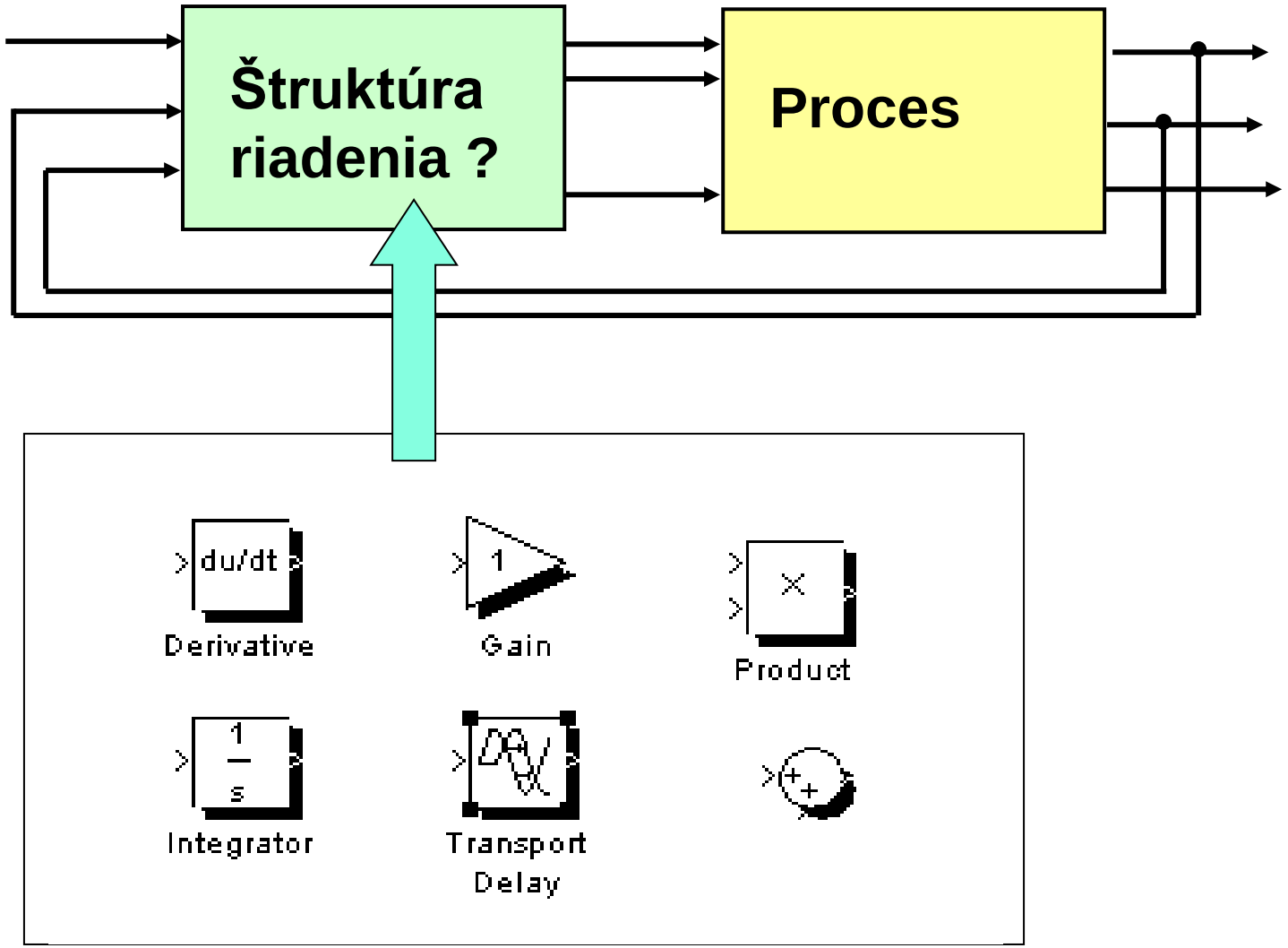
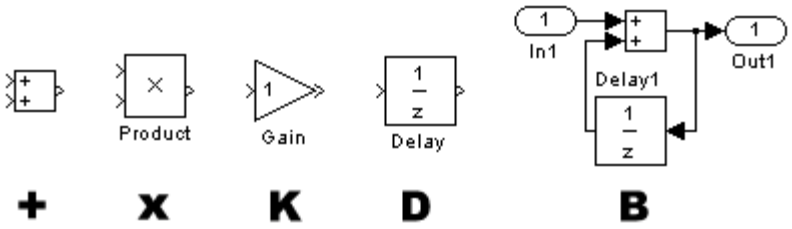
2.5 Návrh / optimalizácia vnútornej štruktúry regulátora použítm evolučných algoritmov

Genetické programovanie
Kartézske genetické programovanie
Gramatická evolúcia

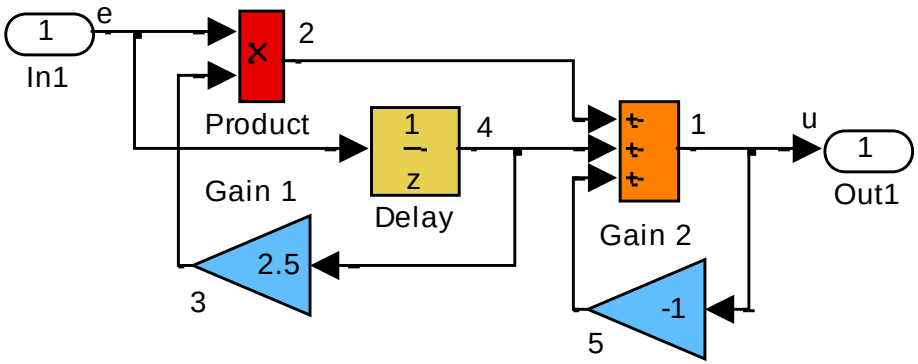
Návrh / optimalizácia štruktúry dynamického systému, jeho vnútorných väzieb a prvkov



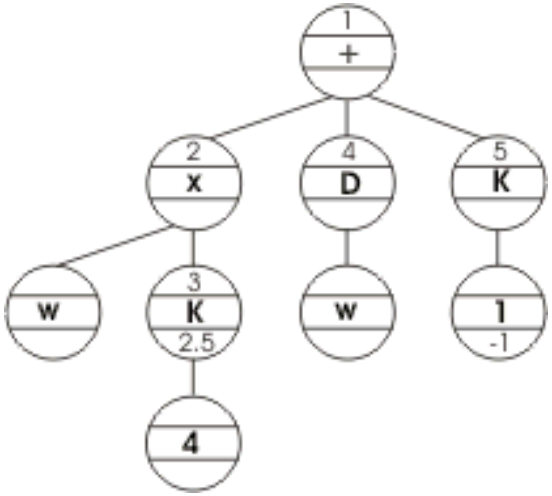
2.5.1 Evolúcia regulačného obvodu pomocou Genetického programovania



Stavebné prvky pre diskrétny prípad



Regulátor v Simulinku - fenotyp



Stromová reprezentácia

$$\begin{pmatrix} 3 & 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ + & x & w & K & 4 & D & w & K & 1 \\ 1 & 2 & & 3 & & 4 & & 5 & \\ & & & 2.5 & & & & -1 & \end{pmatrix}$$

Tabuľková reprezentácia - genotyp

Genetické programovanie (GP) pri návrhu regulačných obvodov

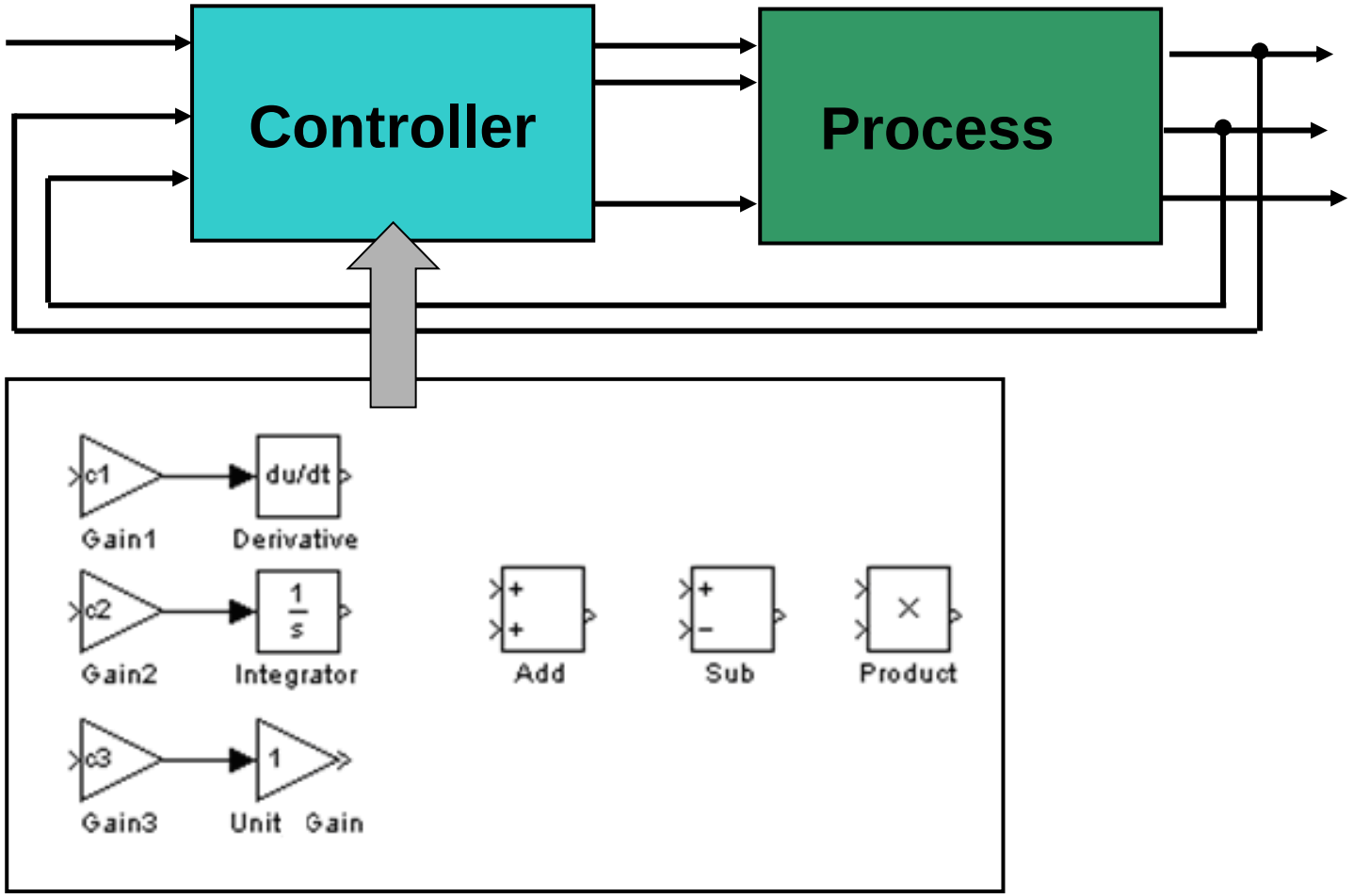
- Návrh aj štruktúry aj parametrov regulačného obvodu
- Rozvetvené regulačné obvody, nelineárne prvky, MIMO
- Neintuitívne riešenia, spravidla lepšie výsledky ako konvenčnými metódami návrhu
- Dlhý čas výpočtu (extrémne dlhý) hodiny-dni

2.5.2 Kartézske genetické programovanie pri návrhu regulačných obvodov

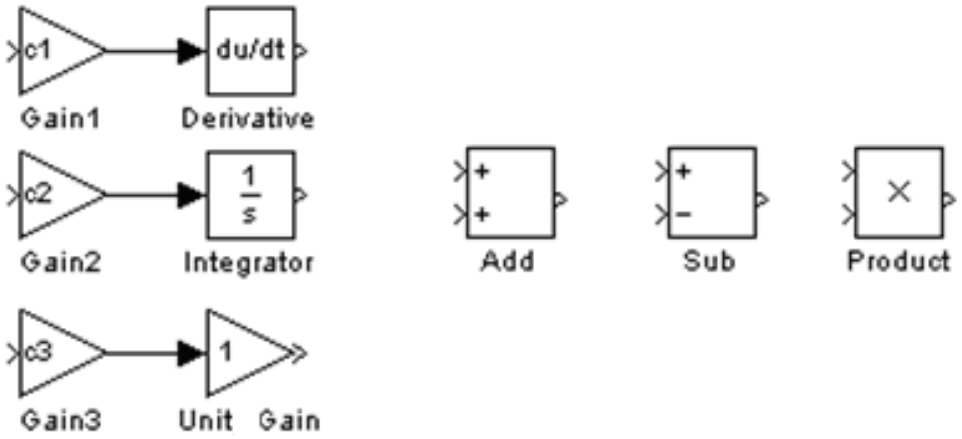
- **Zjednodušenia a obmedzenia aby sa redukovala výpočtová náročnosť GP**
  - obmedzený počet stavebných blokov
  - obmedzené možnosti ich prepojenia (viď príklad ďalej)
  - prvky (gény) sú spravidla organizované v pravouhlej mriežke
- **Väčšina vlastností GP zostane zachovaná.**

Kartézke (genetické) programovanie – návrh regulátora

spojitý prípad



Knižnica nami použitých stavebných blokov



Reprezentácia jedinca

- Každý jedinec polulácie je reprezentovaný reťazcom ktorý pozostáva z [bt, bg, ic, oc, mc].
- bt – typ operácie (zosilnenie, integrátor, derivátor)

  bg – konštanta zosilnenia

  ic – číslo predchádzajúceho modulu

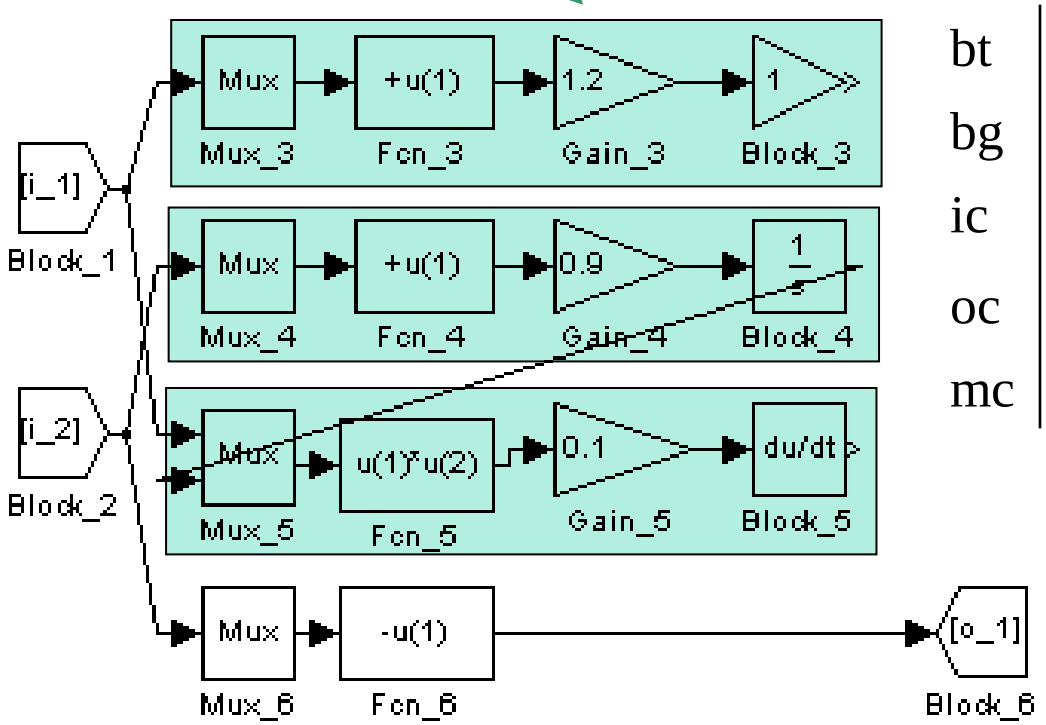
  oc – číslo nasledujúceho modulu

  mc – matematická operácia (sčítavanie odčítavanie, násobenie)
- bt, bg sú vektory dĺžky N a ic, oc, mc sú vektory dĺžky M

Príklad reprezentácie jedinca

Jedinec: [1 2 3 1.2 0.9 0.1 1 1 2 4 2 3 5 4 5 6 1 3 1 3 2]

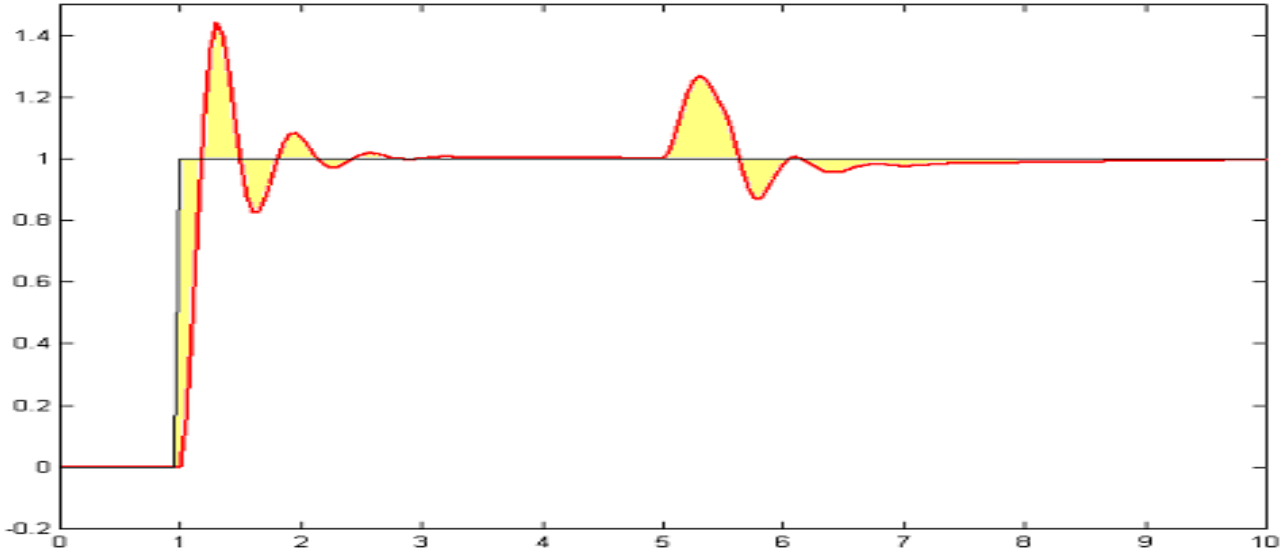
Stavebné bloky



bt	1	2	3			N
bg	1.2	0.9	0.1			N
ic	1	1	2	4	2	M
oc	3	5	4	5	6	M
mc	1	3	1	3	2	M

Fitness funkcia (účelová funkcia)

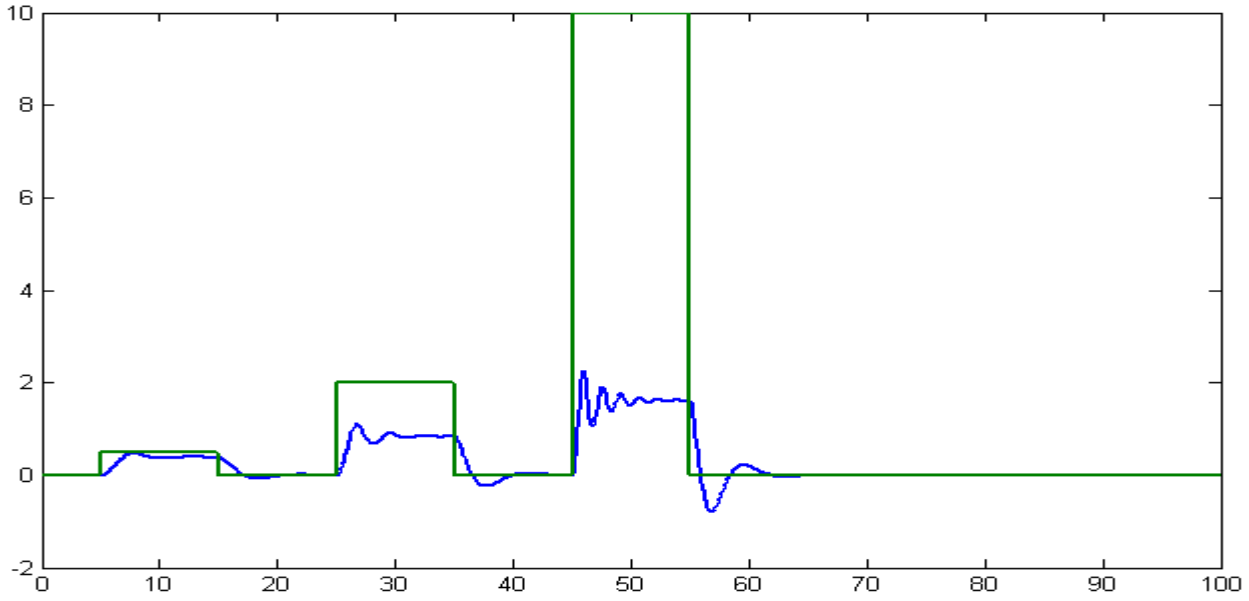
$$I_{AE} = \int_0^T |e(t)| dt \rightarrow \min$$



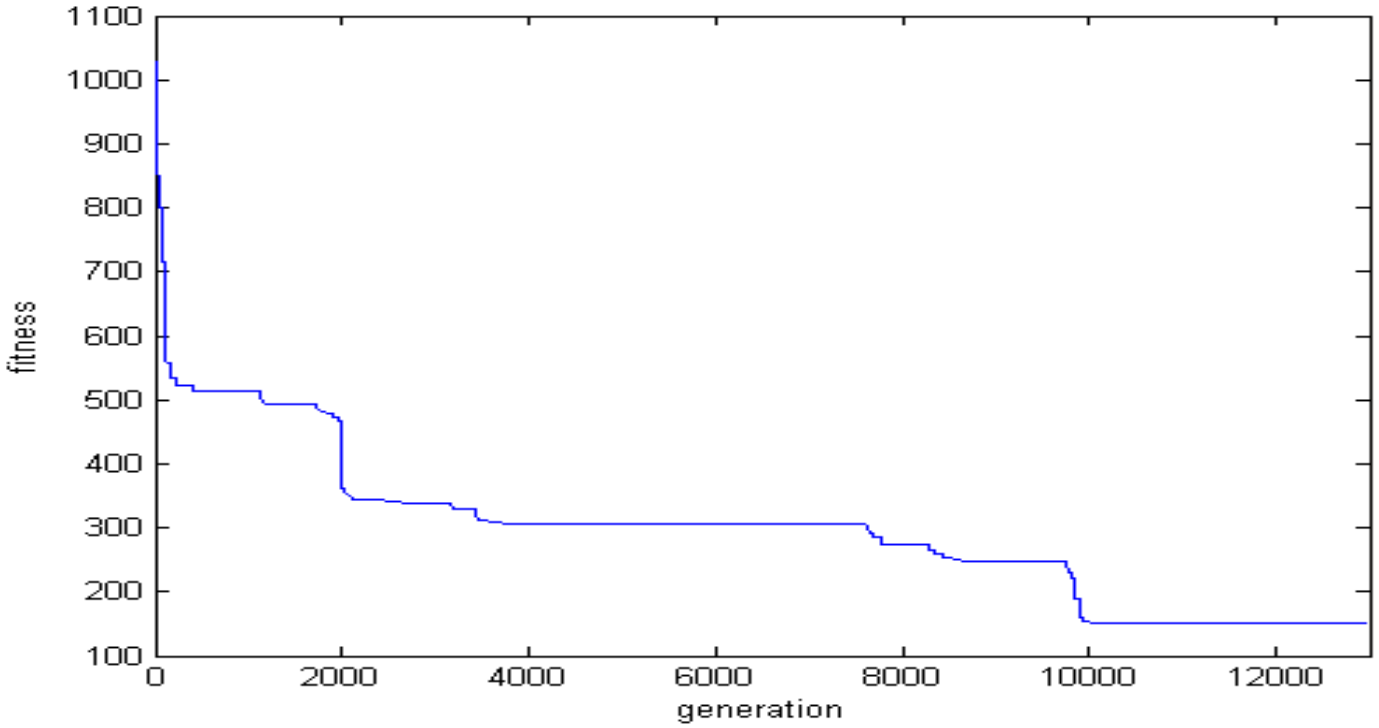
Príklad: Nelineárny SISO- systém

prechodové charakteristiky systému

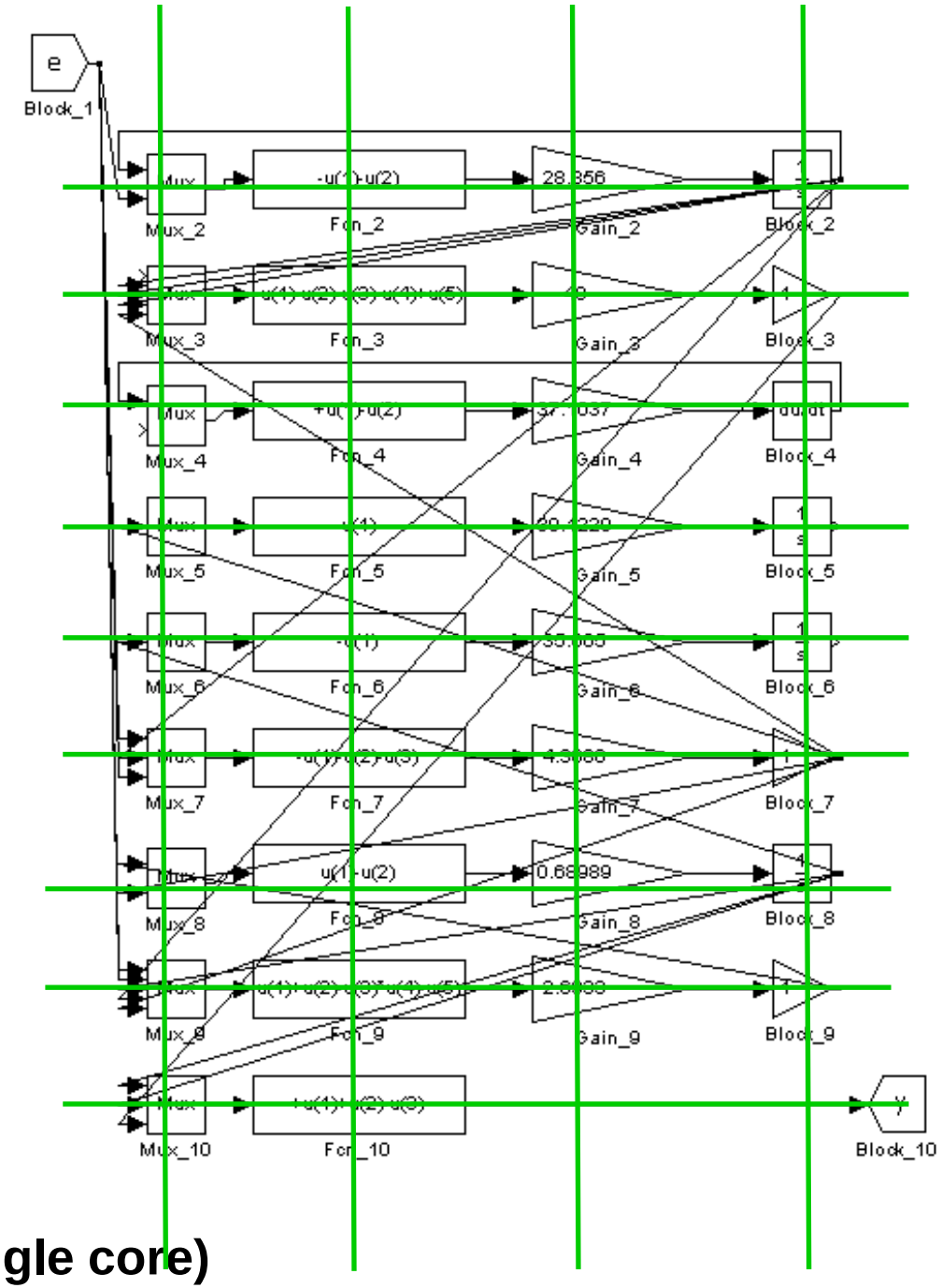
$$y'' + y' + y + 2y^3 - u = 0$$



Graf evolúcie fitness najlepšieho jedinca



Regulátor získaný pomocou Kartézkeho programovania

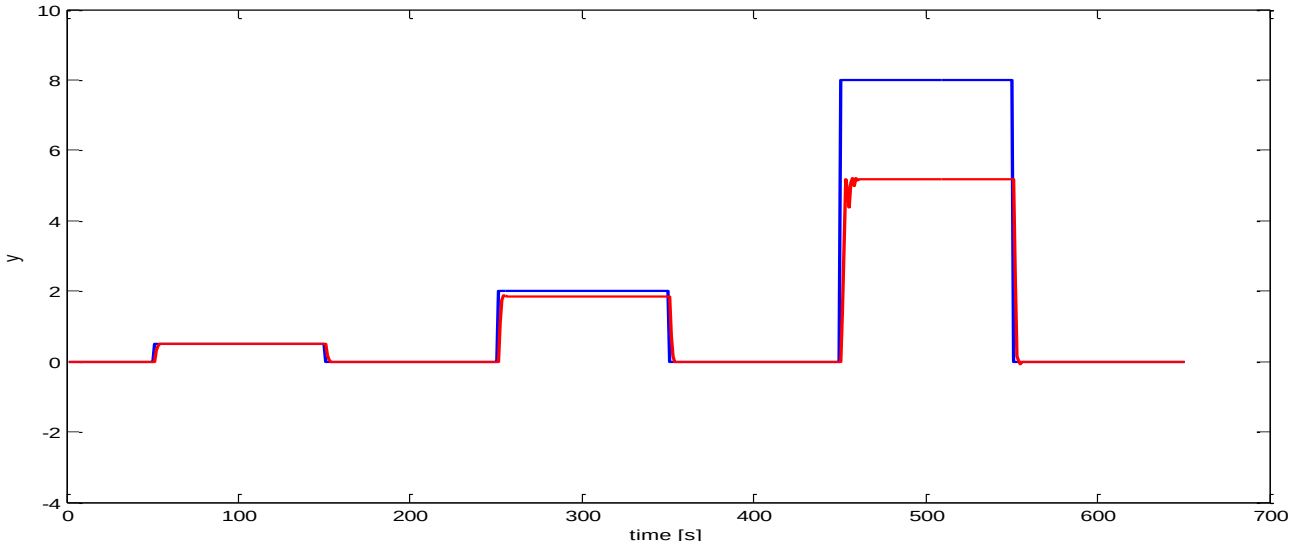


Karteziánska mriežka stavebných prvkov

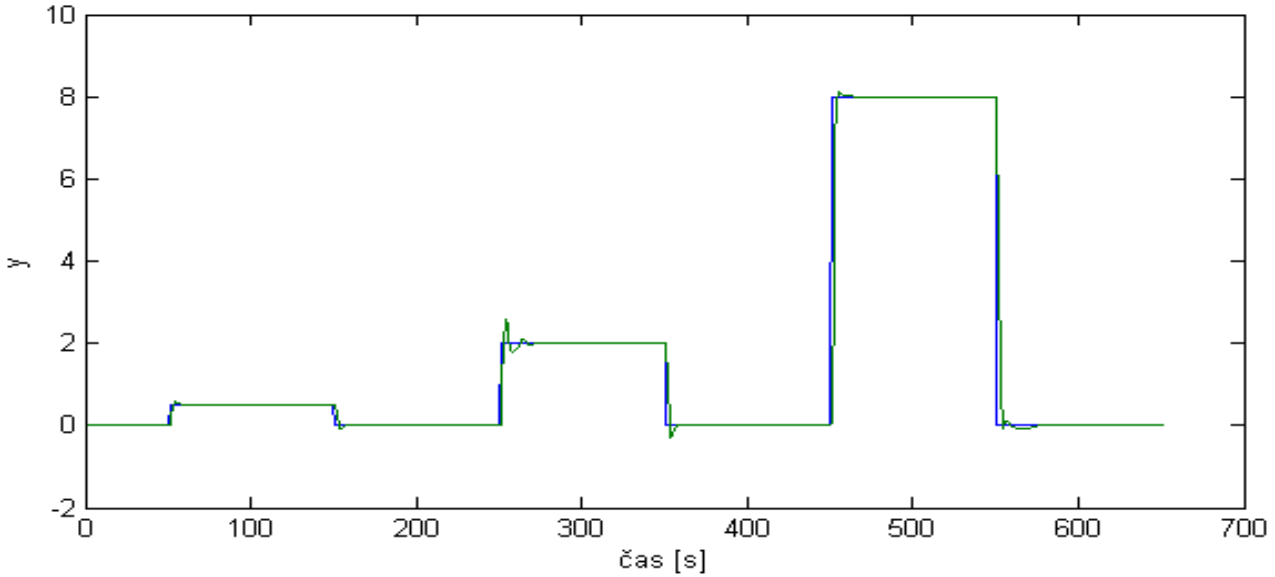
Computation time: 40 hours (single core)

Comparation of the PID controller and controller designed by Cartesian programming

GA-PID



Cartesian programming

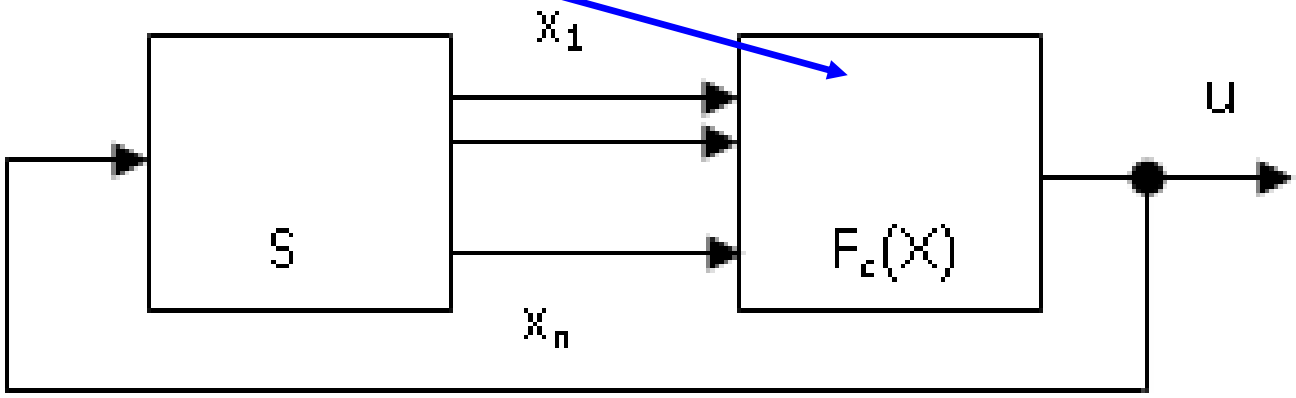


2.5.3 Grammatical Evolution Controller design

$u_k = F_c(e, ie, de, d^2e, y, dy, other)$

$F_c(X)=?$

e – control error
y – controlled value
i – integral
d - derivative



Controlled System

Controller function

For the controller function $F_c(X)$ – the grammatical evolution is used

Individual representation - F_c

Set of mathematical operations = {+, -, *, /, ^} (other...)

individual={$f_1,f_2,\ldots,f_n$, $x_1, x_2,\ldots,x_n$, $p_1,p_2,\ldots,p_n$, $b_1,b_2,\ldots,b_n$ }

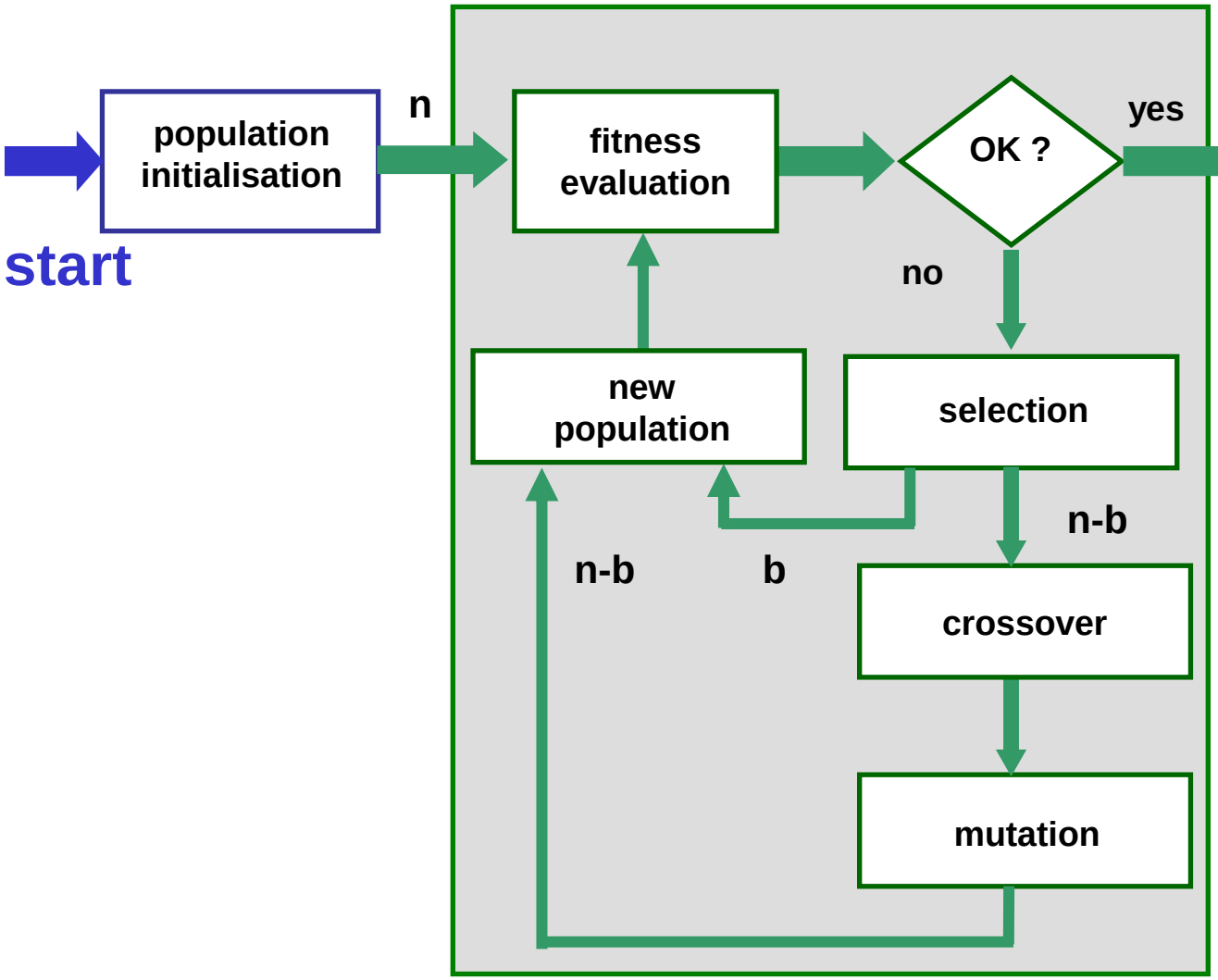
- f_i - mathematical operation
- x_i - argument – controller input variable
- p_i - multiplicative coefficient of each input variable x_i
- b_i - coefficient of the power operation (if neccessary).

f_i in the considered grammar is:

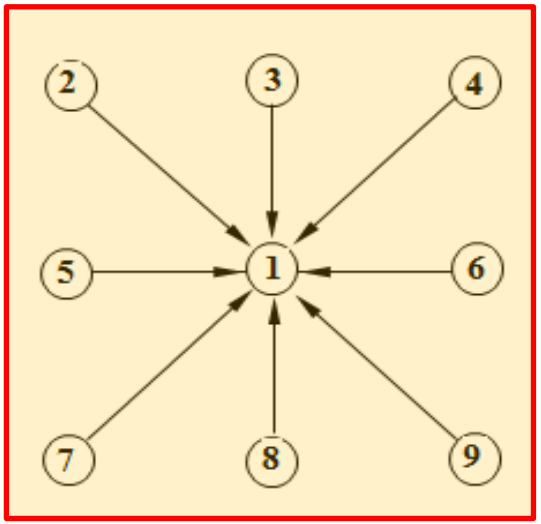
0: $\lambda \rightarrow \lambda$, 1: $\lambda \rightarrow (\lambda + \lambda)$, 2: $\lambda \rightarrow (\lambda - \lambda)$, 3: $\lambda \rightarrow (\lambda * \lambda)$,
4: $\lambda \rightarrow (\lambda / \lambda)$, 5: $\lambda \rightarrow (\lambda)^b$,

The symbol λ is substituted by: $\lambda_i = p_i x_i$.

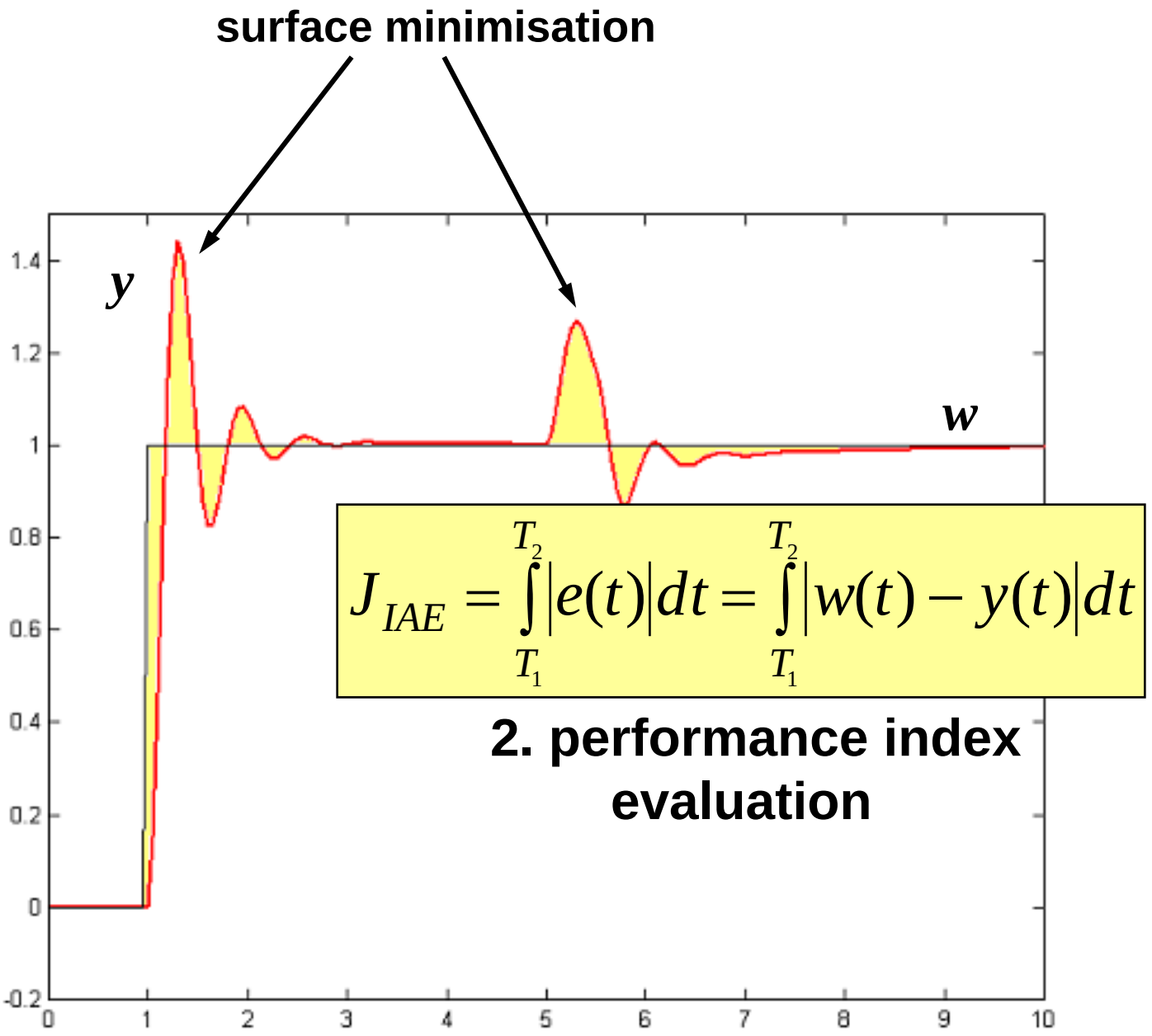
Evolution



Parallel GE



Fitness



$$J_{IAE} = \int_{T_1}^{T_2} |e(t)| dt = \int_{T_1}^{T_2} |w(t) - y(t)| dt$$

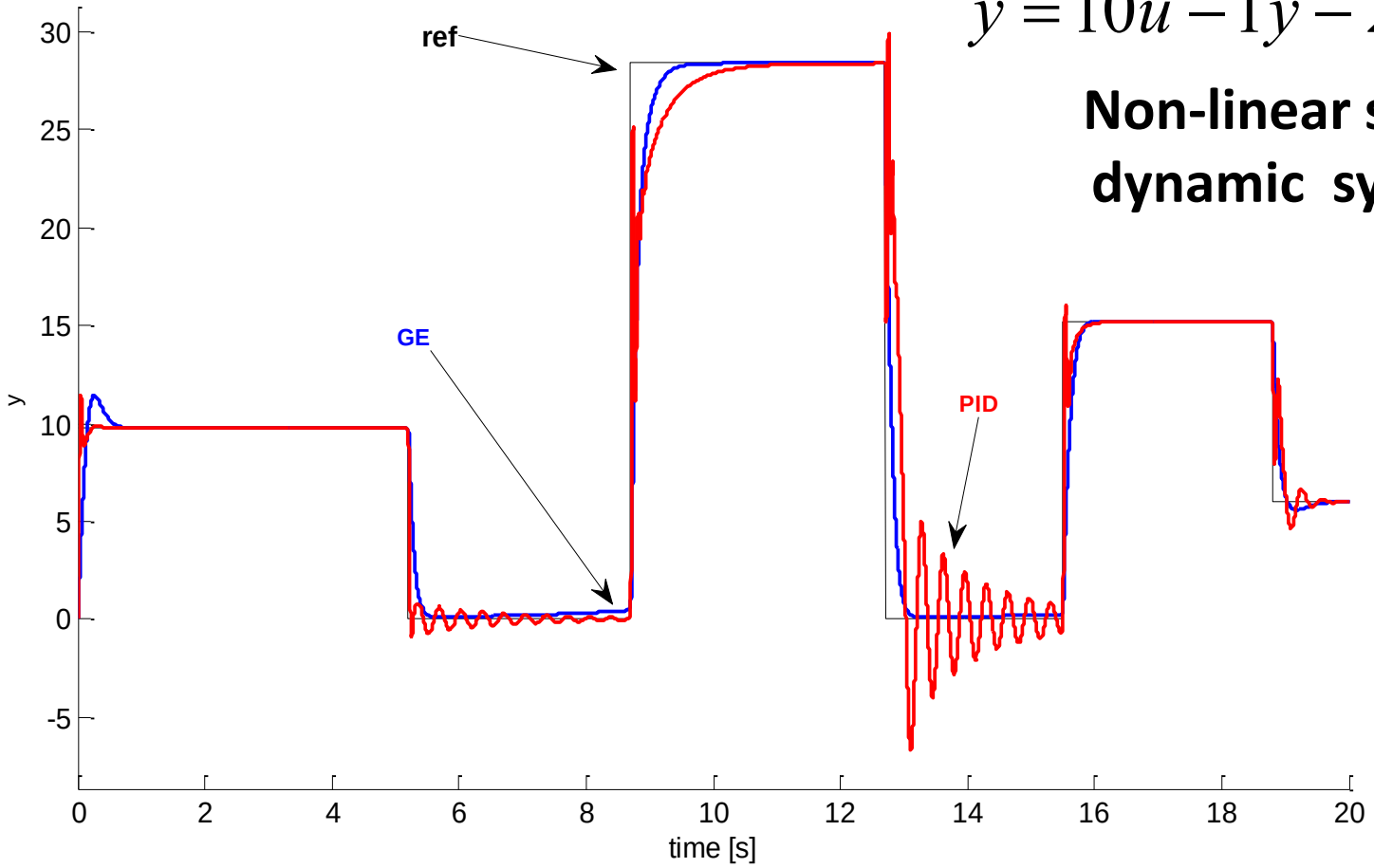
2. performance index evaluation

1. Closed loop simulation (Simulink)

Case study:

$$\ddot{y} = 10u - 1\dot{y} - 2y - 4y^3$$

Non-linear stable dynamic system

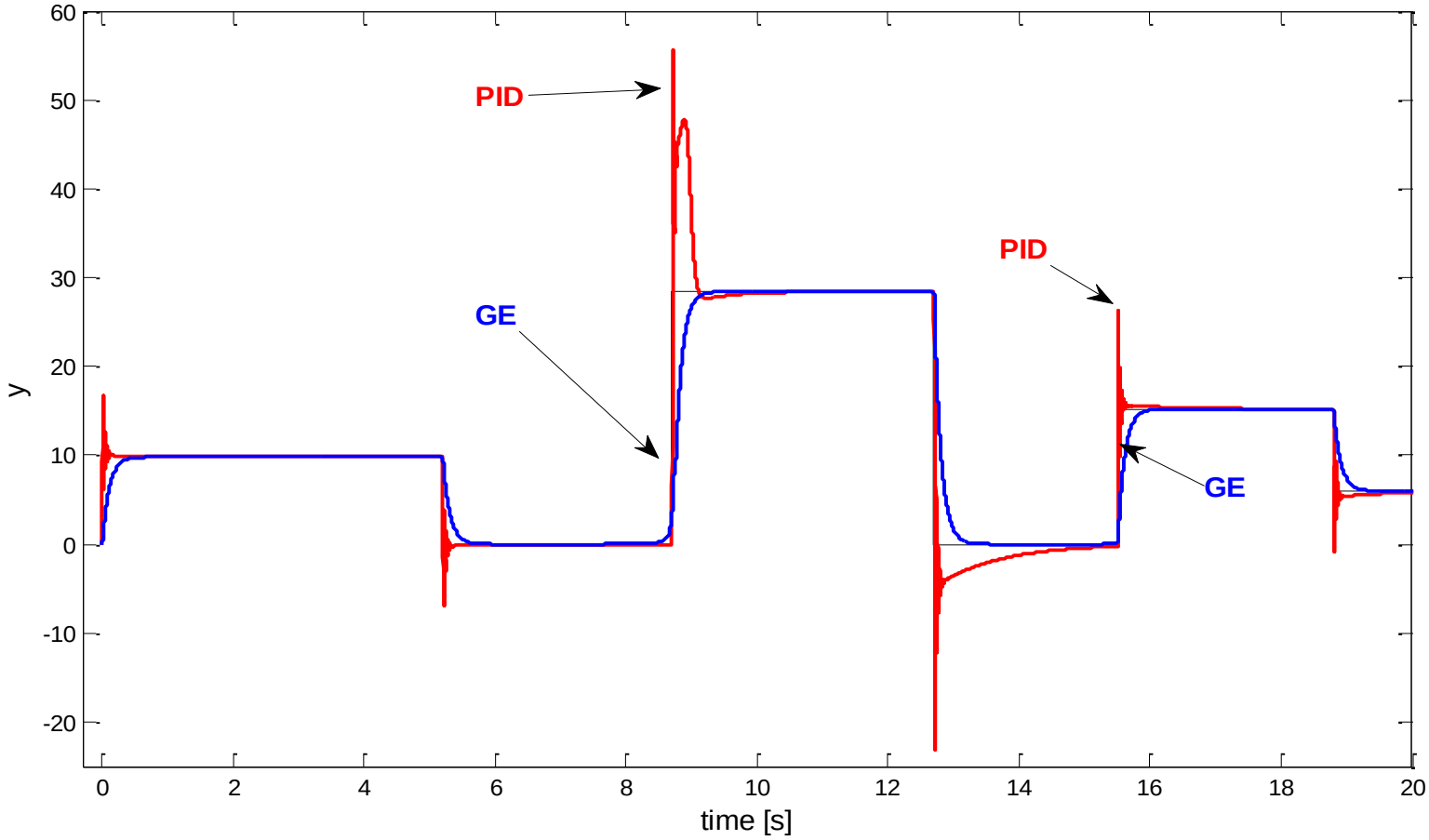


+	+	-	*	(	+	+	+	)	-
98.21	80.77	17.35	1.4988	-	4.79	64.72	90.52	-	-
e	e	dy	ie	-	e	ie	y	-	-

$$F_c: \quad u = 178.98e - 17.35y' + 1.4988\left(\int e\right)(4.79e + 64.72\int e + 90.52y)$$

Non-linear unstable dynamic system

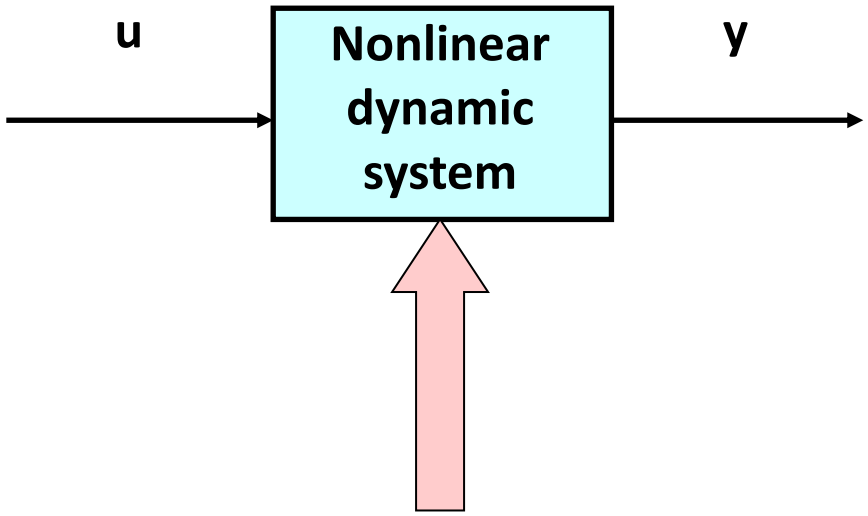
$$\ddot{y} = 10u - 1\dot{y} - 2y + 4y^3$$



$$F_c: \quad u = 183.135e - 21.923y' - 2.518y + 0.0113\left(\int e\right)(1.832y - 16.653\int e)$$

2.5.4 Grammatical Evolution – based model design

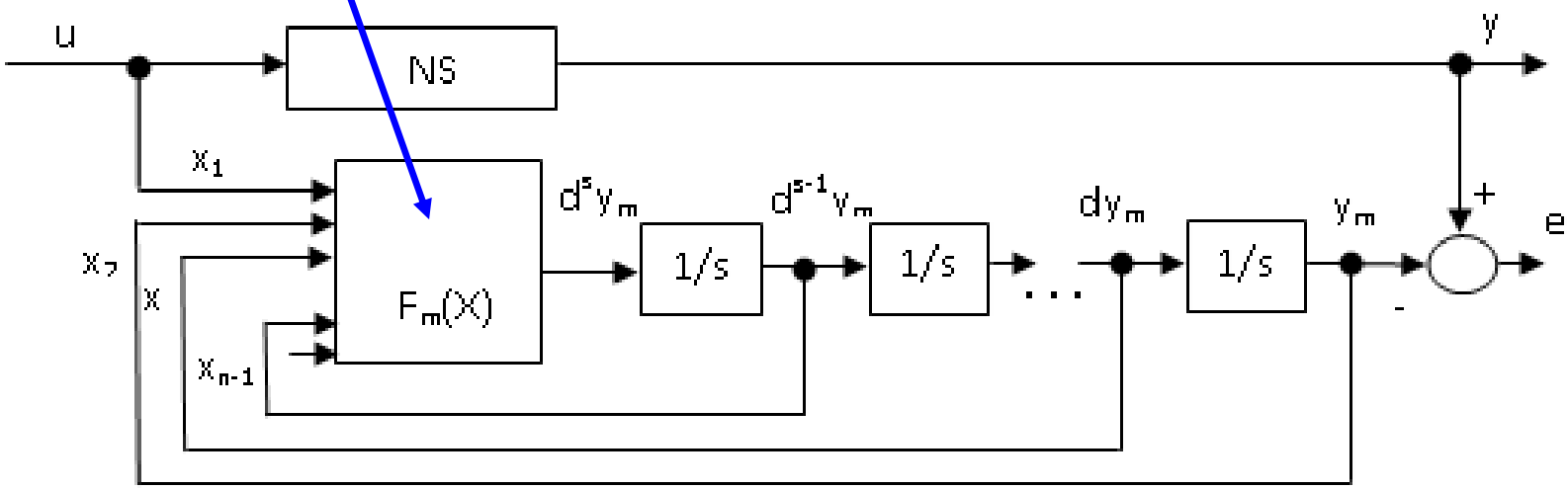
$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), u(t))$$
$$y(t) = h(\mathbf{x}(t))$$



Unknown structure of the model
f=?, h=?
and unknown number of parameters

$y = F_m(u, y_m, dy_m, \ldots, d^{s-1}y_m)$

$F_m(X) = ?$



Individual representation - F_m

Set of mathematical operations = {+, -, *, /, ^} (other...)

individual={$f_1,f_2,\ldots,f_n$, $x_1, x_2,\ldots,x_n$, $p_1,p_2,\ldots,p_n$ }

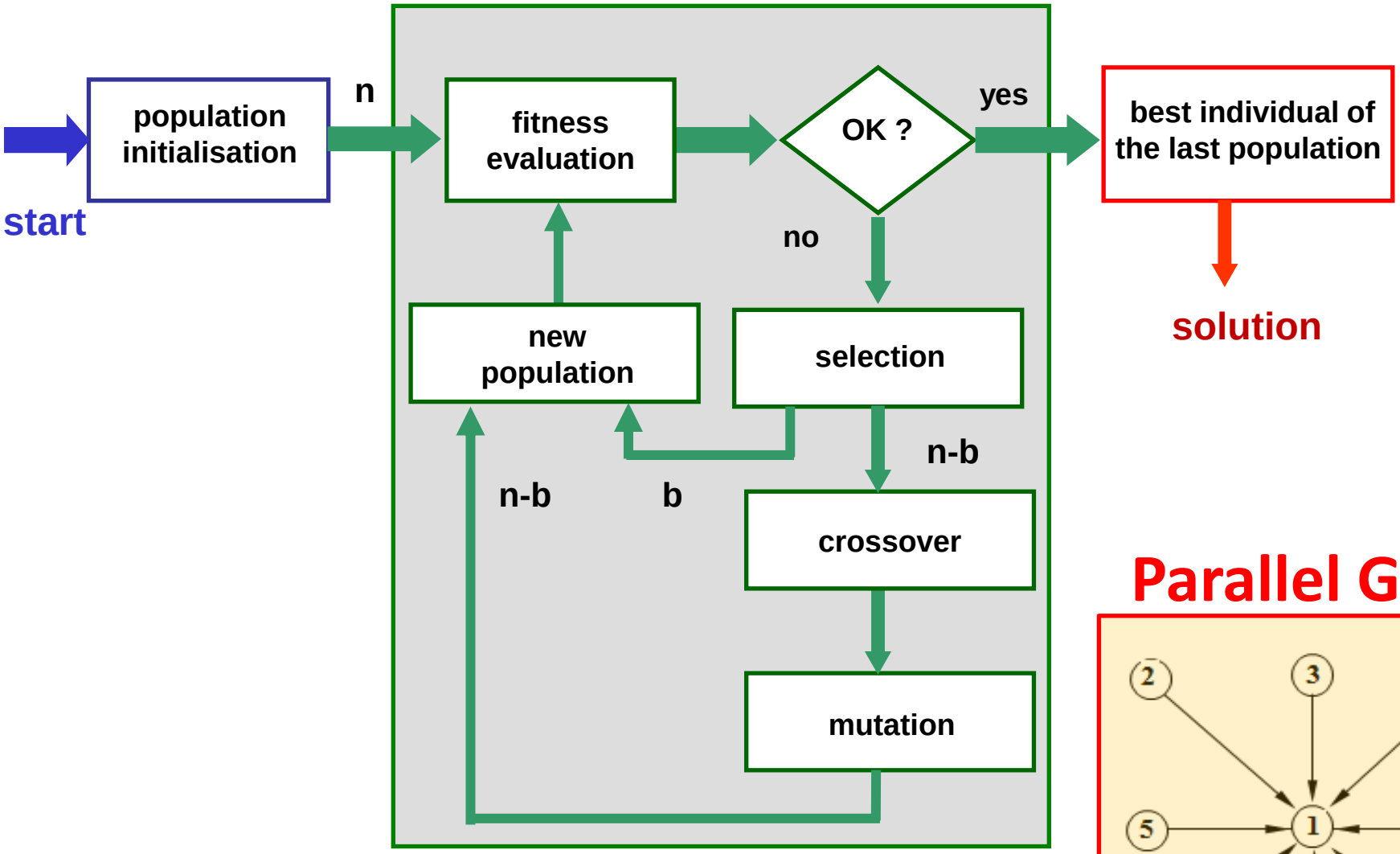
- f_i - mathematical operation
- x_i - argument – model variable
- p_i - multiplicative coefficient of each variable x_i

f_i in the considered grammar is:

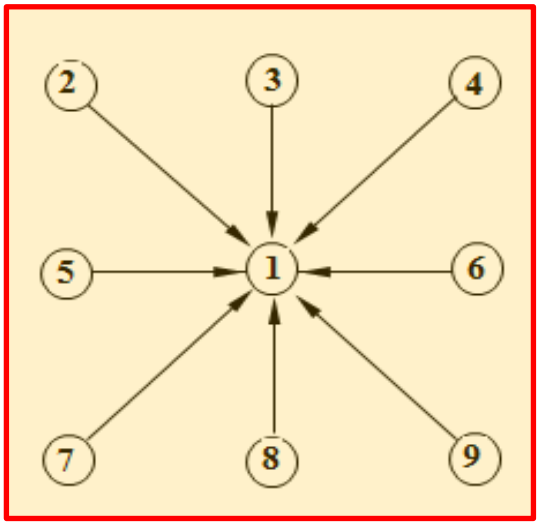
0: $\lambda \rightarrow \lambda$, 1: $\lambda \rightarrow (\lambda + \lambda)$, 2: $\lambda \rightarrow (\lambda - \lambda)$, 3: $\lambda \rightarrow (\lambda * \lambda)$,
4: $\lambda \rightarrow (\lambda / \lambda)$, 5: $\lambda \rightarrow (\lambda)$,

The symbol λ is substituted by: $\lambda_i = p_i x_i$.

Evolution

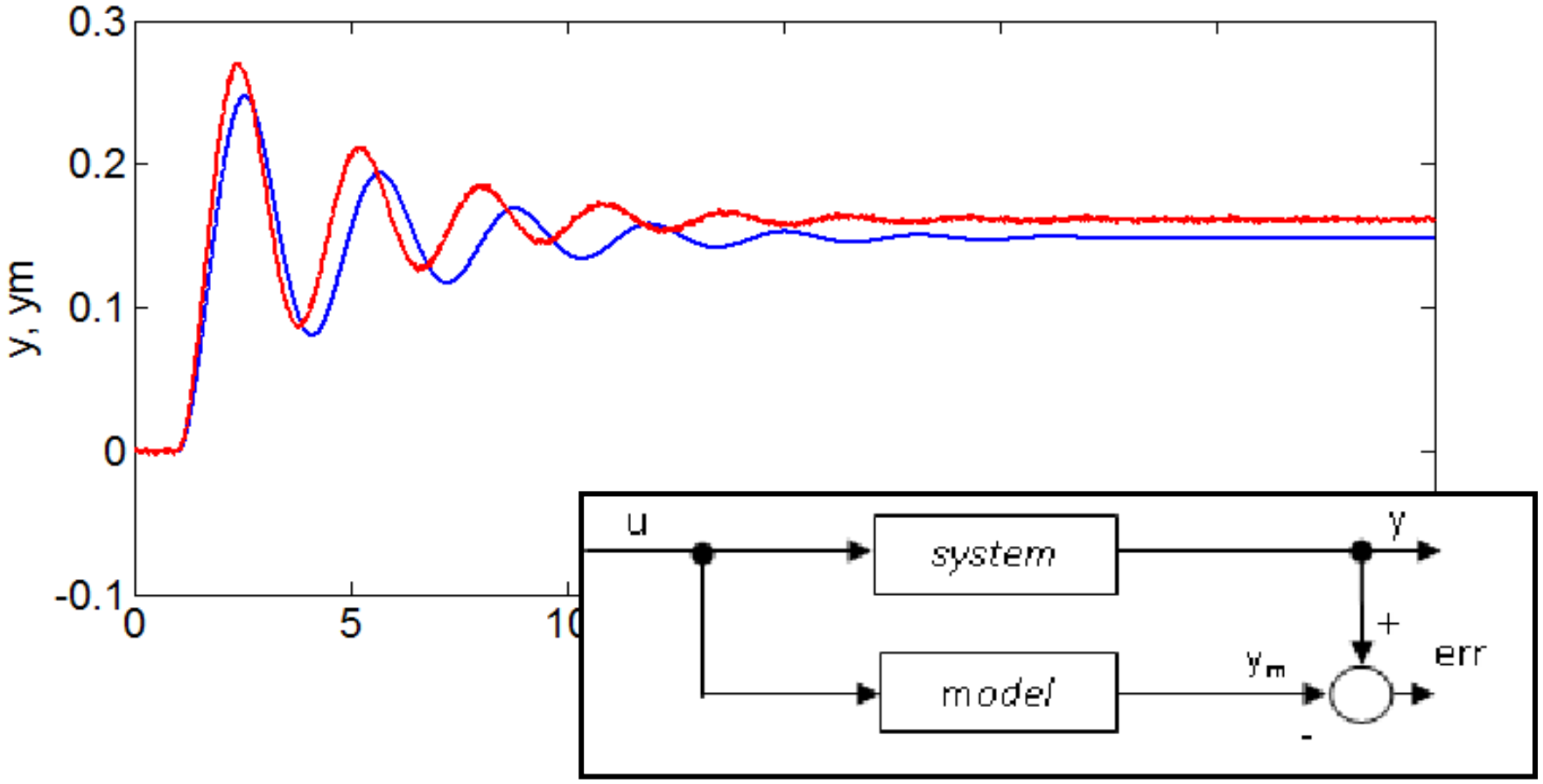


Parallel GE



Fitness

$$J = \sum_{i=1}^{N} (y_i - ym_i)^2 \rightarrow \min$$



Model simulation (Simulink)

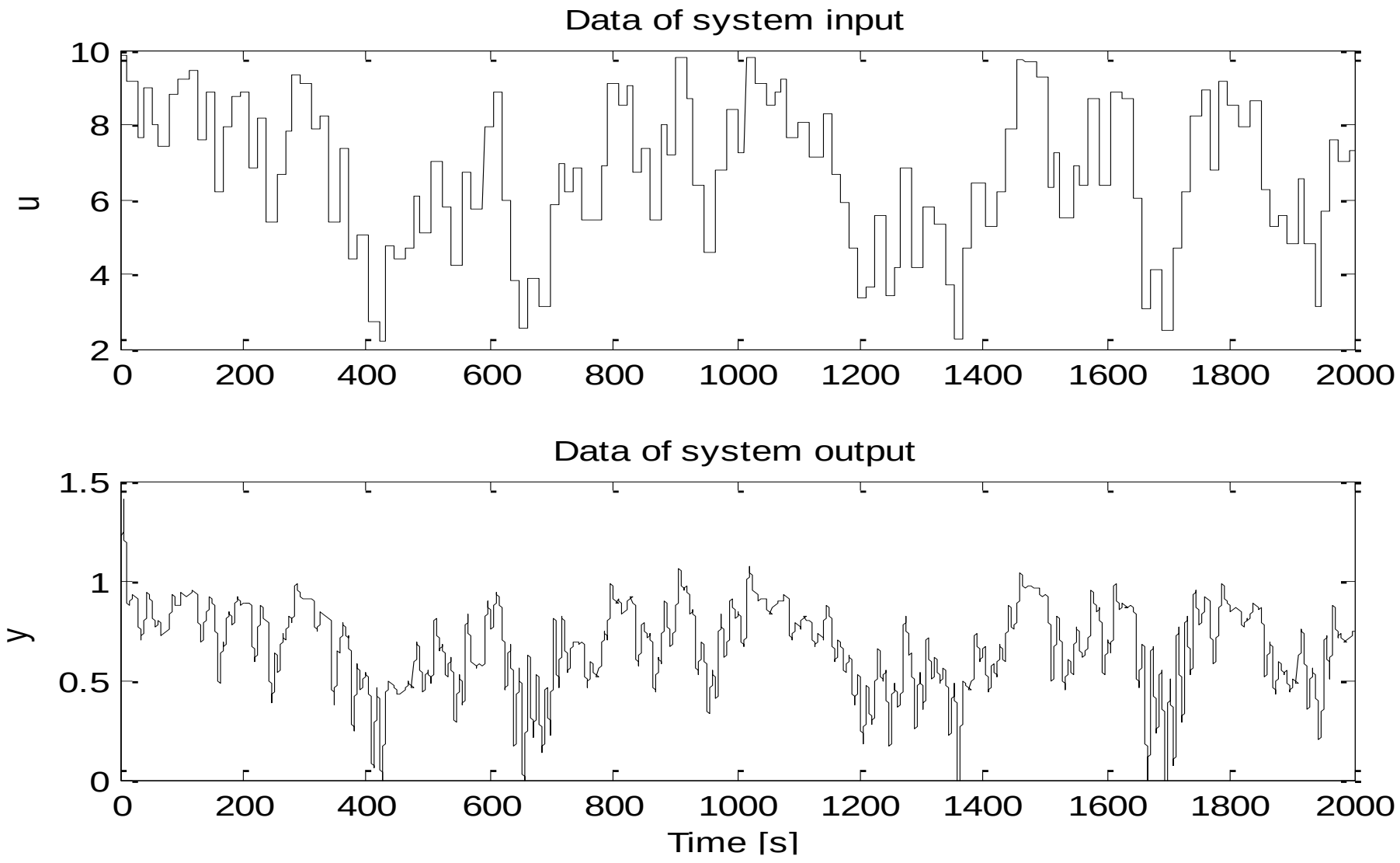
Case study

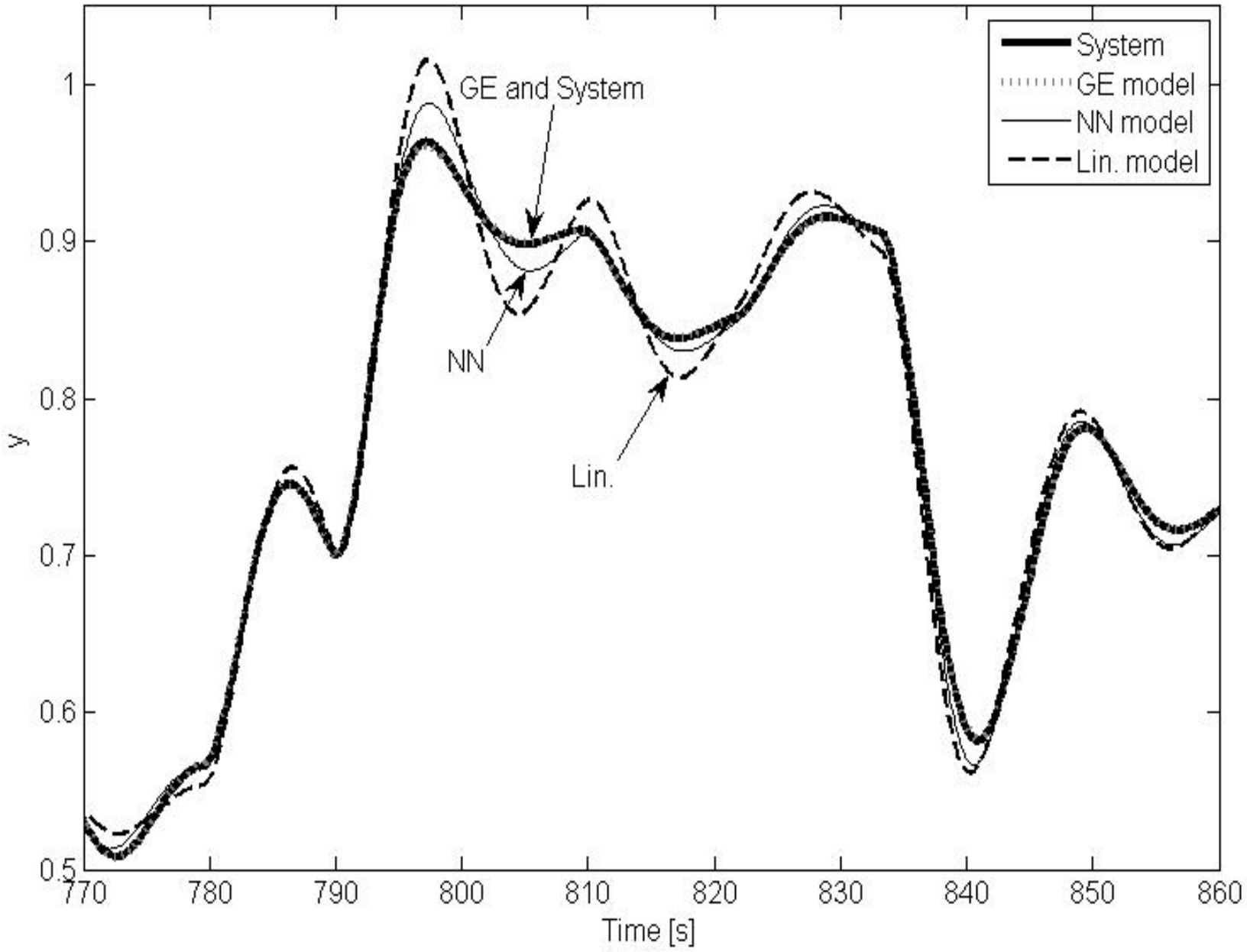
Modeled non-linear dynamic system

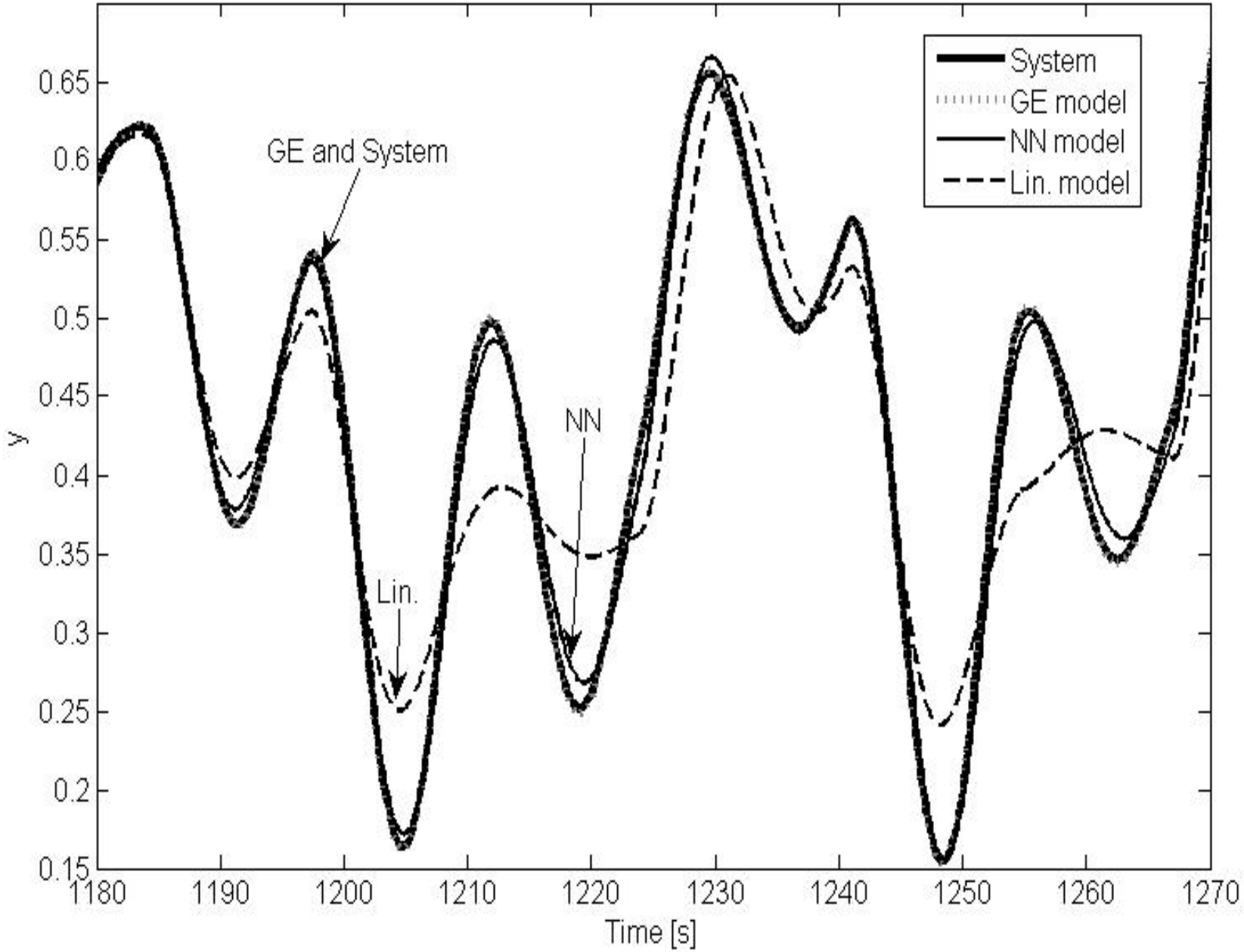
$$\dddot{y} = 0.2u - 10\ddot{y} - 5\dot{y}y^2 - 2y$$

Obtained GE-Model

$$\dddot{y} = -11.0\dot{y}y^2 + (\dot{y}(7.5\dot{y} + 3.7) - 3.9)y - 20.0\ddot{y} + 0.39u -$$
$$-1.0\dot{y}(1.7\dot{y} - 14.0\ddot{y} + 0.25u + 8.0\ddot{y}\dot{y} + 2.2\dot{y}^2)$$


Data of system input
u
10
8
6
4
2
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Data of system output
y
1.5
1
0.5
0
Time [s]


System
GE model
NN model
Lin. model
GE and System
NN
Lin.
y
Time [s]
1
0.9
0.8
0.7
0.6
0.5
770
780
790
800
810
820
830
840
850
860


System
GE model
NN model
Lin. model
GE and System
NN
Lin.
y
Time [s]
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270

Results

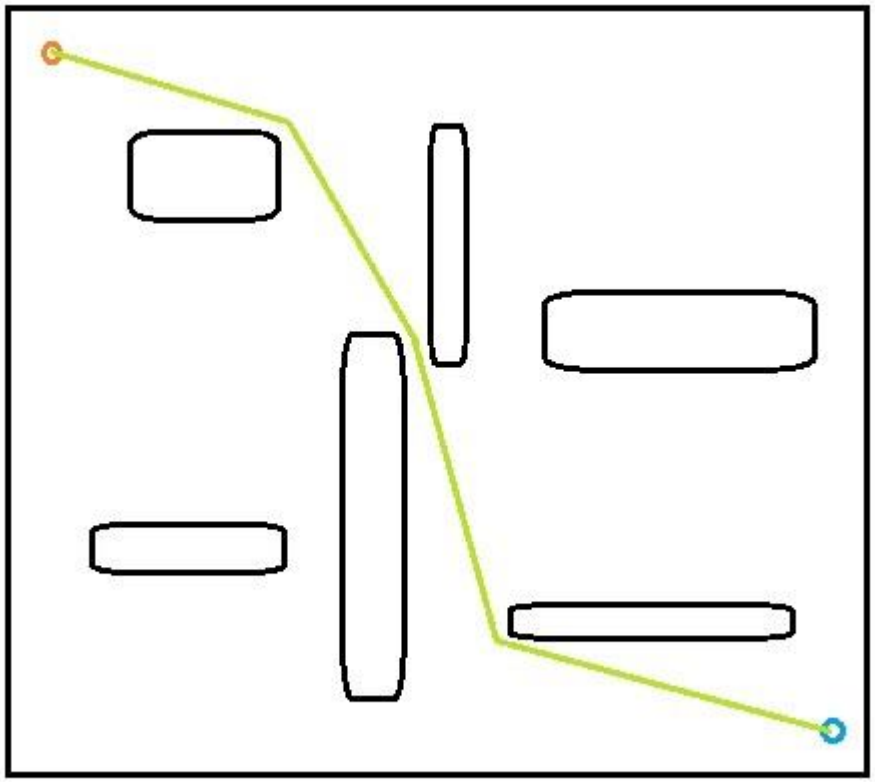
Method	Training data	Testing data
GE-based model	1.4881e-006	1.4052e-006
Neural model	8.0076e-005	4.6814e-005
Linear model	9.8008e-004	5.2461e-004

2.6 Evolučné algoritmy v robotike

Plánovanie trasy mobilného robota

- návrh/optimalizácia trasy mob. robota v 2D (3D) prostredí
- chromozóm obsahuje vhodne reprezentovanú množinu bodov trajektórie priestoru, ktorá je geneticky modifikovaná
- minimalizácia dĺžky/náročnosti trasy
- statické prostredie – jednorázový návrh
- dynamické prostredie – preplánovávanie trasy (výpočtový čas?)
- Prístupy: GA, GP, PSO, ACO...

Optimálna dráha medzi prekážkami v známom prostredí

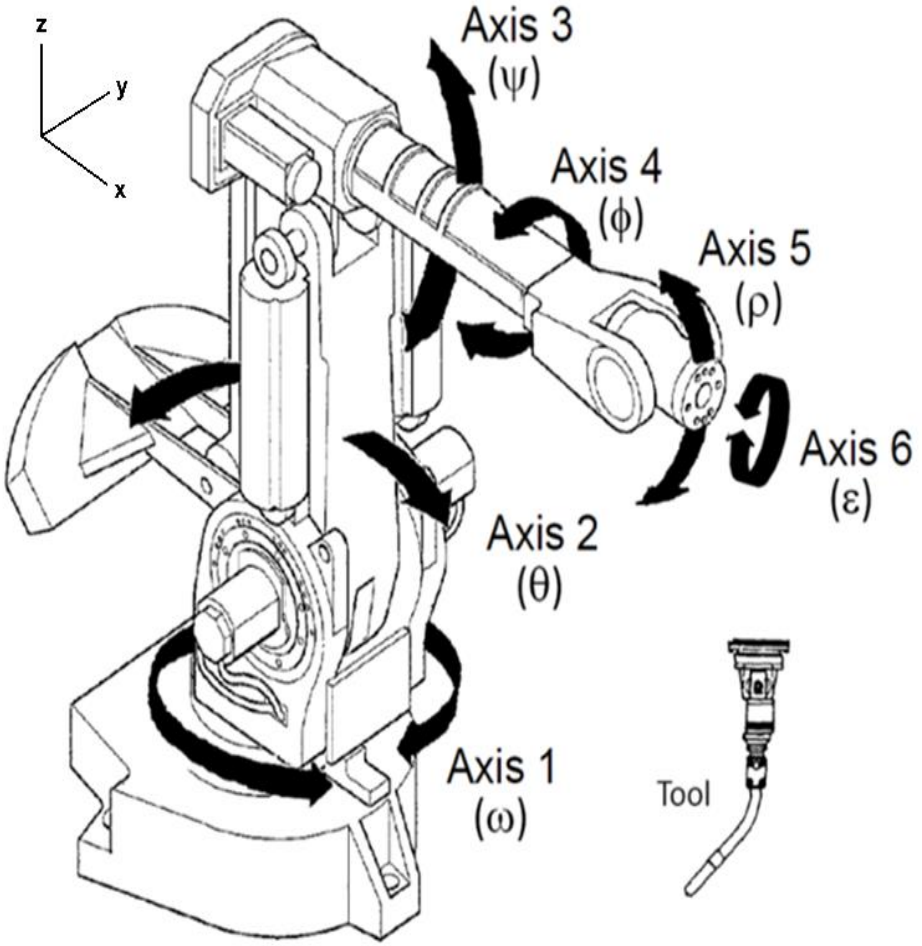


Optimalizácia trajektórie robotického ramena

- návrh/optimalizácia trajektórie robotického manipulátora
- cieľ – bezkolízny, časovo/energeticky optimálny pohyb ramena
- chromozóm obsahuje vhodne reprezentovanú množinu (zlomových) bodov trajektórie pohybu, ktorá je v priebehu evolúcie geneticky modifikovaná
- minimalizácia dĺžky dráhy/času, minimalizácia počtu riadiacich krokov, min. energie
- eliminácia kolízií
- statické prostredie – jednorázový návrh
- dynamické prostredie – preplánovávanie trasy, zabránenie kolíziám
- koordinácia pohybu viacerých ramien

Optimalizácia trajektórie robotického ramena – príklad (Robot ABB IRB 6400)

http://www.youtube.com/watch?v=p2wCSyl_f6M



Definícia úlohy (Inverzná kinematická úloha)

- **Robot s *n* stupňami voľnosti má uhly rotácie v kĺboch $\alpha_1, \alpha_2, ..., \alpha_n$.**
- **Vykonáva operáciu v *N* bodoch P_1 až P_N v pravotočivom karteziánskom súradnicovom systéme $P_i[x_i;y_i;z_i]$.**
- **Každý pracovný bod P_i (koncový bod manipulátora) zodpovedá *n*-uhlom rotácie kĺbov:**

$$P_i[x_i;y_i;z_i] \leftrightarrow f(\alpha_{1i}, \alpha_{2i},...,\alpha_{ni}); i \in \{1,2,3,...,N\}.$$

- **Cieľom je nájsť optimálnu trajektóriu robota, ktorá optimalizuje zvolené kritérium.**

Transformačný model

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = TM * \begin{bmatrix} R6x \\ 0 \\ R6z \\ 1 \end{bmatrix}$$

$$TM = A*B*C*D*E*F*G*H*I$$

$$P_i[x_i;y_i;z_i] \leftrightarrow f(\alpha_{1i}, \alpha_{2i},\ldots,\alpha_{ni})$$

$$A = \begin{pmatrix} \cos(\omega) & -\sin(\omega) & 0 & 0 \\ \sin(\omega) & \cos(\omega) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 & R1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & R2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E = \begin{pmatrix} \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 1 & 0 & 0 & R3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} \cos(\rho) & 0 & \sin(\rho) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\rho) & 0 & \cos(\rho) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad I = \begin{pmatrix} 1 & 0 & 0 & R4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$J = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varepsilon) & -\sin(\varepsilon) & 0 \\ 0 & \sin(\varepsilon) & \cos(\varepsilon) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Úloha

- **Aká má byť postupnosť natočení uhlov jednotlivých kĺbov $\alpha_1, \alpha_2, ..., \alpha_n$**
- **pre každý pracovný bod *i*, $i \in \{1,2,3,...,N\}$: $(\alpha_{1i}, \alpha_{2i},...,\alpha_{ni})$,**
- **aby bola dosiahnutá požadovaná presnosť polohovania a súčasne bolo minimalizované ďalšie kritérium:**
  - ***a) energia celého pohybového cyklu***
  - ***b) čas operácie celého pohybového cyklu***
  - ***c) suma uhlov natočenia vo všetkých kĺboch v celom pohybovom cykle***

Genetický algoritmus

Chromozóm: $R=\{\alpha_{11}, \alpha_{21},..., \alpha_{n1}, \ ... \ , \alpha_{1N}, \alpha_{2N},..., \alpha_{nN}\}$,

Fitness = presnosť_polohovania + w . ďalšie_kritérium

Presnosť polohovania

$$D_{tr} = \sum_{i=1}^{N} \sqrt{[x_{w,i} - x_{GA,i}]^2 + [y_{w,i} - y_{GA,i}]^2 + [z_{w,i} - z_{GA,i}]^2}$$

Ďalšie kritériá

Energia z bodu do bodu: $$E_{p2p} = \sum_{i=1}^{n} [(\alpha_{b,i} - \alpha_{a,i})] * EP_{r,i}$$

a) Energia celého pohybu: $$E_{tr} = \sum_{j=1}^{N} \sum_{i=1}^{n} [(\alpha_{b,i,j} - \alpha_{a,i,j})] * EP_{r,i}$$

$EP_{r,i}$ - energia i-teho kĺbu na 1° rotácie

b) Čas operácie celého cyklu:

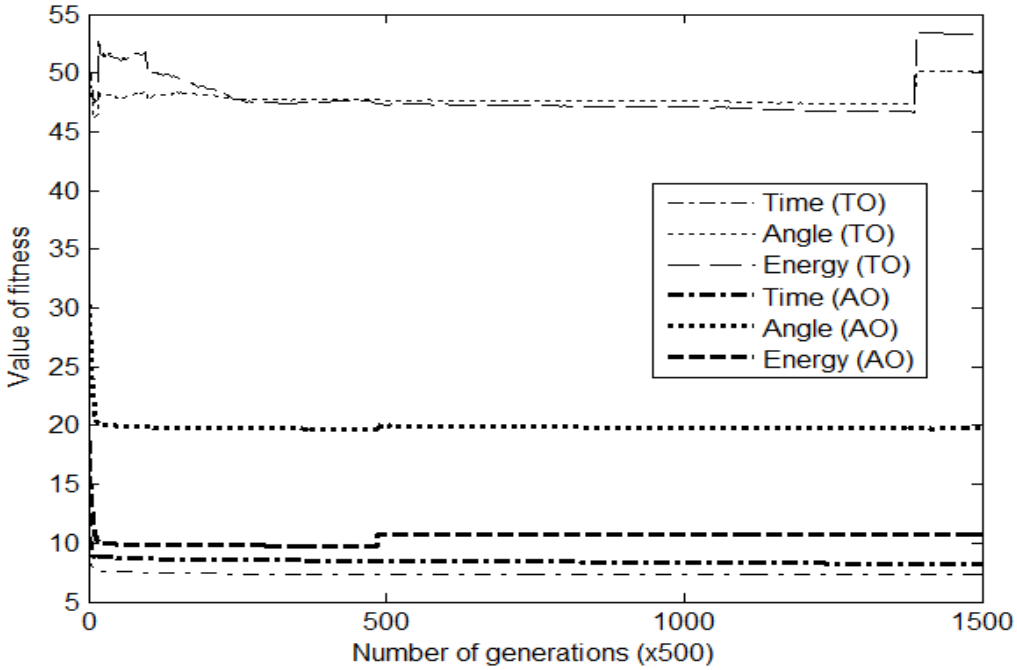
$$\boldsymbol{T_{tr}} = \sum_{j=1}^{N} \max_{i=1..n} \left[\left(\alpha_{b,i,j} - \alpha_{a,i,j} \right) \right] * TP_{r,i}$$

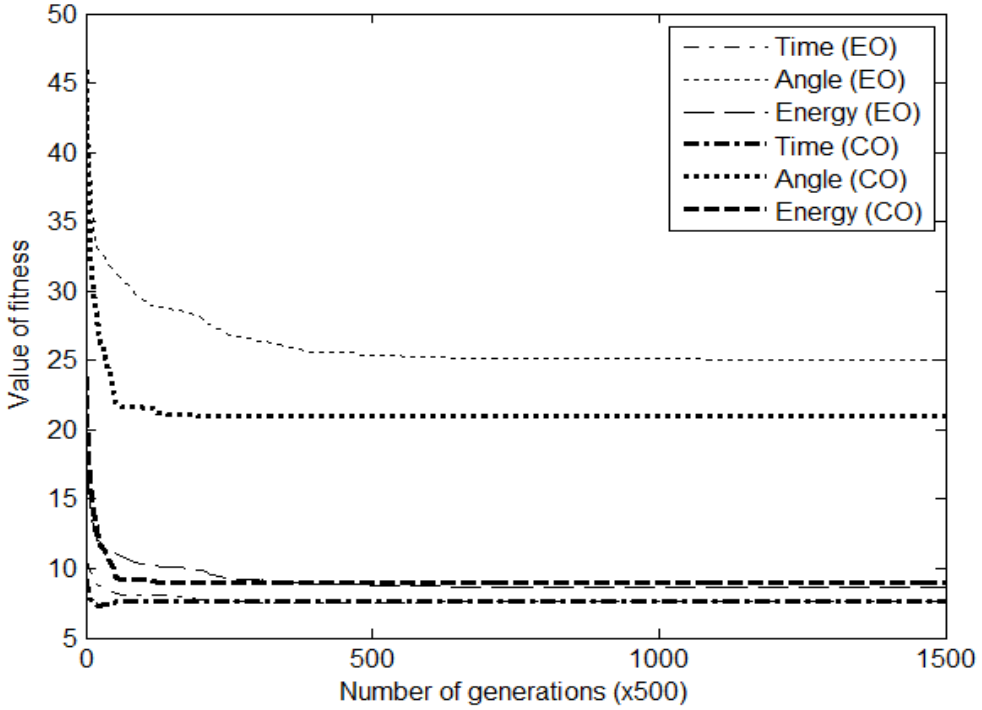
$TP_{r,i}$ je čas pohybu i-teho kĺbu na 1° rotácie

c) Sumárna rotácia všetkých kĺbov:

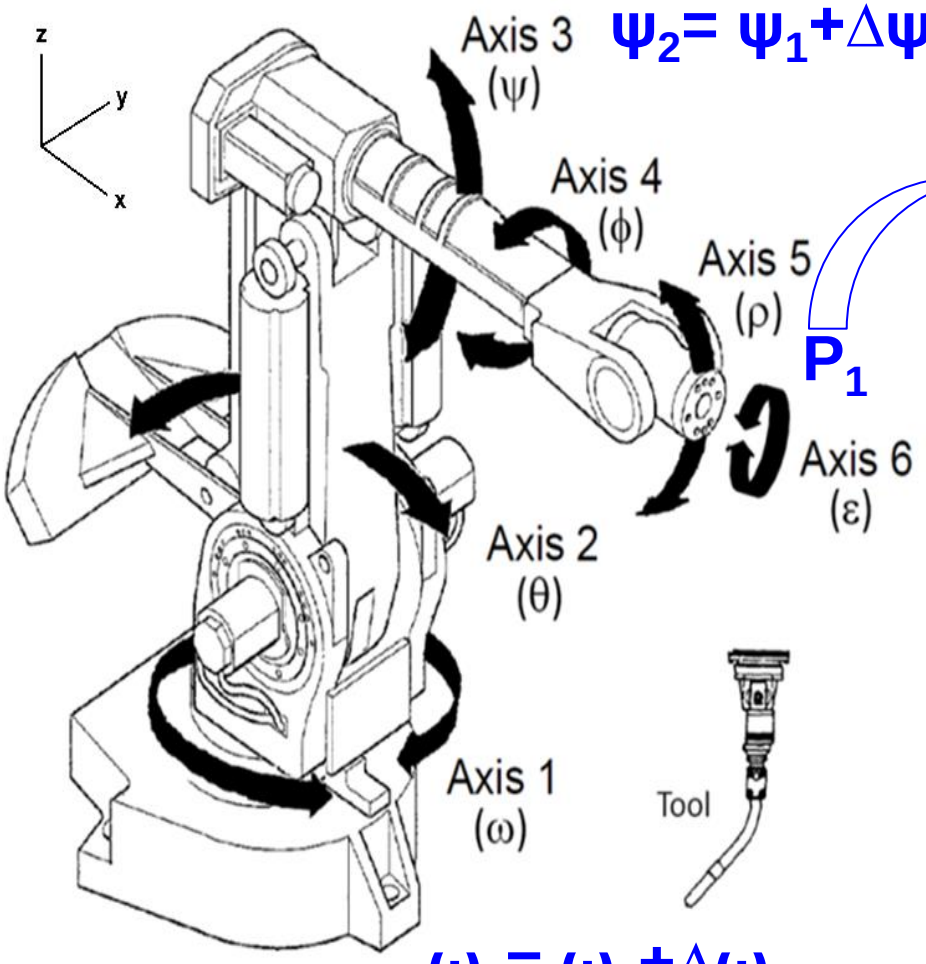
$$A_{tr} = \sum_{j=1}^{N} \sum_{i=1}^{n} \left(\alpha_{b,i,j} - \alpha_{a,i,j} \right)$$

Evolúcia trajektórie robota





Minimalizácia chyby polohovania (Ch_{pol}) a súčasne sumárneho uhla rotácií (S_{rot})



$\psi_2 = \psi_1 + \Delta\psi_{12}$

P_1 P_2

$\omega_2 = \omega_1 + \Delta\omega_{12}$

$$Fit = S_{rot} + Ch_{pol}$$

$$S_{rot} = \sum_j \sum_i (|\Delta\omega_{ji}| + |\Delta\theta_{ji}| + \ldots + |\Delta\rho_{ji}|)$$

j – index prechodu prac. bodu
i – index klbu

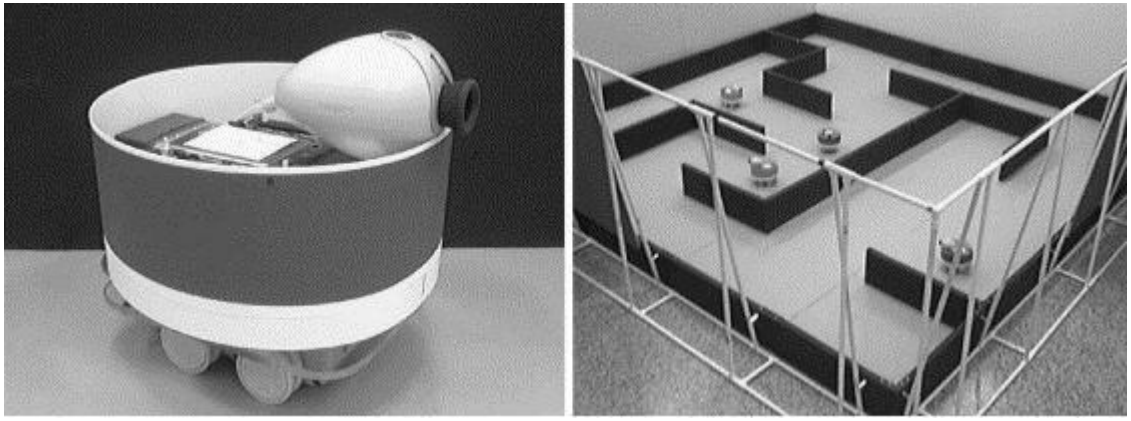
$$Ch_{pol} = \sum_{i=1}^{N} \sqrt{[x_{w,i} - x_{GA,i}]^2 + [y_{w,i} - y_{GA,i}]^2 + [z_{w,i} - z_{GA,i}]^2}$$

Evolúcia morfológie robota

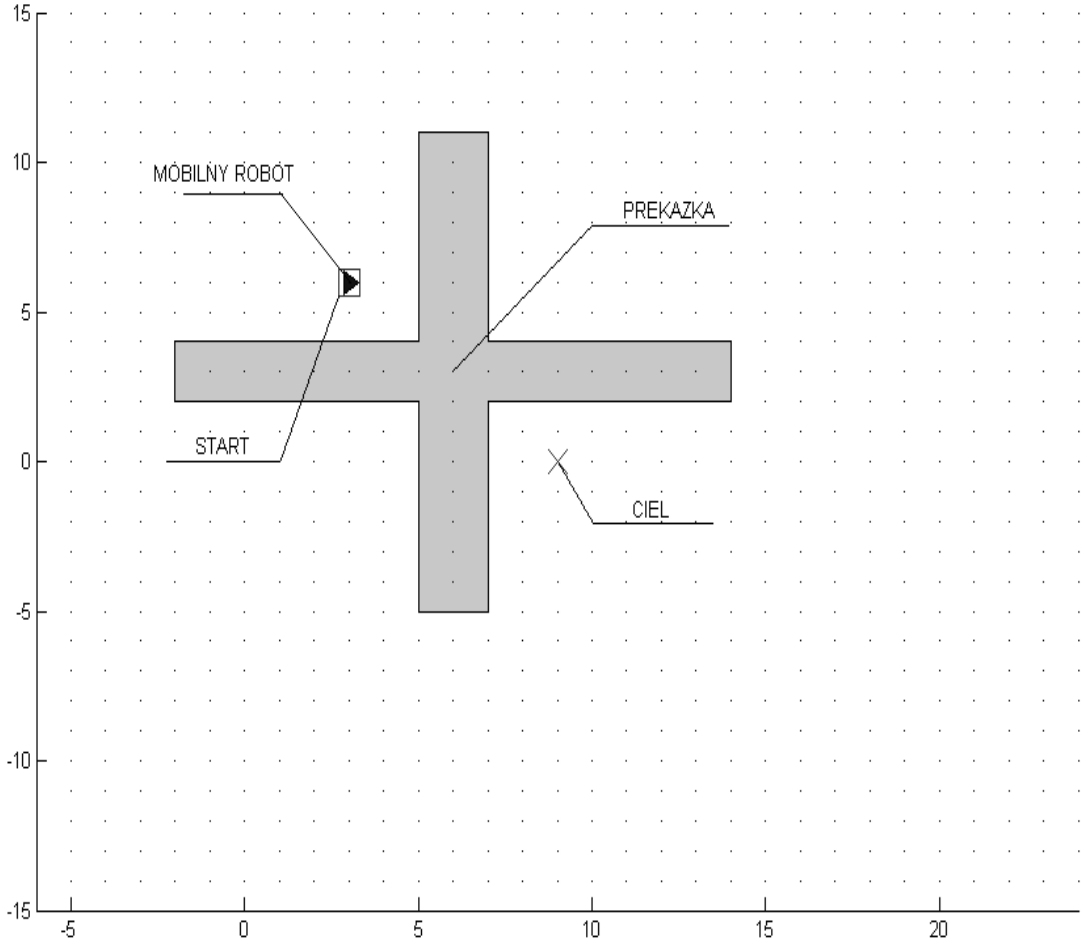
- robot v populácii je tvorený určenými konštrukčnými prvkami ako sú: ramená, kĺby, pohony, prevody a iné prvky o rôznych rozmeroch a parametroch (dĺžka, priemer, uhol...), ktoré sú zakódované v chromozómoch
- EA hľadá ich optimálnu topológiu a parametre
- účelová funkcia ohodnocuje úspešnosť jednotlivých jedincov (schopnosť vykonávať pohyby, efektivitu, výkon, silu, moment sily a pod.)

Trénovanie (učenie) správania sa mobil. robota

- trénovanie riadiaceho programu robota efektívne interagovať so svojim prostredím - riadiaci algoritmus sa pomocou EA naparametrizuje („naučí“) efektívne sa správať
- tu sa často používajú aj umelé neurónové siete
- chromozóm obsahuje zakódovný riadiaci program, resp. jeho parametre
- fitness funkcia obsahuje vyhodnotenie miery úspešnosti správania sa robota v prostredí



Príklad:
Evolúcia riadiaceho programu mobilného robota – genetické programovanie

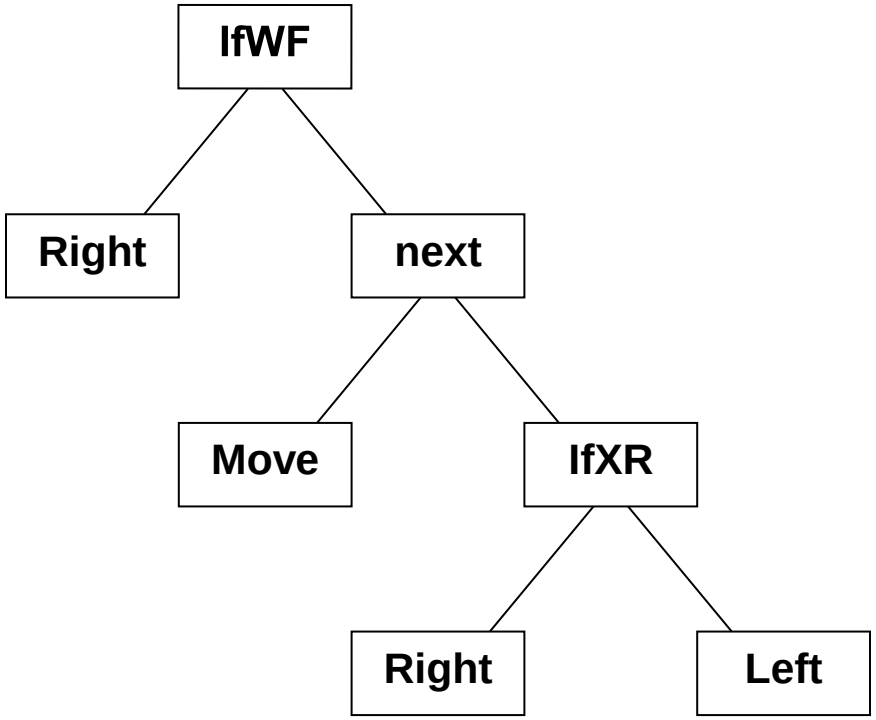


Inštrukčný súbor robota

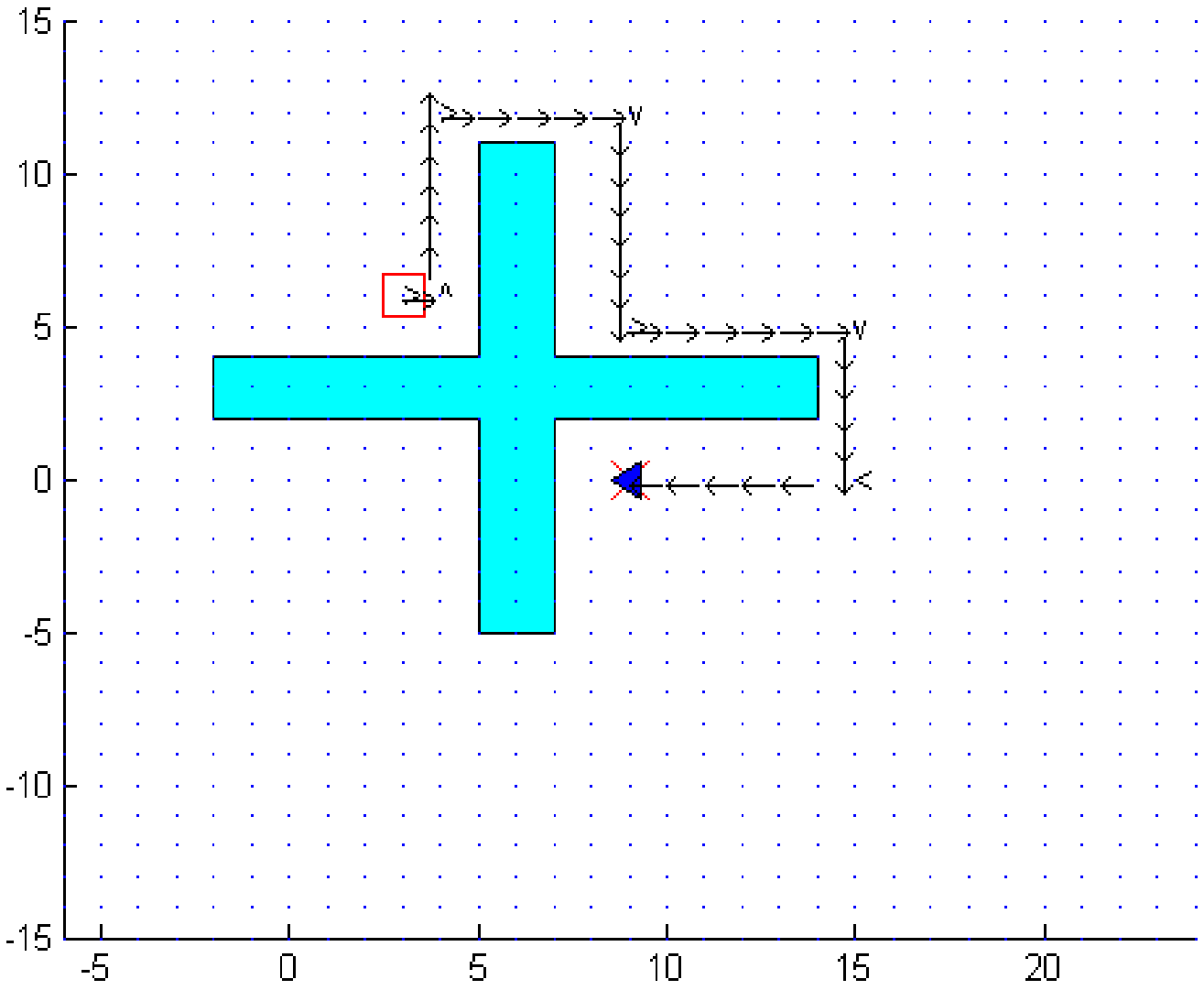
	Symbol inštrukcie (gén)	Popis inštrukcie
Vetviace inštrukcie	***IfWF, IfWR, IfWL***	**Ak je prekážka lokalizovaná snímačmi vpredu, napravo, naľavo od robota.**
	IfXF, IfXR, IfXL	**Ak sa nachádza cieľ smerom vpredu, napravo, naľavo od robota.**
	Prog2	**Sekvenčne vykonaj inštrukcie ľavého a potom pravého podstromu.**
Výkonné inštrukcie	***Move***	**Pohyb o jeden krok v smere natočenia robota.**
	Right	**Otočenie robota doprava o 90°.**
	Left	**Otočenie robota doľava o 90°.**

Príklad riadiaceho programu a jeho stromová reprezentácia

```
IfWF
    Right
else
     Move
     IfXR
          Right
     else
          Left
     end
end
```



Výsledná dráha robota



Nájdený algoritmus:

```
IfWF
    IfXF
        Left
    else
        Right
        IfWR
            Right
        else
            Move
        end
    end
else
    IfWF
        IfXL
            Left
            Left
        else
            IfWF
                Right
            else
                Move
            end
        end
    else
        IfXF
            Move
        else
            IfWR
                Move
            else
                IfXL
                    Move
                else
                    IfXR
                        Right
                        Move
                        Move
                    else
                        Move
                    end
                end
            end
        end
    end
end
```