T(t) + T_{dis}(t)], \quad (2.4)$$

where φ is the angular position [rad], $\vartheta = \dot{\varphi}$ is the angular velocity [rad/s], J is the moment of inertia [kgm^2], D is the damping factor [Nms/rad], T_{in} is the total torque acting on the load [Nm], T is the drive (input) torque [Nm] and T_{dis} is the (external) disturbance torque [Nm]. The system differential equation of the drive-load system can also be represented schematically as in the following Figure 2.2.

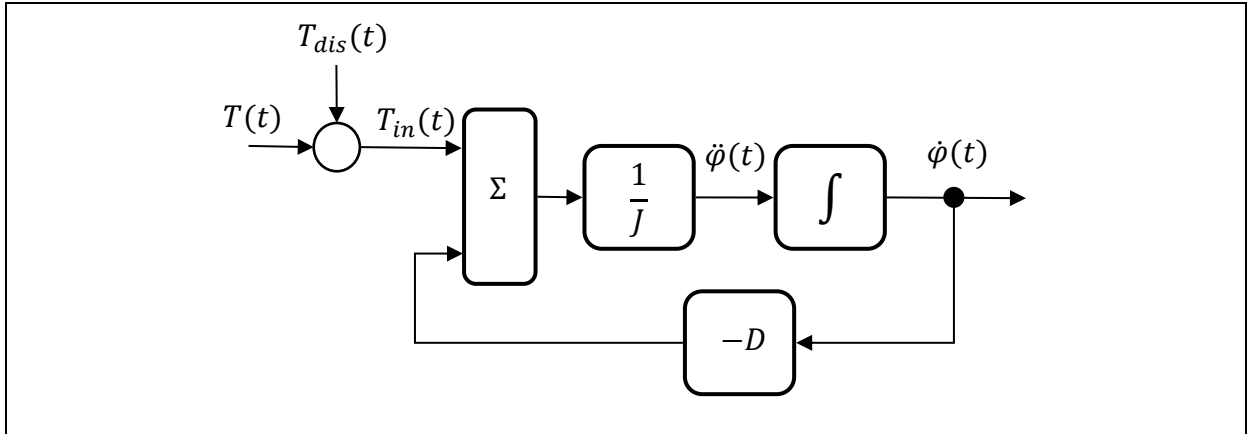


Figure 2.2: Drive-load system using the physical parameters.

Equation (2.4) has been written using the physical parameters of the plant, alternatively, it can also be rewritten using mathematical parameters, when $a = D/J$ and $b = 1/J$, as following

$$\dot{\vartheta}(t) = -a\vartheta(t) + b[T(t) + T_{dis}(t)]. \quad (2.5)$$

The system block diagram of equation (2.5) is presented in Figure 2.3. Furthermore, the drive-load system transfer function can also be put in this form

$$\vartheta(s) = \frac{b}{s + a} [T(s) + T_{dis}(s)]. \quad (2.6)$$

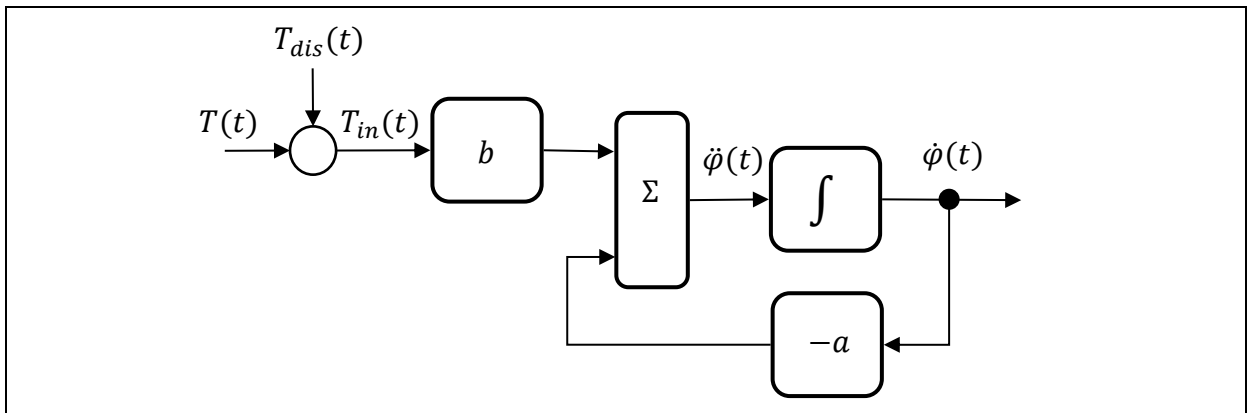


Figure 2.3: Drive-load system using the mathematical parameters.

2.2.1 Externally Excited Periodic Disturbances

If the oscillation is not related to any of the system variables or parameters, or in other words, it is not generated because of internal system linear or nonlinear dynamics, then it is assumed to come from an external source (forced oscillation), and so it will be modeled and called as external periodic disturbance. Since the external disturbance is a periodic signal, which is the case for the considered oscillation, then it can be represented by the sine cosine (*Fourier*) series

$$T_{dis}(t) = \sum_{i=1}^{N_x} \alpha_{xi} \sin(i\omega t) + \beta_{xi} \cos(i\omega t), \quad (2.7)$$

where ω is the disturbance principal angular frequency [rad/s]. The system is also represented by using equation (2.6) as shown in Figure 2.4.

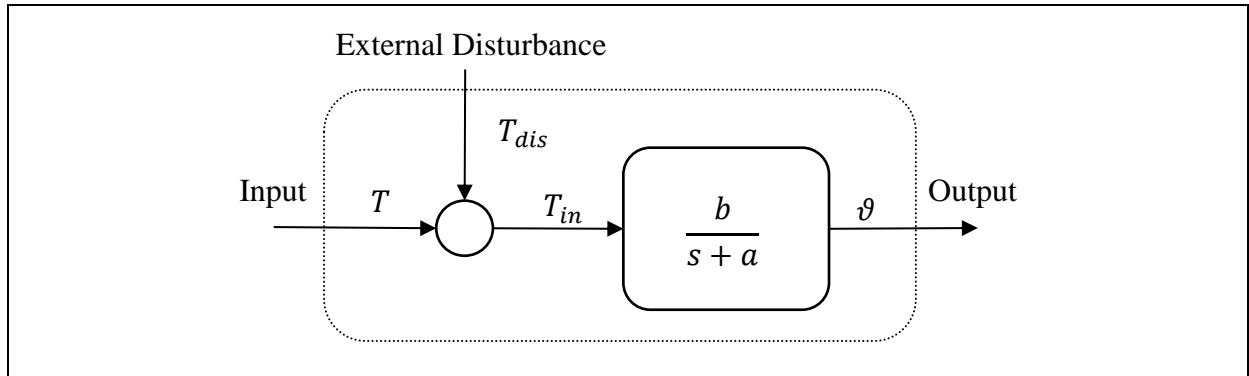


Figure 2.4: Externally disturbed system.

Figure 2.5 shows the frequency response of the linear dynamics for the mathematical parameters $a = 1$ and $b = 1$. Figure 2.6 shows the step response when the external disturbance has 1, 10 or 100 [rad/s] frequency (ω) and defined by

$$T_{dis}(t) = \sin(\omega t). \quad (2.8)$$

It can be seen from the Figure 2.5 and Figure 2.6 that, although the disturbance frequencies 10 and 100 [rad/s] are outside the system bandwidth and they have received a substantial rejection, nevertheless, their effect on the output is still obvious and it can never be small enough in some system applications, where they sometimes demand a complete cancellation. In other words, the external disturbance is only attenuated here, but its effect on the system output depends directly on the external disturbance amplitude. Therefore, in some applications, it demands a perfect disturbance attenuation either through the system dynamics, or a disturbance cancellation by feeding forward an anti-disturbance signal to the system input.

2.2.2 Internally (Self) Excited Periodic Disturbances

In the following subsection 2.2.2.1, a model of stiff rotational drive-load system with angle dependent spring, damper and moment of inertia load elements is introduced, moreover in subsection 2.2.2.2 the physical parameters of the system representing the spring, damper and moment of inertia load elements are transferred into mathematical parameters, while in subsection 2.2.2.3 the angle dependent part parameters are separated from the linear part parameters and transferred as internally (self) excited input periodic disturbance. In addition, the state space model is presented with state (angle) dependent parameters in subsection 2.2.2.4 as well as the state space model of the separated internal input periodic disturbance torque model in subsection 2.2.2.5. Furthermore, an approximation of the angle dependent moment of inertia load element is introduced in subsection 2.2.2.6.

2.2.2.1 Angle dependent spring, damper and moment of inertia load elements

Angle dependent spring (torque) $K(\varphi)$ [Nm], damper $D(\varphi)$ [Nms/rad] and moment of inertia $J(\varphi)$ [kgm²] functions, which represent nonlinear load elements, are defined as following:

The angle dependent spring (torque) function

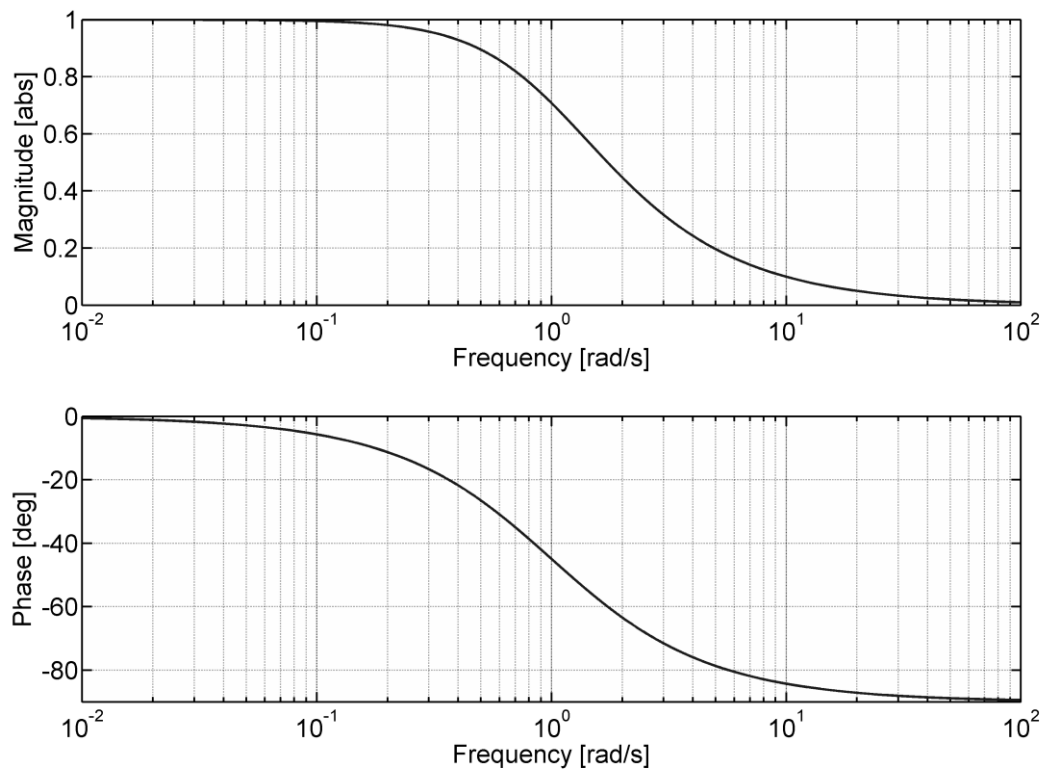


Figure 2.5: Frequency response.

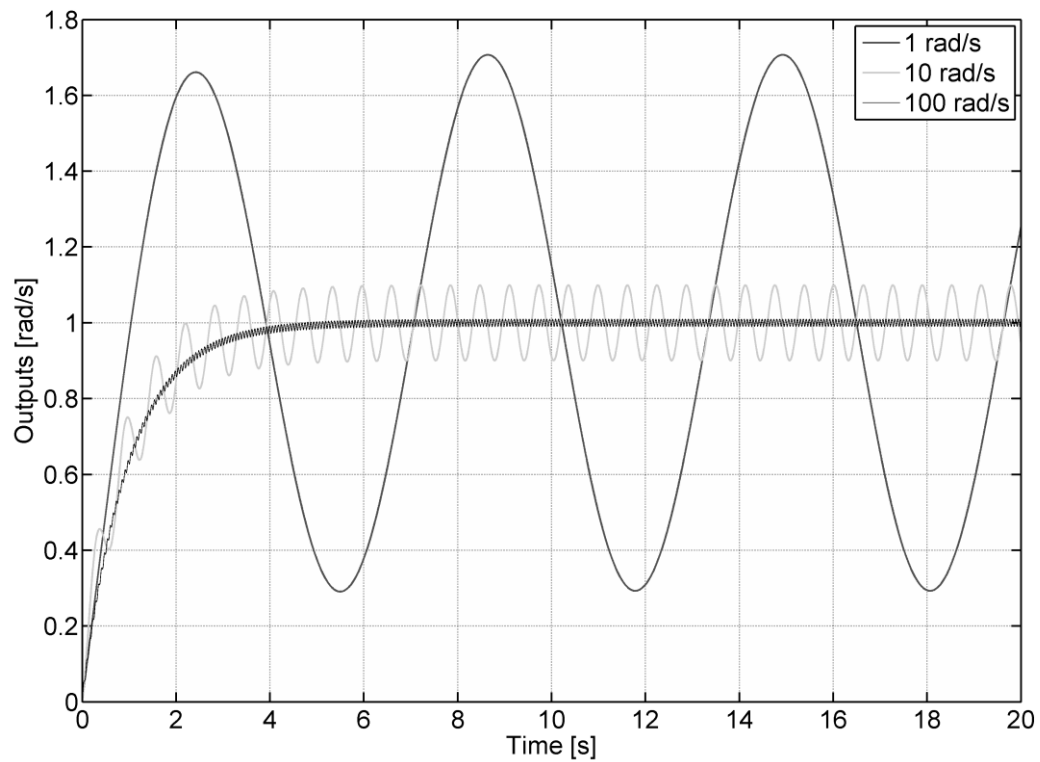


Figure 2.6: Unit step response with different external periodic disturbance frequencies.

$$K(\varphi) = \sum_{i=1}^{N_K} k_i \sin(i\varphi + \phi_{Ki}), \quad (2.9)$$

the angle dependent damper function

$$D(\varphi) = d_0 + \sum_{i=1}^{N_D} d_i \sin(i\varphi + \phi_{Di}), \quad (2.10)$$

the angle dependent moment of inertia function

$$J(\varphi) = j_0 + \sum_{i=1}^{N_J} j_i \sin(i\varphi + \phi_{Ji}), \quad (2.11)$$

and its derivative

$$\frac{dJ(\varphi)}{dt} = \sum_{i=1}^{N_J} i j_i \cos(i\varphi + \phi_{Ji}) \dot{\varphi}(t). \quad (2.12)$$

Therefore, the system differential equation is given by

$$\frac{d}{dt}(J(\varphi)\dot{\varphi}(t)) + D(\varphi)\dot{\varphi}(t) + K(\varphi) = T(t), \quad (2.13)$$

or

$$J(\varphi)\ddot{\varphi}(t) + \frac{dJ(\varphi)}{dt}\dot{\varphi}(t) + D(\varphi)\dot{\varphi}(t) + K(\varphi) = T(t). \quad (2.14)$$

Solving for the acceleration, equation (2.14) becomes

$$\ddot{\varphi}(t) = -\frac{\frac{dJ(\varphi)}{dt} + D(\varphi)}{J(\varphi)}\dot{\varphi}(t) - \frac{K(\varphi)}{J(\varphi)} + \frac{1}{J(\varphi)}T(t). \quad (2.15)$$

Although, these angle dependent rotational load elements are only presented mathematically, they can be practically and simply generated by using a rotational mechanism with eccentricity that translates the rotational motion into a reciprocating linear one connected to linear motion spring, damper and mass load elements, for example, Scotch yoke, cam or crank mechanism, see Figure 2.7. In addition, the analysis and the modeling of a simple Scotch yoke mechanism with spring, damper and mass linear motion load elements are presented in Appendix A, as well as the kinematic analysis of a crank mechanism.

2.2.2.2 Angle dependent mathematical parameter format

The factors of the previous equation (2.15) are defined in terms of the physical parameters, therefore now, these factors are redefined as mathematical parameters ($a(\varphi)$, $\alpha(\varphi)$, $b(\varphi)$ and $c(\varphi)$) and approximated by finite Fourier series expansions as following

$$a(\varphi) = \frac{D(\varphi)}{J(\varphi)} = a_0 + \sum_{i=1}^{N_a} a_i \sin(i\varphi + \phi_{ai}), \quad (2.16)$$

$$\alpha(\varphi) = \frac{\frac{dJ(\varphi)}{dt}}{J(\varphi)\dot{\varphi}(t)} = \frac{\sum_{i=1}^{N_J} i j_i \cos(i\varphi + \phi_{Ji})}{J(\varphi)} = \sum_{i=1}^{N_\alpha} \alpha_i \sin(i\varphi + \phi_{\alpha i}), \quad (2.17)$$

$$b(\varphi) = \frac{1}{J(\varphi)} = b_0 + \sum_{i=1}^{N_b} b_i \sin(i\varphi + \phi_{bi}), \quad (2.18)$$

and

$$c(\varphi) = -\frac{K(\varphi)}{J(\varphi)} = \sum_{i=1}^{N_c} c_i \sin(i\varphi + \phi_{ci}). \quad (2.19)$$

So again, the system differential equation having angle dependent parameters, as presented in Figure 2.8, becomes

$$\ddot{\varphi}(t) = -a(\varphi)\dot{\varphi}(t) - \alpha(\varphi)\dot{\varphi}^2(t) + c(\varphi) + b(\varphi)T(t). \quad (2.20)$$

2.2.2.3 Internal periodic disturbance format

Now, the angle dependent torque variations can be separated and redefined as internal disturbances as following

$$T_{dis0}(\varphi) = \frac{1}{b_0} \left[\sum_{i=1}^{N_c} c_i \sin(i\varphi + \phi_{ci}) \right], \quad (2.21)$$

$$T_{dis1}(\varphi, T) = \frac{1}{b_0} \left[\sum_{i=1}^{N_b} b_i \sin(i\varphi + \phi_{bi}) \right] T(t), \quad (2.22)$$

$$T_{dis2}(\varphi, \dot{\varphi}) = \frac{-1}{b_0} \left[\sum_{i=1}^{N_\alpha} \alpha_i \sin(i\varphi + \phi_{\alpha i}) \right] \dot{\varphi}(t), \quad (2.23)$$

and

$$T_{dis3}(\varphi, \dot{\varphi}) = \frac{-1}{b_0} \left[\sum_{i=1}^{N_\alpha} \alpha_i \sin(i\varphi + \phi_{\alpha i}) \right] \dot{\varphi}^2(t). \quad (2.24)$$

So, the system differential equation becomes

$$\ddot{\varphi}(t) = -a_0\dot{\varphi}(t) + b_0T(t) + b_0T_{dis0}(\varphi) + b_0T_{dis1}(\varphi, T) + b_0T_{dis2}(\varphi, \dot{\varphi}) + b_0T_{dis3}(\varphi, \dot{\varphi}). \quad (2.25)$$

Also, the total internal periodic disturbance can be defined as

$$T_{dis}(\varphi, \dot{\varphi}, T) = T_{dis0}(\varphi) + T_{dis1}(\varphi, T) + T_{dis2}(\varphi, \dot{\varphi}) + T_{dis3}(\varphi, \dot{\varphi}), \quad (2.26)$$

and the system dynamics as

$$\ddot{\varphi}(t) = -a_0\dot{\varphi}(t) + b_0T(t) + b_0T_{dis}(\varphi, \dot{\varphi}, T). \quad (2.27)$$

Moreover, using equations (2.21) to (2.24), new angular dependent mathematical parameters can be defined as following

$$\dot{c}(\varphi) = \frac{1}{b_0} \left[\sum_{i=1}^{N_c} c_i \sin(i\varphi + \phi_{ci}) \right], \quad (2.28)$$

$$\dot{b}(\varphi) = \frac{1}{b_0} \left[\sum_{i=1}^{N_b} b_i \sin(i\varphi + \phi_{bi}) \right], \quad (2.29)$$

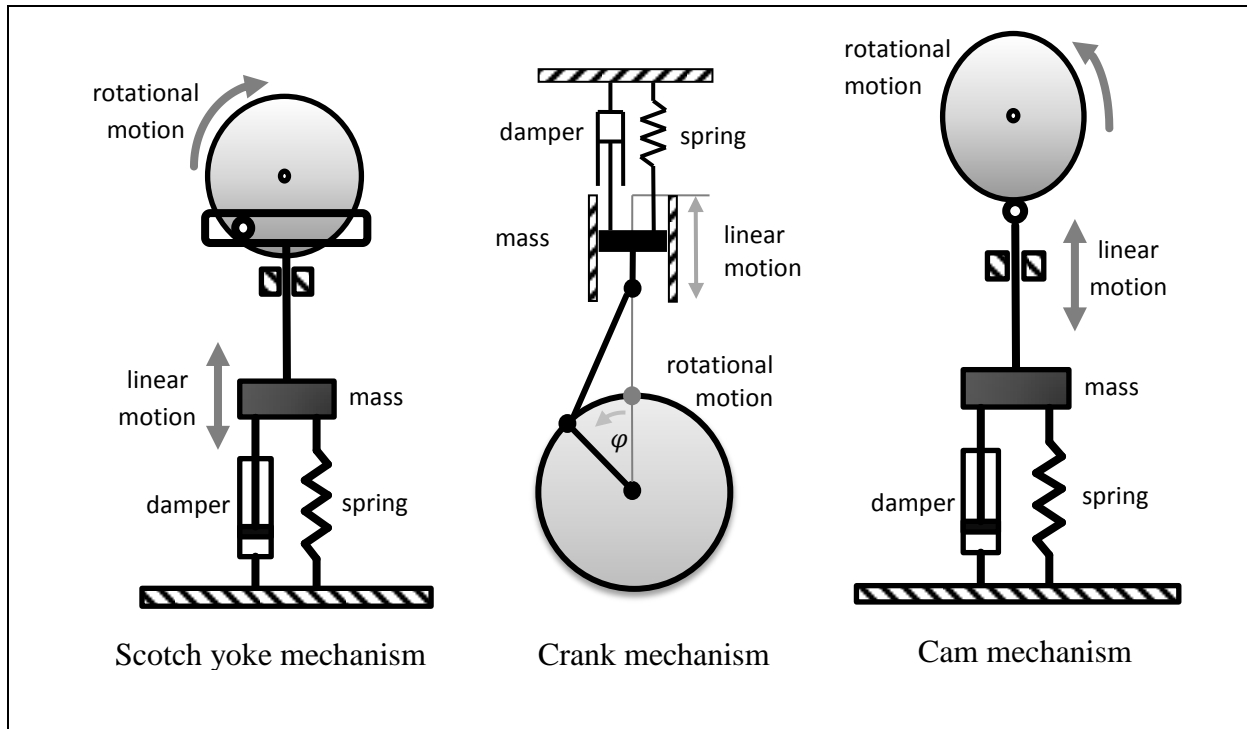


Figure 2.7: Eccentric mechanisms

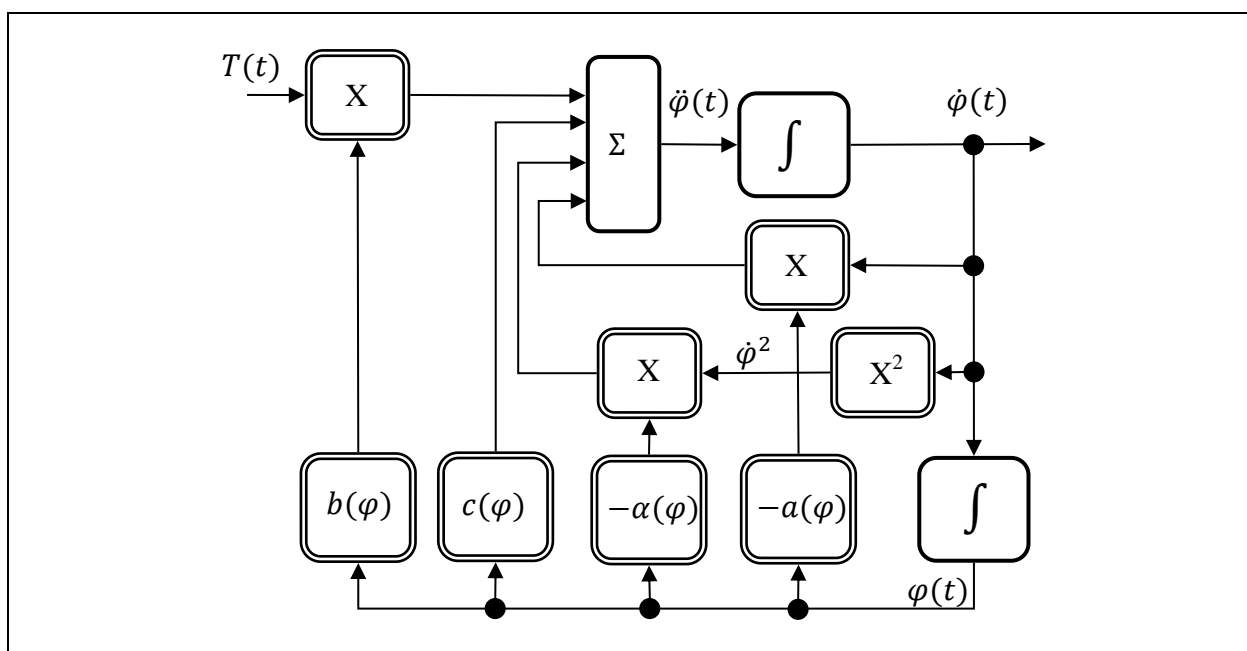


Figure 2.8: Drive-load system with angle dependent mathematical parameters.

$$\dot{a}(\varphi) = \frac{-1}{b_0} \left[\sum_{i=1}^{N_a} a_i \sin(i\varphi + \phi_{ai}) \right], \quad (2.30)$$

$$\dot{\alpha}(\varphi) = \frac{-1}{b_0} \left[\sum_{i=1}^{N_\alpha} \alpha_i \sin(i\varphi + \phi_{\alpha i}) \right], \quad (2.31)$$

so that the separated system can be put in the following form

$$\ddot{\varphi}(t) = -a_0 \dot{\varphi}(t) + b_0 [T(t) + \dot{c}(\varphi) + \dot{b}(\varphi)T(t) + \dot{a}(\varphi)\dot{\varphi}(t) + \dot{\alpha}(\varphi)\dot{\varphi}^2(t)]. \quad (2.32)$$

The system of separated internal periodic disturbance and input-output dynamics, equation (2.32), is depicted in Figure 2.9.

2.2.2.4 Angle dependent parameter format in state space

Using the already defined differential equation (2.20) with mathematical parameters, the system state equations can be defined as following

$$\begin{bmatrix} \dot{\varphi}(t) \\ \dot{\dot{\varphi}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{c(\varphi)}{\varphi} & -[a(\varphi) + \alpha(\varphi)\dot{\varphi}] \end{bmatrix} \begin{bmatrix} \varphi(t) \\ \dot{\varphi}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ b(\varphi) \end{bmatrix} T(t); \quad (2.33)$$

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{x})\mathbf{x}(t) + \mathbf{B}(\mathbf{x})T(t), \\ y(t) &= [0 \ 1]\mathbf{x}(t); \\ y(t) &= \mathbf{C}\mathbf{x}(t). \end{aligned} \quad (2.34)$$

The block diagram describing the system of equations (2.33) and (2.34) is presented in Figure 2.10.

2.2.2.5 State space internal input periodic disturbance torque format

The model of separated input-output dynamics with a state dependent periodic disturbance is defined by splitting the linear dynamics and putting them in the following state equations (2.35) and (2.36), and assuming that the unwanted nonlinear behavior of the state dependent load elements as an internal input disturbance source defined by equation (2.38), as following

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -a_0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ b_0 \end{bmatrix} T_{in}(t), \quad (2.35)$$

$$y(t) = [0 \ 1]\mathbf{x}(t), \quad (2.36)$$

where the total input torque is as defined in equation (2.2) by

$$T_{in}(t) = T(t) + T_{dis}(\varphi, \dot{\varphi}, T), \quad (2.37)$$

and the input format internal angle dependent periodic disturbance is defined by

$$\begin{aligned} T_{dis}(\varphi, \dot{\varphi}, T) &= T_{dis0}(\varphi) + T_{dis1}(\varphi, T) + T_{dis2}(\varphi, \dot{\varphi}) + T_{dis3}(\varphi, \dot{\varphi}) \\ &= \dot{c}(\varphi) + \dot{b}(\varphi)T(t) + \dot{a}(\varphi)\dot{\varphi}(t) + \dot{\alpha}(\varphi)\dot{\varphi}^2(t). \end{aligned} \quad (2.38)$$

The system of equations (2.35), (2.36) and (2.38) is presented in Figure 2.11.

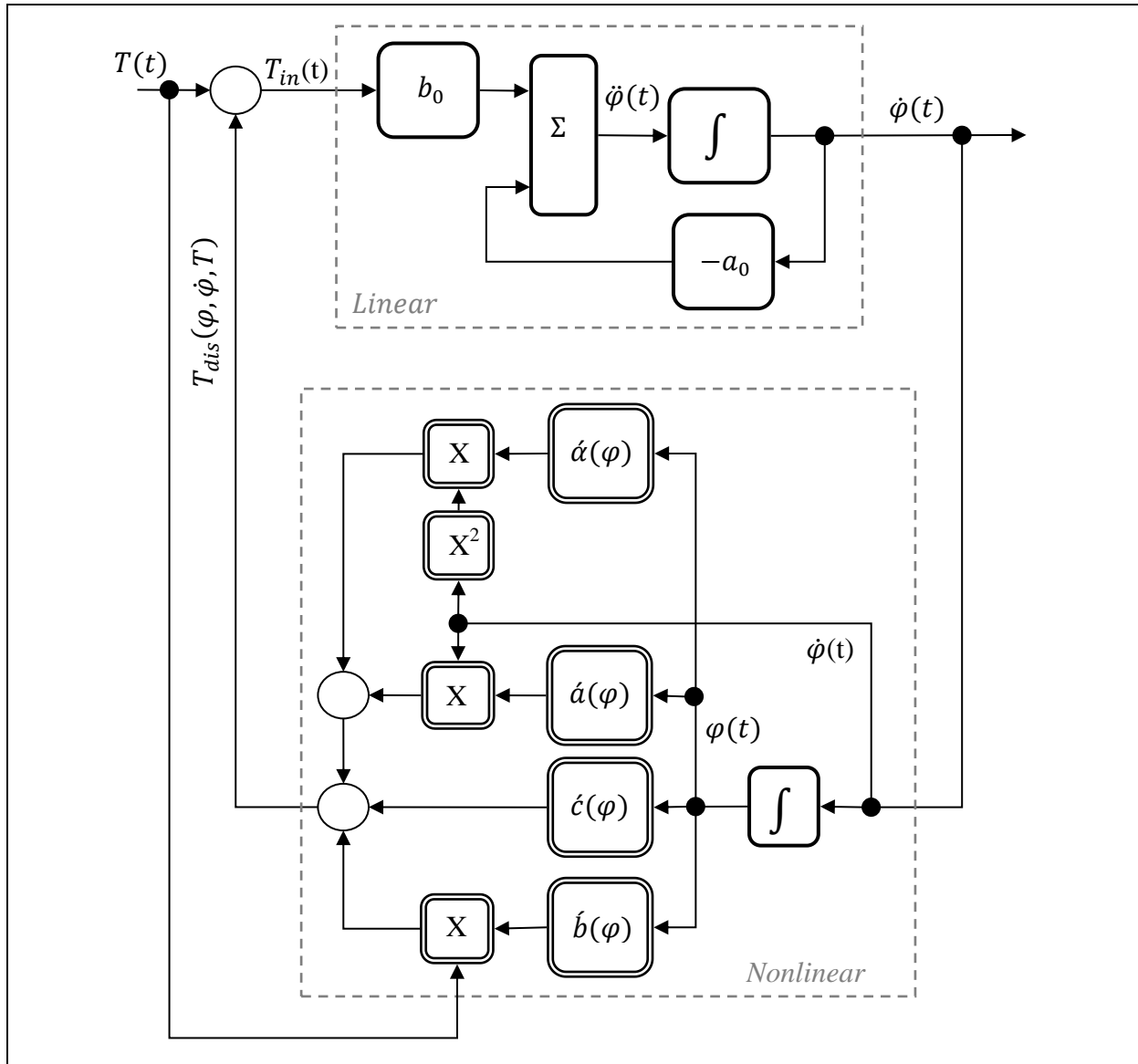


Figure 2.9: Drive-load system with internal periodic disturbances.

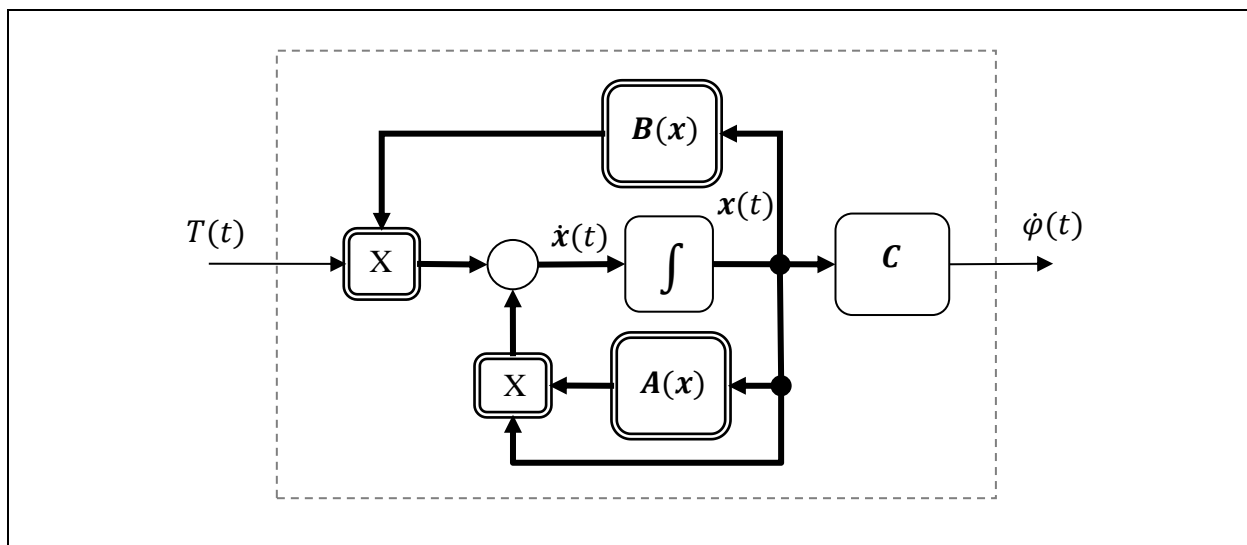


Figure 2.10: Nonlinear state space representation of the drive-load system.

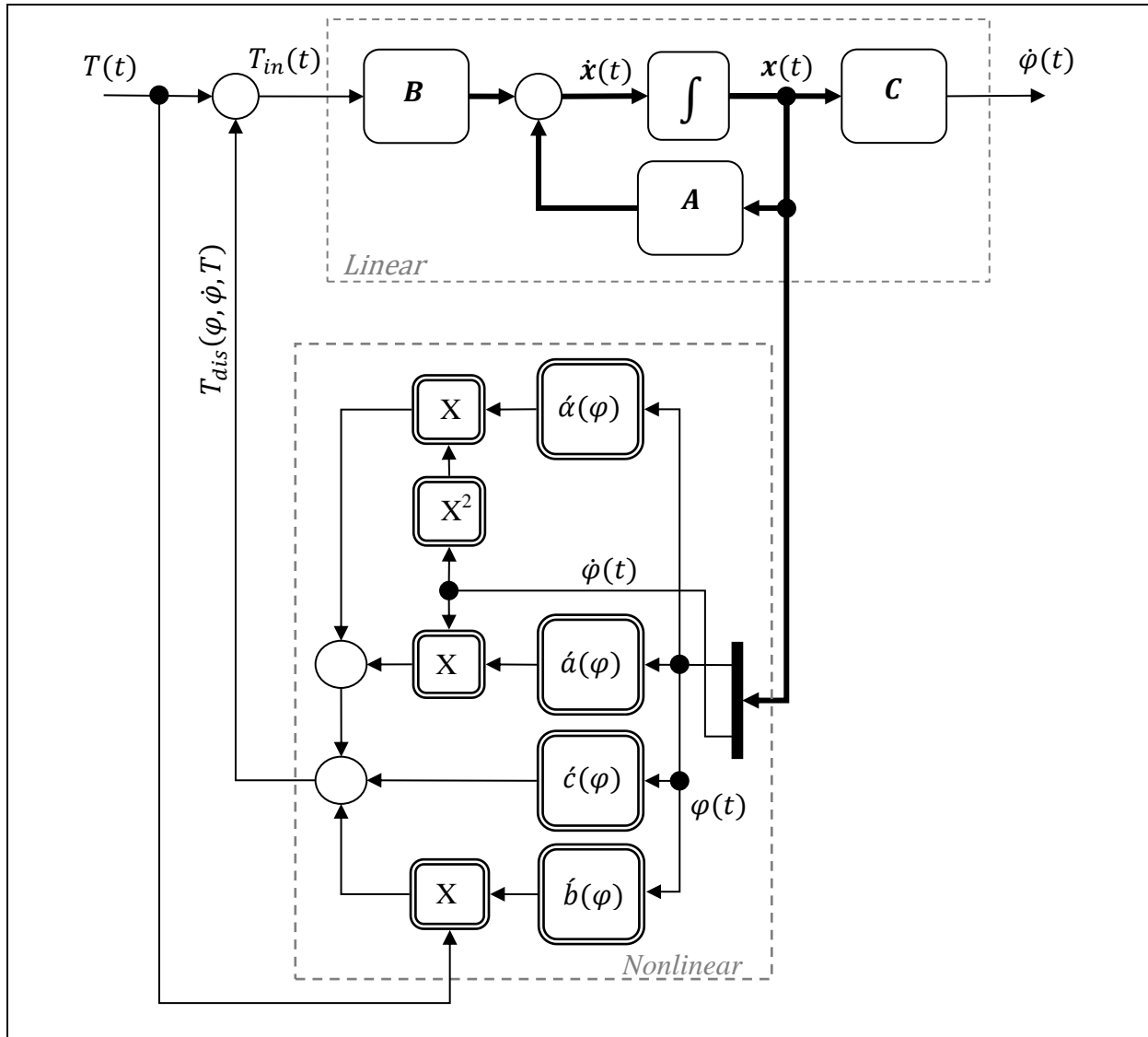


Figure 2.11: Separated linear nonlinear state space representation of the drive-load system.

2.2.2.6 Approximation of the state dependent moment of inertia element

Now, restarting again from equation (2.15) which can be reformulated as

$$\ddot{\varphi}(t) = \frac{1}{J(\varphi)} \left\{ - \left[\frac{dJ(\varphi)}{dt} + D(\varphi) \right] \dot{\varphi}(t) - K(\varphi) + T(t) \right\}. \quad (2.39)$$

Since the factor $(1/J(\varphi))$ is common to all terms in equation (2.39), its variation, especially after the integration, will almost have no effect on the output, and it will happen only in the extreme case when moment of inertia function varies between extremely small minimum and extremely large maximum values, see subsection 2.2.4.3. Therefore, the variation contribution of this factor will be ignored and assumed to be constant as following

$$\frac{1}{J(\varphi)} = \frac{1}{J} = \frac{1}{j_0} = b_0, \quad (2.40)$$

so that input periodic disturbance, equation (2.22), is neglected or put equal to zero

$$T_{dis1}(\varphi, T) = \frac{1}{b_0} \left[\sum_{i=1}^{N_b} b_i \sin(i\varphi + \phi_{bi}) \right] T(t) = 0, \quad (2.41)$$

while

$$\dot{b}(\varphi) = \frac{1}{b_0} \left[\sum_{i=1}^{N_b} b_i \sin(i\varphi + \phi_{bi}) \right] = 0. \quad (2.42)$$

So, the system equation becomes

$$\ddot{\varphi}(t) = -a_0 \dot{\varphi}(t) + b_0 [T(t) + \dot{c}(\varphi) + \dot{a}(\varphi) \dot{\varphi}(t) + \dot{\alpha}(\varphi) \dot{\varphi}(t)], \quad (2.43)$$

or alternatively, as separated input angle dependent disturbance and input-output dynamics

$$T_{dis}(\varphi, \dot{\varphi}) = T_{dis0}(\varphi) + T_{dis2}(\varphi, \dot{\varphi}) + T_{dis3}(\varphi, \dot{\varphi}), \quad (2.44)$$

$$\ddot{\varphi}(t) = -a_0 \dot{\varphi}(t) + b_0 T(t) + b_0 T_{dis}(\varphi, \dot{\varphi}). \quad (2.45)$$

The system of equation (2.43) is depicted in Figure 2.12.

2.2.3 The Disturbance Format of the Rigid Drive-load System

The stiff drive-load system represented by 1DOF system, as presented in previous subsections 2.2.1 and 2.2.2, can generally be put in the model structure form of external and internal input periodic disturbance format system as shown in Figure 2.13. The drive-load input-output dynamics and disturbance equations are given as

$$\dot{\vartheta}(t) = -a\vartheta(t) + b[T(t) + T_{dis}(t)], \quad (2.46)$$

$$T_{dis}(t) = T_{external}(t) + T_{internal}(\varphi). \quad (2.47)$$

Especially, if the approximated version of the inverse of moment of inertia is considered, the externally and internally periodically disturbed system will have the structure of the input disturbance format as shown in Figure 2.13. This model structure, input external and internal periodic disturbance and input-output linear or nonlinear dynamics, is actually the main inspiration in building a model structure of the identification model that will be presented later on in chapter 4 as a general strategy of feedback control for set point tracking and feed-forward control for periodic disturbance compensation.

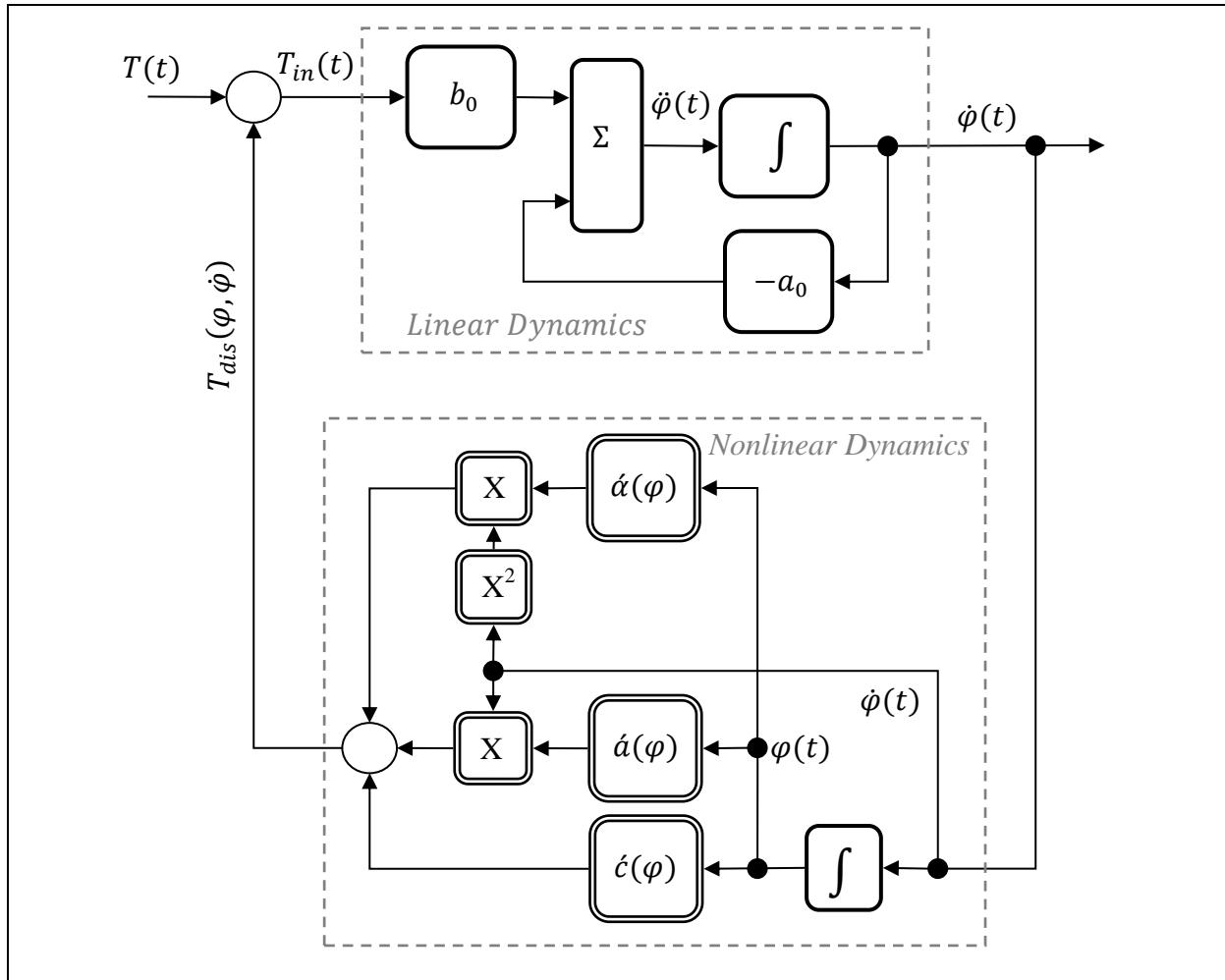


Figure 2.12: Drive-load system with nonlinear spring, damper and approximate moment of inertia load elements.

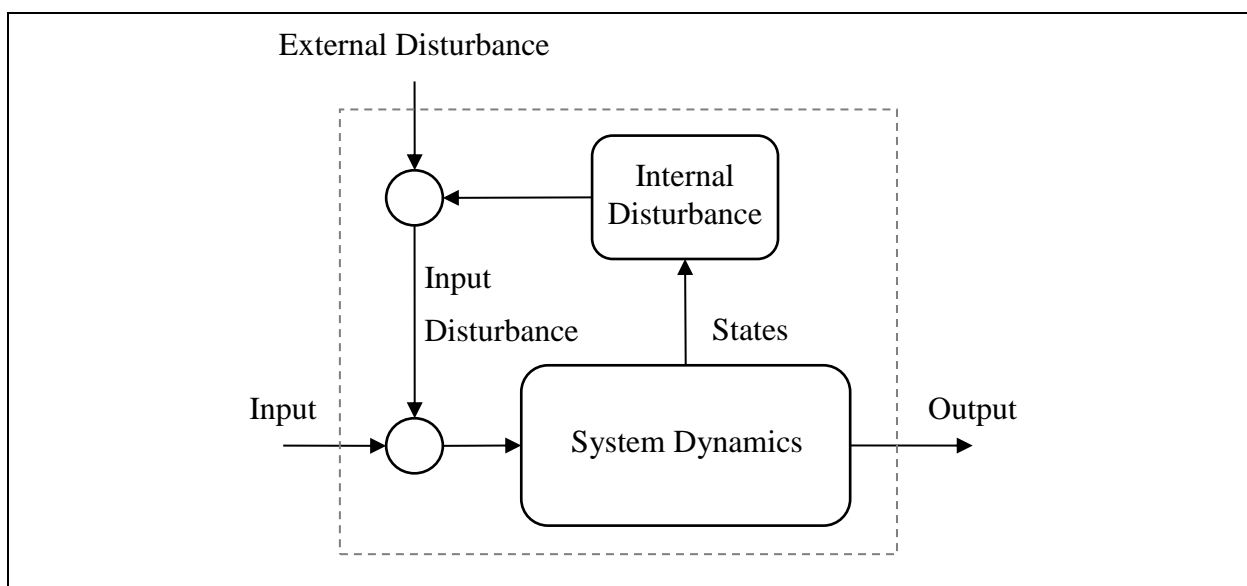


Figure 2.13: Externally and internally disturbed rigid drive-load system.

2.2.4 Simulation Examples

In the following, some (digital) simulation examples are presented in order to demonstrate the oscillation effect on the system output for the case of angle dependent spring load element in subsection 2.2.4.1, for the case of angle dependent damper load element in subsection 2.2.4.2 and for the case of angle dependent moment of inertia load element and its approximation in subsection 2.2.4.3, when the torque input is a stair case function.

2.2.4.1 Angle dependent spring load element

Figure 2.14 shows the system output response and the variable nonlinear angle dependent spring element torque to the stair case torque input, when the system physical parameters are given by

$$\begin{aligned} J &= 1 \text{ [kgm}^2\text{]}; \quad D = 1 \text{ [Nms/rad]}; \\ K(\varphi) &= 0.5 \sin(\varphi) \text{ [Nm]}. \end{aligned} \quad (2.48)$$

From Figure 2.14, it can be seen that, the higher the angular velocity is, the lower the effect of the self-excited periodic disturbance is, and this is because the system has low pass filter dynamics. However, this is like the case of external periodic disturbance in subsection 2.2.1, where the oscillation receives a certain amount of rejection only, and therefore, its effect on the output also depends directly on its amplitude.

2.2.4.2 Angle dependent damper load element

The system stair case step response and the state dependent damper load element are plotted in Figure 2.15, when the system physical parameters are given by

$$\begin{aligned} J &= 1 \text{ [kgm}^2\text{]}; \quad K(\varphi) = 0 \text{ [Nm]}; \\ D(\varphi) &= 1 + 0.5 \sin(\varphi) \text{ [Nms/rad]}. \end{aligned} \quad (2.49)$$

Here, the periodic disturbance effect will linearly increase as the angular velocity increases causing that the effect of the self-excited periodic disturbances to be present even at the higher frequencies outside the bandwidth of the system dynamics.

2.2.4.3 Angle dependent moment of inertia load element and its approximation

The system parameters and state dependent moment of inertia load element are given by

$$\begin{aligned} D &= 1 \text{ [Nms/rad]}; \quad K(\varphi) = 0 \text{ [Nm]}; \\ J(\varphi) &= 1 + 0.2 \sin(\varphi) \text{ [kgm}^2\text{]}. \end{aligned} \quad (2.50)$$

The system stair case step response, the state dependent moment of inertia load element and its constant reciprocal approximation are plotted in Figure 2.16. As already expected, there are almost no differences in oscillation amplitudes, except that there will be a phase difference between them, since the two systems are not synchronized.

Moreover, the effect of the self-excited periodic disturbances, caused by state dependent moment of inertia load element at the higher frequencies outside the system dynamics bandwidth, is much higher than in the case of state dependent damper load element, since its amplitude is nonlinearly related to the squared angular velocity of the system.

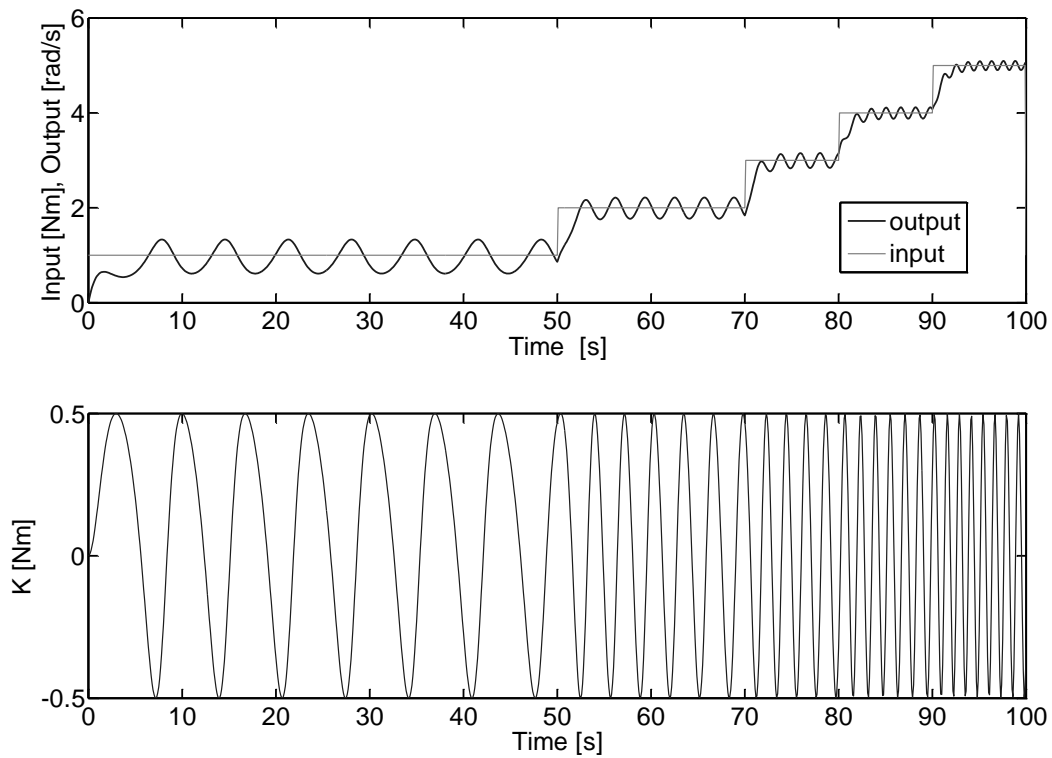


Figure 2.14: Stair case step response and the variable nonlinear spring torque value.

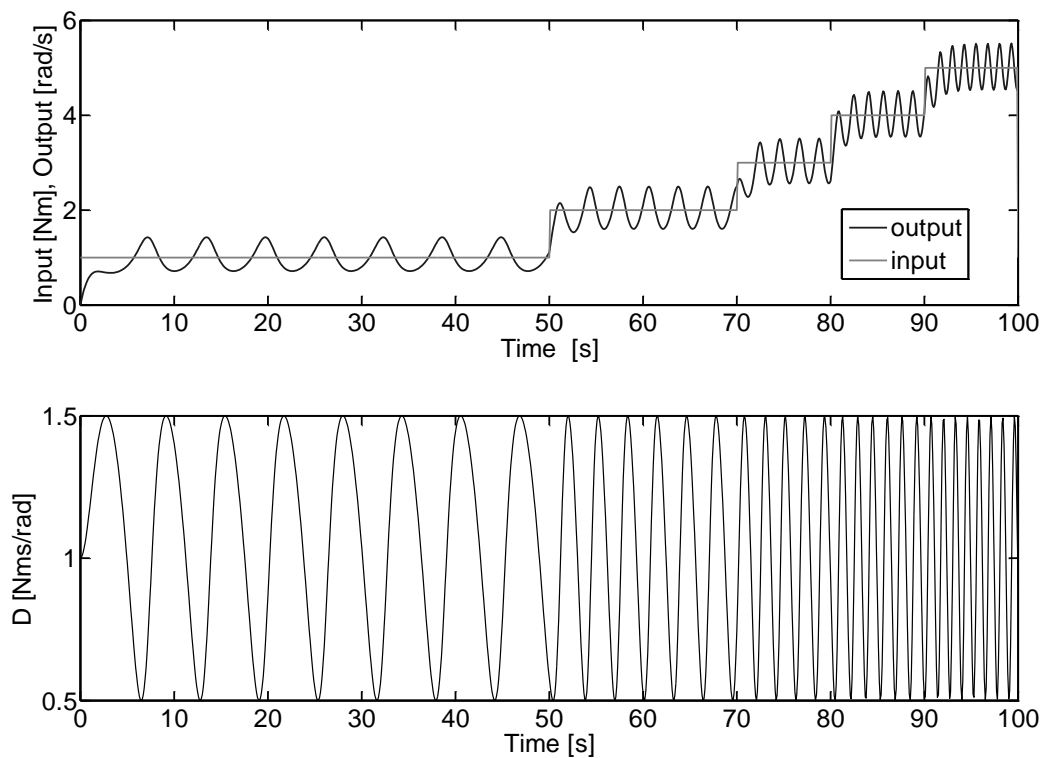


Figure 2.15: Stair case step response and the variable state dependent damper load element.

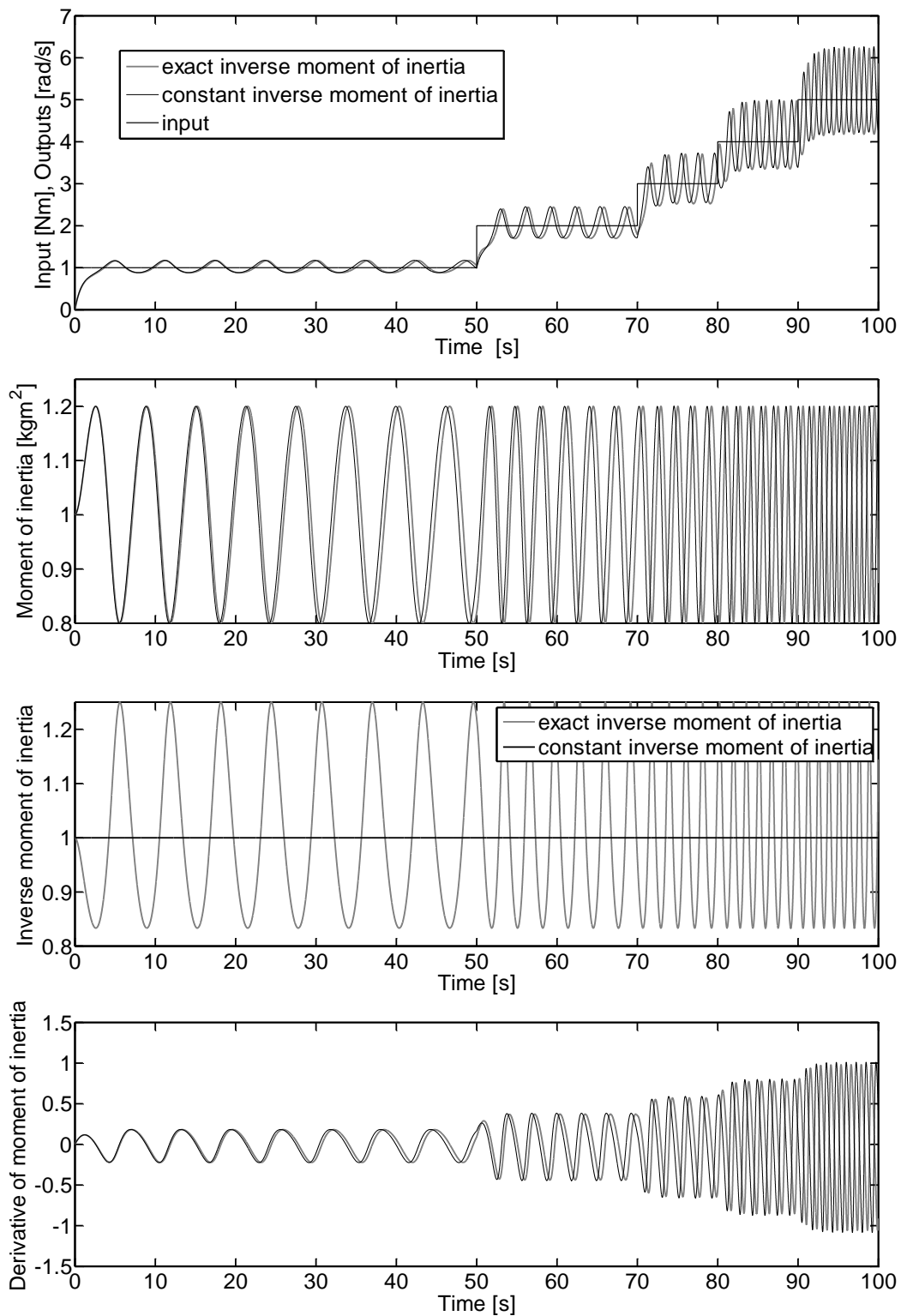


Figure 2.16: Stair case step response and the variable angle dependent moment of inertia for exact $1/J$ and constant approximation of $1/J$ cases.

2.3 Flexible Drive-load System

In this section, the flexibly linked drive-load system is constructed from a drive motor linked to a load through a flexible mechanical link and a gear as shown in Figure 2.17, also there is some disturbance torque acting on the load side. Therefore, the system has two inputs, one is the usual drive (motor) torque input T_D acts upon the drive-side, and the other is the disturbance torque input T_L acts upon the load-side. The mechanical link is modeled by stiffness and damping factors (spring and damper elements). Figure 2.18 shows the physical parameters and the mechanical structure of the rotational drive-load system.

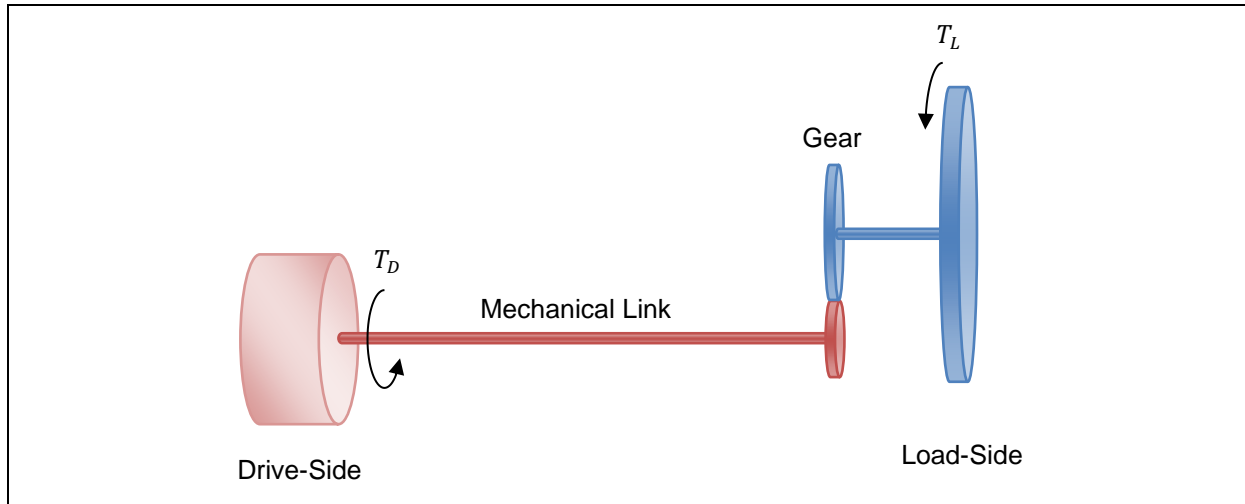


Figure 2.17: The drive-load system with gear.

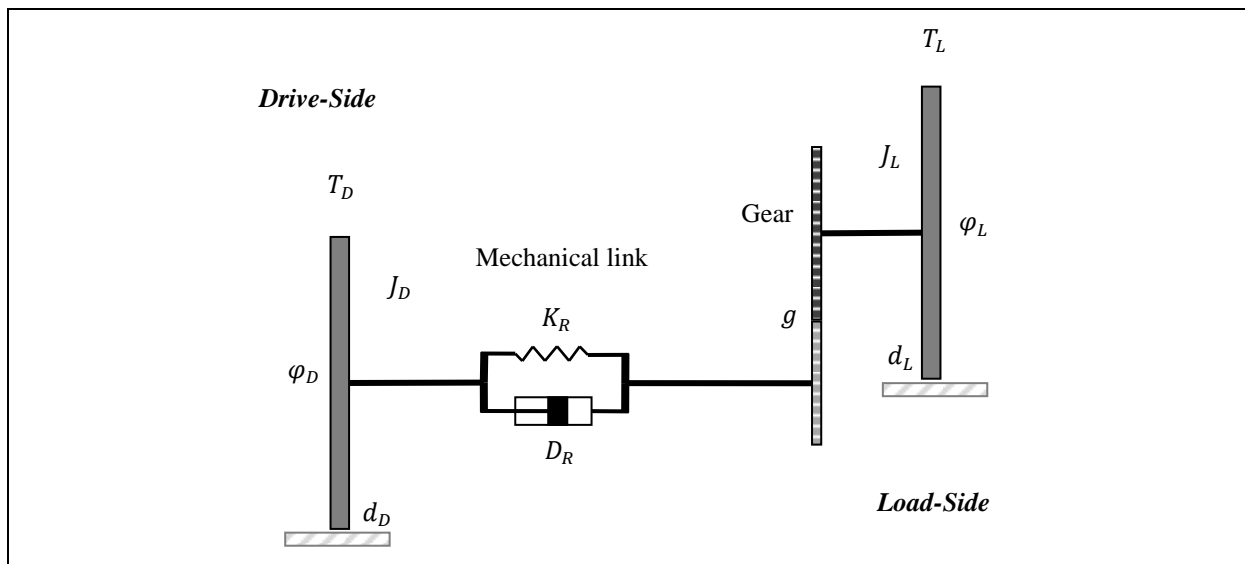


Figure 2.18: The physical parameters of the flexibly linked drive-load system with gear.

Where φ_D & φ_L are the angular positions of the drive and the load sides respectively, T_D is the motor torque acts on the drive-side, T_L is the disturbance (load) torque acts on the load-side, J_D & J_L are the moments of inertia of the drive and load sides respectively, d_D & d_L are the damping factors of the drive and the load sides respectively, K_R & D_R are the spring and the damping factors of the mechanical link between the drive and the load sides. Finally, g is the gear box ratio between the drive and load sides, which is defined by

$$g = \frac{\varphi_D}{\varphi_L} = \frac{\dot{\varphi}_D}{\dot{\varphi}_L} = \frac{T_L}{T_D}. \quad (2.51)$$

The differential equation of the drive-side can be written as

$$J_D \ddot{\varphi}_D(t) + d_D \dot{\varphi}_D(t) + D_R(\dot{\varphi}_D(t) - g\dot{\varphi}_L(t)) + K_R(\varphi_D(t) - g\varphi_L(t)) = T_D(t), \quad (2.52)$$

or after rearrangement

$$J_D \ddot{\varphi}_D(t) + (d_D + D_R) \dot{\varphi}_D(t) + K_R \varphi_D(t) = D_R g \dot{\varphi}_L(t) + K_R g \varphi_L(t) + T_D(t). \quad (2.53)$$

Now, by taking the Laplace transform of the last equation (2.53), the transfer function, between T_D and φ_L as inputs and φ_D as output, can be defined as

$$\varphi_D(s) = \frac{g(D_R s + K_R)\varphi_L(s) + T_D(s)}{J_D s^2 + (d_D + D_R)s + K_R}. \quad (2.54)$$

The differential equation of the load-side can also be written

$$J_L \ddot{\varphi}_L(t) + d_L \dot{\varphi}_L(t) + gD_R(g\dot{\varphi}_L(t) - \dot{\varphi}_D(t)) + gK_R(g\varphi_L(t) - \varphi_D(t)) = T_L(t), \quad (2.55)$$

or after rearrangement

$$\begin{aligned} J_L \ddot{\varphi}_L(t) + d_L \dot{\varphi}_L(t) + g^2 D_R \dot{\varphi}_L(t) + g^2 K_R \varphi_L(t) \\ = gD_R \dot{\varphi}_D(t) + gK_R \varphi_D(t) + T_L(t). \end{aligned} \quad (2.56)$$

Also, by taking the Laplace transform of equation (2.56), the transfer function, between T_L and φ_D as inputs and φ_L as output, can be defined as

$$\varphi_L(s) = \frac{g(D_R s + K_R)\varphi_D(s) + T_L(s)}{J_L s^2 + (d_L + g^2 D_R)s + g^2 K_R}. \quad (2.57)$$

The block diagram of the drive-load system, using equation (2.54) and (2.57), is constructed and presented in Figure 2.19 and in more compact format in Figure 2.20. The overall transfer function between the inputs T_D the drive motor torque and T_L the disturbance torque and the output angular velocity of the load-side $\dot{\varphi}_L$ is given by

$$\begin{aligned} \dot{\varphi}_L(s) = & \quad (2.58) \\ & \frac{[g(D_R s + K_R)]T_D(s) + [J_D s^2 + (d_D + D_R)s + K_R]T_L(s)}{J_D J_L s^3 + [J_D(d_L + g^2 D_R) + J_L(d_D + D_R)]s^2 + [(g^2 J_D + J_L)K_R + d_D d_L + (g^2 d_D + d_L)D_R]s + [(g^2 d_D + d_L)K_R]} \end{aligned}$$

Now, the system mathematical parameters are defined as following

$$\begin{aligned} a_{DL2} &= \frac{[J_D(d_L + g^2 D_R) + J_L(d_D + D_R)]}{J_D J_L}; & a_{DL1} &= \frac{[(g^2 J_D + J_L)K_R + d_D d_L + (g^2 d_D + d_L)D_R]}{J_D J_L}; \\ a_{DL0} &= \frac{[(g^2 d_D + d_L)K_R]}{J_D J_L}; & b_{D1} &= \frac{gD_R}{J_D J_L}; \\ b_{D0} &= \frac{gK_R}{J_D J_L}; & b_{L2} &= \frac{1}{J_L}; \\ b_{L1} &= \frac{(d_D + D_R)}{J_D J_L}; & b_{L0} &= \frac{K_R}{J_D J_L}. \end{aligned} \quad (2.59)$$

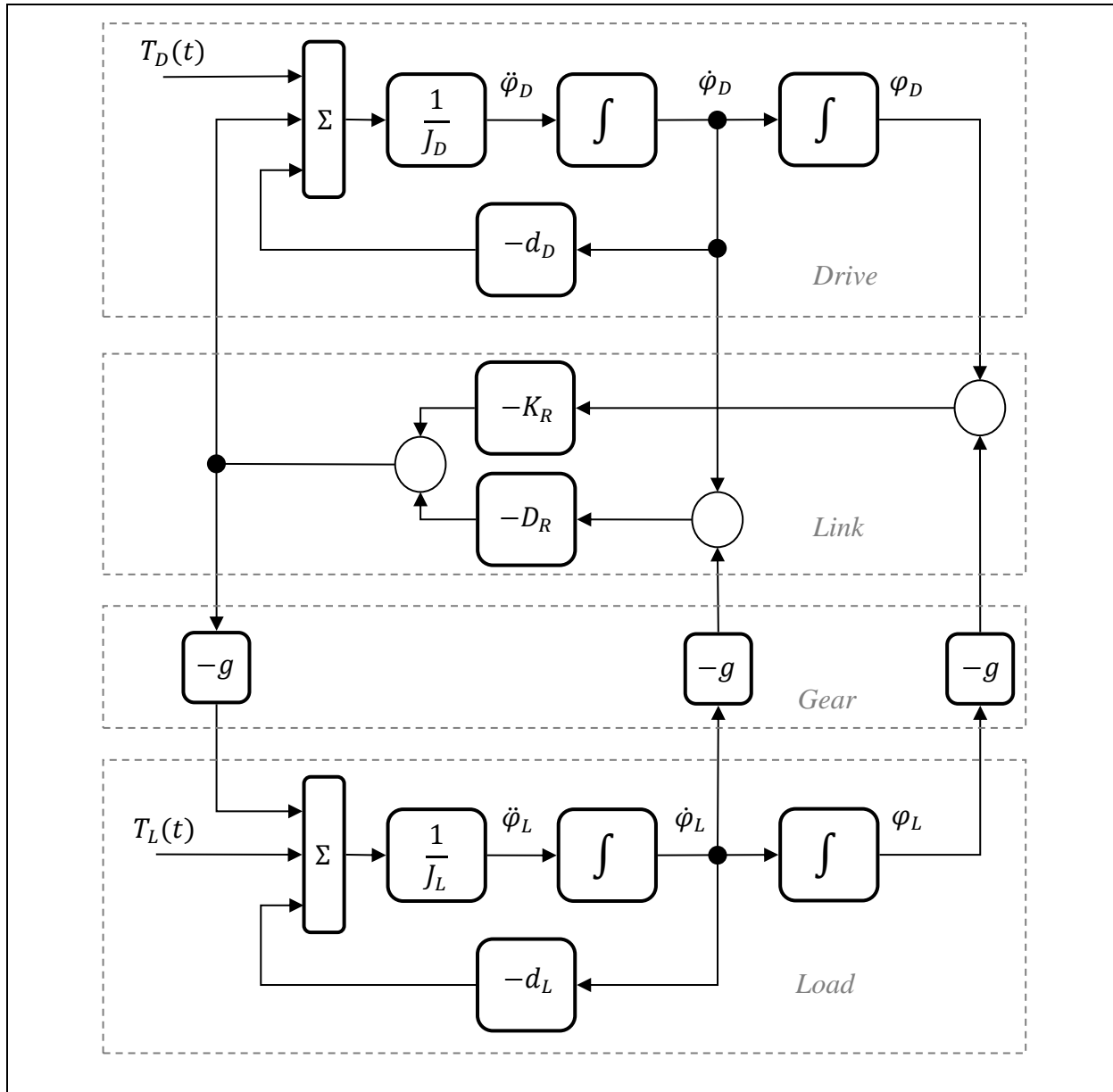


Figure 2.19: Block diagram of the flexibly linked drive-load system.

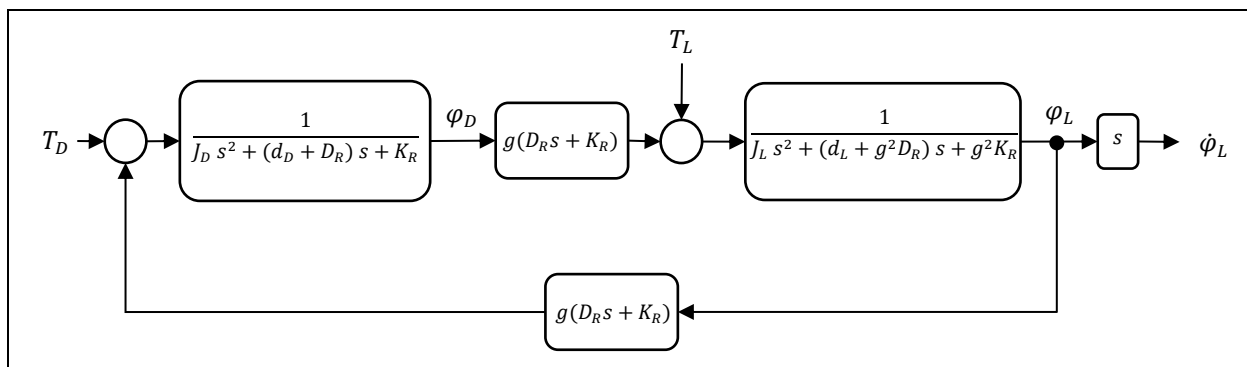


Figure 2.20: Compact block diagram of the flexibly linked drive-load system.

Alternatively, by using the defined mathematical parameters in equation (2.59), the system of equation (2.58) becomes

$$\dot{\phi}_L(s) = \frac{(b_{D1}s + b_{D0})T_D + (b_{L2}s^2 + b_{L1}s + b_{L0})T_L}{s^3 + a_{DL2}s^2 + a_{DL1}s + a_{DL0}}, \quad (2.60)$$

$$\dot{\phi}_L(s) = \frac{b_{D1}s + b_{D0}}{s^3 + a_{DL2}s^2 + a_{DL1}s + a_{DL0}} T_D(s) + \frac{b_{L2}s^2 + b_{L1}s + b_{L0}}{s^3 + a_{DL2}s^2 + a_{DL1}s + a_{DL0}} T_L(s), \quad (2.61)$$

$$\dot{\phi}_L(s) = \frac{P_{ND}(s)}{P_{DL}(s)} T_D(s) + \frac{P_{NL}(s)}{P_{DL}(s)} T_L(s). \quad (2.62)$$

The system of equation (2.62) is depicted in Figure 2.21 by two variants A and B.

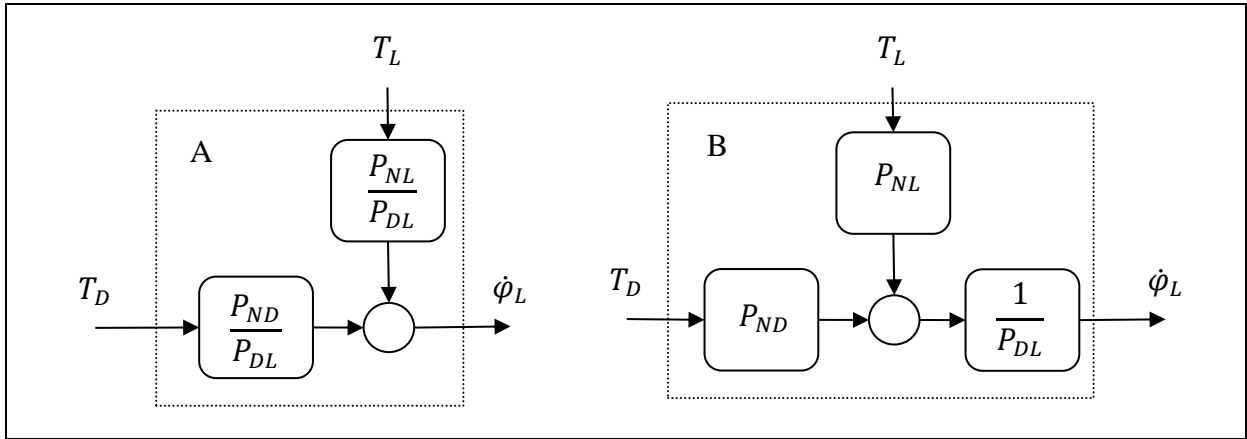


Figure 2.21: Block diagram of the flexible drive-load system.

Numerical Example:

In this example, a comparison is made between the rigid and the flexible systems frequency response. Assume that the flexible system has the physical parameters of the motor side $J_D = 0.1$ [kgm²] and $d_D = 0.1$ [Nms/rad], the mechanical link $K_R = 10^5$ [Nm/rad] and $D_R = 1$ [Nms/rad] with gear ratio $g = 1$ and the load side $J_L = 1$ [kgm²] and $d_L = 1$ [Nms/rad]. So, the system transfer function is given by

$$\frac{\dot{\phi}_L(s)}{T_D(s)} = \frac{10s + 10^6}{s^3 + 13s^2 + 1.1 \cdot 10^6 s^2 + 1.1 \cdot 10^6}, \quad (2.63)$$

with poles at $-6.0 + i1.05$, $-6.0 - i1.05$ and -1.0 .

The system has one dominant pole at -1 and can be approximated by a rigid one which is represented by the physical parameters $J = 1$ [kgm²] and $D = 1$ [Nms/rad], so, its transfer function is given by

$$\frac{\dot{\phi}_2(s)}{T_D(s)} = \frac{1}{s + 1}. \quad (2.64)$$

But this is only valid up to the frequency about 400 [rad/s]. Therefore, the system can be represented (approximated) by a rigid system as long as the targeted operational bandwidth is far below the resonance frequency of the corresponding flexible system.

2.3.1 State Space Model of the Linear Flexible Drive-load System

Starting by defining the angular position difference between the drive and the load sides as

$$\varphi_{\Delta}(t) = (\varphi_D(t) - g\varphi_L(t)), \quad (2.65)$$

the system of differential equations (2.52) and (2.55) can be rewritten as

$$\ddot{\varphi}_D(t) = -\frac{K_R}{J_D}\varphi_{\Delta}(t) - \frac{d_D + D_R}{J_D}\dot{\varphi}_D(t) + \frac{gD_R}{J_D}\dot{\varphi}_L(t) + \frac{1}{J_D}T_D(t), \quad (2.66)$$

$$\ddot{\varphi}_L(t) = \frac{gK_R}{J_L}\varphi_{\Delta}(t) - \frac{d_L + g^2D_R}{J_L}\dot{\varphi}_L(t) + \frac{gD_R}{J_L}\dot{\varphi}_D(t) + \frac{1}{J_L}T_L(t). \quad (2.67)$$

Now, by using the equations (2.66) and (2.67), the state space model of the linear flexible system can be defined, when $\mathbf{x}^T(t) = [\varphi_{\Delta}(t) \ \dot{\varphi}_D(t) \ \dot{\varphi}_L(t)]$, as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{bmatrix} 0 & 1 & -g \\ -K_R/J_D & -(d_D + D_R)/J_D & gD_R/J_D \\ gK_R/J_L & gD_R/J_L & -(d_L + g^2D_R)/J_L \end{bmatrix} \mathbf{x}(t) \\ &+ \begin{bmatrix} 0 & 0 \\ 1/J_D & 0 \\ 0 & 1/J_L \end{bmatrix} \begin{bmatrix} T_D(t) \\ T_L(t) \end{bmatrix}; \\ y(t) &= [0 \ 0 \ 1]\mathbf{x}(t). \end{aligned} \quad (2.68)$$

2.3.2 Externally and Internally Disturbed Flexible Drive-load System

In this subsection, the flexible drive-load system with angle dependent load elements spring, damper and moment of inertia is modeled in subsection 2.3.2.1 as 2DOF system with state dependent parameters. While in subsection 2.3.2.2, it is also modeled as 2DOF system but with a separated external and internal periodic disturbance torque function.

2.3.2.1 Angle dependent parameters format

The modeling of the flexible drive-load system is considered for the simple case when the reciprocal of the angle dependent moment of inertia load element is assumed constant. The angle dependent spring (torque) $K_L(\varphi_L)$, the damper $D_L(\varphi_L)$ and the moment of inertia $J_L(\varphi_L)$ functions of the load-side are defined as following

$$K_L(\varphi_L) = \sum_{i=1}^{N_K} k_i \sin(i\varphi_L + \phi_{Ki}), \quad (2.69)$$

the angle dependent damper function

$$D_L(\varphi_L) = d_{L0} + \sum_{i=1}^{N_D} d_i \sin(i\varphi_L + \phi_{Di}), \quad (2.70)$$

the angle dependent moment of inertia function

$$J_L(\varphi_L) = J_{L0} + \sum_{i=1}^{N_J} j_i \sin(i\varphi_L + \phi_{Ji}), \quad (2.71)$$

and its time derivative

$$\dot{J}_L(\varphi_L) = \frac{dJ_L(\varphi_L)}{dt} = \sum_{i=1}^{N_J} ij_i \cos(i\varphi_L + \phi_{ji}) \dot{\varphi}_L(t). \quad (2.72)$$

The differential equation of the load side can be written as

$$J_L(\varphi_L) \ddot{\varphi}_L(t) + \dot{J}_L(\varphi_L) \dot{\varphi}_L(t) + D_L(\varphi_L) \dot{\varphi}_L(t) + K_L(\varphi_L) + gD_R(g\dot{\varphi}_L(t) - \dot{\varphi}_D(t)) + gK_R(g\varphi_L(t) - \varphi_D(t)) = T_L(t), \quad (2.73)$$

or alternatively as

$$\ddot{\varphi}_L(t) = \frac{1}{J_L(\varphi_L)} \{ [-\dot{J}_L(\varphi_L) + D_L(\varphi_L)] \dot{\varphi}_L(t) - K_L(\varphi_L) - gD_R(g\dot{\varphi}_L(t) - \dot{\varphi}_D(t)) - gK_R(g\varphi_L(t) - \varphi_D(t)) + T_L(t) \}. \quad (2.74)$$

Now, substituting the inverse variable moment of inertia function with its average constant value, this will be true as long as the variation is not extreme, as shown in the rigid system case in subsection 2.2.2.6. So equation (2.74) becomes

$$\ddot{\varphi}_L(t) = \frac{1}{J_{L0}} \{ [-\dot{J}_L(\varphi_L) + D_L(\varphi_L)] \dot{\varphi}_L(t) - K_L(\varphi_L) - gD_R(g\dot{\varphi}_L(t) - \dot{\varphi}_D(t)) - gK_R(g\varphi_L(t) - \varphi_D(t)) + T_L(t) \}. \quad (2.75)$$

So, Figure 2.22 shows the block diagram of flexible drive-load system presented using the physical parameters.

Now, by using equations (2.65), (2.66) and (2.75), a state space model with state (angle) dependent parameters can be constructed, when $\mathbf{x}^T(t) = [\varphi_D(t) \ \dot{\varphi}_D(t) \ \varphi_L(t) \ \dot{\varphi}_L(t)]$, as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K_R & -d_D - D_R & gK_R & gD_R \\ J_D & J_D & J_D & J_D \\ 0 & 0 & 0 & 1 \\ gK_R & gD_R & -g^2K_R & -[g^2D_R + \dot{J}_L(\varphi_L) + D_L(\varphi_L)] \\ J_{L0} & J_{L0} & J_{L0} & J_{L0} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 0 \\ \frac{1}{J_D} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_{L0}} \end{bmatrix} \begin{bmatrix} T_D(t) \\ T_L(t) \end{bmatrix}; \quad (2.76)$$

$$y(t) = [0 \ 0 \ 0 \ 1] \mathbf{x}(t),$$

which has the form of state dependent parameters, that can be generally presented, in corresponding to equation (2.77), as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{x}) \mathbf{x}(t) + \mathbf{B} \mathbf{T}(t); \\ y(t) &= \mathbf{C} \mathbf{x}(t). \end{aligned} \quad (2.77)$$

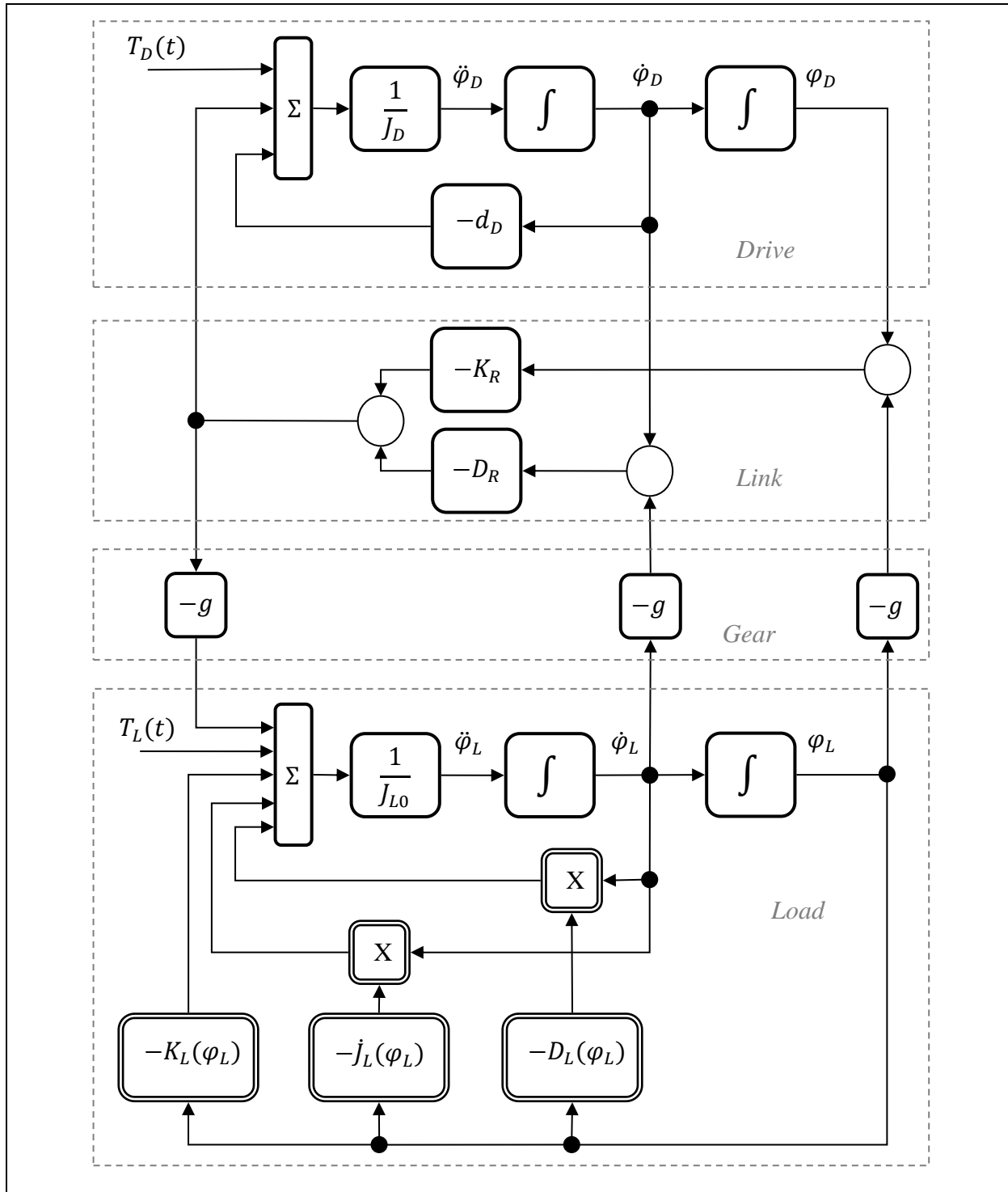


Figure 2.22: Block diagram of the flexibly linked drive-load system with angle dependent physical parameters.

2.3.2.2 Separated periodic disturbance format

In this subsection, the model of the flexible drive-load system is reconstructed in order that it will have a separate periodic disturbance part and the dynamical part. Now, the external and internal periodic disturbance torque functions are introduced as following:

The external disturbance is defined by

$$T_{disx}(t) = \sum_{i=1}^{N_x} \beta_{xi} \sin(i\omega t + \phi_{xi}), \quad (2.78)$$

and the internal disturbance

$$T_{dis2}(\varphi_L, \dot{\varphi}_L) = \dot{J}_L(\varphi_L) \dot{\varphi}_L(t) = - \left[\sum_{i=1}^{N_J} i j_i \sin(i\varphi_L + \phi_{Ji}) \right] \dot{\varphi}_L^2(t) = \alpha(\varphi_L) \dot{\varphi}_L^2(t); \quad (2.79)$$

where

$$\alpha(\varphi_L) = - \left[\sum_{i=1}^{N_J} \alpha_i \sin(i\varphi_L + \phi_{Ji}) \right],$$

$$T_{dis1}(\varphi_L, \dot{\varphi}_L) = - \left[\sum_{i=1}^{N_D} d_i \sin(i\varphi_L + \phi_{Di}) \right] \dot{\varphi}_L(t) = d(\varphi_L) \dot{\varphi}_L(t);$$

where

$$d(\varphi_L) = - \left[\sum_{i=1}^{N_D} d_i \sin(i\varphi_L + \phi_{Di}) \right], \quad (2.80)$$

and

$$T_{dis0}(\varphi_L) = k(\varphi_L) = - \left[\sum_{i=1}^{N_K} k_i \sin(i\varphi_L + \phi_{Ki}) \right]. \quad (2.81)$$

Now, using the defined external and internal periodic disturbance torques, the load side differential equation can be rewritten as

$$J_{L0} \ddot{\varphi}_L(t) = -gD_R (g\dot{\varphi}_L(t) - \dot{\varphi}_D(t)) - gK_R (g\varphi_L(t) - \varphi_D(t)) - d_{L0} \dot{\varphi}_L(t) + T_{disx}(t) + T_{dis0}(\varphi_L) + T_{dis1}(\varphi_L, \dot{\varphi}_L) + T_{dis2}(\varphi_L, \dot{\varphi}_L). \quad (2.82)$$

The detailed block diagram of the separated external and internal periodic disturbance drive-load system is presented in Figure 2.23. The system can also be put in a state space model, when $\mathbf{x}^T(t) = [\varphi_D(t) \ \dot{\varphi}_D(t) \ \varphi_L(t) \ \dot{\varphi}_L(t)]$, as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K_R & -d_D - D_R & gK_R & gD_R \\ J_D & J_D & J_D & J_D \\ 0 & 0 & 0 & 1 \\ gK_R & gD_R & -g^2K_R & -g^2D_R - d_{L0} \\ J_{L0} & J_{L0} & J_{L0} & J_{L0} \end{bmatrix} \begin{bmatrix} \varphi_D \\ \dot{\varphi}_D \\ \varphi_L \\ \dot{\varphi}_L \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{J_D} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_{L0}} \end{bmatrix} \begin{bmatrix} T_D \\ T_L \end{bmatrix}; \quad (2.83)$$

$$y(t) = [0 \ 0 \ 0 \ 1] \mathbf{x}(t),$$

where the external and the internal periodic disturbance torque function is defined by

$$f(\mathbf{x}) = T_L(t, \varphi_L, \dot{\varphi}_L) = T_{disx}(t) + T_{dis0}(\varphi_L) + T_{dis1}(\varphi_L, \dot{\varphi}_L) + T_{dis2}(\varphi_L, \dot{\varphi}_L). \quad (2.84)$$

The system compact transfer function block diagram and state space models are also presented in Figure 2.24 and Figure 2.25 respectively.

However, in order to reject the internal and the external periodic disturbances acting up on the load side of the flexible drive-load system and without using extra drive acting directly on the load side, then these external and internal periodic disturbances are needed to be transferred from the load side to the drive side so that this transformed input periodic disturbance signal can be directly used to compensate those disturbances of the load side. More details will be discussed in chapters 3 and 4.

2.4 The Decision of How Many Degrees of Freedom

Since real systems could have practically any number of degrees of freedom, the question arises, when and how many of these should be taken into account. The first aspect is when the physical parameter distribution of a system looks exactly like the considered drive-load system where it has two mass concentrations represented by two inertial masses (lumped parameter system). If the driving torque of the system is slow enough or alternatively the mechanical link between the drive and the load is stiff enough so that under the normal operation there will be almost no position difference between these masses ever, or considered unnoticeable. Then, the system can be very well represented by one-mass system or in other words by one degree of freedom system. On the other hand, either if the driving torque is quick enough or working at high frequency, or alternatively, the mechanical link is so flexible that there is always a noticeable difference between the two masses so that it cannot be ignored. Then the system is represented by a flexibly linked two-mass system with two degrees of freedom.

However, in real drive-load systems with explicit two-mass concentrations, practically, the system could pass over many resonances, according to the driving frequency, then the system is modeled mathematically between the input and the output, because the modeling analogy has been lost between the mathematical model and the physical one. Therefore, the answer of this question is simply, it depends on the interested operating frequency range. For example, if the interested frequency range is operating at low frequencies where there is no resonance region then a 1DOF is chosen, if the operating frequency goes near to the first resonance frequency then a 2DOF is chosen, and so on for higher number of DOF dynamics.

Moreover, in general, for real physical systems even if there is no resonance, the dynamics sometimes are accompanied with large dead time, then the system will have or should be represented by higher order dynamics and not only with first order one. Finally, the highest control performance level is directly limited by the characteristics of every component/device in the control loop, the real-time control computer (hardware-software) and the process with its sensors and actuators.

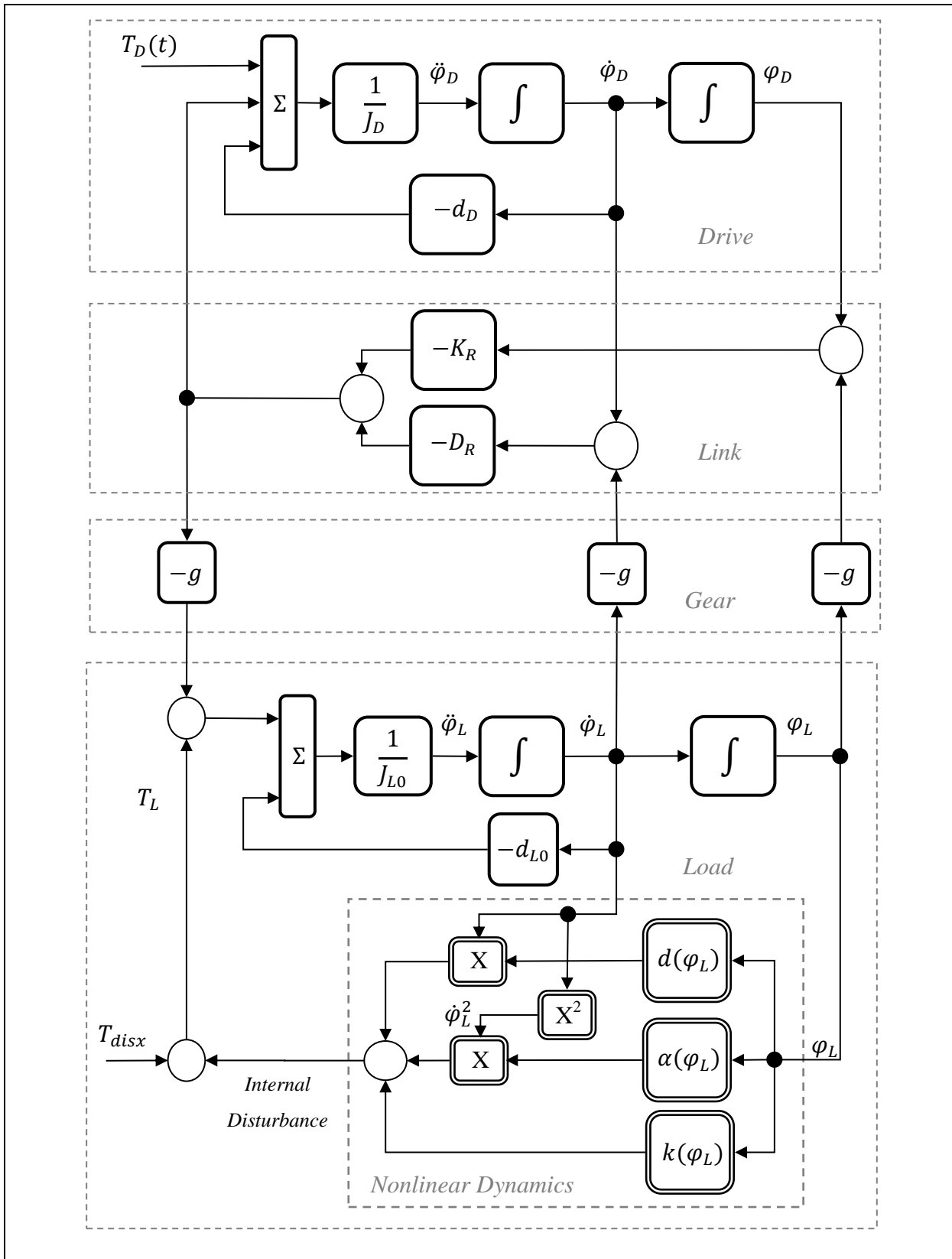


Figure 2.23: Flexible drive-load system with a separate external and internal periodic disturbance torque function.

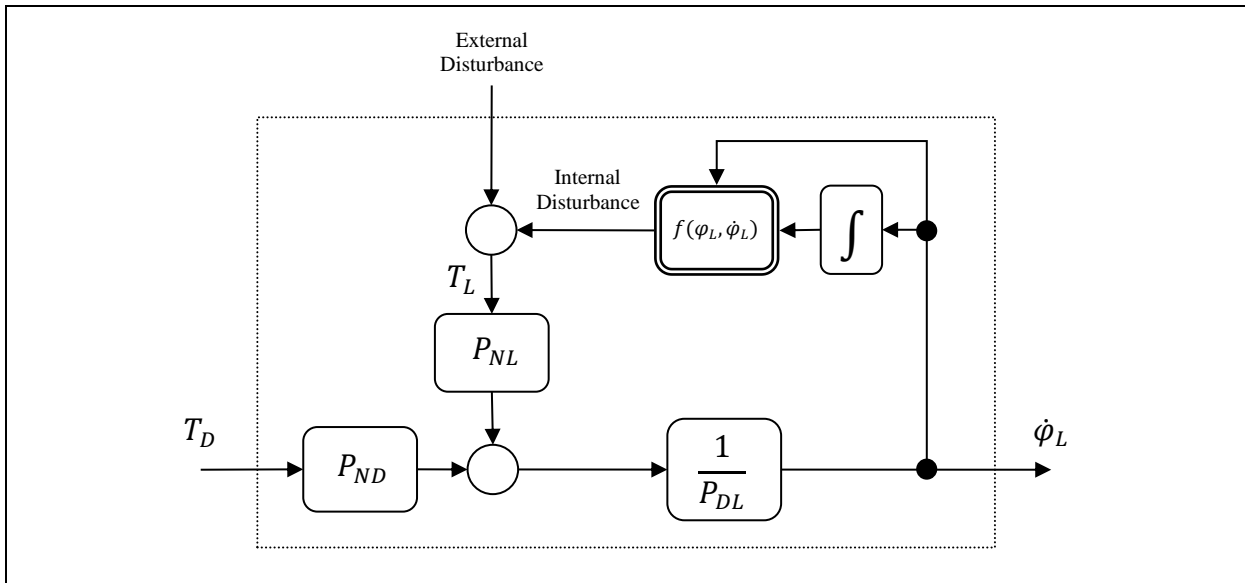


Figure 2.24: Block diagram of externally and internally disturbed flexible drive-load system.

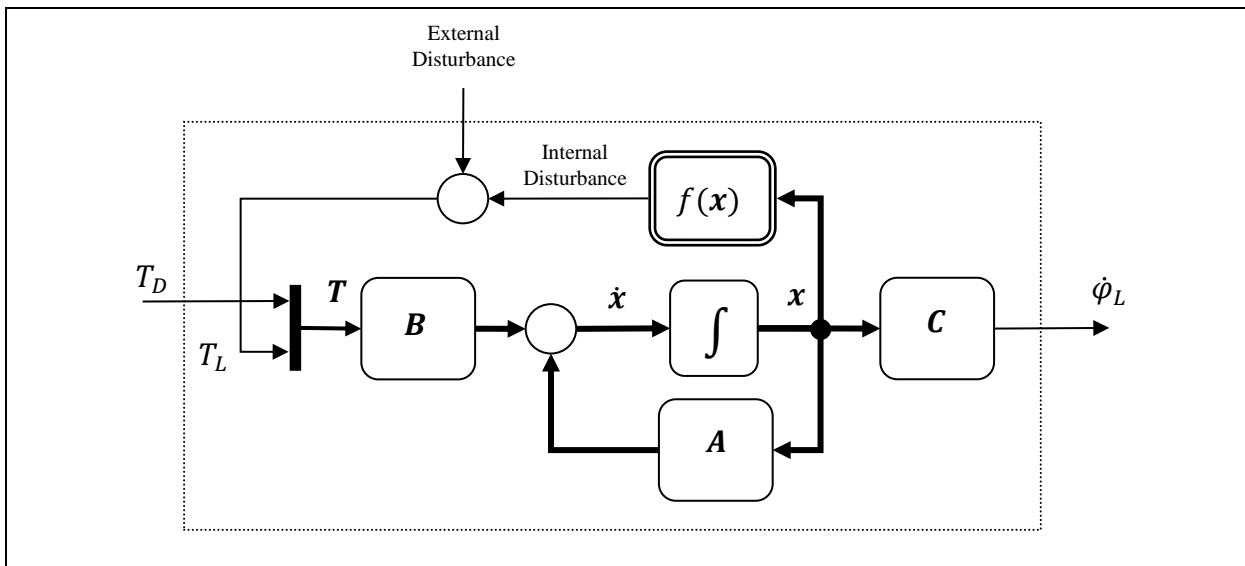


Figure 2.25: Externally and internally disturbed flexible drive-load system in state space representation.

3 PERIODIC DISTURBANCE REJECTION

In this chapter, a brief introduction to the disturbance rejection methods is given in general and to the periodic disturbance rejection methods in particular. Therefore in the following, methods of using only feedback technique to achieve both set point tracking and periodic disturbance rejection are presented. Also, the interaction problem between the set point design demands and the periodic disturbance rejection is discussed. Then, the interaction is minimized by using a separate feed-forward controller to reject the periodic disturbances as an add-on compensator to the pre-existing set point tracking feedback controller. Furthermore, the idea of the disturbance observer is introduced in general. Moreover, the analysis of a feed-forward periodic disturbance controller is presented to the rigid and flexible drive-load system.

3.1 Feedback Disturbance Rejection

Set point tracking and disturbance rejection are the main objectives of the classical and the modern control system engineering, where both feedback as well as feed-forward can be used to achieve the design objectives of the set point tracking and the disturbance rejection. But, using feedback control only to achieve these objectives could lead in some circumstances to an interaction between the design demands, for example, the multi-objective design problem described in (Pipeleers, Demeulenaere and Swevers 2009, subsection 2.3.3), which results in a compromising solution between the set point tracking demands and disturbance rejection. Therefore, the problem must be separated or separate solutions should be made to this controversy. However, in this section, the disturbance is tried to be rejected by using only the feedback control with set point tracking as a primary goal to achieve. Now, the feedback control system structure, pictured in the following Figure 3.1, is assumed. The disturbed process is put under feedback control.

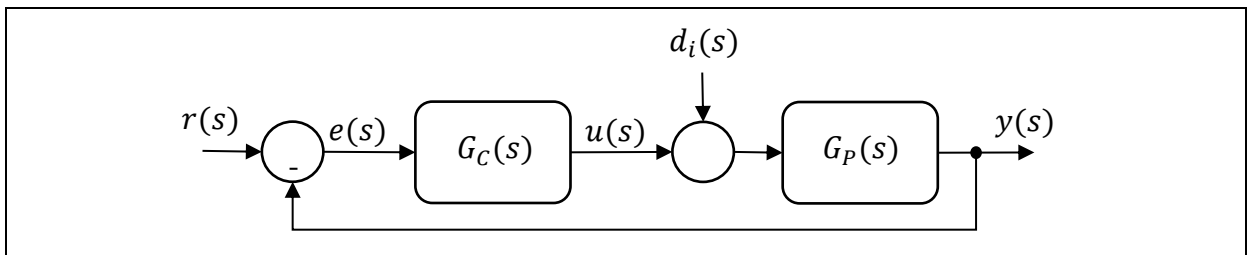


Figure 3.1: Feedback control of disturbed process.

The closed loop frequency response of the set point and the disturbance, for the system shown in Figure 3.1, are given below respectively

$$\frac{y(s)}{r(s)} = \frac{G_P(s)G_C(s)}{1 + G_P(s)G_C(s)}, \quad (3.1)$$

$$\frac{y(s)}{d_i(s)} = \frac{G_P(s)}{1 + G_P(s)G_C(s)}. \quad (3.2)$$

From the equations (3.1) and (3.2), it can be seen that the condition for good set point tracking, which is a primary design objective here, as

$$G_P(s)G_C(s) \gg 1, \quad (3.3)$$

to be valid in the desired system bandwidth, which yields

$$\frac{y(s)}{r(s)} \approx 1; \quad \frac{y(s)}{d_i(s)} \approx \frac{1}{G_C(s)}. \quad (3.4)$$

But for a perfect disturbance rejection, especially outside the systems desired bandwidth, another more condition is needed, which is

$$G_C(s) \approx \infty, \quad (3.5)$$

so that the disturbance response becomes

$$\frac{y(s)}{d_i(s)} \approx 0. \quad (3.6)$$

However, for dynamic systems, these conditions can only be hold for specific frequency spectrum (bandwidth), so that from the disturbance rejection perspective, if the disturbance frequency still lies outside the bandwidth then it will get a poor rejection.

3.1.1 Periodic Disturbance Rejection Filter

Now, the best condition for a perfect disturbance rejection is checked in the frequency domain. The second condition (of course after the validation of the first one) actually reveals how to be done. For example, if the disturbance is just a constant in the time domain, which is represented in the frequency domain by a component at frequency zero. This means that, an integral action (pole at frequency zero) must be added to the controller in order to guarantee the second condition to reject the effect of this disturbance on the system output. Therefore, for a periodic disturbance, complex conjugate poles at the disturbance frequency are needed to be added to the controller in order to keep the second condition valid for perfect disturbance rejection, which actually makes a notch in the closed loop frequency response at the disturbance frequency. This Periodic Disturbance Rejection Filter (PDRF), also “inverse” notch filter, is according to the internal model principle (Francis and Wonham 1976; Bodson 2005). However, most of the use of the notch filters in control applications is to shape the system frequency response of the feedback loop. They are usually implemented in a feedback control systems to suppress the resonance modes of flexible structure characteristics, which are actually the causes of the oscillations (see, for example, Schmidt and Rehm 1999; Yoo, et al. 2011). But now, a PDRF is considered to reject the periodic disturbance that come from an external source. The PDRF can simply be defined, for instance, by an under damped second order transfer function as

$$G_N(s) = \frac{K_d \omega_d^2}{s^2 + 2\xi_d \omega_d s + \omega_d^2}. \quad (3.7)$$

This part is simply added to the original feedback controller as shown in Figure 3.2. For example, if the controller originally has the proportional, integral and the derivative actions then this part is assumed to be as an extension to the integral part to counteract the corresponding periodic disturbance. Figure 3.3 shows the poles of the PDRF when disturbance frequency is at γ [rad/s] and its damping ratio ξ_d is equal to zero. Furthermore, Figure 3.2

shows two variations of adding the PDRF to a preexisting feedback controller. The resulted closed loop transfer function will be for the variant A as

$$y_A(s) = \frac{[G_C(s) - G_N(s)]G_P(s)}{1 + [G_C(s) - G_N(s)]G_P(s)} r(s) + \frac{G_P(s)}{1 + [G_C(s) - G_N(s)]G_P(s)} d_i(s), \quad (3.8)$$

and for the variant B as

$$y_B(s) = \frac{G_C(s)G_P(s)}{1 + [G_C(s) - G_N(s)]G_P(s)} r(s) + \frac{G_P(s)}{1 + [G_C(s) - G_N(s)]G_P(s)} d_i(s). \quad (3.9)$$

Obviously, there is no difference between the variants A and B in the disturbance rejection curves. But there is a difference between the set point response curves. Variant A tries to give a large open loop gain (or even infinite for perfect disturbance rejection) at the disturbance frequency which leads to that the controller amplifies this frequency in the frequency response of the set point. This means, it gives very high open loop gain at this frequency for the set point. Therefore, this variant will be the choice to reject a periodic disturbance when its frequency is inside the system bandwidth. Alternatively, variant B suppresses both the set point and the periodic disturbance by creating a notch in the closed loop set point and the disturbance frequency responses at the periodic disturbance frequency. This is needed when the disturbance frequency is outside the (demanded) system bandwidth. Figure 3.4 shows the variant A closed loop frequency response for the set point and the disturbance, while Figure 3.5 shows the variant B closed loop frequency response. More simulation examples will be presented in chapter 5 and real-time implementation in chapter 6.

Generally, for multi-harmonic disturbances, a number of PDRF can be designed and used to reject every single harmonic distinctively or one PDRF for a band of disturbance frequencies. For the case of infinite harmonics a repetitive controller (Kobayashi, Hara and Tanaka 1990; Steinbuch 2002; Quan and Cai 2010) can be used to reject them.

In conclusion, the perfect condition for disturbance rejection will add some extra lag to the open loop path. Regrettably, this will deteriorate the closed loop set point tracking characteristics in terms of dynamics and stability. It will be even worse if there are multi-harmonics. This will make the design more complex. And at the end, there will be only a compromising solution between system (relative) stability and disturbance rejection design demands.

The design of the PDRF, added to a closed loop negative feedback controller, can actually be carried out by all classical as well as modern control design techniques. For example, if the PDRF is added to a PID controller, then the design could simply be done by tuning the PID as well as the PDRF parameters to get the desired set point tracking and periodic disturbance rejection demands. But this is not going to be an easy task for complex systems.

Alternatively, the design can be done using classical control design techniques, e.g. in frequency domain to get the corresponding phase and gain margins for a specific relative stability, or by using the pole placement technique, to get the desired stable closed loop poles. Moreover, the design can also be carried out by using modern control design techniques, e.g. using state feedback pole placement or optimal control, and furthermore alternatives are the design methods based on the robust control theory to achieve a robust controller.

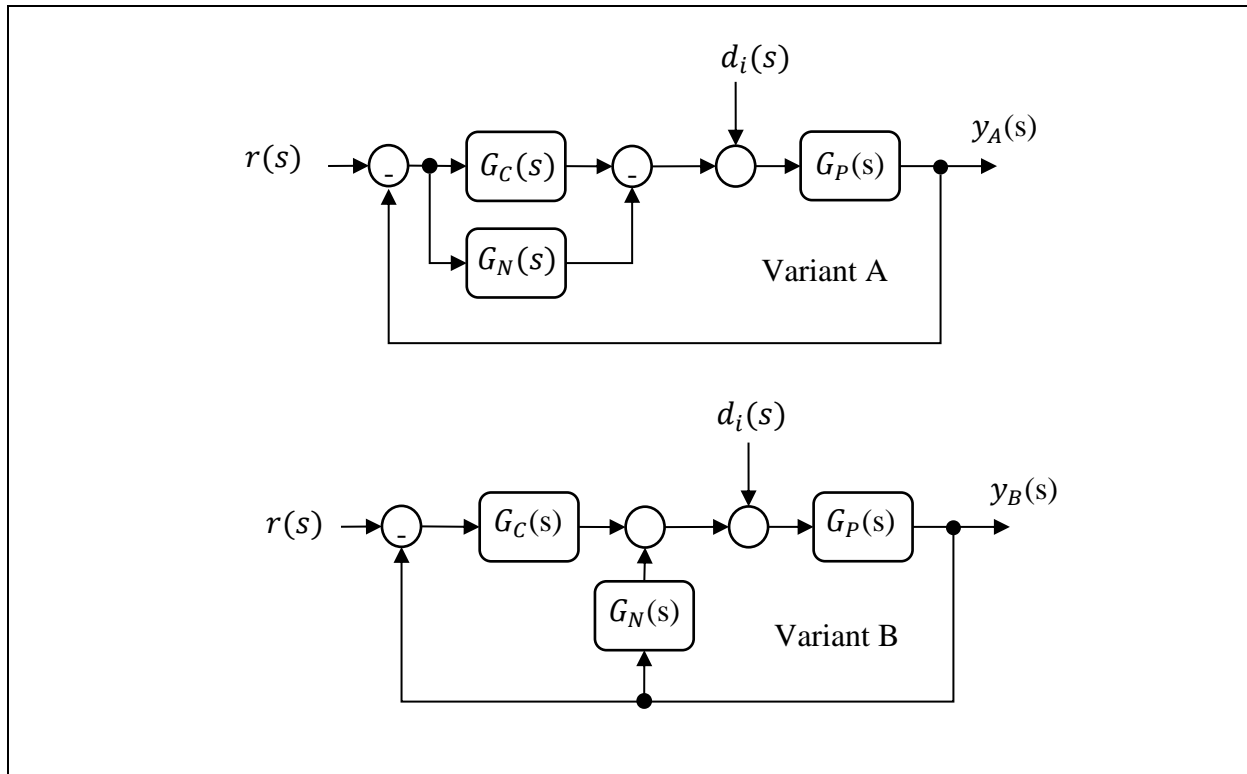
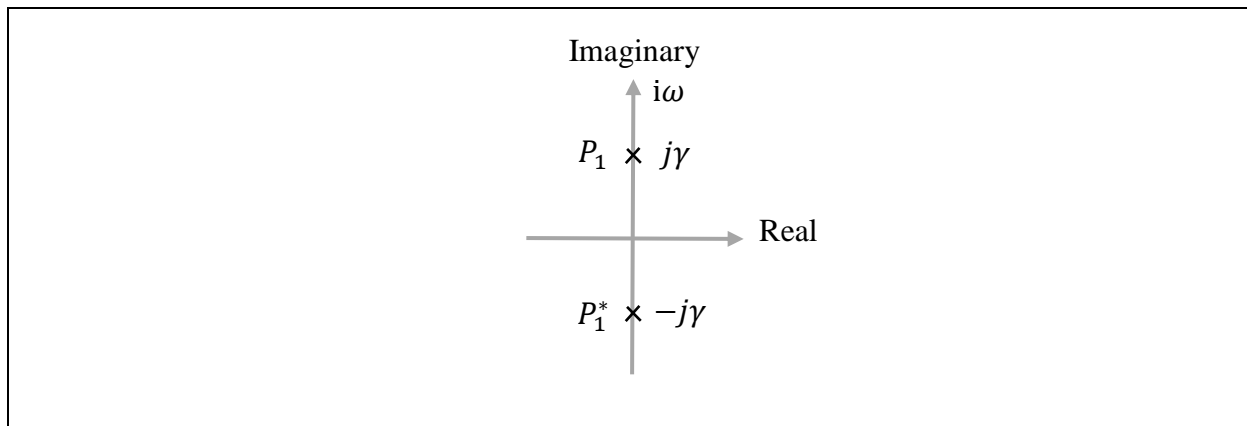


Figure 3.2: Addition of PDRF in two variations A and B.

Figure 3.3: The PDRF poles in s -plane for $\omega_d = \gamma$ and $\xi_d = 0$.

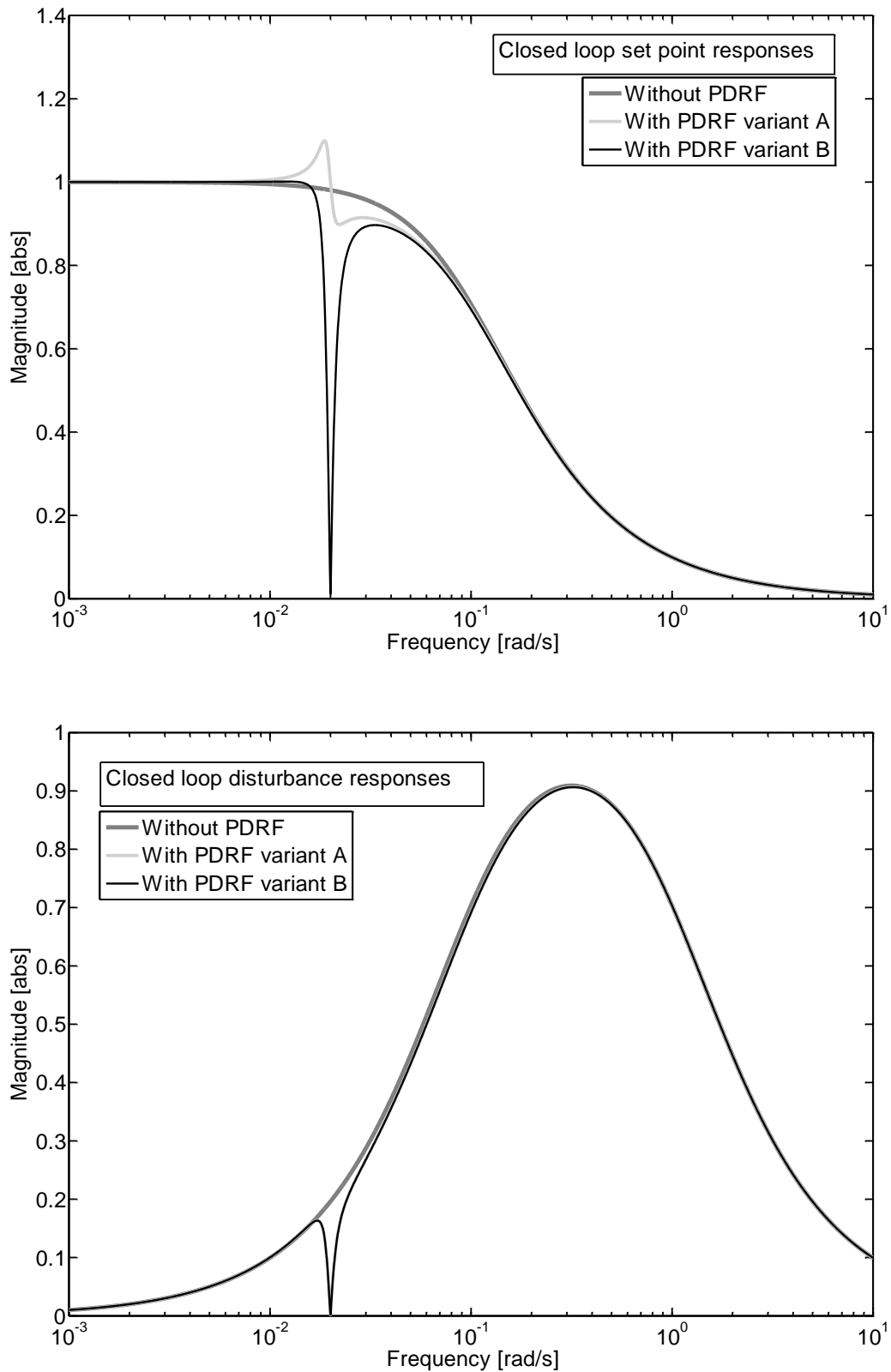


Figure 3.4: Disturbance frequency inside the (demanded) system bandwidth.

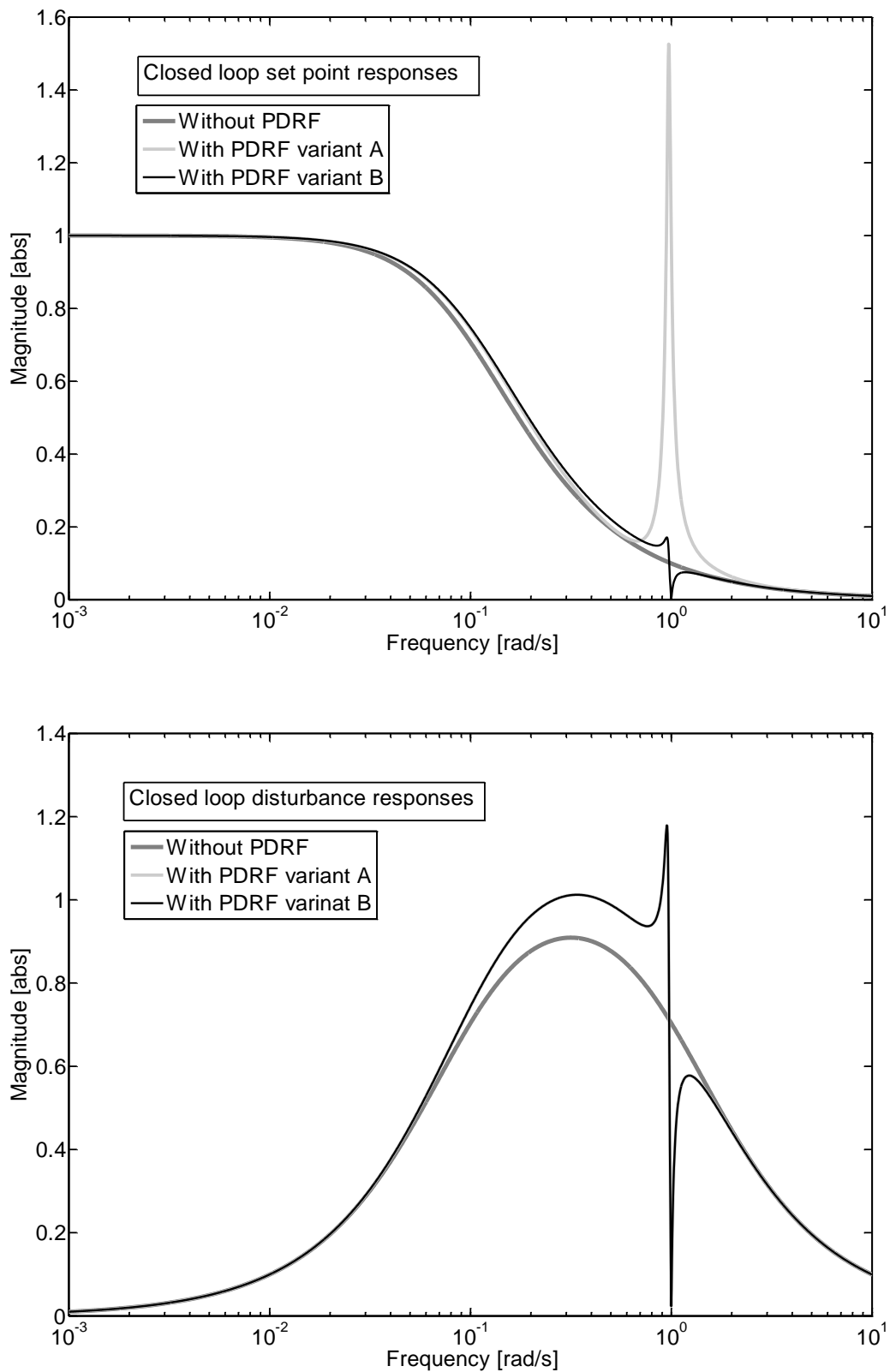


Figure 3.5: Disturbance frequency outside (demanded) system bandwidth.

3.2 Feed-forward Disturbance Compensation

3.2.1 Introduction to the Feed-forward Disturbance Compensation

To achieve feed-forward compensation, both the disturbance signal and the model characteristics of the system should be known, this can be explained by the following. A system, as shown in Figure 3.6.A, is assumed, which is described by functional (operator)

$$y(t) = f(u(t), d(t)), \quad (3.10)$$

where $y(t)$ is the output, $u(t)$ is the (manipulated) input and $d(t)$ is the disturbance input of the system. Furthermore, the disturbance is assumed to be measurable and its effect on the system output is needed to be cancelled by using the manipulated input. A further assumption is to be made that the input signal and the disturbance signal have an independent action on the system output. This means, that the output functional is separable, see Figure 3.6.B1, and can be separated into independent functionals as

$$y(t) = f_u(u(t)) + f_d(d(t)). \quad (3.11)$$

Now, if the disturbance action is to be compensated by the input signal, this makes its effect on the output equal to zero, see Figure 3.6.B2. From this, the input compensation signal (feed-forward control law) can be calculated by

$$u(t) = -f_u^{-1}(f_d(d(t))), \quad (3.12)$$

where $f_u^{-1}(u(t))$ is the inverse of the input functional. Now, if the disturbance action on the output is to be compensated by the input, then the disturbance signal must be available either by direct measurement, observation (if it is not directly measured), estimation (measurements are stochastic) or prediction (if the needed value is in future, especially for systems with large time delay). In addition, the system input functional, its inverse and the disturbance functional should be also available (known). Furthermore, if the input and the disturbance functionals can be represented by linear transfer functions, as shown in Figure 3.6.C1, then

$$y(s) = G_u(s)u(s) + G_d(s)d(s). \quad (3.13)$$

So again, the condition for disturbance compensation by the input, as shown in Figure 3.6.B2, is given by

$$u(s) = -\frac{G_d(s)}{G_u(s)}d(s). \quad (3.14)$$

Apart from the lately discussed conditions, the input transfer function has to have minimum phase (stable) zeros and the ratio of the disturbance transfer function divided by the input transfer function should be causal (denominator degree is higher than the degree of the numerator). Otherwise, the feed-forward control law is unrealizable. Notwithstanding this, an exception can though be made, if the disturbance is a periodic signal, and the feed-forward law is non-causal. Therefore, it will be a matter of finding the right amplitude and phase shift to compensate (cancel out) the respective periodic signal. This will be discussed more in subsection 3.2.4 and chapter 4.

Figure 3.6.D1 shows the output disturbance format and Figure 3.6.E1 shows the input disturbance format. The transformation of the output disturbance into input disturbance is given by

$$d_i(s) = \frac{1}{G_u(s)} d_o(s), \quad (3.15)$$

where d_o is the output disturbance ($d_o(s) = G_d(s)d(s)$), while the transformation of the input disturbance to output disturbance is given by

$$d_o(s) = G_u(s)d_i(s), \quad (3.16)$$

It is clear from the transformation equations that the input to output disturbance transformation is causal, but the output to input disturbance is not, since the reciprocal of causal dynamics yields a non-causal system. For real application, an exception can be made only for periodic disturbances, since the periodic signals are completely predictable in future when their amplitude, phase shift and frequency are constants.

Now, assuming that the structures given in Figure 3.6.A, B1, C1 and D1 can be transformed into the form of the direct input disturbance, see Figure 3.6.E1, by having this signal as a direct measurement, observation, estimation or prediction, then it can be used directly without extra computation, or simply the feed-forward control law becomes as

$$u(t) = -d_i(t). \quad (3.17)$$

So at the end, it is better to get the input disturbance signal even when the real system has another disturbance signal, e.g. output disturbance signal, so that to get the simplest feed-forward control law, as shown in Figure 3.6.E2.

Regrettably, in practice, most of the cases are either the direct measurement, by using extra sensor(s), is impossible, or it is commercially too expensive to realize. Therefore, indirect methods are used, for example, using a disturbance observer (estimator, predictor) to construct the disturbance signal by monitoring the system input and the disturbance effect on the output. However, the implementation of this type of disturbance observer to estimate the disturbance and at the same time to cancel its effect on the output, transforms the principle idea of feed-forward control back into feedback control, but one exception could be made when the process model is perfect. Therefore, this type of feed-forward control is sometimes alternatively called estimated feedback control, pseudo or virtual feed-forward control (Pipeleers, Demeulenaere and Swevers 2009).

3.2.2 Introduction to Disturbance Observers

In this subsection, some simple disturbance observers are briefly introduced and discussed, first using transfer function filter formats and then later in state space.

3.2.2.1 Disturbance observer construction

The problem here is to compute the output disturbance of a process from its input and output measurements, where the output measurements are also assumed to be corrupted with a measurement error (noise), as shown in Figure 3.7.

$$\begin{aligned} y_m(t) &= y_u(t) + d_o(t) + e_m(t); \\ y_m(s) &= G_P(s)u(s) + d_o(s) + e_m(s). \end{aligned} \quad (3.18)$$

Now, if the disturbance estimate is needed and the process dynamics can be represented or approximated by some mathematical model, so that the response of the process dynamics to

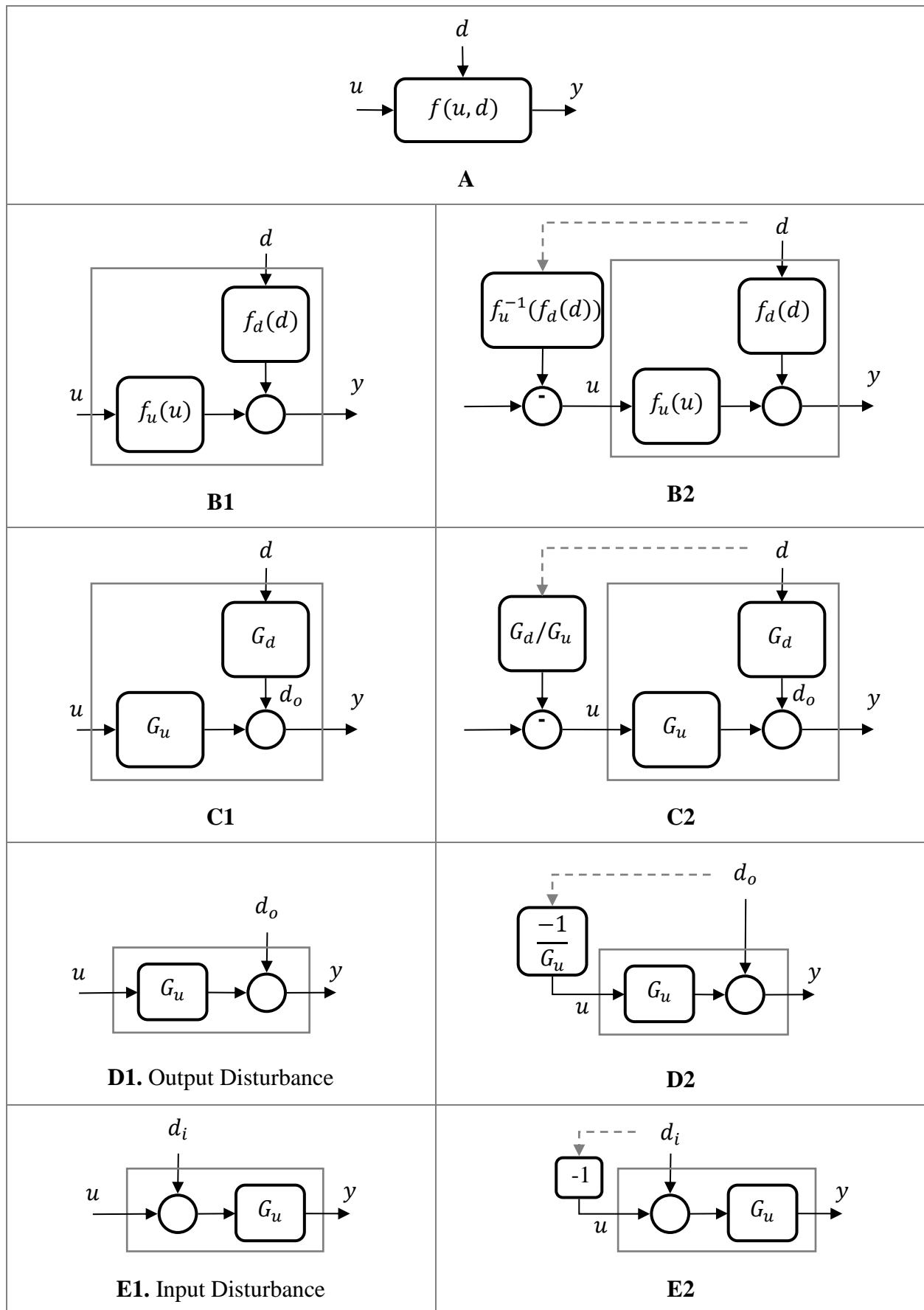


Figure 3.6: Disturbances and their feed-forward control.

the input can be estimated by

$$\hat{y}_u(s) = \hat{G}_P(s)u(s). \quad (3.19)$$

Consequently, the output disturbance can be computed (estimated) by

$$\hat{d}_o(t) = y_m(t) - \hat{y}_u(t). \quad (3.20)$$

Since the process dynamics cannot in reality be structured in a model without any error, or in other words, there is actually an error, because of a model structure that cannot take in consideration all modes or nonlinear characteristics of the process. Therefore, this modeling error is defined as a disturbance added to the estimated output of the process dynamics

$$y_u(t) = \hat{y}_u(t) + d_{ME}(t). \quad (3.21)$$

So, the disturbance computation will be

$$\hat{d}_o(t) = y_u(t) + d_o(t) + e_m(t) - (y_u(t) - d_{ME}(t)), \quad (3.22)$$

which gives

$$\hat{d}_o(t) = d_o(t) + e_m(t) + d_{ME}(t). \quad (3.23)$$

Thus at the end, the computation of the disturbance is corrupted with the measurement and the modeling error of the system. Moreover, the last disturbance observer computes the estimate of the output disturbance, which is not directly applicable if its effect on the output is wanted to be compensated. Since it must be first converted to an input disturbance and that needs the inverse of the estimated process dynamics, as in equation (3.24) and Figure 3.7 show.

$$\hat{d}_i(s) = (\hat{G}_P(s))^{-1} \hat{d}_o(s), \quad (3.24)$$

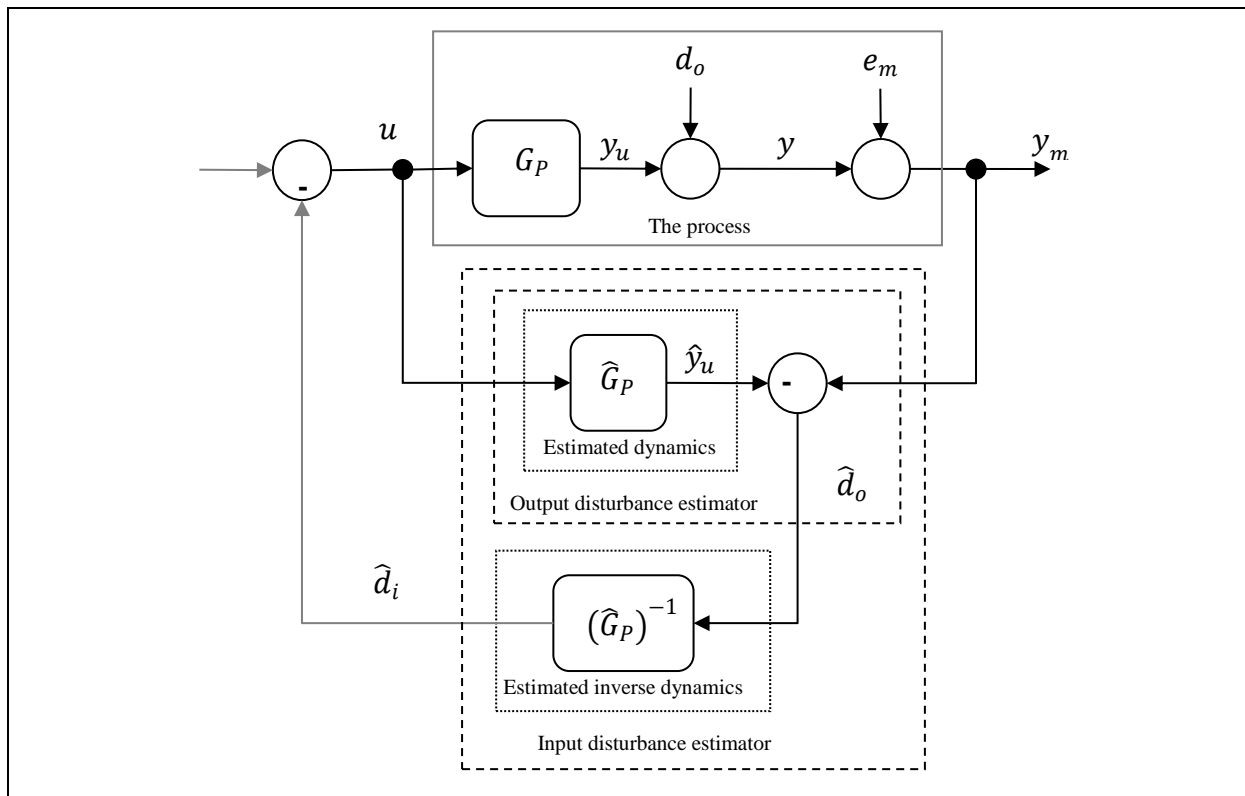


Figure 3.7: Output-input disturbance observer.

provided that the (estimated) process has no non-minimum phase (unstable) zeros and its inverse is a causal system. This way of computing the input disturbance is indirect and susceptible to a lot of computation errors, since the estimated process dynamics and their inverse are needed in the computation. For a better performance, an input disturbance observer is introduced that uses only the estimate of the process dynamics, plus a feedback correction is applied by amplifying the observer error between the real (measured) and the estimated output as shown in the Figure 3.8. So, the estimated input disturbance is given as

$$\hat{d}_i(s) = \frac{G_c(s)G_P(s)}{1 + G_c(s)\hat{G}_P(s)} d_i(s). \quad (3.25)$$

For the case that $G_c(s)\hat{G}_P(s) \gg 1$, the estimated disturbance becomes

$$\hat{d}_i(s) = \frac{G_P(s)}{\hat{G}_P(s)} d_i(s), \quad (3.26)$$

and here, the input disturbance is estimated without the need to compute the inverse of the estimated dynamics with the condition that $\hat{G}_P(s) = G_P(s)$. The estimated input disturbance will be exactly equal to the real input disturbance. Of course, the condition will hold only for a specific bandwidth determined by the closed loop poles.

Moreover, the design of the G_c depends on the type of the disturbance either be a real external independent disturbance acting on the system or an internal disturbances coming from unknown or un-modeled dynamics. For more practical information refer to Ellis (2002), and for the disturbance observer Q-Filter type refer to Du, et al. (2010); Ohishi 1987; Ohishi (2010). So generally, when the disturbance is simply a constant bias or its time or frequency behavior is unknown. Then, it can be generally modeled as (variable) constant when the observer bandwidth or their poles are (as a rule of thumb) at least ten times faster than the disturbance bandwidth. This leads of course to high gain design that its application could be limited by

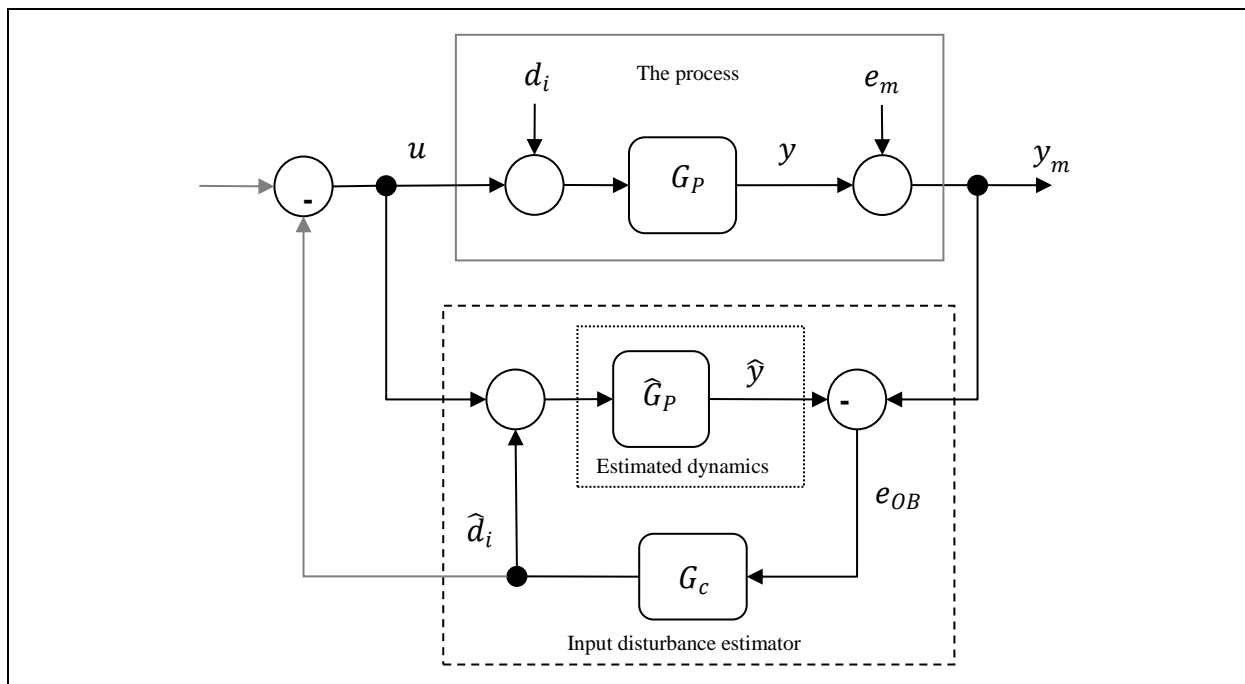


Figure 3.8: Computing the input disturbance without using the inverse of estimated dynamics.

the noise ratio in the system. Otherwise, it would be better to be modeled within G_c according to the IMP, this could achieve better observation with relatively low observer gain. The idea is also presented using state space models in subsection 3.2.3.

3.2.2.2 Model following as disturbance canceller

Now, instead of observing the (input) disturbance acting on a process and then using its estimated signal to compensate the disturbance effect on the output, alternatively, the process can be forced to follow specified dynamics given as a (linear/nonlinear) model. By doing so, the output response becomes as a projection of the specified dynamics, even when the process has external disturbances, as long as these disturbances and model mismatches are in the working bandwidth of the closed loop system. This can be done, for example, by considering the (inexact) model following case, particularly when the process inverse dynamics are not available or unknown, therefore from Figure 3.9, when the block $(\hat{G}_P(s)/G_P(s))$ is substituted by or put equal to one, then the system output becomes

$$y_m(s) = \frac{G_P(s) (1 + G_c(s)\hat{G}_P(s))}{1 + G_c(s)G_P(s)} r(s) + \frac{d_o(s)}{1 + G_c(s)G_P(s)} + \frac{e_m(s)}{1 + G_c(s)G_P(s)}. \quad (3.27)$$

From equation (3.27), if $\hat{G}_P(s) = G_P(s)$ then the closed loop response to the new input $r(s)$ becomes $G_{CL}(s) = G_P(s)$, and for the case, when $\hat{G}_P(s) \neq G_P(s)$, $G_c(s)G_P(s) \gg 1$ and $G_c(s)\hat{G}_P(s) \gg 1$ the closed loop response becomes $G_{CL}(s) \approx \hat{G}_P(s)$.

This means that the closed loop system follows the desired dynamics $\hat{G}_P(s)$ in perspective of the new input $r(s)$. This concept is a complementary concept of the disturbance observer, where the disturbance cancelling is done by using the estimated dynamics and the disturbances, while (inexact) model following is done by giving the desired system dynamics.

Moreover, the system suppresses the effect of the disturbance and the measurement error in the specified system bandwidth where the condition $(G_c(s)G_P(s) \gg 1)$ holds, for more details about model following control refer, for example, to Skoczowski, Domek and Pietrusiewicz (2003).

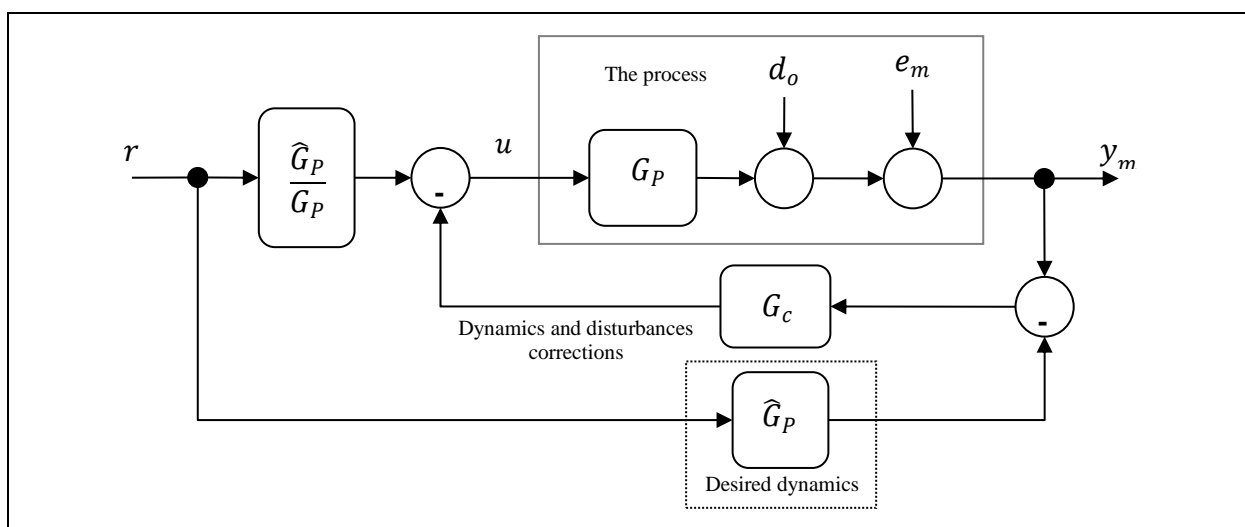


Figure 3.9: Model following control concept.

3.2.3 State and Disturbance Observer in State Space

The modern state observers can be traced back to 1960, where the idea of the stochastic state observer (Kalman 1960) was introduced and in 1961 developed as Kalman-Bucy-Filter (Kalman and Bucy 1961). Later in 1964, the deterministic version was reintroduced by Luenberger (Luenberger 1964). For more information about the state and disturbance observers refer, for example, to Radke and Gao (2006).

3.2.3.1 Extended state space observer

Now, the normal state observer, which is usually used to estimate the system states, is extended by adding an extra state to represent the input disturbance to the system. Therefore, if the system dynamics are represented by the following state space equations

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} u(t); \quad y(t) = \mathbf{C} \mathbf{x}(t), \quad (3.28)$$

then the new extended system equations become

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{x}_d(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ x_d(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u(t); \\ y(t) &= [\mathbf{C} \quad 0] \begin{bmatrix} \mathbf{x}(t) \\ x_d(t) \end{bmatrix}, \end{aligned} \quad (3.29)$$

where the extended state (disturbance) is assumed to be constant

$$x_d(t) = \text{constant}; \quad \dot{x}_d(t) = 0, \quad (3.30)$$

in continuous time, or in discrete-time format

$$x_d(t+1) = x_d(t). \quad (3.31)$$

The extended state observer can be designed, for example, by using pole placement design methods to find the observer gain vector. For this case, if the disturbances are not constant the observer will catch the change (as a rule of thumb) as long as it is at least ten times slower than the slowest observer pole. But for more specific disturbances, e.g. periodic disturbances, to get an exact estimation of these disturbance signals, an internal model should be used to represent these signal dynamics as shown briefly in the following subsection.

3.2.3.2 Extended sinusoidal disturbance observer

The extended disturbance can be generated by using a sinusoidal signal generator in state space format with a constant angular frequency (see, e.g., Kinney and Callafon 2006) as

$$x_{d1}(t) = \sin(\omega t), \quad (3.32)$$

$$x_{d2}(t) = \dot{x}_{d1}(t) = \omega \cos(\omega t), \quad (3.33)$$

$$\dot{x}_{d2}(t) = -\omega^2 \sin(\omega t) = -\omega^2 x_{d1}(t). \quad (3.34)$$

The sinusoidal generator system can be put in a state space model as

$$\begin{aligned} \begin{bmatrix} \dot{x}_{d1}(t) \\ \dot{x}_{d2}(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \end{bmatrix}; \\ y_d(t) &= [1 \quad 0] \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \end{bmatrix}. \end{aligned} \quad (3.35)$$

Alternatively, the states can also be defined as

$$x_{d1}(t) = \sin(\omega t); x_{d2}(t) = \cos(\omega t), \quad (3.36)$$

$$\dot{x}_{d1}(t) = \omega \cos(\omega t) = \omega x_{d2}(t), \quad (3.37)$$

$$\dot{x}_{d2}(t) = -\omega \sin(\omega t) = -\omega x_{d1}(t). \quad (3.38)$$

Then, the state space sinusoidal generator system can alternatively be put in the form

$$\begin{aligned} \begin{bmatrix} \dot{x}_{d1}(t) \\ \dot{x}_{d2}(t) \end{bmatrix} &= \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \end{bmatrix}; \\ y_d(t) &= [1 \quad 0] \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \end{bmatrix}, \end{aligned} \quad (3.39)$$

or generally as variable frequency derived from a time variant angular position $\varphi(t)$

$$x_{d1}(t) = \sin(\varphi(t)); x_{d2}(t) = \cos(\varphi(t)), \quad (3.40)$$

$$\dot{x}_{d1}(t) = \dot{\varphi}(t) \cos(\varphi(t)) = \dot{\varphi}(t) x_{d2}(t), \quad (3.41)$$

$$\dot{x}_{d2}(t) = -\dot{\varphi}(t) \sin(\varphi(t)) = -\dot{\varphi}(t) x_{d1}(t). \quad (3.42)$$

Therefore, the state space model becomes as

$$\begin{aligned} \begin{bmatrix} \dot{x}_{d1}(t) \\ \dot{x}_{d2}(t) \end{bmatrix} &= \begin{bmatrix} 0 & \dot{\varphi}(t) \\ -\dot{\varphi}(t) & 0 \end{bmatrix} \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \end{bmatrix}; \\ y_d(t) &= [1 \quad 0] \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \end{bmatrix}. \end{aligned} \quad (3.43)$$

There are a lot of periodic disturbance compensation methods based on disturbance observers, for example, by designing a set of linear time-invariant observers for a set of disturbance frequencies in some operating regions, that results in gain scheduled control, for example, Bohn et al. (2004) have suggested that the observer design can be done by using pole placement techniques or by designing an optimal stationary Kalman filter. Moreover, robust control LPV-techniques can also be used to design the observer and controller vectors, for example, Du and Shi (2002), Du, et al. (2003) and Kinney and Callafon (2006) have developed H_∞ and LPV methods respectively for active vibration control applications. Furthermore, Ballesteros and Bohn (2011) have developed and applied the robust control LPV design algorithms in the discrete-time format. In addition, Kinney and Callafon (2011) have developed a study of rapidly varying frequencies regulation guarantee. For extra discussion about the so called “waterbed effect” or “spillover” that accompanies these algorithms, see for example, (Bohn, et al. 2004; Hong and Bernstein 1998).

3.2.3.3 State and disturbance observer as model follower

The state observer can also be interpreted to work as a model follower, for instance, if the desired system dynamics are given in the state observer model, then the disturbance state can be used to cancel the external disturbances and internal disturbances in terms of dynamic difference between the desired model and the real one. But these again need high loop gain, or it will only work in the observer bandwidth represented by the observer poles, which have been set by the observer gain vector.

3.2.4 Adaptive Periodic Disturbance Cancellation

Another way of cancelling the periodic disturbance effect on the system output is by using a direct estimation and cancellation methods of the input periodic disturbance. These methods have been mostly developed by the signal processing community for active noise (sound) and vibration cancellation (control) applications. For example, consider the system presented in Figure 3.10 below, where the process is disturbed by a harmonic input disturbance with a known frequency (ω_d). Therefore, a simple solution is to generate a harmonic (sinusoidal) signal as following equation

$$u(t) = -\hat{d}_i(t) = \alpha \sin(\omega_d t) + \beta \cos(\omega_d t), \quad (3.44)$$

or alternatively as

$$u(t) = -\hat{d}_i(t) = a_d \sin(\omega_d t + \phi_d), \quad (3.45)$$

where

$$a_d = \sqrt{\alpha^2 + \beta^2} ; \quad \phi_d = \tan^{-1}\left(\frac{\beta}{\alpha}\right), \quad (3.46)$$

and to tune its parameters, the amplitudes α and β , or magnitude a_d and phase shift ϕ_d , until the effect of the harmonic disturbance is completely (vanished) canceled out (Lueg 1936). The parameter adaptation could be practically done manually as Conover (1956) demonstrated an active noise cancellation system for transformer noise, or by using adaptive algorithms, for example, the least mean squares algorithm and its variants (Morgan 1980; Kuo and Morgan 1996; Widrow and Stearns 1985; Hansen 2001; Elliott 2001). These algorithms are usually known as adaptive feed-forward in signal processing literature, although they are “truly and purely a feedback control law” (Bodson 2005). Moreover, Bodson (2004) has

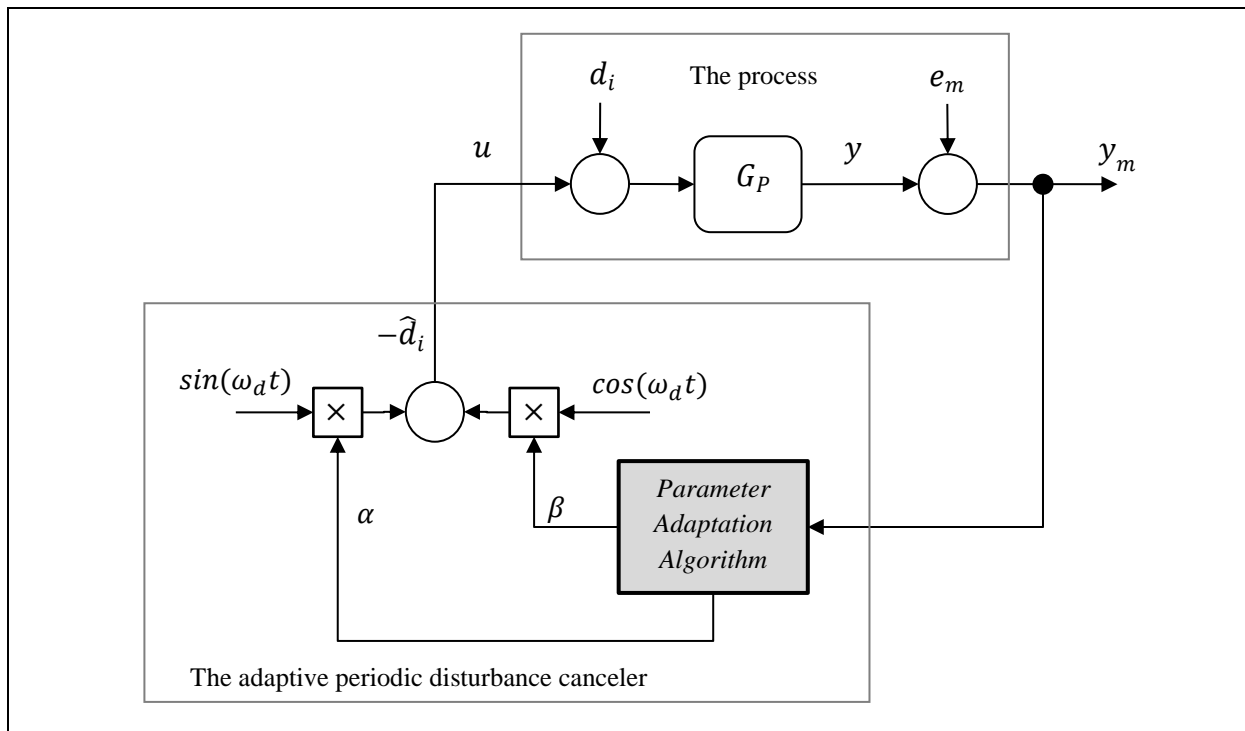


Figure 3.10: The adaptive periodic disturbance canceler

shown that these adaptive feed-forward controllers can be equivalent under certain conditions to the internal model principle linear controllers.

However, the advantage of using the LTI-IMP periodic disturbance observer is that the closed loop stability can be computed and proven, provided that an accurate mathematical model for the targeted process is available at least in the operating bandwidth. But, on the other hand, for this adaptive feed-forward controller, the closed loop stability analysis depends on the adaptation algorithm used to tune its parameters which has usually a time-varying nonlinear characteristics, this makes the stability analysis difficult to prove in general.

Nevertheless, an advantage of this adaptive feed-forward control, when the adaptive algorithm is deactivated after the convergence of its parameters to the optimal ones, especially when the disturbance parameters are not time varying, so that the estimated disturbance signal compensates or cancels out the periodic disturbance effect on the system output. Then, at this situation the pseudo feed-forward controller becomes a true feed-forward controller, and therefore it will not affect the system closed loop dynamics anymore, where they can then be freely designed to fulfil the stability and set point design demands separately.

3.3 Feed-forward Disturbance Compensation of the Drive-load System

3.3.1 Rigid Drive-load System

The model of the rigid drive-load system as presented in chapter 2 and shown in Figure 3.11, it has an input disturbance format. This format has the simplest feed-forward control law, which is a static and not dynamic, since the disturbance and the drive torques are acting simultaneously (in phase) at the same inertia. Therefore, the feed-forward control law is simply given by

$$G_{FF} = -1, \quad (3.47)$$

since

$$T = -T_{dis}. \quad (3.48)$$

To achieve this design the disturbance signal must be available in form of measurement, observation, estimation or prediction.

Although, the feed-forward control law for this model structure is very simple, still the problem is not trivial. Since as already stated, the disturbance signal should be directly available in the first place. Therefore, the methods based on periodic disturbance observers can be used to estimate the input periodic disturbance to cancel the effect of the internal and the external disturbances on the system output. Moreover, the adaptive periodic disturbance canceler (the adaptive feed-forward control) can also be used to cancel the external and the internal periodic disturbances of the drive-load system. But, applying this technique at the same time with a set point tracking negative feedback controller is not preferable generally for reasons that were already discussed in introduction (section 1.2) and will be discussed more in chapter 4 again. Instead, the modeling and the identification of the internal periodic disturbance function will be considered and explained in chapter 4.

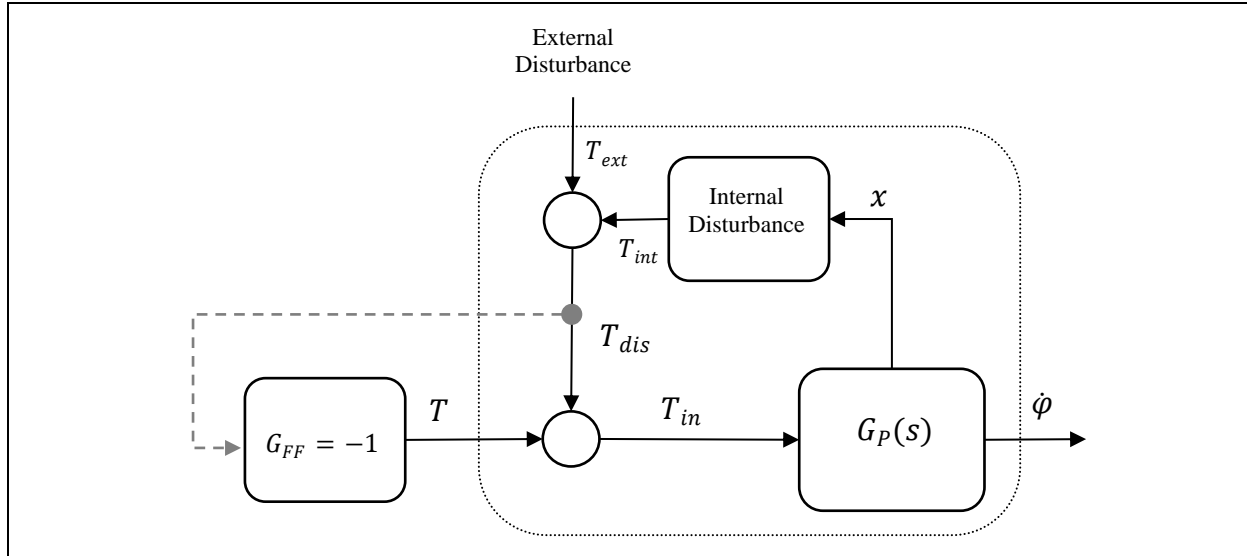


Figure 3.11: Feed-forward control of internally and externally disturbed system.

3.3.2 Flexible Drive-load System

The feed-forward control law for the flexible drive-load system is not only dynamic, it is also non causal system, since the degree of the numerator polynomial is higher than the denominator polynomial degree as can be seen from the modeling of the system, see Figure 3.13. The feed-forward control law is given by

$$G_{FF} = -\frac{P_{NL}}{P_{ND}} = -\frac{b_{L2}s^2 + b_{L1}s + b_{L0}}{b_{D1}s + b_{D0}}. \quad (3.49)$$

In other words, if a non-causal system is to be realized, then it needs to have the future value of the input to compute the current output. Fortunately, oscillations are periodic signals, their future value can easily be predicted (determined) provided that their parameters (frequency, amplitude and phase-shift) are known. Therefore, the causality for the case of periodic disturbance or oscillations makes no problem as long as the Fourier series expansion is used to represent the disturbance signal, where it can be shifted backward or forward until the right phase is achieved. For example, if there is a sine wave disturbance with phase of $+45^\circ$ and unity magnitude, then this can also be done by making phase shift of -135° with magnitude of minus unity, as shown in the Figure 3.12 below. Now, if the model structure of the flexible drive-load system of Figure 3.13 is transformed into input disturbance format, as the rigid case, but the system dynamics can have higher order. Then the feed-forward control law will also have the simplest format as in rigid case, see Figure 3.14. So by doing this, instead of measuring (observing, estimation or prediction) the real disturbance, the input disturbance is directly measured and used. Therefore, as Figure 3.14 shows, if a local or a global model can in particular be built for the internal periodic disturbance path, and consequently identified, then it can be used to compensate this internal disturbance locally or globally respectively. Finally, the algorithm of modeling, identification and control of periodic disturbances will be introduced in chapter 4.

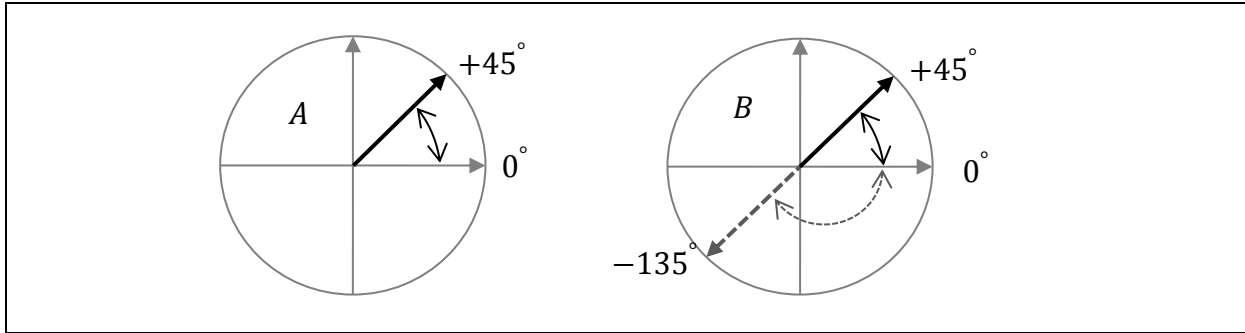


Figure 3.12: Phasor diagram.

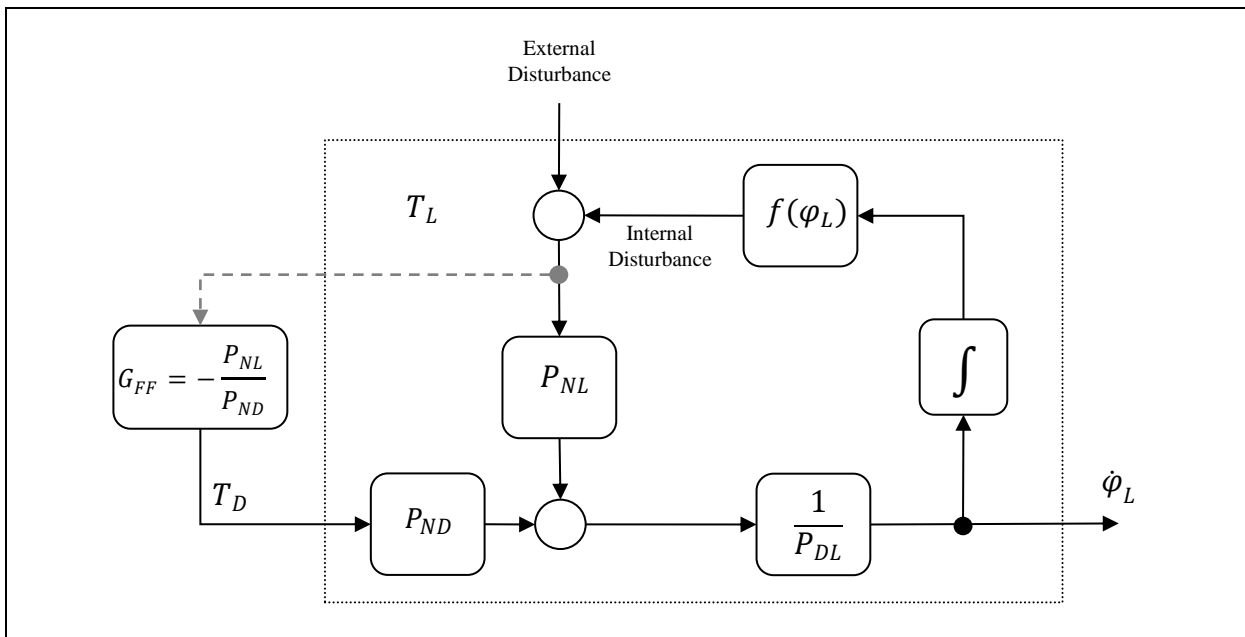


Figure 3.13: Load-side disturbance feed-forward control of the flexible drive-load system.

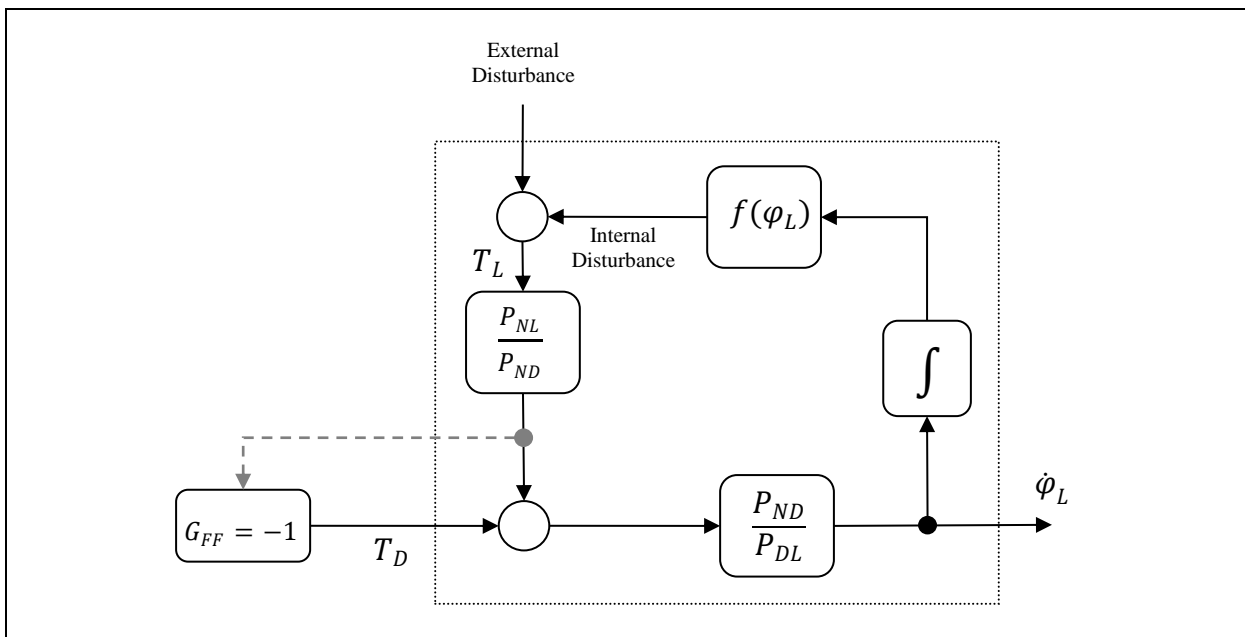


Figure 3.14: Input disturbance feed-forward control of the flexible drive-load system.

4 MODELING, IDENTIFICATION AND CONTROL

In this chapter, the general strategy of the feedback control for set point tracking and feed-forward control for periodic disturbance compensation is introduced. Thereby, the building up of the identification model and its parameterization as continuous or discrete time are presented. Moreover, the implementation of the nonlinear least squares recursive Gauss-Newton and the linear recursive least squares methods are presented to identify the identification model parameters.

4.1 Introduction

Most of the adaptive feed-forward control strategies, e.g. Least Mean Squares (LMS) family methods (Widrow and Stearns 1985), developed for pure active noise and vibration control applications do not suit in particular this control problem because of stability issue, parameter convergence and the direct online adaptation (tuning) problems (Elliott 2001; Bodson 2005). This means that as long as the algorithm does not converge, the system operates with very poor performance and that will be until and only if the parameters converge.

Wit and Parly (2000) have developed an adaptive eccentricity compensation algorithm, which is an adaptive feed-forward control, Maier, Bals and Bodson (2011) have also developed an adaptive feed-forward to cancel the ripples of a permanent magnet synchronous motor, but both of them are with a direct controller parameter (tuning) optimization algorithm. Na and Park (1997) have developed an adaptive feed-forward control that works like the typical LMS-algorithm but in indirect way, by using pre-estimated process and closed loop transfer functions, so that the controller works as an add-on controller to reject the periodic disturbances without affecting the original feedback closed loop system characteristics; also, Wang and Ren (1994) have developed an indirect adaptive feed-forward control for and to be applied exclusively in active noise and vibration control systems. While in this work, the problem of feed-forward control adaptation will be transferred into identification (parameter optimization) problem. Where, the identification model is constructed by two parts. The dynamic part represents the input to output dynamics and made in terms of linear, nonlinear differential or difference equation. And the disturbance part, which made in terms of sine cosine sum functions representing the input periodic disturbances (Bodson 2005). This will allow the identification to gather information online (offline as a priori) about the process, and to pass the converged parameters to use in the feed-forward control law, even when the feed-forward controller is deactivated, particularly at the starting phase, as the dynamic and disturbance parameters are far from the optimal ones.

So in the next, a new method is presented for feed-forward control of periodic disturbances (Alsogkier and Bohn 2012), that is originally stemmed from the adaptive feed-forward active noise and vibration control applications, for example, LMS method families (Elliott 2001). Nevertheless, it will be changed so that, it will suit the usual control applications. The main problem of the adaptive feed-forward control, used in active vibration control applications, is that the online (direct) tuning (adaptation) of the controller parameters, which will of course deteriorate the system performance in the tuning phase, when the parameters are far from the

optimal ones. The new method is to shift this online direct adaptation of the controller parameters into an identification problem. This yields an indirect adaptive control strategy (Aström and Wittenmark 1995; Landau, Lozano and M'Saad 1998).

The disturbed process is modeled by a dynamic model plus a disturbance model. By this type of modeling and identification, the adaptation can practically be done for both controllers. Since, the identified dynamic parameters can be used to compute the feedback control law, and the identified disturbance parameters are used to construct the feed-forward controller. This strategy, adaptive feedback control for set point tracking and adaptive feed-forward control for disturbance compensation, is schematically presented in Figure 4.1. Nevertheless, the rest of the chapter is only concerned with main target of this work the add-on adaptive feed-forward control to the preexisting set point tracking feedback controller, therefore, the feedback controller is kept as simple as possible in the rest of this work. The identification model structure can be parameterized and identified by any parametric nonlinear identification technique (Aström and Eykhoff 1971; Ljung and Söderström 1983; Söderström and

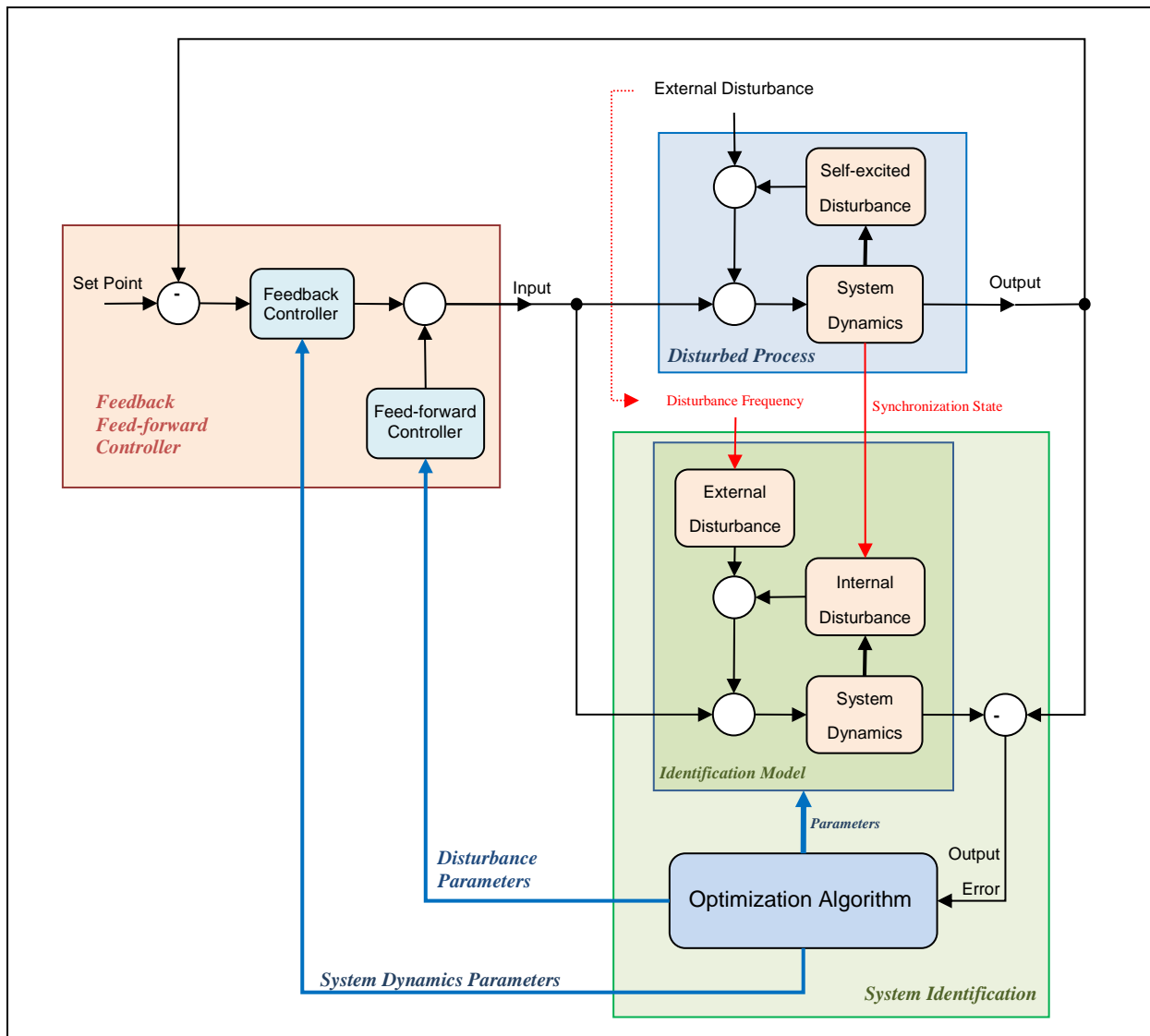


Figure 4.1: General control strategy of the adaptive set point feedback controller and periodic disturbance feed-forward controller.

Stoica 1989; Ljung 1999; Verhaegen and Verdult 2007; Mikles and Fikar 2007; Garnier and Wang 2008), for example, output error, prediction error or even with the method of linear simple recursive least squares, provided that the model structure has linear in parameter form. The choice of the identification procedure depends on many conditions, some of them are model structure, measurement noise, type of model based or state based control strategy, conditions for identification in closed loop, identification for control, etc. If the parameters converge, this means the parameters of the system dynamics and the disturbance model. Then, the parameters of disturbance model can be used to generate an anti-disturbance signal that can be fed forward to the system input to compensate the disturbance action on the system output. In general, the successfulness of the procedure is directly dependent on the identification algorithm. So, if the identification algorithms converge to parameters, and these parameters represent the real dynamics and disturbance of the real system at the interested frequencies (bandwidth), then these parameters can be used to compensate the disturbance. Otherwise, the parameters are useless, since they do not represent the real process.

In the next, an identification model is built up especially for the externally internally periodically disturbed drive-load system already presented in chapter 2. The system is mainly constructed as angular velocity servo control system, so that its input is the drive torque and its output is the load angular velocity.

4.2 Externally and Internally Disturbed Identification Modell

Now, an identification model is developed for an externally and internally periodically disturbed drive-load process that was already introduced in chapter 2. The identification model structure and its parameterization should take into consideration all of the available priory knowledge about the real (physical) plant as much as possible. Moreover, the disturbance model is constructed to have all its disturbances added to its input, i.e. input disturbance format, see Figure 4.2. This makes the use of identified disturbance model and its application very direct and without extra computations, even if the real system has another format, e.g. output disturbance format, as long as these disturbances are periodic. Moreover, it is assumed that the external periodic disturbance frequency (ω [rad/s]) is known either by direct measurements or by estimation. Furthermore, the synchronization signal (φ_m [rad]), in this case the angular position of the load side, is also available either by direct measurements or indirectly as observation, estimation or prediction.

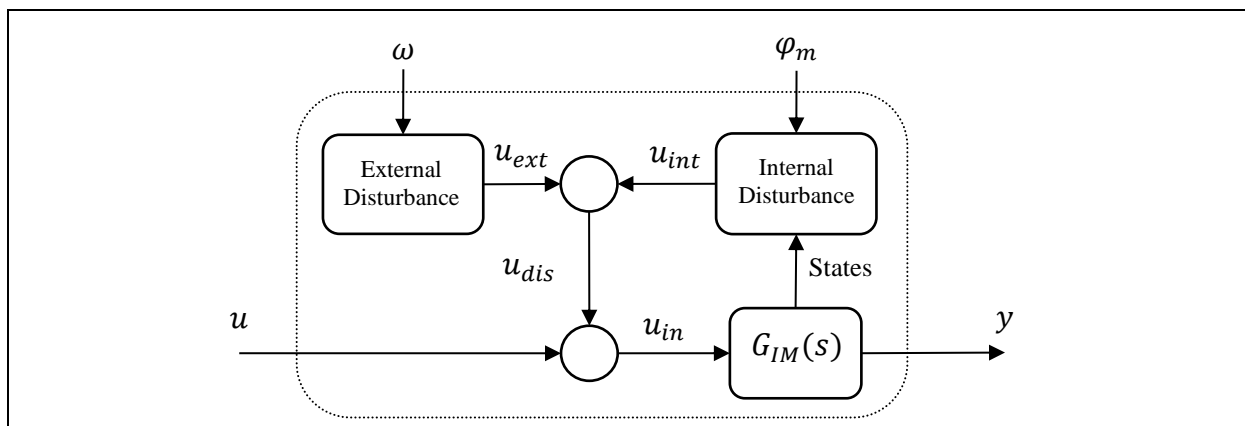


Figure 4.2: Externally and internally disturbed identification model.

4.2.1 Building up the Identification Model

The identification model is constructed by two parts, the dynamic part and the disturbance part. The dynamic part represents the effect of the total input on the output, in other words, the input-output dynamics. While the disturbance part represents the external and the internal periodic disturbance of the process which are mainly to be compensated. Therefore, the system dynamics are represented by an order n linear transfer function as

$$G_{IM}(s) = \frac{y(s)}{u_{in}(s)} = \frac{b_{n-1}s^{n-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}. \quad (4.1)$$

The identification model input-output dynamics can also be extended to some nonlinear dynamics or time delay if necessary. The input of the system dynamics are the sum of the manipulated (control) input variable and the input periodic disturbance as

$$u_{in}(t) = u(t) + u_{dis}(t). \quad (4.2)$$

The input periodic disturbance is in turn also the sum of the external and the internal input disturbance as

$$u_{dis}(t) = u_{ext}(t) + u_{int}(t). \quad (4.3)$$

Moreover, the periodic external and internal input disturbances are approximated by their Fourier series expansion as

$$\begin{aligned} u_{dis}(t) = & \sum_{i=1}^{N_0} [\alpha_{0,i} \sin(i\omega t) + \beta_{0,i} \cos(i\omega t)] \\ & + \sum_{i=1}^{N_1} [\alpha_{1,i} \sin(i\varphi_m) + \beta_{1,i} \cos(i\varphi_m)] \\ & + \sum_{i=1}^{N_2} [\alpha_{2,i} \sin(i\varphi_m) + \beta_{2,i} \cos(i\varphi_m)] y(t) \\ & + \sum_{i=1}^{N_3} [\alpha_{3,i} \sin(i\varphi_m) + \beta_{3,i} \cos(i\varphi_m)] y^2(t). \end{aligned} \quad (4.4)$$

The first sum is for the external disturbance, the second, the third and fourth are for the state dependent internal disturbances built to represent the angle dependent load elements spring, damper and moment of inertia as introduced in chapter 2. The approximation of the periodic disturbances by their finite Fourier series is valid as long as the input-output system dynamics have a low pass filter behavior. Moreover, the disturbance series format (4.4) can be transformed into vector format as

$$\begin{aligned} u_{dis}(t) = & [\alpha_0^T \sin(\omega t) + \beta_0^T \cos(\omega t)] \\ & + [\alpha_1^T \sin(\varphi_m) + \beta_1^T \cos(\varphi_m)] \\ & + [\alpha_2^T \sin(\varphi_m) + \beta_2^T \cos(\varphi_m)] y(t) \\ & + [\alpha_3^T \sin(\varphi_m) + \beta_3^T \cos(\varphi_m)] y^2(t), \end{aligned} \quad (4.5)$$

where

$$\boldsymbol{\alpha}_i = \begin{bmatrix} \alpha_{i,1} \\ \alpha_{i,2} \\ \vdots \\ \alpha_{i,N_i} \end{bmatrix}; \boldsymbol{\beta}_i = \begin{bmatrix} \beta_{i,1} \\ \beta_{i,2} \\ \vdots \\ \beta_{i,N_i} \end{bmatrix}, \quad (4.6)$$

and

$$\begin{aligned} \mathbf{sin}(\omega t) &= \begin{bmatrix} \sin(\omega t) \\ \sin(2\omega t) \\ \vdots \\ \sin(N_0\omega t) \end{bmatrix}; \mathbf{cos}(\omega t) = \begin{bmatrix} \cos(\omega t) \\ \cos(2\omega t) \\ \vdots \\ \cos(N_0\omega t) \end{bmatrix}; \\ \mathbf{sin}(\varphi_m) &= \begin{bmatrix} \sin(\varphi_m) \\ \sin(2\varphi_m) \\ \vdots \\ \sin(N_i\varphi_m) \end{bmatrix}; \mathbf{cos}(\varphi_m) = \begin{bmatrix} \cos(\varphi_m) \\ \cos(2\varphi_m) \\ \vdots \\ \cos(N_i\varphi_m) \end{bmatrix}. \end{aligned} \quad (4.7)$$

Moreover, the vector of sine cosine disturbance components can be defined as

$$\begin{aligned} \mathbf{u}_{SC}(t)^T &= [\mathbf{sin}(\omega t)^T \quad \mathbf{cos}(\omega t)^T \quad \mathbf{sin}(\varphi_m)^T \quad \mathbf{cos}(\varphi_m)^T \\ &\quad y(t) \mathbf{sin}(\varphi_m)^T \quad y(t) \mathbf{cos}(\varphi_m)^T \quad y^2(t) \mathbf{sin}(\varphi_m)^T \quad y^2(t) \mathbf{cos}(\varphi_m)^T]. \end{aligned} \quad (4.8)$$

Therefore, the total disturbance, defined in equation (4.4), can also be computed by

$$\mathbf{u}_{dis}(t) = [\boldsymbol{\alpha}_0^T \quad \boldsymbol{\beta}_0^T \quad \boldsymbol{\alpha}_1^T \quad \boldsymbol{\beta}_1^T \quad \boldsymbol{\alpha}_2^T \quad \boldsymbol{\beta}_2^T \quad \boldsymbol{\alpha}_3^T \quad \boldsymbol{\beta}_3^T] \mathbf{u}_{SC}(t). \quad (4.9)$$

Now, the state model of an externally and internally periodically disturbed drive-load process with linear input-output dynamics is presented in the following equations

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \mathbf{x}(t) \\ &+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \boldsymbol{\alpha}_0^T & \boldsymbol{\beta}_0^T & \boldsymbol{\alpha}_1^T & \boldsymbol{\beta}_1^T & \boldsymbol{\alpha}_2^T & \boldsymbol{\beta}_2^T & \boldsymbol{\alpha}_3^T & \boldsymbol{\beta}_3^T \end{bmatrix} \begin{bmatrix} u(t) \\ \mathbf{sin}(\omega t) \\ \mathbf{cos}(\omega t) \\ \mathbf{sin}(\varphi_m) \\ \mathbf{cos}(\varphi_m) \\ \mathbf{sin}(\varphi_m) y \\ \mathbf{cos}(\varphi_m) y \\ \mathbf{sin}(\varphi_m) y^2 \\ \mathbf{cos}(\varphi_m) y^2 \end{bmatrix}; \end{aligned} \quad (4.10)$$

$$y(t) = [b_0 \quad b_1 \quad \cdots \quad b_{n-1}] \mathbf{x}(t).$$

So, the identification model can generally be defined by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t, \mathbf{x});$$

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \alpha_0^T & \beta_0^T & \alpha_1^T & \beta_1^T & \alpha_2^T & \beta_2^T & \alpha_3^T & \beta_3^T \end{bmatrix} \begin{bmatrix} u(t) \\ \sin(\omega t) \\ \cos(\omega t) \\ \sin(x_m) \\ \cos(x_m) \\ \sin(x_m) \mathbf{C}\mathbf{x}(t) \\ \cos(x_m) \mathbf{C}\mathbf{x}(t) \\ \sin(x_m) [\mathbf{C}\mathbf{x}]^2(t) \\ \cos(x_m) [\mathbf{C}\mathbf{x}]^2(t) \end{bmatrix}; \quad (4.11)$$

$$y(t) = \mathbf{C}\mathbf{x}(t);$$

$$y(t) = [b_0 \quad b_1 \quad \cdots \quad b_{n-1}] \mathbf{x}(t),$$

where x_m is the measured synchronization state signal, which in this case is the measured angular position φ_m . The parameter vector of the state space identification model, defined by equations (4.11), is given as

$$\boldsymbol{\theta}^T = \begin{bmatrix} -a_0 & -a_1 & \cdots & -a_{n-1} & \vdots & b_0 & b_1 & \cdots & b_{n-1} \\ \alpha_{0,1} & \alpha_{0,2} & \cdots & \alpha_{0,N_0} & \vdots & \beta_{0,1} & \beta_{0,2} & \cdots & \beta_{0,N_0} \\ \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,N_1} & \vdots & \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,N_1} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,N_2} & \vdots & \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,N_2} \\ \alpha_{3,1} & \alpha_{3,2} & \cdots & \alpha_{3,N_3} & \vdots & \beta_{3,1} & \beta_{3,2} & \cdots & \beta_{3,N_3} \end{bmatrix}. \quad (4.12)$$

Now, since the developed identification model is in the form of nonlinear in parameters, their parameters can be identified by any nonlinear in parameter identification algorithm, for example, Gauss-Newton output error, prediction error or even with dual (extended) Kalman filter to predict the system states and its parameters in a stochastic environment. For more details about system identification in general, see Appendix B.

Then, the identified disturbance parameters can be used in the design of the disturbance feed-forward controller, as well as the system dynamic parameters can also be used in the design of the feedback controller. Actually, this formulation fits the general idea of (adaptive) model based control design, for example, state feedback control strategies.

4.2.2 Parameter Identification of the Identification Model

In the following, the Gauss-Newton output error method, presented in Appendix B.3, is used to identify the parameters of the identification model. First, the partial derivatives of the model matrices \mathbf{A} , \mathbf{B} and \mathbf{C} as defined in equation (4.11) are taken with respect to every corresponding parameter of the parameter vector given in equation (4.12) in order to compute the gradients as following:

For the parameters of ($i = 1$ to n)

$$\begin{aligned} \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) &= [0 \quad \dots \quad 0 \quad 1]^T x_i(t, \boldsymbol{\theta}); \\ \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) &= \mathbf{0}; \\ \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{u}(t, \boldsymbol{\theta}) &= \mathbf{0}; \end{aligned}$$

$$\frac{\partial \mathbf{u}(t, \boldsymbol{\theta})}{\partial \theta_i} = \begin{bmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \sin(x_m) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \\ \cos(x_m) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \\ \sin(x_m) 2\mathbf{C}\mathbf{x}(t, \boldsymbol{\theta}) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \\ \cos(x_m) 2\mathbf{C}\mathbf{x}(t, \boldsymbol{\theta}) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \end{bmatrix}, \quad (4.13)$$

and their gradients are computed by simulating the systems

$$\begin{aligned} \frac{\partial \dot{\mathbf{x}}(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{A}(\boldsymbol{\theta}) \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + \mathbf{B}(\boldsymbol{\theta}) \frac{\partial \mathbf{u}(t, \boldsymbol{\theta})}{\partial \theta_i} + [0 \quad \dots \quad 0 \quad 1]^T x_i(t, \boldsymbol{\theta}); \\ \frac{\partial y(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{C}(\boldsymbol{\theta}) \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i}. \end{aligned} \quad (4.14)$$

For the parameters of ($i = n + 1$ to $2n$)

$$\begin{aligned} \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) &= \mathbf{0}; \\ \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) &= x_{i-n}(t, \boldsymbol{\theta}); \\ \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{u}(t, \boldsymbol{\theta}) &= \mathbf{0}; \end{aligned} \quad (4.15)$$

$$\frac{\partial \mathbf{u}(t, \boldsymbol{\theta})}{\partial \theta_i} = \begin{bmatrix} 0 \\ \mathbf{0} \\ 0 \\ \mathbf{0} \\ 0 \\ \sin(x_m) \left[\mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) \right] \\ \cos(x_m) \left[\mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) \right] \\ \sin(x_m) 2\mathbf{C}\mathbf{x}(t, \boldsymbol{\theta}) \left[\mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) \right] \\ \cos(x_m) 2\mathbf{C}\mathbf{x}(t, \boldsymbol{\theta}) \left[\mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) \right] \end{bmatrix},$$

and their gradients are computed by simulating the systems

$$\begin{aligned} \frac{\partial \dot{\mathbf{x}}(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{A}(\boldsymbol{\theta}) \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + \mathbf{B}(\boldsymbol{\theta}) \frac{\partial \mathbf{u}(t, \boldsymbol{\theta})}{\partial \theta_i}; \\ \frac{\partial y(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{C}(\boldsymbol{\theta}) \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + x_{i-n}(t, \boldsymbol{\theta}). \end{aligned} \quad (4.16)$$

And for ($i = 2n + 1$ to $2n + 2N_0 + 2N_1 + 2N_2 + 2N_3$) parameters, their gradients are

$$\begin{aligned} \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) &= \mathbf{0}; \\ \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{x}(t, \boldsymbol{\theta}) &= \mathbf{0}; \\ \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{u}(t, \boldsymbol{\theta}) &= [0 \quad \dots \quad 0 \quad 1]^T u_{SC(i-2n)}(t); \end{aligned}$$

$$\frac{\partial \mathbf{u}(t, \boldsymbol{\theta})}{\partial \theta_i} = \begin{bmatrix} 0 \\ \mathbf{0} \\ 0 \\ \mathbf{0} \\ 0 \\ \sin(x_m) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \\ \cos(x_m) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \\ \sin(x_m) 2\mathbf{C}\mathbf{x}(t, \boldsymbol{\theta}) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \\ \cos(x_m) 2\mathbf{C}\mathbf{x}(t, \boldsymbol{\theta}) \mathbf{C} \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} \end{bmatrix}, \quad (4.17)$$

and their corresponding gradients are computed from

$$\begin{aligned}\frac{\partial \dot{\mathbf{x}}(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{A}(\boldsymbol{\theta}) \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i} + \mathbf{B}(\boldsymbol{\theta}) \frac{\partial \mathbf{u}(t, \boldsymbol{\theta})}{\partial \theta_i} + [0 \quad \cdots \quad 0 \quad 1]^T u_{SC(i-2n)}(t); \\ \frac{\partial y(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{C}(\boldsymbol{\theta}) \frac{\partial \mathbf{x}(t, \boldsymbol{\theta})}{\partial \theta_i}.\end{aligned}\quad (4.18)$$

The previous continuous models can also be developed in discrete-time, this is straight forward. For more details refer to the system identification Appendix B. The presented algorithm will be tested in simulation as well as on real-time control experiments in chapter 5 and chapter 6 respectively.

4.3 Discrete Periodic Disturbance Compensation Using RLS Method

The identification model, as it is shown in following Figure 4.3, has state dependent nonlinear oscillations because of its periodic parameters. This model structure is going to be modeled as linear in parameter discrete model. So that, the recursive least squares identification algorithm is used to identify the linear discrete dynamics and the disturbance model in the form of nonlinear sine cosine function parameters representing the external and the internal periodic disturbances, where the identified sine cosine function parameters are used in a feed-forward controller to compensate the system oscillations.

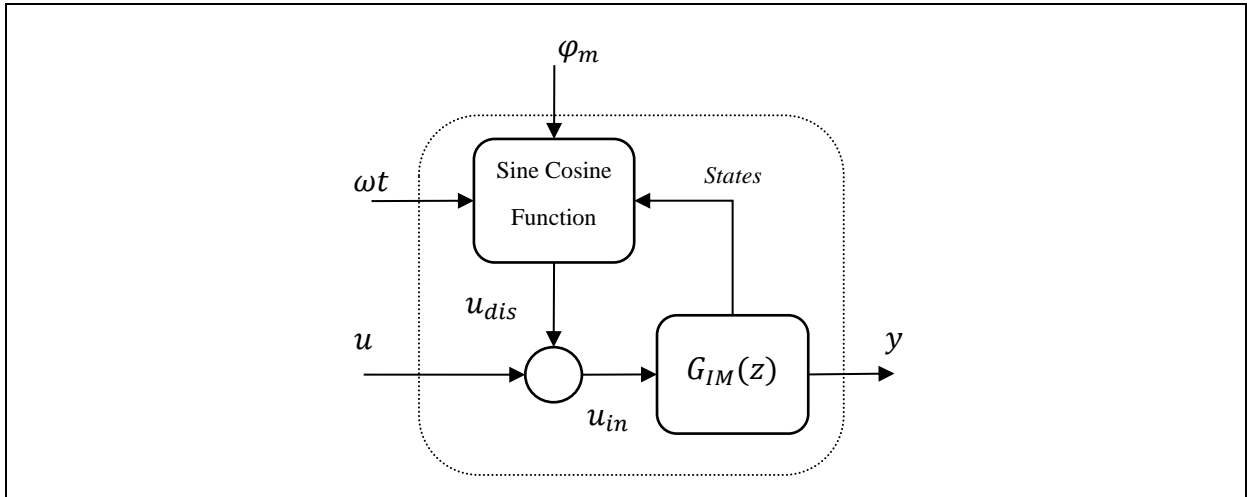


Figure 4.3: Identification model for externally and internally disturbed system.

The system input-output dynamics, shown in Figure 4.3, is defined as a discrete ARX, where it can be described by a discrete linear difference equation as

$$\begin{aligned}y(kT) &= a_1 y(T_s(k-1)) + a_2 y(T_s(k-2)) + \cdots + a_n y(T_s(k-n)) \\ &+ b_1 u_{in}(T_s(k-1)) + b_2 u_{in}(T_s(k-2)) + \cdots + b_n u_{in}(T_s(k-n)),\end{aligned}\quad (4.19)$$

where $u_{in}(kT_s)$ is the total input of the system dynamics, $y(kT_s)$ is the system output, n is order of the system dynamics, T_s is the sampling time and k is the sampling time sequence index (integer, $k = 0, 1, 2, \dots$). Therefore, the discrete-time is given by

$$t = kT_s. \quad (4.20)$$

In the following, the discrete-time will be represented only by the index k , but the sampling period T_s will appear only when it is needed in the computation.

The input of the dynamics is the manipulated variable plus the periodic disturbance as already defined in equation (4.2). And the input disturbance is the sum of the external and the internal disturbances as defined in equation (4.3), where the synchronization signal of the external and the internal disturbances $(\omega t, \varphi_m)$ are assumed to be known (measured, observed, estimated or predicted). The external and the internal disturbances are defined by their Fourier series expansions also as defined in equation (4.4). The internal disturbance is modeled to take into consideration all of spring, damper and moment of inertia angle dependent load elements as given in chapter 2.

Now, the shifted in sampling time sine and cosine vectors, that are representing the external and the internal periodic disturbance components, are defined as

$$\begin{aligned} \mathbf{sin}(\omega T_s(k-i)) &= [\sin(\omega T_s(k-i)) \quad \sin(2\omega T_s(k-i)) \quad \cdots \quad \sin(N_0\omega T_s(k-i))]; \\ \mathbf{cos}(\omega T_s(k-i)) &= [\cos(\omega T_s(k-i)) \quad \cos(2\omega T_s(k-i)) \quad \cdots \quad \cos(N_0\omega T_s(k-i))]; \end{aligned} \quad (4.21)$$

$$\begin{aligned} \mathbf{sin}(\varphi_m(k-i)) &= [\sin(\varphi_m(k-i)) \quad \sin(2\varphi_m(k-i)) \quad \cdots \quad \sin(N_1\varphi_m(k-i))]; \\ \mathbf{cos}(\varphi_m(k-i)) &= [\cos(\varphi_m(k-i)) \quad \cos(2\varphi_m(k-i)) \quad \cdots \quad \cos(N_1\varphi_m(k-i))], \end{aligned} \quad (4.22)$$

$$\begin{aligned} \mathbf{sin}(\varphi_m(k-i))y(k-i) &= [\sin(\varphi_m(k-i))y(k-i) \quad \sin(2\varphi_m(k-i))y(k-i) \\ &\quad \cdots \quad \sin(N_2\varphi_m(k-i))y(k-i)]; \\ \mathbf{cos}(\varphi_m(k-i))y(k-i) &= [\cos(\varphi_m(k-i))y(k-i) \quad \cos(2\varphi_m(k-i))y(k-i) \\ &\quad \cdots \quad \cos(N_2\varphi_m(k-i))y(k-i)], \end{aligned} \quad (4.23)$$

and

$$\begin{aligned} \mathbf{sin}(\varphi_m(k-i))y^2(k-i) &= [\sin(\varphi_m(k-i))y^2(k-i) \quad \sin(2\varphi_m(k-i))y^2(k-i) \\ &\quad \cdots \quad \sin(N_3\varphi_m(k-i))y^2(k-i)]; \\ \mathbf{cos}(\varphi_m(k-i))y^2(k-i) &= [\cos(\varphi_m(k-i))y^2(k-i) \quad \cos(2\varphi_m(k-i))y^2(k-i) \\ &\quad \cdots \quad \cos(N_3\varphi_m(k-i))y^2(k-i)]. \end{aligned} \quad (4.24)$$

Moreover, the multiplication vectors of the numerator polynomial parameters (b_i) of the system dynamics and sine and cosine factors ($b_i\alpha_{j,N_j}$) and ($b_i\beta_{j,N_j}$) respectively are defined as

$$\mathbf{b}_i\boldsymbol{\alpha}_j = \begin{bmatrix} b_i\alpha_{j,1} \\ b_i\alpha_{j,2} \\ \vdots \\ b_i\alpha_{j,N_j} \end{bmatrix}^T; \quad \mathbf{b}_i\boldsymbol{\beta}_j = \begin{bmatrix} b_i\beta_{j,1} \\ b_i\beta_{j,2} \\ \vdots \\ b_i\beta_{j,N_j} \end{bmatrix}^T. \quad (4.25)$$

Therefore, the data vector is defined as

$$\begin{aligned}
\mathbf{Z}(k) &= [y(k-1) \quad y(k-2) \quad \cdots \quad y(k-n) \\
&\quad u(k-1) \quad u(k-2) \quad \cdots \quad u(k-n) \\
&\vdots \quad \mathbf{sin}(\omega T_s(k-1)) \quad \mathbf{sin}(\omega T_s(k-2)) \quad \cdots \quad \mathbf{sin}(\omega T_s(k-n)) \\
&\vdots \quad \mathbf{cos}(\omega T_s(k-1)) \quad \mathbf{cos}(\omega T_s(k-2)) \quad \cdots \quad \mathbf{cos}(\omega T_s(k-n)) \\
&\vdots \quad \mathbf{sin}(\varphi_m(k-1)) \quad \mathbf{sin}(\varphi_m(k-2)) \quad \cdots \quad \mathbf{sin}(\varphi_m(k-n)) \\
&\vdots \quad \mathbf{cos}(\varphi_m(k-1)) \quad \mathbf{cos}(\varphi_m(k-2)) \quad \cdots \quad \mathbf{cos}(\varphi_m(k-n)) \\
&\vdots \quad \mathbf{sin}(\varphi_m(k-1))y(k-1) \quad \mathbf{sin}(\varphi_m(k-2))y(k-2) \quad \cdots \quad \mathbf{sin}(\varphi_m(k-n))y(k-n) \\
&\vdots \quad \mathbf{cos}(\varphi_m(k-1))y(k-1) \quad \mathbf{cos}(\varphi_m(k-2))y(k-2) \quad \cdots \quad \mathbf{cos}(\varphi_m(k-n))y(k-n) \\
&\vdots \quad \mathbf{sin}(\varphi_m(k-1))y^2(k-1) \quad \mathbf{sin}(\varphi_m(k-2))y^2(k-2) \quad \cdots \quad \mathbf{sin}(\varphi_m(k-n))y^2(k-n) \\
&\vdots \quad \mathbf{cos}(\varphi_m(k-1))y^2(k-1) \quad \mathbf{cos}(\varphi_m(k-2))y^2(k-2) \quad \cdots \quad \mathbf{cos}(\varphi_m(k-n))y^2(k-n)],
\end{aligned} \tag{4.26}$$

and the parameter vector

$$\begin{aligned}
\boldsymbol{\theta}^T &= [\quad a_1 \quad a_2 \quad \cdots \quad a_n \quad \vdots \quad b_1 \quad b_2 \quad \cdots \quad b_n \quad \vdots \\
&\quad \mathbf{b}_1\boldsymbol{\alpha}_0 \quad \mathbf{b}_2\boldsymbol{\alpha}_0 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\alpha}_0 \quad \vdots \quad \mathbf{b}_1\boldsymbol{\beta}_0 \quad \mathbf{b}_2\boldsymbol{\beta}_0 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\beta}_0 \quad \vdots \\
&\quad \mathbf{b}_1\boldsymbol{\alpha}_1 \quad \mathbf{b}_2\boldsymbol{\alpha}_1 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\alpha}_1 \quad \vdots \quad \mathbf{b}_1\boldsymbol{\beta}_1 \quad \mathbf{b}_2\boldsymbol{\beta}_1 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\beta}_1 \quad \vdots \\
&\quad \mathbf{b}_1\boldsymbol{\alpha}_2 \quad \mathbf{b}_2\boldsymbol{\alpha}_2 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\alpha}_2 \quad \vdots \quad \mathbf{b}_1\boldsymbol{\beta}_2 \quad \mathbf{b}_2\boldsymbol{\beta}_2 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\beta}_2 \quad \vdots \\
&\quad \mathbf{b}_1\boldsymbol{\alpha}_3 \quad \mathbf{b}_2\boldsymbol{\alpha}_3 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\alpha}_3 \quad \vdots \quad \mathbf{b}_1\boldsymbol{\beta}_3 \quad \mathbf{b}_2\boldsymbol{\beta}_3 \quad \cdots \quad \mathbf{b}_n\boldsymbol{\beta}_3 \quad].
\end{aligned} \tag{4.27}$$

So, the identification model output is computed by

$$y(k) = \mathbf{Z}(k)\boldsymbol{\theta}. \tag{4.28}$$

Now, instead of the model output, the measured (process) output is used in the data vector \mathbf{Z} , the discrete model, equation (4.28), becomes in form of linear in parameters, so that the RLS algorithm can be used to estimate the parameter vector. The disturbance parameters are calculated by averaging all of the available redundant parameters as

$$\alpha_{j,i} = \frac{1}{n} \sum_{k=1}^n \frac{b_k \alpha_{j,i}}{b_k}; \quad \beta_{j,i} = \frac{1}{n} \sum_{k=1}^n \frac{b_k \beta_{j,i}}{b_k}. \tag{4.29}$$

In the following subsections 4.3.1 and 4.3.2, simple cases of identification models are presented for the first order discrete dynamics with single harmonic external periodic disturbance, as well as the case for second order discrete dynamics with single harmonic internal periodic disturbance.

4.3.1 Discrete First Order Identification Model

Now, the case of first order ($n = 1$) identification model and external disturbance with one harmonic is assumed ($N_0 = 1, N_1 = N_2 = N_3 = 0$). So, the input signal is defined as

$$u_{in}(t) = u(t) + \alpha_{0,1} \sin(\omega t) + \beta_{0,1} \cos(\omega t), \tag{4.30}$$

This models an external disturbance of a sinusoidal signal or a mono frequency. Moreover the difference equation of the system is given by

$$\begin{aligned}
y(T_s k) &= a_1 y(T_s(k-1)) \\
&+ b_1 [u(T_s(k-1)) + \alpha_{0,1} \sin(T_s \omega(k-1)) + \beta_{0,1} \cos(T_s \omega(k-1))],
\end{aligned} \tag{4.31}$$

where this can be rewritten as

$$y(T_s k) = a_1 y(T_s(k-1)) + b_1 u(T_s(k-1)) + b_1 \alpha_{0,1} \sin(T_s \omega(k-1)) + b_1 \beta_{0,1} \cos(T_s \omega(k-1)), \quad (4.32)$$

or in the vector format

$$y(T_s k) = [y(T_s(k-1)) \quad u(T_s(k-1)) \quad \sin(T_s \omega(k-1)) \quad \cos(T_s \omega(k-1))] \begin{bmatrix} a_1 \\ b_1 \\ b_1 \alpha_{0,1} \\ b_1 \beta_{0,1} \end{bmatrix}, \quad (4.33)$$

where

$$\theta = [a_1 \quad b_1 \quad b_1 \alpha_{0,1} \quad b_1 \beta_{0,1}]^T, \quad (4.34)$$

The disturbance model parameter can be calculated as

$$\alpha_{0,1} = \frac{\theta(3)}{\theta(2)}; \quad \beta_{0,1} = \frac{\theta(4)}{\theta(2)}. \quad (4.35)$$

4.3.2 Discrete Second Order Identification Model

Alternatively, for the second order case ($n = 2$), the system is described by the following discrete transfer function

$$y(z) = \frac{b_1 z + b_2}{z^2 - a_1 z - a_2} [u(z) + u_{dis}(z)], \quad (4.36)$$

and the external disturbance has one harmonic ($N_0 = 0$, $N_1 = 1$, $N_2 = N_3 = 0$), so

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + b_1 u(k-1) + b_2 u(k-2) + b_1 \alpha_{1,1} \sin(\varphi_m(k-1)) + b_2 \alpha_{1,1} \sin(\varphi_m(k-2)) + b_1 \beta_{1,1} \cos(\varphi_m(k-1)) + b_2 \beta_{1,1} \cos(\varphi_m(k-2)), \quad (4.37)$$

or in vector format

$$y(k) = [y(k-1) \quad y(k-2) \quad u(k-1) \quad u(k-2) \quad \sin(\varphi_m(k-1)) \quad \sin(\varphi_m(k-2)) \quad \cos(\varphi_m(k-1)) \quad \cos(\varphi_m(k-2))] \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \\ b_1 \alpha_{1,1} \\ b_2 \alpha_{1,1} \\ b_1 \beta_{1,1} \\ b_2 \beta_{1,1} \end{bmatrix} \quad (4.38)$$

So, the disturbance model parameters are calculated by

$$\alpha_{1,1} = \frac{1}{2} \left[\frac{\theta(5)}{\theta(3)} + \frac{\theta(6)}{\theta(4)} \right] = \frac{1}{2} \left[\frac{b_1 \alpha_{1,1}}{b_1} + \frac{b_2 \alpha_{1,1}}{b_2} \right]; \quad (4.39)$$

$$\beta_{1,1} = \frac{1}{2} \left[\frac{\theta(5)}{\theta(3)} + \frac{\theta(6)}{\theta(4)} \right] = \frac{1}{2} \left[\frac{b_1 \beta_{1,1}}{b_1} + \frac{b_2 \beta_{1,1}}{b_2} \right].$$

4.4 Identification and Control Strategies

There are a lot of strategy variations and combinations of setting up the identification and the control procedure together. The simplest one, for example, is to make an offline open loop process parameter identification in order to get the disturbance model parameters and then use them to compensate the disturbance through the feed-forward control either with or without closing the loop, which is a true feed-forward control that causes no interaction to the closed loop (relative) stability or set point tracking design characteristics. This is good only if the disturbance parameters do not change with time at all. But if the parameters change from phase to phase or in certain operating regions or conditions, then the system parameters can be identified in these regions/conditions offline and make a correspondent controller design for each and use them as scheduled control strategy. This can be done provided that some offline data are available in advance.

For the case, the process is under closed loop control and it is not allowed to open the loop, the parameter identification can also be set up online while the process under the closed loop control. If the parameters converge, then the online parameter identification is set off and the converged parameters are used to compute the control law. This strategy would face the problem of identifiability in the closed loop since the input signal is directly correlated with the output signal, unless an external signal or the set point is used to provide enough excitation to the plant modes (Van den Hof 1998; Forssell and Ljung 1999). One alternative solution though is to use closed loop identification methods, e.g., indirect process identification by identifying the total closed loop system and then computing the process parameters from the identified closed loop system and known feedback controller, this method is computationally more extensive.

The second alternative is to use the adaptive mode in form of online identification and control by using *certainty equivalence principle* (Harris and Billings 1981). Under this condition the identifiability and identification conditions are more improved. Particularly, for the method of recursive least squares according to Åström and Wittenmark (1995), the problem of linear parameter dependence is solvable provided that the controller is a complex high order or time-varying, see also Appendix B.4. But caution should be taken at the initialization phase, where the initial parameters are to be carefully chosen to get a safe start by using good initial parameter guess. This could lead to use the adaptive dual control strategy (Filatov and Unbehauen 2000; Filatov and Unbehauen 2004), where the algorithm uses the identified controller parameters according to their parameter certainty and make sure to excite the plant in order to accelerate and to guarantee a correct and consistent parameter estimation.

However, as long as the algorithms of online identification and control are active, the true feed-forward conditions are not met, this means that the algorithms are working as a feedback controller that interacts with the pre-existing set-point tracking feedback controller. Nevertheless, if the parameters converge and the direct parameter adaptation is set off, then the algorithm becomes back again working as a true feed-forward controller.

In general, the stability analysis of the adaptive control strategies, which can be done for example, by using the direct method of Lyapunov (Åström and Wittenmark 1995), is directly linked to the parameter adaptation (identification) algorithms that are used to identify the

identification model parameters and their interaction with the controller, which are usually a non-trivial analytic problems, because they are time varying and nonlinear systems (Landau, Lozano and M'Saad 1998). Therefore, in this work instead, the developed control strategies will be intensively tested and validated in simulation, where they will be presented in the next chapter 5, as well as in real-time control with real hardware in the loop, where they will be presented in chapter 6.

5 SIMULATION EXAMPLES

In this chapter, the drive-load system is simulated by using its rigid and flexible models with external and internal (self-excited) periodic disturbances that have been developed in chapter 2. In these simulation examples, the system is constructed as an angular velocity servo control system to follow the given set point tracks. And because of the external and/or internal periodic disturbances, oscillations appear on the system output, therefore, the methods introduced in chapters 3 and 4 are applied in order to reject these periodic disturbances.

First, the problem of the set point tracking and periodic disturbance rejection is tried to be done by using feedback control only and as well as by incorporating a periodic disturbance internal model according to IMP (Francis and Wonham 1976). Then, the problem is done by using the developed strategy of feedback control for set point tracking and an add-on feed-forward for periodic disturbance compensation introduced in chapter 4. The strategy is tested on rigid and flexible drive-load system, and external as well as on internal disturbance, where according to the identification model discrete or continuous linear or nonlinear in parameters, the corresponding identification method is used as introduced in Appendix B.

5.1 Rigid Drive-load System

In this section, some simulation examples are presented on periodic disturbance rejection by using feedback and feed-forward control techniques. The source of periodic disturbances is external, internal or a combination of them. Furthermore, the following examples use the drive-load process given by the following differential equation, as already defined in subsection 2.2,

$$J\dot{\vartheta}(t) + D\vartheta(t) = T(t) + T_{dis}(t), \quad (5.1)$$

where the physical parameters are given by

$$J = 1 \text{ [kgm}^2\text{]}; D = 1 \text{ [Nms/rad]}; \quad (5.2)$$

5.1.1 Feedback Control

In this subsection, only a feedback control technique is used to guarantee the set point tracking demands and to reject the periodic disturbances. First, only a PI (Proportional-Integral) feedback controller is used, then a periodic disturbance rejection filter, by incorporating an internal model of the periodic disturbance, is used as an add-on to the PI feedback controller to reject the external periodic disturbance.

5.1.1.1 Simulation Example 1.1: Set point tracking and periodic disturbance rejection using only feedback control

The process of drive-load system, equations (5.1) and (5.2), is put under closed loop feedback control. Now, if it is required that the system has to be more slowly than it is, but keeping at the same time the steady state error to step input equal to zero. This can be achieved by changing the dynamics of the system by using a feedback controller (dynamic compensator), as already presented in chapter 3 Figure 3.1.

Now, the feedback controller is a PI controller with the parameter set A ($K_p = 0.1, K_I = 0.1$). This parameter set will yield set point and disturbance frequency responses shown in Figure 5.1. From the figure, it can be seen that the goal is achieved, since the system is now ten times slower and the step input steady state error is zero. But now, if the disturbance frequency response is carefully inspected, it shows a bad disturbance rejection in the region between 0.1 to 1 [rad/s]. So, this case has contradictory demands between slow response and good disturbance rejection which demands that the system has to have faster response or wider bandwidth instead. On the other hand, for more disturbance rejection the system should be made faster (wider bandwidth), for example, the following PI controller parameter set B ($K_p = 10.0, K_I = 0.1$) is considered. Figure 5.1 plots the frequency response of closed loop system for both parameter sets A and B, while Figure 5.2 plots the unit step response, when the external periodic torque disturbance is defined by ($T_{dis}(t) = \sin(t)$ [Nm]), where it can be seen that the case of parameter set B is much better than the case of parameter set A in terms of periodic disturbance rejection. But even in the better case B, the system output is still affected by the external periodic disturbance.

For perfect rejection, an infinite loop gain is needed at the frequency disturbance. This could be done by introducing an internal disturbance model to the feedback controller in form of “inverse” notch filter (periodic disturbance rejection filter) for one harmonic or repetitive control for infinite harmonics case. Therefore, in case A, an extra method of compensation should be used to cancel any undesirable disturbances especially outside the bandwidth of the working feedback controller. This could be done by using an internal model principle by adding an “inverse” notch filter, as in example 1.2, or feed-forward controller to the already existing set point tracking feedback controller, as in examples 1.3 and 1.4 respectively.

5.1.1.2 Simulation Example 1.2: Feedback with IMP to reject the periodic disturbances

The system is the same as in example 1.1 and the feedback controller is a PI with the parameter set A. Now, the Periodic Disturbance Rejection Filter (PDRF) is defined by

$$G_N(s) = \frac{K_d \omega_d^2}{s^2 + 2 \xi_d \omega_d s + \omega_d^2}, \quad (5.3)$$

where $K_d = 0.5$, $\xi_d = 0$ and $\omega_d = 1$ [rad/s]. The PDRF is implemented by using variant A and B, as previously explained in subsection 3.1.1 and shown in Figure 3.2. Figure 5.3 and Figure 5.4 show the frequency and the step responses for only PI feedback controller with parameter set A, PI with PDRF of variant A and PI with PDRF of variant B cases.

Actually, both of the variants have the same disturbance rejection characteristics. The difference is only in the set point response characteristics. Variant A amplifies the set point at the designed rejection frequency, which makes this variant the choice when the disturbance frequency is inside the demanded closed loop system bandwidth. While variant B suppresses the set point at the designed rejection frequency, this in turn, makes this variant the choice when the disturbance frequency is outside the demanded closed loop system bandwidth, which is the case in this example. But, both of the variants A and B affect the closed loop transient characteristics.

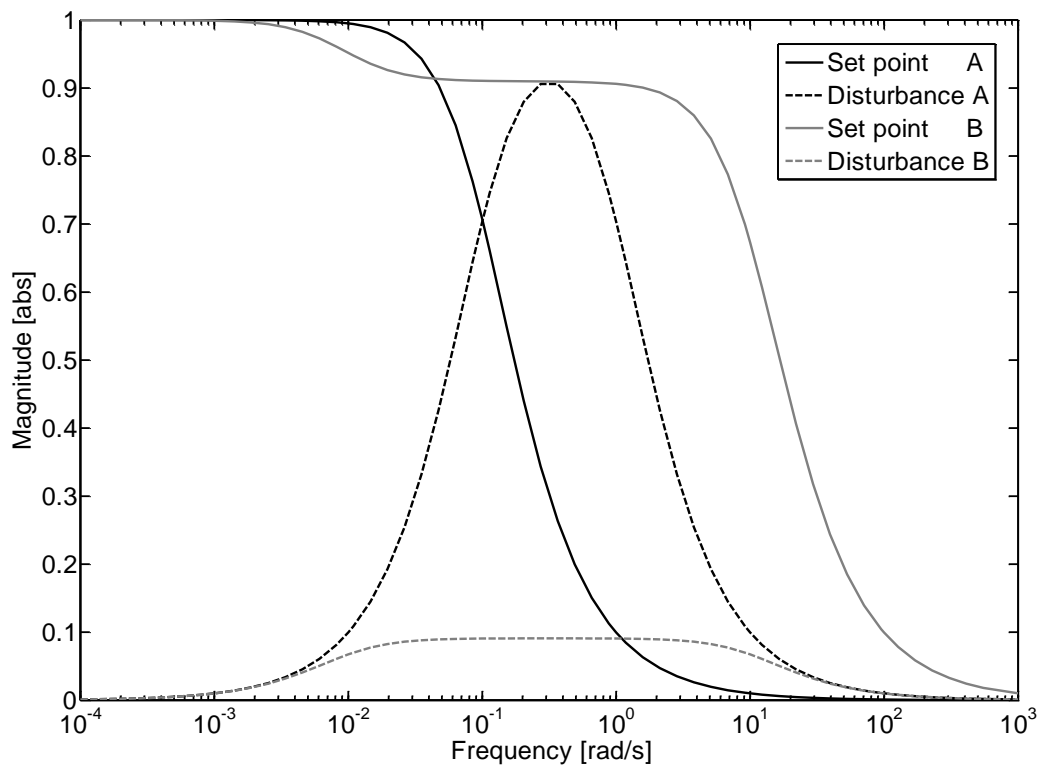


Figure 5.1: Example 1.1: Closed loop frequency response to the set point and the disturbance for the parameter sets A and B of the PI feedback controller.

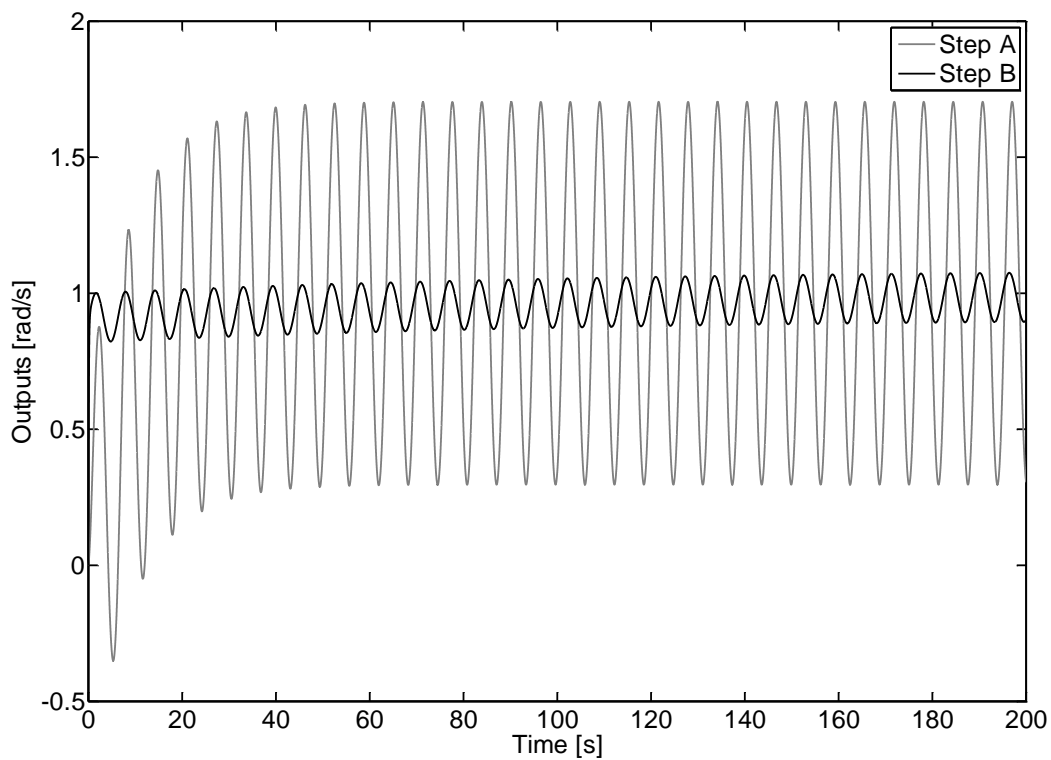


Figure 5.2: Example 1.1: Closed loop set point step response for parameter sets A and B of the PI feedback controller.

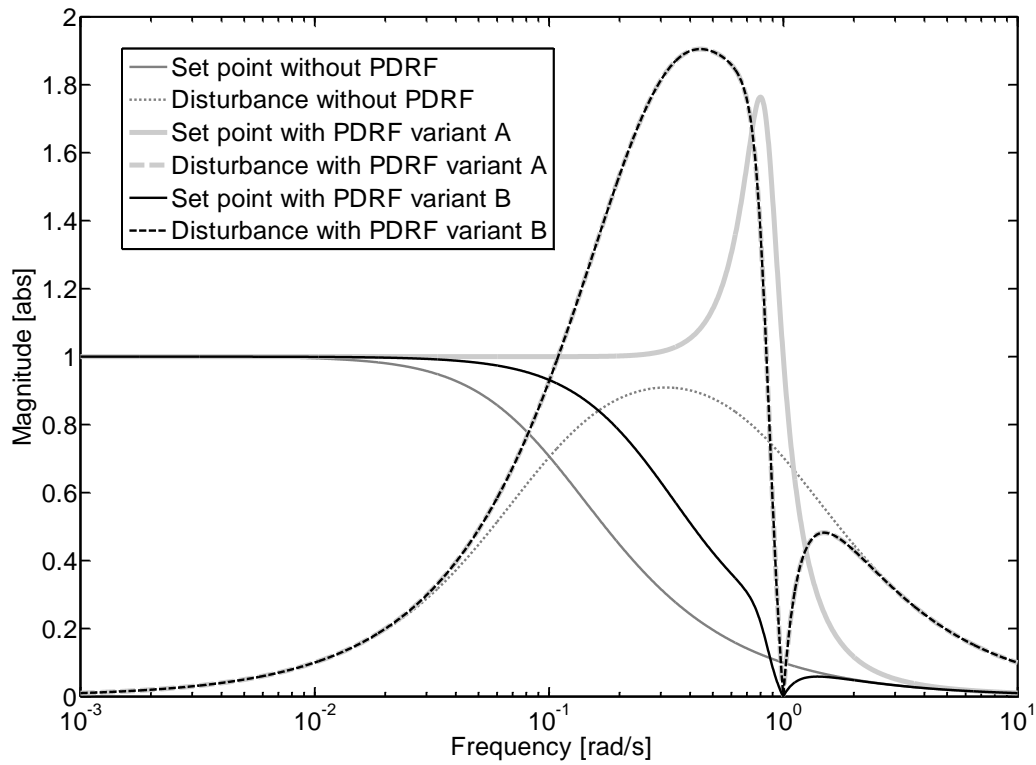


Figure 5.3: Example 1.2: Set point and disturbance frequency response of closed loop system for only PI controller (parameter set A), with PDRF variant A and with PDRF variant B.

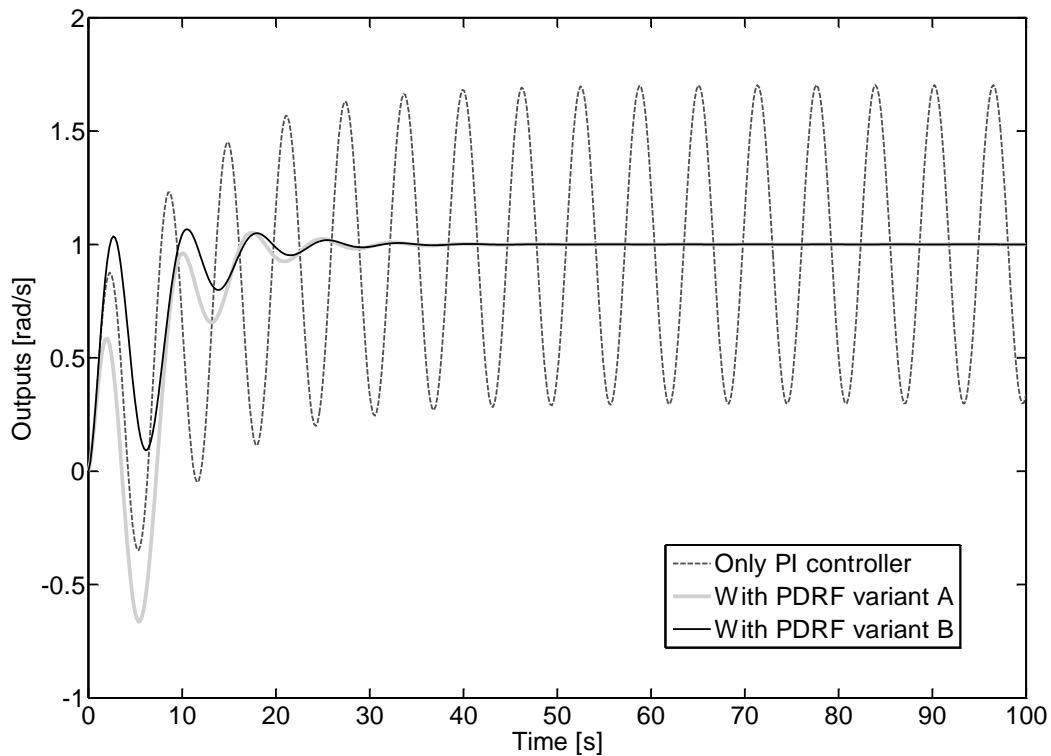


Figure 5.4: Example 1.2: Set point step response of closed loop system for only PI (parameter set A), with PDRF variant A and with PDRF variant B.

5.1.2 Feedback Feed-forward Control

Now in this subsection, the strategy of feedback control for set point tracking and (adaptive) feed-forward control for periodic disturbance rejection, introduced in section 4.2, is used to compensate the external single and multi-harmonic disturbances as well as the internal periodic disturbances induced from nonlinear angle dependent load elements.

5.1.2.1 Simulation Example 1.3: Feedback for set point tracking and feed-forward for periodic disturbance compensation

In this example, the drive-load model is as already defined by equations (5.1) and (5.2) with external periodic torque disturbance as already defined in example 1.1.

Now, the identification model structure as presented in subsection 4.2.1, the dynamic and the disturbance parts, is defined by

$$\begin{aligned} \dot{y}(t) &= ay(t) + b[u(t) + u_{dis}(t)]; \\ u_{dis}(t) &= \alpha_{0,1} \sin(t) + \beta_{0,1} \cos(t), \end{aligned} \quad (5.4)$$

with the parameter vector

$$\boldsymbol{\theta} = [a \ b \ \alpha_{0,1} \ \beta_{0,1}]^T. \quad (5.5)$$

The parameter vector is identified by using the parametric output error method, since the identification model is in the form of nonlinear in parameters, as introduced in details in Appendix B.3. The identification procedure is started with initial parameter vector of $[-2, 2, 0, 0]^T$, initial covariance matrix of $1000\mathbf{I}$, forgetting factor of 0.999 and sampling time of 1 [s]. The open loop identification session is shown in the Figure 5.5, and the yielded final parameter vector $[-1, 1, 1, 0]^T$ which it has converged to the true process parameters. The system is identified in open loop and then the identified disturbance parameters are used in the feed-forward controller to compute the anti-disturbance signal to compensate the external periodic disturbance. Figure 5.6 shows the feed-forward disturbance compensation in open loop, while Figure 5.7 in closed loop under the PI feedback controller with parameter set A given in example 1.1. The online identification and control of the process in closed loop, the adaptive session, is also done and presented in Figure 5.8. As can be seen from the figures, the algorithm was capable of identifying and compensating the external periodic disturbances both in open loop as an (offline/online) identification and indirect adaptation algorithm to compensate the periodic disturbance in open/closed loop, as well as in closed loop as an online identification and direct adaptation algorithm.

An advantage of this algorithm is that it becomes a pure feed-forward control when the parameter identification is done and so the direct adaptation is set off. In this situation, the algorithm does not affect the closed loop dynamics anymore, where they can then be freely designed to fulfil the set point tracking design demands separately.

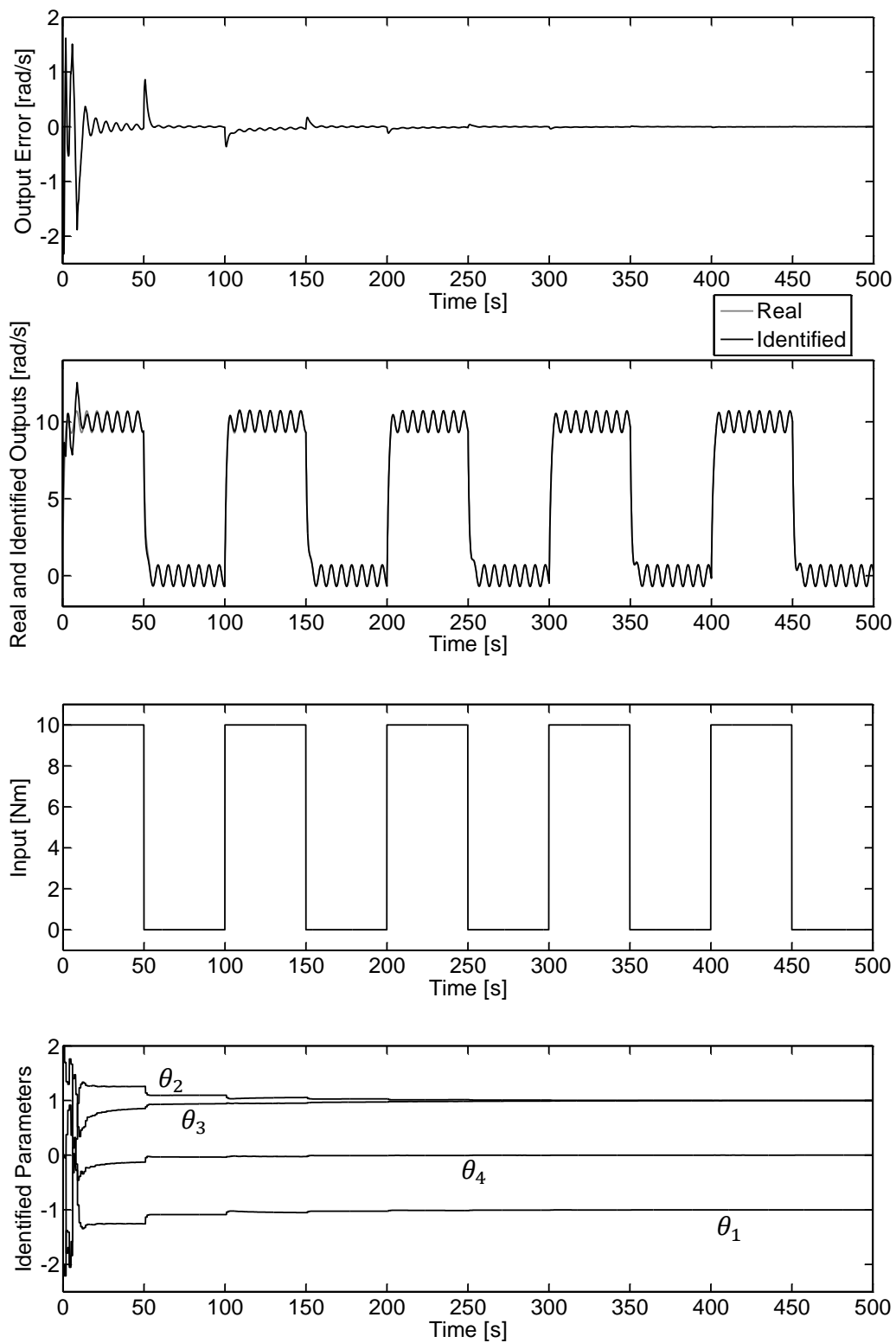


Figure 5.5: Example 1.3: Open-loop identification run.

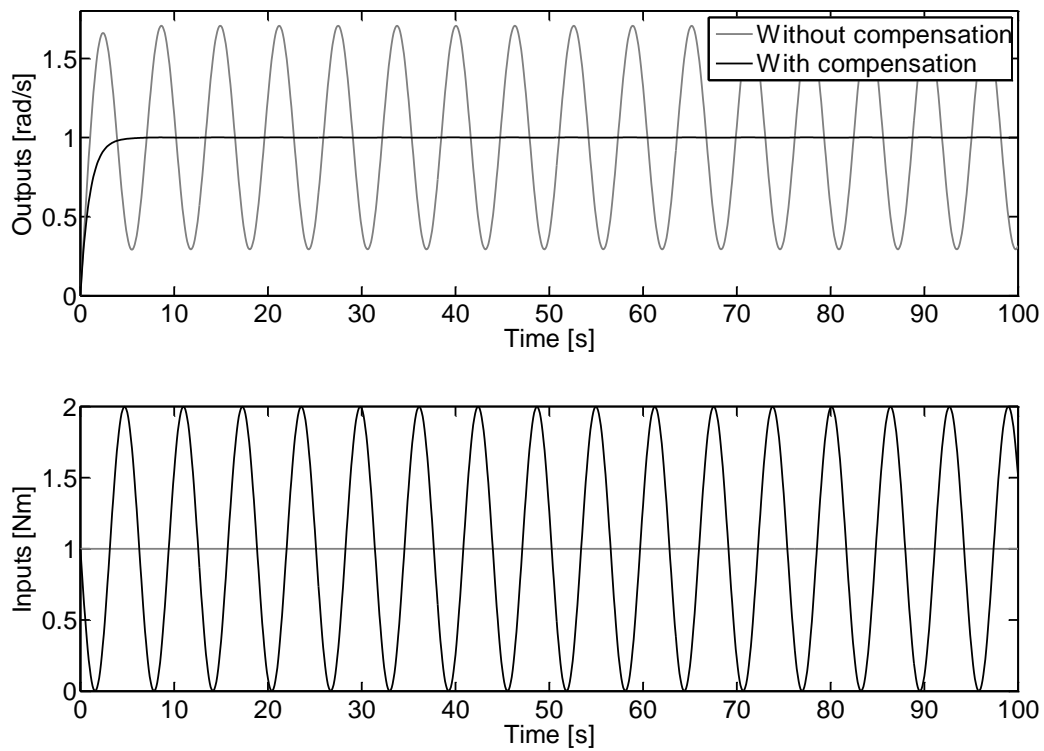


Figure 5.6: Example 1.3: Compensation in open loop.

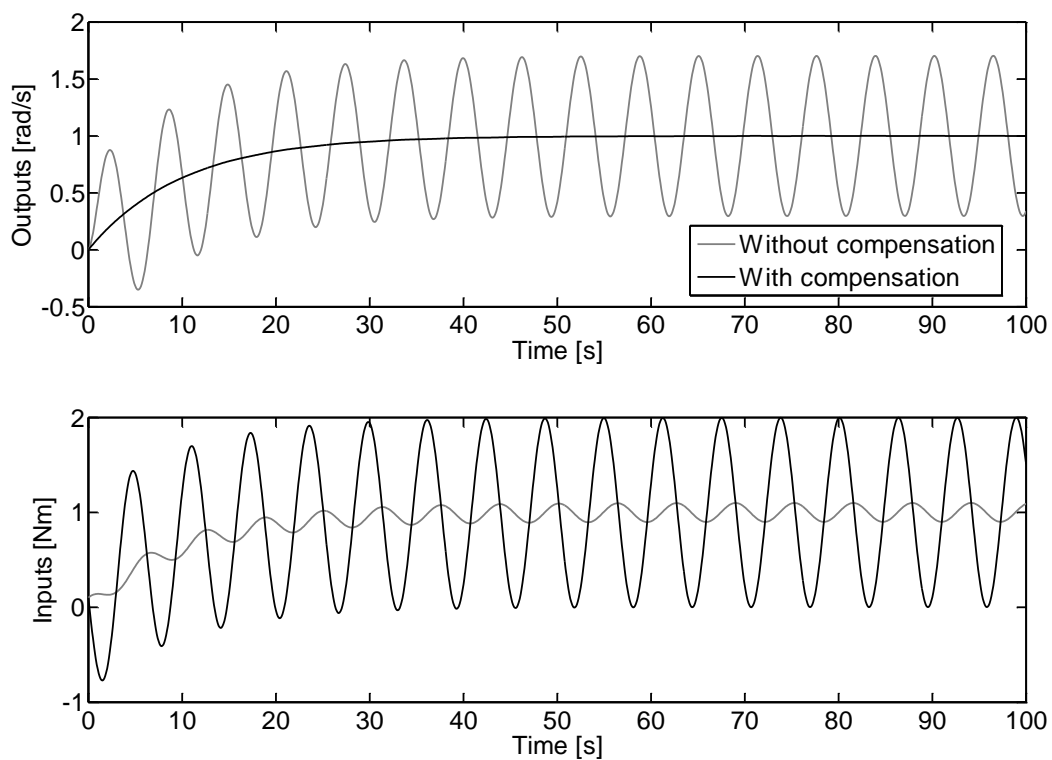


Figure 5.7: Example 1.3: Compensation in closed loop under PI controller.

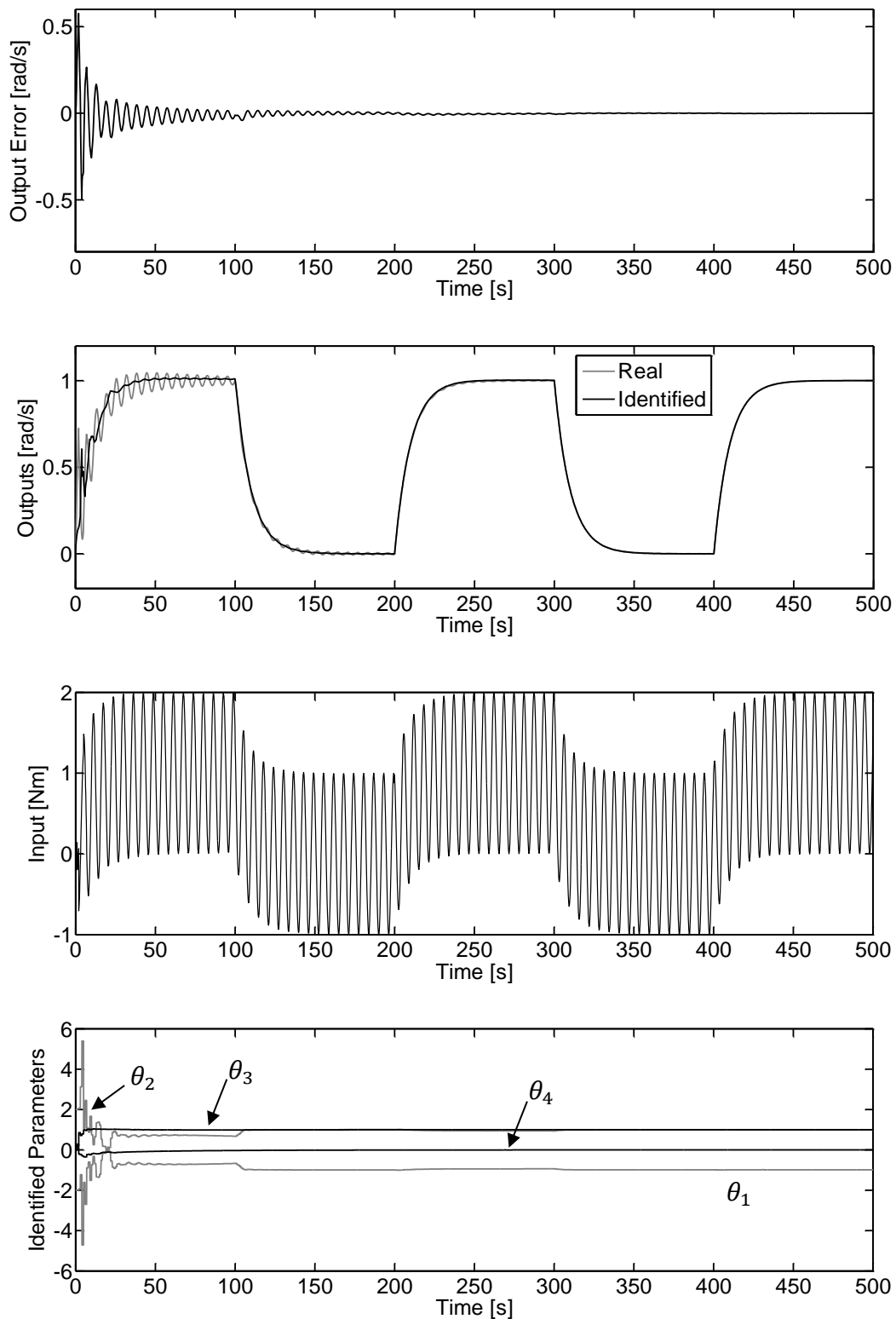


Figure 5.8: Example 1.3: Closed loop adaptive run.

5.1.2.2 Simulation Example 1.4: Multi-harmonic external periodic disturbance

In this simulation example, the process physical parameters and the closed loop controller parameters are the same as given in the previous examples, except that the external periodic torque disturbance is defined as

$$\begin{aligned} T_{dis}(t) = & \sin(t) + 0.5 \sin(2t) + 0.2 \cos(2t) \\ & + 0.2 \sin(3t) + 0.1 \cos(3t) + 0.1 \sin(4t) \\ & + 0.4 \cos(4t) + 0.01 \sin(5t) + 0.5 \cos(5t) \text{ [Nm]}. \end{aligned} \quad (5.6)$$

Therefore, the identification model is accordingly defined as

$$\begin{aligned} \dot{y}(t) = & ay(t) + b[u(t) + u_{dis}(t)]; \\ u_{dis}(t) = & \sum_{i=1}^{N_0=5} \alpha_{0,i} \sin(it) + \beta_{0,i} \cos(it), \end{aligned} \quad (5.7)$$

with the parameter vector

$$\boldsymbol{\theta} = [a \ b \ \alpha_{0,1} \ \alpha_{0,2} \ \alpha_{0,3} \ \alpha_{0,4} \ \alpha_{0,5} \ \beta_{0,1} \ \beta_{0,2} \ \beta_{0,3} \ \beta_{0,4} \ \beta_{0,5}]^T. \quad (5.8)$$

Now, the parameter vector is identified by using the output error method with initial parameter vector $[-2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, covariance matrix = $100000\mathbf{I}$, forgetting factor 0.99 and sampling time of 1 [s]. The number of considered harmonics are 5 and the principal disturbance frequency is 1[rad/s]. The system is identified in open loop, see Figure 5.9, which ended with the final parameter vector $[-1.0002, 1.0002, 1.0002, 0.4990, 0.1985, 0.0983, 0.0112, -0.0009, 0.1993, 0.1002, 0.4015, 0.5024]^T$. Then, the identified disturbance parameters are used in the feed-forward controller to compute the anti-disturbance signal to compensate the disturbance. Figure 5.10 shows the feed-forward disturbance compensation in open loop, while Figure 5.11 in closed loop by using the PI feedback controller with parameter set A given in example 1.1. Figure 5.12 presents the adaptive session, where the online direct identification of the process in the closed loop and direct adaptation of the feed-forward controller algorithm are implemented. Also in this experiment, the algorithm was capable of identifying and compensating the external multi-harmonic disturbances successfully.

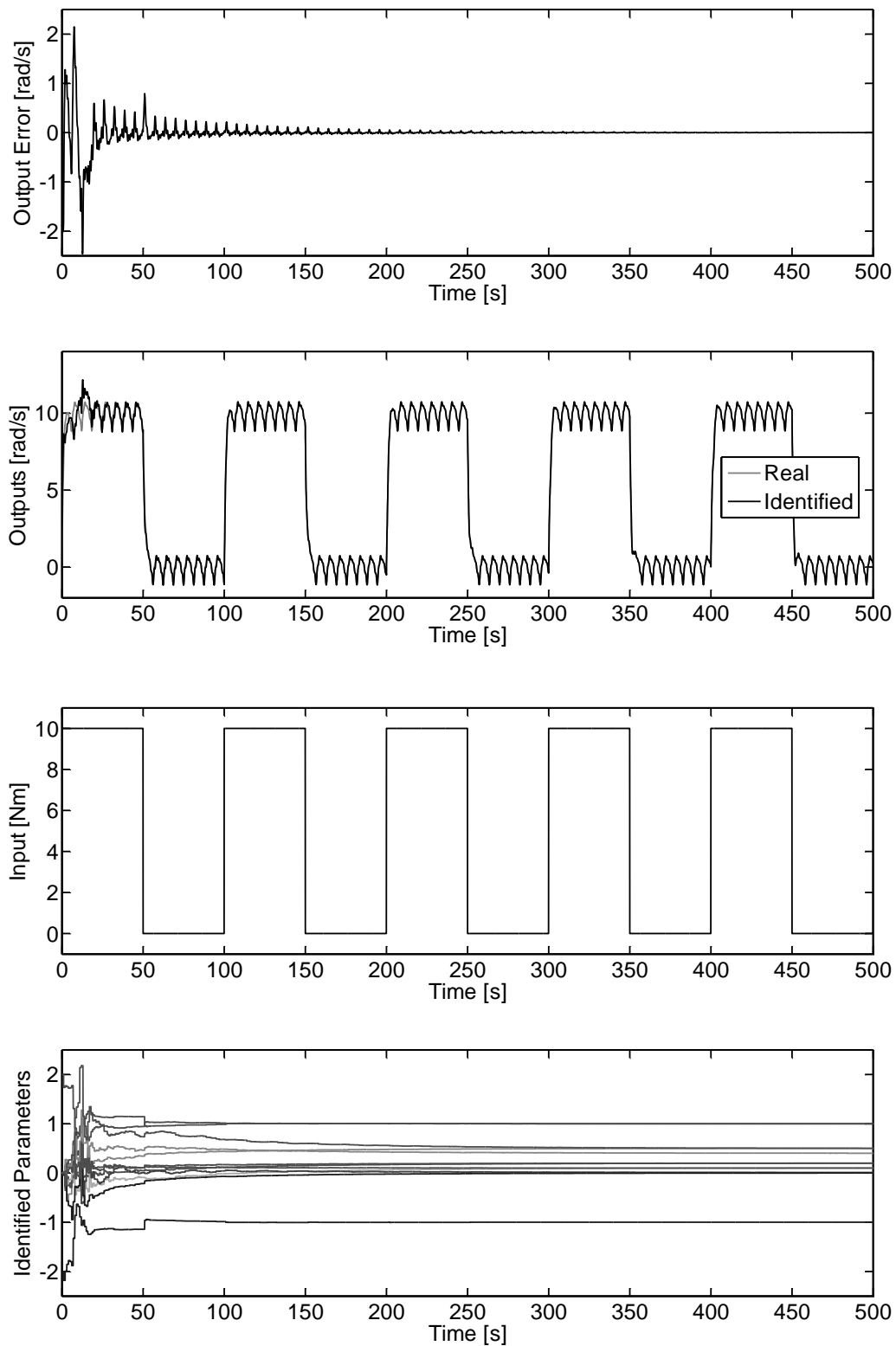


Figure 5.9: Example 1.4: 5-harmonics open loop identification run.

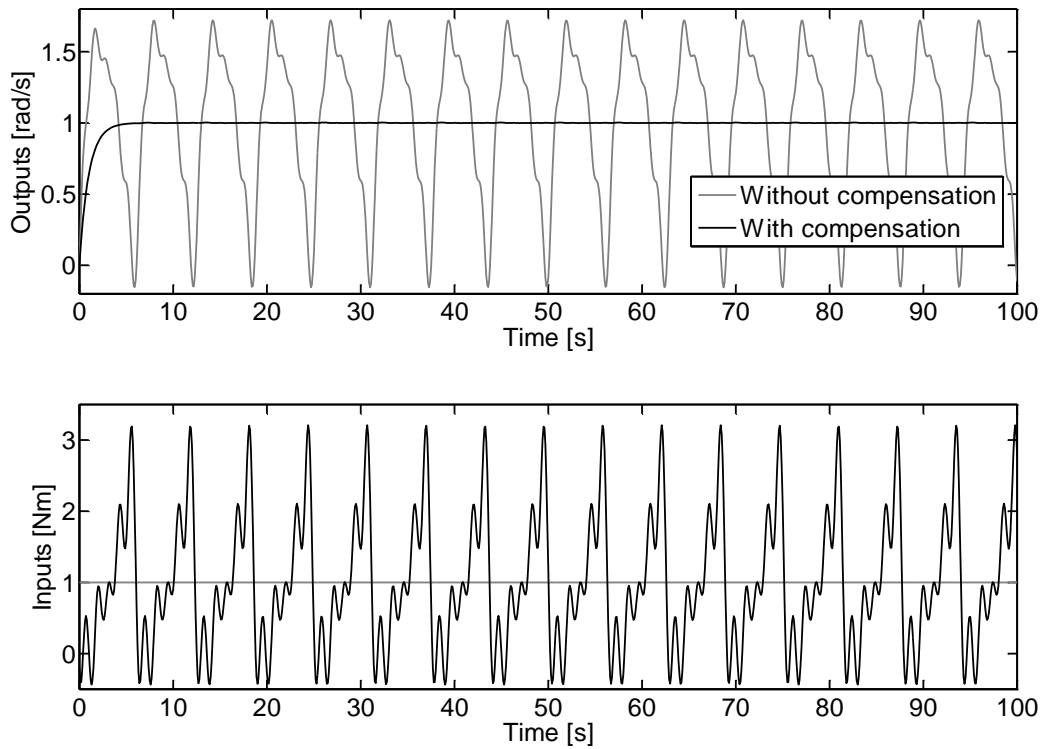


Figure 5.10: Example 1.4: 5-Harmonics open loop compensation.

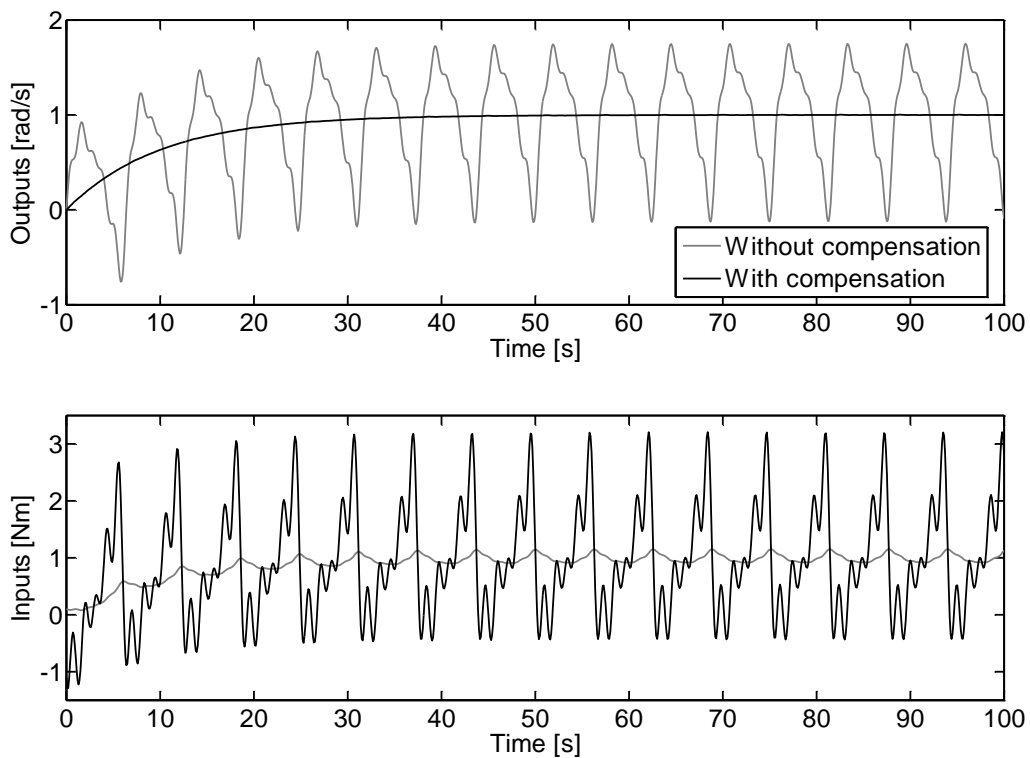


Figure 5.11: Example 1.4: 5-Harmonics closed loop compensation.

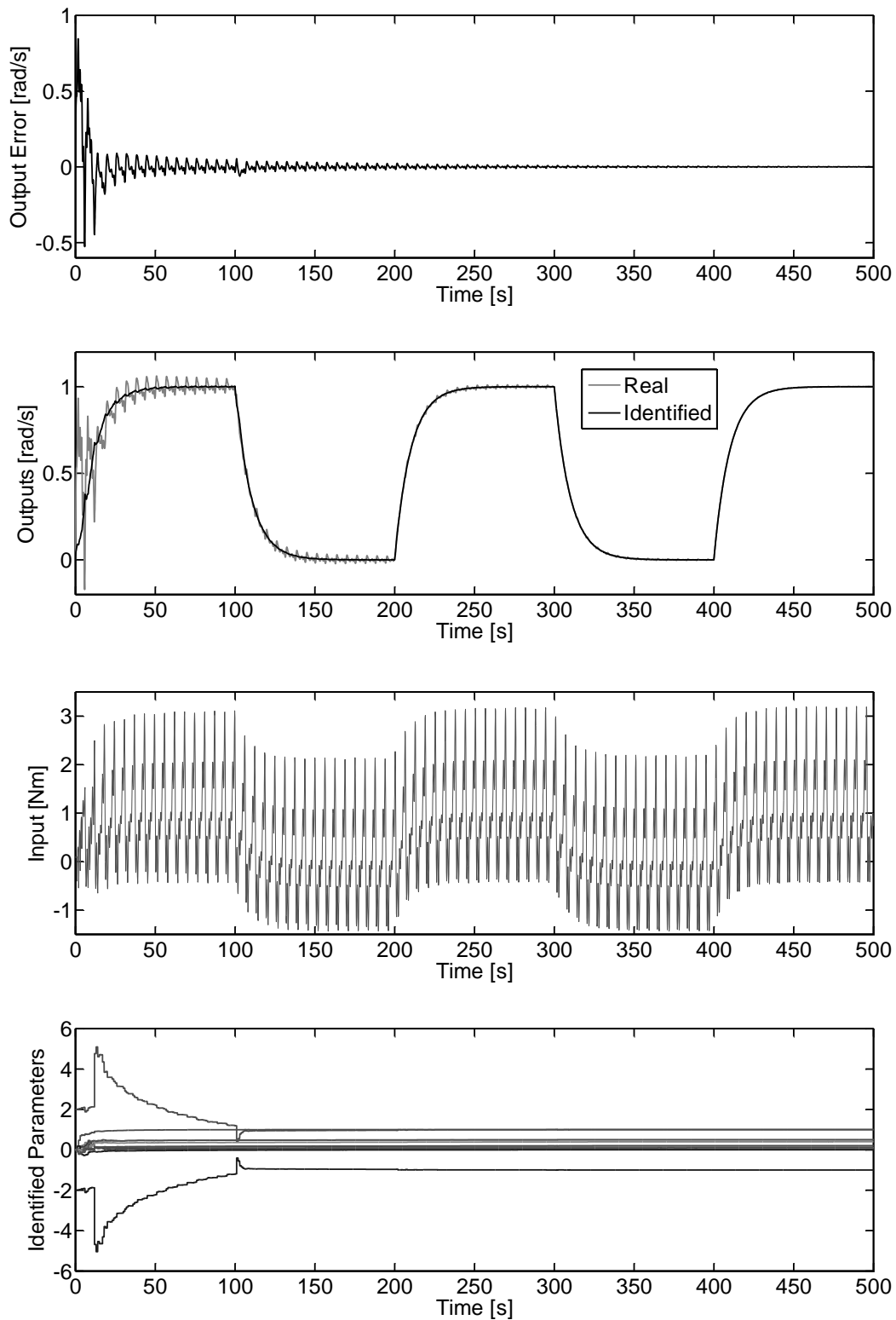


Figure 5.12: Example 1.4: Adaptive run.

5.1.2.3 Simulation Example 1.5: Externally and internally (self-excited oscillations) periodically disturbed system

Now the drive-load system is externally as well as internally disturbed. The external periodic torque disturbance source is defined by

$$T_{dis}(t) = 0.5 \sin(t) \text{ [Nm]}. \quad (5.9)$$

While the internal periodic disturbance is caused by the angle dependent spring, damper and moment of inertia load elements defined as

$$\begin{aligned} K(\varphi) &= 0.5 \sin(\varphi) \text{ [Nm]}; \\ D(\varphi) &= 1 + 0.5 \sin(\varphi) \text{ [Nms/rad]}; \\ J(\varphi) &= 1 + 0.05 \sin(\varphi) \text{ [kgm}^2\text{]}. \end{aligned} \quad (5.10)$$

The identification model dynamic part is a first order continuous and the disturbance part has external periodic disturbance and angle dependent internal disturbance of zero, first and second degree dependence of identification model angular velocity, plus first degree dependence of input signal. Therefore, the identification model is accordingly constructed by

$$\dot{y}(t) = ay(t) + b[u(t) + u_{dis}(t)], \quad (5.11)$$

where the input disturbance is defined by

$$\begin{aligned} u_{dis}(t) &= \sum_{i=1}^{N_0} [\alpha_{0,i} \sin(i\omega t) + \beta_{0,i} \cos(i\omega t)] \\ &+ \sum_{i=1}^{N_1} [\alpha_{1,i} \sin(i\varphi_m) + \beta_{1,i} \cos(i\varphi_m)] \\ &+ \sum_{i=1}^{N_2} [\alpha_{2,i} \sin(i\varphi_m) + \beta_{2,i} \cos(i\varphi_m)] y(t) \\ &+ \sum_{i=1}^{N_3} [\alpha_{3,i} \sin(i\varphi_m) + \beta_{3,i} \cos(i\varphi_m)] y^2(t) \\ &+ \sum_{i=1}^{N_4} [\alpha_{4,i} \sin(i\varphi_m) + \beta_{4,i} \cos(i\varphi_m)] u(t). \end{aligned} \quad (5.12)$$

Since the identification model has a form of nonlinear in parameters, therefore, the parameters of the identification model are identified by using the parametric output error method with the initial dynamic part parameters $[-2, 2]^T$ and the disturbance part started with zeros as an initial guess. The identification routine run with a sampling time of 1[s], the considered numbers of harmonics are $(N_0 = 2, N_1 = 2, N_2 = 2, N_3 = 2$ and $N_4 = 2)$, the covariance matrix is started by $10\mathbf{I}$ and the forgetting factor by 0.99.

At the beginning, an open loop identification run is done as shown in Figure 5.13. Then, the identified disturbance parameters are used in a feed-forward controller to compensate the periodic disturbances in the open loop operation; this run is plotted in Figure 5.14. Moreover, the feed-forward controller with the identified parameters is also applied to compensate the periodic disturbances in the closed loop operation under the PI controller with parameter set A, as given in example 1.1, this run is presented in Figure 5.15. Furthermore, an adaptive run, online and direct identification of the process model in closed loop and direct parameter adaptation of the feed-forward controller, is also done and presented in Figure 5.16.

5.1.3 Comments on the Rigid Drive-load System

So, the implementation of the algorithm can be done offline by doing offline identification with some previously recorded input-output data or online identification with an operating process. Consequently, as the parameters converge, the identified periodic disturbance parameters can be used to construct a “true” feed-forward controller to compensate the unwanted periodic disturbance in open as well as in closed loop. This is reasonable when the process and the disturbance parameters do not change with time. Otherwise, if the parameters are time-varying or the identification model is modeled locally while the process operates globally, then the online (adaptive) identification and control is the better way, but this makes the feed-forward controller a pure feedback controller again.

Also for the adaptive mode, a single adaptive run can be done at the system start-up to optimize the parameters and then the adaptation set off when the parameters converge. Moreover, the adaptation can be reassumed again when the performance deteriorates as the process behavior changes. Furthermore, a consistent adaptation can be used to track the parameter change, especially, when a local identification model is used and the operating point changes continuously. However, the activation of the adaptation is very important in order to achieve good identifiability conditions in closed loop operation, see Appendix B.4.

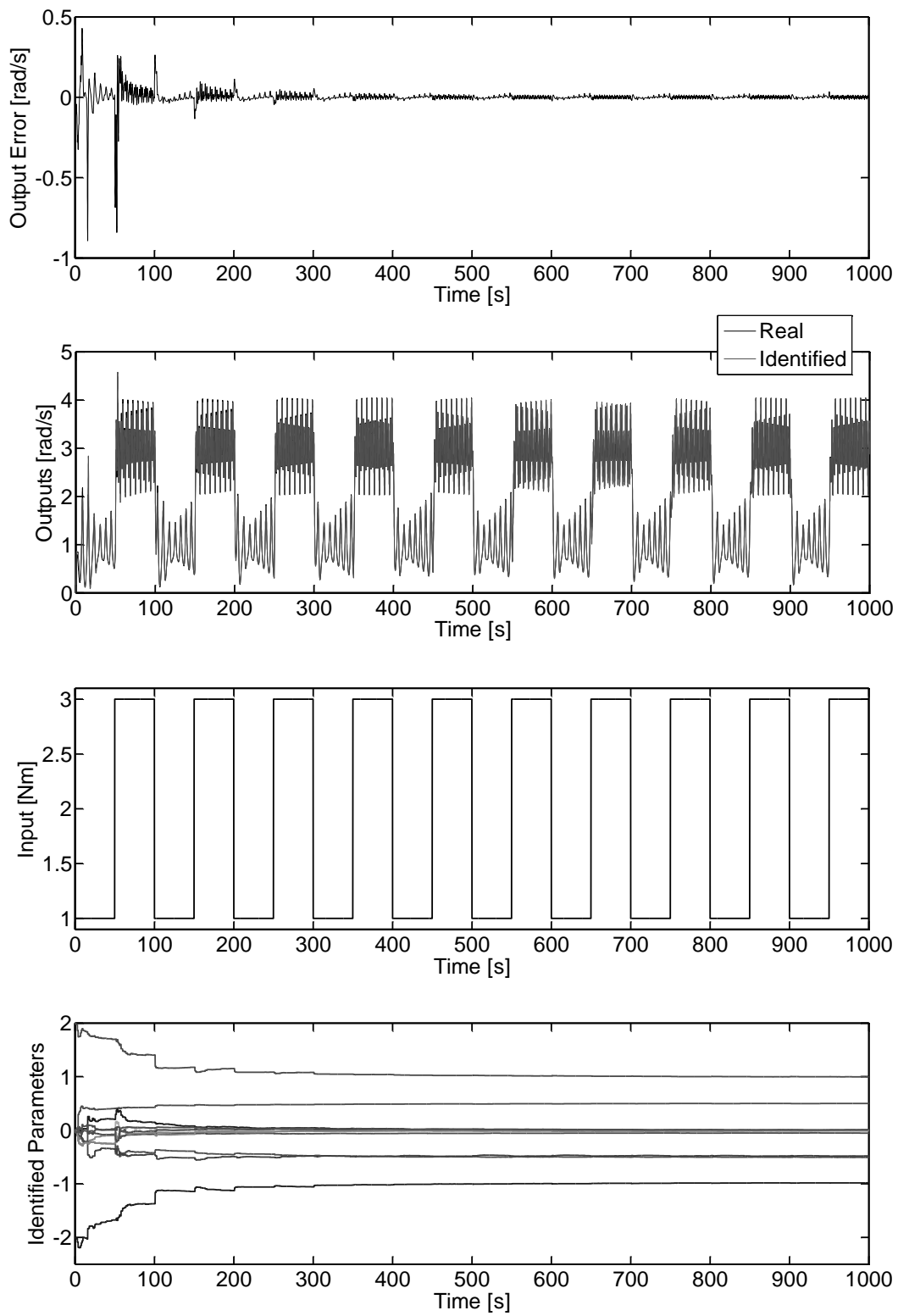


Figure 5.13: Example 1.5: Open loop identification.

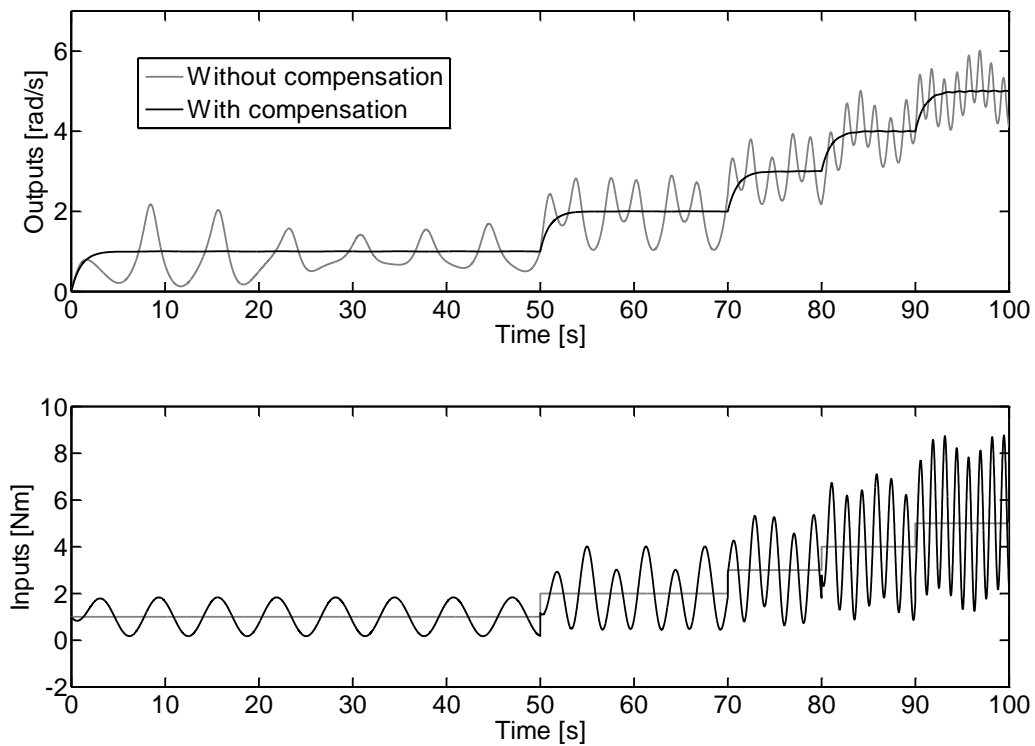


Figure 5.14: Example 1.5: Periodic disturbance compensation in open loop.

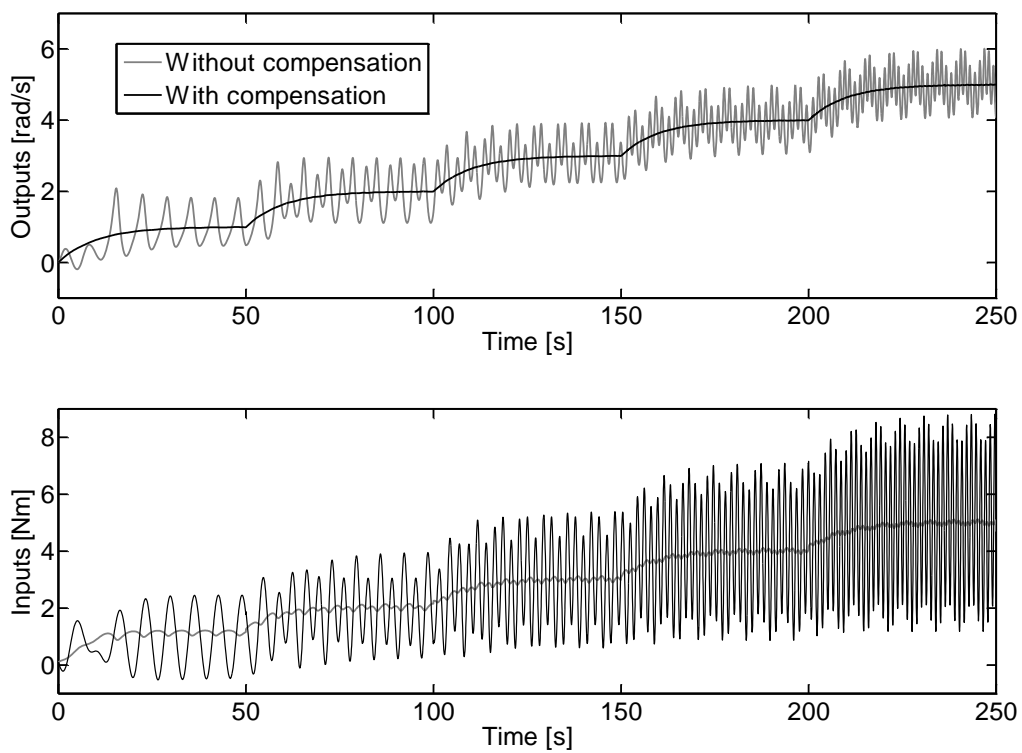


Figure 5.15: Example 1.5: Periodic disturbance compensation in closed loop.

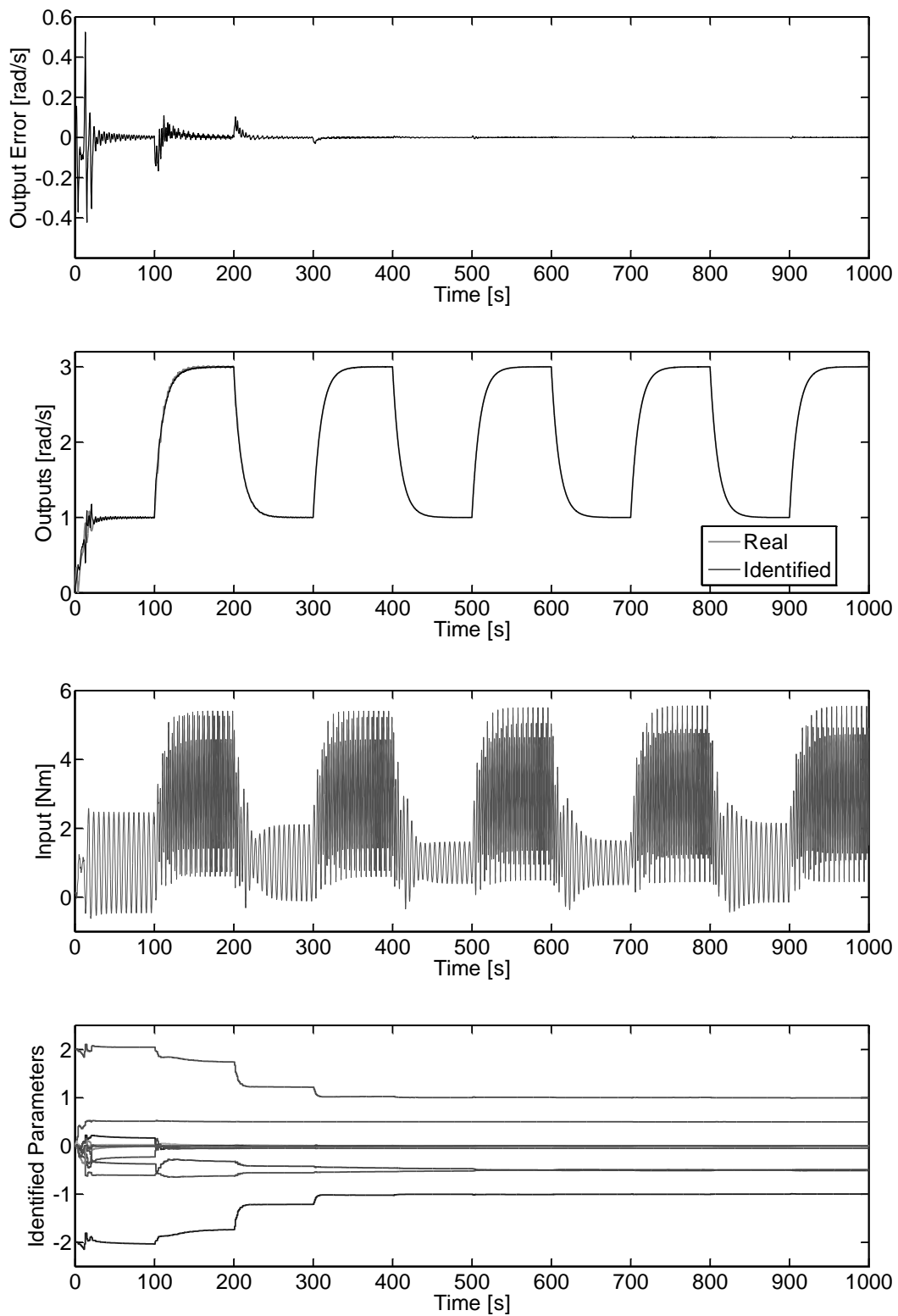


Figure 5.16: Example 1.5: Adaptive run.

5.2 Flexible Drive-load System

In this section, the drive-load system is modeled and simulated as flexible system. The flexible 2DOF model is already given in section 2.3 by equations (2.51)-(2.62).

5.2.1 Externally Excited Periodic Disturbance Compensation

In this subsection, the drive-load system is driven to its physical and operational limits, by reducing the mechanical link material relative to the load inertia, which is done either by reducing the stiffness and the damping factors of the mechanical link or by increasing the load inertia. The operational limit is when the drive-load system is driven to and beyond its resonance region.

In the following, the first example operates the flexible drive-load system with external periodic disturbance in the region far before its resonance frequency, where its behavior is the same as a rigid body one which can be described by 1DOF system. In the second example, the drive-load system is more flexible than in the first one but still operating in the rigid region. The third and fourth examples are operating at the flexible regions at their resonance frequencies.

5.2.1.1 Simulation Example 2.1: Rigid link case

In this example, the principle idea of the algorithm is presented to demonstrate that the algorithm will identify and approximate the system dynamics and the disturbance model even if the real system has a different dynamics and structure. The system has a rigid link between the drive and the load sides. The system physical parameters are given as following:

$$\begin{aligned}
 \text{Drive side:} \quad & J_D = 0.1 \text{ [kgm}^2\text{]}; \\
 & d_D = 0.1 \text{ [Nms/rad]}; \\
 \text{Mechanical link:} \quad & K_R = 100000.0 \text{ [Nm/rad]}; \\
 & D_R = 1.0 \text{ [Nms/rad]}; \\
 & g = 1; \\
 \text{Load side:} \quad & J_L = 1.0 \text{ [kgm}^2\text{]}; \\
 & d_L = 1.0 \text{ [Nms/rad]}.
 \end{aligned} \tag{5.13}$$

This gives the following transfer function of the drive-side T_D and the load-side T_L torques to the load-side angular velocity $\dot{\phi}_L$ as

$$\begin{aligned}
 \dot{\phi}_L(s) = & \frac{10(s + 100000)}{s^3 + 13s^2 + 1.1 \cdot 10^6 s + 1.1 \cdot 10^6} T_D(s) \\
 & + \frac{s^2 + 11s + 10^6}{s^3 + 13s^2 + 1.1 \cdot 10^6 s + 1.1 \cdot 10^6} T_L(s).
 \end{aligned} \tag{5.14}$$

The external periodic disturbance torque acting on the load side defined by

$$T_L(t) = 5 \cdot \sin(10t) \text{ [Nm]}. \tag{5.15}$$

The system dynamics have two complex conjugate poles at $(-6.0 + i1050, -6.0 - i1050)$ and one real pole at (-1) . Figure 5.17 shows the frequency response of the load-side angular velocity output to the drive torque and the load torque inputs. It is clear from the poles and the frequency response that, there is only one dominant pole at (-1) , and so the behavior of the system can be very good approximated by a first order system as

$$\phi_L(s) = \frac{0.9091}{s + 1} [T_D(s) + T_L(s)]. \quad (5.16)$$

Moreover, this can also be seen from the frequency response shown in Figure 5.17.

Therefore, the identification model is constructed as first order dynamic part and one harmonic external disturbance part as following

$$\begin{aligned} \dot{y}(t) &= ay(t) + b[u(t) + u_{dis}(t)]; \\ u_{dis}(t) &= \alpha_{0,1} \sin(\omega t) + \beta_{0,1} \cos(\omega t), \end{aligned} \quad (5.17)$$

with the parameter vector

$$\boldsymbol{\theta} = [a \ b \ \alpha_{0,1} \ \beta_{0,1}]^T. \quad (5.18)$$

The identification model parameter vector is identified by using the parametric output error method, because the identification model has a form of nonlinear in parameters, with initial parameter vector $[-2, 2, 0, 0]^T$, initial covariance matrix of $100\mathbf{I}$, forgetting factor of 0.99, sampling time of 1 [s] and external disturbance frequency 10 [rad/s]. An open loop identification session is done as shown in Figure 5.19 that yields the final parameter vector $[-1.0000, 0.9091, 4.9998, 0.0000]^T$, which shows that the parameters are perfectly converged to the disturbed first order dynamics, this can be seen in Figure 5.18, which plots the bode diagram of both the real and the identified dynamics. Figure 5.20 shows the open loop disturbance compensation using the disturbance parameters identified in the open loop session. Figure 5.21 shows the application of the feed-forward compensation using the disturbance model while the disturbed system is operating under negative feedback PI controller with $(K_P = 0.1, K_I = 0.1)$. Figure 5.22 presents the adaptive run, where it demonstrates clearly, as the identification model parameters converge, the applicability of the algorithm to identify and to compensate the periodic disturbances in the closed loop.

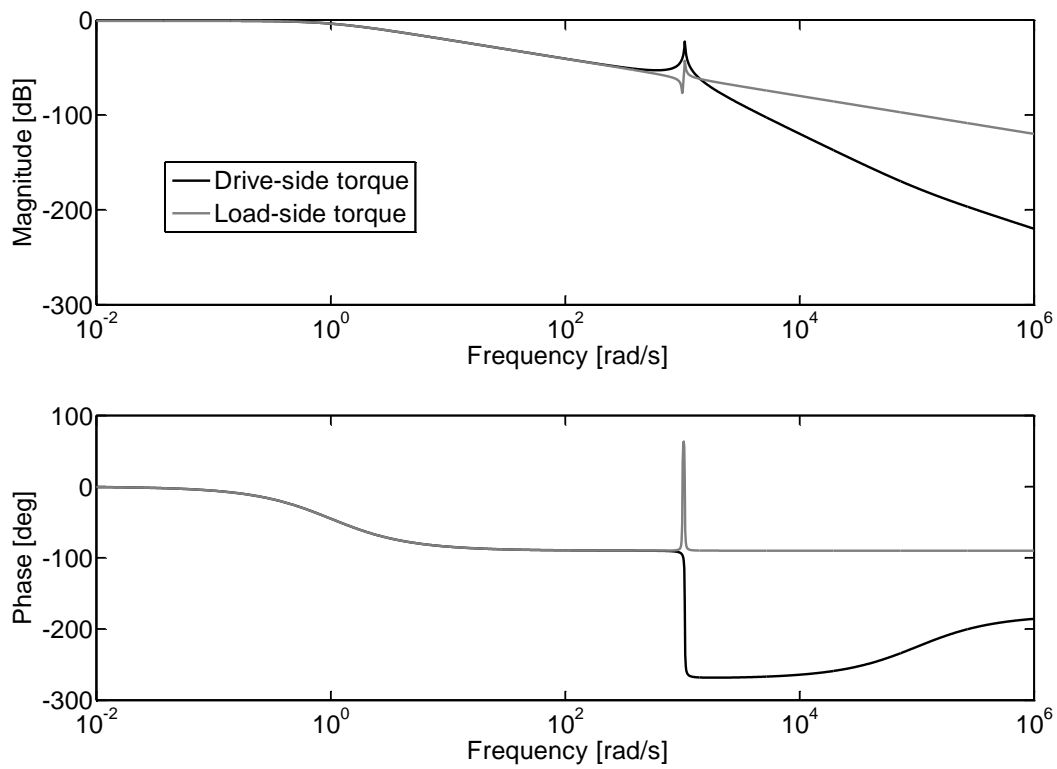


Figure 5.17: Example 2.1: Frequency response of the drive and disturbance load torques.

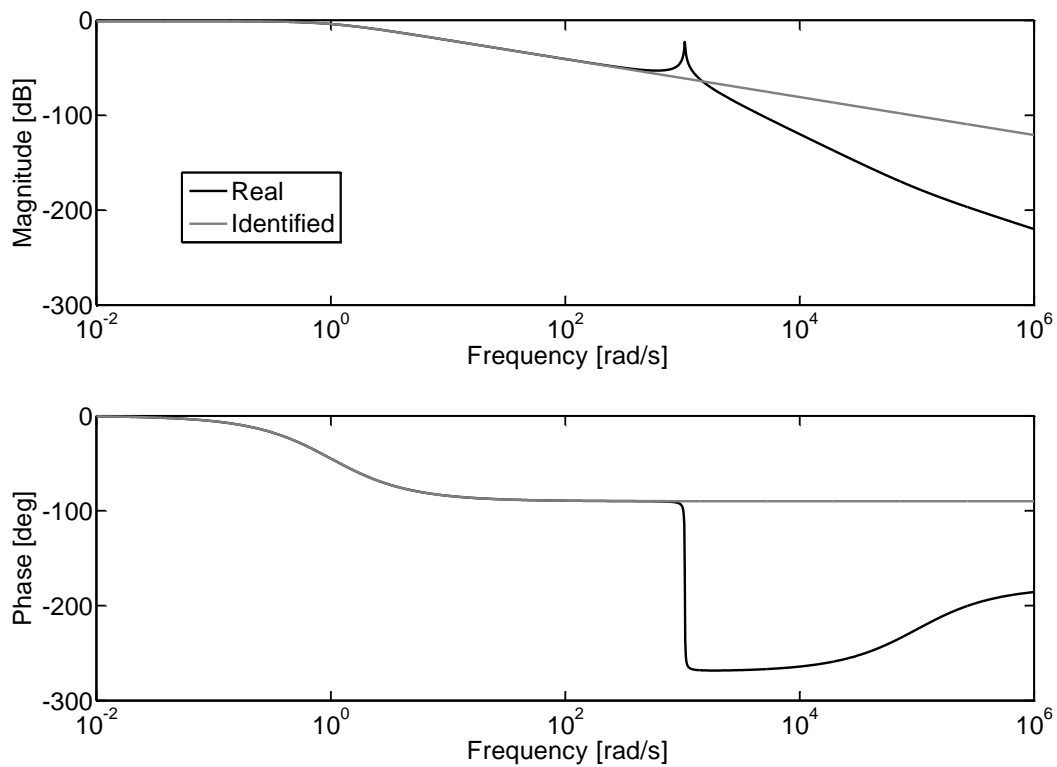


Figure 5.18: Example 2.1: Comparison between real and identified dynamics.

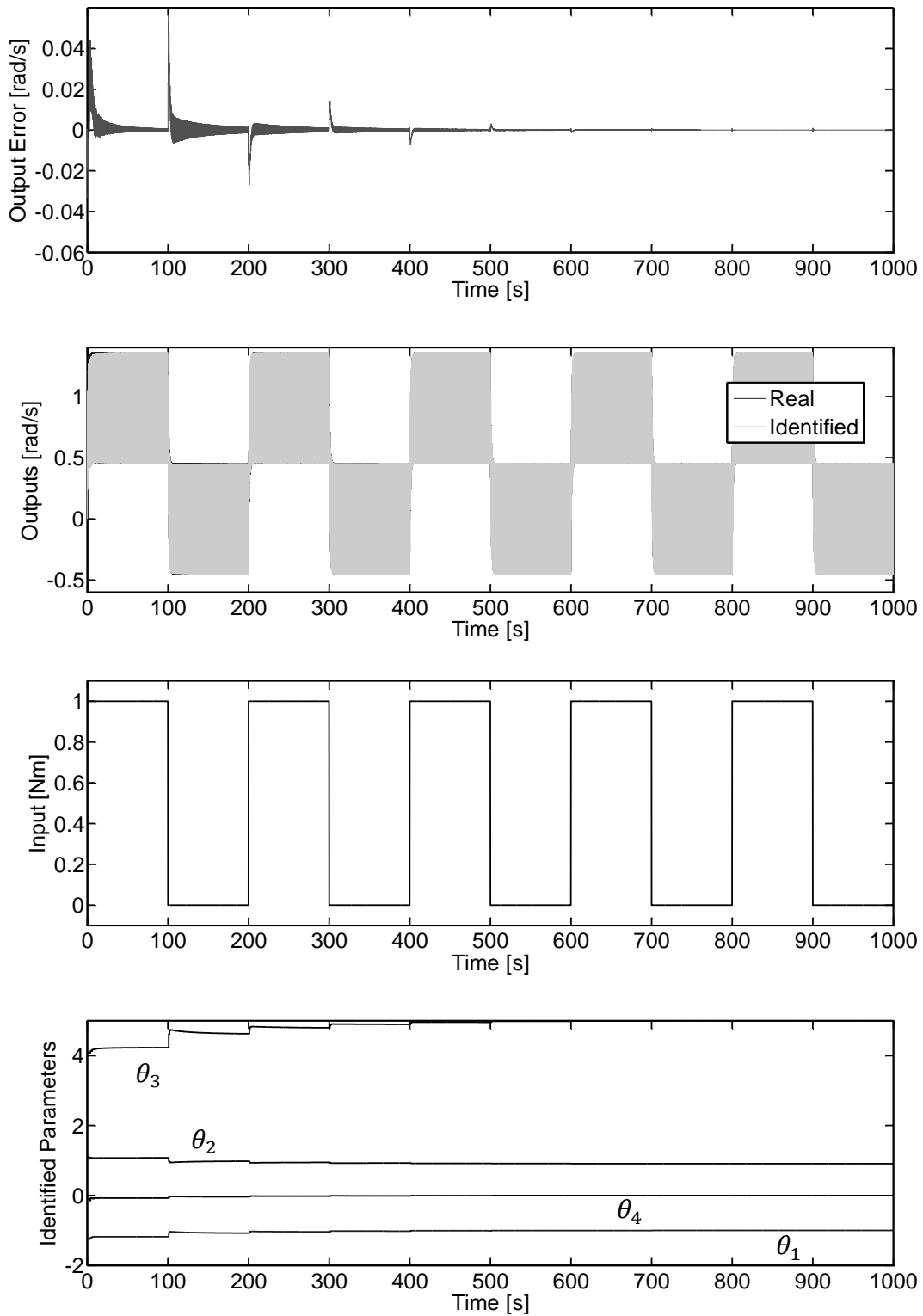


Figure 5.19: Example 2.1: Open loop identification run.

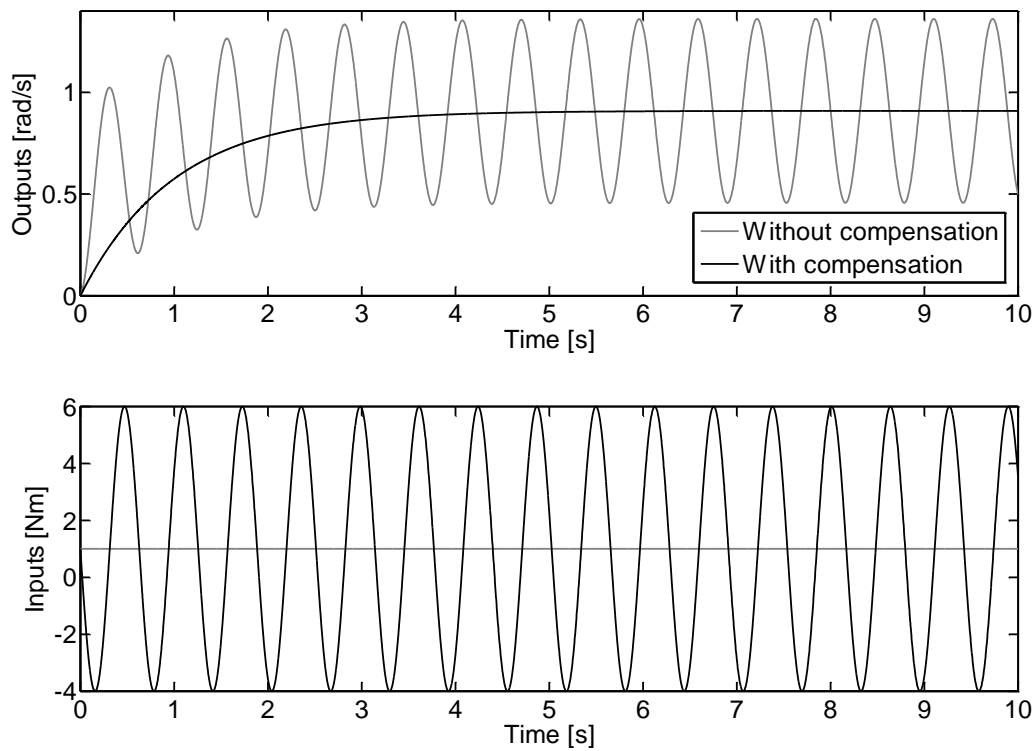


Figure 5.20: Example 2.1: Periodic disturbance compensation in open loop.

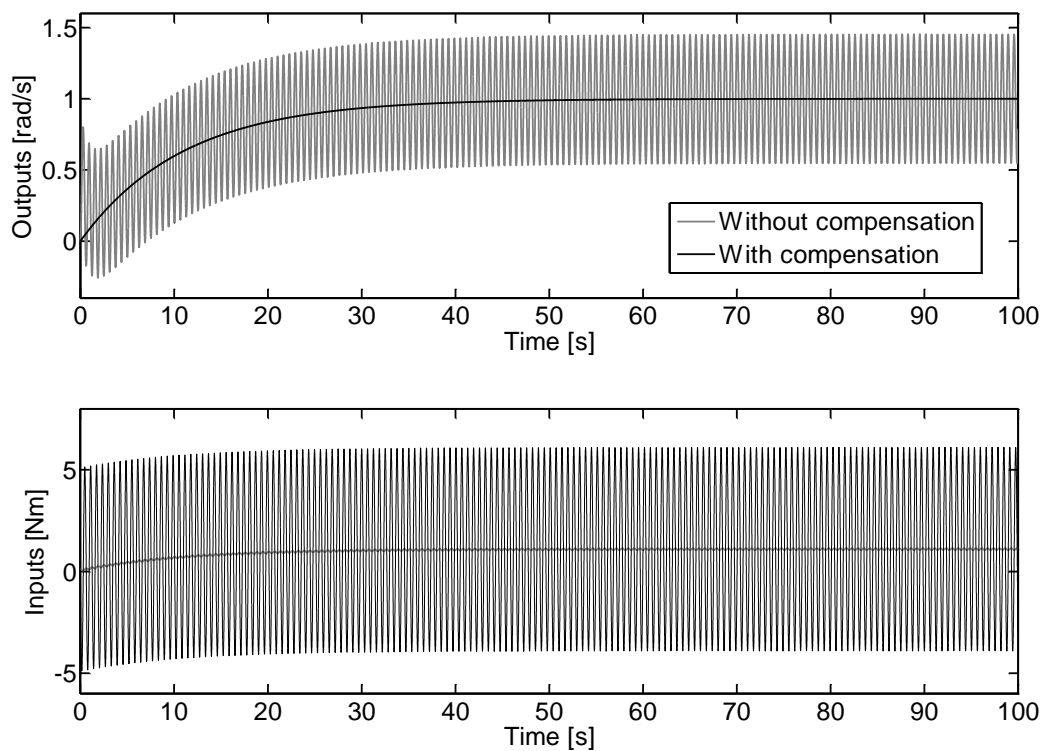


Figure 5.21: Example 2.1: Periodic disturbance compensation in closed loop.

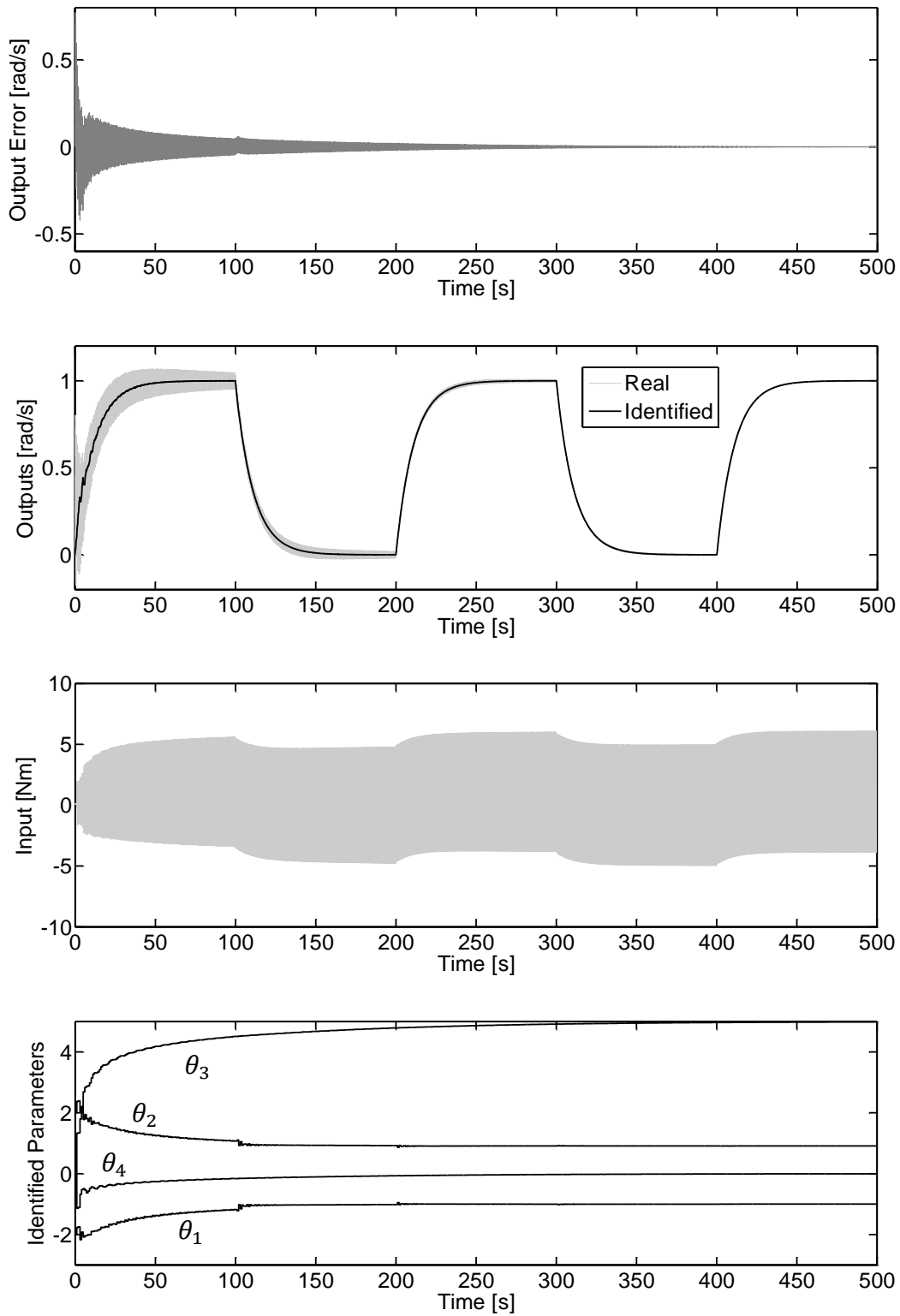


Figure 5.22: Example 2.1: Adaptive run.

5.2.1.2 Simulation Example 2.2: Flexible link case

The system has a flexible link between the drive and the load sides. The system physical parameters are given as following:

$$\begin{aligned}
 \text{Drive side:} \quad & J_D = 0.1 \text{ [kgm}^2\text{]}; \\
 & d_D = 0.1 \text{ [Nms/rad]}; \\
 \text{Mechanical link:} \quad & K_R = 100.0 \text{ [Nm/rad]}; \\
 & D_R = 0.1 \text{ [Nms/rad]}; \\
 & g = 1; \\
 \text{Load side:} \quad & J_L = 1.0 \text{ [kgm}^2\text{]}; \\
 & d_L = 1.0 \text{ [Nms/rad]}.
 \end{aligned} \tag{5.19}$$

As given in equations (5.19) the mechanical link stiffness and damping factors are thousand and hundred times less than the factors of the previous example 2.1 respectively. These parameters yield the following transfer function

$$\begin{aligned}
 \phi_L(s) = & \frac{s + 1000}{s^3 + 3.1s^2 + 1102s + 1100} T_D(s) \\
 & + \frac{s^2 + 2s + 1000}{s^3 + 3.1s^2 + 1102s + 1100} T_L(s).
 \end{aligned} \tag{5.20}$$

The system poles are at $(-1.05 + 33.1496i, -1.05 - 33.1496i, -1)$. The system is definitely a third order system but from the frequency response in Figure 5.23, it can be seen for frequencies lower than 30 [rad/s], that the system can still be approximated by a first order system as in previous example.

The identification model is chosen to be the same as given in previous example represented by equations (5.17). The process is also run with the same external disturbance as in previous example. The real and identified input-output transfer functions are plotted in Figure 5.24, where it can be seen that the first order dynamics give a perfect representation to the flexible dynamics up to 20 [rad/s]. Figure 5.25 shows the adaptive run of system identification and compensation, where the algorithm has succeeded in identification and compensation of the external periodic disturbance, since the targeted external disturbance, in this case, has frequency of 10 [rad/s].

5.2.1.3 Simulation Example 2.3: Flexible link case

This is the same as in example 2.2, where the identification model dynamics are represented by a first order linear transfer function. But in this example, the frequency of the external disturbance is 40 [rad/s]. This means, the external periodic disturbance signal is defined by

$$T_L(t) = 5 \cdot \sin(40t) \text{ [Nm]}. \tag{5.21}$$

With the same initial parameters as given in example 2.1, an adaptive session is done and plotted in Figure 5.26, where it shows that the parameters diverge, and the identification algorithm could not have found a local minimum. The main reason is that the identification model could not represent the flexible region by just a first order identification model.

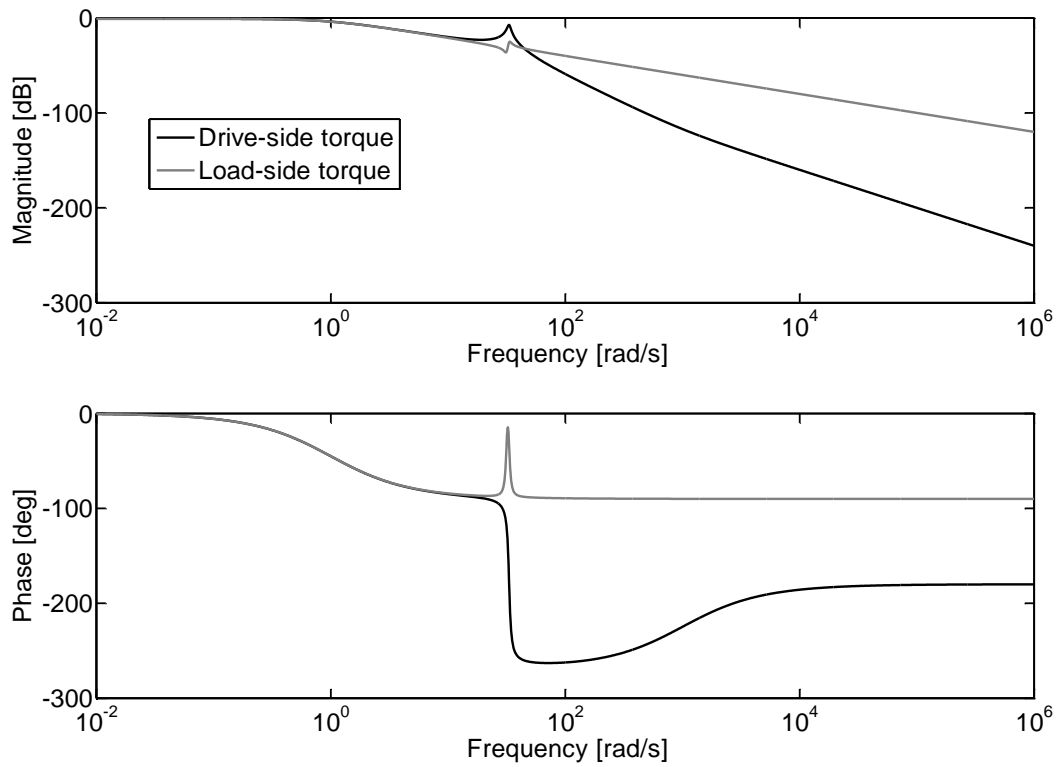


Figure 5.23: Example 2.2: Frequency response of the flexible drive-load system.

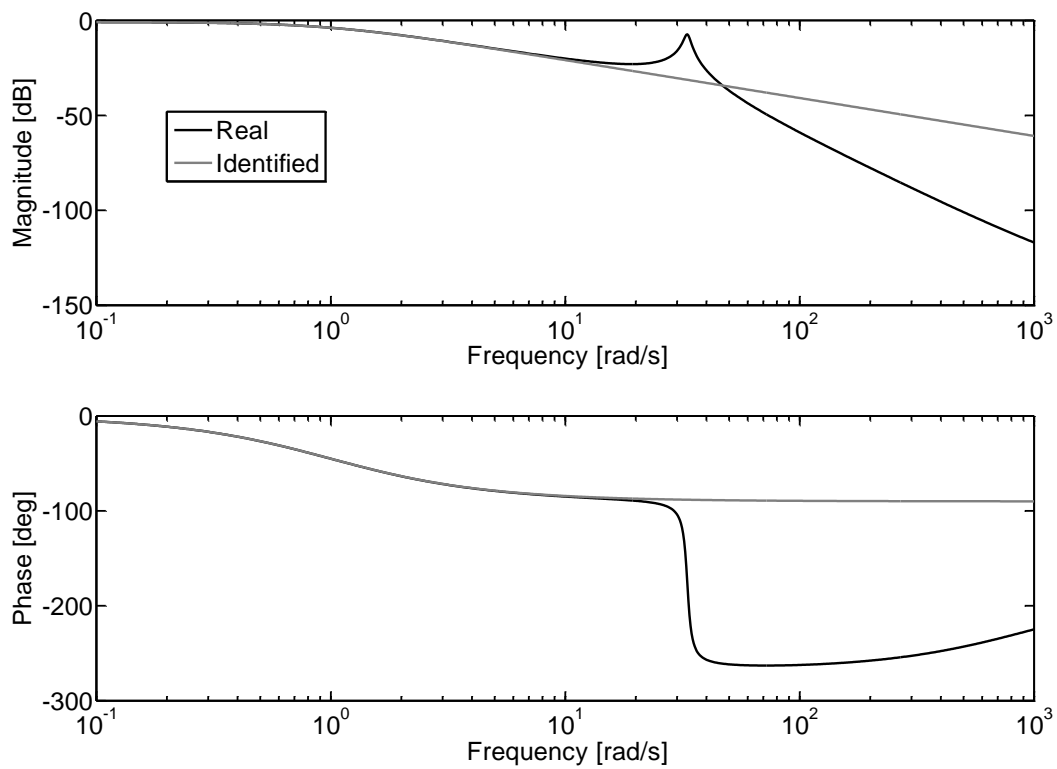


Figure 5.24: Example 2.2: The frequency response of the real flexible and the identified first order system dynamics.

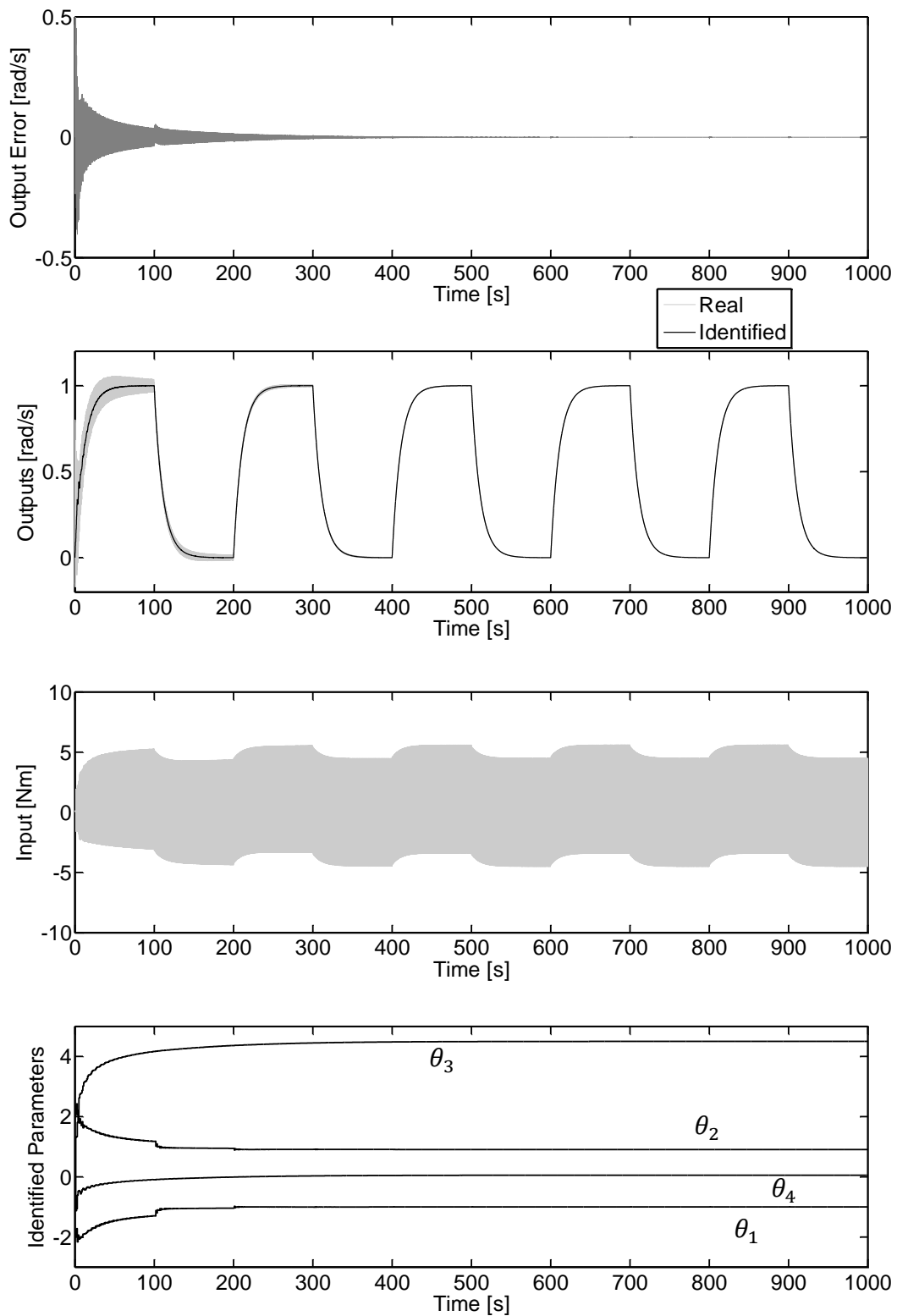


Figure 5.25: Example 2.2: Closed loop adaptive run.

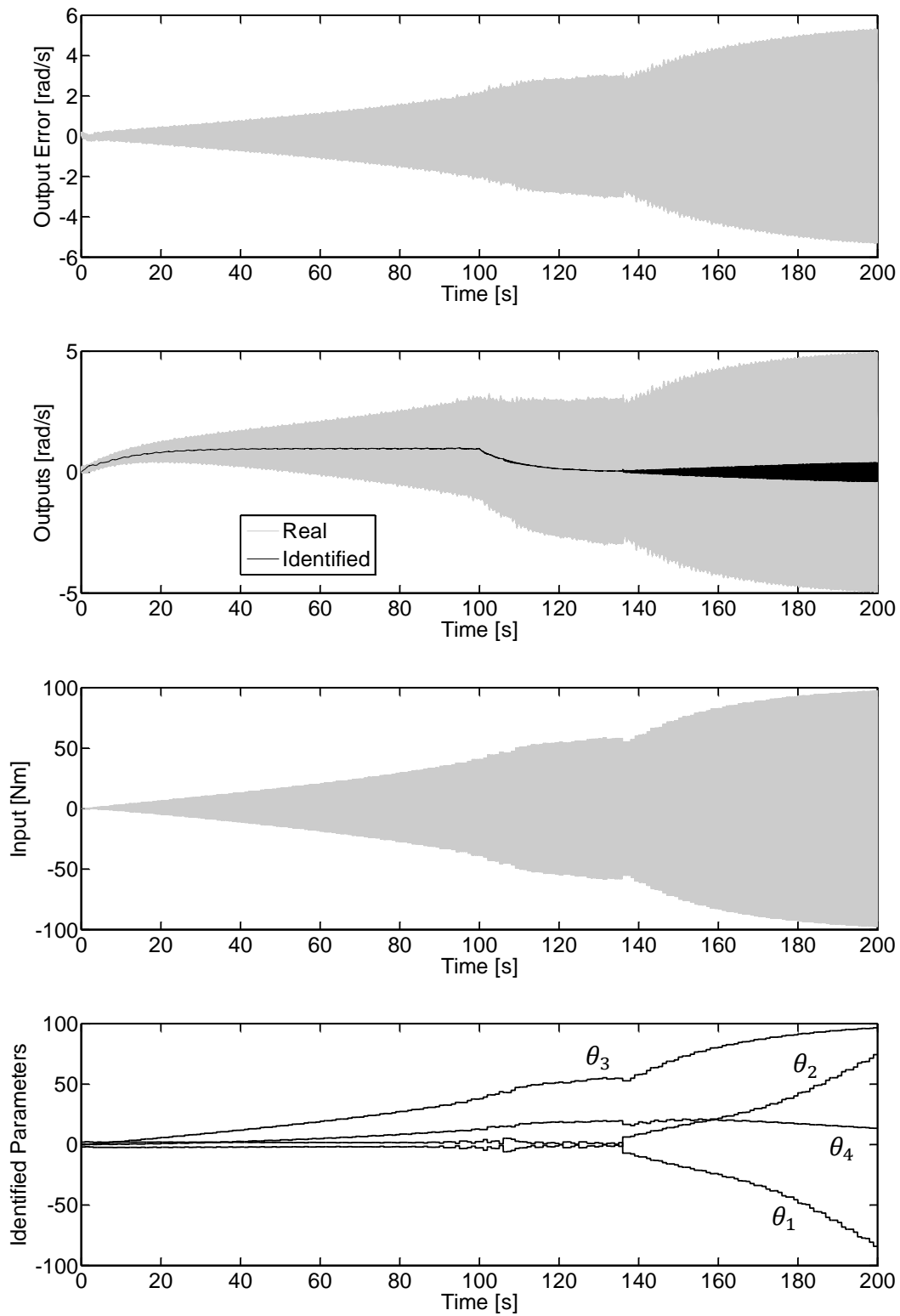


Figure 5.26: Example 2.3: Closed loop adaptive run.

5.2.1.4 Simulation Example 2.4: Flexible link case

In this simulation example, the identification model dynamics are chosen to be a third order as

$$y(s) = \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} [u(s) + u_{dis}(s)]; \quad (5.22)$$

$$u_{dis}(t) = \alpha_{0,1} \sin(\omega t) + \beta_{0,1} \cos(\omega t),$$

with the parameter vector

$$\theta = [-a_0 \quad -a_1 \quad -a_2 \quad b_0 \quad b_1 \quad \alpha_{0,1} \quad \beta_{0,1}]^T, \quad (5.23)$$

and the initial parameter vector $[-1150, -1100, -3, 1000, 1, 0, 0]^T$.

An adaptive session is started with the same initialization parameters as in example 2.1 and plotted in Figure 5.28, which ended with the final parameter vector $[-1128.8, -1100.4, -2.9, 1023.1, 0.4, -3.0, 0.5]^T$. Figure 5.27 plots the frequency response of the real and the identified system dynamics.

As the case of first order identification model, the identification model has failed to represent the dynamics of the real flexible system especially at the resonance region, as demonstrated in the previous example 2.4. This example, on the other hand, showed that an identification model of third order could represent the real system dynamics, and therefore, the identification algorithm has managed to find good parameters of both the dynamic and disturbance parts. Consequently, the identified parameters of the identification model can be used to compensate the system disturbances as well as its dynamics.

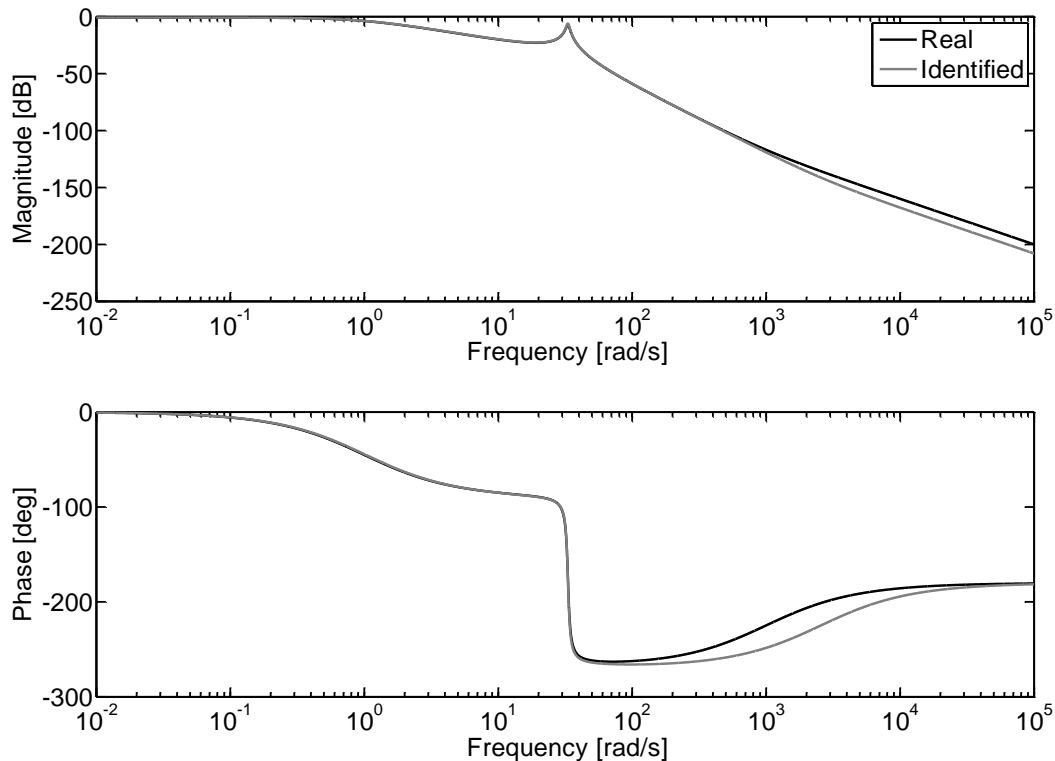


Figure 5.27: Example 2.4: Frequency response of the real and the identified system dynamics.

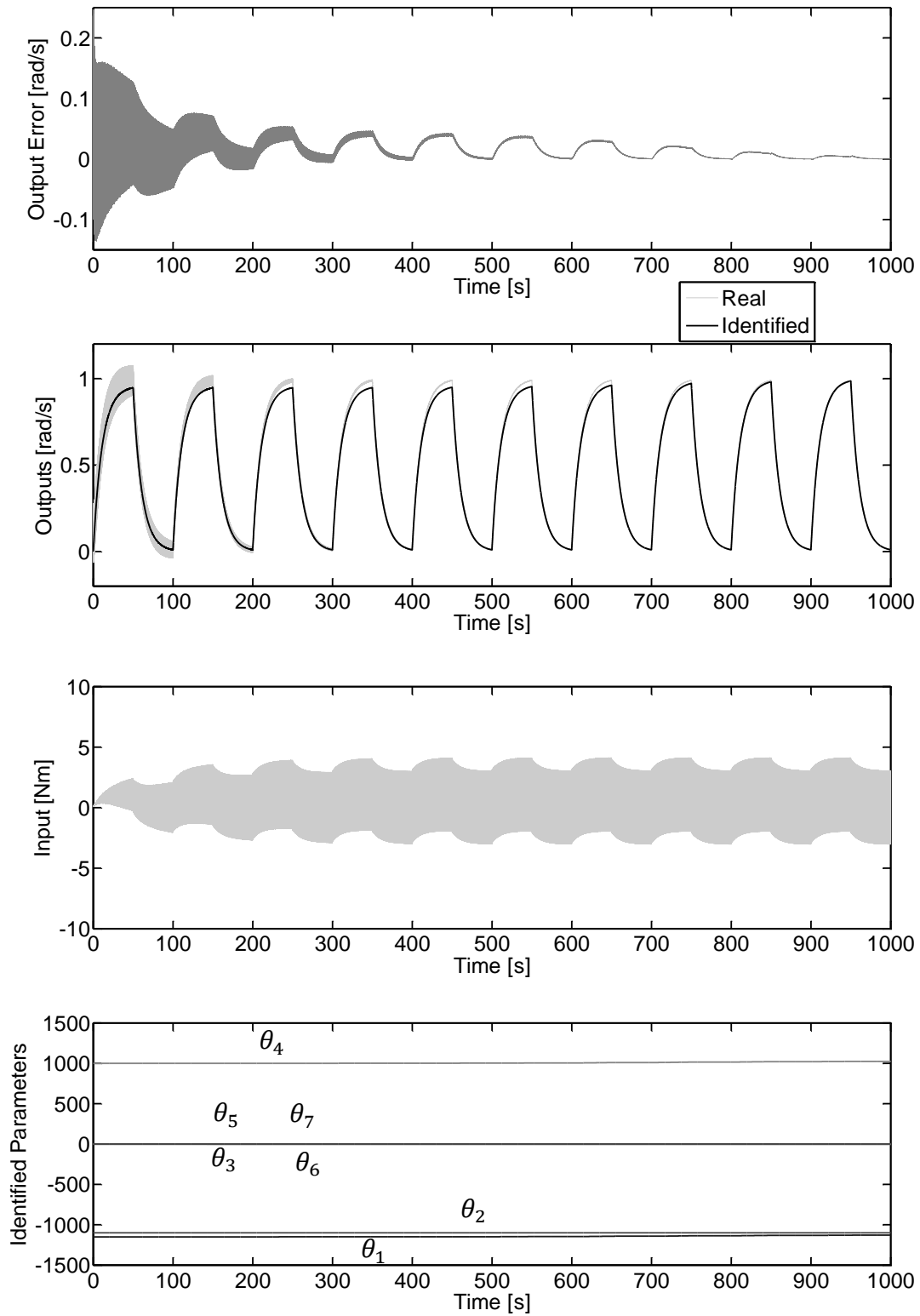


Figure 5.28: Example 2.4: Closed loop adaptive run.

5.2.2 Internally Excited Periodic Disturbance Compensation

In this subsection, the drive-load system is defined by the following physical parameters:

$$\begin{aligned}
 \text{Drive side:} \quad & J_D = 0.1 \text{ [kgm}^2\text{]}; \\
 & d_D = 0.1 \text{ [Nms/rad]}; \\
 \text{Mechanical link:} \quad & K_R = 100.0 \text{ [Nm/rad]}; \\
 & D_R = 0.1 \text{ [Nms/rad]}; \\
 & g = 1; \\
 \text{Load side:} \quad & J_L = 1 + 0.1 \sin(\varphi_L) \text{ [kgm}^2\text{]}; \\
 & D_L = 1 + 0.1 \sin(\varphi_L) \text{ [Nms/rad]}; \\
 & K_L = 0.1 \sin(\varphi_L) \text{ [Nm]}.
 \end{aligned} \tag{5.24}$$

The load side has angle dependent spring, damper and moment of inertia load elements that cause the self-excited nonlinear oscillations. In the following, two examples are presented, where the drive-load system is constructed as velocity servo control system using a PI controller with the parameters ($K_P = 0.1, K_I = 0.1$) that presumably satisfy the set point tracking demands. Furthermore, an add-on feed-forward controller, as developed in chapter 4, is used to identify and to compensate the self-excited periodic disturbances appear on the system output, which is the angular velocity of the load side. In the first example, the internal periodic disturbance is represented locally, while in the second example the disturbance part of the identification model is represented globally.

5.2.2.1 Simulation Example 2.5: Local identification model

The identification model has a first order continuous dynamics model and the disturbance model is angle dependent and of zero degree dependence on the output angular velocity as following

$$\begin{aligned}
 \dot{y}(t) &= ay(t) + b[u(t) + u_{dis}(t)]; \\
 u_{dis}(t) &= \sum_{i=1}^{N_1} \alpha_{1,i} \sin(i\varphi_m) + \beta_{1,i} \cos(i\varphi_m),
 \end{aligned} \tag{5.25}$$

with the parameter vector

$$\boldsymbol{\theta} = [a \ b \ \alpha_{1,1} \ \alpha_{1,2} \ \cdots \ \alpha_{1,N_1} \ \beta_{1,1} \ \beta_{1,2} \ \cdots \ \beta_{1,N_1}]^T. \tag{5.26}$$

The parameter vector is identified using the parametric recursive output error method, because the identification model is in the form of nonlinear in parameters, as introduced in Appendix B. The identification procedure has started with initial parameter vector $[-2, 2, 0, 0, 0, 0, 0, 0]^T$, initial covariance matrix of $10\mathbf{I}$, forgetting factor of 0.98, sampling time of 1 [s] and number of harmonics considered ($N_1 = 3$).

As can be seen from Figure 5.29, the system starts under feedback PI controller with only identification to track 2 [rad/s] step set point, then at about time 250 [s], the feed-forward controller is activated, so that it uses the last identified disturbance parameters in its control law in order to compensate the identified self-excited periodic disturbances.

As expected, at the activation time of 250 [s], see Figure 5.29, the identifyability and the identification conditions (as discussed in subsection B.4.1) of direct process identification in closed loop become better, therefore, the identification model parameters in this phase converge asymptotically quickly to the optimal ones.

It can be seen that, the local model can work very well when the output follows a constant set point; otherwise, disturbance model will never reach a steady state, particularly since the identification needs some time to gather information about the process change in order to take it in the identified model.

5.2.2.2 Simulation Example 2.6: Global identification model

This example is the same as the previous example 2.5 except that the disturbance model is angle dependent and has zero first and second degree dependence on the output angular velocity. In general, the disturbance part can now globally describe the internal input periodic disturbance of the given drive-load process. The identification model is constructed as following

$$\begin{aligned} \dot{y}(t) &= ay(t) + b[u(t) + u_{dis}(t)]; \\ u_{dis}(t) &= \sum_{i=1}^{N_1} [\alpha_{1,i} \sin(i\varphi_m) + \beta_{1,i} \cos(i\varphi_m)] \\ &\quad + \sum_{i=1}^{N_2} [\alpha_{2,i} \sin(i\varphi_m) + \beta_{2,i} \cos(i\varphi_m)] y(t) \\ &\quad + \sum_{i=1}^{N_3} [\alpha_{3,i} \sin(i\varphi_m) + \beta_{3,i} \cos(i\varphi_m)] y^2(t). \end{aligned} \tag{5.27}$$

The initial dynamic parameters are the same as given in previous example, all disturbance parameters are initialized by zeros, with numbers of harmonics considered ($N_1 = 1$, $N_2 = N_3 = 5$). The system is also run with the given PI feedback velocity servo controller as given in previous example. The system is to follow a square wave set point and to reject the self-excited periodic disturbance by using the adaptive feed-forward controller. The session is plotted in Figure 5.30, where at the beginning of the session, only identification is done without feed-forward periodic disturbance compensation, later at time about 375 [s], the feed-forward control is activated with a direct adaptation mode by using directly the last identified disturbance parameters in its control law. Moreover, after the adaptive identification run is finished, the final parameters are used to compensate the system Figure 5.31 in open loop and Figure 5.32 in closed loop with the PI controller. Both cases use stair case input signals in order to show the global validation of the identified identification model dynamic and disturbance parameters.

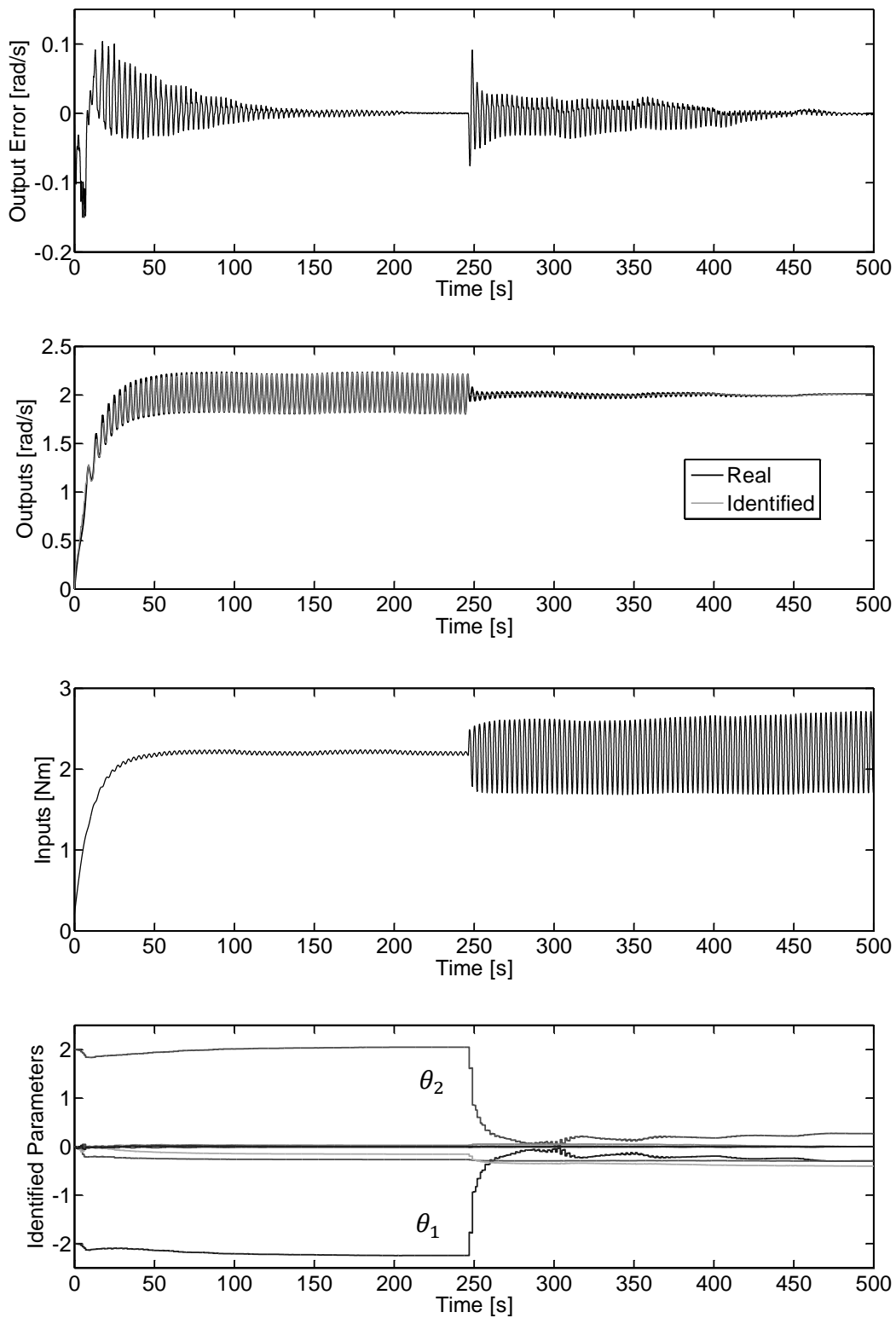


Figure 5.29: Example 2.5: Closed loop adaptive run.

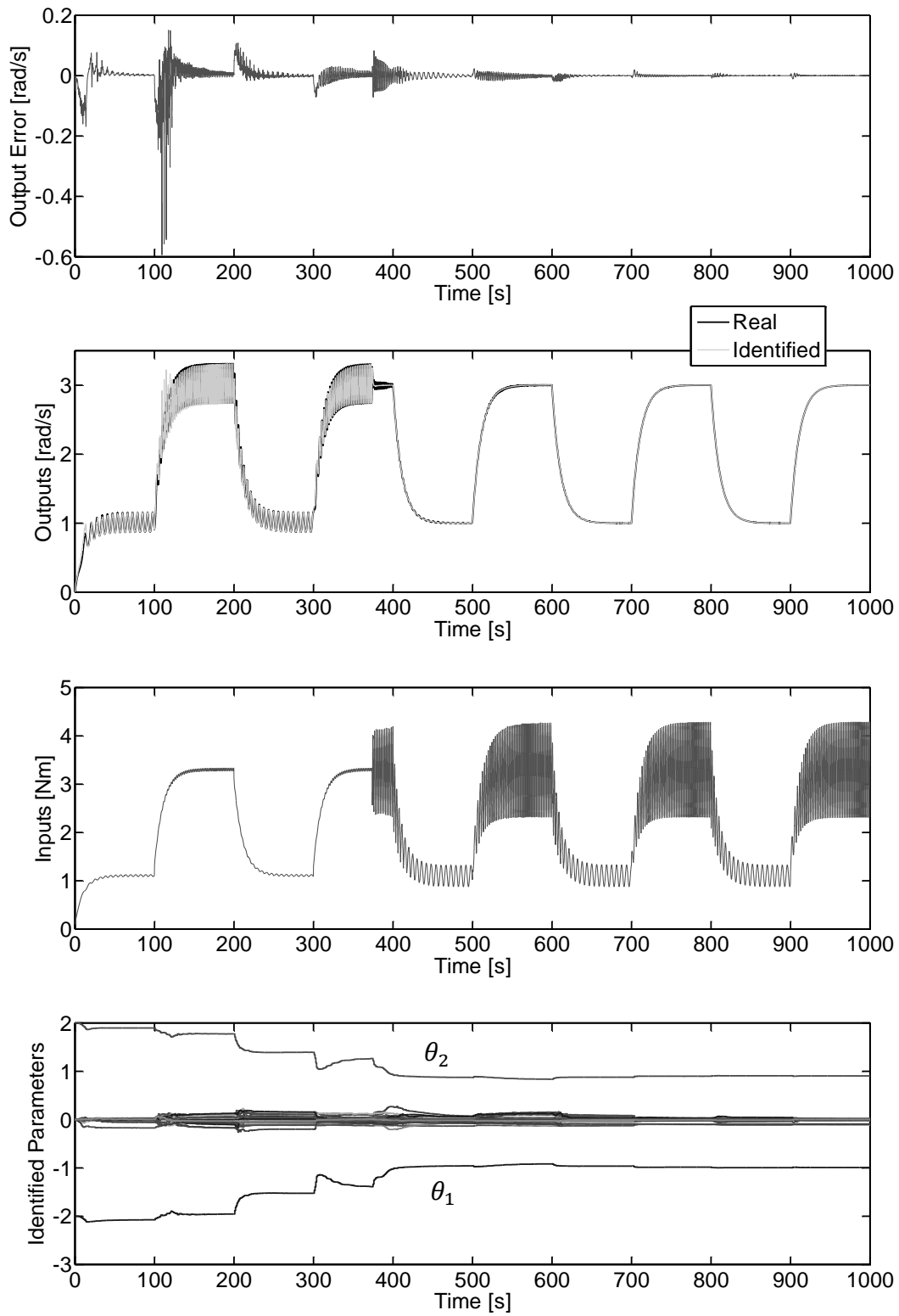


Figure 5.30: Example 2.6: Closed loop adaptive run.

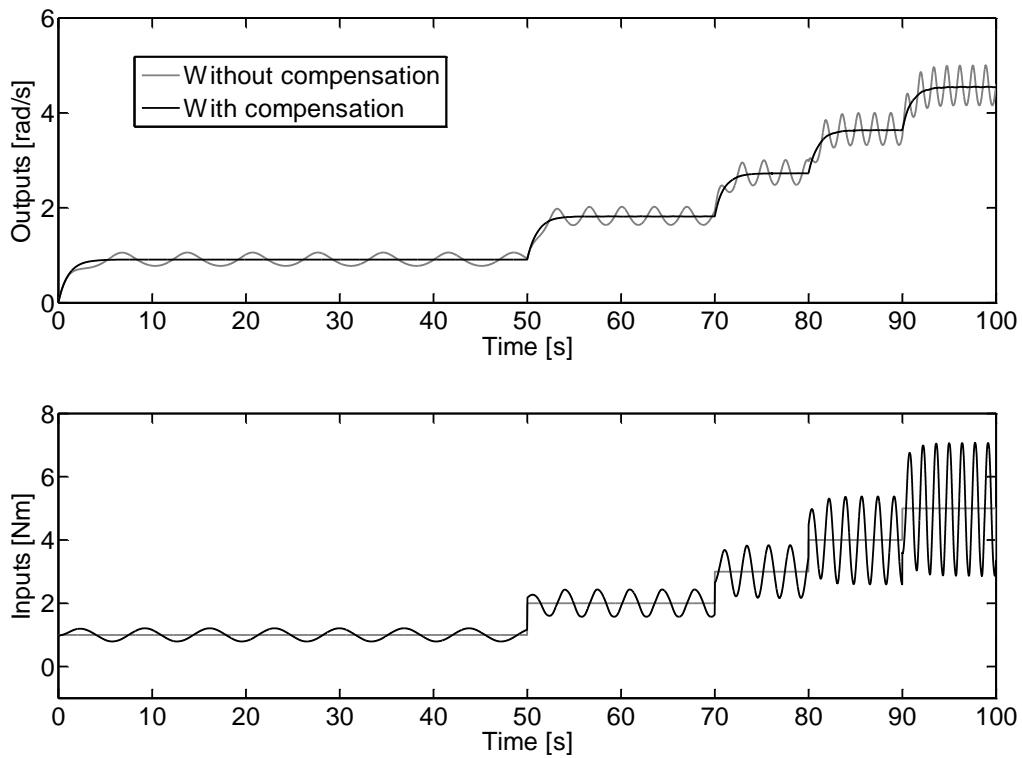


Figure 5.31: Example 2.6: Periodic disturbance compensation in open loop.

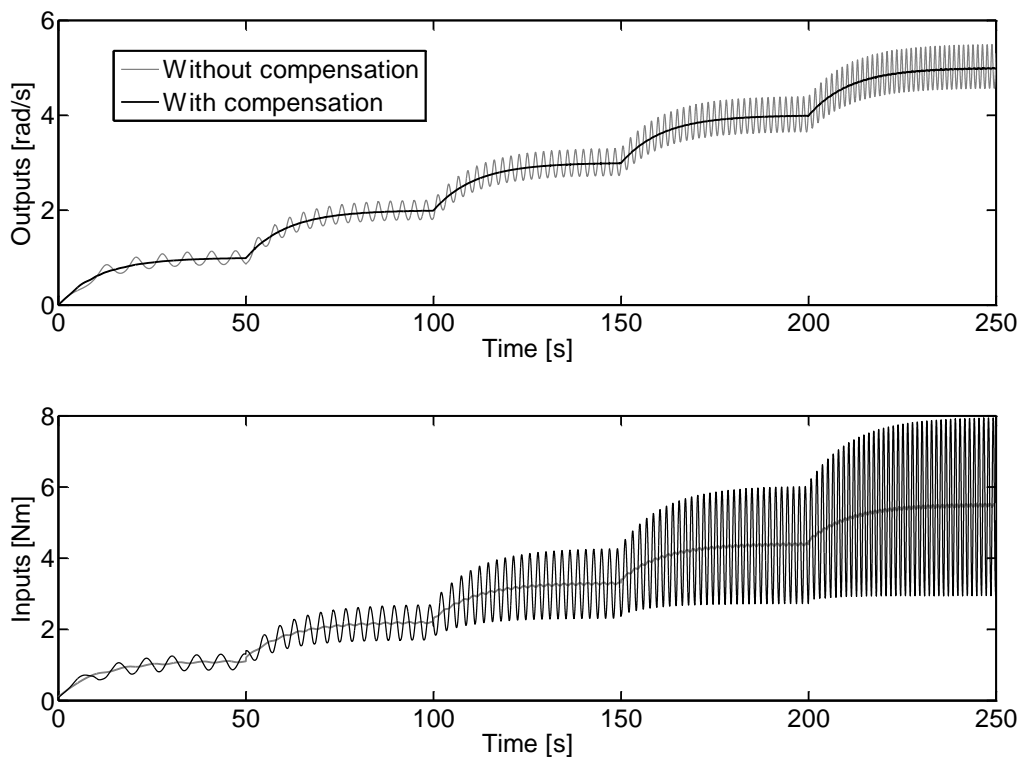


Figure 5.32: Example 2.6: Periodic disturbance compensation in closed loop.

5.2.3 Discrete Periodic Disturbance Compensation

In this subsection, the method of Recursive Least Squares (RLS) is used to identify the linear in parameters identification model as constructed in section 4.3, in other words, the RLS is chosen to identify the identification model because it has a linear in parameters format. In the next, two simulation examples are presented, the first one has simple local internal disturbance model while the second one has more global internal disturbance model to work better in a wider operating region.

5.2.3.1 Simulation Example 2.7: Local model

First of all, the drive-load system physical parameters are the same as in example 2.5, with feedback PI controller. Moreover, in this example, a discrete identification model, as defined in section 4.3, is used to identify the system dynamics and the periodic disturbance in order to compensate it. Furthermore, the identification model dynamics are of the first order and the internal disturbance is angle dependent of zero degree dependence on the identification model angular velocity output ($n = 1$, $N_0 = 0$, $N_1 = 1$, $N_2 = N_3 = 0$).

Now, since the identification model is linear in parameters, the RLS algorithm, as introduced in Appendix B.2.3, is used to identify the drive-load system online with the initial parameters: sampling time 0.1 [s], number of harmonics 5, initial dynamic parameters are all set to ones and all initial disturbance parameters are set to zeroes, initial covariance matrix is $100\mathbf{I}$ and forgetting factor of 0.998.

The experiment starts from the very beginning with an active periodic disturbance adaptive feed-forward controller. After about 725 [s], the feed-forward periodic disturbance compensator is deactivated in order to show the impact of the self-excited periodic disturbance on the system output, as it can be seen in the Figure 5.33.

5.2.3.2 Simulation Example 2.8: Global model

In this example, the system and the initial identification model parameters are the same as in the previous one, except that the internal disturbance part is angle dependent and of zero, first and second degree dependence on the output angular velocity, with only 3-harmonics are considered $n = 1$, $N_0 = 0$, $N_1 = N_2 = N_3 = 3$. This means, this identification model disturbance part is more global than the one in the previous example. This explains the better performance of the periodic disturbance feed-forward compensator as shown in the Figure 5.34. Particularly, it can be seen from the output error graph, the first one from the top, in comparison with the previous simulation example 2.7.

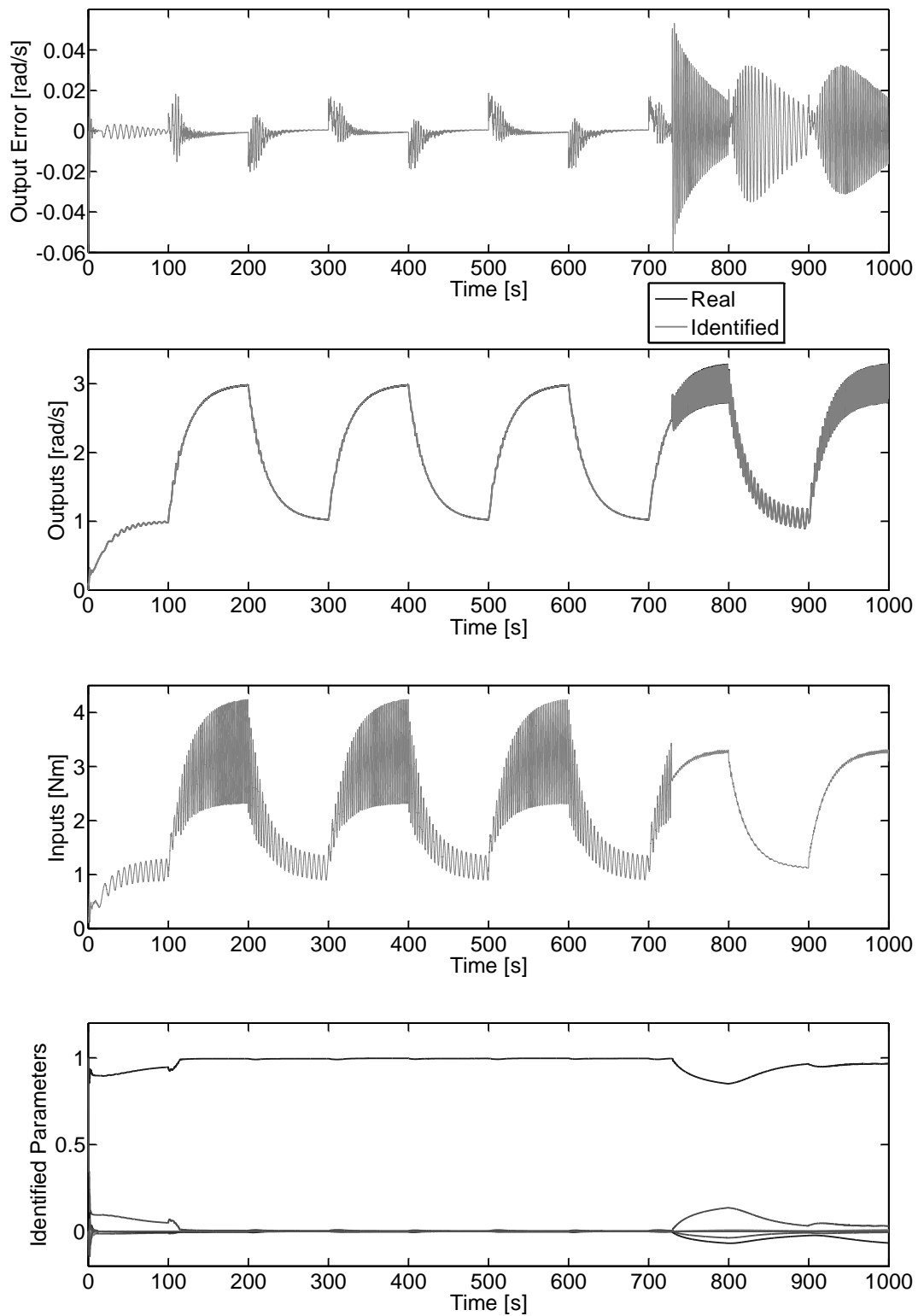


Figure 5.33: Example 2.7: Closed loop adaptive with 5-harmonics.

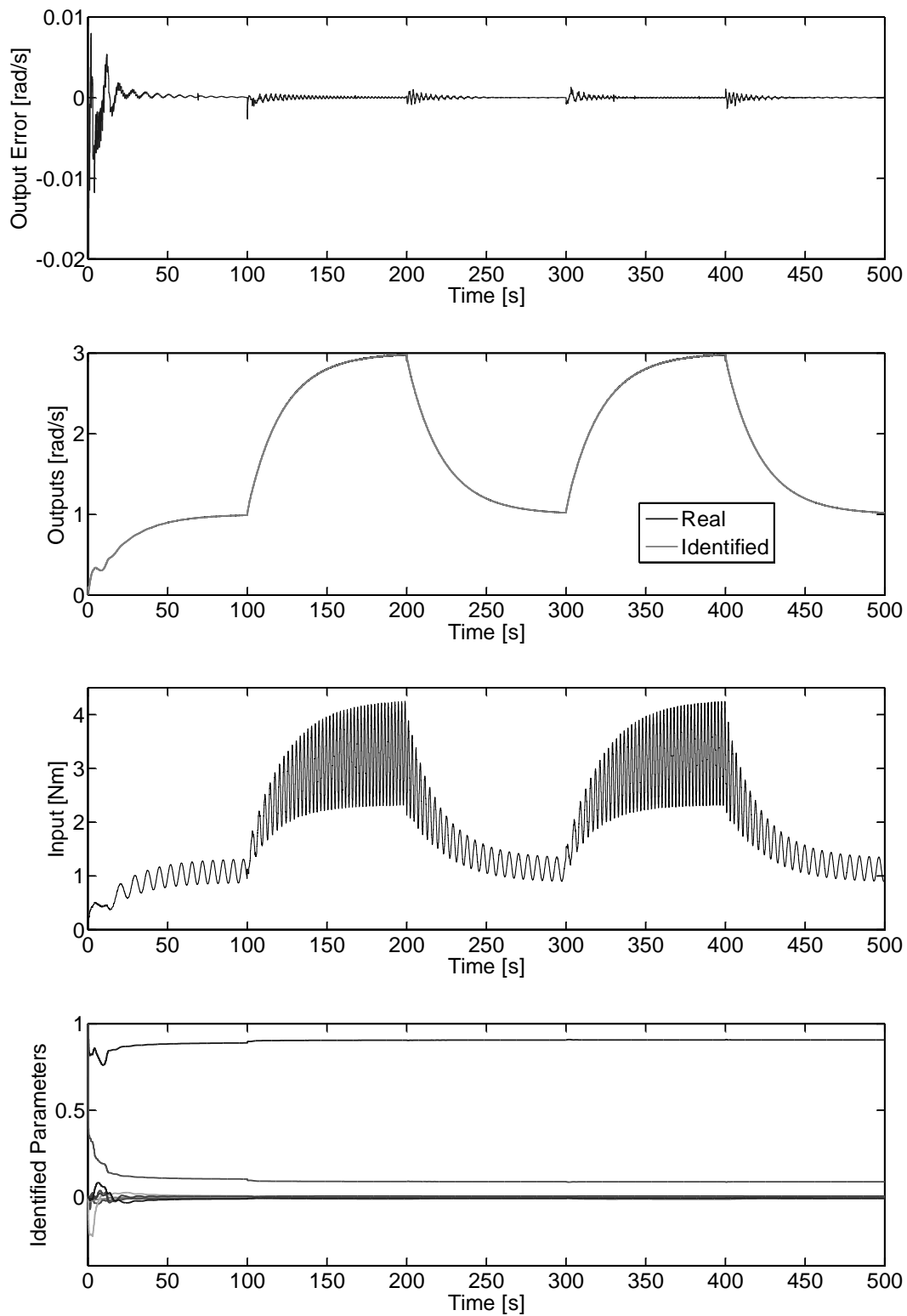


Figure 5.34: Example 2.8: Closed loop adaptive with 3-harmonics.

5.2.4 Comments on the Flexible Drive-load System

The algorithm is applied to compensate the periodic disturbances of a flexible drive-load system in both regions, the rigid and the flexible one, successfully, provided that the identification model is capable of approximating the system input-output dynamics in the operating region either locally or globally. Also, the algorithm of periodic disturbance identification and control using the method of RLS to identify the discrete (linear in parameters) identification model has been tested successfully in compensating the internal and external periodic disturbances.

The flexible drive-load system dynamics can be modeled by an identification model with first order dynamics, when the operating bandwidth of the system is below its resonance frequency. In other words, it works like a rigid body and can be represented with 1DOF system as shown in the simulation examples 2.1 and 2.2. On the other hand, when the flexible drive-load system is operating around the resonance frequency, its behavior is dominated by the third order dynamics, and it can only be modeled by an identification model with a third order dynamics. Otherwise, the identified dynamic parameters will never represent the drive-load system dynamics and consequently the identified periodic disturbance parameters will also be useless as already shown in the simulation examples 2.3 and 2.4.

Modeling of the internal periodic disturbance locally leads also to its compensation locally. But if the local modeling is used to target a wider operating region, then the algorithm must adapt the identification model parameters consistently to cope with the system nonlinear behavior. Alternatively, modeling the identification model globally will allow a wider operational implementation of the identification model parameters. This has been demonstrated in the simulation examples 2.5 and 2.6 respectively using continuous time identification model and in the simulation examples 2.7 and 2.8 respectively using discrete-time identification model. Moreover, modeling and identification globally makes it possible to do one shot adaptation and then the identified internal periodic disturbance parameters can be used to construct a compensation function to compensate this undesirable nonlinear behavior of the drive-load system that caused by the state dependent (periodic) parameters.

The identification model parameters converge in closed loop identification quickly when the certainty equivalence adaptive mode, when the identified parameters are directly used in control law without questioning their credibility, is active, as it is expected in Appendix B.4. This can be seen in example 2.5, where the parameters have rapidly converged to the optimal parameters as soon as the adaptive mode is activated, or in example 2.7, where the parameters have diverged from the optimal parameters as soon as the adaptive mode is deactivated.

Finally, the introduced algorithms of modeling, identification and control of periodic disturbances have been applied and verified in a series of simulation tests. Where the algorithm has showed a good behavior in compensation of the external and the internal periodic disturbances as long as the identification model is capable of approximating the real system input dynamics and the disturbance either locally or globally according to its targeted operating region. These good results gained from the extensive simulation tests are the prompt for the next step of applying the algorithm on a real-time control experiment with a real mechanical plant in the loop. This will be presented in the next chapter 6.

6 REAL-TIME EXPERIMENTS

In this chapter, the developed algorithms of periodic disturbance compensation, after their successful implementation in the simulation experiments, are implemented now in real-time controllers and tested on real mechanical plants. The torsional plant is presented in the first section, while the self-excited machine is presented in the second section. The torsional test platform is ready to buy commercially. It was also ready to use, therefore, it has been used from the very beginning to emulate the drive-load system with external and or internal periodic disturbances and to evaluate the control algorithms on it. Furthermore, as next step to the successful simulation and emulation implementations of the algorithms, an experimental drive-load system with crankshaft mechanism is built up as a self-excited machine together with a powerful real-time controller in order to test the developed algorithms ultimately. Final comments are given at the end of each section correspondingly.

6.1 Torsional Test Platform

In this section, the drive-load system with external and internal periodic disturbance is emulated and the developed algorithms of periodic disturbance rejection/compensation are implemented and tested on it.

6.1.1 Test Platform Description

The torsional plant model 205 is from the Educational Control Products Company, as illustrated in Figure 6.1, the system is constructed by a controller/data board and firmware plugged in a host PC through its PCI bus, the I/O electronics unit and the electromechanical torsional plant. The controller board runs the control algorithms compiled from the Matlab-Simulink program using Real-Time Windows Target (RTWT) software. The controller board receives angular incremental position signal from the angular position encoders installed on the electromechanical plant via I/O electronics unit. Moreover, the board sends the control signals to the main drive-motor and the secondary (disturbance) motor. I/O electronics unit also contains the power stage (servo-amplifiers) to control the two motors, for more details, see the manual (Parks 1999). The electromechanical plant is used to construct rigid or flexible mechanical plants driven by the main motor with external and or internal (self-excited or state dependent) disturbances generated by the secondary motor.

6.1.2 External Periodic Disturbance

The test platform is set up to have only disc No. 1 installed without extra weights. The secondary motor is installed in order to work as an external periodic disturbance source applied on disc No. 1, where disc No. 1 is at the bottom, disc No. 2 at the middle and disc No. 3 is at the top of the flexible transmission rod (the mechanical link). There is also an incremental position sensor at every disc, see Figure 6.1. The system is constructed as velocity servo control system, where the angular velocity is derived from the angular position sensor at disc No. 1 and compared with the set point, making a closed loop negative feedback under a PI controller.

The system design objectives are to ensure the set point tracking as primary goal as well as periodic disturbance rejection as a secondary goal to achieve.

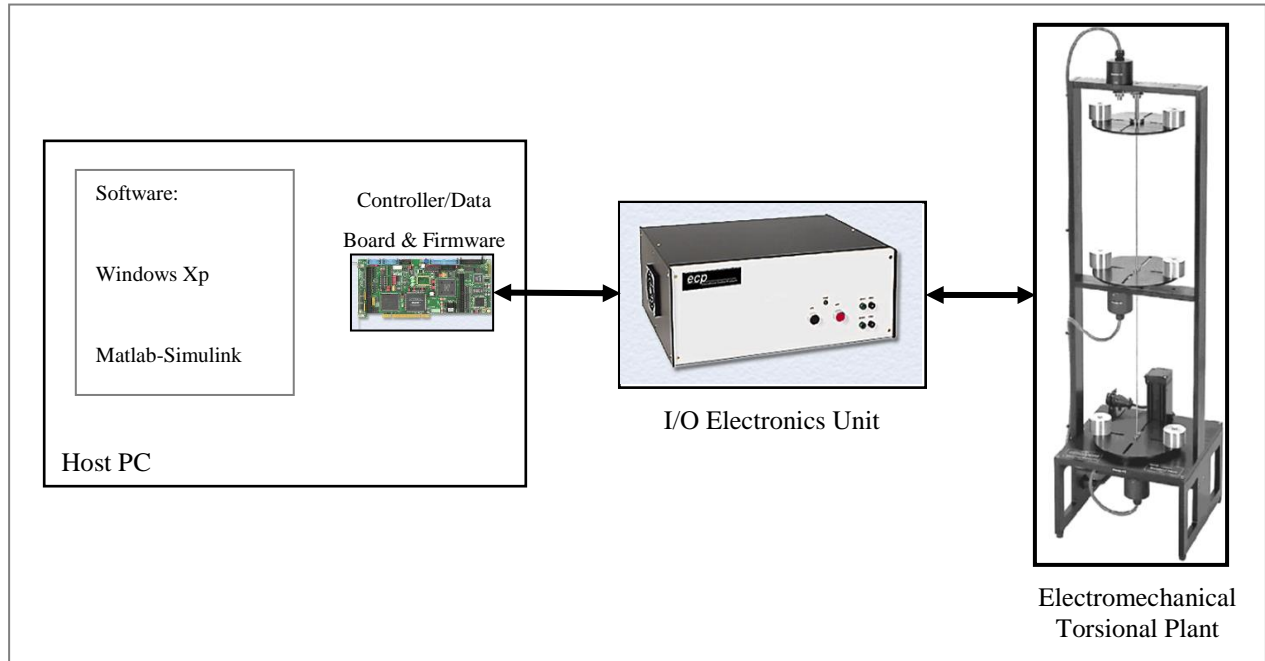


Figure 6.1: Overview of the torsional plant real-time control system from Educational Control Products Company.

6.1.2.1 Experiment 1.1: External periodic disturbance rejection using periodic disturbance rejection filter

To ensure the secondary objective, a periodic disturbance rejection filter is introduced as an add-on to the already existing PI feedback controller already designed to ensure the set point tracking objectives, see Figure 6.2. The discrete filter is the z-transform of the continuous under damped second order system given by

$$G_N(s) = \frac{K_d \omega_d^2}{s^2 + 2 \xi_d \omega_d s + \omega_d^2}, \quad (6.1)$$

where the disturbance frequency is defined by $\omega_d = 10$ [rad/s] and the damping ratio by $\xi_d = 0.001$. By using the z-transformation table (Phillips and Nagle 1995), the continuous transfer function with zero order hold is transformed into discrete transfer function, with a sampling time equal to 0.01 [s], this yields

$$G_N(z) = K_d \frac{4.996 \cdot 10^{-3} z + 4.995 \cdot 10^{-3}}{z^2 - 1.99 z + 0.9998}. \quad (6.2)$$

With K_d as tuning parameter, this will be tuned until a satisfactory performance in terms of disturbance rejection and set point tracking performance. In this experiment, K_d was tuned to the value of (17.33). Now as shown in Figure 6.3, the experiment was started with only the PI

controller with the parameters of ($K_p = 100.0, K_I = 50.0$) and without the external disturbance, at time about 30 seconds, the external periodic disturbance source ($1000\sin(10t)$) was switched on by *manual switch2*, at time equal to 50 seconds, the *manual switch1* was turned on to activate the discrete periodic disturbance rejection filter to reject the external periodic disturbance that comes from the secondary motor. As it can be seen from Figure 6.3, the filter has managed to reject the external periodic disturbance asymptotically.

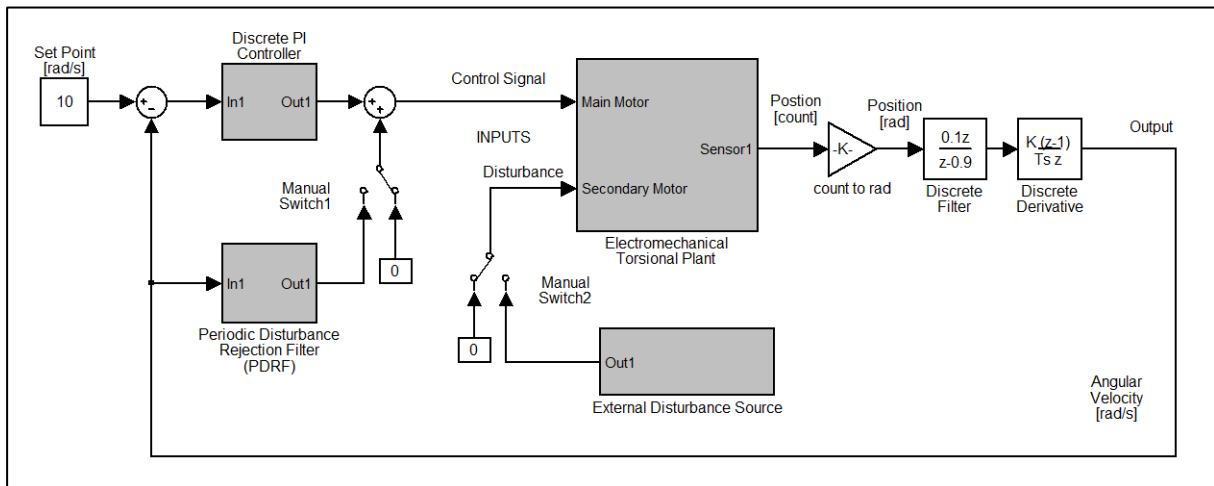


Figure 6.2: Matlab-Simulink diagram of experiment 1.1.

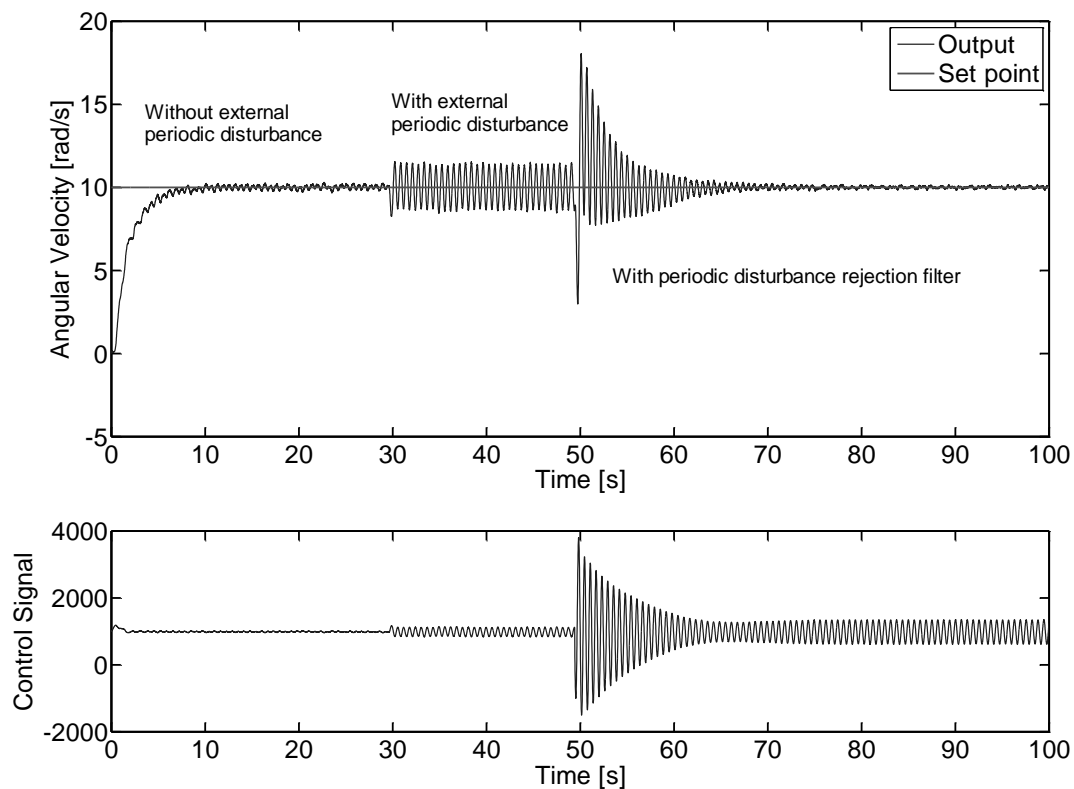


Figure 6.3: The system time response of experiment 1.1.

6.1.2.2 Experiment 1.2: Continuous modeling, identification and feed-forward control

Now, the developed periodic disturbance feed-forward controller in section 4.2, the identification and compensation block, as shown in the Matlab-Simulink diagram Figure 6.4, is added on to the existing set point tracking PI feedback controller. The algorithm was started with the following initial parameters: Sampling period equal to 0.0177 second, first order continuous input-output identification model dynamics, the number of external disturbances harmonics ($N_0 = 2$), and without any internal periodic disturbances ($N_1 = N_2 = N_3 = 0$), initial parameter vector $[-1.94, 0.0146, 0, 0, 0, 0]^T$, initial covariance matrix of $1000\mathbf{I}$ and forgetting factor of 0.998.

Then, the algorithm was compiled and ran in the real-time mode, at the beginning only the PI controller and the identification were active, then at the time about 33 seconds the feed-forward controller was activated (put in certainty equivalence adaptive mode) by using the identified disturbance parameters directly to generate an anti-disturbance (compensation) signal added to the system input. The experiment run is presented in Figure 6.5. As stated in subsection B.4.1, the parameters converge better in the closed loop as the identifyability and the identification conditions are becoming better when the direct parameter adaptation is on. Moreover, it can also be seen from the figure the perfect disturbance compensation is an indication of good parameter identification.

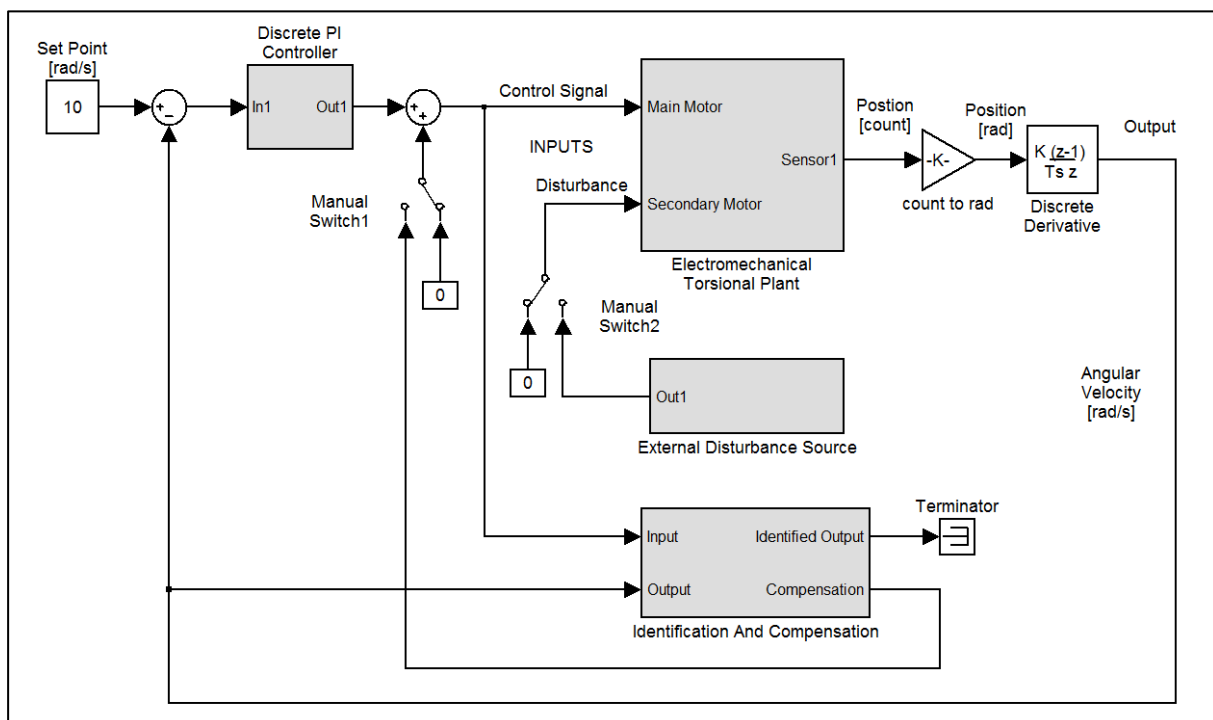


Figure 6.4: The Matlab-Simulink diagram of experiment 1.2.

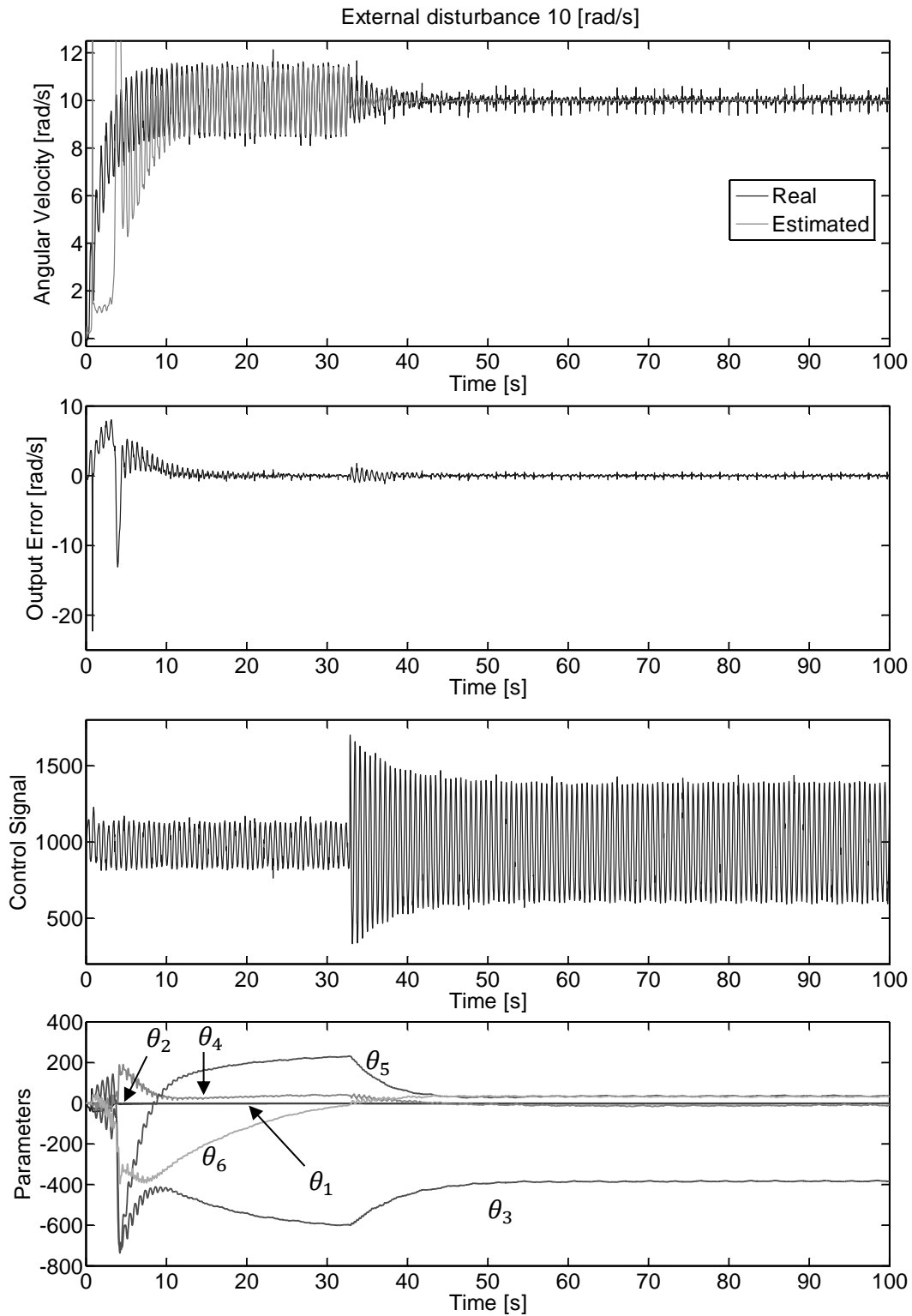


Figure 6.5: Experiment 1.2 run.

6.1.2.3 Experiment 1.3: Discrete modeling, identification and feed-forward control

The identification model used in this experiment is a linear in parameter discrete model. Therefore, the method of recursive least squares is used to identify the identification model as presented in section 4.3. The Matlab-Simulink diagram of the experiment is generally looks like the diagram of the previous experiment shown in Figure 6.4. The algorithm was started with the following initial parameters: Sampling period 0.0177 second, first order discrete input-output identification model dynamics, number of external periodic disturbance harmonics ($N_0 = 2$) and without any internal periodic disturbances ($N_1 = N_2 = N_3 = 0$), initial parameter vector $[1, 1, 0, 0, 0, 0]^T$, covariance matrix of $100\mathbf{I}$ and forgetting factor of 0.999. Then, the algorithm was compiled and ran in the real-time mode, at the beginning, only the PI controller and the identification were active, then at the time about 30 seconds, the feed-forward controller was activated by using the identified disturbance parameters directly to generate an anti-disturbance (compensation) signal added to the system input. The experiment run is presented in Figure 6.6.

6.1.2.4 Comments on external disturbance experiments

The first experiment 1.1 demonstrates the extension of the feedback PI controller to compensate the periodic disturbance by adding an internal disturbance model in form of an under damped second order linear system representing the periodic disturbance rejection filter as introduced in chapter 3. The advantage of the filter is its simplicity, its disadvantage though, it changes the closed loop feedback system characteristics in terms of relative stability this leads into a compromising solution between perfect periodic disturbance rejection and the closed loop feedback system relative stability.

The second experiment 1.2 uses the PI feedback controller mainly to set up the set point response characteristics and uses a feed-forward controller as an add-on only to compensate the rest of the external periodic disturbances with a minimum interaction with feedback controller objectives. This is the advantage of the algorithm, the pseudo feed-forward controller becomes a true feed-forward controller, especially, when the identification model parameters converge and the adaptation is set off or pre-identified disturbance parameters are used in the feed-forward controller.

The disadvantage though is the complexity of the algorithm by using a complex and computational extensive nonlinear least squares of output error algorithm to identify the continuous system dynamics and the disturbance parameters. The algorithm of the third experiment 1.3 is like the second one except by using discrete linear in parameter model identified by the recursive least squares method, makes this approach less computational extensive than the case in the second experiment.

However, the objective of these experiments is to demonstrate the capability of these methods to reject or to model, identify and to compensate the external periodic disturbances as an add-on to an existing feedback controller designed primarily for set point tracking demands.

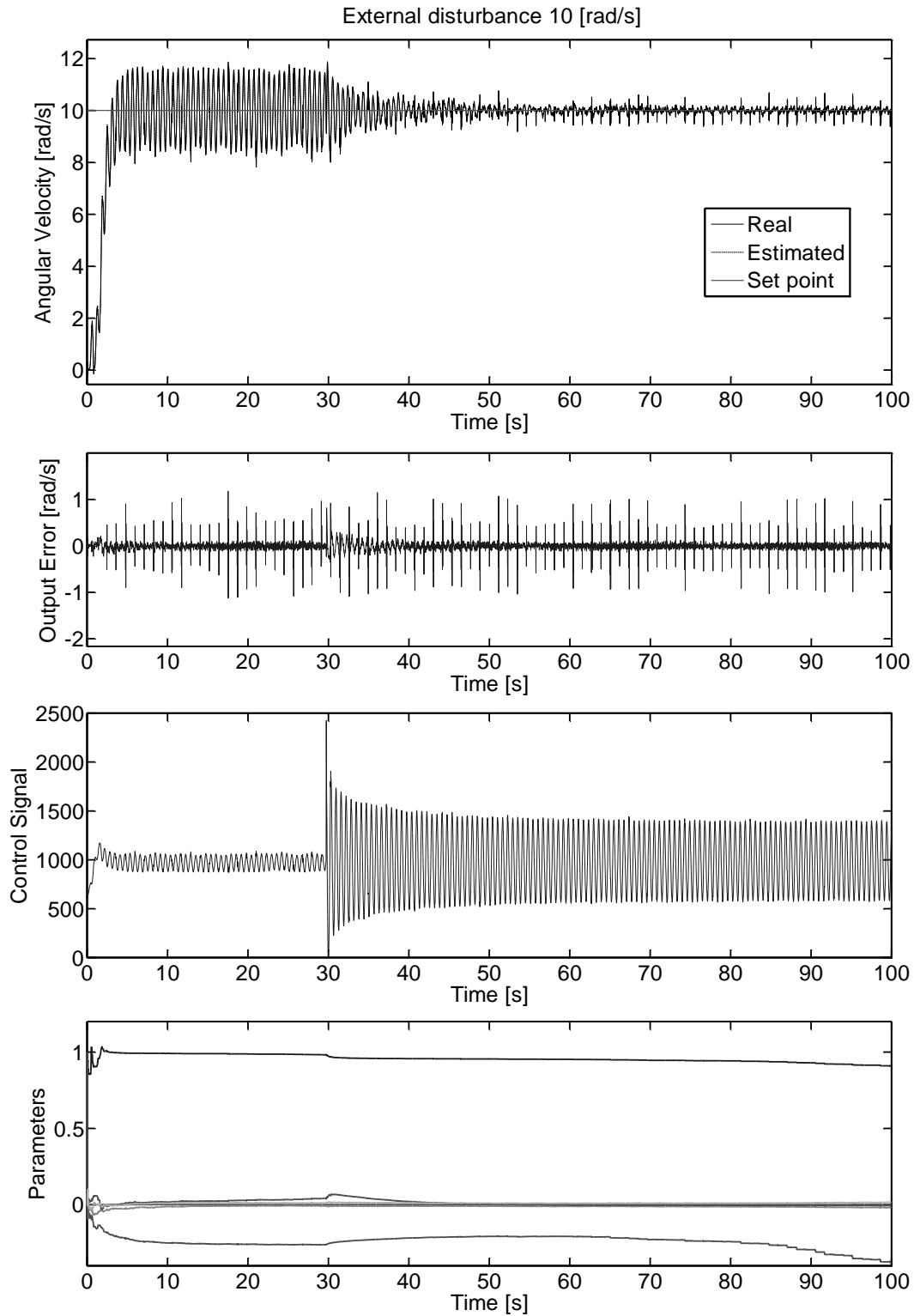


Figure 6.6: Experiment 1.3 run.

6.1.3 Internal Periodic Disturbance

The test platform has the same set up as in subsection 6.1.2. But in the next subsections, the drive-load velocity servo control system has an internal (self-excited) periodic disturbance. The internal periodic disturbance is also a state dependent, which in this case is the (measured) angular position; therefore, it is defined as

$$d(t) = 1000 \sin(\varphi_m(t)). \quad (6.3)$$

This is used as an input to the secondary disturbance motor. The system is constructed as PI feedback controller to ensure set point tracking demands; moreover, a feed-forward controller is used as an add-on to compensate the (emulated) self-excited periodic disturbance.

In the following, experiments are presented to demonstrate the use of continuous as well as discrete identification models to identify and compensate these emulated internal state dependent disturbances.

6.1.3.1 Experiment 1.4: Continuous modeling, identification and control

The periodic disturbance feed-forward controller is used to identify and compensate the internal periodic disturbances. The experiment Matlab-Simulink diagram is shown in Figure 6.7. The identification algorithm was started with following parameters: Sampling period 0.0177 second, first order continuous input-output identification model dynamics, number of harmonics ($N_0 = 0$, $N_1 = 2$, $N_2 = N_3 = 0$), initial parameter vector $[-1.94, 0.0146, 0, 0, 0, 0]^T$, initial covariance matrix $1000\mathbf{I}$ and forgetting Factor 0.998.

The experiment run is presented in Figure 6.9, where at the beginning only the PI controller for set point tracking objective and system identification were active, afterwards, at about the time 27 seconds, the feed-forward disturbance compensation was activated. This had yielded the final parameter vector $[-1.0674, 0.0110, -405.1971, 23.1688, 24.5031, -4.0879]^T$. The disturbance parameters have taken almost 13 seconds to get to the optimal ones.

6.1.3.2 Experiment 1.5: Discrete modeling, identification and control

In this experiment, the system structure and their parameters are the same as given in previous experiment, except that the identification model is discrete linear in parameters, therefore, the method of recursive least squares is used to identify the parameters of the identification model. Moreover, a discrete first order filter with pole at 0.9 is used to smooth the measured position signal as shown in Figure 6.8.

The algorithm was started with the following parameters: Sampling period 0.0177 second, first order discrete input-output identification model dynamics, number of harmonics ($N_0 = 0$, $N_1 = 1$, $N_2 = N_3 = 0$), the identification model initial parameter vector $[1, 1, 0, 0]^T$, which is first order discrete difference equation with time delay of two sampling periods, initial covariance matrix $100\mathbf{I}$ and forgetting factor 0.999.

Then, the algorithm was compiled and ran in the real-time mode, at the beginning only the PI controller and the identification were active, then at the time about 17 seconds the feed-

forward controller was activated using directly the identified disturbance parameters to generate an anti-disturbance (compensation) signal added to the system input. The experiment run is presented in Figure 6.10, which yielded the final parameter vector $[0.9916, 0.0001, -0.0357, 0.0007]^T$.

6.1.3.3 Experiment 1.6: Discrete modeling, identification and control with variable set point

This is the same as in experiment 1.5, except that, the set point is defined as two levels, 7 and 10 [rad/s], and linear time-varying from 7 to 10 and vice versa, see Figure 6.11, where it shows that the algorithm has succeeded in tracking and compensation of variable internal periodic disturbances.

6.1.3.4 Comments on internal disturbances experiments

In the last three experiments, the algorithm of modeling and identification of internal periodic disturbances has been implemented on the drive-load test platform. Where, the experiment 1.4 has used a continuous identification model, therefore, the continuous parameter identification output error method has been used to identify the identification model parameters.

On the other hand, experiment 1.5 has used a discrete identification model so that its parameters have been identified by using RLS method. Both methods of continuous and discrete modeling and identification have been used to reject the internal periodic disturbances successfully as shown in Figure 6.9 and Figure 6.10.

Moreover in experiment 1.6, a variable set point has been given to the system to test the algorithm behavior when the internal periodic disturbance consistently changes with the time as shown in Figure 6.11.

It is also to remark again that a quick parameter convergence happens usually when the adaptive feed-forward controller is activated as direct adaptive controller (experiment 1.4 is at time 27 seconds and experiment 1.5 is at time 17 seconds, see Figure 6.9 and Figure 6.10 respectively), which enhances the identifiability and the identification conditions in closed loop.

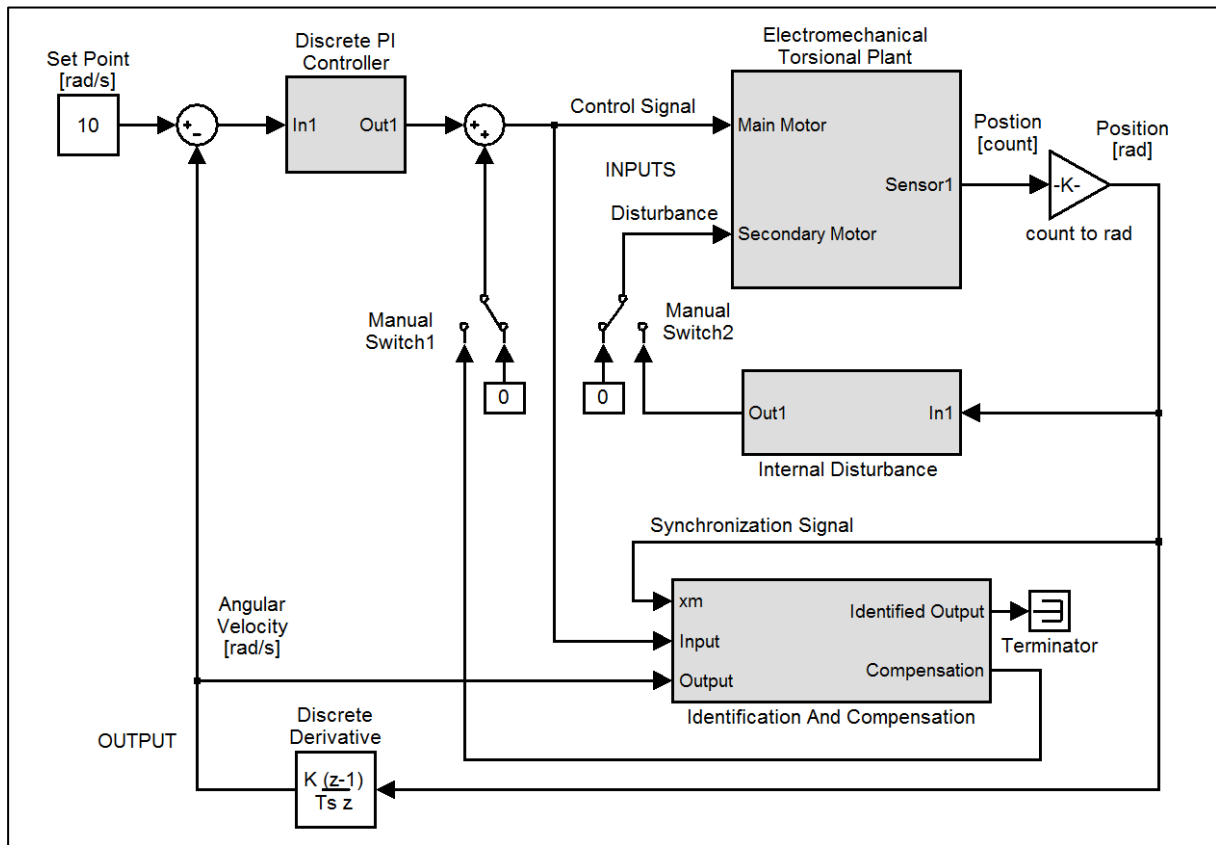


Figure 6.7: MatLab-Simulink diagram of experiment 1.4.

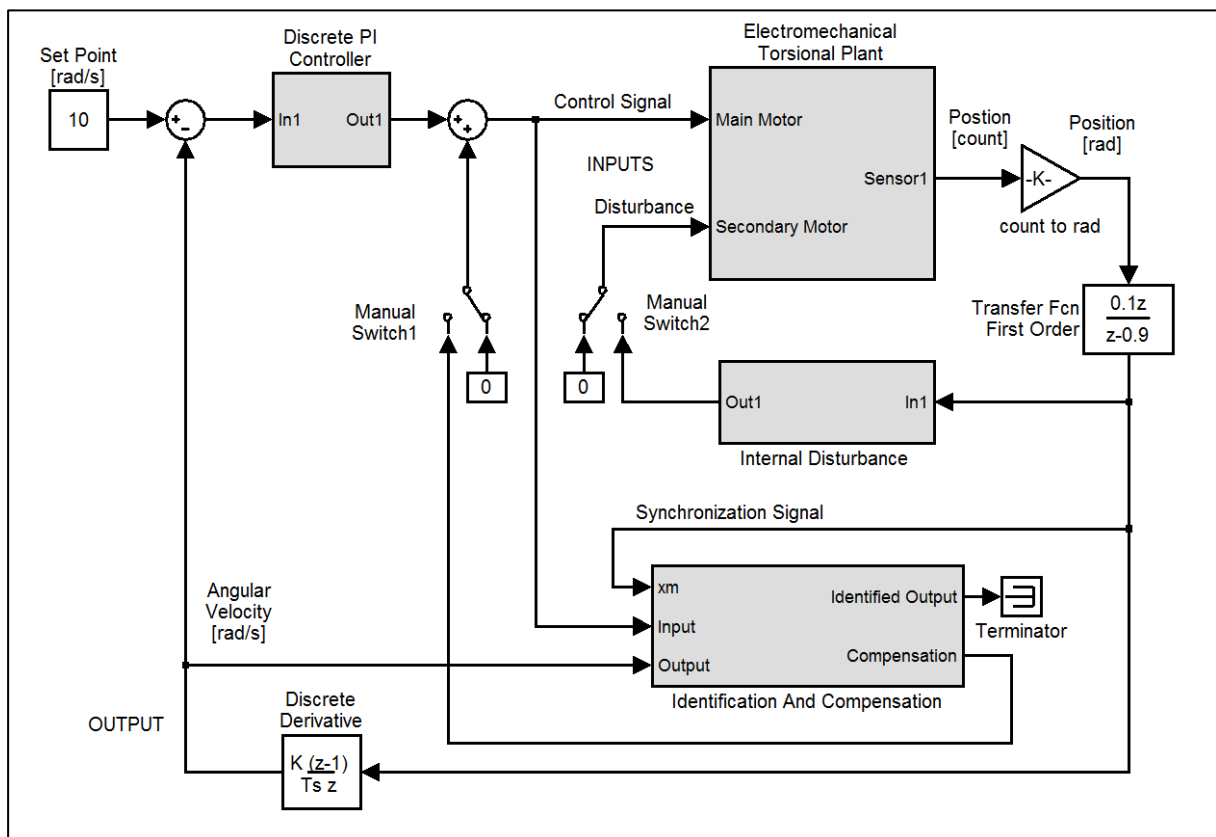


Figure 6.8: MatLab-Simulink diagram of experiment 1.5.

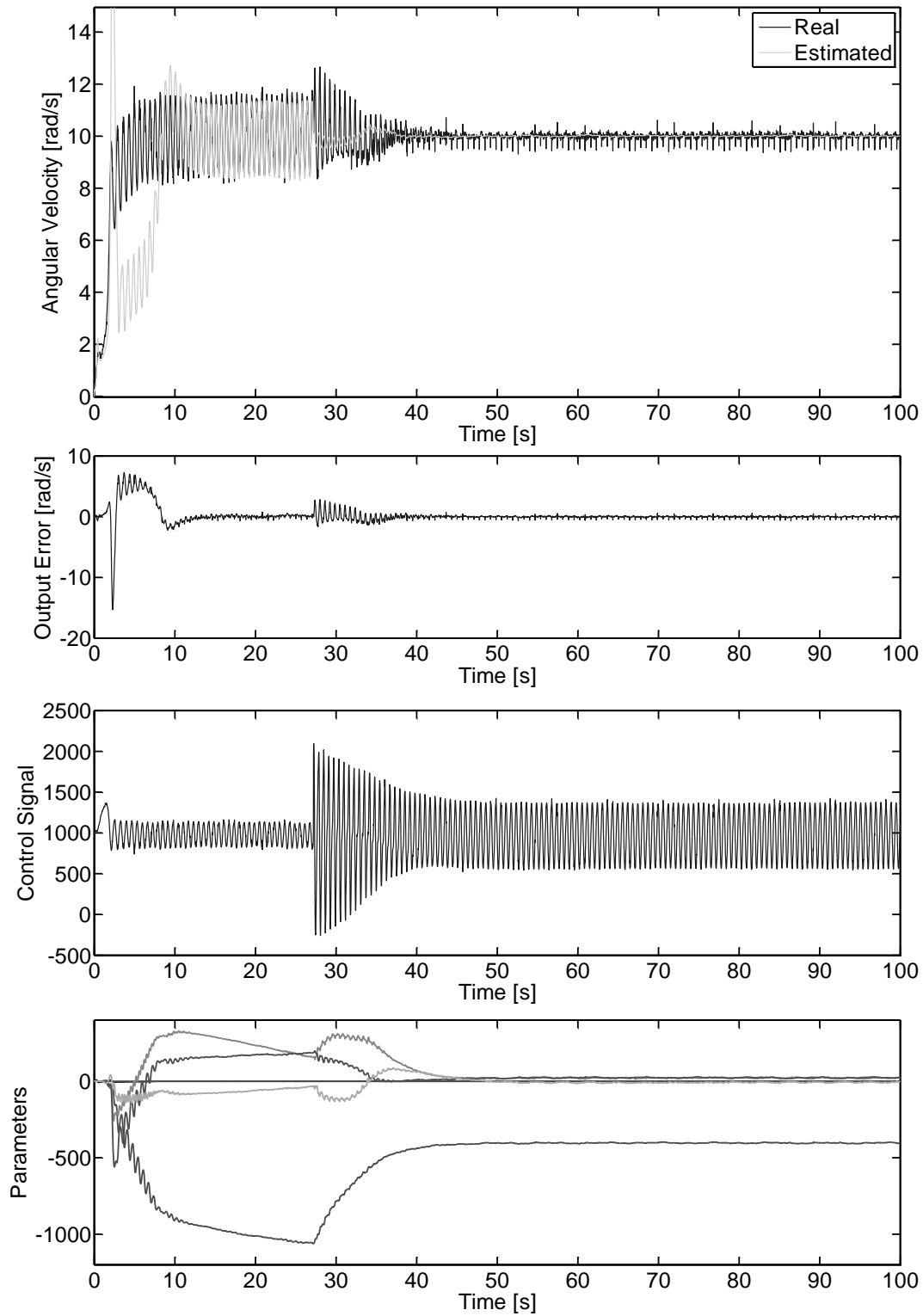


Figure 6.9: Experiment 1.4 run.

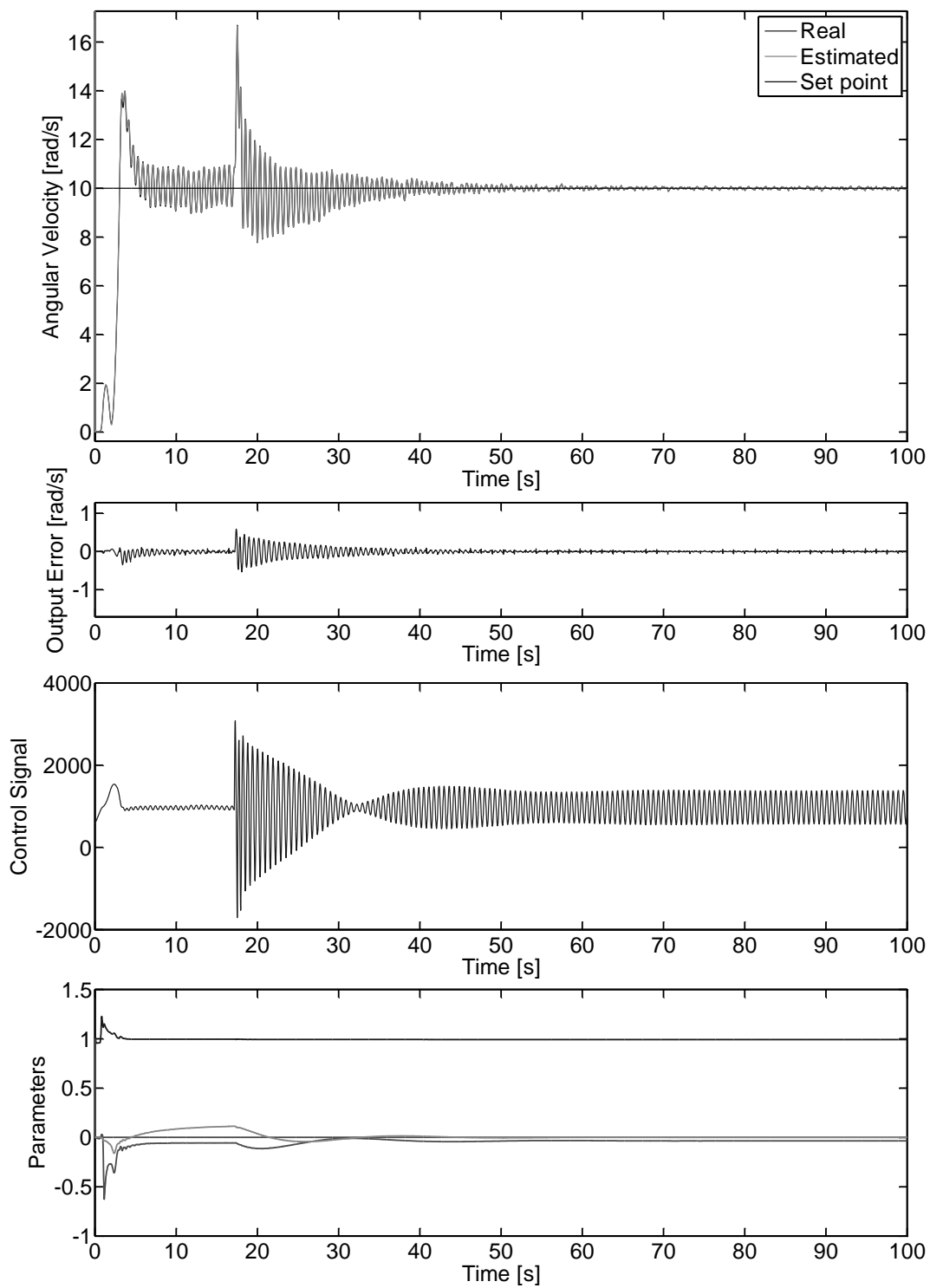


Figure 6.10: Experiment 1.5 run.

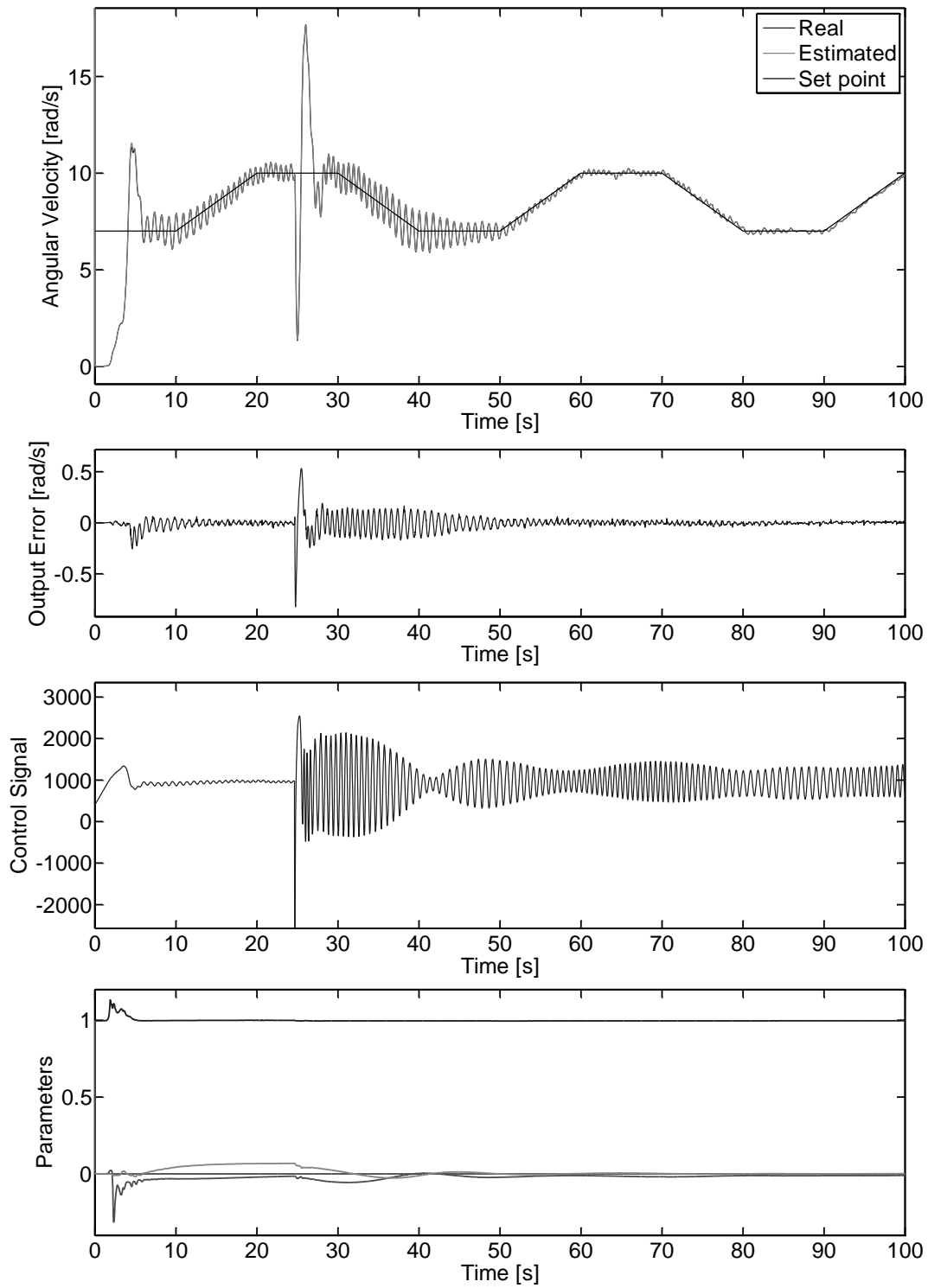


Figure 6.11: Experiment 1.6 run.

6.1.4 Comments on the Torsional Platform Experiments

As it can be seen, the experiments have demonstrated that the developed algorithms in chapter 4, for continuous and discrete linear or nonlinear in parameters identification models, were very successful in real-time applications where they showed that as the identification model parameters converged, the dynamics as well as the disturbance parts, the disturbance parameters could be used to generate an anti-disturbance compensation signal that cancels the periodic disturbance effect on the output. This means, the direct process identification in closed loop, by using open loop identification methods, was also successful, particularly when the algorithm is switched from the passive to the adaptive mode. One problem that accompanies this approach is that the inactive adaptation will not get quickly to the optimal parameters, particularly, when the set point is not consistently exciting, e.g. step function, until the adaptive mode is activated.

One solution is to use closed loop process identification methods. For example, indirect closed loop identification methods, where the closed loop system is identified then by considering the controllers in the closed loop, the process model is computed. This approach will guarantee the identifiability of the process in the closed loop provided that the set point is persistently exciting. The drawback of this approach is the intensive and complex online computation of the identification model from the identified closed loop model and the feedback controller.

Furthermore improvement can be done by the extension of the simple certainty equivalence adaptation strategies to implement the adaptive dual control methods, which they care about both the control and the parameter estimation certainty to generate a cautious control signal accordingly, and to guarantee optimum conditions for the parameter estimation (Filatov and Unbehauen 2000; Filatov and Unbehauen 2004).

The nonlinear in parameter identification models employed in this algorithm are explicit and direct in terms of model structure, especially the continuous ones, but their identification computations are rather complex and extensive. On the other hand, the linear in parameter identification models are complex and have many redundant parameters especially when the system has second order dynamics or higher, but their parameter identification computations are relatively simpler than those of nonlinear in parameter ones.

Finally, the developed algorithms of modeling, identification and control of periodic disturbances have been applied and tested successfully on the torsional test platform, where the algorithms have shown its capability to compensate both the external and the internal periodic disturbances in real-time control.

6.2 Self-excited Machine Test Platform

In this section, the developed algorithm of identification and control of periodic disturbances is tested in real-time control on a drive-load system with crankshaft load mechanism. The system is first described in general, then frequency spectrum analysis, set point tracking and internal periodic disturbance compensation experiments are presented.

6.2.1 Test Platform Description

The test platform, as shown in Figure 6.12 and Figure 6.13, consists of a drive-load machine. The drive part is a synchronous-motor (type: MDD1-33-3G6-CFA), which is driven by Variable-Frequency-Drive unit (VFD) (type: CDD32.008.C1.1), both are from the SERVAX DRIVERS Company. The load part is simply a crank mechanism connected to a spring, which due to its eccentric structure it generates the self-excited periodic disturbances that are dependent on the crankshaft angular position. The drive-load machine is mechanically connected by a rubber timing belt.

The VFD unit is powered from the AC mains 230 volts through a circuit breaker equipped with emergency stop button and enable/disable automation functions. Moreover, the unit is set up as torque control unit that can be controlled by an analogue external signal, where in this case it comes from the real-time controller board. Also, the motor has an embedded angular position incremental encoder, where its signal is sent to VFD unit, which also in turn resends this signal to the real-time control board. For more details about the setup, start-up or more information, please refer to the manufacturer manual (Servax Drives 2003a, b and c).

The Real-Time Controller Board (RTCB) is the (DS1104 R&D) board from dSPACE Company. The board is installed within a host PC, using one of its PCI extension slots. The Matlab-Simulink software from Mathworks Company and Control Desk from dSPACE Company are installed in the host PC and used as a working environment to develop a rapid prototyping real-time control applications (dSPACE 2005a). So, having Matlab-Simulink developed control algorithm, this can be compiled and send to the RTCB to be implemented in real-time, and the Control Desk helps to interface and control this implemented algorithm online and also in real-time control (dSPACE 2005b), for more information about rapid control prototyping in general refer to, for example, Abel and Bollig (2006).

The load side, as presented schematically in Figure 6.14, is a crank mechanism with crank radius ($r = 0.03$ [m]) and link rod ($l = 0.07$ [m]). The crankshaft mechanism translates the rotational motion φ into linear motion x , or in other words, the crankshaft moves the mass m_c that linked to a spring up and down against the friction force, where the spring and friction elements are represented by a stiffness factor k_c and a damping constant d_c respectively.

Eventually, the drive-load machine together with hardware and software accessories is set up to construct an angular velocity servo control system, where a feedback PI controller is used to guarantee the set-point tracking characteristics. Moreover, the designed periodic disturbance feed-forward controller is implemented to compensate the rest of the self-excited oscillations in the angular velocity.

In the next, the developed algorithm of periodic disturbance feed-forward controller is applied on the test platform. Where first, the frequency spectrum estimates for different angular

velocities are computed and analyzed for the cases without and with periodic disturbance compensation of four harmonics, the principal frequency and three super-harmonics, as well as for the case of five harmonics, the principal frequency plus one sub-harmonic and three super-harmonics. Moreover, the periodic disturbance compensation algorithm is tested when the set point is a stepwise variable and linear time variable. The feedback PI controller parameters remain the same for all next experiments.

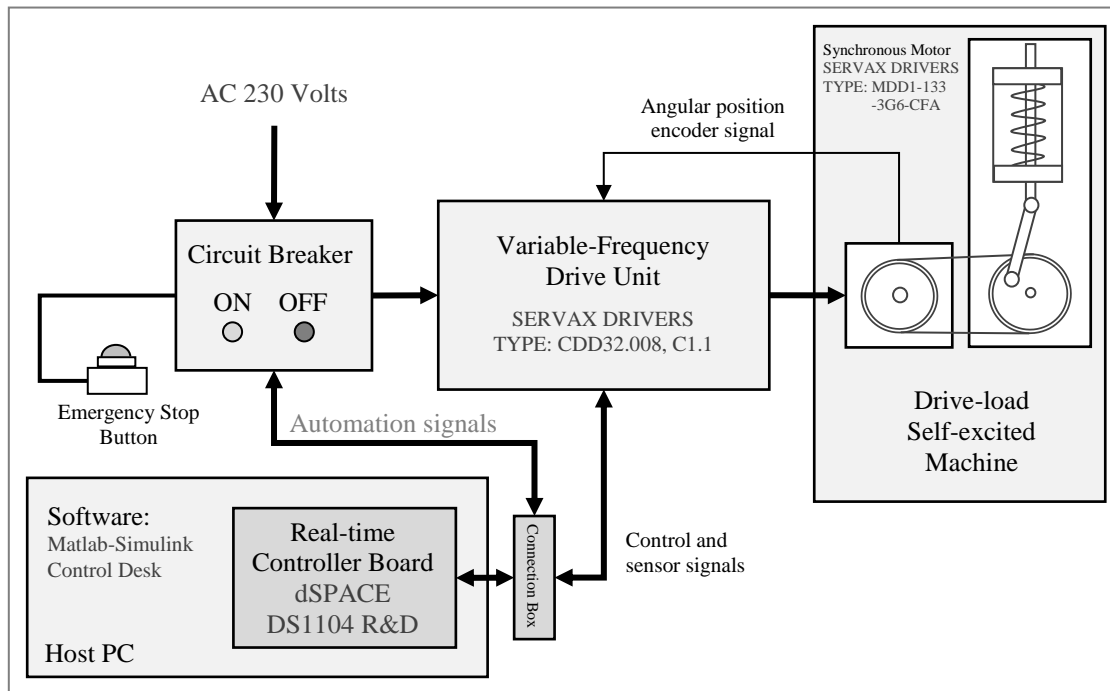


Figure 6.12: The schematic diagram of the drive-load self-excited machine.

6.2.2 Frequency Spectrum Analysis

First of all, the oscillations of angular velocity at different constant values are analyzed by computing their finite Discrete Fourier Transform (DFT) by using the Fast Fourier Transformation (FFT) algorithm to estimate their frequency spectrums (Smith 1997). This will help in deciding how many harmonics are important that should be considered in the identification model in order to be compensated.

6.2.2.1 Frequency spectrum without periodic disturbance compensation

In this experiment, the self-excited machine is under the PI feedback velocity servo controller with the parameters ($K_p = 0.02, K_I = 0.02$). The system is run and recorded at different angular velocities of 10, 15, 20 and 25 [rad/s] without any periodic disturbance compensation. These runs, their time response and frequency spectrums, are presented in Figure 6.15, where the time responses are from the top to the bottom on the left side and then their corresponding finite DFT frequency spectrum estimates are computed and presented at the right side. From the plots in Figure 6.15, it can be seen that the system has a principal frequency equal to its rotational velocity as well as super-harmonics equal to the multiple of rotational (velocity) frequency. Therefore, in the next experiment the same as in this experiment is done but with the compensation of the principal and three super-harmonics of the periodic disturbance frequencies.



Figure 6.13: The drive-load self-excited rotational machine.

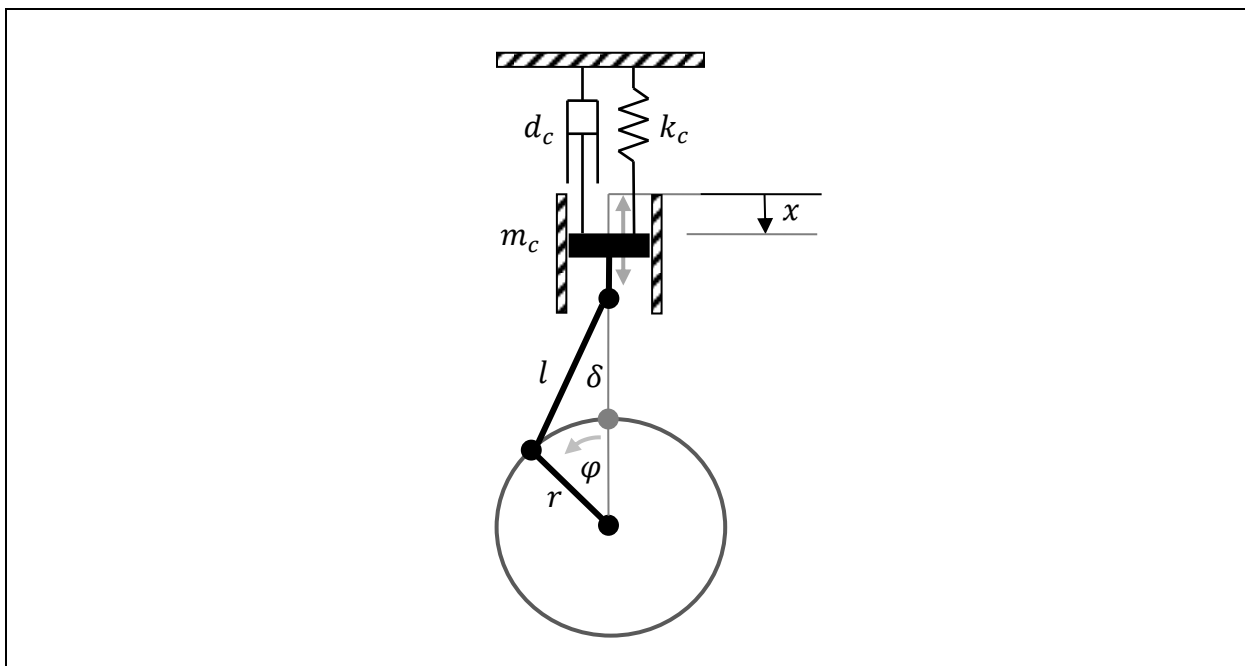


Figure 6.14: The crankshaft mechanism moving a mass, spring and damper load.

6.2.2.2 Frequency spectrum with periodic disturbance compensation

This experiment is the same as the previous one but the periodic disturbance feed-forward controller is run with 4-harmonics periodic disturbance compensation, which are at rotation rates of 10, 15, 20 and 25 [rad/s]. The uncompensated and compensated outputs are presented on the left side of Figure 6.16, while their uncompensated and compensated output finite DFT frequency spectrum estimates are presented on the right side. By a careful investigation of the signals on the left and their spectrum on the right, it can be seen that there is actually still some rest oscillations due to sub-harmonics; the first sub-harmonic is about 80% of the principal frequency. So in the next experiments, the principal frequency plus one sub-harmonic and three super-harmonics are considered in the periodic disturbance model and consequently compensated.

6.2.3 Identification and Control of Periodic Disturbances

In the following experiments, the identification model is a discrete first order linear in parameters as defined in section 4.3 with the following parameters ($N_0 = 0$, $N_1 = 5$, $N_2 = N_3 = 0$) with the internal periodic disturbance frequency vector is chosen to be, according to the last subsection spectral analysis, as

$$[0.8 \ 1 \ 2 \ 3 \ 4]\omega, \quad (6.4)$$

where ω is the system rate of rotation (angular velocity [rad/s]). So, the frequencies considered in the periodic disturbance model are the principal frequency plus one sub-harmonic and three super-harmonics. These frequencies are going to be modeled, identified and used to compensate the system periodic disturbances in the following experimental examples.

6.2.3.1 Identification and control of 5-harmonics

In this experiment, the set point input is a step from zero to 15 [rad/s] at time about 14 seconds. The experiment run graphs are presented in Figure 6.17, where the first graph from the top shows the real system (measured) output, the identified output and the set point at 15 [rad/s]. The operation was made so, at the beginning only identification is done then at time about 48 seconds the feed-forward periodic disturbance compensation is activated to compensate the identified 5-harmonics periodic disturbance. Moreover, the second graph presents the output error while the third graph plots the control signal, the fourth graph shows the estimated dynamic parameters a_1 and b_1 , while the fifth and the sixth graphs show the periodic disturbance model parameters $\alpha_{1,i}$ and $\beta_{1,i}$ respectively. Figure 6.18 reveals a comparison between the system output of uncompensated case, compensated with four harmonics case and the case of one principal-harmonic, one sub-harmonic and three super-harmonics, while Figure 6.19 compares their correspondent finite DFT frequency spectrum estimates.

6.2.3.2 Variable set point tracking and periodic disturbance rejection

This experiment has been done to show that the algorithm is able to cope with the set point sudden change. The set point changes stepwise from 10 to 15, 20, 25 and 30 [rad/s]. The experiment run is presented in Figure 6.20. Also, another run is done when the set point is saw-tooth signal. So in this run, the algorithm was tested under constantly varying set point. This run is presented in Figure 6.21. As the figures show, the algorithm could cope with the set point change and compensate the targeted periodic disturbance.

6.2.4 Comments on the Self-excited Machine Experiments

The algorithm of modeling, identification and control of periodic disturbances has been applied and tested experimentally in real-time control on a real mechanical process successfully. Where, first the frequency spectrum analysis is done for the self-excited drive-load machine output oscillations, and upon that the dominant oscillation harmonics are considered in the internal periodic disturbance part of the identification model.

Therefore, the algorithm is applied by using the identified internal periodic disturbance part parameters of the identification model to generate the anti-disturbance signal to cancel the internal periodic disturbances that generated by the angle dependent load. The algorithm has been applied with constant, staircase variable and linear time variable set point cases to demonstrate the applicability of the algorithm locally and globally.

Moreover, as expected and said in subsection 6.1.4 and subsection B.4.1, that although identifyability and identification conditions in the closed loop are very poor in general. But exception could be made, if the controller changes or it is a complex one as in this case, when it is adaptive, then the plant identifyability and identification conditions became better in the closed loop. This can be seen from the estimated parameter plot as the feed-forward controller is activated and becomes online adaptive by using the certainty equivalence principle, the parameters converge more quickly to the optimal one, so that the disturbance parameters could be successfully used to compensate the periodic disturbances.

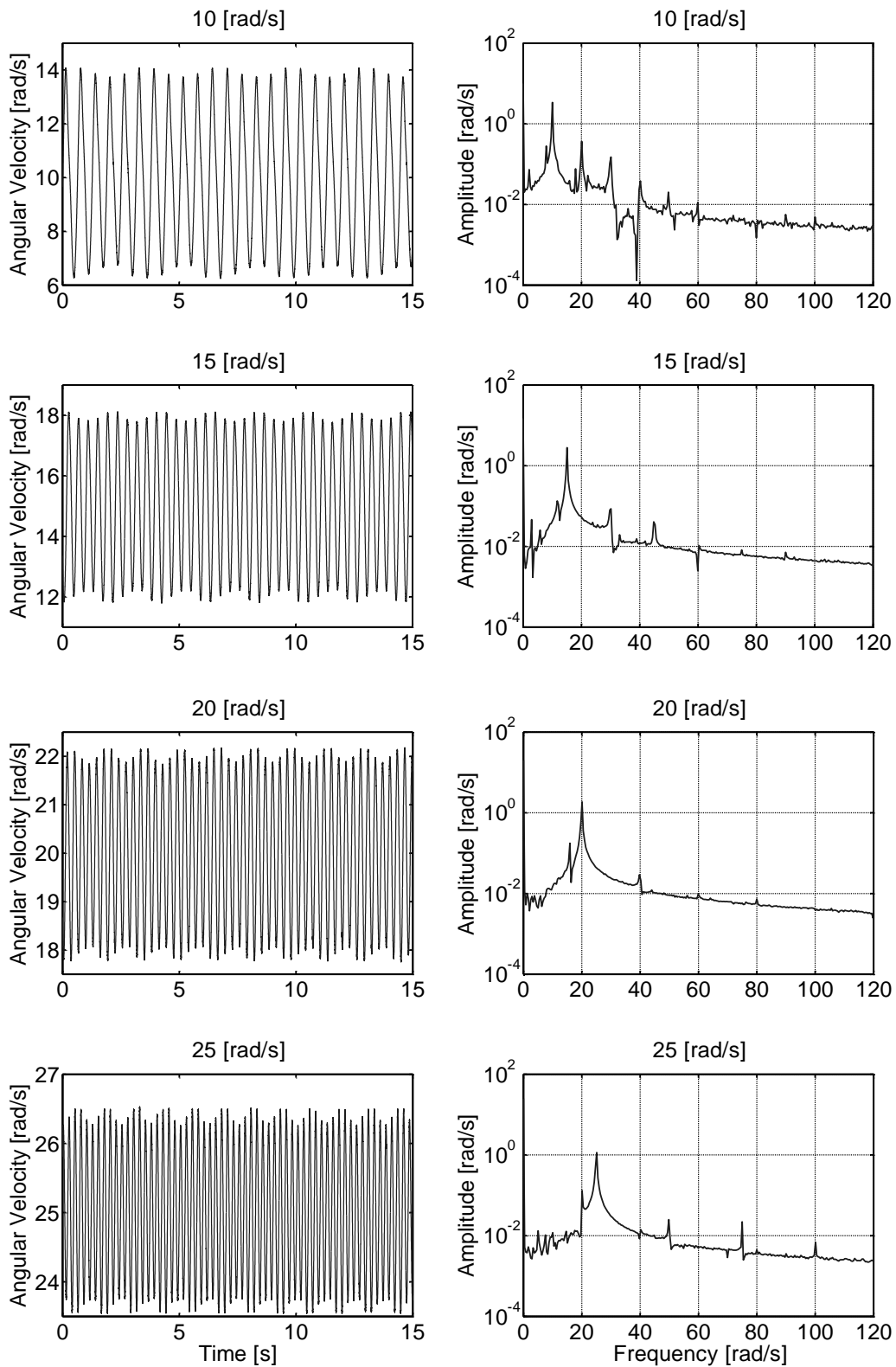


Figure 6.15: Time and finite DFT frequency spectrum estimate without compensation.

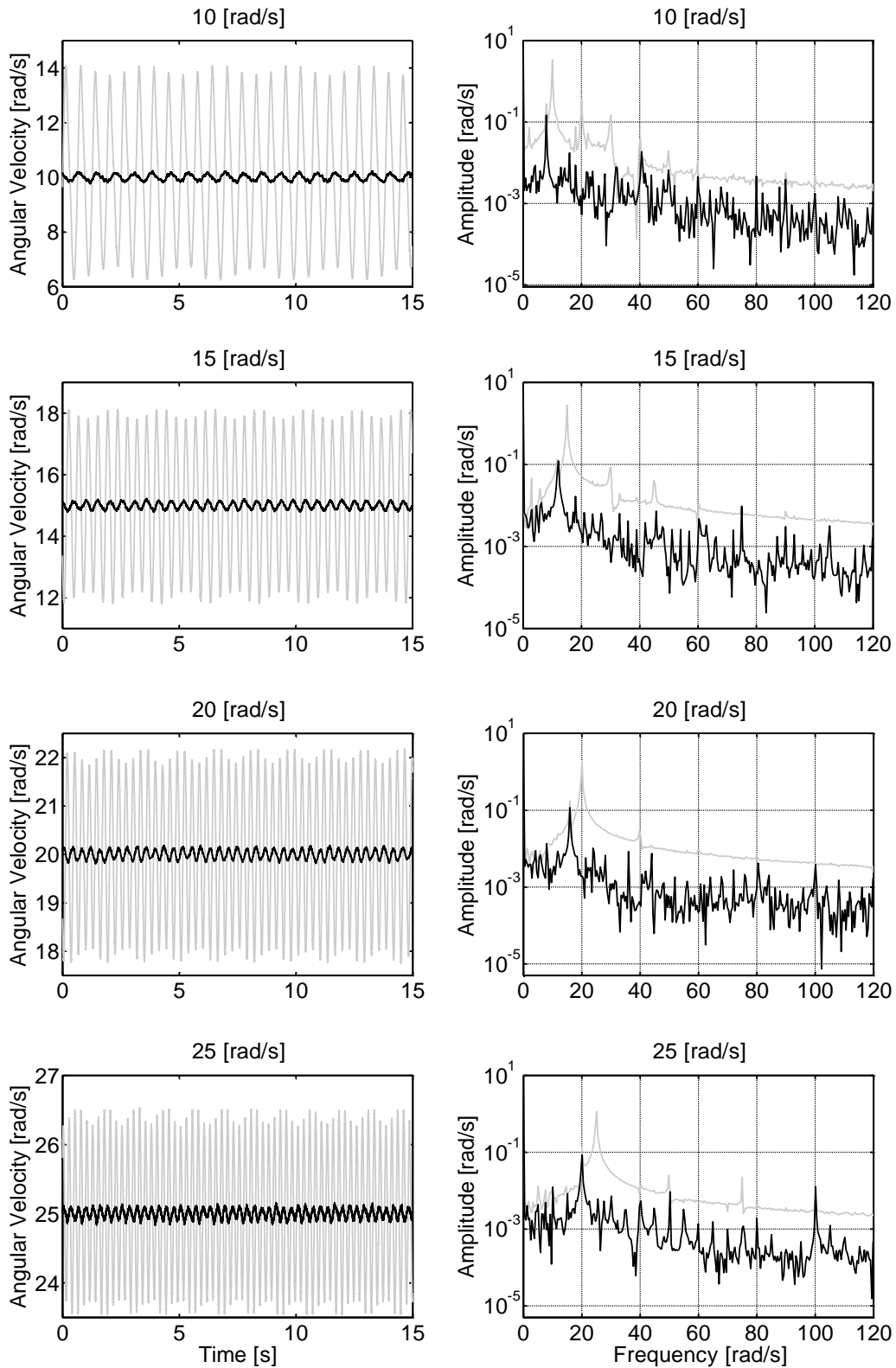


Figure 6.16: Time and finite DFT frequency spectrum estimate without compensation (gray line) and with 4-harmonics compensation (black line).

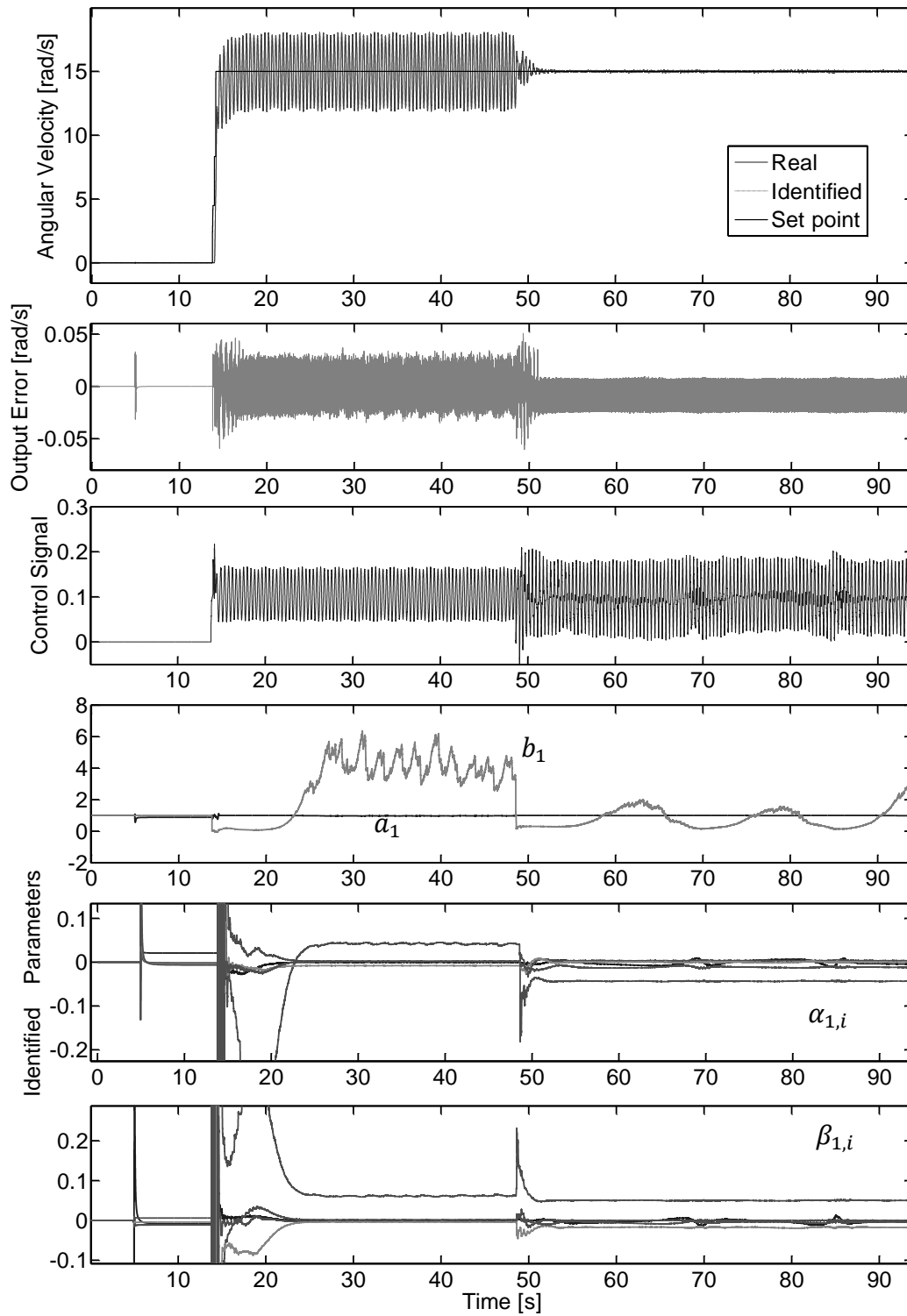


Figure 6.17: Identification and compensation of 5-harmonics.

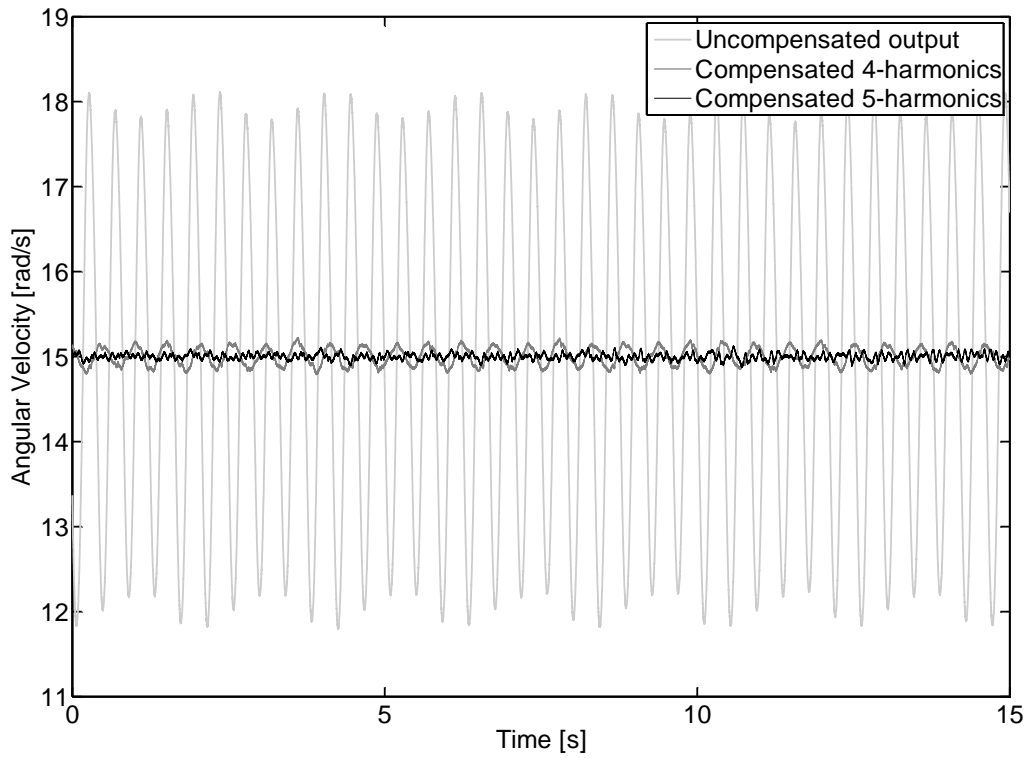


Figure 6.18: System output of uncompensated; compensated with 4-harmonics and compensated with 5-harmonics.

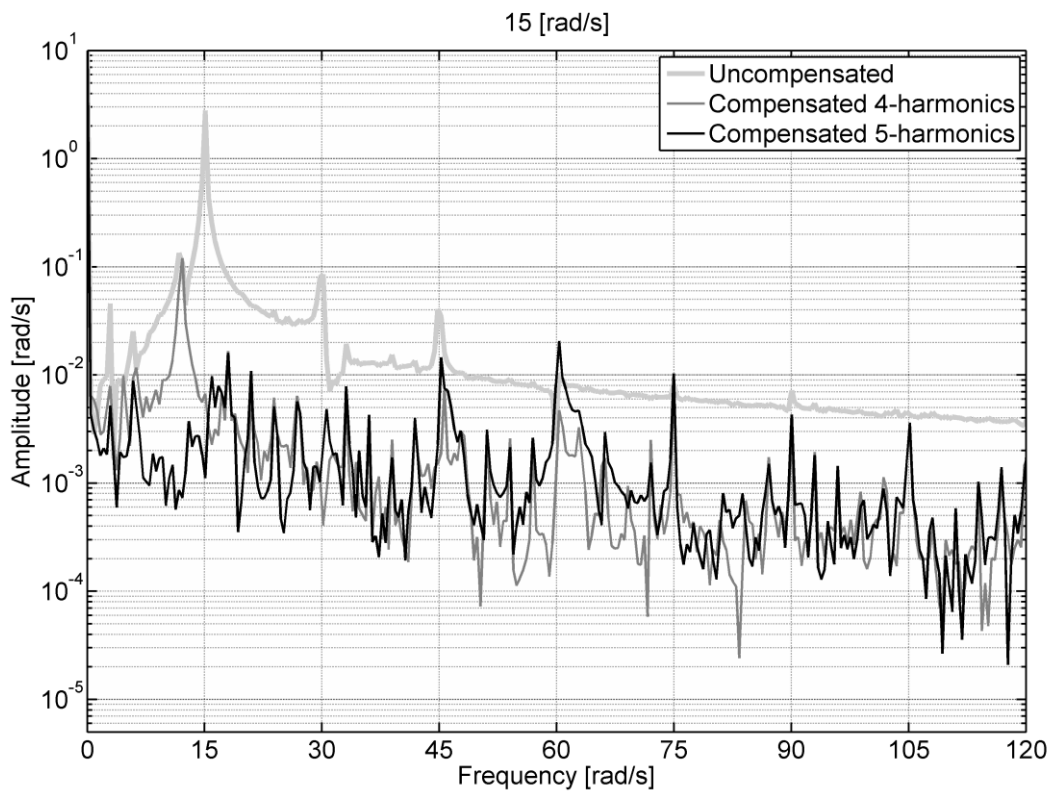


Figure 6.19: The finite DFT frequency spectrum estimates of uncompensated; compensated with 4-harmonics and compensated with 5-harmonics.

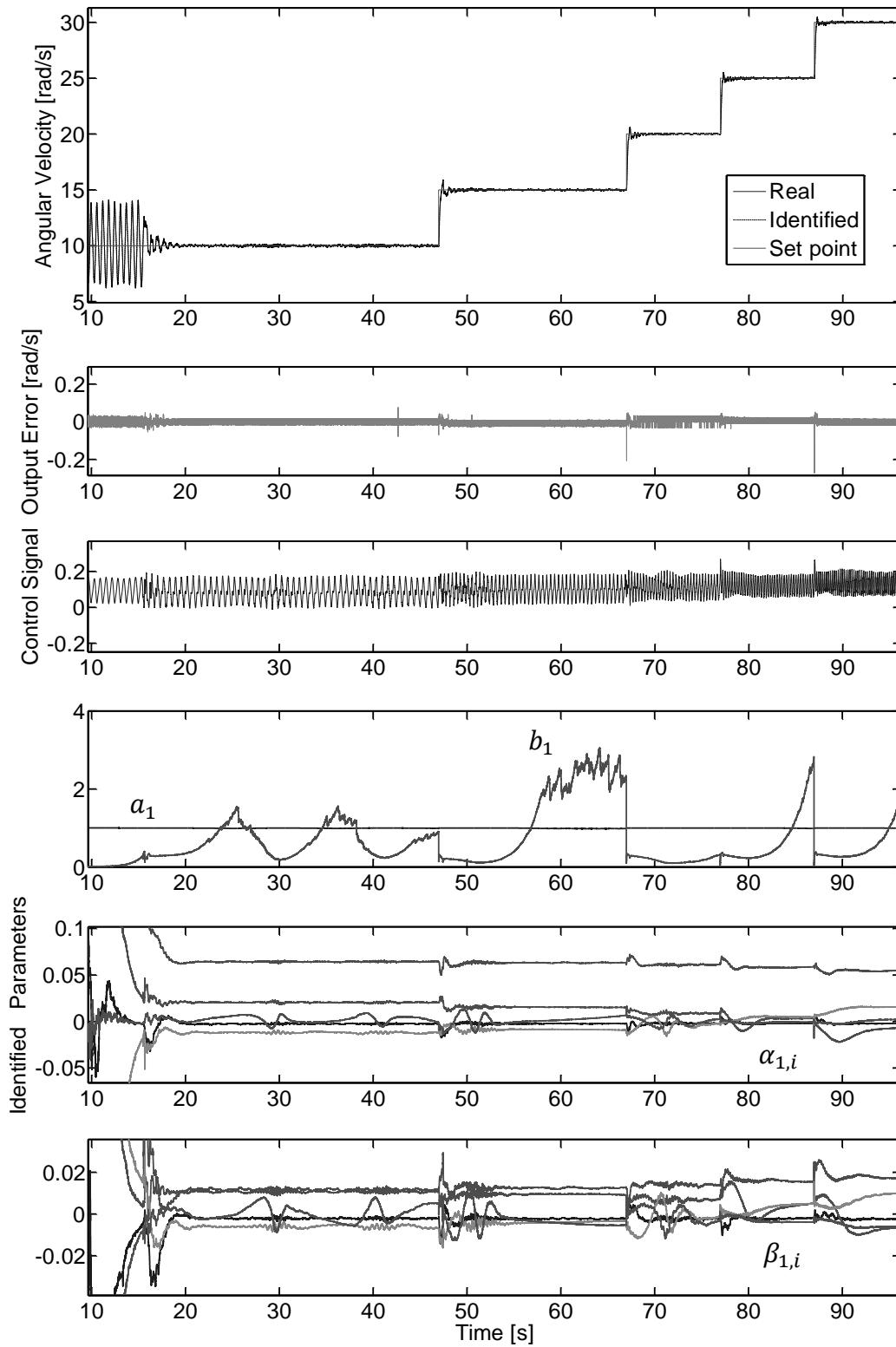


Figure 6.20: Stepwise variable set point and periodic disturbance compensation.

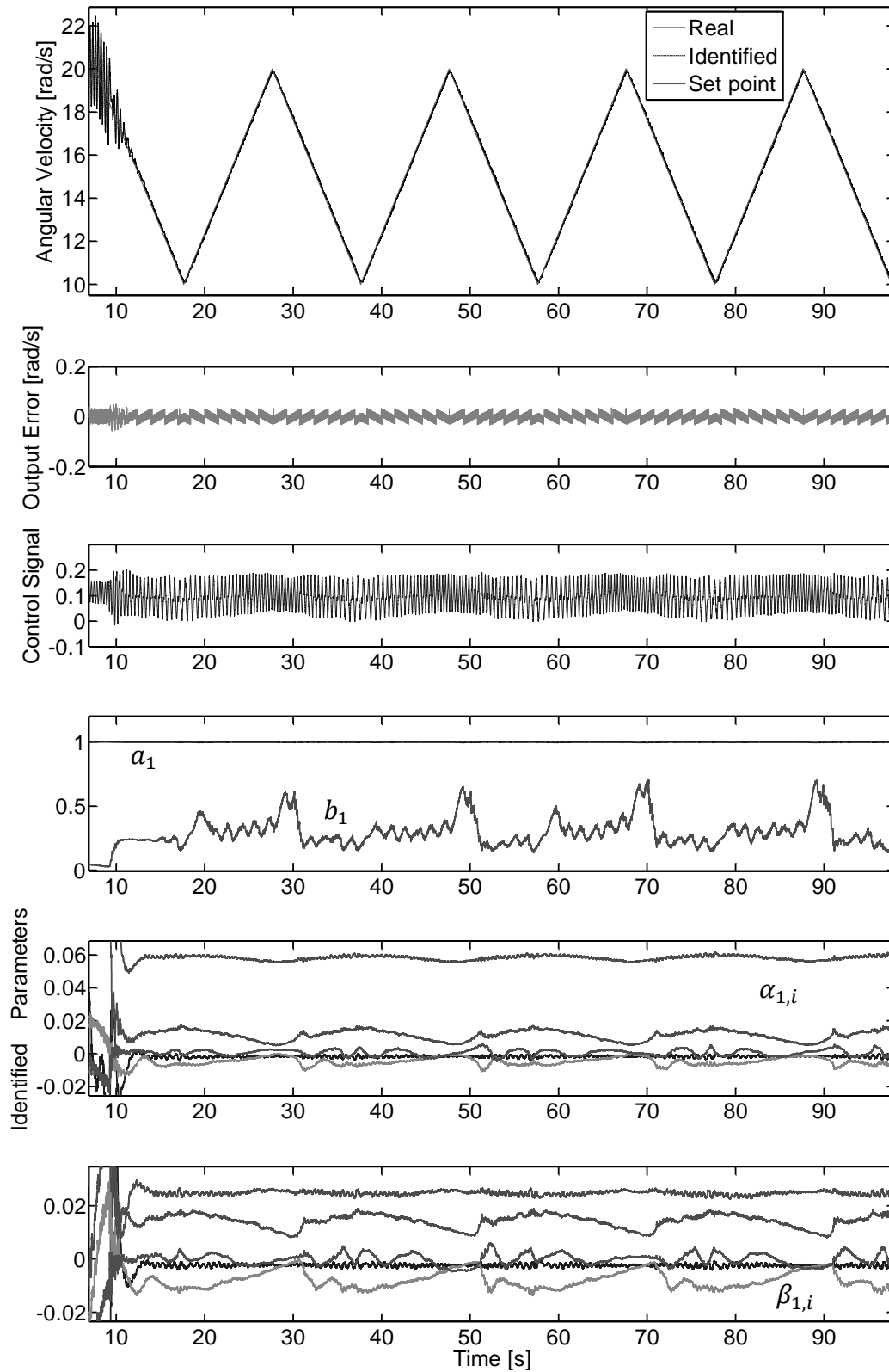


Figure 6.21: Consistently varying set point tracking and periodic disturbance compensation.

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The control problem, addressed in this work, is based on that the system has a set-point tracking objective, which is set as main target to achieve. Moreover, there are oscillations interpreted as periodic disturbances that also needed to be rejected in order to obtain higher performance level. However, this has to be done with minimum interaction with the already existing feedback controller preserved mainly to guarantee the set point tracking objective.

The problem is very well solvable by using a Linear Time-Invariant Internal Model Principle (LTI-IMP) based feedback control, for example, realized either as periodic disturbance rejection filter, periodic disturbance observer/compensator or generally as a repetitive controller. But, there will be an inevitable interaction between the set point tracking and the periodic disturbance rejection design demands, where it could at least happen in the system transient behavior or affects the closed loop relative stability, particularly, when the disturbance frequency is outside the set point tracking demanded bandwidth. Since adding extra filter at this high frequency, will consequently add extra lag and will obviously affect the set point tracking design transient characteristics.

For a minimum interaction between these two design objectives, a new feed-forward periodic disturbance compensation method was introduced in this work. It was actually inspired from the active noise and vibration control society applications. For example, the filtered X-LMS methods are mainly direct adaptive feed-forward algorithm, which in return makes them as a pure feedback control. These are particularly not suitable for this control problem. Since, they need to be tuned online and this will affect the performance of the set point tracking objective as well as the disturbance rejection, particularly at the initial tuning phase when the parameters are far from the optimal ones.

The problem of the direct adaptive feed-forward control has been solved in this work by transferring the direct adaptation problem into system identification (optimization) problem. By doing this, it is now possible to make both direct and indirect identification and adaptation schemes. Direct is when the identification and controller parameter adaptation are active, while indirect is when the identification is active but the controller parameter adaptation is passive. Indirect with offline identification is needed especially when the open loop process input-output data is available, then the parameters can be identified offline in advance and used in the feed-forward control law. In this case, the feed-forward controller is a true feed-forward controller, which yields a minimum interaction to the closed loop designed characteristics.

Also, indirect identification can be made online, especially, when there is no pre-recorded data available from the process or the initial parameters are uncertain. With this mode, the algorithms can be started as a cautious procedure until the parameters converge, then the parameter identification can be set off and the converged parameters can be used to construct a “true” feed-forward controller to compensate the periodic disturbances.

However, in real applications, the process usually works under a closed loop feedback controller, and the operating conditions do not allow interrupting the operation, for example, opening the loop and recording input-output data for an identification experiment. Therefore, the parameter identification of the process is done in the closed loop which leads to the problem of the identifiability condition in the closed loop is no longer more valid, particularly, when the applied identification algorithm is originally designed for open loop identification. A solution to this problem is to use closed loop identification methods, e.g., indirect process identification methods in closed loop by identifying the closed loop system model and computing the process model indirectly by using the known controller model. This is a computationally intensive method, especially when applied as an online parameter identification algorithm in an adaptive control system.

Fortunately, the direct algorithm of direct online process parameter identification and directly using them in the controller, according to the certainty equivalence principle, solves the problem of the identifiability and the identification conditions in the closed loop (Appendix B.4.1). But this makes the feed-forward controller to work as a feedback controller and will make an interaction with the closed loop characteristics when the parameters are far from the optimal one. But this sacrifice will end when the parameters converge and the adaptation is switched off, making the pseudo feed-forward controller back to a true one, which is the advantage of this algorithm over methods based on the LTI-IMP.

The system identification of the periodically disturbed process is done by constructing an identification model to represent this periodically disturbed process. The identification model is made by two parts, a dynamic part and a disturbance part. The dynamic part describes the system input-output behavior and can be represented by (linear/nonlinear) differential or difference equations, while the disturbance part is made to represent the external disturbance and or the internal disturbance in the input disturbance format. The input disturbance format is chosen in order to make the later use of disturbance parameters in the feed-forward control law easy and straightforward without any extra transformations. The disturbance model is constructed by sine cosine sums that are function of the external disturbance frequency and or the internal disturbance synchronization signal (e.g., angular position in case of rotational systems).

Additionally, the internal disturbance function could have a linear or nonlinear dependence to system states in order to make a local or a global identification model. By modeling (globally) the nonlinear behavior of the process, that actually causes the problem of self-excited oscillations in the first place, makes it also possible to cancel (globally) this disturbing behavior. This is the main advantage of the method introduced in this work, the strategy of modeling, identification and control of periodic disturbances, over the classical methods based on LTI-IMP, for example, the periodic disturbance rejection filter.

Now, the choice of the parameter identification method, which is used to identify the identification model, is made according to the problem environment. For example, whether the model is discrete or (quasi) continuous, linear or nonlinear in parameters, or whether the process is directly measurable or needs some observations, deterministic or stochastic. For instance, if the process were a stochastic, then the Kalman filter or the dual extended Kalman filter could be used to estimate the states and the parameters of a nonlinear system. Besides,

the process could be stochastic and have large dead time which leads to a prediction and estimation problem.

Furthermore, by modeling the identification model as dynamic and disturbance parts, the adaptive algorithm can be theoretically and practically applied for both the feedback and the feed-forward controller, and by developing the states, the modern control design methods can be used to design a state feedback controller, for example, pole placement method. In general, one way of utilizing adaptive algorithms is to use them as automatic tuning algorithms that carry out the design quickly and find the optimal parameters. Therefore, when the parameters converge then the parameter adaptation can be frozen, hence the system characteristics can also be analyzed around these parameters.

The successfulness of the algorithm depends significantly on the capability of the implemented identification algorithm to identify the identification model and the capability of the identification model to represent (approximate) the process input to output and disturbance dynamics around the targeted operating region either locally or globally. Where only on the condition that: If the identification model parameters converge, and so the identification model makes a good representation of the process. Then, the disturbance parameters can be used to compensate the disturbance of the process. Otherwise, the parameters are useless, since they do not or cannot represent the real process.

The algorithm was applied and verified in simulation as well as in emulation of drive-load velocity servo control of rotational machines that have an external periodic disturbance (oscillation) source or and self-excited oscillation source caused by nonlinear angle dependent spring, damper or moment of inertia load elements and their combinations. The cases of rigid and flexible drive-load system were considered by using both continuous and discrete identification model dynamics. The algorithm was very successful to reject the targeted periodic disturbances both in simulation and in real-time drive-load system emulation. Furthermore, the algorithm has been applied and tested in real-time control on self-excited drive-load machine test platform where the algorithm was also very successful to reject the targeted self-excited oscillations generated due to the crankshaft load mechanism.

Advantage and disadvantage between the LTI-IMP based periodic disturbance rejection techniques, see table 7.1, either implemented as feedback controller or disturbance observer/compensator, over the algorithm developed in this work are almost the same. Since the modeling of the periodic disturbance in this work is also considered to be according to the IMP (Bodson 2004). However, the algorithm developed in this work needs adaptation, which is done as (online) identification and control, actually, this is an indirect adaptive scheme (Landau, Lozano and M'Saad 1998). On the other hand, the IMP disturbance rejection filter algorithm needs to know in advance the (accurate) system dynamics in order to make a design done in terms of closed loop relative stability, demanded set point response and disturbance rejection. This is usually an extensive and complex computation commonly done by using robust control theory techniques. Nevertheless, the advantage of this work algorithm is the explicit modeling and identification of the unwanted behavior that generates the periodic disturbances as a compensation function that can be directly embedded in a control system to compensate the disturbances in a specific operating region locally or globally.

Table 7.1: Advantages and disadvantages of feedback and (adaptive) feedforward techniques.

Technique		Description	Advantage/Disadvantage
Feedback control only		For both set point tracking and periodic disturbance rejection. By using IMP, for example, “inverse” notch filter for harmonic disturbances in particular or repetitive control for periodic disturbances in general.	- There will be an interaction between the closed loop set point design specifications and periodic disturbance rejection. + Only the disturbance frequencies need to be known.
Feed-forward		As an add-on to a pre-existing set point tracking feedback controller.	+ It does not interact with the set point closed loop design characteristics. - The disturbance signal should be (directly) available.
Feed-forward disturbance observer based		The disturbance signal is estimated by using a disturbance observer either transfer function or state space based, also here by implementing IMP.	- It works as a true feed-forward only when the observer model is exactly as the process model. - Otherwise, it is a feedback controller, particularly when it uses the process output to observe the disturbance, and it does interact with set point closed loop design characteristics. + Only the disturbance frequencies need to be known, as in the case of feedback control only.
Adaptive feed-forward methods	Direct adaptive methods	By estimating the controller parameters directly usually in a form of (anti-) disturbance parameters (amplitude, phase and the frequency if necessary) by using adaptation (optimization) algorithms.	- It has to be directly tuned online. - This will affect the system performance, when the algorithm starts with very poor initial parameters. - It works as feedback method, when it uses the process output in the parameter adaptation, which does interact with the set point closed loop design characteristics. + It will work as true feed-forward when the parameters converge and the parameter adaptation algorithm is set off.
	Indirect adaptive methods	By estimating the identification model parameters to represent the process input-output dynamics and the (external and internal) periodic disturbance parameters, and then the identified disturbance parameters are used to construct the feed-forward controller parameters, as presented in this work. Also, the identified input-output dynamic parameters can be used to compute the feedback controller parameters.	+ It can be started and tuned offline by using prerecorded input-output data of the process, this is important when the initial parameters are far from the optimal ones. - In certainty equivalence mode, it works as feedback method, which does interact with the set point closed loop design characteristics. + It will work as true feed-forward when the parameters converge and the parameter adaptation is set off.

7.2 Future Work

7.2.1 Theoretical

The certainty equivalence principle adaptive control strategies, presented in this work, can be and have been started and operated safely with the recommended precautions (for example, starting with very good initial conditions by using all available and proper pre-knowledge about the system). Nevertheless, the extension of the strategies to implement the closed loop identification methods (Landau, Lozano and M'Saad 1998), as well as, the extension of the simple algorithm that uses the certainty equivalence principle to adaptive dual control algorithm is therefore recommended to be explored, where both the control and the identification problems are considered in the analysis and the solution of the adaptive problem (Filatov and Unbehauen 2004).

Moreover, the developed algorithms in this work need to know the synchronization signals, the angular position for angular dependent periodic disturbance and the principal frequency for external disturbance. But, for the cases when the continuous angular position is not available, for example, because of the implemented sensor that develops only one or more than pulse per revolution. Then an observer can be designed to reconstruct the continuous angular position signal. Also, for the case of unknown external disturbance frequency, it can be measured or estimated by using frequency estimation algorithms (Bodson 2005).

7.2.2 Practical

Although the presented algorithms have been implemented in particular on an angular velocity servo control drive-load systems with periodic disturbances, the development and the design of the algorithms have been done in general format and they are easily extendable to almost any practical control problems with periodic disturbances. This makes the implementation spectrum of the algorithms in industrial applications very wide.

However, as an example is the extension to drive-load angular position servo control systems with self-excited oscillations (vibrations) due to flexible structure and angle dependent parameters. This extends the problem from single-input-single-output system into multi-input-multi-output system particularly when more than one drive is allowed to be used and more than one targeted control variable is needed to be controlled. This can be encountered, for example, in control systems of machine tools in general, and robot control systems in particular with flexible arms and with external or internal oscillation or vibration sources due to the robot drives (joint actuators/motors), or they come from the machine tool mounted in its tool center.

Furthermore, as another example is the implementation of the algorithm in an Active Noise and Vibration Control (ANVC) applications in general and the ANVC in automobile in particular. For example, the developed methods can be applied to reduce the engine oscillations and to reject their effect, the noise and the vibrations, on the passenger compartment, as well as on the exhaust system actively instead of traditional passive isolation methods. This will help in reducing the automobile weight and eventually its energy consumption which will favorably lead to increase the comfort and the dynamic performance of the automobile in general.

Appendix A Eccentric Mechanisms

In this appendix, simple eccentric mechanisms, Scotch yoke and crank mechanisms, are introduced. Where, in the first section, the kinematics and the dynamics are presented when the Scotch yoke mechanism has spring, damper and mass linear motion load elements. Moreover, a simple dynamic model for this system is constructed. Consequently, a frequency spectrum analysis for the angular velocity oscillation is presented for different linear motion load parameters. While, in the second section, the kinematics of crank mechanism is presented, as well as the frequency spectrum analysis of its transformed reciprocating linear motion, velocity and acceleration.

A.1 Scotch Yoke Mechanism

The Scotch yoke (slotted link) mechanism converts the rotational motion into a reciprocating linear motion or vice versa. Historically, the Scotch yoke mechanism was used in motors or machines that generate or translate high torques, for example, in steam and combustion engines, but with very low revolution rates (Wikipedia 2012).

A.1.1 Scotch Yoke Kinematics

The Scotch yoke mechanism, as shown in Figure A.1, translates the rotational motion into a pure sinusoidal (reciprocating) linear motion as

$$x = r \sin(\varphi), \quad (\text{A.1})$$

where φ is the rotational motion, angle of rotation in [rad], with a radius of r [m] and x is the translated linear motion in [m]. Now, the velocity of the linear motion can be derived from equation (A.1) as

$$\dot{x} = r \cos(\varphi) \dot{\varphi}. \quad (\text{A.2})$$

Also, the acceleration of the linear motion is derived from equation (A.2) as

$$\ddot{x} = r[\ddot{\varphi} \cos(\varphi) - \dot{\varphi}^2 \sin(\varphi)]. \quad (\text{A.3})$$

If the rotational motion rate of change is constant, then equation (A.3) becomes

$$\ddot{x} = -r \dot{\varphi}^2 \sin(\varphi) \quad (\text{A.4})$$

A.1.2 Scotch Yoke Dynamics

As can be seen from the Figure A.1 the mechanism translates the rotational motion, generated by a torque source (drive) T_D in [Nm], into linear motion. Therefore, the rotational torque source faces the reaction torques caused by the linear forces, which are generated by the linear motion spring, damper and mass load elements as presented in the following subsections.

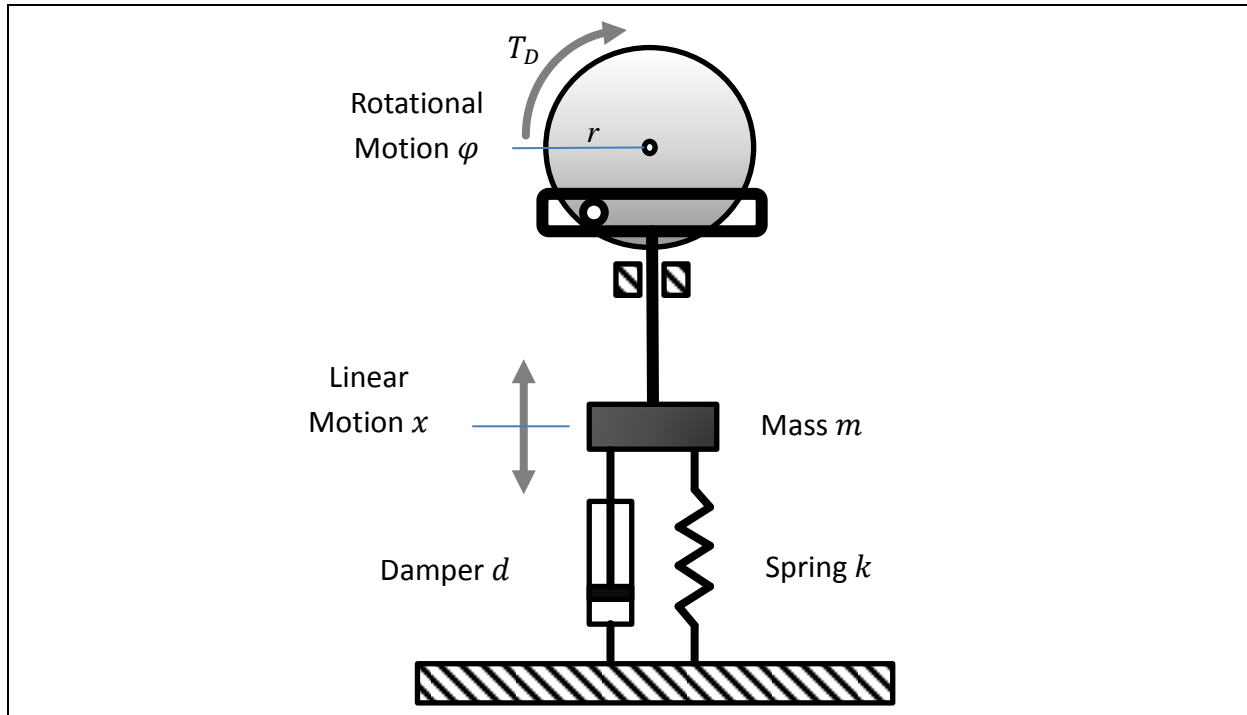


Figure A.1: Scotch yoke mechanism

A.1.2.1 Spring load element

The force generated by the linear motion spring element, using equation (A.1), is computed by

$$F_k = kx = kr \sin(\varphi), \quad (\text{A.5})$$

where k is the stiffness constant [N/m] of the linear motion spring load element. Moreover, the spring force can be converted into a reaction torque acting on the rotational system as

$$T_k = rF_k = rkx = r^2k \sin(\varphi). \quad (\text{A.6})$$

Therefore, the resulted rotational (angle dependent) spring (torque) function can be defined as

$$K(\varphi) = K_r \sin(\varphi), \quad (\text{A.7})$$

where

$$K_r = r^2k \text{ [Nm]}. \quad (\text{A.8})$$

This can also be done to the damper and the mass forces as in the following.

A.1.2.2 Damper load element

The damping force generated by the linear motion damper load element is computed by

$$F_d = d\dot{x} = drcos(\varphi)\dot{\varphi}, \quad (\text{A.9})$$

where d is the damping coefficient [Ns/m] of the linear motion damper. Therefore, its reaction torque at the rotational (motion) system is computed by

$$T_d = rF_d = r^2dcos(\varphi)\dot{\varphi}. \quad (\text{A.10})$$

Therefore, from equation (A.10), the converted rotational (angle dependent) damper function can be defined as

$$D(\varphi) = r^2 d \cos(\varphi) = D_r \cos(\varphi), \quad (\text{A.11})$$

where

$$D_r = r^2 d \text{ [Nms]} \quad (\text{A.12})$$

is the converted rotational damper constant. Moreover, the total rotational damping function of the rotational system is given by the sum of rotational damping coefficient D_0 and the retranslated (converted) rotational (angle dependent) damping function $D(\varphi)$ as following

$$D_{total}(\varphi) = D_0 + D(\varphi). \quad (\text{A.13})$$

A.1.2.3 Mass load element

Furthermore, the force generated due to acceleration of the mass is computed by

$$F_m = \frac{d}{dt}(m\dot{x}) = \frac{d}{dt}[mr \cos(\varphi)\dot{\varphi}], \quad (\text{A.14})$$

where m is the mass in [kg]. Note that the constant force due to gravity is neglected. Also, the mass force can be converted into load torque acting on the rotational system as

$$T_m = rF_m = \frac{d}{dt}[mr^2 \cos(\varphi)\dot{\varphi}]. \quad (\text{A.15})$$

Moreover, the torque in equation (A.15) can be redefined to fit the form

$$T_m = rF_m = \frac{d}{dt}[J(\varphi)\dot{\varphi}], \quad (\text{A.16})$$

where

$$J(\varphi) = mr^2 \cos(\varphi) = J_r \cos(\varphi) \quad (\text{A.17})$$

is the converted angle dependent rotational moment of inertia load element function and

$$J_r = mr^2 \text{ [Kgm}^2\text{]} \quad (\text{A.18})$$

is its constant.

Moreover, by reconsidering the derivation of the converted mass torque of equation (A.15), this yields

$$T_m = rF_m = \frac{d}{dt}[mr^2 \cos(\varphi)\dot{\varphi}] = mr^2[\ddot{\varphi} \cos(\varphi) - \dot{\varphi}^2 \sin(\varphi)]. \quad (\text{A.19})$$

Therefore, when the Scotch yoke is rotating with constant rotational speed then the first factor of equation (A.19) is zero, which yields

$$T_m = -mr^2 \dot{\varphi}^2 \sin(\varphi). \quad (\text{A.20})$$

Moreover, the total moment of inertia of the rotational system is given by the sum of rotational mass (moment of inertia J_0) and the retranslated (converted) rotational mass (angle dependent moment of inertia $J(\varphi)$) as following

$$J_{total}(\varphi) = J_0 + J(\varphi). \quad (\text{A.21})$$

A.1.3 Scotch Yoke Dynamic Model

Now, the driving torque is applied against the torque generated by total moment of inertia, the total rotational damping and the reflected angle dependent spring functions. Therefore, the system differential equation can be written as

$$\frac{d}{dt}([J_0 + J(\varphi)]\dot{\varphi}) + D_{total}(\varphi)\dot{\varphi} + K(\varphi) = T_D. \quad (A.22)$$

Or alternatively, equation (A.22) can be also rewritten as

$$J_0\ddot{\varphi} + \frac{d}{dt}[J(\varphi)\dot{\varphi}] + D_{total}(\varphi)\dot{\varphi} + K(\varphi) = T_D. \quad (A.23)$$

Now, substituting the rotational spring, damper and moment of inertia load elements defined in equations (A.7), (A.11), (A.13) and (A.17) into equation (A.23) yields

$$J_0\ddot{\varphi} + \frac{d}{dt}(J_r \cos(\varphi)\dot{\varphi}) + [D_0 + D_r \cos(\varphi)]\dot{\varphi} + K_r \sin(\varphi) = T_D. \quad (A.24)$$

After derivation and reorganization of equation (A.24), it becomes

$$[J_0 + J_r \cos(\varphi)]\ddot{\varphi} - J_r \sin(\varphi)\dot{\varphi}^2 + [D_0 + D_r \cos(\varphi)]\dot{\varphi} + K_r \sin(\varphi) = T_D. \quad (A.25)$$

Moreover, after substituting of the constants of equation (A.25) with their values defined in equations (A.8), (A.12) and (A.18), it becomes

$$[J_0 + mr^2 \cos(\varphi)]\ddot{\varphi} - mr^2 \sin(\varphi)\dot{\varphi}^2 + [D_0 + r^2 d \cos(\varphi)]\dot{\varphi} + r^2 k \sin(\varphi) = T_D. \quad (A.26)$$

Now, solving for the angular acceleration, the highest derivative in equation (A.26), yields

$$\ddot{\varphi} = \frac{[mr^2 \sin(\varphi)\dot{\varphi}^2 - [D_0 + r^2 d \cos(\varphi)]\dot{\varphi} - r^2 k \sin(\varphi) + T_D]}{[J_0 + mr^2 \cos(\varphi)]}. \quad (A.27)$$

Alternatively, equation (A.27) can be put back in a compact form as

$$\ddot{\varphi} = \frac{1}{J_{total}(\varphi)} [\alpha(\varphi)\dot{\varphi}^2 - D_{total}(\varphi)\dot{\varphi} - K(\varphi) + T_D], \quad (A.28)$$

where

$$\alpha(\varphi) = mr^2 \sin(\varphi). \quad (A.29)$$

Now, the block diagram of the Scotch yoke mechanism dynamic model with spring, damper and mass linear motion load elements, by using equation (A.28), is presented in Figure A.2.

A.1.4 Frequency Spectrum Analysis

In this subsection, the frequency spectrum analysis of the angular velocity is done for different parameter settings of the Scotch yoke mechanism dynamic model developed in previous subsection A.1.3 equation (A.28). The Scotch yoke mechanism is set up as angular velocity servo control system under a PI-controller with proportional and integral parameters ($K_p = 0.01$, $K_i = 0.1$). And the input set point is set to a constant at 100 [rad/s]. After the output reaches the steady state, the frequency spectrum analysis is done to the output angular velocity by computing the finite Discrete Fourier Transform (DFT) estimate by using the Fast Fourier Transform (FFT) algorithm and plotted for each parameter set in the respective figure.

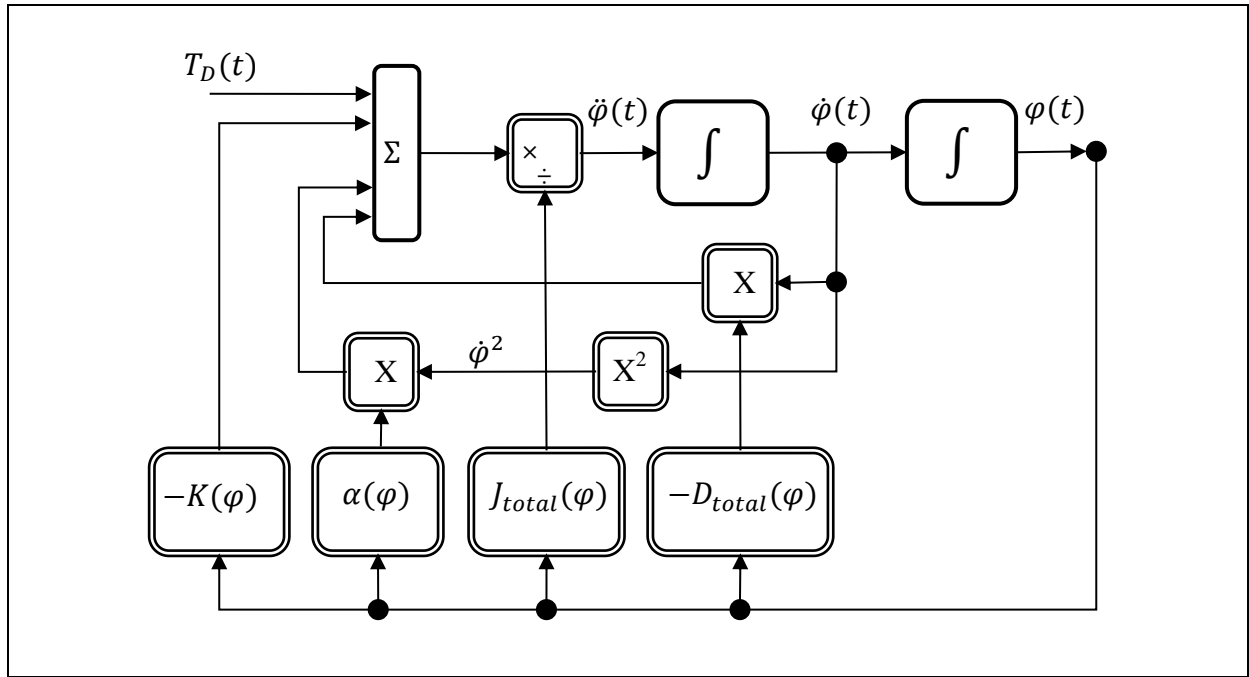


Figure A.2: Block diagram of the Scotch yoke dynamic model.

A.1.4.1 Linear motion spring and damper load elements

First, the frequency spectrum analysis is done, when Scotch yoke dynamic model has the following physical parameters:

$$\begin{array}{llll}
 J_0 = 1.0 & [\text{kgm}^2] & m = 0.0 & [\text{kg}] \\
 D_0 = 0.01 & [\text{Nms/rad}] & d = 1.0 & [\text{Ns/m}] \\
 r = 0.05 & [\text{m}] & k = 1.0 & [\text{N/m}]
 \end{array}$$

This means that, the mechanism drives a load with only damper and spring linear motion elements without a mass. The finite DFT frequency spectrum estimate of the steady state angular velocity output is computed and presented in Figure A.3. Where, it shows that, the output angular velocity oscillation has only one principal harmonic.

A.1.4.2 Linear motion mass load element

In this case, the Scotch yoke mechanism model has the following physical parameters:

$$\begin{array}{llll}
 J_0 = 1.0 & [\text{kgm}^2] & m = 1.0 & [\text{kg}] \\
 D_0 = 0.01 & [\text{Nms/rad}] & d = 0.0 & [\text{Ns/m}] \\
 r = 0.05 & [\text{m}] & k = 0.0 & [\text{N/m}]
 \end{array}$$

Therefore, in this case, the Scotch yoke mechanism drives a load with only a linear motion mass element, or in other words, without spring and damper linear motion load elements. The DFT frequency spectrum estimate of the steady state angular velocity output is computed and presented in Figure A.4. Where, it shows that, output angular velocity oscillation has two harmonics, the principal one and the relatively very small first super-harmonic.

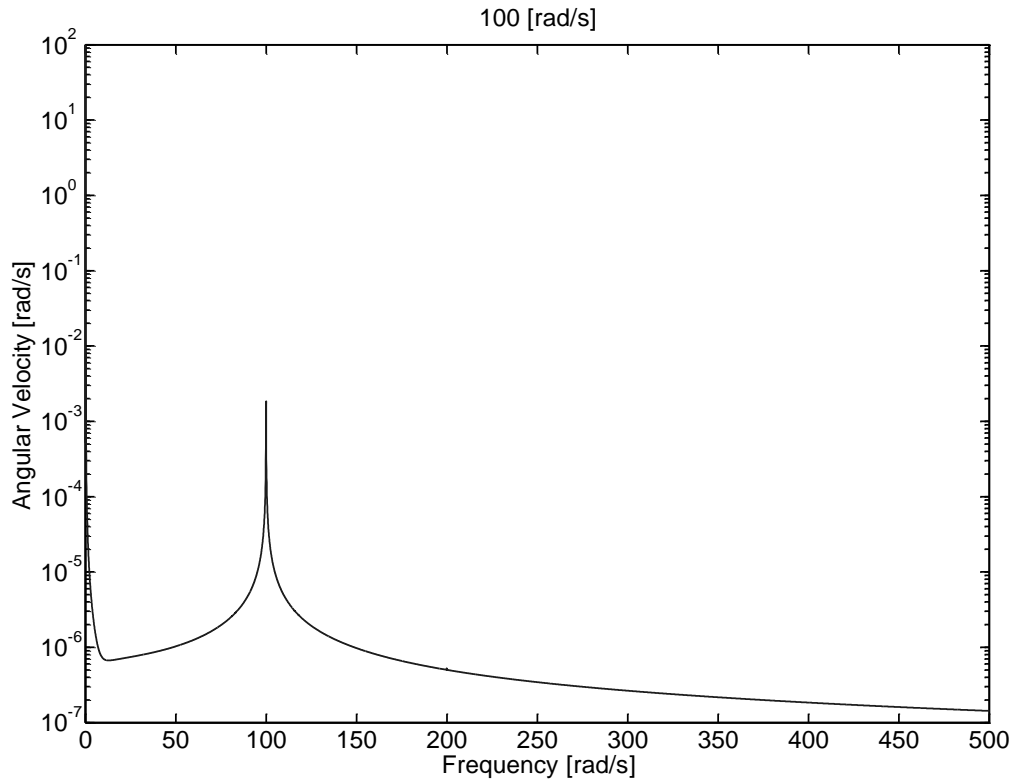


Figure A.3: Angular velocity finite DFT frequency spectrum estimate of Scotch yoke mechanism with linear motion damper and spring load elements.

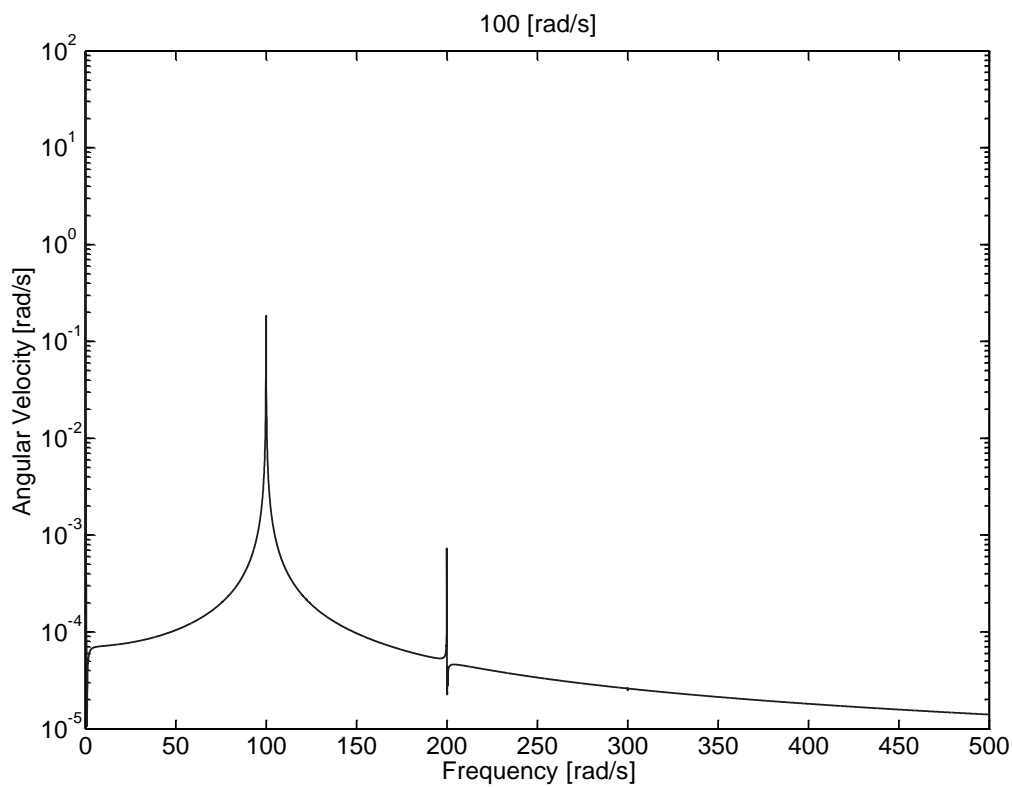


Figure A.4: Angular velocity finite DFT frequency spectrum estimate of Scotch yoke mechanism with linear motion mass load element.

A.2 Crank Mechanism

The crank mechanism is another example of eccentric mechanisms, that also transforms the rotational motion φ into a reciprocating linear motion x , which is connected to linear motion spring damper and mass load elements, as shown in the following Figure A.5.

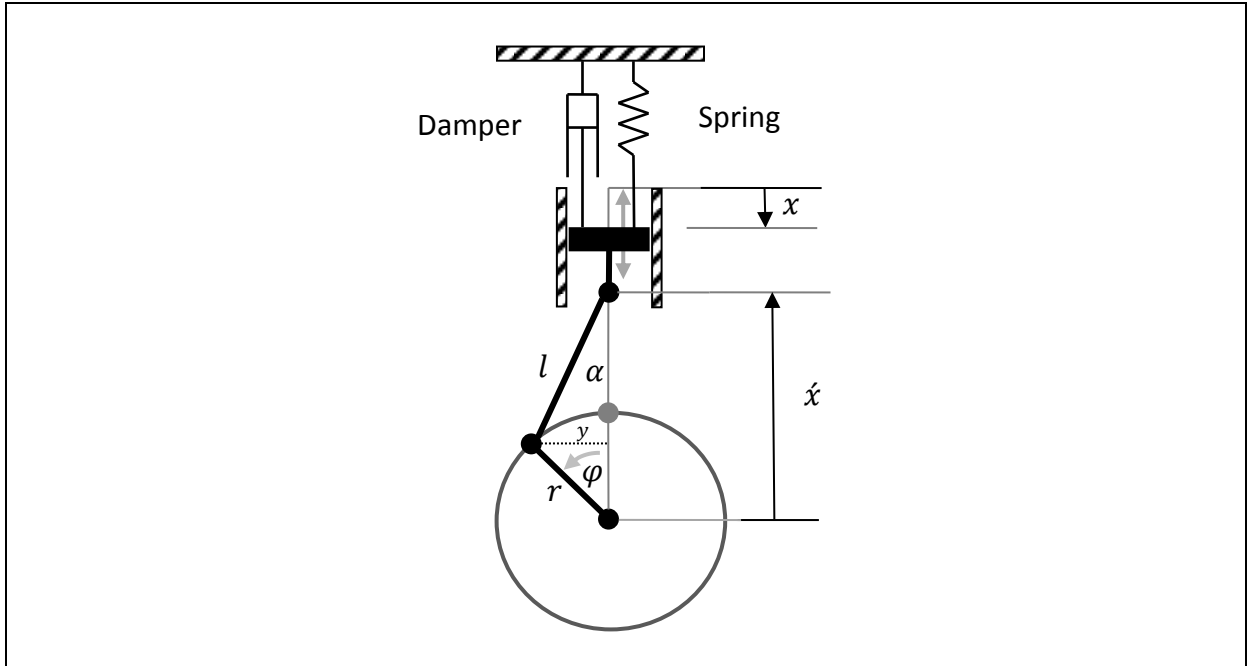


Figure A.5: Crank mechanism moving a mass, spring and damper load elements.

A.2.1 Crank Mechanism Kinematics

Now, the rotational motion φ of the crank mechanism, shown in Figure A.5, is transformed into the linear motion x , this can be done by starting by the relation between the rotational motion φ and the angle of the linking rod α , which is

$$r \sin(\varphi) = y = l \sin(\alpha), \quad (\text{A.30})$$

where r and l are the lengths [m] of the rotational rod and its link to the linear moving part.

Now, by squaring both sides of equation (A.30), it becomes

$$r^2 \sin^2(\varphi) = l^2 \sin^2(\alpha). \quad (\text{A.31})$$

And by using the trigonometric identity

$$\sin^2(\alpha) + \cos^2(\alpha) = 1, \quad (\text{A.32})$$

equation (A.31) becomes

$$r^2 \sin^2(\varphi) = l^2 [1 - \cos^2(\alpha)]. \quad (\text{A.33})$$

Solving equation (A.33) for $\cos(\alpha)$ results

$$\cos^2(\alpha) = 1 - \frac{r^2}{l^2} \sin^2(\varphi) \quad (\text{A.34})$$

or

$$\cos(\alpha) = \sqrt{1 - \frac{\sin^2(\varphi)}{n^2}} \quad (\text{A.35})$$

where

$$n = \frac{l}{r} \quad (\text{A.36})$$

is the ratio between the lengths of linking rod and the rotational rod.

Now, the translation between the rotational motion to the linear motion can be computed by

$$\dot{x} = r \cos(\varphi) + l \cos(\alpha). \quad (\text{A.37})$$

Moreover, substituting equation (A.35) in equation (A.37) yields

$$\begin{aligned} \dot{x} &= r \cos(\varphi) + l \sqrt{1 - \frac{\sin^2(\varphi)}{n^2}}; \\ &= r \left[\cos(\varphi) + n \sqrt{1 - \frac{\sin^2(\varphi)}{n^2}} \right], \end{aligned} \quad (\text{A.38})$$

or

$$\dot{x} = r \left[\cos(\varphi) + \sqrt{n^2 - \sin^2(\varphi)} \right]. \quad (\text{A.39})$$

Now, the linear motion is computed by

$$x = r + l - \dot{x}. \quad (\text{A.40})$$

Therefore, the computation of the linear motion from the rotational motion can be done by

$$x = r + l - r \left[\cos(\varphi) + \sqrt{n^2 - \sin^2(\varphi)} \right]. \quad (\text{A.41})$$

Moreover, the derivation of equation (A.41) yields the velocity of the linear motion, which is given by

$$\dot{x} = r \left[\sin(\varphi) \dot{\varphi} + \frac{2 \sin(\varphi) \cos(\varphi) \dot{\varphi}}{2 \sqrt{n^2 - \sin^2(\varphi)}} \right]. \quad (\text{A.42})$$

Using the double angle formula

$$2 \sin(\varphi) \cos(\varphi) = \sin(2\varphi) \quad (\text{A.43})$$

in equation (A.42), it becomes

$$\dot{x} = r \left[\sin(\varphi) \dot{\varphi} + \frac{\sin(2\varphi) \dot{\varphi}}{2 \sqrt{n^2 - \sin^2(\varphi)}} \right], \quad (\text{A.44})$$

or

$$\dot{x} = r\dot{\phi} \left[\sin(\phi) + \frac{\sin(2\phi)}{2\sqrt{n^2 - \sin^2(\phi)}} \right]. \quad (\text{A.45})$$

Furthermore, the derivation of equation (A.45) results the linear motion acceleration computation by

$$\begin{aligned} \ddot{x} = r\ddot{\phi} \left[\sin(\phi) + \frac{\sin(2\phi)}{2\sqrt{n^2 - \sin^2(\phi)}} \right] \\ + r\dot{\phi}^2 \left[\cos(\phi) + \frac{1}{2} \frac{[\sqrt{n^2 - \sin^2(\phi)}] \cos(2\phi) 2\dot{\phi} - \sin(2\phi) \frac{-2\sin(\phi)\cos(\phi)\dot{\phi}}{2\sqrt{n^2 - \sin^2(\phi)}}}{\sqrt{n^2 - \sin^2(\phi)}^2} \right]. \end{aligned} \quad (\text{A.46})$$

Also here, by using the double angle formula equation (A.43) in equation (A.46) results

$$\begin{aligned} \ddot{x} = r\ddot{\phi} \left[\sin(\phi) + \frac{\sin(2\phi)}{2\sqrt{n^2 - \sin^2(\phi)}} \right] \\ + r\dot{\phi}^2 \left[\cos(\phi) + \frac{1}{2} \frac{[\sqrt{n^2 - \sin^2(\phi)}] \cos(2\phi) 2 + \frac{\sin^2(2\phi)}{2\sqrt{n^2 - \sin^2(\phi)}}}{\sqrt{n^2 - \sin^2(\phi)}^2} \right], \end{aligned} \quad (\text{A.47})$$

or

$$\begin{aligned} \ddot{x} = r\ddot{\phi} \left[\sin(\phi) + \frac{\sin(2\phi)}{2\sqrt{n^2 - \sin^2(\phi)}} \right] \\ + r\dot{\phi}^2 \left[\cos(\phi) + \frac{\cos(2\phi)}{\sqrt{n^2 - \sin^2(\phi)}} + \frac{\sin^2(2\phi)}{4\sqrt{n^2 - \sin^2(\phi)}^3} \right]. \end{aligned} \quad (\text{A.48})$$

In the next subsection, the frequency spectrum analysis is done to only the angle dependent parts of the linear motion and to its derived linear velocity and acceleration, when the crank mechanism rotational angular velocity is constant.

A.2.2 Frequency Spectrum Analysis

In this subsection, the frequency spectrum of the angle dependent linear motion part in equation (A.39), which is expressed by

$$\left[\cos(\phi) + \sqrt{n^2 - \sin^2(\phi)} \right], \quad (\text{A.49})$$

is computed at constant angular velocity of 50 [rad/s] (with $n = 1$) by using the FFT algorithm and plotted in Figure A.6. This angle dependent linear motion part generates or can be converted as angle dependent spring torque that affects back the rotational system. Figure A.6 shows that this part of angle dependent linear motion expressed by (A.49), where the first term of the angle dependent linear motion part (A.49) generates the first harmonic with fre-

quency at the rate of angular rotation (50 [rad/s]), while the second term of the expression generates the second harmonic with twice the frequency of the first term component as well as super-harmonics. Moreover, the angle dependent linear motion part has a constant level component.

Also with the same conditions in this subsection, the frequency spectrum of the angle dependent linear velocity part, equation (A.45), which is expressed by

$$\left[\sin(\varphi) + \frac{\sin(2\varphi)}{2\sqrt{n^2 - \sin^2(\varphi)}} \right], \quad (\text{A.50})$$

is done and presented in Figure A.7, which reveals that, this component has also, apart from the principal one, some super-harmonics.

Furthermore, the frequency spectrum of the angle dependent linear motion acceleration part, the second term of equation (A.48), which is expressed by

$$\left[\cos(\varphi) + \frac{\cos(2\varphi)}{\sqrt{n^2 - \sin^2(\varphi)}} + \frac{\sin^2(2\varphi)}{4\sqrt{n^2 - \sin^2(\varphi)}^3} \right], \quad (\text{A.51})$$

is also computed and plotted, with the same conditions, in Figure A.8, which also reveals that, this component has also, apart from the principal one, some super-harmonics as well as a constant level.

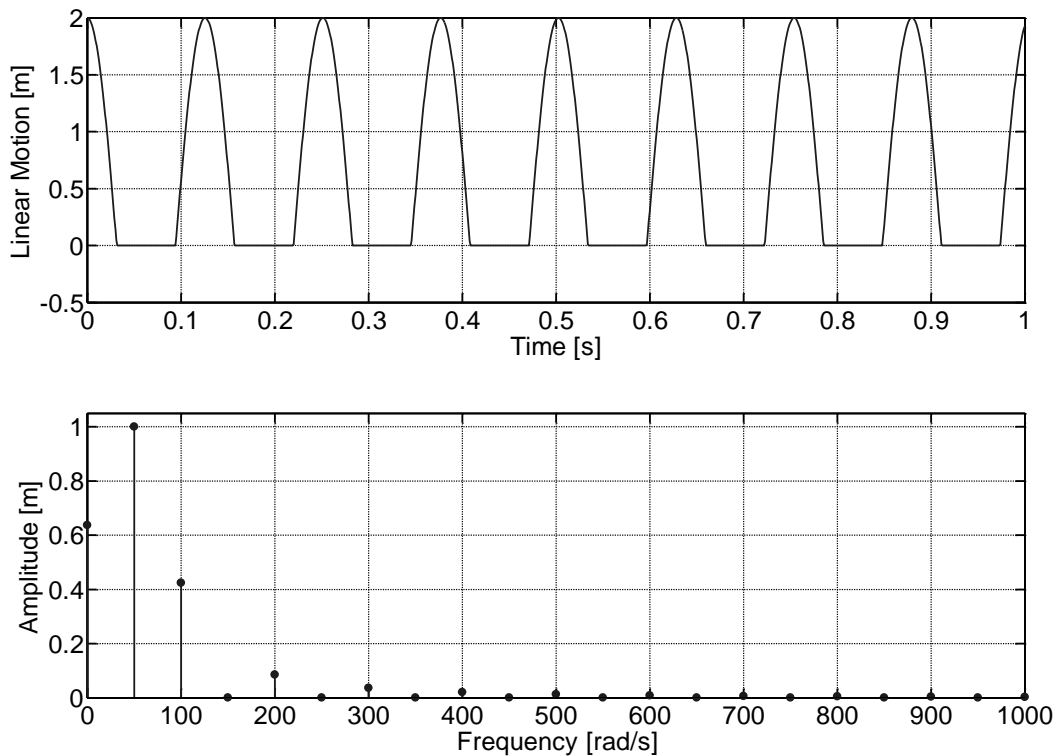


Figure A.6: Time plot (top) and frequency spectrum plot (bottom) of the angle dependent linear motion part at 50 [rad/s] rotational velocity.

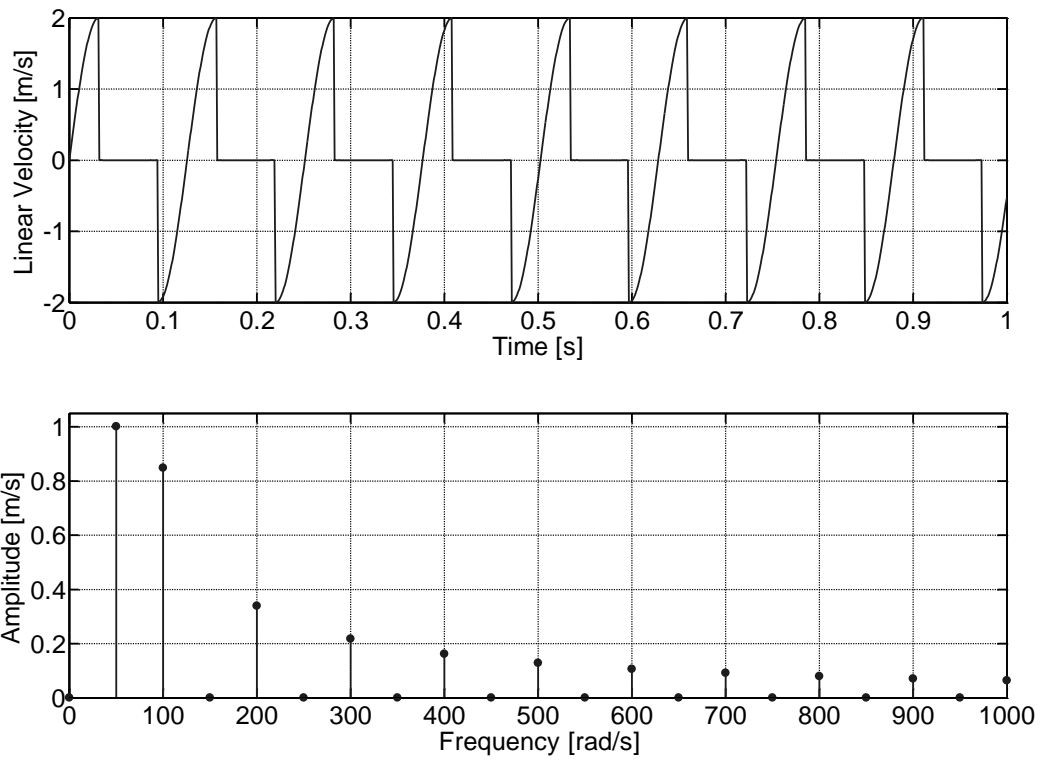


Figure A.7: Time plot (top) and frequency spectrum plot (bottom) of angle dependent linear motion velocity part at 50 [rad/s] rotational velocity.

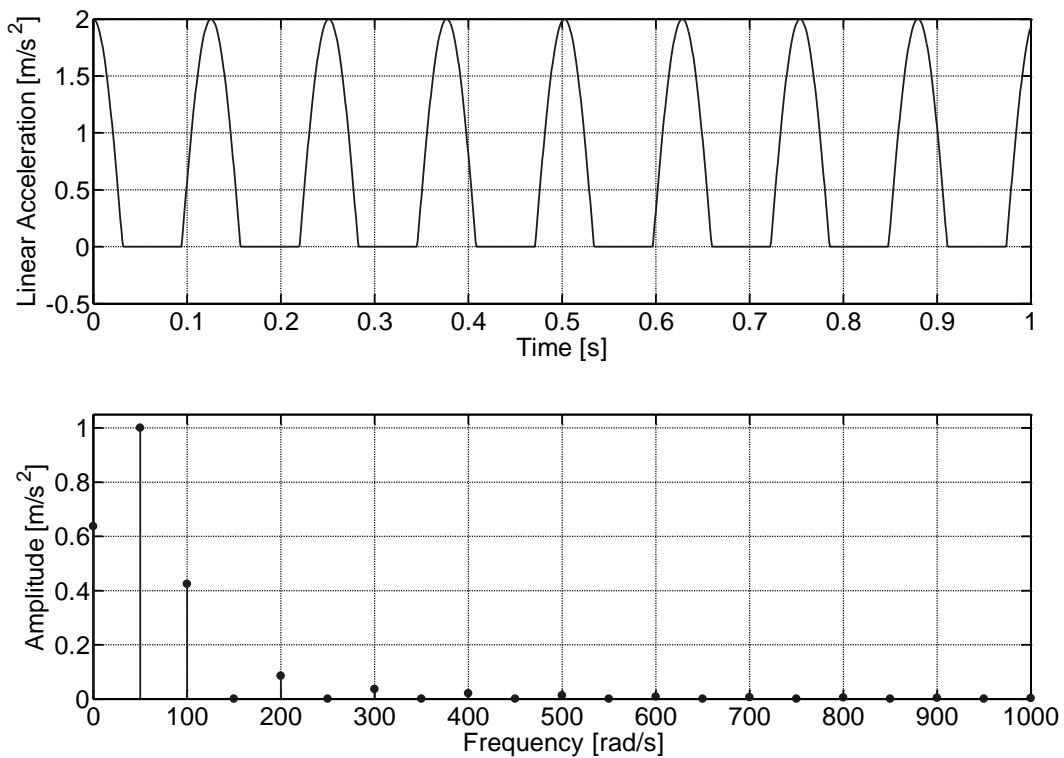


Figure A.8: Time plot (top) and frequency spectrum plot (bottom) of angle dependent linear motion acceleration part at 50 [rad/s] rotational velocity.

Appendix B System Identification

In this appendix, only a concise literature review is presented to system identification in general. Then, a short definition to the system identification main task is given. Furthermore, some system identification issues and aspects, which are related to this work, are very briefly discussed.

On the other hand, parametric identification algorithms, for continuous and discrete models with linear or nonlinear in parameter model structures are presented, and reintroduced in graphical block diagram format, so that they can be easily programmed in Simulink-Matlab from MathWorks Company, which is a graphical-oriented programming environment. This will make them easy to compile and to implement online and in real-time control by using, for example, the dSPACE Company real-time controller utilities.

B.1 Introduction

The modern system identification history can be traced back to time of Gauss (1809), where he had used the method of least squares to determine the orbits of celestial bodies from their observations (Gevers 2006). But the new era of the system identification, which was developed mainly for control applications, stemmed from the work of Ho and Kalman (1965) and Aström and Bohlin (1965), which have marked the development of system identification until these days. Ho and Kalman (1965) have dealt with the minimal realization problem of the state space, which has led to the development of subspace identification techniques. While, Aström and Bohlin (1965) have introduced the maximum likelihood framework, which has led in turn to the parametric prediction error methods. More details of this history are in the reference (Gevers 2006). The work of Aström and Eykhoff (1971) stands since 1971 as a survey for system identification in general. For system identification references, the system identification tutorials (Fasol and Jörgl 1980; Rake 1980; Godfrey 1980; Strejc 1980; Aström 1980; Isermann 1980) given in the special issue of *Automatica* volume 16 number 5, are suggested, as well as the text books (Norton 1986; Söderström and Stoica 1989; Ljung 1999; Isermann and Münchhof 2011) for system identification in general, and (Haber and Keviczky 1999; Nelles 2001) for nonlinear system identification in particular. Furthermore, more specific references will be given in the context accordingly.

System identification is the art of finding a mathematical model that gives the best description of a system using its input-output history. But there are actually two assumptions in the definition. The first one is that the model structure and its parameterization are known from the physical principles, for example, a first order time-invariant linear system. Therefore, the job here is to find its differential equation, transfer function or impulse response parameters, and that is of course by using the system input-output data collected in an experiment. This aspect can be called light gray model identification. The second one, the system behavior (input-output data) is to be fit in some model, with the fact that the model structure is not known. By using the input-output data, the model structure and its parameters are to be calculated or estimated. This type of identification is called approximation, which also called dark gray model identification.

In the following, some modeling and identification aspects are discussed, distinguished and explained why they are chosen to work within this context.

- ***Parametric and Nonparametric System Identification***

Theoretically, nonparametric models have an infinite number of parameters (data), for example, the time (impulse, step or ramp) response or frequency (correlation or spectral function) response, and practically, nonparametric models can be approximated by a relatively large finite number of parameters (data), for example, time or frequency response table records with a large number of data. The advantage of the nonparametric models is that they can be directly measured and recorded without extensive pre-knowledge about the system; therefore, they are always used as a quick way to start acquainting the knowledge about the process under the analysis. On the other hand, their disadvantage though, they need a relatively large memory to be stored in a digital computer.

For the parametric models, it is almost the contrary to the nonparametric ones, where a pre-knowledge is very essential to choose a candidate function that can perfectly fit the response (behavior) of the system with as small number of parameters as possible, for example, the candidate function of the system step response differs according to the number and the position of the poles of the linear time-invariant system. So, the crucial issue for parametric models is the candidate function or the model structure that must be previously selected before the phase of parameter computation. The positive issue though, the number of parameters is very small in comparison with an approximated nonparametric one. In this context, mainly the parametric identification methods are considered for later use in control law design phase.

- ***Linear or Nonlinear Models***

Linear models are very well established and have a general structure with its algorithms. On the contrary, the nonlinear models are very difficult to handle because they are very challenging to put in a general form, and even though, it will be done with very huge and tremendous efforts. Although dealing with the linear models is easier than the nonlinear one for some cases, there are some other cases, where a more global behavior is interested, that make the use of the nonlinear models inevitable to increase the performance of a system.

- ***Deterministic or Stochastic Environment***

Deterministic is always when the input-output data is well measured (with high fidelity), this means there is almost no noise or error by measurements or the signal to noise ratio is very high. Vice versa, in the stochastic case, the data is corrupted with errors (noise) or the plant is disturbed by stochastic variable or parameter. Therefore, by using methods designed accordingly, a better performance can be achieved than using only deterministic cases. Some of these methods are instrumental variables, extended and generalized least squares, maximum likelihood, prediction error method and dual (extended) Kalman filter. In this work, the process assumed to have no stochastic variables or parameters and the input-output measurements are available with very good signal to noise ratio.

- ***Continuous or Discrete Models***

Now days, the most recent control applications are applied using digital computers. This means that the applications are discrete in nature, and therefore, discrete models comply with them. Nevertheless, continuous models have more appeal to the physical nature of the process than the discrete ones. The discrete models have rather more mathematical appeal than the physical ones. This means, for a continuous model, its parameters are directly linked to its physical parameters, while on the other hand, the discrete model parameters are rather more directly linked to the sampling rate. Furthermore, if nonlinear models are interested, then only continuous modeling makes sense to observe and represent the process, even if it is applied by a digital computer. The quasi-continuous mode should be considered, by sampling the system variables with a very high sampling rate, or simulating the system with a very small numerical integration step size with respect to the targeted system bandwidth.

- ***Linear or Nonlinear in Parameters***

System identification problem, in general, is an optimization problem, where a performance measure or index has to be defined which usually incorporates, for example, a sum of squared errors between the outputs of the process to be identified and the outputs of the mathematical model that will represent or approximate the real physical process. The sum of the squared errors index is usually chosen because of its analytical simplicity. For the case, when the error used in the performance index is linear in parameters, this will result in a simple least squares problem with a direct solution. On the other hand, the case of nonlinear in parameters problem is more complex than the linear in parameter one, where it leads to the nonlinear least squares problem, which has no direct (trivial) mathematical solution. Therefore, only numerical iterative (successive) approximation solutions are available. Some of these methods are gradient based algorithms (line search, finite difference techniques, steepest descent, Newton's method, quasi-Newton methods, Gauss-Newton method, conjugate gradient methods, etc.). And some other methods depend on direct search algorithms (simplex search method, Hooke-Jeeves method). For more detail, for example, see the references (Ljung 1999; Nelles 2001).

- ***Offline or Online***

Most of the system identification algorithms were derived in a form suitable for batch processing in which the amount of data storage and computations increase with the number of collected input and output data pairs. Therefore, these algorithms are usually used in offline system identification. On the other hand, these batch algorithms are obviously undesirable for online identification as required, for example, in adaptive control schemes. Therefore, recursive identification algorithms have been developed directly from the original batch processing identification algorithms by processing the every newly collected input-output data pair immediately as a recursion to the old processed data, so that their computational requirements are kept at minimum and do not increase as time progresses, which makes them eligible to online implementations (Ljung and Söderström 1983).

- ***Identification for control***

Sometimes, although the process can have a complex behavior (high order, nonlinear, time-variant) that needs a complex model to describe it. Nevertheless, in many control applications, it needs only a part of its behavior to be known (identified) in order to achieve the

control design goals. For example, considering such a complex system that only needs to be controlled in the steady state region, therefore, for this design demand, only its gain factor is needed to be identified around the operating point. Moreover, trying to make the design more dynamic, this could lead the system to operate in an unknown (unidentified) region that could make the design, for example, unstable. This leads to the case where a more complex dynamic model should be used. Another example is when a larger operating region is considered, that could lead to more nonlinear characteristics to appear at the system behavior, which they need of course to be considered in order to get a better performance.

In the next section B.2, the least squares algorithm is used to minimize the sum of squared prediction and output errors for the case when the model parameters are or assumed linear in parameter with respect to their modeling error, particularly for ARX (AutoRegressive model with eXogenous input) discrete models. Also, in section B.3, the general case is considered, when the model parameters are nonlinear in parameters, where the Gauss-Newton method is introduced and formulated as recursive identification algorithm. At the end, some practical aspects of system identification for control are discussed in section B.4.

B.2 Linear Least Squares Parametric Identification Methods

In this section, the least squares optimization algorithm is used to identify processes that are modeled with linear in parameters discrete models, see e.g. (Söderström and Stoica 1989). Therefore, the algorithm is used to minimize the sum of squared prediction errors, resulting in prediction error method. Moreover, an explanation is given why it cannot be directly applied to minimize the sum of squared output errors. Also, the recursive least squares algorithm is introduced.

Now, a general parametric dynamic process is defined to have linear time-invariant dynamics that are stimulated by the control input ($u(t)$) and disturbed by an unmeasurable as well as an uncontrollable input ($d_o(t)$) called the disturbance. Moreover, the output of the process is measured with a measurement error ($e_m(t)$), which is also a kind of disturbance. But in this context, the disturbance represents here a real physical action on the process while the measurement error is a falsifying value added to the real output value induced in the measurement phase. So, the process is represented by following equation (B.1), and presented in Figure B.1,

$$y_m(t) = y_u(t) + d_o(t) + e_m(t). \quad (\text{B.1})$$

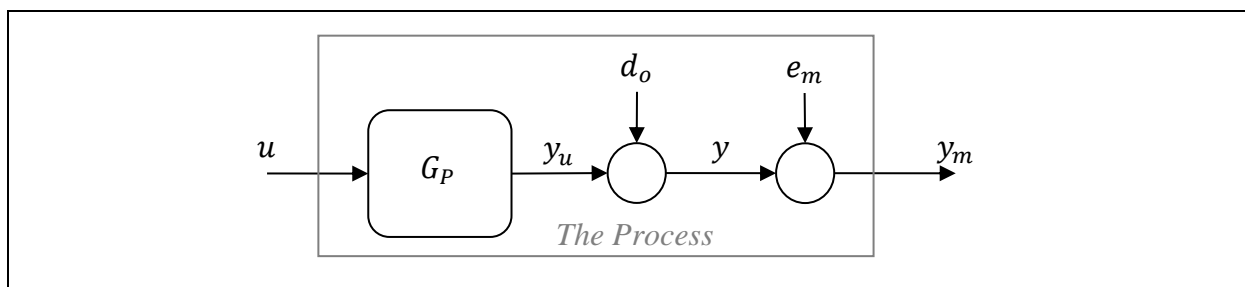


Figure B.1: Disturbed process with output measurement error.

In the following, the process dynamics are defined as continuous and discrete strictly causal, which means that the current output depends only on the past input-output data, as given in the following equations in continuous s - and discrete z -domain transfer functions, as well as in discrete-time domain series and vector format. The continuous transfer function

$$\frac{y_u(s)}{u(s)} = G_P(s) = \frac{B(s)}{A(s)} = \frac{\beta_{n-1}s^{n-1} + \dots + \beta_0}{s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_0}, \quad (\text{B.2})$$

and the discrete transfer function

$$\frac{y_u(z)}{u(z)} = G_P(z) = \frac{B(z)}{A(z)} = \frac{b_1z^{n-1} + \dots + b_n}{z^n + a_1z^{n-1} + \dots + a_n}, \quad (\text{B.3})$$

in past inputs and outputs series

$$y_u(t) = -\sum_{i=1}^n a_i y_u(t-i) + \sum_{i=1}^n b_i u(t-i), \quad (\text{B.4})$$

where $(t = 0, 1, 2, \dots)$ is sampling time sequence index, also in compact vector format

$$y_u(t) = \boldsymbol{\phi}_u^T(t) \boldsymbol{\theta}, \quad (\text{B.5})$$

where the past input-output data vector is constructed as following

$$\boldsymbol{\phi}_u^T(t) = [-y_u(t-1) \quad \dots \quad -y_u(t-n) \quad u(t-1) \quad \dots \quad u(t-n)], \quad (\text{B.6})$$

and the parameter vector

$$\boldsymbol{\theta} = [a_1 \quad \dots \quad a_n \quad b_1 \quad \dots \quad b_n]^T. \quad (\text{B.7})$$

B.2.1 Simple Least Squares of Prediction Error

First, the process, as defined in equation (B.1), is assumed to be without disturbance, so the process output becomes

$$y_m(t) = y_u(t) + e_m(t), \quad (\text{B.8})$$

also, the sampled measured output of the process is the sum of the process input-output dynamics and the measurement error, as following

$$y_m(t) = \boldsymbol{\phi}_u^T(t) \boldsymbol{\theta} + e_m(t). \quad (\text{B.9})$$

Since the input-output data vector in equation (B.9) contains the past outputs of the process dynamics which is not available as measurements, therefore, equation (B.9) is reformed to have the measured outputs instead as

$$y_m(t) = \boldsymbol{\phi}_m^T(t) \boldsymbol{\theta} + e_{EE}(t), \quad (\text{B.10})$$

where the data vector and the equation error are defined by

$$\boldsymbol{\phi}_m^T(t) = [-y_m(t-1) \quad \dots \quad -y_m(t-n) \quad u(t-1) \quad \dots \quad u(t-n)], \quad (\text{B.11})$$

$$e_{EE}(t) = e_m(t) + a_1 e_m(t-1) + \dots + a_n e_m(t-n). \quad (\text{B.12})$$

Now, an identification experiment is done by injecting a proper (persistently exciting) input signal to the process and recording a number (N) of input-output pairs. The prediction of the process measured output is (estimated) computed by using the past input and the past measured output data as following

$$\hat{y}_m(t) = \boldsymbol{\phi}_m^T(t)\hat{\boldsymbol{\theta}}, \quad (\text{B.13})$$

where the estimated parameter vector is defined by

$$\hat{\boldsymbol{\theta}} = [\hat{a}_1 \quad \cdots \quad \hat{a}_n \quad \hat{b}_1 \quad \cdots \quad \hat{b}_n]^T, \quad (\text{B.14})$$

Now, the output prediction error is defined as a modeling error, which is the difference between the measured output and the computed output, see Figure B.2, as following

$$e_{PE}(t) = y_m(t) - \hat{y}_m(t), \quad (\text{B.15})$$

$$e_{PE}(t, \hat{\boldsymbol{\theta}}) = y_m(t) - \boldsymbol{\phi}_m^T(t)\hat{\boldsymbol{\theta}}. \quad (\text{B.16})$$

Now, the method of Simple Least Squares (SLS) is used as an optimization method to estimate the process parameters by minimizing a performance index, which is defined by the sum of the squared prediction errors as given by

$$J_N = \frac{1}{2} \sum_{t=n}^N [e_{PE}(t, \hat{\boldsymbol{\theta}})]^2. \quad (\text{B.17})$$

For N input-output data pairs, the data can be batched in vectors and matrices and that is by starting up from the system order n . So, the data matrix and the measured output array are constructed as following

$$\boldsymbol{\Phi}_m = \begin{bmatrix} \boldsymbol{\phi}_m^T(n) \\ \vdots \\ \boldsymbol{\phi}_m^T(N) \end{bmatrix} = \begin{bmatrix} -y_m(n-1) & \cdots & -y_m(0) & u(n-1) & \cdots & u(0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_m(N-1) & \cdots & -y_m(N-n) & u(N-1) & \cdots & u(N-n) \end{bmatrix}, \quad (\text{B.18})$$

$$\mathbf{Y}_m = [y_m(n) \quad \cdots \quad y_m(N)]^T, \quad (\text{B.19})$$

and the modeling (prediction) error vector as

$$\mathbf{E}_{PE} = \mathbf{Y}_m - \boldsymbol{\Phi}_m \hat{\boldsymbol{\theta}}, \quad (\text{B.20})$$

as well as the sum of squared prediction error performance index

$$J_N = \frac{1}{2} \sum_{t=n}^N [e_{PE}(t)]^2 = \frac{1}{2} \mathbf{E}_{PE}^T \mathbf{E}_{PE}. \quad (\text{B.21})$$

Now, minimizing (B.21) will give the optimal least squares solution of the prediction error, see e.g. (Norton 1986), as

$$\hat{\boldsymbol{\theta}} = [\boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_m]^{-1} \boldsymbol{\Phi}_m^T \mathbf{Y}_m. \quad (\text{B.22})$$

To check the parameter consistency, the expected value of the estimated parameters is computed as following

$$\mathcal{E}\{\hat{\boldsymbol{\theta}}\} = \mathcal{E}\{[\boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_m]^{-1} \boldsymbol{\Phi}_m^T \mathbf{Y}_m\}, \quad (\text{B.23})$$

since the measured output vector, generated from equation (B.10), is given by

$$\mathbf{Y}_m = \boldsymbol{\Phi}_m \boldsymbol{\theta} + \mathbf{E}_{EE}. \quad (\text{B.24})$$

Now, the substitution of the measured output vector equation (B.24) into (B.23) yields

$$\begin{aligned} \mathcal{E}\{\hat{\boldsymbol{\theta}}\} &= \mathcal{E}\{[\boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_m]^{-1} \boldsymbol{\Phi}_m^T [\boldsymbol{\Phi}_m \boldsymbol{\theta} + \mathbf{E}_{EE}]\}; \\ &= \boldsymbol{\theta} + \mathcal{E}\{[\boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_m]^{-1} \boldsymbol{\Phi}_m^T \mathbf{E}_{EE}\}. \end{aligned} \quad (\text{B.25})$$

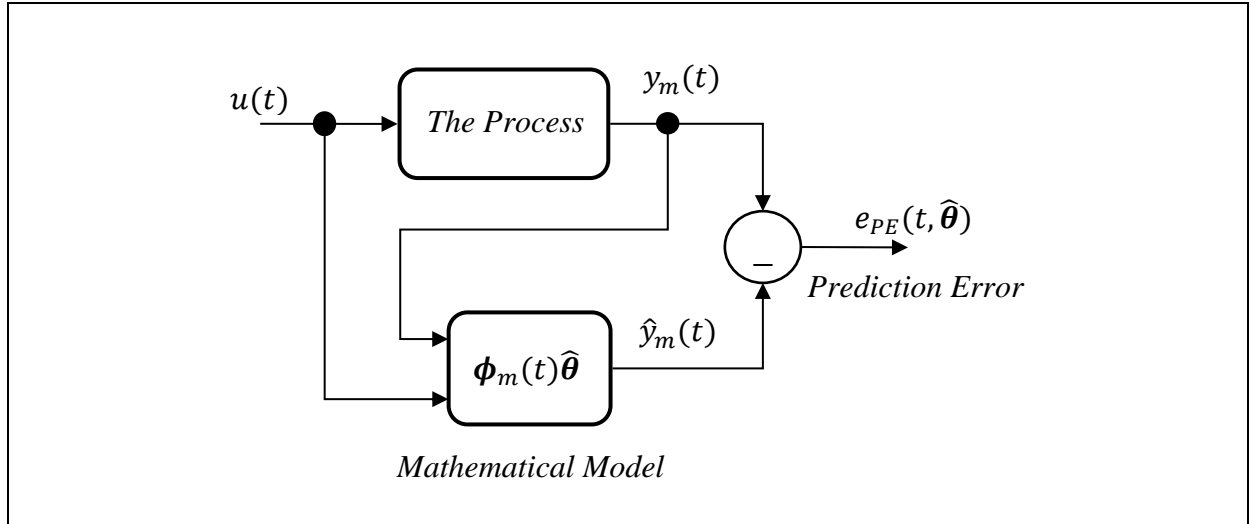


Figure B.2: Prediction error system identification.

Now, if the measurement error is zero, then the equation error is also zero, the second term of equation is equal to zero, and therefore, the parameter vector estimate is equal to the process (true) parameter vector

$$\mathcal{E}\{\hat{\boldsymbol{\theta}}\} = \boldsymbol{\theta}. \quad (\text{B.26})$$

Moreover, for the case, when the equation error is a white noise with zero mean and uncorrelated with the input, then the second term of equation (B.25) goes to zero as the number of sampled input-output data goes to infinity, which leads to a consistent parameter vector estimate (Isermann and Münchhof 2011). However, the last condition is only realizable when the measurement error is a filtered white noise through the true process dynamics

$$e_m(t) = -a_1 e_m(t-1) - \dots - a_n e_m(t-n) + \zeta(t), \quad (\text{B.27})$$

where $\zeta(t)$ is a white noise with zero mean signal.

This condition is seldom to happen in real processes. In other words, the parameter estimation will bias even when the measurement error is a white noise with zero mean. Therefore, there are modified least squares methods to deal with this problem, for example, generalized least squares and extended least squares methods (Isermann and Münchhof 2011). Nevertheless, the estimated model will give a good prediction to the measured and disturbed output of the process.

B.2.2 Least Squares of Output Error

In order to identify the true parameters of the process input-output dynamics, given in equation (B.1), in spite of the presence of the measurement error or the disturbance affecting the measured output. The data vector should be changed so that it has no more components from these measurements. Therefore, the algorithm will correlate the relation between the input and the measured disturbed output and exclusively builds up the relation just between the input and its contribution to the output. This can be done as following: The output estimation of the process dynamics stimulated from the input is computed by

$$\hat{y}_u(t) = \hat{\boldsymbol{\phi}}_u^T(t)\hat{\boldsymbol{\theta}}, \quad (\text{B.28})$$

where the parameter vector is defined as in equation (B.14), the data vector and matrix are constructed using the computed outputs instead of the measured outputs respectively as

$$\hat{\boldsymbol{\phi}}_u^T(t) = [-\hat{y}_u(t-1) \quad \cdots \quad -\hat{y}_u(t-n) \quad u(t-1) \quad \cdots \quad u(t-n)], \quad (\text{B.29})$$

$$\hat{\boldsymbol{\Phi}}_u = \begin{bmatrix} \hat{\boldsymbol{\phi}}_u^T(n) \\ \vdots \\ \hat{\boldsymbol{\phi}}_u^T(N) \end{bmatrix} = \begin{bmatrix} -\hat{y}_u(n-1) & \cdots & -\hat{y}_u(0) & u(n-1) & \cdots & u(0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\hat{y}_u(N-1) & \cdots & -\hat{y}_u(N-n) & u(N-1) & \cdots & u(N-n) \end{bmatrix}. \quad (\text{B.30})$$

And so here, the output (estimation) error is defined by the difference between the measured output and the output of the estimated dynamics, see Figure B.3, as

$$\begin{aligned} e_{OE}(t) &= y_m(t) - \hat{\boldsymbol{\phi}}_u^T(t)\hat{\boldsymbol{\theta}}; \\ &= y_m(t) - \hat{y}_u(t), \end{aligned} \quad (\text{B.31})$$

or in array format

$$\mathbf{E}_{OE} = \mathbf{Y}_m - \hat{\boldsymbol{\Phi}}_u\hat{\boldsymbol{\theta}}. \quad (\text{B.32})$$

This modeling error is defined as the output error, since it uses the past input and past estimated output data to compute the current output estimate. Now, the least squares criterion index is applied on the output error as

$$J_N = \frac{1}{2} \sum_{t=n}^N [e_{OE}(t)]^2 = \frac{1}{2} \mathbf{E}_{OE}^T \mathbf{E}_{OE}. \quad (\text{B.33})$$

Unfortunately, the output error is nonlinearly related to the parameters, since the data vector has estimated output values instead of the measured output values, where the estimated output values depend on the estimated parameter vector (Nelles 2001, subsection 16.5.4). Therefore, the solution for an optimal parameter vector is a nontrivial one and usually a numerical and iterative approximation, e.g. the solutions of nonlinear in parameter problem presented in subsection B.3.1, and the pseudo linear regression for model reference techniques presented in (Ljung and Söderström 1983, subsection 2.5.2; Landau 1976).

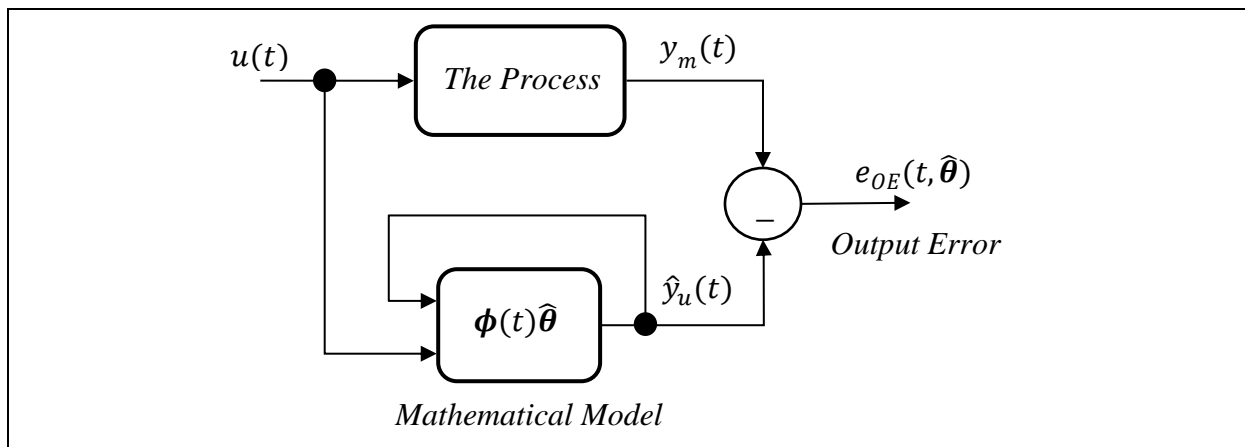


Figure B.3: Output error system identification.

B.2.3 The Recursive Least Squares Algorithm with Forgetting Factor

The simple least squares as introduced in equation (B.22) processes a batch of input-output data at time. This is not perfect for online identification. Hence, a Recursive Least Squares (RLS) algorithm is introduced from the simple one, which only requires the new data sample to compute the new parameter estimate recursively (see, e.g., Söderström and Stoica 1989). Moreover, in the case, where the process is (slowly) time-varying, and in order to get the actual parameter estimate, the least squares algorithm has to forget the old input-output data information and to pay more attention to the recent input-output data. This is done by introducing a weighting (*forgetting*) factor to the sum of squared prediction error index as

$$J = \frac{1}{2} \sum_{t=n}^N \lambda^{N-t} [e_{PE}(t)]^2, \quad (\text{B.34})$$

where λ is the forgetting factor. Now define the covariance matrix as

$$\mathbf{P}(t-1) = [\boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_m]^{-1}. \quad (\text{B.35})$$

The minimization of the performance index of equation (B.34) leads to the recursive algorithm with the initial conditions $\hat{\boldsymbol{\theta}}(0)$ and $\mathbf{P}(0)$. After defining all initial conditions and parameters the RLS algorithm, equations (B.36), (B.37) and (B.38), is run repetitively at every sample hit.

$$\mathbf{K}(t) = \frac{\mathbf{P}(t-1)\boldsymbol{\phi}_m(t)}{\lambda + \boldsymbol{\phi}_m^T(t)\mathbf{P}(t-1)\boldsymbol{\phi}_m(t)}, \quad (\text{B.36})$$

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \mathbf{K}(t)[y_m(t) - \boldsymbol{\phi}_m^T(t)\hat{\boldsymbol{\theta}}(t-1)], \quad (\text{B.37})$$

$$\mathbf{P}(t) = \frac{1}{\lambda} \left[\mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\phi}_m(t)\boldsymbol{\phi}_m^T(t)\mathbf{P}(t-1)}{\lambda + \boldsymbol{\phi}_m^T(t)\mathbf{P}(t-1)\boldsymbol{\phi}_m(t)} \right]. \quad (\text{B.38})$$

B.2.4 Starting the RLS for Online Identification

The algorithm is normally initialized by setting the covariance matrix to a diagonal matrix whose elements reflect how much confidence is in the initial vector of parameter estimates. In other words, the initial covariance matrix can be set as

$$\mathbf{P}(0) = \gamma \mathbf{I}, \quad (\text{B.39})$$

where \mathbf{I} is the identity matrix, γ is a scalar and chosen such that γ is high when the initial estimates are poor (i.e. little confidence). On the other hand, α is chosen low when there are reasonable initial estimates (i.e. reasonable confidence).

The initial vector of parameter estimates is usually set to a “good guess” based on prior knowledge about the system to be identified. In this way, γ is set to a small to medium value depending on the relative quality of the “good guess”. Alternatively, it can be waited until sufficient data is available to perform one matrix inversion i.e. that of $[\hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\Phi}}]$ to get $\hat{\boldsymbol{\theta}}(0)$. The RLS would, of course, then require a small γ -value.

Therefore, if a priori-knowledge about the process is available then it should be used to guess the initial value of the parameter vector with its initial covariance matrix. If there is not enough prior knowledge, then a simple way is to make the parameters equal to one with very high covariance matrix.

B.2.5 Extra Applications of Linear Least Squares Algorithm

The extension of using least squares algorithm to identify nonlinear discrete models, provided that these models are still in linear in parameter format, is a straightforward. Moreover, the least squares algorithm can also be applied to estimate the parameters of linear time-invariant continuous system in form of differential equation or transfer function. This is by making the higher derivatives equal to the rest of other derivatives of the output and the input. This form is linear in parameter and therefore the least squares algorithm can be used to estimate the parameters (Mikles and Fikar 2007). Also, an extension to some nonlinear terms is possible here, as long as the formulation stays in the form of linear in parameters. But, the need of calculating the derivatives of the input and the output signals and the need of relatively high sampling rates makes this numerically unattractive in online practical applications.

B.3 Nonlinear Least Squares Parametric Methods

In this section, an algorithm of ‘*Gauss-Newton*’ for minimizing the functional nonlinear least squares performance index is introduced. This can be applied on output error or prediction error parametric identification methods. The algorithm is first developed as an offline and iterative algorithm, and then transformed into a recursive algorithm to be used as an online identification algorithm especially in real-time control applications. Moreover, this algorithm can be applied to identify the parameters of continuous as well as discrete models.

B.3.1 Output Error Method

The method is a parametric method that searches for optimal model parameters that minimize the performance index the sum of squared errors, which are the difference between the process outputs and the estimated model outputs that are or were generated in an identification experiment by using consistently exciting inputs, as shown in the following Figure B.4. The difference between the process and the model output is called the output error and calculated by

$$e_{OE}(t, \boldsymbol{\theta}) = y_m(t) - \hat{y}_u(t, \boldsymbol{\theta}), \quad (\text{B.40})$$

where $y_m(t)$ is the measured output of the process that meant to be identified, and $\hat{y}_u(t)$ is the calculated output of the model. The model output $\hat{y}_u(t)$ is calculated using the model parameters and the input signal as

$$\hat{y}_u(t) = \mathcal{M}(\boldsymbol{\theta}, u), \quad (\text{B.41})$$

where \mathcal{M} is a mathematical model, which it can be linear or nonlinear differential or difference equation, continuous or discrete transfer function or in state space format, the variable $\boldsymbol{\theta}$ is the parameter vector of the model \mathcal{M} , and u is the input signal. The sum of squared errors performance index, for N samples, is defined as

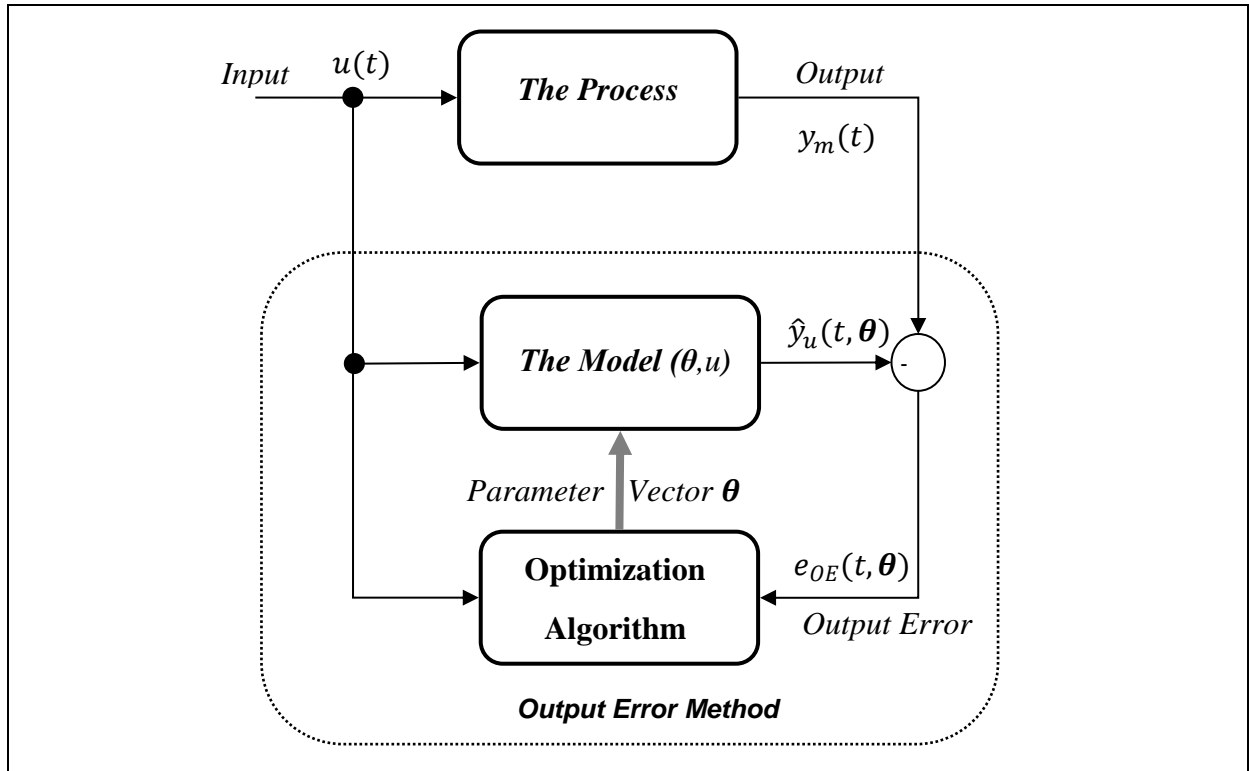


Figure B.4: Output error parameter identification strategy.

$$J_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{t=1}^N e_{OE}^2(t, \boldsymbol{\theta}). \quad (\text{B.42})$$

If the output error is a nonlinear function of the model parameters, for example, continuous transfer function, continuous and discrete state space models, then the optimization solution can only be found numerically by successive approximation methods (Verhaegen and Verdult 2007). For example, the solution can be searched for by approximating or expanding the performance index functional (B.42) in Taylor series around an initial estimate of the parameter vector $\boldsymbol{\theta}(k)$

$$\begin{aligned} J_N(\boldsymbol{\theta}) &= J_N(\boldsymbol{\theta}(k)) + [J'_N(\boldsymbol{\theta}(k))]^T [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)] \\ &\quad + \frac{1}{2} [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)]^T J''_N(\boldsymbol{\theta}(k)) [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)] \\ &\quad + \text{higher order terms}, \end{aligned} \quad (\text{B.43})$$

where $J'_N(\boldsymbol{\theta}(k))$ is the Jacobian of the functional $J_N(\boldsymbol{\theta})$ at $\boldsymbol{\theta}(k)$ and defined as

$$J'_N(\boldsymbol{\theta}(k)) = \frac{\partial J_N(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial J_N(\boldsymbol{\theta})}{\partial \theta_1} \\ \frac{\partial J_N(\boldsymbol{\theta})}{\partial \theta_2} \\ \vdots \\ \frac{\partial J_N(\boldsymbol{\theta})}{\partial \theta_{N_p}} \end{bmatrix}, \quad (\text{B.44})$$

where N_p is the number of the parameters, and $J''_N(\boldsymbol{\theta}(k))$ is the Hessian of the functional $J_N(\boldsymbol{\theta})$ at $\boldsymbol{\theta}(k)$ and defined as

$$J_N''(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \frac{\partial}{\partial \boldsymbol{\theta}^T} J_N(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_{N_p}} \\ \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_2} & \cdots & \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_{N_p}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_{N_p} \partial \theta_1} & \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_{N_p} \partial \theta_2} & \cdots & \frac{\partial^2 J_N(\boldsymbol{\theta})}{\partial \theta_{N_p} \partial \theta_{N_p}} \end{bmatrix}. \quad (\text{B.45})$$

Generally, the computational solution of the functional $J_N(\boldsymbol{\theta})$ in equation (B.43) and its Jacobian and Hessian are very burdensome for nonlinear in parameters model structures. Therefore, in order to reduce the computational effort and complexity, some approximation procedures are introduced in the next subsections B.3.1.1 and B.3.1.2.

B.3.1.1 The Newton method

The Newton method approximates the cost functional $J_N(\boldsymbol{\theta})$ by ignoring all of its higher order terms in Taylor series expansion equation (B.43) as following

$$J_N(\boldsymbol{\theta}) = J_N(\boldsymbol{\theta}(k)) + [J_N'(\boldsymbol{\theta}(k))]^T [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)] + \frac{1}{2} [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)]^T J_N''(\boldsymbol{\theta}(k)) [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)]. \quad (\text{B.46})$$

Now, the minimum of the functional $J_N(\boldsymbol{\theta})$ is found by deriving its derivative with respect to $\boldsymbol{\theta}$ and made equal to zero

$$J_N'(\boldsymbol{\theta}) = 0 = J_N'(\boldsymbol{\theta}(k)) + J_N'(\boldsymbol{\theta}(k)) + [J_N''(\boldsymbol{\theta}(k))]^T [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)] + [J_N''(\boldsymbol{\theta}(k))] [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)] + \frac{1}{2} [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)]^T J_N'''(\boldsymbol{\theta}(k)) [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)], \quad (\text{B.47})$$

and by ignoring the third order derivative term of equation (B.47)

$$\frac{1}{2} [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)]^T J_N'''(\boldsymbol{\theta}(k)) [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)] = 0, \quad (\text{B.48})$$

then equation (B.47) can be approximated by

$$J_N'(\boldsymbol{\theta}) = 0 = J_N'(\boldsymbol{\theta}(k)) + [J_N''(\boldsymbol{\theta}(k))] [\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)], \quad (\text{B.49})$$

and therefore, the next parameter update is calculated by

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - J_N''(\boldsymbol{\theta}(k))^{-1} J_N'(\boldsymbol{\theta}(k)). \quad (\text{B.50})$$

This type of parameter update, equation (B.50), is called Newton method, where it needs an initial guess of the parameter vector and the values of the Jacobian and Hessian at the initial guess to compute the next parameter estimate. Now, the output error vector is defined by

$$\mathbf{E}_N(\boldsymbol{\theta}) = [e_{OE}(0, \boldsymbol{\theta}) \quad e_{OE}(1, \boldsymbol{\theta}) \quad \cdots \quad e_{OE}(N-1, \boldsymbol{\theta})]^T, \quad (\text{B.51})$$

so that the cost functional can be rewritten as

$$J_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{t=1}^N e_{OE}^2(t, \boldsymbol{\theta}) = \frac{1}{N} \mathbf{E}_N^T(\boldsymbol{\theta}) \mathbf{E}_N(\boldsymbol{\theta}). \quad (\text{B.52})$$

Also, the gradient matrix of the error vector with respect to the parameters is defined by

$$\Psi_N(\boldsymbol{\theta}) = \frac{\partial \mathbf{E}_N(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} = \begin{bmatrix} \frac{\partial e_{OE}(0, \boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial e_{OE}(0, \boldsymbol{\theta})}{\partial \theta_2} & \dots & \frac{\partial e_{OE}(0, \boldsymbol{\theta})}{\partial \theta_{N_p}} \\ \frac{\partial e_{OE}(1, \boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial e_{OE}(1, \boldsymbol{\theta})}{\partial \theta_2} & \dots & \frac{\partial e_{OE}(1, \boldsymbol{\theta})}{\partial \theta_{N_p}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{OE}(N-1, \boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial e_{OE}(N-1, \boldsymbol{\theta})}{\partial \theta_2} & \dots & \frac{\partial e_{OE}(N-1, \boldsymbol{\theta})}{\partial \theta_{N_p}} \end{bmatrix}. \quad (\text{B.53})$$

By using the calculus of functional differentiation (Brewer 1978), the Jacobian and Hessian of $J_N(\boldsymbol{\theta})$ can be expressed as

$$\begin{aligned} J'_N(\boldsymbol{\theta}) &= \frac{2}{N} \left(\frac{\partial \mathbf{E}_N(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right)^T \mathbf{E}_N(\boldsymbol{\theta}); \\ &= \frac{2}{N} \Psi_N^T(\boldsymbol{\theta}) \mathbf{E}_N(\boldsymbol{\theta}), \end{aligned} \quad (\text{B.54})$$

and

$$\begin{aligned} J''_N(\boldsymbol{\theta}) &= \frac{2}{N} \frac{\partial^2 \mathbf{E}_N^T(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T \partial \boldsymbol{\theta}} (\mathbf{I}_P \otimes \mathbf{E}_N(\boldsymbol{\theta})) + \frac{2}{N} \left(\frac{\partial \mathbf{E}_N(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right)^T \frac{\partial \mathbf{E}_N(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}; \\ &= \frac{2}{N} \frac{\partial^2 \mathbf{E}_N^T(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T \partial \boldsymbol{\theta}} (\mathbf{I}_P \otimes \mathbf{E}_N(\boldsymbol{\theta})) + \frac{2}{N} \Psi_N^T(\boldsymbol{\theta}) \Psi_N(\boldsymbol{\theta}), \end{aligned} \quad (\text{B.55})$$

which are still extensive to compute, therefore, it needs more simplifications as presented in the next subsection.

B.3.1.2 The Gauss-Newton method

Moreover, the Gauss-Newton method will approximate the Hessian $J''_N(\boldsymbol{\theta})$ by the matrix

$$\mathbf{H}_N(\boldsymbol{\theta}) = \frac{2}{N} \Psi_N^T(\boldsymbol{\theta}) \Psi_N(\boldsymbol{\theta}). \quad (\text{B.56})$$

This approximation is valid around the optimal parameters as long as the correlation between the error and its second derivative is weak, so that the first term of equation (B.55) is omitted. Doing so will make the computation much easier. If the matrix $\mathbf{H}_N(\boldsymbol{\theta})$ is invertible, then the Gauss-Newton method parameter update formula can be written as

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mathbf{H}_N(\boldsymbol{\theta}(k))^{-1} J'_N(\boldsymbol{\theta}(k)). \quad (\text{B.57})$$

From last equation (B.57), the Jacobean and the Hessian approximate are needed to be calculated at every iteration. Therefore, equations (B.53) and (B.54) lead to the fact that only $\mathbf{E}_N(\boldsymbol{\theta})$ and $\Psi_N(\boldsymbol{\theta})$ are needed to be calculated. Furthermore, $\mathbf{E}_N(\boldsymbol{\theta})$ is found by computing the output estimate $\hat{y}_u(t, \boldsymbol{\theta})$ for $(t = 1, 2, \dots, N)$, which is done by simulating the following discrete state space system equation (B.58), or alternatively continuous, if the identification model is continuous.

$$\begin{aligned} \hat{\mathbf{x}}(t+1, \boldsymbol{\theta}) &= \mathbf{A}(\boldsymbol{\theta}) \hat{\mathbf{x}}(t, \boldsymbol{\theta}) + \mathbf{B}(\boldsymbol{\theta}) u(t); \\ \hat{y}_u(t, \boldsymbol{\theta}) &= \mathbf{C}(\boldsymbol{\theta}) \hat{\mathbf{x}}(t, \boldsymbol{\theta}). \end{aligned} \quad (\text{B.58})$$

This will also give the signal $\hat{\mathbf{x}}(t, \boldsymbol{\theta})$, which is needed in computing $\boldsymbol{\Psi}_N(\boldsymbol{\theta})$, as shown in the following

$$\boldsymbol{\Psi}_N(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial e_{OE}(0, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \\ \frac{\partial e_{OE}(1, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \\ \vdots \\ \frac{\partial e_{OE}(N-1, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \hat{y}_u(0, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \\ \frac{\partial \hat{y}_u(1, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \\ \vdots \\ \frac{\partial \hat{y}_u(N-1, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \end{bmatrix}, \quad (\text{B.59})$$

and that

$$\frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} = \left[\frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_1} \quad \frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_2} \quad \dots \quad \frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_{N_p}} \right], \quad (\text{B.60})$$

where θ_i is the i th entry of the vector $\boldsymbol{\theta}$, and for every parameter θ_i there will be

$$\begin{aligned} \frac{\partial \hat{\mathbf{x}}(t+1, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{A}(\boldsymbol{\theta}) \frac{\partial \hat{\mathbf{x}}(t, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \hat{\mathbf{x}}(t, \boldsymbol{\theta}) + \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i} u(t); \\ \frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{C}(\boldsymbol{\theta}) \frac{\partial \hat{\mathbf{x}}(t, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \hat{\mathbf{x}}(t, \boldsymbol{\theta}). \end{aligned} \quad (\text{B.61})$$

By defining $\mathbf{X}_i(t, \boldsymbol{\theta}) = \frac{\partial \hat{\mathbf{x}}(t, \boldsymbol{\theta})}{\partial \theta_i}$, equations (B.61) can be rewritten as

$$\begin{aligned} \mathbf{X}_i(t+1, \boldsymbol{\theta}) &= \mathbf{A}(\boldsymbol{\theta}) \mathbf{X}_i(t, \boldsymbol{\theta}) + \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \hat{\mathbf{x}}(t, \boldsymbol{\theta}) + \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i} u(t); \\ \frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{C}(\boldsymbol{\theta}) \mathbf{X}_i(t, \boldsymbol{\theta}) + \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \hat{\mathbf{x}}(t, \boldsymbol{\theta}). \end{aligned} \quad (\text{B.62})$$

This means that the derivative of $\hat{y}_u(t, \boldsymbol{\theta})$ with respect to θ_i is obtained by simulating a linear system with state vector $\mathbf{X}_i(t, \boldsymbol{\theta})$ and inputs $\hat{\mathbf{x}}(t, \boldsymbol{\theta})$ and $u(t)$. The partial derivatives of model matrices with respect to the parameters, equation (B.63), are normally constant.

$$\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i}, \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i}, \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i}. \quad (\text{B.63})$$

Therefore, the calculation of $\boldsymbol{\Psi}_N(\boldsymbol{\theta})$ is simply done by simulating a linear system for every parameter. This means at the end, $N_p + 1$ linear systems are needed to be simulated for N_p parameters. Figure B.5 shows the schematic diagram of the method, where starting from an initial parameter vector, a simulation run is made to accumulate the necessary vectors and matrices, after accumulating N samples, a new update of the parameter vector is made. This calculation method is iterative, therefore, in order to get close enough to the optimal parameters, the algorithm needs to be repeated many times until the performance index reaches the desired minimum. The method is a batch (offline) algorithm, since it needs to collect the whole data over the simulation period to make up the necessary vectors and matrices in order to compute the new estimate of the parameter vector.

Moreover, as can be seen from the derivation of the Gauss-Newton method, it is an approximate and iterative (successive) solution when the initial parameter guess is around or near the optimal one. But, when the initial guess is far from the optimal one, it takes a lot more

iterations to get near to the optimal one (provided that all convergence conditions are available). Therefore, for a better global behavior, the Gauss-Newton method needs to be conditioned, for example, by extending it to the Levenberg-Marquardt method, see , for example, Verhaegen and Verdult (2007) or Nelles (2001). Where it works as steepest decent method when the estimate is far from the optimal one, which in this case makes a better approach behavior to the optimal region than Gauss-Newton method, in other words, it needs less iterations to go towards the optimal region. And it works as Gauss-Newton method, when the estimate is around the optimal region, and in this case, it has a better approach behavior to the optimal estimate than the steepest decent method.

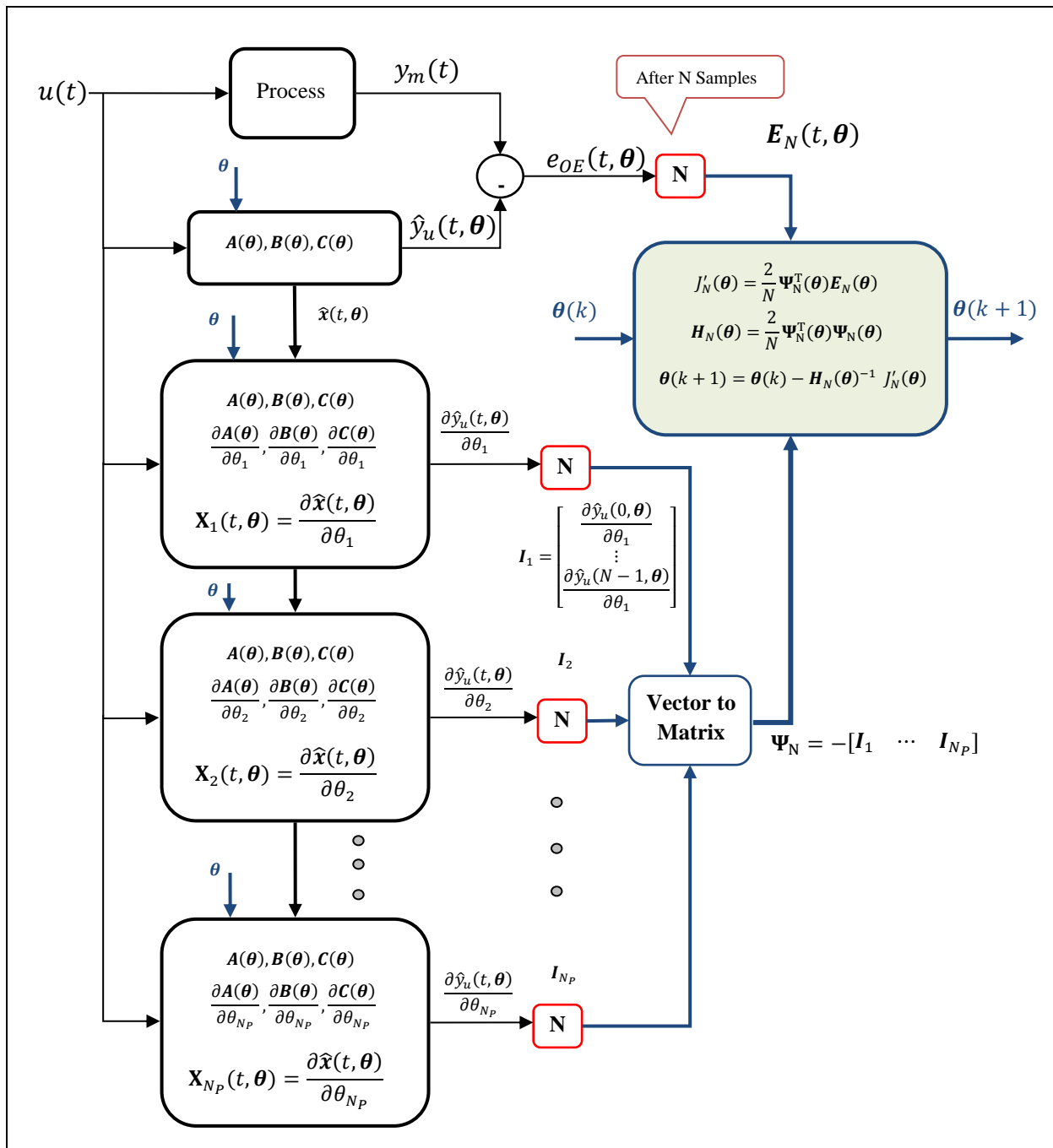


Figure B.5: Batch (offline) iterative (successive) Gauss-Newton output error method.

B.3.2 Recursive Gauss-Newton Method

The so far presented way of calculation is good for batch computations, but the problem is that it needs to memorize and compute large vectors and matrices. For online identification and control, it is better to use recursive calculation methods, where a new parameter vector update is calculated in every simulation update or sample hit, and thereby, only the last sample hit data is needed. The new parameters update can be computed recursively (Ljung and Söderström 1983) as

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \frac{1}{t} \mathbf{H}^{-1}(t) \boldsymbol{\Psi}(t) e_{OE}(t), \quad (\text{B.64})$$

Where variable $\boldsymbol{\Psi}(t)$ is the gradient vector. The Hessian matrix is calculated by

$$\mathbf{H}(t) = \mathbf{H}(t-1) + \frac{1}{t} [\boldsymbol{\Psi}(t) \boldsymbol{\Psi}^T(t) - \mathbf{H}(t-1)]. \quad (\text{B.65})$$

Alternatively, to avoid the Hessian matrix inversion at every parameter update, the covariance matrix can be defined as

$$\mathbf{P}(t) = \frac{1}{t} \mathbf{H}^{-1}(t), \quad (\text{B.66})$$

with its recursive computation formula

$$\mathbf{P}(t) = \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1) \boldsymbol{\Psi}(t) \boldsymbol{\Psi}^T(t) \mathbf{P}(t-1)}{1 + \boldsymbol{\Psi}^T \mathbf{P}(t-1) \boldsymbol{\Psi}(t)}. \quad (\text{B.67})$$

So, the recursive (online) algorithm with the forgetting factor (λ) becomes as following:

Starting with these initial conditions: $\hat{\boldsymbol{\theta}}(0)$ and $\mathbf{P}(0)$.

Get the input and the output samples $u(t), y_m(t)$.

Run the system model

$$\begin{aligned} \hat{\mathbf{x}}(t+1, \boldsymbol{\theta}) &= \mathbf{A}(\boldsymbol{\theta}) \hat{\mathbf{x}}(t, \boldsymbol{\theta}) + \mathbf{B}(\boldsymbol{\theta}) u(t); \\ \hat{y}_u(t, \boldsymbol{\theta}) &= \mathbf{C}(\boldsymbol{\theta}) \hat{\mathbf{x}}(t, \boldsymbol{\theta}). \end{aligned} \quad (\text{B.68})$$

Calculate the output error

$$e_{OE}(t, \boldsymbol{\theta}) = y_m(t) - \hat{y}_u(t, \boldsymbol{\theta}). \quad (\text{B.69})$$

Run the systems, for $i = 1$ to N_p of parameters

$$\begin{aligned} \mathbf{X}_i(t+1, \boldsymbol{\theta}) &= \mathbf{A}(\boldsymbol{\theta}) \mathbf{X}_i(t, \boldsymbol{\theta}) + \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \hat{\mathbf{x}}(t, \boldsymbol{\theta}) + \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i} u(t); \\ \frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_i} &= \mathbf{C}(\boldsymbol{\theta}) \mathbf{X}_i(t, \boldsymbol{\theta}) + \frac{\partial \mathbf{C}(\boldsymbol{\theta})}{\partial \theta_i} \hat{\mathbf{x}}(t, \boldsymbol{\theta}). \end{aligned} \quad (\text{B.70})$$

And let

$$\boldsymbol{\Psi}(t) = \left[\frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_1} \quad \dots \quad \frac{\partial \hat{y}_u(t, \boldsymbol{\theta})}{\partial \theta_{N_p}} \right]^T. \quad (\text{B.71})$$

Update the covariance matrix

$$\mathbf{P}(t) = \mathbf{P}(t-1) - \frac{1}{\lambda} \frac{\mathbf{P}(t-1)\boldsymbol{\Psi}(t)\boldsymbol{\Psi}^T(t)\mathbf{P}(t-1)}{\lambda + \boldsymbol{\Psi}^T\mathbf{P}(t-1)\boldsymbol{\Psi}(t)}. \quad (\text{B.72})$$

Update the parameters

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}(t-1) + \mathbf{P}(t)\boldsymbol{\Psi}(t)e_{OE}(t). \quad (\text{B.73})$$

Loop

The algorithm is also presented schematically in the following Figure B.6.

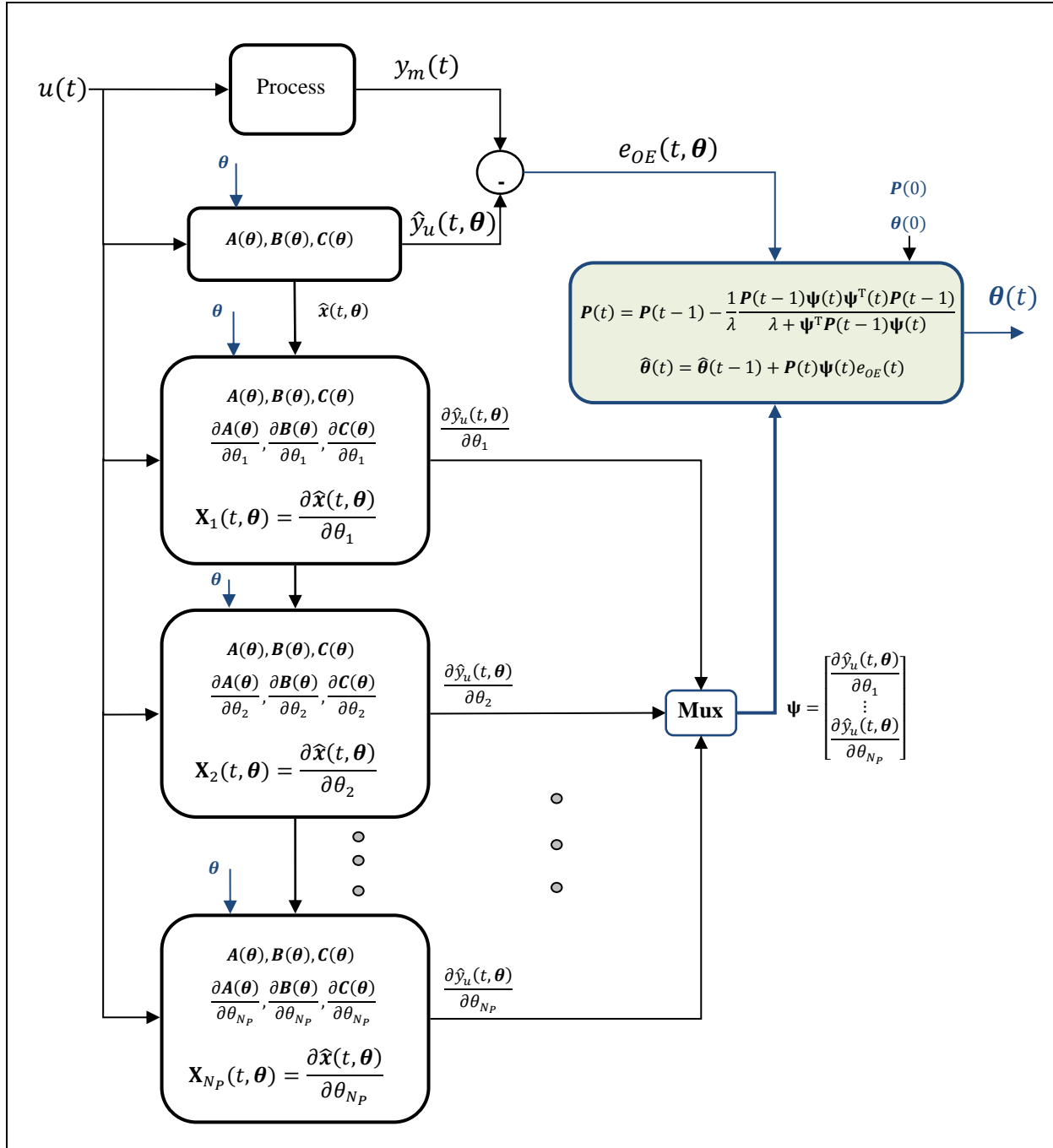


Figure B.6: Online recursive Gauss-Newton output error method.

B.3.3 Prediction Error Method

The prediction error method is made by optimizing the parameters of a predictor in order that a cost functional of prediction error least squares sum is minimal (Verhaegen and Verdult 2007). The prediction error is the difference between the measured output and the prediction model output, see Figure B.7, as

$$e_{PE}(t, \boldsymbol{\theta}) = y_m(t) - \hat{y}_m(t, \boldsymbol{\theta}). \quad (\text{B.74})$$

And the cost functional, the sum of squared prediction errors, is defined as

$$J_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{t=1}^N e_{PE}^2(t, \boldsymbol{\theta}). \quad (\text{B.75})$$

The predictor can be constructed, especially in state space, by an observer. The observer uses both the input and the measured output signals to estimate the output prediction as following

$$\begin{aligned} \hat{\mathbf{x}}(t+1, \boldsymbol{\theta}) &= \mathbf{A}(\boldsymbol{\theta})\hat{\mathbf{x}}(t, \boldsymbol{\theta}) + \mathbf{B}(\boldsymbol{\theta})u(t) + \mathbf{K}_o(\boldsymbol{\theta})e_{PE}(t); \\ \hat{y}_m(t, \boldsymbol{\theta}) &= \mathbf{C}(\boldsymbol{\theta})\hat{\mathbf{x}}(t, \boldsymbol{\theta}), \end{aligned} \quad (\text{B.76})$$

or

$$\begin{aligned} \hat{\mathbf{x}}(t+1, \boldsymbol{\theta}) &= [\mathbf{A}(\boldsymbol{\theta}) - \mathbf{K}_o(\boldsymbol{\theta})\mathbf{C}(\boldsymbol{\theta})]\hat{\mathbf{x}}(t, \boldsymbol{\theta}) + \mathbf{B}(\boldsymbol{\theta})u(t) + \mathbf{K}_o(\boldsymbol{\theta})y_m(t); \\ \hat{y}_m(t, \boldsymbol{\theta}) &= \mathbf{C}(\boldsymbol{\theta})\hat{\mathbf{x}}(t, \boldsymbol{\theta}). \end{aligned} \quad (\text{B.77})$$

So, the already developed Gauss-Newton method can be used to search for the parameters of the innovation model $\{\mathbf{A}(\boldsymbol{\theta}), \mathbf{B}(\boldsymbol{\theta}), \mathbf{C}(\boldsymbol{\theta}), \mathbf{K}_o(\boldsymbol{\theta})\}$ that will minimize the performance index (B.75). Although the method is developed in discrete linear state space the reformulation to the continuous state space is straightforward.

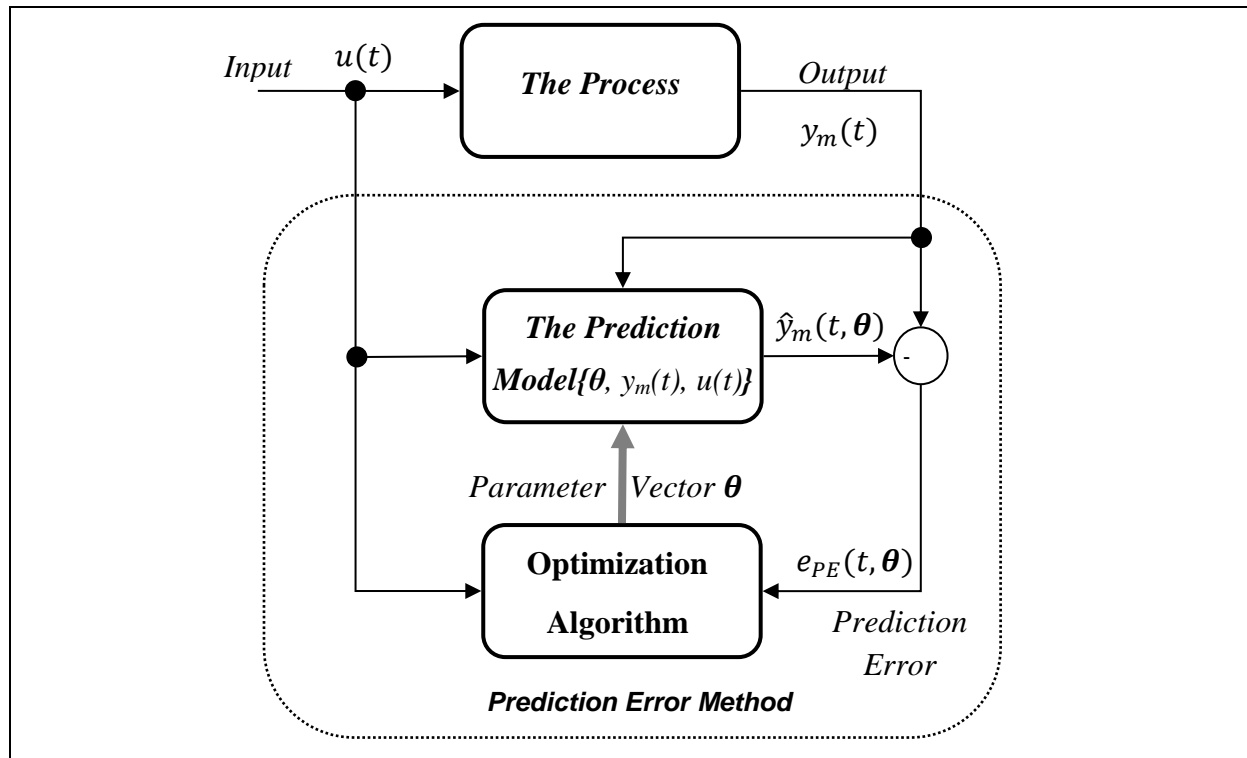


Figure B.7: Prediction error parameter identification strategy.

B.4 Practical Aspects of Parametric System Identification

For a process, which is operating locally around an operating point, its identification input signal should have frequency spectrum rich enough to consistently excite its dynamics that are needed to be identified. Moreover, the input signal amplitude should also be small enough to keep the operation around the targeted operating point as well as large enough to get a good signal to noise ratio. On the other hand, if the process is operating more globally around the operating point, which gives more emphasize to some of its nonlinear characteristics so that a nonlinear identification model is needed. Therefore, for this case, the input signal should have the proper frequency spectrum and the amplitude should have all level variations in order to identify the global nonlinear behavior of the process in the specified operating region.

B.4.1 Identification in Closed Loop

The most of system identification methods, like the previously presented in this chapter, are originally designed to work with data sets recorded in open loop experiments. Therefore, applying these methods by using data sets gathered in closed loop operation will make one of their main design assumptions; the input is uncorrelated with the output measurement error, no longer more valid (Van den Hof 1998; Forssell and Ljung 1999). This makes the process unidentifiable directly in closed loop. This is why alternative indirect methods are developed to identify processes operating in the closed loop by using external signals, for example, by using the set point as an input to identify the closed loop with a known controller, therefore, the process model is computed by using the identified closed loop model and the known controller model indirectly. However, there are some restricted experimental conditions that lead to a consistent direct parameter identification in closed loop operation (Nelles 2001), especially when the system is sufficiently excited either by the set point or by an external disturbance, and the controller must be complex enough. Particularly for recursive least squares method, it is to remove the linear correlation between the input and the output signals in order to identify the process parameters uniquely. This can be solved by letting the controller, for example, to have a higher order, switching or different settings during the identification phase, a nonlinear behavior or time-varying parameters.

Fortunately, these conditions confirm with the application of the adaptive controller (Aström and Wittenmark 1995), especially when the adaptive controller is at the tuning phase, where the controller parameters are time-varying and switch between different values. This is generally a nonlinear behavior (see also Söderström and Stoica 1989, section 10.3). These controller parameter variations could lead to un-correlation between the input and the output measurement error. This has been already seen in the experimental results, where the estimated parameter quick convergence always happens in closed loop operation particularly in the direct adaptive phases. Also, this assumption could lead to the fact that after the convergence phase, especially for identification with small forgetting factor, the parameters will possibly diverge again if the mentioned conditions disappear. So, it is wise to deactivate the adaption phase, when the parameters have converged, especially, when the process is clearly time-invariant for a period of time, and to reactivate it, when the process parameters vary or the operational conditions change again.

References

- Abel, D., and A. Bollig. *Rapid control prototyping*. Springer-Verlag, 2006.
- Ahn, H.-S., Y. Chen, and K. L. Moore. "Iterative learning control: Brief survey and Categorization." *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, 2007: 1099-1121.
- Alsogkier, I., and C. Bohn. "Identification and control of periodic disturbances." *Proceedings of the 20th Mediterranean Conference on Control and Automation*. Barcelona, 2012. 265-271.
- Aström, K. J. "Maximum likelihood and prediction error methods." *Automatica*, 1980: 551-574.
- Aström, K. J., and B. Wittenmark. *Adaptive control systems*. Addison-Wesley Publishing Company, 1995.
- . *Computer-controlled systems theory and design*. Prentice Hall, 1997.
- Aström, K. J., and P. Eykhoff. "System identification - A survey." *Automatica*, 1971: 123-162.
- Aström, K. J., and T. Bohlin. "Numerical identification of linear dynamic systems from normal operating records." *Proceedings of the IFAC Symposium Self-Adaptive Systems*. Teddington, U.K., 1965. 96-111.
- Ballesteros, P., and C. Bohn. "A frequency-tunable LPV controller for narrowband active noise and vibration control." *Proceedings of the American Control Conference*. San Francisco, 2011a. 1340-1345.
- . "Disturbance rejection through LPV gain-scheduling control with application to active noise cancellation." *Proceedings of the 18th IFAC World Congress*. Milano, 2011b. 7897-7902.
- Basu, K., J. S. Prasad, G. Narayanan, H. K. Krishnamurthy, and R. Ayyanar. "Reduction of torque ripple in induction motor drives using an advanced hybrid PWM technique." *IEEE Transactions on Industrial Electronics*, 2010: 2085-2091.
- Bodson, M. "Equivalence between adaptive cancellation algorithms and linear time-varying compensators." *Proceedings of the 43rd Conference on Decision and Control*. Atlantis, 2004. 638-643.
- . "Rejection of periodic disturbances of unknown and time varying frequency." *International Journal of Adaptive Control and Signal Processing*, 2005: 67-88.
- Bohn, C., A. Cortabarría, V. Härtel, and K. Kowalczyk. "Active control and of engine-induced vibrations in automotive vehicles using disturbance observer gain scheduling." *Control Engineering Practice*, 2004: 1029-1039.

-
- Brewer, J. W. "Kronecker products and matrix calculus in system theory." *IEEE Transactions on Circuits and Systems*, 1978: 772-781.
- Bristow, D. A., M. Tharayil, and A. G. Alleyne. "A survey of iterative learning control, a learning-based method for high-performance tracking control." *IEEE Control Systems Magazine*, 2006: 96-114.
- dSPACE. *DS1104 R&D controller board hardware installation and configuration for release 4.2*. dSPACE GmbH, 2005a.
- . *Real-Time interface (RTI and RTI-MP) implementation guide for release 4.2*. dSPACE GmbH, 2005b.
- Du, C., H. Li, C. K. Thum, F. L. Lewis, and Y. Wang. "Simple disturbance observer for disturbance compensation." *IET Control Theory and Applications*, 2010: 1748-1755.
- Du, H., and X. Shi. "Gain-scheduled H-infinity control for use in vibration suppression of system with harmonic excitation." *Proceedings of the American Control Conference*. Anchorage, 2002. 4668-4669.
- Du, H., L. Zhang, Z. Lu, and X. Shi. "LPV technique for the rejection of sinusoidal disturbance with time-varying frequency." *IEE Proceedings on Control Theory and Applications*, 2003: 132-138.
- Elliott, S. *Signal processing for active control*. Academic Press, 2001.
- Ellis, G. *Observers in control systems a practical guide*. Academic Press, 2002.
- Fasol, K. H., and H. P. Jörgl. "Principles of model building and identification." *Automatica*, 1980: 505-518.
- Filatov, N. M., and H. Unbehauen. *Adaptive control theory and applications*. Springer-Verlag, 2004.
- . "Survey of adaptive dual control methods." *IEE Proceedings on Control Theory and Applications*, 2000: 118-128.
- Forsell, U., and L. Ljung. "Closed-loop identification revisited." *Automatica*, 1999: 1215-1241.
- Francis, B. A., and W. M. Wonham. "The internal model principle of control theory." *Automatica*, 1976: 457-465.
- Gan, W., and L. Qui. "Torque and velocity ripple elimination of AC permanent magnet motor control systems using the internal model principle." *IEEE/ASME Transactions on Mechatronics*, 2004: 436-447.
- Garimella, S. S., and K. Srinivasan. "Application of repetitive control to eccentricity compensation in rolling." *Proceedings of the American Control Conference*. Baltimore, 1994. 2904-2908.
- Garnier, H., and L. Wang. *Identification of continuous-time models from sampled data*. Springer-Verlag, 2008.

- Gevers, M. "A personal view of the development of system identification. A 30-year journey through an exciting field." *IEEE Control System Magazine*, 2006: 93-105.
- Godfrey, K. R. "Correlation methods." *Automatica*, 1980: 527-534.
- Gusev, S. V., W. Johnson, and J. Miller. "Active flywheel control based on the method of moment restrictions." *Proceedings of the American Control Conference*. Albuquerque, 1997. 3426-3430.
- Haber, R., and L. Keviczky. *Nonlinear system identification input-output modeling approach*. Kluwer Academic Publishers, 1999.
- Hansen, C. H. *Understanding active noise cancellation*. Spon Press, 2001.
- Harris, C. J., and S. A. Billings. *Self-Tuning and adaptive control theory and applications*. Peter Peregrinus Ltd., 1981.
- Harris, C. M., and A. G. Piersol. *Harris' shock and vibration handbook*. McGraw-Hill, 2002.
- Ho, B. L., and R. E. Kalman. "Effective construction of linear state-variable models from inputoutput functions." *Regelungstechnik*, 1965: 545-548.
- Holm, M., P. Ballesteros, S. Beitler, A. Tarasow, and C. Bohn. "Active control of speed fluctuations in rotating machines using feedback linearization." *Proceedings of the UKACC International Conference on Control*. Cardiff, 2012. 64-69.
- Holtz, J., and L. Springob. "Identification and compensation of torque ripple in high-precision permanent magnet motor drives." *IEEE Transactions on Industrial Electronics*, 1996: 309-320.
- Hong, J., and D.S. Bernstein. "Bode integral constraints, colocation, and spillover in active noise and vibration control." *IEEE Transactions on Control Systems Technology*, 1998: 111-120.
- Isermann, R. "Practical aspects of process identification." *Automatica*, 1980: 575-587.
- Isermann, R., and M. Münchhof. *Identification of dynamic systems*. Springer-Verlag, 2011.
- Itoh, K., and H. Kubota. "Thrust ripple reduction of linear induction motor with direct torque control." *Proceedings of the 8th International Conference on Electrical Machines and Systems*. Nanjing, China, 2005. 655-658.
- Jahns, T. M., and W. L. Soong. "Pulsating torque minimization techniques for permanent magnet AC motor drives - A review." *IEEE Transactions on Industrial Electronics*, 1996: 321-330.
- Kalman, R. E. "A new approach to linear filtering and prediction problems." *Transactions of the ASME-Journal of Basic Engineering*, 1960: 35-45.
- Kalman, R. E., and R. S. Bucy. "New results in linear filtering and prediction theory." *Transactions of the ASME-Journal of Basic Engineering*, 1961: 95-108.
- Kim, Y.-H., S.-H Han, S.-I Cho, and I.-J Ha. "Learning approach to control of servomotors under disturbance torque dependent on time and states." *IEE Proceedings- Control Theory and Applications*, 1998: 251-258.

-
- Kinney, C. E., and R. A. de Callafon. "The internal model principle for periodic disturbances with rapidly time-varying frequencies." *International Journal of Adaptive Control and Signal Processing*, 2011: 1006-1022.
- . "Scheduling control for periodic disturbance attenuation." *Proceedings of the American Control Conference*. Minneapolis, 2006. 4788-4793.
- Kobayashi, F., S. Hara, and H. Tanaka. "Reduction of motor speed fluctuation using repetitive control." *Proceedings of the 29th Conference on Decision and Control*. Honolulu, 1990. 1697-1702.
- Kugi, A., W. Hass, K. Schlacher, K. Aistleitner, H. Frank, and W. R. Günter. "Active compensation of roll eccentricity in rolling mills." *IEEE Transactions on Industry Applications*, 2000: 625-632.
- Kuo, S.M., and D.M. Morgan. *Active noise control systems*. John Wiley and Sons, 1996.
- Landau, I. D. "Unbiased recursive identification using model reference adaptive techniques." *IEEE Transactions on Automatic Control*, 1976: 194-202.
- Landau, I. D., R. Lozano, and M. M'Saad. *Adaptive control*. Springer-Verlag, 1998.
- Li, T., and G. Slemon. "Reduction of cogging torque in permanent magnet motors." *IEEE Transactions on Magnetics*, 1988: 2901-2903.
- Ljung, L. *System identification theory for the user*. Prentice Hall, 1999.
- Ljung, L., and T. Söderström. *Theory and practice of recursive identification*. MIT Press, 1983.
- Lueg, P. Process of silencing sound oscillations. US Patent 2,043,416. 1936.
- Luenberger, D. G. "Observing the state of a linear system." *IEEE Transactions on Military Electronics*, 1964: 74-80.
- Maier, S., J. Bals, and M. Bodson. "Periodic disturbance rejection of a PMSM with adaptive control algorithms." *Proceedings of the IEEE International Electric Machines and Drives Conference*. Niagara Falls, 2011. 1070-1075.
- Mikles, J., and M. Fikar. *Process modeling, identification and control*. Springer-Verlag, 2007.
- Morgan, D.R. "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path." *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1980: 454-467.
- Na, H., and Y. Park. "An adaptive feed-forward controller for rejection of periodic disturbances." *Journal of Sound and Vibration*, 1997: 427-435.
- Nagashima, M., K. Usui, and M. Kobayashi. "Rejection of unknown periodic disturbances in magnetic hard disk drives." *Proceedings of the Asia-Pacific Magnetic Recording Conference*. Singapore, 2006. 1-2.
- Nelles, O. *Nonlinear system identification: From classical approach to neural networks and fuzzy models*. Springer-Verlag, 2001.

- Njeh, M., S. Cauet, and P. Coirault. "Persistent disturbances rejection on internal combustion engine torque in hybrid electric vehicles." *Proceedings of the 49th Conference on Decision and Control*. Atlanta, 2010. 6421-6426.
- Norton, J. P. *An introduction to identification*. Academic Press, 1986.
- Ohishi, K. "Robust position servo system based on vibration suppression control for industrial robotics." *Proceedings of the International Power Electronics Conference*. Sapporo, 2010. 2230-2237.
- . "A new servo method in mechatronics." *Transactions of Japan Society of Electrical Engineering*, 1987: 83-86.
- Östman, F., and H. T. Toivonen. "Model-Based torsional vibration control of internal combustion engines." *IET Control Theory and Applications*, 2008: 1024-1032.
- Parks, T. R. *Manual for model 205/505a, torsional control system (instructor's edition)*. ECP Educational Control Products, 1999.
- Petrovic, V., R. Ortega, A. M. Stankovic, and G. Tadmor. "Design and implementation of an adaptive controller for torque ripple minimization in PM synchronous motors." *IEEE Transactions on Power Electronics*, 2000: 871-880.
- Phillips, Charles L., and H. Troy Nagle. *Digital control system analysis and design*. Prentice Hall, 1995.
- Pipeleers, G., B. Demeulenaere, and J. Swevers. *Optimal linear controller design for periodic inputs*. Springer-Verlag, 2009.
- Quan, Q., and K. Cai. "A survey of repetitive control for nonlinear systems." *Journal of Science Foundation in China*, 2010: 45-53.
- Radke, A., and Z. Gao. "A survey of state and disturbance observers for practitioners." *Proceedings of the American Control Conference*. Minneapolis, 2006. 5183-5188.
- Rake, H. "Step response and frequency response methods." *Automatica*, 1980: 519-526.
- Sacks, A., M. Bodson, and P. Khosla. "Experimental results of adaptive periodic disturbance cancellation in a high performance magnetic disk drive." *Proceedings of the American Control Conference*. San Francisco, 1993. 686-690.
- Schmidt, P., and T. Rehm. "Notch filter tuning for resonant frequency reduction in dual inertia systems." *Proceedings of the IEE Industry Applications Society Conference: The 34th IAS Annual Meeting*. Phoenix, 1999. 1730-1734.
- Servax Drives. *Anwendungshandbuch TorqueChampion Direktantriebe Direktantriebsregler CDD3000*. Servax Drives, 2003a.
- . *Betriebsanleitung TorqueChampion Direktantriebe Direktantriebsregler CDD3000*. Servax Drives, 2003b.
- . *Handbuch PC-Benutzersoftware DriveManager TorqueChampion Direktantriebe*. Servax Drives, 2003c.

-
- Shin, K., J. Jang, J. Kang, and S. Song. "Compensation method for tension disturbance due to an unknown roll shape in a web transport system." *IEEE Transactions on Industry Applications*, 2003: 1422-1428.
- Skoczowski, S., S. Domek, and K. Pietruszewicz. "Model following PID control system." *Kybernetes*, 2003: 818-828.
- Smith, S. W. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, 1997.
- Smolders, K., B. Demeulenaere, J. D. Schutter, and J. Swevers. "Average speed control of reciprocating machinery." *Proceedings of the 13th Mediterranean Conference on Control and Automation*. Cyprus, 2005. 71-76.
- Söderström, T., and P. Stoica. *System identification*. Prentice Hall, 1989.
- Steinbuch, M. "Repetitive control for systems with uncertain period-time." *Automatica*, 2002: 2103-2109.
- Strejc, V. "Least squares parameter estimation." *Automatica*, 1980: 535-550.
- Van den Hof, P. "Closed-loop issues in system identification." *Annual Reviews in Control*, 1998: 173-186.
- Verhaegen, M., and V. Verdult. *Filtering and system identification a least squares approach*. Cambridge University Press, 2007.
- Vinogradov, O. *Fundamentals of kinematics and dynamics of machines and mechanisms*. CRC Press LLC, 2000.
- Wang, A. K., and W. Ren. "A new indirect adaptive algorithm for feed forward control." *Proceedings of the American Control Conference*. Baltimore, 1994. 2990-2994.
- Wang, J., W. C. Gan, and L. Qiu. "A gain scheduled controller for sinusoidal ripple elimination of AC PM motor systems." *Proceedings of the 15th Mediterranean Conference on Control and Automation*. Athens, 2007. 1-6.
- Widrow, B., and S. D. Stearns. *Adaptive signal processing*. Prentice Hall, 1985.
- Wikipedia. 2012. http://en.wikipedia.org/wiki/Scotch_yoke (accessed Januar 9, 2012).
- Wit, C. C., and L. Parly. "Adaptive eccentricity compensation." *IEEE Transactions on Control System Technology*, 2000: 757-766.
- Yoo, S., W. Lee, Y. Bea, and M. D. Noh. "Optimal notch filter for active magnetic bearing controllers." *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Budapest, 2011. 707-711.
- Zaremba, A. T., and R. I. Davis. "Control design for active engine damping using a starter/alternator." *Proceedings of the American Control Conference*. Chicago, 2000. 2043-2047.
- Zaremba, A. T., I. V. Burkov, and R. M. Stuntz. "Active damping of engine speed oscillations based on learning control." *Proceedings of the American Control Conference*. Philadelphia, 1998. 2143-2147.

CURRICULUM VITA

Name: Izziddien Ahmed Abdella
Family name: Alsogkier
Date and place of birth: 09.09.1972, Tripoli, Libya.
Nationality: Libyan.
Passport No.: 753574.
Marital status: Married with two children.
Address: Kleiner Bruch 2, 38678 Clausthal-Zellerfeld.

School

1978-1883 Elementary school in Alkhoms, Libya.
1984-1986 Preparatory school in Alkhoms, Libya.
1987-1990 Secondary school in Alkhoms, Libya.

Study

1991-1995 Faculty of Electronic Engineering, Control Department (B.Sc. degree), Nasser University, Bani-Walid, Libya.

Work

1996 National service.
1997 Libya Telecom Company.
1998 Assistant lecturer in the Faculty of Engineering (Electricity and Computer Department), Nasser University.
2001 A scholarship award for Master of Science degree study.

Study

2002 German Language course at Goethe Institute in Bonn.
2003-2006 Study for M.Sc. degree in Hannover University.

Work

2006-2008 Lecturer at the University of El-Mergeb, Faculty of Engineering-Electrical Engineering/Control department.
2008 A scholarship award for promotional (Ph.D.) study.

Study

2009-2014 Promotion student at the institute of Electrical Information Technology, Faculty of Mathematics/Computer Science and Mechanical Engineering, Clausthal University of Technology.