Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

WHEEL MOBILE ROBOTS
LECTURE– 05 – IDENTIFICATION

dr inż. Tomasz Buratowski

Faculty of Mechanics and Robotics
Department of Robotics and Mechatronics

Identification of mobile wheeled robots

Identification – the best choice among class of models based on mathematical description of the physical phenomena occurred in analyzed object.

Methods of identification presented below will be discussed during this lecture:

- Fuzzy logic
- Genetic algorithms

Fuzzy Logic

Fuzzy set theory is used to describe phenomena and terms, which have ambiguous and imprecise character.

With the use of fuzzy sets we can describe imprecise and ambiguous terms such as “high speed”, “low speed”. In creation of fuzzy sets the most important is determination of so called “set of dissertation”.

Area of dissertation also called space or set, will be most often denoted as X. We should remember that set X is not fuzzy set.

Fuzzy logic

Fuzzy set A in some non-empty space X, what is denoted as $A \subseteq X$, is called sets of pairs:

$$A = \{(x, \mu_A(x)); x \in X\}$$

where: $\mu_A : X \rightarrow [0,1]$ is a membership function of fuzzy set A. This function assigns to each element of set its level of membership of fuzzy set A. We can distinguish three cases:

- Full membership of fuzzy set A of x element
- No membership of fuzzy set A of x element
- Partial membership of fuzzy set A of x element

Fuzzy logic

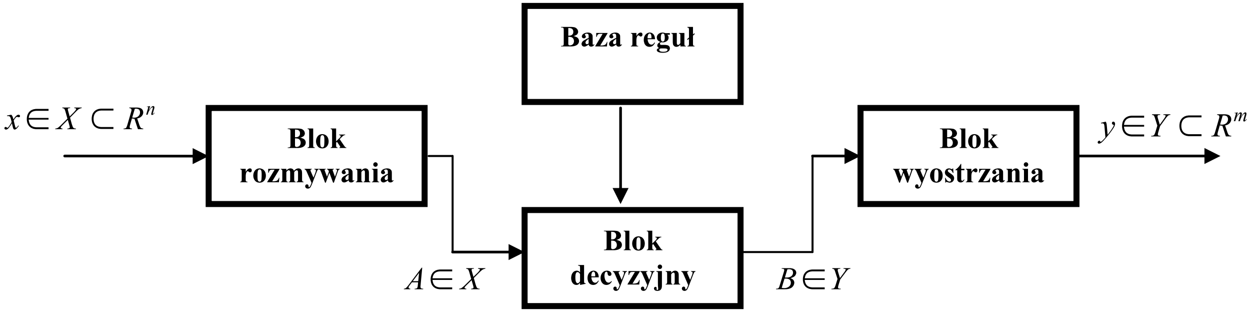
In systems with fuzzy logic we use symbolic expressions „IF-THEN”, qualitative variables described with the use of linguistic variables and fuzzy operators „AND”, so for example we can write:

IF x1 is low **AND** x2 is high **THEN** y is middle

Fuzzy logic

Scheme of system with fuzzy logic, in which we can distinguish four basic elements:

- Fuzzy block
- Decision block
- Rules database
- Sharpening block



Fuzzy logic

Basic rules of fuzzy modeling:

- Use of the linguistic variables instead of or in addition to numerical variables
- Assume relationships between variables based on conditional fuzzy sentences
- Linguistic description is made base on a priori knowledge of the system

Genetic algorithms

Idea of use of genetic algorithms is as follows: right algorithm processed by a computer can gives ways of solving difficult problems in a way as nature solves it – by evolution.

Aim of this algorithms is use of analogical mechanism to selection and reproduction which occur in the nature.

Genetic algorithms

Selection is a choice of chromosomes with the highest value (best adapted), which leans to reproduction more often than those with low value (poorly adapted).

Reproduction is making of new chromosomes as a result of recombination of genes of parental chromosomes.

Recombination is a process in which new combination of genes are created.

State-space

Based on Maggie's dynamics equations (with out castor/ idler wheel) we write them in form which allows further calculations to be easier:

$$\begin{aligned} a_2\ddot{\alpha}_2 + a_1\ddot{\alpha}_1 + a_3\dot{\alpha}_2^2 - a_3\dot{\alpha}_1\dot{\alpha}_2 &= M_1 - a_4\,\mathrm{sgn}(\dot{\alpha}_1) \\ a_2\ddot{\alpha}_1 + a_1\ddot{\alpha}_2 + a_3\dot{\alpha}_1^2 - a_3\dot{\alpha}_1\dot{\alpha}_2 &= M_2 - a_5\,\mathrm{sgn}(\dot{\alpha}_2) \end{aligned} \tag{5.1}$$

where:

$$a_1 = \frac{1}{4}\,\frac{r^2\, m_4\, l_1^{\,2} + r^2\, m_4\, l_2^{\,2} + r^2\, Iz_4 + 2\, r^2\, Ix_1 + 4\, r^2\, m_1\, l_1^{\,2} + 4\, Iz_1\, l_1^{\,2}}{l_1^{\,2}}$$

$$a_3 = \frac{1}{4}\,\frac{r^3\, m_4\, l_2}{l_1^{\,2}}$$

$$a_2 = \frac{1}{4}\,\frac{-r^2\, m_4\, l_2^{\,2} + r^2\, m_4\, l_1^{\,2} - r^2\, Iz_4 - 2\, r^2\, Ix_1}{l_1^{\,2}}$$

$$a_4 = N_1 f_1 \qquad a_5 = N_2 f_2$$

State-space

Assuming state variables as:

$$\alpha_1 = x_1, \dot{\alpha}_1 = \dot{x}_1 = x_2, \alpha_2 = x_3, \dot{\alpha}_2 = \dot{x}_3 = x_4$$

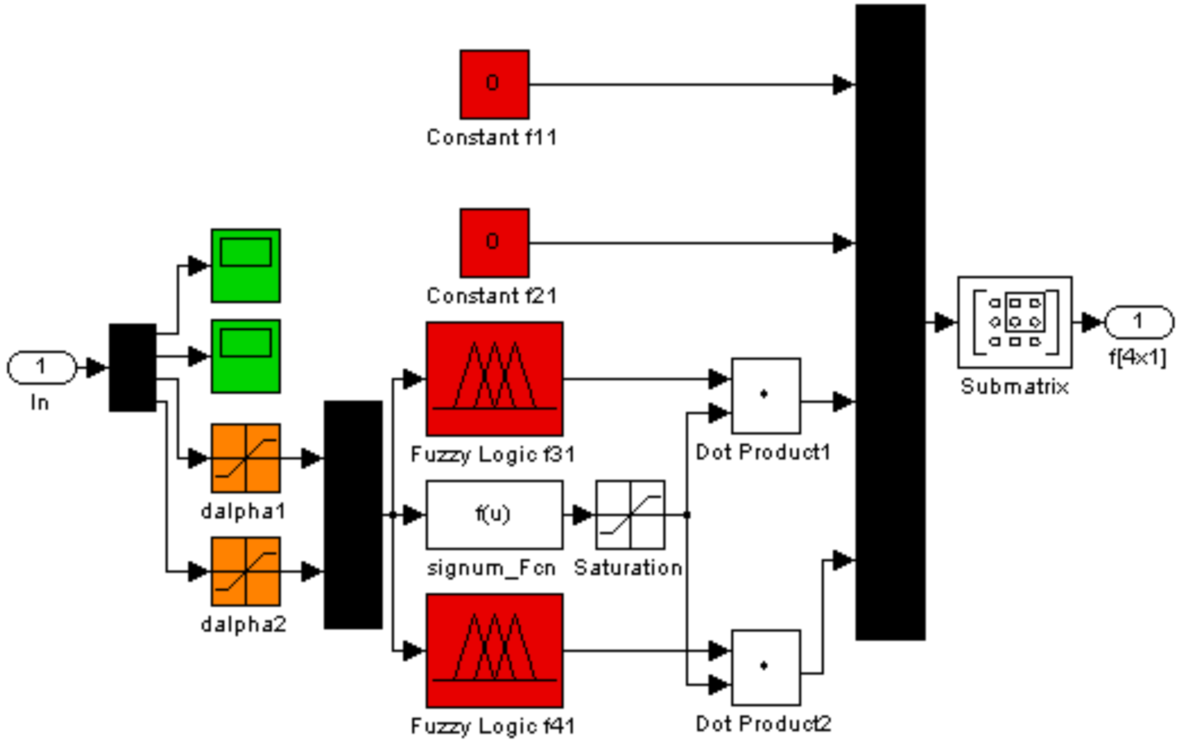
Dynamic equation of motion *(5.1)* can be presented in a form of a state matrix:

$$\dot{x} = Ax + B[f(x,a) + G(x,a)u] \qquad (5.2)$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$f(x,a) = \begin{bmatrix} 0 \\ 0 \\ \dfrac{a_2 a_5 \,\mathrm{sgn}(x_4) + a_2 a_3 x_2^2 - a_2 a_3 x_2 x_4 + a_1 a_3 x_2 x_4 - a_1 a_3 x_4^2 - a_1 a_4 \,\mathrm{sgn}(x_2)}{a_1^2 - a_2^2} \\ \dfrac{-a_1 a_5 \,\mathrm{sgn}(x_4) - a_1 a_3 x_2^2 + a_1 a_3 x_2 x_4 - a_2 a_3 x_2 x_4 + a_2 a_3 x_4^2 - a_2 a_4 \,\mathrm{sgn}(x_2)}{a_1^2 - a_2^2} \end{bmatrix} \qquad G(x,a) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{a_1}{a_1^2 - a_2^2} & -\dfrac{a_2}{a_1^2 - a_2^2} \\ -\dfrac{a_2}{a_1^2 - a_2^2} & \dfrac{a_1}{a_1^2 - a_2^2} \end{bmatrix}$$

Identification – fuzzy logic



Systems with fuzzy logic was used for approximation of non-linear functions and was modeled in a way presented above

Identification structures

Fuzyy Logic f31 and Fuzzy Logic f41 are fuzzy systems, which use numerical information connecting input and output signals in explicit way. Those systems only approximate properties of an object with the use of linguistic information.

In order to make identification of the system two type of identification of the parameters where proposed:

- Parallel
- Serial-parallel

Parallel structure

Identification system in parallel structure has form of

$$\dot{\hat{x}} = A\hat{x} + B[\hat{f}(\hat{x},\hat{a}) + G(a)u] + K\tilde{x} \qquad (5.3)$$

where $\hat{x}$ is an estimation of state vector x,

$\hat{f}(\hat{x},\hat{a})$ is estimation of non-linear function present in *(5.2)* equations

Parallel structure

Assuming estimation error of state vector in a form of

$\tilde{x} = x - \hat{x}$ and subtracting equation *(5.3)* from equation *(5.2)* we get description of the identification system in an error-space.

$$\dot{\tilde{x}} = A_H \tilde{x} + B[\tilde{f}(x, \hat{x}, a, \hat{a}) + G(a)u] \tag{5.4}$$

where: A_H=A-K, and K matrix is matched in such a way that characteristic equation of matrix A_H is strictly stable.

$$\tilde{f}(x, \hat{x}, a, \hat{a}) = f(x, a) - \hat{f}(\hat{x}, \hat{a})$$

Serial-parallel structures

Adding and subtracting equations *(5.2)* expression $A_m x$ where A_m is stable project matrix we got

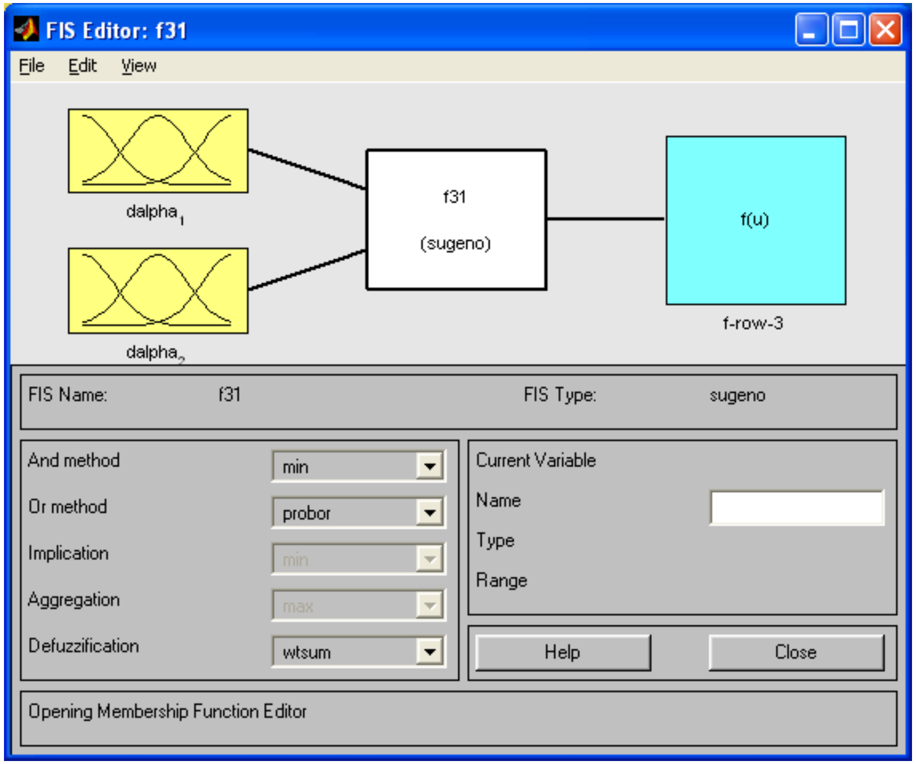
$$\dot{x} = A_m x + (A - A_m)x + B[f(x,a) + G(a)u] \qquad (5.5)$$

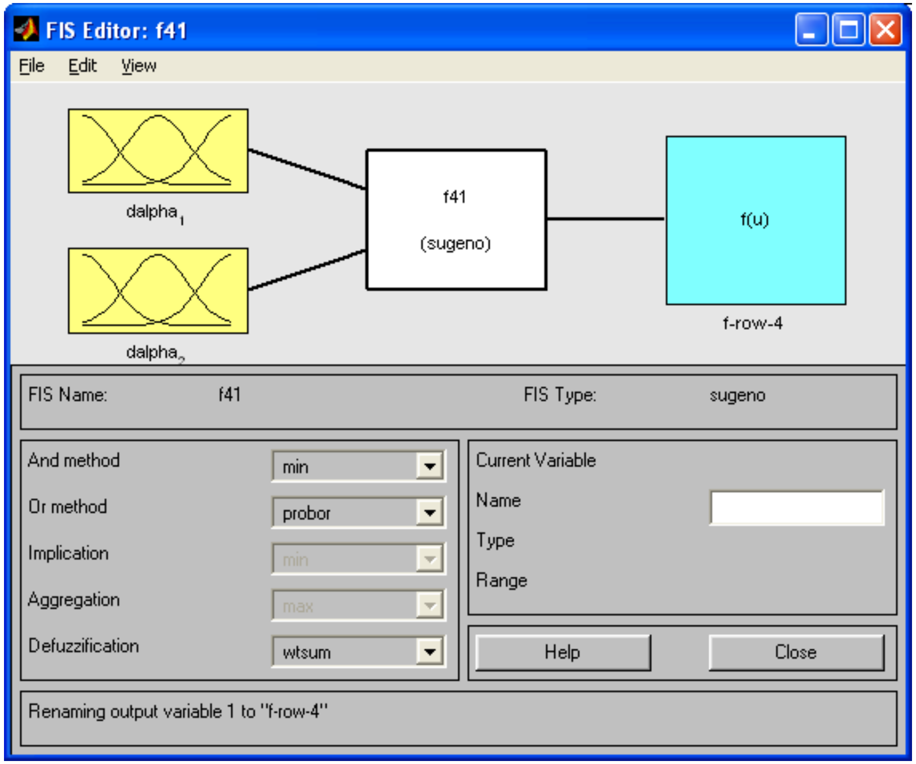
Equation *(5.5)* define serial-parallel identification structure, given in a form of:

$$\dot{\hat{x}} = A_m \hat{x} + (A - A_m)x + B[\hat{f}(x,\hat{a}) + G(a)u] \qquad (5.6)$$

Identification of mobile wheeled robots – modeling

To determine function $\hat{f}(\hat{x},\hat{a})$ there where used system with fuzzy logic created in Matlab software





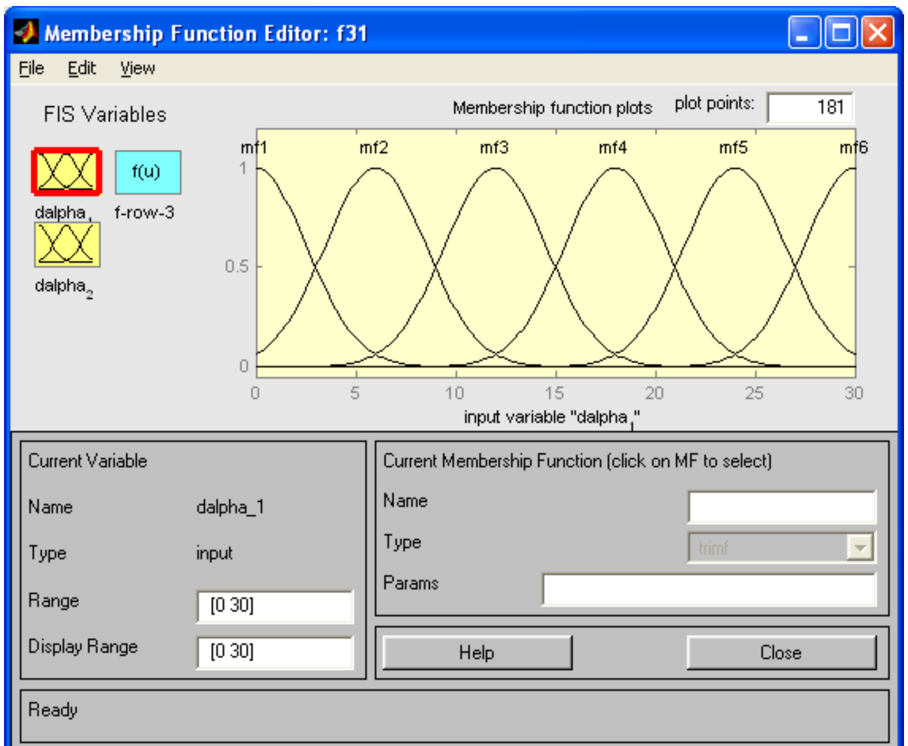
Identification of mobile wheeled robots – modeling

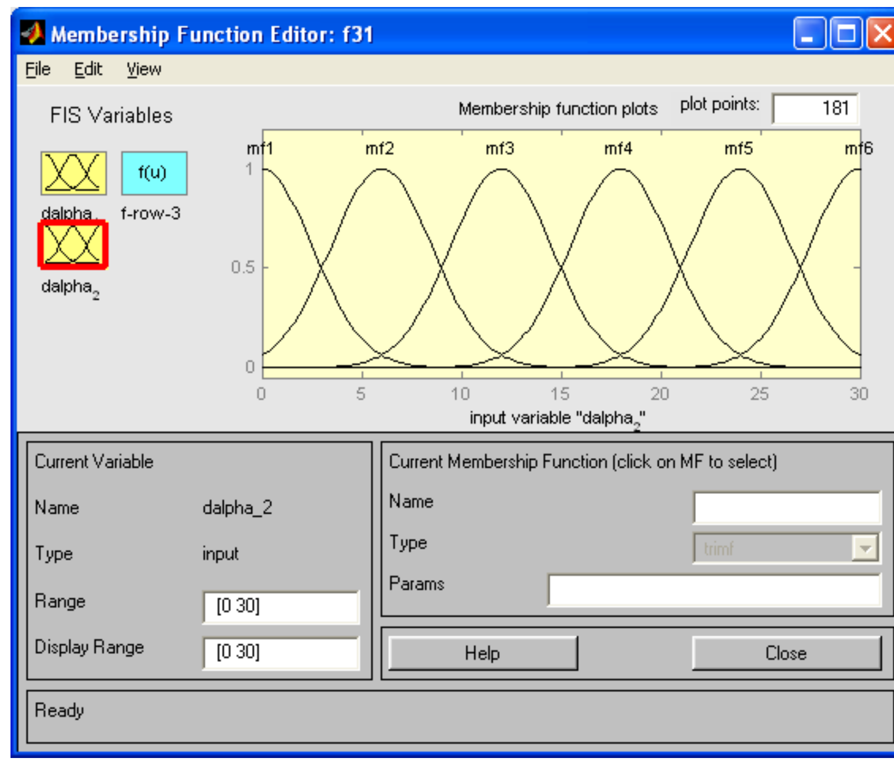
As for fuzzy model type we assumed Takagi-Sugeno.

Fuzzy block transforms input space in a form of $X = [\dot{\alpha}_{1a}, \dot{\alpha}_{1b}] \times [\dot{\alpha}_{2a}, \dot{\alpha}_{2b}] \subset R^n$ into fuzzy set $A \in X$ characterized with membership function $\mu_A(x) : X \to [0,1]$

Identification of mobile wheeled robots – modeling

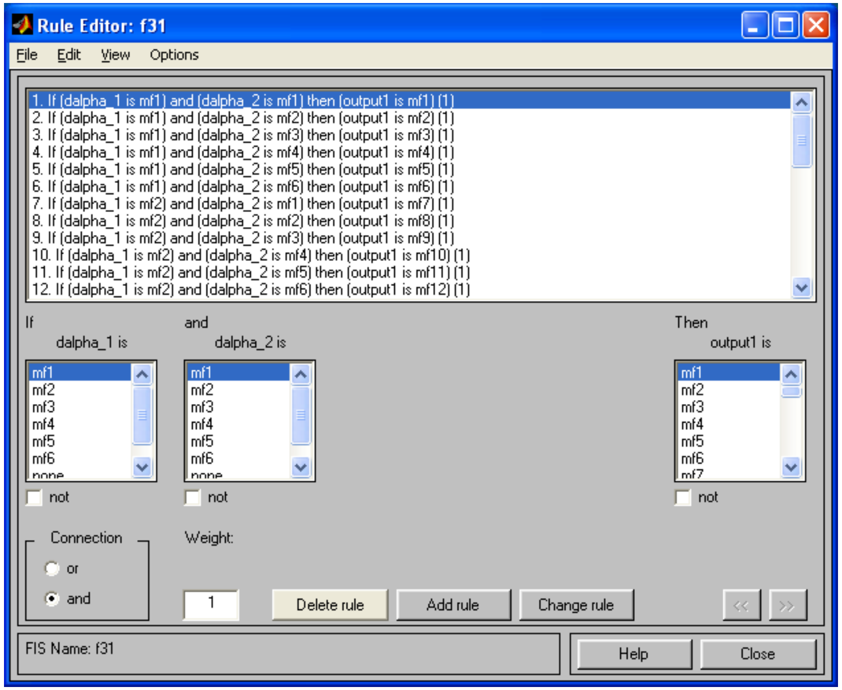
Membership functions with variability ranges of the inputs are presented below





Identification of mobile wheeled robots – modeling

Rules database is assumed as in the picture presented by. Due to the fact that we have 6 inputs to the fuzzy block the 36 rules was stated in a manner each input with every other one.



Identification of mobile wheeled robots – modeling

On input to the decision block we assumed fuzzy set A with T-norm of minimum type

$$\mu_{A_1^j \times \ldots \times A_n^j}(x) = \min[\mu_{A_1^j}, \ldots, \mu_{A_n^j}]$$

On output from the fuzzy model of Takagi-Sugeno type we get signal in a from of normalized weighted sum of individuals outputs $\bar{y}_j$

$$y(x) = \frac{\sum_{j=1}^{M} \bar{y}_j \tau_j}{\sum_{j=1}^{M} \tau_j}$$

where: $\tau_j = \prod_{i=1}^{n} \mu_{A_i^j}(x_i)$

is level of ignition of j-th rule

Identification of mobile wheeled robots – fuzzy logic scripts

- ```
  clear all
  global Cy
  global y
  global f31
  global f41
   load mapleparam.mat
   load pomiar3\m1m2.mat
   load pomiar3\x_vector.mat
   m1m2=ans';
   x_vector=[t,pa1,pa2,pda1,pda2];
   f31 = readfis('f31.fis');
   f41 = readfis('f41.fis');
  ```
- ```
  function [y] = funkkg1f31(p);
   global f31;
   global Cy;
   f31.output(1).mf(1).params = p;
   sim('s_r_e_fuzzy');
   y = Cy;
  ```

```
for kk = 1:1,
    kk
for k = 1:36,
if k == 1,
  k
  x = fminbnd('funkkg1f31', -25, 10);    % obliczenie minimum funkcji
  f31.output(1).mf(1).params = x;        % przypisanie wartości minimalnej dla zmiennej w simulinku
Cy                                       % współczynnik jakości
end;
if k == 2,
  k
  x = fminbnd('funkkg2f31', -25, 10);    % obliczenie minimum funkcji
  f31.output(1).mf(2).params = x;        % przypisanie wartości minimalnej dla zmiennej w simulinku
Cy
end;
  ...
if k == 36,
  k
  x = fminbnd('funkkg36f31', -25, 10);   % obliczenie minimum funkcji
  f31.output(1).mf(36).params = x        % przypisanie wartości minimalnej dla zmiennej w simulinku
Cy
end;
writefis(f31, 'f31.fis');
end;
for k = 1:36,
if k == 1,
  k
  x = fminbnd('funkkg1f41', -25, 10);    % obliczenie minimum funkcji
  f41.output(1).mf(1).params = x;        % przypisanie wartości minimalnej dla zmiennej w simulinku
Cy
end;
if k == 2,
  k
  x = fminbnd('funkkg2f41', -25, 10);    % obliczenie minimum funkcji
  f41.output(1).mf(2).params = x;        % przypisanie wartości minimalnej dla zmiennej w simulinku
Cy
end;
...
if k == 36,
  k
  x = fminbnd('funkkg36f41', -25, 10);   % obliczenie minimum funkcji
  f41.output(1).mf(36).params = x;       % przypisanie wartości minimalnej dla zmiennej w simulinku
Cy
end;
end;
writefis(f41, 'f41.fis');
end;
```

Genetic algorithm

Genetic algorithms similarly to fuzzy logic was used also in two structures described before. However instead of fuzzy sets used to determine values of algebraic expressions present in the matrix f(x,a), presented below form was assumed:

$$f(x,a) = \begin{bmatrix} 0 \\ 0 \\ p_1 \operatorname{sgn}(x_4) + p_2 x_2^2 + p_3 x_2 x_4 + p_4 x_4^2 + p_5 \operatorname{sgn}(x_2) \\ p_5 \operatorname{sgn}(x_4) + p_4 x_2^2 + p_3 x_2 x_4 + p_2 x_4^2 + p_1 \operatorname{sgn}(x_2) \end{bmatrix}$$

where p are determined parameters including coefficients a.

Genetic algorithms - scripts

```
clear all
load C:\matlabR12\simulink\doktorat\matlabv6_0_0_88\maple_matlab\model_ruchu\mapleparam.mat
%Pomiar3
load C:\matlabR12\simulink\doktorat\matlabv6_0_0_88\maple_matlab\dane\pomiar3\m1m2.mat
load C:\matlabR12\simulink\doktorat\matlabv6_0_0_88\maple_matlab\dane\pomiar3\x_vector.mat
m1m2=ans';
x_vector=[t,pa1,pa2,pda1,pda2];
%Dane do zainicjowania obliczen
x(1)=-0.5;
 x(2)=0.5;
  x(3)=-0.5;
   x(4)=0.5;
    x(5)=-0.5;
```

init_param.m

gen_genetic_f.m

```
global y3;
 global y4
  global y3r;
   global y4r;
y1r=y1r;
 y2r=y2r;
  y3r=y3r;
   y4r=y4r;
y1=y1(:);
 y2=y2(:);
  y3=y3(:);
   y4=y4(:);
t=time.time;
ep =    130;
min_p= -6;
max_p = 6;
bounds=ones(5,1)*[min_p max_p];
initPop=initializega(150,bounds,'genetic_f');
[x endPop] = ga(bounds,'genetic_f',[5..8],initPop,[1e-7 1 1],'maxGenTerm',ep,...
'tournSelect',[2],['heuristicXover'],[50..10],'multiNonUnifMutation',[10 ep 9]);
 x
```

Genetic algorithms - scripts

```
function [sol,val]=gaDemo1Eval(sol,options)
global y3;
  global y4
    global y3r;
      global y4r;
p1=sol(1);
  p2=sol(2);
    p3=sol(3);
      p4=sol(4);
        p5=sol(5);
     z=p1*sign(y4)+p2*y3.^2+p3*y3.*y4+p4*y4.^2+p5*sign(y3);
     w=p1*sign(y4r)+p2*y3r.^2+p3*y3r.*y4r+p4*y4r.^2+p5*sign(y3r);
     r=abs(z-w);
     val = sum(r.^2);
     val=-val;
```

genetic_f.m

Experiments to conduct

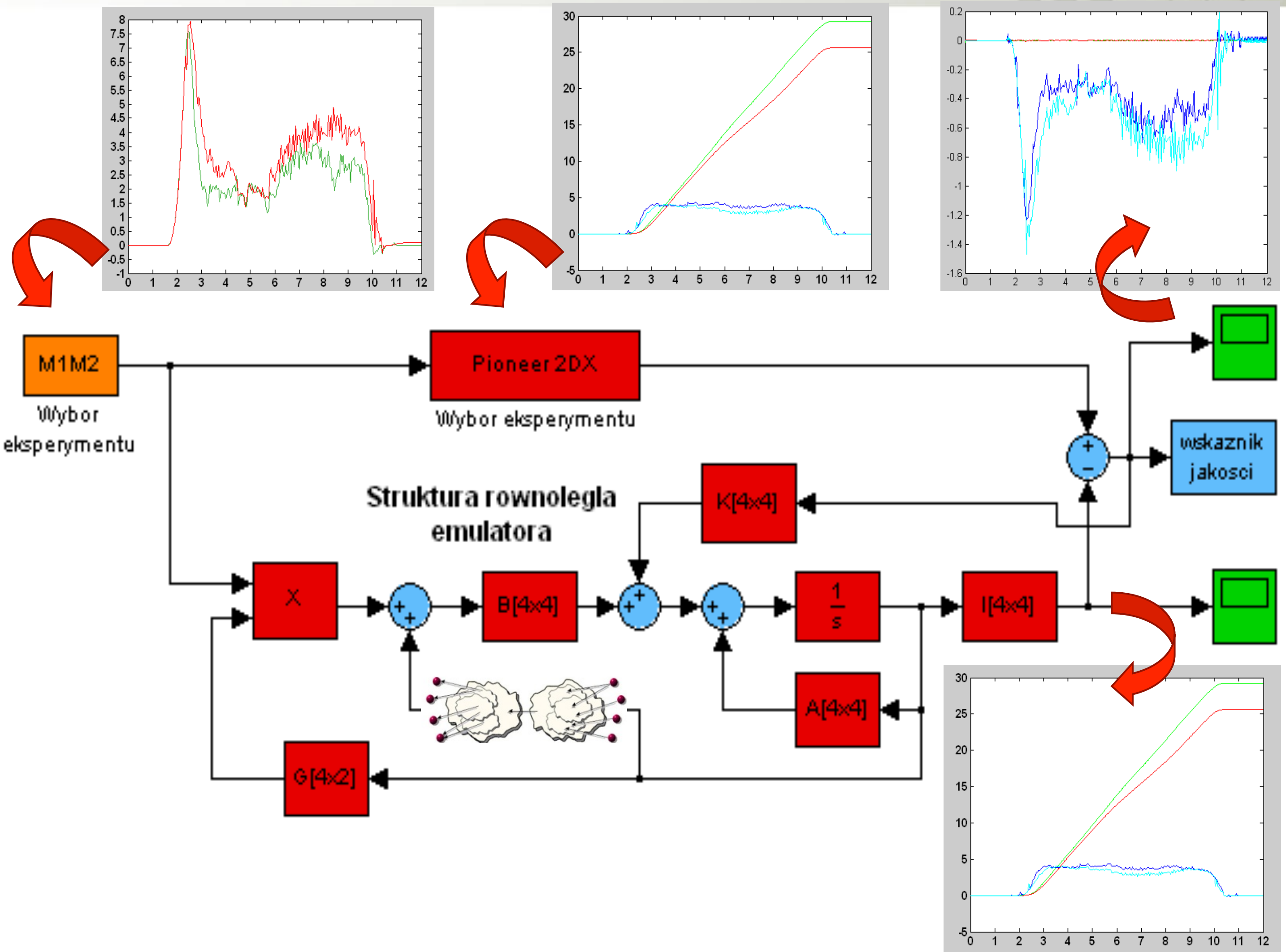
experiment 1	Straight line motion model
experiment 2	Straight line motion model (mathematical model with noise)
experiment 3	Curve line motion model
experiment 4	Curve line motion model (mathematical model with noise)
experiment 5	Straight line motion model
experiment 6	Straight line motion model with parametric disturbance
experiment 7	Curve line motion model
experiment 8	Curve line motion model with parametric disturbance

Parallel state emulator structure

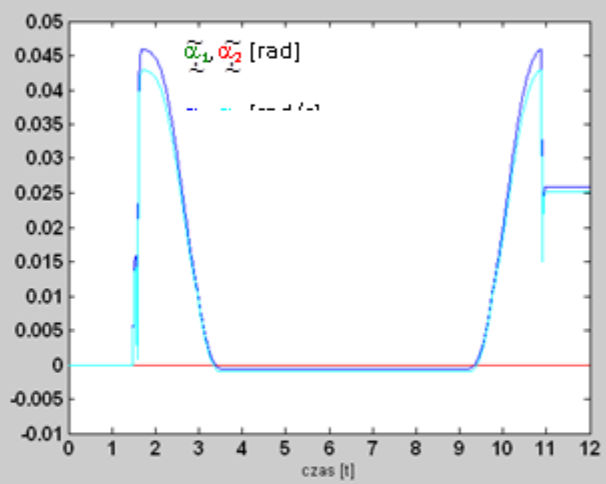
If we fully know state vector we can use parallel state emulator structure during identification process.

In the structure presented next input U is known and represents driving torques of wheel 1&2. System denoted as Pioneer 2DX represents real model of the robot to which the emulator is adjusted. On output of this system we get kinematic parameters – displacements and angular velocities of wheel 1&2.

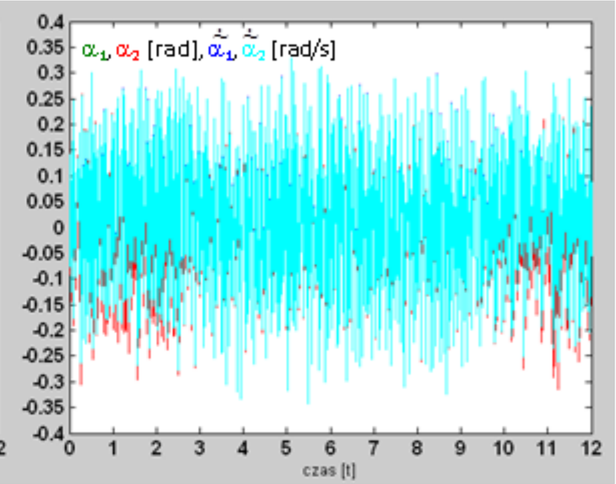
Parallel state emulator structure



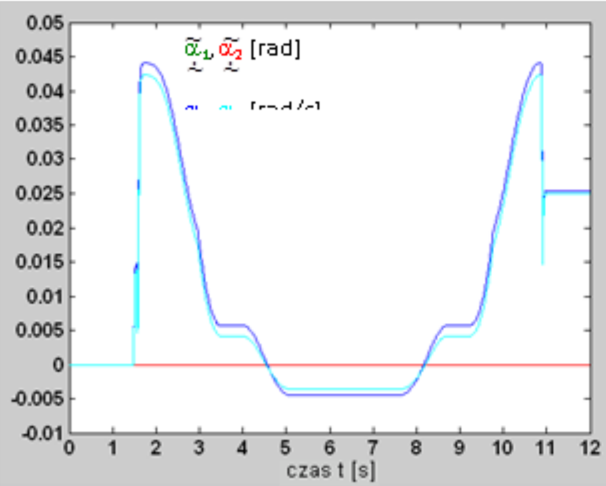
Parallel state emulator structure



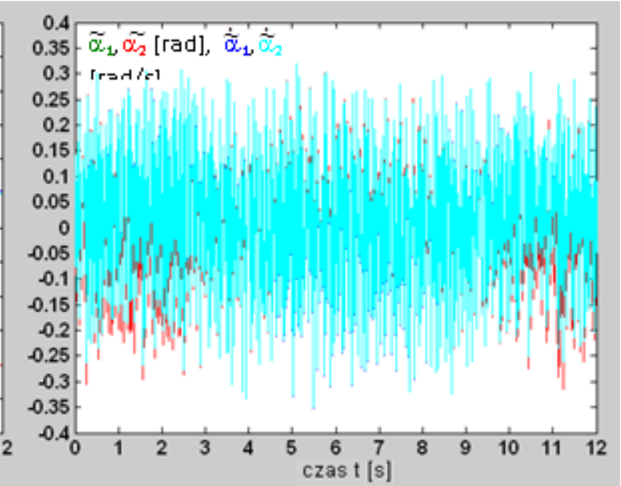
eksperyment1



eksperyment2

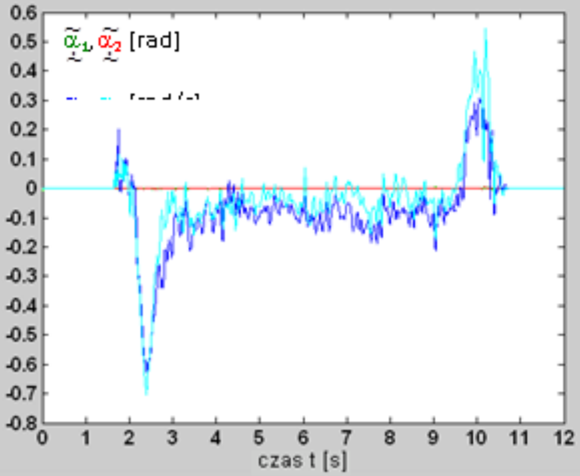


eksperyment3

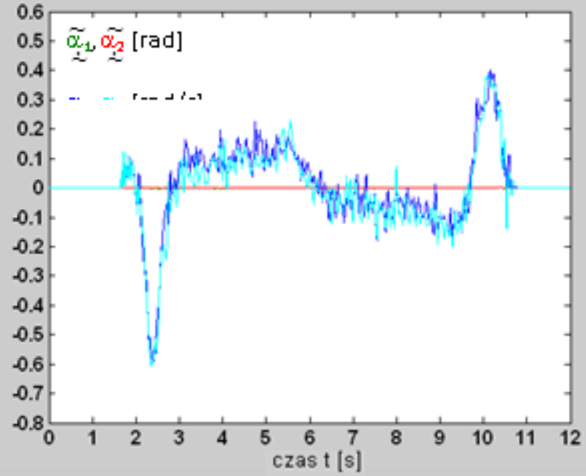


eksperyment4

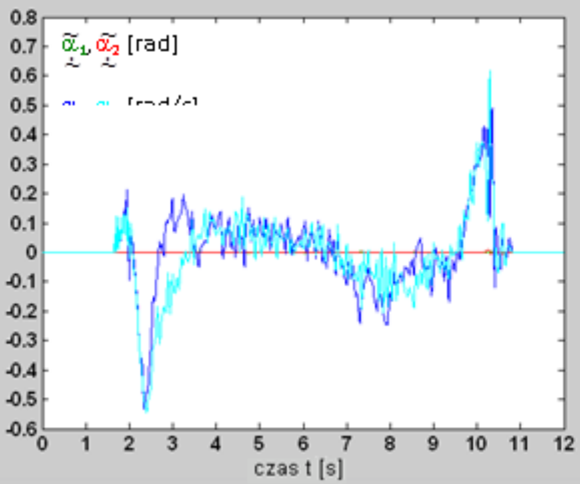
Parallel state emulator structure



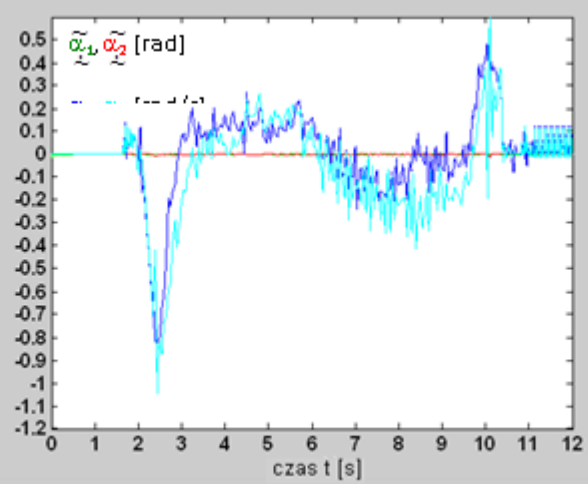
eksperyment5



eksperyment6



eksperyment7



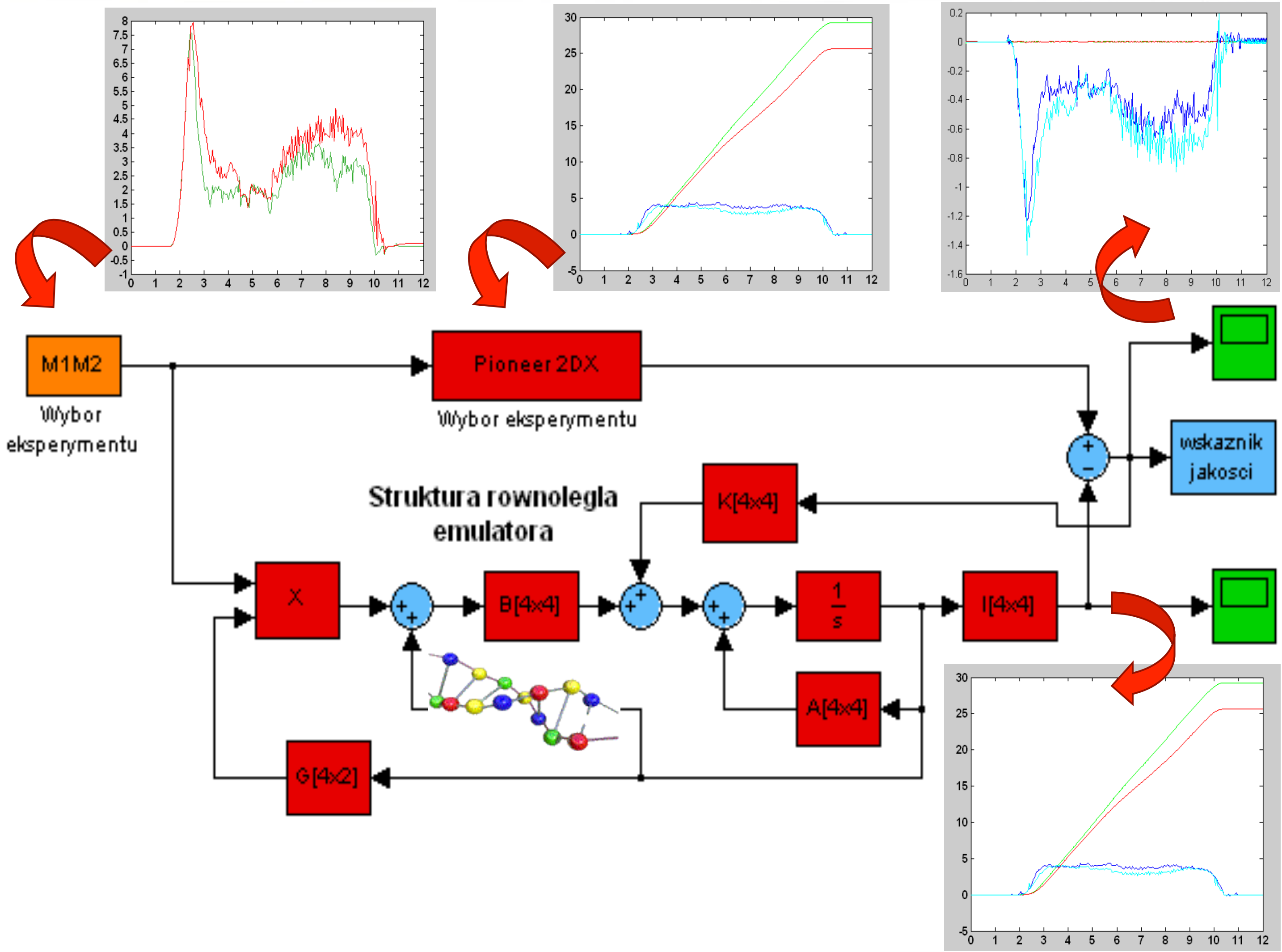
eksperyment8

Parallel state emulator structure

Values of quality coefficient Cy received for the experiments with the use of parallel state emulator structure.

experiment 1	Cy= 0.0076	experiment 2	Cy= 0.2474
experiment 3	Cy= 0.0083	experiment 4	Cy= 0.2486
experiment 5	Cy= 0.4578	experiment 6	Cy= 0.4263
experiment 7	Cy= 0.3729	experiment 8	Cy= 0.8895

Parallel state emulator structure

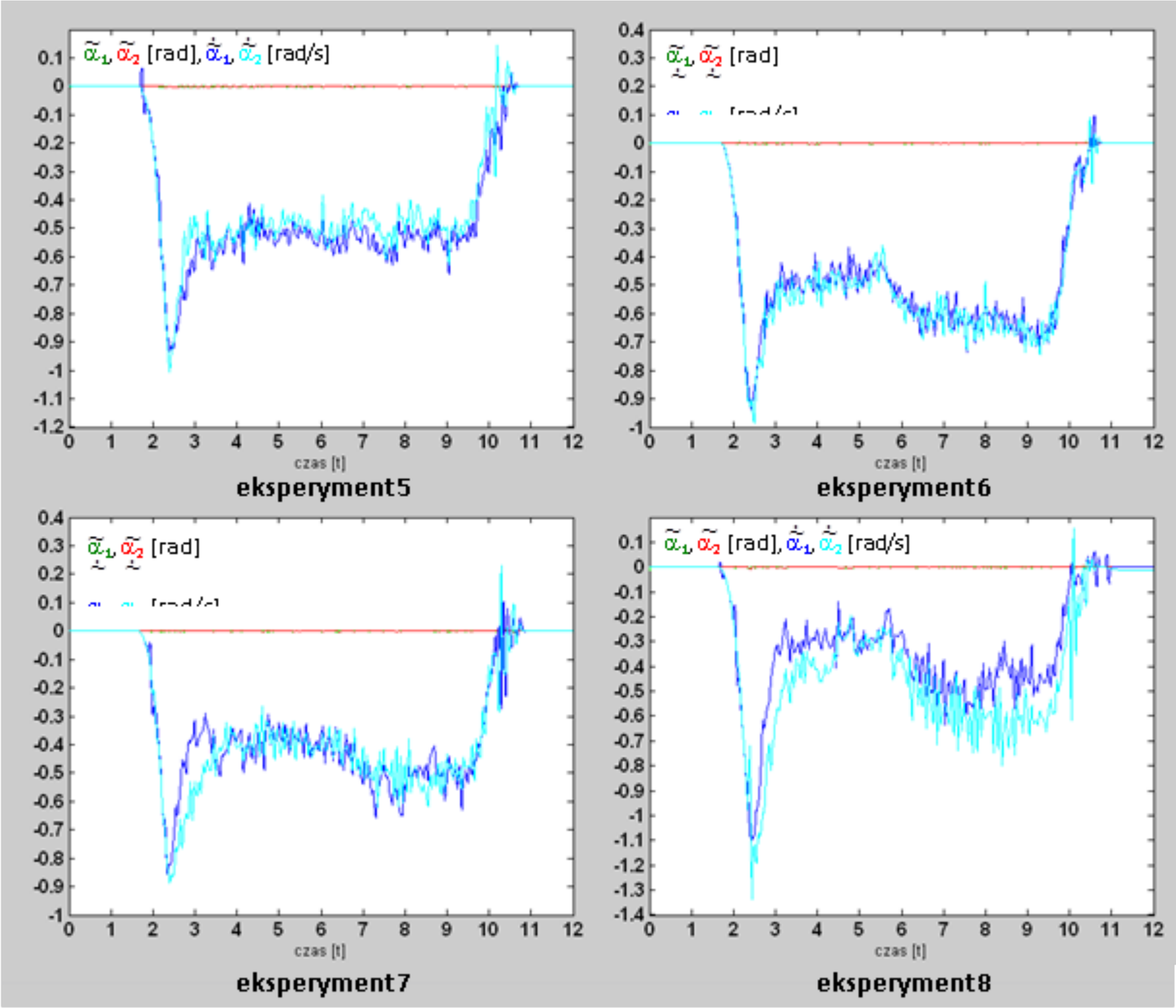


Parallel state emulator structure

- Values of the p parameters and Wj factor for individual experiments

	p_1	p_2	p_3	p_4	p_5	Wj
Experiment 5	0.0043	0.0596	-0.1403	0.0809	0.0030	0.0211
Experiment 6	-0.0315	2.1511	-4.1457	2.0010	0.0081	0.9671
Experiment 7	-0.0013	-0.1245	0.2755	-0.1526	0.0019	0.0560
Experiment 8	0.0125	-0.2635	0.4962	-0.2408	0.0009	0.8857

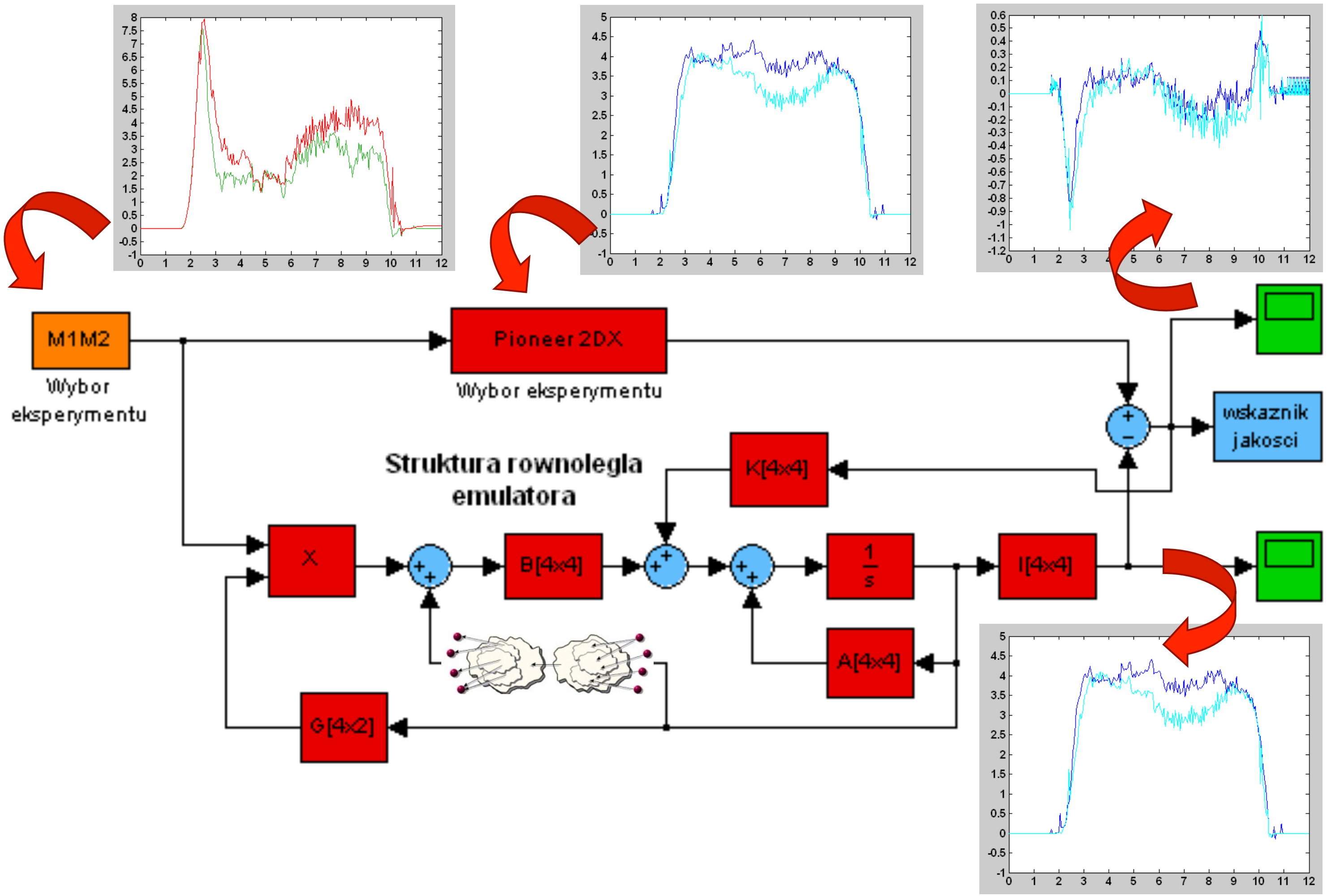
Parallel state emulator structure



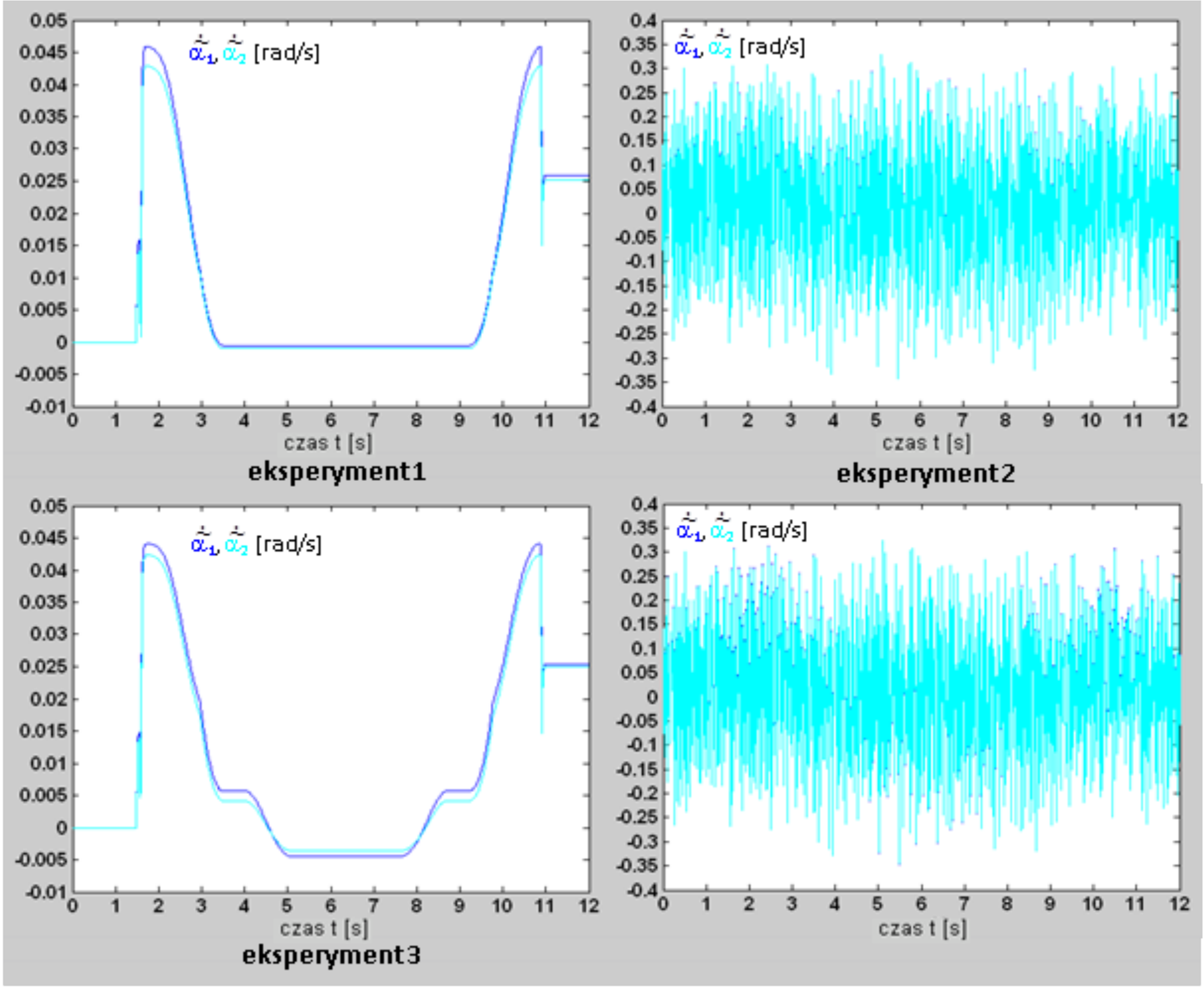
Parallel velocity identifier structure

In case when we don’t know fully state vector, but only time series of angular velocities of wheel 1&2, we can use parallel velocity identifier structure.

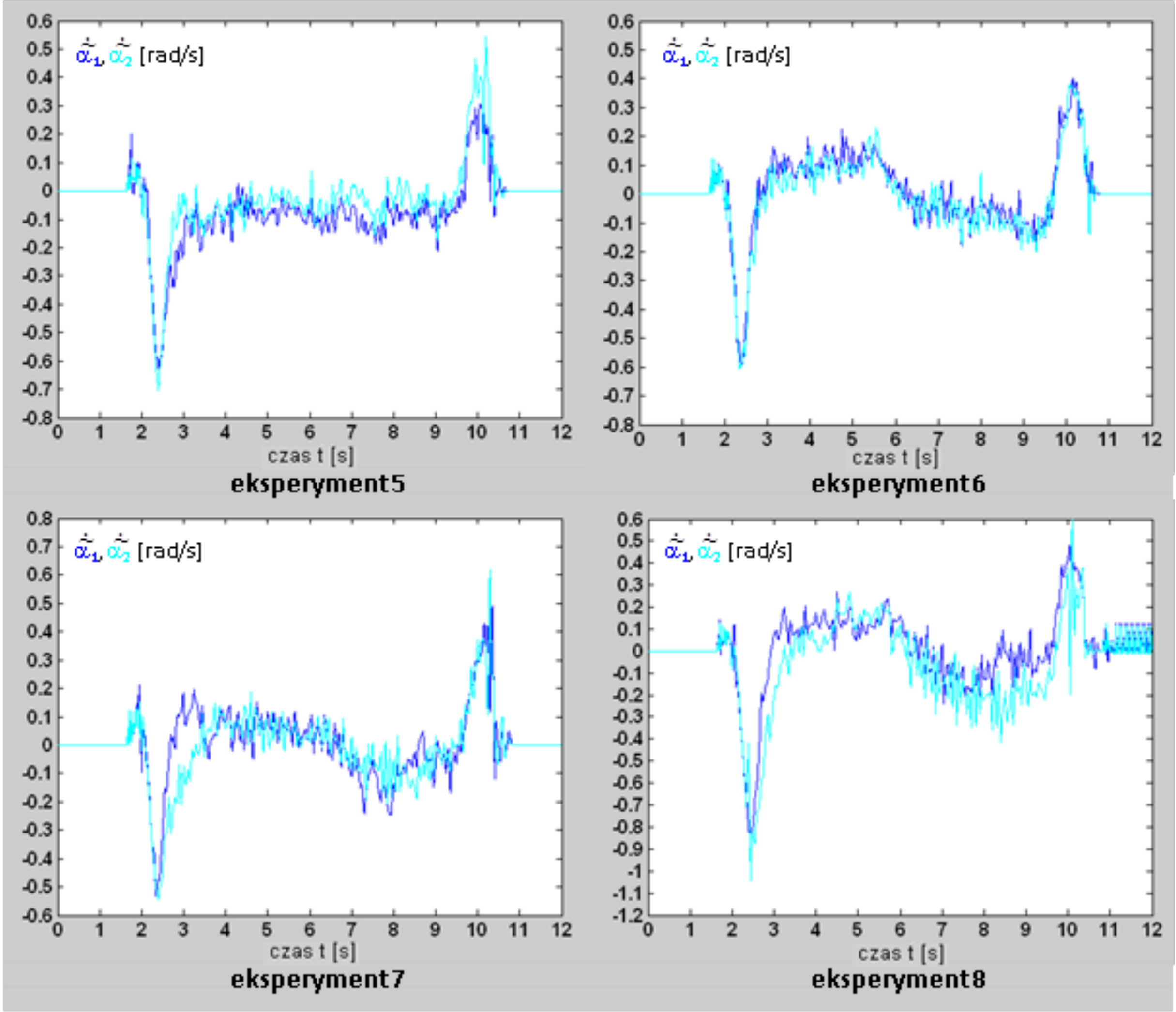
Parallel velocity identifier structure – fuzzy logic



Parallel velocity identifier structure – fuzzy logic



Parallel velocity identifier structure – fuzzy logic



Parallel velocity identifier structure – fuzzy logic

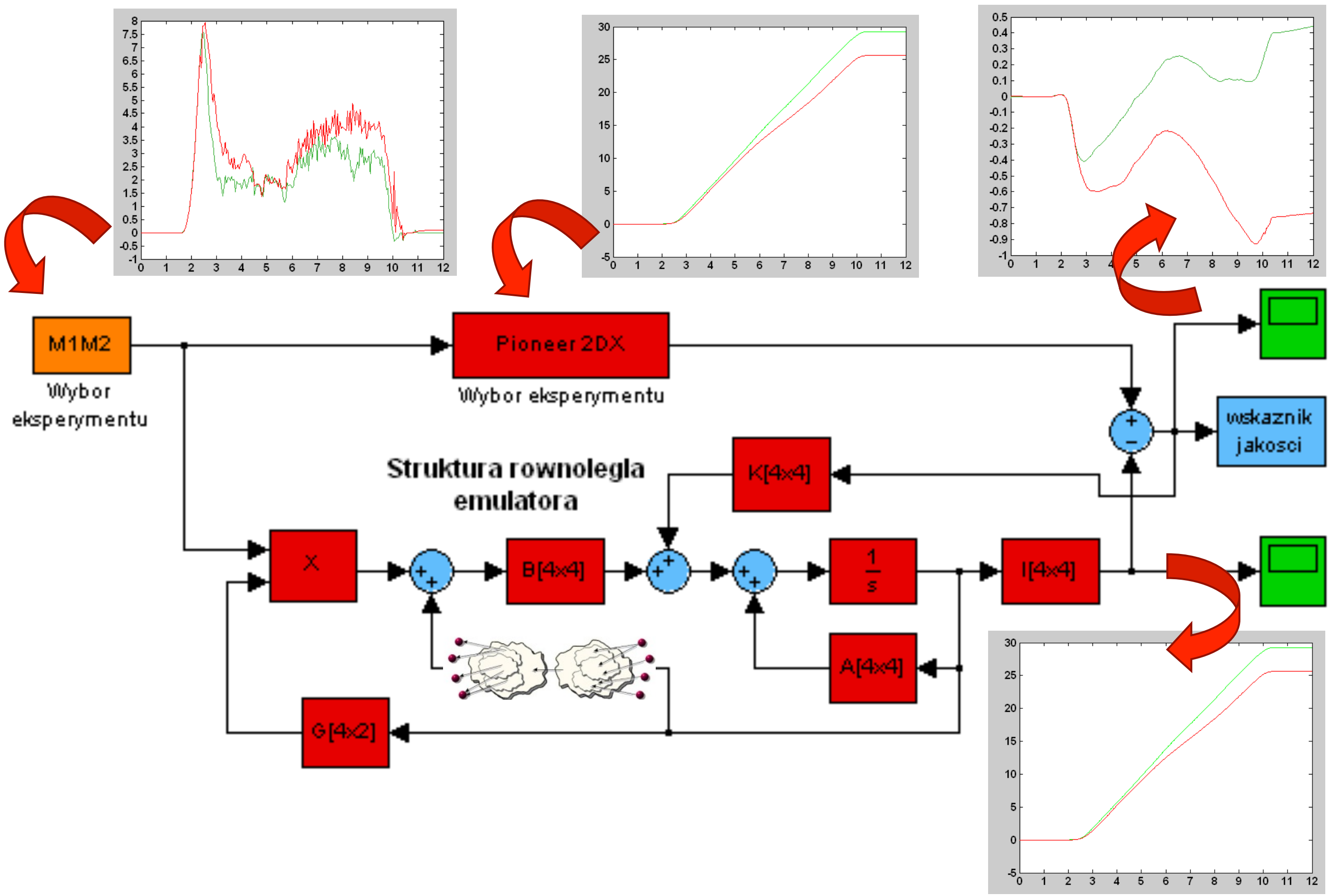
- Values of quality coefficient Cy received for individual experiments with the use of parallel velocity identifier structure

Experiment 1	Cy= 0.0076	Experiment 2	Cy= 0.2474
Experiment 3	Cy= 0.0083	Experiment 4	Cy= 0.2486
Experiment 5	Cy= 0.4578	Experiment 6	Cy= 0.4263
Experiment 7	Cy= 0.3729	Experiment 8	Cy= 0.8895

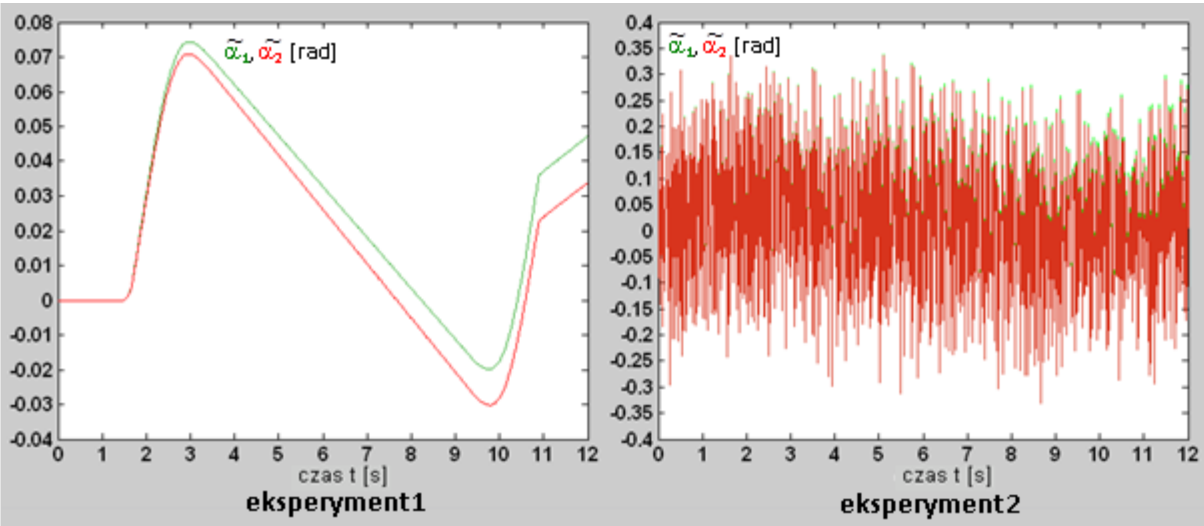
Parallel displacement identifier structure

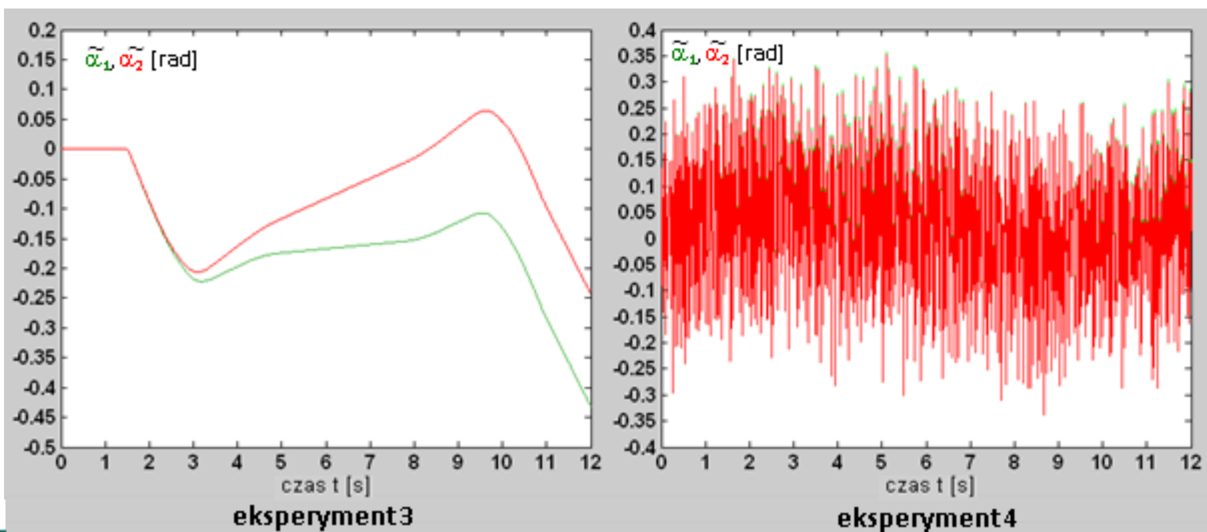
In case when we don’t know fully state vector, but only time series of angular displacements of wheel 1&2, we can use parallel displacement identifier structure.

Parallel displacement identifier structure

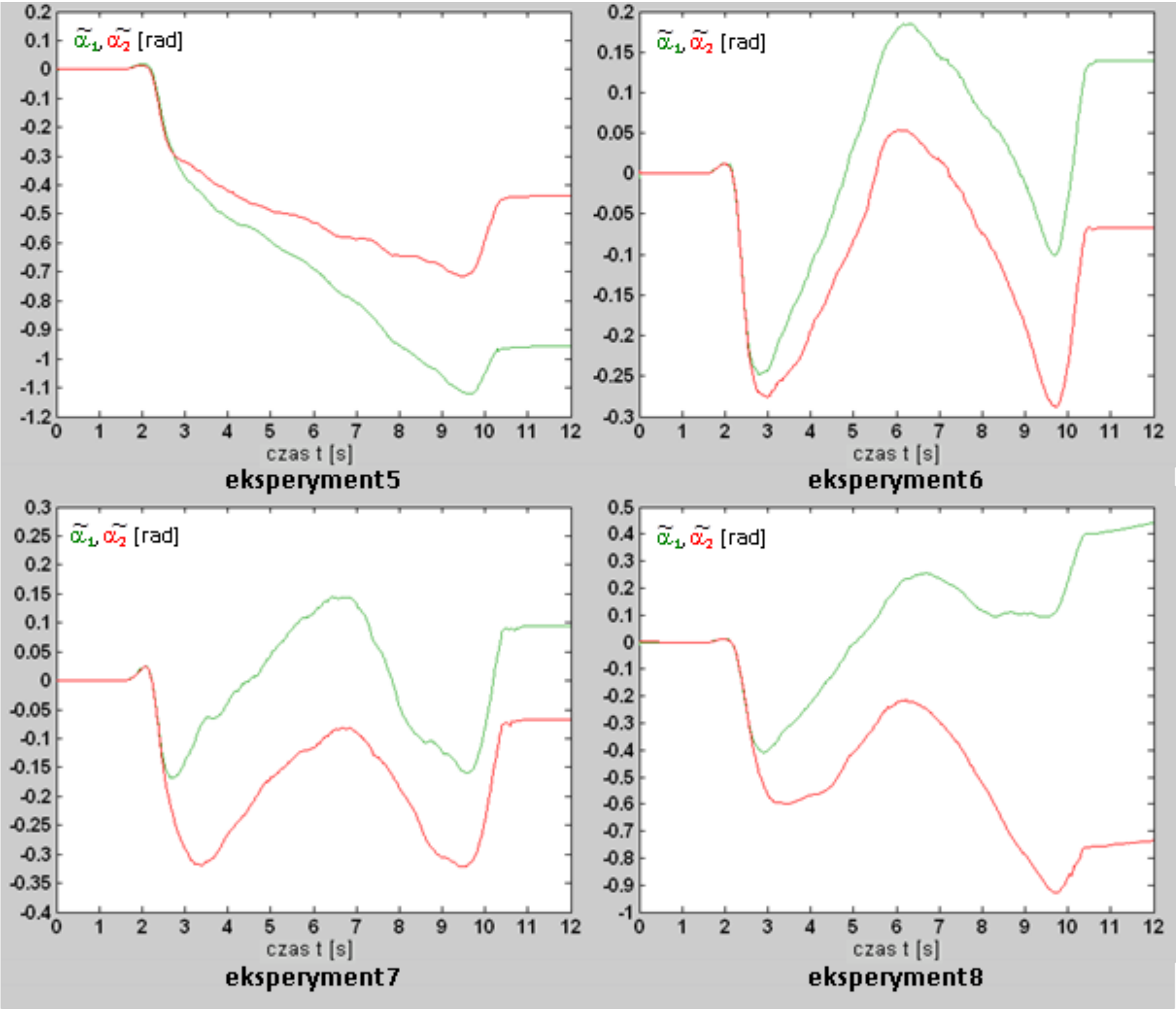


Parallel displacement identifier structure





Parallel displacement identifier structure

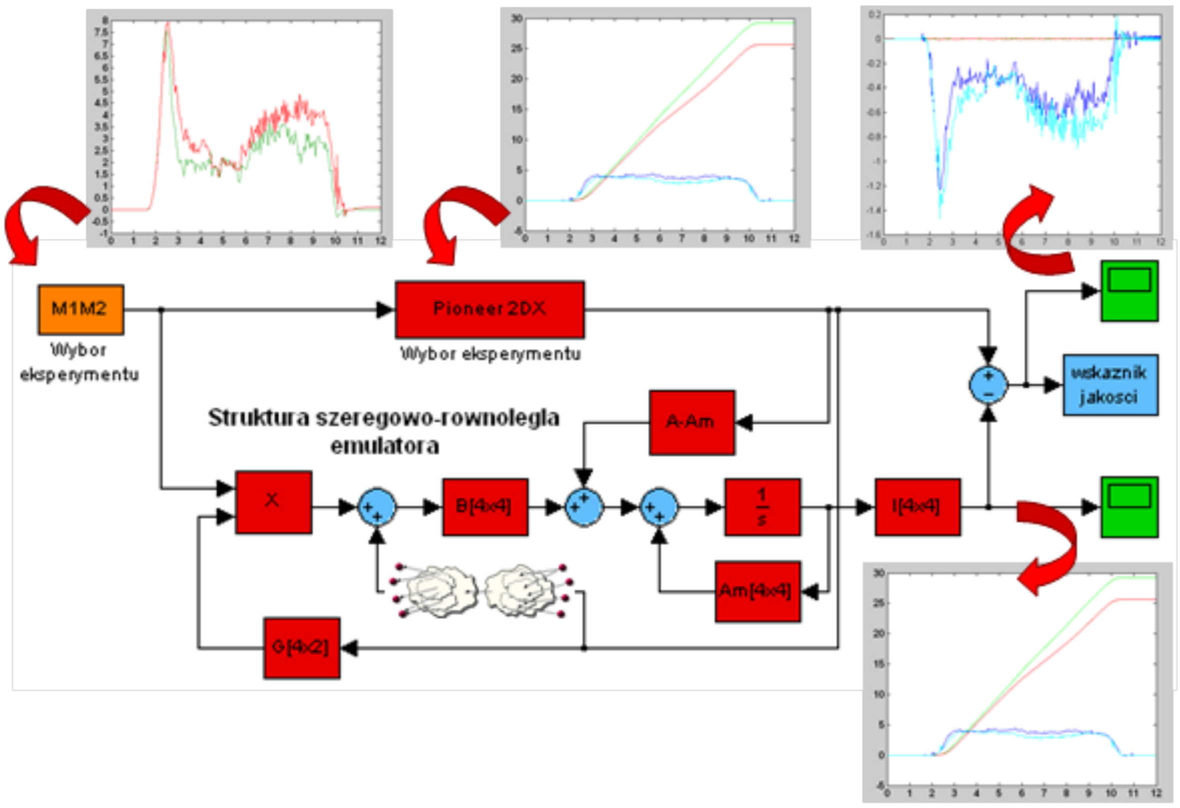


Parallel displacement identifier structure

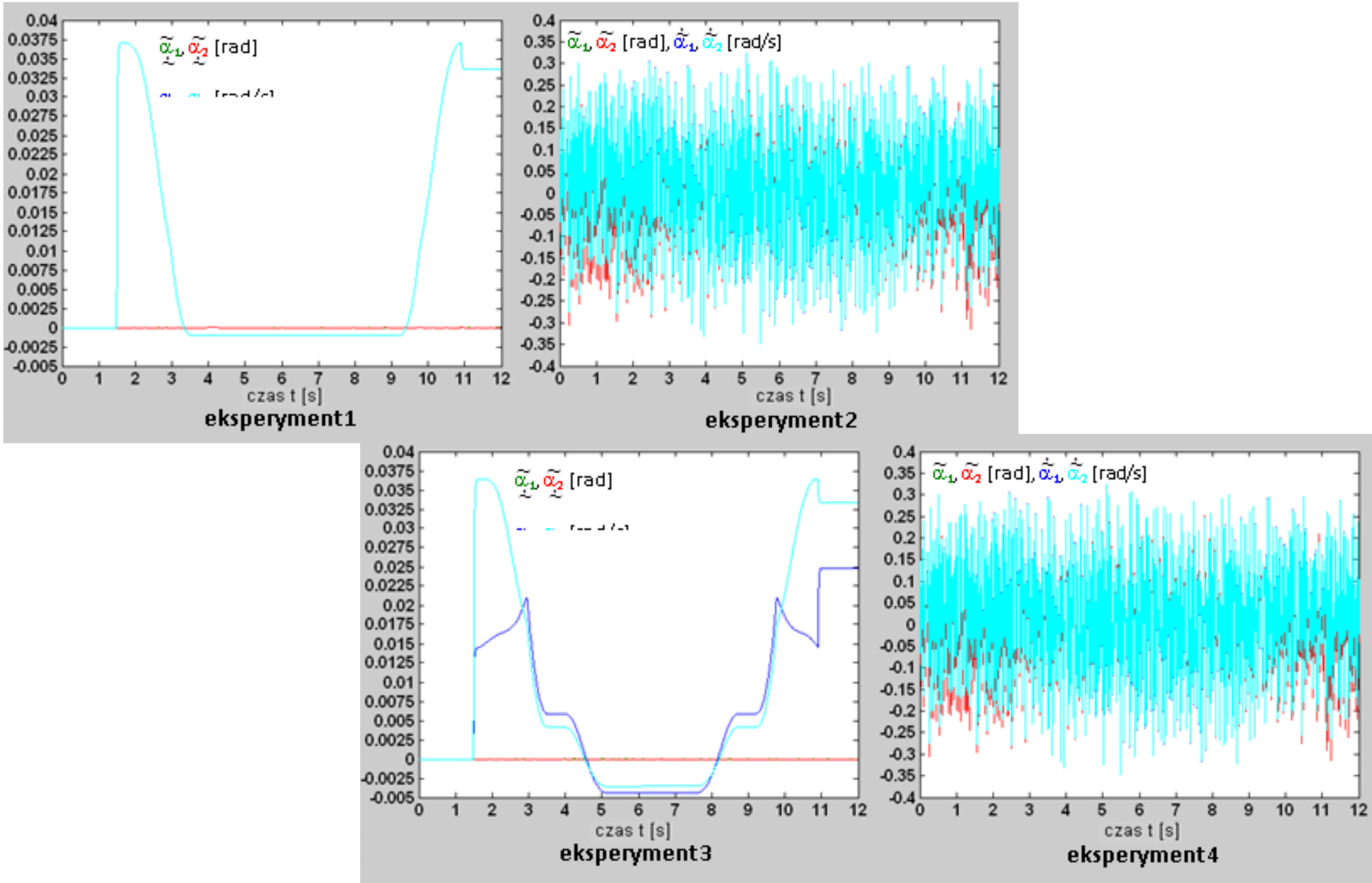
- Values of quality coefficient Cy received for individual experiments with the use of parallel displacement identifier structure

Experiment 1	Cy=0.0304	Experiment 2	Cy=0.2721
Experiment 3	Cy=0.5298	Experiment 4	Cy=0.2851
Experiment 5	Cy=9.0312	Experiment 6	Cy=0.3859
Experiment 7	Cy=0.4906	Experiment 8	Cy= 4.0674

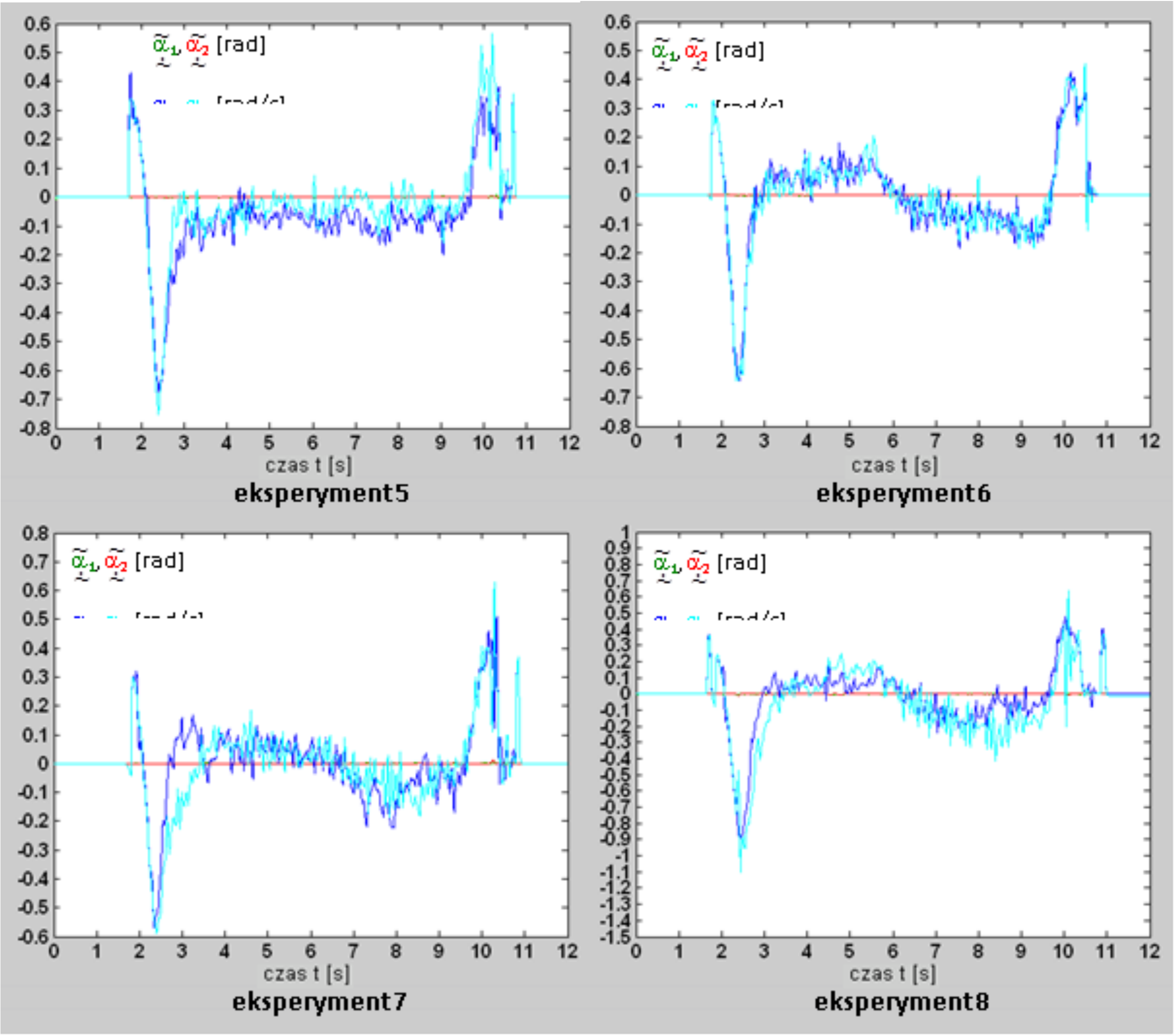
Serial-parallel state emulator structure



Serial-parallel state emulator structure



Serial-parallel state emulator structure

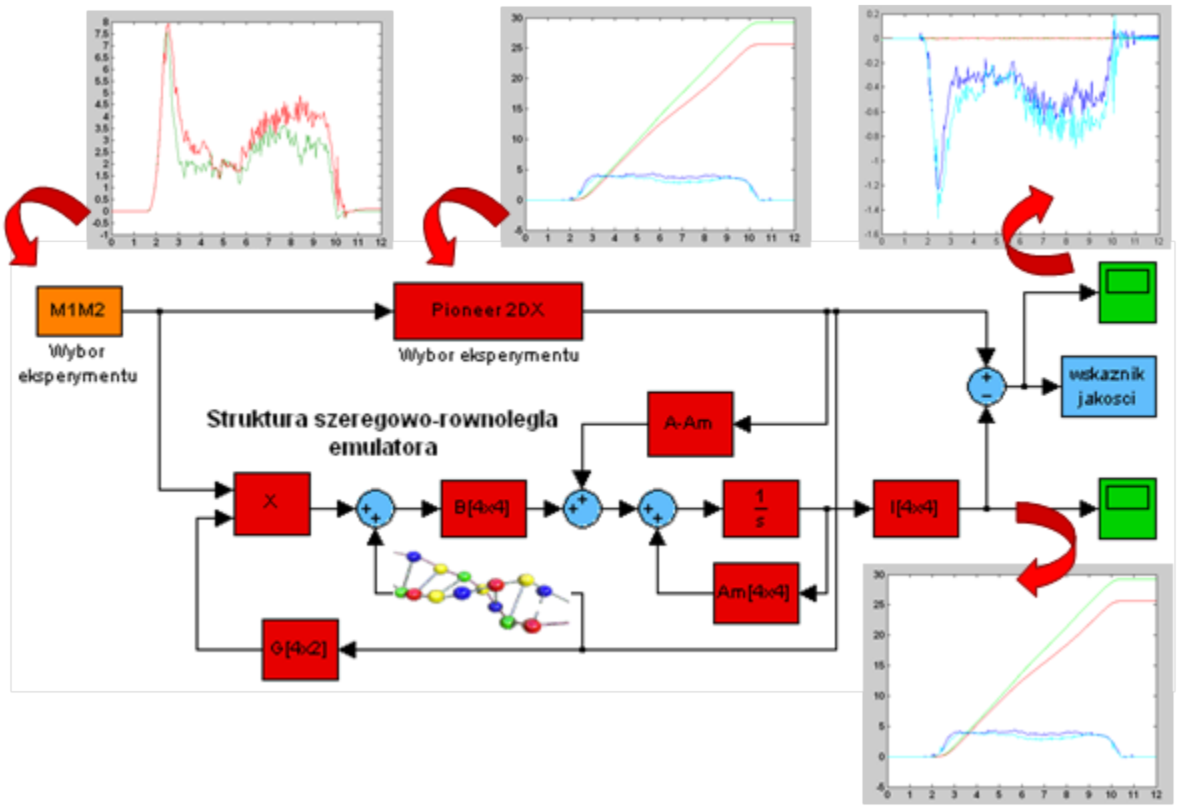


Serial-parallel state emulator structure

Values of quality coefficient Cy received for individual experiments with the use of serial-parallel state emulator structure.

Experiment 1	Cy= 0.0070	Experiment 2	Cy= 0.2507
Experiment 3	Cy= 0.0054	Experiment 4	Cy= 0.2513
Experiment 5	Cy= 0.5494	Experiment 6	Cy= 0.4975
Experiment 7	Cy= 0.4331	Experiment 8	Cy= 0.9843

Serial-parallel state emulator structure

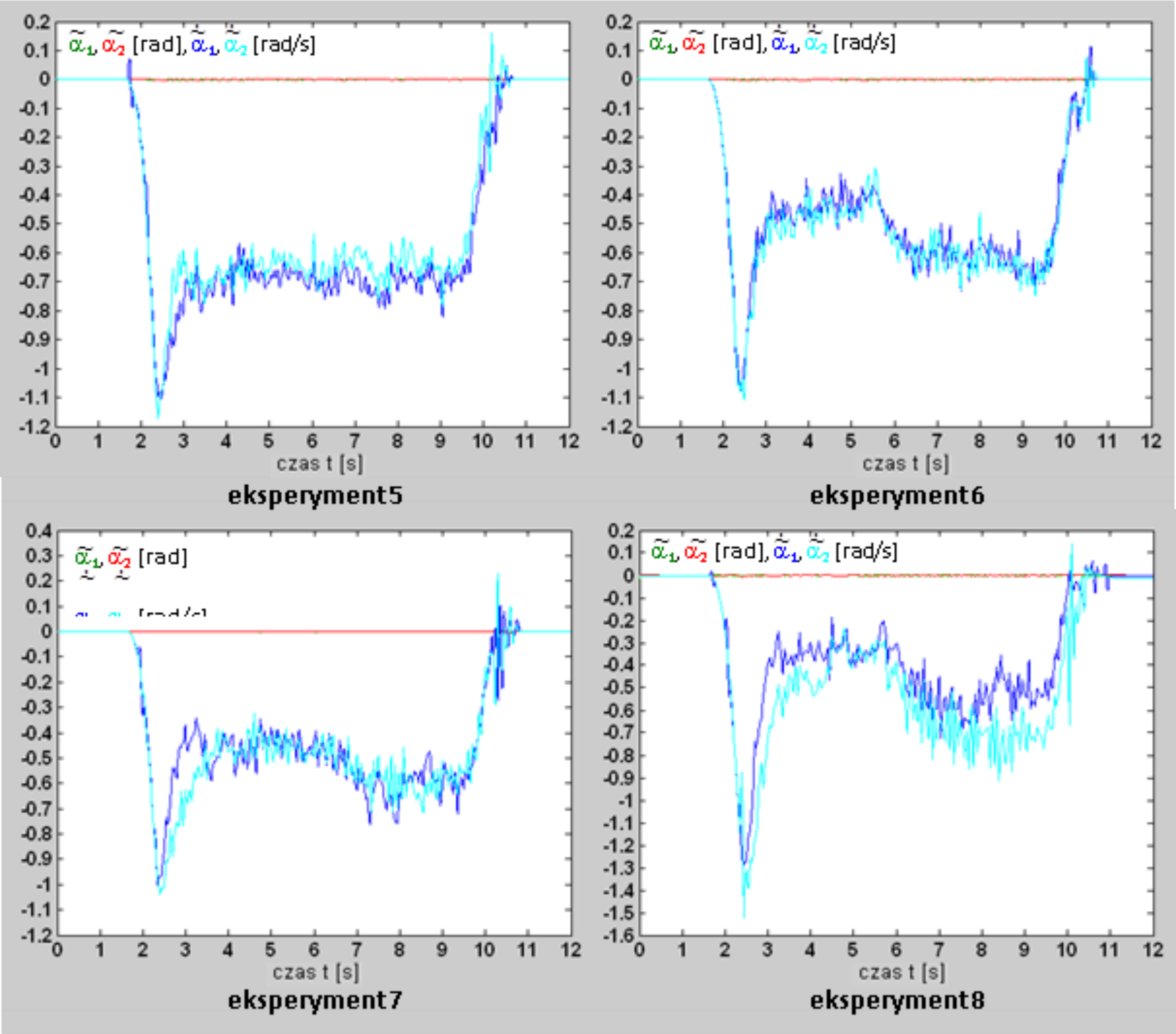


Serial-parallel state emulator structure

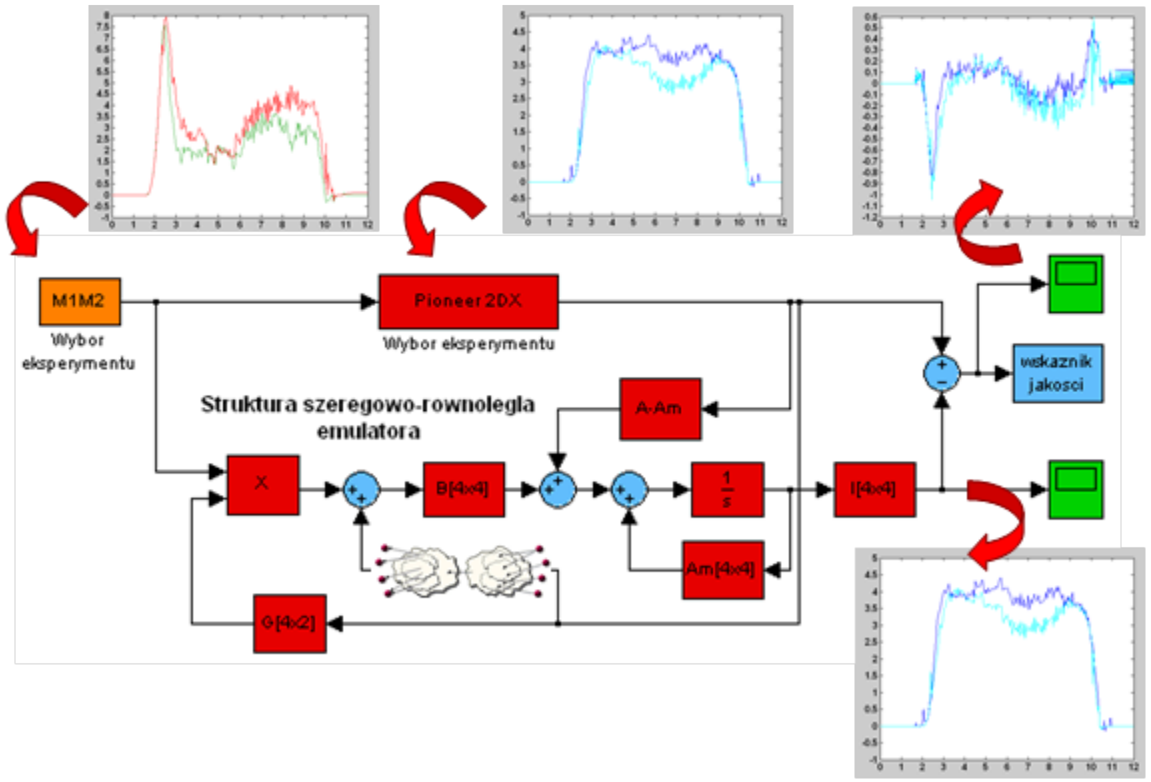
Values of the p parameters and Wj factor for individual experiments

	p_1	p_2	p_3	p_4	p_5	Wj
Experiment 5	0.0132	-0.2517	0.5486	-0.2998	-0.0168	0.2841
Experiment 6	-0.0004	0.4952	-0.9938	0.4991	0.0007	0.0265
Experiment 7	0.0010	0.0960	-0.2090	0.1140	-0.0008	0.0331
Experiment 8	0.0007	-0.0560	0.1104	-0.0557	-0.0010	0.0483

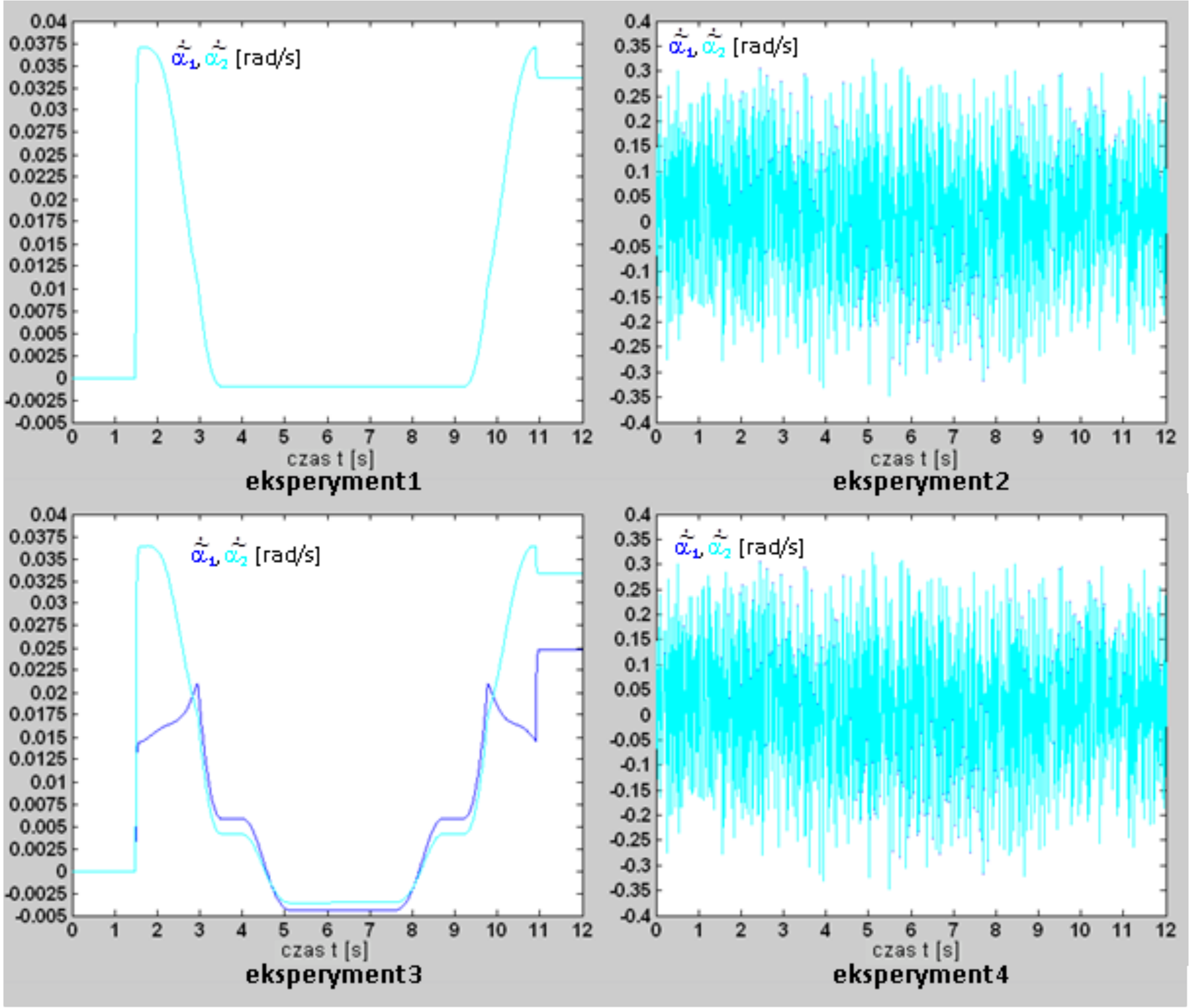
Serial-parallel state emulator structure



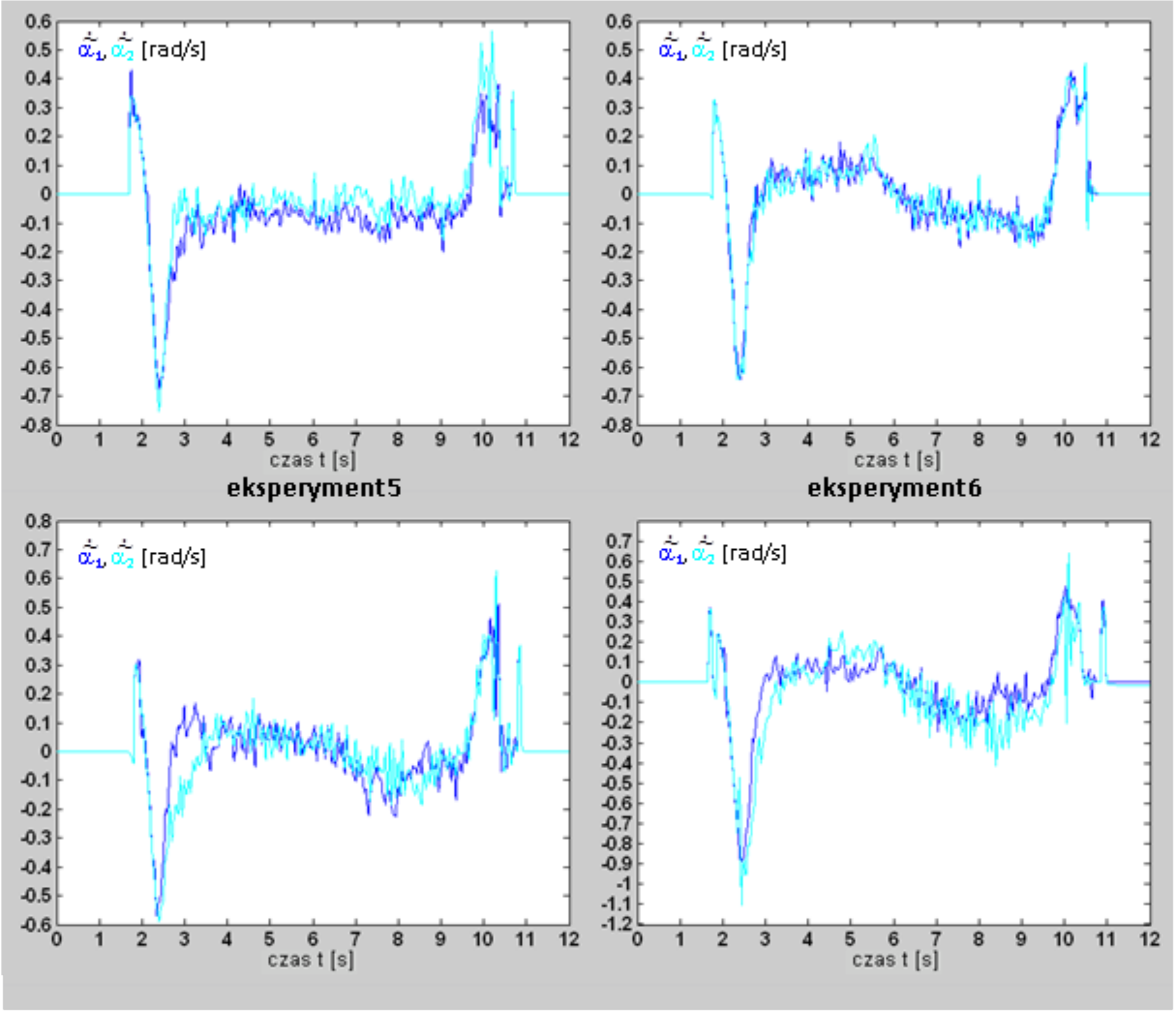
Serial-parallel velocity identifier structure



Serial-parallel velocity identifier structure



Serial-parallel velocity identifier structure

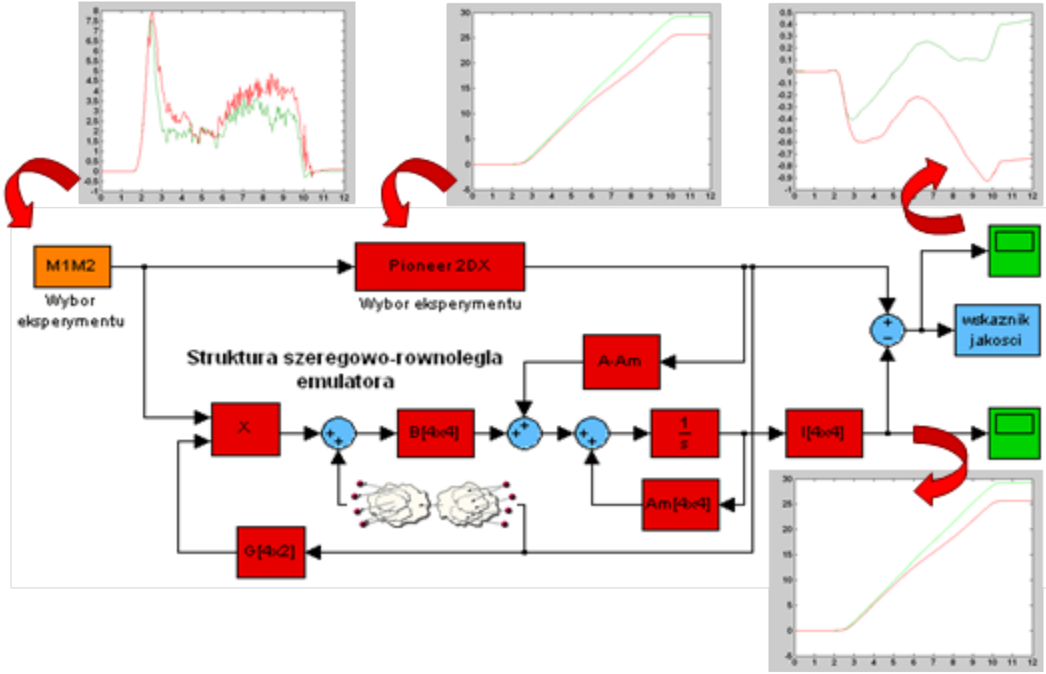


Serial-parallel velocity identifier structure

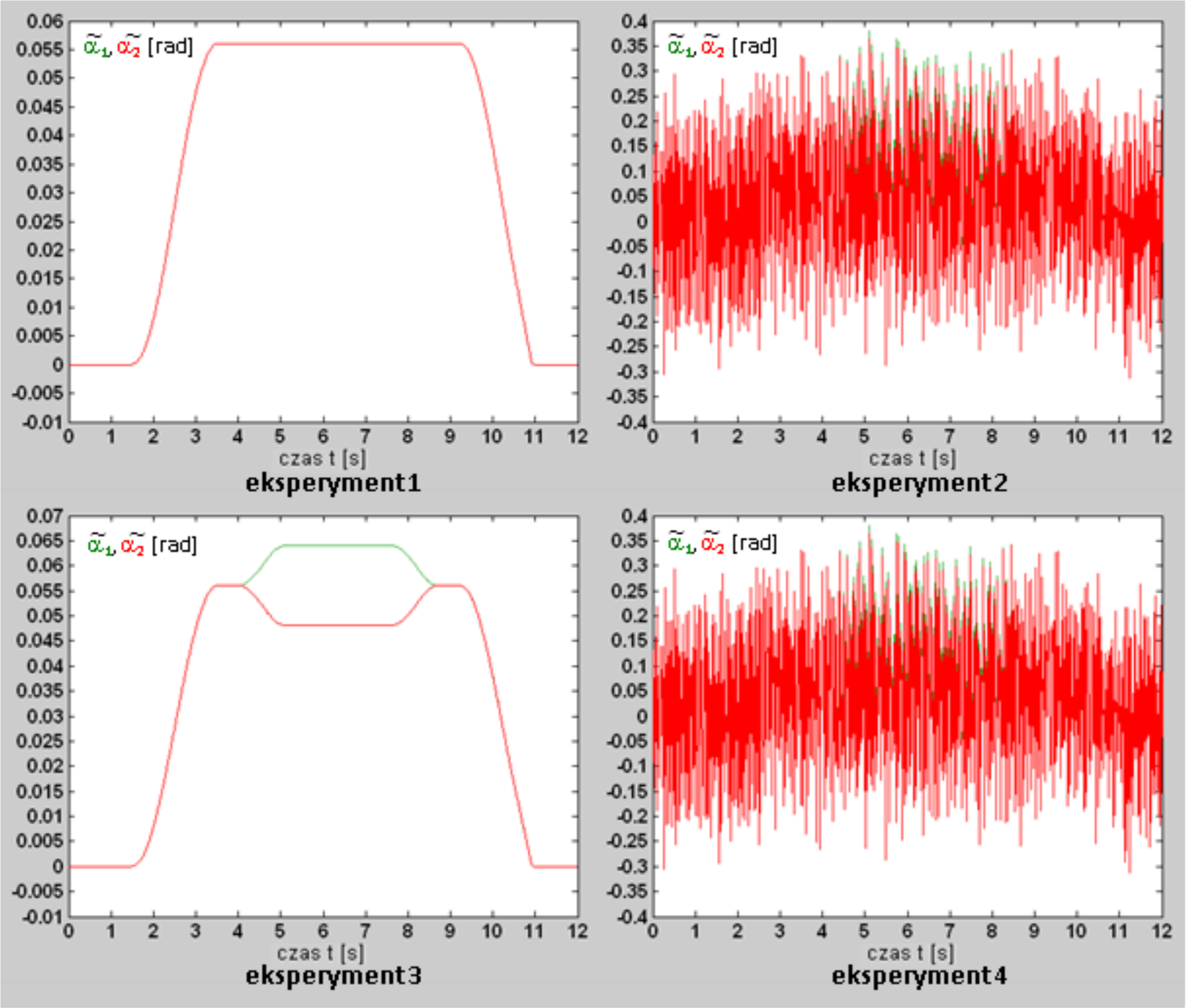
- Values of quality coefficient Cy received for individual experiments with the use of serial-parallel velocity identifier

Experiment 1	Cy= 0.0070	Experiment 2	Cy= 0.2507
Experiment 3	Cy= 0.0054	Experiment 4	Cy= 0.2513
Experiment 5	Cy= 0.5494	Experiment 6	Cy= 0.4975
Experiment 7	Cy= 0.4331	Experiment 8	Cy= 0.9843

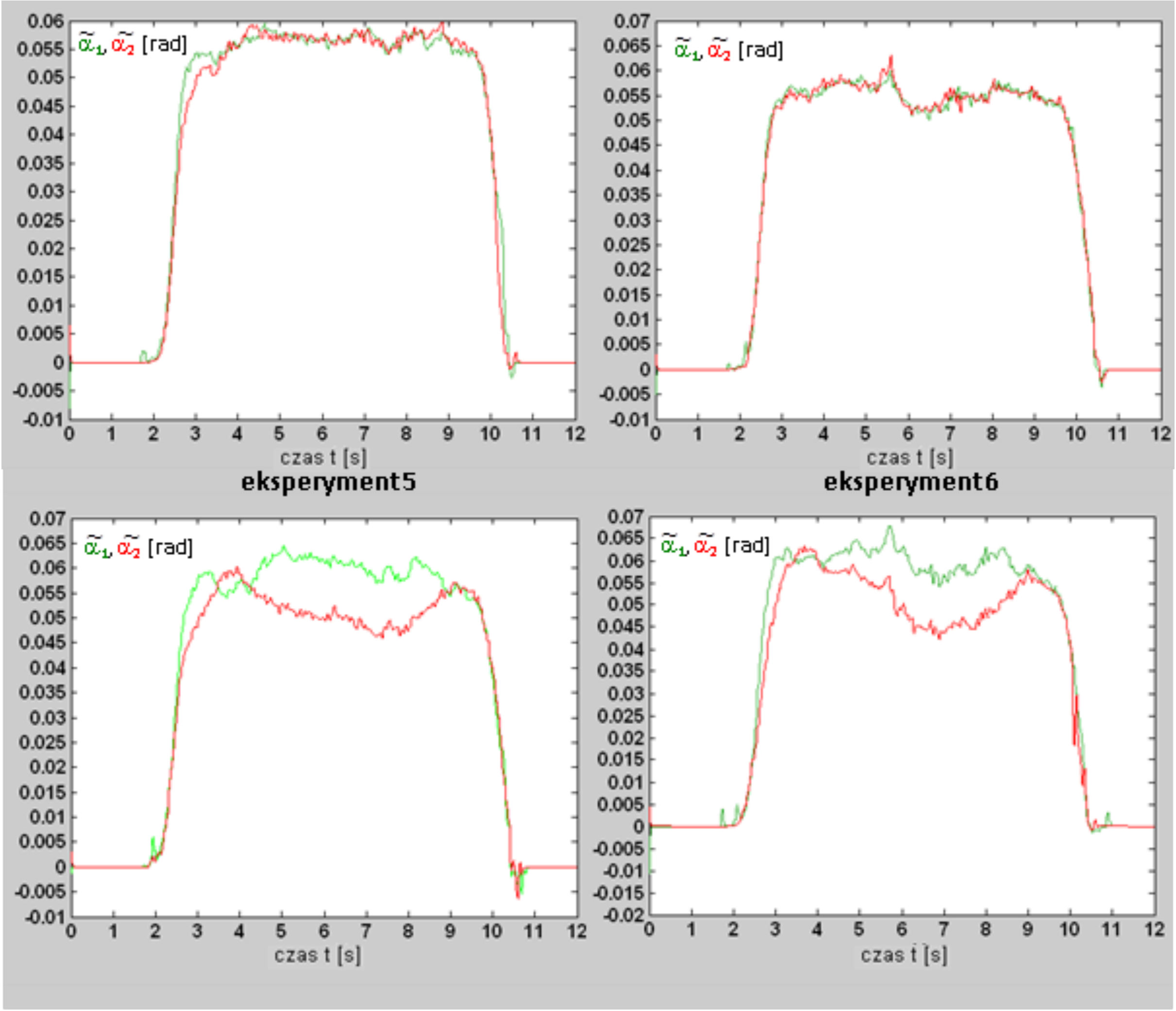
Serial-parallel displacement identifier structure



Serial-parallel displacement identifier structure



Serial-parallel displacement identifier structure



Serial-parallel displacement identifier structure

- Values of quality coefficient Cy received for individual experiments with the use of serial-parallel displacement identifier structure

Experiment 1	Cy= 0.0454	Experiment 2	Cy= 0.2784
Experiment 3	Cy= 0.0458	Experiment 4	Cy= 0.2789
Experiment 5	Cy= 0.0465	Experiment 6	Cy= 0.0455
Experiment 7	Cy= 0.0460	Experiment 8	Cy= 0.0464

genfis2 – automatic generation of fuzzy systems

genfis2 function is created based on *subclust* function used for ”subtractive clustering” in order to provide fast method of acquisition of input and output data and creation of fuzzy system of Sugeno type which models behavior of the data.

In order to implement genfis2 function we must have input data for fuzzy system and output data from the model.

genfis2 – automatic generation of fuzzy systems

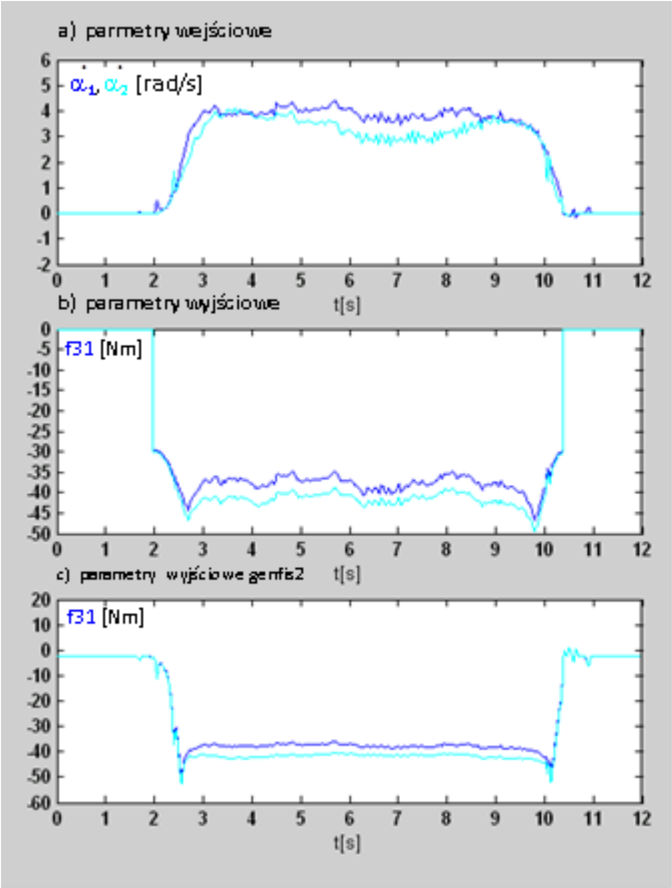
There were used data from off-line identification with the use of fuzzy logic systems made with the use of *fuzzy logic toolbox* and function to select minimum of the function (*fminbnd)*.

Based on the example of parallel structure of state emulator, angular velocities of wheel 1&2 were measured which are inputs to both fuzzy logic systems in state emulator and output parameters of each fuzzy system.

genfis2 – automatic generation of fuzzy systems – example of use

```
in=[f3c(1,:)',f4c(1,:)'];
  t=time.time;
    out1=f3row(1,:)';
      out2=f4row(1,:)';
         out=[out1,out2];
subplot(3,1,1), plot(t, in(:,1),'b')
  hold
subplot(3,1,1), plot(t, in(:,2),'c')
  subplot(3,1,2), plot(t,out1,'b')
hold
subplot(3,1,2), plot(t,out2,'c')
f31=genfis2(in,out1,0.5)
  f41=genfis2(in,out2,0.5)
fuzout1=evalfis(in,f31);
  fuzout2=evalfis(in,f41);
    fuzout=[fuzout1 fuzout2];
subplot(3,1,3), plot(t,fuzout1,'b')
hold
subplot(3,1,3), plot(t,fuzout2,'c')
```

genfis2 – automatic generation of fuzzy systems – example of use



Experiment 5	*Cy= 0.5002*
Experiment 6	*Cy= 0.4428*
Experiment 7	*Cy= 0.3749*
Experiment 8	*Cy= 0.9611*

On-line identification

Identification of dynamic equation of motion in real time is crucial process, which can be used in control of the mobile wheeled robots. On-line identification is based on fuzzy logic systems described previously and use of the anfis function (adaptive neuro-fuzzy inference) based on the genfis2 function combined with neural network.

On-line identification

Using available input and output data anfis function creates fuzzy logic system (FIS), which membership function parameters are tune with the use of back error propagation algorithm or combination of that algorithm with estimation of the least square method.

Online identification – anfis modeling

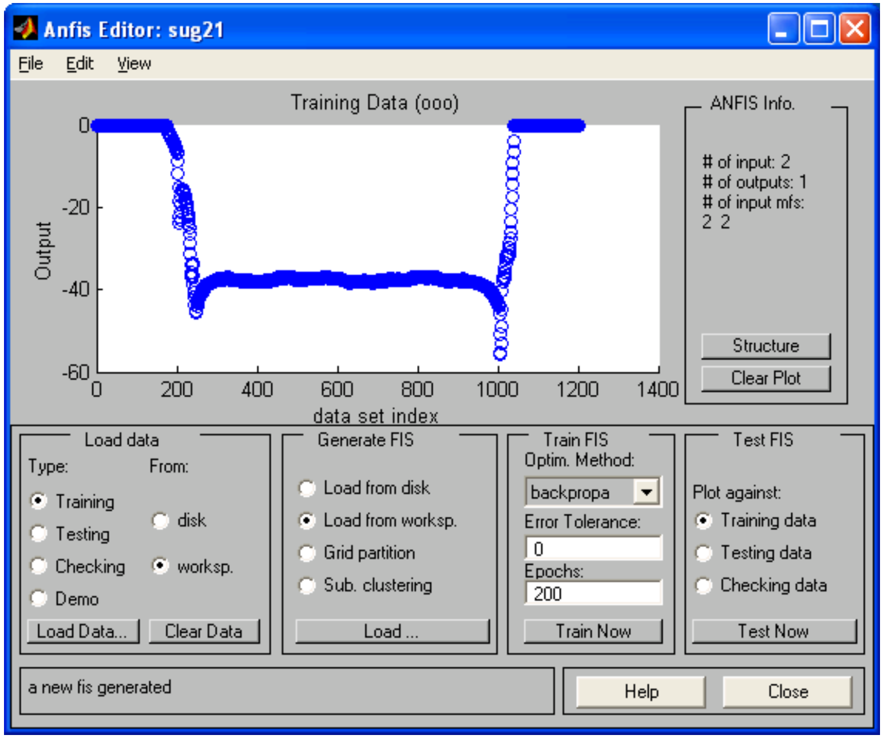
Firstly we assume structure of a parameterized model (assigning: inputs to membership functions, membership functions to rules, rules to outputs). Next we gather input/output data in a form which can be used by anfis in learning process with the FIS in order to imitate learning data presented by the modification of the membership function according to the chosen error criteria.

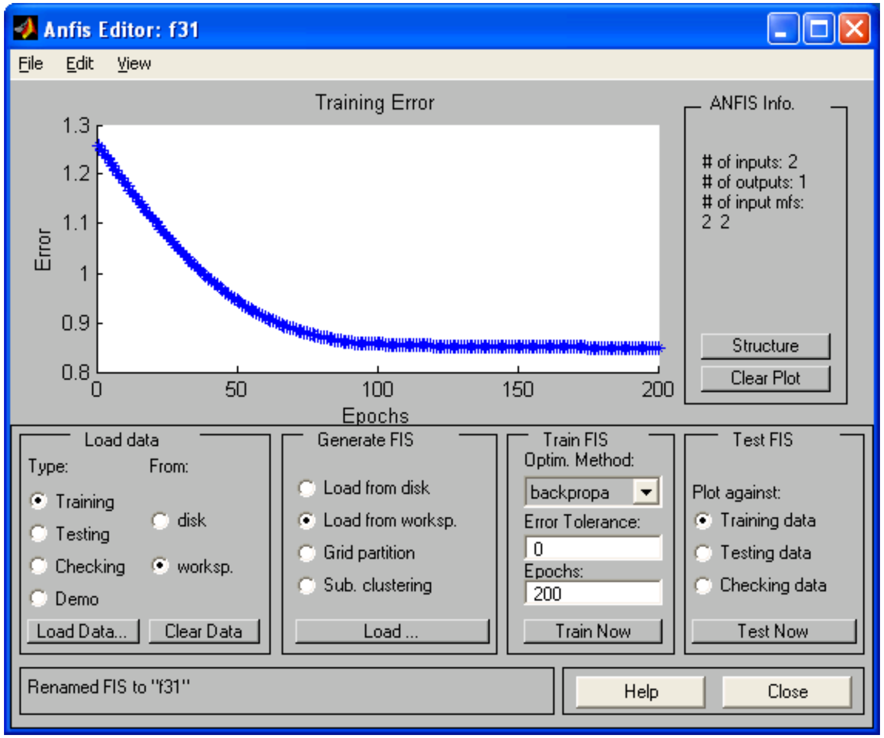
Online identification

In order to use anfis function we make use of automatically generated fuzzy logic systems with the use of the function *genfis2* (for the parallel structure of the state emulator). Generated two fuzzy models f31 and f41 for 8th experiment were used to teach new fuzzy logic systems with the use of the anfis function using the script for acquisition of data from computing model presented below.

```
in1=f3c(1,:)';
  in2=f4c(1,:)';
out1=f3row(1,:)';
  out2=f4row(1,:)';
data1=[in1,in2,out1];
  data2=[in1,in2,out2];
```

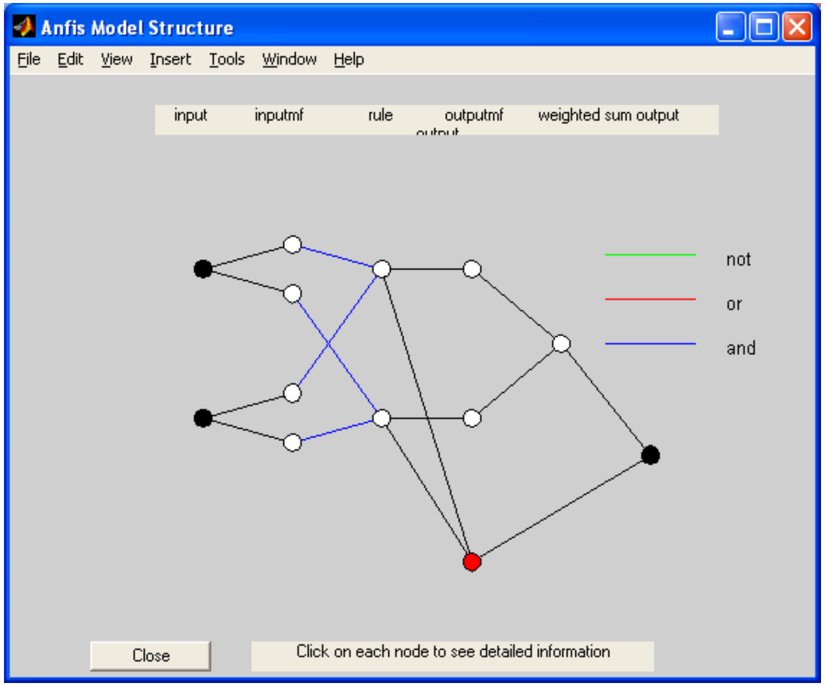
On-line identification





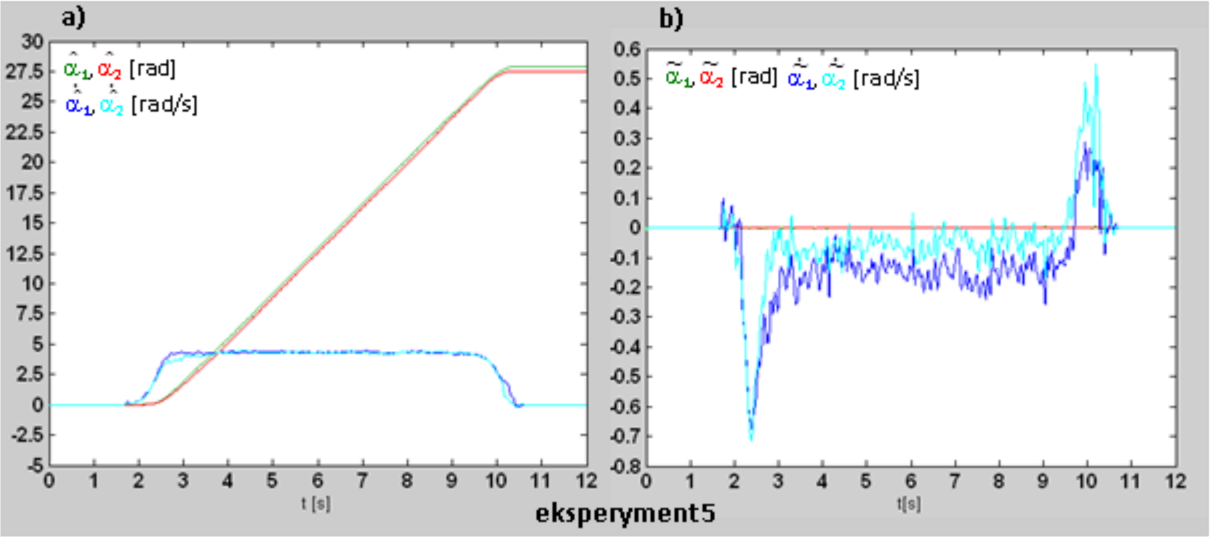
anfis function editor for new adaptive fuzzy logic system f31

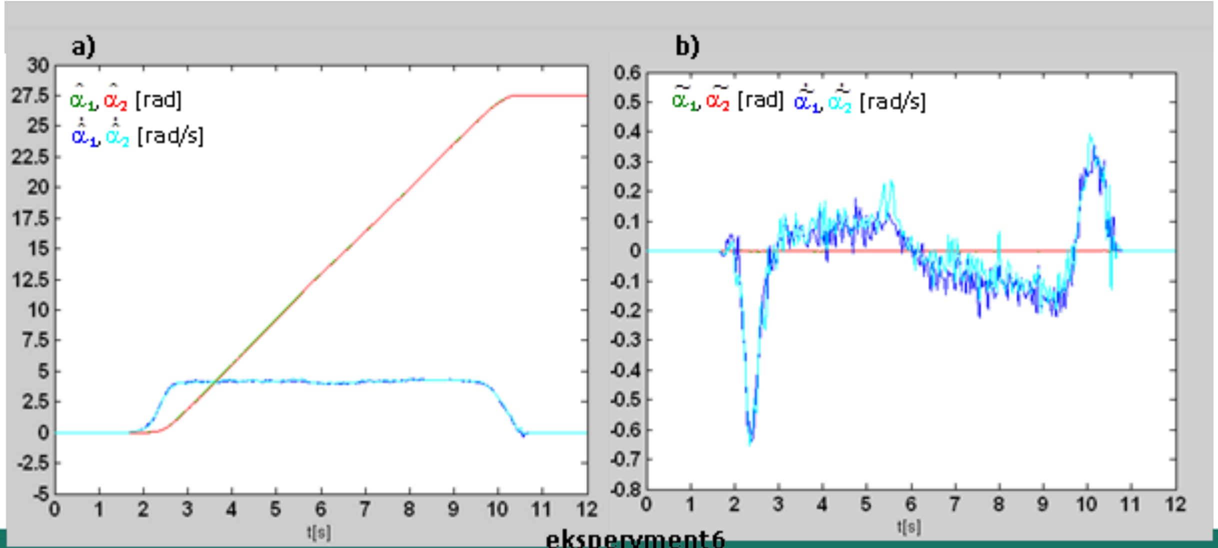
On-line identification



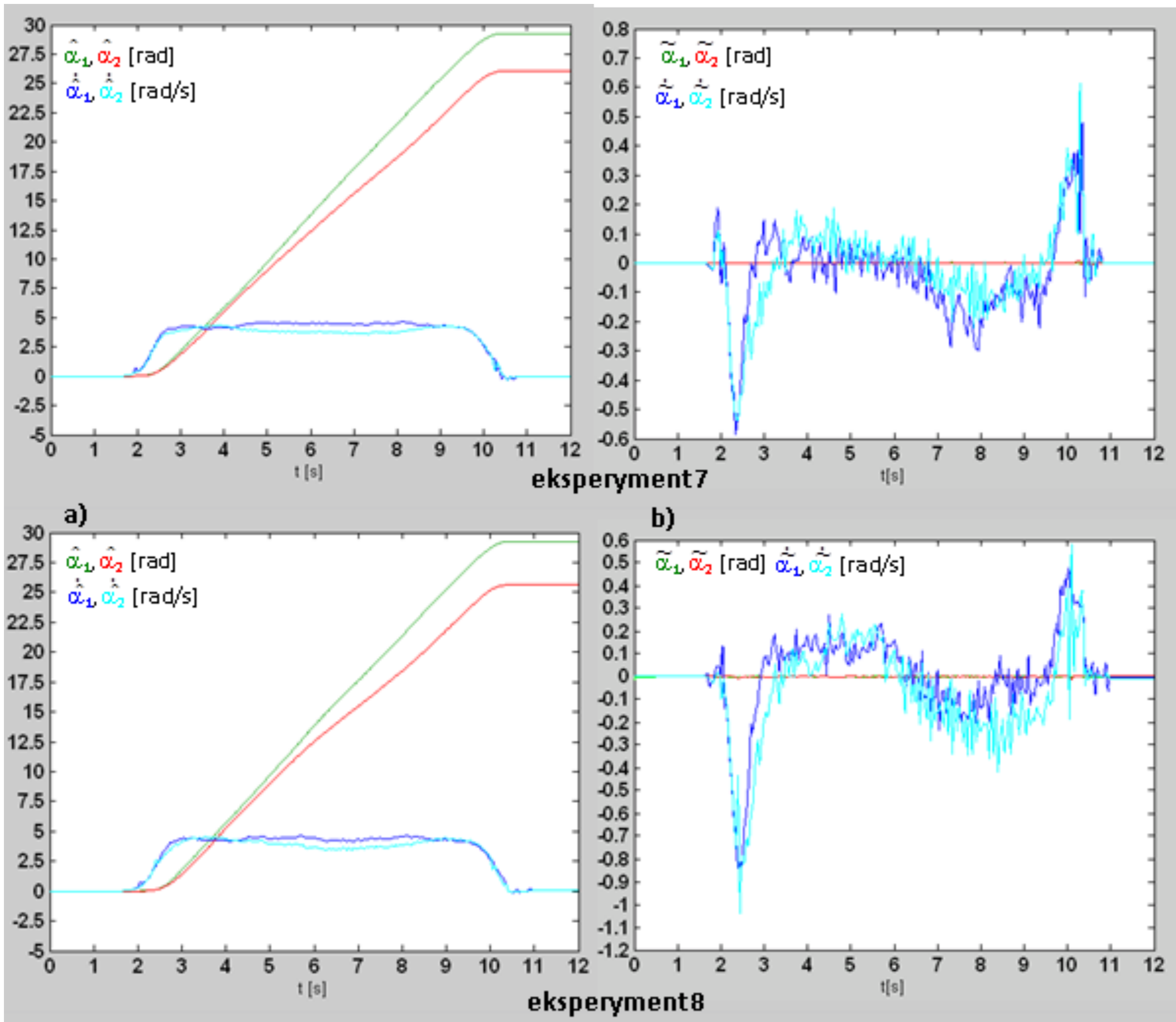
Structure of a neural network

On-line identification





On-line identification



On-line identification

Values of the quality coefficient Cy received for individual experiments with the use of parallel structure of state emulator

Experiment 5	*Cy=0.6755*
Experiment 6	*Cy=0.4836*
Experiment 7	*Cy=0.3846*
Experiment 8	*Cy=0.9640*

THANK YOU FOR YOUR ATTENTION