

# Robust Stabilization of a Servomechanism with Respect to Time-Delay

Baris Samim Nesimioglu <sup>\*1</sup>, Sabri Yilmaz <sup>2</sup>, Emre Dincel <sup>2</sup>

Accepted 3<sup>rd</sup> September 2016

**Abstract:** In this paper, a servomechanism under teleoperation is considered. Since the teleoperation itself can result in large amount of time-delays and this amount can change operation to operation, it can be difficult to control such mechanisms in order to accomplish the desired tasks. From the robust control viewpoint, a methodology that guarantees the stability in worst case is essential. Based on a simple methodology to find the delay independent stabilizing proportional (P) controller regions, just by forming the magnitude polynomial and employing the root locus technique, the stability of the robot is guaranteed, even in the worst case: the system becomes stable even if the connection has huge amount of time-delays. This fact is evidenced first by the simulations. To simulate the real system, as there is no information about the motor parameters, the motor is modeled by a global optimization methodology, named Genetic Algorithm in order to obtain a valid model for the system as accurate as possible. Then the resulting P controllers are applied to the real system, the results of which are found in accordance with the simulation results; the stability of the operation is not affected by the time-delay.

**Keywords:** Teleoperation System Control, Time-Delay Systems, Robust Control, Low-Order Controllers, Delay Independent Stability

## 1. Introduction

Teleoperation is one of the important control strategy which especially suits the unfavourable environmental conditions, such as underwater or mountainous terrains and for the processes that is dangerous for human life, such as nuclear plants, toxic chemical reactions etc. This methodology has been more widely used especially for a few decades due to the advances of the internet based technologies [1, 2]. Nevertheless, the most important drawback of this methodology is the phenomenon of the propagation time-delay, amount of which can be huge. Moreover, the amount of time-delay can be affected from operational conditions and hence can be changed operation to operation [3-7].

Time-delay, which is almost an inevitable phenomenon for most of the physical systems, has generally adverse effects on the system performance [8]. In other words, a designed control system to behave in a prescribed way with the assumption of the absence of the time-delay does not work as it is intended under the emergence of the time-delay. The amount of the performance deviation might become greater, if the amount of the time-delay increases; even the system might be unstable.

From the mathematical point of view, for Linear Time Invariant (LTI) and Single Input Single Output (SISO) systems, this fact is mostly caused by the fact that time-delay term adds infinite number of poles in the closed-loop. As a result, analysing the performance of time-delay systems, even assessing the stability becomes more complicated in such systems [9- 16].

As might be expected from the operational and the mathematical viewpoints and as stated above, the amount of the time-delay may considerably attenuate the overall control performance, even it may lead the system to instability. As the stability is the first and the foremost important requirement of the design, an effective

methodology providing the time-delay independent stability is essential from the robust control viewpoint.

To address this problem, in [17], an effective and simple methodology to compute all P type controllers providing time-delay independent stability is given for SISO-LTI systems. Although there are some methods about delay-independent stability available in the literature [18-20], the method given in [17] is much easy to understand and provides a simple and fast solution for the SISO LTI plants. The methodology is based on the stability analysis given in [9] and it consists of forming the magnitude polynomial, sketching the resulting root locus with the suitable transformations of the variables and searching the regions of gains for which any positive real roots do not exist.

Using the methodology given in [17], a servomechanism subject to teleoperation is considered in this paper. The motors are selected as Dynamixel MX-106T [21] and as the motor parameters are not explicitly given, the global optimization method, named Genetic Algorithm (GA) is employed to obtain a proper model of the motors. Then using the obtained mathematical model, all gains providing time-delay independent stability is computed as proposed in [17], and it is shown that the simulation results are as expected. To verify the simulation results in the real environment, the resulting gains are applied the servomechanism subject to teleoperation under different scenarios, i.e. for different values of time-delay. Then it is shown that, the application results are in accordance with the simulations, i.e. the stability of the system is not affected by the time-delay even if the delay is of huge values, such as 12 seconds.

The paper is organized as follows: The second section is devoted to the theoretical background, in order for the readers to follow the paper easily. In this section, the methodology given in [17] is presented and the brief summary about the GA is given, respectively. In the third section, it is explained how the mathematical model of the motors are obtained and the methodology given in [17] is applied to this model, i.e. all the gains providing time-delay independent stability is calculated. In the fourth section, the simulation and the real system results are

<sup>1</sup> Istanbul Technical University, Turkey

\* Corresponding Author: Email: -

Note: This paper has been presented at the 3<sup>rd</sup> International Conference on Advanced Technology & Sciences (ICAT'16) held in Konya (Turkey), September 01-03, 2016.

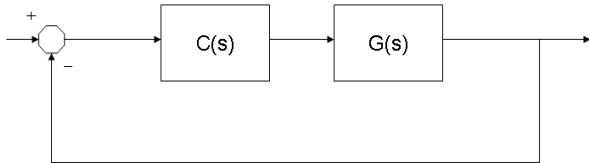
shown, in a comparative manner. In the last section, further discussions are given and the possible extensions of this study is stated.

## 2. Theoretical Background

In this section, the methodology for the computation of all gains providing time-delay independent stability proposed in [17] will be briefly explained and the Genetic Algorithms (GA) will be briefly overviewed, respectively.

### 2.1. Calculation of All Gains Providing Time-Delay Independent Stability Using the Root Locus

In this subsection, the methodology given in [17] will be briefly



explained. Since this subsection is a summary only, the reader is referred to [17] for further discussions and explanations.

To begin with, let the interested control scheme is given in figure 1.

**Figure 1.** The interested control scheme

Where the  $G(s)$  and the  $C(s)$  is given as follows:

$$G(s) = \frac{N(s)}{D(s)} e^{-sL} = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0} e^{-sL}, \quad C(s) = k_p \quad (1)$$

Where  $N(s)$ ,  $D(s)$  and  $C(s)$  denote the numerator - the denominator polynomials and the P type controller respectively. Additionally,  $L$  symbolizes the nonnegative time-delay of the system.

Apparently, when  $L=0$ , i.e. the system is delay-free, the characteristic structure of the closed loop system constitutes a polynomial; i.e the closed loop poles are finite in number and it is easy to determine the stability of the system. However, when  $L$  takes the values greater than zero, the closed loop structure begins to constitute an expression named quasipolynomial, which possess infinite number of roots; resulting in infinite number of closed-loop poles. Additionally, these poles move in the  $s$  plane with the change of the  $L$  value.

To track these movements in terms of stability problem [8], let an infinitesimally small  $L$  is introduced to the system. In that case, the number of the closed loop poles become infinite, and these new poles appears at infinity in the  $s$  plane. From the stability viewpoint, all of these roots must occur in the left half  $s$  plane. To sum up, if the following conditions are satisfied, all the new roots occur in the left half plane:

- i)  $m < n$ , i.e the system is strictly proper, or
- ii)  $m = n$ , i.e. the system is biproper and  $|k_p b_n / a_n| < 1$

In other words, if the system is strictly proper, all the new roots

<sup>1</sup>Mechatronics Engineering Department, Engineering Faculty, KTO-Karatay University, 42030 Konya/Turkey

<sup>2</sup>Department of Control and Automation Engineering, Electrical and Electronics Faculty, 34467 Istanbul/Turkey

\* Corresponding Author: Email: baris.samim.nesimioglu@karatay.edu.tr

occur in the left half  $s$  plane without any further condition. If the system is biproper, an additional condition have to be satisfied about the ratio of the principal coefficients, i.e.  $|k_p b_n / a_n| < 1$ .

Assuming that either of these conditions are satisfied, the roots

begin to move in the  $s$  plane with the increase of the  $L$ . When  $L$  reaches to a critical value, say  $L_{cr}$ , a pair of roots begin to locate on the imaginary axis just before changing their half planes, which is crucial as it implies a possible stability change. When such a situation happens, the closed loop expression satisfies the following equation:

$$D(j\omega) + e^{-j\omega L} k_p N(j\omega) = 0 \Rightarrow e^{-j\omega L} = -\frac{D(j\omega)}{k_p N(j\omega)} \quad (2)$$

Obviously, this expression implies two condition to be satisfied:

$$\left| -D(j\omega) / k_p N(j\omega) \right| = 1 \quad (3)$$

$$\angle -D(j\omega) / k_p N(j\omega) = -\omega L \quad (4)$$

The first one can be considered as the magnitude condition, which determines the location of the closed loop pole pairs on the imaginary axis, just before changing their half planes. The second one can be thought as the phase condition, from which, the corresponding  $L$  value, i.e.  $L_{cr}$  is determined.

If the magnitude condition given in (3) is elaborated, the following equivalent condition can be obtained:

$$W(\omega^2) := D(j\omega)D(-j\omega) - k_p^2 N(j\omega)N(-j\omega) = 0 \quad (5)$$

As might be expected, the roots of this equation correspond to the solution of (3). Apparently, (5) is a polynomial of  $\omega^2$  implying that the closed loop poles cross the imaginary axis as pairs by locating at  $\pm j\sqrt{\omega}$ . Moreover, it is independent of  $L$  and its roots change with  $k_p$ .

Clearly, the complete imaginary axis crossing of the closed-loop poles, i.e. possible stability changes with respect to time-delay are governed by (5). Then if there can be found some  $k_p$  values and/or regions such that (5) has no positive real roots, it can be stated that for these  $k_p$  values, no pair of closed-loop poles cross the imaginary axis i.e. there is no stability change. Since (5) is a polynomial that is independent of  $L$ , this situation does not change with the  $L$  values; implying that the stability of the time-delay system is not affected by the time-delay.

In this way, if (5) is elaborated, it can be observed that the problem can be converted into a simple root locus sketch. To accomplish that, let the following variable transformations are done in (5):

$$v := \omega^2, \quad \alpha := k_p^2 \quad (6)$$

$$A(v) := D(j\omega)D(-j\omega), \quad B(v) := N(j\omega)N(-j\omega) \quad (7)$$

If both sides of (5) are divided into  $A(v)$  along with the other transformations above, (5) can be rewritten as

$$1 - \alpha \frac{B(v)}{A(v)} = 0 \quad (8)$$

Obviously, this defines a negative root locus in the  $v$  plane. The all needs to be done is to interpret this root locus whether there exists  $\alpha$  gain values and/or intervals such that (8) has no positive real roots. If such an  $\alpha$  gain value and/or interval can be found, for those gains (5) has also no positive real roots, implying that no root pair can cross the imaginary axis, regardless of the value of  $L$ . Obviously, by considering (6), those  $\alpha$  gain values are squared gains and needs to be retransformed into  $k_p$  values again. Let such a squared gain interval  $(\alpha_1 \alpha_2)$  is found for which (8) has no positive real roots. Then from (6), the corresponding gains can

be given as the following intervals:

$$k_p \in (-\sqrt{\alpha_2}, -\sqrt{\alpha_1}) \cup (\sqrt{\alpha_1}, \sqrt{\alpha_2}) \quad (9)$$

To sum up, for those gains, the stability of the system given in (1) with figure 1 is not affected by the time-delay; if the delay-free system is stable (unstable) the time-delay system is stable (unstable) for all values of L.

## 2.2. Genetic Algorithms (GA)

Here, a brief overview of the Genetic Algorithms is given, and the reader is referred to [22] for much more detailed explanations. Genetic Algorithms are known as their robustness and usefulness as optimization algorithms. In principle, GAs are search algorithms based on mechanics of natural selection and natural genetics. They combine survival of the fittest among the string structures with randomized yet structured information exchange to form a search algorithm with innovative flair of natural evolution [22].

In general, it can be mentioned that GA mainly works at three stages which are briefly described below:

- 1) **Reproduction:** The stage where the candidates are copied by their fitness function values.
- 2) **Crossover:** The stage where some parts of the candidates are changed within themselves.
- 3) **Mutation:** The stage where the randomization is applied.

## 3. Identification of The System and The Calculation of The Time-Delay Independent Stabilizing Gains

In this chapter, it will be described how the system model is obtained and using the obtained mathematical model, the corresponding gains will be calculated employing the methodology explained in the subsection 2.1.

### 3.1. Identification of the System

As mentioned before, the utilized motors are selected as the Dynamixel MX-106T. It is a permanent magnet direct current (PMDC) motor and all the necessary information and all operating conditions about the motor can be found in [21]. However, since the motor parameters are not explicitly known, a system identification process is taken into account from the real

data obtained from the motors. However, before describing the identification process, it is desirable to discuss the model properties of the servomechanisms.

In general a permanent magnet direct current (PMDC) motor can be modelled by the following equations:

$$\tau_m = \tau_L + J_m \frac{d\omega_m}{dt} + B_m \omega_m \quad (10)$$

$$V_a = V_b + L_m \frac{dI_m}{dt} + R_m I_m \quad (11)$$

$$\tau_m = K_t I_m \quad (12)$$

$$V_b = K_b \omega_m \quad (13)$$

where  $\tau_m$  and  $\tau_L$  represents the motor and load torque,  $J_m$  and  $B_m$  represents the equivalent inertia and the viscous damping at the armature,  $V_a$  and  $V_b$  represents the applied voltage and the back electromotive force (back emf), and the  $K_t$  and the  $K_b$  represents the motor torque and the motor back emf constants, respectively. The corresponding block diagram of such a servomechanism can be seen in figure 2.

And also from the motor document [22], it is known that, motor has a closed loop block, which does not affect the system order but affects the form of the transfer function, i.e. the integral term at the output of the figure 2 cannot be present in the transfer function of the motor. From the equations above, from the block diagram in figure 2 and from [22], it is clear that, the transfer function from the input ( $V_a(s)$ ) to the output ( $\theta_m(s)$ ) results in a third order system no longer comprising the integral effect and can be given as follows::

$$G(s) = \frac{K}{s^3 + as^2 + bs + c} \quad (14)$$

To obtain a valid mathematical model, the real data are taken from the motor to employ them in the system identification toolbox of the MATLAB. The inputs applied to the motor and the output of the motor can be seen in the figure 3.

Using the data shown in figure 3, and with the priori of the order of the model, the parameters given in (14), i.e. K, a, b and c are obtained, and they are used to determine the initial ranges for the GA. These obtained parameter values are shown in the Table 1.

Also, the parameters for GA are shown in Table 2. In this table,

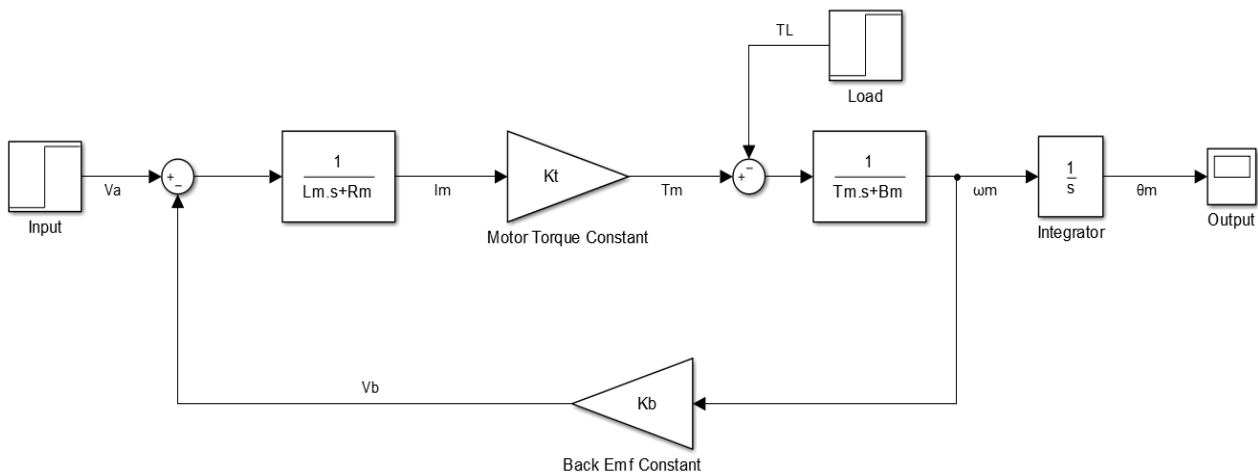


Figure 2. Block diagram of a servomechanism.

the population size is the number of candidates, crossover fraction is the factor for reproduction, mutation is selected as constraint dependent, the nonlinear constraint is the penalty factor and iteration is the number of iterations.

As the last, the obtained motor parameters from the GA is shown in the Table 3.

In other words, the transfer function of the servomechanism given in (14) can be written as follows:

$$G(s) = \frac{11520.33}{s^3 + 449.371s^2 + 4103.139s + 11599.107} \quad (15)$$

To show the validity of the model, the model output and the actual motor output is compared in figure 4.

After obtaining the motor parameters and the equivalent servomechanism transfer function, it is desirable to describe the system subject to teleoperation. As might be expected, the system consists of a master and the slave operators (i.e. servomechanisms) which are sufficiently far from each other (up to several kilometres) and it is expected that the slave operator to follow the position of the master operator. The real system and the corresponding block diagram is shown in figures 5 and 6, respectively.

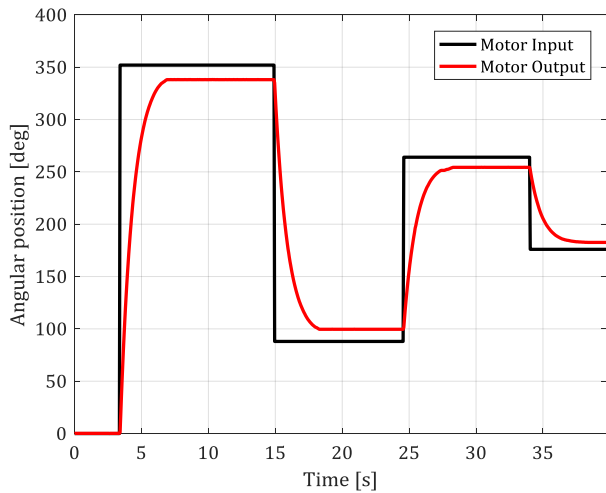


Figure 3. The input applied to the motor and the response of the motor.

Table 1. System identification toolbox results

Parameter	Value
<b>K</b>	11820.461
<b>a</b>	532.478
<b>b</b>	4786.245
<b>c</b>	11816.266

Table 2. Genetic algorithm parameters

Parameter	Value
<b>Population Size</b>	50
<b>Crossover Fraction</b>	0.8
<b>Penalty Factor</b>	100
<b>Iteration</b>	50

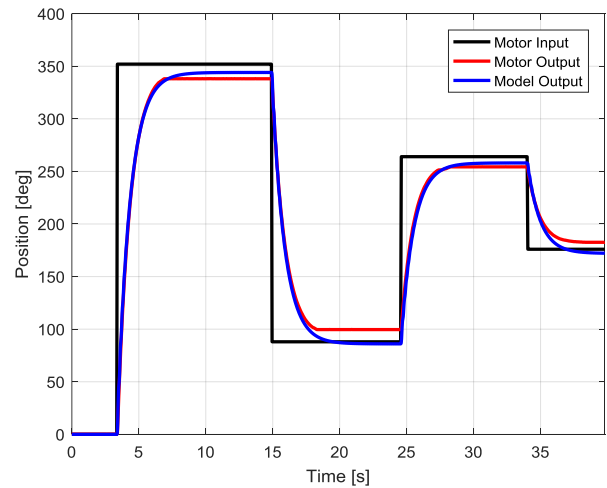


Figure 4. Motor input, output and the model output.

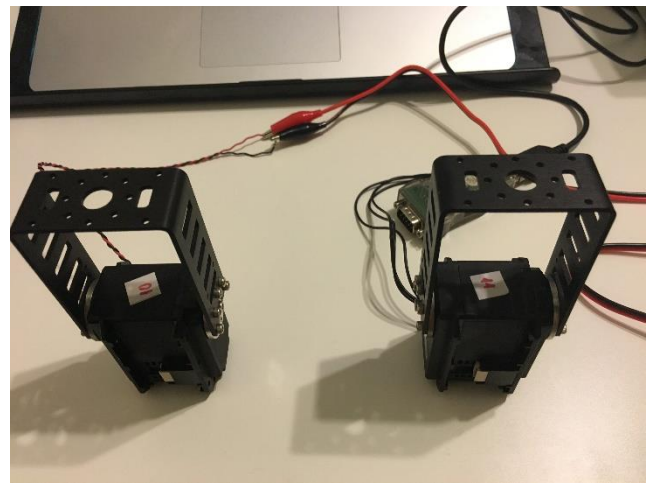


Figure 5. The utilized system (the master and the slave servomechanisms)

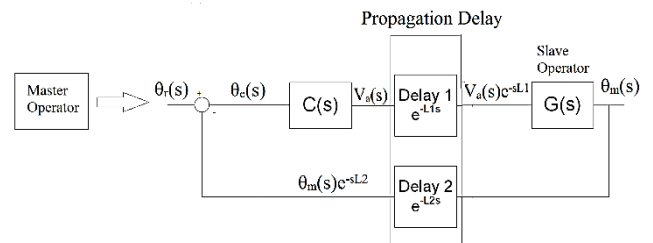


Figure 6. The simplified block diagram of the complete system

In the block diagram, the reference input is the angular position input produced by the master operator,  $C(s)$  is the proportional controller, the values of which will be calculated in the next subsection, and  $L_1$  and  $L_2$  is the propagation delay from master to slave operator and vice versa respectively, along with  $G(s)$  is the obtained transfer function of the servomechanism.

Supposing that the propagation delay from master to slave is  $L_1$  seconds and from slave to master is  $L_2$  seconds, the closed-loop transfer function,  $P_c(s)$  becomes as follows:

$$P_c(s) = \frac{11520.33k_p e^{-L_1 s}}{s^3 + 449.371s^2 + 4103.139s + 11599.107 + 11520.33k_p e^{-(L_1+L_2)s}} \quad (16)$$

### 3.2. Calculation of All Time-Delay Independent Stabilizing Gains

In this subsection, based on the mathematical model given in (16), all the stabilizing gains providing time-delay independent

stability will be computed using the methodology summarized in the subsection II-A.

To begin with, it is noticeable to mention that, the polynomials  $N(s)$  and  $D(s)$  given in (1) are as follows in (16):

$$N(s) = 11520.33 \quad (17)$$

$$D(s) = s^3 + 449.371s^2 + 4103.139s + 11599.107 \quad (18)$$

along with a time-delay of  $L=L_1+L_2$  seconds.

To compute all the time-delay independent stabilizing gains, if the polynomial is formed as suggested in (5), the following is obtained:

$$W(\omega^2) = \omega^6 + 193728\omega^4 + 6411145\omega^2 + 134539283 - 132718003k_p^2 = 0 \quad (19)$$

If the transformations given in (6)-(7) is done, the polynomials  $A(v)$  and the  $B(v)$  is found as follows:

$$A(v) = v^3 + 193728v^2 + 6411145v + 134539283 \quad (20)$$

$$B(v) = 132718003 \quad (21)$$

Considering these new polynomials and variables in (19) and dividing the both sides by  $A(v)$  gives the following root locus problem:

$$1 - \alpha \frac{132718003}{v^3 + 193728v^2 + 6411145v + 134539283} = 0 \quad (22)$$

The corresponding root locus is sketched via MATLAB, and can

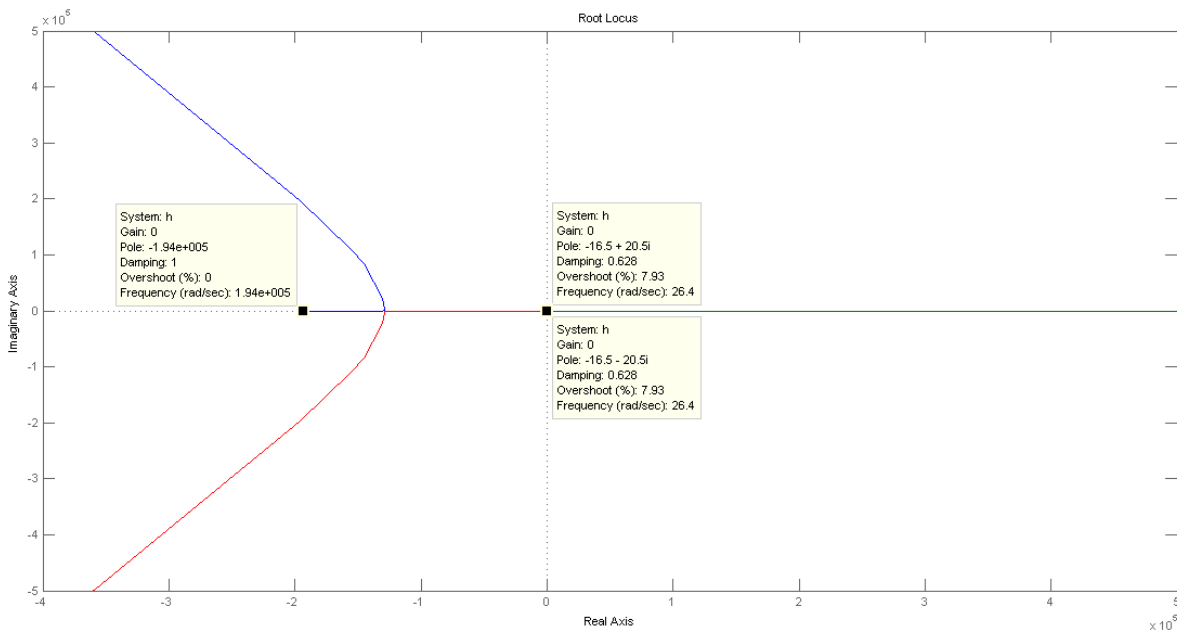


Figure 8. The root locus of (22) (complete)

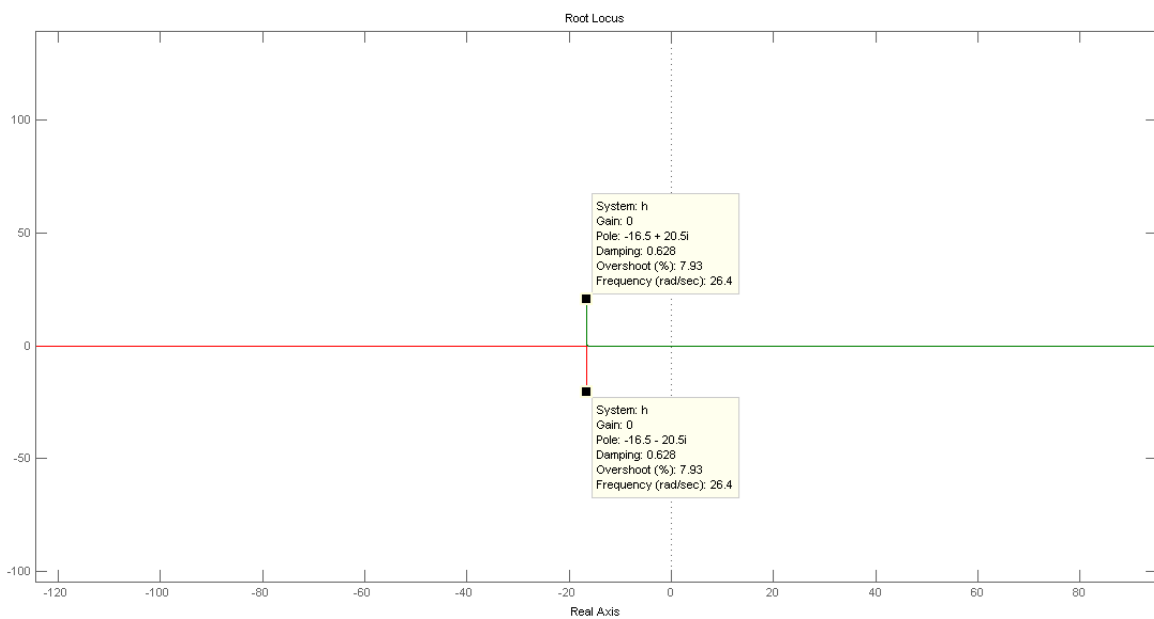


Figure 8. The root locus of (22) (zoomed-in)

be seen in figure 7. The initial roots of the locus can be found as  $v_1 = -193694.92$  and  $v_{2,3} = -16.548 \pm 20.513j$ . Since the initial roots are extremely far from each other, the zoomed in version of the root locus can be seen in figure 8.

To interpret the root locus, it is clear from the figures that, there is no positive real roots for  $\alpha=0$ . When  $\alpha$  begins to increase, the complex conjugate roots break into the real axis, and one of them moves to positive real axis by passing the point  $v=0$  for a critical value of  $\alpha$ , say  $\alpha_{cr}$ . For  $\alpha$  values greater than  $\alpha_{cr}$ , the corresponding root always locate at the positive real axis, thereby, there exists always a positive root on the real axis. Then it can be summarized that there is no positive root of (22) for the  $\alpha$  values such that  $\forall \alpha \in (0, \alpha_{cr})$ .

In order to compute the  $\alpha_{cr}$ , if the corresponding  $\alpha$  is found at the point  $v=0$  using the root locus arguments, the following is computed:

$$1 - \alpha \frac{132718003}{v^3 + 193728v^2 + 6411145v + 134539283} \Big|_{v=0} = 0 \quad (23)$$

The critical  $\alpha$ , i.e.  $\alpha_{cr}$  is found as 1.0137. Then it can be concluded that (22) has no positive real roots for  $\forall \alpha \in (0, 1.0137)$ . To obtain the corresponding  $k_p$  values, if  $\alpha$  is retransformed into  $k_p$  using (9) gives the  $k_p$  set:

$$k_p \in (-1.0068, 0) \cup (0, 1.0068) \quad (24)$$

Obviously, for these gains the stability of the closed-loop time-delay system is independent of time-delay, i.e. the stability of the time-delay system is identical of the delay-free one. Then all need is to check whether the delay-free system is stable for those  $k_p$  values. Computation of the stabilizing gains of  $P_c(s)$  given in (16) for  $L=0$  is straightforward and can be found as follows:

$$-1.0068 < k_p < 159.0434 \quad (25)$$

It is clear from (25) that for the gain values given in (24), the delay-free system is stable. Therefore, it can be concluded that for gain values given in (24) the time-delay system given in (16) is stable, regardless of the value of the time-delay.

#### 4. Simulation-The Real System Results

In this section, for the gain values given in (24), the responses of the slave operator to the signals produced by master operator are obtained under different scenarios, i.e. under different values of propagation delays. From both simulations and the real system responses, it can be observed that the system is stable and the real system results are in accordance with the simulation ones.

In addition, for the gain values outside of the given region in (24), it is also shown that the stability of the system is not independent of the time-delay and the system can be unstable for smaller propagation delays.

To overcome the intrinsic steady state error problem, a pre-amplifier is put after the input signal. The utilized simple block diagram in both simulations and for the real system is shown in figure 9.

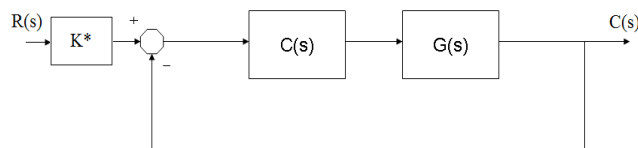


Figure 9. The simplified block diagram used for simulations and the real system.

Here, the  $K^*$  is the gain of the pre-amplifier the value of which is the inverse of the steady state value of the system. Using the gain set given in (24), with a priori of possessing a stable closed-loop system in (16), the gain of the preamplifier can be computed via final value theorem as follows:

$$K^* = \lim_{s \rightarrow 0} \frac{1}{T(s)} \rightarrow K^* = 1 + \frac{11599.107}{11520.33k_p} \quad (26)$$

At first, the controller value, i.e.  $k_p$  is taken as 0.5 and the simulation-real system results under different propagation delays are shown in figures 10-11. From the figures, it is clear that the system remains stable even if the total propagation delay is 12 seconds.

After that, the  $k_p$  is selected as 0.85, more close to the borders of the  $k_p$  set given in (24). In this case, the system remains stable even if the total propagation delay is 12 seconds. The simulation-real system results are shown in the figures 12-13.

As the last, the  $k_p$  is selected as 1.05, which is in the close proximity of the given set in (24), but out of borders. Although the selected  $k_p$  is very close to the borders of the set, it is clear from the simulation-real system results that, the robust stability is no longer guaranteed, i.e. the system is unstable, although the total propagation delay is 4 seconds. The results can be seen in figures 14-15.

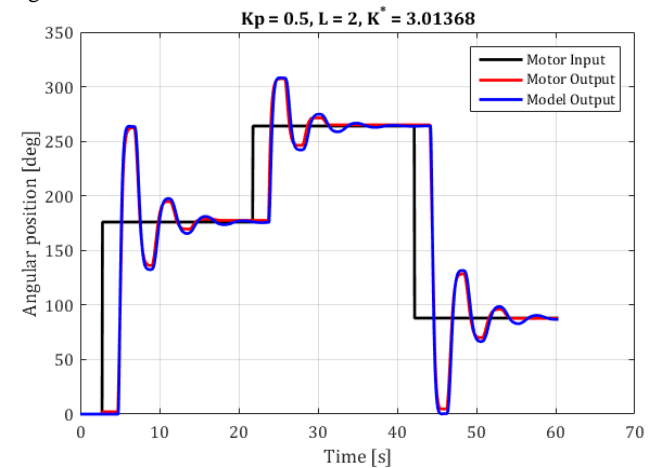


Figure 10. The simulation-real system results for  $k_p=0.5$  and  $L_1+L_2=2$  seconds.

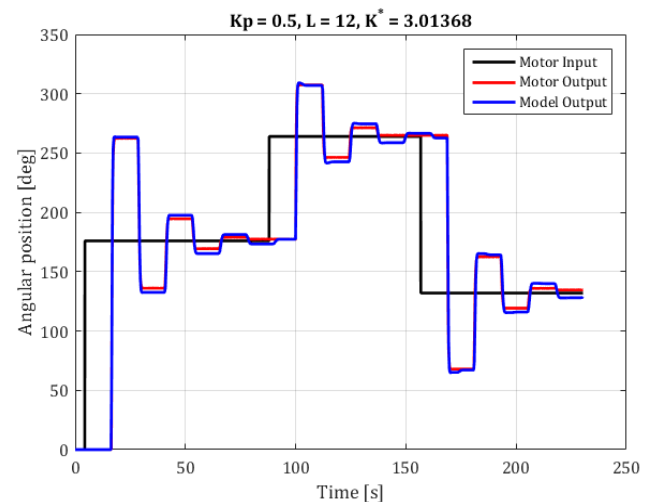


Figure 11. The simulation-real system results for  $k_p=0.5$  and  $L_1+L_2=12$  seconds.



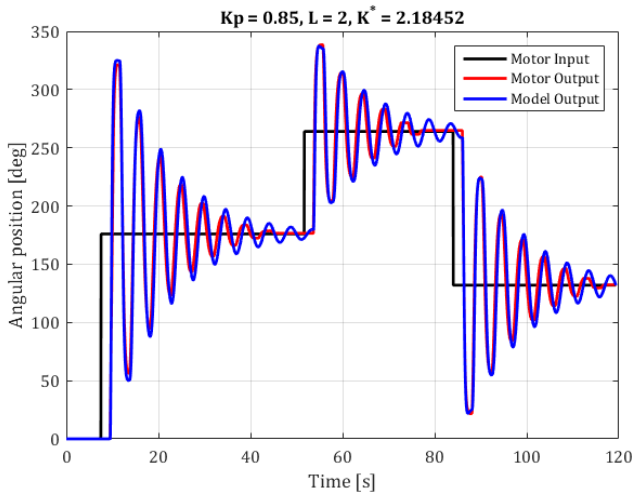


Figure 12. The simulation-real system results for  $k_p=0.85$  and  $L_1+L_2=2$  seconds.

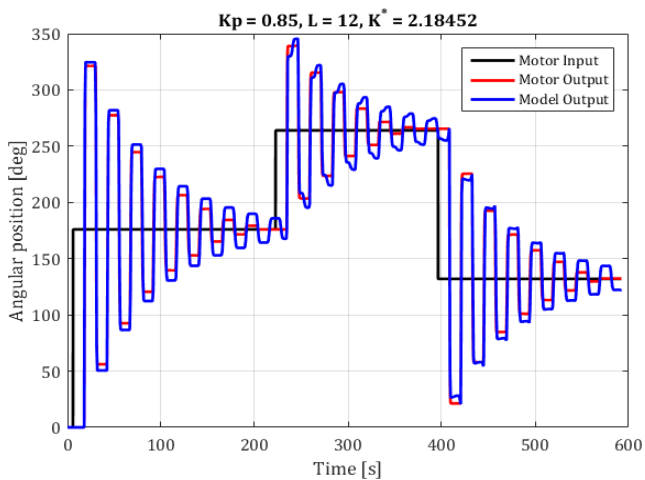


Figure 13. The simulation-real system results for  $k_p=0.85$  and  $L_1+L_2=12$  seconds.

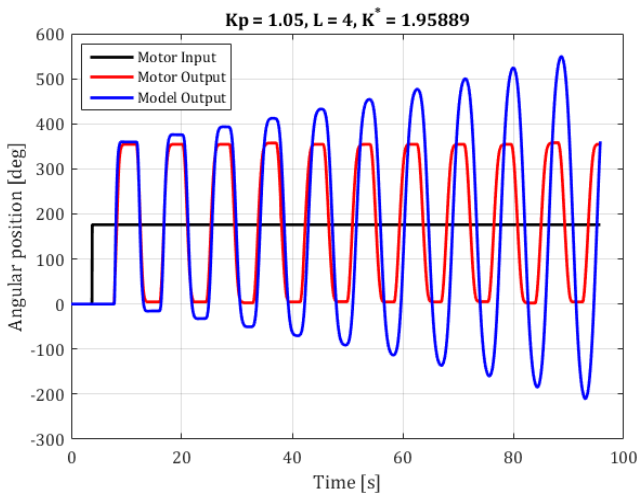


Figure 14. The simulation-real system results for  $k_p=1.05$  and  $L_1+L_2=4$  seconds.

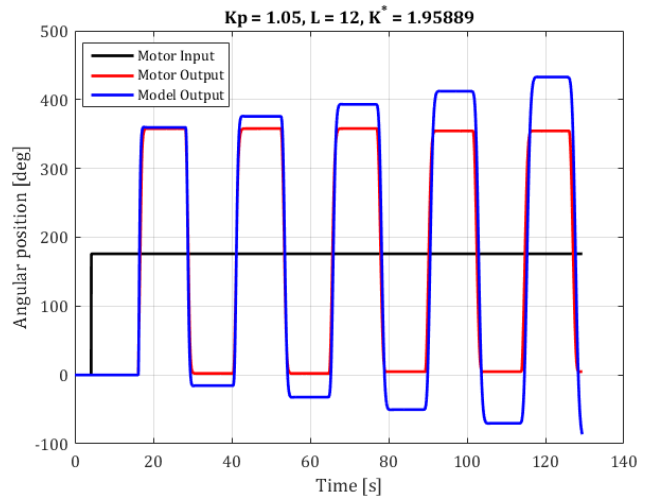


Figure 15. The simulation-real system results for  $k_p=1.05$  and  $L_1+L_2=12$  seconds.

## 5. Conclusions

In this paper a servomechanism subject to teleoperation is considered. The teleoperation can possess huge time-delays and these delays may vary operation to operation, which makes developing efficient control strategies extremely difficult. Since the stability is the first and the foremost important criterion for all of the control systems, the robust stabilization of such servomechanisms with respect to huge and varying time-delays considered. To achieve this goal, an efficient method in the literature is employed, and such kind of a mechanism is robustly stabilized in both simulations and in a real environment.

Since the interested controller is P type only, the robust stabilization problem is the only focused one in this paper. This can be thought somewhat as a drawback of the methodology. To improve the other performance criterion/criteria along with the robust stabilization problem, more complicated controller structures such as PD and lead/lag are planned to be taken account in the possible future publications.

## References

- [1] Hokayen P.F. & Spong, M.W., Bilateral teleoperation: A historical survey, *Automatica*, Vol.42, 2006, pp. 2035-2057.
- [2] Cui J. et. al. A Review of Teleoperation System Control, Proc. FCRAR, Boca Raton, Florida, USA 2003.
- [3] Alfi A. & Farrokhi, F., A Simple Structure for Bilateral Transparent Teleoperation Systems With Time Delay, *Journal of Dynamic Systems Measurement and Control*, Vol. 130 Number 4, 2008.
- [4] Arcara P. & Melchiorri, C., Control Schemes for Teleoperation with time delay: A comparative study, *Robotics and Autonomous Systems*, Vol 38, 2002, pp. 49-64.
- [5] Al-Mutairi, N.B. Adaptive Fuzzy Modulation for Networked PI Control Systems, Ph.D. Dissertation, Dept. Elec. and Comp. Eng., North Carolina State University, 2002.
- [6] Alfi, A., Bakhshi, A., Yousefi, M. & Talebi, H.A. Design and Implementation of Robust-Fixed Structure Controller for Telerobotic Systems, *J. Intell Robot Syst*, 2016. doi: 10.1007/s10846-016-0335-2.
- [7] Slawinski, E., Mut V. & Santiago D., PD-like controller for delayed bilateral teleoperation of wheeled robots

International Journal of Control, 2016, doi: 10.1080/00207179.2016.1144234.

- [8] Silva, G.J., Datta, A. & Bhattacharyya, S.P. PID Controllers for Time-Delay Systems, Birkhäuser: Boston, 2005.
- [9] Walton M., & Marshall, J.E., Direct Method for TDS Stability Analysis, IEE Proceedings – Control Theory & Applications, Vol. 134, Number 2, pp. 101 – 107, 1987.
- [10] Nesimioglu, B.S. & Soylemez, M.T., A simple derivation of all stabilizing proportional controllers for first order time-delay systems, Asian Journal of Control, Vol. 14, Number 2, 2012, pp. 598-604.
- [11] Olgac N., & Sipahi, R., A Practical Method for Analyzing the Stability of Neutral Type LTI-Time Delayed Systems, Automatica, Vol. 40, 2003 pp. 847 – 853.
- [12] Hohenbichler, N., All Stabilizing PID Controllers for Time Delay Systems, Automatica, Vol. 45, Number 11, 2009, pp. 2678 – 2684.
- [13] Lee, B.N., Wang Q.G. & Lee, T.H., Development of D-decomposition method for computing stabilizing gain ranges for general delay systems, Journal of Process Control, Vol. 25, 2015, pp. 94-104.
- [14] Nesimioglu, B.S. & Soylemez, M.T. All- Stabilizing Proportional Controllers for First- Order Bi- Proper Systems with Time Delay: An Analytical Derivation, Asian Journal of Control, Vol.18 Number 6, 2016, pp. 2203-2220.
- [15] Wang, D.J. A PID Controller Set of Guaranteeing Stability and Gain and Phase Margins for Time-Delay Systems, Journal of Process Control, Vol. 22, 2012, pp. 1298 – 1306.
- [16] Nesimioglu, B.S, Yilmaz, S., & Dincel E., Robust stabilization of a servomechanism with respect to time-delay, International Conference on Advanced Technology & Sciences (ICAT), 1-3 September 2016, Turkey, Konya.
- [17] Le, B.N., Wang, Q.G., Lee, T.H. & Nie, Z. On computation of stabilizing loop gain and delay ranges for bi-proper delay systems, ISA Transactions, Vol.53, pp. 1705 – 1715, 2014.
- [18] Nesimioglu, B.S. & Soylemez, M.T., Calculation of All Gains Providing Time-Delay Independent Stability via Root Locus, Int. Conf. on Control Decision and Information Technologies (CoDIT), Metz, France, pp. 566-571, 2014.
- [19] Thowsen A. Delay independent stability of linear systems, IEE Proceedings D-Control Theory and Applications, Vol. 129, Number 3, 1982 pp. 73-75.
- [20] Ergenc, A.F., A New Method for Delay-Independent Stability of Time-Delayed Systems, 9<sup>th</sup> IFAC Workshop on Time Delay Systems, Published by Elsevier Ltd., 7-9 June 2010, Czech Republic, Prague.
- [21] Michiels, W., & Niculescu S. I., Characterization of Delay-Independent Stability and Delay Interference Phenomena, SIAM Journal of Control and Optimization, Vol.45, Number 6, 2007, pp. 2138-2155.
- [22] Dynamixel MX 106T User's Manual, [http://support.robotis.com/en/product/dynamixel/mx\\_series/mx-106.htm](http://support.robotis.com/en/product/dynamixel/mx_series/mx-106.htm)
- [23] Garg, D.P. & Kumar, M. Optimization techniques applied to multiple manipulators for path planning and torque minimization, Engineering Applications of Artificial Intelligence, Vol. 15, 2002 pp. 241-252.