A Case-Study in Multidisciplinary Modeling of Dynamic Embedded Systems[1]

Björn Bukkems,[a,§] Heico Sandee,[a] Jan Beckers,[b] Zhaorui Yuan,[b] Berry van der Wijst,[c] René van de Molengraft[a]
[a] Technische Universiteit Eindhoven, [b] Océ Technologies B.V., [c] Philips CFT
[§] Department of Mechanical Engineering, Dynamics and Control Technology Group
WH -1.132, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Phone: +31 40 247 2841, Fax: +31 40 246 1418
b.h.m.bukkems@tue.nl

Abstract – **In this paper a case-study in multidisciplinary modeling of dynamic embedded systems is described. This case-study represents our initial step towards the development of a design methodology for this type of systems, which is the aim of the research project "Boderc". The objectives and composition of the project are given, together with a motivation why such a method is needed. After this, the modeling of parts of the paper path of a high-volume black and white multi-functional copier is described. The goals and the results of the case-study are discussed in detail. Next, the way the knowledge from different disciplines is linked is shown, and the possible couplings between models are explained. In order to judge the predicting capabilities of the models, experiments for validation have been carried out. The obtained measurements show that the created models form a fair description of the system behavior.**

I. INTRODUCTION

Throughout the past decade, a considerable amount of attention has been given to the design of effective and efficient product development processes, e.g. see [1] and the references therein. However, although the academic literature has generated many contributions to the understanding of these processes, numerous (industrial) organizations face difficulties following them. In this paper, we focus on a case-study for the development of a multidisciplinary design methodology for dynamic embedded systems. In our approach, also known as "industry-as-laboratory" [2], academic research is directly confronted with requirements and constraints from industry.

Within an industrial environment, a traditional product development process often starts with the design of the mechanics. In this phase the mechanical layout of the system under consideration is designed. After adding actuators, sensors, processor boards, etc., software needs to be developed for the integration of all (dynamic) embedded subsystems. During this phase of the development process, it often becomes clear that in an earlier stage assumptions have been made that do not hold on the realized design. Consequently, neither can the desired integration. As a result, returning to the earlier stages, and carrying out a redesign of certain subsystems, is often inevitable. This process is not only costly, but it also slows down the product development phase.

The sequential way of working, described above, is likely not the optimal method to carry out the development process. This might originate from the fact that engineers from different disciplines are often not familiar with each other's way of describing, modeling, and designing systems. This knowledge gap might be caused by the fact that the various disciplines use different formalisms, such as differential equations, process algebra, state-machines, etc. Some initial steps are taken in academia to mix these described formalisms, leading to the lively field of hybrid systems [3]. This active research area, however, is still in its infancy and the gap towards industrial design of dynamic embedded systems is huge. This forms one of the major challenges in the 21st century [4].

In our opinion, a better and faster development process can be realized when engineers from different disciplines work in a parallel fashion. This implies a close coupling of the expertise of those disciplines in an early design stage. Academic research is required to create a model-based design methodology that can effectively bridge the gap between the different disciplines. Such a methodology must take into account the demanding industrial product requirements; e.g. serviceability, exception handling, real-time behavior, and safety [5].

Our first step in developing such a design methodology for dynamic embedded systems is performing a multidisciplinary modeling case-study. In this paper, our first explorations towards the coupling of models from different disciplines are presented. More specifically, we focus on the possibilities of coupling a software model with a mechatronics model. The developed models are validated on an industrial motion system in order to judge their predicting capabilities. If successful, a useful starting point for the following step in the development of the design methodology has been realized. The above mentioned modeling case-study is the main subject of this paper. The outline is as follows: in Section II, the "Boderc" project, i.e. the project which the presented work is part of, will be discussed. An example of the modeling activities, carried out in the scope of this project, will be discussed in Section III, whereas in the fourth section we will present the experimental results, used to validate the models. In Section V, we will show an example of interdisciplinary coupling and Section VI gives a broader evaluation of the modeling activities. Conclusions will come at the end.

[1] This work has been carried out as part of the Boderc project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Senter TS program.

II. THE BODERC RESEARCH PROJECT

The modeling and validation activities, presented in this paper, were carried out as part of the research project "Boderc". The name of the project is an acronym and stands for "**B**eyond the **O**rdinary: **D**esign of **E**mbedded **R**eal-time **C**ontrol". The goal of the project is to develop design methods and modeling techniques that can effectively bridge the gap between the different disciplines [5]. The multidisciplinary coupling between the three main disciplines involved in the project, i.e. Mechanical engineering (M), Electrical engineering (E), and Software engineering (S), is schematically depicted in Fig. 1. Such multidisciplinary modeling techniques can help in making decisions in an early design phase, taking into account aspects over more than one discipline. This can lead to the development of an effective and efficient design methodology for dynamic embedded systems.

The first step in realizing the goals is to improve the communication between engineers from different disciplines, which means leaning to understand each other's vocabulary and each other's way of looking at systems. This step creates an awareness of potential problems other disciplines face when designing dynamic embedded systems. A second step is to make models in the early stage of the development process that exploit the input from the various disciplines.

To realize the Boderc goals, a large research team has been composed which consists of both academic and industrial project partners. This team is located at the Embedded Systems Institute, which carries the project management responsibility, and enables a close cooperation between academia and industry. The total occupation of the project consists of approximately 17 full time employees. From the industrial project partners, Océ Technologies B.V. defined the problem statement. Furthermore, this company provided all background information related to the case-study, and equipped the project with a representative industrial carrier. This is the Océ VarioPrint^{TM} 2090, which is a high-volume black-and-white multifunctional copier, depicted in Fig. 2.

III. CASE-STUDY: MODELING MOTOR BEHAVIOR

One example of the modeling activities carried out in the first year of the Boderc project is about the controlled behavior of one of the motors in the paper path. More precisely, the activities deal with the last motor before the Fuse Pinch, which is called Motor 5. In the Fuse Pinch the image is printed onto the sheet. Therefore Motor 5 has large influence on the printing accuracy. A schematic representation of the related part of the paper path is shown in Fig. 3. In this figure, two pinches can be distinguished: Pinch 5, used to transport the sheet through the paper path, and the Fuse Pinch. The Heater, depicted in the figure as well, heats the sheet to its optimal printing temperature, whereas the fourth optical sensor in the paper path, OS4, detects the arrival of a sheet. The Fuse Pinch is driven by the Main Motor, which is the motor that imposes the velocity of the image. Pinch 5 and the Heater are driven by Motor 5, which enables final adjustments to the sheet velocity. Consequently, the sheet meets its corresponding image in the Fuse Pinch right on time. The upper of the two lines in the bottom of Fig. 3 shows the distance from Pinch 5 and OS4 to the Fuse Pinch, whereas the lower one shows the corresponding nominal times the leading edge of the sheet has to travel to this pinch.

In this case-study, two types of models have been created: qualitative models, which are models for understanding, and quantitative models, which are models for simulation. The first type of models gives a better insight into the how and why of the machine parts and process. The quantitative models, on the other hand, can be executed and give a numerical prediction of the behavior of the system.

A. Qualitative modeling: Describing the Process

Before initiating the real explorations on the other disciplines' way of modeling, first a qualitative model is

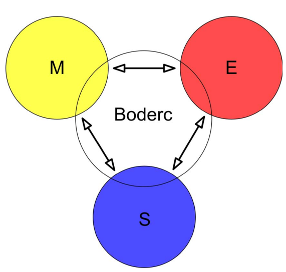

Fig. 1. A schematic representation of the desired coupling between Mechanics (M), Electronics (E), and Software (S)

Fig. 2. The Océ VarioPrint^{TM} 2090

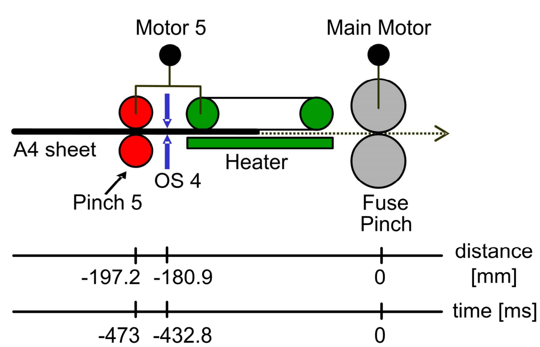

Fig. 3. Schematic representation of Motor 5 and its environment, together with distances and traveling times to the Fuse Pinch

made to understand what happens in the relevant part of the paper path, when printing one A4 sheet. More specifically, the model should tell us where the sheet is, by which pinch or heater it is driven, and which motor is in control. This model has mainly been built from Océ documentation. Fig. 3 shows that the leading edge of a sheet enters the region of interest at Pinch 5. When this edge of the sheet triggers OS4, the control software knows the exact sheet position, as well as the image position. The relative positions of the sheet and the image can thus be synchronized. Next, the sheet enters the Heater and its temperature is increased to approximately 100°C. This is necessary for the image - which consists of toner particles carried by a rubber transport belt - to be melted onto the sheet correctly. The melting process takes place in the Fuse Pinch, which is located just behind the Heater.

The movement of the sheet through the part of the paper path under investigation is shown graphically in Fig. 4. More specifically, this figure shows which pinch or heater is in contact with the A4 sheet, as a function of the distance to the Fuse Pinch. The 1 and 0 on the vertical axis imply that the pinch or heater is or is not in contact with the sheet, respectively. The figure also shows that an A4 sheet can have contact with more than just one pinch at a time. For example, when the leading edge of the sheet has just entered the Fuse Pinch, the sheet is driven by Pinch 5, the Heater, and the Fuse Pinch simultaneously. Since Pinch 5 and the Heater on the one hand, and the Fuse Pinch on the other hand are driven by Motor 5 and the Main Motor, respectively, it is important to know which motor prescribes the velocity of the sheet, since the motors are not running at the same velocity. This is also indicated in Fig. 4. It can be seen that at the moment the sheet enters Pinch 5, Motor 5 is in control, and remains in control until the sheet is just a few millimeters in the Fuse Pinch. From that moment on, the Main Motor imposes the sheet velocity.

From this qualitative model, a better idea of what is happening in the process has been obtained. Based on this model, our modeling activities can be continued, looking for the relevant interaction between the various disciplines. In the remainder of this section, two models will be presented regarding the controlled behavior of Motor 5. The first model has been built from a control engineering viewpoint, describing the controlled dynamic behavior of Motor 5. The second model has been built from the software engineering viewpoint, focusing on the execution time of Motor 5 control. In this software module, the control algorithm for Motor 5 is implemented.

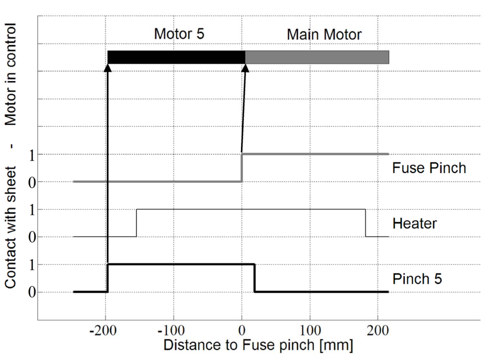

Fig. 4. Pinches and Heater in contact with an A4 sheet as a function of the distance to the Fuse Pinch, together with the motor in control

B. Quantitative modeling: Dynamic Behavior of Motor 5

The control engineering model is a quantitative model that describes the dynamic behavior of Motor 5. It was built in Matlab/Simulink and it consists of several components. First of all, it contains a coupled model of the electrical part and the mechanical part of the motor. The electrical part contains the motor inductance and resistance, whereas the mechanical part contains the inertia and damping. Both parts are linear differential models described by transfer functions from voltage to current, and from torque to angular velocity, respectively. Besides the motor model, a reference trajectory generator was designed, calculating the desired sheet position, velocity, and acceleration. The input of the Proportional-Integral-Derivative (PID) feedback controller is the difference between the desired sheet position and the actual one. The latter is calculated from the angular velocity of Motor 5, assuming no-slip conditions. Together with an acceleration and velocity feedforward controller, the feedback controller should deliver the control input needed for accurate tracking of the reference profile.

The results of a simulation of this dynamic model of Motor 5 can be seen in Fig. 5. The horizontal axis shows the time in seconds relative to the moment the leading edge of the sheet enters the Fuse Pinch, whereas the vertical axis shows the velocity of the sheet in millimeters per second. Although a position controller was used, we show the sheet velocity, since this is the main performance variable in the sheet profile specifications. The vertical lines indicate the locations of Pinch 5, OS4, and the Fuse Pinch, respectively. The reference velocity profile is depicted in black. This profile represents the nominal sheet velocity, obtained from Océ documentation. In the real copier, the reference profiles are calculated on-the-fly. The simulated sheet velocity is represented by the grey line. This velocity is calculated from

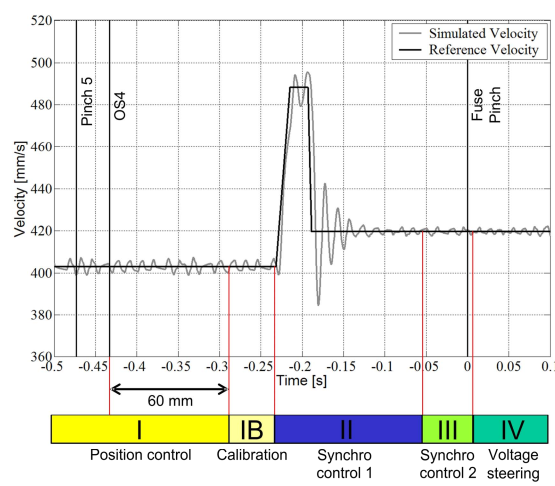

Fig. 5. Simulation results of the dynamic model of Motor 5, together with the four different control modes of Motor 5 control

the Motor 5 velocity, again assuming no-slip conditions. The change in velocity indicates that the sheet has to accelerate and decelerate in order to catch up with its corresponding image. Quite some overshoot can be observed, but this will not be a problem, as long as the sheet position is correct when entering the Fuse Pinch. The results of this simulation were obtained using a fixed reference profile, i.e. it was not corrected for varying image velocities. The behavior after the sheet enters the Fuse Pinch, i.e. a change in Motor 5 velocity when the Fuse Pinch takes over control, has not been incorporated either.

In designing the Motor 5 motion controller, the assumption was made that a controller sampling time of 1 [ms] can be achieved in the real system. More specifically, given a sampling frequency of 1 [kHz], a controller can be designed that achieves the required bandwidth of the closed-loop system, below which disturbance attenuation takes place. Hence, we expect the embedded processor to have sufficient processing power to calculate both Motor 5 control and all other tasks it has to perform, within 1 [ms]. To verify if this assumption is valid, a model predicting the execution time of Motor 5 is derived in the next subsection.

C. Qualitative modeling: Motor 5 Control Execution Time

The model described in this subsection is a qualitative model. It should give better insight into what happens inside the software when calculating Motor 5 control, and more specifically, how much time it takes to perform all calculations. From Océ documentation and from analysis of the existing software, we know that Motor 5 control consists of 4 different control modes. These modes start when the sheet is at predefined locations from OS4 and are indicated in Fig. 5. The first mode (I) is called position control. During the last part of this control mode, a time interval of calibration begins at a predefined distance of 60 [mm] from OS4. This calibration, indicated as mode IB, is used to determine the voltage needed to drive the motor during control mode IV. This fourth control mode is called Voltage steering and starts just after the leading edge of the sheet enters the Fuse Pinch. As mentioned before, this pinch is driven by the Main Motor which now takes over control from Motor 5. After mode IB, the sheet position is synchronized with the image position, in order to meet the image in the Fuse Pinch at the exact right moment. During the first synchronization mode, the sheet velocity is briefly increased to its maximum bound in order to correct for position mismatches with the image position. In the second synchronization mode, the bounds on the velocity are much more conservative. During this control mode the sheet enters the Fuse Pinch and the actual printing process starts, so only fine-tuning can be performed. However, Fig. 5 does not show a difference in the simulated velocity during both synchronization modes. This is because the bounds on the sheet velocity in both synchronization modes are not taken into account in the simulation, in which only the (nominal) reference velocity has to be tracked.

From the control software implementation and from knowledge obtained from Océ documentation, the software execution times can be estimated. The constructed model is shown graphically in Fig. 6. The horizontal axis shows the time needed to perform certain actions, e.g. taking samples or generating a setpoint. On the vertical axis, the four control modes are shown. As mentioned before, the controller sampling frequency is equal to 1 [kHz] for all modes. This frequency is the result of the occurrence of a hardware timer interrupt after every millisecond. Within each millisecond, the control inputs for different motors are calculated sequentially. After reading all motor encoders, first the control input for Motor 5 is calculated, since this motor has the highest influence on the printing accuracy. By calculating this control input first, least delay between measuring and actuating will be experienced, as well as least variation on this delay. After calculating Motor 5 control, the control input for Motor 3 is calculated. Finally, the control inputs for Motor 1 or Motor 6 are calculated in turn, since the sampling frequency of these motor control loops is 500 [Hz].

The model depicted in Fig. 6 shows that in every control mode the calculation of Motor 5 control is preceded by taking samples, i.e. obtaining information from all motor encoders in the paper path. After this, the calculation of Motor 5 control starts. In estimating the relative execution time for each control mode with respect to the other modes, the existing machine software is analyzed. For each mode, the number and the nature of the calculations are investigated. From this analysis, it follows that the amount of time needed in each control mode differs, as can be seen in Fig. 6. The model assumes a constant execution time in mode I. During calibration, a little more time is needed to store the control inputs. At the beginning of the first and second synchronization mode the setpoint generation takes a little longer. This instant corresponds to the changes on the bounds of Motor 5 velocity. Mode IV takes least execution time, since no setpoint generation and calculation of the control input are expected to take place. According to Océ documentation, the time slot for Motor 5 control is 162 ± 70 [μs]. Consequently, the remainder of the millisecond is left for calculation of the other motor controllers and for other tasks, e.g. error handling and communication with other processors.

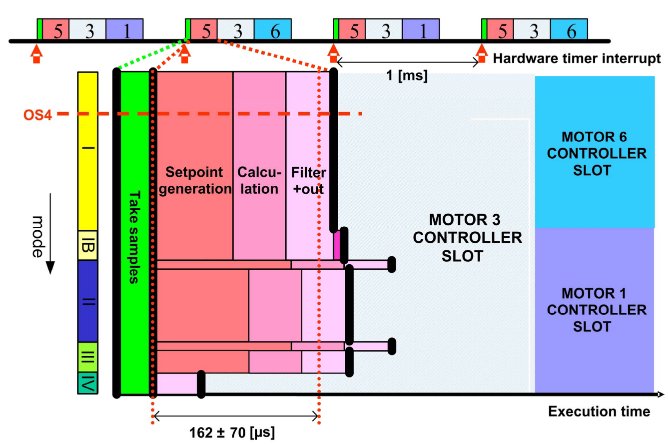

Fig. 6. Graphic representation of the Motor 5 control execution time model

IV. EXPERIMENTAL RESULTS

A. Model validation: Motor 5 Control Execution Time

Since the software model described in the previous section is qualitative, an attempt has been made to make it more quantitative by performing a validation experiment. The first step in the preparation of this experiment consists of a slight adaptation of the existing machine software. Right before and just after taking samples and at the end of calculating the control inputs for the motors of interest, several I/O pins on one of the generic I/O boards of the copier are toggled by the software. These I/O pins are connected to the parallel port of a PC, on which an RTAI Linux application is polling them with a frequency of 500 [kHz].

The results of the measurement, obtained when printing a single A4 sheet, are shown in Fig. 7. This figure shows both resemblances and differences with the model. It can be seen that mode IV requires somewhat less execution time than mode III. As predicted, the spikes corresponding to the switches from the first to the second mode, and from the second to the third mode are indeed present. From the experimental results, we can see that the prediction of the execution time in mode I shows least overlap with measured data. It is not constant and neither can the increase in execution time during the calibration period be observed. The model did also not predict the difference in execution time during mode II.

The measurements show that the execution time of Motor 5 control is approximately 145 ± 110 [μs]. Comparing these values with the values obtained from Océ documentation, it can be seen that the mean value is a little bit smaller, but the deviations from this mean are significantly larger. The mean total amount of time needed to calculate all motor controls is approximately 340 [μs], whereas the maximum value is equal to 475 [μs], as can be seen in Fig. 7. From these experimental results it can be concluded that the Motor 5 control loop can indeed run at 1 [kHz]. In worst-case situations, a little bit more than 0.5 [ms] is left for all other tasks the processor has to perform. We assume this is enough, but to verify this assumption, additional analysis of software timing behavior has to be carried out.

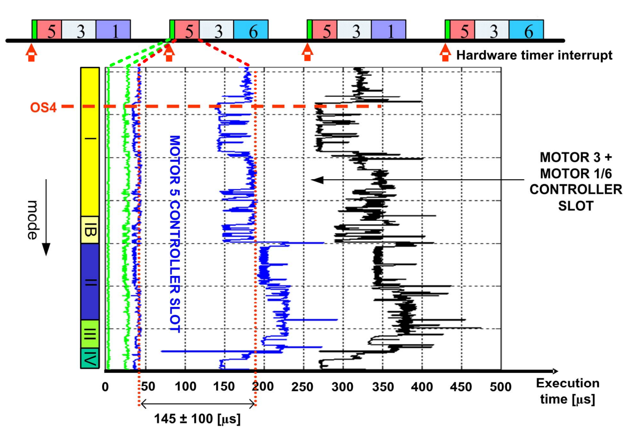

Fig. 7. Experimental results of the Motor 5 control execution time measurement

B. Model validation: Dynamic Behavior of Motor 5

In order to validate the model describing the dynamic behavior of Motor 5, experiments were carried out. The results of this second validation experiment, obtained when printing a single A4 sheet, are shown in Fig. 8. The horizontal axis of this figure shows the time in seconds, relative to the moment the leading edge of the sheet enters the Fuse Pinch. The vertical axis shows the velocity of the sheet in millimeters per second. The four different control modes are shown as in Fig. 5. The sheet velocity is represented by the black line, and is calculated from the measured Motor 5 position, again assuming no-slip conditions. The velocity of the image, reconstructed from the image position, is represented by the grey line.

The results show quite some overshoot of the sheet velocity. As in the simulation results, this is again no problem, as long as the sheet position is correct when entering the Fuse Pinch. In the second part of mode II and in mode III, i.e. the first and second synchronization modes, the sheet velocity is indeed seen to be synchronized with the image velocity. Note that the sheet position is also synchronized with the image position, although this cannot be seen in Fig. 8. In mode IV, the sheet velocity is not represented by the black line anymore, but it has the same value as the image velocity. This is due to the fact that the Fuse Pinch is in control of both the sheet and the image. Consequently, in this mode, the black line represents the Motor 5 velocity, converted into sheet velocity units. It can be seen that this velocity is approximately equal to the velocity during the calibration interval. This is to be expected, since the average control input stored during the calibration period is again applied to Motor 5. Slight changes in velocity are the result of a change in load, since in mode IV the Fuse Pinch is pulling the sheet. From the moment control mode IV starts, slip occurs between the sheet and the parts driven by Motor 5 that are still in contact with the sheet, see Fig. 4. This is caused by the difference in velocity between the sheet and the parts driven by Motor 5. The slip prevents folding of the sheet.

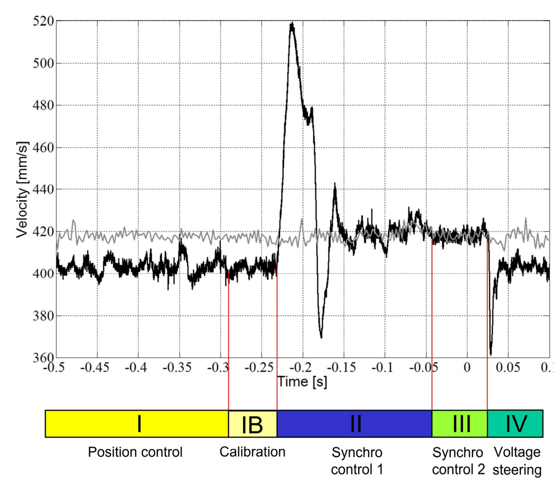

Fig. 8. Measurement results of Motor 5 (black) and image (grey) velocity, together with the four different control modes

The main difference between the simulation and the experimental results can be observed in the two synchronization modes. In the simulation, only an ideal reference velocity profile has to be tracked, i.e. the position and velocity of the image are not taken into account. In reality, however, the sheet is synchronized with the image.

V. INTERDISCIPLINARY COUPLING

To create an interdisciplinary coupling, the measured software execution time is incorporated in the quantitative model describing the dynamic behavior of Motor 5. This is done by delaying the summed output of the feedback and feedforward controller with a varying time, which is obtained from the measurements. The effect of the varying execution time on the dynamic response can now be analyzed, and a better prediction of the actual system behavior can be obtained. The results of this simulation are shown in Fig. 9. In the upper plot, the reference velocity profile and the simulated velocity without delay are represented by the black, thin line and the grey line, respectively. The black, thick line shows the velocity obtained from the simulation with the delay included. In the lower plot, the difference between the responses of the two simulations is shown. As can be seen from the figure, there is not much difference between the two simulated velocities. In the beginning of the simulation, the two responses seem to be a little out of phase, which causes a somewhat larger difference. The mean of the difference, however, is close to zero. These simulation results indicate that the actual closed-loop system implemented in the copier will probably work correctly when a sampling frequency of 1 [kHz] is used. This indication was already justified by the second experiment of the previous section.

VI. EVALUATION

In the modeling case-study presented, several models have been developed. The created models themselves are rather mono-disciplinary, but links between them do exist. More

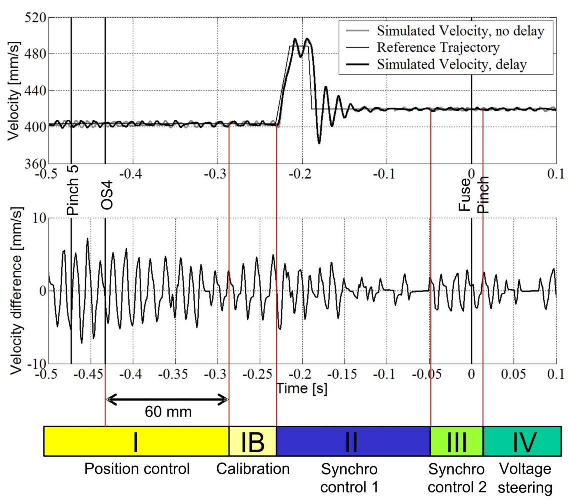

Fig. 9. Simulation results of the dynamic model of Motor 5; with and without the software execution time incorporated in the model

specifically, the model of the control engineer assumed a sampling time of 1 [ms], and the model of the software engineer, together with the corresponding validation results, justified this assumption. A tighter coupling between the models was obtained by incorporating the measured software execution time into the dynamic model. Using this model, the influence of the execution time on the dynamic behavior can be analyzed. In developments of dynamic embedded systems this model can be used to predict performance when an estimation of the delay is present.

The modeling exercise presented in this paper resulted in a quantitative model with inputs from different disciplines. However, this model only contains measurements of the software, so no modeled software behavior is incorporated. This is mainly caused by the approach we followed. Due to the high complexity of the application-specific software, much effort was put into the analysis of it, mainly resulting in qualitative models. During the second phase of the Boderc project, models will be made of new dynamic embedded systems of which no experimental set-up will be available. As a result, more effort will be needed in creating qualitative models, using the knowledge of the different disciplines.

The described modeling case-study has resulted in a better understanding of the work carried out by engineers from different disciplines. Furthermore, it has resulted in models with reasonably good predicting capabilities. Therefore, the goals of the modeling case-study have been realized and a starting point for the second step in the development of the desired design methodology has been created.

VII. CONCLUSIONS

This paper presents a case-study in multidisciplinary modeling of dynamic embedded systems. In this case, which is part of the Boderc project, the effort of engineers from different disciplines is combined. The goal of this case-study was to investigate the possibility of making multidisciplinary models of a part of a high-volume industrial copier. Several models have been created, and validation experiments have been conducted. Eventually, a model with input from both disciplines has been developed which, in future case-studies, can be used to predict dynamic behavior taking into account delays present in the embedded control software.

VIII. REFERENCES

[1] N.P. Repenning, "Understanding fire fighting in new product development," *Journal of Product Innovation Management*, vol. 18, no. 5, Sept. 2001, pp. 285-300.

[2] C. Potts, "Sotware-engineering Research Revisited," *IEEE Software*, vol. 10, no. 5, Sept. 1993, pp. 19-28.

[3] R. Alur, G. Pappas (Eds.), *Proc. 7th Int. Workshop HSCC 2004*, Philadelphia, PA, USA, March 2004.

[4] R.M. Murray et. al., "Future directions in control in an information-rich world," *Control Systems Magazine*, vol. 23, no. 2, April 2003, pp. 20-33.

[5] Embedded Systems Institute, Summary of the Boderc Project Plan, [online] http://www.embeddedsystems.nl