An Algorithm For Optimal Design of Mechanical Systems Based On Machine Learning

v. 0.1

July 7, 2018

Fazel Arasteh
Dr.Saeed Khodayegan
arastehfazel@gmail.com

Department of Mechanical Engineering
Sharif University
Tehran
Iran

Agenda I

Agenda II

Minimum is optimal

- Minimize f(x)
  subject to:
  gi(x) $\leq$ *bi* $i = 1, ..., m$
  hi(x)=0 i=1,,k
- Types of Optima:
  1. Local Optima
  2. Global Optima
- Types of Optimization:
  1. Local Optimization
  2. Global Optimization
- Can we ever make sure that we have found the Global Optima?
  1. Convex Optimization
  2. Non Convex Optimization

- An Intelligent way to search the space
- Inspired by Natural Selection
- Most fit with the environment individuals, can survive

Genetic Operators:

- Handling the constraints
- Code into Chromosomes
- Cross Over
- Mutation

Handling the constraints

Minimize $f(\mathbf{X})$

$$g_i(\mathbf{X}) \leq 0, \quad i = 1, 2, \ldots, m$$

$$h_j(\mathbf{X}) = 0, \quad j = 1, 2, \ldots, p$$

$\Longrightarrow$

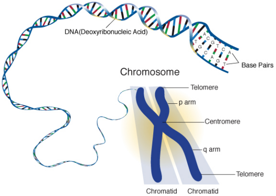
Maximize

$$F(\mathbf{X}) = \frac{1}{1 + \phi(\mathbf{X})}$$

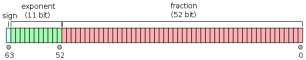
$$\phi(\mathbf{X}) = f(\mathbf{X}) + \sum_{i=1}^{m} r_i \langle g_i(\mathbf{X}) \rangle^2 + \sum_{j=1}^{p} R_j \left(h_j(\mathbf{X})\right)^2$$

Code into Chromosomes

- Chromosome is a string of information



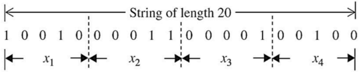
- Coding numbers into strings of 0 and 1:
  1. Bit string
  2. Double vector



$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right)\times 2^{e-1023}$$

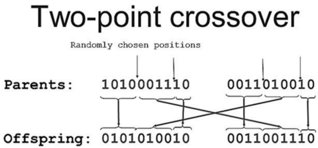
Code into Chromosomes

e= 00000000001_2 = 001_{16} =1: $2^{1-1023} = 2^{-1022}$ (smallest exponent for normal numbers)
e= 01111111111_2 = $3ff_{16}$ =1023: $2^{1023-1023} = 2^{0}$ (zero offset)
e= 10000000101_2 = 405_{16} =1029: $2^{1029-1023} = 2^{6}$
e= 11111111110_2 = $7fe_{16}$ =2046: $2^{2046-1023} = 2^{1023}$ (highest exponent)



Cross Over

- a procedure to make new individuals from previous ones



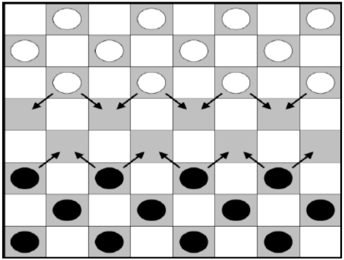
Mutation

- a procedure that let's the algorithm avoid the local optimas

Parent: 01010111**000**1001

Child: 01010111**010**1001

- Arthur Samuel's definition
- Tom Mitchel's definition
- Checkers game
  - E: experience of playing Checkers
  - T: playing Checkers to win
  - P: probability that computer wins the game



- Supervised Learning VS Unsupervised Learning
  - Support Vector Machine is a supervised learning method

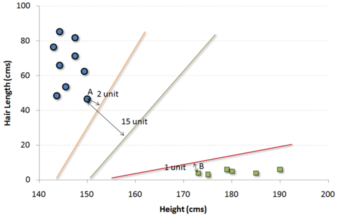
- consider a Training Set of labeled data as follows:

$$TS = \{(x_i, y_i), x_i \in R^n, y_i \in \{0, 1\}, i = 1, ..., N\}$$

- with the following hypothesis:

$$h_\theta(x) = \begin{cases} 1 & \text{if } \Theta^T x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- Example: Prediction of Sexuality from Hair Length and Height



- Cost function is:

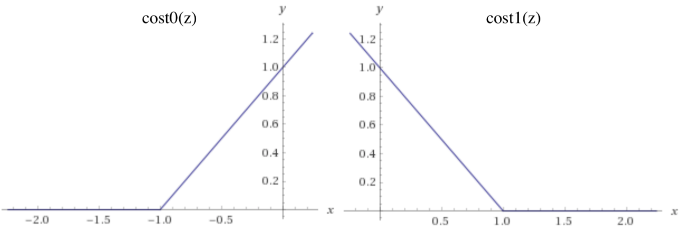
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} y^{(i)}\,\text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)})\,\text{cost}_0(\theta^T x^{(i)})$$

$$z = \theta^T x$$

$$\text{cost}_0(z) = \max(0, k(1+z))$$

$$\text{cost}_1(z) = \max(0, k(1-z))$$

- Penalty functions for wrong predictions:

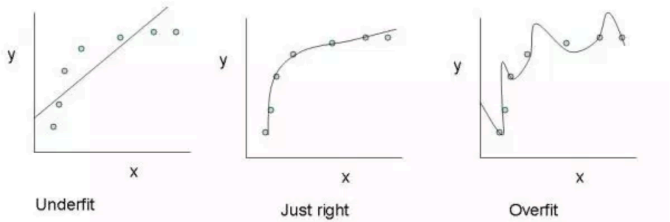


- solve the following problem to find θ :

  Min J(θ)

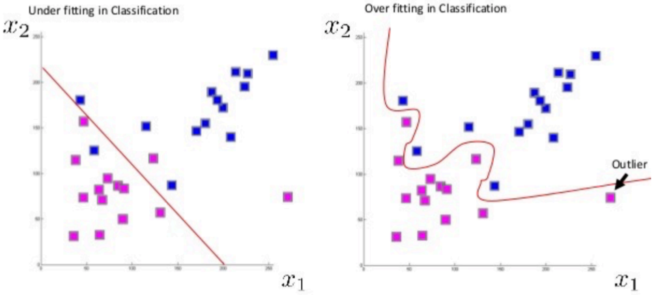
Under Fitting and Over Fitting

- Under Fitting = High Bias
- Over Fitting = High Variance
- 1D Regression:



Under Fitting and Over Fitting

- 2D Classification:

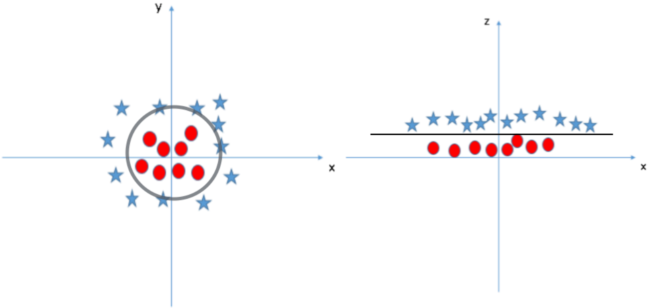


- Cost function with Regularization Parameter (C):

$$J(\theta) = C \sum_{i=1}^{m} y^{(i)} \; \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \; \text{cost}_0(\theta^T x^{(i)}) + \frac{1}{2} \sum_{j=1}^{n} \Theta_j^2$$

Non Linear Boundaries

- Consider the following case.Linearly separable in 2D? in 3D?



- $$Z^2 = X^2 + Y^2$$

Non Linear Boundaries

- No need to have extra knowledge. Use Kernel Trick!
- Just provide the inner product kernel.For instance Gaussian Kernel:

$$f_i = similarity(x, l^{(i)}) = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right)$$

$$l^{(1)} \rightarrow f_1$$

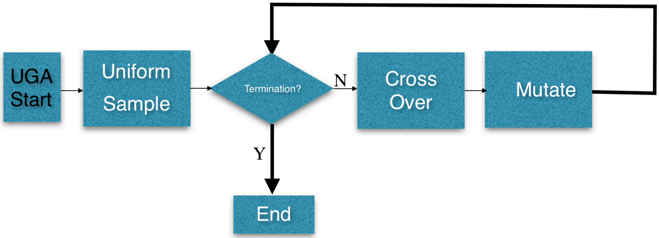
$$l^{(2)} \rightarrow f_2$$

$$l^{(3)} \rightarrow f_3$$

$$h_\Theta(x) = \Theta_1 f_1 + \Theta_2 f_2 + \Theta_3 f_3 + \dots$$

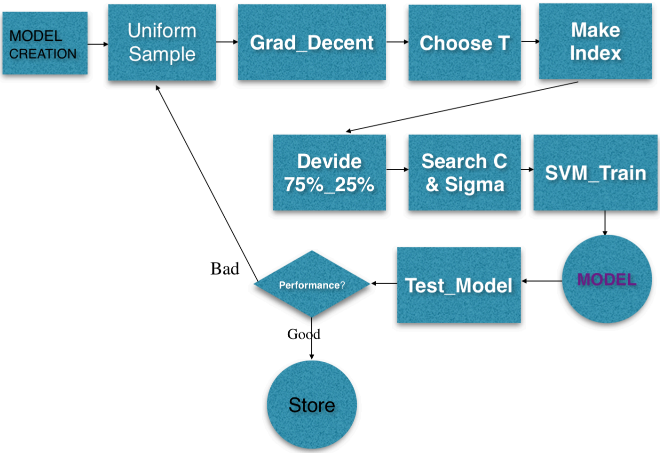
- A good initial guess = Faster convergence = More accurate results
- Exploit the existing knowledge of the problem
- Knowledge is coded inside a Machine Learning Model
- It is possible to classify optimization problems (or maybe problems of other disciplines) in to several classes and train a model for each class according to their specific features. (This is not the case in this article)
- The goal is to show that if such a model were available, how effective it can be.

- UGA = Uniformly initiated Genetic Algorithm :
  initial population is uniformly selected from feasible space



- LGA = Learned Genetic Algorithm :
  Initial population is uniformly selected from feasible space that is accepted by the M.L model
- There are 2 phases:
  1. Learning Phase
  2. Utilization phase

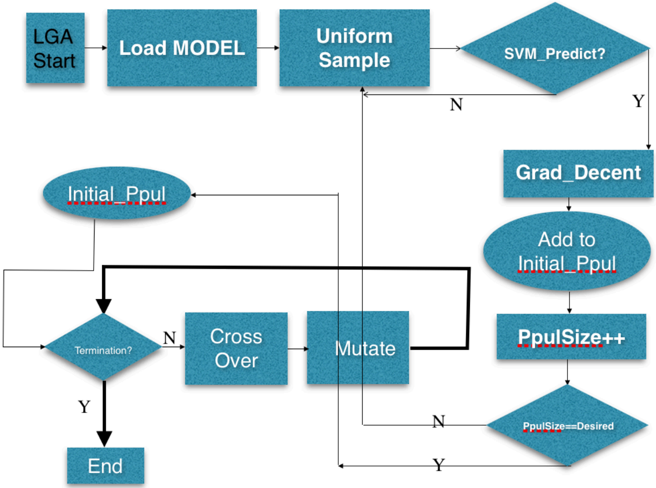
Learning Phase



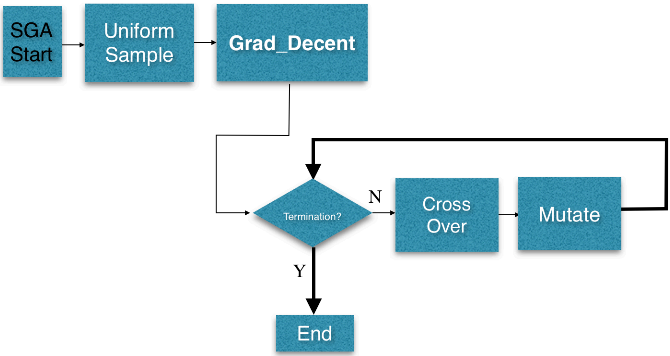
Learning Phase

- Choosing T: using 2sigma rule in Statistics
- Creating the Data Set:
  Local Optima Below T is labeled by "1"
  Local Optima Over T is labeled by "-1"
- Divide Data set into Training Set and Cross Validation Set: 75% , 25 %
- Grid search over C and sigma parameters using TS and CVS
- Train a SVM using whole data and gained C and Sigma
- The model MUST be tested
- Accepted Model will be stored in memory

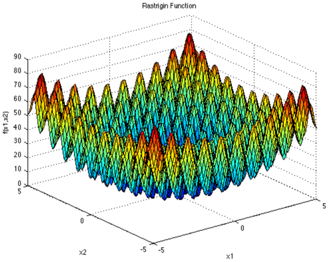
Utilization Phase



- SGA = Sampled Genetic Algorithm :
  initial population is uniformly selected from feasible space and then from each point a local search is initiated. The obtained local optima is sent to GA.
- This algorithm is proposed to be compered with LGA



Rastrigian



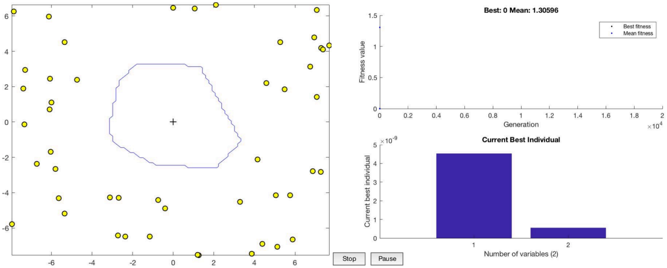
$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i)\right]$$

- Domain:

$$x_i \in [-10, 10], \textit{for } i = 1, , n.$$

Rastrigian

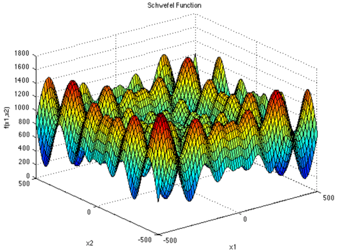
- Model acceptance boundary and GA run:



- Global optima:

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \ldots, 0)$$

Schwefel



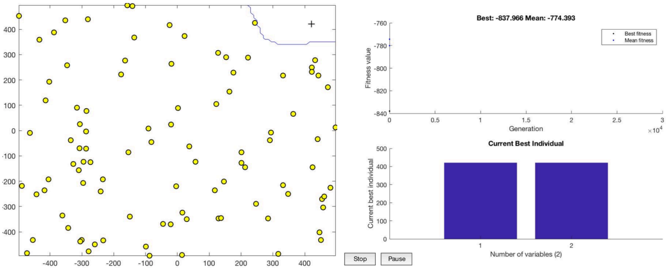
$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}),$$

- Domain:

$$x_i \in [-500, 500],\ \ for\ i = 1, , n.$$

Schwefel

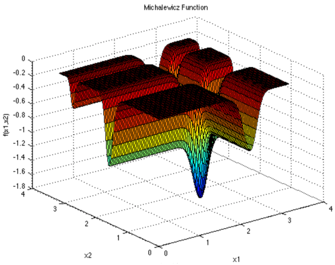
- Model acceptance boundary and GA run:



- Global optima:

$$X_i = 420.9687 \; and \; Value = -418.9829 * Dimension$$

Michalewicz



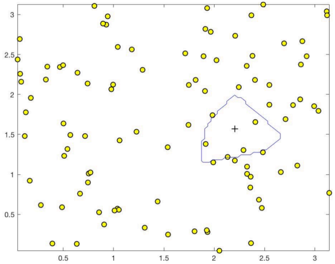
$$f(\mathbf{x}) = -\sum_{i=1}^{d} \sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$$

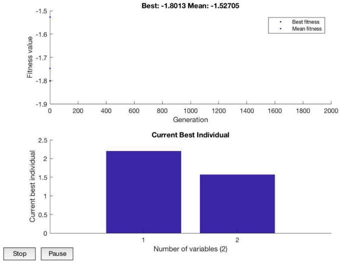
- Domain:

$$x_i \in [0, \pi], \; for \; i = 1, , d.$$

Michalewicz

- Model acceptance boundary and GA run:





- Global optima:

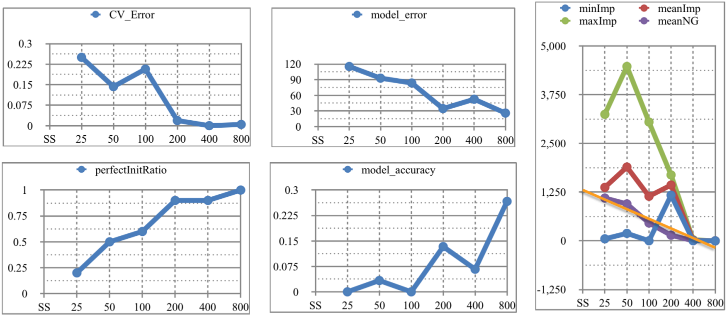
$$\text{at } d = 2: \ f(\mathbf{x}^*) = -1.8013, \text{ at } \mathbf{x}^* = (2.20, 1.57)$$
$$\text{at } d = 5: \ f(\mathbf{x}^*) = -4.687658$$
$$\text{at } d = 10: \ f(\mathbf{x}^*) = -9.66015$$

- CV Error
- Perfect Initiation Ratio
- Model Error
- Model Accuracy
- meanNG
- meanImp,minImp,maxImp

Rastrigian 10D

- Dimension: 10
  Number of GA: 20
  Fitness Limit:2*10⁻9
  Max Stall Generations: 10,000
  Population Size: 30

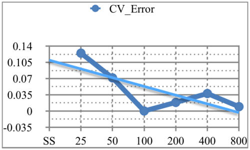
Rastrigian 10D

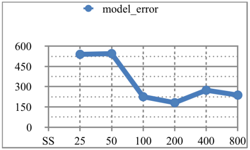


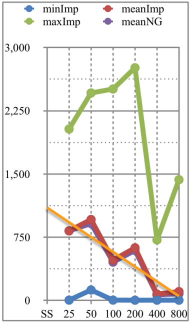
Schwefel 6D

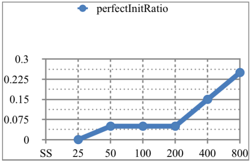
- Diminsion: 6
  Number of GA: 20
  Fitness Limit:-418.9*Dimension
  Max Stall Generations: 20,000
  Population Size: 30

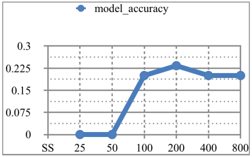
Schwefel 6D


CV_Error
0.14
0.105
0.07
0.035
0
-0.035
SS
25
50
100
200
400
800

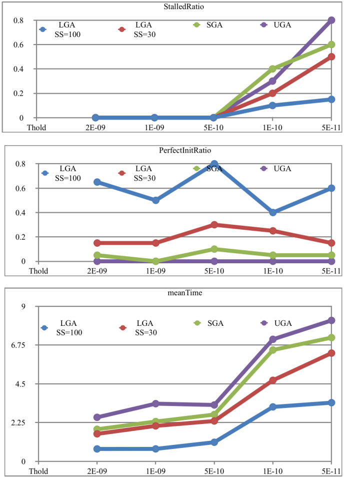

model_error
600
450
300
150
0
SS
25
50
100
200
400
800


minImp
meanImp
maxImp
meanNG
3,000
2,250
1,500
750
0
SS
25
50
100
200
400
800

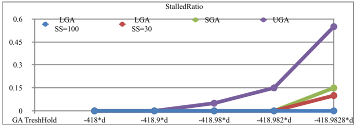

perfectInitRatio
0.3
0.225
0.15
0.075
0
SS
25
50
100
200
400
800

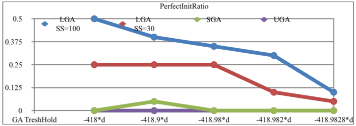

model_accuracy
0.3
0.225
0.15
0.075
0
SS
25
50
100
200
400
800

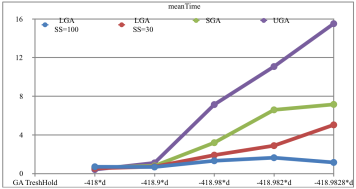
Rastrigian 10D



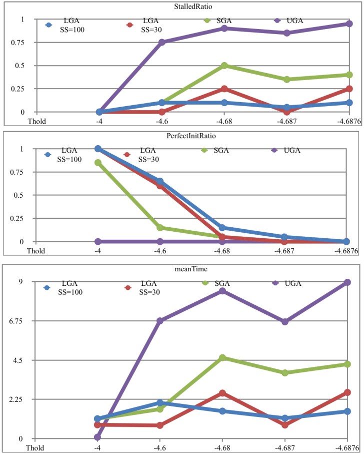
Schwefel 6D


StalledRatio
LGA SS=100
LGA SS=30
SGA
UGA
0.6
0.45
0.3
0.15
0
GA TreshHold
-418*d
-418.9*d
-418.98*d
-418.982*d
-418.9828*d

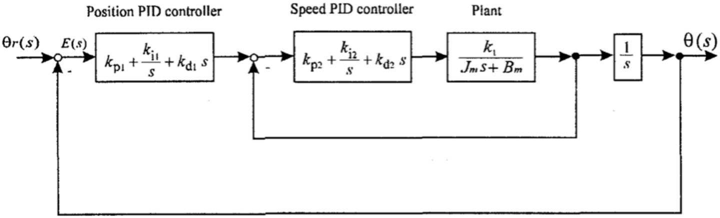

PerfectInitRatio
LGA SS=100
LGA SS=30
SGA
UGA
0.5
0.375
0.25
0.125
0
GA TreshHold
-418*d
-418.9*d
-418.98*d
-418.982*d
-418.9828*d


meanTime
LGA SS=100
LGA SS=30
SGA
UGA
16
12
8
4
0
GA TreshHold
-418*d
-418.9*d
-418.98*d
-418.982*d
-418.9828*d

Michalewicz 5D



- consider the following system
- two PID controller
  1. position
  2. speed



- transfer function of the plant

$$G(s) = \frac{k_t}{J_m s + B_m}$$

- transfer function of PID controllers

$$C(s) = k_p + \frac{k_i}{s} + k_d s$$

- transfer function of system for unit step response

$$E(s) = \frac{d_0 s^3 + d_1 s^2 + d_2 s + d_3}{a_0 s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4}$$

Controller Design for an AC Servo Motor

Problem Definition

- Nominator coefficients:

$$d_0 = J_m + k_{d2}.k_t$$
$$d_1 = B_m + k_{p2}.k_t$$
$$d_2 = k_t.k_{i2}$$
$$d_3 = 0$$

- Denominator coefficients:

$$a_0 = J_m + k_{d2}.k_t + k_{d1}.k_{d2}.k_t$$
$$a_1 = B_m + k_{d2}.k_{p_1}.k_t + k_{p2}.k_t + k_{d1}.k_{p2}.k_t$$
$$a_2 = k_{d2}.k_{i1}.k_t + k_{i2}.k_t + k_{d1}.k_{i2}.k_t + k_{p1}.k_{p2}.k_t$$
$$a_3 = k_{i2}.k_{p1}.k_t + k_{i1}.k_{p2}.k_t$$
$$a_4 = k_{i1}.k_{i2}.k_t$$

- ITSE criteria:

$$W = \int_0^{\infty} t e^2(t) dt$$

- Laplace Transform:

$$= -\frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} \frac{d}{ds}\left(E(s)\right) E(-s)\, ds$$

$$= -\frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} \frac{d}{ds}\left(\frac{\sum_{h=0}^{n-1} d_h s^{n-1-h}}{\sum_{h=0}^{n} a_h s^{n-h}}\right) \frac{\left(\sum_{h=0}^{n-1} d_h (-s)^{n-1-h}\right)}{\left(\sum_{h=0}^{n} a_h (-s)^{n-h}\right)} ds = W_n$$

- Residual Theorem:

$$W_4(k_{p1}, k_{p2}, k_{i1}, k_{i2}, k_{d1}, k_{d2}) = A - B + C$$

$$A = \frac{d_3^2}{4a_4^2}$$

$$B = \frac{Z_1}{N_1}$$

$$Z_1 = 2a_4\, d_0^2 + 2a_0\left(d_2^2 - 2d_1\, d_3\right) + a_2\left(d_1^2 - 2\, d_0\, d_2\right) + \\ + a_1\left(d_1\, d_2 - 3d_0\, d_3\right) + a_3\, d_0\, d_1 + \frac{\left(a_1\, a_2 - a_0\, a_3\right)}{a_4} d_2\, d_3 + \\ + \frac{d_3^2}{a_4}\left(a_0\, a_2 + a_1^2\right)$$

$$N_1 = 2\left(a_1\, a_2\, a_3 - a_0\, a_3^2 - a_1^2 a_4\right)$$

$$C = \frac{Z_2}{N_2}$$

$$Z_2 = \left(a_3^2\, d_0^2 + a_1^2\left(d_2^2 - 2d_1\, d_3\right) + a_1\, a_3\left(d_1^2 - 2d_0\, d_2\right) + \frac{a_1}{a_4}\left(a_1\, a_2 - a_0\, a_3\right)d_3^2\right)\left(a_1\, a_3 - 4a_0\, a_4 + a_2^2\right)$$

$$N_2 = 2\left(a_1\, a_2\, a_3 - a_0\, a_3^2 - a_1^2\, a_4\right)^2$$

- Optimization problem:

$$\min_{K\ \text{stabilizing}} \; W_n(K)$$

- constrained to:

$$\forall i : 0 \le k_i \le 10$$

- and stability check: Routh array or place of Zeros and Poles

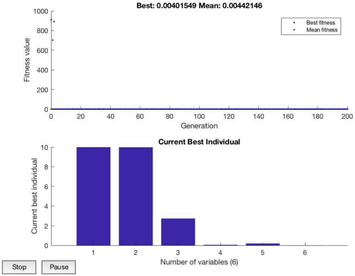
- Data of reference [2] :

$$k_t = 4.42\ 10^{-2}\ \mathrm{Nm/A}$$
$$J_m = 2.5\ 10^{-4}\ \mathrm{kg\ m^2}$$
$$B_m = 8.59\ 10^{-5}\ \mathrm{Nm/(rad/s)}$$

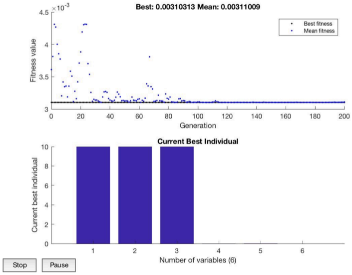
- Comparison of LGA and UGA and Krohling Results:

	Kp1	Ki1	kd1	Kp2	Ki2	kd2	W
R.A.Krohlong Results	10	9.98	1.675	0.071	0.362	0	0.0040
UGA	9.9959	9.9883	2.7340	0.0509	0.1910	0.0029	0.0040
LGA_100	10.0000	10.0000	10.0000	0.0104	0.0171	0.0000	0.0030

- UGA Design:


Best: 0.00401549 Mean: 0.00442146
Best fitness
Mean fitness
Fitness value
Generation
Current Best Individual
Current best individual
Number of variables (6)
Stop
Pause

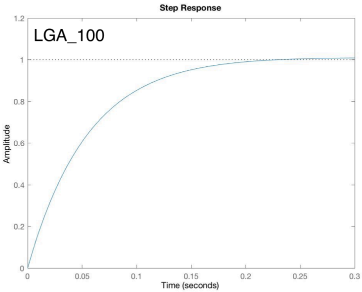
- LGA Design:

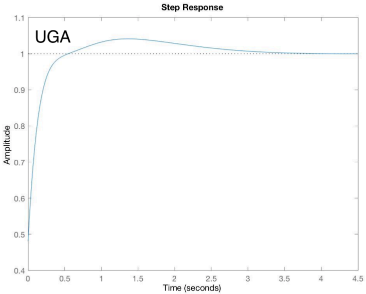

Best: 0.00310313 Mean: 0.00311009
Best fitness
Mean fitness
Fitness value
Generation
Current Best Individual
Current best individual
Number of variables (6)
Stop
Pause

Controller Design for an AC Servo Motor

Implementation and results

- Step Response:

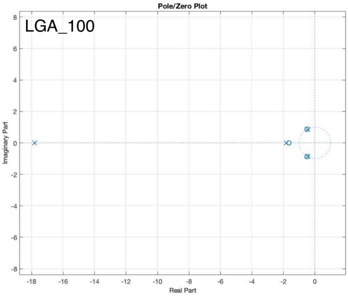


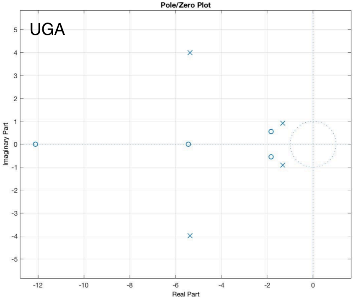


The AAU Sidebar Beamer Theme

Fazel Arasteh

- Poles and Zeroes:





Feedback
Contact Information

In case you have comments or suggestions, please do not hesitate to contact me. You can find my contact details below.

Fazel Arasteh
Dr.Saeed Khodayegan
`arastehfazel@gmail.com`

Thank you for your attention