FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 13, N° 3, 2014, pp. 159 - 168

DESIGN AND IMPLEMENTATION OF A DIGITAL POSITIONING SYSTEM WITH CONTROLLED JERK*

UDC (681.523.4+681.5.01):621.316.729

Mihajlo J. Stojčić

University of Banja Luka, Faculty of Mechanical Engineering, Republic of Srpskam
Bosnia and Herzegovina

Abstract. *In this paper design and implementation of a digital tracking control system is considered. This system provides tracking a trajectory where the jerk is changed to a pre-determined sine manner. As a control system there is used the combination of a feedforward and feedback controller. An electromechanical system that is formed from an actuating device and a load is used as an object. To advance certain changes of the jerk and the given maximum values of acceleration, velocity and displacement, the algorithm of the trajectory planning is also proposed. The designed control system is experimentally realized, where a DC motor with a load is used as the object. To measure the position and other parameters of movement, a two-channel optical encoder by appropriate resolution was used. The control software is implemented using the Arduino microcontroller's platform, where the control signals are realized as a PWM signals. The experimental results confirm the proposed theory.*

Key words: *design and implementation, tracking trajectory, positioning system, controlled jerk, microcontroller, Arduino Uno*

1. INTRODUCTION

The problem of tracking a given trajectory occurs in most of the positioning systems. The main task of any such system is to achieve adequate movement between two arbitrary points, from point A to point B, where the system at the initial and the final points is in the idle state. In these cases, the quality of tracking depends on the manner of changing velocity. In this paper we assume that the time of acceleration and deceleration are the same, which means that the trajectory is symmetric with respect to the velocity. In order to design and implement such a system it is necessary to solve several different problems such as:

Received November 15, 2014
Corresponding author: Mihajlo J. Stojčić
Faculty of Mechanical Engineering, Vojvode Stepe Stepanovića 71, 78000 Banja Luka, Republic of Srpska
E-mail: stojcicmihajlo@gmail.com
* **Acknowledgement**: This paper was supported by BANOROB project which is funded by the Norwegian government. The author is very grateful for their support.

- trajectory planning: determining allowable trajectory and all parameters of motion (jerk, acceleration, velocity, ...) for all degrees of freedom and for each actuating device separately,
- controller design: design feedback and/or feed-forward controller that ensures the achievement of the desired trajectory for each actuating devices separately, although disturbances (internal and/or external) are acting on the object and there exists an unmodeled dynamics of the object, and
- other problems, such as: implementation, diagnostic, internal checks, communications, etc.

The solutions of the above problems usually are reduced to one actuating unit, assuming that the solutions for the rest units are the same.

The motion of the object (plant) between two arbitrary points can typically be divided into three phases: acceleration, motion with constant speed, and deceleration. Traditionally, a trapezoidal (or triangular) speed profile has been mainly used. This means that acceleration of the object (for a time t_a) is constant until it reaches the maximum velocity, then it keeps this velocity (for a time t_v), and it decelerates by a constant deceleration (also, for a time t_a), so that the total time of the movement t_u is $t_u = 2t_a + t_v$. One of the main problems with the trapezoidal speed profile is large changes of the jerk, and consequently large inertial forces. Furthermore, that can induce large vibrations of mechanical parts of the system which leads to a large stationary error and too long a settling time, which can often be unacceptable for the overall system. But, there exist several different approaches to improve the performance of these systems, that are described in [2, 11] and which can roughly be split as: *i*) trajectory smoothing or shaping (different example of this approach can be found in papers [2, 3, 7, 9]), *ii*) feedforward control based on plant inversion (see the papers [2-4]) and *iii*) feedback control optimization (see [1, 9, 10]).

In this paper are given the design and implementation of a digital control system (controller) that are very similar to the design described in [12, 13] and whose implementation is similar to the system in [13]. Herein, the proposed control system is the combination of a feedback and feedforward controller. For the implementation of this control system an Arduino platform is used, whose hardware is based on Atmel MEGA 328 microcontroller, the software is based on C++ programming language and the environment (IDE) is based on Java platform. The mini DC motor with load and two-channel optical encoder is used as an object and a sensor, respectively.

And at the end of this chapter, a few words about the structure of the paper. The second chapter is related to the trajectory planning, which includes: the mathematical modelling of the object, determining the manner of the object motion parameters changes and an algorithm of the trajectory planning. In the third chapter there is the design of the digital control system that consists from the feedback and feedforward controllers. And, in the fourth chapter this control system is implemented. The fifth and sixth chapters include conclusions and references, respectively.

2. Trajectory Planning

A. Mathematical model of the object

We observe an electromechanical positioning system, which consists of a permanent magnet direct current (DC) motor with a load. At the initial time $t_o = 0$, we assume that the angular jerk $j(0)$, angular acceleration $\varepsilon(0)$, angular velocity $\omega(0)$ and angular displacement $\theta(0)$ of the motor are zero. Also, all the frictions (e.g. Coulomb friction) except viscous friction are neglected and they are hereinafter modeled as disturbances $w(t)$. In this case, the transfer function $G_{\theta u}(s) = P(s)$ from the applied voltage $u(t)$ to the angular position $\theta(t)$ and the transfer function $G_{\omega u}(s) = P_v(s)$ from $u(t)$ to the angular velocity $\omega(t)$ are given as

$$P(s)=\frac{\Theta(s)}{U(s)}=\frac{K}{s(\tau s+1)},\quad P_v(s)=\frac{\Omega(s)}{U(s)}=\frac{K}{\tau s+1},\tag{1}$$

where K and τ are the static gain and the time constant respectively. They are given as $K \approx k_t / (R_a b_m)$ and $\tau \approx J_m / (R_a b_m)$, where k_t, J_m, b_m and R_a are: the torque constant, moment of inertia of all rotating masses (rotor and load), damping coefficient and resistance of the armature winding, respectively, the known parameters of the motor. In doing so, it is assumed that the inductance of the armature winding L_a is negligibly small compared to the other sizes.

B. Smoothing trajectory

In order to get little changes of acceleration (trajectory smoothing) we assume the sinusoidal change of the jerk as:

$$j(t)=\begin{cases} J\sin\frac{2\pi}{T}t, & t\in[0,T] \\ 0, & t\in[T,t_1] \\ -J\sin\frac{2\pi}{T}(t-t_1), & t\in[t_1,t_u] \end{cases},\tag{2}$$

where: $T = t_a$, $t_1 = T + t_v$ and J are: the acceleration time, the time of the start deceleration and the maximum value of the jerk, respectively. According to the jerk from (2) and using: $\varepsilon(t) = \int j(t)dt + C_\varepsilon$, $\omega(t) = \int\varepsilon(t)dt + C_\omega$ and $\theta(t) = \int\omega(t)dt + C_\theta$, we (for each time interval) obtain changes of the acceleration, velocity and movement respectively, see [11], whose changes are showed in the Fig. 1. The integration constants C_ε, C_ω and C_θ are obtained from the initial conditions for each time interval separately. So, at initial time $t = 0$ the values $\varepsilon(0)$, $\omega(0)$ and $\theta(0)$ are the zero, and at the rest time intervals, the value of each parameter at the end interval is the same as the value at the start of the next time interval.

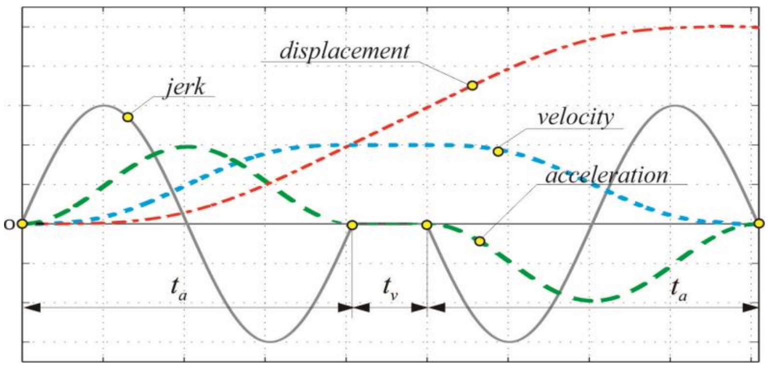

Fig. 1 Changes of jerk, acceleration,velocity and displacement at the time interval $[0,t_u]$

Over the time interval $[0,T]$ their maximum values are given ([11]) as:

$$\mathrm{E} = \varepsilon\left(\tfrac{T}{2}\right) = \frac{JT}{\pi},\ \Omega = \omega(T) = \frac{JT^2}{2\pi},\ \theta(T) = \frac{JT^3}{4\pi}\ . \qquad (3)$$

From a practical viewpoint, it is the best to give the maximum values of angular jerk J, angular acceleration E, angular velocity Ω and the total angular displacement Θ at the beginning of the trajectory planning. These values depend on the possibility of the actuator (force or torque), on the application of the positioning system as well as the possibility of the control system. Then it is necessary to determine the shortest time $T = t_a$ and t_v so that the given limitations are not exceeded. In this sense, we propose an algorithm that follows.

C. Trajectory planning

The shortest time within which motion can be performed is calculated from (3) as:

$$\Theta = 2\times\theta(T) = \frac{JT^3}{2\pi} \Rightarrow T = \sqrt[3]{\tfrac{2\pi\Theta}{J}}\ . \qquad (4)$$

Using this time, given the jerk J and (3) we can calculate the maximum value of the angular acceleration $\varepsilon_{\max}$ as, $\varepsilon_{\max} = JT / \pi$. Now we can test whether $\varepsilon_{\max}$ exceeds the limit value of the acceleration E. If $\varepsilon_{\max} \le \mathrm{E}$ we continue, but if $\varepsilon_{\max} > \mathrm{E}$ we recalculate T with respect E as, $T = \mathrm{E}\pi / J$. In the similar way we test whether the velocity bound is satisfied. The maximal angular velocity, from (3), is $\omega_{\max} = JT^2 / 2\pi$. Test $\omega_{\max} \le \Omega$. If this is true we continue, but if it is false we have to recalculate T with respect Ω as $T = \sqrt{2\Omega\pi / J}$. And finally we determine the time t_v as $t_v = (\Theta - \bar{\omega}T)/\bar{\omega}$, where angular velocity $\bar{\omega}$ is $\bar{\omega} = \min(\omega_{\max}, \Omega)$. Now, the total time of the movement t_u is given as $t_u = 2T + t_v$. The flowchart of the above algorithm is given in Fig. 2.

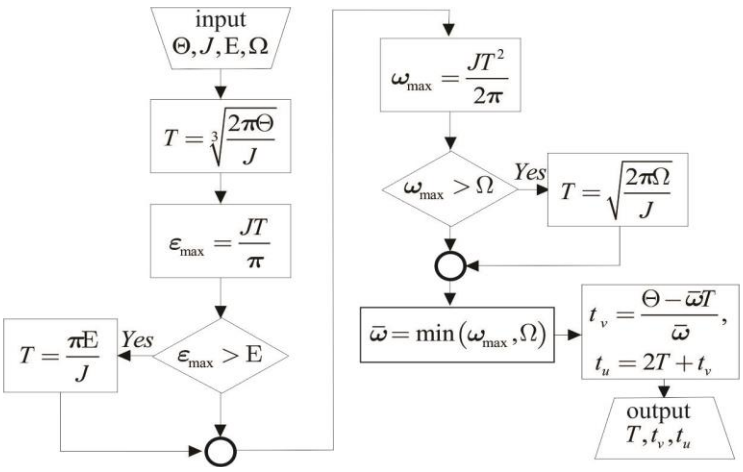

Fig. 2 Flowchart of the trajectory planning algorithm

3. DESIGN OF THE CONTROL SYSTEM

Configuration of the overall system is given in Fig. 3. The combination of the feedback $B(s)$ and feedforward $F(s)$ controller is used as a control system. In this figure with $r(s)$, $y(s)$ and $e(s)$ are denoted reference, output and output error respectively and with $W(s)$ is denoted the transfer function from disturbance $w(s)$ to the output $y(s)$.

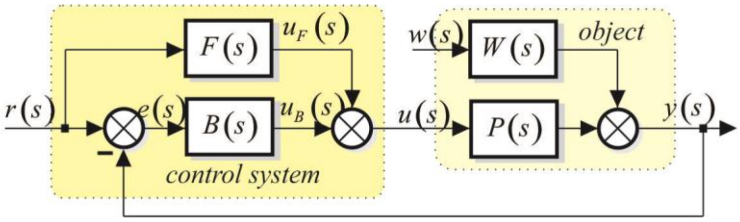

Fig. 3 Configuration of the overall system

The output $y(s)$ of the overall system is given as

$$y(s) = \frac{P(s)F(s)}{1+P(s)B(s)} r(s) + \frac{W(s)}{1+P(s)B(s)} w(s)\,, \tag{5}$$

and the output error $e(s) = r(s) - y(s)$ is obtained as

$$e(s) = \underbrace{\frac{1-P(s)F(s)}{1+P(s)B(s)} r(s)}_{e_r} - \underbrace{\frac{W(s)}{1+P(s)B(s)} w(s)}_{e_w}\,. \tag{6}$$

A. Design of feedforward controller

From the last equation we can see that the output error e contains two components, one due to the reference r (the error $e_r = (1-PF)/(1+PB)\times r(s)$) and the other due to the disturbance w (the error $e_w = W/(1+PB)\times w(s)$). In order for the object $P(s)$ to track asymptotically the reference r, the condition $e_r = 0$ must be satisfied. This leads that the transfer function of feedforward controller $F(s)$ is given as

$$F(s) = P^{-1}(s) = \frac{1}{K}(\tau s^2 + s) = \frac{u_F(s)}{r(s)}\,, \tag{7}$$

which causes, according to [2, 11] and for $r(s) = \theta(s)$, that the feedforward control in time domain $u_F(t)$ is given, using differentiation, as

$$u_F(t) = \frac{1}{K}(\tau\ddot{\theta}(t) + \dot{\theta}(t)) = \frac{1}{K}(\tau\varepsilon(t) + \omega(t))\,. \tag{8}$$

From (6) is seen that the feedforward controller (7) cannot eliminate the effect of the disturbance $w(t)$, so that the second part of the error (6), the error e_w, is not zero. For the step disturbance $w(t) = w_o 1(t)$[1] and for $W(s) = P(s)$, its stationary value becomes

$$e_w(\infty) = \lim_{s\to 0} s \frac{w_o}{s} \frac{P(s)}{1+P(s)B(s)} = \frac{w_o P(0)}{1+P(0)B(0)} \,. \tag{9}$$

As in our case the $P(0)$ is always different from zero, $P(0) \neq 0$, then from (9) we can conclude that $e_w(\infty)$ is zero if only the stationary gain of the feedback controller $B(0)$ is infinity, $B(0) = \infty$, so that a PID controller is used as the feedback controller.

B. Design of feedback controller

Thus, the main task of the feedback controller is to compensate some unknown external disturbances and some unmodeled dynamics of the object. The transfer function of the continuous PID controller is

$$B(s) = k_p + \frac{k_i}{s} + k_d s = k_p\left(1 + \frac{1}{T_i s} + T_d s\right),\ \ T_i = \frac{k_p}{k_i},\ T_d = \frac{k_d}{k_p}, \tag{10}$$

where the parameters k_p, k_i and k_d are proportional, integral and derivative constants, and T_i, T_d are integral and derivative time respectively. There are several different approaches for determining the above parameters. In this paper we use the poles placement method that is in [11, 12] described in detail[1]. Based on this, we obtain

$$\begin{gathered} k_p = \frac{\omega_n \tau(2\varsigma\sigma + \omega_n)}{K},\ k_i = \frac{\tau\sigma\omega_n^{\,2}}{K},\ k_d = \frac{\tau(2\varsigma\omega_n + \sigma) - 1}{K}, \\ \text{and} \quad T_i = \frac{(2\varsigma\sigma + \omega_n)}{\sigma\omega_n},\ T_d = \frac{\tau(2\varsigma\omega_n + \sigma) - 1}{\omega_n \tau(2\varsigma\sigma + \omega_n)}, \end{gathered} \tag{11}$$

where K and τ are the object's parameters and ω_n, ς and σ are undamped frequency, damping ratio and real pole respectively. Through these parameters locations of the poles in the open LHP of $s-$plane are determined and, equivalently, the dynamics of the overall system. In our case it is adopted that $\varsigma = 1$ (the output error tends to the zero value without oscillations), which induces that the dominant poles are $s_{1,2} = -\omega_n$ and pole $s_3 = -\sigma$, $\sigma \geq 7\omega_n$. For implementation of the feedback controller there is used a discrete PID controller, obtained from (10) by approximations. The integral part is approximated by trapezoidal $(s = 2/T_s \times (z-1)/(z+1))$ and differential part by backward approximation $(s = (z-1)/zT_s)$, so we finally get the discrete transfer functions of the feedback controller $B(z)$, as

$$B(z) = \frac{u_B(z)}{e(z)} = k_{pd} + \frac{k_{id}}{1-z^{-1}} + k_{dd}(1-z^{-1}), \tag{12}$$

where

$$k_{pd} = k_p\left(1 - \frac{T_s}{2T_i}\right),\ k_{id} = k_p \frac{T_s}{T_i},\ k_{dd} = k_p \frac{T_d}{T_s} \,. \tag{13}$$

---

[1] desired characteristic equation of the overall system is $\Delta_d(s) = (s+\sigma)(s^2 + 2\varsigma\omega_n s + \omega_n^{\,2})$

In the above equation T_s is the sampling time and the values T_i, T_d and k_d are given in (11). Now, based on (12) and (13), the feedback control $u_B(z)$ in complex domain is obtained as

$$u_B(z) = \underbrace{k_{pd}e(z)}_{P(z)} + \underbrace{\frac{k_{id}}{1-z^{-1}}e(z)}_{I(z)} + \underbrace{k_{dd}(1-z^{-1})e(z)}_{D(z)} . \tag{14}$$

This feedback control has three components, $P(z)$, $I(z)$ and $D(z)$, which in the next chapter are implemented separately.

4. Implementation of the Overall System

For implementation of the overall system it is necessary to solve two tasks: to determine the transfer function of the object and to realize the above proposed control system.

A. Determination transfer function of the object

As object we use DC M28N-1 (manufacturer is MABUCHI, for details to see http://www.mabuchi-motor.co.jp/cgi-bin/catalog/e_catalog.cgi?CAT_ID=rs_385ph) DC mini motor with a load and two-channel optical encoder whose resolution is $N_c=448$ pulses per rotation (see Fig. 4). One channel (channel A) is used to measure the current position, while the second channel (channel B) is used for determining the direction of the motor rotation. To determine the transfer functions of the object (1) we have to find values of the parameters K and τ. For their determination we use MatLab (more precisely, its System Identification Toolbox) and an Arduino software. First, we made an Arduino software that for the known voltage (*uex* [V]) which is connected to the motor and using interrupts, every 0.005[sec] (*Tsam* [sec]) we read the number of pulses which coming from the encoder and write them (as a angle *yex* [rad]) to an excel file. Based on these data and using *data=iddata(yex,uex,Tsam)* there is formed MatLab object (named as *data*), from which and using the function *Ps=pem(data,'P2')*, was found the transfer function $P(s)$ (named as *Ps*), where: K = 32.4 [rad/Vsec] and τ = 0.025[sec]. Based on the time τ, sampling time T_s is determined as, $T_s = \tau_m$ / (2~5) = 0.005[sec], where the $\tau_m = \tau$ is the least time constant.

B. Implementation of the control system

For the implementation of the overall system there is used the speed control system with angular velocity as reference, $r(t) = \omega_d(t)$, and $P_v(s) = \Omega(s)/U(s) = K/(\tau s+1)$ that is used as the object. The feedforward controller is implemented based on (8) and (2), and feedback based on (11), (13) and (14). After trajectory planning the times T, t_1 and t_u are known, so using (2) for each time interval we calculate the jerk $j(t)$ at every instant of time $t_k = kT_s$, k = 0,1,2,... Using this jerk, and using backward approximation, $s = (1-z^{-1})/T_s$, the values of the angular acceleration, velocity and position in (8) are obtained, so we have

$$\begin{aligned}
\varepsilon(k) &= \varepsilon(k-1) + j(k)T_s, \\
\omega(k) &= \omega(k-1) + \varepsilon(k)T_s, \\
\theta(k) &= \theta(k-1) + \omega(k)T_s.
\end{aligned} \tag{15}$$

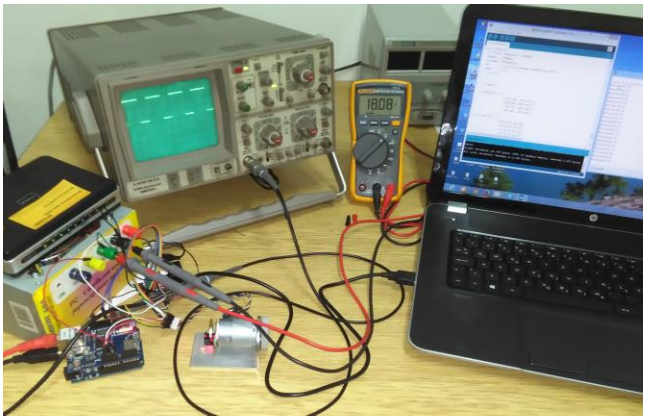

Fig. 4 Arduino Uno with Ethernet shield, DC motor with encoder and load, L298 H-bridge driver, DC power supply, notebook and oscilloscope

At the next instant of time, the k_{th} instants from (15) become $(k-1)_{th}$ (e.g. $\varepsilon(k-1) = \varepsilon(k)$) and so on. Now, feedforward control in discrete time domain becomes

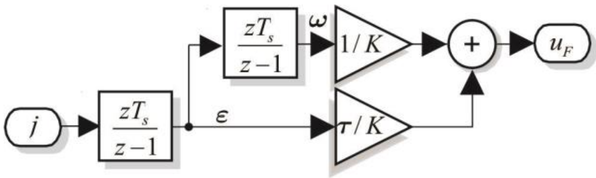

Fig. 5 Realization of the feedforward controller

$$u_F(k) = \frac{\tau}{K}\varepsilon(k) + \frac{1}{K}\omega(k)\,, \tag{16}$$

where $\varepsilon(k)$ and $\omega(k)$ are given in (15). This is shown in Fig. 5, and it is very suitable for realization using the microcontroller.

For the realization of the feedback control $u_B(k)$ it is necessary to calculate the output error $e(k)$. It is calculated for every moment k as $e(k) = \omega_{Td}(k) - \omega_T(k)$, where the $\omega_{Td}(k)$ is the reference $r(t)$ expressed as a number of pulses during time T_s, and that is obtained from (15). The velocity $\omega_T(k)$ is measured as a number of pulses coming from the encoder during T_s also. Based on this error and known values (from (11) and (13)) of k_{pd}, k_{id} and k_{dd}, for every discrete time k we calculate all parts of the control (14) as

$$\begin{aligned} p(k) &= k_{pd}e(k), \\ i(k) &= i(k-1) + k_{id}e(k), \\ d(k) &= k_{dd}(e(k) - e(k-1)), \end{aligned} \tag{17}$$

so that the feedback control is

$$u_B(k) = p(k) + i(k) + d(k)\,, \tag{18}$$

see Fig, 6.

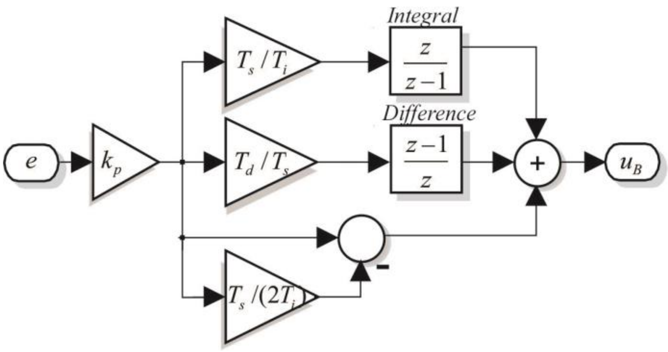

Fig. 6 Realization of the feedback controller

At the initial time, $k = 0$, the values of $(k - 1)$ are zero, ($e(k - 1) = 0$ and $i(k - 1) = 0$). At the next instant of time, k_{th} values become $(k - 1)_{th}$ and so on (for example, the current values $i(k) = i_{old}$, $e(k) = e_{old}$ at the next k_{th} instant become $i(k-1) = i_{old}$, $e(k-1) = e_{old}$).

The total control $u(k)$ is the sum of the feedforward (16) and the feedback control (18), i.e. $u(k) = u_F(k) + u_B(k)$. For its realization a 10-bit PWM signal is used which the Arduino generates. Since the Arduino, default, generates only PWM signals with 8-bit resolution, which is insufficient in this case, were certain software interventions that have been realized using 16-bit Timer 2 of microcontroller are necessary. In this way a 10-bit PWM signal is obtained whose frequency is 8KHz, which is here quite enough.

All the above theoretical considerations are practically realized using the Arduino Uno microcontroller's platform. This platform is based on 8-bit microcontroller, Atmel MEGA 328, with 32Kb programmable flash memory, 1Kb EEPROM and 2Kb RAM memory. As a driver for DC motor is used H-bridge module based on a dual full-bridge driver L298D, see Fig. 7. In our case a remote wireless interface it is implemented, too.

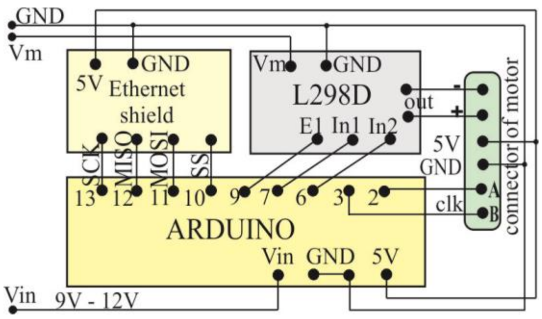

Fig. 7 Implementation of the overall system using Arduino Uno, L298 H-bridge motor driver and Arduino compatible Ethernet modules

It is achieved via Arduino compatible 10/100 Mbps Ethernet module (based upon W5100 chip), where Arduino together with this module and appropriate software are configured as a TCP/IP server (based on C++ and HTML language). As a client we can use some Windows computer and/or Android mobile phone.

As the operating voltages there are used DC power supply, for motor Vm: 12 - 30V, 800 mA and for Arduino and encoder Vin: 9-12V, 100mA.

6. Conclusion

The design and implementation of a digital position tracking system with controlled jerk is performed. For in advance determined sine changes of the jerk and the given maximum values of the motion parameters, the algorithm of the trajectory planning is also developed. As the controlled object the mini DC motor with encoder and load is used. The combination of the feedforward and feedback controller is used as the control system. The control software is developed and realized experimentally. For its implementation the Arduino Uno platform and H-bridge motor driver are used. The remote user interface is performed using the HTML software and compatible Ethernet module. The experimental results confirm the proposed theory.

References

[1] A. Cervin, J. Eker, B. Bernhardsson, K. Eric, *Feedback Feedforward Scheduling of Control Tasks, Real Time Systems.* Special Issure on Control Theoretical Approaches to Real Time Computing, 2002.

[2] P. Lumbrechts, *Trajectory planning and feedforward design for electromechanical motion systems*. Report No. *DCT 2003-08*, University of Technology Eindhoven, 2003.

[3] P. Lumbrechts, M. Boerlage, M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," Control Engineering Practice, vol. 13, no. 2, pp. 145–157, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2004.02.010

[4] V. Cerone, M. Milanese, D. Regruto, "Robust feedforward design for a two-degrees of freedom controller," System & Control Letters, vol. 56, no. 11-12, pp. 736–741, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.sysconle.2007.04.010

[5] H. Z. Li, Z. M. Gong, W. Lin, T. Lippa, "Motion profile planning for reduced jerk and vibration residuals", SIMTech Technical Reports, vol. 8, no. 1, pp. 32–37, 2007. [Online]. Available: http://www.google.rs/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CBwQFjAA&url=http%3A%2F%2Fcfile25.uf.tistory.com%2Fattach%2F167811264CAEC38A136E4C&ei=4AKYVIeKB8isU9KIgZgL&usg=AFQjCNGV9HbEcW2XgI_j45JIqOe_Y0plEA

[6] K. D. Nguyen, T. C. Ng , I. M. Chen, "On algorithms for planning S-curve motion profiles," International Journal of Advanced Robotic Systems, vol. 5, no. 1, pp. 99–106, 2008. [Online]. Available: http://cdn.intechopen.com/pdfs-wm/4267.pdf

[7] G. M. Clayton, S. Tien, A. J. Fleming, S. O. R. Moheimani, S. Devasia, "Inverse feedforward of charge controlled piezopositioners," Mechatronics, vol. 18, no. 5-6, pp. 273–281, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.mechatronics.2007.07.006

[8] M. J. Stojčić, B. Knežević, "The controller design for tracking trajectory with controlled jerk," in *Proceedings of DEMI 2011*, Banja Luka, Bosnia and Herzetgovina, pp. 905–911, 2011.

[9] M. J. Stojčić, B. Knežević, "Design of digital controller that ensures robust tracking trajectory with controlled jerk," in *Proceedings of the JUPITER 37*, Belgrade, Serbia, pp. 4.11–4.16, 2011.

[10] M. J. Stojčić, "Design of electromechanical positioning system with controlled jerk," in *Proceedings of the Haevy Mashinery – HM2011*, Kraljevo, Serbia, pp. 13–17, 2011.

[11] M. Y. Stoychitch, "Design of an electromechanical positioning system with sinusoidal change of the jerk," in *Proceedings of the International Conference on Future Trends in Structural, Civil, Environmental and Mechanical Engineering - FTSCEM 2013*, pp. 35–38, 2013. [Online]. Available: http://dx.doi.org/10.3850/978-981-07-7021-1_48

[12] M. Y. Stoychitch, "Design of a digital electromechanical positioning system with sinusoidal change of the jerk," Applied Mechanics and Materials, vol. 474, pp. 255–260, 2014. [Online]. Available: http://dx.doi.org/10.4028/www.scientific.net/AMM.474.255

[13] M. Y. Stoychitch, "Design and Implementation of a Digital Control System for Tracking a Trajectory with Controlled Jerk," in *Proceedings of the XII International SAUM Conference on Systems, Automatic Control and Measurements*, Niš, Serbia, pp. 188–191, 2014.