Control of the Rotary Inverted Pendulum through Threshold-Based Communication

Abstract

This paper deals with the real implementation of an event-based control structure for the classical rotary inverted pendulum. The communication between controller and plant is performed through Ethernet (TCP/IP) which leads to a Networked Control System. The bandwidth used by the control loop is reduced, compared with the one that needs a conventional control, by using a threshold-based communication. The values of the thresholds have been determined by means of simulation techniques. The results over the real plant show how this technique can reach a significant reduction of the bandwidth consumed with a negligible worsening of the performance.

Keywords

Event-based control systems, networked control systems, bandwidth saving.

1. - Introduction

When dealing with a Networked Control System (NCS) the most valuable resource is the bandwidth. In this kind of control systems, the necessary communication between the control device and the controlled plant is carried out through a shared communication medium ([1]–[7]). A potentially large number of devices are using the same medium to transmit information for different purposes: control loops, monitoring, supervising, alarms, maintenance and even transmission of information not related with control tasks (real and non-real time data).

Depending on the physical medium used to transmit the information (electrical, optical, wireless, infrared …) and on the distance between emitter and receiver devices, a certain bandwidth is available ([8]–[11]). This must be understood as the amount of information that can be transmitted per time unit. The bandwidth must be shared out between all the information generators (writer devices) attached to the medium. When the number of sharing devices increases, the real bandwidth available to each *writer* decreases.

In a typical control loop there are two devices able to generate information to be sent through the shared medium: sensor and controller. There are also two information consumers (reader devices): controller and actuator. The information generated by the sensor (output samples) is consumed by the controller. The information generated by the controller (control actions) is consumed by the actuator. Two links must be implemented: sensor-to-controller (SC) and controller-to-actuator (CA). In a conventional control system each link is implemented through its own physical medium but in a NCS they are virtual links implemented through the same medium. If a physical medium offers a bandwidth B bps (bits-per-second) and L control loops are going to be closed through it, each *writer* has a real bandwidth of B/(2*L) bps. Of course this happens when all the *writers* have the same priority and the bandwidth is assigned no matter if it is needed or not. This is the policy when all the control loops are using the same sample time and the same number of samples/actions per second is transmitted. It is also possible that different *writers* have different needs and they get a different bandwidth. Loops with smaller sample time get a larger portion of the shared bandwidth. In this case the bandwidth is assigned to each *writer* depending on the control frequency and is constant during its operating life. This pre-assigned bandwidth policy is often used in industrial communication media as, for example, Profibus-DP ([12]–[14]).

But as any control engineer knows, the need for information transmission depends on the state of the controlled plant. There are periods of time in which the system is quite stable. There are neither reference changes nor perturbations and the controlled signal (i.e. the plant output) remains (almost) in the same value. Along these periods the assigned bandwidth is wasted transmitting (almost) the same sample and (almost) the same control action. To optimize the use of the precious resource an 'on-demand' policy appears to be appropriated. The main idea is to send only those sample/actions which are significant enough according to the control system behavior. The information not sent (because it has been considered negligible) leave the resource free to be used by other device. Obviously, this strategy needs a communication medium in which the bandwidth is not pre-assigned but is requested by the *writer* device. The goal is to use the minimum bandwidth to get the desired behavior for the controlled plant. TCP-IP follows this 'on-demand' policy and it is the communication protocol to be used in this work ([15]–[18]).

This event-triggered control strategy ([19]–[27]) differs from conventional control in the fact that the sample period is not determined only by the plant dynamics and the design specifications. The effective sample time (which determines the consumed bandwidth) is variable depending on the state of the controlled plant. When the plant is in a stable situation (the reference to be followed is not changing and there are no significant disturbances to cancel) the sample period increases (consuming less bandwidth) and when the plant is moving, the sample period decreases (consuming more bandwidth) towards the one used in conventional (constant sample time) control.

This can be compared with the variable simulation step in simulation theory ([28]-[29]). When implementing an integration method it can be done in fixed or variable step. In fixed step the simulation step is constant during the life of the simulation. Variable step (frequently used as default in general purpose simulation tools) means that the simulation step increases or decreases depending on the computation needs to reach good quality results, according to a certain tolerance. In fixed step the number of simulation steps (and so, the computational cost) is constant and known and it is independent on the simulated system behavior. In variable step computational cost depends on the system dynamics. When simulating a system with long stable periods of time (in which the state does not change significantly) some simulation steps can be skipped (i.e. saved) to reduce the computational cost and to finish the simulation earlier. These variable step simulation based techniques are going to be applied in this work to a classical control problem. They will reduce, in a significant way, the need of bandwidth without degrading the system performance. Instead of saving computational cost the technique is used here to save bandwidth.

In variable step simulation the decision of changing the simulation step (i.e. computing or not each step) is based on an estimation of the simulation error. Two different integration methods are used and the results are compared. If the difference is below a certain tolerance it is assumed that the simulation is performing well and the step is increased, saving computational cost. In the control strategy proposed in this paper the condition to change the effective sampling period (i.e. transmitting or not each sample/action) is slightly different. If the difference between one sample/action and the previous one is below a certain threshold, the sample is not transmitted, saving bandwidth for other devices. This threshold must be chosen not too small to get a significant reduction on the used bandwidth and not too large to avoid a significant worsening of the system performance. And this is the main challenge: to choose the appropriated threshold to get a good bandwidth/performance ratio.

The threshold is the parameter to decide if the difference between one sample/action and the previously transmitted one is negligible or not. This parameter must be pre-arranged to an appropriated value. It can be done experimentally working with the real plant to be controlled or with a reliable enough mathematical model. Only constant thresholds are used in this work, that is intended to be continued using variable thresholds. In variable threshold it will be dynamically adapted during the life of the control system in a similar way as it is modified the size of the step in variable step simulation.

The fact that not all the samples/actions are going to be transmitted, according to its difference with the previous one, leads to an event sampling control system. The transmission of one sample/action is not caused by an external event but by an event which depends on the plant state which is, in fact, modified by the event itself.

The aim of this work is to prove that the previously described threshold-based control strategy can be applied to a real plant with a significant gain on the bandwidth consumption. In conventional control the easy way to save bandwidth is to increase the sampling period (i.e. to decrease the control frequency). This is simple and efficient but only useful when the sampling time requirements of the controlled plant are low. When controlling a plant with small time constants, the lower bound of the control frequency is easily achieved. In this case, decreasing the control frequency will lead to performance degradation and instability. The proposed strategy only makes sense if it is better than the conventional solution so, it must be tested in rough conditions, in the limits of the control frequency. The plant to be controlled will be the rotary inverted pendulum (RIP) for several reasons:

- RIP is a classical control problem frequently used as a benchmark to measure the goodness of control structures.
- RIP is an unstable and non-minimum phase plant, with a small time constant, which becomes easily unstable when the control frequency is decreased.

- RIP is a multivariable plant with (at least) three signals to transmit (two output signals and one control action). The bandwidth gain will be larger when more information is transmitted between controller and plant.
- RIP is available at our laboratory and the proposed control strategy can be implemented in real conditions not only over the simulation model.

The paper is organized as follows. Section 2 shows the behavior of the control in ideal conditions; conventional control and small enough control frequency to reach the desired behavior. Section 3 is shows the bandwidth saving when using threshold-based communications. Section 4 presents conclusions of this paper and future work.

2. - Conventional control of the rotary inverted pendulum

The well-known rotational inverted pendulum (RIP), also named Furuta pendulum, has a DC motor that moves an arm attached to its shaft. At the end of this arm there is a joint in which it is attached the pendulum ([30]–[35]). The goal is to keep the pendulum in its unstable equilibrium upwards position, while a certain reference signal is followed by the motor arm. In this work, the RIP developed by Quanser Consulting Inc. is going to be used. Figure 1 shows the plant, in the upwards position.

The results in this section have been obtained with the real RIP when using a conventional control, without bandwidth limitations. This will be the nominal behavior to compare with the one that will be achieved when reducing the amount of information transmitted between the plant and the control structure.

RIP is a continuous underactuated plant with one input, the control action applied to the DC motor carrying the pendulum (*u(t)*, measured in volts) and two outputs:

- $\theta(t)$: Angular position of the motor shaft around the vertical axis.
- $\alpha(t)$: Angular position of the pendulum rod around the motor shaft axis.

The theoretical multivariable model includes the first derivative of these signals (velocity) but is not going to be considered as the real plant does not have sensors to measure them. Figure 2 shows a schematic representation (top and front view) of RIP.

The aim of this work is to prove, over a real plant, that threshold-based communication can reduce the amount of information between controller and plant without a significant worsening of the system behavior. No new controller is going to be designed to reduce the bandwidth. The controller will be the one developed to operate in a conventional control structure without bandwidth limitations. Quanser provides a LQR controller suitable to be used in ideal conditions. The controller uses as inputs *θ(t)* and *α(t)* (measured in the real plant), its derivatives (estimated by discrete-time derivators) and the integral of *θ(t)* (estimated by a discrete time integrator). The integral of *θ(t)* has been added to the original state vector of the plant to improve the robustness and to allow that the motor shaft can follow step references with zero error, even with the small dead zone present in the real plant. The augmented five-state vector of the plant used to design the LQR is:

$$\tilde{X} = \left[\theta(t) \quad , \quad \alpha(t) \quad , \quad \frac{d\,\theta(t)}{dt} \quad , \quad \frac{d\,\alpha(t)}{dt} \quad , \quad \int \theta(t)\,dt \right]^T$$

From this five-state vector, Quanser provides the gain vector for the LQR and using these gains, the control action to be applied to the plant can be calculated by the control structure as follows:

$$U = K \cdot \tilde{X} = \left[k_1 \quad , \quad k_2 \quad , \quad k_3 \quad , \quad k_4 \quad , \quad k_5 \right] \cdot \tilde{X} = k_1 \cdot \theta(t) + k_2 \cdot \alpha(t) + k_3 \cdot \frac{d\,\theta(t)}{dt} + k_4 \cdot \frac{d\,\alpha(t)}{dt} + k_5 \cdot \int \theta(t)\,dt =$$
$$= \left(k_1 \cdot \theta(t) + k_3 \cdot \frac{d\,\theta(t)}{dt} + k_5 \cdot \int \theta(t)\,dt \right) + \left(k_2 \cdot \alpha(t) + k_4 \cdot \frac{d\,\alpha(t)}{dt} \right)$$

As it can be seen in the previous expression, the LQR controller has been divided in two independent sub-controllers for a better manipulation of the feedback signals. These two sub-controllers become a PID one for *θ(t)* and a PD controller for *α(t)*. The parameters of these PID and PD controller, in the standard industrial form are:

$$Kp_\theta = k_1 \quad ; \quad Td_\theta = {}^{k_3}\!/_{k_1} \quad ; \quad Ti_\theta = {}^{k_1}\!/_{k_5}$$
$$Kp_\alpha = k_2 \quad ; \quad Td_\alpha = {}^{k_4}\!/_{k_2} \quad ; \quad Ti_\alpha = \infty$$

The discretized version of these controllers are the ones used to obtain the results presents in this work. The addition of the two outputs of these sub-controllers becomes the control action to be applied to the plant. The resulting control structure is shown in Figure 3. Details on this control structure can be found in [9], referred to the double rotary inverted pendulum.

The control structure must use a small enough sample period (T) to reach a good behavior. Working with the real plant it has been stated T=10 ms. as the most appropriated sample period to be used. Larger periods are not able to keep the plant stable while following the reference in an appropriated way. Smaller periods mean an unnecessary large need of bandwidth and computational cost. So, T=10 ms. has been found as the period to use in conventional control to reach a satisfactory behavior with the minimum amount of information transmitted between controller and plant. With this sample period, a bandwidth of 100 samples/second is consumed in each signal. As there are three transmitted signals, the control in ideal conditions needs a total bandwidth of 300 samples/second (200 in SC link and 100 in the CA link). This will be the nominal value of the consumed bandwidth (BW_0) used for comparison with the one consumed when the threshold-based communication will be implemented.

The results these conditions are shown in Figure 4. Both controllers have been implemented in one computer (the remote side of the NCS). AD and DA converters have been placed in a different computer (the local side of the NCS). Ethernet (TCP/IP) has been used as the communication medium to implement SC and CA links. The distance between remote and local computer in the experiments is short enough to consider negligible the transmission delays. No information is lost during the communication.

A square wave of ±45° amplitude has been used as reference signal to be followed by $\theta(t)$. Upper graphic shows *θ(t)* versus reference and lower one shows *α(t)*. As can be seen, conventional control makes the motor shaft to follow the desired reference while keeps the angle of the pendulum rod around 0º, compensating the unavoidable mechanical disturbances. This nominal behavior is not going to be improved in this work. The goal is to keep the system as close as possible to this behavior while reducing the bandwidth used to feedback the controlled signals.

Conventional solution to reduce bandwidth is to increase the sample period. Using T=15 ms. (33% bandwidth reduction) the behavior is significantly worse, as shown in Figure 5. Using a T=20 ms, (50% bandwidth reduction) the plant becomes completely unstable. As will be shown in the following section, threshold-based communication will get a significant reduction of the bandwidth without a significant worsening of the system behavior.

To measure the worsening of the system behavior a quality index is needed. The quality measure used in this work involves the integral of the unsigned tracking error (i.e. the difference between reference and angular shaft position) and the integral of the unsigned pendulum angle:

$$ER = \int_0^t \left| r(\tau) - \theta(\tau) \right| \; d\tau + \int_0^t \left| \alpha(t) \right| \; d\tau$$

Using the results with conventional control the nominal value of this error measure has been calculated. Comparing this nominal error (*ER_0*) with the one obtained with threshold-based communication the system worsening can be quantitatively measured.

3. - Threshold-based control of the rotary inverted pendulum

Threshold-based communication is implemented to reduce the amount of information transmitted through the shared communication medium. In the RIP control loop there are three discrete signals that must be transmitted between controller and plant. So, there are three data to be sent in every sample period:

the samples of the two controlled signals captured by the sensors (i.e. the shaft and pendulum angles) and the control action generated by the controller and to be applied by the actuator. The three information generators (sensors and controller) are time-based devices. The clock to determine the generation of samples and control actions works with the period used in ideal conditions: T=10 ms. As it has been said, with a conventional time-based control 300 samples per second must use the link between controller and plant. Using an event-based control policy the exchange of information is not necessary to be equal spaced in time. The main idea is to transmit only the significant information, leaving the shared medium idle to be used for a different purpose when the information is negligible. To reach this goal, it must be determined by some means which samples/actions are significant and which ones are negligible, according to the system stability and performance.

The implementation in this work uses thresholds to determine the significance of every sample/action to be sent through the shared medium. Each sample captured by the sensors and each action generated by the controller is compared with the previously transmitted one. If the difference between one generated data and the last significant one is below a certain pre-arranged threshold is considered negligible. In this case the sample/action it is not transmitted to the other side of the shared communication medium. Note that the comparison is between the new sample/action and the last non-negligible one, not the previous captured/generated one. This must be done this way because if not, small differences between samples/actions can become an important difference with the last transmitted sample/action. Being y_k the information captured/generated in a given sample instant and TRH the value of the threshold, the transmitted information, $\hat{y}_k$, will be:

$$\hat{y}_k = \begin{cases} \hat{y}_{k-1} & \text{when } \left| y_k - \hat{y}_{k-1} \right| < TRH \qquad \text{(not transmitted)} \\ y_k & \text{when } \left| y_k - \hat{y}_{k-1} \right| > TRH \qquad \text{(transmitted)} \end{cases}$$

Note that when $\hat{y}_k = \hat{y}_{k-1}$ the new sample action is not transmitted. Data not transmitted leave the medium idle to be used by some other device, saving bandwidth for a different purpose. The main idea is not to waste bandwidth when the new data does not provide significant information. The algorithm implemented between the information generators and the shared medium is depicted in Figure 6.

The key point in this communication policy is the value of the threshold. A too small threshold will cause that too many data will be sent and there will not be a significant bandwidth saving. If the threshold is too large significant information will be lost, leading to performance degradation and even instability. In addition, if the threshold is too large, the degradation of the system performance will cause larger changes in the data to be sent. In this case the bandwidth is only not reduced but can be even increased. There must be a compromise between the reduction of the consumed bandwidth and the increasing of the error index.

The most appropriated value for the thresholds are going to be determined by means of simulation. The values will be chosen to reach a good error/bandwidth ratio, measured by a certain index. In this work, the procedure to determine the thresholds will be applied to the RIP control but can be applied to any other control problem to reduce the consumed bandwidth. The only requirement to apply the procedure described in this work is to have a good simulation model of the controlled plant. This model can include any kind of non-linear behavior, disturbances and uncertainties for a better representation of the real plant.

3.1. - Optimal threshold values determination

There is not an easy way to determine the optimal values of the thresholds. These values depends on the dynamics of the control system which depends itself on the thresholds. But if a good mathematical model of the controlled plant is available they can be calculated by means of simulation. This will not lead to the optimal thresholds but to the best among all the values tested in the simulation study. The procedure is as follows:

- The behavior of the control system using certain values for the thresholds is simulated.
- The quality of the response is measured calculating the error index (*ER*)
- The consumed bandwidth (*BW*) is measured counting the number of samples/actions transmitted during the simulation.

Repeating the procedure for a large enough collection of possible values for the thresholds, the most appropriated values for the three thresholds have been determined. To reach these values a large number of trials have been done varying the thresholds and measuring the error index and the consumed bandwidth.

The ratio between the error index and the one obtained with conventional control (ER_0) provides the relative error (ER_R), always above one. The larger this relative error is, the worse behavior is caused.

$$ER_R = ER / ER_0$$

The ratio between the number of data above the threshold and the consumed bandwidth with time-based control (BW_0) is the relative bandwidth index (BW_R), always below one. The smaller this relative bandwidth is, the larger bandwidth saving is gained.

$$BW_R = BW / BW_0$$

The results of this study are presented in Figure 7 (relative error and relative bandwidth versus theta threshold), Figure 8 (relative error index and relative bandwidth versus alpha threshold) and Figure 9 (relative error index and relative bandwidth versus control action threshold).

The results from this experiment show that, even if threshold-based communication is used in only one of the three signals, a significant reduction of the consumed bandwidth can be achieved without a significant worsening of the behavior. But if the threshold is increased beyond a certain point the bandwidth gain is much smaller than the performance worsening. An index combining these two experimental measures must be used to decide the optimal value of the threshold. The error/bandwidth index can be calculated giving the appropriated weigh to each one. It is not an easy problem to choose this weigh because it depends on when the bandwidth improvement is less significant than the behavior worsening. After some experiments with the real plant and the simulation model, it has been decided that the performance index to be used in this work will be:

$$IX = 0.2 \cdot ER_R + 0.8 \cdot BW_R$$

The values of the threshold that minimize this index will be the ones to use in the real implementation. Figure 10 shows this performance index versus the three thresholds, considered one by one. The minimum value (marked with a circle) gives the optimal (according to the selected index) value of the thresholds.

Using only one threshold an index of around 0.8 can be achieved. It is obvious that better results can be achieved if two thresholds are used together. Figures 11, 12 and 13 show the index IX when varying a pair of thresholds. The color in these figures indicates the value of the index to be minimized. The white dot marks the minimum of the index and the values chosen for the thresholds. These results, obtained using the simulation model of the RIP, show that when using a pair of thresholds the performance index can be improved up to around 0.6 as the bandwidth is saved in two of the transmitted signals instead of only one as in the first experiment.

Finally, it is easy to think that the results can be improved by using event-based communication for the three signals. After a (large) simulation study with three nested loops the optimal (again, according to the chosen index) values of the thresholds have been stated as follows:

- Theta threshold: 0.36
- Alpha threshold: 0.29
- Control threshold: 0.26

As there are four variables involved (the index and the three thresholds) there is no possible graphic representation of the results in this experiment but the simulation shows that using these values for the thresholds, the performance index can be reduced below 0.5. In these conditions the error index is only 4.5% larger than the one with conventional control but the consumed bandwidth is only the 34% than the one needed with time-based communication. If these results are confirmed in the real plant, a reduction of the bandwidth around 60-70% will be achieved with a behavior that will be almost the same than the one with conventional control. Using the conventional control structure it is not possible to stabilize the RIP using such a small bandwidth.

3.2. - Experimental results with threshold-based control

Using the values determined in the previous section for the thresholds, the results in the real plant are shown in Figure 14. Again, Ethernet (TCP/IP) has been used to send the information between controller and plant. As can be seen the behavior is slightly worse than in ideal conditions (relative error is 1.06) but the consumed bandwidth is much better (relative bandwidth is 0.31). This means that constant threshold has been achieved a bandwidth saving of 69%, regarding to the consumed in nominal conditions.

Figure 15 shows a detail of a small piece of the theta angle. Upper plot is the signal captured by the sensor and lower one is the signal effectively transmitted to the controller. Samples marked with a circle are the ones which passes the threshold condition and are sent through the shared medium (consuming bandwidth). Samples marked with a solid circle are the ones not transmitted (saving bandwidth). Similar results could be shown in alpha and control action signals.

To understand how this saving is achieved Figure 16 shows the transmission flag for the theta angle. This flag indicates when a sample is effectively transmitted through the shared medium (marked as ones) or skipped as being considered negligible (marked as zeros). The upper plot show data belonging to the transient and the lower show data belonging to the steady state. As can be seen, during the transient as the state of the plant is changing, most of the samples are transmitted. When the plant reaches the steady state, much more samples are discarded, increasing the bandwidth saving. This different need of bandwidth suggest that the values of the thresholds can be dynamically changed for a better use of the available bandwidth. Variable thresholds are proposed as an extension to this work.

The experiment has been repeated (over the real plant) using only one and two thresholds to see how different relative errors and bandwidths can be achieved. The results are summarized in Table 1. As expected, the more thresholds are used the best bandwidth saving and performance index are achieved.

4. - Conclusions and future work

The work in this paper shows the implementation of an event-sampling control, over a real plant and using a real shared communication medium to close the loop. The goal has been to reduce the amount of information exchanged between controller and plant, without a significant loss of control performance. This band width reduction has been achieved by means of using a threshold-based communication. Rotary inverted pendulum has been used for this implementation due to its bandwidth needs and sensitivity to the sampling period. TCP/IP has been used as communication protocol because it follows an 'on-demand' policy and in this kind of sharing policy is where the bandwidth saving is effective.

The results show how the threshold-based communication can be easily used to reduce significantly the consumed bandwidth and the behavior of the system is almost the same than the one with conventional control. Simulation model of the controlled plant has been used to select the most appropriated values for the thresholds to minimize an index, involving bandwidth and performance.

The procedure to determine the best values for the thresholds have been determined using the simulation model of the controlled plant. This procedure, applied in this work to the particular case of controlling the rotary inverted pendulum, can be exported to the control of any other plant no matter if it is linear or not, stable or not, as far as a good enough model is available.

A detailed analysis of the results when using threshold-based communication shows that the need for bandwidth is not constant during the life of the system. It depends on the state of the controlled plant which depends on the reference signal and on the presence of significant disturbances. The pre-arranged values for the thresholds has been obtained in certain environment conditions (reference signal to be followed, communication delays, signals noise, performance index …). If these conditions change, maybe the pre-arranged values are not the most appropriated. This suggest an adaptive, variable threshold-based communication in which the values for the thresholds are dynamically modified depending on the plant state. This adaptive policy is expected to improve the results in this paper.

In addition, when using constant thresholds the system does not reach an equilibrium position. A certain ripple around the desired value appears when the motor shaft reference is constant. The presence of this undesired characteristic it is difficult to be seen in this particular plant because the upwards position is an unstable equilibrium point and same ripple is always present, even when conventional control is applied. Using an stable plant the ripple caused by the threshold can be considered in the performance index as a new factor to measure the system behavior. Variable thresholds are expected to help in solving this problem without increasing the consumed bandwidth.

Another improvement to be considered is to use a different sample period in the threshold comparison, leading to a dual-rate structure that could improve the performance without increasing the consumed bandwidth..

Acknowledgements

This work was supported by the Spanish Ministerio de Economía y Competitividad under Grant referenced TEC2012-31506.

References

[1] L.G. Bushnell. “Networks and control”. IEEE Control Systems Magazine, vol. 21, no. 1, pp. 22–23, 2001.

[2] W. Zhang, M.S. Branicky and S.M. Phillips. “Stability of networked control systems,” IEEE Control Systems Magazine, vol. 21, no. 1, pp. 84–99, 2001.

[3] Y. Tipsuwan and M-Y. Chow. “Control methodologies in networked control systems”. Control Engineering Practice, vol. 11, no. 10, pp. 1099–1111, 2001.

[4] L.A. Montestruque and P. Antsaklis. ”Stability of model-based networked control systems with time-varying transmission times”. IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1562–1572, 2004.

[5] J.P. Hespanha, P. Naghshtabrizi and X. Yonggang. “A Survey of Recent Results in Networked Control Systems”. Proceedings of the IEEE, vol. 95, no. 1, pp. 138–162, 2007.

[6] W. Chen and L. Qiu. “Stabilization of networked control systems with multirate sampling”. Automatica, vol. 49, no. 6, pp. 1528–1537, 2013.

[7] H.D. Tran, Z.H. Guan, X.K. Dang, X.M. Cheng and F.S. Yuan. “A normalized PID controller in networked control systems with varying time delays”. ISA Transactions, vol. 52, no. 5, pp. 592–599, 2013.

[8] J. Baillieul and P.J. Antsaklis. “Control and Communication Challenges in Networked Real-Time Systems”. Proceedings of the IEEE, vol. 95, no. 1, pp. 9–28, 2007.

[9] V. Casanova, J. Salt, A. Cuenca and R. Piza. “Networked Control Systems: control structures with bandwidth limitations”, International Journal of Systems, Control and Communications, vol. 1, no. 3, pp. 267–296, 2009.

[10] M. Liu, Q. Wang and H. Li. “State estimation and stabilization for nonlinear networked control systems with limited capacity channel”. Journal of the Franklin Institute, vol. 348, no. 8, pp. 1869–1885, 2011.

[11] L. Zhang, H. Gao and O. Kaynak “Network-Induced Constraints in Networked Control Systems—A Survey”. IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 403–416, 2013.

[12] E. Tovar and F. Vasques, “Cycle time properties of the PROFIBUS timed-token protocol”. Computer Communications, vol. 22, no. 13, pp. 1206–1216, 1999.

[13] K.C., Lee, S. Lee, Suk and M.H. Lee “QoS-based remote control of networked control systems via Profibus token passing protocol”. IEEE Transactions on Industrial Electronics, vol. 1, no. 3, pp. 183–191. 2005.

[14] M. Felser. “PROFIBUS Manual—A collection of information explaining PROFIBUS”. ePubli GmbH. 2011.

[15] D. E. Comer. “Internetworking with TCP/IP: Principles, Protocols, and Architecture, volume I”. Prentice-Hall, 1991.

[16] F.L. Lian, J.R. Moyne, D.M. Tilbury. “Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet”. IEEE Control Systems Magazine, vol. 21, no. 1, pp. 66–83, 2001.

[17] J.D. Decotignie. “Ethernet-Based Real-Time and Industrial Communications”. Proceedings of the IEEE, vol. 93, no. 6, pp. 1102–1117, 2005.

[18] A. Cuenca, J. Salt, A. Sala and R. Piza "A Delay-Dependent Dual-Rate PID Controller over an Ethernet Network". IEEE Transactions on Industrial Informatics, vol. 7, no. 1, pp. 18–29, 2011.

[19] W.P.M.H. Heelmes, J.H. Sandee, and P.P.J. Van Den Bosch. “Analysis of event-driven controllers for linear systems”. International Journal of Control, vol. 81, no. 4, pp. 571–590, 2007.

[20] J. Komenda and J.H. van Schuppen. “Control of discrete-event systems with modular or distributed structure”. Theoretical Computer Science, vol. 388, no. 1, pp. 199–226, 2007.

[21] K.J. Astrom. “Event Based Control”. Analysis and design of nonlinear control systems, pp. 127–147, Springer, 2008.

[22] A. Eqtami, D.V. Dimarogonas and K.J. Kyriakopoulos. “Event-triggered control for discrete-time systems”. 2010 American Control Conference, pp. 4719–4724, 2010.

[23] X. Wang and M.D. Lemmon. “Event-Triggering in Distributed Networked Control Systems”. IEEE Transactions on Automatic Control, vol. 53, no. 3, pp. 586–601, 2011.

[24] S. Hu and D. Yue. “Event-triggered control design of linear networked systems with quantizations”. ISA Transactions, vol. 51, no. 1, pp. 153–162, 2012.

[25] W.P.M.H. Heemels and M.C.F. Donkers. “Model-based periodic event-triggered control for linear systems”. Automatica, vol. 49, no. 3, pp. 698–711, 2013.

[26] U. Premaratne, S.K. Halgamuge, and I.M.Y. Mareels. “Event Triggered Adaptive Differential Modulation: A New Method for Traffic Reduction in Networked Control Systems”. IEEE Transactions on Automatic Control, vol. 58, no. 7, pp. 1696–1706, 2013.

[27] A. Ruiz, J.E. Jiménez, J. Sánchez and S. Dormido. “A practical tuning methodology for event-based PI control”. Journal of Process Control, vol. 24, no. 1, pp. 278–295, 2014.

[28] J.D. Lambert “Numerical Methods for Ordinary Differential Systems: The Initial Value Problem”, John Wiley and Sons Ltd., 1991.

[29] F.E. Cellier and E. Kofman. “Continuous System Simulation”. Springer, 2006.

[30] K. Furuta, M. Yamakita, S. Kobayashi, and M. Nishimura “A new inverted pendulum apparatus for education”. IFAC Advances in Control Education Conference, pp. 191-196, 1991.

[31] I. Fantoni, I. and R. Lozano. “Non-linear control of underactuated mechanical systems”, Springer-Verlag, 2002.

[32] P.X. La Hera, L.B. Freidovich, A.S. Shiriaeva and U. Mettin. “New approach for swinging up the Furuta pendulum: Theory and experiments”. Mechatronics, vol. 19, no. 8, pp. 1240–1250, 2009.

[33] V. Casanova, J. Salt, R. Pizá and A. Cuenca. “Controlling the double rotary inverted pendulum with multiple feedback delays”. International Journal of Computers, Communications & Control, vol. 7, no. 1, pp. 20–38, 2012.

[34] J. Aracil, J.A. Acosta and F. Gordillo. “A nonlinear hybrid controller for swinging-up and stabilizing the Furuta pendulum”. Control Engineering Practice, vol. 21, no. 8, pp. 989–993, 2013.

[35] M. Ramírez-Neria, H. Sira-Ramírez, R. Garrido-Moctezuma and A.Luviano-Juárez. “Linear active disturbance rejection control of underactuated systems: The case of the Furuta pendulum”. ISA Transactions, vol. 53, no. 4, pp. 920–928, 2014.

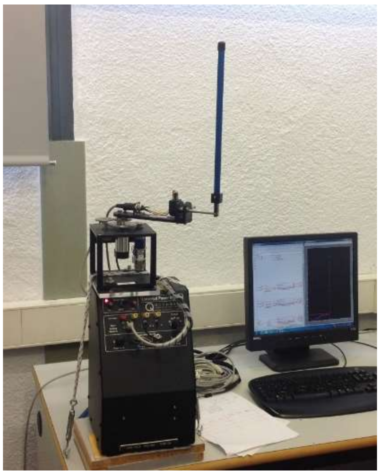


Figure 1. Quanser Rotary Inverted Pendulum

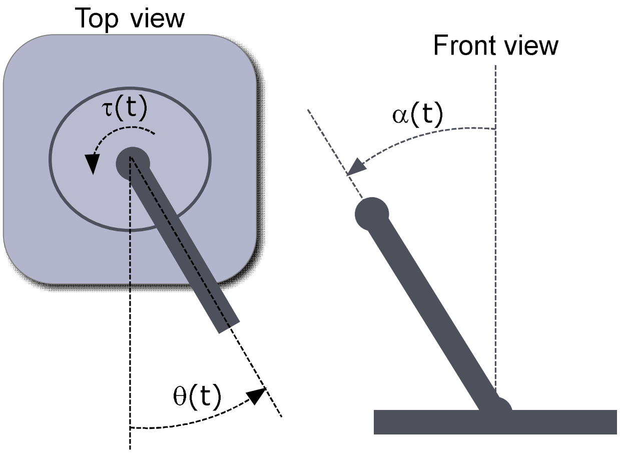


Figure 2. Top and front view

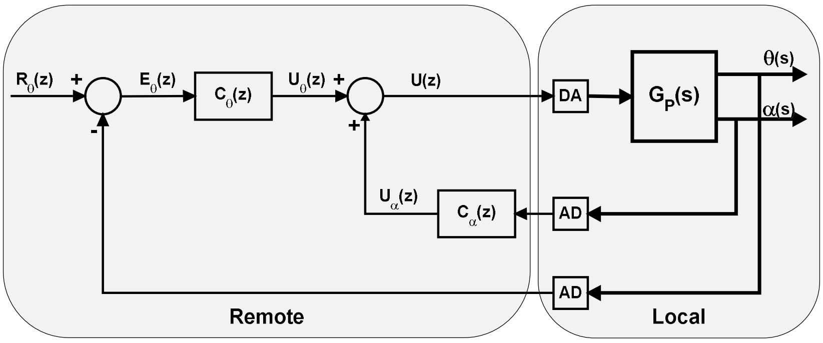
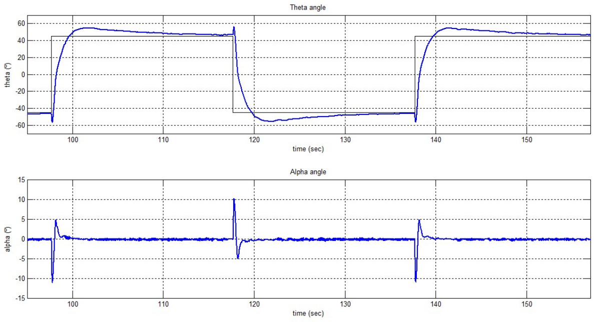


Figure 3. Control structure in ideal conditions



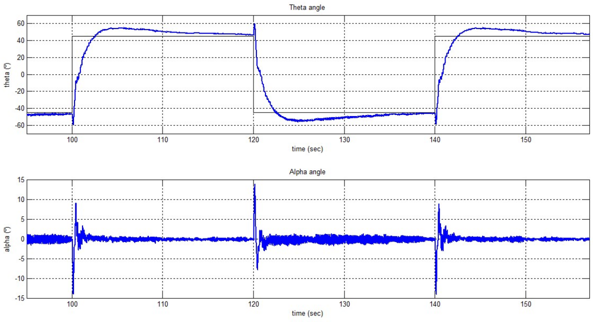


Figure 4. Theta and alpha angles: T=10ms

Figure 5. Theta and alpha angles: T=15 ms

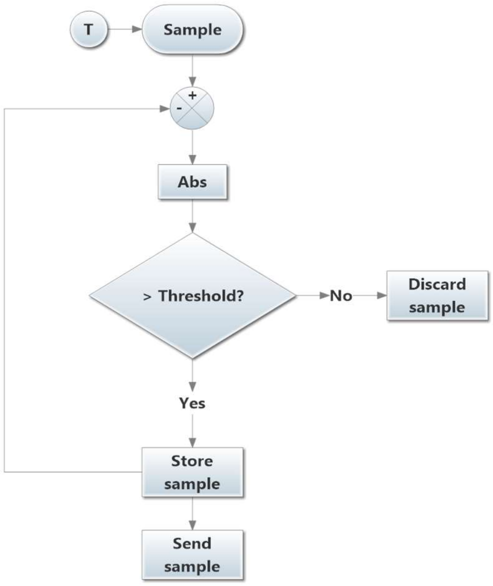


Figure 6. Threshold-based communication flowchart

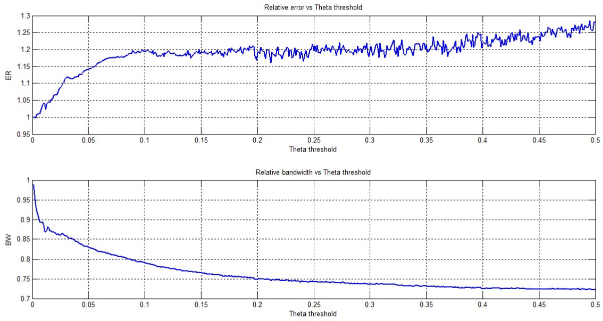


Figure 7. Theta threshold

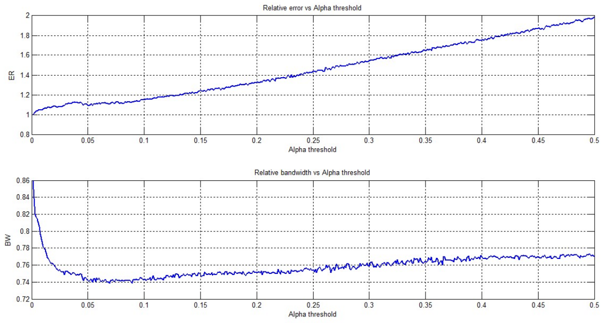


Figure 8. Alpha threshold

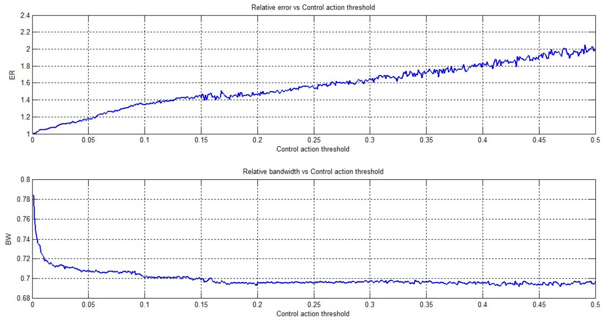


Figure 9. Control action threshold

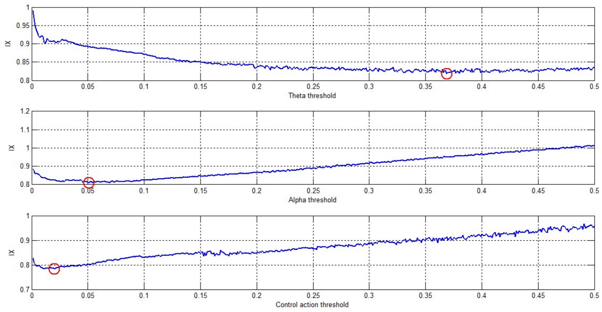


Figure 10. Performance index

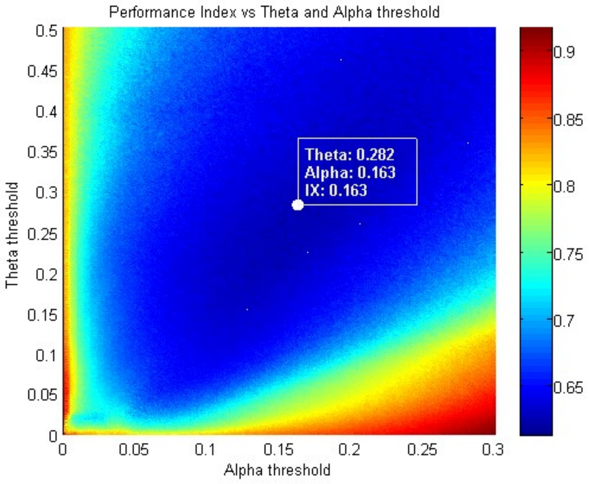


Figure 11. Theta and alpha thresholds

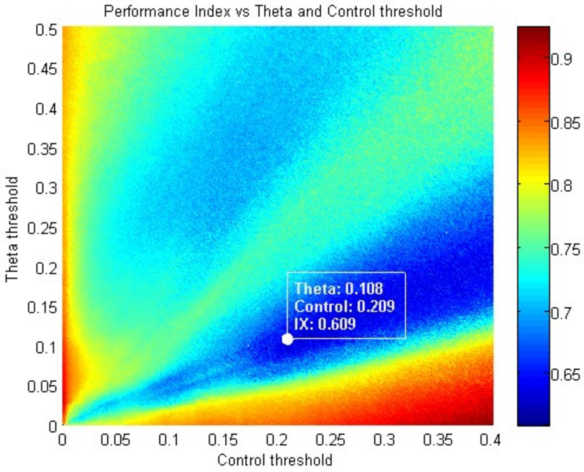


Figure 12. Theta and control thresholds

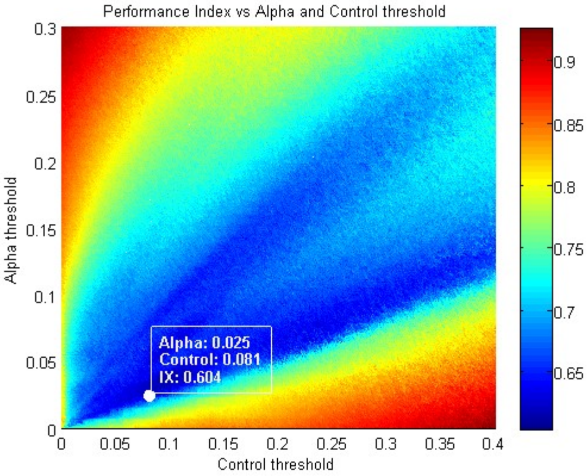


Figure 13. Alpha and control thresholds

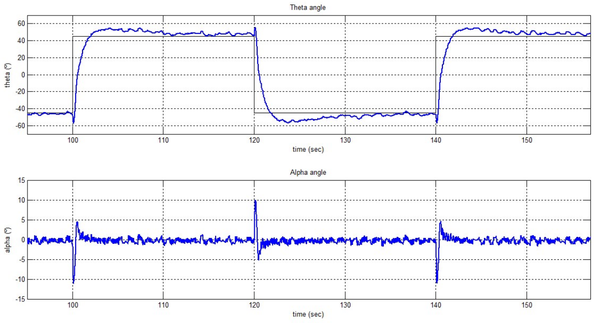


Figure 14. Theta and alpha: Threshold control

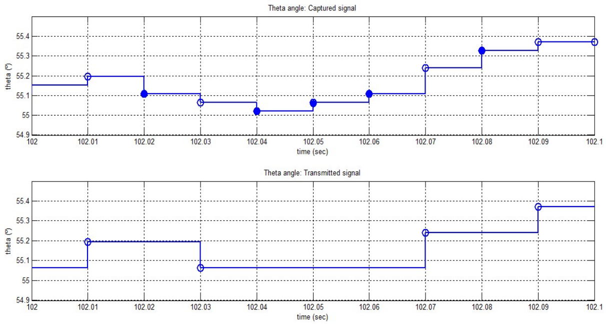


Figure 15. Theta angle: Captured and transmitted

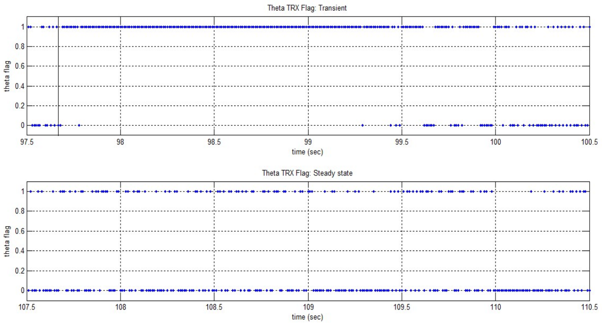


Figure 16. Theta transmission flag

Thresholds	ER_R	BW_R	IX
Theta	1.0290	0.7273	0.7876
Alpha	1.0103	0.7468	0.7995
Control	1.0171	0.7803	0.8277
Theta & Alpha	1.0108	0.4787	0.5851
Theta & Control	1.0408	0.6270	0.7097
Alpha & Control	1.0279	0.6188	0.7006
Theta, Alpha & Control	1.0614	0.3136	0.4632

Table 1. Alpha and control action thresholds