Intelligent Systems

A strongly intended application to control systems

Nelson Carlos de Sousa Campos

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

June 2016

Introduction

- What is an **Intelligent System**?
- **Answer 1**: An **intelligent system** is a machine with an embedded, Internet-connected computer that has the capacity to gather and analyze data and communicate with other **systems**.
- **Answer 2**: An **intelligent system** must be **highly adaptable** to significant unanticipated changes, and so **learning** is essential, exhibit high degree of **autonomy** in dealing with changes, be able to deal with significant **complexity**, and exhibit certain sparse types of functional architectures such as **hierarchies**.

- What is **Adaptive Control**?
- **Answer**: **Adaptive control** is the **control** method used by a **controller** which must adapt to a **controlled** system with parameters which vary, or are initially uncertain.

Adaptive Control - Tasks 1, 2 and 3

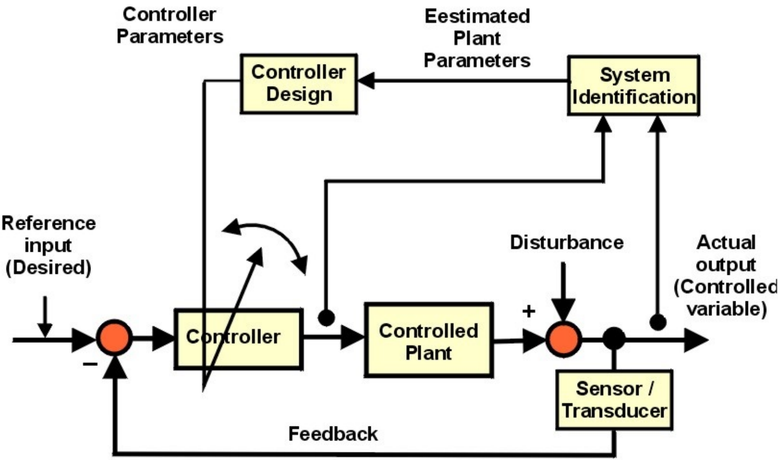

Figure 1 An adaptive control system

Selected Problem from task 1: Processes with parameters variations

Consider a process $G(s) = \frac{K}{a_0 s^2 + a_1 s + a_2}$, where $K = K_0 + \Delta K$, $a_i = a_{i0} + \Delta a_i$ and let the desired close loop response be given by $Y_m = \frac{\omega_n^2}{s^2 + 2\zeta\omega s + \omega_n^2}$
In order to simulate the process, the simulink model shown in Figure 2.

Selected Problem from task 1: Processes with parameters variations

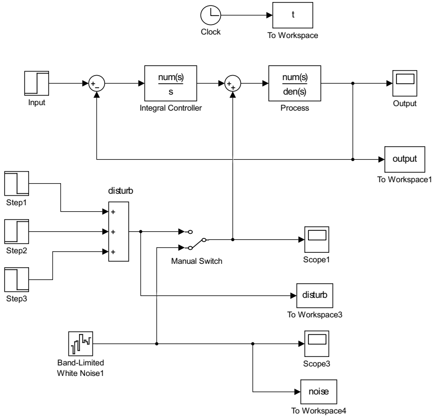

Figure 2 Processes with parameters variations

Selected Problem from task 1: Processes with parameters variations

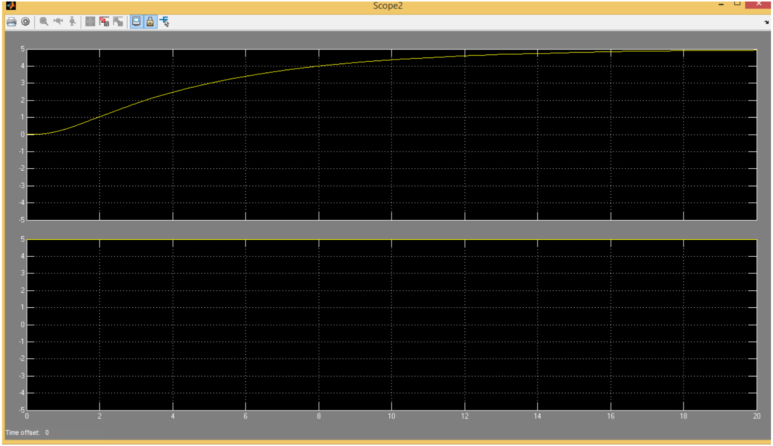

Figure 3 Processes with parameters variations: output closed loop response

Selected Problem from task 2: Estimating Parameters in Dynamical Systems

- Mathematical model on parameters: $y = \phi^T \theta$
- **regressors**: $\phi^T(i) = \begin{pmatrix} \phi_1(i) & \phi_2(i) & \dots & \phi_N(i) \end{pmatrix}$
- **parameters**: $\theta^0 = \begin{pmatrix} \theta_1^0 & \theta_2^0 & \dots & \theta_N^0 \end{pmatrix}^T$
- **Cost function**:
  $V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t}(y(i) - \phi^T(i)\theta)^2 = \frac{1}{2}\sum_{i=1}^{t}(y(i) - y_{est}(i))^2$
- where $\epsilon(i) = y(i) - y_{est}(i)$

Selected Problem from task 2: Estimating Parameters in Dynamical Systems

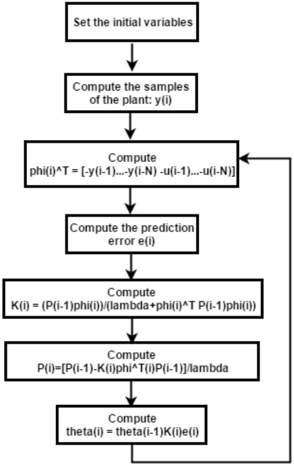

Figure 4 Recursive Least squares algorithm

Selected Problem from task 2: Estimating Parameters in Dynamical Systems

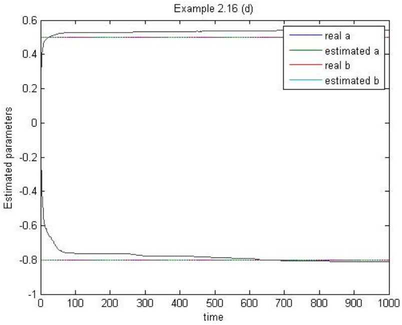

Figure 5 Recursive Least squares algorithm results

Selected Problem from task 3: Deterministic Self-Tuning Regulators

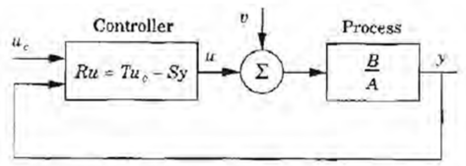

Figure 6 Adaptive Control block diagram

$$y(t) = \frac{BT}{AR+BS} u_c(t) + \frac{BR}{AR+BS} v(t)$$

$$u(t) = \frac{AT}{AR+BS} u_c(t) + \frac{BS}{AR+BS} v(t)$$

Selected Problem from task 3: Algorithm for the pole placement

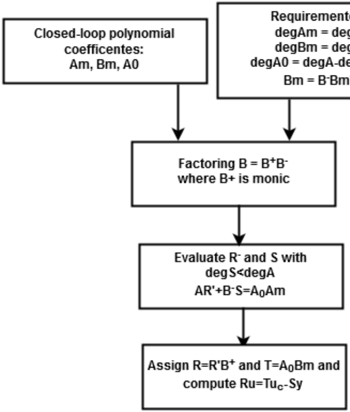

Figure 7 Algorithm for the pole placement

Selected Problem from task 3: Effect of Load Disturbances (results)

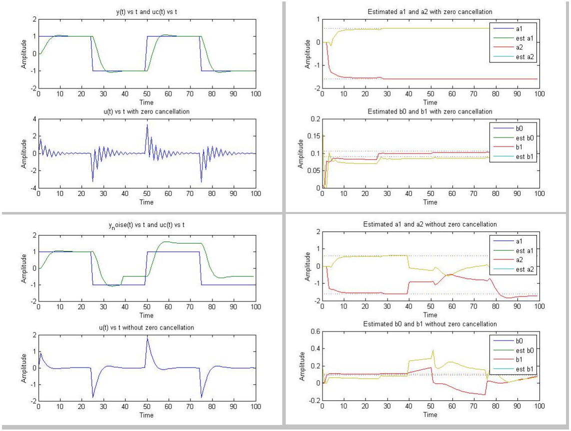

Figure 8 Effect of Load Disturbances (results)

Fuzzy Logic - Tasks 4, 5 and 10

- Machine Learning technique that mimics human reasoning
- Multi value logic that allows intermediate values compared to the binary logic in which only two values are allowed
- Inference systems are defined by rules with different degrees of priority
- Two types of defuzzyfication:
  - **mandani**: Fuzzy outputs are defined as sets
  - **sugeno**: Fuzzy outputs are defined as points

Fuzzy Logic - Design steps

- **step 1**: define inputs and outputs as member functions of the Fuzzy space
- **step 2**: define rules from the fuzzy inputs and outputs
- **step 3**: assign weights for each defined rule
- **step 4**: map these rules to generate human understandable results

Expert Systems for PID Tuning (Tasks 4 and 5)

1) Increasing K_c decreases period and vice versa
2) Increasing K_c increases overshoot and vice versa
3) Increasing K_c decreases rise time and vice versa
4) Increasing K_c increases damping and vice versa
5) Decreasing T_i increases overshoot ratio and vice versa
6) Decreasing T_i increases damping and vice versa
7) Decreasing T_i decreases stability and vice versa
8) Increasing T_i decreases overshoot and vice versa
9) Increasing T_d increases stabilty and vice versa
10) Increasing T_d decreases rise time and vice versa.

Figure 9 Rules for PID Tuning

Expert Systems for PID Tuning (Tasks 4 and 5)

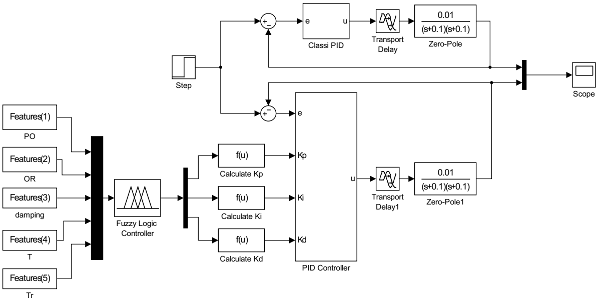

Figure 10 Simulink model for Fuzzy PID Tuning (Tasks 4 and 5)

Expert Systems for PID Tuning (Tasks 4 and 5)

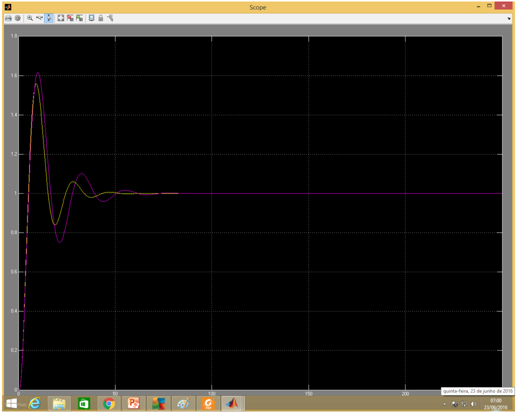

Figure 11 Fuzzy PID Tuning (Tasks 4 and 5): fuzzy output (purple) and traditional PID output (yellow)

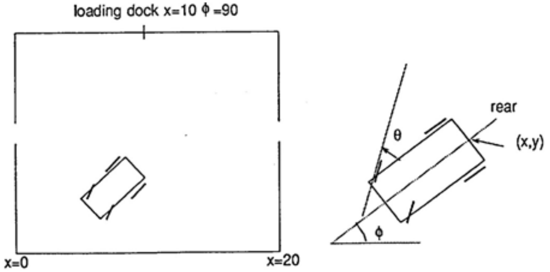

Figure 12 Truck backing up loading zone

$$x(t+1) = x(t) + cos(\phi(t) + \theta(t)) + sin(\theta(t))sin(\phi(t))$$
$$y(t+1) = y(t) + cos(\phi(t) + \theta(t)) - sin(\theta(t))cos(\phi(t))$$
$$\phi(t+1) = \phi(t) - arcsin(\frac{2sin(\theta(t))}{b})$$

Fuzzy Logic for Truck Backing up

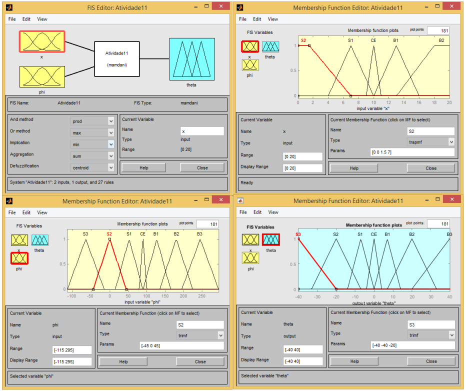

Figure 13 Truck backing up - Fuzzy member functions for inputs and outputs

Fuzzy Logic for Truck Backing up

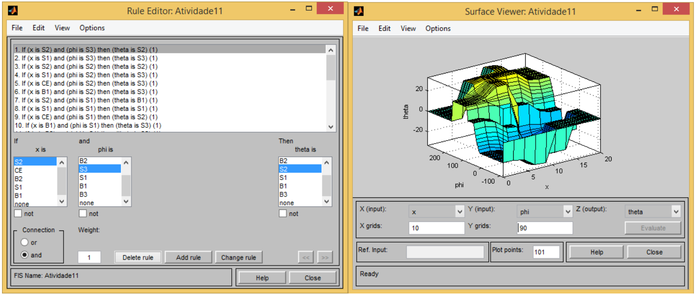

Figure 14 Truck backing up - Fuzzy definitions and maping rules

Fuzzy Logic for Truck Backing up

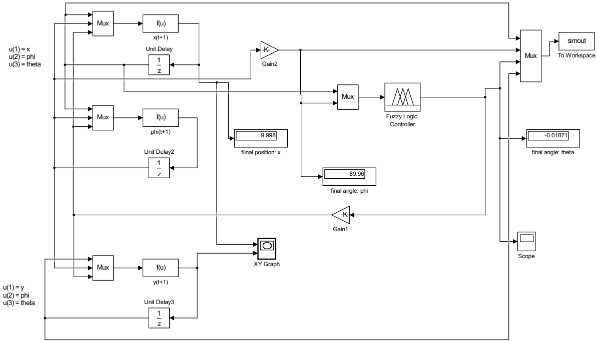

Figure 15 Truck backing up - Simulink model for the dynamic system

Fuzzy Logic for Truck Backing up

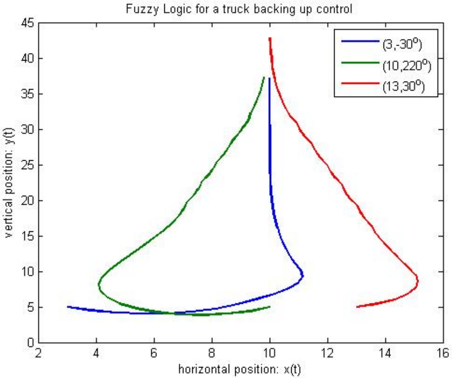

Figure 16 Truck backing up - Simulation results

- An **Expert System** is a computer program that contains subject-specific human knowledge.
- Compomsed by:
  - Knowledge base (facts)
  - Production rules (**"if ... , then ..."**)
  - Inference engine (controls how the rules are applied to the facts)
  - User interface

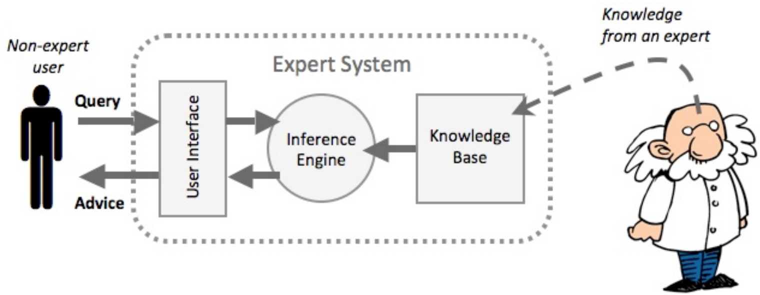

Figure 17 Expert Systems block diagram

CLIPS Expert System - Task 6

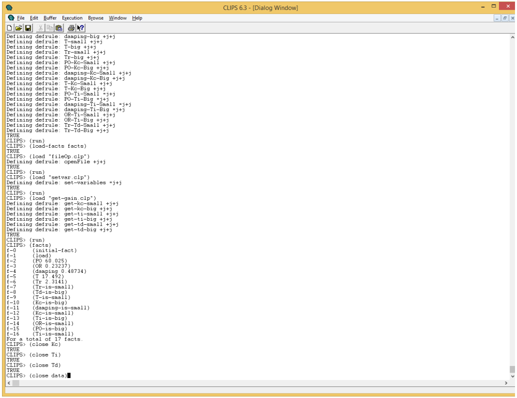

Figure 18 CLIPS (C Language Integrated Production System)

Task 7 - Stochastic Filtering: Kalman Filter

- **Stochastic Filtering** is a good tool when there is the need to track time-varying processes
- **Kalman Filter** is a **recursive** data processing algorithm that generates **optimal** estimate of desired parameters given samples of measurements
- **Optimal** because for linear sistems and for white noise errors, Kalman is the best estimate method based on previous measurements
- **Recursive** because it does not need to store data, once it is reprocessed in each iteration.

Suppose the signal $x(t) = Asin(\omega t)$

Let's define $\phi = \omega t$, $\dot{\phi} = \omega$, $\dot{\omega} = 0$ and $\dot{A} = 0$

The state-space of the system is: $\begin{bmatrix} \dot{\phi} \\ \dot{\omega} \\ \dot{A} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \omega \\ A \end{bmatrix} + \begin{bmatrix} 0 \\ u_{s_1} \\ u_{s_2} \end{bmatrix}$

Let's also define:

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_{s_1} & 0 \\ 0 & 0 & \Phi_{s_2} \end{bmatrix},$$

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } F^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Task 7 - Stochastic Filtering: Kalman Filter

It can be seen that: $\Phi(t) = e^{Ft} = I + Ft$

$$\Phi(t) = \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \Phi(kT_s) = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the measurement vector is:

$$x^* =< \nabla x, \begin{bmatrix} \Delta\phi & \Delta\omega & \Delta A \end{bmatrix} > +v$$

and v is Gaussian noise with zero mean and variance σ_v.

Thus, the linearized System is: $C = \begin{bmatrix} Acos(\phi) & 0 & sin(\phi) \end{bmatrix}$

and the update system state function can be applied by the extended Kalman filter:

$$\begin{bmatrix} \bar{\phi}_k \\ \bar{\omega}_k \\ \bar{A}_k \end{bmatrix} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_{k-1} \\ \hat{\omega}_{k-1} \\ \hat{A}_{k-1} \end{bmatrix}$$

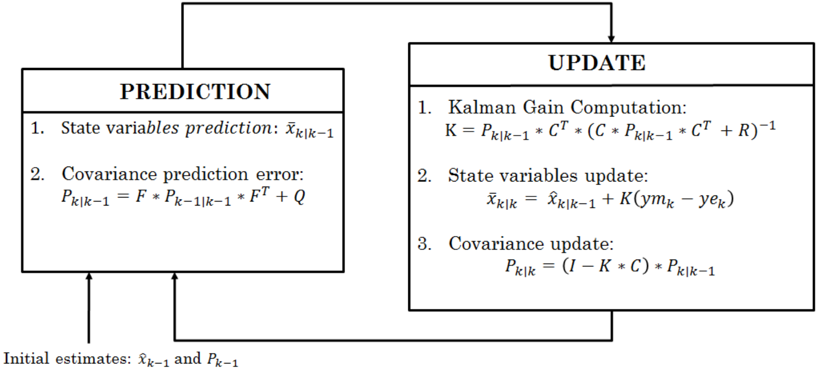

Figure 19 Kalman Filter Algorithm

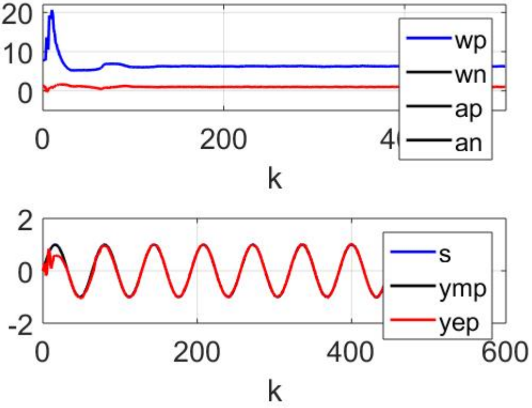

Figure 20 Kalman Filter for sinuisoid tracking (results of the selected Kalman Filter)

Task 8 - Multi-Objective Genetic Algorithm

- Computerized search and optimization algorithms based on Darwin's Principle of Natural Selection
- **Genetic Algorithms (GA)** converts design space into genetic space
- Traditional optimization techniques are deterministic in nature, but GA uses randomized operators
- **Fitness Function**: is a particular type of objective function that is used to summarise, as a single figure of merit, how close a given design solution is to achieving the set aims.
- **Definition and implementation of genetic representation**
- **Definition and implementation of genetic operators**

Task 8 - Multi-Objective Genetic Algorithm

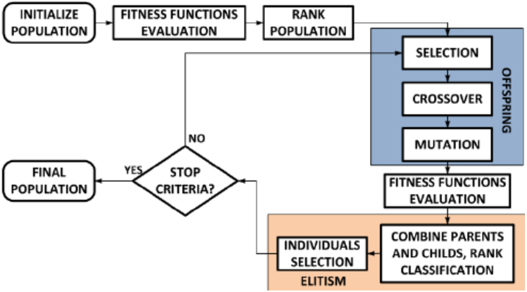

Figure 21 Multi-objective Genetic Algorithm diagram

Task 8 - Multi-Objective GA for Induction Machine parameters estimation

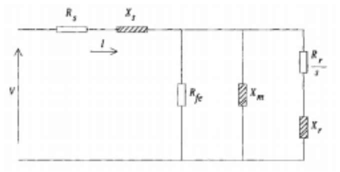

Figure 22 Induction Machine equivalent circuit

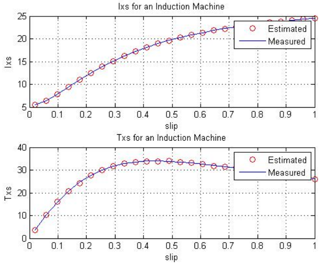

Figure 23 Multi-Objective GA for Induction Machine parameters estimation

A **Hopfield network** is a form of **recurrent artificial neural network** popularized by John Hopfield in 1982, but described earlier by Little in 1974.

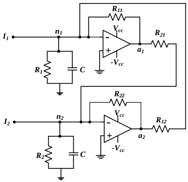

Figure 24 Circuit of Hopfield Network for 2 neurons

$$C\frac{dn_i(t)}{dt} = \sum_{i=1}^{S} T_{i,j}a_j - \frac{n_i(t)}{R_i} + I_i$$
$$\text{where } |T_{i,j}| = \frac{1}{R_{i,j}}$$

Neural Networks for Sinusoidal Frequency Estimation - Task 9

Statement of the problem: estimate A_k, f_k and Φ_k

$$x_n = \sum_{k=0}^{m} A_k cos(2\pi f_k n + \Phi_k),, \text{ for } n = 0, 1, 2, ...N - 1$$

Applying Least-squares optimization in the circuit model:

$$\frac{dE}{dn} = \frac{d}{dn}\left[-\frac{1}{2}\sum_{n=0}^{N-1}[\,x_n - \sum_{k=1}^{m} A_k cos(2\pi f_k n + \Phi_k)]^2\right]$$

$$\frac{dE}{dn} = -2\pi \sum_{n=0}^{N-1}[f_1 A_1 sin(2\pi f_1 + \Phi_1) + f_2 A_2 sin(2\pi f_2 + \Phi_2)]$$

$$\frac{dE}{df_k} = -\sum_{n=0}^{N-1} 2\pi n A_k sin(2\pi n f_k + \Phi_k) d_n, \text{ for } k = 0, 1, 2, ...m$$

$$\frac{dE}{dA_k} = \sum_{n=0}^{N-1} cos(2\pi n f_k + \Phi_k) d_n, \text{ for } k = 0, 1, 2, ...m$$

$$\frac{dE}{d\Phi_k} = -\sum_{n=0}^{N-1} A_k sin(2\pi n f_k + \Phi_k) d_n, \text{ for } k = 0, 1, 2, ...m$$

$$dn = x_n - \sum_{i=1}^{m} A_i cos(2\pi f_i n + \Phi_i)$$

Neural Networks for Sinusoidal Frequency Estimation - Task 9

tab. 1 Parameters estimation of x_n

	Real Value	Initial Value	Estimated Value
A_1	1.0	1.25	1.1299
A_2	0.8	1.0	0.9215
f_1	0.3	0.375	0.3762
f_2	0.32	0.4	0.3988
Φ_1	0.7854	1.1781	0.0804
Φ_2	0.3927	0.589	-0.1170

References - Tasks 1, 2 and 3

[1] Åström, Karl J., and Björn Wittenmark. Adaptive control. Courier Corporation, 2013. APA
[2] http://www.omative.com/ACM.html
[3] www.dictionary.com
[4] Landau, Ioan Doré, et al. Adaptive control: algorithms, analysis and applications. Springer Science Business Media, 2011.

References - Tasks 4, 5 and 6

[1] Ziegler, John G., and Nathaniel B. Nichols. "Optimum settings for automatic controllers." trans. ASME 64.11 (1942).
[2] Litt, Jonathan. "An expert system to perform on-line controller tuning." Control Systems, IEEE 11.3 (1991): 18-23.
[3] Zhao, Zhen-Yu, Masayoshi Tomizuka, and Satoru Isaka. "Fuzzy gain scheduling of PID controllers." Control Applications, 1992., First IEEE Conference on. IEEE, 1992.
[4] Zadeh, Lotfi A. "Is there a need for fuzzy logic?." Information sciences 178.13 (2008): 2751-2779.
[5] Riley, Gary. "Clips: An expert system building tool." (1991).

References - Task 7

[1] Bikash Lamsal and Naofumi Matsumoto. Effects of the unscented kalman filter process for high performance face detector. International Journal of Information and Electronics Engineering, 5(6):454, 2015.

[2] Macharla Vinaykumar and Ravi Kumar Jatoth. Performance evaluation of alpha-beta and kalman filter for object tracking. In Advanced Communica- tion Control and Computing Technologies (ICACCCT), 2014 International Conference on, pages 1369-1373. IEEE, 2014.

[3] Wikipedia. Kalman filter - wikipedia. https://en.wikipedia.org/wiki/ Kalman_filter, 2016.
[4] Paul Zarchan. Progress In Astronautics and Aeronautics: Fundamentals of Kalman Filtering: A Practical Approach, volume 208. Aiaa, 2005.

References - Task 8

[1] Multi-objective optimization.
https://en.wikipedia.org/wiki/Multiobjective optimization.
[2] Arshad Afzal and Kwang-Yong Kim. Three-objective optimization of a staggered herringbone micromixer. Sensors and Actuators B: Chemical, 192:350- 360, 2014.
[3] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. Evolutionary Computation, IEEE Transactions on, 6(2):182-197, 2002. [4] Antonio Marcus Nogueira Lima, Cursino Brandao Jacobina, and Eurico Bezerra de Souza Filho. Nonlinear parameter estimation of steady-state induction machine models. Industrial Electronics, IEEE Transactions on, 44(3):390-397, 1997.
[5] Joao B. M. SANTOS and ANTONIO M. N. Lima. Multi-objective identification of induction machine models. Simposio Brasileiro de Automacao Inteligente, 2015.

References - Task 9

[1] Lifang Han and Saroj K Biswas. Neural networks for sinusoidal frequency estimation. Journal of The Franklin Institute, 334(1):1-18, 1997.

[2] John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of the national academy of sciences, 81(10):3088-3092, 1984.

[3] Steven M Kay. Modern spectral estimation. Pearson Education India, 1988.

[4] Steven M Kay and Stanley Lawrence Marple Jr. Spectrum analysis - a modern perspective. Proceedings of the IEEE, 69(11):1380-1419, 1981.

References - Task 10

[1] Seong-Gon Kong and Bart Kosko. Comparison of fuzzy and neural truck backer-upper control systems. In Neural Networks, 1990., 1990 IJCNN In- ternational Joint Conference on, pages 349-358. IEEE, 1990.
[2] Derrick Nguyen and Bernard Widrow. Truck backer-upper: an example of self-learning in neural networks. In Orlando'90, 16-20 April, pages 596-602. International Society for Optics and Photonics, 1990.
[3] L-X Wang and Jerry M Mendel. Generating fuzzy rules by learning from examples. IEEE Transactions on systems, man, and cybernetics, 22(6):1414- 1427, 1992.
[4] Li-Xin Wang and Jerry M Mendel. Generating fuzzy rules from numerical data, with applications. Signal and Image Processing Institute, University of Southern California, Department of Electrical Engineering-Systems, 1991.