International Journal of Computer Science and Applications,
Vol. 14, No. 1, pp. 26 – 46, 2017

1REAL TIME STATE ESTIMATORS INTEGRATED IN A DC SERVOMOTOR ANGULAR SPEED CONTROL SYSTEM – RESULTS COMPARISON

ROXANA-ELENA TUDOROIU†
Mathematics and Informatics, University of Petrosani, 20 University Street, Petrosani, 332006, Romania
tudelena@mail.com

SORIN MIHAI RADU
Mechanical and Electrical Engineering, University of Petrosani, 20 University Street, Petrosani, 332006, Romania
sorin_mihai_radu@yahoo.com

WILHELM KECS
Mathematics and Informatics, University of Petrosani, 20 University Street, Petrosani, 332006, Romania
wwkecs@yahoo.com

NICOLAE ILIAS
Mechanical and Electrical Engineering, University of Petrosani, 20 University Street, Petrosani, 332006, Romania
iliasnic@yahoo.com

DUMITRU DAN BURDESCU
Software Engineering Department, University of Craiova, Craiova, 200585, Romania
dburdescu@yahoo.com

NICOLAE TUDOROIU
John Abbott College 2127 Lakeshore Road, Sainte-Anne-de-Bellevue, QC, H9X 3L9, Canada
Email: ntudoroiu@gmail.com

This paper is an interesting extension of the results presented in detail in the recent conference paper concerning the real time implementation of a sliding mode observer state estimator integrated in a dc Servomotor angular speed control system. In the extended version of this paper we introduced another linear estimator as a possible alternative to the Sliding Mode Observer estimator, more precisely a linear Kalman Filter. Based on their capability to perform well, and also to be robust to the possible changes in the control system dynamics, we are focused in our research to find the most suitable estimator for this kind of application. For comparison purpose of the estimators performance a detailed analysis is done at the end of this paper using as criterion the evolution of the residuals between the estimated values of the states and theirs true values during the transient and the steady state. Extensive simulations were carried out in this direction, in real-time environment, on the real time implementation platform MATLAB and SIMULINK.

Keywords: Kalman Filter; Sliding Mode Observer; mean square root error, dc servomotor.

1. Introduction

This paper is an extension of the results disseminated in the recent international conference paper, FedCSIS, September 11-13, 2016 from Gdansk, Poland, concerning the real time implementation of a sliding mode observer state estimator integrated in a dc

Bvd. Cetatii, Bloc 1/1, Sc:D, Ap. 10, Timisoara, Timis, Romania

Servomotor angular speed control system [Tudoroiu R-E et al., (2016)]. In the following, the extended version of this paper will be organized in two main parts, the first part related to the conference paper, and the second part that presents the new approach to implement in real time a second estimator based on Kalman Filter estimation. In the most situations, the design and the implementation of the real time control systems deals with two main concepts, such as the critical time and the timing constraints. Typically, the hard tasks are required to meet strictly the critical time constraints, while the soft tasks must to meet only the time constraints. Moreover, a real-time operating system exhibits several real-time multitasking features and also the most control systems of high complexity could have an embedded structure that requires a suitable modeling approach. For this reason we prefer to use in our simulations, as the most appropriate tool, the MATLAB/SIMULINK software package, due to its special features for real time implementation by its extensions Real-Time Workshop (RTW) and the Real-Time Windows Target (RTWT). As a case study is considered an embedded structure in a closed-loop speed control system of a dc servomotor, extensively used in the majority of the control applications field due to its high start torque characteristics, high response performance, and its speed easier to be controlled by varying the input voltage, compared to those that need expensive frequency drivers [Tudoroiu R-E et al., (2016)]. Moreover, the real-time dc servomotor speed control can be easily interfaced with MATLAB/SIMULINK platform. This paper is also based on our preliminary results to design and implement different real time estimators [Tudoroiu, N. et al., (2007)], and now we are interested to prove the effectiveness of the both proposed estimators. Concluding, the novelty of our research is to develop a more suitable, accurate and consistent real-time estimators to be used in our future real-time control strategies.

2. The DC Servomotor Dynamics Description

The electric circuit of the dc servomotor armature and the free body diagram of the rotor are shown in figure 1. For simulation purpose we will assume the same experimental values for the physical parameters as in the conference paper [Tudoroiu, R-E., (2016)]: moment of inertia of the rotor: $J = 0.001\ [\frac{kgm^2}{s^2}]$,damping ratio of the mechanical system: $b = 0.01[Nms]$,counter electromotive force coefficient $k_e = k_t = k = 0.0517$ [Nm/A], motor electric resistance: R = 1 [Ω], motor electric inductance: $L = 0.5[H]$, motor initial angular speed: $\omega = 1[\frac{rad}{s}]$, input dc power supply: $V = 12[V]$, load torque: $T_L = 0.1\sin(t)$ [Nm].

The dynamics of the dc servomotor actuator is described by the following input-state-output equations [Tudoroiu R-E et al., (2016)]:

$$J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} = k_t I_a - T_L$$
$$L\frac{dI_a}{dt} + RI_a = V - k_e\frac{d\theta}{dt}\ . \tag{1}$$

where $T_e = T = k_t I_a$ is the dc servomotor torque developed to the shaft, T_L is the load torque, and $e = k_e\frac{d\theta}{dt} = k_e\omega$ is the counter electromotive force. The equivalent state-space dynamics representation of the dc servomotor is given by the following equations [Tudoroiu R-E et al., (2016)]:

(i) State Equation:

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{k_t}{J} \\ -\frac{k_e}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -\frac{T_L}{J} \\ \frac{u}{L} \end{bmatrix} \tag{2}$$

$x_1 = \omega, x_2 = I_a, u = V, x_1(0) = 1 \left[\frac{rad}{s}\right], x_2(0) = 0[A]$

(ii) Output equation:

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3}$$

where x_1 is the measurable state, and the input voltage command $u = V$ is the dc power supply. At this stage the load torque T_L is assumed to be known and bounded.

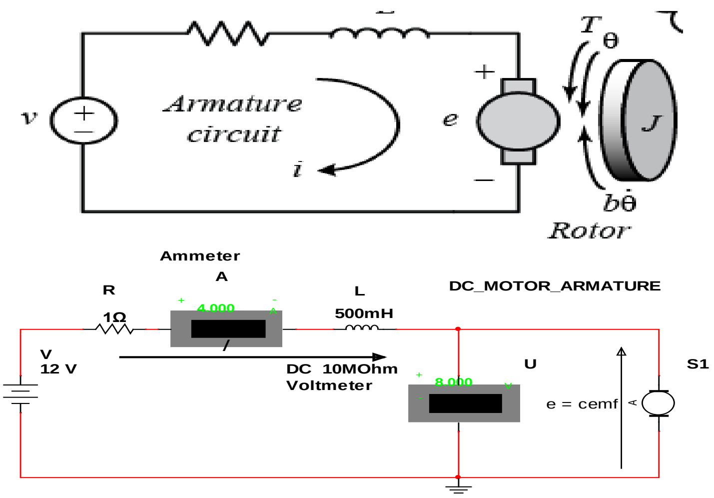

Fig. 1 The simplified equivalent electrical circuit of the dc servomotor (Reproduced from [Tudoroiu R.E., (2012); Tudoroiu R.E. et al., (2016)])

3. Linear Sliding Mode Observer Estimator Design of a DC Servomotor

The sliding mode methods in combination with observer control problems provide the ability to generate a sliding motion on the error between the measured plant output and the output of the observer such that to ensure that a sliding mode observer (SMO) produces a set of state estimates precisely matching with the actual output of the plant [Tudoroiu R.E. et al., (2016)]. Also the analysis of the average value of the applied observer injection signal, the so-called equivalent injection signal, contains useful information about the mismatch between the model used to define the observer and the actual plant. The development of SMO control strategy design in the extended version follows the same design guidelines suggested in the conference paper [Tudoroiu R.E. et al., (2016)]) related to the design of sliding mode observer (SMO) control strategies based on the equivalent injection signal principle [Spurgeon (2015a); Spurgeon (2015b); Yan et al., (2007)].

With the experimental values of the physical parameters introduced in section 2, the dynamics of the dc servomotor is described in state-space representation by a similar set of first order differential equations (2), (3):

$$\frac{dx}{dt} = A_{n\times n}x + B_{n\times m}u + D_{n\times q}\Psi(x,u,t)$$
$$y = C_{p\times n}x = [1 \quad 0]x \tag{4}$$
$$A = \begin{bmatrix} -10 & 51.7 \\ -0.1034 & -2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, D = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\Psi(x,u,t) = (-1000T_{Load}) = -10\sin(t).$$

where $x \in R^n$ is a n - dimensional state vector $(n = 2)$, $y \in R^p$ is a p-dimensional output vector $(p = 1)$, and $u \in R^m$ is a m-dimensional input vector $(m = 1)$. For our case study the load torque disturbance uncertainty is assumed to be $T_{Load} = 0$, (free load speed) and so $\Psi(x,u,t) = 0$. In the following we investigate only the linear case. By some manipulations of the matrices A, B, C we can easily find that B, C have a full rank and the pair (A, C) is observable, as main requirements assumed in [Spurgeon (2015a); Spurgeon (2015b); Yan et al., (2007)].

To design a sliding mode observer (SMO) firstly we attach to the original system an Utkin observer [Spurgeon (2015a); Spurgeon (2015b)] in a canonical form. For this task we need to find a nonsingular state transform $T_c \in R^{n\times n}$ that changes the state vector x in a state vector

$$z = T_c x = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, z_1 \in R^{n-p}, z_2 \in R^p, T_c = \begin{bmatrix} N_c^T \\ C \end{bmatrix}, N_c \in R^{n\times(n-p)} \tag{5}$$

where the column of the matrix N_C spans the null space of $C, \exists z_1 \neq 0_{(n-p)\times 1} \xrightarrow{\text{yields}} N_c \times z_1 = 0_{(n-p)\times 1}$.

The non-singular state transforms T_c is used to convert the nominal system (4) in the following canonical form:

$$\frac{dz_1}{dt} = A_{11}z_1(t) + A_{12}z_2(t) + B_1u(t) \tag{6}$$
$$\frac{dz_2}{dt} = A_{21}z_1(t) + A_{22}z_2(t) + B_2u(t)$$

Now the dynamics of the observer is described by the following similar equations:

$$\frac{d\hat{z}_1}{dt} = A_{11}\hat{z}_1(t) + A_{12}\hat{z}_2(t) + B_1u(t) + L\vartheta$$
$$\tag{7}$$
$$\frac{d\hat{z}_2}{dt} = A_{21}\hat{z}_1(t) + A_{22}\hat{z}_2(t) + B_2u(t) - \vartheta$$

where the pair $(\hat{z}_1, \hat{z}_2)$ represent the estimated values of the transformed components state vector z, and $L \in R^{(n-p)\times p}$ is the observer gain matrix, given by

$$\vartheta_i = M\,\text{sgn}(\hat{z}_{2,i} - z_{2,i}), M \in R_+, i = 1, \dots, p \tag{8}$$

The dynamics of the system errors is described by the following first order differential equations:

$$\frac{de_1}{dt} = A_{11}e_1(t) + A_{12}e_2(t) + L\vartheta$$
$$\frac{de_2}{dt} = A_{21}e_1(t) + A_{22}e_2(t) - \vartheta \tag{9}$$
$$e_1(t) = \hat{z}_1(t) - z_1(t),\ e_2(t) = \hat{z}_2(t) - z_2(t)$$

The observer gain matrix $L \in R^{(n-p)\times p}$ is chosen in order to make the spectrum of the matrix $(A_{11} + LA_{21})$ to lie in C_{-}, where the pair matrices (A_{11}, A_{21}) is observable due to the fact that the pair (A, C) is also observable.

Without to lose the generality we can choose the coordinates transform matrix such as:

$$T_c = \begin{bmatrix} N_c^T \\ C \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{10}$$

that converts the triple (A, B, C) into $(\widetilde{A}, \widetilde{B}, \widetilde{C})$, where the lines of the matrix N_c^T span the null space of the vector C, and also:

$$\widetilde{A} = T_cAT_c^{-1} = \begin{bmatrix} -2 & -0.1034 \\ 51.70 & -10 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$
$$\widetilde{B} = T_cB = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \widetilde{C} = CT_c^{-1} = [0 \quad 1] = [C_1 \quad C_2] \tag{11}$$

where $A_{11} = -2$ (stable), and $A_{12} = -0.1034$, $A_{21} = 51.70, A_{22} = -10$, $B_1 = 2, B_2 = 0, C_1 = 0, C_2 = 1$.

The value of the observer matrix gain *L* can be choose such as $A_{11} + LA_{21} < 0$, let take this $L = -1 < -\frac{A_{11}}{A_{21}} = 0.0387$. Setting the observer matrix gain *L* to 0.01 the dynamics of the linear observer and of its error are described by the following first order differential equations:

$$\frac{d\hat{z}_1}{dt} = -2\hat{z}_1(t) - 0.1034\hat{z}_2(t) + 2u(t) - \vartheta$$
$$\frac{d\hat{z}_2}{dt} = 51.70\hat{z}_1(t) - 10\,\hat{z}_2(t) - \vartheta$$
$$\frac{de_1}{dt} = -2e_1(t) - 0.1034e_2(t) + 0.01\vartheta \tag{12}$$
$$\frac{de_2}{dt} = 51.70e_1(t) - 10e_2(t) - \vartheta$$
$$\vartheta = \text{sgn}(\hat{z}_2 - z_2) = \text{sgn}(e_2(t)), M = 1$$

The dc servomotor is mostly used as an actuator in feedback closed-loop control systems, but in this research for simulation purposes it is design as a controlled plant. The main goal of the overall proposed control strategy is to control its angular speed or its position, or the both. Consequently, the dc servomotor can be connected in the hybrid control system with its integrated structure as in [Tudoroiu, (2012); Tudoroiu et al., (2015a); Tudoroiu et al., (2015b)], represented in figure 2. The dc power supply for dc servomotor is provided in this case by a dc buck converter driver. For simulation purpose, in order to prove the effectiveness of our proposed hybrid control strategy we investigate a 12V dc servomotor, and the experimental values of the physical parameters

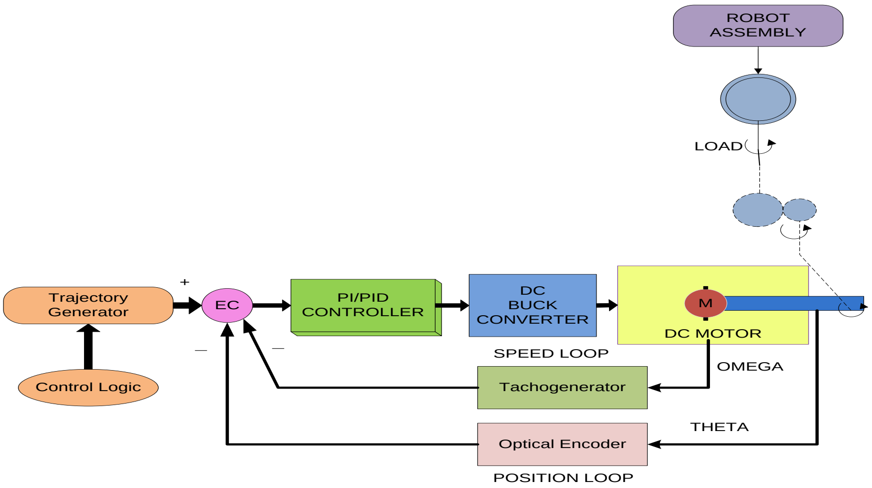

Fig. 2. The feedback closed-loop control system of the dc servomotor angular speed- schematic diagram (Reproduced from [Tudoroiu, (2012)]).

are closed enough to those derived by experiment from an actual dc servomotor in Carnegie Mellon's control lab of the University Michigan [Tudoroiu, (2012)]. The equivalent electrical schematic of the selected dc servomotor is presented in the SIMULINK model provided in MATLAB/SIMULINK library, as is shown in figure 3.

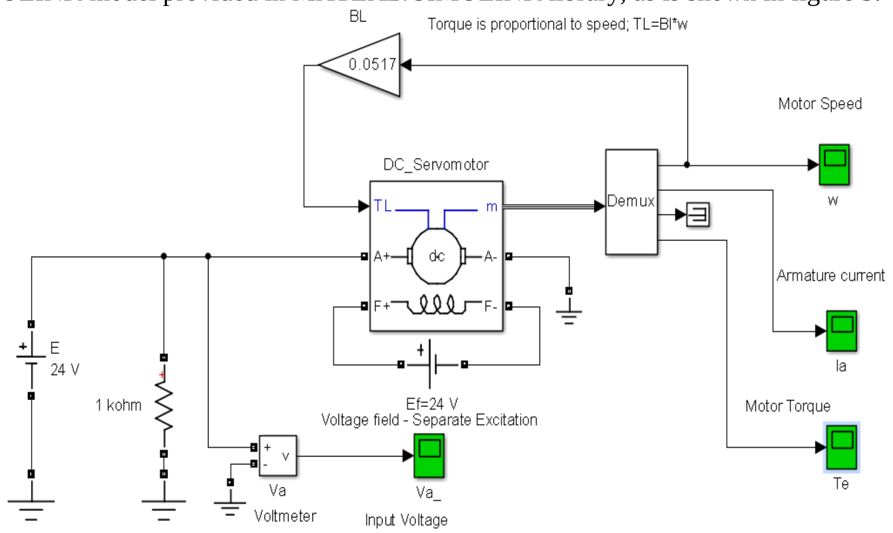

Fig. 3. The SIMULINK model of 24 V dc servomotor reproduced from MATLAB/SIMULINK library

In our case study the input voltage is set to 12V compared to the schematic from figure 3 where is set to 24V.

4. Sliding Mode Observer open loop simulation results

The model of the original system (2) in SIMULINK is shown in figure 4, and the evolution of the states, i.e. angular speed (x_1) and armature current (x_2) are shown in the figures 5 and 6.

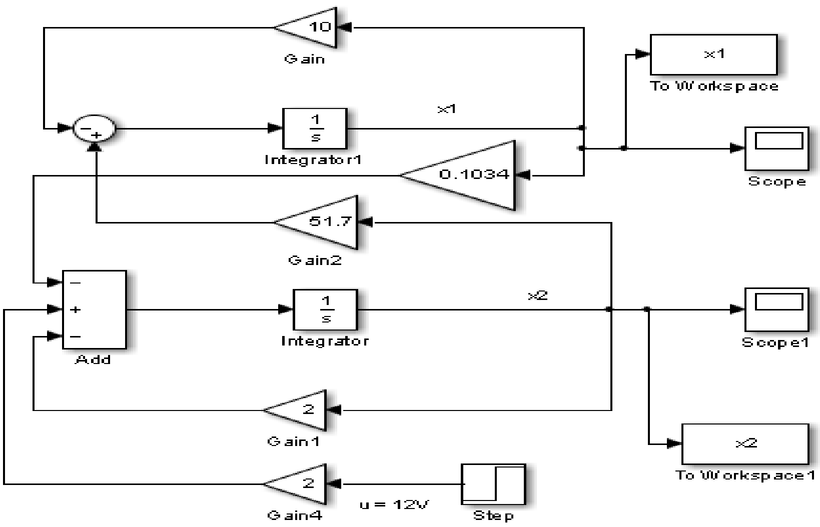

Fig.4. DC servomotor state space-representation of nominal model in MATLAB/SIMULINK

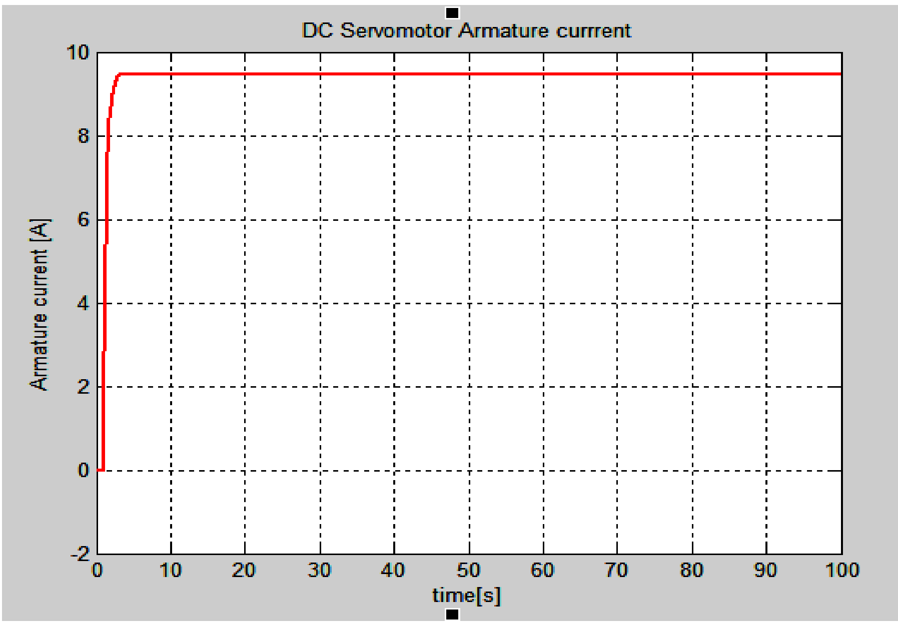

Fig. 5 DC servomotor armature current- original state-space representation model in MATLAB/SIMULINK

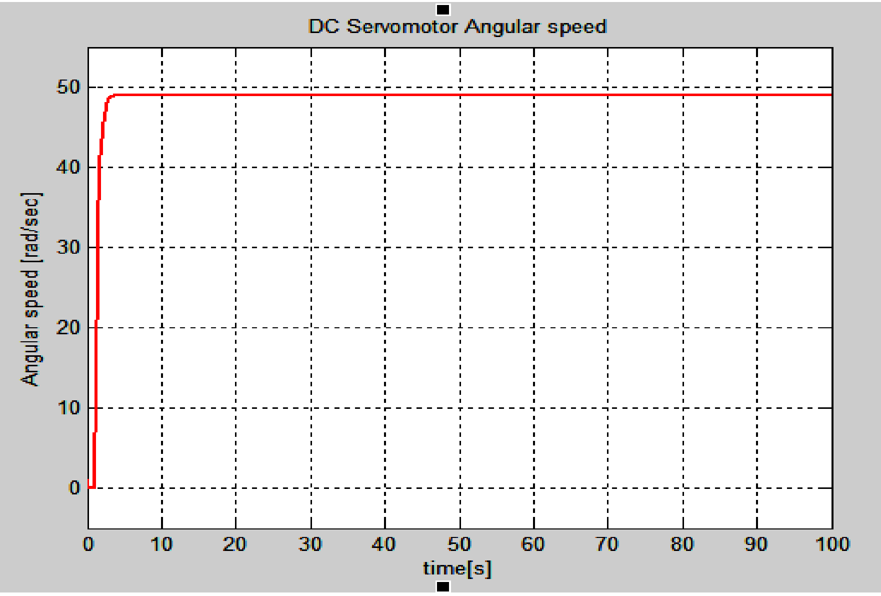

Fig. 6 DC servomotor angular speed- original state-space representation model in MATLAB/SIMULINK

The state-space representation model of the dc servomotor in the canonical form representing in the same time the Utkin classical observer SIMULINK model embedded
in the integrated control structure is shown in figure 7.

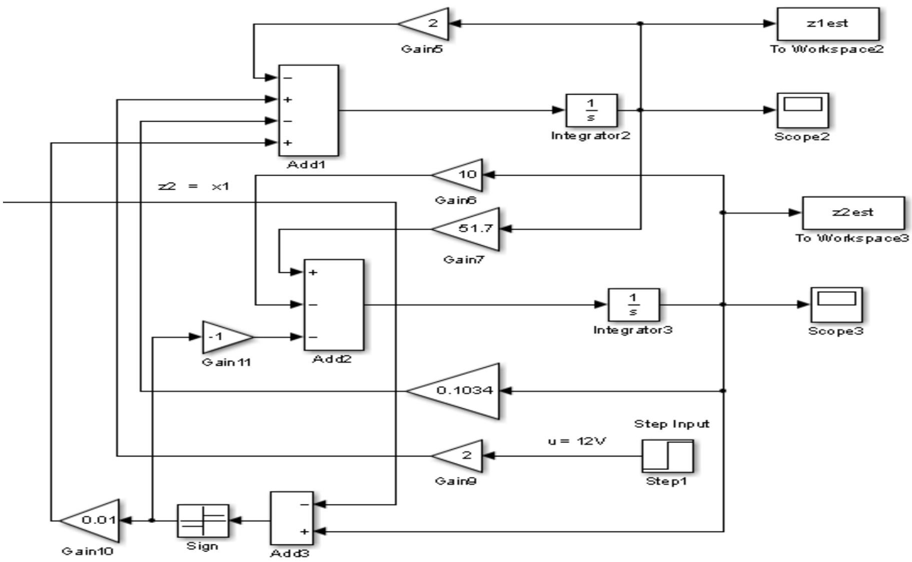

Fig. 7 Sliding Mode Utkin Observer - state space-representation in MATLAB/SIMULINK

The evolution of the estimated states, namely the angular speed (z_2) and the armature current (z_1) versus their true values are shown in figures 8 and 9.

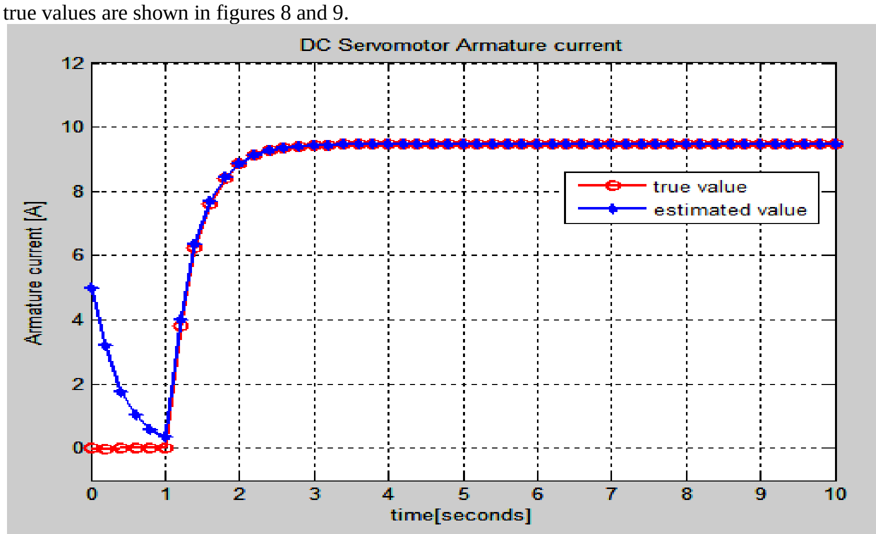

Fig. 8 DC servomotor armature current estimated versus nominal model using SMO control strategy in MATLAB/SIMULINK

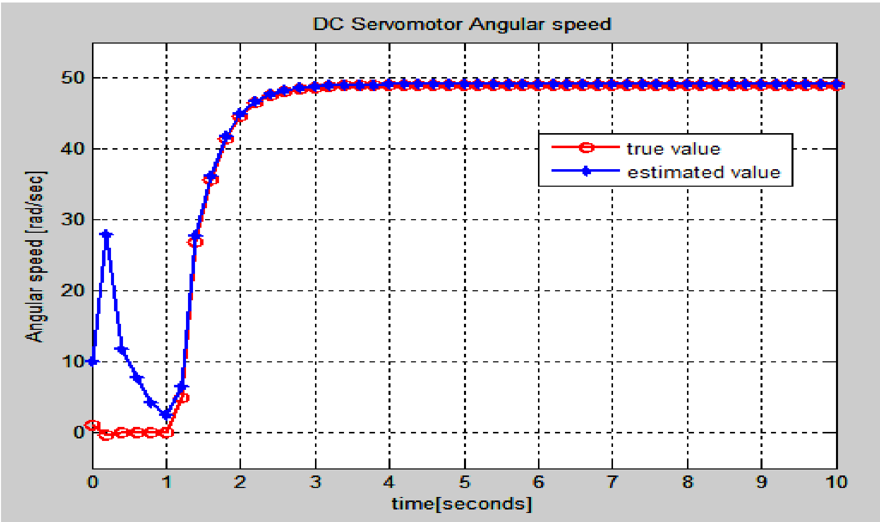

Fig. 9 DC servomotor angular speed estimated versus nominal angular speed model using SMO control strategy in MATLAB/SIMULINK

The SMO residuals of angular speed (e_2) and armature currents (e_1) are shown also in figures 10 and 11. For simplicity model purpose it is assumed that the rotor and the shaft of dc servomotor are rigid.

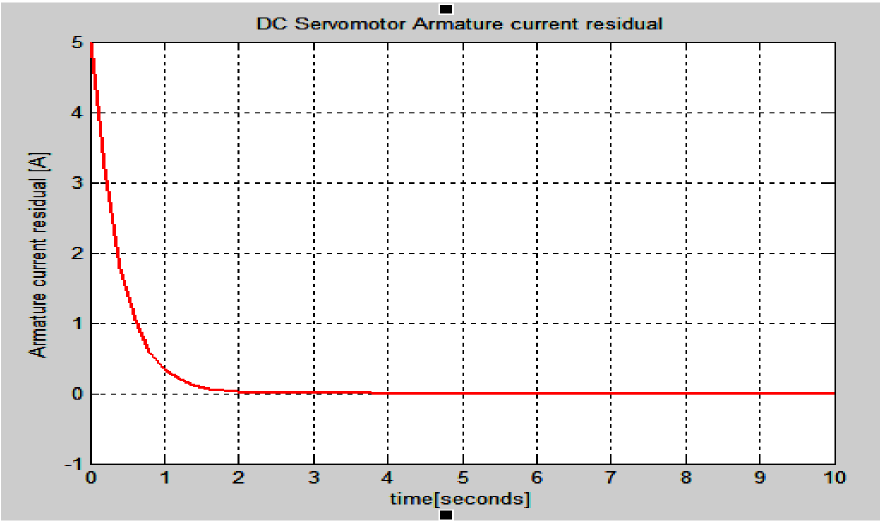

Fig. 10 DC servomotor SMO armature currents residual

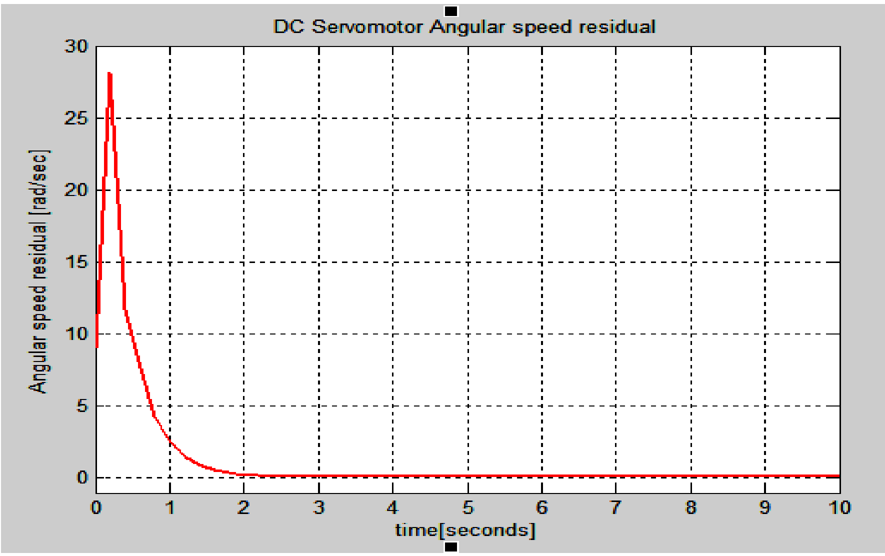

Fig.11 DC servomotor SMO angular speed residual

An ideal sliding motion will take place on the sliding surface [Tudoroiu, (2012); Tudoroiu et al., (2015a); Tudoroiu et al., (2015b)]:

$$S_w = \{(e_1, e_2)|e_2 = 0)\} \tag{13}$$

The corresponding switching function for observer gain set to L = 0.01 and M = 1 is shown in figure 12.

After some finite time t_s, for all subsequent time:

$$e_2 = 0, \text{ and } \quad de_2/dt = 0$$

and the corresponding sliding mode dynamics are given by

$$\frac{d\tilde{e}_1(t)}{dt} = \widetilde{A}_{11}\tilde{e}_1(t) \tag{14}$$

where

$$\tilde{e}_1(t) = e_1(t) + Le_2(t) = e_1(t) - e_2(t)$$

$$\text{and, also} \qquad \widetilde{A}_{11} = A_{11} + LA_{21} = -2 - 51.70 = -53.70 < 0.$$

Since $\widetilde{A}_{11} < 0$, the linear homogenous equation (14) has a stable solution:

$$\tilde{e}_1(t) = C_0 e^{-53.70t},$$

with C_0 as an integration constant determined from the initial condition $\tilde{e}_1(0) = \tilde{e}_{10}$. By a suitable choice of the gain *L*, such as in our case study $L = -1 < 0.0387$,we can conclude that always the system is stable, therefore

$$\tilde{e}_1(t) \xrightarrow{\text{yields}} 0, \qquad \text{and also} \qquad \hat{z}_1(t) \xrightarrow{\text{yields}} z_1(t) \text{ as } t \overset{\Delta}{\to} \infty.$$

If now we replace $\omega = \frac{d\theta}{dt}$ in the nominal system equations (2), (3) representing the link between angular speed (ω) and angular position(θ), the dc servomotor dynamics can be described by the following standard second order differential equation:

$$JL_a \frac{d^2\omega}{dt^2} + 2\zeta\omega_n \frac{d\omega}{dt} + \omega_n^2\omega = k_t u \tag{15}$$

where

$$y = \omega, u = V_1,\ \omega_n = \sqrt{\frac{R_a B_m + k_t^2}{JL_a}}, \zeta = \frac{B_m * L_a + R_a * J}{2}\sqrt{\frac{J * L_a}{R_a * B_m + k_t^2}}, k_m = \frac{k_t}{J * L_a}, \tag{16}$$

- ω_n represent the natural frequency of the free oscillations, ζ the damping factor, and k_m is the dc servomotor gain.

Based on the same set up, by replacing the numerical values of the electrical and mechanical parameters of our proposed dc servomotor machine we get the following dc servomotor transfer function representation:

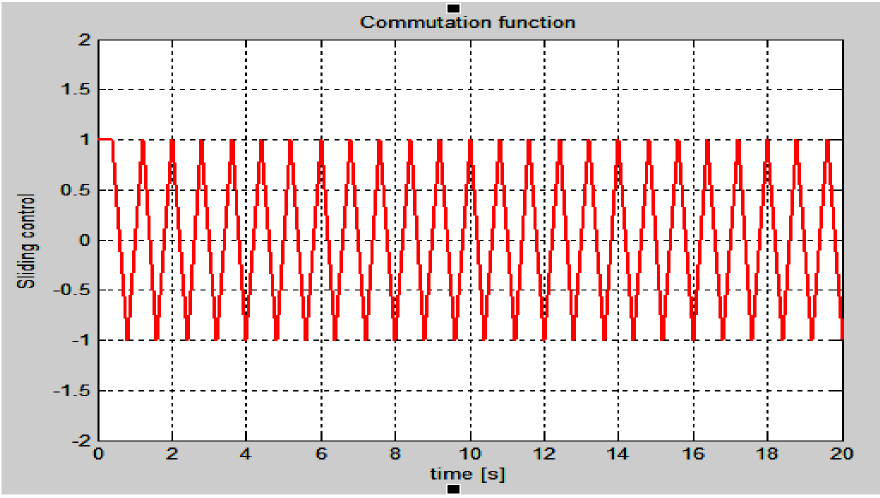

Fig. 12 SMO control switching function around sliding line

$$H(s) = \frac{k_m}{s^2+2\zeta\omega_n s+\omega_n^2} = \frac{Y(s)}{U(s)} = \frac{103.4}{s^2+3s+25.3458} \quad (17)$$

where s represents the equivalent of the derivative operator in complex domain, and $U(s), Y(s)$ represent the Laplace images of u(t) (input voltage) and $y(t)$ (output voltage), respectively. In MATLAB simulations environment the equation (15) is modeled by using MATLAB step response command provided by CONTROL SYSTEMS MATLAB TOOLBOX as is done in [Tudoroiu, (2012); Tudoroiu et al., (2015a); Tudoroiu et al., (2015b)]. The dc servomotor step response is represented by using MATLAB software package as is shown in figure 13, very similar in steady state to those obtained in SIMULINK simulations, as is depicted in figure 5.

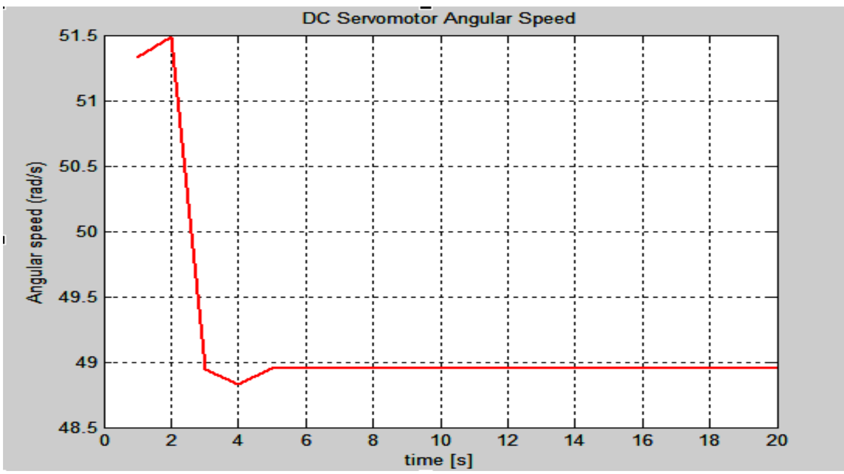

Fig.13The dc servomotor angular speed step response – MATLAB simulation results

5. Standard Linear Kalman Filter Estimator

In this section we investigate a new state estimation strategy approach such as the linear Kalman Filter estimator, capable also to provide an elegant and powerful solution for our case study. In the last five decades Kalman Filter estimation technique proved already a good ability as a suitable tool for dynamic system state estimation. It is extensively used in many applications area, such as the target tracking, global positioning, dynamic systems control, navigation, and communication [Plett, (2004)].

Essentially, the Kalman filter estimator consists of a set of recursive equations that are repeatedly evaluated during the dynamic evolution of the control system. The main idea to develop this new estimation technique is based on the simple fact that any causal dynamic system generates its outputs by considering only the measurable data set of the past and present inputs values. This is nothing else than a new systemic concept which considers that the state vector of any dynamical system (which may not be directly measurable) accumulates entirely the effect of all past inputs values on the system. The actual value of the system output may be computed only based on the actual input and actual state values, without to store also its past input values. We apply the linear Kalman filter to estimate the both dc servomotor states (armature current and angular speed) based on the measurable data input-output set values (the armature supply voltage as the input, and the angular speed as the output). In the new approach the dynamics of dc servomotor is described in discrete-time state space representation by the following equations wrote at time index "*k*" [Plett, (2004)]:

$$x(k+1) = \mathrm{A}_{2\times2}x(k) + \mathrm{B}_{2\times1}u(k) + w(k) \tag{18}$$

$$y(k) = \mathrm{C}_{1\times2}x(k) + v(k) \tag{19}$$

with $\mathrm{A}_{2\times2} = \begin{bmatrix} -10 & 51.7 \\ -0.1034 & -2 \end{bmatrix}, \mathrm{B}_{2\times1} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \mathrm{C}_{1\times2} = \begin{bmatrix} 1 & 0 \end{bmatrix}$

First equation (18) is the "*state equation*", more precisely the "*input-state process equation*" that describes completely the dynamic evolution of the control system. In addition, the key system characteristics such as stability, controllability and sensitivity to disturbance can be entirely analyzed based on the same equation (18) [Plett, (2004)]. In this equation the known/deterministic input to the system is represented by $u(k)$, and $w(k)$ describes the so called stochastic "*process noise*" that models some unmeasured input which affects the state of the control system. The output of the system is designated by $y(k)$ computed by the "*output equation*" (19) as a linear combination of the overall control system states, its input $u(k)$ and an additional "*disturbance*" input $v(k)$, as a modelling "*sensor noise*" that affects the measurements on the system output. Based on the dynamic model described by the equations (18) and (19) and assuming that there is enough knowledge about the measurements data set on the input-output system's signals we can estimate in real time, in a dynamic environment, the unmeasured state $x(k)$ of the corresponding physical system, as is shown in [Plett, (2004)]. According to this research document the Kalman Filter is an optimum numerical recursive method to solve the state estimation problem under certain assumptions. The first assumption is asking that the both process, $w(k)$, and measurement, $v(k)$, noises are mutually uncorrelated white Gaussian random processes, with zero mean and covariance matrices with known value, Q_w, respectively R_v. In practice, this requirement is rarely met, but the consensus of the literature is that the Kalman Filter estimation method still is working very well.

Moreover, by modeling the dc servomotor dynamics with the desired unknown quantities in the model state vector, the Kalman Filter is capable to compute automatically the best state estimate of their actual values [Plett, (2004)].

Summarizing, the Kalman Filter problem can be formulated as:

"*Given the dynamic model of the control system described by the equations (18),(19), a complete set of observed measurements input-output data set, with the assumptions on w(k) and v(k), find the minimum mean squared error estimate of the true state x(k)*".

The solution to this problem is widely known and is given in the following steps, and for more details, see for example the research documents [Haykin, (1996)], [Plett, (2004)]. More precisely, the core of the solution to this problem is a set of computationally efficient recursive relationships that involve both an estimate of the state itself, and also the covariance matrix of the state estimate error, that indicates the uncertainty of the state estimate, and may be used to generate error bounds [Plett, (2004)] According to [Plett, (2004)] a "large" covariance matrix of the state estimate error in terms of its singular values, i.e. one with large singular values, indicates a high level of uncertainty in the state estimate in comparison with a "small" covariance matrix of the state estimate error, so one with small singular values, indicates confidence in the estimate. The discrete-time Kalman filter computes two different estimates of the state and covariance matrix each sampling interval, more precisely the recursive relationships can be grouped in the following two phases, starting at time index k:

Phase 1- Prediction or forecasting phase:

Step 1.1: Calculate the "priori" state $\hat{x}(k+1|k)$:

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) \tag{20}$$

Step 1.2: Calculate the prediction covariance matrix of the state estimate

$$\hat{P}(k+1|k) = A\hat{P}(k|k)A^T + Q_w = E\{[\hat{x}(k+1|k) - x(k+1)][\hat{x}(k+1|k) - x(k+1)]^T\} \ , \tag{21}$$

were E{.} is the statistical expectation operator and a superscript T is the vector or matrix transpose.

Step 1.3: Calculate the Kalman Filter Matrix Gain, K

$$K = \hat{P}(k+1|k)C(C\hat{P}(k+1|k)C^T + R_v)^{-1} \tag{22}$$

Phase 2: Correction or measurement update phase:

Step 2.1: Calculate the state estimate measurement update $\hat{x}(k+1|k+1)$

$$\hat{x}(k+1|k+1) = \hat{x}(k|k+1) + K(y(k) - C\hat{x}(k+1|k) \quad (23)$$

Step 2.2: Calculate the error covariance state measurement

$$\hat{P}(k+1|k+1) = (I - KC)\hat{P}(k+1|k) \quad (24)$$

The Kalman Filter is initialized with the best available information on the state estimate $\hat{x}(0|0)$, and its error covariance $\hat{P}(0|0)$.

Habitually, these statistics values are not precisely known, and initialization must be performed in a randomly manner. This is not a problem as the Kalman Filter is known to be very robust to poor initialization, and will quickly converge to the true values as it runs [Plett, (2004)]. Following initialization, the Kalman Filter repeatedly performs the above two phases' calculations steps in each measurement interval.

6. Kalman Filter linear estimator open loop simulation results

Kalman Filter estimate of the dc servomotor armature current versus its true value is shown in figure 14.

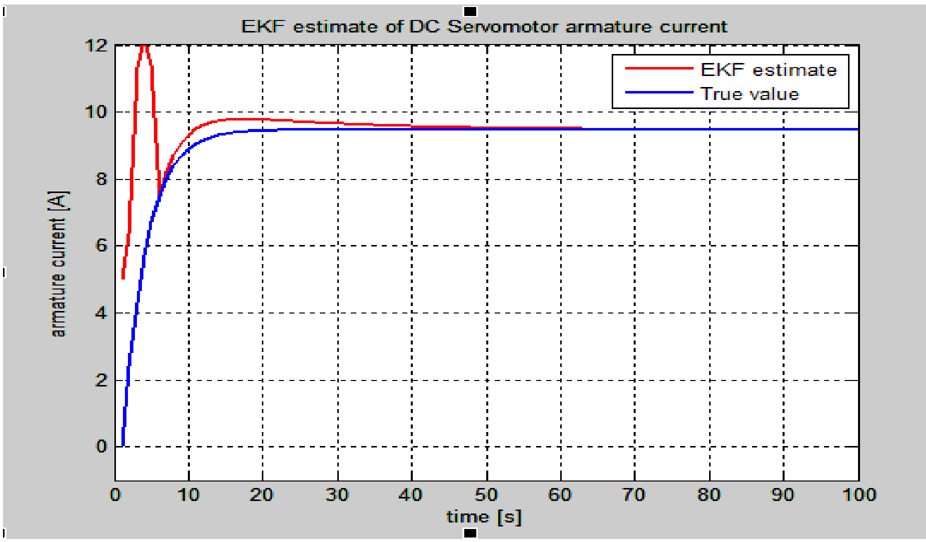

Fig.14. Kalman Filter estimate of the dc servomotor armature current versus its true value – MATLAB simulation results for the level noise with the standard deviation 0.01.

In figure 15 is shown the dc servomotor angular speed Kalman Filter estimate versus its true value. The both figures reveal the robustness of the Linear Kalman Filter estimator to the changes in the initial condition value for armature current and angular speed, respectively. The noise level in the both figures is 0.01 standard deviation, the same for process and measurement noise. The armature current and angular speed residuals are shown in figures 16 and 17.

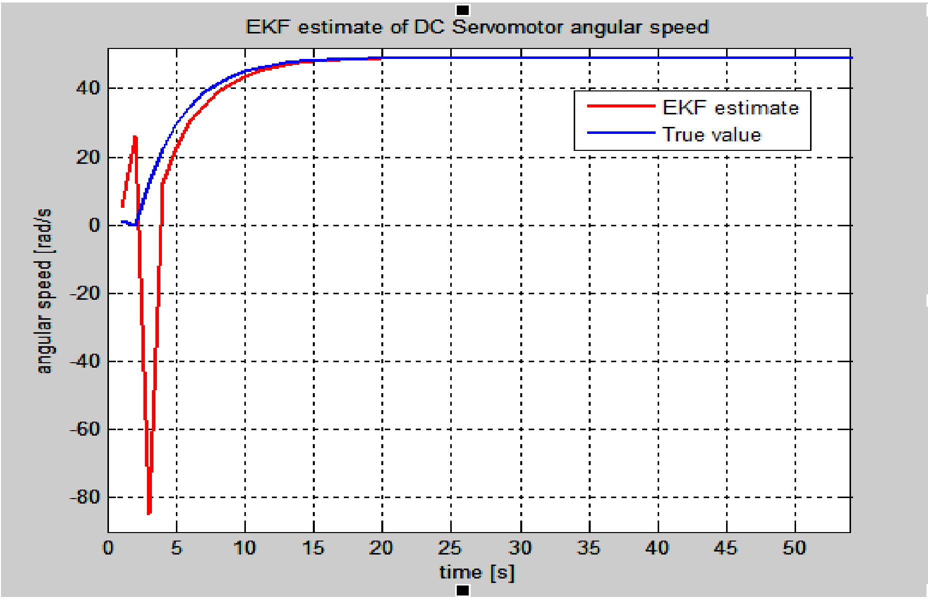

Fig.15. Kalman Filter estimate of the dc servomotor angular speed versus its true value – MATLAB simulation results for the level noise with the standard deviation 0.01.

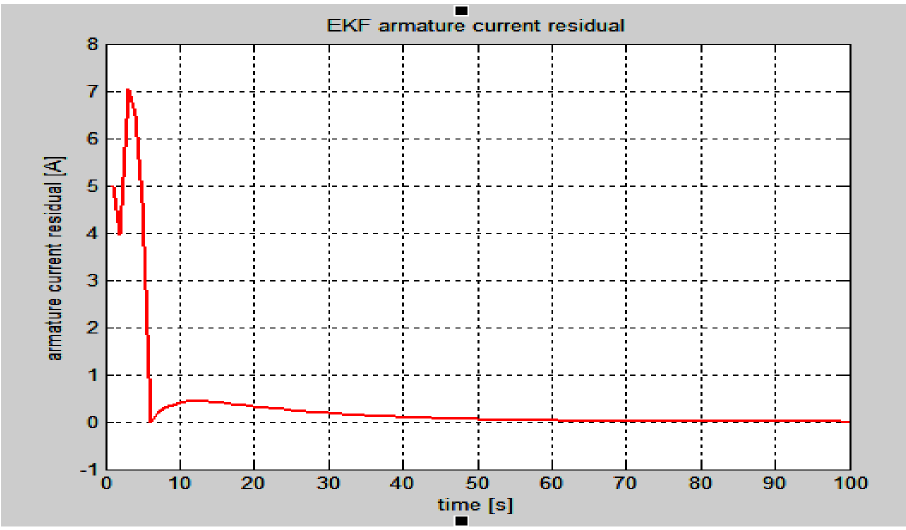

Fig.16. Kalman Filter armature current residual – MATLAB simulation results for the level noise with the standard deviation 0.01.

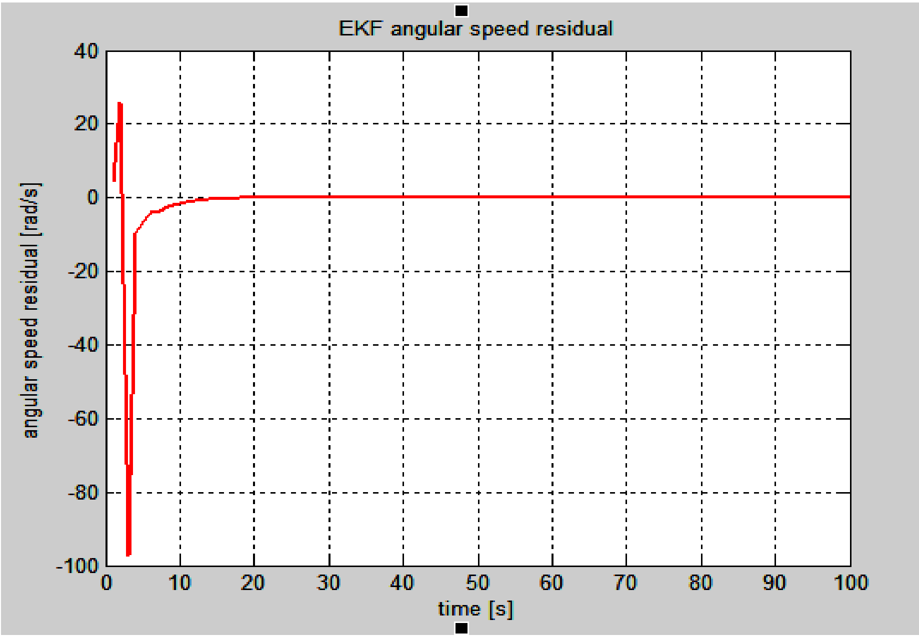

Fig.17. Kalman Filter angular speed residual – MATLAB simulation results for the level noise with the standard deviation 0.01.

If the level of the process and measurement noise increases 10 times we get the following changes in the dynamic evolution of the armature current and angular speed, as is shown in the figures 18 and 19.

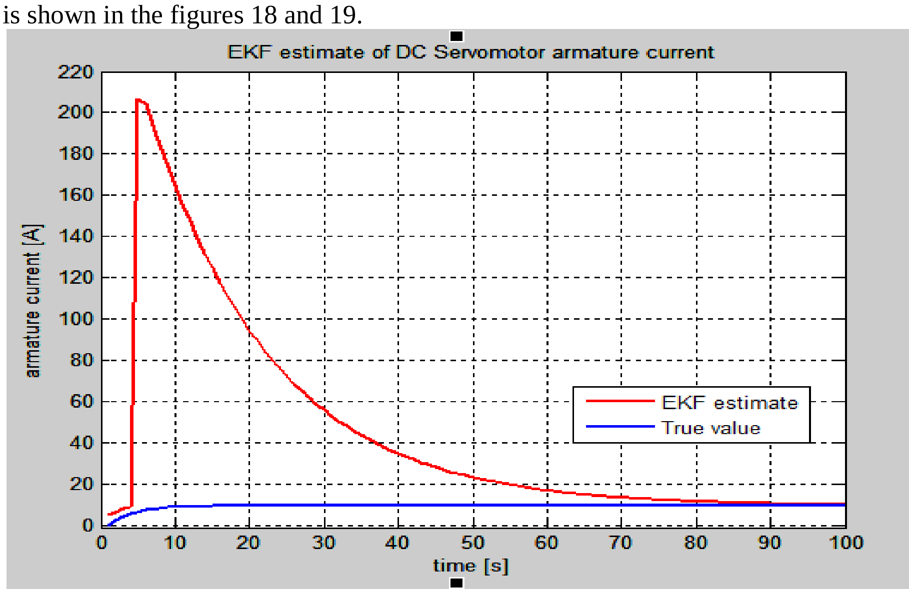

Fig.18. Kalman Filter estimate of the dc servomotor armature current versus its true value – MATLAB simulation results, for the level noise with the standard deviation 0.1.

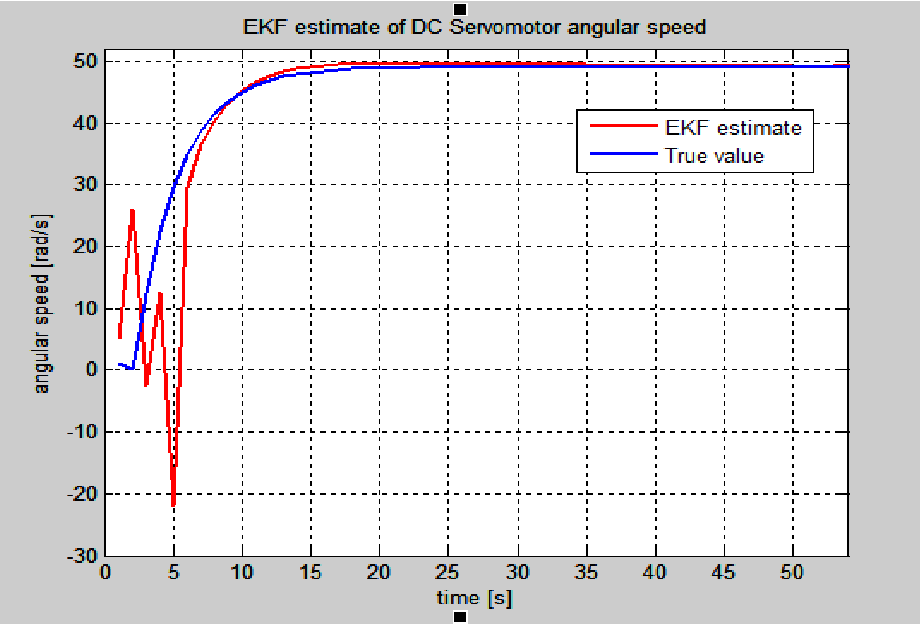

Fig.19. Kalman Filter estimate of the dc servomotor angular speed versus its true value – MATLAB simulation results for the level noise with the standard deviation 0.1.

The armature current and angular speed residuals are shown in figures 20 and 21 for the noise level increased 10 times.

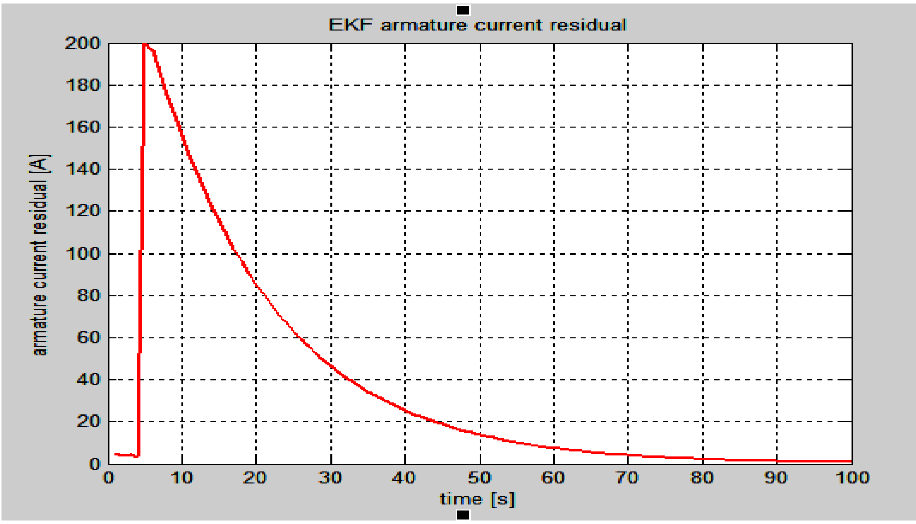

Fig.20. Kalman Filter armature current residual – MATLAB simulation results for the level noise with the standard deviation 0.1.

From these two last figures we notify a significant variation of the armature current residual during the transient due to the effect of the increasing noise 10 times.

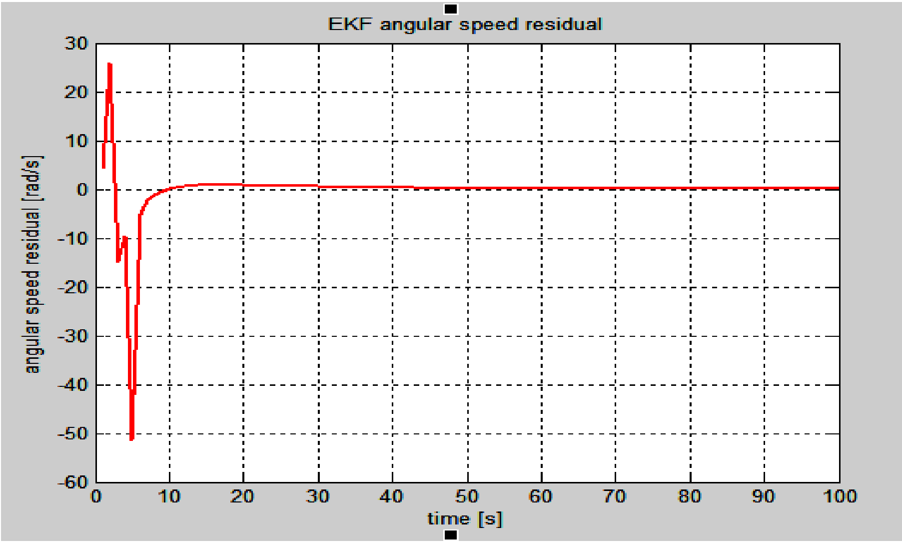

Fig.21. Kalman Filter angular speed residual – MATLAB simulation results for the level noise with the standard deviation 0.1.

We also observe a very good robustness of Kalman Filter estimator for angular speed and a robustness degrade for armature current estimation when the level process and measurement noises increases.

7. Sliding Mode Observer Estimator versus Linear Kalman Filter Estimator - Results Comparison

Comparing the results depictured in the figures 8, 9 related to the Sliding Mode Observer performance with those offered by the figures 14, 15, 18 and 19 related to the Linear Kalman Filter Estimator it is not difficult to see the fast convergence of the both algorithms during the transient, with a slightly superiority of the Sliding Mode Observer, that is not dealing with the level of noise as in the case of the Linear Kalman Filter that is sensitive. We could notify easily the same aspects if we analyze the residuals of the both estimators depictured in the figures 10, 11, and 16, 17, 20, 21, respectively. Also for the Linear Kalman Filter estimator we notify a big variation in the dynamic evolution of the armature currents and angular speed, revealing a big spike due to the numerical algorithm used by the MATLAB solver. Perhaps using SIMULINK simulations as in the case of Sliding Mode Observer estimator the solver selected improve significantly the convergence of this algorithm.

Concluding, it is our opinion that for this kind of application the both estimators perform very well, with a slightly superiority of the Sliding Mode Observer, since is not dealing directly with the stochastic noise and its statistics. The Sliding Mode Observer is dealing directly with the dynamic models affected only by the deterministic disturbances that meet some assumptions. Perhaps for the applications with nonlinear models we expect that the performance of both estimators to change significantly.

8. Sliding Mode Observer and Linear Kalman Filter Estimators - Real-Time Implementation

In the control systems literature rarely we find details about the real-time software and hardware implementation aspects, and no sufficient attention is given about the algorithms and the sampling time selection [Tudoroiu R-E et al., (2016)]. Usually the implementation aspect and real-time control systems design are connected together but in the most cases this connection is always ignored. Furthermore the real-time control systems design is treated from control perspective ignoring the implementation aspects of the control algorithms. Luckily, the most recent real-time implementation and design aspects acquire a considerable attention from part of control engineering community by the conceiving of new software tools, especially MATLAB/SIMULINK with new features, such as RTW (Real-Time Workshop) and RTWT (Real-Time Windows Target) Toolboxes. For our research case study we use MATLAB R2013a with SIMULINK as a real-time platform for which the real-time simulations run on two processors WINDOWS OS machine [Tudoroiu R-E et al., (2016)]. Also the implementation of real-time control applications is easier and time saving. Well, some drawbacks regarding a good perception on the real-life control systems applications could appear in a real-time implementation environment provided by these platforms.

9. Conclusions

In this paper, we have studied the possibility of using two linear estimators, namely a Sliding Mode Observer and the Kalman Filter to estimate the states of a dc Servomotor actuator that can be integrated in the same control system structure. The purpose of the implementation in real time of both estimators is to find the best estimation tools for our future developments, especially in fault detection and isolation (FDI) control. The main contributions in our research are summarized briefly as follows:

a) The development of two different approaches to design in real-time two different estimators., such as a Sliding Mode Observer (SMO) and a Linear Kalman Filter
b) Comparison of their performance capabilities and the advantage of the real-time implementation
c) An extensive investigation of using these estimators tools in our future research developments.

References

Gambier, A. (2004): Real-time Control Systems: A Tutorial, http://www.ascc2004.ee.mu.oz.au/proceedings/papers/P150.pdf

Haykin, S. (1996): Adaptive Filter Theory, 3rd ed., Prentice-Hall, Upper Saddle River, NJ.

Plett, G.L., (2004): Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1. Background, Journal of Power Sources, ELSEVIER, **134**, pp. 252–261.

Spurgeon, Sarah K. (2015): Sliding Mode Observers - historical background and basic introduction, Spring School, Aussois.

Spurgeon, Sarah K. (2015): Sliding Mode Observers – toward a constructive design framework, Spring School, Aussois.

Tudoroiu, R-E, Kecs, W., Dobritoiu, M., Ilias, N., Casavela, S-V., Tudoroiu, N. (2016): Real-Time Implementation of DC Servomotor Actuator with Unknown Uncertainty using a Sliding

Mode Observer, Proceedings of the Federated Conference on Computer Science and Information Systems, Gdansk, Poland, DOI: 10.15439/2016F95, ACSIS, ISSN 2300-5963, Vol. 8, pp. 841–848.

Tudoroiu, R-E., Astilean, A., Letia, T., Iliasi, N., Tudoroiu, N., Burdescu, D.(2015): Wireless UML-RT Simulator for Modeling and Implementing Dynamics Hybrid Structure of a Real Time PI DC Motor Control System, The Journal of Economics and Technologies Knowledge (JETK), 1(4), pp.69-81.

Tudoroiu, R-E., Astilean, A, Letia, T., Cretu, V., Iliasi, N., Tudoroiu, N., (2015): UML- RT Simulator for Modeling and Implementing Hybrid Structure Dynamics of a Real Time PID DC Servomotor Position Control System Strategy, The Journal of Economics and Technologies Knowledge (JETK), 1(5), pp. 68-77.

Tudoroiu, N., and Khorasani, K. (2007): Satellite Fault Diagnosis using a Bank of Interacting Kalman Filters, Journal of IEEE Transactions on Aerospace and Electronic Systems, DOI: 10.1109/TAES.2007.4441743, Vol. 43, no.4, pp. 1334-1350.

Tudoroiu, R-E., (2012): Conceiving and Implementing Applications using Real-Time UML, PhD Thesis, Cluj-Napoca Technical University, Romania.

Yan, X.G., Eduards, C., (2007): Nonlinear robust fault reconstruction and estimation using a sliding mode observer, Elsevier, ScienceDirect, Automatica, DOI:10.1016/j.automatica.2007.02.008, Vol. 43, pp.1605 – 1614.