Proceedings of the International Conference of Control, Dynamic Systems, and Robotics
Ottawa, Ontario, Canada, May 15-16 2014
Paper No. 106

Reducing Negative Effects of Jitter in Digital Control Systems Through a Jitter Compensating PID Controller

Chamira Perera, Trevor Pearce
Carleton University, Department of Systems and Computer Engineering
1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6
kaluarachchigeperera@cmail.carleton.ca; pearce@sce.carleton.ca

Miodrag Bolic
University of Ottawa, School of Electrical Engineering and Computer Science
161 Louis Pasteur, Ottawa, Ontario, Canada K1N 6N5
mbolic@eecs.uottawa.ca

Abstract - Software induced delays in a digital control system can lead to degradation in performance and as a result plant response characterizing performance requirements such as percent offshoot, rise time, and settling time can be violated. In the worst case the digital control system can become unstable. These delays can be constant or varying from one period to the other. The latter type of delay is known as *jitter*. Existing solutions attempt to improve the performance of digital control systems in the presence of software induced delays notwithstanding shortcomings. These include but are not limited to, a lack of regard for satisfying plant response characterizing performance requirements, implementation complexity, and an increase in the runtime overhead to execute the solution. In addition, a solution to compensate for input-output and output jitter for proportional integral derivative (PID) controllers is not available in the literature. Taking the problem to be solved and shortcomings in the literature into account, this paper proposes method to design a discrete-time PID controller that compensates for sampling, input-output, and output jitter. The solution initially models the worst case delay between input to output and then places it in a model of the digital control system in the forward loop between the PID controller and plant. Subsequently, the PID controller is tuned to satisfy plant response characterizing performance requirements. The two main advantages of this solution is that plant response characterizing performance requirements are satisfied and the execution time of the PID controller remains unchanged. An evaluation of the performance using a step input experiment validate that the proposed solution is capable of satisfying plant response characterizing performance requirements whereas the uncompensated counterpart failed to satisfy one requirement. A square wave tracking experiment also demonstrated that the proposed solution improved performance.

Keywords: Jitter compensation, real-time digital control systems, software execution delay induced performance degradation, PID Controllers.

1. Introduction

Software that implements the functionality of a digital control system usually executes in a periodic manner. The three main operations of any task that implements the functionality of a digital control system (referred to as a *control task* in this paper) are: input sampling, calculation of the control law, and outputting the control law to the plant. Usually, the input sampling and outputting the control law are performed by hardware via the analog to digital converter (ADC) and digital to analog converter (DAC) respectively. Software is typically involved in calculating the control law. However, either software or hardware can be used to trigger the ADC and/or DAC.

Execution of the analog to digital (A/D) conversion via the ADC, digital to analog (D/A) conversion via the DAC, control law, and other software routines such as interrupt service routines (ISRs), and real-time operating system

(RTOS) scheduler and context switching introduce a delay between sampling of inputs and outputting the control law to the plant. To make matters worse, the timing of these two events can vary from one sampling period to the other. This phenomenon is known as *jitter*. Therefore the type of delays considered in this paper vary from one period to the other, and it also considers the hardware induced delays from the ADC and DAC. This paper considers three types of jitters viz. sampling, input-output, and output jitter.

The negative effect of software induced delays (both constant and jitter) is a performance degradation, which in the worst case can lead to a violation of performance requirements. Ultimately, it could result in destabilizing the digital control system. This is the problem focused in this paper. Traditional digital control system design methodologies assume that the input sampling, control law calculation, and outputting to the plant all occur simultaneously without any delay. This oversight is one of the contributors to the performance degradation issue.

Literature in the field of improving digital control system performance due to software induced delays have several shortcomings. Firstly, not prioritizing satisfying plant response characterizing performance requirements, which are: percent offshoot, rise time, and settling time. Secondly, increased complexity and execution overhead to execute the solution. Finally, a solution to compensate for input-output and output jitter using PID controllers to the best of our knowledge was not found despite its simplicity. A review of the current state of the art is presented in Section 2.

This paper considers a collection of multiple independent single input single output (SISO) digital control systems executing in parallel on a single processor and this collection is referred to as a *digital control system set*. The task model, model of the digital control system set, along with four representative implementations of digital control systems, which refine the digital control system model within a set are discussed in Section 3. Each implementation is referred to as a *configuration* in this paper.

The solution that this paper proposes is a discrete-time PID controller capable of compensating for software induced jitter. The solution involves modelling the worst case delay between input sampling and outputting to the plant, and placing this delay in a model of the digital control system in the forward loop between the PID controller and plant. This worst case delay between input to output is assumed to be one sampling period and this assumption is referred to as the *one sampling period delay assumption*. The discrete-time PID controller is tuned to satisfy performance requirements. A discussion of the proposed solution is provided in Section 4.

An evaluation of the proposed solution which involves controlling the voltage stabilizer example plant model described by Lozoya et al. (2008) and Marti et al. (2010) is presented in Section 5, while the performance results from the evaluation are presented in Section 6. The performance evaluation makes use of the four configurations described in Section 3. Performance results reveal that the proposed solution is capable of satisfying plant response characterizing performance requirements in the presence of software induced jitter when its uncompensated counterpart failed to fulfill at least one requirement. Finally, Section 7 provides concluding remarks.

2. Related Work

The literature for improving digital control system performance in the presence of software induced delays can be classified into three perspectives and they are described in the following subsections.

2.1. Control Systems Perspective

Compensating for software induced delays in the digital controller is the hallmark feature of solutions from the control systems perspective. Åström and Wittenmark (2011) describe a method to model a constant delay between the input and response from the plant. A tutorial by Mathworks (2014) discuses how to compensate only for a constant delay using a PID controller, which leverages the method described by Åström and Wittenmark (2011).

The research by Marti et al. (2001), to the best of our knowledge is the only solution that compensates for sampling jitter using a PID controller. Additionally, this solution is also intended for state feedback controllers. Smeds and Lu (2012) propose to cascade an add-on compensator to the main controller to compensate for input-output jitter. This add-on compensator adds a zero near the Nyquist frequency to attenuate the control gain. The one-shot solution by Lozoya et al. (2008) make use of prediction to compensate for input-output jitter for state feedback controllers. This solution allows sampling jitter to be present, and predicts the control law which is applied at the start of the next sampling period.

2.2. Real-Time Systems Perspective

Solutions from the real-time systems perspective improve control system performance indirectly by reducing delays and jitter. Adjusting the relative deadline is one method to reduce jitter, because this reduces the the slack time between task response and relative deadline. Hong et al. (2010) proposed an approach to adjust the relative deadline where the relative deadline is initially adjusted offline and subsequently re-adjusted online as the workload changes.

Changes can also be made to the scheduling policy to minimize jitter. The solution described by Nasri and Kargahi (2012) modify the task schedule generated by the earliest deadline first (EDF) scheduling policy. Liu et al. (2009) proposed a different solution where the preemption segment of a task is changed statically to reduce software induced jitter.

2.3. Control Scheduling Co-Design Perspective

The control scheduling co-design type solutions aim to apply real-time systems and control systems approaches in a collaborative manner to design multiple digital controllers executing as real-time tasks. In addition, solutions from this perspective attempt to maximize performance of the digital controllers by applying optimization based approaches in an iterative manner. Solutions by Wu et al. (2010), and Aminifar et al. (2013) are examples of solutions from this perspective.

3. System Modelling

All details related to the model of the digital control system used in this paper is elaborated in this Section. The control task model is initially discussed, followed by a description of structure of the digital control system set model. Finally, the structure and timing of the digital control system model within a set is refined to represent four realistic ways of implementing a digital control system.

3.1. Task Model

The functionality of a digital control system set considered in this paper can be modeled as a control task set: $T_i \in \{T_1, T_2, \ldots\}$. Each control task has an infinite periodic instantiation set otherwise known as a job set: $J_{i,j} \in \{J_{i,1}, J_{i,2}, \ldots\}$. The execution of four consecutive jobs from $J_{i,k-2}$ to $J_{i,k+1}$ is depicted in Figure 1.

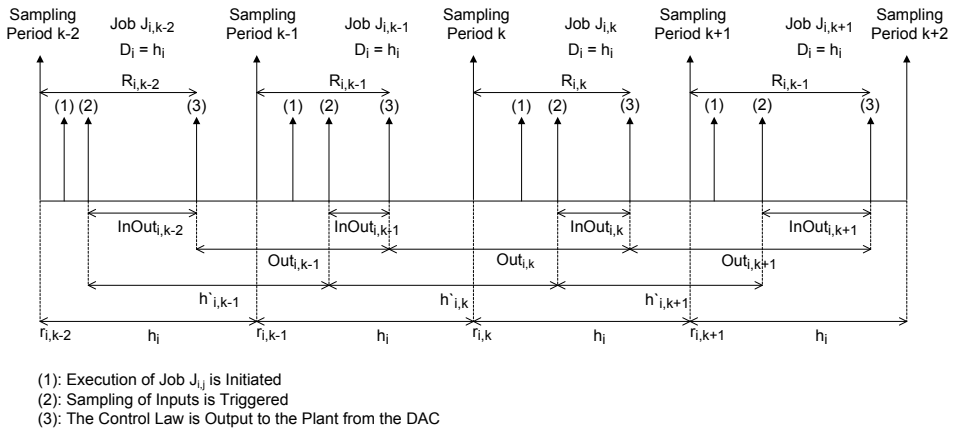

Fig. 1. Timing Diagram for a Typical Digital Control System.

The response time, $R_{i,j}$ (start of sampling period to event (3)) for job $J_{i,j}$ is the time it takes from the start of the sampling period to output the control law to the plant. Taking the response time of a control task and one sampling period delay assumption into account, each control task has to adhere to the timing constraint given by (1),

$$R_i \leq h_i \tag{1}$$

$InOut_{i,j}$ is the input sampling to outputting to the plant delay (event (2) to (3)) for job $J_{i,j}$. $Out_{i,j}$ is the delay to output to the plant from the previous sampling period to the current sampling period (event (3)s from jobs $J_{i,k-1}$ to $J_{i,k}$).

$\acute{h}_{i,j}$ is the sampling interval for job $J_{i,j}$, which is the delay from the input sampling event from the previous sampling period to the current sampling period (event (1)s from jobs $J_{i,k-1}$ to $J_{i,k}$). The sampling interval is distinct from the sampling period h_i. The latter is always constant while the former varies from one period to the other.

Using the terminology developed in this Section, the three types of jitters considered in this paper can be defined as follows:

- **Sampling Jitter:** Variation in $\acute{h}_{i,j}$ for T_i.
- **Input-output Jitter:** Variation in $InOut_{i,j}$ for T_i.
- **Output Jitter:** Variation in $Out_{i,j}$ for T_i.

When there is only one digital control system in the set, input-output and output jitter also exist. This is because the A/D and D/A conversions, RTOS scheduler execution, and control law calculation can jitter. However, sampling jitter does not exist in this scenario because no other control tasks exist to interfere the trigger for the A/D conversion. As additional digital control systems are introduced into the set, the three types of jitters increase in tandem because each task interferes with the other tasks' execution.

3.2. Model Structure

The digital control system set used in this paper (also used in the performance evaluation) consists of five SISO independent digital control systems that execute in parallel. All five digital controllers of the five digital control systems execute on a single processor. The model diagram of the structure of a digital control system in a set is depicted in Figure 2.

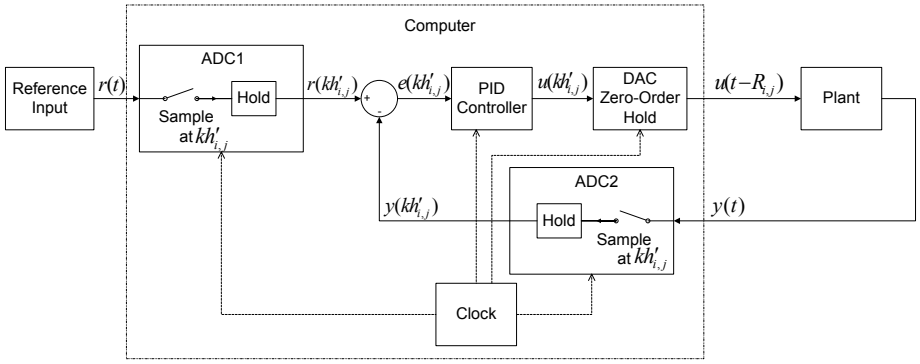

Fig. 2. Model of a Digital Control System.

The two ADCs and DAC are modeled as sample and hold, and zero order hold devices respectively. Both types of devices have variable execution times so that they model real ADCs and DACs. The periodic workflow of the model occurs as follows:

1. Discretizing the inputs $r(t)$ and $y(t)$ using the ADCs via sample and hold to produce $r(k\acute{h}_{i,j})$ and $y(k\acute{h}_{i,j})$.
2. Comparing $r(k\acute{h}_{i,j})$ and $y(k\acute{h}_{i,j})$ to produce the error signal $e(k\acute{h}_{i,j})$.
3. Executing the control law by the PID controller to produce the control signal $u(k\acute{h}_{i,j})$.
4. Converting $u(k\acute{h}_{i,j})$ to a continuous time signal $u(t)$ using the DAC and outputting it to the plant.

3.3. Configurations

A configuration is determined by the trigger method, which is either software or hardware to trigger the ADCs and DAC for all digital control systems within a set. These four configurations are described as follows:

Configuration 1: This is the most conservative configuration out of the four because minimization of jitter is the main priority. Hardware is used to trigger both the ADCs and DAC at precise instants in time.

Configuration 2: Software is used to trigger the DAC whereas hardware is used to trigger both the ADCs. Sampling jitter is completely minimized in this configuration while input-output and output jitter are much larger compared to configuration 1.

Configuration 3: Hardware is used to trigger the DAC while software is used to trigger both the ADCs. In this configuration both sampling and input-output jitter are present whereas output jitter is minimized.

Configuration 4: This is the least conservative configuration among the four due to not prioritizing jitter minimization. Software is used to trigger both the ADCs and DAC and hence, sampling, input-output, and output jitter are present.

4. Jitter Compensating PID Controller

The design of the proposed solution is carried out in two main steps, which are, continuous-time design and discrete-time design. The purpose of the continuous-time design is to determine the appropriate rise time, which is used to derive the sampling period. The equation to calculate the sampling period according to Åström and Wittenmark (2011) is given by (2),

$$h = \frac{T_r}{N} \tag{2}$$

T_r is the rise time and N is an integer. Åström et al. suggests $[4,10]$ whereas J. Liu suggests $[10,20]$ for the value of N. This research uses the value of 10 because it is the middle value of the two approaches, and it results in a good trade-off between scheduling periodic software tasks and quality of control. From a control systems point of view having a low sampling period yields good control performance. On the other hand, a low sampling period creates problems for scheduling multiple tasks as it increases the processor utilization.

The resulting rise time after the design is usually less or equal to the required rise time. Therefore, according to (2) a lower sampling period can be chosen if the rise time is decreased. Consequently, the performance of the digital control system is improved by utilizing a smaller sampling period.

After obtaining the sampling period from the continuous-time design, the next step is to discretize the continuous-time plant. Subsequently, the worst case delay between input sampling to outputting to the plant is taken as one sampling period, and placed in the forward loop between the PID controller and discrete-time plant. This worst case delay between input sampling to output is the one sampling period delay assumption. In the z domain the one sampling period delay is modeled as z^{-1}. The diagram of this discrete-time design is shown in Figure 3.

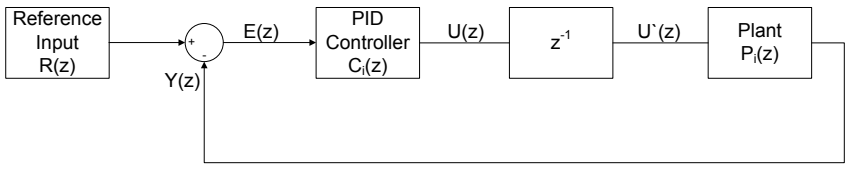

Fig. 3. Discrete-Time Control System with a Delay.

The next step in the design process is to tune the PID controller to satisfy performance requirements. This research used the PID tuner described by Mathworks (2013) from Simulink. Tuning the PID controller involves using a graphical user interface with a slider that updates the plant output response when the position of the slider changed.

Finally, the designed PID controller is interfaced with a continuous-time plant and reference input through two ADCs and a DAC. A comparison of the underlying Simulink simulation model of the PID controller that compensates for jitter against a non-jitter compensating counterpart revealed that only the P, I, and D constants were updated. This implies that the execution time of the PID controller remains unchanged. Therefore, digital control systems with already strict timing budgets can be benefited from this solution.

5. Performance Evaluation

A Simulink simulation based evaluation of the proposed solution is carried out to determine its effectiveness in the presence of software induced jitter. The system model described in Section 3 is simulated. The TrueTime simulation platform by Cervin et al. (2003) is used to simulate multiple control tasks executing in parallel on a single

processor. Three types of digital control system sets are compared for the evaluation viz. *NewComp* (uses the proposed solution and four configurations), *UnComp* (does not use the proposed solution but uses four configurations), and *CompOnePeriod* (only uses configuration 1 and the proposed solution; the sampling to output delay is set to one sampling period).

Only the first digital control system in the set controls an actual plant, whereas the software for the other four execute periodically. The only difference between first control task and the four periodic control tasks is the latter do not read from and write to ADCs and DAC respectively. Apart from that everything else that is executed in the task remains the same. The voltage stabilizer example plant presented by Lozoya et al. (2008) and Marti et al. (2010) is used as the plant for the first digital control system. The transfer function of this plant is given by (3),

$$G(s) = \frac{918.27}{s^2 + 90.9s + 918.27} \tag{3}$$

Design parameters including plant response characterizing performance requirements for the voltage stabilizer plant is depicted in Table 1. Reasonable requirements were formulated as these types of requirements were not provided in the literature.

Table 1. Design Parameters for the Voltage Stabilizer Digital Control System Example.

Parameter	**Value**
h_1	3ms
% Overshoot	$< 10\%$
Rise time	< 30ms
Settling time	< 100ms

Two types of experiments are carried out as part of the performance evaluation which are: step input and tracking. In the first experiment, performance is measured using four metrics viz. percent overshoot, rise time, settling time, and integrated absolute error (IAE) cost function. The fourth metric is only used in the tracking experiment because the other three are not applicable. According to Åström and Wittenmark (1995), the IAE cost function integrates the errors up to infinity, however, in this evaluation a finite duration is considered. For the step input experiment the experiment duration is 100 h_1 periods whereas for the tracking experiment 1000 h_1 periods are used as the duration. In addition, the error between the reference input and response of the plant is used in the IAE cost function.

6. Performance Results

Step input experiment performance results are shown in Table 2. Performance results in bold font indicate a violation of performance requirements in Table 1. When compared to Table 1, NewComp and CompOnePeriod types satisfied performance requirements, whereas UnComp was unable to satisfy the percent offshoot requirement.

Table 2. Voltage Stabilizer Digital Control System Average Step Input Performance.

System Type	Metric	Cfg 1	Cfg 2	Cfg 3	Cfg 4
NewComp	% Overshoot	4.10	2.92	2.87	2.19
	RT (ms)	21.18	21.66	21.77	22.14
	ST (ms)	87.19	87.75	85.75	85.29
	IAE	0.0203	0.0196	0.0198	0.0196
UnComp	% Overshoot	**12.52**	**11.33**	**11.57**	**10.96**
	RT (ms)	26.50	26.99	26.95	27.24
	ST (ms)	93.90	93.08	93.99	93.89
	IAE	0.0282	0.0272	0.0276	0.0272
Comp1P	% Overshoot	7.69	N/A	N/A	N/A
	RT (ms)	19.95	N/A	N/A	N/A
	ST (ms)	91.33	N/A	N/A	N/A
	IAE	0.0218	N/A	N/A	N/A

From a real-time systems perspective, satisfying timing constraints is important, however, results from UnComp show that despite meeting these constraints it is not a sufficient condition to ensure satisfying performance requirements. Therefore, compensation of delays is essential. An overall improvement in NewComp compared to the other two system sets is observed according to the IAE results.

The 95% confidence intervals for the performance data in Table 2 is shown in Table 3. When comparing the confidence intervals for each metric among the three different system set types for each indivudual configuration, there is no overlap in the confidence intervals. This implies that the difference of a particular metric among the three system set types for each configuration is statistically significant.

Table 3. Voltage Stabilizer Digital Control System Step Input Performance 95% Confidence Intervals.

System Type	Metric	Cfg 1	Cfg 2	Cfg 3	Cfg 4
NewComp	% Overshoot	4.10,4.10	2.92,2.92	2.86,2.88	2.18,2.20
	RT (ms)	21.18,21.18	21.66,21.66	21.77,21.78	22.09,22.19
	ST (ms)	86.68,87.71	87.23,88.26	85.07,86.42	83.48,87.10
	IAE	0.0197,0.0208	0.0191,0.0201	0.0192,0.0205	0.0191,0.0202
UnComp	% Overshoot	**12.52,12.52**	**11.33,11.33**	**11.57,11.58**	**10.86,11.07**
	RT (ms)	26.50,26.50	26.99,26.99	26.95,26.96	27.21,27.27
	ST (ms)	93.38,94.41	92.57,93.59	93.32,94.66	93.50,94.28
	IAE	0.0277,0.0287	0.0267,0.0277	0.0269,0.0282	0.0266,0.0277
Comp1P	% Overshoot	7.69,7.69	N/A	N/A	N/A
	RT (ms)	19.95,19.95	N/A	N/A	N/A
	ST (ms)	90.82,91.85	N/A	N/A	N/A
	IAE	0.0213,0.0223	N/A	N/A	N/A

Tracking performance results are shown in Table 4 and it also demonstrates that NewComp performed better compared to CompOnePeriod and UnComp types.

Table 4. Voltage Stabilizer Digital Control System Average Tracking Performance.

System Type	Cfg 1	Cfg 2	Cfg 3	Cfg 4
NewComp	1.064	1.040	1.032	1.019
UnComp	1.433	1.405	1.398	1.381
Comp1P	1.143	N/A	N/A	N/A

Table 5 shows the 95% confidence intervals of the performance results in Table 4. There is no overlap in the confidence intervals among the three different system set types for each configuration. This implies that the difference in performance is statistically significant.

Table 5. Voltage Stabilizer Digital Control System Tracking Performance 95% Confidence Intervals.

System Type	Cfg 1	Cfg 2	Cfg 3	Cfg 4
NewComp	1.049,1.079	1.026,1.055	1.021,1.044	1.009,1.030
UnComp	1.418,1.448	1.390,1.420	1.386,1.410	1.369,1.392
Comp1P	1.128,1.158	N/A	N/A	N/A

The performance results are not conclusive enough to judge which configuration has the best overall performance as there is overlap in the confidence intervals for the tracking performance results among the configurations for each system type. However, configuration 2 seems to be a popular choice because most modern microcontrollers have a facility to trigger ADCs automatically without software intervention.

7. Conclusion

This paper presents a solution to compensate for software induced delays in digital control systems using PID controllers, and it focuses mainly on jitter compensation. The two main advantages of this solution are: plant response characterizing requirements are satisfied and the PID controller execution time remains unchanged. The latter is a huge benefit for digital controllers that execute on computer systems that already have tight timing budgets. Simulation based experiments showed that the proposed solution satisfied plant response characterizing performance requirements for the voltage stabilizer digital control system example. Performance results also revealed that satisfying timing constraints is a necessary condition to satisfy performance requirements, however, it is not a sufficient condition. Compensation of delays essential to obtain the required performance. According to the IAE metric, an improvement in performance is observed in the proposed solution when used in digital control systems that output the control law to the plant with an input to output delay less than one sampling period compared to digital control systems that have an input to output delay of one sampling period. This implies that reducing the delay between input to output is necessary to improve performance.

References

Aminifar, A., Eles, P., Peng, Z., and Cervin, A. (2013). Control-quality driven design of cyber-physical systems with robustness guarantees. In *Design, Automation & Test in Europe (DATE 2013)*, pages 1093–1098, Grenoble, France.

Åström, K. J. and Wittenmark, B. (1995). *PID Controllers: Theory, Design, and Tuning*. The Instrument Society of America, Research Triangle Park, NC, second edition.

Åström, K. J. and Wittenmark, B. (2011). *Computer-Controlled Systems*. Dover Publications, Mineola, NY, third edition.

Cervin, A., Henriksson, D., Lincoln, B., Eker, J., and Årzén, K. E. (2003). How does control timing affect performance? analysis and simulation of timing using Jitterbug and TrueTime. *IEEE Control Systems Magazine*, 23(3):16–30.

Hong, S., Hu, X. S., and Lemmon, M. D. (2010). Reducing delay jitter of real-time control tasks through adaptive deadline adjustments. In *Real-Time Systems (ECRTS), 2010 22nd Euromicro Conference on*, pages 229–238, Brussels, Belgium.

Liu, Z., Zhao, H., Li, P., and Wang, J. (2009). An optimization model for IO jitter in device-level RTOS. In *Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on*, pages 1528–1533, Las Vagas, Nevada, USA.

Lozoya, C., Velasco, M., and Marti, P. (2008). The one-shot task model for robust real-time embedded control systems. *IEEE Transactions on Industrial Informatics*, 4(3):164–174.

Marti, P., Fuertes, J. M., Fohler, G., and Ramamritham, K. (2001). Jitter compensation for real-time control systems. In *Real-Time Systems Symposium, 2001. (RTSS 2001). Proceedings. 22nd IEEE*, pages 39–48, London, England.

Marti, P., Velasco, M., Fuertes, J., Camacho, A., and Buttazzo, G. (2010). Design of an embedded control system laboratory experiment. *IEEE Transactions on Industrial Electronics*, 57(10):3297–3307.

Mathworks (2013). *Simulink©Control Design™Users Guide*.

Mathworks (2014). Modeling computational delay and sampling effects. consulted 31 January 2014.

Nasri, M. and Kargahi, M. (2012). A method for improving delay-sensitive accuracy in real-time embedded systems. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on*, pages 378–387, Seoul, Korea.

Smeds, K. and Lu, X. (2012). Effect of sampling jitter and control jitter on positioning error in motion control systems. *Precision Engineering*, 36(2):175–192.

Wu, Y., Buttazzo, G., Bini, E., and Cervin, A. (2010). Parameter selection for real-time controllers in resource-constrained systems. *IEEE Transactions on Industrial Informatics*, 6(4):610–620.