

Dynamic Simulations & Animations of the Classical Control Techniques with Linear Transformations

Ahmet ALTINTAŞ^{*}, Mehmet Emin GÜVEN

*Dumlupınar University, Faculty of Technical Education, Kütahya
Gazi University, Faculty of Technical Education, Ankara*

Received: 22.05.2006 Accepted: 30.04.2007

ABSTRACT

Teaching and learning techniques using computer-based resources greatly improve the effectiveness and efficiency of the learning process. Currently, there are a lot of simulation and animation packages in use, and some of them are developed for educational purposes. The dynamic simulations-animations (DSA) allow us to see physical movement of the different pieces according to the modeled system. Education-purposed packages cannot be sufficiently flexible in different branches of science. Therefore, some educators prefer developing their own simulation and animation packages. This paper reports on the developing of dynamic simulation/animation using classical control techniques to enhance learning and interest for the courses of automatic control systems. Thus, the effects of control techniques on system performance will be observed visually with ease. A new point of view is also introduced to educators in order to realize personal DSAs. DSAs have been realized with computer graphics using fundamental linear transformations. In addition, the paper briefly introduces several exemplary applications related to electrical and robotics engineering.

Key Words: Control techniques, Animation, Linear transformations, Computer graphics

1. INTRODUCTION

When a new mechanism is developed or a system is formulated, it is often convenient to study the system's performance by simulation before building the breadboard or prototype. The simulation not only validates the system's performance, but also permits optimization of the system's performance by iteration of its parameters. Thus, valuable time can be saved in the development and design of a product, and the failure of components of poorly designed systems can be avoided. Animation is any application that generates a series of frames, so that each frame appears as an alteration of the previous one, and where the sequence of frames is determined either by the designer or the user. Thus, animations not only depict the current state of objects; they also deliver information concerning changes in objects and their position over time as well as information concerning the direction of these changes. DSAs are realized by using both the data obtained from the simulation of a mathematically-modeled system and the frames formed with sequential movements. The term animation covers a broad range of software applications such as kinematics and dynamics. For kinematic animations, use of keyframes and motion capture constitutes the primary means of driving the animation sequence (e.g. Poser, 3D Studio Max, Jack, etc.). Dynamic animation is driven by the outputs of a

simulation program to provide a 2D or 3D display of the physical characteristics of the application. The output may contain simplified geometric objects such as lines, rectangles, circles, etc. A number of researchers have reported their efforts in using animation to enhance the learning process [1-3]. In [1], an engineering animation tool was introduced. In [2], a real time simulation/animation tool was developed to facilitate the evaluation of an active suspension system. In [3], animation of a flexible manufacturing system was carried out. Also, Visual C++, Direct-3D, Matlab, VisSim, and Lab VIEW-based softwares were generated for interactive modeling, simulation and animation [4-8].

In the sense of teaching pedagogy, the traditional treatment of all fields of engineering tends to be highly theoretical and mathematical with heavy emphasis on equation derivation and algorithmic development. Such an approach is convenient from the instructor's point of view, but may not be beneficial to the students. DSAs often enrich modern education in all areas. Today, there are a lot of simulation and animation packages in use, and each of them has some strong and some weak features. Important demands for education are visualization of the simulation results and the interactivity of the simulation. Animations based on

^{*}Corresponding author, e-mail: a_altintas@dumlupinar.edu.tr

interactive simulations are an effective way to go deeper inside a problem. The user/student has to have the ability to influence parameters and/or conditions during the simulation/animation and thereby see the effect of these variations immediately in his simulation/animation. The package programs of DSAs provide a convenient tool whereby many scenarios can be tried with ease. This is very helpful for better understanding. However, some educators are recommending and using their own products for simulation and animation, because the package programs can not be sufficiently flexible in different scientific branches, and may not produce satisfactory solutions under given conditions.

Proportional-Integral-Derivative (PID) controllers are still extensively used in numerous industrial applications despite the significant developments of recent years in control theory and technology. It was recently estimated that more than 90% of all control loops involve PID controllers. This is because they perform very well for a wide class of processes. They give robust performance for a wide range of operating conditions. Also, they are easy to implement by using analogue or digital hardware. Furthermore, their simplicity, availability, reliability and cost-effectivity are among the reasons for their success [9, 10].

In this paper, the aim is to introduce a different point of view on the use of DSAs in automatic control systems; thus, an educator realizes his/her own package program easily, improves the effectiveness of utilizing animations-simulations, and enhances learning and interest for the courses of automatic control systems. In this method, DSAs are created with the Matlab package program by using the linear transformation matrixes. To realize a DSA, first of all the mathematical model of the physical system is expressed with a proper-mathematical method (differential equations, transfer functions, state-space model, etc.), and the control technique (P, PI, PID, etc.), which will be used as a controller in the modeled system, and the controller parameters (K_p, K_i, K_d) are determined. Then, simulation of the modeled system is performed with the Matlab Control System Toolbox, and the step response of that is obtained graphically and numerically. After getting the necessary data from step response, the stage of dynamic animation can be carried out. The necessary part of the system is formed with position vectors graphically, and the moving (animated) parts, which will be controlled with the selected controller type, are treated successively with a proper linear transformation depending on the step response of the model. Finally, the successive figure pages are combined with the 'movie' function of Matlab, and the DSA is realized successfully. As a consequence, the user/student will see visually the effects of both controller type and controller parameters on system performance.

2. CLASSICAL CONTROL TECHNIQUES AND THEORETICAL CHARACTERISTICS

A control system consists of an interconnection of components so that the output of the overall system will follow as closely as possible the desired signal. Many industrial automatic controllers are electronic, hydraulic, pneumatic, or combinations thereof. Industrial controllers, also, may be classified according to control actions. Generally there are four basic control actions: on-off control, proportional (P) control, integral (I) control and derivative (D) control. Sometimes it is necessary to combine control actions. So, there are also some combinational control actions consisting of P, I, D control actions: proportional-integral (PI) control, proportional-derivative (PD) control and proportional-integral-derivative (PID) control [9].

2.1. The Effects of Control Actions on System Performance

The simplest control action is the proportional control. This type control action speeds up the system performance to some extent. The proportional control provides a control signal that is proportional to error.

In the proportional control of a plant whose transfer function does not possess an integrator $1/s$, there is a steady-state error, or offset, in the response to a step input. Such an offset can be eliminated if the integral control action is included in the controller.

Derivative control action, when added to a proportional control, provides high sensitivity. An advantage of using derivative control action is that it responds to the rate of change of the actuating error and can produce a significant correction before the magnitude of the actuating error becomes too large. Derivative control thus anticipates the actuating error, initiates an early corrective action, and tends to increase the stability of the system [9, 11, 12].

2.2. Performance Criteria

In general, the performance of control systems is divided into two parts: the steady-state performance, which specifies accuracy, and transient performance, which specifies the speed of response. These performances are defined with respect to test signals.

In many practical cases, the desired performance characteristics of control systems are specified in terms of time domain quantities. The steady-state performance is concerned with the response of $y(t)$ as t (time) approaches infinity. It is defined for step, ramp and parabolic inputs. Transient performance is concerned with the speed of response or the speed at which the system reaches the steady-state. Although the steady-state performance is defined for a step, ramp or parabolic reference input; the transient performance is defined only for a step reference input. The transient response of a practical control system often exhibits damped oscillations before reaching steady-state. The

transient performance is generally specified in terms of the rise time, settling time, peak time and maximum overshoot.

The time-domain specifications are quite important because most control systems are time-domain systems; that is, they must exhibit acceptable time responses. This means that the control system must be modified until the transient response is satisfactory. Transient performance of the controlled system directly depends on the controller parameters. Depending on controller parameters, the system may respond in one of four ways: undamped system, underdamped system, critically damped system, overdamped system (Fig.1) [9, 10].

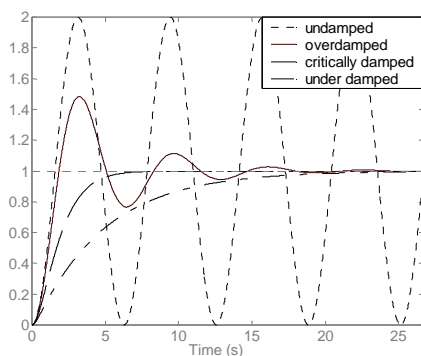


Figure 1. The four ways of transient response.

3. MATERIAL AND METHOD

3.1. Material

Simulations and animations are based on computer graphics. Many of the most important programs for computer graphics have been written in traditional programming languages (Fortran, Pascal, C, etc.). However, in the last recent years, the general-purpose numerical computation programs (NCP) are gaining more and more popularity. Today, they are well-established as a powerful alternative to traditional programming languages in many different areas, such as electrical engineering, mechanical engineering, signal processing, power systems, etc. Nevertheless, not all the NCPs offer the same advantages and features. In this paper we have decided to use Matlab package program in order to realize educational-purposed DSAs.

Matlab is a matrix-based software for scientific and engineering numeric computation and visualization [13]. Matlab is chosen as the programming tool primarily because it contains Control System Toolbox, interactive mode of work, immediate graphics facilities, built-in functions, the possibility of adding user-written functions, and simple programming and because of its wide availability on computing platforms. These factors make Matlab an excellent language for teaching and a powerful tool for research and practical problem solving. Matlab also allows one to create movies either by saving a number of different pictures and then playing them back or by continually erasing and

redrawing the objects on the screen. The first method is more advisable in situations in which each frame is fairly complex and cannot be redrawn rapidly. As simplified flowchart of the developed Matlab program, performing DSAs of the classical control techniques, is given in Fig.2.

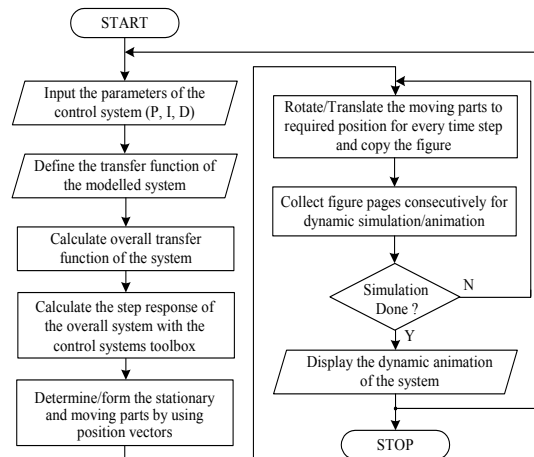


Figure 2. Simplified flowchart of the developed Matlab program performing DSAs of the classical control techniques.

According to Fig.2, after determining the transfer function of the system, the DSA can be carried out. The first step for realizing the DSA is to decide a controller type and calculate the step response of the overall system. The second step is to constitute the stationary parts of the system simply by using position vectors. The third step is to form the moving parts and animate the moving parts according to the step response. The final step is to accumulate the figure pages consecutively and display them sequentially.

3.2. Method

Computer graphics, which play an important role in simulations and animations, are based on the fundamental linear transformations. Linear transformations have a matrix-based structure. Computer graphics is a complex and diversified technology. Presently this technology is used for a large variety of purposes; e.g., computer-aided design and manufacturing, architectural rendering, advertising illustrations, animated movies, etc. Computer graphics are usually formed by rotating, translating, scaling and performing various projections on the data. These basic orientations or viewing preparations are generally performed using 3×3 or 4×4 transformation matrix operating on the data represented in homogeneous coordinates. When a sequence of transformations is required, each individual transformation matrix can be sequentially applied to the points to achieve the desired result [14, 15].

A point is represented in two dimensions by its coordinates. These two values are specified as the

elements of a 1-row, 2-column matrix: $[x \ y]$. Alternately, a point is represented by a 2-row, 1-column matrix: $[x \ y]^T$. These two representation matrixes are commonly called position vectors. In this study, a row matrix formulation of the position vector is used. If the origin of the coordinate system is variant, the homogeneous coordinate systems must be used. The homogeneous coordinates of a non-homogeneous position vector $[x \ y]$ are $[x \ y \ 1]$. A series of points, each of which is a position vector relative to some coordinate systems, is stored in a computer as a matrix. By drawing lines between the position vectors, lines, curves or figures are generated. After defining the required transformation matrix and multiplying it by the position vectors, the transformed shape can be attainable.

A number of transformations such as rotation, reflection, scaling, shearing, etc., can be realized by the general 3×3 transformation matrix. The general transformation matrix is given in Eq.1. The effect of the terms a , b , c and d in the 3×3 matrix can be identified separately. The terms b and c cause a shearing in the y and x directions respectively. The terms a and d act as scale factors. Thus, the general 3×3 matrix produces a combination of shearing and scaling. In the same manner, different combinations of the terms a , b , c and d , also, perform rotation and reflection.

$$[T] = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ m & n & 1 \end{bmatrix} \quad (1)$$

3.2.1. Translation

The pure two-dimensional translation process is given in Eq.2. The elements m and n are the translation factors in the x and y directions, respectively;

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} \quad (2)$$

where x^* and y^* are coordinates of the translated position vector.

3.2.2. Scaling

Scaling is controlled by the magnitude of the two terms on the primary diagonal of the matrix: a , d (Eq.3). If the magnitudes are equal, uniform scaling occurs about the origin; if the magnitudes are not equal, a distortion occurs. For a uniform scaling, if $a=d>1$, a uniform expansion occurs; thus the figure gets larger. If $a=d<1$, then a uniform compression occurs; thus the figure gets smaller;

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} a & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where x^* and y^* are coordinates of the scaled position vector.

3.2.3. Rotation

In general, a rotation about an arbitrary point can be accomplished by first translating the point to the origin, performing the required rotation, and then translating the result back to the original centre of rotation. Thus, rotation of the position vector $[x \ y \ 1]$ about the point p, q through an arbitrary angle can be accomplished by Eq.4;

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ -p(\cos \theta - 1) & -q(\cos \theta - 1) & 1 \\ +q \sin \theta & -p \sin \theta & 1 \end{bmatrix} \quad (4)$$

where x^* and y^* are coordinates of the rotated position vector [15].

4. EXEMPLARY APPLICATIONS

4.1. A DC Motor and Measurement Apparatus

The armature-controlled permanent-magnet DC motor is a type of motor that is commonly used for control system applications. The operation of the DC motor with armature control provides a nearly linear ratio of steady-state velocity to input voltage over a speed range that extends from zero to the maximum rated angular velocity [16,17]. Voltage control of an armature-controlled DC motor provides an innate feedback system, as shown in Fig. 3. Thus, the motor exhibits an automatic feedback compensation that tends to maintain a constant angular velocity. The overall transfer function relating angular velocity to input voltage is given in Eq.5.

$$\frac{\Omega(s)}{V_a(s)} = \frac{K_t}{L_a J s^2 + (R_a J + L_a B) s + (R_a B + K_t K_b)} \quad (5)$$

where the parameters are: the armature resistance R_a in ohms ; the armature inductance L_a in henrys ; the induced emf constant K_b in $\text{volt}/(\text{rad}/\text{sec})$; the torque constant K_t in $\text{N}\cdot\text{m}/\text{Amp}$; the moment of inertia J in $\text{kg}\cdot\text{m}^2$; the viscous friction coefficient B in $\text{N}\cdot\text{m}/(\text{rad}/\text{sec})$.

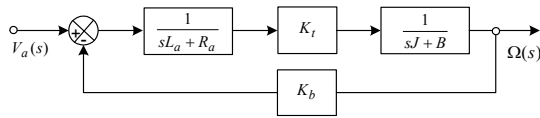


Figure 3. Block diagram of the DC motor.

Let the DC motor with the rated values of $V_t = 24V$, $I_a = 2.5A$, $n = 1200rpm$ have those parameters: $R_a = 1\Omega$, $L_a = 0.5H$, $K_t = K_b = 0.01$, $J = 0.01 \text{ kg-m}^2$, $B = 0.1 \text{ N-m/(rad/sec)}$. The open-loop step response of the uncontrolled DC motor should firstly be calculated. The step response is the time behaviour of the outputs of a general system when its inputs change from 0 to unity value (1 Volt in this example) in a very short time. Knowing the step response of a dynamical system gives information on the stability of such a system, and on its ability to reach a stationary state starting from another when required. Once a control signal is designed on the bases of test signals, the performance of the system in response to actual inputs is generally satisfactory. So, from the calculation of step response we see that the motor can only rotate at 0.1 rad/sec with an input voltage of 1 Volt, ten times smaller than our desired speed. Also, it takes the motor 3 seconds to reach its steady-state speed. Another matter is the steady-state error of 90%.

A simple DSA about the DC motor mentioned above is given in Fig.4. In the DSA both the stationary parts of the DC motor (frame, base, cooling fins, end cap and bolts) and rotary part of that (shaft) are firstly constituted with position vectors. Then, a long arrow showing the reference rotating angle and a short arrow showing the actual rotating angle are added to the figure. Thus, the physical parts of DC motor have been materialized visually, and are ready to use in the DSA. Now, suppose that we want to use a P-controller in order to improve the DC motor's performance, and see the effects of control action on the motor's performance. Let the P-controller parameter K_p be 200.

The developed Matlab program first calculates the step response of the motor with The Control System Toolbox, and then animate both the long (reference) arrow according to the reference input and the short (shaft) arrow according to the step response. The calculated step-response of the motor, given in Fig.5.a, shows that the P-controlled DC motor has better transient and steady-state performances than the uncontrolled case. Fig.4.a, the first frame of the DSA, shows the initial positions of the arrows ($t=0s$), and Fig.4.b shows the positions of those at the instant $t=0.0828s$.

The effects of control actions on rotary parts of a system as in the DC motor generally can not be illustrative and instructive due to the harmonic motion. Despite this disadvantage, the DC motor has been given here as an example because it is widely used for control system applications widely. In addition to this, the effects of any control techniques on the DC motor can be easily seen when the step response of the DC motor is applied to some other equipment; this method is more explanatory than before. For this purpose, step response of the DC motor with PI-controller is applied to the pointer of a measurement apparatus. The apparatus is planned as an ammeter. As explained before, both stationary (base, frame, scale) and moving (pointer) parts of the ammeter are constructed in the beginning, and then the pointer is rotated appropriately depending on step response of the DC motor (Fig.4.b) and a measured (steady-state) value. So, in this example, motion of the pointer shows the dynamic behaviour of the PI-controlled DC motor from a different point of view (Note that we did not use the mathematical model of the measurement apparatus). A PI controller with $K_p = 50$, $K_i = 90$ is tested for the DSA; the measured value was chosen as 3A. Fig.6.a shows the initial position of the pointer ($t=0s$), and Fig.6.b shows the position of it at the instant $t=0.319s$, which is in the overshoot-time region.

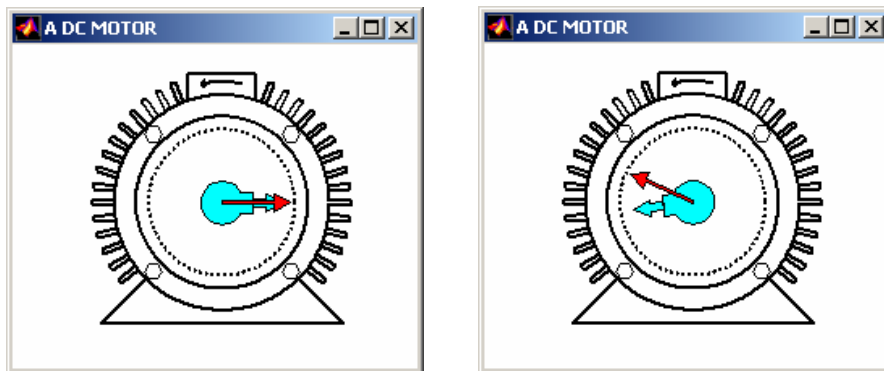


Figure 4. Instantaneous positions of the DSA of the DC motor, a) $t=0s$, b) $t=0.0828s$.

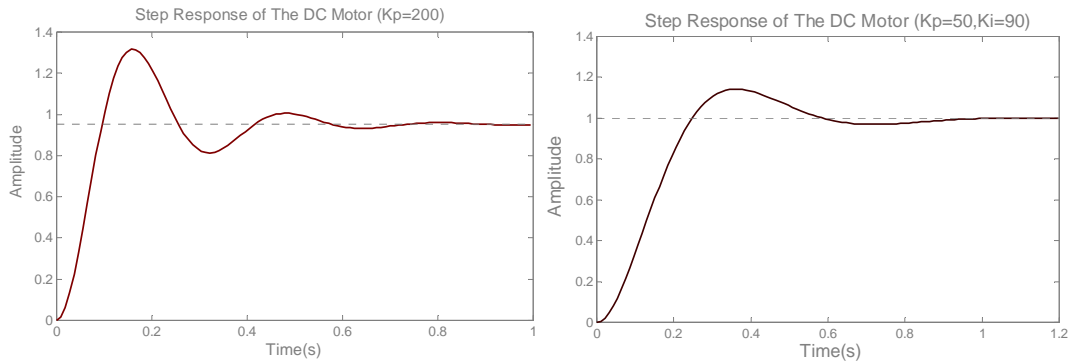


Figure 5. Step responses of the DC motor with, a) $K_p=200$, b) $K_p=50$, $K_i=90$.

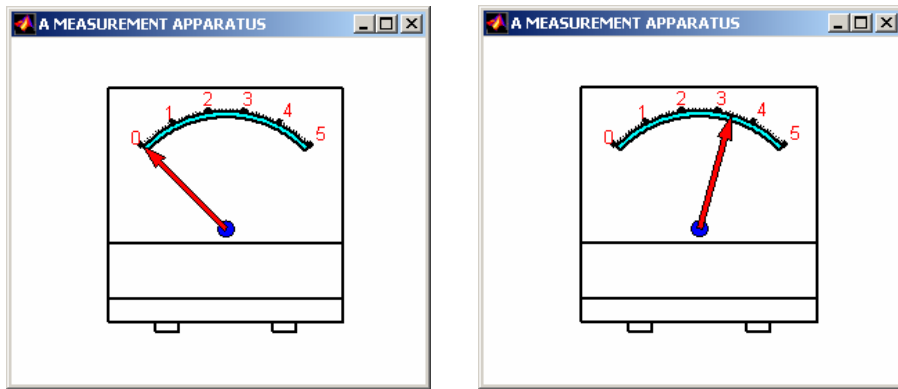


Figure 6. Instantaneous positions of the DSA of the ammeter, a) $t=0s$, b) $t=0.319s$.

4.2. A Robot Arm and A Tank Barrel

A robot is a mechanical system that must be controlled in order to accomplish a useful task. The task involves the movement of the manipulator arm, so the primary function of the robot control system is to position and orient the wrist with a specified speed and precision. Robots are employed in a wide assortment of applications in industry [17,18]. Although a robot manipulator is mechanically far more complex than the DC motor which we have analyzed above, it exhibits many of the same operating features. The resulting motions of the robot arm behave in a manner similar to the performance of a system described by a second-order linear differential equation with constant coefficient. In this section, we will deal with making a DSA on a robot manipulator having a single joint. The main components of the joint include a DC servomotor, a gear train, a tachometer, a position encoder, and a

controller (Fig.7). A simplified block-diagram of the robot arm is given in Fig.8. Briefly, the developed Matlab program firstly calculates step response of the robot arm (Fig.9.a) by using the desired control technique and the transfer function of the robot arm, and takes the rotation angle as in degree from the user, and forms the stationary and moving parts, and then translates the moving parts to the position where they should be. With this method the realized DSA having both a PID controller ($K_p = 120$, $K_i = 200$, $K_d = 10$) and a rotation angle of -45° is given in Fig.10. The preferential rotation angle is restricted with $\mp 45^\circ$ due to hardware limitations; for this reason, the maximum overshoot must be also taken into consideration. From this point of view, the robot-arm example is a very suitable exemplary application for testing all of the control techniques.

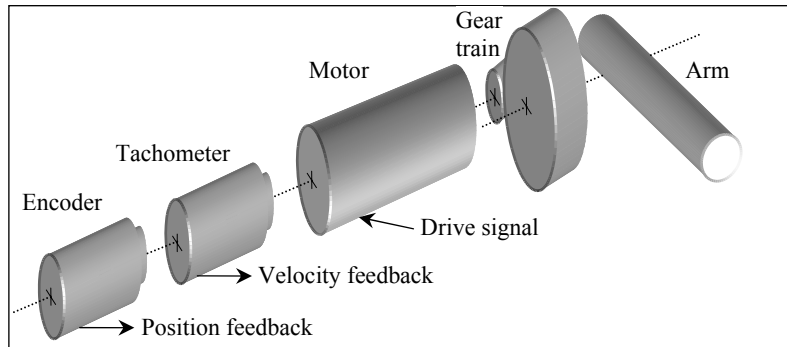


Figure 7. Schematic diagram of the robot arm having a single joint.

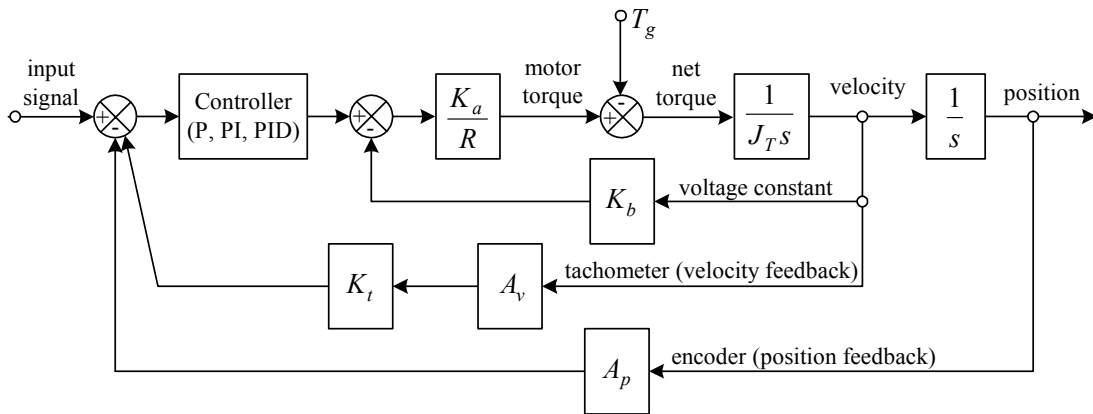


Figure 8. Block diagram of the robot arm.

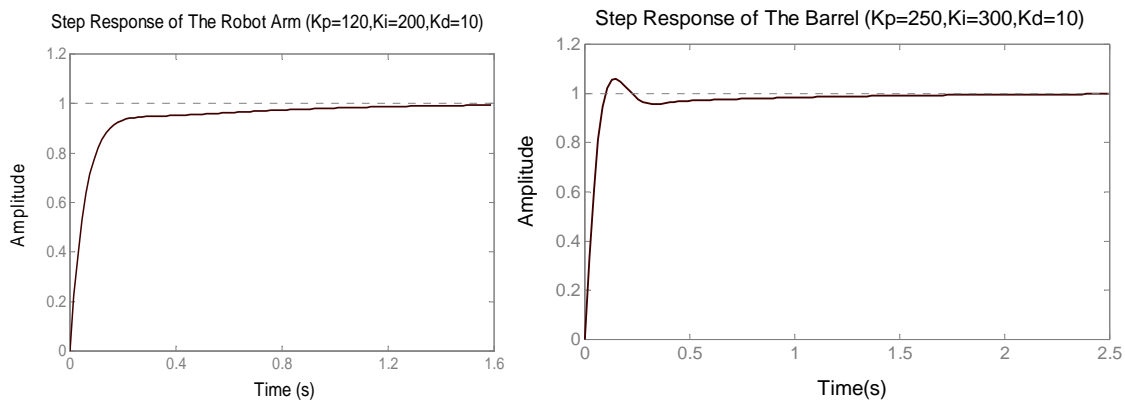


Figure 9. Step responses of the robot arm with a PID controller a) $K_p = 120, K_i = 200, K_d = 10$, b) $K_p = 250, K_i = 300, K_d = 10$.

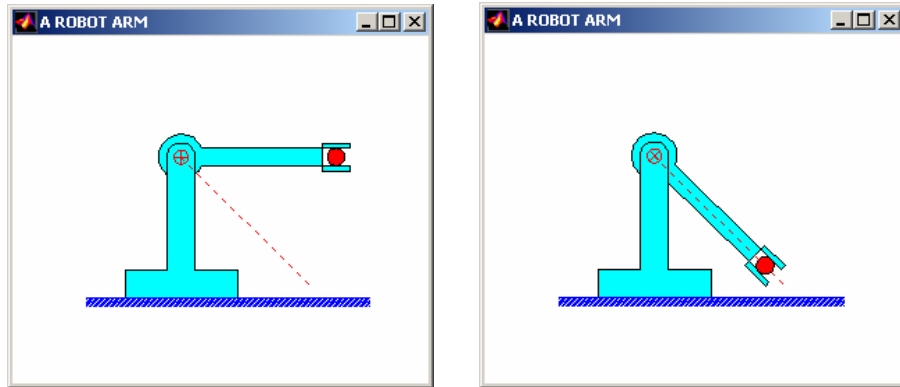


Figure 10. Instantaneous positions of the DSA of the robot arm, a) $t=0s$, b) $t=1.5935s$.

The barrel of a gun or firearm is a metal tube through which a controlled explosion is released in order to propel a projectile or a bullet out of the end at great speed. There are generally two basic categories of firearms: small arms and artillery pieces. Small arms are generally small, very portable firearms and are aimed visually at their targets using sights. Artillery pieces, such as a field gun, a tank gun, an anti-tank gun, or a naval gun, are much larger, mounted on a movable carriage, and possibly weighing many tons. Pointing the barrel at a target as soon as possible is a very important detail in artillery pieces because they have huge bodies. This can be realized by either a human or an automatic tracker. Naturally, an automatic tracker uses the automatic control systems to achieve good shots.

A tank barrel controlled with a classical control technique is analogous a single-jointed robot arm with

the exception of hardware limitations and rotation angle. Therefore, without knowing the mathematical model or block diagram of the barrel system, the block diagram of the robot arm given in Fig.8 is used easily for realizing a DSA on a tank barrel. In this exemplary DSA, a PID controller ($K_p = 250$, $K_i = 300$, $K_d = 10$) and a rotating angle of 45° are tested. Under these conditions, the step response of the robot arm produced by the control system toolbox is given in Fig.9.b. After the values of the step response are normalized with the desired rotation angle, the moving part (the barrel only) is treated appropriately. Also, instantaneous positions of the DSA of the tank barrel are given in Fig.11. The dashed line in Fig.11 denotes the desired rotating angle which should be entered by the user.

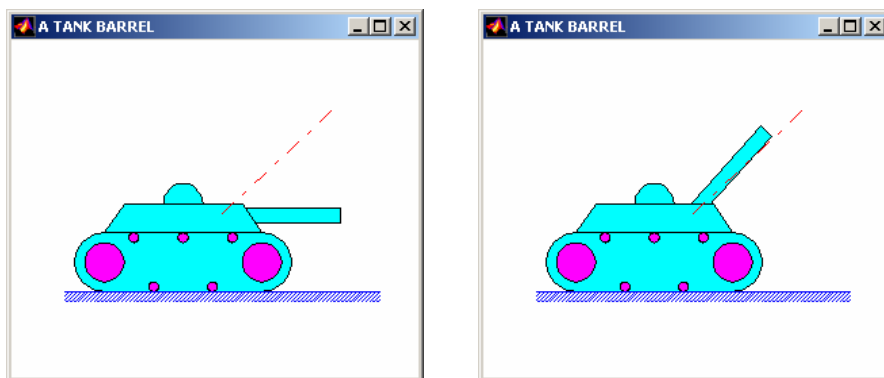


Figure 11. Instantaneous positions of the DSA of the tank barrel, a) $t=0s$, b) $t=0.1492s$.

4. CONCLUSIONS

Computer-based teaching and learning techniques, using linear transformation and computer graphics, are greatly enhancing the learning process. Computer graphics used for educational purposes are requiring more features such as flexibility, reliability, visuality and user-friendliness. In this study, a different approach to the DSAs has been introduced. This approach is recommended to some educators who want to develop

their own animation package program in order to improve the effectiveness of utilizing animations and simulations in their control courses. With the aid of this method an educator can demonstrate the effects of control actions on system performance visually, which will enhance interest, understanding and learning. In this approach, Matlab package program is the main equipment to deal with linear transformation matrix because of it is matrix-based software. Equipment of secondary importance is linear transformation matrixes

and position vectors. While linear transformation matrixes performs transformations such as scaling, translating, rotating, etc.; position vectors constitute the physical part of the system in 2D or 3D-space. After performing a step response of the system, the animation is realized by carrying the moving part to the required position at the right time. Meanwhile, the stationary parts are not taken into consideration. In this paper, how to make DSAs in control of electromechanical systems is explained with exemplary applications. Thus, one can constitute special-purposed animations by using the method introduced. Linear transformation matrixes can also be used for three-dimensional DSAs; for this purpose, new algorithms and package programs may be developed in further studies.

REFERENCES

- [1] Neelamkavil, F., O'Tuathail, E. "Systems for engineering animation with automatic motion control", *Computer-Aided Engineering Journal*, vol:8/3, 103-115, 1991.
- [2] Huang, N., Cheok, T.G., Settle, T., "Real-time simulation and animation of suspension control system using TI TMS320C30 DSP", *Simulation*, vol:61/6, 405-416, 1993.
- [3] Dimirovski, G.M., Radojicic, P.C., and etc., "Modeling, control and animated simulation of complex processes in robotized FMS", *Proc. of the 20th Int. Conference on Industrial Electronics, Control and Instrumentation*, part:2/3, 1141-1146, 1994.
- [4] Watkins, J., Piper, G., and etc, "Computer animation: A visualization tool for dynamic system simulations", *Proc. of the 1997 ASEE Annual Conference*, 1997.
- [5] Villareal, S., Wynn, C., Eastwood, D., Zoghi, B., "Design, development and evolution of Web-based materials featuring computer-animated simulations", *Proc. of 1998 28th Annual Frontiers in Education Conference*, FIE, part:2/3, 588-593, 1998.
- [6] Tilbury, D., Messner, W., "Control education on the WWW: Using Matlab for Control design, simulation and visualization", *Proc. of the 1998 ASME Int. Mechanical Eng. Congress and Exposition*, 65-70, 1998.
- [7] Ravn, O., Larsen, T., Andersen, N., "Simulation and animation in Simulink and VRML", *Proc. of the 1999 IEEE Int. Symposium on Computer Aided Control System Design*, 120-125, 1999.
- [8] Rios, C., Lim, C.I., and etc, "Multivariable analysis and control of a cart-pendulum-seesaw system using an animation tool", *Proc. of the IEEE Conference on Decision and Control*, 825-860, 1999.
- [9] Katsuhiko, O., "*Modern Control Engineering, 3rd ed.*", Prentice-Hall International Editions, Upper Saddle River, New Jersey, 1997.
- [10] Chi-Tsong, C., "*Control System Design, Conventional, Algebraic and Optimal Methods*", Pond Woods Press, NY, 1987.
- [11] Lewis, P.H., Yang, C., "*Basic Control Systems Engineering*", Prentice Hall Inc., Upper Saddle River, New Jersey, 1997.
- [12] Katsuhiko, O., "*Designing Linear Control System with Matlab*", Prentice-Hall, Englewood Cliffs, New Jersey, 1994.
- [13] Biran, A., Breiner, M., "*Matlab for Engineers*", Addison-Wesley Pub. Comp., 1995.
- [14] Rogers, D.F., "*Procedural elements for computer Graphics*", McGraw-Hill Pub., New York, 1985.
- [15] Rogers, D.F., Adams, J.A., "*Mathematical elements for computer graphics, International 2nd Edition*", McGraw-Hill Pub. Comp., 1990.
- [16] Krishnan, R., "*Electric Motor Drives: Modeling, Analysis, and Control*", Prentice Hall, Inc., 2001.
- [17] Groover, M.P., Weiss, M., Nagel, R.N., Odrey, N.G., "*Industrial Robotics: Technology, Programming and Applications*", McGraw-Hill, Inc., 1986.
- [18] Dorf, R.C., "*Robotics and Automated Manufacturing*", Reston, VA, 1983.