

Remote Control of Mechatronic Systems over Communication Networks*

Otto Roesch and Hubert Roth
Institute of Automatic Control Engineering
University Siegen
Hoelderlinstr. 3, 57068 Siegen, GERMANY
{otto.roesch, hubert.roth}@uni-siegen.de

Silviu-Iulian Niculescu
HEUDIASYC, UMR CNRS 6599
Université de Technologie de Compiègne
BP 20529, 60205 Compiègne, cedex, FRANCE
niculescu@hds.utc.fr

Abstract—This paper presents different techniques for the remote control of mechatronic systems over communication networks. The first part is devoted to a virtual laboratory for educational purposes. This system can be activated at distance, but the control loop itself runs locally. The second part of the paper describes a new approach for the control of dynamical systems with a stochastically varying delay in the input, caused by the signal transmission over the network. The delay is described by a probability distribution model, and some simple control laws are proposed. Furthermore, the properties (asymptotic stability, sensibility mechanism with respect to parameters' variations) are analyzed for the corresponding closed-loop system. An illustrative example (simplified helicopter model) ends the paper.

Index Terms—Virtual labs, network control, distributed delay, gamma distribution.

I. INTRODUCTION

The presence of time delays in industrial processes is an important phenomenon and must be considered in the stability analysis and controller design [5]. Without any loss of generality, any data communication process includes transmission delays, which can induce bad performances for the overall schemes if their effects are not completely understood, and taken in consideration. The delays can be classified as constant or time-varying, bounded or unbounded, distributed or not, deterministic or stochastic. Control of time-delay systems is an active research area in the last decade (see, for instance [3], [1], [9] and the references therein), and the stability analysis of the corresponding closed-loop system is not an easy task, mainly to infinite-dimensional character of the problem (infinite number of poles). Some simplifications of the problem can be done if one assumes that the stability property is *delay independent* (see, e.g. [3], [1], [9]), but such an hypothesis is not realistic in the case of communication networks.

In this paper, signal delays due to data-transmission are considered, which could occur e.g. by large/long bus-system or by the use of the internet (with TCP/IP or UDP as protocols). There are different ways to overcome this problem. One

of the strategies is represented by a *remote control* mechanism (control loop runs locally). This is nowadays often used in the engineering education with virtual laboratories, as briefly presented in section II.

The other approach is to use the communication delays in the definition of the control law. Whereas such an approach seems natural, the problem is largely more complicated since the delays are *time varying*, and the time-dependence is very complicated, and it depends on a lot of network parameters, as for example the network load and the available bandwidth, to cite only a few. Furthermore, if the controller and the plant are spatially separated, the delays occur two times in the control loop, once from the plant to the controller (forward direction), and secondly from the controller back to the plant (backward direction), see Fig. 1. Thus, the closed-loop system will include several delays (Forward Trip Time, Backward Trip Time, but also the Round Trip Time) characterizing all possible communication delays between the corresponding "nodes" (controller, plant). In conclusion, the stability analysis becomes more complicated, as seen in section III. The model used to describe the communication delay is represented by some stochastic γ -distribution and the control law is defined by some simple P (Proportional) or PD (Proportional-Derivative) law.

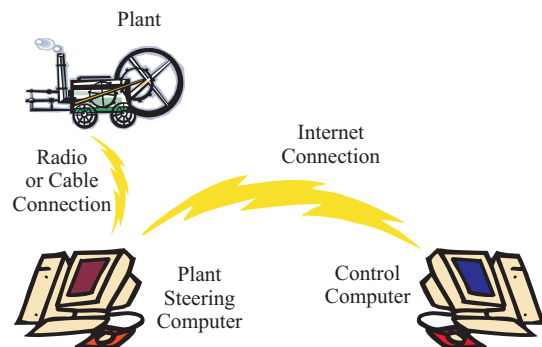


Fig. 1. Schematic drawing of a remote control setup, where time delay occurs in the Internet connection.

*Work was (partially) supported through a European Community Marie Curie Fellowship and in the framework of the CTS, contract number: HPMT-CT-2001-00278

II. VIRTUAL LABORATORIES

Virtual laboratories entered the engineering education, where real laboratory hardware experiments are getting controlled over the internet. Because of sharing resources and being time independent for the students, such an investigation is made. New internet programming techniques offer a simple and easy way to put real hardware onto the internet.

The experiment explained here is a test-platform for vibration damping and high-precision positioning of a flexible structure. The aim of this online experiment is to study the identification and control of the mechanical vibration in flexible structures. The students have to identify the system dynamics and test different control strategies. Finally, they have to evaluate the quality of the control algorithms. In order to provide a handy and convenient user-oriented laboratory experiment, a complete educational unit has to be provided. To help students to get into the appropriate field, links to general tutorials as well as a set of documentation to the particular experiment is accessible over the internet. Before the student can execute the experiment, a knowledge check has to be passed. This consists of online questions like multiple-choice.

The access to the remote experiment is done by an e-Learning web-portal. This includes a booking system, in which the registered students can reserve time slices for the online laboratory experiments.

A. Swinging Rod Hardware Setup

In the design of spacecraft, aircraft, and even building, the flexibility of structural elements is of concern. This is especially pronounced in space structures and aircrafts where large size coupled with lightweight materials emphasize structural flexibility. The focus of the experiment¹ is an aluminium rod, suspended on a motor. Its sensors and actuators can be controlled via the Internet. The rod's length coupled with its small cross section makes the system quite flexible. The purpose of this lab is to provide a test platform for students to analyze structural vibrations, model system behavior, and design controllers. They can implement their work on an actual physical system via the Internet or locally in the laboratory.

The main feature of the "Swinging Rod" test facility is shown in Figure 2. It uses a 1.6 meter long aluminium rod with a small cross section (4x10 mm) as the flexible component [7]. An excitement of the rod is possible in one orthogonal direction at the upper end of the rod. The system is equipped with a DC-Motor as an actuator, used for disturbing and afterwards to control the rods movement. The following two sensors are used for the vibration measurement [4]:

¹proposed at the Control Engineering Department in the University of Siegen

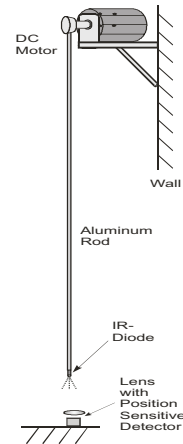


Fig. 2. Swinging Rod Schematic Drawing.

- A PSD (Position Sensitive Detector) sensor is used for the position measurement at the tip of the rod. It is an opto-electronic device which converts an incident light spot into continuous position data. A lens is fixed at top of this photo-element; by this an extension of the working range of 20 centimeters in diameter can be achieved. The PSD is located on the floor, perpendicular to the rod. An infrared diode, fixed at the tip of the rod, beams down towards the sensor.
- The second sensor measures the angular position of the motor shaft, therefore a built in 4-quadrant encoder inside the motor is used.

An analogue amplifier drives the motor and delivers the necessary current. In comparison to PWM (Puls Width Modulation) amplifiers, no high frequency noise inside the system can disturb the very small analogue sensor signal from the PSD with this analogue amplifier.

B. Swinging Rod Software Setup

The software is divided into two main parts, one is the software to communicate directly with the hardware over a data acquisition board, and the other program establishes a remote control over the internet. The control software to

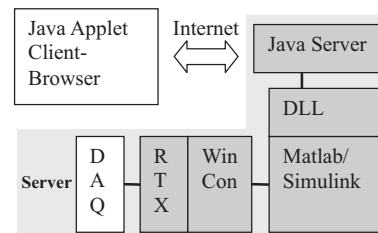


Fig. 3. Swinging Rod Software Setup and Signal Flow.

communicate with the hardware is divided into four sub-modules, see Figure 3.

- The Java Server handles on one side the communication to the client (student) over the internet, and on the other side to access the hardware through a DLL library.
- The DLL library is an interface between the Java server application and the Matlab program, to enable a communication between these applications.
- The real-time code is firstly created by a Matlab/Simulink model. During the experiment, Matlab starts and stops the real-time process and saves the sampled data through the WinCon application.
- The WinCon software controls the data acquisition board of the real-time process (both from the Quanser [10]) over the RTX (real time kernel) from VentureCom.

The remote control software is implemented in Java, whereas on the server-side a Java application and on the client (student) side a Java-Applet is getting executed. The Java-Applet is integrated into a web page and accessible inside a web browser without any special software. The control signals from the client to the server and the feedback information backwards get transferred over a socket connection [14]. The sensor data are transferred to the student in form of a graph after the particular test-run is finished. The rod is also observed by a web camera to get a better expression of the oscillating rod and to see how well the controller works.

The real-time control loop is running here locally on a real time hardware platform. The remote control is only used to transfer parameters and to start the system. No time delay difficulties occur in this control loop. The next section will explain a remote controllable laboratory experiment with a spatially distributed controller and hardware plant over the network, where stochastic delays occur.

III. CONTROL OVER THE NETWORK

A simplified control scheme is shown in Figure 4, where G_c denotes the controller, G_{heli} the plant, r the desired input and y the plant output. The time delay occurs from the controller to the plant and again from the plant backward to the control unit through some network connections.

One of the problems to be considered is represented by the delay models of the corresponding network connections. The simplest model is represented by a time-delay block, where the delay is assumed *constant*, but *uncertain* inside the prescribed, and known interval $[\tau_{min}, \tau_{max})$, where $\tau_{min} > 0$. In such a case, the analysis of the closed-loop system can be done using frequency- and time-domain methods similar to the ones proposed by [3] and [8]. Thus, for example, one (necessary and sufficient) delay margin can be explicitly computed using matrix pencil methods. However, such a model is quite restrictive since it does not take into account the real influence of the network environment (bandwidth available; number of users; control mechanism strategies).

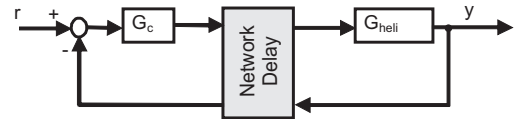


Fig. 4. Closed loop system with time-delays in the communication channels.

A second idea is to consider the delay as a *bounded time varying* function and to use some time-domain methods to perform the stability analysis, such an approach was considered by [11], for which a Lyapunov-Razumikhin function was used. The controller's gains have been computed using some genetic algorithm strategies. Furthermore some limitations on the approach have been discussed in [11].

One of the ways to reduce the conservatism of the method is to derive a more accurate model for representing the delay in the network. In this sense, it seems (see, for instance, section III-A) that a gamma distribution delay is better adapted for the control problem considered here. The proposed controller approach uses a probability model of the time delay in the network to estimate the data package delay, and the closed-loop system reduces to some particular integro-differential equations. Next, the analysis will be performed with respect to some controller parameters. This approach is applied on some helicopter laboratory experiment, controlled over Internet. To the best of the authors' knowledge, such an approach for controlling helicopter elevation over networks was not considered before.

A. Gamma Model of the Network Delay

Roughly speaking, the delay in the network is the sum of a constant component and a "dynamic" (time-varying) component: while the constant term depends on the signal propagation time (from some source to the corresponding destination), which is assumed to be the absolute minimum propagation delay value, occurring during the overall measurement, the varying delay occurs due to data collisions and routing problems on the network and is considered to have a stochastic nature.

For the delay analysis, we need to measure the round trip time between two computers in the network. In Figure 5 are some measurements shown, which describe the connection between a computer in Compiègne/France to one in Ohio/USA. The minimum delay is always larger than 60 ms, which results in the signal propagation time, and is assumed to be the minimum delay which could occur during this overall connection. The maximum delay goes up to 333 ms for just a few peaks, while the average delay is 101 ms.

The next step is now to describe this delay measurement in a probability density sense. Mathematical models for describing the time delay in a network are mainly used in internet telephony, also known as VoIP (Voice over IP). In order to deliver a good quality of service to the people who

use the Internet connection to make phone calls, signal delays must be considered carefully. In general, the network delay is modelled as a gamma distribution model or as a power approximation, as in [2]. In the sequel, we shall consider, and focus on a gamma distribution model.

Figure 6 shows the probability density distribution of the measured signal delay from Figure 5. In addition, the estimated mathematical model of the gamma distribution is overlapped. It can be seen that the main (highest) peak occurs for both curves (more or less) at the same time delay. The parameters of the identified gamma density distribution are $\alpha_{med} = 5$ and $\beta_{med} = 0.020$, and within a 95% confidence bound we get $3 < \alpha < 7$ and $0.012 < \beta < 0.032$ for $\alpha \in N, \beta \in R$. The gamma probability density function has the following form

$$g(t|a, b) = \frac{t^{\alpha-1} \cdot e^{-t/\beta}}{\beta^\alpha \cdot \Gamma(\alpha)}. \quad (1)$$

The gamma function is defined as follows

$$\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1} dt. \quad (2)$$

The dotted line shows in addition the cumulative distribution function. According to [13], a gamma distribution model for time-delays in the network “fits” relatively well to the real behavior. The delay behavior changes in general for each new established connection, therefore different parameters for the gamma model should be used. To solve this problem, a software module was developed by [13], to perform a delay measurement and get directly the parameters for the corresponding gamma distribution model. These parameters are computed and explicitly given in some intervals. Note that this software could be used for a real application directly before the remote control starts, to calculate the gamma function parameters in order to get further the controller parameters.

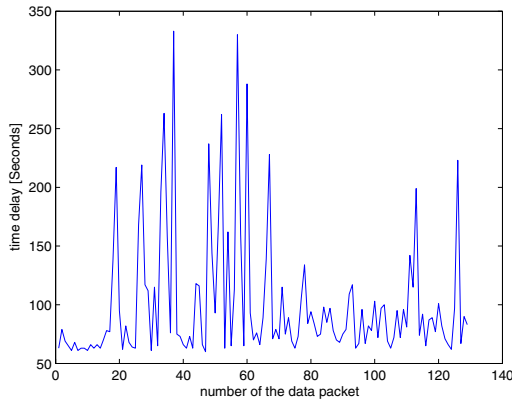


Fig. 5. Time delay measurement between two computers in the network.

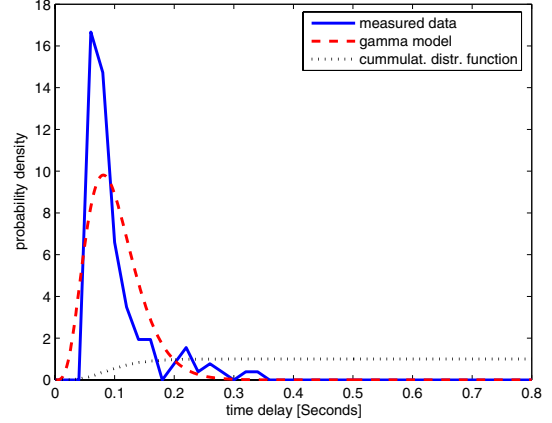


Fig. 6. Gamma distribution model of the network delay.

B. Controller Design

The design of a stabilizing controller for a system with a stochastically varying time delay is in general a challenge. As briefly presented in the previous section, the gamma distribution can be used to model the delay in networks. In the sequel, we focus on designing a controller for a dynamical system in closed-loop under some *network uncertain* evolution that is a “trade-off” between network environment and system’s behavior.

If we consider a network model according to Figure 7, it can be described as in the sequel, where $g(t)$ denotes the (time-varying) network delay, which follows the distribution law presented in section III-A. Such a distributed delay can be used to design a controller, where the following equation holds.

$$FTT : y_1(t) = \int_{-\infty}^t g_1(t - \theta - \tau_{prop,FTT}) u_1(\theta) d\theta, \quad (3)$$

$$BTT : y_2(t) = \int_{-\infty}^t g_2(t - \theta - \tau_{prop,BTT}) u_2(\theta) d\theta. \quad (4)$$

The network delay can be partitioned as FTT (Forward Trip Time) and BTT (Backward Trip Time). The gamma distribution presented above gives the total RTT (Round Trip Time). The network delay model is described by g_1 and g_2 , the network outputs are y_1 and y_2 , the network inputs are u_1 and u_2 , according to Figure 7. The signal propagation delay, when no charge is in the network, is given by $\tau_{prop,BTT}$ and $\tau_{prop,FTT}$ (no collisions and no routing problems occur). For simplification purposes, we assume that $\tau_{prop,BTT} \cong 0$ and $\tau_{prop,FTT} \cong 0$. One assumes similar behaviors on FTT and BTT, which allow to define symmetric channels, so we can assume to have $FTT(\alpha_1, \beta)$ and $BTT(\alpha - \alpha_1, \beta)$ according to the distribution derived in III-A, with $\alpha_1 \in N$.

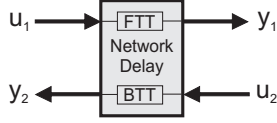


Fig. 7. Network delay, partitioned in FTT and BTT.

Consider now a dynamical system given by $G_p(s) = \frac{P(s)}{Q(s)}$. The Laplace transform of the gamma probability density function of equation (1) is derived for FTT and BTT as:

$$FTT_{\beta}^{\alpha_1}(s) = \frac{1}{(\beta s + 1)^{\alpha_1}}, \quad BTT_{\beta}^{\alpha - \alpha_1}(s) = \frac{1}{(\beta s + 1)^{\alpha - \alpha_1}}.$$

Assuming a controller of the general form $G_c(s) = \frac{N(s)}{D(s)}$, then we get the following closed loop transfer function

$$G_{cl} = \frac{G_c G_p FTT_{\beta}^{\alpha_1}}{1 + G_c G_p FTT_{\beta}^{\alpha_1} BTT_{\beta}^{\alpha - \alpha_1}}. \quad (5)$$

Thus, the complete closed-loop transfer function becomes:

$$G_{cl} = \frac{N(s) P(s) (\beta s + 1)^{\alpha - \alpha_1}}{D(s) Q(s) (\beta s + 1)^{\alpha} + N(s) P(s)}. \quad (6)$$

Among the ideas that can be used to construct a controller, we consider the standard *pole placement approach*.

C. Sensitivity Computing

It is now of interest to know the way the poles of the closed-loop system move by varying the gamma parameters. In the sequel, the average delay defined by $\tau = \alpha \cdot \beta$ is used as a parameter for defining the sensitivity of the roots $\lambda = \lambda(\tau)$, for all $\tau \in [\underline{\tau}, \bar{\tau}]$ of the corresponding characteristic function. For the sake of simplicity, assume that the dominant (“rightmost”) roots are simple, and assume also that the function $\lambda(\tau)$ is differentiable. Under these assumptions, define now $R(\tau) =$:

$$\frac{d}{d\tau} \left[D(\lambda(\tau)) Q(\lambda(\tau)) \left(\frac{\tau}{\alpha} \lambda(\tau) + 1 \right)^{\alpha} + N(\lambda(\tau)) P(\lambda(\tau)) \right]. \quad (7)$$

Since we consider only the stability issues, the main problem will be to analyze the cases when dominant (“rightmost”) roots cross the imaginary axis, *if any*, when τ is varying inside the corresponding interval $[\underline{\tau}, \bar{\tau}]$. More explicitly, we shall focus on the *crossing direction*, which is given by:

$$\text{sgn} \left\{ \Re e [R(s)]|_{s=j\omega} \right\}, \quad (8)$$

with the rule that the crossing is towards stability (instability) if the value of (8) is $-1(+1)$ (see also the example below).

D. Illustrative Example

The system under consideration is represented by a Helicopter model [10], see Fig. 8, and it consists of a fixed base, on which a rotary arm is mounted. The arm carries the helicopter body on one end, and a counterweight on the other.

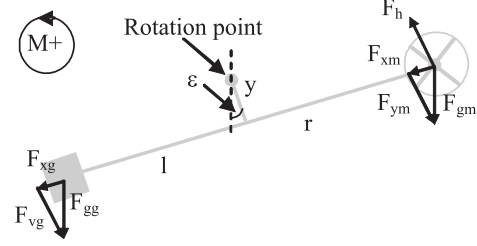


Fig. 8. Schematic drawing of the helicopter model.

The arm can make an elevation motion around the angle epsilon. An encoder mounted on the axis allows measuring the elevation angle of the arm. The helicopter body is fixed at one end of the arm and is free to elevate in a certain range. Two motors with propellers mounted on the helicopter arm can generate a force proportional to the voltage applied to the motors. The force generated by the propellers cause the helicopter body to lift off the ground. The purpose of the counterweight is to reduce the power requirements on the motors around equilibrium. The corresponding nonlinear mathematical model is:

$$J_{ges} \cdot \ddot{\epsilon} = -g \cdot y \cdot (M + m) \cdot \sin(\epsilon) + 2 \cdot k_t \cdot r \cdot v(t), \quad (9)$$

where k_t, g represent the motor, and gravity constants, m and M denote the mass of the helicopter blades inclusive motors and the fixing devices, and the counterweight, respectively, $v(t)$ the corresponding voltage, and J_{ges} the moment of inertia around the rotating point (see also Fig. 8). In order to simplify the analysis, the model will be linearized around the quiescent point. For $\epsilon = 0$, one gets:

$$J_{ges} \cdot \ddot{\epsilon} = -g \cdot y \cdot (M + m) \cdot \epsilon + 2 \cdot k_t \cdot r \cdot v, \quad (10)$$

and finally, after the damping factor identification, the helicopter transfer function results in [12]:

$$G(s) = \frac{K_p}{s^2 + d s + c} = \frac{0.2607}{s^2 + 0.07441 s + 2.904} \quad (11)$$

In the sequel, we shall use the plant model given by (11).

1) *Controller Design*: Consider now a simple PD-controller for improving the system response of the above helicopter laboratory experiment, control law given by $G_c = (K_1 s + K_2)$. Next we analyze the closed-loop stability with respect to the controller gains K_1, K_2 , that is the distribution of roots of the following characteristic equation:

$$Q(s, \alpha, \beta) = (s^2 + d \cdot s + c) (\beta s + 1)^{\alpha} + (K_1 s + K_2) K_p, \quad (12)$$

where α and β are the gamma probability density function parameters. We assume first that $\alpha = \alpha_{med}$ and $\beta = \beta_{med}$ for defining appropriate controller gain parameters. Next, we increase successively the control parameters K_1 and K_2 , while the real part of the dominant (“rightmost”) poles is regarded, as we can see in Fig. 9, and the controller gains

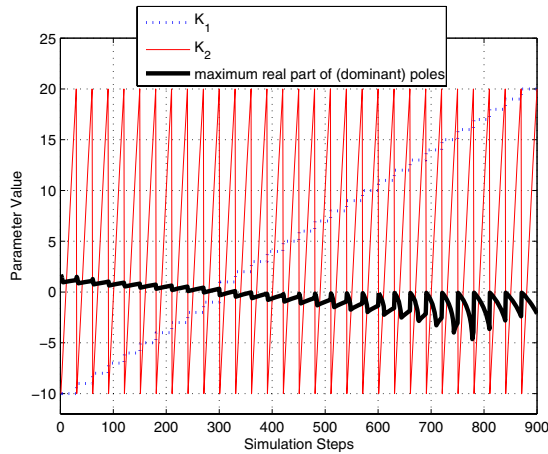


Fig. 9. Maximum real part of the (dominant) poles by varying the controller parameters K_1 and K_2 .

can be chosen, for example, by using the distance to the imaginary axis as criterion. The optimization approach for getting better controller parameters can be improved. One of the ideas is to use a more advanced search like a genetic algorithm.

2) *Sensitivity computing*: The next step is now to look how stable the corresponding closed-loop system is for varying time delays. In section III-A we gave a confidence interval for the gamma distribution parameters. It is further of interest how the roots change while varying α and β inside this given intervals. For a better understanding of the gamma probability density function parameters, we look to the average delay value, which is defined as $\tau = \alpha \cdot \beta$. Inside the 95% confidence interval, the average delay varies in $0.036 < \tau < 0.256$.

Take for instance $K_1 = 16.5$, and $K_2 = 19.5$ as gain parameters, corresponding to the step 780 in fig.9, and let us analyze the pole locations in the complex s-domain for different gamma parameters. Fig.10 shows the way the roots of the corresponding characteristic equation are moving by varying β inside the given interval for a fixed α to $\alpha_{med} = 5$, which means by varying τ inside the interval $[0.036, 0.256]$. It is easy to see that all the roots move towards the imaginary axis by increasing the average delay value. However, the real part of the dominant (“rightmost”) roots $\lambda_d(\tau)$ stays in the left half plane, and we have: $Re(\lambda_d(\tau)) < -0.3563$.

IV. CONCLUSION

The remote controllable experiment presented in the context of the virtual laboratory in the first part of the paper is a simple approach for a remote controllable system, where the control loop runs local. But if a distributed control has to be achieved, the stochastic nature of the time-delay in the network must be considered. Simulation results showed a stable behavior of the corresponding closed-loop system.

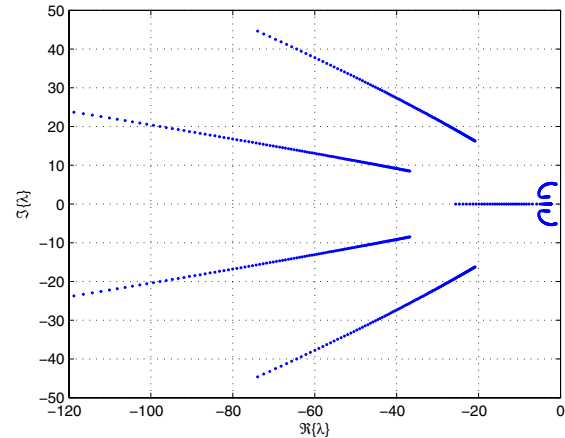


Fig. 10. The sensitivity graph shows the root locations by varying the gamma parameter β for a fixed α to $\alpha_{med} = 5$.

The further work is to implement the above controller design on the real system with a remote control computer, and to analyze its behavior in the real environment.

REFERENCES

- [1] Boukas, E.-K., and Liu, Z.-K.: *Deterministic and Stochastic Time Delay Systems* (Birkhauser: Boston, 2002).
- [2] Broido, A., Basic E., Claffy, Kc.: www.caida.org/~broido, 2002.
- [3] Gu, K., Kharitonov, V. L., and Chen, J.: *Stability of Time Delay Systems* (Birkhauser: Boston, 2003).
- [4] Irwin, R. D., T. M. Adami, H. Roth, G. Muenst, and O. Roesch: “Sensor and Control Concepts for the Internet-Based Flexlab Experiment,” in *Workshop on Internet Based Control Education, IBCE’01*, Madrid - Spain, 231 - 234, 2001.
- [5] Kolmanovskii, V.B., and Nosov, V.R.: *Stability of Functional Differential Equations* (Academic Press: New York, 1986).
- [6] Marshall J.E., H. Gorecki, A. Korytowski, and K. Walton: *Time Delay Systems, Stability and Performance Criteria with Applications* (Ellis Horwood: 1992).
- [7] Muenst, G., Roth, H.: “Sensing and Control for Flexible Structures,” in *Workshop on Tele Education in Mechatronics Based on Virtual Laboratories*, Weingarten, Germany. ISBN 3-9255359-003, p. 60., 2001
- [8] Niculescu S.-I., and K. Gu (Eds.) *Advances in Time Delay Systems* (Springer, LNCSE, vol. 38, 2004).
- [9] Niculescu S.-I.: *Delay Effects on Stability. A Robust Control Approach* (Springer, LNCIS, vol. 269, 2001)
- [10] Quanser, *3D Helicopter Experiment with DAQ Board and WinCon Software*, www.quanser.com, 2004.
- [11] Roesch, O., Roth, H., and Iqbal, A.: “Extended Stability margins on controller Design for Nonlinear Input Delay Systems,” *IFAC World Congress*, Prague, July 2005.
- [12] Roesch, O.: “Control of Dynamical Systems with Stochastic Time-Delay in the Communication Channel,” PhD. Thesis in preparation, Siegen, Germany, 2005.
- [13] Salza, S., Draoli, M., Gaibisso, C., Palma, A.L., and Puccinelli, R.: “Methods and Tools for the Objective Evaluation of Voice-over-IP Communications,” in *INET 2000, The 10th Annual Internet Society Conference*, Yokohama, Japan, 2000.
- [14] Schilling, K.: “Remote Sensor Data Acquisition and Control,” *Smart Systems and Devices*, Tunisia, 2001.