An Evaluation Framework for Vision-in-the-Loop Motion Control Systems

Chaitanya Jugade, Daniel Hartgers, Phan Dúc Anh, Sajid Mohamed, Mojtaba Haghi,
Dip Goswami, Andrew Nelson, Gijs van der Veen, and Kees Goossens

***Abstract*—Industrial applications and processes such as quality inspections, pick and place operations, and semiconductor manufacturing require accurate positioning control for achieving the high throughput of the assembly machines. Vision-based sensing is considered to be a potential means to achieve robust positioning control which is referred to as a vision-in-the-loop (VIL) system. In such motion systems, the point-of-control and the point-of-interest are often different due to several physical factors. In this case, validation of a system is done only when a machine prototype is available. A physical prototype is often expensive and infeasible in real-life. This paper proposes an evaluation framework for VIL systems targeting a predictable multi-core embedded platform. The presented framework offers model-in-the-loop (MIL), software-in-the-loop (SIL), and processor-in-the-loop (PIL) simulation features for evaluating the closed-loop performance of industrial motion control systems. As a deployment platform, we consider a predictable embedded platform CompSOC. The predictable nature of the CompSOC platform guarantees periodic and deterministic execution of the control applications and allows verification of the timing properties and performance of the VIL system. Additionally, the framework offers automatic code generation feature targeting the CompSOC platform. Closed-loop simulation setup models the system dynamics and camera position in the CoppeliaSim physics simulation engine and simulates the system software in C and MATLAB. CoppeliaSim runs as a server and MATLAB as a client in synchronous mode. We show the effectiveness of our framework using a vision-based motion control example.**

***Index Terms*—Vision-in-the-loop, processor-in-the-loop simulation, embedded control, industrial motion control system.**

I. INTRODUCTION

Vision-based motion control is gaining significant momentum, and in recent years various vision-based approaches in control applications have been presented [1]. Especially in many automated manufacturing systems, vision-based accurate position control is one of the essential techniques. In such an industrial process, assembly machines that assemble products require accurate positioning control to achieve high productivity [2], [3], [4]. But one of the challenges is positional errors that occur due to uncertain operating environments, e.g., ageing deterioration, external disturbances such as vibrations, the elastic deformation of the wafer surface, mechanical joint misalignment, etc. These factors are non-negligible since their magnitude is relatively large compared to the target accuracy. Often in these systems, the point-of-control (i.e., where a direct position controller is available) and the point-of-interest (i.e., the position of primary interest to the application) are different due to several physical factors. Here, the point-of-interest cannot be directly controlled/measured. A vision-based sensing system is a potential means to measure the position at the point-of-interest, and the measurements can be used as a feedback for the point-of-control. Therefore, in today's assembly machines, vision-based positioning control is increasingly becoming common to improve positioning accuracy, which senses the point-of-interest and overcomes these unmeasurable errors/disturbances caused by the operating environment. By obtaining "relative" positional errors between a controlled object and a target position from visual information, a vision-in-the-loop system aims to accomplish robust and precise positioning control in uncertain environments.

Another challenge is the embedded implementation of the feedback control system. In this paper, we consider the closed-loop control system with a vision sensor in feedback which is referred to as a vision-in-the-loop (VIL) system. A VIL system performs three operations [5] – (i) a sensing task i.e., the image processing algorithm, which processes the image captured from the camera sensor, (ii) a control computation task, which executes the control algorithm, and (iii) an actuator task, which applies the control action obtained from the controller to the system. The common challenge in the embedded implementation of a VIL system is to achieve strictly periodic and jitter-free execution of the control applications of the above three operations. Embedded platforms that offer deterministic execution of algorithms and composability for multi-application scenarios are desired for the deployment of the control applications [6]. The CompSOC platform is a predictable and composable multiprocessor system-on-chip (MPSoC) platform suitable for the development of embedded control applications with strict performance requirement [7]. For functional validation of such systems, an evaluation framework is crucial to model the behavior of the operating environment and to allow simulation of the system behavior with different ranges of the relevant parameters to test, validate, and debug the performance of overall closed-loop systems for industrial applications.

In this paper, we propose an evaluation framework for a VIL motion control system that is mapped on the CompSOC platform. The developed framework allows the model-in-the-loop (MIL), software-in-the-loop (SIL), and the processor-in-

C. Jugade, D. Hartgers, P. D. Anh, S. M. Haghi, D. Goswami, A. Nelson, and K. Goossens are with the Department of Electrical Engineering, Eindhoven University of Technology (TU/e) (e-mail: c.jugade@tue.nl, d.hartgers@student.tue.nl, d.phan.duc.anh@student.tue.nl, s.m.haghi@tue.nl, D.Goswami@tue.nl, A.T.Nelson@tue.nl, K.G.W.Goossens@tue.nl).

S. Mohamed and G. van der Veen are with ITEC B.V., Netherlands (e-mail: sajid.mohamed@itecequipment.com, gijs.van.der.veen@itecequipment.com).

the-loop (PIL) simulations to verify and debug the closed-loop control system under test.

The MIL simulation tests the developed control law with the simulated plant model. The SIL simulation, on the other hand validates the system behavior taking into account the effects of the operating environment. The SIL framework supports C and MATLAB code running in a host PC (for the image processing and controller) interfaced with CoppeliaSim. In the PIL simulation, the controller code and vision processing algorithm runs on the target platform in isolation but not in real-time. The PIL configuration allows measurement of the parameters such as execution time, and memory usage of the control and vision algorithm on the embedded platform as they will be in the final implementation.

The proposed framework offers the following key features:

- The SIL simulation to evaluate the effects of vision sensor feedback and the operating environment on the motion control system. To model the system and vision sensor setup, a physics simulator CoppeliaSim is used.
- The PIL simulation of C code for discrete-time controller and image processing algorithm on an instance of the CompSOC platform. This feature further establishes interfacing with CoppeliaSim using application programming interfaces (APIs).
- Automatic code generation for the VIL system software targeting the CompSOC platform.

II. RELATED WORK

Recent works are reported on evaluation framework of vision-based control systems. Authors in [8], [9] proposed the IMACS framework for evaluating the performance of image approximation in a closed-loop image-based control system for automotive control applications. The framework helps to evaluate the impact of image approximation in the image-based control system. The performance has been evaluated by implementing the framework on NVIDIA platforms [10], [5] – Drive PX2 and AGX Xavier. Such platforms have limitations in terms of predictability and composability of the application software. In the vision-based control system, it is important to maintain jitter-free periodic execution of sensing and control tasks. Our framework provides the support to evaluate the performance of the vision-based control system on the predictable and composable platform for industrial control applications. Using the framework, one can achieve the deterministic periodic execution of the sensing and control operations. Also, the framework allows scheduling of the different tasks on multi-processors and evaluation of control performance and resource utilization on the platform.

III. CASE STUDY

We evaluate VIL system performance targetting the ITEC ADAT3-XF (Automatic Die ATtach - ADAT) die bonder platform as a case study [11]. Fig. 1 shows the ADAT die bonder platform used for semiconductor assembly. To evaluate the performance of the VIL system, we have approximated the wafer stage in the ADAT platform as a second-order mass-damping system, where a mass is moved across a surface by applying force. The system also experiences damping and friction forces that reduce the impact of the externally applied force. The dynamics of the mass-damper system models the movement across one axis of a mass sliding across a surface.

Fig. 1. ITEC's ADAT3-XF die bonder platform. [11]

IV. AN EVALUATION FRAMEWORK

In this section, we explain the proposed framework for evaluating the VIL system for motion control systems. Fig. 2 shows the proposed framework prototyped on a PYNQ-Z2 FPGA board. The framework consists of different blocks which are explained below.

A. Framework Components

Plant/System: To simulate the motion control system, the physics simulator CoppeliaSim is used. The simulation environment models the system described in Section III including its operating environment. The motion system is represented by the mass-damper system. We model the single-axis movement of the moving conveyor belt that is controlled by a single motor for positioning. Fig. 3 (a) shows the model of a moving mass along one axis and a vision sensor. The mass modelling the belt moves across a surface by applying a force as a control action $u[k]$. The CoppeliaSim is interfaced with MATLAB using the MATLAB remote API.

Vision Sensor: The ADAT platform has a camera in a place to observe the wafer stage. Hence, a CoppeliaSim vision sensor with the appropriate region-of-interest (RoI) is used. Fig. 3 (a) shows the simulation setup that contains a vision sensor preview window displaying the vision sensor data for direct user feedback. Fig. 3 (b) shows the image captured by the camera. The image processing algorithm uses the captured image to extract the position information.

Image Processing: After obtaining the image, the next step is to extract the position information from the captured image of the die position on the wafer stage. For position information extraction, an algorithm based on Hough transforms is used [12]. The algorithm extracts the line segments and returns the x-coordinate of the center point of a product within the RoI. The ratio between the center point and full image width determines the absolute position of the center point with respect to the axis in the simulation setup. It should be noted that any other computer vision algorithm can be used depending on the sensing requirements. In Fig. 3 (b), the image processing algorithm detects line segments shown by blue color vertical lines, and based on detected lines, we compute the x-coordinate of the center point of a product which is the output position $y[k]$.

Controller: A proportional (P), integral (I), and derivative (D) (PID) control method is used in the framework, where the input signal to the plant is the control action computed based on the current measurements of the plant. For the mass damper system model, the control action from the PID controller is a Force signal. As shown in Fig. 3 (a), the applied force i.e., control action $u[k]$, moves the mass across the surface and the vision sensor captures the image of the moving mass when in the RoI. In general, any other control law can replace the PID controller. The discrete-time PID controller equation is given as follows:

$$u[k] = k_p e[k] + k_i \sum_{k=1}^{N} e[k]h + k_d \left(\frac{e[k] - e[k-1]}{h} \right) \tag{1}$$

where $u[k]$ is the control action, k_p, k_i, and k_d are the proportional, integral, and derivative controller gains respectively. h is the sample time, which is 25 ms and $e[k]$ is the error signal i.e., difference between reference signal $r[k]$ and current output state $y[k]$. k is the discrete time step.

Reference Generator: The reference signal $r[k]$ in Fig. 2 is a fixed reference signal, which is 0.04 m. The reference position is decided as a centre of total image width and we consider a pixel ratio as 1 mm.

Data Logger: The measured outputs from the closed-loop simulation are logged for debugging and verification in the Simulink environment.

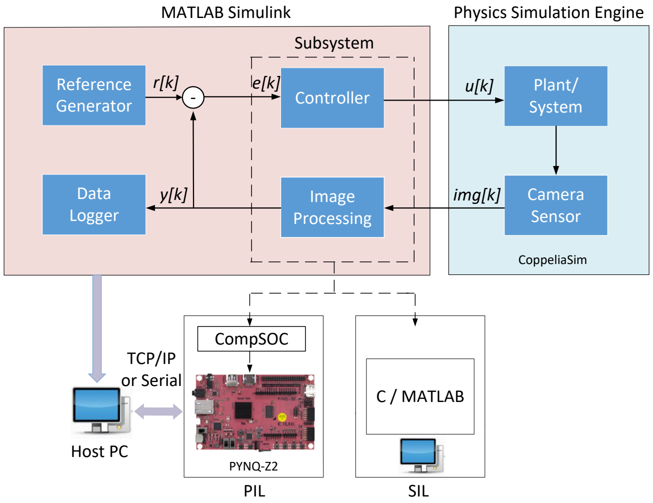

Fig. 2. Proposed framework for the VIL systems. The subsystem for PIL is implemented on CompSOC prototyped on the PYNQ-Z2 FPGA board. The MATLAB Simulink runs on the host PC and communicates with the board through TCP/IP or serial connection.

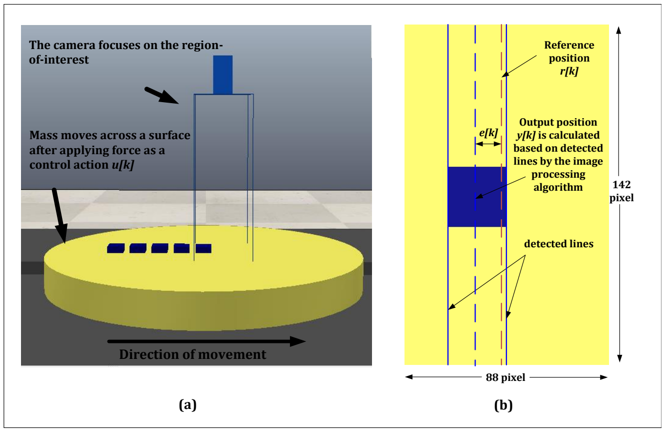

Fig. 3. CoppeliaSim simulation setup with mass moving across a single axis. The control action is applied force which moves the mass block and the camera captures the current position of the block under the region-of-interest.

B. Interfaces and Data Exchange

The schematic of the evaluation framework is shown in Fig. 2. At each discrete time step k, the controller reads the current output measurement $y[k]$. The output measurement is the absolute position of the center point of the die in the RoI. The controller also reads the reference signal $r[k]$ received from the reference generator. Based on these two inputs the controller computes the control action $u[k]$ and applies it to the plant which is simulated in CoppeliaSim. The plant reads the current control action $u[k]$ and the camera sensor modeled in CoppeliaSim produces output $img[k]$. The camera sensor captures the image based on the current position of the die on the product handler. This process is then repeated periodically. MATLAB Simulink is used to simulate and evaluate the closed-loop control performance. Simulink is responsible for providing reference data $r[k]$ at each discrete time step k and the data logger stores the output measurements $y[k]$ in the host PC.

C. Composable Multi-Core Platform

The embedded platform targeted in this paper is CompSOC [6]. It has a configuration number of a predictable RISC-V cores and memories. The platform uses a predictable and composable micro-kernel (CoMiK) to create a virtual execution platform (VEP). A VEP is composable, i.e., a CompSOC can run multiple VEPs concurrently and independent of other VEPs. A periodic time-division-multiplexing (TDM) policy is used on all processors. This enables the platform to achieve real-time performance with cycle-accurate time granularity.

D. Automatic Code Generation

We use standard Simulink code generation tool optimized for CompSOC. The designer may decide which part of the Simulink model is to be executed on the platform. As shown in Fig. 2, controller and image processing blocks are deployed on the targeted platform as a PIL block.

E. *PIL Simulation*

After the automatic code generation, PIL simulation compiles the code and uploads it on the embedded platform. Before uploading the code on the CompSOC platform, we define the processor tile, the number of virtual platforms, TDM allocation, and the size of the virtual platform allocated for the application. In this way, the designer can upload the executable on the corresponding VEPs of the processor tile. Authors in [13] proposed the framework which explains the different configurations for PIL targetting multi-processor platforms. In this paper, we demonstrate PIL simulation for the vision-based control on CompSOC.

V. XIL RESULTS

In this section, we discuss evaluation of the closed-loop control performance in the proposed framework. Fig. 4 shows the PIL simulation output for the mass-damper system. The output position (m) is extracted from camera modeled in CoppeliaSim. We validate the designed controller by performing a model-in-the-loop (MIL) simulation. Once the controller is verified, we perform SIL and PIL simulations. We can observe that PIL and SIL simulation results are relatively similar. It further shows the developed software functionalities for the VIL system are correct. Also, when the framework is implemented on the CompSOC, the PIL results show that output trajectory approaches the reference and the steady-state error becomes zero and we achieve the settling time of 1.5 sec.

In the proposed framework, we ensure that the PIL simulation runs in synchronized mode. The remote API in CoppeliaSim offers functionality to synchronize control between the remote API client and server. PIL simulation allows evaluation of the execution time and memory utilization on the target platform. We observed the execution time of the subsystem consisting of the image processing algorithm and PID controller to be 24 ms and memory usage of 32kB on CompSOC platform.

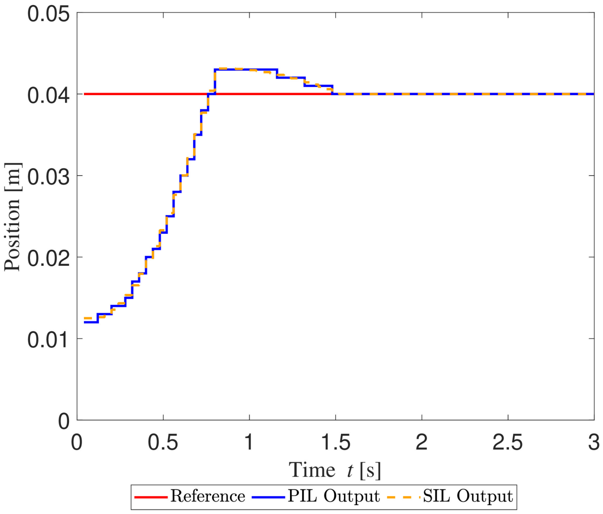

Fig. 4. PIL and SIL simulation results using the vision-based sensor modeled in CoppeliaSim. Output is the position.

VI. CONCLUSIONS

In this paper, we present an evaluation framework for the vision-based closed-loop control system considering motion control applications. Further, we demonstrate the MIL, SIL, and PIL simulation which targets a composable multi-core platform. The proposed framework allows the designer to test and debug the development of the VIL system for the industrial control application and perform PIL simulation with the strictly periodic deterministic executions by implementing it on the CompSOC platform.

Future direction is to extend the framework to perform HIL simulations on multiple cores of the platform and employ different scheduling techniques such as parallel and pipelining for improving the execution time and the closed-loop performance of the system.

ACKNOWLEDGMENT

This work is supported by the ECSEL Joint Undertaking under grant agreement 101007311 (IMOCO4.E).

REFERENCES

[1] M. H. Khan, M. S. Farid, and M. Grzegorzek, “Vision-based approaches towards person identification using gait,” *Computer Science Review*, vol. 42, p. 100432, 2021.

[2] M. Iwasaki, K. Seki, and Y. Maeda, “High-precision motion control techniques: A promising approach to improving motion performance,” *IEEE Industrial Electronics Magazine*, vol. 6, no. 1, pp. 32–40, 2012.

[3] M. Čech, A.-J. Beltman, and K. Ozols, “Pushing mechatronic applications to the limits via smart motion control,” *Applied Sciences*, vol. 11, no. 18, p. 8337, 2021.

[4] M. Čech, A.-J. Beltman, and K. Ozols, “Digital Twins and AI in Smart Motion Control Applications.” in *Emerging Technologies and Factory Automation (ETFA), paper in press*. IEEE, 2022.

[5] S. Mohamed, D. Goswami, S. De, and T. Basten, “Optimising multiprocessor image-based control through pipelining and parallelism,” *IEEE Access*, vol. 9, pp. 112 332–112 358, 2021.

[6] K. Goossens, M. Koedam, A. Nelson, S. Sinha, S. Goossens, Y. Li, G. Breaban, R. van Kampenhout, R. Tavakoli, J. Valencia, H. A. Balef, B. Akesson, S. Stuijk, M. Geilen, D. Goswami, and M. Nabi, *NoC-Based Multiprocessor Architecture for Mixed-Time-Criticality Applications*. Springer, 2017, pp. 491–530.

[7] M. Haghi, F. Wenguang, D. Goswami, and K. Goossens, “Delay-based design of feedforward tracking control for predictable embedded platforms,” in *American Control Conference (ACC)*. IEEE, 2019, pp. 3726–3733.

[8] S. Mohamed, S. De, K. Bimpisidis, V. Nathan, D. Goswami, H. Corporaal, and T. Basten, “Imacs: a framework for performance evaluation of image approximation in a closed-loop system,” in *Mediterranean Conference on Embedded Computing (MECO)*, 2019, pp. 1–4.

[9] S. De, S. Mohamed, D. Goswami, and H. Corporaal, “Approximation-aware design of an image-based control system,” *IEEE Access*, vol. 8, pp. 174 568–174 586, 2020.

[10] S. Mohamed, D. Goswami, V. Nathan, R. Rajappa, and T. Basten, “A scenario-and platform-aware design flow for image-based control systems,” *Microprocessors and Microsystems*, vol. 75, p. 103037, 2020.

[11] G. van der Veen, J. Stokkermans, N. Mooren, and T. Oomen, “How learning control supports industry 4.0 in semiconductor manufacturing,” in *ASPE Spring Topical Meeting on Design and Control of Precision Mechatronic Systems*, 2020, pp. 1–5.

[12] Y. Liu and S. Zhou, “Detecting point pattern of multiple line segments using hough transformation,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 1, pp. 13–24, 2014.

[13] M. Haghi, M. Koedam, D. Goswami, and K. Goossens, “Model-based processor-in-the-loop framework for composable multi-core platforms,” in *Euromicro Conference on Digital System Design (DSD)*. IEEE, 2019, pp. 592–596.