

Publication [P4]

M. Pohjola, L. Eriksson, H. Koivo, "Tuning of PID Controllers for Networked Control Systems", in *Proc. The 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON'06)*, Paris, France, November 6-10, 2006, 6 p.

© 2006 IEEE. Reprinted, with permission.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Helsinki University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Tuning of PID Controllers for Networked Control Systems

Mikael Pohjola
Control Engineering Laboratory
Helsinki University of Technology
P.O. Box 5500, FI-02015 TKK
Espoo, Finland
mikael.pohjola@tkk.fi

Lasse Eriksson
Control Engineering Laboratory
Helsinki University of Technology
P.O. Box 5500, FI-02015 TKK
Espoo, Finland
lasse.eriksson@tkk.fi

Heikki Koivo
Control Engineering Laboratory
Helsinki University of Technology
P.O. Box 5500, FI-02015 TKK
Espoo, Finland
heikki.koivo@tkk.fi

Abstract - As the use of computer networks grows rapidly, applications, such as networked control systems (NCS), have emerged. NCS suffer from varying time-delays that bring new problems to the control loop. This paper discusses tuning of a discrete-time PID controller for NCS. A discrete-time PID controller is selected as a controller, because it is widely used in industry. Minimising of a cost function is a powerful tool in control design. Here the optimisation technique is used to find the best parameters of a PID controller, when the system has time-varying or stochastic delays. The cost variance is used as a robustness measure of the tuning. Comparisons to the Ziegler-Nichols and IMC tuning methods are given in a case where the process and the controller are distributed over the Internet. The tunings done in simulations are further verified on a real process. Despite of the simplicity of the PID controller adequate performance is still achieved and it is shown that the optimisation tuning is robust in the presence of a varying time-delay.

I. INTRODUCTION

Control theory is extending to new areas. The idea of using closed-loop control with COTS (Commercial off-the-shelf) hardware using standard Ethernet networks, or even over the Internet, has emerged, and this direction is now researched widely. The traditional local control loop is expected to expand to tomorrow's control over large communication networks [1]. The benefits in this approach are that cheap COTS hardware can be used. With wireless control networks cabling costs are saved, flexibility in positioning the sensors is gained, as no cables need to reach the sensor and it is easy to add devices to the network. Control over Internet is the ultimate case with no boundaries between the factory floor and the world. In the near future we will probably see control systems built with dedicated Ethernet hardware.

The problem is that the network induces varying time-delays into the control loop, which have to be taken into account in the control design. Conventional control design usually assumes constant delays and thus new methods must be developed.

The results so far include stability conditions for delays smaller than the sampling time and stability under packet dropout [2]. It is recognised that as the delay increases the bookkeeping gets increasingly harder and the systems are more difficult to analyse analytically.

Model predictive controllers have been proposed to control varying time-delay systems [3], but they are complicated to implement in practice. Dynamic programming is another approach [4]. Still simple controllers and tuning rules for varying time-delay systems are needed for applying the methods in practice. The problem is difficult because the time-delay in NCS often has a stochastic nature and it is therefore complicated to approach analytically. This paper presents a controller optimisation tuning method using simulation for varying

time-delay systems, especially for NCS.

The simulation results are further verified in practice on a real process. The MoCoNet (Monitoring and Controlling Laboratory Processes over Internet) system developed in the Control Engineering Laboratory at Helsinki University of Technology is used [5].

The paper is arranged as follows. First networked control systems are introduced. A simulation model of the network is described in Section III. In Section IV the optimisation tuning method is presented. In the following sections the method is applied on a discrete-time PID controller in simulations and tuning results are given as a function of sampling time. The optimisation tuning is compared to the Ziegler-Nichols and IMC tuning methods. The tunings are further demonstrated on a real process in Section VI. Finally conclusions are given.

II. NETWORKED CONTROL SYSTEMS

In distributed NCS the controller and the process are physically separate and connected with a network, as depicted in Fig. 1. The measurement and control signals are sent over the network. There are two network induced varying time-delays in the control loop: τ_{ca} and τ_{sc} [6].

The network can be an industrial-, Ethernet or wireless network. The network considered in this paper is the Internet. The Internet is chosen because it is a difficult environment for a control system running on COTS equipment.

A discrete-time PID controller structure is selected to control the system as it is well understood and intuitive to tune. It is also used extensively in the industry [7] and there is a benefit of using a well-known and trusted controller in future applications. This is a simple and desirable solution, since it does not require new control algorithms, just new tunings to cope with the varying time-delay. With a more complex controller the delay could be explicitly taken into account in the control algorithm, and better results could be possible. With a PID controller adequate performance is nevertheless achieved. An investigation of several types of PID controllers can be found in [8].

The controller is tuned with the optimisation method presented in Section IV. Because of network bandwidth arbitrary high sampling rates cannot be used. There is a trade-off: increased utilisation and congestion of the

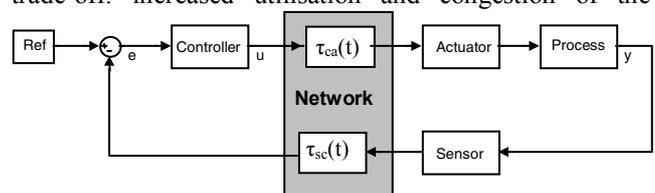


Fig. 1. Fully-distributed NCS where controller, actuator and sensor are distributed and connected with a network.

network increases the delay times. The controller is tuned over a range of sampling times, to investigate the impact of different sampling times on the control loop.

III. NETWORK DELAY AND PACKET LOSS

For investigation of the impact on network induced delay in the control loop a network simulator is chosen to enable one to specify the network properties, such as delay and packet loss, and to replicate the delay for comparisons. Research can thus be conducted in a controlled fashion.

A system, called MoCoNet, has been developed in the Control Engineering Laboratory at Helsinki University of Technology for studying real processes in a networked control system setting [5]. The system is used in Section 6 to verify the results achieved in simulations.

Measurement and control signals are transmitted through the network simulator (NetSim) as indicated in Fig. 2 to form a fully distributed NCS (Fig. 1). The measurement, control and actuation are performed by the same device, but the overall system is virtually distributed.

The varying time-delay that the measurement and control signals experience when transmitted over the network is approximated by a delay distribution. This approximation holds if there is no correlation between consecutive packets sent over the network. The assumption is true if the packets are sent sufficiently far apart and the packets take up less than 10% of the capacity of the link [9]. For higher accuracy the network can be simulated with models such as TrueTime [10].

The delay distribution of the Internet has been shown to be a shifted gamma distribution [11], where the gamma distribution model was statistically verified with quantile-quantile plots. The parameters of the gamma distribution were identified with properties of the network, such as the number of hops. The gamma probability distribution function is given by

$$P(x) = \frac{\alpha}{\Gamma(n)} (\alpha x)^{n-1} e^{-\alpha x}, \quad (1)$$

where Γ is the gamma function

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

and n (the shape parameter) is the number of hops between the first and last node and $\alpha = n / T$, where T is the mean delay [11].

Every network has a shortest possible delay. The delay distribution has to be shifted by this amount to the right to get the correct minimum delay. The total network delay consists of a *static* and a *stochastic* component

$$T_{network} = T_{static} + T_{stochastic}, \quad (3)$$

where T_{static} is the minimum delay for the network and $T_{stochastic}$ the stochastic delay induced by other traffic, approximated by the gamma distribution.

The total delay from sensor to controller (τ_{sc}) is composed of the time it takes to process the measurement (τ_p), the network delay time ($\tau_{network}$) and the time the measurement has to wait to be processed at the next time step: the synchronisation delay (τ_s) [6]. The synchronisation delay arises because the controller is time-driven and works with

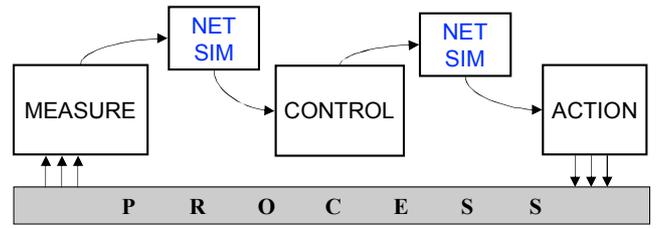


Fig. 2. Network simulator in the control loop.

a fixed sampling time. The synchronisation delay satisfies $\tau_s < h$, where h is the controller sampling time, as it rounds the total delay to the next time-step. In other words:

$$\tau_{sc} = \tau_p + \tau_{network} + \tau_s = nh, \quad n \in \mathbb{N}. \quad (4)$$

The controller to actuator delay (τ_{ca}) is composed of the same components.

The packets can also get lost. The loss is due to transmission errors and queue overflow at the routers. The packet loss is random and independent of the delay with the same assumptions as when approximating the delay with a distribution [9].

IV. OPTIMISATION BY SIMULATION TUNING

The tuning of the controller in a NCS is not straightforward because of the varying delay. Traditional tuning methods, such as pole placement, can not be used. Tuning based on optimisation can be applied and it is introduced in this section.

The tuning of a parameterised controller, such as a PID controller, is based on minimisation of an optimisation criterion, J . In this paper only PID controllers are tuned with the optimisation method, whereas the method is applicable to other parameterised controllers as well.

Well-known and used optimisation criteria are the error integrals IAE, ISE, ITAE and ITSE [7]. The ITAE (Integral Time Absolute Error) criterion is used in this paper because it gives step responses with a short settling time.

The cost function is evaluated with simulation, thus a simulation model of the process is needed. In [12] it is described how the tuning is done in practice with MATLAB/Simulink and the Optimisation Toolbox. The procedure is followed in this paper, with the modification that because of the random time-delay many step response tests are performed and the maximum cost is minimised. The cost function becomes

$$J = \max_{k=1 \dots K} \int_0^{\infty} t |e_k(t)| dt, \quad (5)$$

over K step response tests with errors $e_k(t)$. This restrains the method from optimising for a certain instance of a varying time-delay and at the same time minimising the worst case. The resulting optimal tuning will thus be robust.

The cost can also be an average of several step response costs as in [13]

$$J = \frac{1}{K} \sum_{k=1}^K \int_0^{\infty} t |e_k(t)| dt, \quad (6)$$

but in that case the tuning may allow some bad responses.

For NCS the error, e , in the cost function can be calculated as the difference between the reference and the output of the process model. The delayed output in the feedback loop (which the PID controller receives) is simply the process measurements arriving to the controller from the real process at the other end of the network.

The variance of the cost from run to run can be a measure of the robustness of the tuning. Because of the natural increase of the cost as a function of sampling time, the cost variance is normalised by the average cost to obtain a relative cost variance, which is estimated with

$$\bar{\sigma}_J^2 = \frac{1}{K-1} \sum_{k=1}^K \left(\frac{J_k - \bar{J}}{\bar{J}} \right)^2 = \frac{\sigma_J^2}{\bar{J}^2}. \quad (7)$$

J_k is the cost at run k and \bar{J} the average cost over all the runs, all K step tests evaluated with the same controller tuning. It is zero for no impact on the cost and positive as the effect of the varying time-delay affects the closed loop performance.

V. SIMULATIONS

The optimisation by simulation tuning method is applied on an example process, an electric RC-circuit, where the control objective is to make the output voltage of the circuit follow the reference voltage. The tuning results and control performances are compared to other tuning methods with simulations. The results are further verified on the real process in the next section.

The example process is modelled as a transfer function

$$G(s) = \frac{U_{out}(s)}{U_{in}(s)} = \frac{bs+1}{a_2s^2+a_1s+1} = \frac{0.4624s+1}{0.311s^2+1.228s+1}. \quad (8)$$

It is placed in a fully distributed network setting such that a varying time-delay is appended to the model at the input and the output. The network is simulated with the delay distribution and packet loss described in Section 3. The network properties are those of a typical short-range Internet connection, listed in Table 1 [5].

The controller is a discrete-time PID controller derived with the backward derivative approximation from the textbook version of the continuous PID controller

$$\begin{aligned} u(t) &= K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) = \\ &= Pe(t) + I \int_0^t e(\tau) d\tau + D \frac{de(t)}{dt} \end{aligned} \quad (9)$$

The resulting discrete-time controller with constant sampling time (h) and filtering of the derivative part:

$$u(k) = u_p(k) + u_i(k) + u_d(k)$$

$$u_p(k) = K_p e(k)$$

$$u_i(k) = u_i(k-1) + \frac{K_p h}{T_i} e(k) \quad (10)$$

$$u_d(k) = \frac{T_d}{T_d + Nh} u_d(k-1) + \frac{K_p T_d N}{T_d + Nh} (e(k) - e(k-1))$$

Table 1: Network properties, typical Internet parameters.

Hops, n	Static delay, τ_{static}	Stochastic delay, $\tau_{stochastic}$	Packet loss
3	35 ms	100 ms	1 %

In this paper the P , I and D gains of the PID controller are used. They are related to the K_p , T_i and T_d gains by

$$P = K_p, I = K_p / T_i \text{ and } D = K_p T_d \quad (11)$$

as indicated in (9). $N = 10$ is selected for the filtering constant.

A. Optimisation Tuning - Simulation Results

The optimisation of the PID controller is done with the cost function (5) over $K = 20$ unit steps. The tuning is done for sampling times ranging from 0.01 to 0.25 seconds and the optimal controller parameters are shown in Fig. 3.

The average cost (6) and relative cost variance (7) are estimated with $K = 100$ and using different realisations of the network delay compared to those used in the optimisation to avoid over-optimisation. The estimated average and maximum cost are displayed in Fig. 4 with the maximum cost from the optimisation. The cost rises roughly linearly with sampling time. The maximum cost is larger than the optimisation maximum cost. This indicates that the optimisation did not capture all the possible delay variations. A larger K should be used in optimisation, but then the optimisation time would increase. The results are nevertheless robust according to the cost variance.

The relative cost variance is shown in Fig. 5. The cost variance is low at small sampling times. It increases and has a local minimum at about $h = 0.16$ s. With higher sampling time the variance is larger. At the highest sampling times the variance decreases as the controller tuning is loose.

At $h = 0.16$ s the effective delay varies between two values: h and $2h$, and 85 % of the stochastic time-delay is smaller than the sampling time. This suggests that the PID controller works well when most of the variation of the delay is in the order of the sampling time. This supports the work that considers the delay variation to be at most one sampling time (e.g. [6]).

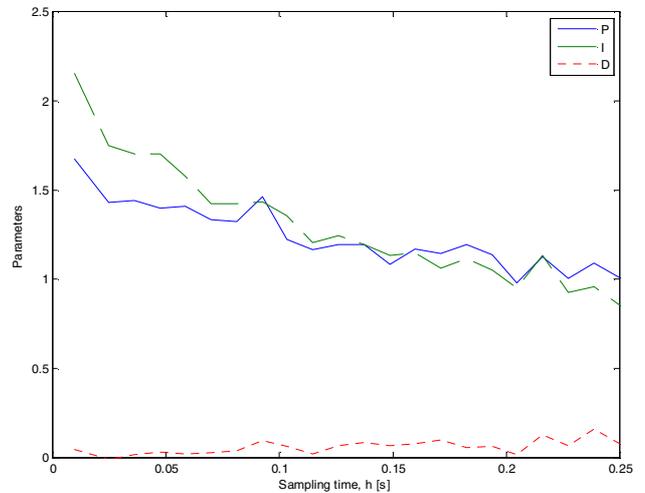


Fig. 3. Optimal PID controller parameters for example process. P, I and D as functions of sampling time, h .

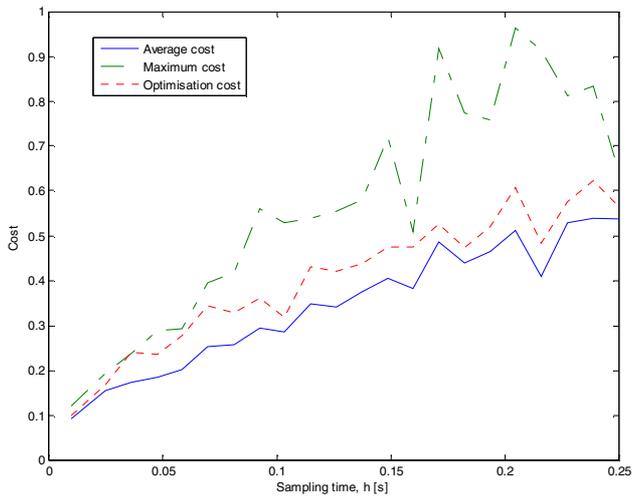


Fig. 4. Average ITAE cost, Maximum ITAE cost and Cost from optimisation as a function of sampling time, h , for example process.

In Fig. 6 some step responses are shown with different sampling times. At the shortest sampling time the system has a good response. With the sampling time of $h = 0.16$ s there is some overshoot and oscillations. This is the most preferred sampling time, as concluded in the previous paragraph. The longest sampling time results in a slower response with longer rise time and settling time, because of the synchronisation delay. The responses are generally good with any sampling time.

B. Comparison of Tuning Methods

For comparison of the optimisation method to traditional controller tuning methods, the Ziegler-Nichols (Z-N) frequency and step response test methods [7] and the IMC scheme [14] are also applied.

The Z-N tunings are done with tests on the simulation model. The obtained gains are listed in Table 2. Since the Z-N and the IMC methods are for continuous controllers the PID parameters are used as is and a small sampling time of $h = 0.025$ is used to approximate the continuous controller.

The IMC controller is obtained with

$$G_{IMC}(s) = \frac{G_m^-(s)^{-1}G_f(s)}{1 - G_m^+(s)G_f(s)} \quad (12)$$

where $G_m^-(s)$ is the invertible and $G_m^+(s)$ the non-invertible part of the model and

$$G_f(s) = \frac{1}{(\lambda s + 1)^n} \quad (13)$$

is a filter with appropriate n to make the controller proper. The controller is tuned with the parameter λ .

As the process model is invertible $G_m^-(s) = G(s)$ and the non-invertible part $G_m^+(s)$ consists only of the time-delay. The varying time-delay is first approximated with a constant delay, τ , and this is further approximated with a first order Taylor series.

$$e^{-\tau s} \approx 1 - \tau s \quad (14)$$

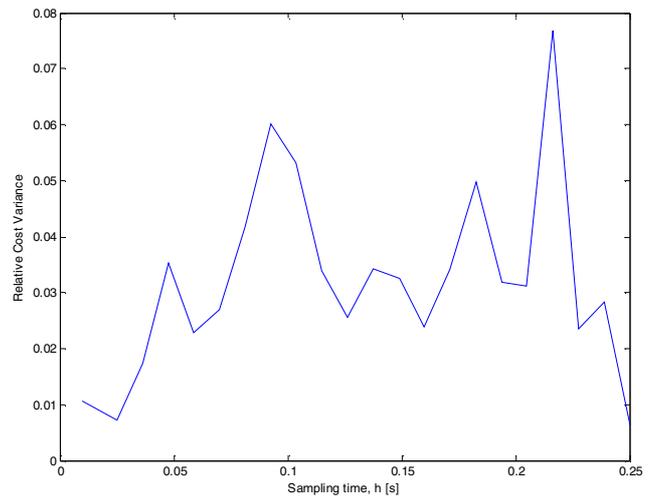


Fig. 5. Relative ITAE cost variance as a function of sampling time, h , for example process.

Several approximation alternatives exist, but this is selected because it leads to a PID controller structure. The resulting IMC controller according to (12) is

$$G_{IMC}(s) = \frac{a_2 s^2 + a_1 s + 1}{(\lambda + \tau)(bs + 1)s} = \frac{1}{(bs + 1)} \frac{1}{(\lambda + \tau)} \left(a_1 + \frac{1}{s} + a_2 s \right) \quad (15)$$

which equals to a PID controller with

$$P = \frac{a_1}{\lambda + \tau}, \quad I = \frac{1}{\lambda + \tau} \quad \text{and} \quad D = \frac{a_2}{\lambda + \tau}, \quad (16)$$

and a pre-filter

$$G_{pf}(s) = \frac{1}{bs + 1}. \quad (17)$$

The pre-filter can be omitted if the process zeros are not modelled. The controller is tested with and without the pre-filter. The value of λ is optimized to give minimum ITAE cost for the controller. For the IMC controller with the pre-filter $\lambda = 0.46$ and without $\lambda = 0.53$.

The average costs and the cost variances are evaluated over $K = 100$ unit steps in simulations for all the tuning methods. The comparison is summarised in Table 2 with the corresponding PID tuning parameters. Also the IAE cost is given as comparison to a cost criterion without time-weighting. Other cost criteria, such as the ISE criterion, give comparable results.

Simulated responses of the tunings on a series of reference step changes are displayed in Fig. 6 and Fig. 7. The Z-N step response method is not shown because it oscillates heavily, so does also the Z-N frequency response method and IMC without the pre-filter. The optimisation method and the IMC with the pre-filter have fast responses with no overshoot. The IMC method has a smooth response but it is slower than the optimisation method.

The optimisation and IMC methods have a comparable performance based on the cost. The IMC is easier to tune as it has only one tuning parameter. It does not, however, take the robustness of the tuning explicitly into account

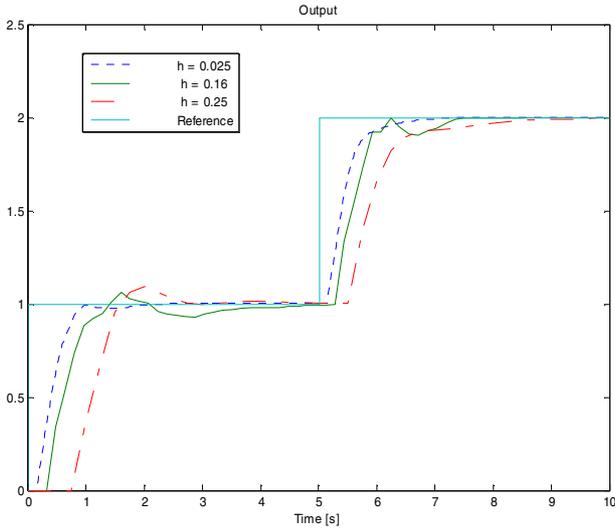


Fig. 6. Step responses with optimisation tuning. Sampling times: $h = 0.025, 0.16$ and 0.25 seconds.

since it is based on approximating the varying delay with a constant. The cost variance of the optimisation method is slightly larger than for the IMC method. The low cost variance of the IMC method is due to the pre-filter. Without the pre-filter the variance and cost are considerably higher.

The cost variance of the optimisation tuning method is at any sampling time reasonably small compared to the other tuning methods. It results in both a low cost and a low cost variance. It is thus both fast and robust.

VI. TEST RUNS WITH THE MOCONET SYSTEM

The tunings done in simulations are verified on a real process using the MoCoNet system. The system has a network simulator presented in Section III and it enables investigations with real processes in a networked control system setting. The average run ITAE costs and the cost variance are estimated with $K = 20$. Step responses are evaluated visually. The results are given in Table 2 with the simulation results.

Compared to the simulation results the optimisation tuned controller performs roughly equally well on the real system. It has the lowest cost with the IMC controller with pre-filter. The Z-N step response tuning is unstable. The frequency response and the IMC without the pre-filter are highly oscillating. The cost variances for both methods are high.

The cost variances are about the same as in the simulations, except for the IMC without the pre-filter,

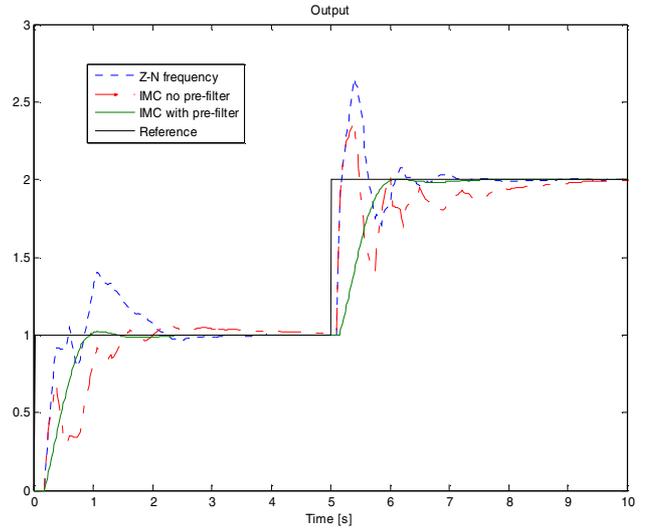


Fig. 7. Simulated step responses for example process. Other tuning methods.

which oscillates in the test run. The conclusion is that the tunings are equally robust on the real process as in the simulations. The IMC controller with the pre-filter had the lowest cost variance in the simulations, but the optimisation method has lowest variance in the test runs.

As the optimisation method has only minor deviations from the simulation results in terms of the cost and cost variance, the conclusion is that optimisation tuning is robust to network delay, measurement noise and modelling errors when the tuning is done as presented in this paper.

All the tuning methods except the Z-N methods and the IMC without the pre-filter gave satisfactory results based on the cost and step response tests, and can thus be used to tune a PID controller for practical use in NCS. Evaluating the performances of the controllers visually leads to the conclusion that the optimisation method has the most desirable step response. As an example of the test runs, a run with the optimisation tuned controller is shown in Fig. 8. There are some minor oscillations, not present in the simulation, but the response is fast and stable.

In this case the simulation results gave a good indication on the performance of the real system. This is mainly because the process is simple to model and the resulting simulation model is sufficiently accurate. As the optimisation method relies on the process model, it is expected to give comparable results also in other cases, as long as the simulation model is reasonably accurate, including the network delay model.

Table 2: PID tuning, ITAE simulation and test run cost, and cost variance for Ziegler-Nichols frequency and step method, optimisation with $h = 0.025$ s and $h = 0.16$ s, IMC with and without a pre-filter.

Method	P	I	D	Simulation ITAE	Run ITAE	Simulation IAE	Simulation Cost variance	Run Cost variance
Z-N step response	3.36	6.72	0.42	1.3	Unstable	0.66	2	Unstable
Z-N freq. response	2.7	5.57	0.33	0.38	0.6	0.10	0.03	0.1
Opt., $h = 0.025$	1.43	1.75	0.00	0.15	0.2	0.057	0.0002	0.0004
Opt., $h = 0.16$	1.16	1.15	0.08	0.39	0.5	0.13	0.0003	0.01
IMC (w/o pre-filter)	1.84	1.50	0.47	0.19	0.7	0.20	0.05	0.1
IMC (with pre-filter)	2.06	1.68	0.52	0.07	0.2	0.074	0.00003	0.002

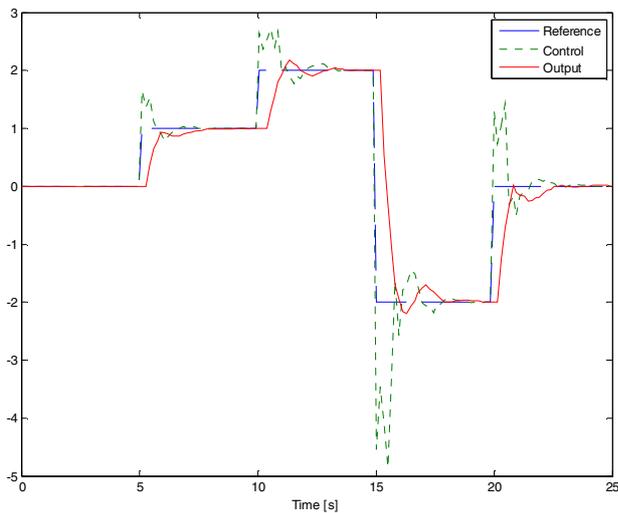


Fig. 8. Optimisation tuning run with $h = 0.016$ s. Reference, control signal and output response.

VII. CONCLUSIONS

A discrete-time PID controller was selected to control a process in a networked control system. The optimisation tuning method by simulation was applied, where the network was modelled as the Internet with the gamma delay distribution. The Internet was used as a difficult test case. In applications dedicated Ethernet and wireless networks will probably be used.

The tuning criterion was set up to minimise the maximum cost of several step responses. Other tuning methods were compared with the optimisation method in simulations. The optimisation tuning method was shown to perform better than the Ziegler-Nichols step and frequency response test methods and the IMC tuning method.

The average optimal tuning cost as a function of sampling time increases linearly. There is a trade-off between the controller performance (sampling time) and network congestion. The relative cost variance for the optimal tuning is low at all sampling times. The tuning is thus robust at any sampling time. The controller can therefore be used at any desirable sampling time where the network utilisation is at a suitable level. The favourable sampling time is such that the resulting effective delay varies between two values.

The MoCoNet system was introduced and the simulation results were verified on a real process with it. With a network simulator MoCoNet enables the investigation of control systems with real processes in a NCS setting.

The performances of the controllers were equal to the simulation performances and the cost variances were similar to those estimated in the simulations. The conclusion was that with an adequate process model and network delay model the discrete-time PID controller tuned with the optimisation tuning method can be used in practice on processes in a NCS setting. The optimisation tuning method performed better and was more robust to the varying network time-delay than the other tuning methods were.

This paper showed that a discrete-time PID controller can be used to control a process with varying time-delay. The performance is good as long as the delay variation is small, in the order of the sampling time. Traditional tuning

methods do not perform well in the new setting of NCS. The tuning can instead be done with the described optimisation tuning method using simulation.

VIII. REFERENCES

- [1] T. Katayama, T. McKelvey, A. Sano, C. Cassandras, M. Camp. "Trends in Systems and Signals". In *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- [2] W. Zhang, M.S. Branicky, S. M. Phillips. "Stability of Networked Control Systems". *IEEE Control Systems Magazine*, Vol. 21, Issue 1, February 2001. ISSN: 0272-1708.
- [3] S. Chai. "Design and Implementation of Networked Predictive Control Systems". In *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- [4] B. Lincoln. "Dynamic programming and Time-Varying Delay Systems", *Ph.D. thesis*, Department of Automatic Control, Lund Institute of Technology, Sweden, 2003.
- [5] M. Pohjola, L. Eriksson, V. Hölttä, T. Oksanen, "Platform for Monitoring and Controlling Educational Laboratory Processes over Internet". In *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- [6] J. Nilsson. "Real-Time Control Systems with Delays". *Master's thesis*, Department of Automatic Control, Lund Institute of Technology, Sweden. 1998. ISSN 0280-5316.
- [7] K. J. Åström and T. Hägglund. *PID Controllers: Theory, Design, and Tuning*, 2nd edition. Instrument Society of America, 1995.
- [8] L. Eriksson and H. N. Koivo. "Comparison of PID, IMC and Fuzzy Gain Scheduling Controllers in Varying Time-Delay Systems". In *proc. 5th Vienna Symposium on Mathematical Modelling - MATHMOD*, Vienna, Austria, February 8-10, 2006.
- [9] J. C. Bolot. "Characterizing End-to-End Packet Delay and Loss in the Internet". *Journal of High-Speed Networks*. Sigcomm 1993, pp. 289 – 298.
- [10] A. Cervin, D. Henriksson, B. Lincoln, K-E. Årzén. "Jitterbug and TrueTime: Analysis Tools for RealTime Control Systems". *2nd Workshop on RealTime Tools*. Copenhagen, Denmark, 2002.
- [11] A. Mukherjee. "On the Dynamics and Significance of Low Frequency Components of the Internet Load". In *Proc. First ACM SIGCOMM Workshop on Internet Measurement*, 2001. p: 281 – 293. ISBN: 1-58113-435-5.
- [12] A. Reijonen. "Tuning of PID Controller for Varying Time-Delay Systems". *Master's thesis*, Department of Automation and Systems Technology, Helsinki University of Technology, Finland, 2003.
- [13] M. Pohjola. "PID Controller Design in Networked Control Systems". *Master's thesis*, Department of Automation and Systems Technology, Helsinki University of Technology, Finland. 9.1.2006.
- [14] C. E. Garcia and M. Morari. "Internal Model Control. 1. A Unifying Review and Some New Results". *Industrial Engineering Chemistry Process Design and Development*. Vol. 21, No. 2, 1982.