Bridging the Controller Design-Implementation Gap for Image-based Control Systems

Sajid Mohamed\textsuperscript{1}, Dip Goswami\textsuperscript{1}, Twan Basten\textsuperscript{1,2}

\textsuperscript{1}Electronic Systems Group, Eindhoven University of Technology, The Netherlands
\textsuperscript{2}ESI, TNO, Eindhoven, The Netherlands
\{s.mohamed, d.goswami, a.a.basten\}@tue.nl

\section{I. Motivation}

Image-Based Control (IBC) systems have a long sample period. Sensing in these systems consists of compute-intensive image processing algorithms whose response times are dependent on image workload. The state-of-the-art design for IBC systems considers worst-case workload and is generally platform independent \cite{1}. This results in a long sample period for the controller and hence suboptimal quality-of-control (QoC). Such a generic controller design fails to efficiently utilise the advantages of the state-of-the-art multiprocessor platforms (e.g. MPSoC) that allow parallel and pipelined executions.

An embedded systems engineer, who implements the controller, considers this worst-case based design for mapping of controller tasks, allocating platform resources and satisfying the timing requirements imposed by the control engineer. We observe that a controller design-implementation gap exists that results in significant resource over-provisioning and inefficient platform utilisation. A platform-aware co-design approach would bridge this gap to a great extent \cite{1}.

Another crucial observation we noticed is that the execution time for the sensing task is dependent on image workload variations. In a real world situation, the worst-case image workload is rarely encountered and the response time distribution due to workload variations can be statistically analysed (e.g. as a PERT distribution) \cite{2}. Such a distribution helps the designer to classify frequently occurring workload scenarios and almost all the times we see that these frequent scenarios do not include the worst case. Intuitively, control performance can be improved if we take this into account. We can also avoid inefficient resource utilisation due to idling compared to the worst-case design.

\section{II. Approach}

We present a structured (co-)design flow for IBC systems to optimise QoC by sampling as fast as possible for a given platform allocation. Our method enables us to relate throughput, a parameter relevant for embedded systems engineers, to sample period, relevant for control engineer. This helps to design, analyse and optimise the IBC systems jointly from both control and embedded perspective.

The key idea is to characterize and model workload variations to identify a set of workload scenarios. For a given platform allocation, for each scenario, we find the optimal mapping configuration that maximises the effective resource utilization. Each workload scenario leads to a sampling period of the IBC system. A controller is designed for each sampling period optimising the QoC and ensuring switching stability of the overall system. The combination of a sampling period and the corresponding controller defines a controller configuration. The combination of controller and mapping configuration is a system configuration. At runtime, the IBC system switches (using a reconfiguration mechanism) between the system configurations depending on the actual image workload. As opposed to operating under the worst-case design with sampling period $h_{wc}$, the IBC system most of the time runs in system scenarios \cite{3} with a shorter sampling period. A system scenario abstracts multiple workload scenarios having the same sampling period either due to platform constraints or due to our design choice. This reduces the average sampling period for the IBC system and improves its QoC.

Our Scenario and Platform-Aware Design (SPADe) flow is illustrated in Fig. 1. SPADe involves the following design aspects:
1) Formal Modelling: i) identify and model the parameters that characterise workload variations, and ii) model application considering workload variations and platform considering platform constraints.
2) Analysis and Design: Analyse application and platform models to design system configurations.
3) Reconfiguration mechanism for run-time implementation.

III. RELATED WORK

Our SPADe approach models and explicitly considers the image processing in the sensing algorithm. One main challenge of IBC is the delay of the sensor data to reach the controller[4][5]. Generally, the analysis of the image processing step is limited to a measurement or estimate of the computational delay and is assumed to be negligible compared to other delays in the control loop [4][6][7].

Control engineers tackle a long sample period of IBC systems using state estimation [8], robust design [9], predictive control [10], observer-based [1], multi-rate sampling [11] and reconfigurable pipelining methods [12]. However, these approaches do not model and consider platform constraints like resource availability and mapping, and/or workload variations in image processing [1]. SPADe can explicitly handle these constraints using the proposed co-design approach.

In [3], a system-scenario-based design approach is introduced from an embedded systems engineering perspective. However, control performance and long sample period are not explicitly considered. The SPADe approach bridges the design gap between a control engineer and an embedded systems engineer by relating image processing workloads and platform mappings to sample periods and quality of control.

IV. RESULTS AND CONCLUSION

We observe that our switching designs of SPADe (plot \((h_1h_2h_{wc})^*\) and \((h_2h_2h_{wc})^*\)) settle faster (and hence have better QoC) than the worst-case sampling period based design (see plot \((h_{wc})^*\)) in Fig. 2. Our SPADe approach and the results are explained in detail in [13].

In conclusion, we presented a structured platform-aware (co-)design flow, SPADe, that considers application dependencies, platform settings and control parameters for an efficient design and implementation of an IBC system. Our SPADe approach optimises QoC and maximises the effective resource utilisation for a given platform allocation.

The main drawback of the current approach is that a better controller can be designed only if we could identify the existence of a common quadratic lyapunov function. Further, our approach performs proficiently when the image processing algorithm is known, i.e. it is a white/gray box approach. In this case, SPADe assumes that information on parameters that relate to workload is known or can be computed during the frame pre-processing.

V. NEXT CHALLENGES

Future work is to study how we can extend our approach for IBC applications to pipelined and reconfigurable platform implementation. multiple IBC applications sharing the platform study the impact of approximate image processing algorithms on QoC distributed automotive platform implementation considering communication busses, e.g. CAN, FlexRay study the effect of alternate control design methods like markov jump linear systems.

ACKNOWLEDGMENT

This project has received funding from the European Union’s Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 674875.

REFERENCES