WELCOME IN ROMANIA 2026!

NATIONAL AUTORITY OF QUALIFICATION (ANC)

&

INTERNATIONAL UNIVERSITY ASSOCIATION OF SCIENCE AND TECHNOLOGY FROM ROMANIA (UASTRO)

(www.uastro.space)

Organize

between 25 June to 29 June 2026

The 9-th edition of the International Conference of the Space Launching Systems (SLS)

&

The 21-th Edition of the International Conference on Artificial Intelligence applied in the fields of Aerospace, Robotics, Manufacturing Systems, Mechanical Engineering, Biomechanics, Power Energy, Technology of Materials and Neurorehabilitation (OPTIROB)

OPEN CEREMONY- TURQUOISE HOTEL, VENUS, CONSTANTA

FRIDAY 26 June 2026 10:00

www.uastro.space

www.ijmo.org

IJMO
ISSN 2010-3697
Home
Current issue
Archive
Author Guidelines
Reviewers Guidelines
Editorial Board
About IJMO
Conference
International Journal of Modeling and Optimization
SUBMIT Manuscript
Archive
please select
What's New
Nov 20, 2013 News!Welcome to Dr. John Kaiser Calautit join the Editorial Board of IJMO
Nov 19, 2013 News!Vol.2, No.3 has been indexed by Crossref
Nov 15, 2013 News!Vol.2, No.2 has been indexed by Crossref
General Information
ISSN: 2010-3697
Frequency: Bimonthly
DOI: 10.7763/IJMO
Editor-in-Chief: Prof. Adrian Olaru
Executive Editor: Mr. Ron C. Wu
Abstracting/ Indexing: Engineering & Technology Digital Library, ProQuest, Crossref, Electronic Journals Library, DOAJ, Google Scholar, EI (INSPEC, IET).
READ NOW
International Journal of Modeling and Optimization
International Journal of Modeling and Optimization (IJMO) is an international academic journal which gains a foothold in Singapore, Asia and opens to the world. It aims to promote the integration of modeling and optimization. The focus is to publish papers on state-of-the-art modeling and optimization. Submitted papers will be reviewed by technical committees of the Journal and Association. The audience includes researchers, managers and operators for modeling and optimization as well as designers and developers.
All submitted articles should report original, previously unpublished research results, experimental or theoretical, and will be peer-reviewed. Articles submitted to the journal should meet these criteria and must not be under consideration for publication elsewhere. Manuscripts should follow the style of the journal and are subject to both review and editing.
Interested in being a peer-reviewer?
Please fill in the application form and send it back to the editor, then your application will be processed soon..
Any opinion about IJMO?
You are encouraged to write a letter to the editor and share your valuable Opinions with us.

www.ijeee.net

mathematics
an Open Access Journal by MDPI

IMPACT FACTOR 2.592

CITESCORE 2.9 SCOPUS

SPECIAL ISSUE
"DYNAMIC MODELING AND SIMULATION FOR CONTROL SYSTEMS"

- **GOAL:** this special issue will address topics including the mathematical modeling of dynamic behavior; optimization algorithms; assisted theoretical and experimental research; the control of physical engineering systems; mechanical, electrical and fluid interaction components; system response analysis; feedback control systems; numerical software and software for dynamic simulation and optimization; system stability; dynamic behavior in the frequency field. papers related to the modeling, simulation, and optimization of control systems are welcome.
- **Guest Editors:** Prof. Ph.D. Eng. Adrian Olaru,
  Prof.Ph.D.Eng. Gabriel Frumusanu,
  Prof.Ph.D.Eng. Catalin Alexandru
- **Website:** https://www.mdpi.com/si/mathematics/Dyn_Modeling_Simul_Control_Syst_III
- **DEADLINE** FOR MANUSCRIPT SUBMISSIONS: 31.03.2026 (AN EXTENSION CAN BE GRANTED, IF NECESSARY)
- **AUTHOR BENEFITS:** OPEN ACCESS; HIGH VISIBILITY; NO EXTRA CHARGES; THOROU PEER-REVIEW AND DECISIONS; COVERAGE BY LEADING INDEXING SERVICES

Special Issue QR:

Dynamic modelling and simulation for control systems using the transfer multipol functions

Adrian Olaru[1][0000-0001-6062-4657] , Serban Olaru [2][0000-0001-6993-8599] , Niculae Mihai [3]

[1] National University for Science and Technology Politehnica Bucharest, Romania
[2]Military Equipment and Technologies Research Agency (METRA), Romania
[3]TechnoAccord, Leuval, Canada

ICAPE 2025
Beihang University International Conference
Beijing, China
12-15 December 2025

CONTENT

1. Introduction of the dynamic modelling and simulation for control systems
2. Theory of the transfer functions with multipol modelling
3. Neural Network modelling
4. Example of the multipol transfer functions analyse
5. State of art of components and satellite orientation ways
6. Application in mechanical inertial orientation satellite- mathematical modelling and simulation of the proposed method
7. Assisted analyse of the proposed algorithm and some results
8. Analyse of the obtained results
9. Conclusions
10. Future work
11. References

1. INTRODUCTION OF THE DYNAMIC MODELLING AND SIMULATION FOR CONTROL SYSTEMS

- Modelling and simulating servo systems using transfer functions and multipole modules (multiple inputs, multiple outputs and multiple reference quantities) are strictly necessary steps in the analysis of dynamic behaviour and the transition to the next stage: servo system synthesis.

These steps include the following activities:

- (i) writing the mathematical model for each component of the servo system;
- (ii) eliminating intermediate variables and determining a relationship between input and output quantities;
- (iii) creating the block diagram using elementary transfer functions of the first and second order proportional derivative type, first and second order inertia integrator, first and second order proportional with inertia, multiplier, adder, constants;

- (iv) creating the block diagram using subroutines of the LabView software platform for simulation and assisted analysis;
- (v) determining real and frequency characteristics with highlighting the parameters and performances of the dynamic behaviour;
- (vi) the actual analysis by drawing comparative characteristics with the modification of some constructive-functional sizes and highlighting the comparative dynamic behaviour parameters and performances.

2. THEORY OF THE TRANSFER FUNCTIONS WITH MULTIPOL MODELLING

- The transfer functions theory applied to the elements and the systems **using the LabVIEW** non linear components assure one very easily mode of the **modeling, simulation and validation** of the elements and systems, finally to obtain by sinthesys one integrated and intelligent system;

- In the paper will be presented **one virtual LabVIEW propre library** for the assisted research of the electrical and hydralic elements and systems with many results what will be possible to use in the curently research;

- In the optimising field was used some neural network and the **on-line research** of the network influences to the finally target of the servo driving system;

- With applied this theory was possible **to design one proper smart system to decrease the vibration** and optimise the space trajectory;

- With designed LabVIEW VI-s will be possible **to choose the optimal values of the constructive-functional parameters** of the components of the system.

Table 1. Some expressions and virtual LabVIEW instruments of transfer functions

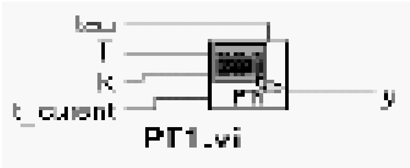
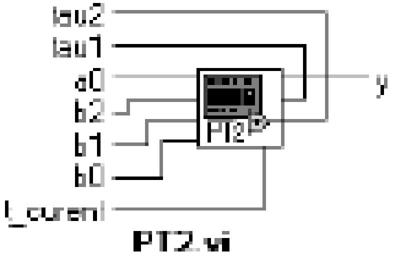
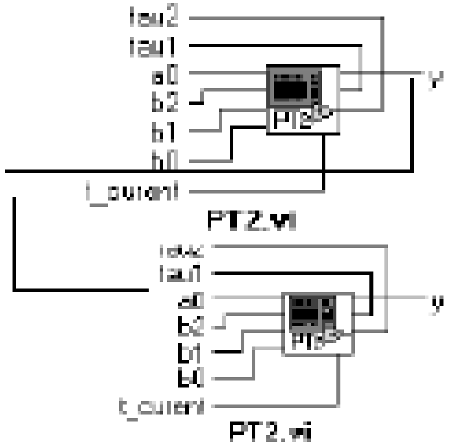
Type	Expresion of transfer function	Virtual LabVIEW instrument
PT_1	$H(s)=\dfrac{k}{T_1 s+1}$	
PT_2	$H(s)=\dfrac{k}{(s+a)(s+b)}$ $\xi>1$	
	$H(s)=\dfrac{k}{s^2+a^2}$ $\xi=0$	
	$H(s)=\dfrac{k}{(s+a)^2}$ $\xi=1$	
	$H(s)=\dfrac{k\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$ $0<\xi<1$	
PT_3	$H(s)=\dfrac{k}{(s+a)(s+b)(s+c)}$	
PT_4	$H(s)=\dfrac{k}{(s+a)(s+b)(s+c)(s+d)}$	

Table contents: type of the transfer functions; Mathematical model; virtual LabVIEW instrumentation

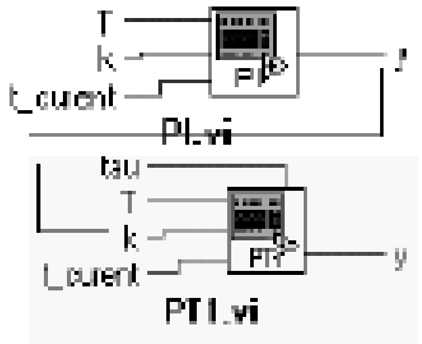
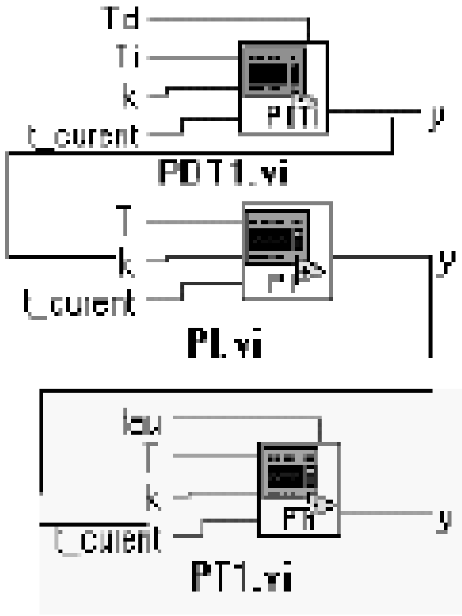
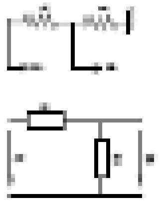
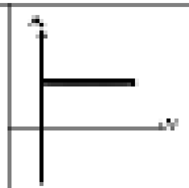
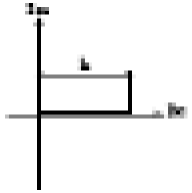
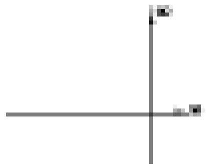
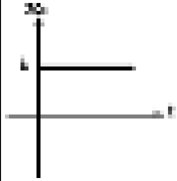
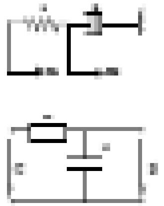
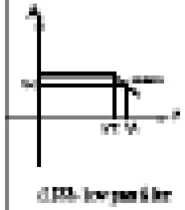
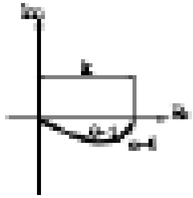
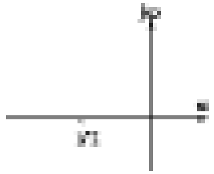
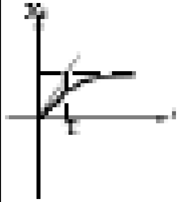
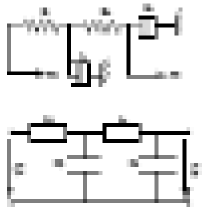
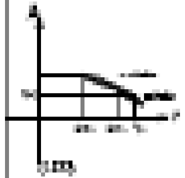
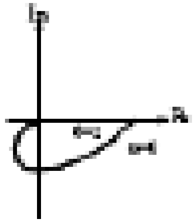
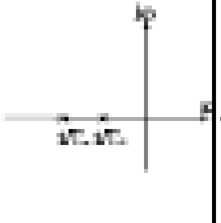
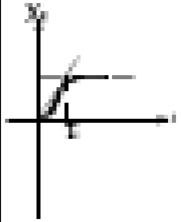
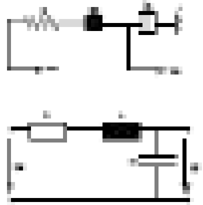
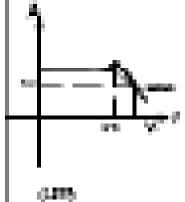
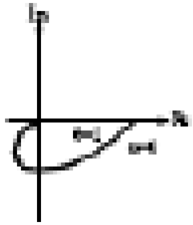
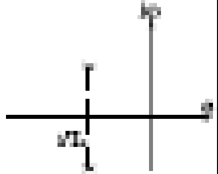
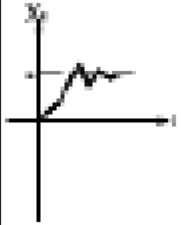
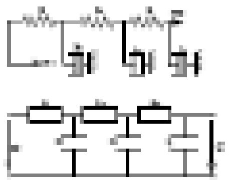
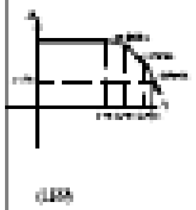
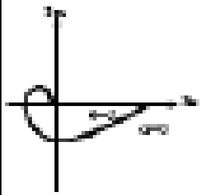
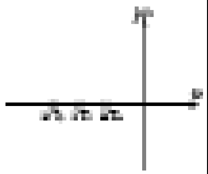
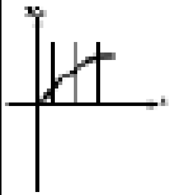
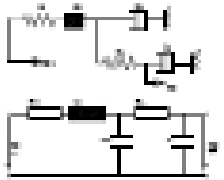
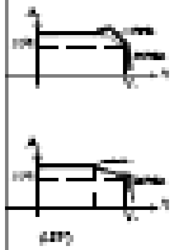
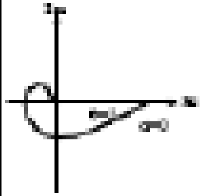
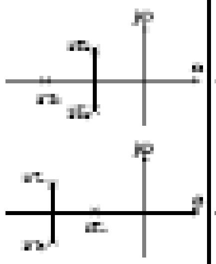
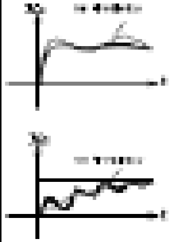
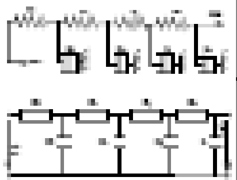
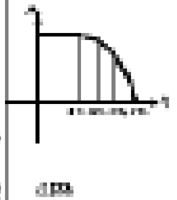
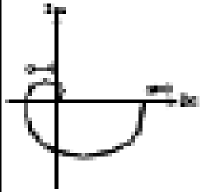
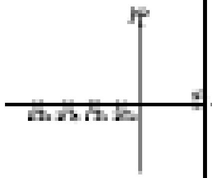
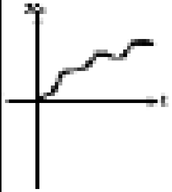
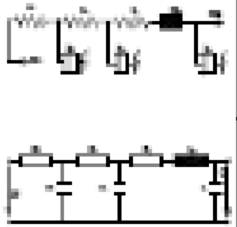
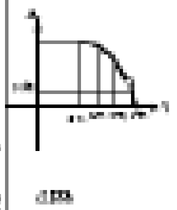
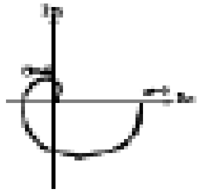
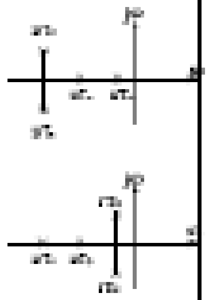
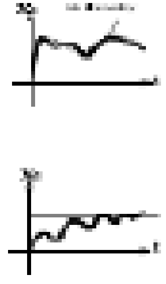
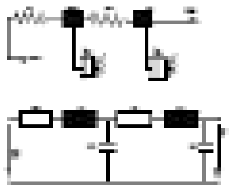
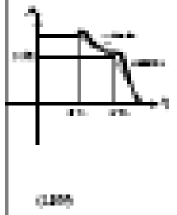
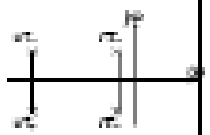
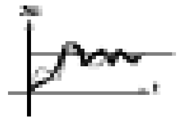
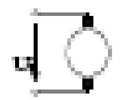
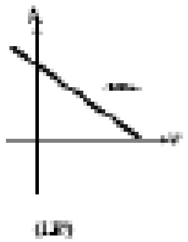
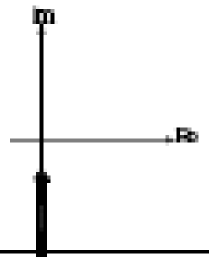
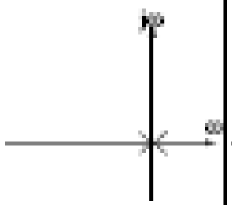
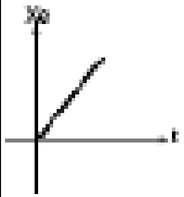
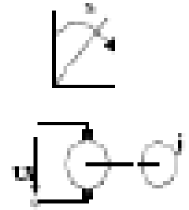
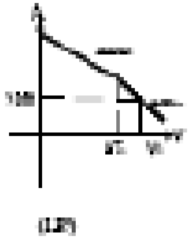
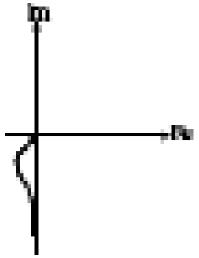
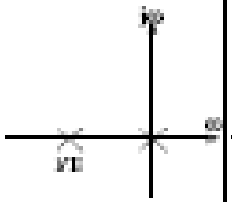
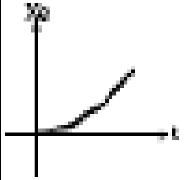
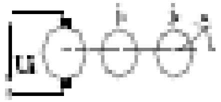
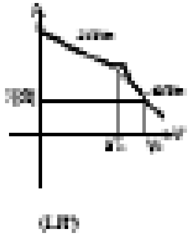
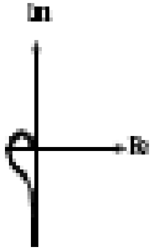
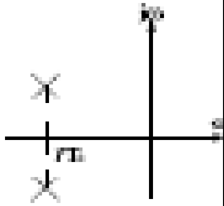
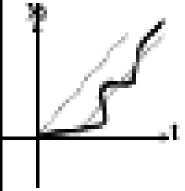
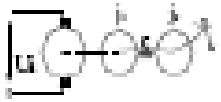
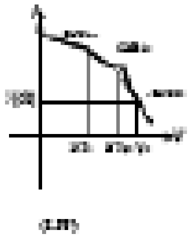
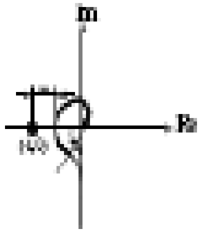
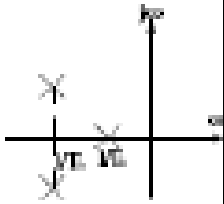
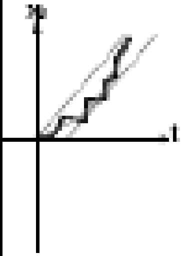
I	$H(s)=\frac{k}{s}$	
IT_1	$H(s)=\frac{k}{s}\cdot\frac{1}{T_i s+1}$	
PDT_1	$H(s)=\frac{k(T_d s+1)}{T_i s+1}$	
DT_1	$H(s)=\frac{T_d s}{Ts+1_i}$	
PID	$H(s)=k(1+T_d s+\frac{1}{Ts_i})$	
PID T1	$H(s)=k(1+T_d s+\frac{1}{Ts_i})\cdot\frac{1}{T_1 s+1}$	

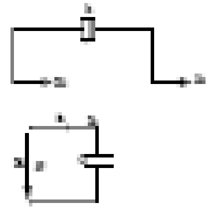
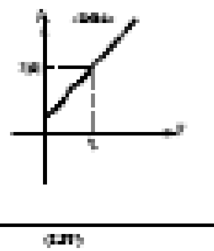
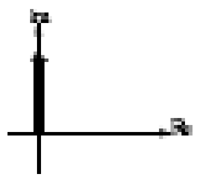
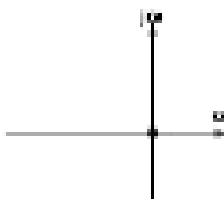
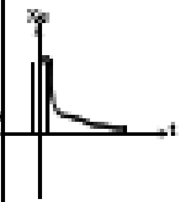
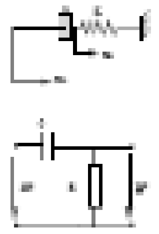
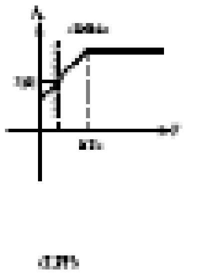
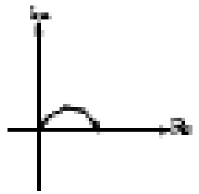
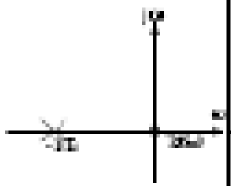
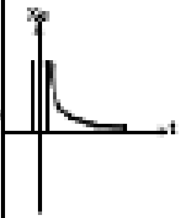
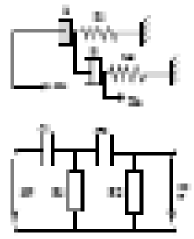
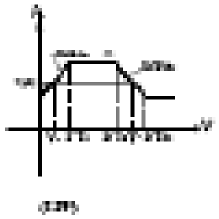
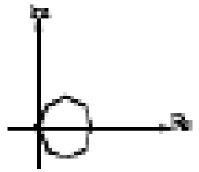
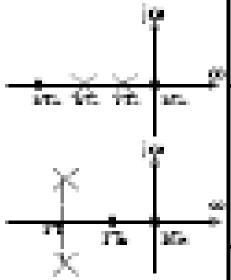
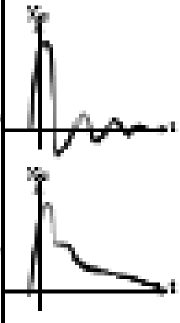
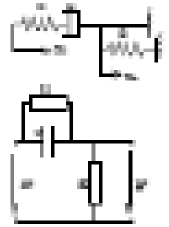
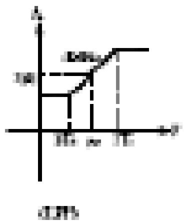
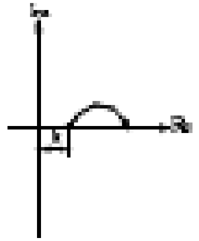
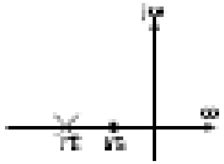
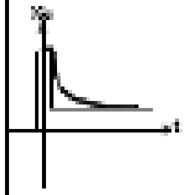
Table 2

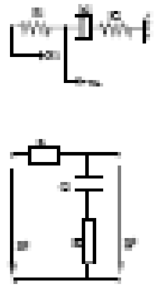
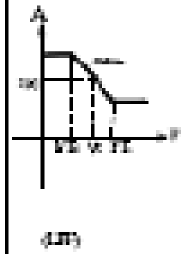
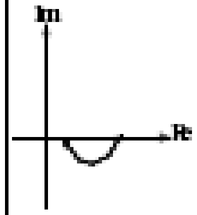
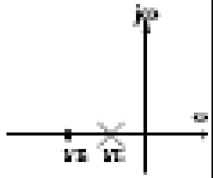
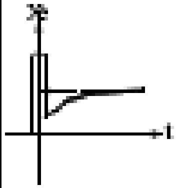
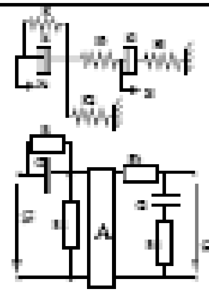
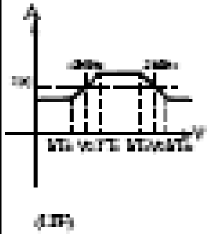
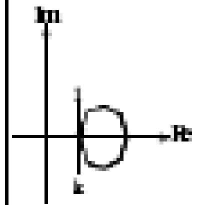
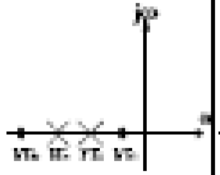
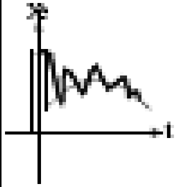
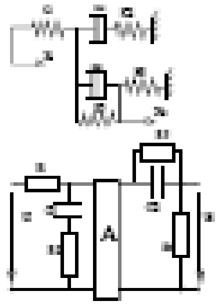
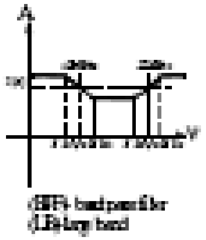
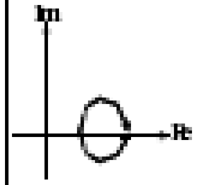
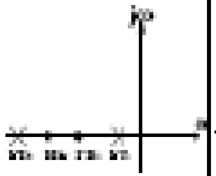
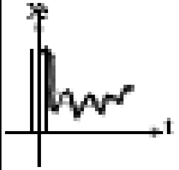
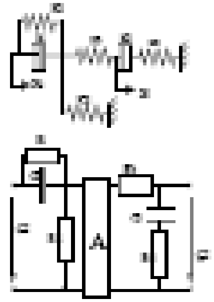
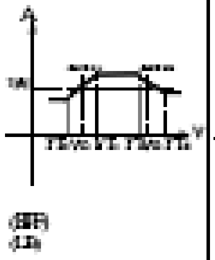
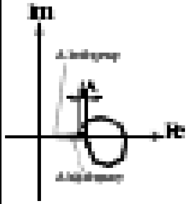
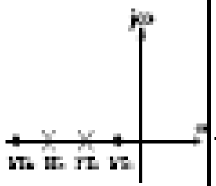
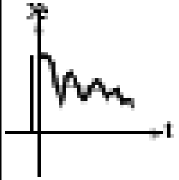
Some models of transfer functions and his characteristics, mathematical and physical models

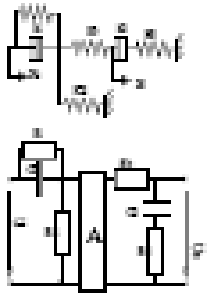
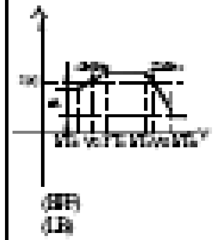
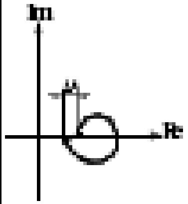
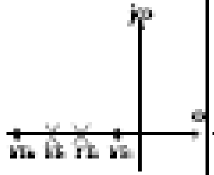
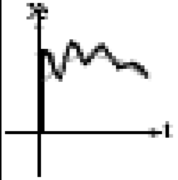
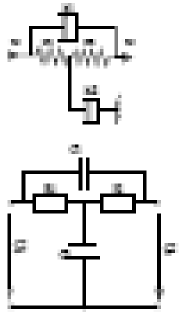
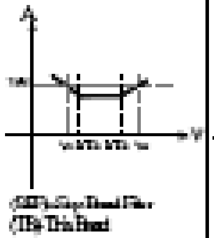
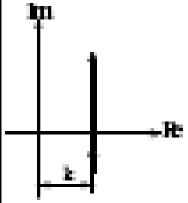
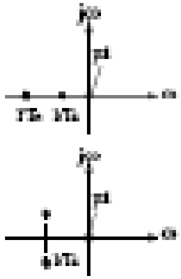
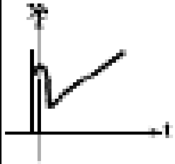
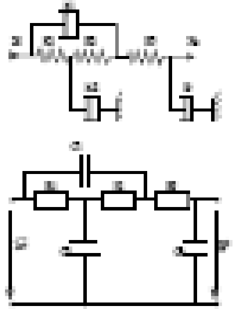
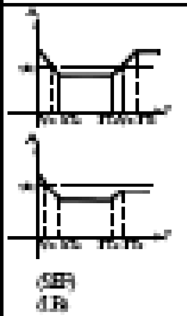
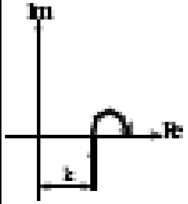
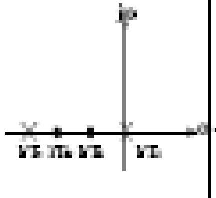
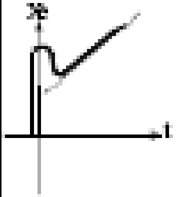
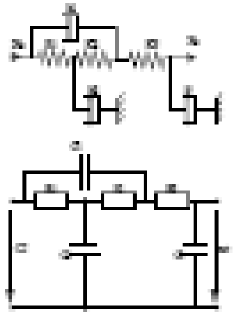
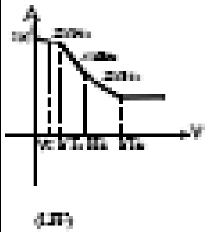
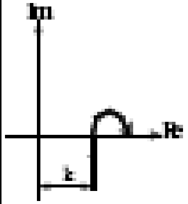
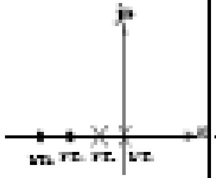
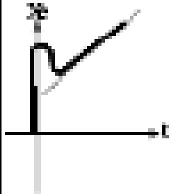
P	$H(s) = K$					
PT1	$H(s) = \frac{K}{T_i s + 1}$					
PT2	$H(s) = \frac{K}{(T_{i_1} s + 1)(T_{i_2} s + 1)}$					
	$H(s) = \frac{K}{(T_{i_1} s + 1)(T_{i_2} s + 1)}$					

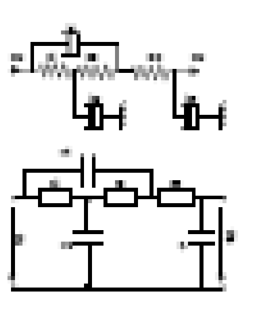
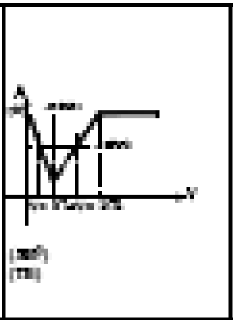
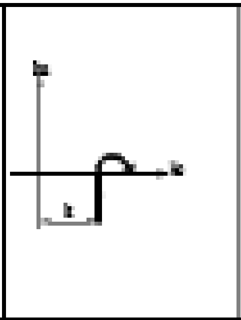
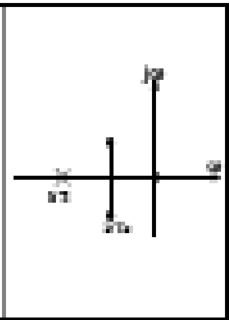
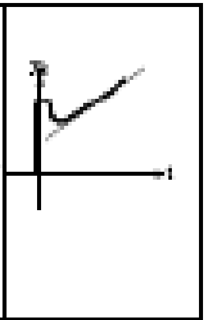
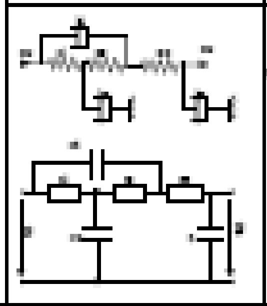
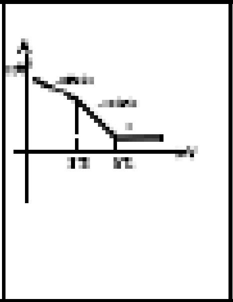
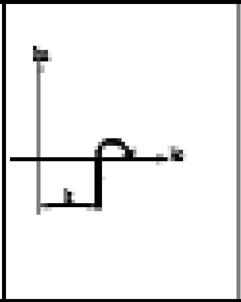
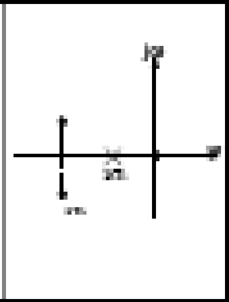
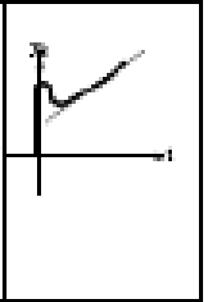
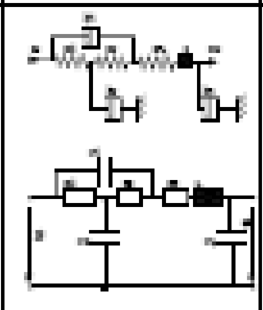
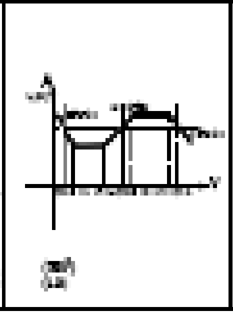
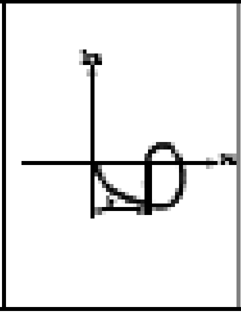
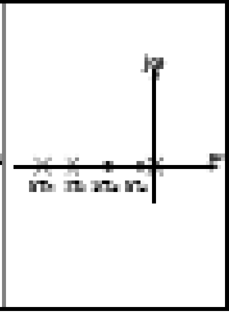
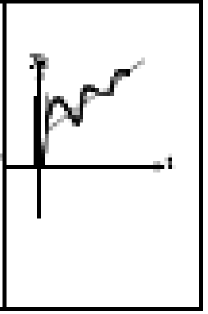
PT3	$H(s)=\dfrac{K}{(T_{i_1}s+1)(T_{i_2}s+1)(T_{i_3}s+1)}$					
	$H(s)=\dfrac{K}{(T_{i_1}s+1)(T_{i_2}s^2+T_{i_3}s+1)}$					
PT4	$H(s)=\dfrac{K}{(T_{i_1}s+1)(T_{i_2}s+1)(T_{i_3}s+1)(T_{i_4}s+1)}$					
	$H(s)=\dfrac{K}{(T_{i_1}s+1)(T_{i_2}s+1)(T_{i_3}s^2+T_{i_4}s+1)}$					
	$H(s)=\dfrac{K}{(T_{i_2}s^2+T_{i_1}s+1)(T_{i_3}s^2+T_{i_4}s+1)}$					

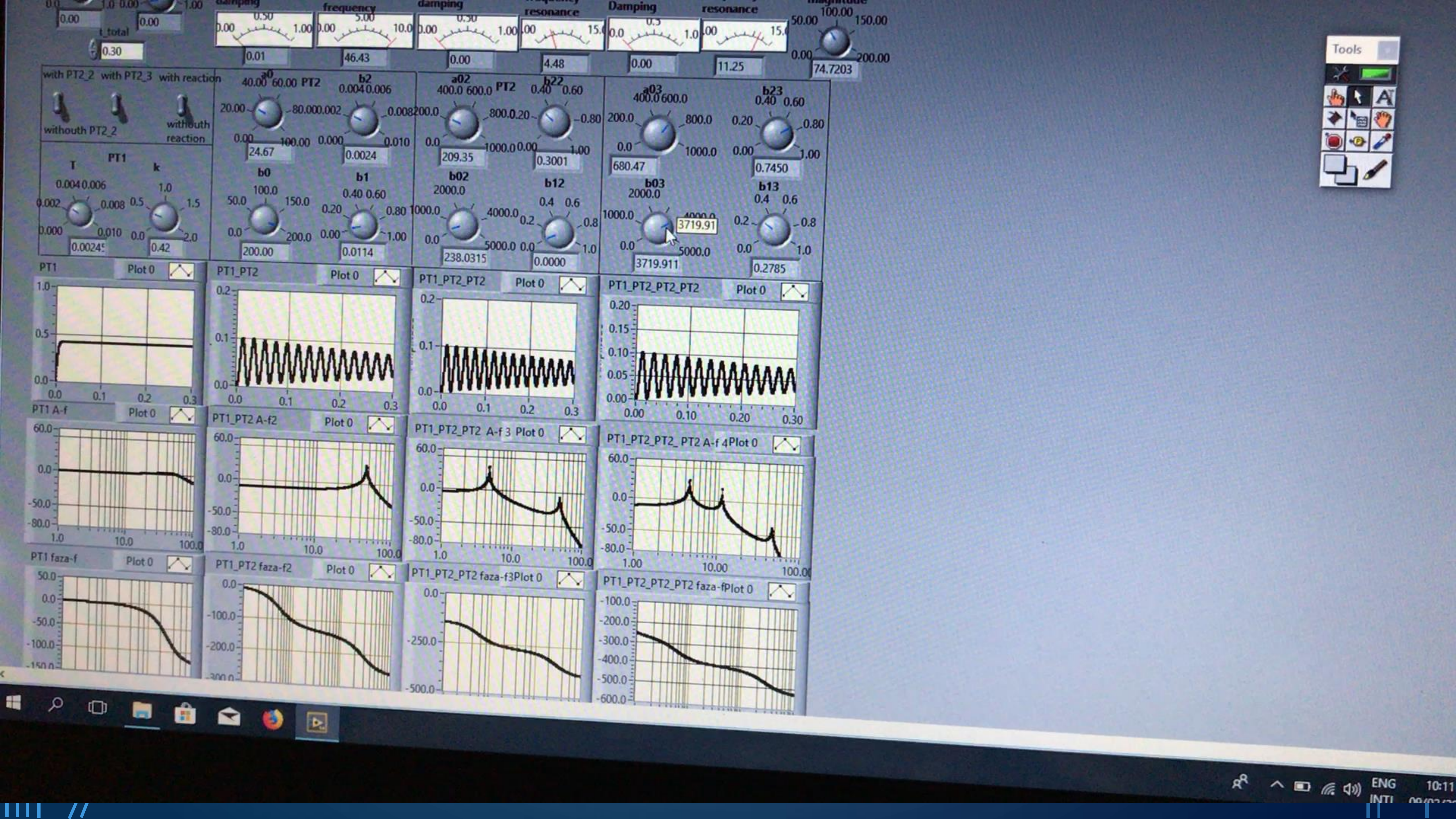
I	$H(s)=\frac{K}{s}$					
IT1	$H(s)=\frac{K}{s}\frac{1}{T_i s+1}$					
IT2	$H(s)=\frac{K}{s}\frac{1}{T_{i_2}s^2+T_{i_1}s+1}$					
IT3	$H(s)=\frac{K}{s}\frac{1}{(T_i s+1)(T_{i_2}s^2+T_{i_1}s+1)}$					

D	$H(s)=T_D s$					
DT1	$H(s)=\frac{T_D s}{T_i s+1}$					
D2T2	$H(s)=\frac{T_{D_2} s^2+T_{D_1} s}{T_{i_2} s^2+T_{i_1} s+1}$					
PDT	$H(s)=\frac{K(T_D s+1)}{T_i s+1}$ $T_D > T_i$					

PDT1	$H(s)=\dfrac{K(T_D s+1)}{T_i s+1}$ $T_D < T_i$					
PD2T2	$H(s)=\dfrac{s^2+(a_1+b_2)s+a_1b_2}{s^2+(a_2+b_1)s+a_2b1}$ $a_1b_2=a_2b_1;\ (a_1+b_2)>(a_2+b_1)$					
PD2T2	$H(s)=\dfrac{s^2+(a_1+b_2)s+a_1b_2}{s^2+(a_2+b_1)s+a_2b1}$ $a_1b_2=a_2b_1;\ (a_1+b_2)<(a_2+b_1)$					
PD2T2	$H(s)=\dfrac{s^2+(a_1+b_2)s+a_1b_2}{s^2+(a_2+b_1)s+a_2b1}$ $a_1b_2=a_2b_1;\ (a_1+b_2)>(a_2+b_1)$ $a>b$					

PD2T2	$H(s)=\frac{s^2+(a_1+b_2)s+a_1b_2}{s^2+(a_2+b_1)s+a_2b_1}$ $a_1b_2=a_2b_1;\ (a_1+b_2)>(a_2+b_1)$ $a<b$					
PID	$H(s)=\left(1+T_Ds+\frac{1}{T_is}\right)K$ $4Td > T$					
PIDT1	$H(s)=K\left(1+T_ds+\frac{1}{T_is}\right)\frac{1}{T_1s+1}$					
						

						
						
PIDT2	$H(s)=K\left(1+T_d s+\frac{1}{T_i s}\right)\frac{1}{T_{t_2} s^2+T_{t_1} s+1}$					



frequency
damping
resonance
Damping
t_total
with PT2_2
with PT2_3
with reaction
withouth PT2_2
withouth reaction
PT1
T
k
PT2
a0
b0
b1
b2
a02
b02
b12
b22
a03
b03
b13
b23
Tools
PT1
PT1_PT2
PT1_PT2_PT2
PT1_PT2_PT2_PT2
PT1 A-f
PT1_PT2 A-f2
PT1_PT2_PT2 A-f 3
PT1_PT2_PT2_ PT2 A-f 4
PT1 faza-f
PT1_PT2 faza-f2
PT1_PT2_PT2 faza-f3
PT1_PT2_PT2_PT2 faza-f
Plot 0
ENG
10:11

CASE STUDY OF THE LINEAR HYDRAULIC MOTOR (LHM)-OPTIMIZING THE DYNAMIC BEHAVIOR OF THE DRIVERS

$$[H(s)] = \frac{\begin{bmatrix} x_{e1}(s) \\ x_{e2}(s) \end{bmatrix}}{\begin{bmatrix} x_{i1}(s) \\ x_{i2}(s) \end{bmatrix}} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$

$$T_1 T_2 \frac{dx_e^2}{dt^2} + h(T_1 + T_2)\frac{dx_e}{dt} + x_e = kU$$

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{T_1 T_2} & -\frac{h(T_1 + T_2)}{T_1 T_2} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{k}{T_1 T_2} \end{pmatrix} U(t)$$

$$Y = (1 \quad 0)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\nu = \frac{\prod \nu_i}{\sum \nu_i}$$

The frequency of the system will be done by the element with the smaller frequency of the system aproximatelly determined by:

$$(x'(t)) = [A](x(t)) + [B](u(t))$$
$$(y(t)) = [C](x(t)) + [D](u(t))$$

$$Y(s) = C^T[sI - A]^{-1}x_0 + C^T[sI - A]^{-1}BU(s) + DU(s)$$

where:

$$T_1T_2 = \frac{m\dfrac{A_1c}{2E}}{A_1^2(1-c_{fu}) + a_mb_m}; h(T_1 + T_2) = \frac{ma_m + \dfrac{A_1c}{2E}b_m}{A_1^2(1-c_{fu}) + a_mb_m};$$

$$kU = \frac{A_1(1-c_{fu})Q}{A_1^2(1-c_{fu}) + a_mb_m}.$$

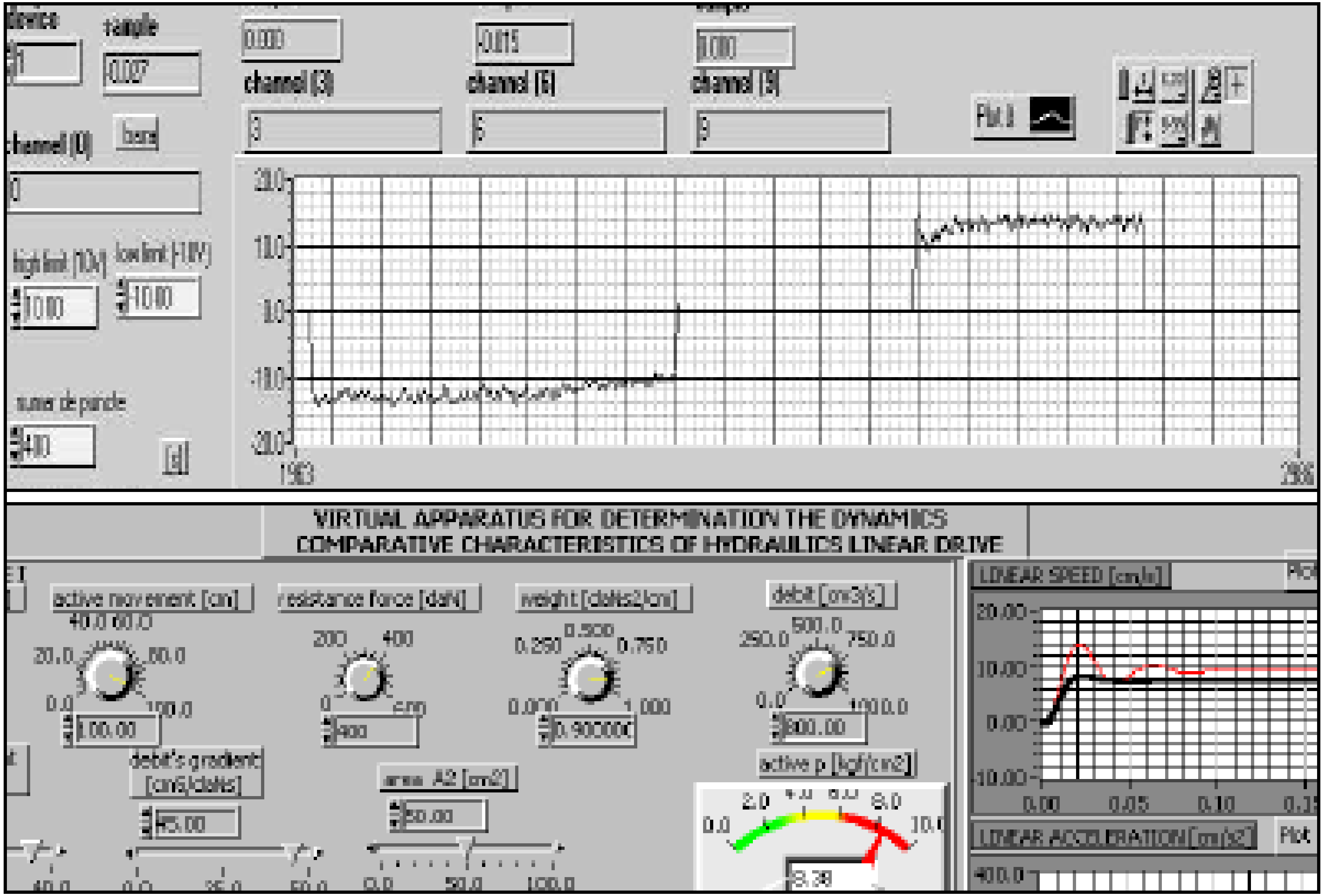


Fig.1: The front panel of the acquisition and theoretical virtual LabVIEW instruments: validation of the LHM mathematical model

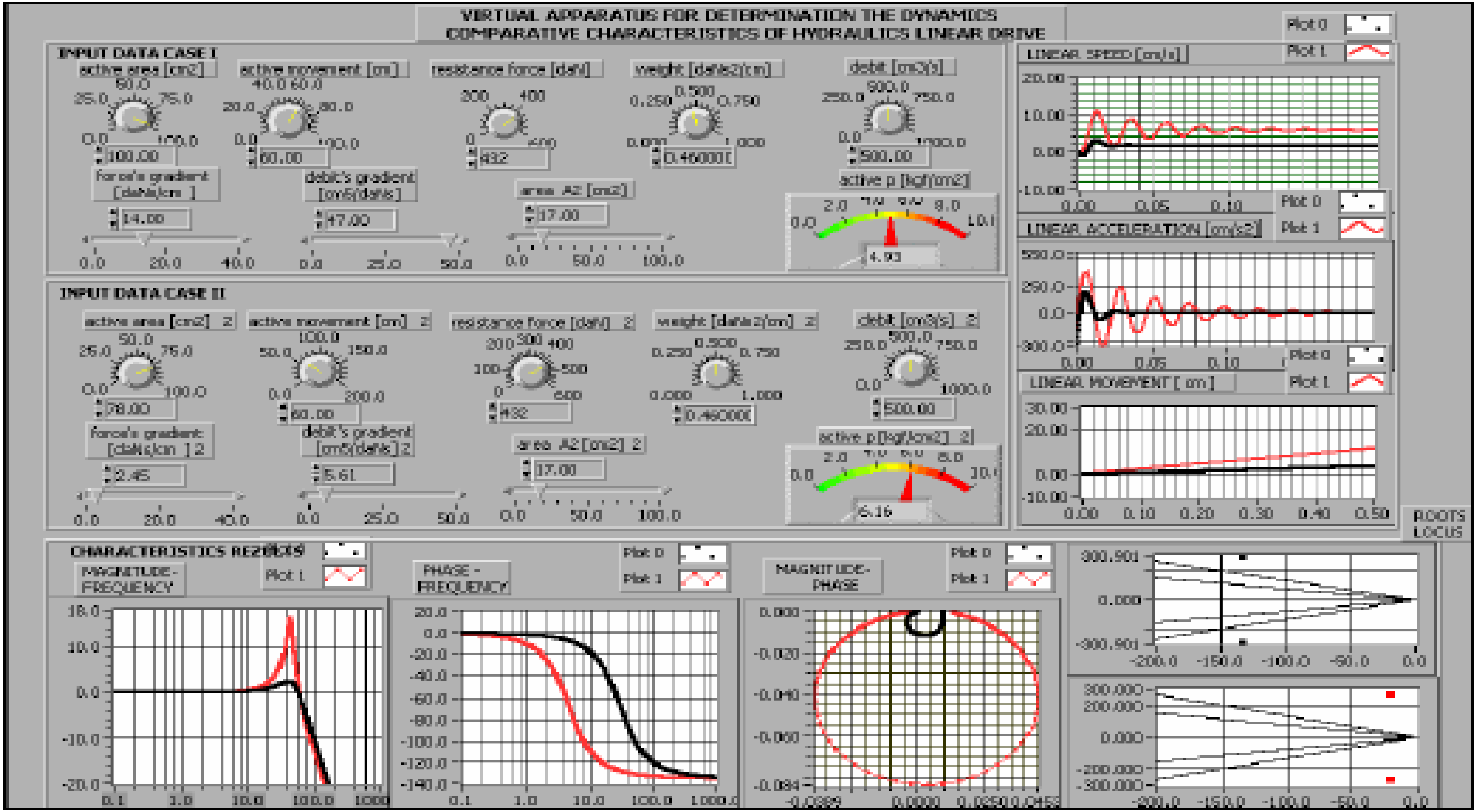
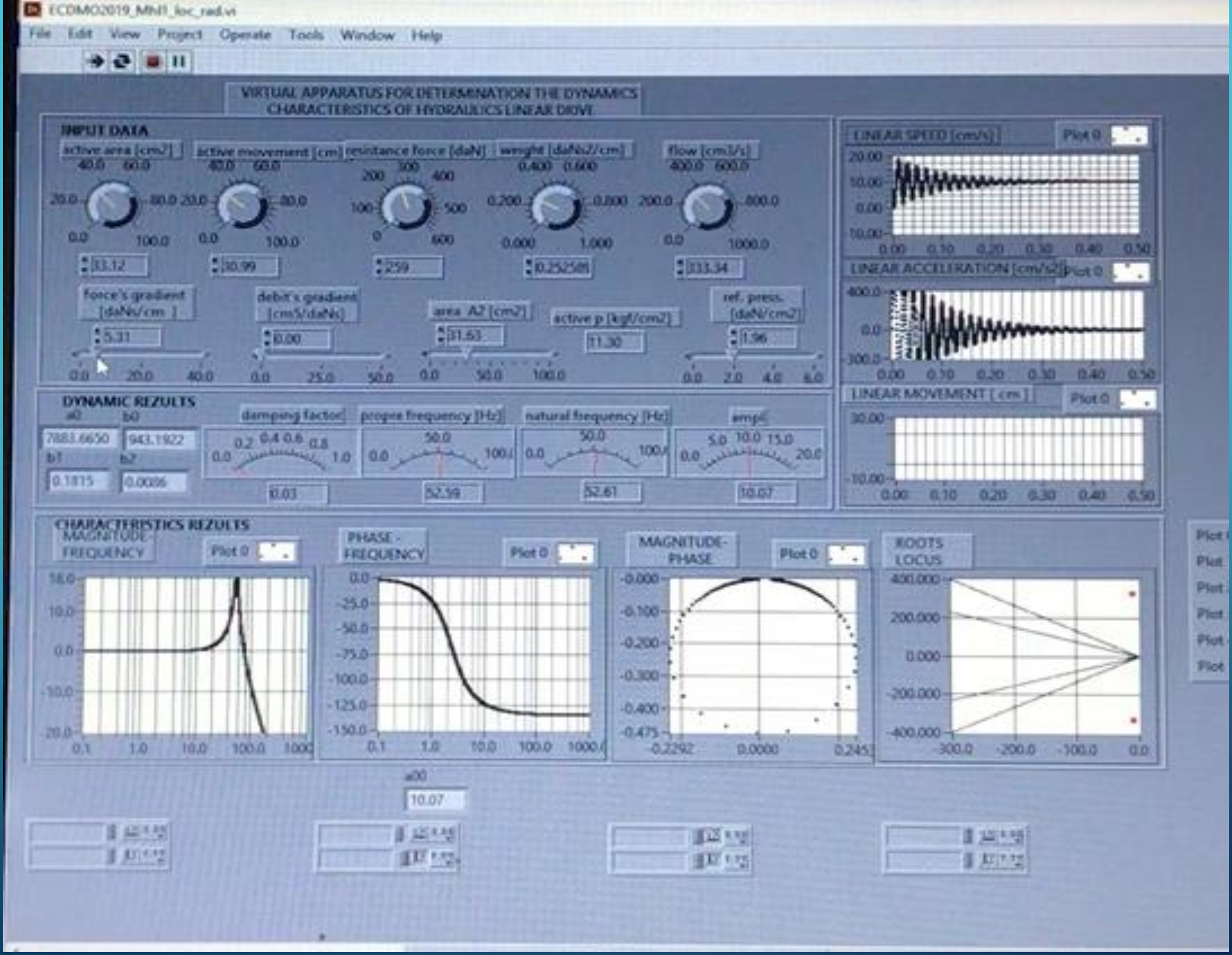


Fig.3: Front panel of the virtual LabVIEW LHM instrument for the comparative analyze, when was been changed the active area, A_1



ECDIMO2019_MN1_loc_rad.vi
File Edit View Project Operate Tools Window Help
VIRTUAL APPARATUS FOR DETERMINATION THE DYNAMICS CHARACTERISTICS OF HYDRAULICS LINEAR DRIVE
INPUT DATA
active area [cm2]
active movement [cm]
resistance force [daN]
weight [daNs2/cm]
flow [cm3/s]
force's gradient [daNs/cm]
debit's gradient [cm5/daNs]
area A2 [cm2]
active p [kgf/cm2]
ref. press. [daN/cm2]
DYNAMIC REZULTS
damping factor
proper frequency [Hz]
natural frequency [Hz]
ampl
LINEAR SPEED [cm/s]
LINEAR ACCELERATION [cm/s2]
LINEAR MOVEMENT [cm]
CHARACTERISTICS REZULTS
MAGNITUDE-FREQUENCY
PHASE-FREQUENCY
MAGNITUDE-PHASE
ROOTS LOCUS

3. Neural Network modelling

- In many applications where the extreme precision is needed, like: trajectory control, guidance, accuracy of taking the image, accuracy to talk or to hear, will be necessary to apply the neural networks. A first wave of interest in neural networks was introduction of simplified neurons by McCulloch and Pitts in 1943 [1]. There neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks.
- Many other applications of the neural networks have trying to developing this field, most notably were Teuvo Kohonen, Stephen Grossberg, James Anderson and Kunihiko Fukushima [5].
- The training of the neural network is made in iterative manner for each iteration (Widrow and Hoff) [3], or by adjusting all neurons weights and biases by calculus of the difference between the target output and the network output, Least Mean Squares (LMS) algorithm. To ensure stable learning, the learning rate must be less than the reciprocal of the largest eigenvalue of the correlation matrix pp^T of the input vector [2, 7]. If the learning rate is set too high, the algorithm can oscillate and become unstable. If the learning rate is too small, the algorithm takes too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

- Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called “squashing” functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. This causes a problem when you use steepest descent to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, cause small terms discussed up to this point, a learning rate is used to determine the length of the weight update.
- Another version of the conjugate gradient algorithm was proposed by Polak and Ribiére [2, 3, 6]. Quasi-Newton Algorithms BFGS, Algorithm Newton’s method is an alternative to the conjugate gradient methods for fast optimization. The basic step of Newton’s method is where is the Hessian matrix (second derivatives) of the performance index at the current values of the weights and biases. Newton’s method often converges faster than conjugate gradient methods. Unfortunately, it is complex and expensive to compute the Hessian matrix for feed forward neural networks.
- Levenberg-Marquardt method, like the quasi-Newton methods, was designed to approach the second-order training speed without having to compute the Hessian matrix [6, 7, 8, 9].

WHAT WAS PROPOSED IN THE ASSISTED RESEARCH OF THE PROPER NEURAL NETWORK

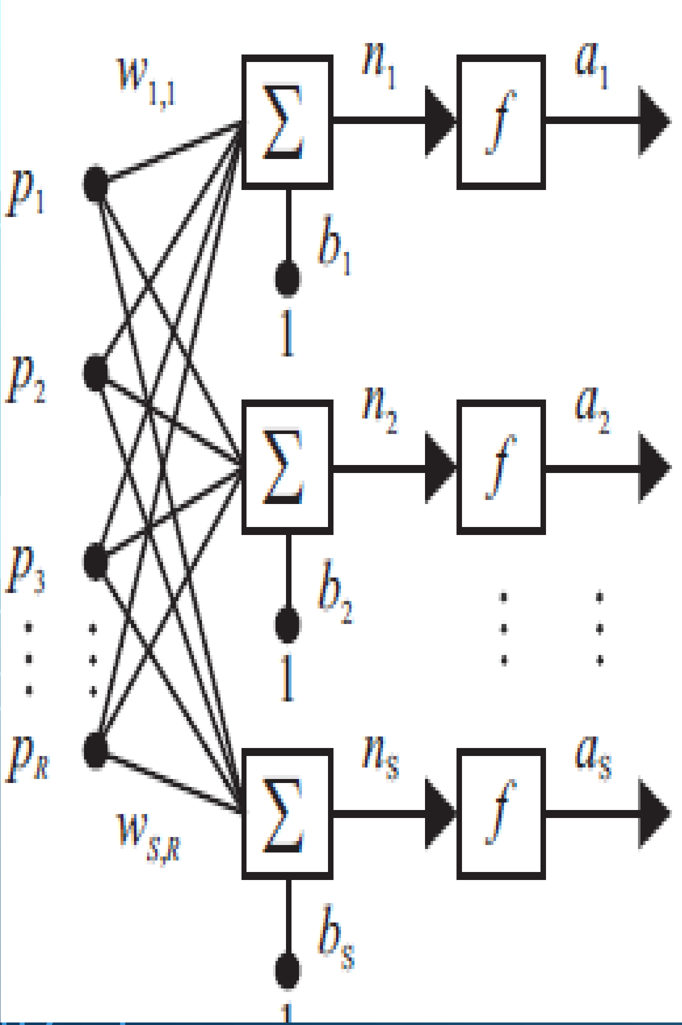
- Design and simulate the known important sensitive functions- for each of them were designed the proper LabVIEW *VI*-s;
- Design and simulate the simple different neurons with LabVIEW;
- Design and simulate the input network (for different sensitive functions with only one layer);
- Design and simulate the simple neural network with 2 or 3 different layers;
- Design and simulate the more known complex neural network;
- Critical analyze of the actual stage of the designed neural networks and proposed new one;
- Assisted research of the new proposed neural network type- Bipolar Sigmoid Hiperbolic Tangent Neuronal Network with Time Delay and Recurrent Links (BSHTNN(TDRL));
- Apply the new NN in one intelligent system to solve the inverse kinematics in Aerospace orientation and optimize the vibration spectrum answer with rheological damper.

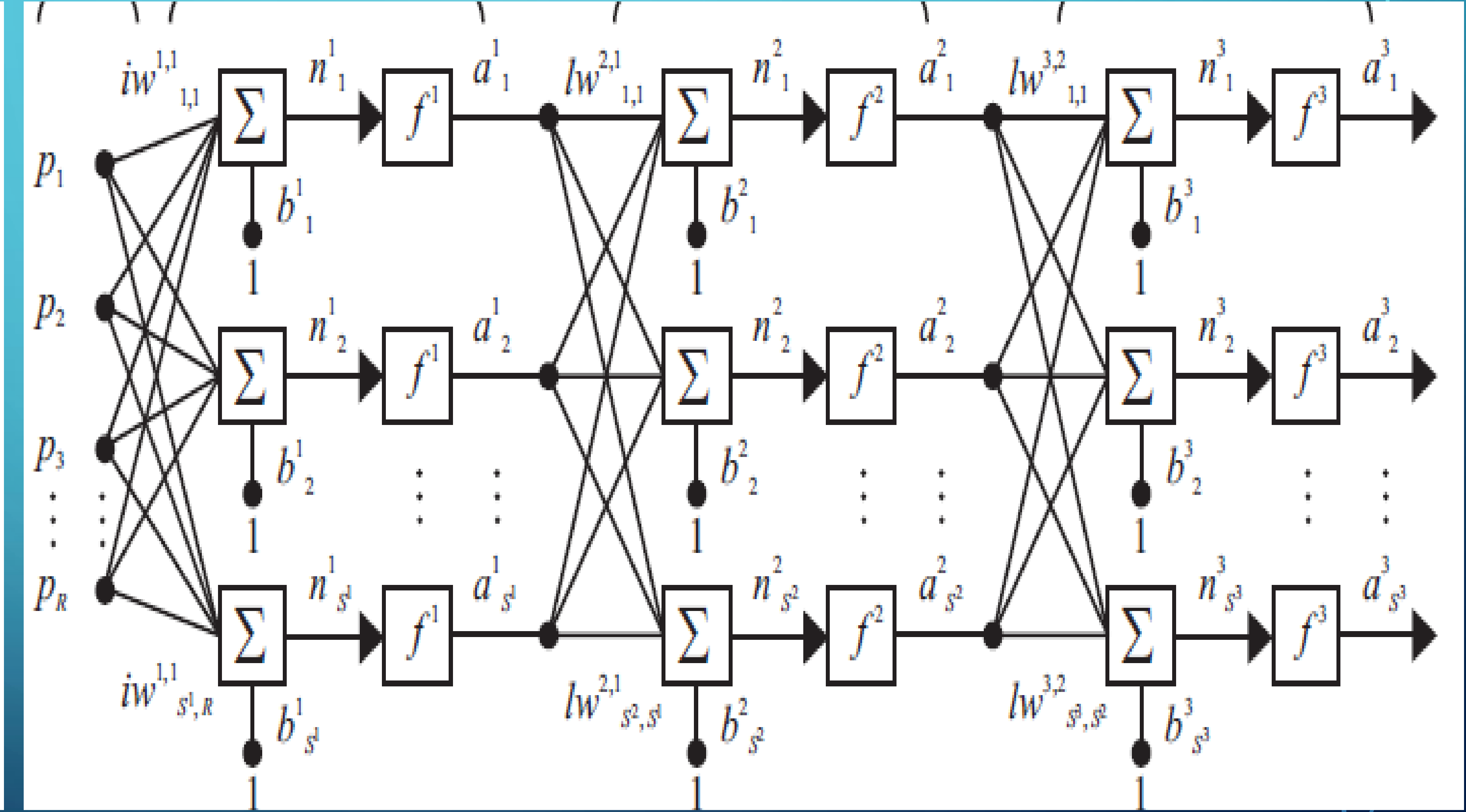
WHAT WILL BE PRESENTED IN THIS RESEARCH AND WHAT ARE THE NOVELTY OF THE RESEARCH

- Establish the proper mathematical model and design the Virtual Instrumentation LabVIEW, *VI*-s, for more known and very used Neural Network;
- Assisted analyze of the more known complex dynamic neural network and establish what are the deficient of them;
- Propose one new Neural Network- Bipolar Sigmoid Hyperbolic Tangent Neural Network with Time Delay and Recurrent Links (BSHTNN(TDRL)) and assisted research of him in parametrized form:
  - with different recurrent links;
  - with different position of the time delay and with different step of the delay;
  - with different number of neurons in each layer;
  - with different associated form of the recurrent links and time delay;

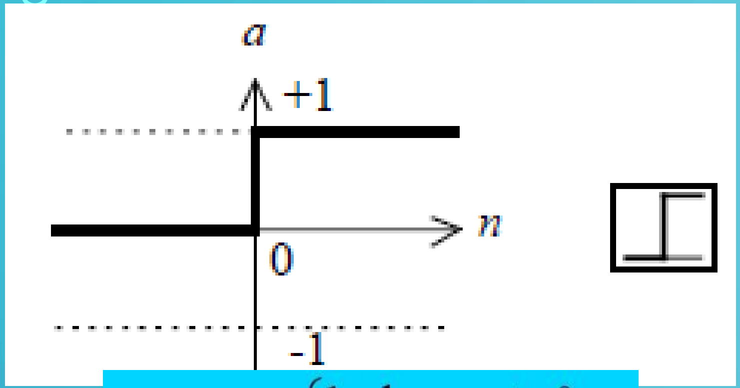
Establish the final form of the proposed Neural Network and apply him in solving the IK and change the Fourier spectrum in one intelligent optimizing damper system.

COMPONENTS OF THE NEURAL NETWORK



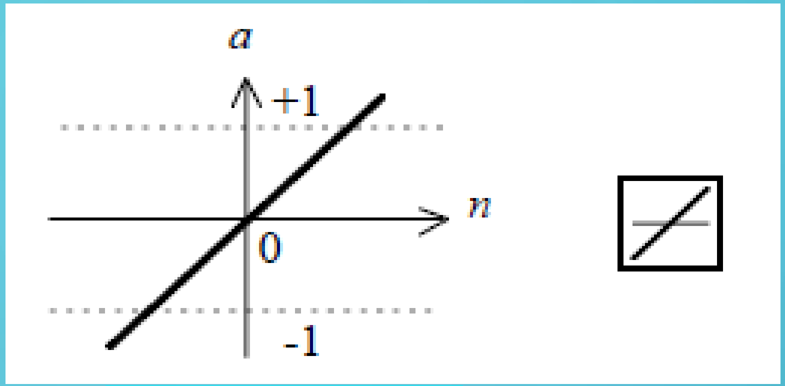


SOME OF THE SENSITIVE USUAL FUNCTIONS



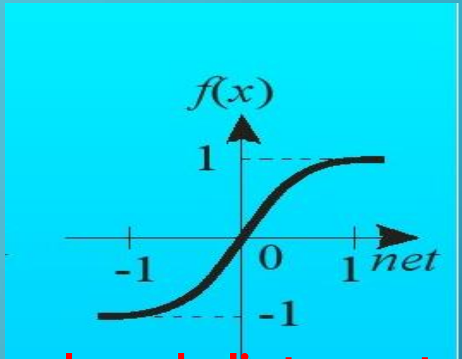
$$f(x)=\begin{cases}1 & \text{daca } x\geq 0\\ 0 & \text{daca } x<0\end{cases}$$

step



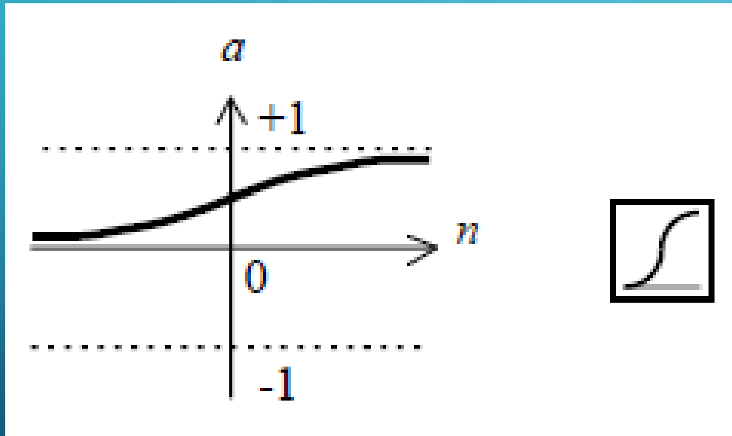
$$f(x)=ax+b$$

ramp



hyperbolic tangent

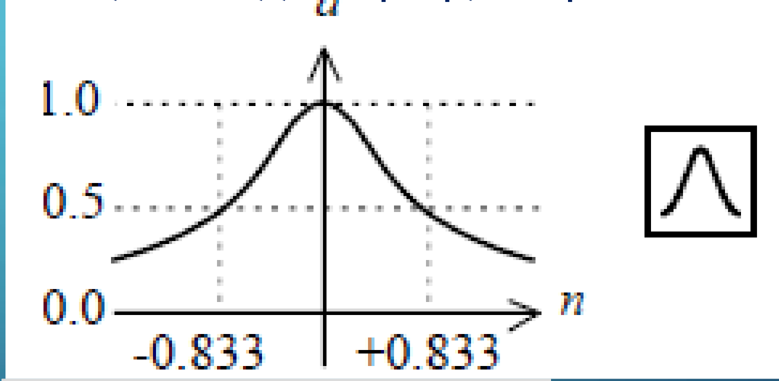
$$f(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}=\tanh(x)$$



$$f(x)=\frac{1}{1+e^{-x}}$$

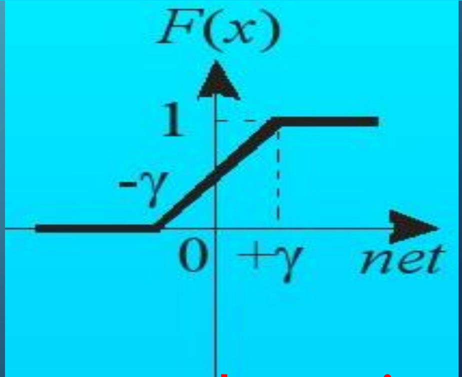
sigmoid

, slinear, , ramptep, ramp



unctii de activare

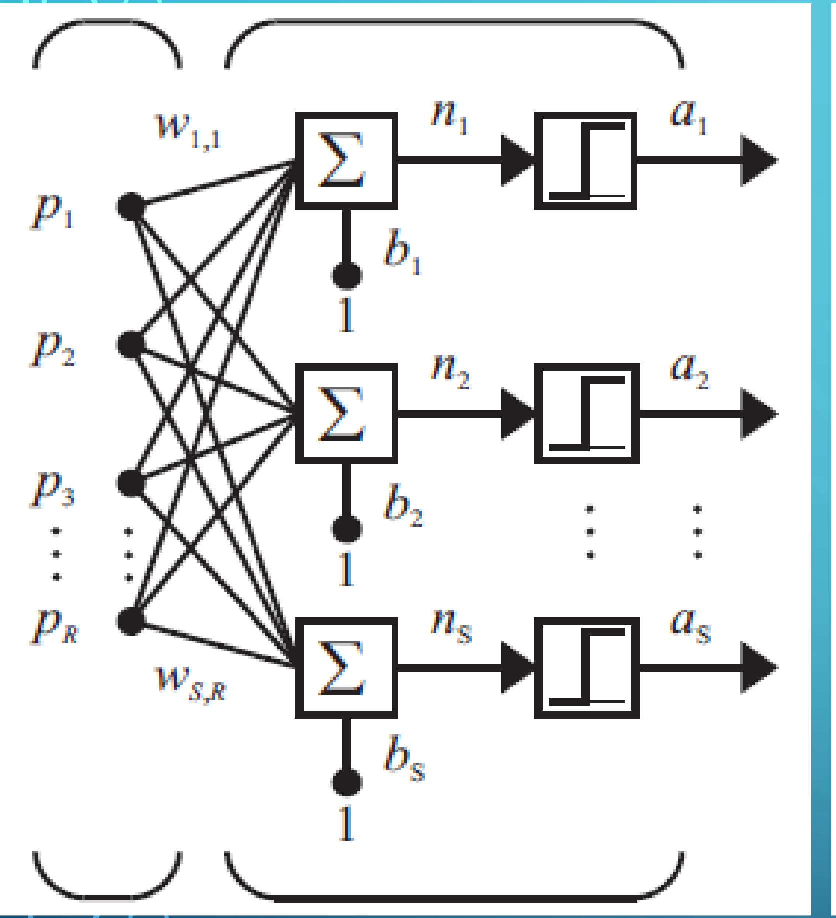
gaussian

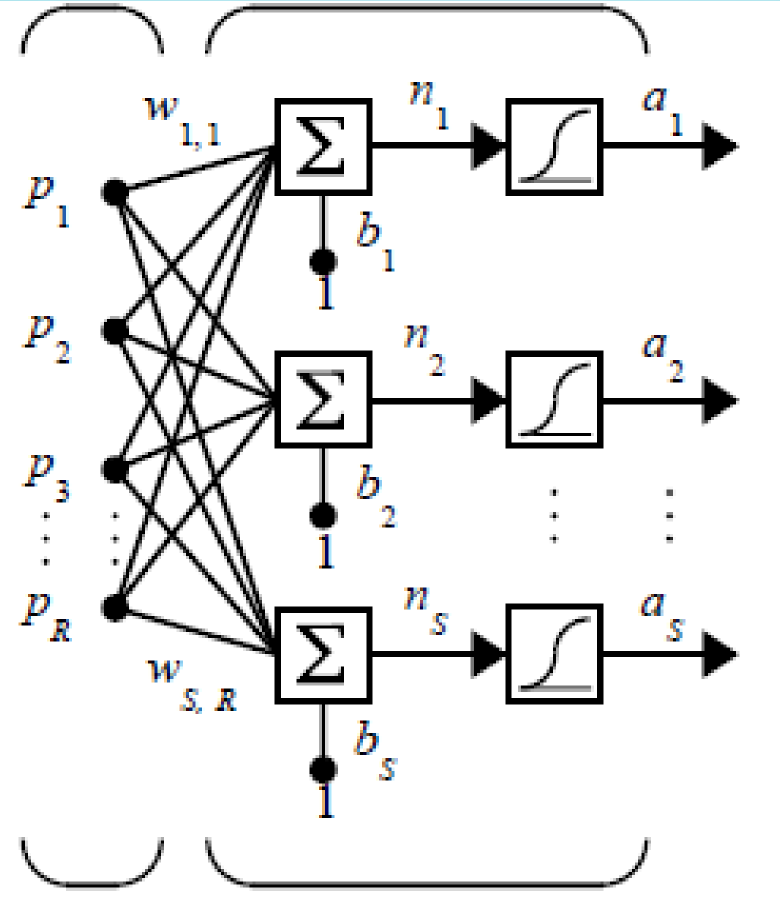


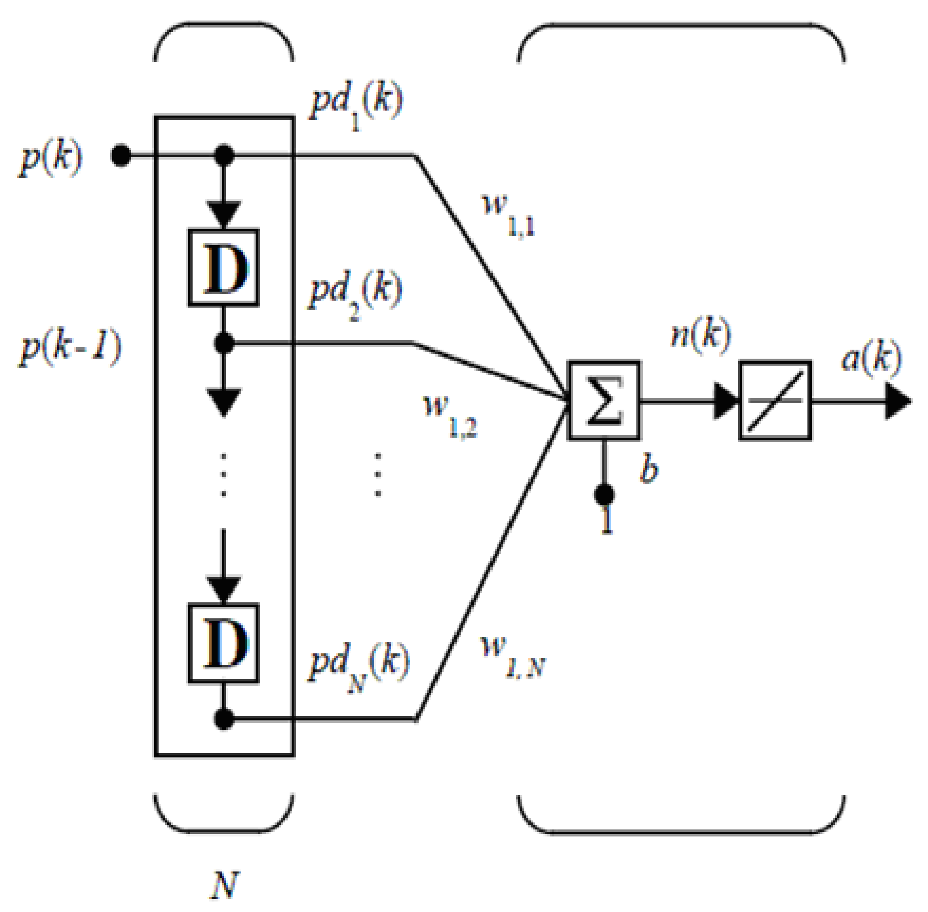
ramp and saturation

$$f(x)=\begin{cases}0 & \text{daca } x<-\gamma\\ \frac{1}{2\gamma}x+\frac{1}{2} & \text{daca } x\in[-\gamma,\gamma]\\ 1 & \text{daca } x>\gamma\end{cases}$$

DIFFERENT EXAMPLES OF SIMPLE NEURAL NETWORKS





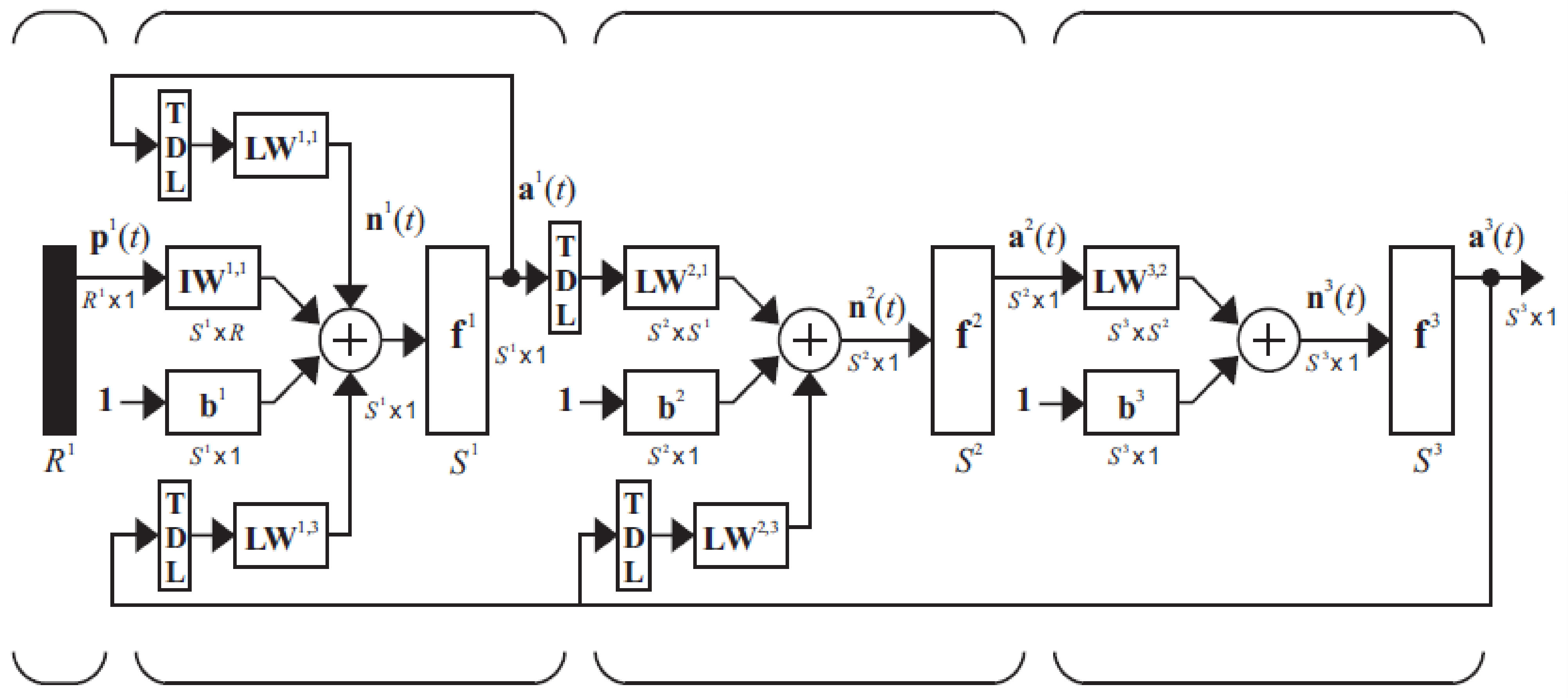


MATHEMATICAL MODELLING OF THE MORE IMPORTANT NN

$$a^1(t) = f^1[IW^{1,1}p^1(t) + b^1 + LW^{1,1}a^1(t-1) + LW^{1,3}a^3(t-1)]$$

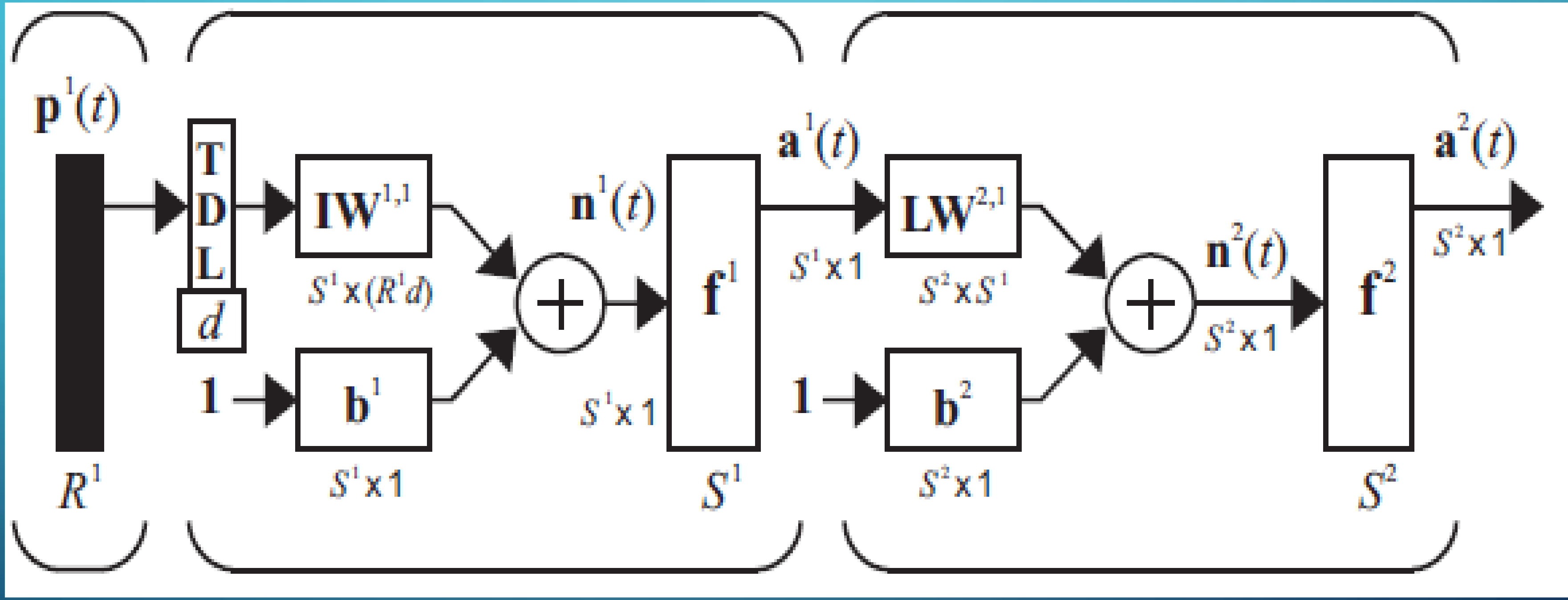
$$a^2(t) = f^2[LW^{2,1}a^1(t-1) + b^2 + LW^{2,3}a^3(t-1)]$$

$$a^3(t) = f^3[LW^{3,2}a^2 + b^3]$$



Layered Digital Dynamic Network (LDDN)

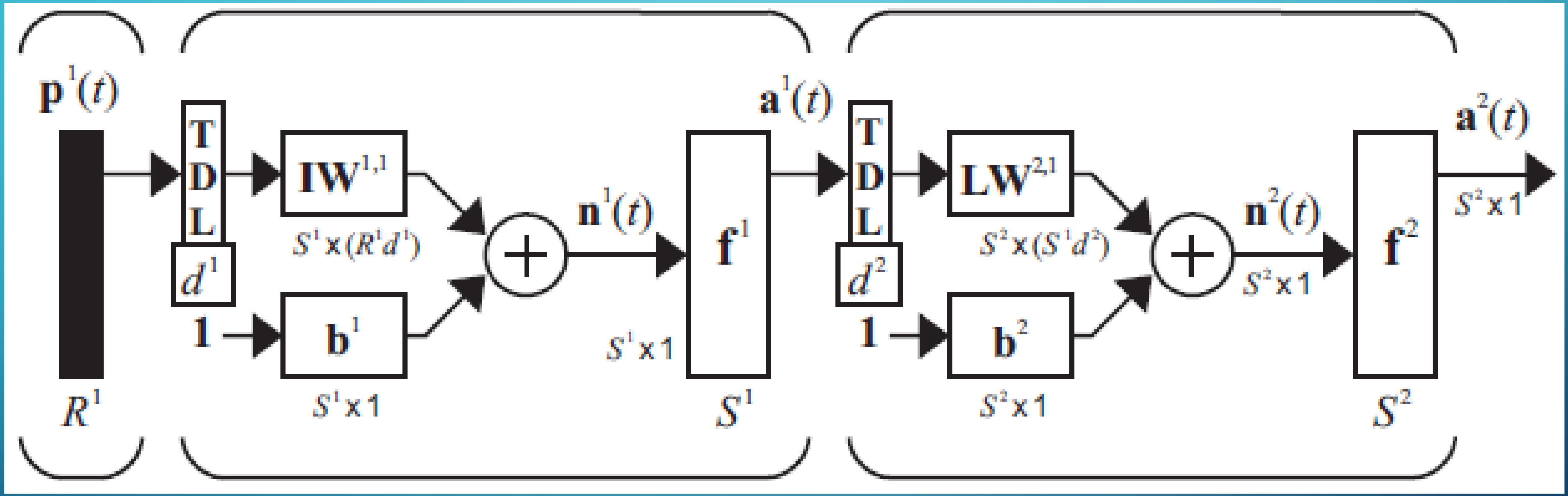
$$a^1(t) = f^1[IW^{1,1} p(t-d+1) + b^1]$$
$$a^2(t) = f^2[LW^{2,1} a^1(t) + b^2]$$



Focused Time-Delay Neural Network (FTDNN)

$$a^1(t) = f^1[IW^{1,1} p^1(t - d^1 + 1) + b^1]$$

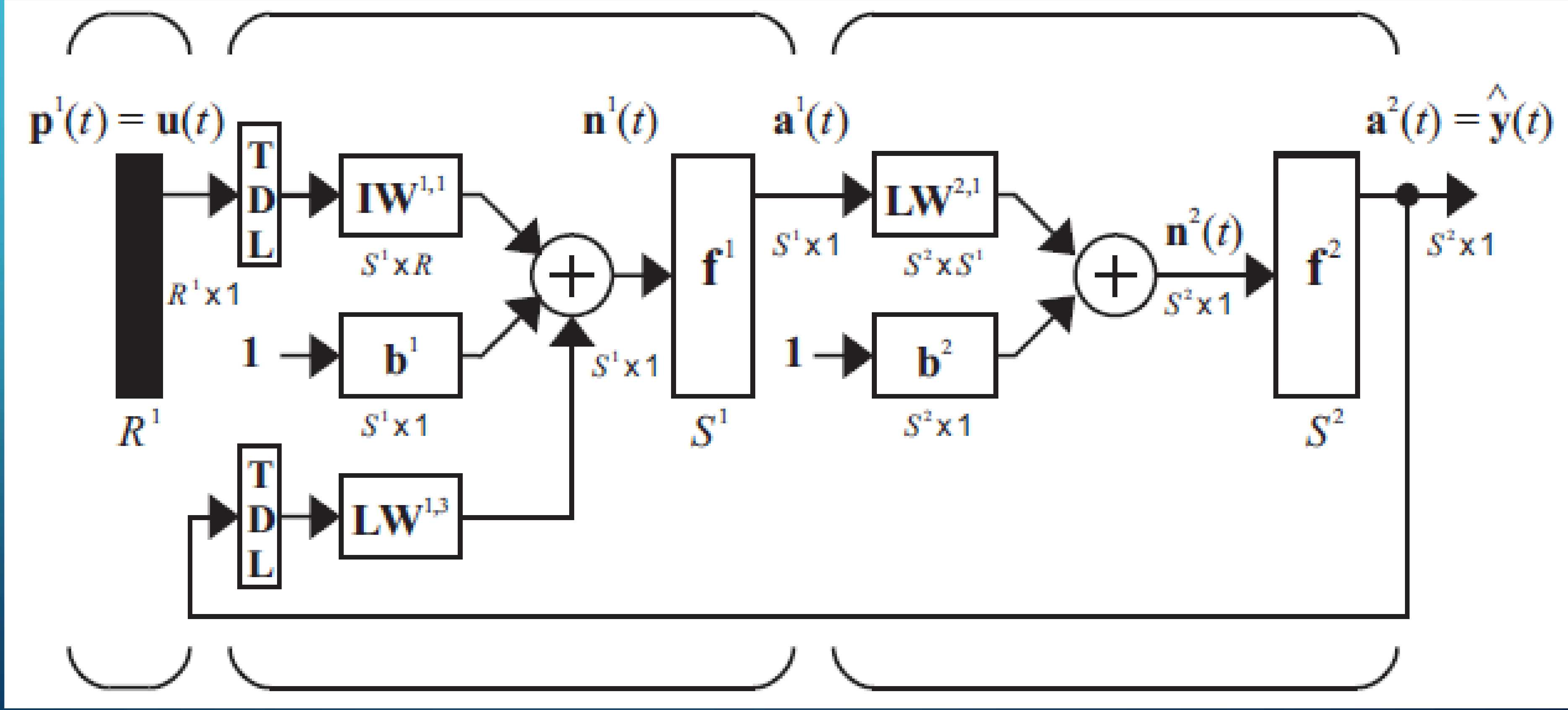
$$a^2(t) = f^2[LW^{2,1} a^1(t - d^2 + 1) + b^2]$$



Distributed Time-Delay Neural Network (DTDNN)

$$a^1(t) = \frac{1}{1+e^{-[IW^{1,1}u(t-1)+b^1+LW^{1,3}a^2(t-1)]}}$$

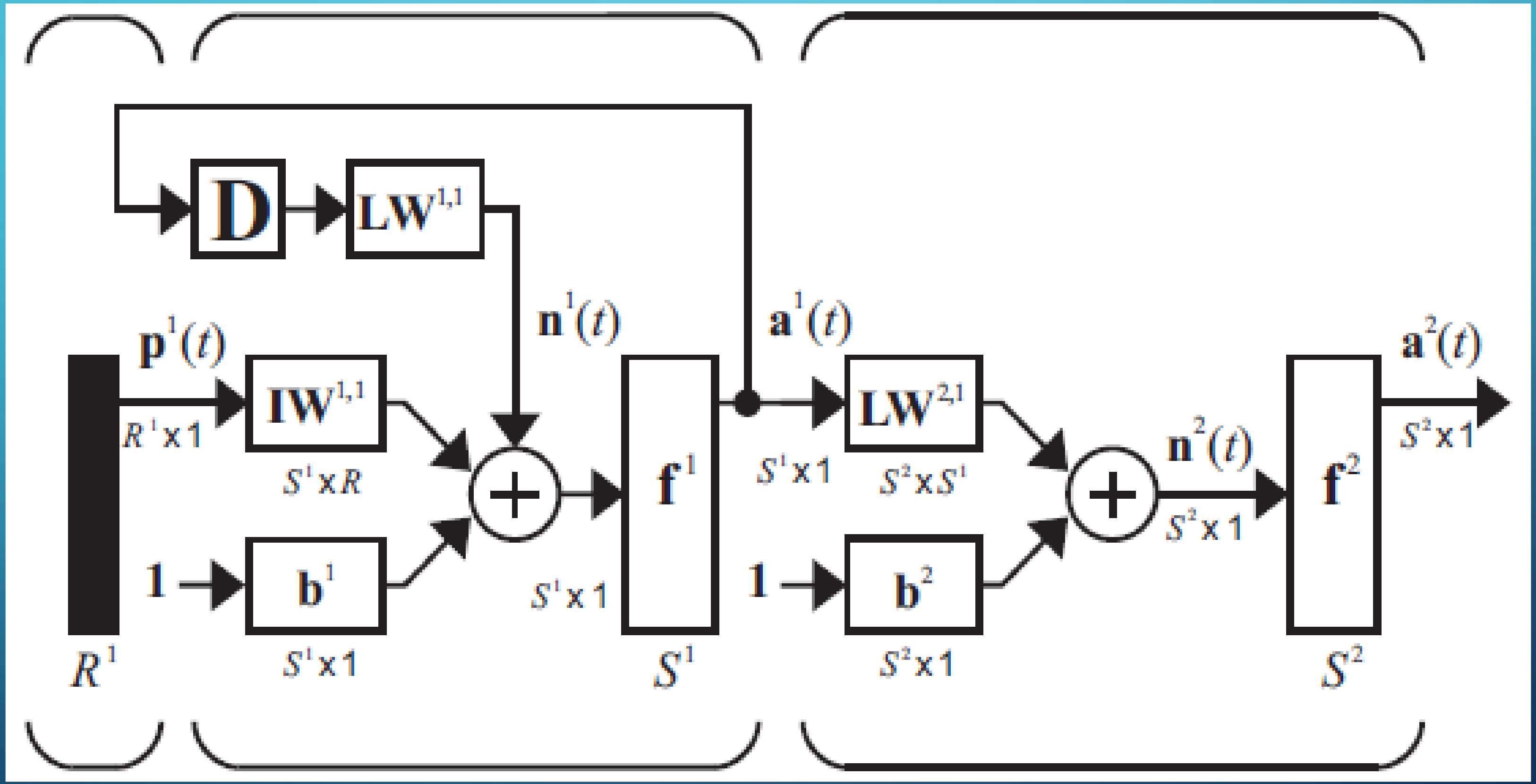
$$a^2(t) = \frac{1}{1+e^{-[LW^{2,1}a^1(t)+b^2]}}$$



Nonlinear antiregresive with exogene inputs NARX

$$a^1(t) = \frac{1}{1 + e^{-[IW^{1,1}p^1(t) + b^1 + LW^{1,2}a^1(t-1)]}}$$

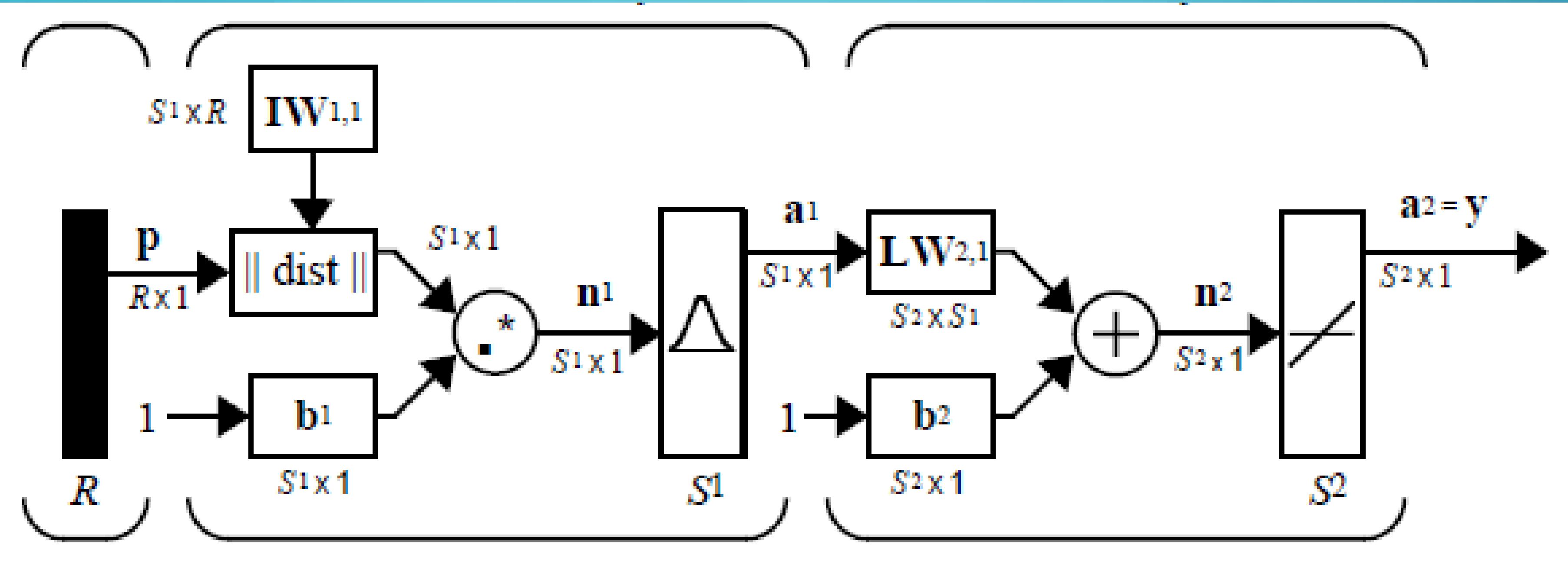
$$a^2(t) = LW^{2,1}a^1(t) + b^2$$



Layer-Recurrent Network (LRN)

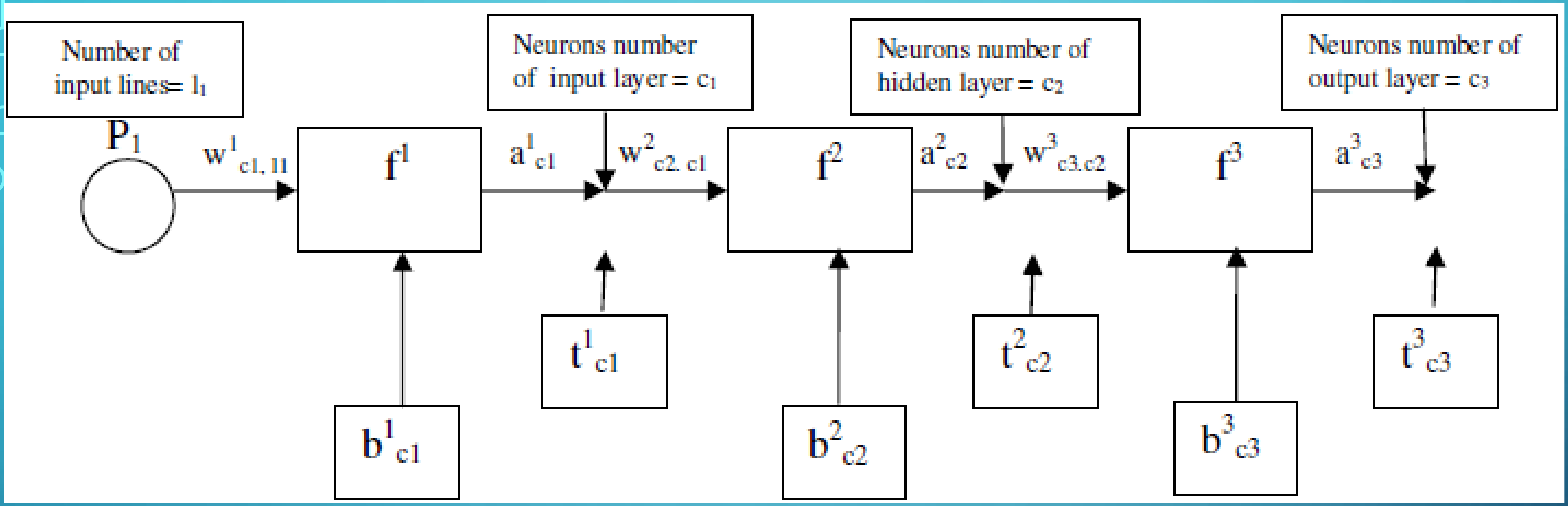
$$a^1 = \begin{cases} 1 \forall \left| p^1 - IW^{1,1} + b^1 \right| \neq 0 \\ 0 \forall \left| p^1 - IW^{1,1} + b^1 \right| = 0 \end{cases}$$

$$a^2 = LW^{2,1} a^1 + b^2$$



Radial base Neural Network

ASSISTED ANALYSE OF THE NN AND PROPER NN



The parameters that were studied with numerical simulation with LabVIEW were:

- the number and the data of the input vector;
- the number of neurons for each of the layers;
- the sensitive functions in each of layers;
- the teaching gain for each layer;
- the sigmoid bipolar gain;
- the target output data and the calculus of the inverse functions of the target in each layer.

THE PROPOSAL OF THE NEW DYNAMIC NEURAL NETWORK

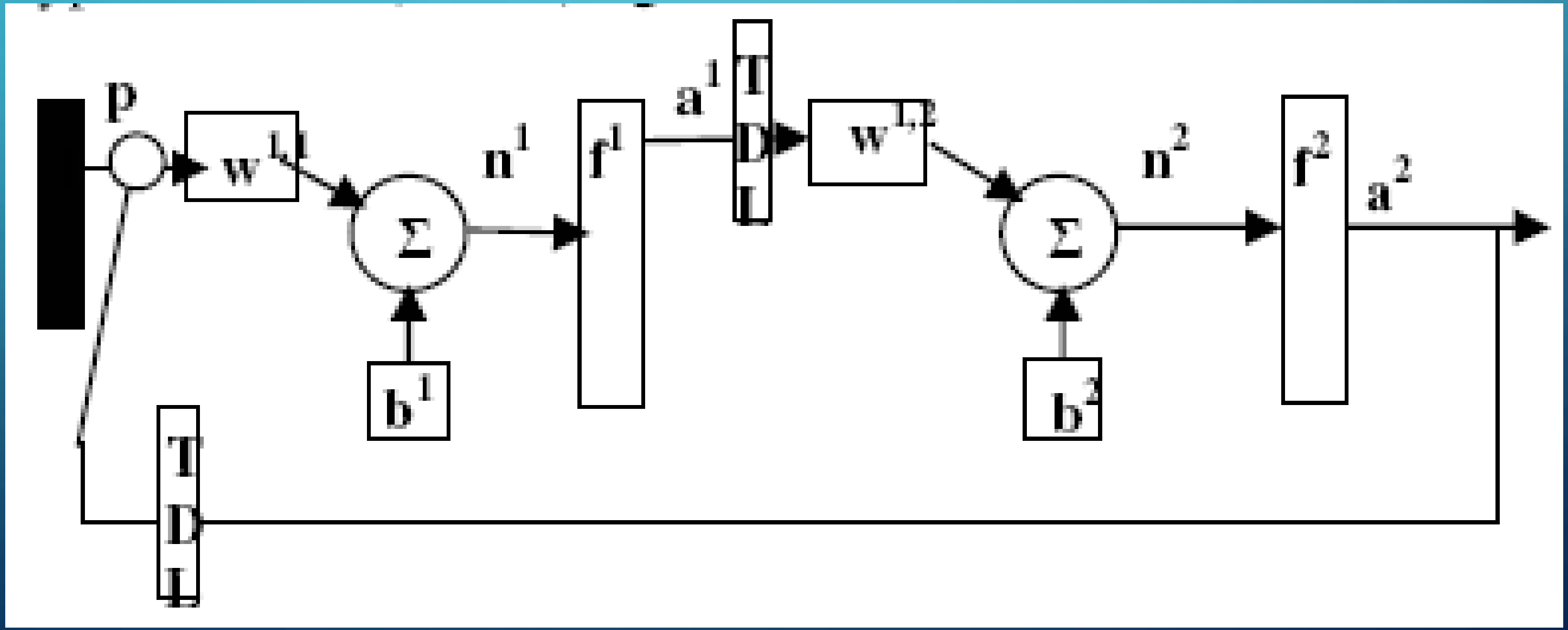
$$n^1(t) = w^{1,1}p^1 + w^{1,2}a^2(t-1) + b^1$$

$$a^1(t) = \frac{a(1-e^{-n^1})}{1+e^{-n^1}}$$

$$n^2(t) = w^{2,1}a^1(t-1) + b^2$$

$$a^2(t) = \frac{a(1-e^{-n^2})}{1+e^{-n^2}}$$

f*[1]** and ***f*[2]** are the sigmoid activation functions, and ***a is the gain of these functions



ANALIZE OF THE RESULTS AFTER ASSISTED SOME VERY USED NEURAL NETWORK AND NEW PROPOSAL

After simulation of the more important neural networks we can remark the following:

- the time delay with **the parameter *d* determines one general** possibility to adjust the parameters;
- it is necessary to respect the essential condition that ***d* must be one odd number comparing with the next layer** to assure the alternative oscillation near the target curve;
- in the network with many time-delay, **the delay parameter d_2 must be odd number comparing with d_1;**
- **the reaction with time-delay** applied with one weight matrix **can determine the instability** of the network solution.

To eliminate some of these deficient of the researched neural network on propose one new structure of network what was analyzed in the paper (BSHTNN(TDRL)).

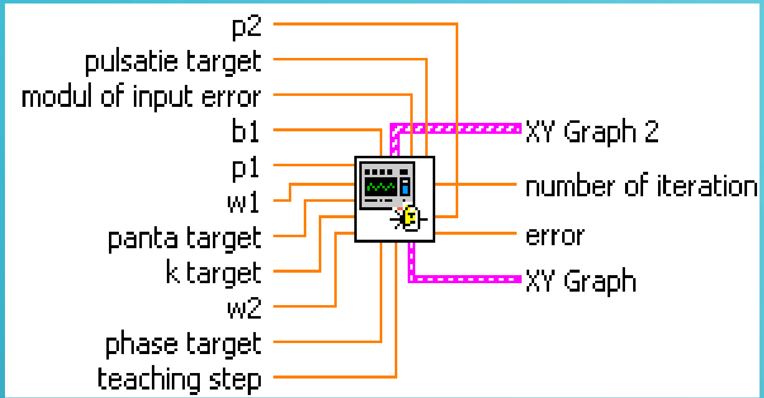
The researched cases of the proper neural network were:

- without time-delay;
- with time- delay after output $a^1(t)$;
- with time- delay after output $a^1(t)$ and $a^2(t)$;
- with time-delay after $a^1(t)$ and one recurrent link between the output and the input vector *with* and *without* weights matrix.

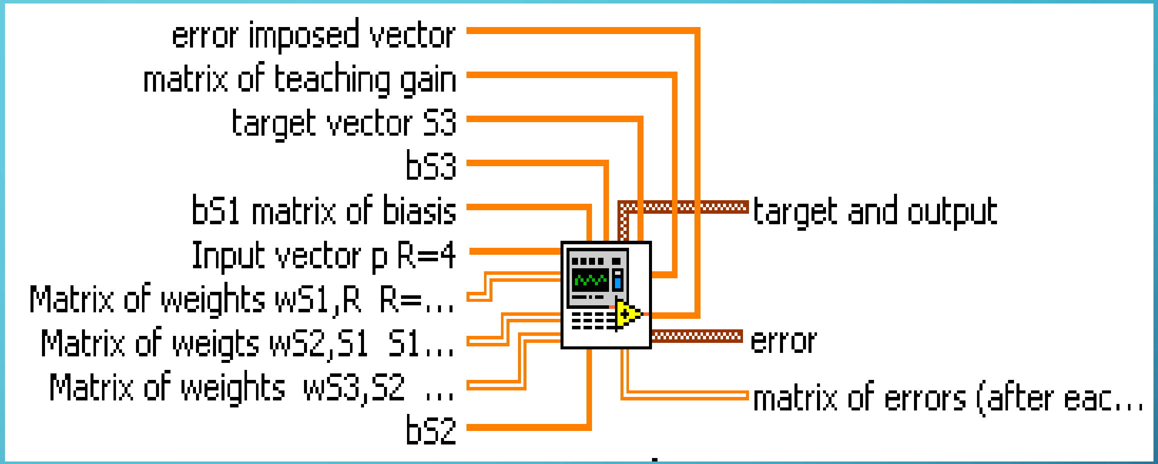
For the simulation was used one proper virtual LabVIEW instrument type BSHTNN(TDRL) (Bipolar Sigmoid Hiperbolic Tangent with Time –Delay and Recurrent Links).

The comparative research was made **respect the same all input, output and parameters data .**

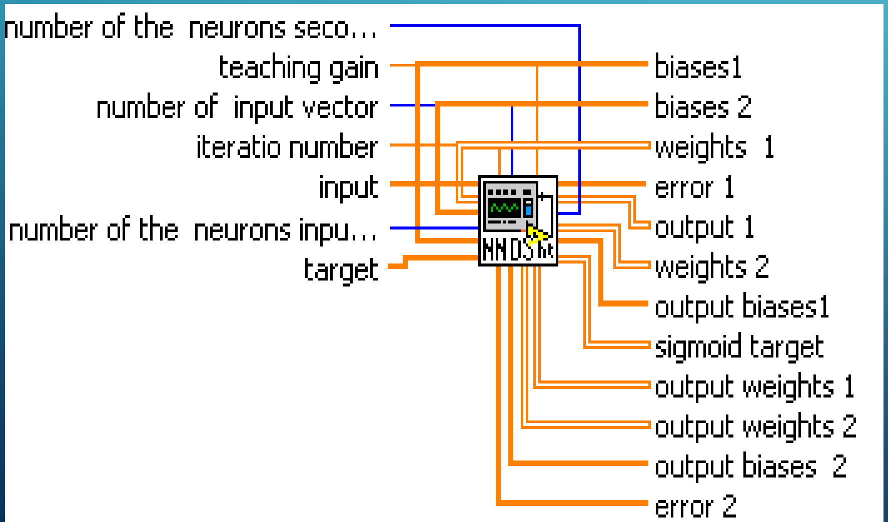
THE PROPER LABVIEW *VI*-s FOR SIMULATION



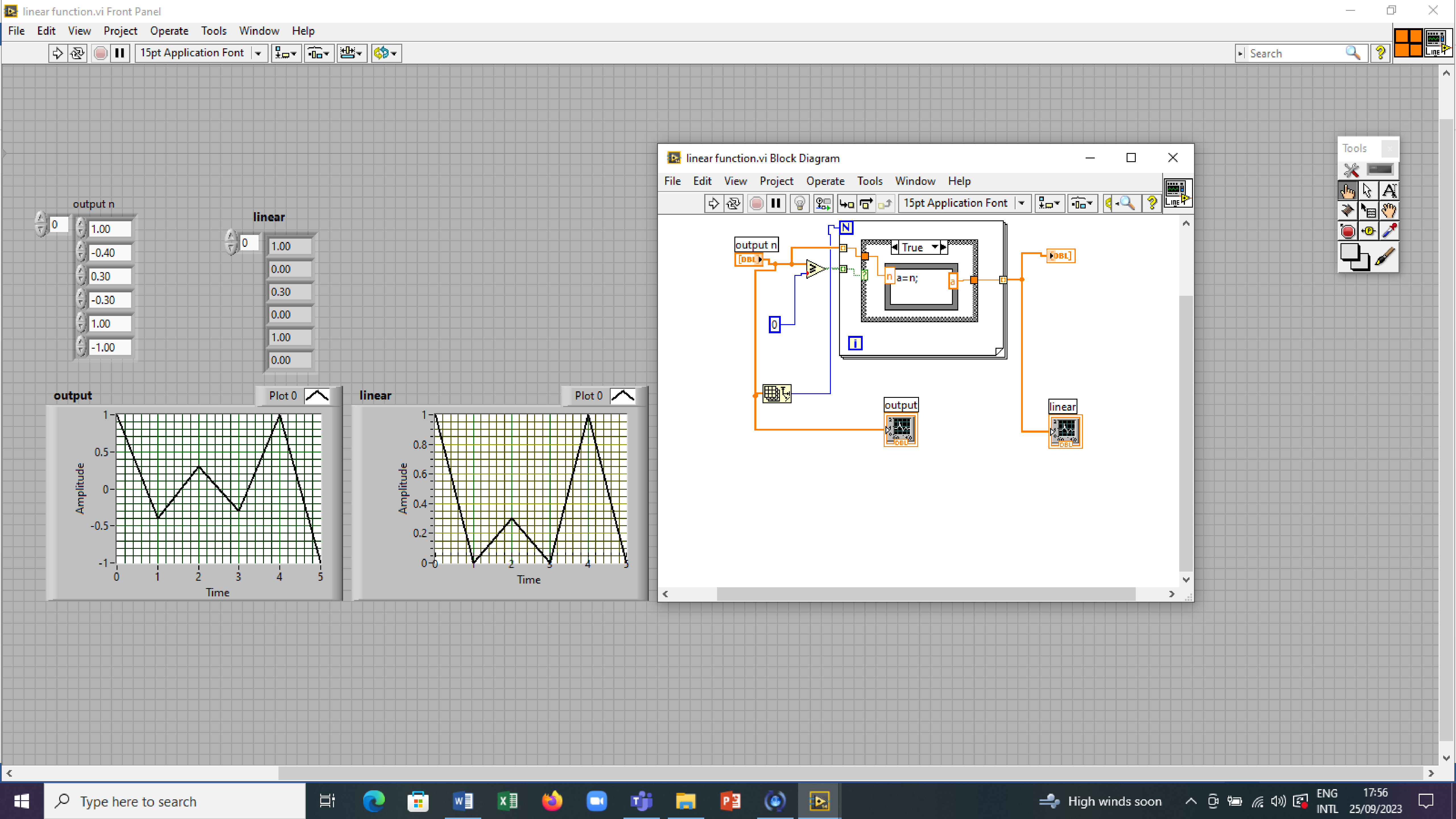
Icon of the VI-s for the assisted theoretical research **of the simple linear neuron**

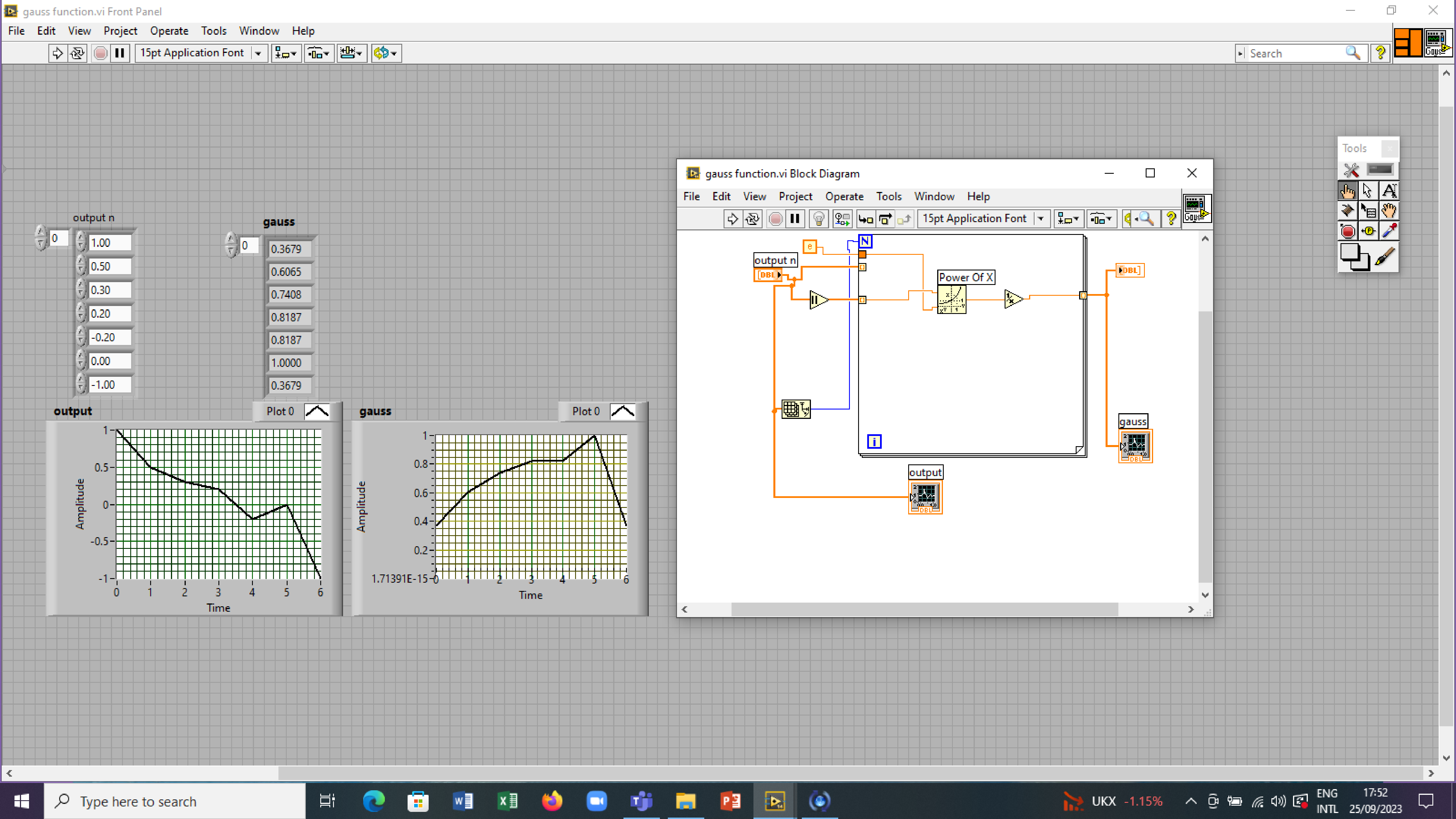


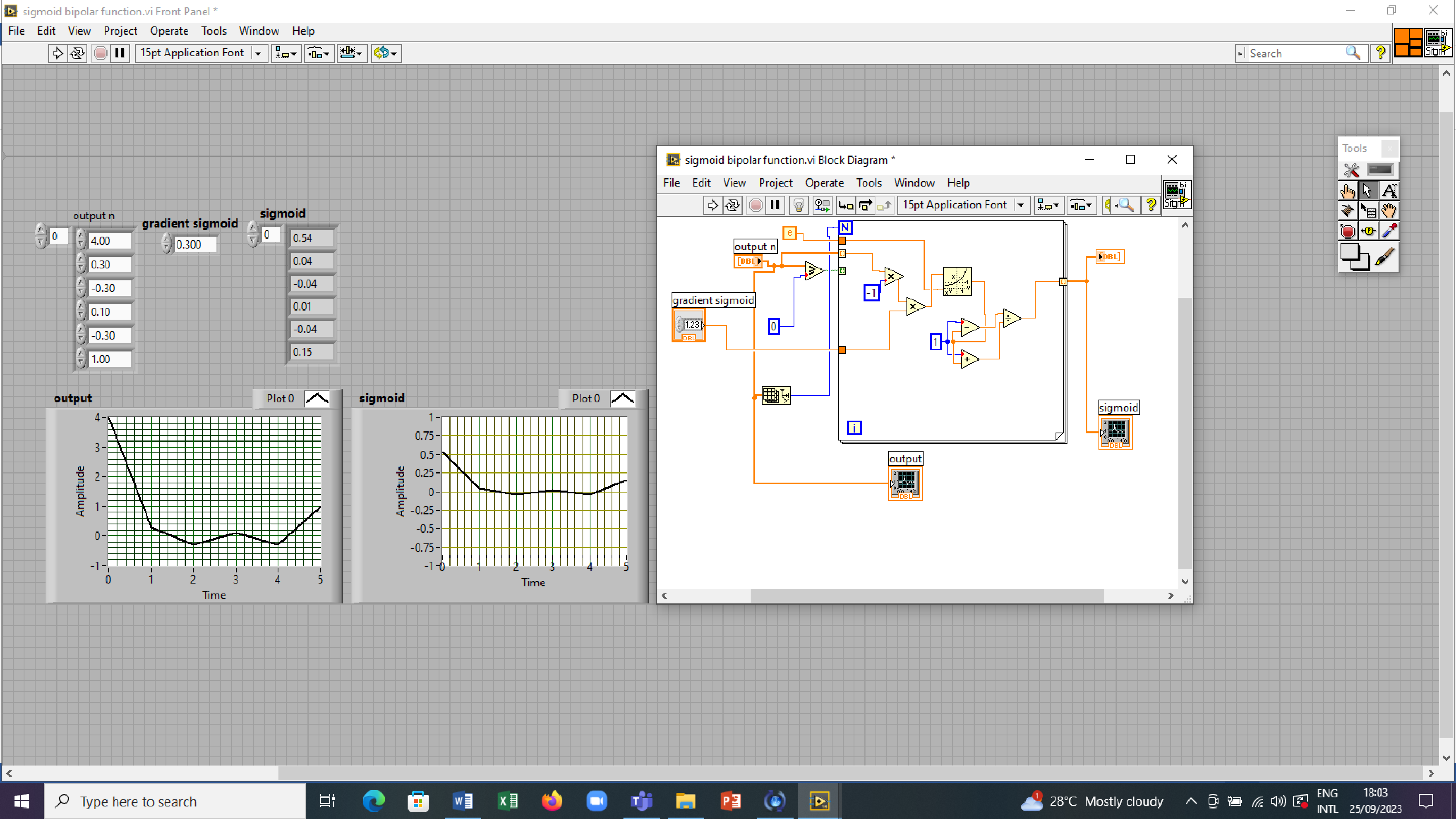
Icon of the VI-s for the assisted theoretical research **of the linear neural network**

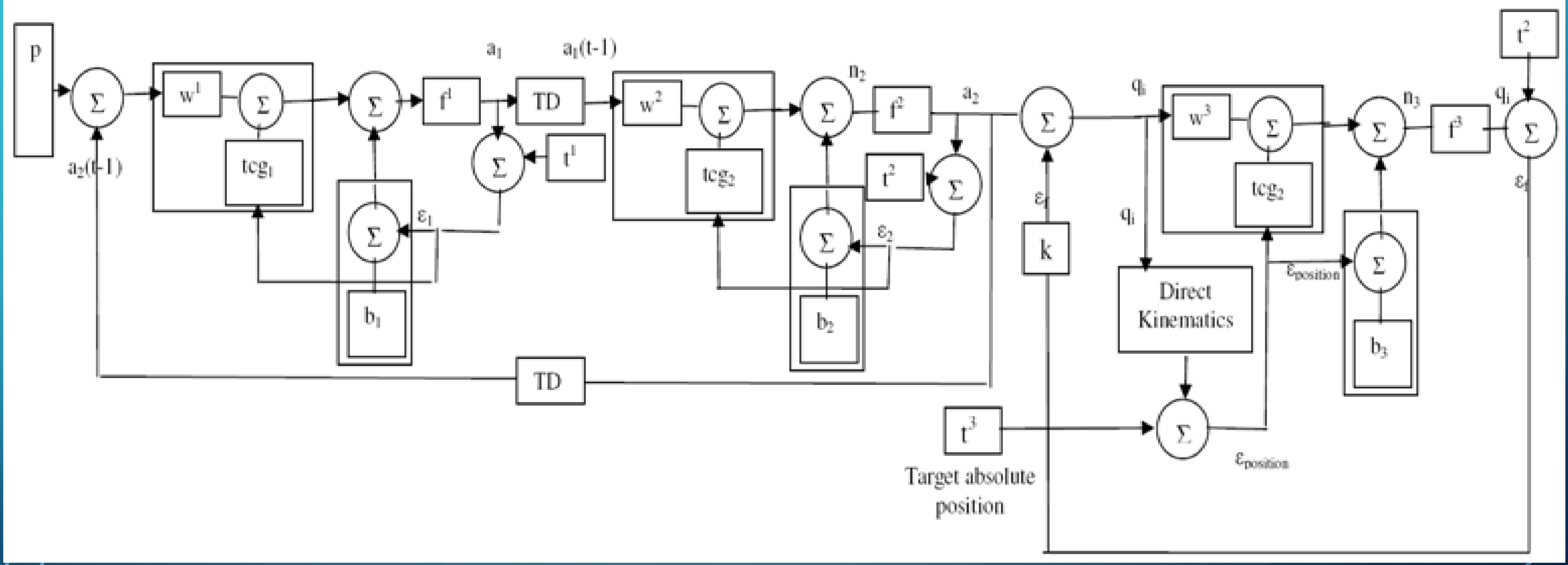


Icons of the virtual LabVIEW instruments for the assisted research of the time-delay and external closed loop **of proper neural network- Bipolar Sigmoid Hiperbolic Tangent Neural Network- BSHTNN**


linear function.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
output n
0
1.00
-0.40
0.30
-0.30
1.00
-1.00
linear
0
1.00
0.00
0.30
0.00
1.00
0.00
output
Plot 0
Amplitude
Time
linear
Plot 0
Amplitude
Time
linear function.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
output n
N
True
a=n;
i
output
linear
Tools
Type here to search
High winds soon
ENG
INTL
17:56
25/09/2023


gauss function.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
output n
0
1.00
0.50
0.30
0.20
-0.20
0.00
-1.00
gauss
0
0.3679
0.6065
0.7408
0.8187
0.8187
1.0000
0.3679
output
Plot 0
Amplitude
Time
gauss
Plot 0
1.71391E-15
gauss function.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
output n
DBL
N
e
Power Of X
i
DBL
gauss
output
Tools
Type here to search
UKX -1.15%
ENG
INTL
17:52
25/09/2023


sigmoid bipolar function.vi Front Panel *
File Edit View Project Operate Tools Window Help
15pt Application Font
output n
gradient sigmoid
sigmoid
0
4.00
0.30
-0.30
0.10
-0.30
1.00
0.300
0.54
0.04
-0.04
0.01
-0.04
0.15
output
Plot 0
Amplitude
Time
sigmoid bipolar function.vi Block Diagram *
output
sigmoid
Tools
Type here to search
28°C Mostly cloudy
ENG
INTL
18:03
25/09/2023



p
Σ
w1
tcg1
a2(t-1)
f1
a1
TD
a1(t-1)
t1
ε1
b1
w2
tcg2
n2
f2
a2
t2
ε2
b2
TD
qi
k
εi
w3
tcg2
n3
f3
t2
Direct Kinematics
εposition
b3
t3
Target absolute position

4. EXAMPLES OF THE MULTIPOL TRANSFER FUNCTIONS ANALYSE

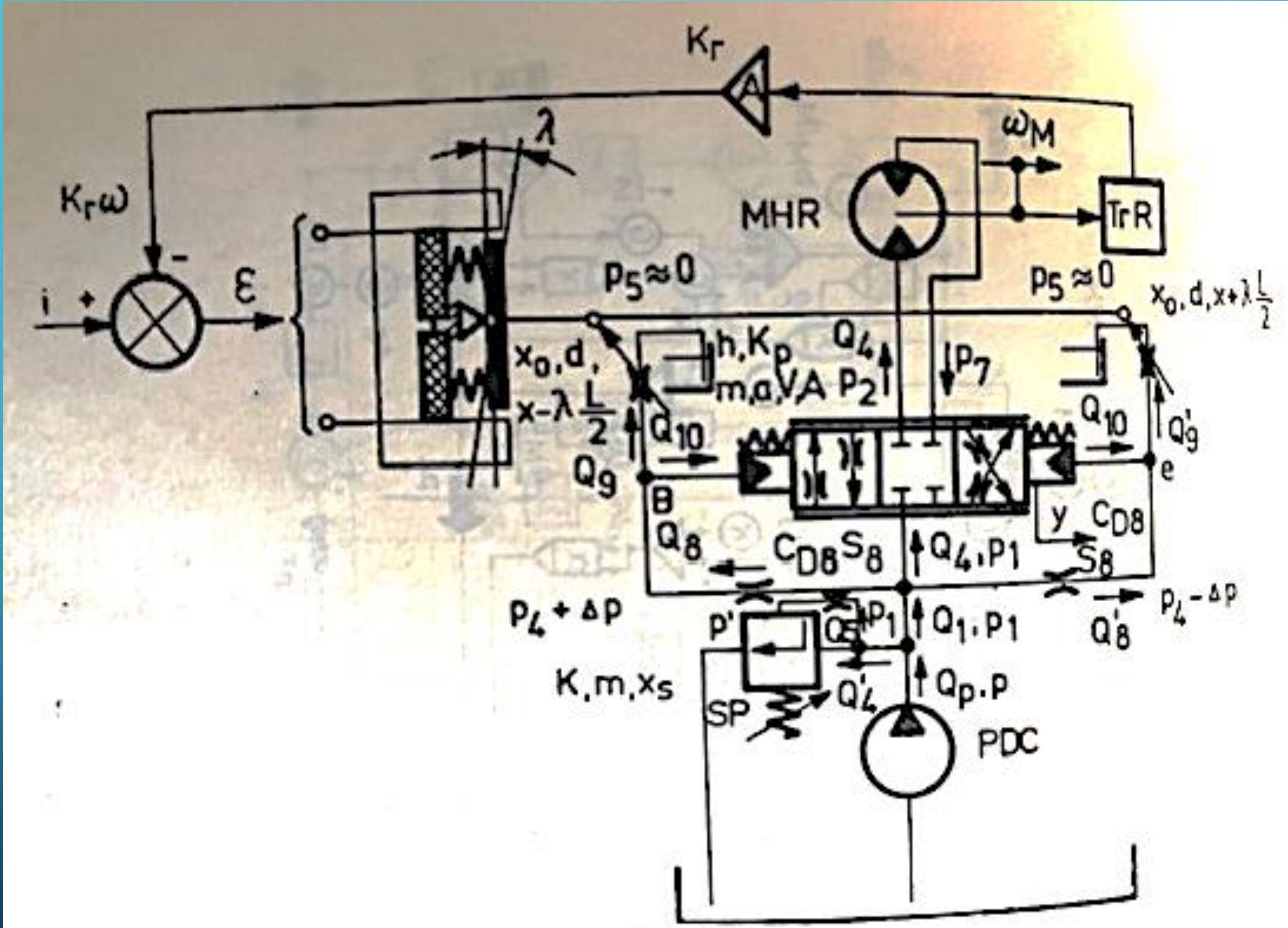
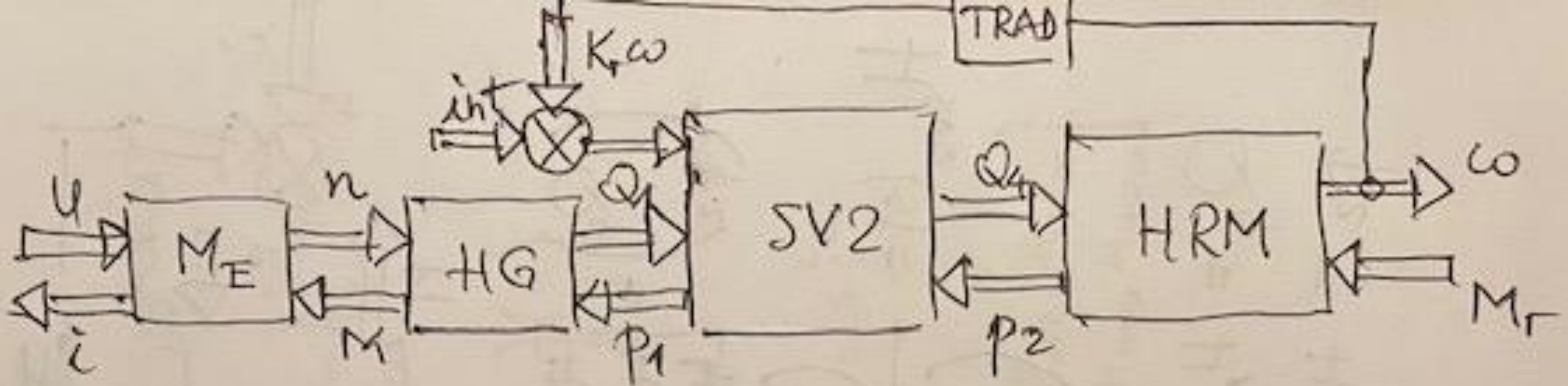


Fig. 4.7



M_E – electrical motor
HG – hydraulic generator
SV2 – servo-valve
HRM – hydraulic rotating motor

$$H(s)_{servo} = \frac{\begin{pmatrix} \omega \\ M_r \end{pmatrix}}{\begin{pmatrix} int \\ u \\ i \end{pmatrix}} = \begin{bmatrix} H_{servo11} & H_{servo12} & H_{servo13} \\ H_{servo21} & H_{servo22} & H_{servo23} \end{bmatrix}$$

$$\begin{pmatrix} \omega \\ M_r \end{pmatrix} = \begin{bmatrix} H_{servo11} & H_{servo12} & H_{servo13} \\ H_{servo21} & H_{servo22} & H_{servo23} \end{bmatrix} \begin{pmatrix} int \\ u \\ i \end{pmatrix}$$

int – command input
u – electrical tension
i – electrical current

Supapă de presiune:

$$Q_1 = Q_p - Q'_4 - Q'_5;$$

$$Q'_4 = C_{DSP}\pi D x_s \sqrt{2/\rho}\,\sqrt{p};$$

$$Q'_5 = C_D S \sqrt{2/\rho}\,\sqrt{p - p'};$$

$$Q'_5 = A_s \frac{dx_s}{dt} + a_1 p' + \frac{V'}{E}\frac{dp'}{dt}; \quad (4.15)$$

$$m\frac{d^2x_s}{dt^2} + C\frac{dx_s}{dt} + K(x_s - x_{0s}) = \rho L\frac{dQ'_4}{dt} + p'A - 2C_{DSP}\pi D x_s \cos\varphi\, p.$$

Filtru:

$$Q_1 R_{cf} = p - p_1. \quad (4.16)$$

Servovalvă cu două etaje de amplificare:

$$K_c(i - K_r\omega_M) + (p_4 - \Delta p)\frac{\pi d^2}{4}\cos\lambda - (p_4 + \Delta p)\frac{\pi d^2}{4}\cos\lambda =$$

$$= I_K\frac{d^2\lambda}{dt^2} + \varphi_K\frac{d\lambda}{dt} + K_K\lambda;$$

$$Q_9 = C_{D9}\pi d\left(x_0 - \lambda\frac{L}{2}\right)\sqrt{2/\rho}\,\sqrt{p_4 + \Delta p - p_5};$$

$$Q_9' = C_{D9}\pi d\left(x_0 + \lambda\frac{L}{2}\right)\sqrt{2/\rho}\,\sqrt{p_4 - \Delta p - p_5};$$

$$Q_8 = Q_9 + Q_{10};$$

$$Q_8' + Q_{10} = Q_9';$$

$$Q_8 = C_{D8}S_8\sqrt{2/\rho}\,\sqrt{p_1 - p_4 - \Delta p};$$

$$Q_8' = C_{D8}S_8\sqrt{2/\rho}\,\sqrt{p_1 - p_4 + \Delta p};$$

$$p_1 = \left[1 + \left(\frac{C_{D9}}{C_{D8}}\right)^2\left(\frac{\pi d x_0}{S_8}\right)^2\right]p_4, \quad \text{unde} \quad \frac{C_{D9}}{C_{D8}} < 1; \quad \frac{\pi d x_0}{S_8} < 1 > p_4 \ll p_1;$$

$$Q_{10} = A\frac{dy}{dt} + a(p_4 + \Delta p) + \frac{V}{E}\frac{d(p_4 + \Delta p)}{dt};$$

$$Q_1 = Q_8 + Q_4 + Q_8'; \qquad (4.17)$$

$$Q_4 = C_D\pi D(y - y_0)\sqrt{2/\rho}\,\sqrt{p_1 - p_2};$$

$$(p_4 + \Delta p)A - (p_4 - \Delta p)A = m\frac{d^2y}{dt^2} + h\frac{dy}{dt} + K_p y + \rho L_r\frac{dQ_4}{dt} +$$

$$+ 2\pi C_D D y(p_1 - p_2)\cos\varphi.$$

Motor hidraulic rotativ:

$$Q_M = Q_4 = \frac{q_M}{2\pi}\omega_M + a_M p_2 + \frac{q_M}{2E}\frac{dp_2}{dt}; \qquad (4.18)$$

$$J_M\frac{d\omega_M}{dt} + b_M\omega_M + \frac{\omega_M}{|\omega_M|}c_{fM}\frac{q_M}{2\pi}(p_2 - p_7) + M = \frac{q_M}{2\pi}(p_2 - p_7).$$

$$Q_p(s) = \frac{q_p}{2\pi}\,\Omega_p(s) - a_p P(s) - \frac{q_p}{2E}\,sP(s);$$

$$J_p s\Omega_p(s) + b_p\Omega_p(s) + \left(\frac{q_p}{2\pi} + \frac{q_p}{2\pi}c_{fp}\right)P(s) = K_e(\Omega_s(s) - \Omega_p(s));$$

$$Q_1(s) = Q_p(s) - Q'_4(s) - Q'_5(s);$$

$$Q'_4(s) = C_{DSP}\pi D X_s(s)\,\sqrt{2/\rho}\,\sqrt{P(s)};$$

$$Q'_5(s) = C_D S\,\sqrt{2/\rho}\,\sqrt{P(s) - P'(s)};$$

$$Q_6'(s) = A_s s X_s(s) + a_1 P'(s) + \frac{V}{E} s P'(s);$$

$$(ms^2 + Cs + K)\, X_s(s) + K X_{0s} = \rho L s Q_1'(s) - $$
$$+ P'(s)\, A - 2C_{Ds\rho}\pi D X_s(s) \cdot P(s) \cos\varphi;$$

$$Q_1(s)\, R_{i,f} = P(s) - P_1(s);$$

$$K_c [I(s) - K_s \Omega_M(s)] + [P_4(s) - \Delta P(s)] \frac{\pi d^2}{4} \cos\lambda$$

$$-[P_4(s) + \Delta P(s)] \frac{\pi d^2}{4} \cos\lambda = [I_K s^2 + \varphi_\lambda s + K_\lambda] \lambda(s);$$

$$Q_9(s) = C_{D9}\pi d \left[x_0 - \lambda(s)\frac{L}{2}\right] \sqrt{2/\rho}\, \sqrt{P_4(s) + \Delta P(s) - P_r};$$

$$Q_9'(s) = C_{D9}\pi d \left[x_0 + \lambda(s)\frac{L}{2}\right] \sqrt{2/\rho}\, \sqrt{P_4(s) - \Delta P(s) - P_r};$$

$$Q_8(s) = Q_9(s) + Q_{10}(s);$$

$$Q_8'(s) + Q_{10}(s) = Q_9'(s);$$

$$Q_8(s) = C_{D8} S_8 \sqrt{2/\rho}\, \sqrt{P_1(s) - P_4(s) - \Delta P(s)};$$

$$Q_8'(s) = C_{D8} S_8 \sqrt{2/\rho}\, \sqrt{P_1(s) - P_4(s) + \Delta P(s)};$$

$$P_1(s) = \left[1 + \left(\frac{C_{D9}}{C_{D8}}\right)^2 \left(\frac{\pi d x_0}{S_8}\right)^2\right] P_4(s)$$

(relație care stabilește echilibru valoarea lui P_4 funcție de P_1)

$$Q_{10}(s) = A s Y(s) + a(P_4(s) + \Delta P(s)) + \frac{V}{E} s (P_4(s) + \Delta P(s))$$

$$Q_1(s) = Q_8(s) + Q_4(s) + Q_8'(s);$$

$$Q_4(s) = C_D \pi D (Y(s) - Y_0) \sqrt{2/\rho}\, \sqrt{P_1(s) - P_2(s)};$$

$$(P_4(s) + \Delta P(s))\, A - [P_4(s) - \Delta P(s)]\, A = (ms^2 + hs +$$
$$+ K_s)\, Y(s) + \rho L_s s Q_4(s) + 2\pi C_D D Y(s)\, (P_1(s) - P_2(s)) \cos\varphi;$$

$$Q_4 = \frac{q_M}{2\pi} \Omega_M(s) + a_M P_2(s) + \frac{q_M}{2E} s P_2(s);$$

$$J_M s \Omega_M(s) + b_M \Omega_M(s) + \frac{\Omega_M(s)}{|\Omega_M(s)|} c_{f_s} \frac{q_M}{2\pi} (P_2(s) - P_r(s)) +$$

$$+ M = \frac{q_M}{2\pi} (P_2(s) - P_r(s)).$$

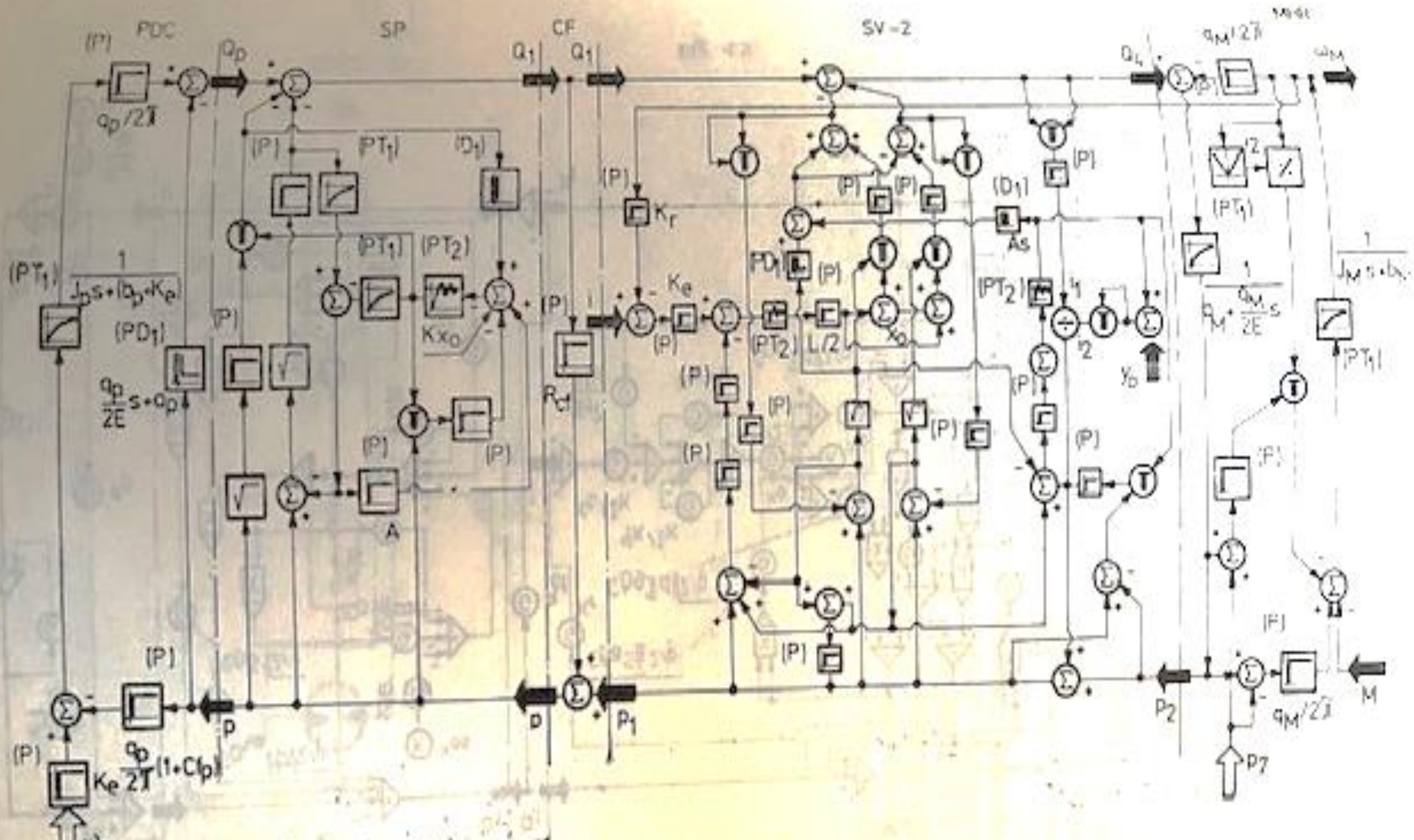


Fig. 4.8

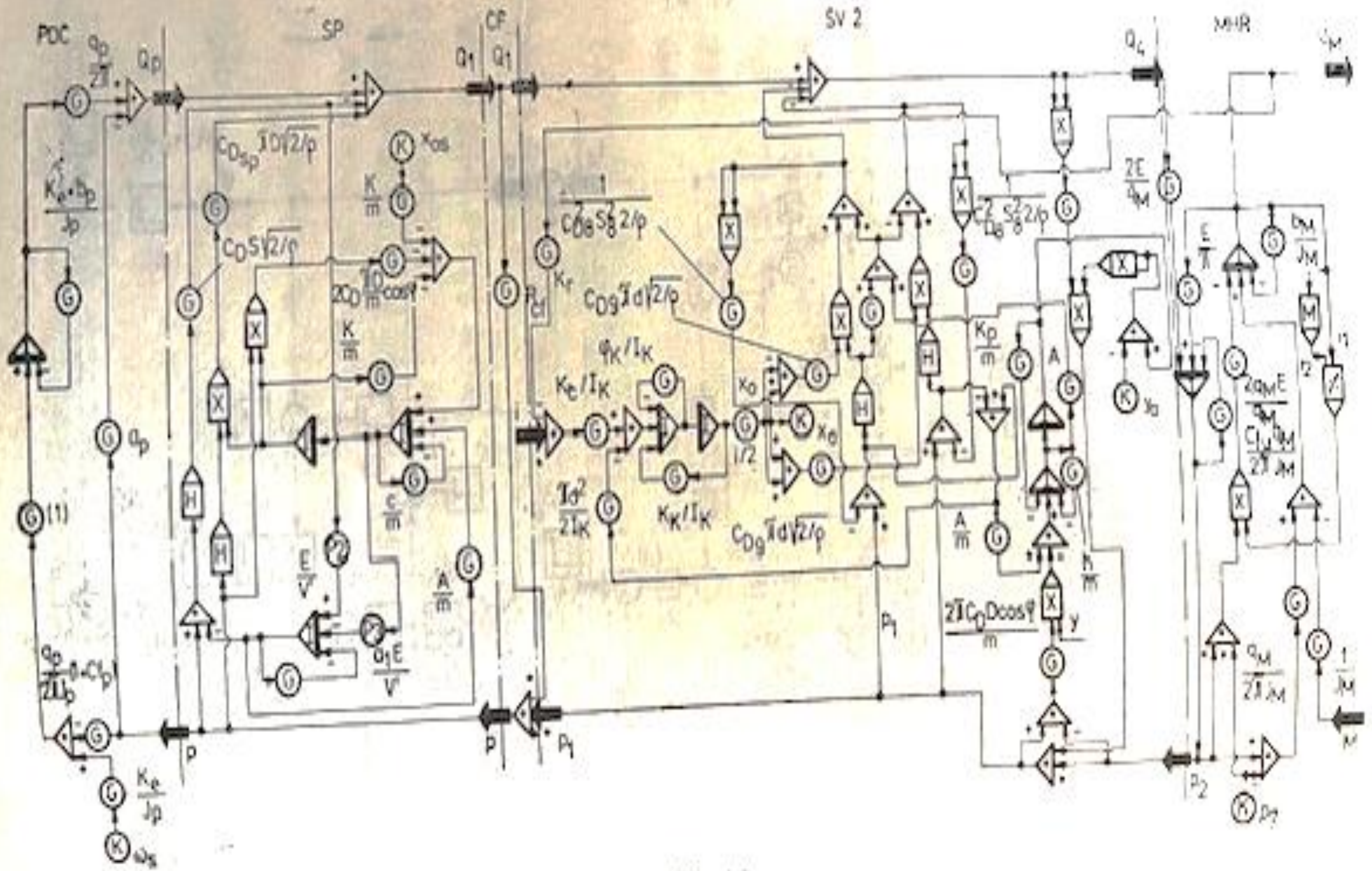
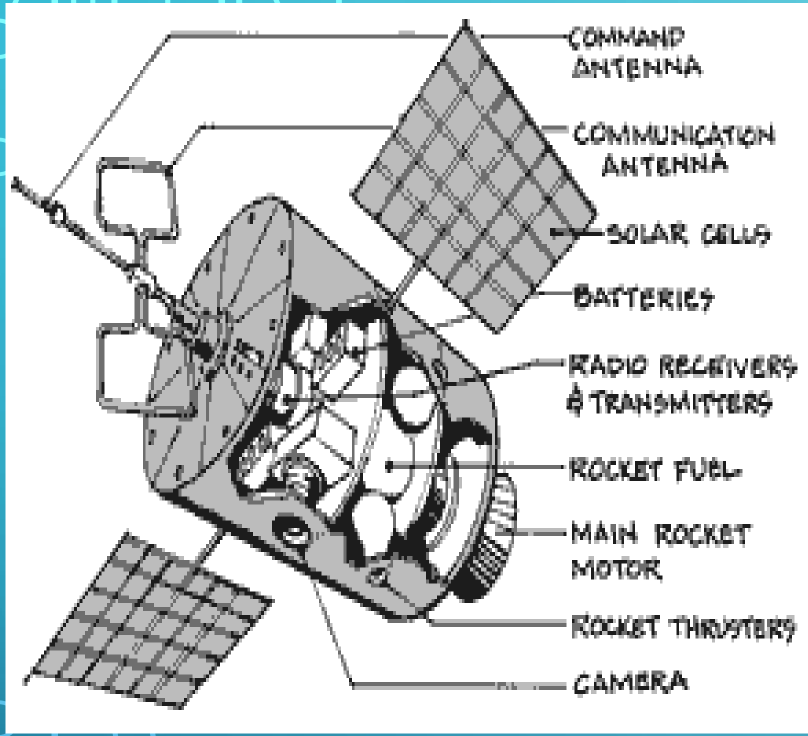
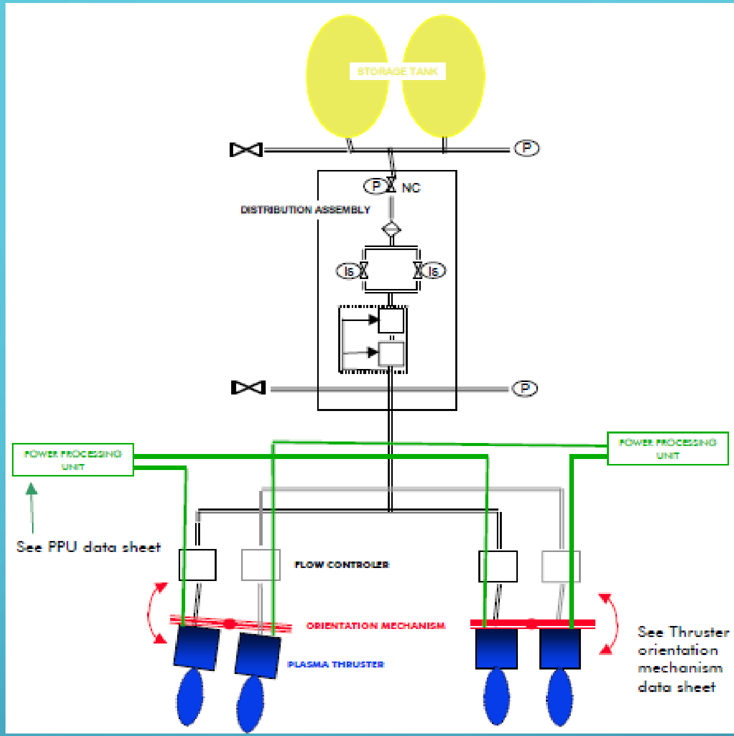

PDC
SP
CF
SV 2
MHR
Q_p
Q_1
Q_1
Q_4
q_M
P
P
P_1
P_1
P_2
M
$\frac{q_p}{2\pi}$
$\frac{K_e \cdot \theta_p}{J_p}$
$C_{Dsp}\pi D\sqrt{2/\rho}$
$C_D S\sqrt{2/\rho}$
K_{os}
$\frac{K}{m}$
$\frac{c}{m}$
$\frac{A}{m}$
$\frac{E}{V'}$
$\frac{q_1 E}{V'}$
$2C_D\frac{\pi D}{m}\cos\varphi$
$\frac{q_p}{2\pi J_p}$
$\frac{K_e}{J_p}$
ω_s
K_r
K_e/I_K
φ_K/I_K
K_K/I_K
$\frac{\pi d^2}{2 I_K}$
$C_{Dg}\pi d\sqrt{2/\rho}$
$C_{De}S_e^2\,2/\rho$
K_o
X_e
1/2
$\frac{K_p}{m}$
A
$\frac{2\pi C_D D\cos\varphi}{m}$
y
$\frac{2E}{q_M}$
$\frac{E}{\pi}$
y_o
$\frac{b_M}{J_M}$
$\frac{2 q_M E}{q_M}$
$\frac{C_{1M}\,q_M}{2\pi J_M}$
$\frac{q_M}{2\pi J_M}$
$\frac{1}{J_M}$
p_1

Fig. 4.9

5. STATE OF ART OF COMPONENTS AND SATELLITE ORIENTATION





There are 4 major components of the small satellite fig.1 [1-6]:

(i) ***The transponder and antenna system*** that contents the high – frequency radio receiver, a frequency down-converter and a power amplifier, which is used to transmit the signal. The antenna system contains the antennas and the mechanism to position them correctly;

(ii) ***Power System*** that is a power supply to the satellite. The satellite must be powered either from a battery or a solar energy system.

(iii) ***Control and information system and rocket thruster system*** contains the control system, on-board processors, the rocket thruster system, the fuel tanks; the satellite is weightless in the space, it is very susceptible to external forces of the Sun and Earth that cause the satellite to disorient and move. **This are called the satellite keeping system;** the thrusters are needed to decrease the natural forces influenced by the Earth's and moon gravity, the solar wind and variations in the Earth's magnetic field. This system try to keep the satellite in the correct orbit with the antennas pointed in the exact desired direction [1-6].

(iv) ***The thermal control system*** is necessary **to keep enough cool the electronics on board of the satellite**, to work properly. Without thermal control, electronics would overheat and cease to work.

6. APPLICATION IN MECHANICAL INERTIAL ORIENTATION SATELLITE-MATHEMATICAL MODELLING AND SIMULATION OF THE PROPOSED METHOD

(i) Dynamic behaviour of the satellite servo driving and Forward Kinematics (FK)

$$K_m \cdot (i) = [J_{red}] \cdot \frac{d}{dt}(\omega_m) + b_m \cdot (\omega_m)$$

$$(U_m) - K_e \cdot (\omega_m) = R_a \cdot (i) + L_a \cdot \frac{d}{dt}(i)$$

$$\frac{1}{2}(\omega_V)^T [J_V](\omega_V) = \frac{1}{2}(\omega_S)^T ([J_S] + [J_R])(\omega_S)$$

$$(\omega_V) = \frac{1}{i_{red}}(\omega_m); [J_{red}] = \frac{1}{i_{red}^2}[J_V] + [J_m]$$

$$(U_m) = \frac{L_a}{K_m} \cdot (M) + (\frac{R_a}{K_m} \cdot [J_{red}] + L_a \cdot \frac{b}{K_m}) \frac{d}{dt}(\omega_m) + (R_a \cdot \frac{b_m}{K_m} + K_e) \cdot (\omega_m)$$

$$P_{STOP} = T(\varphi_{XS}, \varphi_{YS}, \varphi_{ZS}) P_{START}; T = \begin{bmatrix} c_2 c_3 & s_1 s_2 c_3 - c_1 s_3 & c_1 s_2 c_3 + s_1 s_3 \\ c_2 s_3 & s_1 s_2 s_3 + c_1 c_3 & c_1 s_2 s_3 - s_1 c_3 \\ -s_2 & s_1 c_2 & c_1 c_2 \end{bmatrix}$$

$$s_1 = \sin(\varphi_{XS}); s_2 = \sin(\varphi_{YS}); s_3 = \sin(\varphi_{ZS})$$

$$c_1 = \cos(\varphi_{XS}); c_2 = \cos(\varphi_{YS}); c_3 = \cos(\varphi_{ZS}).$$

$$X_{STOP} = c_2 c_3 X_{START} + (s_1 s_2 c_3 - c_1 s_3) Y_{START} + (c_1 s_2 c_3 + s_1 s_3) Z_{START}$$

$$Y_{STOP} = c_2 s_3 X_{START} + (s_1 s_2 s_3 + c_1 c_3) Y_{START} + (c_1 s_2 s_3 - s_1 c_3) Z_{START}$$

$$Z_{STOP} = -s_2 X_{START} + s_1 c_2 Y_{START} + c_1 c_2 Z_{START}$$

$$\Delta\varphi_{XS} = \int_o^{t_X} \omega_{XS} dt; \Delta\varphi_{YS} = \int_o^{t_Y} \omega_{YS} dt; \Delta\varphi_{ZS} = \int_o^{t_Z} \omega_{ZS} dt;$$

$$N_X = \frac{\varphi_{XS}}{\Delta\varphi_{XS}}; N_Y = \frac{\varphi_{YS}}{\Delta\varphi_{YS}}; N_Z = \frac{\varphi_{ZS}}{\Delta\varphi_{ZS}}$$

K_m- moment motor gradient with the current intensity; (i)- column matrix of the absorbed intensity; $[J_{red}]$- inertial tensor reduced to the motor axes; (ω_m)- column matrix of the angular velocity on the each motor axes; b_m- moment motor gradient with angular velocity; (U_m)- column matrix of the motor electrical tension; (L_a)- column matrix of the motor inductances; (R_a)- column matrix of the electrical resistance; (P_{stop})- column matrix of the final space position of the satellite tool centre point; (P_{start})- column matrix of the home space position of the satellite tool centre point; s_i, c_i- director cosinus of the transfer matrix between Cartesian axes; $[J_V]$- inertia tensor of the metallic disk; $[J_S]$ – inertia tensor of the satellite; $[J_R]$- inertia tensor of the resistive forces from the space; column matrix velocity of the satellite; (ω_V)- angular column matrix velocity of the disk; (ω_S)- angular column matrix of the satellite; $(\omega_{Sx,y,z})$- angular column matrix of the velocity in each axes; $N_{x,y,z}$- number of pulses in each motor axes; t_x, t_y, t_z are the integration times. This algorithm was transposed in LabView subVI-s and were researched how some of these parameters influence the servo driving dynamic behavior [12].

(ii) Complex algorithm of the pseudoinverse Jacobian matrix method with applied neural network to solve the Inverse Kinematics with the errors less than 0.001mm

$$(q_{current}) = (q_{previous}) + (\Delta q)$$

$$(\Delta q) = \propto * \{(P_{target}) - (P_{FK})\} * \{[J_{q_i}]^T * ([J_{q_i}] * [J_{q_i}]^T)^{-1}\}$$

$$n_1 = [\underbrace{w^1}_{p_1} + \underbrace{tcg_1}_{p_2} \cdot \varepsilon_1](p - a_2(t - p_3 + 1)) + (b_1 + \varepsilon_1)$$

$$a_1 = \frac{p_4(1 - e^{-n_1})}{1 + e^{-n_1}}$$

$$\varepsilon_1 = t_1 - a_1$$

$$n_2 = [w^2 + \underbrace{tcg_2}_{p5} \cdot \varepsilon_2](a_1(t - p_6 + 1)) + (b_2 + \varepsilon_2)$$

$$a_2 = \frac{p_7(1 - e^{-n_2})}{1 + e^{-n_2}}$$

$$\varepsilon_2 = t_2 - a_2$$

$$q_i = p_8(a_2 - \varepsilon_f)$$

$$\varepsilon_{pos} = t_3 - r_i$$

$$n_3 = [w^3 + \underbrace{tcg_2}_{p_5} \cdot \varepsilon_{pos}](q_i) + (b_3 + \varepsilon_{pos})$$

$$a_3 = \frac{p_9(1 - e^{-n_3})}{1 + e^{-n_3}}$$

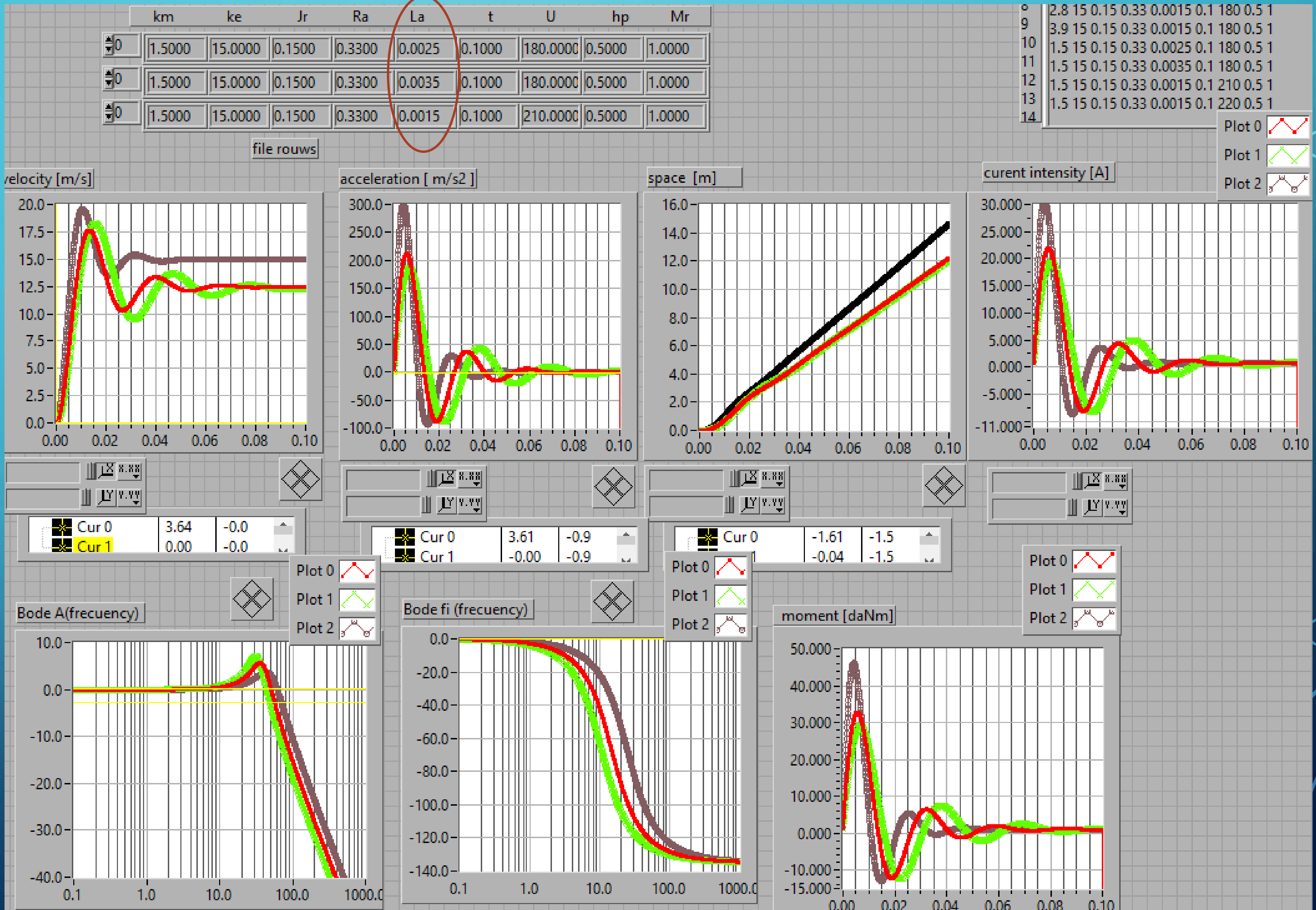
$$\varepsilon_f = t_2 - a_3$$

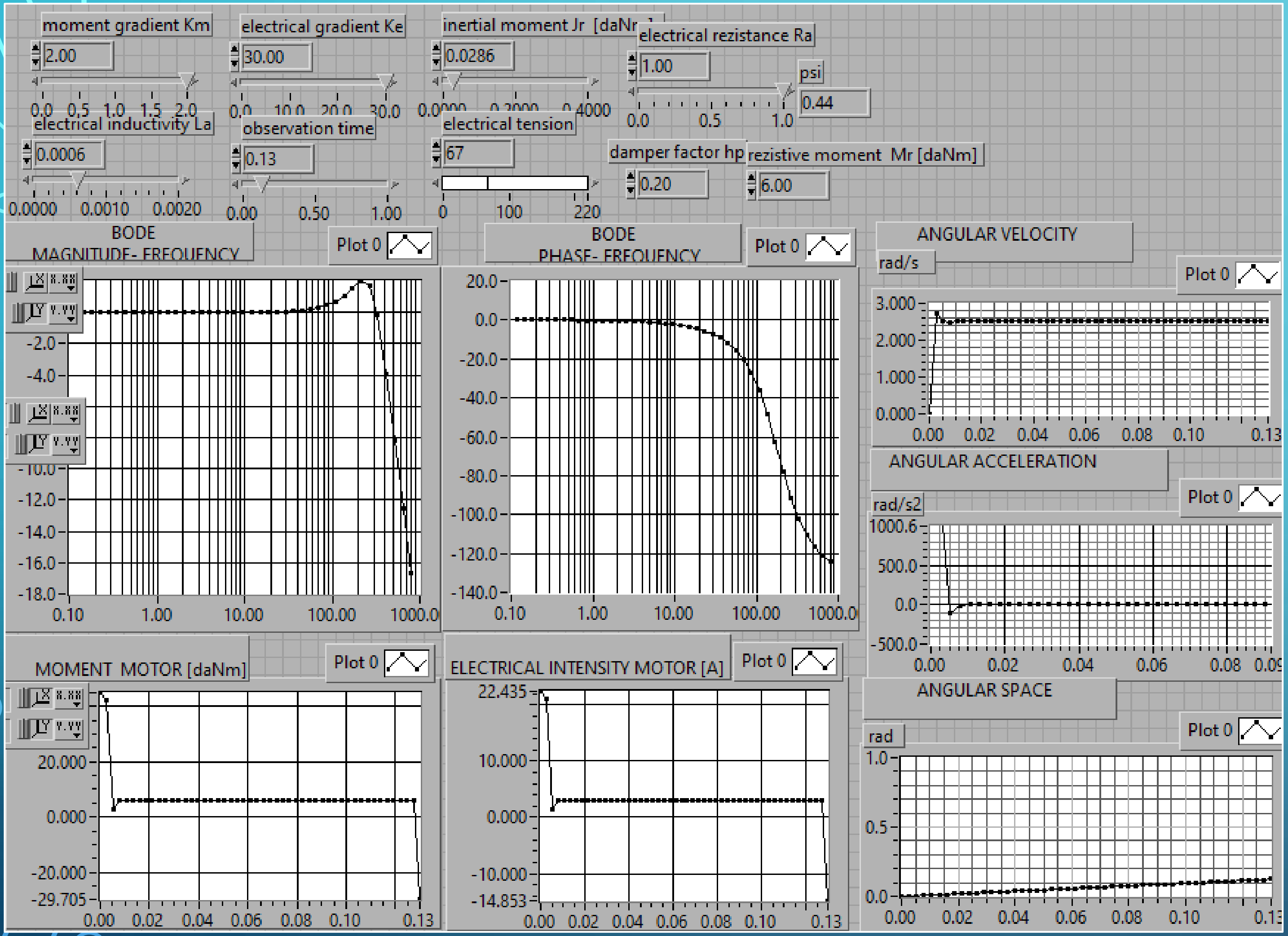
$$J_{x1} = (c1s2c3 + s1s3)y + (-s1s2c3 + c1s3)z$$
$$J_{x2} = -s2c3x + s1c2c3y + c1c2c3z$$
$$J_{x3} = -c2s3x + (-s1s2s3 - c1c3)y + (-c1s2s3 + s1c3)z$$
$$J_{y1} = (c1s2s3 - s1c3)y + (-s1s2s3 - c1c3)z$$
$$J_{y2} = -s2s3x + (s1c2s3 + c1c3)y + c1c2s3z$$
$$J_{y3} = c2c3x + (s1s2c3 - c1s3)y + (c1s2c3 + s1s3)z$$
$$J_{z1} = c1c2y - s1c2z$$
$$J_{z2} = -c2x - s1s2y - c1s2z$$
$$J_{z3} = 0.0$$

(q_{curent})- column matrix of the internal coordinate (rotation in each satellite axes) calculated in current iteration; $(q_{previous})$- column matrix of the internal coordinate calculated in previous iteration; $\propto$ - teaching gain of the iterative Jacobian method; (P_{target})- is the P_{stop}; (P_{FK})- current column matrix of the point P determined by applied the FK; $[J_{q_i}]^T([J_{q_i}][J_{q_i}]^T)^{-1}$- pseudo inverse Jacobian matrix; $[w_i]$- neural network weight matrices; (tcg_i)- different teaching gain for input layer, hidden layer and output layer; $(p_{3,6})$- column matrices of different delay of time; $(p_{4,7,9})$- different hyperbolic tangent gains; (b_i)- different biases column matrices; (n_i)- column matrices of the threshold of the sensitive function of the current layer in the neural network; x,y,z- home know space position.

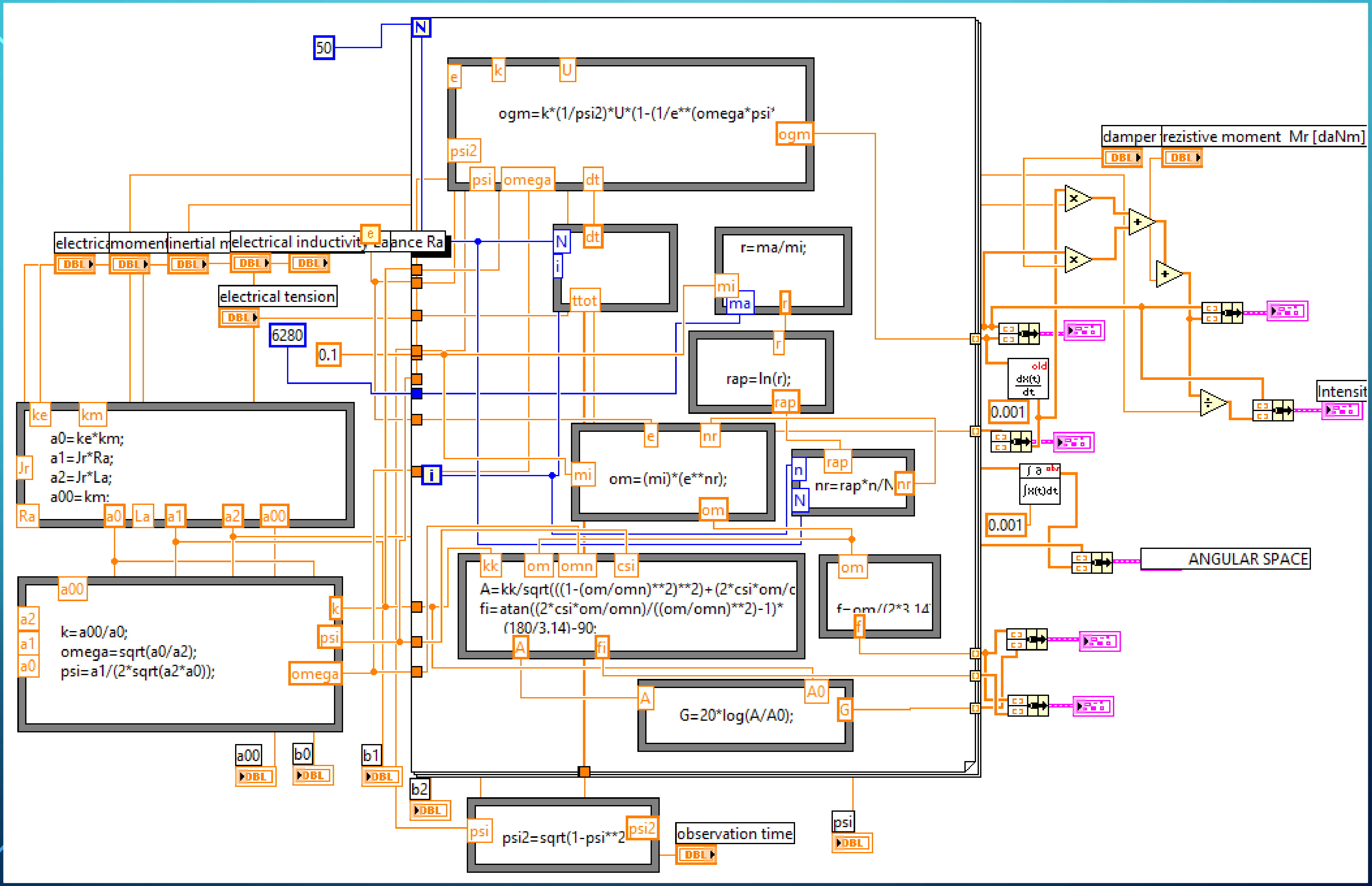
7. ASSISTED ANALYSE OF THE PROPOSED ALGORITHM AND SOME RESULTS

(i). Assisted analyse of the servo driving with DC motor and three stablized inertial wheels

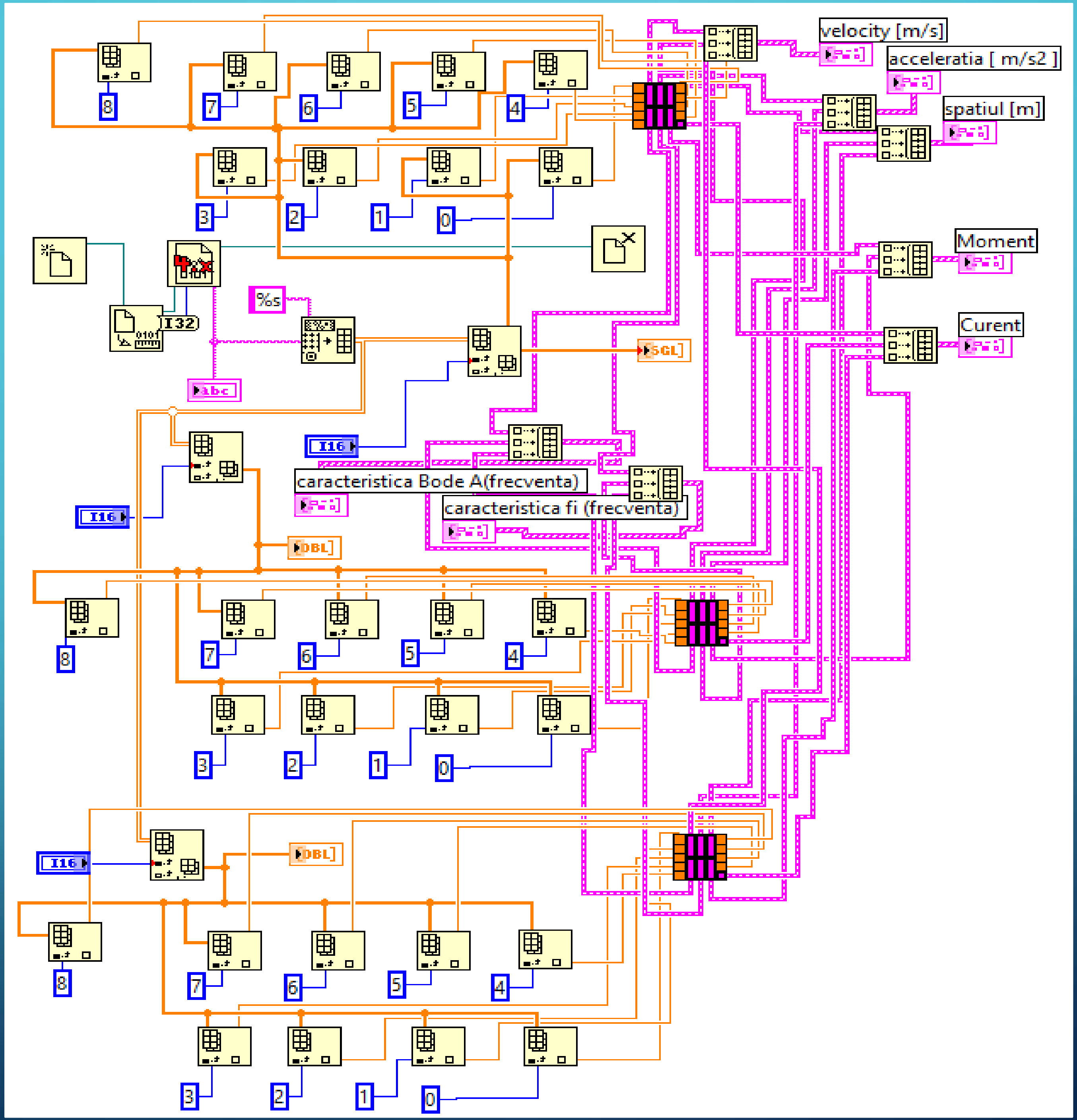




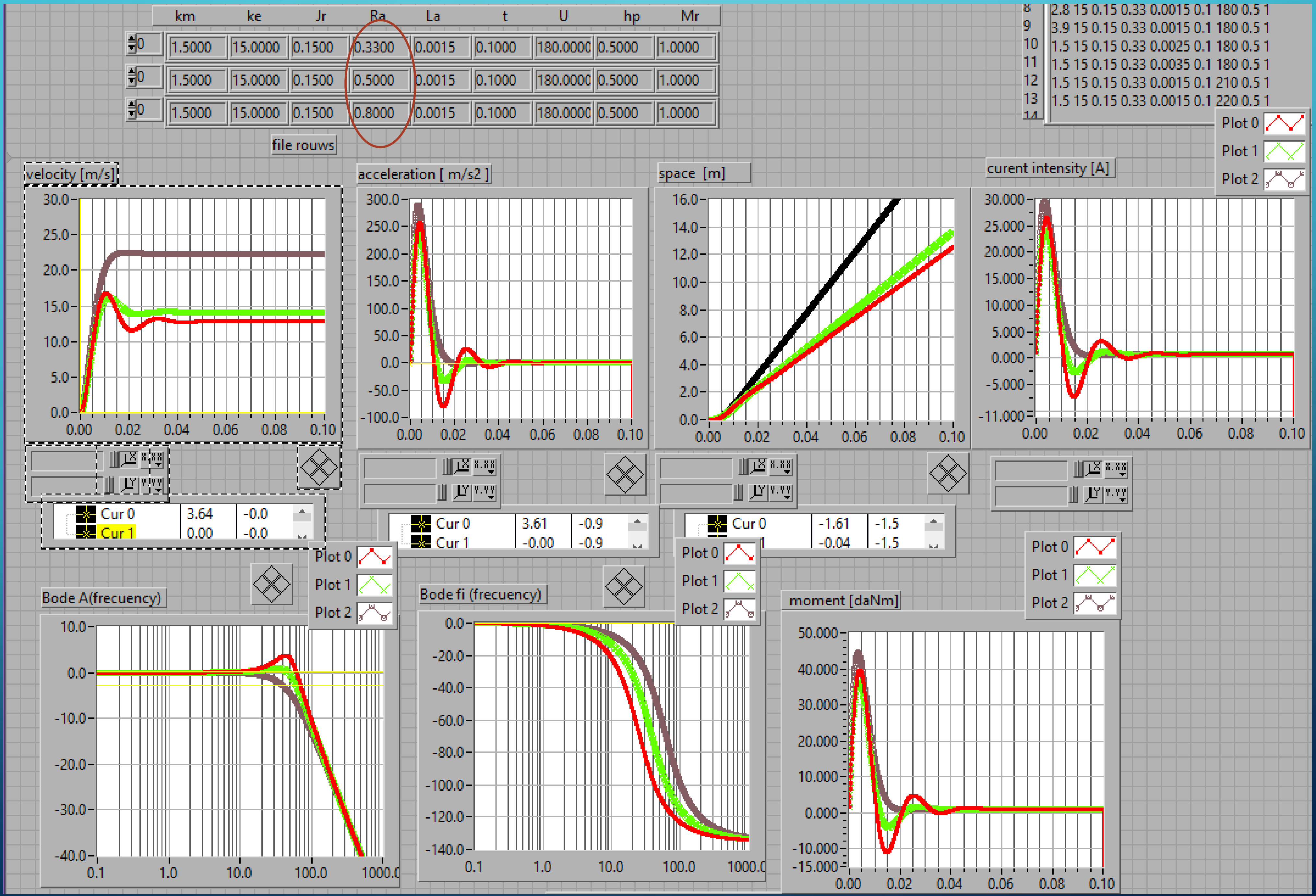
Km	ke	Jr	Ra	La	t	U	hp	Mr	
1.5	15	0.15	0.33	0.0015		0.1	180	0.5	1
1.5	15	0.15	0.5	0.0015		0.1	180	0.5	1
1.5	15	0.15	0.8	0.0015		0.1	180	0.5	1
1.5	15	0.35	0.33	0.0015		0.1	180	0.5	1
1.5	15	0.55	0.33	0.0015		0.1	180	0.5	1
1.5	25	0.15	0.33	0.0015		0.1	180	0.5	1
1.5	35	0.15	0.33	0.0015		0.1	180	0.5	1
2.8	15	0.15	0.33	0.0015		0.1	180	0.5	1
3.9	15	0.15	0.33	0.0015		0.1	180	0.5	1
1.5	15	0.15	0.33	0.0025		0.1	180	0.5	1
1.5	15	0.15	0.33	0.0035		0.1	180	0.5	1
1.5	15	0.15	0.33	0.0015		0.1	210	0.5	1
1.5	15	0.15	0.33	0.0015		0.1	220	0.5	1

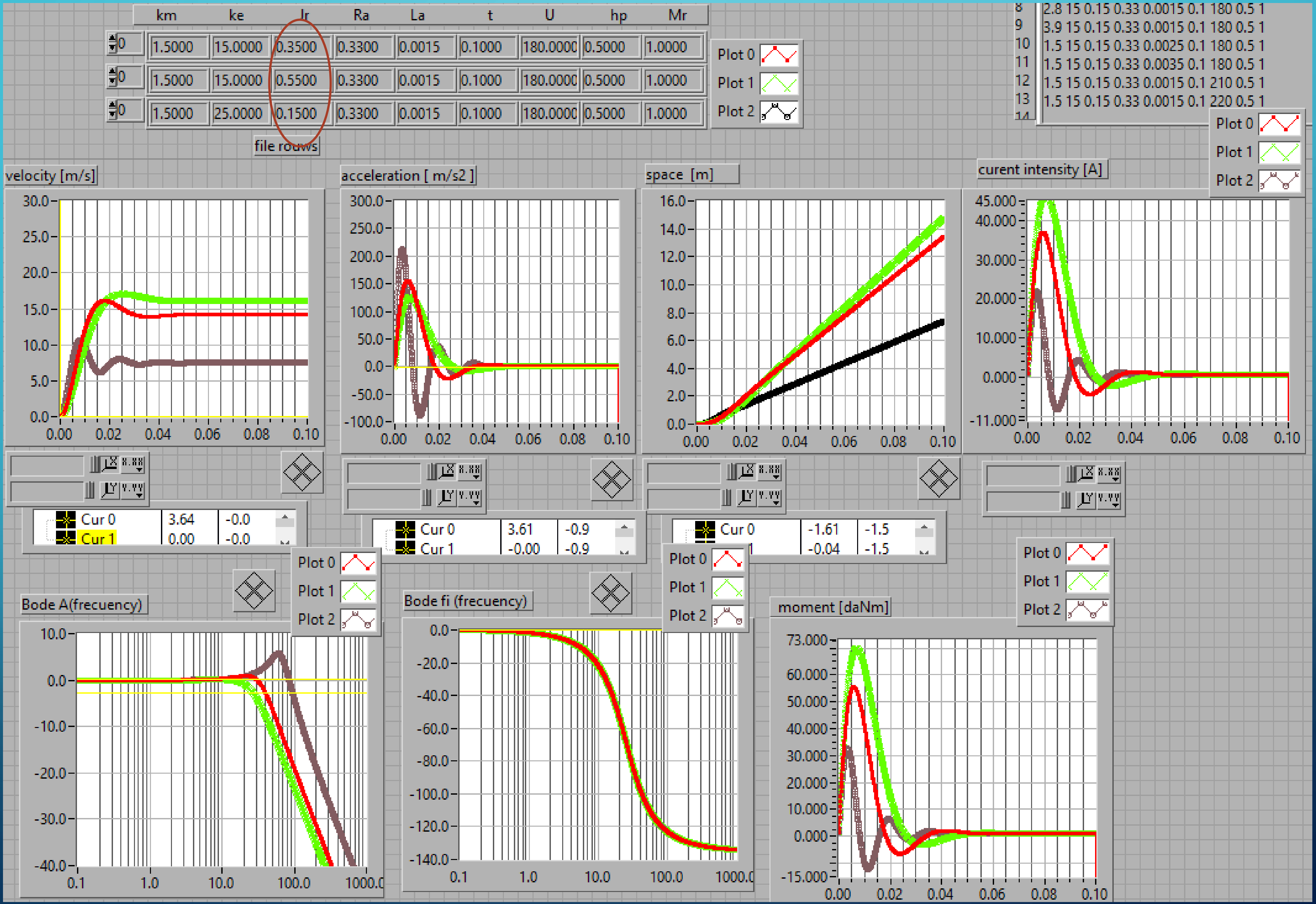


50
N
ogm=k*(1/psi2)*U*(1-(1/e**(omega*psi*
e
k
U
ogm
psi2
psi
omega
dt
damper
rezistive moment Mr [daNm]
electrica
moment
inertial m
electrical inductivit
lance Ra
DBL
electrical tension
6280
0.1
ttot
r=ma/mi;
mi
ma
r
rap=ln(r);
rap
ke
km
a0=ke*km;
a1=Jr*Ra;
a2=Jr*La;
a00=km;
Jr
Ra
a0
La
a1
a2
a00
om=(mi)*(e**nr);
nr
om
nr=rap*n/N
n
i
0.001
Intensit
ANGULAR SPACE
k=a00/a0;
omega=sqrt(a0/a2);
psi=a1/(2*sqrt(a2*a0));
kk
omn
csi
A=kk/sqrt(((1-(om/omn)**2)**2)+(2*csi*om/c
fi=atan((2*csi*om/omn)/((om/omn)**2)-1)*
(180/3.14)-90;
A
fi
f=om/(2*3.14
f
A0
G=20*log(A/A0);
G
b0
b1
b2
psi2=sqrt(1-psi**2
observation time

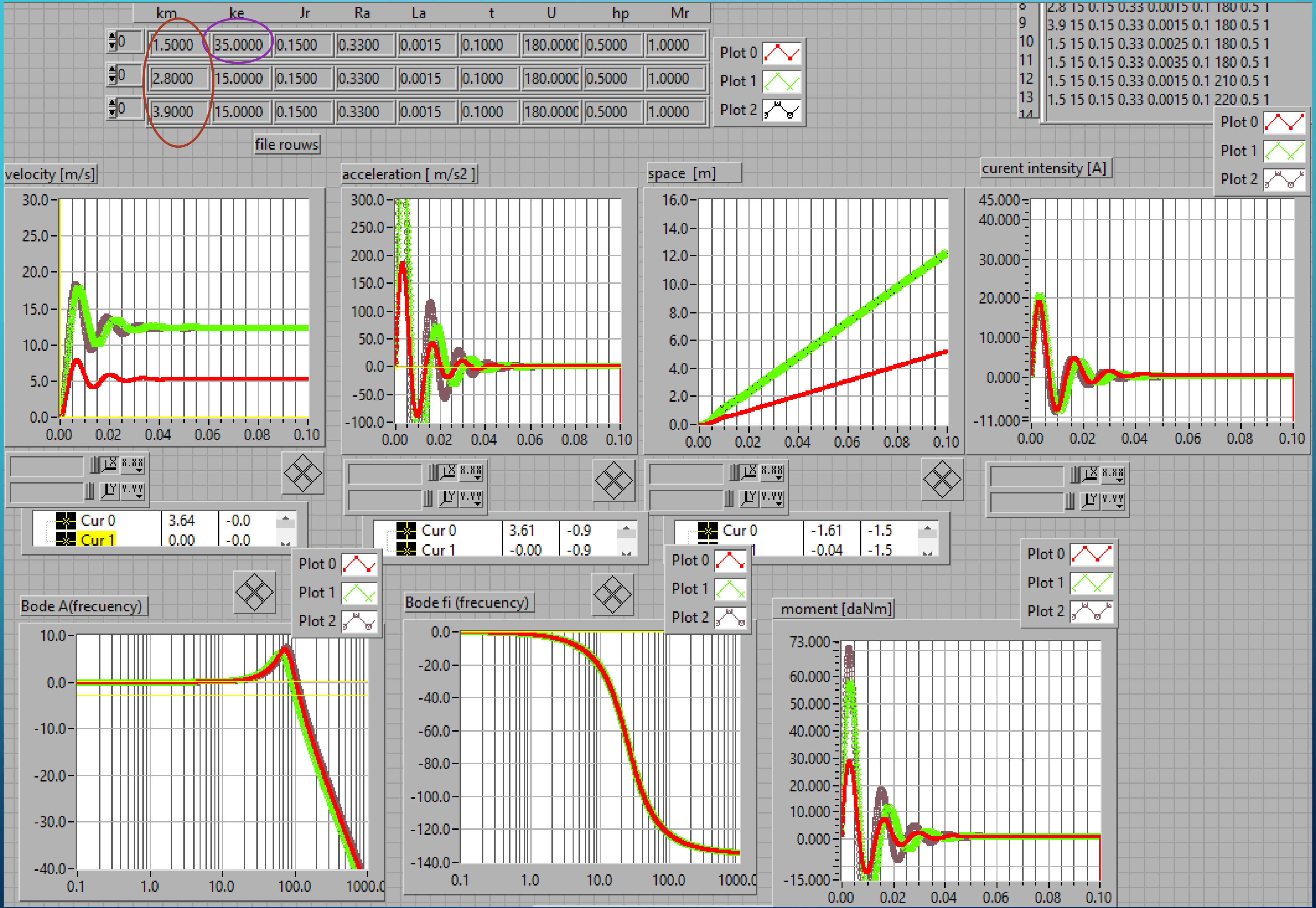


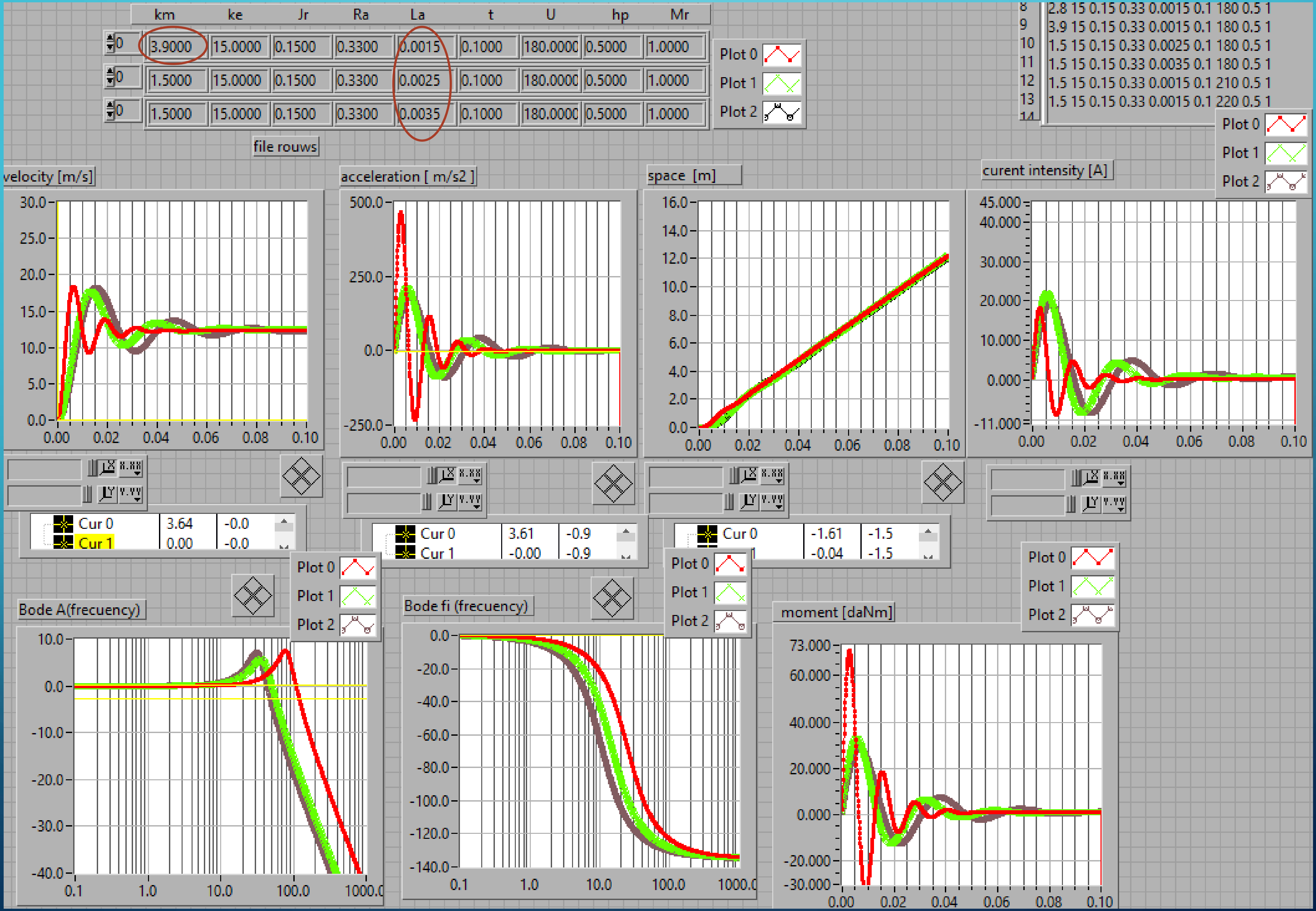
velocity [m/s]
acceleratia [m/s2]
spatiul [m]
Moment
Curent
%s
abc
I32
SGL
I16
caracteristica Bode A(frecventa)
caracteristica fi (frecventa)
DBL
8
7
6
5
4
3
2
1
0

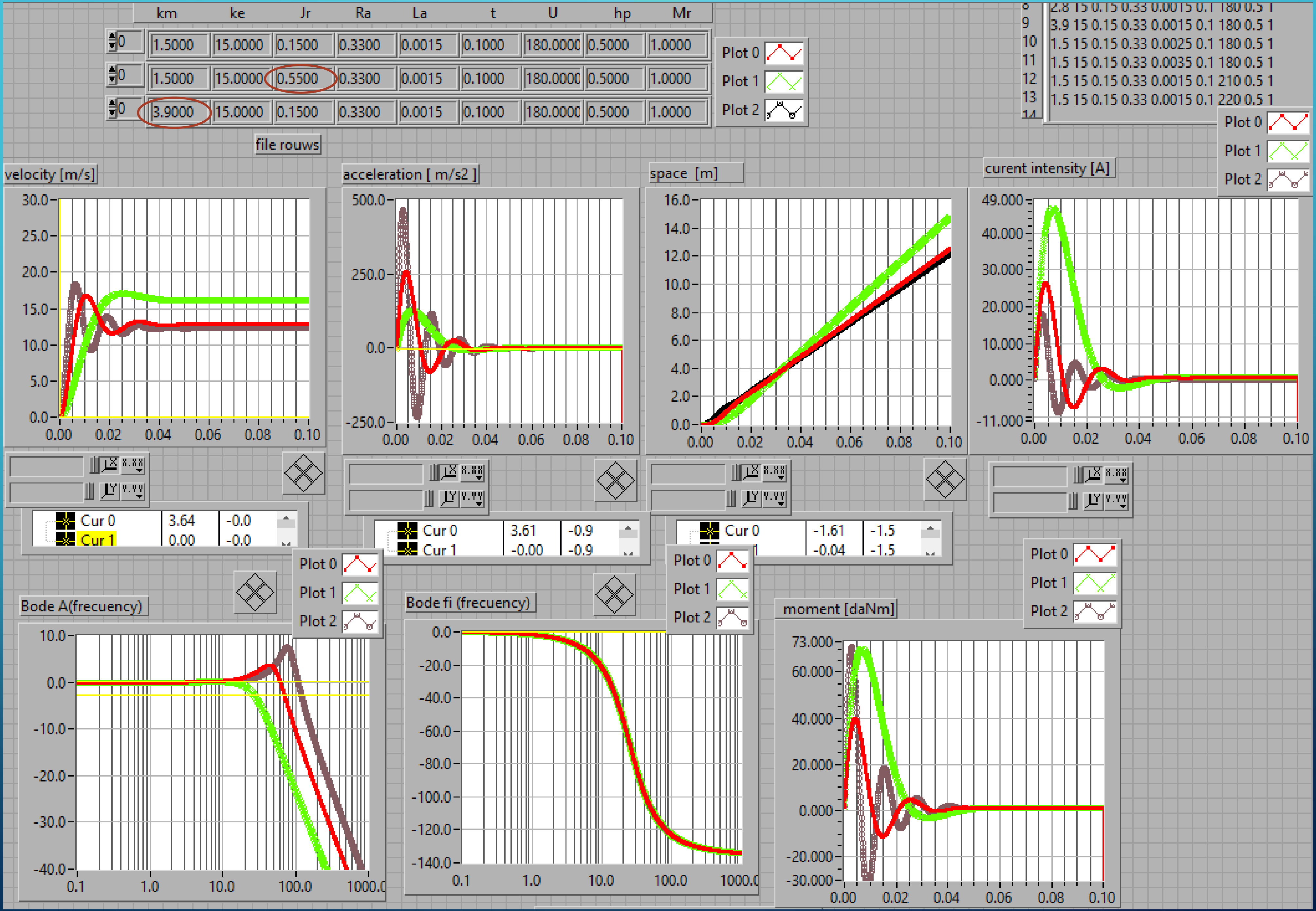

km ke Jr Ra La t U hp Mr
0 1.5000 15.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
0 1.5000 15.0000 0.1500 0.5000 0.0015 0.1000 180.0000 0.5000 1.0000
0 1.5000 15.0000 0.1500 0.8000 0.0015 0.1000 180.0000 0.5000 1.0000
file rouws
8 2.8 15 0.15 0.33 0.0015 0.1 180 0.5 1
9 3.9 15 0.15 0.33 0.0015 0.1 180 0.5 1
10 1.5 15 0.15 0.33 0.0025 0.1 180 0.5 1
11 1.5 15 0.15 0.33 0.0035 0.1 180 0.5 1
12 1.5 15 0.15 0.33 0.0015 0.1 210 0.5 1
13 1.5 15 0.15 0.33 0.0015 0.1 220 0.5 1
Plot 0
Plot 1
Plot 2
velocity [m/s]
acceleration [m/s2]
space [m]
curent intensity [A]
Cur 0 3.64 -0.0
Cur 1 0.00 -0.0
Cur 0 3.61 -0.9
Cur 1 -0.00 -0.9
Cur 0 -1.61 -1.5
-0.04 -1.5
Bode A(frecuency)
Bode fi (frecuency)
moment [daNm]



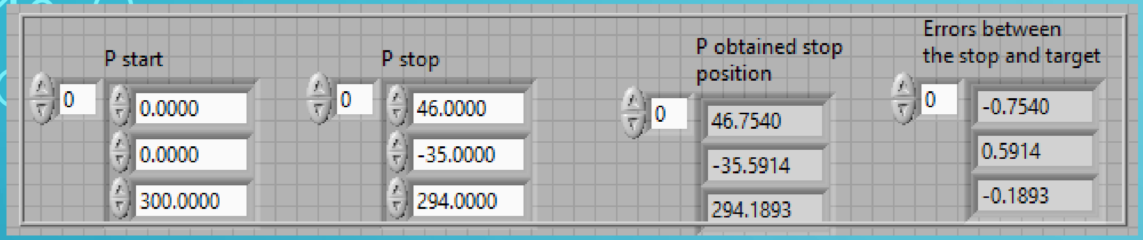
km ke Jr Ra La t U hp Mr
1.5000 15.0000 0.3500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
1.5000 15.0000 0.5500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
1.5000 25.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
file rows
Plot 0
Plot 1
Plot 2
8 2.8 15 0.15 0.33 0.0015 0.1 180 0.5 1
9 3.9 15 0.15 0.33 0.0015 0.1 180 0.5 1
10 1.5 15 0.15 0.33 0.0025 0.1 180 0.5 1
11 1.5 15 0.15 0.33 0.0035 0.1 180 0.5 1
12 1.5 15 0.15 0.33 0.0015 0.1 210 0.5 1
13 1.5 15 0.15 0.33 0.0015 0.1 220 0.5 1
velocity [m/s]
acceleration [m/s2]
space [m]
curent intensity [A]
Cur 0 3.64 -0.0
Cur 1 0.00 -0.0
Cur 0 3.61 -0.9
Cur 1 -0.00 -0.9
Cur 0 -1.61 -1.5
-0.04 -1.5
Bode A(frecuency)
Bode fi (frecuency)
moment [daNm]

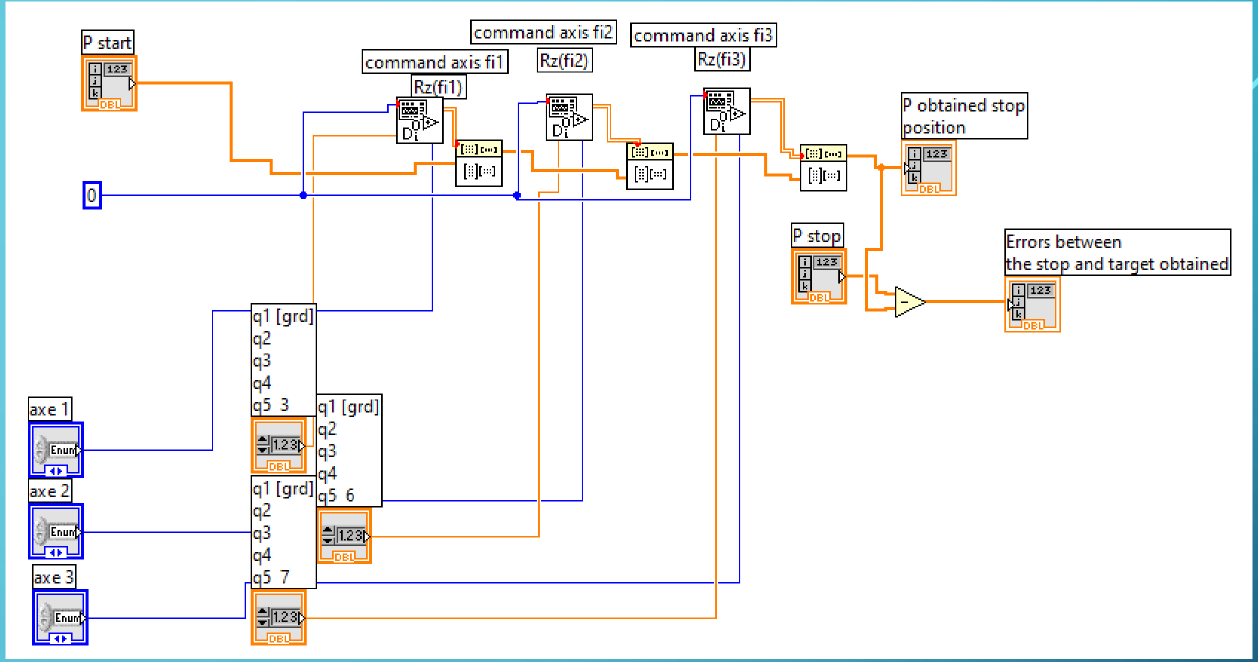

km ke Jr Ra La t U hp Mr
0 1.5000 35.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
0 2.8000 15.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
0 3.9000 15.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
file rouws
9 3.9 15 0.15 0.33 0.0015 0.1 180 0.5 1
10 1.5 15 0.15 0.33 0.0025 0.1 180 0.5 1
11 1.5 15 0.15 0.33 0.0035 0.1 180 0.5 1
12 1.5 15 0.15 0.33 0.0015 0.1 210 0.5 1
13 1.5 15 0.15 0.33 0.0015 0.1 220 0.5 1
Plot 0
Plot 1
Plot 2
velocity [m/s]
acceleration [m/s2]
space [m]
curent intensity [A]
Cur 0 3.64 -0.0
Cur 1 0.00 -0.0
Cur 0 3.61 -0.9
Cur 1 -0.00 -0.9
Cur 0 -1.61 -1.5
-0.04 -1.5
Bode A(frecuency)
Bode fi (frecuency)
moment [daNm]

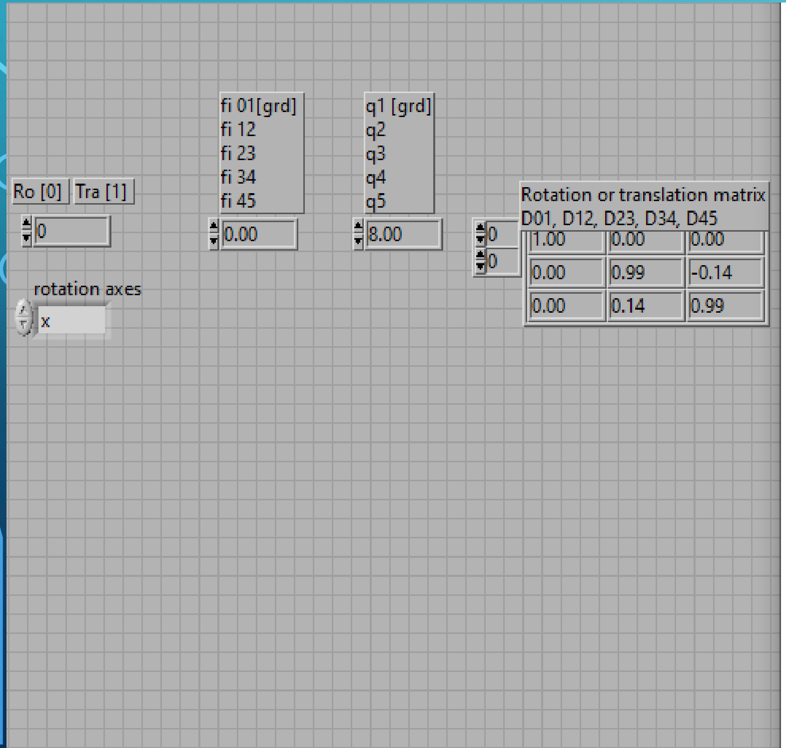

km ke Jr Ra La t U hp Mr
0 3.9000 15.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
0 1.5000 15.0000 0.1500 0.3300 0.0025 0.1000 180.0000 0.5000 1.0000
0 1.5000 15.0000 0.1500 0.3300 0.0035 0.1000 180.0000 0.5000 1.0000
file rouws
Plot 0
Plot 1
Plot 2
8 2.8 15 0.15 0.33 0.0015 0.1 180 0.5 1
9 3.9 15 0.15 0.33 0.0015 0.1 180 0.5 1
10 1.5 15 0.15 0.33 0.0025 0.1 180 0.5 1
11 1.5 15 0.15 0.33 0.0035 0.1 180 0.5 1
12 1.5 15 0.15 0.33 0.0015 0.1 210 0.5 1
13 1.5 15 0.15 0.33 0.0015 0.1 220 0.5 1
velocity [m/s]
acceleration [m/s2]
space [m]
curent intensity [A]
Cur 0 3.64 -0.0
Cur 1 0.00 -0.0
Cur 0 3.61 -0.9
Cur 1 -0.00 -0.9
Cur 0 -1.61 -1.5
-0.04 -1.5
Bode A(frecuency)
Bode fi (frecuency)
moment [daNm]

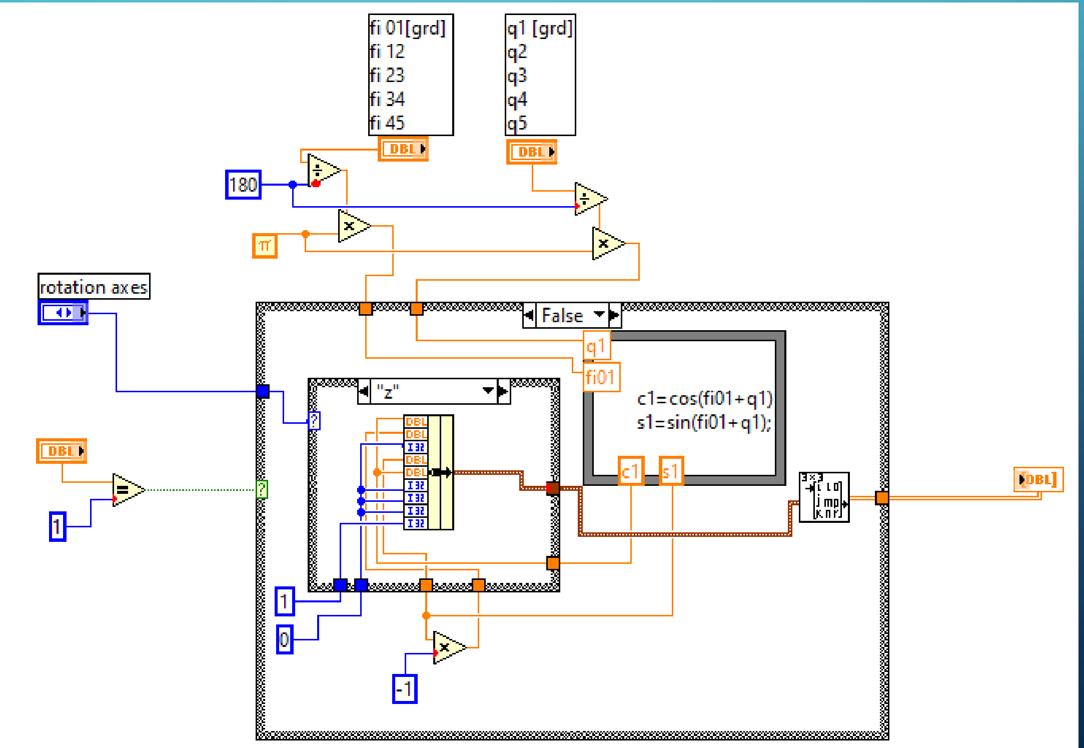

km ke Jr Ra La t U hp Mr
1.5000 15.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
1.5000 15.0000 0.5500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
3.9000 15.0000 0.1500 0.3300 0.0015 0.1000 180.0000 0.5000 1.0000
file rouws
9 3.9 15 0.15 0.33 0.0015 0.1 180 0.5 1
10 1.5 15 0.15 0.33 0.0025 0.1 180 0.5 1
11 1.5 15 0.15 0.33 0.0035 0.1 180 0.5 1
12 1.5 15 0.15 0.33 0.0015 0.1 210 0.5 1
13 1.5 15 0.15 0.33 0.0015 0.1 220 0.5 1
Plot 0
Plot 1
Plot 2
velocity [m/s]
acceleration [m/s2]
space [m]
curent intensity [A]
Cur 0
Cur 1
Bode A(frecuency)
Bode fi (frecuency)
moment [daNm]

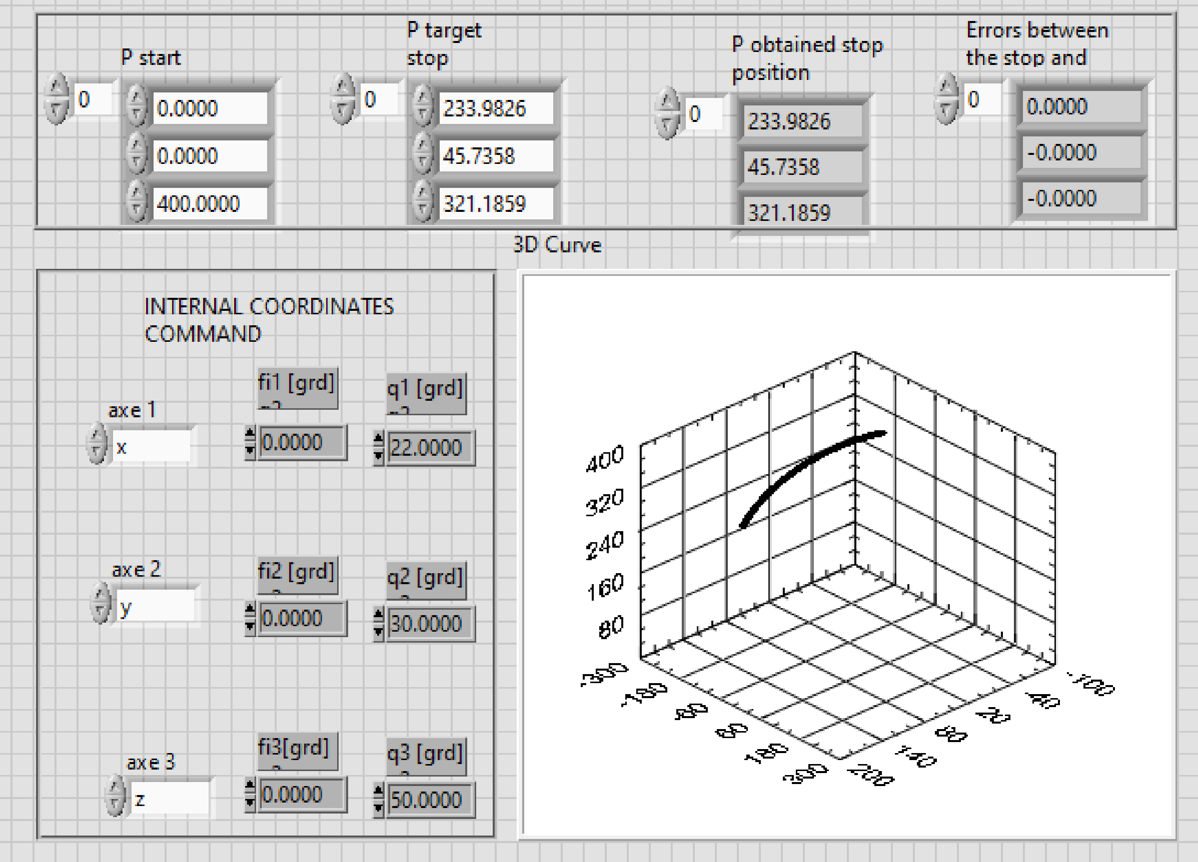
(ii). Assisted analyse of the complex coupled algorithm with Iterative Pseudoinverse Jacobian matrix method with proper neural network

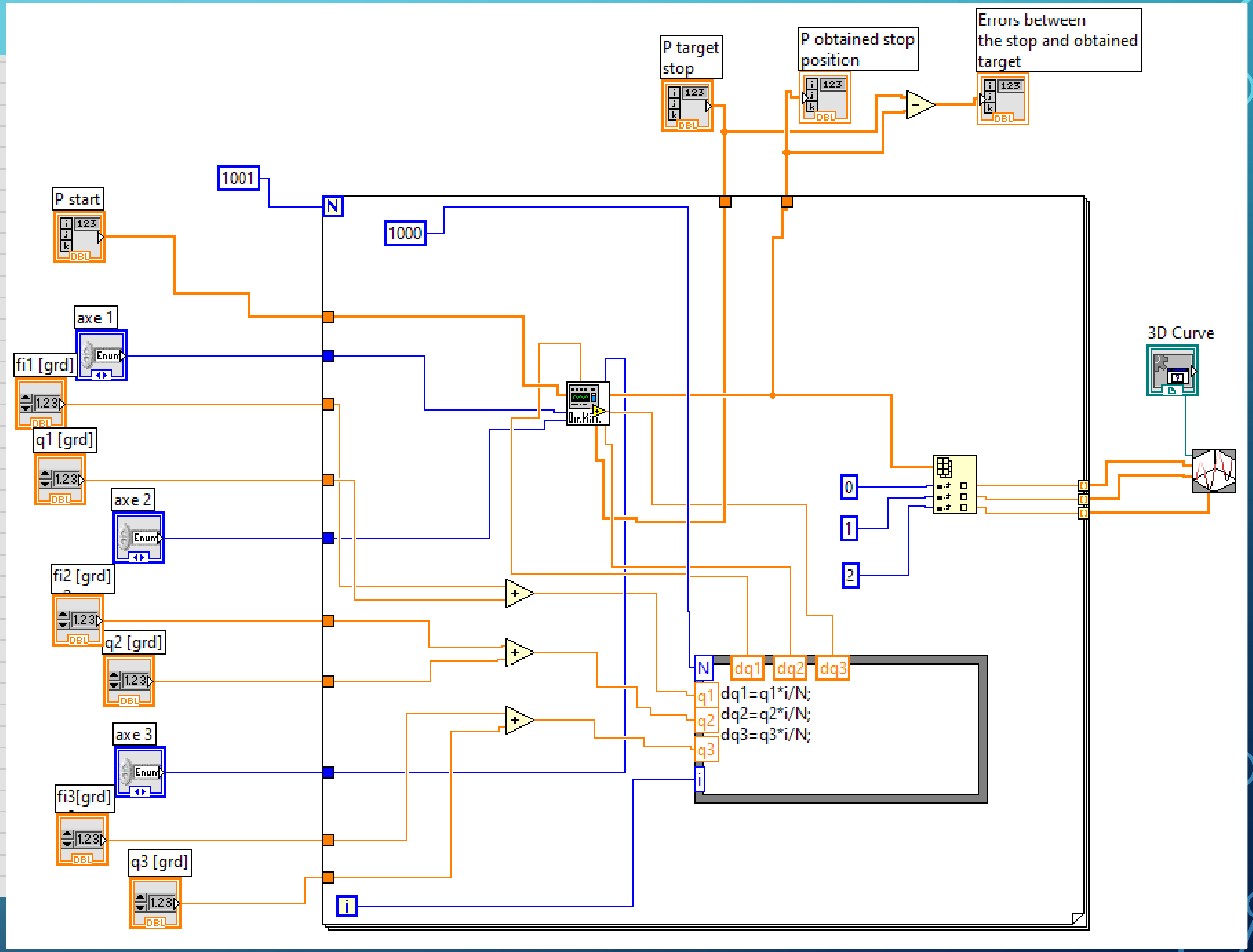


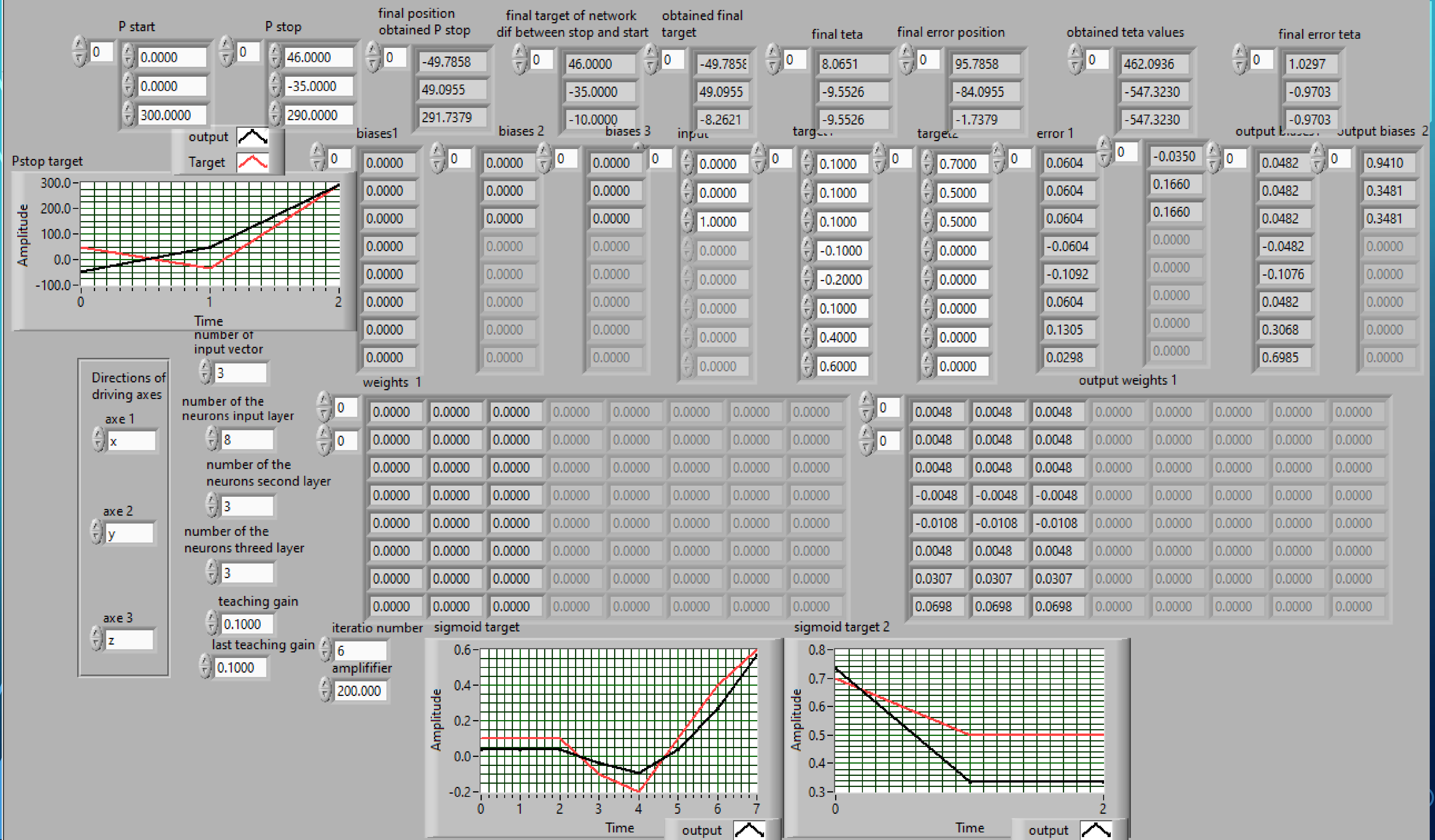




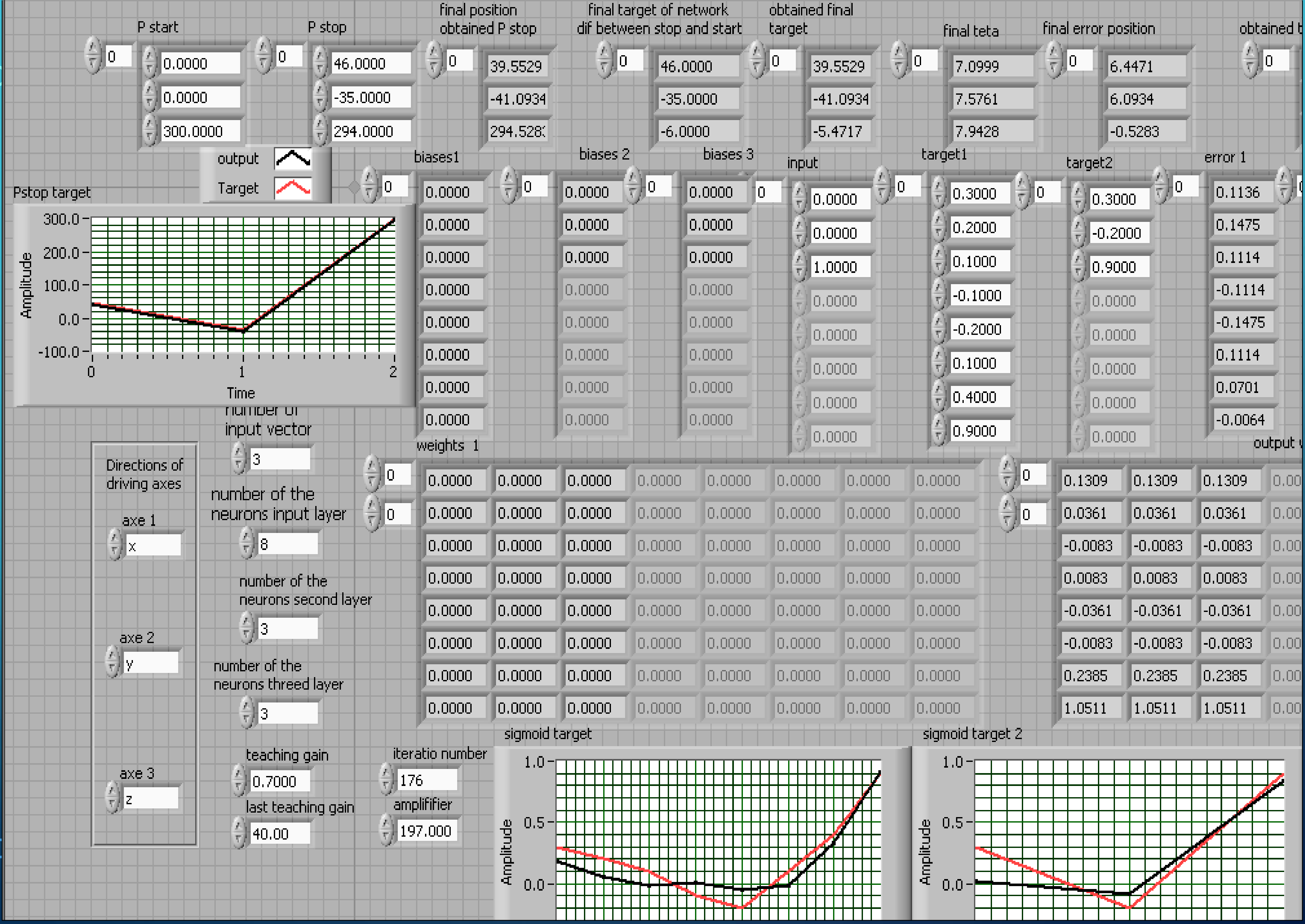



P start
0
0.0000
0.0000
400.0000
P target stop
0
233.9826
45.7358
321.1859
P obtained stop position
0
233.9826
45.7358
321.1859
Errors between the stop and
0
0.0000
-0.0000
-0.0000
3D Curve
INTERNAL COORDINATES COMMAND
axe 1
x
fi1 [grd]
0.0000
q1 [grd]
22.0000
axe 2
y
fi2 [grd]
0.0000
q2 [grd]
30.0000
axe 3
z
fi3[grd]
0.0000
q3 [grd]
50.0000
400
320
240
160
80

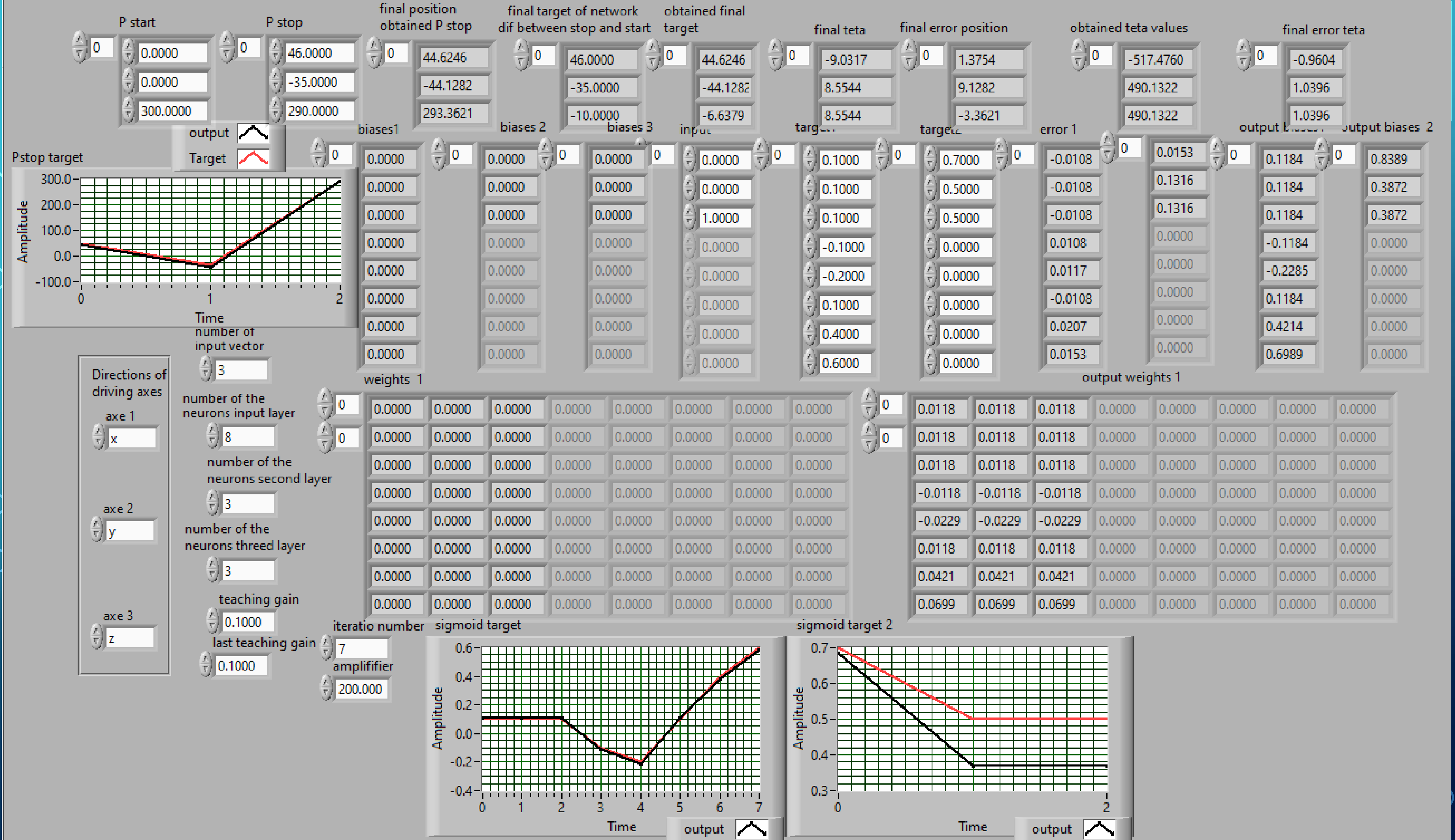

P target stop
P obtained stop position
Errors between the stop and obtained target
P start
1001
N
1000
axe 1
fi1 [grd]
q1 [grd]
axe 2
fi2 [grd]
q2 [grd]
axe 3
fi3[grd]
q3 [grd]
3D Curve
0
1
2
N
dq1
dq2
dq3
q1
q2
q3
dq1=q1*i/N;
dq2=q2*i/N;
dq3=q3*i/N;
i



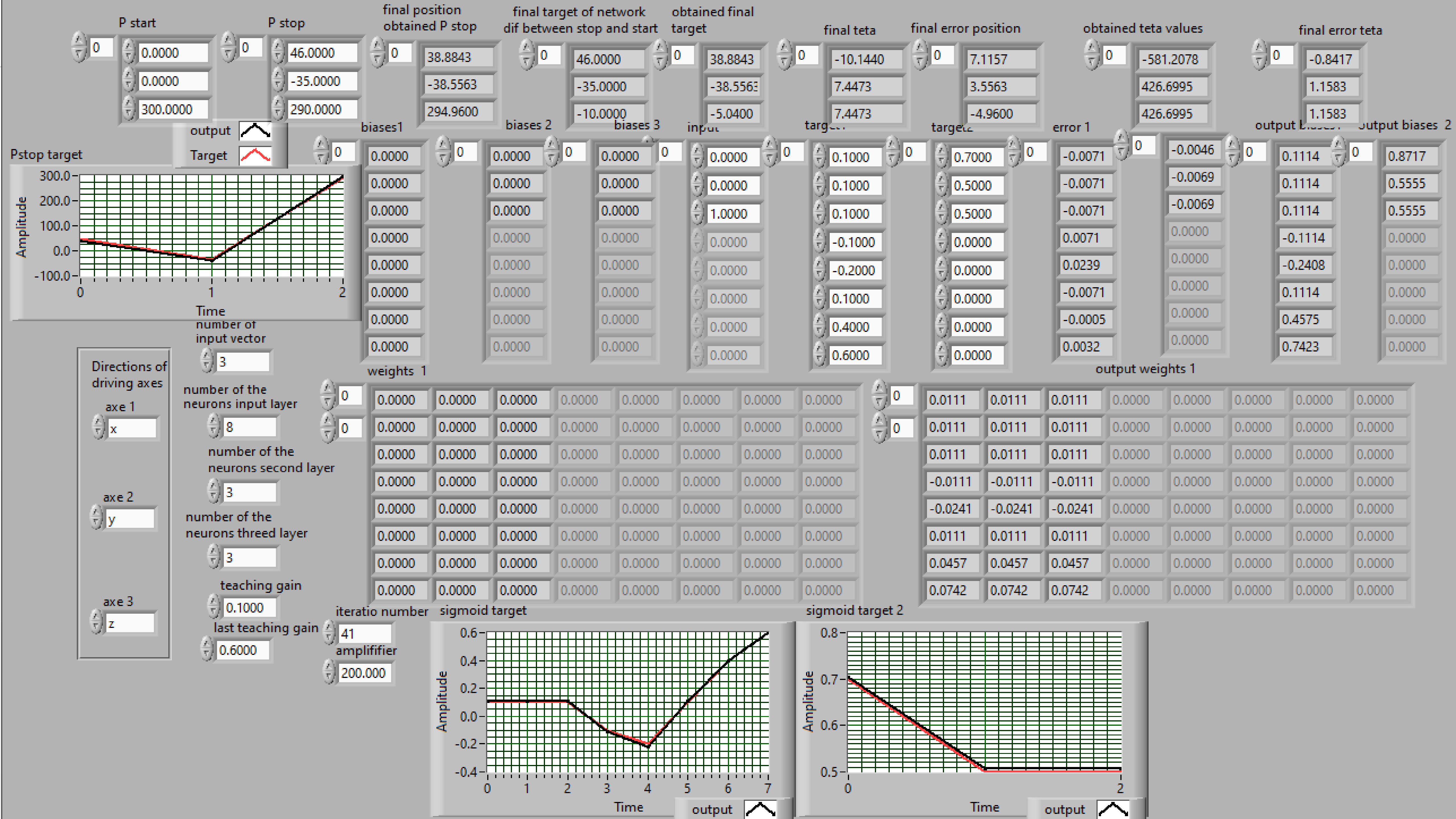
P start
0.0000
0.0000
300.0000
P stop
46.0000
-35.0000
290.0000
final position obtained P stop
-49.7858
49.0955
291.7379
final target of network dif between stop and start
46.0000
-35.0000
-10.0000
obtained final target
-49.7858
49.0955
-8.2621
final teta
8.0651
-9.5526
-9.5526
final error position
95.7858
-84.0955
-1.7379
obtained teta values
462.0936
-547.3230
-547.3230
final error teta
1.0297
-0.9703
-0.9703
Pstop target
output
Target
Amplitude
Time
biases1
biases 2
biases 3
input
0.0000
0.0000
1.0000
0.1000
0.1000
0.1000
-0.1000
-0.2000
0.1000
0.4000
0.6000
0.7000
0.5000
0.5000
error 1
0.0604
0.0604
0.0604
-0.0604
-0.1092
0.0604
0.1305
0.0298
-0.0350
0.1660
0.1660
0.0482
0.0482
0.0482
-0.0482
-0.1076
0.0482
0.3068
0.6985
0.9410
0.3481
0.3481
number of input vector
3
Directions of driving axes
axe 1
x
axe 2
y
axe 3
z
number of the neurons input layer
8
number of the neurons second layer
3
number of the neurons threed layer
3
teaching gain
0.1000
last teaching gain
0.1000
weights 1
output weights 1
0.0048
-0.0048
-0.0108
0.0307
0.0698
iteratio number
6
amplifier
200.000
sigmoid target
sigmoid target 2
output



P start
0
0.0000
0.0000
300.0000
P stop
0
46.0000
-35.0000
294.0000
final position obtained P stop
0
39.5529
-41.0934
294.528
final target of network dif between stop and start
0
46.0000
-35.0000
-6.0000
obtained final target
0
39.5529
-41.0934
-5.4717
final teta
0
7.0999
7.5761
7.9428
final error position
0
6.4471
6.0934
-0.5283
obtained
0
output
Target
Pstop target
300.0
200.0
100.0
0.0
-100.0
Amplitude
0
1
2
Time
biases1
0
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
biases 2
0
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
biases 3
0
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
input
0
0.0000
0.0000
1.0000
0.0000
0.0000
0.0000
0.0000
0.0000
target1
0
0.3000
0.2000
0.1000
-0.1000
-0.2000
0.1000
0.4000
0.9000
target2
0
0.3000
-0.2000
0.9000
0.0000
0.0000
0.0000
0.0000
0.0000
error 1
0
0.1136
0.1475
0.1114
-0.1114
-0.1475
0.1114
0.0701
-0.0064
number of input vector
3
Directions of driving axes
axe 1
x
axe 2
y
axe 3
z
number of the neurons input layer
8
number of the neurons second layer
3
number of the neurons threed layer
3
teaching gain
0.7000
last teaching gain
40.00
iteratio number
176
amplifiier
197.000
weights 1
0
0
output
0
0
0.1309 0.1309 0.1309
0.0361 0.0361 0.0361
-0.0083 -0.0083 -0.0083
0.0083 0.0083 0.0083
-0.0361 -0.0361 -0.0361
-0.0083 -0.0083 -0.0083
0.2385 0.2385 0.2385
1.0511 1.0511 1.0511
sigmoid target
1.0
0.5
0.0
Amplitude
sigmoid target 2
1.0
0.5
0.0
Amplitude



P start
0
0.0000
0.0000
300.0000
P stop
0
46.0000
-35.0000
290.0000
final position obtained P stop
0
44.6246
-44.1282
293.3621
final target of network dif between stop and start
0
46.0000
-35.0000
-10.0000
obtained final target
0
44.6246
-44.1282
-6.6379
final teta
0
-9.0317
8.5544
8.5544
final error position
0
1.3754
9.1282
-3.3621
obtained teta values
0
-517.4760
490.1322
490.1322
final error teta
0
-0.9604
1.0396
1.0396
output
Target
Pstop target
Amplitude
Time
biases1
biases 2
biases 3
input
error 1
output biases 2
number of input vector
3
Directions of driving axes
axe 1
x
axe 2
y
axe 3
z
number of the neurons input layer
8
number of the neurons second layer
3
number of the neurons threed layer
3
teaching gain
0.1000
last teaching gain
0.1000
iteratio number
7
amplifiier
200.000
weights 1
output weights 1
sigmoid target
sigmoid target 2



P start
0.0000
0.0000
300.0000
P stop
46.0000
-35.0000
290.0000
final position obtained P stop
38.8843
-38.5563
294.9600
final target of network dif between stop and start
46.0000
-35.0000
-10.0000
obtained final target
38.8843
-38.5563
-5.0400
final teta
-10.1440
7.4473
7.4473
final error position
7.1157
3.5563
-4.9600
obtained teta values
-581.2078
426.6995
426.6995
final error teta
-0.8417
1.1583
1.1583
output
Target
Pstop target
Amplitude
Time
biases1
biases 2
biases 3
input
0.0000
0.0000
1.0000
target2
0.7000
0.5000
0.5000
error 1
-0.0071
-0.0071
-0.0071
0.0071
0.0239
-0.0071
-0.0005
0.0032
output biases 2
0.8717
0.5555
0.5555
number of input vector
3
Directions of driving axes
axe 1
x
axe 2
y
axe 3
z
number of the neurons input layer
8
number of the neurons second layer
3
number of the neurons threed layer
3
teaching gain
0.1000
last teaching gain
0.6000
iteratio number
41
amplifier
200.000
weights 1
output weights 1
sigmoid target
sigmoid target 2

Teta 1-3 [grd]

22.0000
30.0000
50.0000

Jx1	Jx2	Jx3
216.211	100.112	-225.772

Jy1	Jy2	Jy3
-232.276	146.567	291.153

Jz1	Jz2	Jz3
-67.4745	-360.101	0.00000

Pstart

233.983
45.7358
321.186

Jacobian

216.211	100.112	-225.772
-232.276	146.567	291.153
-67.4745	-360.101	0.00000

Transpose

216.211	-232.276	-67.4745
100.112	146.567	-360.101
-225.772	291.153	0.00000

J x J TRANSP

107742	-101281	-50639.2
-101281	160204	-37106.1
-50639.2	-37106.1	134225

INV (J x J TRANSP)

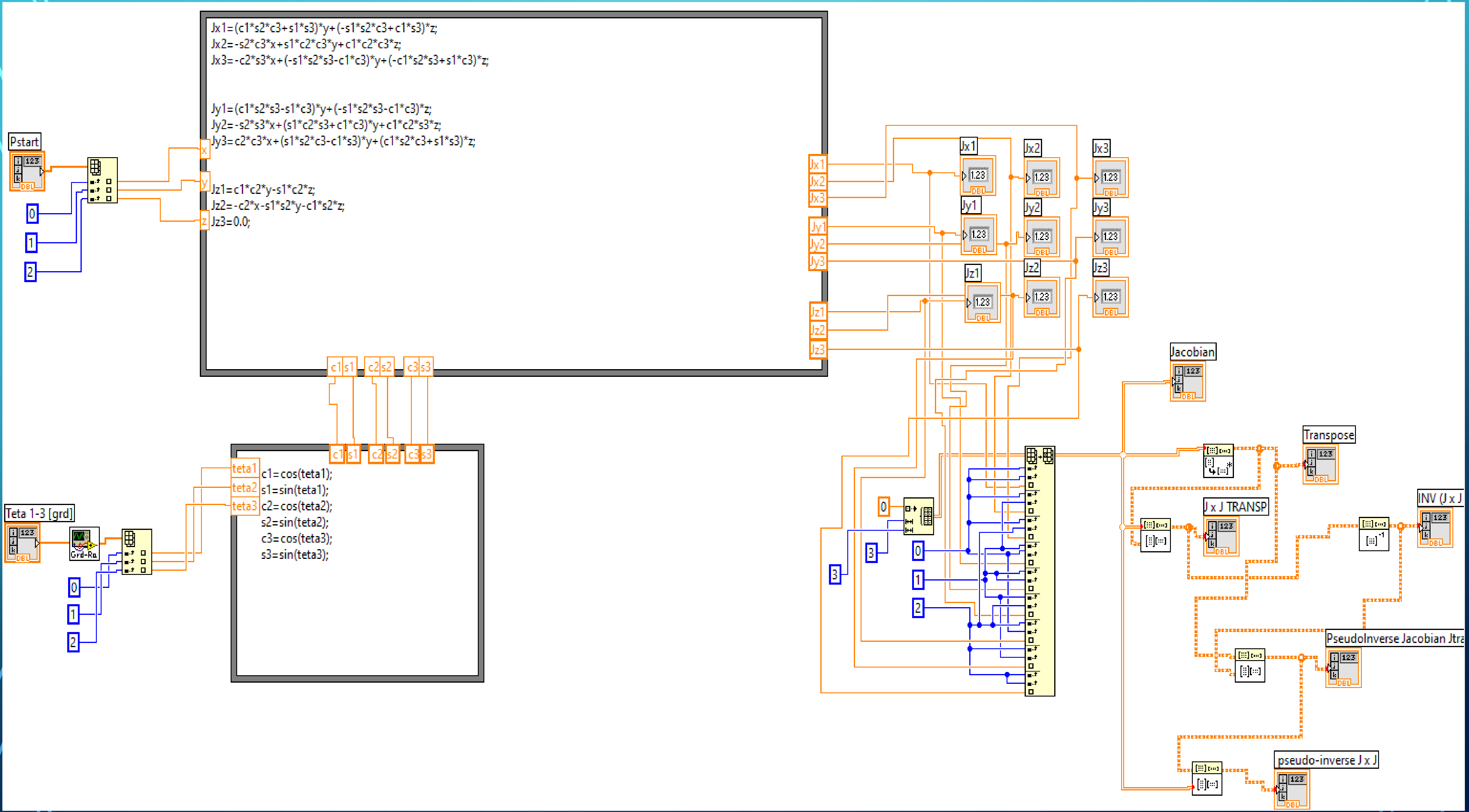
0.116727	0.089741	0.068846
0.089741	0.069000	0.052931
0.068846	0.052931	0.040614

PseudoInverse Jacobian Jtran x Inv(J x Jtran)

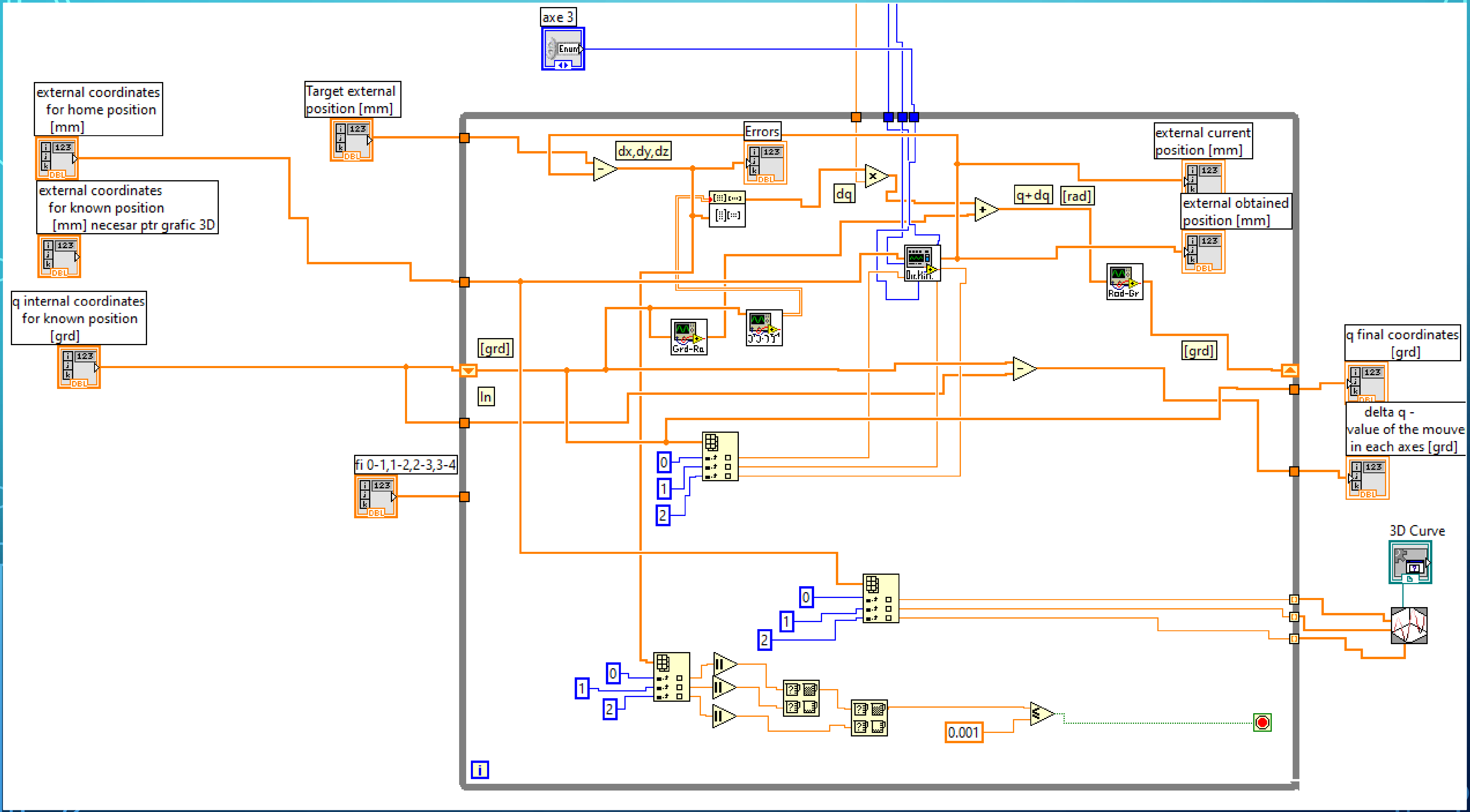
-0.252491	-0.195792	-0.149886
0.047311	0.036686	0.025308
-0.225249	-0.171233	-0.132317

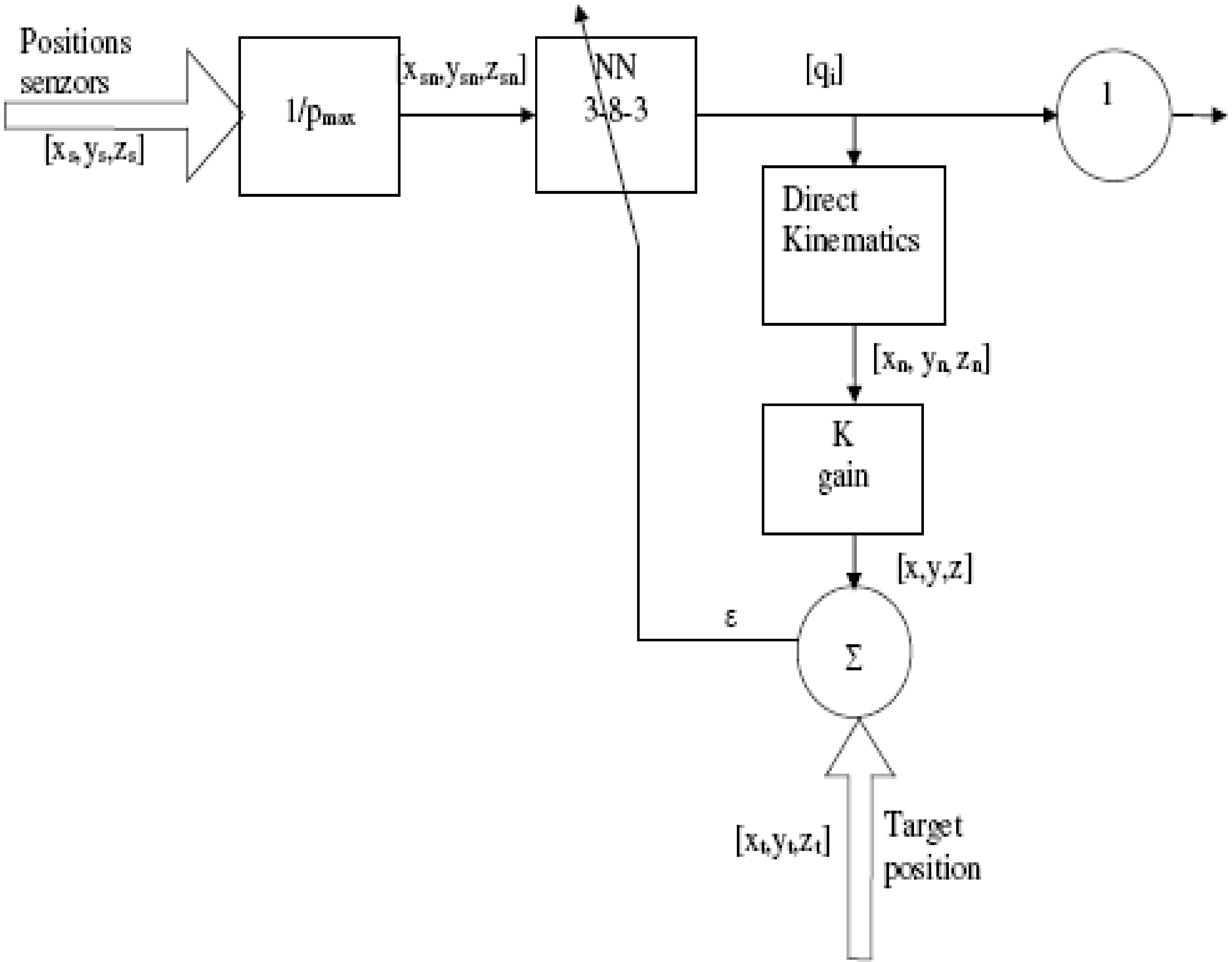
pseudo-inverse J x J

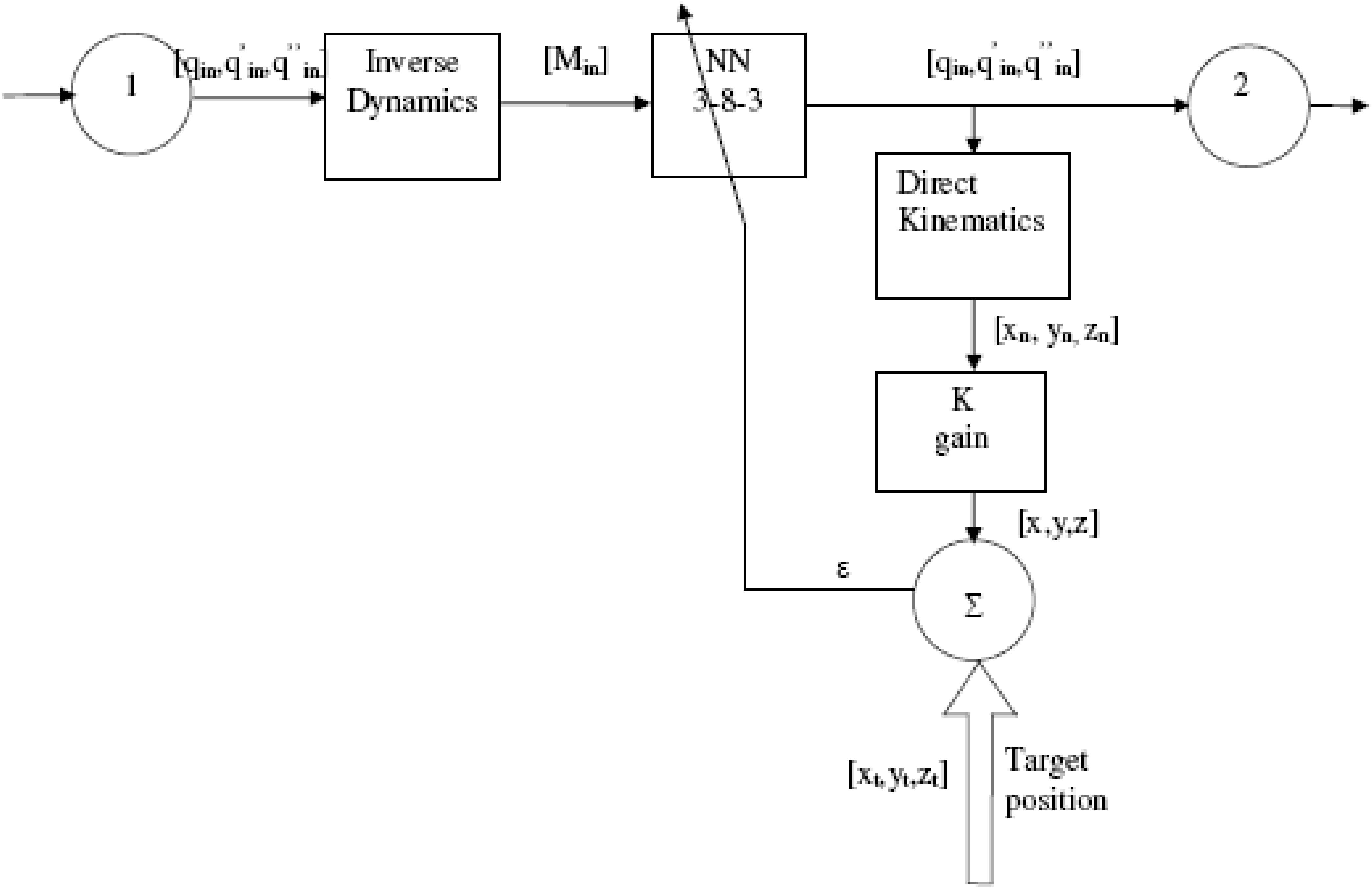
1.00000	8.81073E-	4.12115E-
-8.47100E	1.00000	1.04095E-
-6.66134E	-6.53699E	1.00000

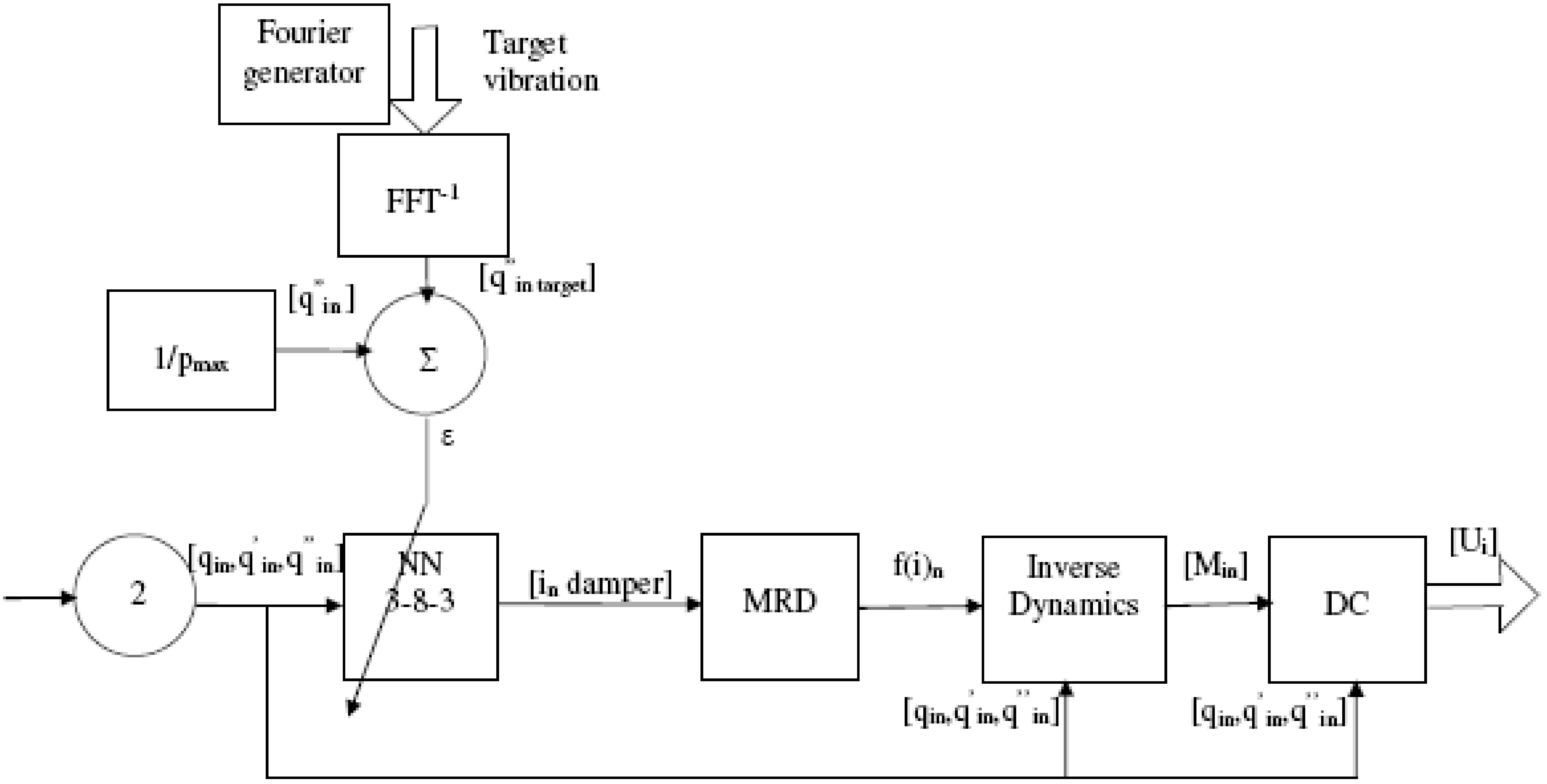


Jx1=(c1*s2*c3+s1*s3)*y+(-s1*s2*c3+c1*s3)*z;
Jx2=-s2*c3*x+s1*c2*c3*y+c1*c2*c3*z;
Jx3=-c2*s3*x+(-s1*s2*s3-c1*c3)*y+(-c1*s2*s3+s1*c3)*z;
Jy1=(c1*s2*s3-s1*c3)*y+(-s1*s2*s3-c1*c3)*z;
Jy2=-s2*s3*x+(s1*c2*s3+c1*c3)*y+c1*c2*s3*z;
Jy3=c2*c3*x+(s1*s2*c3-c1*s3)*y+(c1*s2*c3+s1*s3)*z;
Jz1=c1*c2*y-s1*c2*z;
Jz2=-c2*x-s1*s2*y-c1*s2*z;
Jz3=0.0;
c1=cos(teta1);
s1=sin(teta1);
c2=cos(teta2);
s2=sin(teta2);
c3=cos(teta3);
s3=sin(teta3);
Pstart
Teta 1-3 [grd]
Grd-Ra
teta1
teta2
teta3
Jx1
Jx2
Jx3
Jy1
Jy2
Jy3
Jz1
Jz2
Jz3
Jacobian
Transpose
J x J TRANSP
INV (J x J
PseudoInverse Jacobian Jtra
pseudo-inverse J x J

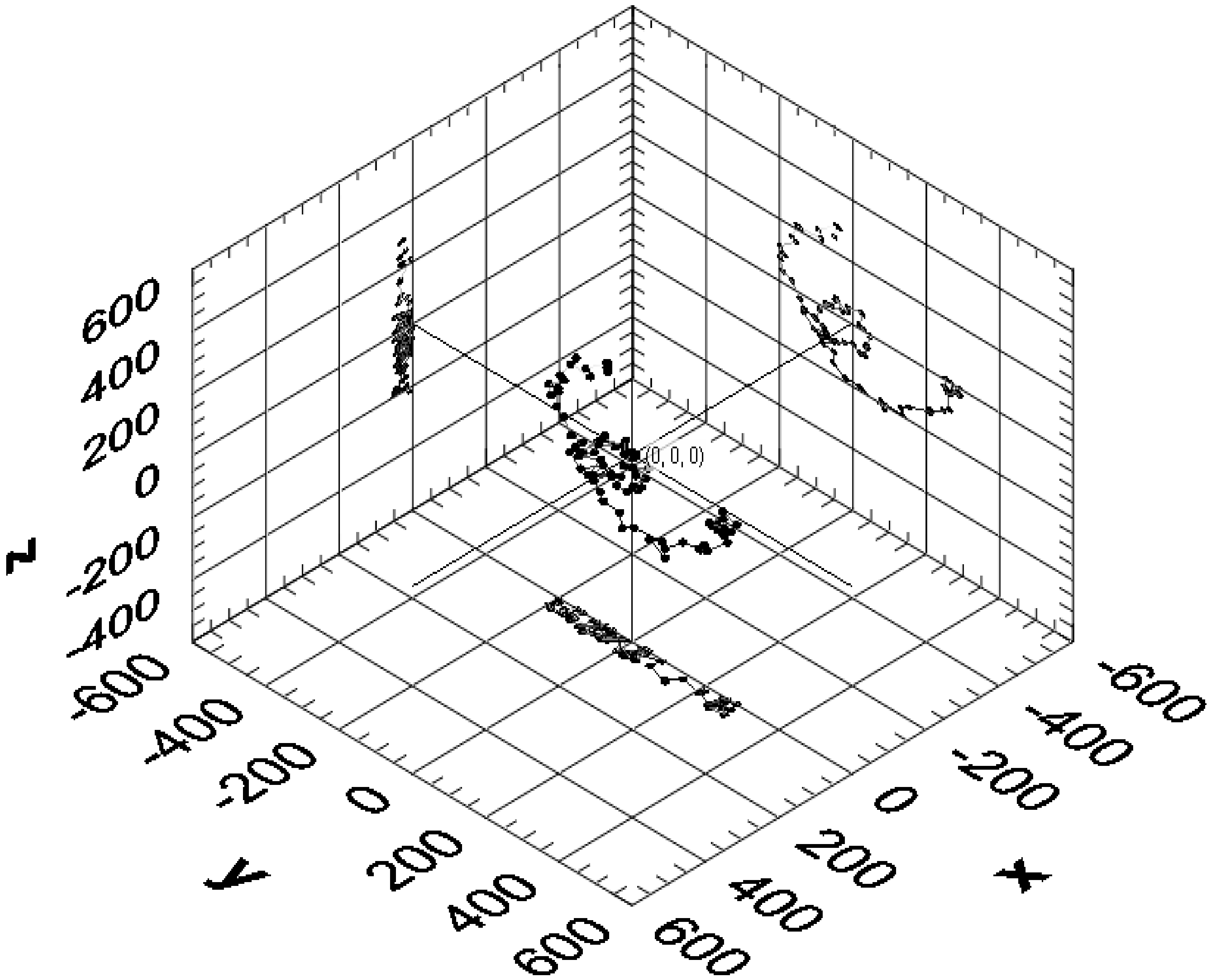

axe 3
external coordinates for home position [mm]
Target external position [mm]
external coordinates for known position [mm] necesar ptr grafic 3D
q internal coordinates for known position [grd]
fi 0-1,1-2,2-3,3-4
Errors
dx,dy,dz
dq
q+dq
[rad]
external current position [mm]
external obtained position [mm]
[grd]
In
Grd-Ra
Dir.Kin.
Rad-Gr
q final coordinates [grd]
delta q - value of the mouve in each axes [grd]
3D Curve
0.001


Positions
senzors
[xs,ys,zs]
1/pmax
[xsn,ysn,zsn]
NN
3-8-3
[qi]
1
Direct
Kinematics
[xn, yn, zn]
K
gain
[x,y,z]
ε
Σ
[xt,yt,zt]
Target
position

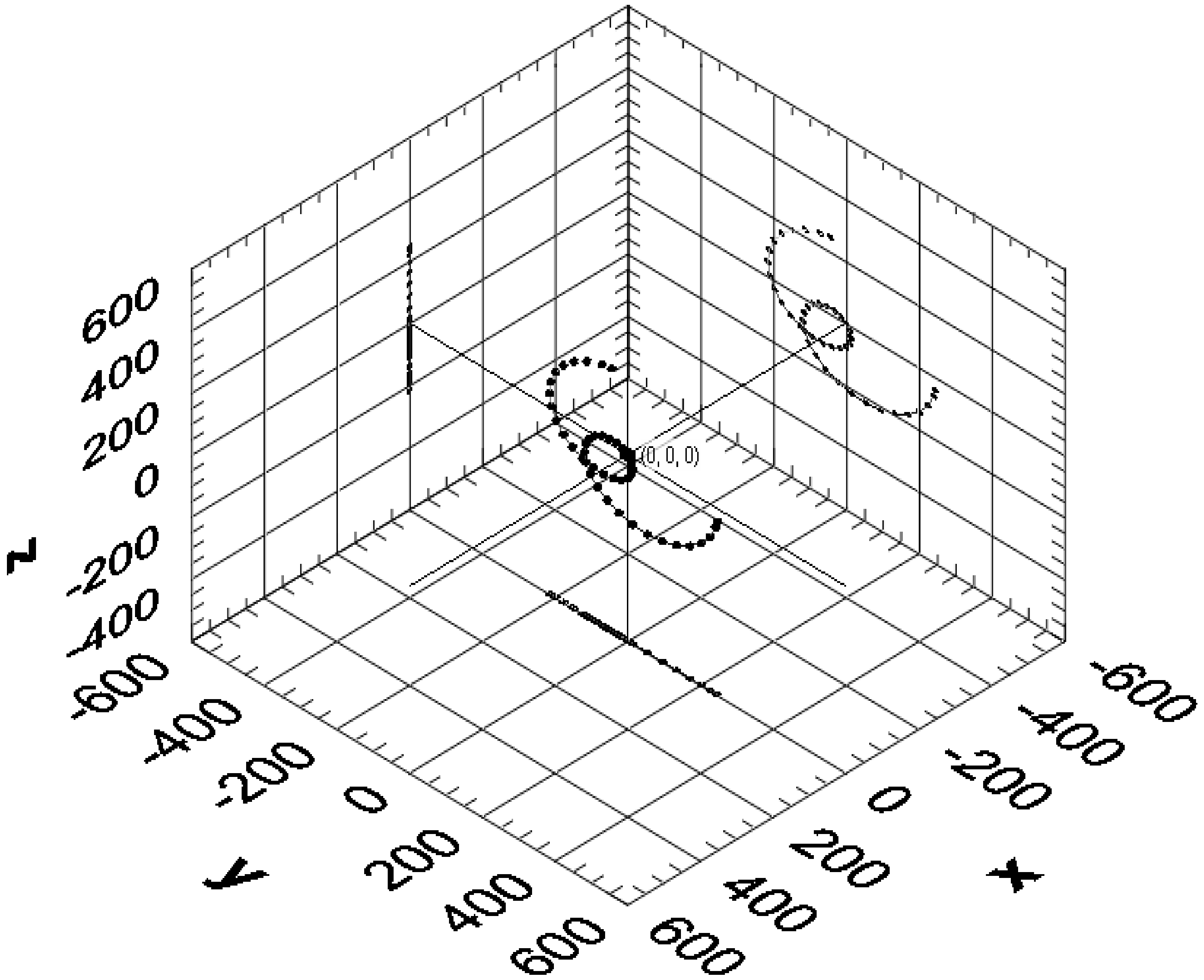

1
[qin,q'in,q''in]
Inverse Dynamics
[Min]
NN 3-8-3
[qin,q'in,q''in]
2
Direct Kinematics
[xn, yn, zn]
K gain
[x,y,z]
ε
Σ
[xt,yt,zt]
Target position



Fourier generator
Target vibration
FFT-1
[q'' in target]
[q'' in]
1/pmax
Σ
ε
2
[qin,q' in,q'' in]
NN 3-8-3
[in damper]
MRD
f(i)n
Inverse Dynamics
[Min]
DC
[Ui]
[qin,q' in,q'' in]
[qin,q' in,q'' in]



600
400
200
0
-200
-400
Z
-600
-400
-200
0
200
400
600
Y
-600
-400
-200
0
200
400
600
X
(0, 0, 0)


(0, 0, 0)
Z
600
400
200
0
-200
-400
-600
-400
-200
0
200
400
600
Y
-600
-400
-200
0
200
400
600
X

8. ANALYSE OF THE OBTAINED RESULTS

(i). Assisted analyse of the servo driving with DC motor and three stablized inertial wheels

From this research way the more important parameters to obtain the best dynamic answer of servo driving process are:

(i) the motor parameters k_e and k_m ;

(ii) the inertial reduced inertial tensor to servo motors, J_{red} ;

(iii) the electrical resistance of the motor R_a ;

(iv) the electrical inductance L_a ;

(ii).Assisted analyse of the complex coupled algorithm with Iterative Pseudoinverse Jacobian matrix method with proper neural network

From this research way the more important parameters to obtain the quickly convergence process and the maximal precision are:

(i) the amplifier gain of the hyperbolic tangent sensitive function;

(ii) the different teaching gain in each layer;

(iii) the intermediary target data of the hidden layer;

(iv) the step of the time delay and the position of them inside of the block schema;

(v) alternative use of the time delay in even or odd time;

(vi) the forward links and the position of them in the used closed loops.

8. CONCLUSIONS

- This research open the way to establish quickly, by using the LabVIEW software, the stability of the three inertial satellite system.
- The used method is very complex and requires to use the high-performance programs to solve problems related to forward and inverse kinematics (FK, IK) and direct and inverse dynamics (DD, ID) applied in the field of correcting the position in the space of the satellites.
- The obtained results confirm the veracity of the used method, the designed programs being able to be used in various applications that require extreme precision of the spatial trajectories.
- All results, the proper virtual instrumentation and the proposed algorithm can be applied in many other mechanical applications what imposed the maximal precisions in the space.

9. FUTURE WORK

- Apply all designed virtual LabView instrumentation in only one assisted platform what must be cover all calculus and optimisation of the kinematic and dynamic behavior of satellite;
- The new assisted platform for satellite will cover all needed calculus for servo driving, establishing the number of pulses for each motor, animation of the satellite in the space, optimising the used neural network;
- The new platform will perform all input data of the complex coupled method Pseudoinverse Jacobian Matrix Method (PIJMM) coupled with Bipolar Hyperbolic Tangent Neural Network with Time Delay and Recurrent Links (BSHTNN(TDRL));

10. REFERENCES

[1] Herbert J. Kramer & Arthur P. Cracknell (2008) An overview of small satellites in remote sensing,*International Journal of Remote Sensing,* 29:15, 4285-4337, DOI: 10.1080/01431160801914952

[2] Sweeting, M.N., 1991, Why satellites are scaling down. *Space Technology International*, pp. 55–59 [the journal was generally referred to as Space Technology, which has since been discontinued].

[3] Encyclopedia Britannica. *Science and Technology Illustrated*, Chicago, 1994.

[4] Guinn, Robert P. *The New Encyclopedia Britannica* Vol. 10. Chicago, 1994.

[5] Hagan, Brian J. "Voyagers Go Lite". *Design News*. 6 Sept. 1996: p 81, 88. [6] Knight-Riddler/Tribune Information Services. *Press-Telegram*. Long Beach, CA., 1996.

[7] MacLean, Hunter. Satellite Communications Services Industry, "A Satellite Out of Tune". Info track Magazine Index Plus. 8 April 1996.

[8] http://www.responsivespace.com/Papers/RS7/SESSIONS/Session%20V /5002_

[9] Dimith, H., Beale M.& Hagan M.: *Neural Network ToolboxTM 6- User Guide*, The MathWorks, Inc.3 Apple Hill Drive Natick, MA 01760-2098, USA.

[10] M. Minsky and S. Papert, *Perceptrons: an introduction to computational geometry,* The MIT Press, Cambridge MA, 1969.

[11] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. of the National Academy of Sciences,* 81, 1984, pp.3088-3092.

[12] Olaru, Adrian. "Optimize the Satellite Orientation by Using the Inertial Pulse Method, Intelligent Damper, Dynamics, Kinematics and Proper NeuralNetwork." *Applied Mechanics and Materials*, vol. 232, Trans Tech Publications, Ltd., Nov. 2012, pp. 665–673, Crossref, doi: 10.4028 /www.scientific.net /amm.232.665.

[13] Olaru, S., Oprean, A., Olaru, A. (2008). Assisted research of the new Bouc-Wen rheological damper, *Proceedings of OPTIROB 2008*, (Ed.) Olaru, A., pp.143- 152, ISBN 978-973-648-784-2, Predeal, may 2008, Bren, Bucharest.

[14] Olaru, A., Olaru, S. (2006). Research of the industrial robots viscose global dynamic damper coefficient with LabVIEW instrumentation, *Proceedings of CAX'2006,* (Ed.)Mikolajczyk T., pp. 73- 81, ISBN 978- 83-89334-49-7, Bydgoszcz, nov. 2006, Akademia Techniczno-Rolnicza, Bydgoszcz.

[15] Olaru, A., Olaru S., Ciupitu, L. Assisted research of the neural network by back propagation algorithm, OPTIROB 2010 International Conference, Calimanesti, Romania, *The RPS Singapore Book*, www.rpsonline.com , pp.194-200 , 2010.

[16] Olaru,A.,Olaru,S.,PauneD.,GhioneaA.Assisted research of the neural network, OPTIROB 2010 International Conference, Calimanesti, Romania, *The Research Publishing Services Singapore Book*, www.rpsonline.com , pp.189-194 , 2010.

[17] Olaru, A., Olaru, S. Assisted research of the neural network with LabVIEW instrumentation, *IEEE ICMENS-2010 Proceedings*, Changsha, China, pp.1-8, 2010.

THANK YOU FOR YOUR ATTENTION !

I hope to see you in Romania, in SLS&OPTIROB 2026 International Conference

www.uastro.space