FPGA-based Embedded System Designed for the Deployment in the Compliant Robotic Leg CARL

Steffen Schütz[a], Atabak Nejadfard[b], Max Reichardt and Karsten Berns[c]

Robotics Research Lab, Department of Computer Science, TU Kaiserslautern, Kaiserslautern, Germany

Keywords: Embedded Systems, Distributed Control, Compliant Actuation, Series Elastic Actuator, Bipedal Walking.

Abstract: The embedded system that is distributed within a bipedal robot is a key component of such a highly interwoven mechatronic system. Generally, it has to handle two competing main tasks – executing the embedded closed-loop control of the actuators and handling the communication with the higher-level control system. As the restrictions on physical size and energy consumption limit its computational resources, the design of the embedded nodes poses a potential bottleneck for the performance of the overall system. Hence, the following presents an approach to mitigate the conflicting requirements by deploying FPGA-based embedded nodes. It is illustrated how the additional flexibility at the logic level is used to implement the closed-loop force and impedance control of a series elastic actuator. Furthermore, it is shown how the consequent hardware/software co-design enables the deployment of a full featured robotic framework. To validate the concept, the properties of the implementation are characterized.

1 INTRODUCTION

In an advanced bipedal robot, the distributed embedded controllers should abstract the mechatronic system for the higher-level control system – e.g., to act like an ideal torque or position source (Radford et al., 2015). This abstraction entails two main tasks at the embedded level – the application specific tasks such as executing the closed-loop control of the actuators and the handling of the communication with the higher-level control. Depending on how much of the closed-loop structure is distributed to the embedded nodes determines how the criticality is distributed between the communication and the application task. If most of the closed-loop control cascades are executed at the embedded nodes, the communication side becomes less performance critical. In this scenario, a high sampling rate and the deterministic execution of the embedded control are the main determinant for the achievable performance (Whitney, 1977; Shirai et al., 2016). On the other hand, if a portion of the closed-loop actuator structure is executed at the high-level controller, the frequency and latency of the communication might limit the actuator performance (Zhao et al., 2015). Thus, a deterministic execution with a high frequency of the two main tasks is advantageous for the performance of the overall biped.

Looking at the literature on the embedded solutions deployed in current walking robots, it can be noted that the degree of the distribution varies among the systems. Table 1 summarizes the implementation details as well as the main performance indicators in form of various execution frequencies of several legged machines. In all systems, with the exception of the BioBiped, a dedicated controller handles the current control of the electromagnetic motors. Thus, the current control and – in case of BLDC motors the commutation – is not mentioned explicitly in Table 1.

The embedded systems developed for StarlETH (Hutter, 2013) and BioBiped (Scholz, 2016) are representatives of a centralized approach. As mentioned above, in case of the BioBiped, even the current control is handled by the central high-level controller. In StarlETH, the current and velocity control of the motors, are handled by the embedded controller. Notably, the purpose of the latter is the local and thereby high-frequent compensation of the undesired motor and gearbox dynamics. Nevertheless, in both cases the actuator behavior desired by the higher control layers are established by the high-level controller. All other implementations that are listed, namely DLR's TORO, NASA's Valkyrie, IIT's

[a] https://orcid.org/0000-0001-7568-6439
[b] https://orcid.org/0000-0001-9074-7519
[c] https://orcid.org/0000-0002-9080-1404

Schütz, S., Nejadfard, A., Reichardt, M. and Berns, K.
FPGA-based Embedded System Designed for the Deployment in the Compliant Robotic Leg CARL.
DOI: 10.5220/0007978205370543
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 537-543
ISBN: 978-989-758-380-3

Table 1: Overview of the control infrastructure deployed in existing walking machines presented in the literature.

Platform	C/D	EC	f_{EC}	Bus	f_{Bus}	HLC	f_{HLC}
BioBiped (Scholz, 2016)	C	-	-	EtherCAT	1 kHz	Orocos/ROS	1 kHz
StarlETH (Hutter, 2013)	C	V	1 kHz	CAN	400 Hz	SL	400 Hz
TORO (Ott et al., 2010)	D	P/I/T	3 kHz	Sercos-II	1 kHz	Matlab	1 kHz
Valkyrie (Radford et al., 2015)	D	I/T	5 kHz	Robonet	-	ROS	< 3 kHz
WALK-MAN (Tsagarakis et al., 2017)	D	P/T	1 kHz	EtherCAT	500 Hz	YARP	500 Hz
ESCHER (Knabe et al., 2015)	D	I/T	2 kHz	CAN	500 Hz	Bitfrost/ROS	500 Hz

C – Centralized, D – Decentralized, EC – Embedded Control, HLC – High-Level Control

P – Position Control, V – Velocity Control, I – Impedance Control, T – Torque Control

WALK-MAN, and Virgnia Tech's THOR/Escher – implement at least the torque/force plus one further control cascade at the embedded node.

Generally, it can be observed that the sampling rate of closed-loop cascades that establish the final abstraction towards the higher control layers is limited in centralized control approaches – mostly due to the bandwidth of the communication bus (Hutter, 2013). Furthermore, as mentioned above, the latency introduced by the communication bus reduces the stability margin of the control system. Nevertheless, even in highly distributed control approaches, a high bandwidth communication with the superimposed control layers is desirable.

In addition to the pure performance related characteristics of an embedded system, another aspect is the ease of handling of such a complex mechatronic system as a bipedal robot. Usually, there is a break in the development process between the powerful PCs that execute the high-level control and the distributed embedded nodes that implement the actuator control. As Table 1 shows the high-level control is usually implemented using a framework that runs on top of a full-featured OS. The embedded nodes in contrast are mostly used bare-metal or at most with a slim real-time OS. Nevertheless, a more unified approach to software development and especially system debugging is desirable and facilitates the system handling tremendously.

Therefore, in prior work, we presented an approach for the decoupling of the two competing tasks by using an FPGA-based embedded node that enabled the deployment of the full-featured robotic framework Finroc[1] to the bare-metal embedded nodes (Schütz et al., 2014; Reichardt et al., 2017). Following a strict HW/SW co-design approach, the flexibility provided by FPGAs ensured this did not negatively impact the performance characteristics of the embedded nodes. The deployment of either task-specific processing units or general-purpose soft-processors allows for parallel computation to mitigate computational bottlenecks.

The concept for the FPGA system is illustrated in Figure 1. At the center of each subsystem – the communication side and the application side – is a soft processor. The communication between the two subsystems is handled via a block of Dual-Port Ram. More details about the Ethernet-based framework-integrated embedded protocol (FinEmbP) and the data handling within the FPGA system are provided in (Schütz et al., 2014).

This system served as the basis for the development of the embedded nodes that are deployed within the compliant robotic leg Compliant Robotic Leg (CARL) developed at the Robotics Research Lab (RRLab) (Schütz et al., 2017). It poses a first iteration, to bring the biologically-inspired behavior-based bipedal locomotion control (B4LC)– also developed at the RRLab – from a simulation to a physical robotic system. Thus, CARL is a bio-inspired system that – inspired by the morphology of humans – integrates mono- as well as biarticular actuations. The redundant drive system – five series elastic actuators (SEAs) that act on three joints – is implemented using two scaled RRLab SEAs developed at the RRLab (Schütz et al., 2016).

Hence, the following is meant to give an insight into the conceptual design and implementation of the embedded nodes at the electronic as well as the FPGA level. The purpose is to encapsulate the RRLab SEAs towards B4LC as force and impedance sources while exploiting the full actuator potential. Especially the extensions to the HW/SW system at the application side of the FPGA and the distribution of the closed-loop tasks within that system are detailed.

2 ELECTRONIC SYSTEM

As a first design iteration, at the electronics level, a modular approach has been adopted. Therefore, as illustrated in Figure 2a, the functionalities are dis-

[1] https://www.finroc.org/

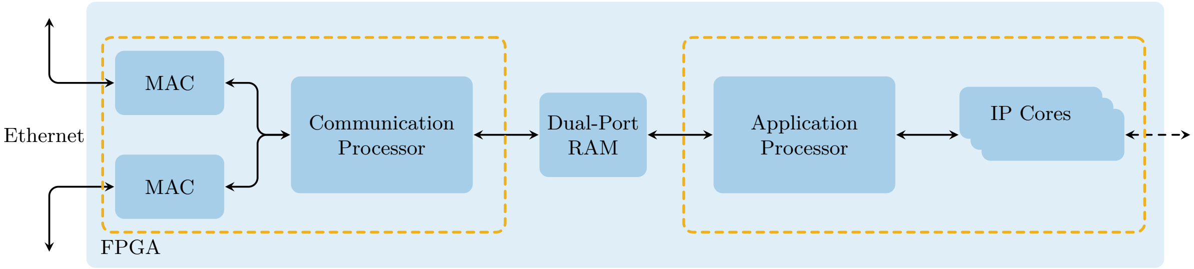


Figure 1: Task decomposition within the FPGA system. Adapted from (Schütz et al., 2014).

tributed across several modules in form of separate PCBs. One module contains the FPGA and the necessary periphery. At the core of that module is the *eSM* Board from elrest Automationssysteme GmbH[2] (for more information refer to (Schütz et al., 2014)). The second module contains the output stage that is used for the commutation and the current control of the BLDC motor. Currently, this is handled by the Gold Twitter servo drive from Elmo Motion Control [3]. Furthermore, that module contains an ADC to capture the winding temperature via the PT1000 temperature sensor. A third module implements the I/O interface to the embedded sensory system. For electric stability and robustness, all signals are differential driven using RS-485 transceiver. The connection between the modules is established using a backplane. Figures 2b and 2c shows the modular backplane system.

In a second iteration, once the exact requirements are known and the functionality of the electronics is ensured, the modular system can be integrated into a compact dedicated embedded node. Especially the commutation and the current control of the BLDC motor can also be handled by the FPGA making the Gold Twitter obsolete.

3 APPLICATION-SIDE HW/SW IMPLEMENTATION

Figure 3 shows a detailed illustration of the FPGA system that implements the application side. At the heart of it is the application soft processor – in this implementation a *Intel Nios II*[4]. It is extended by a multitude of IP Cores that are accessible by the Nios processor as memory-mapped slaves. They handle the serial interfaces to the embedded iC-MU-based sensors. One IP Core directly interfaces the sensor that captures the spring deflection which serves as the feedback for the force control of the SEA. Through the dedicated IP Core, the sensory information can be obtained at the maximum sampling rate and subsequently filtered without causing any overhead at the Nios. The filtered spring position can be obtained by a single memory access.

The sensor that captures the position of the rotor is interfaced by the Elmo Gold Twitter as it requires the information for the commutation of the electromagnetic field. Hence an IP Core within the FPGA taps the SSI in which the Gold Twitter acts as the master. Based on the obtained information another IP Core calculates the absolute length of the ball screw. With the rotational iC-MU being a single turn encoder, this core detects the sensor overflows and hence – once initialized – can calculate the absolute length of the ball screw. After being filtered in HW, the ball screw length can be summed with the spring deflection to obtain the actuator length that forms the feedback for the impedance control.

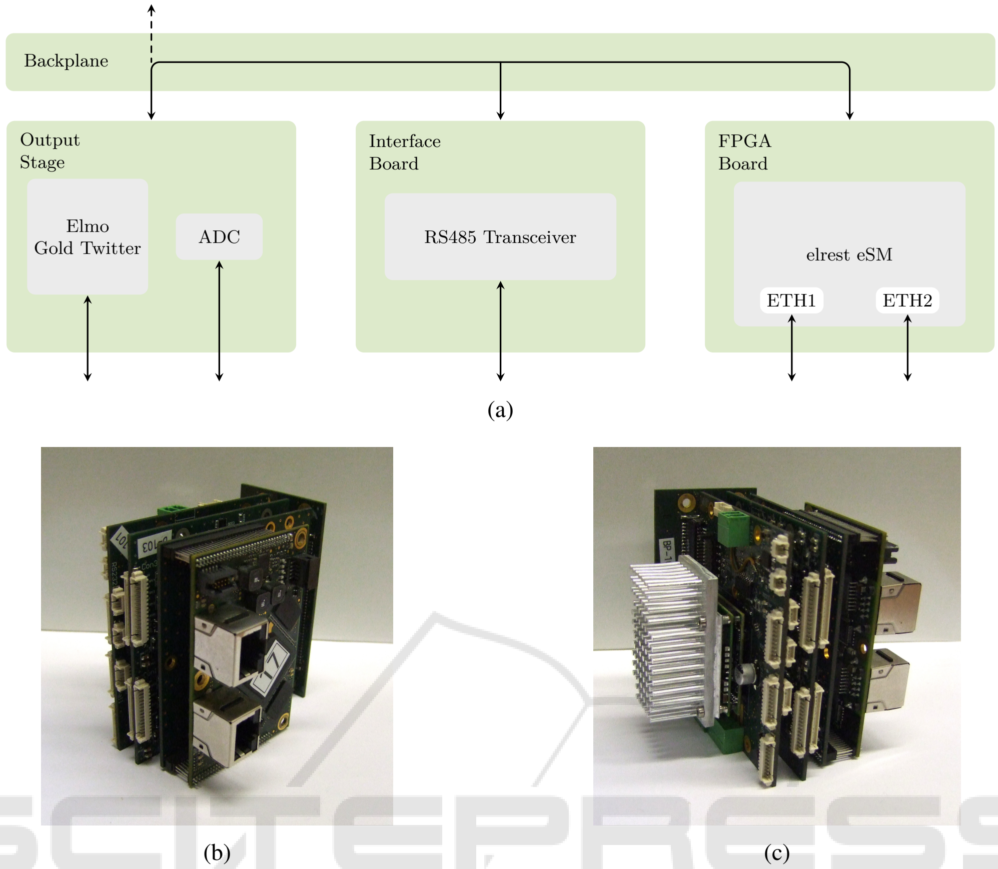
The interface to the Elmo Twitter is implemented by two PWM signals and a single I/O. Due to the flexibility of the FPGA, the properties of the PWM signals could be exactly determined to fully exploit the interface provided by the servo drive. Hence, the PWM signal that encodes the desired current is modulated with a base frequency of 160 MHz as this resembles exactly the frequency at which the servo drive logs the signal. The second PWM signal is the current feedback. For safety reasons and in case of system failure, a watchdog that monitors the Application Nios can disable the servo drive via the I/O that triggers its safe torque off (STO).

Another dedicated IP Core interfaces the serial interface of the ADC that monitors the winding temperature of the BLDC. This information is used in a model of the actuators thermal dynamics that is implemented in SW on the Application Nios. Furthermore, three of the five embedded nodes additionally interface the sensors that are located at the joints. Similar to the spring deflection, the sensor is sampled at maximum frequency and subsequently filtered by dedicated IP Core.

[2] http://http://www.elrest-gmbh.com/

[3] http://www.elmomc.com/

[4] https://www.intel.com/content/www/us/en/products/programmable/processor/nios-ii.html



(a)

(b)

(c)

Figure 2: Embedded electronics developed for CARL– (a) the underlying modular concept and the implementation of (b) the FPGA module and (c) the Elmo based output stage. All the three boards are mounted on the backplane board.

4 IMPLEMENTATION RESULTS

Figure 4 depicts the distribution of the closed-loop cascades, the relevant data flow, and the respective sampling frequencies within the HW/SW system. The current control and the commutation of the BLDC motor are handled by Elmo Gold Twitter at a frequency of 20 kHz. As mentioned above, this frequency also dictates the sampling frequency of the IP Cores that tap the rotor sensor, calculates the ball screw length, and filter the respective data. The spring deflection is sampled and filtered at 40 kHz.

Similar to Hutter (Hutter, 2013), the force control is established by a PID control. As already indicated in Figure 3, the PD-portion of the force control is implemented as a dedicated IP Core. It is executed at a frequency of 10 kHz and, as it is implemented in logic, without any jitter. The integral part of the force control is implemented in software. The combination of a low I-gain and an anti-windup mechanism is used to eliminate the steady-state force error. The force control is complemented by a feed-forward term that is based on the motor torque constant.

The outer impedance cascade is also implemented in software. It relies both on the spring deflection and the ball screw length as feedback signals to determine the current actuator length. All application-side software is implemented within the deployed Finroc instance and are executed with a frequency of 5 kHz. Notably, the application-side software implements more functionality beyond the closed-loop control. It includes the thermal model of the BLDC motor, several safety mechanisms, and the conversion of the raw sensor values to the SI values that are passed to the higher control layers.

Comparing the achieved sampling rates to the numbers summarized in Table 1, it becomes obvious that the presented system outperforms the comparable systems by at least a factor of two.

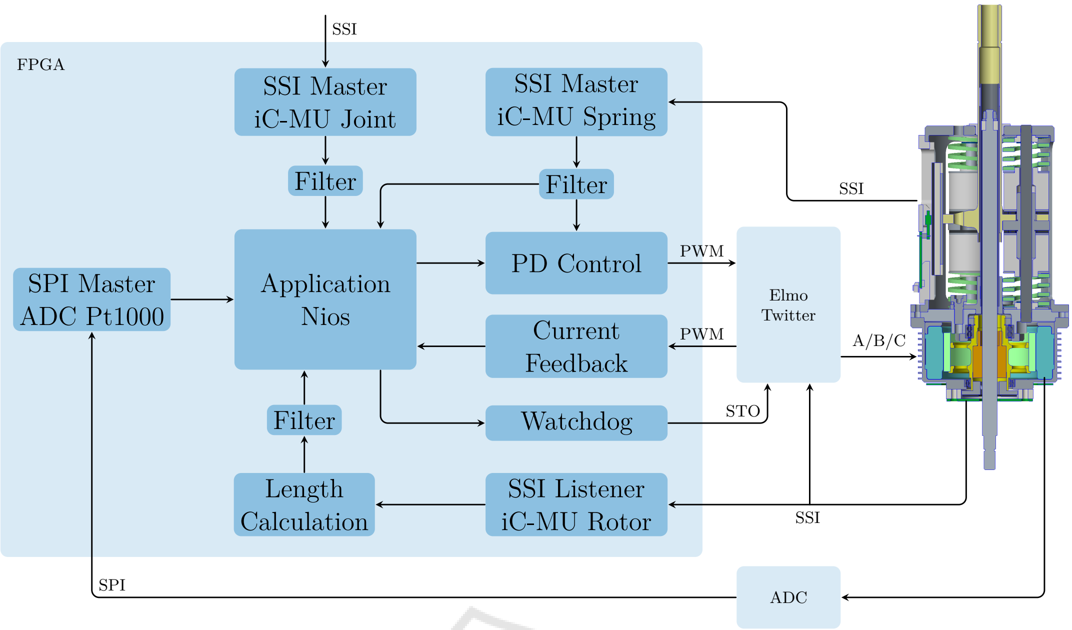


Figure 3: IP Cores acting as coprocessors on the application side of the FPGA system.

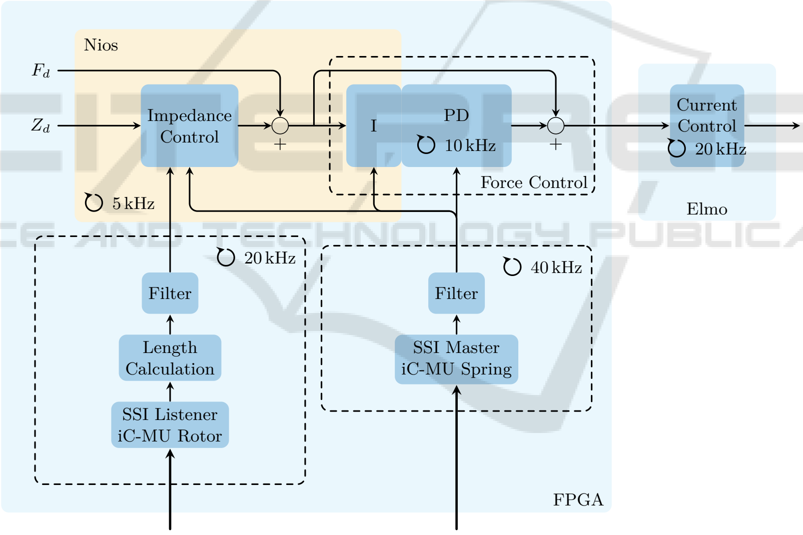


Figure 4: Distribution of the closed-loop cascades within the HW/SW system with the respective sampling frequencies.

4.1 Bare-metal Finroc Execution Timing

Besides the raw sampling rates, the performance of the application-side software has been analyzed regarding the execution timing and jitter (Reichardt et al., 2017). A dedicated IP Core has been used for the measurement of the exact timing in clock cycles of the application soft processor. Within the Finroc execution cycle, the maximum jitter was around 569.4 cycles. This equals a variation of the execution timing by 2.29 % relative the overall cycle length. Thus, the value is well below the desirable 10 % that are determined by Shirai et al. as performance critical (Shirai et al., 2016).

Overall, the numbers validate that no performance limitations regarding the closed-loop control of the actuator originate from the deployment of the robotic framework to the embedded nodes. On the other side it entails the benefit that the nodes are seamlessly integrate into the tooling of the high-level control system without a need to develop any additional tools. Especially, it facilitates the task tremendously when putting a complex system into operation.

4.2 FinEmbP Timing

Although the presented system poses a highly distributed implementation of the actuator control that reduces the requirements on the communication bus, the timing of FinEmbP implementation is investigated in a second experiment. The characteristics are evaluated using the FinEmbP network that is used for the first walking experiments with CARL. It consists of seven FinEmbP nodes – four standard SEA nodes as described above; one SEA node that additionally interfaces the sensory system in the foot to capture the interaction with the ground; a node that controls the test-rig (TR) that consists of a treadmill and a winch mechanism to mimic the second leg; one AD node that monitors the supply voltage and current of the overall system.

Table 2 gives the data that is transmitted and received by the nodes. The table shows that the transmitted data is relatively low – one of the benefits of the actuator abstraction towards the higher control layers. Besides the explicit input/output data, node internal data is divided into blobs with a maximum size of 48 B that are attached to the cyclic output data frames (referred to as *iSend* in the FinEmbP context, refer to (Schütz et al., 2014)). Hence, this mechanism poses a trade-off between the update rate of the internal data and the input/output data.

Table 2: Data sent and received by the FinEmbP nodes in Byte [B].

	SEA		TR	AD	Total
	Std	Foot			
Output	108	140	96	136	804
Input	56	56	4	60	344
Internal	332	368	108	328	-
Blobs	7	8	3	7	-

The seven nodes are arranged in a mixed star/line topology as shown in Figure 5. It includes two switches – one that is located within CARL and the second between CARL and the PC that executes the FinEmbP master.

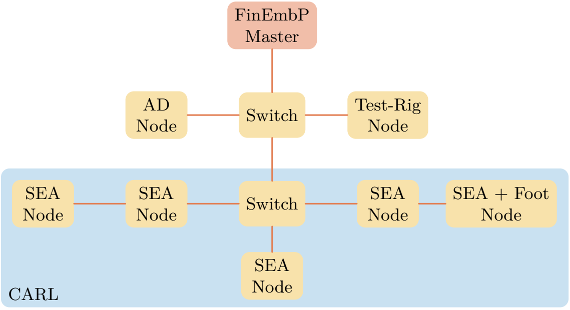


Figure 5: Topology of the FinEmbP network used during the experiment.

For the experiments, the FinEmbP master is running on a standard PC (Intel Core i7-4790K at 4 GHz and 16 GB RAM) with a headless Ubuntu 14.04. The high-level walking control developed for CARL– a subset of B4LC– is executed on top of the FinEmbP master. Hence, the overall setup resembles the normal operating conditions. Notably, the FinEmbP protocol is running at maximum frequency. The idle phase that is part of the protocol is set to zero.

To obtain a representative impression of the properties, 50000 FinEmbP cycles have been recorded. The average cycle time is 736 µs (approximately 1.36 kHz) with a standard deviation of 53 µs (7 % of the average cycle time). The jitter is mostly due to the implementation at the master side. As aforementioned, it is executed on a standard OS using a POSIX UDP socket. Figure 6 shows the histogram of the 50000 cycles.

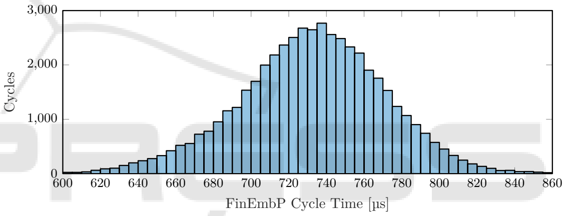


Figure 6: Histogram of the FinEmbP timing. The data is recorded with a network of seven nodes and the high-level walking control of CARL running on the master PC.

The average bus frequency compares well to the comparable systems listed in Table 1. As the actuator control is completely executed at the embedded node, the observed jitter is not critical.

5 CONCLUSIONS AND FUTURE WORK

The paper at hand presents application specific architecture of the dedicated embedded system designed for the use in CARL, a compliant bio-inspired robotic leg. Following a strict HW/SW-codesign approach, the deployment of a full-featured robotic framework to the bare-metal soft processor running at 100 MHz is achieved. This is formulated as an explicit design goal for the development of Robot Operating Software (ROS) 2.0 (Gerkey, 2017).

Besides the soft benefits that come with the extension of the robotic framework to the embedded nodes, the performance critical properties are preserved. This is achieved by implementing the sensor sampling and filtering as well as parts of the force closed-loop control as dedicated IP Core. Especially, the latter removes any significant jitter from the control execution.

It has been validated experimentally that the execution characteristics of the software components implemented within the robotic framework are adequate. Overall, the achieved sampling frequencies and determinism match or even exceed the properties of comparable systems. Similarly, the achieved frequency of the FinEmbP communication bus with 1.36 kHz compares well to other walking machines.

A potential next step would be the implementation of the commutation and the current control of the BLDC within the FPGA fabric as a dedicated co-processor. Thereby, the Elmo Gold Twitter servo drive can be substituted by a bare output-stage that features three half bridges and the appropriate facilities for the current measurement. Subsequently, the modular electronics can be composed to a highly-integrated physically small single PCB. It could be beneficial to integrate the electronics into the actuator.

If the jitter of the communication bus proofs to be critical, firstly a real-time operating system (OS) at the master side poses a potential improvement. Furthermore, the OS's stack could be partially bypassed using, e.g., by using *libpcap*.

ACKNOWLEDGEMENTS

This work was partly funded by the European Commission 7th Framework Program under the project H2R (no.60069).

REFERENCES

Gerkey, B. (2017). Why ROS 2.0? http://design.ros2.org/articles/why_ros2.html. Accessed: 2019-04-16.

Hutter, M. (2013). *StarlETH & co-design and control of legged robots with compliant actuation*. PhD thesis, Diss., Eidgen&quot;ossische Technische Hochschule ETH Z&quot;urich, Nr. 21073, 2013.

Knabe, C., Griffin, R., Burton, J., Cantor-Cooke, G., Dantanarayana, L., Day, G., Ebeling-Koning, O., Hahn, E., Hopkins, M., Neal, J., and others (2015). Designing for Compliance: ESCHER, Team VALOR's Compliant Biped. Technical report, Terrestrial Robotics Engineering & Controls (TREC) Lab, Virginia Tech.

Ott, C., Baumgärtner, C., Mayr, J., Fuchs, M., Burge, R., Lee, D., Eiberger, O., Albu-Schäffer, A., Grebenstein, M., and Hirzinger, G. (2010). Development of a biped robot with torque controlled joints. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 167–173. IEEE.

Radford, N. A., Strawser, P., Hambuchen, K., Mehling, J. S., Verdeyen, W. K., Donnan, A. S., Holley, J., Sanchez, J., Nguyen, V., Bridgwater, L., et al. (2015). Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, 32(3):397–419.

Reichardt, M., Schütz, S., and Berns, K. (2017). One fits more - on highly modular quality-driven design of robotic frameworks and middleware. *Journal of Software Engineering for Robotics (JOSER): Special Issue on Robotic Computing*, 8(1):141–153. ISSN: 2035-3928; this publication is available at http://joser.unibg.it/index.php?journal=joser&page=issue&op=view&path[]=9.

Scholz, D. (2016). *On the Design and Development of Musculoskeletal Bipedal Robots*. PhD thesis, Technische Universität Darmstadt.

Schütz, S., Mianowski, K., Kötting, C., Nejadfard, A., Reichardt, M., and Berns, K. (2016). RRLAB SEA – A highly integrated compliant actuator with minimised reflected inertia. In *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*.

Schütz, S., Nejadfard, A., Mianowski, K., Vonwirth, P., and Berns, K. (2017). CARL – A compliant robotic leg featuring mono- and biarticular actuation. In *IEEE-RAS International Conference on Humanoid Robots*, pages 289–296.

Schütz, S., Reichardt, M., Arndt, M., and Berns, K. (2014). Seamless extension of a robot control framework to bare metal embedded nodes. In *Informatik 2014*, Lecture Notes in Informatics (LNI), pages 1307–1318, Stuttgart, Germany.

Shirai, T., Osawa, K., Chishiro, H., Yamasaki, N., and Inaba, M. (2016). Design and Implementation of a High Power Robot Distributed Control System on Dependable Responsive Multithreaded Processor (D-RMTP). In *IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, pages 19–24.

Tsagarakis, N. G., Caldwell, D. G., Negrello, F., Choi, W., Baccelliere, L., Loc, V., Noorden, J., Muratore, L., Margan, A., Cardellino, A., Natale, L., Hoffman, E. M., Dallali, H., Kashiri, N., Malzahn, J., Lee, J., Kryczka, P., Kanoulas, D., Garabini, M., Catalano, M., Ferrati, M., Varricchio, V., Pallottino, L., Pavan, C., Bicchi, A., Settimi, A., Rocchi, A., and Ajoudani, A. (2017). Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259.

Whitney, D. (1977). Force feedback control of manipulator fine motions. *Trans. ASME J. of Dynam. Syst., Measur. and Control*, 99(2):91–97.

Zhao, Y., Paine, N., Kim, K. S., and Sentis, L. (2015). Stability and Performance Limits of Latency-Prone Distributed Feedback Controllers. *IEEE Transactions on Industrial Electronics*, 62(11):7151–7162.