1. Andrija DEVIĆ, 1. J. VIDAKOVIĆ, 1. N. ŽIVKOVIĆ, 2. M. LAZAREVIĆ

SYNTHESIS OF THE CONTROL UNIT OF THE DESKTOP ROBOT ARM ACTUATED BY STEPPER MOTORS

1. Lola institute, Kneza Viseslava 70a, 11030 Belgrade, SERBIA
2. Faculty of Mechanical Engineering, The University of Belgrade, Kraljice Marije 16, 11120 Belgrade, SERBIA

Abstract: Small–size (desktop) robot arms are being increasingly used in research and education, where low–cost implementation prevails over the need for high dynamic performance. This paper presents a design of a control unit for a 6DoF desktop robot arm with cylindrical joints actuated with stepper motors. Feedforward control, often chosen as a control strategy for stepper motors due to the simplicity of control algorithms, as well as the cost–effectiveness of the solution, is selected as the control method. The applicative software with GUI developed in Matlab is presented. A trajectory planner based on the solution of the inverse kinematics problem and the user–selected joint velocity profile is implemented. In order to avoid step skipping occurrence, the torque in joints necessary to produce the programmed joints motion have to be achievable by the installed stepper motors. As a part of applicative software, the check on the feasibility of the programmed robot trajectories by numerical simulation of solution of inverse dynamics problem in Simscape Multibody is performed. The microcontroller is used as an interface device that controls the stepper motor drivers. The applicative software communicates with the microcontroller using the TCP protocol.

Keywords: robot arm, control, stepper motors

1. INTRODUCTION

In recent technology trends, robot development has attracted more interest due to the introduction of Industry 4.0 which has highlighted automation and robotics as part of the potential fields for raising the quality of production, automotive and human service industries [1]. Robot arms are used for many industrial applications, such as handling, painting, assembling, welding etc. [2]. Due to mentioned trends in robot development, there is a demand on the market for small–size (desktop) robots in research and education, etc. where low–cost implementation prevails over the need for high dynamic performance [3–4].

One of the main components in developing a robot arm includes control unit design. In robotics, stepper motors are used as actuators often for low or predictable force/torque requirements, and have proven to be an excellent choice for low–cost and small–size robot manipulators due to the ease of controlling their position without feedback [3].

In this paper, the design of a control unit for a 6DoF desktop robot arm actuated with stepper motors is presented. Feedforward control is selected as a control method as a simple and cost–effective control solution. To achieve successful trajectory tracking by feedforward control, the step skipping occurrence has to be prevented throughout the robot operation. This implies that the torques in joints necessary to produce the desired joint motion have to be achievable by the installed motors during entire robot operation, i.e. robot must operate under strictly controlled conditions. Herein, this is achieved by verification of given robot trajectories based on the implemented robot dynamic model. The applicative software with GUI developed in Matlab is presented. The application examines the feasibility of the programmed robot trajectories by numerically simulating the solution of the inverse dynamic problem in Simscape Multibody. A microcontroller is used as an interface device and it communicates with the application software via TCP protocol.

This paper is organised as follows. In Section 2, a description of the main features of the 6DoF robot arm is given in short. 3. In Section 3, the control system design for desktop robot actuated by stepper motors is presented, with special emphasis on 1) the designed applicative software with GUI, 2) the implementation of the solution of robot inverse dynamics problem, and 3) microcontroller interface implementation. Concluding remarks are given in Section 4.

2. FEATURES OF 6DOF DESKTOP ROBOT ARM

Development of the robot design starts with a computer–aided design (CAD) model, and simulation of the system motion. The development of the kinematic and dynamic model follows, and is completed by designing a control unit for the robot [5].

At the very beginning of the design of the 6DoF low–cost desktop robot arm with cylindrical joints, it was necessary to adopt the kinematic structure so that the obtained inverse kinematics equations are as simple as possible, as well as to enable construction with relatively simple production of the parts. The design of the 3D model of the mechanical structure is performed in SolidWorks. Stepper motors [6] are selected due to the cost–effectiveness and simplicity of the control unit development. Toothed belts were used herein as a low–cost and simple solution to transmit power from the stepper motor to the joints.

The segments of the robotic arm are made using additive manufacturing which enables relatively simple and cheap production. PET–G plastics that has an enviable load resistance/price ratio is used to fabricate the robot arm on the 3D printer. The robot mass is 3.9kg, while the maximum payload is 1kg. In Figure1, the constructed 6DoF desktop robot arm named AD100 is presented.

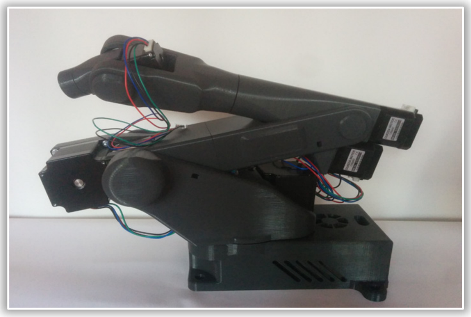


Figure 1. 6DoF desktop robot arm AD100 [7]

3. CONTROL SYSTEM DESIGN FOR DESKTOP ROBOT ACTUATED BY STEPPER MOTORS

Within the development of the control unit, the basic guidelines were the simplicity of development, reliability of the control unit, and cost–effectiveness. In Figure2, the block diagram of the control unit of the robot arm with six degrees of freedom and cylindrical joints actuated by stepper motors is shown.

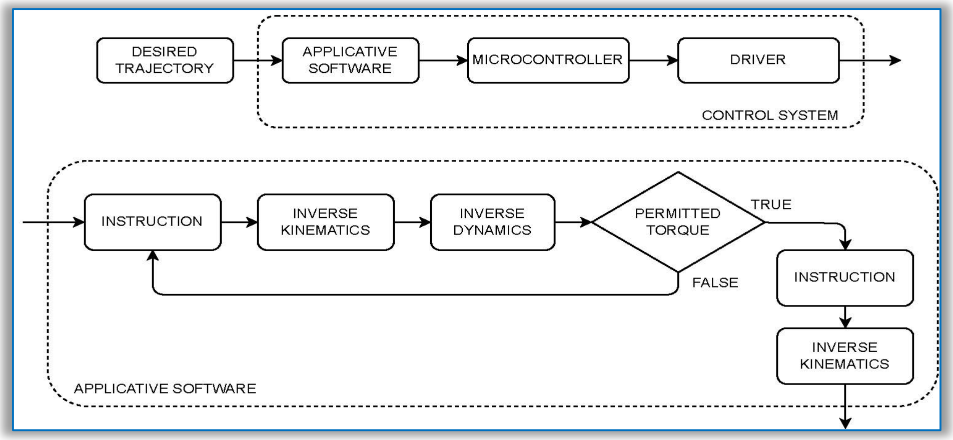


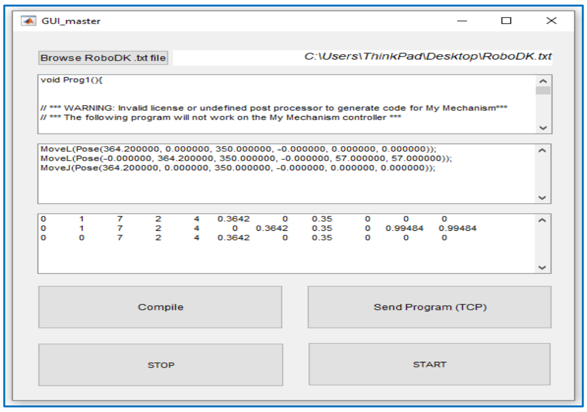
Figure 2. The block diagram of the control unit of the 6DoF robot arm AD100

The applicative software with GUI is developed in Matlab. Programming of the robot is made possible by setting the desired robot trajectory via the internal and/or external coordinates. Programming of the desktop robotic arm is done in offline mode. 3D visualisation of robot motion for the purpose of verifying the given programs is implemented. A separate trajectory planner was developed. Torques required to be produced by motors necessary to achieve the desired robot motion are obtained by numerical solution of robot inverse dynamics problem in Simscape Multibody [8]. TCP standard is used for communication between the Matlab GUI application software and the microcontroller that drives the stepper motors.

Development of Matlab GUI

The applicative software with GUI (Figure 3) designed in Matlab can be run on Windows, Mac, and Linux operating systems. In this study, the software package RoboDK [9] was used for users to generate robot path programs by offline programming methods, and also to produce the robot path for visualisation purposes.

A separate trajectory planner was developed for generating trajectories with an implemented solution of the inverse kinematics problem. Joint robot trajectories (path, velocity, acceleration) are created based on input/output position and/or orientation data, input user choice for generation of the joint velocity profile–linear, trapezoidal, or polynomial, and other parameters. Successive joint positions sent to the microcontroller are obtained by interpolation with



C:\Users\ThinkPad\Desktop\RoboDK.txt

```
void Prog1(){


// *** WARNING: Invalid license or undefined post processor to generate code for My Mechanism***
// *** The following program will not work on the My Mechanism controller ***
```

```
MoveL(Pose(364.200000, 0.000000, 350.000000, -0.000000, 0.000000, 0.000000));
MoveL(Pose(-0.000000, 364.200000, 350.000000, -0.000000, 57.000000, 57.000000));
MoveJ(Pose(364.200000, 0.000000, 350.000000, -0.000000, 0.000000, 0.000000));
```

```
0     1     7     2     4    0.3642      0     0.35     0      0        0
0     1     7     2     4       0     0.3642   0.35     0    0.99484  0.99484
0     0     7     2     4    0.3642      0     0.35     0      0        0
```

Figure 3. Matlab GUI

an interpolation period of 50 ms. That period of time has been practically proven to be adequate. Clicking the Compile button starts the part of the application that calculates the desired motors moments for input joint trajectories, which is performed by the simulation diagram in Matlab Simulink. After verifying the feasibility of the programmed trajectory based on the possibilities of installed motors, a Matlab script is executed, which creates packets of information that are prepared to be sent to the microcontroller.

In order to perform the desired movement of the robot, by pressing the Send Program (TCP) button, the application sends previously formed information packets to the microcontroller using the TCP protocol, and finally, by pressing the Start button, the microcontroller actuates the motors and executes the movement of the robot according to the trajectory set by the program.

Implementation of solution of robot inverse dynamics problem

In order for the stepper motors to be able to achieve the programmed positions and to avoid “skipping steps” (ie. to maintain synchronization) throughout robot operation, it was necessary to examine whether the required motor torques exceed their capabilities for the obtained joints’ motions throughout desired robot operations. A robot dynamic model is time variable, highly non–linear and characterized by coupling effects among the robot joints [10]. The solution of the robot inverse dynamic problem, ie. the calculation of the required robot actuators’ torques/forces from a specification of the robot’s trajectory can be obtained analytically, or by using ready–made modeler/simulator software solutions based on 3D software robot models. The latter avoids solving complicated mathematical expressions [10].

Here, torques required to be produced by motors necessary to achieve the desired robot motion are obtained by numerical solution of robot inverse dynamics problem in Simscape Multibody. The robot model is imported from SolidWorks. In SolidWorks, by selecting the Simscape Multibody Link option from the Tools drop–down menu, a file with the extension .hml is created, which contains all the necessary parameters for creating a dynamic model in Simscape, mass, inertia, center of mass, for each part of the assembly separately, as well as the type of joints by which they are connected.

The simulation model for 6DoF robot AD100 in Simscape Multibody is created and presented in Figure 4.

The torque/speed curve is obtained from motor manufacturer documentation [6], and is used for comparison of maximum permissible load torque without the motor losing the step.

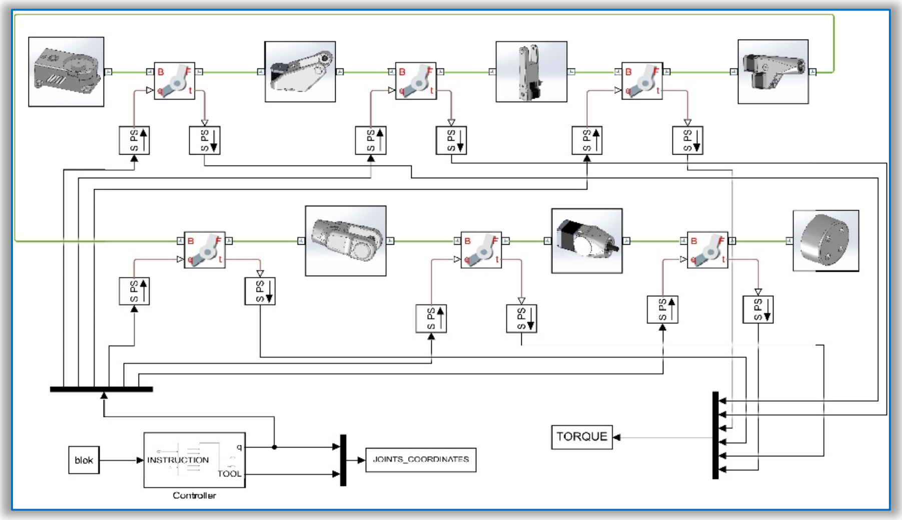


Figure 4. Simulation diagram for the solution of inverse dynamics problem in Simulink Simscape Multibody

Microcontroller interface implementation

The application software communicates with the microcontroller using the TCP protocol. The microcontroller program is written in the mikroC compiler. Microchip PIC32MX534F064H, 32–bit microcontroller [11], controls the motors using a driver–type DRV8825 [12]. Via the SPI bus, the microcontroller communicates with the ENC28J60 gateway [13], which enables communication via ethernet with the application software. The electrical schematic contains six more A3144 [14] hall sensors for detecting the end positions of the robot's joints.

Initial loading (before moving robot) of all packages into the memory of the microcontroller ensures real–time movement control, i.e. avoids traffic jams in communication. The duration of execution of one package is 50ms. An algorithm has been implemented that calculates the moments of time according to the data from the package. Those moments of time are the moments in which it is necessary to change the state of the pins on the microcontroller, for all six motors. The core of the algorithm runs in an interrupt routine that allows the robot to move in real time. Based on this algorithm, the distribution of the signal pulses sent to the motor drivers is determined.

4. CONCLUSION

In this paper, the design of a control unit for a 6DoF desktop robot arm with cylindrical joints actuated with stepper motors is presented. Feedforward control is selected as the control method for stepper motors due to the simplicity of control algorithms and cost–effectiveness. The applicative software with GUI developed in Matlab is presented. A trajectory planner based on the solution of the inverse kinematics problem and user–selected joint velocity profile is implemented. In order to feedforward control to be able to achieve successful trajectory tracking, the check on the torque in joints necessary to produce the desired joints motion is performed by comparison to maximum achievable moments by the installed motors for the desired motor velocity. This check is implemented as a part of applicative software by numerical simulation of solution of inverse dynamics problem in Simscape Multibody based on SolidWorks 3D robot model. A microcontroller is used as a cost–effective interface device, and the application software communicates with the microcontroller using the TCP protocol. It can be concluded that the basic guidelines appointed on the development of the control unit, namely the simplicity of development, reliability of the control unit, and cost–effectiveness have been successfully followed.

Acknowledgement

This research has been supported by the research grants of the Serbian Ministry of Science, Technological Development and Innovations, grant No. 451–03–68/2023–14/200066 and 451–03–47/2023–01/ 200105 from 03.02.2023.

Note: This paper was presented at DEMI 2023 – 16th International Conference on Accomplishments in Mechanical and Industrial Engineering, organized by Faculty of Mechanical Engineering, University of Banja Luka (BOSNIA & HERZEGOVINA), co–organized with the Faculty of Mechanical Engineering University of Niš (SERBIA), Faculty of Mechanical Engineering University of Podgorica (MONTENEGRO), Faculty of Engineering Hunedoara, University Politehnica Timişoara (ROMANIA) and Reykjavik University (ICELAND), in Banja Luka (BOSNIA & HERZEGOVINA), in 01–02 June, 2023

References

[1] Zainudin, W. M. L. W., Shauri, R. L. A., Roslan, M. I., Rosli, M. A., & Ariffin, M. F. M. New Interface using Beaglebone Black for 4–DOF Robot Arm System. IEEE Symposium on Industrial Electronics & Applications (ISIEA), July 2020, pp 1–6
[2] Vidaković, J. Z., Kvrgić, V. M., Lazarević, M. P., Dimić, Z. Z., & Mitrović, S. M. (2017). Procedure for definition of end–effector orientation in planar surfaces robot applications. Tehnika
[3] Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). Robotics: Modelling, Planning and Control. Springer, pp 191–231
[4] Sobaszek, Ł., Gola, A., & Varga, J. (2016). Virtual designing of robotic workstations. Applied Mechanics and Materials, vol. 844, pp 31–37
[5] Dede, M. I. C., & Tosunoglu, S. (2010). Virtual rapid robot prototyping for robot controllers. International Journal of Design Engineering 3(3):276–288
[6] LAM Technologies electronic equipment, NEMA23 Stepper Motor 1.1Nm. From: https://www.lamtechnologies.com/Product.aspx?lng=EN&idp=M1233032 accessed on: February 25, 2023.
[7] Devic, A. (2020). Projektovanje i upravljanje jednog robotskog sistema sa šest stepeni slobode. (In Serbian). MSc thesis. Faculty of Mechanical Engineering, University of Belgrade.
[8] MathWorks. Model and simulate multidomain physical systems. From: https://www.mathworks.com/products/simscape.html, accessed on: February 9, 2023.
[9] RoboDK. Simulate Robot Applications. From: https://robodk.com, accessed on: February 9, 2022.
[10] Vidakovic, J., Devic, A., Zivkovic, N., Kvrgic, V., & Stepanic, P. Practical Approaches for Robot Dynamic Model Implementation for Control and Simulation. Experimental Research and Numerical Simulation in Applied Sciences CNNTech, 2022, Zlatibor, pp 147–163
[11] Microchip. PIC32MX5XX/6XX/7XX Family Data Sheet. From: http://ww1.microchip.com/downloads/en/devicedoc/61156g.pdf, accessed on: February 9, 2023.
[12] TexasInstruments. DRV8825 Stepper Motor Controller IC. From: https://www.ti.com/lit/ds/symlink/drv8825.pdf, accessed on: February 9, 2023.
[13] Microchip. Stand–Alone Ethernet Controller with SPI Interface. From: http://ww1.microchip.com/downloads/en/DeviceDoc/39662e.pdf, accessed on: February 9, 2023.
[14] Allegro MicroSystems. Inc, Sensitive hall–effect switches for high–temperature operation. From: https://www.mpja.com/download/a3144eul.pdf, accessed on: February 9, 2023

ANNALS OF FACULTY ENGINEERING HUNEDOARA
ANNALS of Faculty Engineering Hunedoara
International Journal of Engineering
INTERNATIONAL JOURNAL OF ENGINEERING
ISSN: 1584-2665 (PRINTED, ONLINE)
ISSN: 1584-2673 (CD-ROM, ONLINE)

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN–L 1584 – 2665

5, Revolutiei, 331128, Hunedoara, ROMANIA
http://annals.fih.upt.ro