Doc ID	18010801R03
Doc Creation Date	20 OCT 2017
Doc Revision	03
Doc Revision Date	5 JAN 2018
Doc Status	SUBMITTED

Workpackage	Deliverable ID
WP2, Business requirements and reference system architecture	**D2.3, Overall requirements on I-MECH reference platform**

Executive summary

This deliverable presents the system-level requirements for the I-MECH platform. The input for this document was gathered as part of I-MECH work package 2 and is presented in an unprocessed form in deliverables D2.1 and D2.2. This document focuses on requirements related to the I-MECH platform architecture (and building blocks), consisting of a combination of hardware, firmware and software requirements.

The system-level architecture requirements are organized by I-MECH platform layer (instrumentation, control and system behavior layer). For each layer, the architectural requirements follow from the interfaces between the layers and which functions must be present to make the I-MECH reference platform flexible, modular and interoperable by means of open standards.

This deliverable will be succeeded by and refined in D2.4, incorporating information from the requirements phases in all of the I-MECH work packages.

Author Tom Kampschreur / Gijs van der Veen (Nexperia) / Thomas Lembrechts (Sioux CCM)

Keywords
Motion control platform, Interoperability, Open standards compliant, Instrumentation, Sensors, Actuators, Drives, Vision, Control, Dynamics, Vibrations, Bandwidth, Observers, Learning, Repetitive, Robustness, Trajectory, Filters, Multi-Many-Core, OPC UA, RAMI 4.0

Coordinator	Sioux CCM
Tel.	0031 (0)40.263.5000
E-Mail	info@i-mech.eu
Internet	www.i-mech.eu

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

Smart Mechatronic Solutions

Table of contents

(Open) Issues & Actions

Open Issues (and related actions) that need central attention shall be part of a file called "IAL - Issues & Action List – Partners" which is can be found in the Google Drive Partner Zone.

ID	Description	Next steps	Due date	Owner	IAL ID
OI-0002	Req A12 (code generation) is regarded as too strict. We will keep the two options obligatory/wish open in this deliverable.	The discussion point is added to the WP2 IAL to be discussed at the next face to face consortium meeting.	28-02-2018	Gijs & Thomas	WK1751.01
OI-0004	No clear requirements are in place on how to deal with decentralized control. Centralized control is currently the prefered control strategy for the I-MECH platform. This might however be unsuited for very high speed servo loops.	This is discussion point is added to the WP2 IAL to be discussed at the next face to face consortium meeting.	28-02-2018	Gijs & Thomas	WK1751.02

Document Revision History

Revision	Status	Date	Author	Description of changes	IAL ID / Review ID
R01	Draft	20-OCT-17	Van der Veen	Initiate	
R02	Draft	01-DEC-17	Van der Veen	First draft submitted for review	
R03	Draft	20-DEC-17	Van der Veen	Complete revision based on reviewer feedback	

Contributors

Revision	Affiliation	Contributor	Description of work
R01	Nexperia	Van der Veen	Main content
R02	Nexperia	Van der Veen	Main content
		Kampschreur	Additional feedback
	Sioux CCM	Lembrechts	Various contributions
	Technolution	Marc van Eert	Feedback via e-mail
R03	Nexperia	Van der Veen	Complete revision based on reviewer feedback
	Technolution	Marc van Eert	Feedback and suggestions
	Sioux CCM	Lembrechts	Various contributions, editorial work
	Sioux CCM	Kuppens	Feedback and suggestions

Document control

Reviewer Name	Role	Status / Revision / Selection							
	Status	Draft							
	Revision	01							
Arend-Jan Beltman	Coordinator	X							
Technolution	Reviewer	X							
UNIBS	Reviewer	X							

File Locations (cross reference to I-MECH documents)

Via URL with a name that is equal to the document ID, you shall introduce a link to the location (either in Partner Zone or CIRCABC)

URL	Filename	Date
2017102002R01	I-MECH Requirements Table	21-OCT-2017
2017102001R01	I-MECH State-of-the-art & Requirements	21-OCT-2017

Abbreviations & Definitions

Abbreviation	Description
ADC	Analog Digital Converter
BB	Building Block
CPU	Central Processing Unit
DoA	Description of Action: Annex 1B of the I-MECH grant agreement
EMC	Electromagnetic Compatibility
EMF	Electromagnetic Field
EMI	Electromagnetic Interference
HAL	Hardware abstraction layer
HIL	Hardware in the Loop
HW	Hardware
I2C	Inter IC bus
I2T	I^2T protection, used for motor overheating protection
IEPE	Integrated Electronics Piezo Electric
IPk	Peak Current
MIL	Model in the Loop
OVP	Over Voltage Protection
PIL	Processor in the loop
PWM	Pulse Width Modulation
ROI	Region of interest
RTOS	Real Time Operating System
SIL	Software in the Loop
SPI	Serial Peripheral Interface
STO	Safe Torque Off
SW	Software
Tmax	Maximum temperature
TSN	Time Sensitive Networking

TTL	Transistor-Transistor Logic
UVP	Under Voltage Protection
UWB	Ultra Wideband

1 Introduction

Based on the investigation in "2017102001R01 I-MECH State-of-the-art & Requirements", this document describes the requirements on the I-MECH reference platform. The requirements will be derived from the point of view of the 11 individual I-MECH building blocks (sub-system level) and how they integrate into the system in a flexible way, as well as from the point of view of the overall system (system level).

This document does not include all I-MECH requirements, because the focus is on system-level requirements. Requirements at building block level (sub-system level) can be more extensive and these are left to the respective requirement phases in tasks 3.1, 4.1 and 5.1.

1.1 Relation to description of action and state of the art

As described in the I-MECH description of action, the ECSEL-JU programme goals relate to improving the leadership of European industry concerning industrial motion control, and smart systems integration, which allows heterogeneous systems (building blocks) to be integrated into a functional system. Hence, requirements that distinguish the I-MECH platform from available solutions relate in particular to methods to ensure openness, extensibility and interoperability of the platform and methods to develop and customize hardware and software for the platform. Interface definitions and abstraction layers will facilitate updates to software and hardware without affecting the entire system. Furthermore, building blocks and hardware components will be modular and interoperable, leading to the ability to combine modules into a working system without the need to explicitly configure modules or interfaces. Interface requirements and standardization of interfaces will therefore be crucial from an I-MECH platform perspective, as they shall enable interoperability, modularity and maintainability of the platform. These key aspects define how the I-MECH platform extends beyond the current state of the art.

1.2 Definitions of I-MECH terminology

The I-MECH reference platform is designed to meet the goals of the ECSEL-JU programme, related to interoperability and integration. In this context a number of buzzwords occur which will be defined and made explicit in this brief section, to better appreciate their meaning and impact.

The I-MECH platform will provide:

- A **modular design**: a design approach that subdivides a system into smaller parts called modules, that can be independently created and then used in different systems. A modular system can be characterized by functional partitioning into discrete scalable, reusable modules; rigorous use of well-defined modular interfaces; and making use of industry standards for interfaces;
- **Self-diagnostics**: the ability of a module to examine and report on its own actual functionality compared to the designed functionality at runtime;
- **Self-reflection**: the ability of a module to examine, introspect, and report on its own structure and behavior at runtime;
- **Interoperability**: a characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, at present or future, in either implementation or access, without any restrictions;
- **Maintainability**: the ease with which a product can be maintained in order to correct defects or their cause, repair or replace faulty or worn-out components without having to replace still working parts, prevent unexpected working condition, maximize a product's useful life, maximize efficiency, reliability, and safety, meet new requirements, make future maintenance easier, or cope with a changed environment.
- A **service-oriented architecture**: a style of software design where a module provides services to other modules by application components, through a standardized interface. This allows the functionality to be accessed like a self-contained black-box, disregarding the specific implementation inside the module.
- **I-MECH building blocks**: modules with a specific set of features and interfaces, making the modules maintainable and interoperable on the I-MECH platform.

1.3 Scope of this deliverable

This I-MECH deliverable (D2.3) contains a first iteration of the requirements on the I-MECH reference platform and its architecture. The deliverable provides “a rough sketch” of the architecture of the platform and expected interfaces between building blocks to give direction for I-MECH tasks that will gather more detailed requirements about the implementation of functions and features. It also presents requirements of I-MECH pilots, use cases and demonstrators gathered during task 2.1, 2.2 and 2.3, which mainly concern requirements for specific functionality of building blocks.

The more detailed specifications for the architecture of the I-MECH reference platform will be gathered and defined in I-MECH task 3.1, 4.1, 5.1, 6.1 and 7.1. These tasks focus on requirements for specific layers of the I-MECH platform, implementation requirements and methodology. Deliverable D2.4 will gather the refined requirements and present the detailed specification of the I-MECH reference platform. For instance, where D2.3 prescribes the use of open protocols in layer 1, D2.4 will specify which protocols will be chosen specifically.

2 Requirements

This chapter describes the requirement specifications of the I-MECH reference platform. These requirement specifications must be met in order to achieve the I-MECH goals and ambitions. This means that the requirements focus on making the platform open, flexible, maintainable and conforming to industry standards, i.e on reusability and interoperability. The reference platform will not need to address all conceivable use cases, but rather describe a series of mechanisms or collection of interfaces to allow integration of modules. Following the V-model, the requirements are uniquely identified by a code. Wherever possible, these requirements will be specific, measurable, achievable, realistic and time-bound (SMART).

A distinction is made between *system-level* and *subsystem*-level requirements. The system-level requirements relate to reference platform architecture and the guiding mechanisms and principles. The subsystem-level requirements relate mainly to building-block-specific performance. The focus of this document is only on the former: requirements that should be met to integrate building blocks into the I-MECH platform. Tasks 3.1, 4.1 and 5.1 will zoom in on subsystem-level requirements for the three layers. A revision of the present document will be made in deliverable D2.4.

2.1 Requirement coding scheme

Each requirement ID is prefixed with rq- (for requirement), the deliverable ID (in this case D2.3) and the applicable domain:

- rq-D2.3-A: architecture
- rq-D2.3-L1: layer 1
- rq-D2.3-L2: layer 2
- rq-D2.3-L3: layer 3
- rq-D2.3-BBx: building block x
- rq-D2.3-CB: layer 2 control BB (general)
- rq-D2.3-S: safety
- rq-D2.3-E: electrical safety

We suggest that other I-MECH deliverables use the same coding scheme, implementing at least the “rq-D#.#-” part to ensure that requirement IDs are unique and can be found easily in documents.

The requirement verification method is also indicated. Two methods are foreseen:

- T: test/validate
- I: inspect/demonstrate

A requirement can be:

- R: required (must-have)
- O: optional (nice-to-have)

2.2 System-level requirements

A major target for the I-MECH project is to deliver a flexible, open and future-proof architecture for (motion) control platforms. This target leads to requirements at building block level and at system level, which follow from a careful consideration of interfaces and interactions between building blocks.

2.2.1 Methodology for the I-MECH reference platform

Several I-MECH building blocks can be mixed and matched to form a (motion) control platform. Sensors and actuators (regardless of whether they are I-MECH building blocks or COTS devices) connect to the I/O of a multi/many-core platform. The multi/many core platform and its software take care of the communication with the control software layer (layer 2) and the behaviour level (level 3).

Figure 1 below shows the architecture of the I-MECH reference platform. It is intended to clarify architecture and interfacing requirements, focussing on the the requirements (i.e. the *what* question) and not yet on solutions (i.e. the *how*).

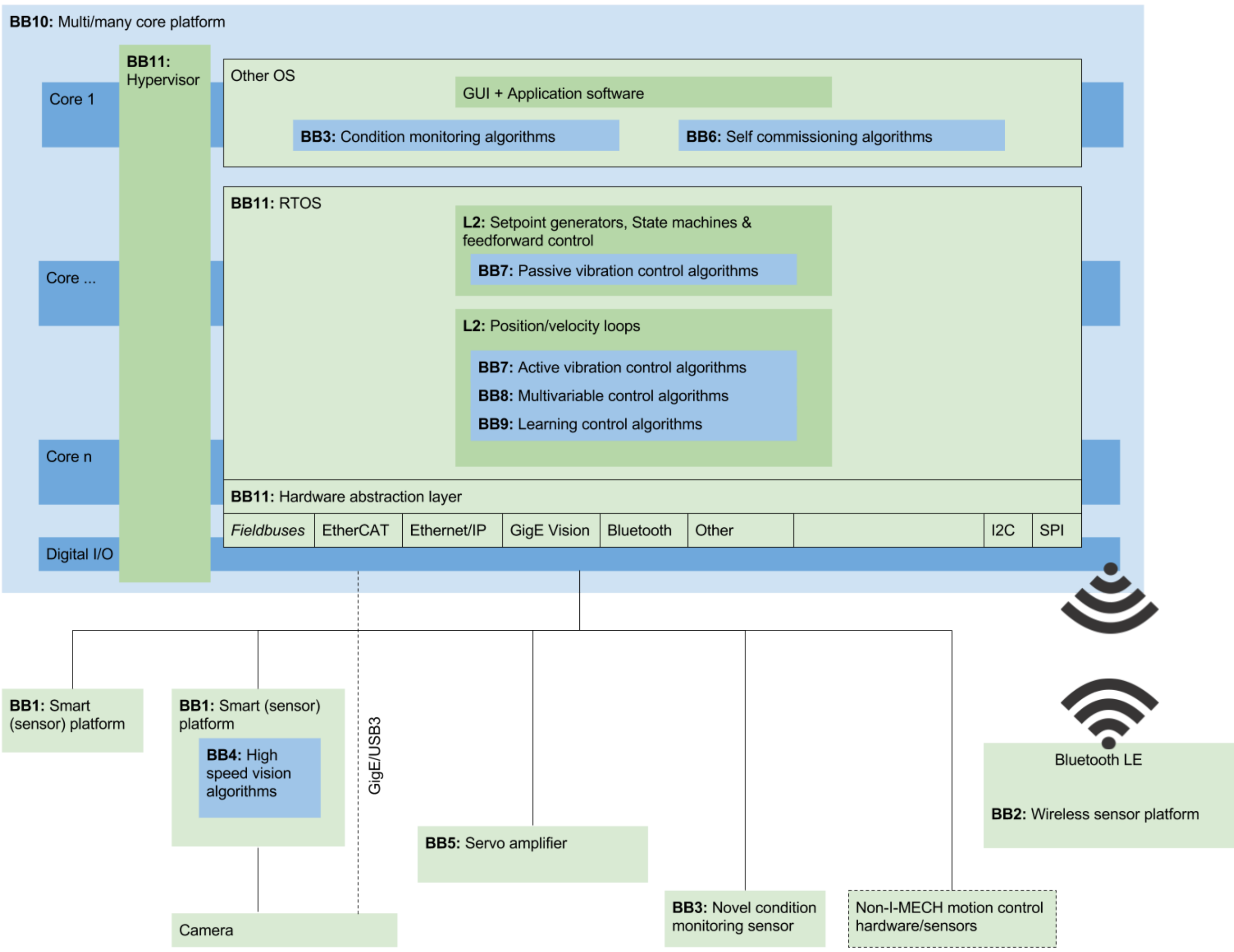

Figure 1: I-MECH platform architecture.

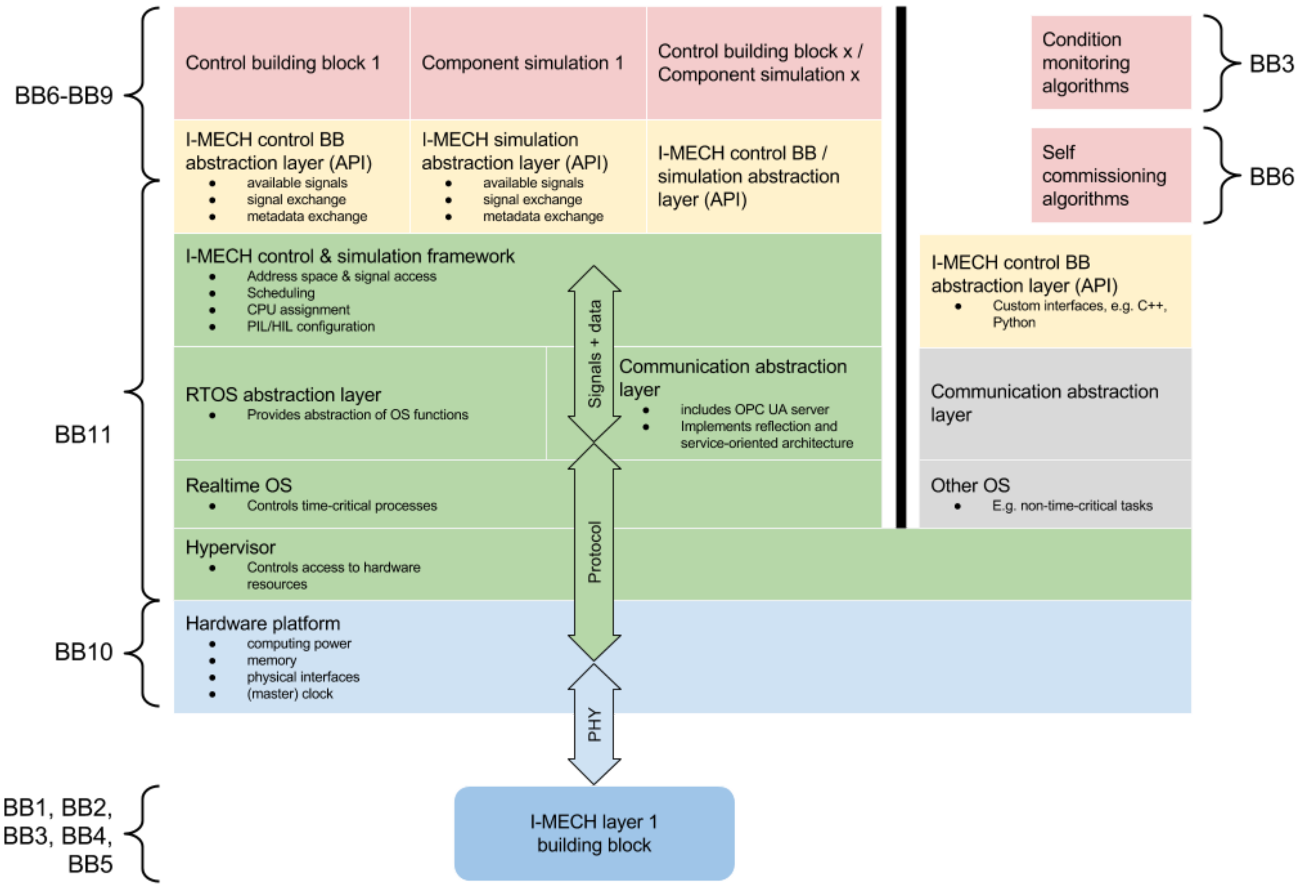

Figure 2: I-MECH platform architecture with abstraction layers.

Figure 2 displays the architecture in a more abstract and layered way and indicates how building blocks are related to the layers. The functions of each layer are also indicated. While most functions are clear, the following functions are emphasised:

- A **sensor/actuator building block**:
    - Communicates via a physical interface (PHY; wired or wireless)
    - Supports a protocol to communicate with the communication abstraction layer, which supports reflection and detection. Note that different protocols for reflection/detection and real-time communication could be used.
    - Transmits signals and metadata to the I-MECH control & simulation framework
    - May provide dedicated sensors and processing for condition monitoring (BB3)
- The **communication abstraction layer** offers the hardware (read: building block) functionality to the software in upper layers by means of a service-oriented software interface and takes care of hardware detection and reflection.
- The **I-MECH control & simulation framework**:
    - Manages all input and output signals relevant for the control & simulation framework
    - Manages controller internal signals (e.g. states)
    - Provides tracing functionality for all (controller) signals
    - Controls timing and synchronization of controllers
    - Assigns controller & simulation computational tasks to computational cores on the hardware platform

- Provides configuration options for PIL/HIL simulation to select which components of the system will be simulated.

- The **I-MECH control BB abstraction layer** is an application programming interface (API) which provides a unified software interface to access input and output signals and other functionality of the I-MECH control framework.
- The **I-MECH simulation abstraction layer** is an application programming interface (API) which provides a unified software interface to provide a simulation of the physical environment to the I-MECH platform. This interface can be used to perform model/processor-in-the-loop (MIL/PIL) simulations.

2.2.1.1 Building block abstraction

Disregarding some subtleties, layer-1 building blocks can be viewed as blocks at sensor and actuator level, while layer-2 control blocks can be viewed as data processing entities. The following abstractions are introduced.

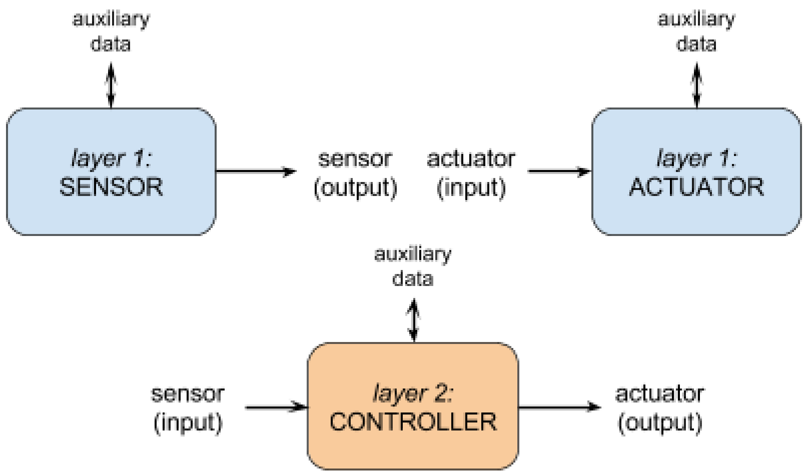

Figure 3: Abstract I-MECH building blocks.

With these abstractions, control chains can be formed (Figure 4).

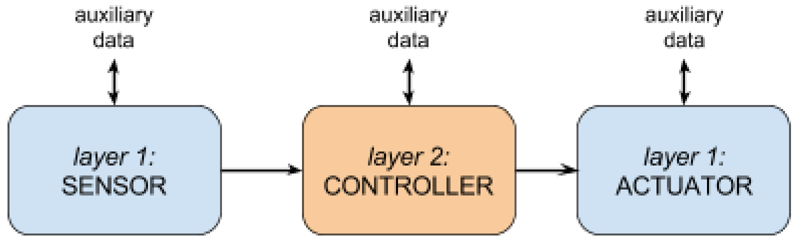

Figure 4: Example: abstract I-MECH building blocks forming a control chain.

The abstractions also allow unit tests to be defined (Figure 5). For each unit, test units should be defined which test all functionality of I-MECH building blocks. That is, there may be a single sensor test unit, which allows any sensor building block (BB1 or BB2) to be verified against the I-MECH requirements. Similar test should be in place for actuators and control building blocks.

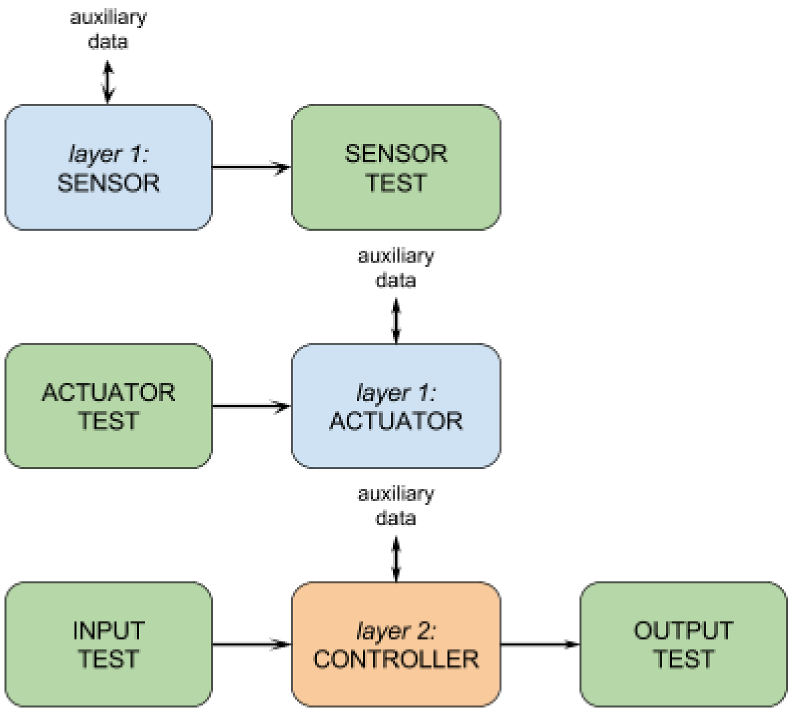

Figure 5: Abstract I-MECH building blocks in a unit test configuration. The green blocks are test blocks.

The abstractions also indicate an auxiliary data flow in addition to the main sensor/actuator data. This data is always of a similar nature for layer 1 building blocks, allowing standardization. For instance, a layer 1 block will typically communicate the following information: condition, state, errors, commands, electronic datasheet, firmware and settings.

2.2.1.2 Hardware and software abstraction

A hardware abstraction layer allows hardware-independent and device-independent software to be written at several levels, by abstracting the hardware-specific implementation details behind generic and well-defined software interfaces. A few examples will be given:

- At instrumentation level, smart sensors and actuators are automatically detected by the instrumentation layer, along with their properties and limitations (e.g. an electronic datasheet). All sensors and actuators (and other communication and network interfaces) are mapped and presented to upper layers. The software is able to access sensor data or send actuator signals through a common software interface which is device-independent. The interface also takes care of status messages and state control.
- In the control layer, any controller software implementation is able to access sensor data from and write output data to all detected devices using a generic software interface, while the real-time OS takes care of deterministic timing. The user (responsible for control software implementation) does not need to worry about low-level communication with hardware.

Advantages of hardware abstraction are clear:

- Higher-level software is not affected by lower-level software changes as long as interfaces are maintained (thus, it is necessary to carefully design standardized, efficient and extensible interfaces)
- Software at all levels is more object-oriented and maintainable
- All types of sensors or actuators share the same interface

2.2.1.3 Model-based system engineering

One of the goals of the I-MECH project is to facilitate model-based systems engineering (MBSE). The essence of MBSE for the I-MECH reference platform is that the platform supports the integration and testing phases in the V-model for development, transitioning from simulation in a virtual environment to testing on real hardware in several steps (Figure 6). That is, one may start with a model of the system, consisting of the control system and the physics which are sensed and manipulated (e.g. a dynamic simulation). In this way, designs can be tested and modified on a **virtual prototype**. With the virtual environment in place, transitions can be made from model-in-the-loop (MIL) to software-in-the-loop (SIL) and processor-in-the-loop (PIL) simulations and hardware-in-the-loop (HIL) for testing purposes. In each stage, a verification and validation cycle can be completed, resulting in the commissioned system in the last step. Because interfaces will be standardized, the same control system which was used in the MIL simulation can be used directly on the real system (possible after code generation/compilation). For instance, a controller designed in Simulink should be testable in combination with a mechanical simulation framework for MIL and SIL testing and the same controller should be deployable onto the I-MECH platform for PIL or HIL testing. This requires that the (software) interfaces are the same and that the controller can be deployed to either environment without any changes.

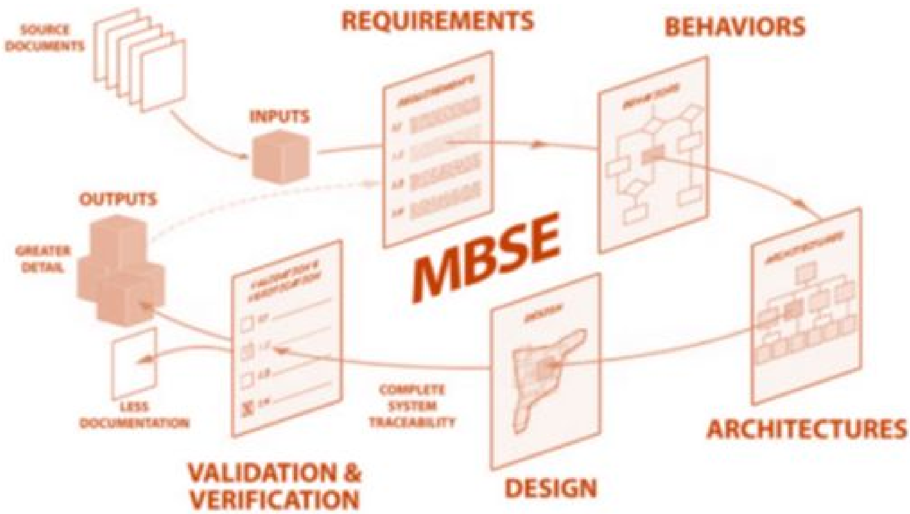

Figure 6: Schematic of the elementary steps in model-based systems engineering.

To support the MBSE paradigm, the I-MECH platform should offer a number of features and tools and a methodology to implement the currently most used software tools for model based engineering. Requirements related to MBSE are listed in 2.2.2 Architectural requirements.

2.2.1.4 System integration

The following table lists which tasks are involved in system integration and how.

Table 1: Tasks involved in system integration

Task	Explanation
All tasks in WP3, WP4 and WP5	All subtasks must keep track of requirements related to system integration, such as interfacing with other parts of the system
Tasks 6.2, 6.3, 6.4, 6.5	Responsible for platform validation, integration and deployment
Task 7	Testing the integrated platform (or partial platform) on pilots
Tasks 3.7, 4.7	Responsible for the multi-many core hardware and software platform and thereby also to a large extent for taking care of hardware abstraction

2.2.2 Architectural requirements

To achieve the I-MECH methodology and abstractions sketched in the previous section, a number of requirements must be satisfied. The following table lists these requirements, which enable a flexible, extensible, and interoperable reference platform. These requirements follow from the desire to create a service-oriented architecture (introduction of abstraction layers and open interfaces) and to reduce (development and commissioning) costs (a modular and maintainable system). All requirements are related to principles and mechanisms; not to solutions.

Table 2: Overall platform requirements

ID	Requirement	Type	Validation	Task
	Interfaces			
rq-D2.3-A.1	The I-MECH reference platform shall only implement hardware interfaces and corresponding protocols with an open standard: interface specification documentation is publicly available (at a reasonable fee). *Note: the specific protocols will be defined in D2.4*	R	I	2.3, 3.1, 4.1, 5.1
rq-D2.3-A.2	All software interfaces in the I-MECH reference platform shall comply with a documented standard (this can be an existing open standard or an open I-MECH standard) *Note: the specific standards for interfaces will be defined in D2.4*	R	I	2.3, 4.1
rq-D2.3-A.3	All software interfaces in the I-MECH reference platform shall be extensible while maintaining backwards compatibility, i.e. an extension of an interface with new functionality shall not cause the system to fail.	O	T	3.1, 4.1, 5.1
rq-D2.3-A.4	Core functionality of I-MECH layer 1 building blocks (BB1, BB2, BB4 & BB5) shall be compatible with non-I-MECH control (layer 2) platforms (e.g. via commonly used industrial interfaces) *Note: how this is offered for the various BBs will be defined in D2.4*	O	T	3.1, 4.1, 5.1
rq-D2.3-A.5	The I-MECH platform shall be extensible to offer compatibility with non-I-MECH layer 1 devices, by adding a communication interface. This applies to: ● I/O, encoder and wireless sensor interfaces ● Servo drives ● Vision modules The hardware abstraction in layer 1 shall be extensible to accommodate such non-I-MECH devices	R	I/T	3.1, 4.1, 5.1
rq-D2.3-A.6	The communication abstraction layer shall offer the possibility to monitor the available capacity on (wireless) communication buses, processing hardware and memory in such a way that it is possible to respond to limitations (i.e.implementation of a graceful failure mechanism).	R	T	3.1, 4.1
	Scalability			
rq-D2.3-A.7	The I-MECH control & simulation software framework shall offer the capability to:	R	T	3.1, 4.1

	• assign execution of functions to specific computational cores • optimize execution of functions for parallel processing, tailored to the capabilities (i.e. number of cores) of the selected multi-many-core platform			
	Tools, documentation and completeness			
rq-D2.3-A.8	The I-MECH platform shall offer interfaces for model-based development, comprising: • Support for MIL/SIL/PIL/HIL simulation (virtual prototyping) of layer 1 devices and plant physics in a (graphical) modelling environment (e.g. Simulink, AMESIM,…) • Possibility to configure which of the layer 1 building blocks/system modules will be simulated and which will be real hardware during HIL testing. • Support for real-time simulations of virtual layer 1 building blocks/system modules on the I-MECH motion control platform for PIL/HIL testing, where the simulation environment is connected to the control & simulation framework via a simulation abstraction layer (real-time performance depending on selected simulation tool, model complexity and computing performance of the I-MECH platform BB10). • Interfaces for physical simulation tools allowing I-MECH controllers to be used to control simulated systems in a (third party) simulation environment for MIL/SIL testing. Functionality shall at least be verified by demonstrating integration with a simulation environment comparable to AMESIM (for 1D simulation purposes) or OOFELIE multiphysics (for 3D simulation purposes). *Note: exact specifications for the interfaces for MIL/SIL/HIL/PIL will be defined in D2.4.*	R	I/T	4.1, 4.3
rq-D2.3-A.9	The I-MECH platform shall be accompanied by a complete list of required deployment (software) tools, i.e. all software and hardware tools required to commission a system	R	I/T	3.1, 4.1, 5.1
rq-D2.3-A.10	Methodology and guidelines for developing and using an I-MECH platform shall be available, including documentation at system and building block level	R	I	3.1, 4.1, 5.1
rq-D2.3-A.11	The I-MECH reference platform shall be complete in the sense that with the available I-MECH building blocks it shall at least be possible to control an AC induction motor with a PID controller, by: • SIL/PIL/HIL simulation of the system using at least a simple 1D mechanical representation. • Composing and configuring the control system using an	R	T	3.1, 4.1, 5.1

	MBSE toolset. ● Deploying and testing the control system. The steps above will demonstrate 'completeness' of the I-MECH platform.			
rq-D2.3-A.12	Software algorithms (i.e. smart processing or control) for I-MECH building blocks in both layer 1 and 2 (e.g. for BB1, BB3, BB6, BB7, BB8) should be deployable on non-I-MECH hardware and software platforms. This could be facilitated by, for instance: ● Developing in a (graphical) environment that supports deployment to various platforms by means of code generation for various platforms (e.g. C, PLC & VHDL code). ● Documenting the algorithm, including the methodology and possible implementation. This will ensure algorithms could also be applicable to non-I-MECH motion control platforms. This will not be a requirement for layer 3 software and low-level software like the RTOS, I-MECH control & simulation framework and the abstraction layers. *Note: development in an environment supporting code generation is recommended for (motion) control algorithms.*	O	I	3.1, 4.1, 5.1
rq-D2.3-A.13	Support for model-based development and MIL/HIL/SIL simulation and validation workflows will be demonstrated (e.g. controller design in Simulink and deployment to the I-MECH platform).	R	T	3.1, 4.1, 5.1
	User experience, maintainability and cost			
rq-D2.3-A.14	The I-MECH platform shall offer any (acquired) data for diagnostics and predictive maintenance via OPC UA. *Note: A standard for the used information model is still to be determined.*	R	T	5.1, 5.2, 5.3
rq-D2.3-A.15	The I-MECH reference platform shall be an open platform, implying that interface specifications (hardware and software) are open and completely documented Note: software code itself will not necessarily be open-source *Note: openness of the platform will be improved when modularity of the platform increases such that small parts of the platform can be exchanged for/extended with custom solutions. D2.4 will specify the modularity of the platform in more detail (e.g. modularity within abstraction layers).*	R	I	3.1, 4.1, 5.1

2.2.3 Requirements on layer 1 sensor and actuator building blocks

A layer 1 device can be a standard sensor or actuator, but also a smart sensor, smart actuator or even a smart module controlling multiple actuators and sensors with a decentralized control strategy. A layer 1 sensor or actuator block will be an I-MECH building block if it meets the following requirements.

Table 3: Layer 1 building block requirements

ID	Requirement	Type	Validation	Task

rq-D2.3-L1.1	There shall be a discovery mechanism for layer-1 I-MECH building blocks (note that IEEE 1451 only partially addresses such needs), i.e presence of layer-1 devices is detected by the I-MECH platform *Note: mechanism will be specified in D2.4*	O	T	3.1, 4.1

rq-D2.3-L1.2	I-MECH building blocks shall support a reflection interface supporting at least:
	● identification (containing identifier and supported functionality)
	● what is your state (installing, idle, error, operational)
	● a metadata description, describing actuator and sensor characteristics (e.g. IEEE 1451: Transducer Electronic Datasheets)
	● calls to inquire/send:
	    ● data types
	    ● value ranges
	    ● sample rate
	    ● status
	    ● calibration data
	    ● configuration
	    ● firmware
	All building blocks shall follow the same standard.
	Note: interface will be specified in D2.4
		R	T	3.1, 4.1

rq-D2.3-L1.3	There shall be a firmware update mechanism for all building blocks via the main communication interface with the central control platform. Local access to hardware shall not be required for firmware updates.	R	T	3.1, 4.1
rq-D2.3-L1.4	I-MECH building blocks shall be able to report their state (i.e. the internal state such as ready, initializing, firmware updating, …) and condition (i.e. health and performance of the device).	R	T	3.1, 4.1
rq-D2.3-L1.5	High-precision time stamping of signals will be possible, e.g. conforming to ISO 8601 for the date/time format *Note: format will be frozen in D2.4*	R	T	3.1, 4.1
rq-D2.3-L1.6	Time will be synchronized (synchronized clocks) across sensors and layers (e.g. using IEEE 1588 [1])[1] *Note: standard will be frozen in D2.4*	R	T	3.1, 4.1
rq-D2.3-L1.7	Whether a sensor is wired or wireless shall not affect the interface or way in which it is used by the I-MECH control & simulation framework (layer 2)	R	T	3.1, 4.1
rq-D2.3-L1.8	A graceful failure mechanism shall be specified for wireless sensors used in closed-loop control, when bandwidth is insufficient or communication is lost.	R	T	3.1, 4.1, 4.7
rq-D2.3-L1.9	Control loops will preferably be closed via level 2 hardware and software. If performance or safety demands preclude such a solution, control loops may be closed locally on layer 1 hardware. In such cases, it should still be possible to monitor/trace and configure the control loop from layer 2, by exchanging the appropriate signals and	O	I	3.1, 4.1, 3.7, 3.3, 4.7

[1] Also see https://www.ohwr.org/projects/white-rabbit

	parameters. *Note: it will be decided whether additional requirements are necessary for smart I-MECH layer 1 modules with local decentralized control loops to improve integration of centralized and decentralized control architectures during the face to face meeting February 2018. A revision of this document will be published after this meeting.*			

2.2.4 Requirements on layer 2 framework

The layer 2 building blocks rely on a software framework on the reference platform, provided by BB11. The layer 2 framework must fulfill the following requirements to enable integration of the control building blocks into the I-MECH platform.

Table 4: Layer 2 framework requirements

ID	Requirement	Type	Validation	Task
rq-D2.3-L2.1	Layer 2 shall offer a control and simulation framework abstraction layer for layer-2 control algorithms and simulation models: ● The API shall allow control software to be developed independent of changes in the control & simulation framework and below ● Knowledge of details of communication with layer 1 devices shall be of no concern to layer 2 algorithms due to the abstraction layer *Note: the API will be specified in D2.4*	R	T	4.1, 4.7
rq-D2.3-L2.2	A detection mechanism for layer-1 building blocks in layer 2 ● Layer 2 is presented with a pool of available layer-1 devices *Note: the mechanism will be specified in D2.4*	R	T	4.1, 4.7
rq-D2.3-L2.3	Time stamping of all signals which are inputs or outputs of layer 2. The electronic datasheet will specify the accuracy.	R	T	4.1, 4.7
rq-D2.3-L2.4	The layer 2 control & simulation framework shall offer the signal handling to accommodate MIMO control (feedforward and feedback). This includes: - Offering all sensor signals synchronized (optionally via interpolation using time stamps) - Processing all actuator signals (i.e. such that outputs are synchronized in time) - Deterministic scheduling of events (e.g. sending outputs to devices) - Support for multirate execution of control algorithms	R	T	4.1, 4.7
rq-D2.3-L2.5	Support to monitor and trace all signals (including current loop signals and layer-1 BB internal signals) with accurate timestamps (see rq-D2.3-L2.3) and at sample rates at which the signals appear in the system, suited for: ● development purposes/troubleshooting, e.g. using a scope application/GUI ● measurement of system dynamics (e.g., frequency responses) (coupled to BB6)	R	T	4.1, 4.2, 4.7 5.1, 5.3

	● monitoring of (closed-loop) performance data (coupled to BB3) *Note: Number of traceable signals and sample rate will depend on hardware performance of BB10. Requirements for the performance of BB10 will be specified by pilots.*			
rq-D2.3-L2.6	Support for flexible and user-defined control algorithms, facilitated by development of algorithms in a high-level language or model based engineering environment or via coding by hand in lower level languages. These user defined algorithms should connect to the control & simulation framework via the I-MECH Control BB abstraction API.	R	T	4.1, 4.7
rq-D2.3-L2.7	Support for real-time execution of user-defined simulation models for PIL simulation and HIL testing. Simulation environments should connect to the control & simulation framework via the I-MECH simulation abstraction API.	R	T	4.1, 4.7

2.2.5 Requirements on interfaces to layer 3

The motion platform will be able to connect to behavioural layers (e.g. a supervisory factory control system) using industry-compliant standards and protocols. Also see “Industrie 4.0 reference architecture RAMI4.0”. The table below summarizes the standards.

Table 5: Layer 3 requirement specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-L3.1	Layer 3 shall at least support OPC Unified Architecture (IEC 62541) as communication protocol with factory systems and communication with subsystems (implementation of client and server), for instance for condition monitoring. *Note: the information model will be specified in D2.4*	R	T	Table	5.1
rq-D2.3-L3.2	Support for OPC Unified Architecture TSN (Time Sensitive Networking, a real-time implementation of OPC UA)	O	T	Table	5.1

2.3 Subsystem-level (building block) requirements

2.3.1 A note on performance specification tables

Most requirement specification tables below include detailed and application-specific specifications. For instance, a power range of 50W-10kW is listed for the High-performance servo amplifier. These values are only intended as a ballpark range corresponding to what pilots and demonstrators require (from the requirements investigation; D2.1/D2.2). It does not mean that the I-MECH partners will create building blocks to meet every possible specification. It means that the I-MECH platform should facilitate designing hardware for these specifications if required.

2.3.2 BB1: Platform for Smart Sensors with Advanced Data Processing

This building block facilitates the incorporation of advanced sensors into the motion control platform. Two of these are *BB2: Real-time wireless sensors* and *BB4: High speed vision*. These building blocks require advanced data processing at high speeds in order to extract the signals relevant for feedback control. The table below describes the requirements in detail.

Note: we envision that BB1 can also be used to create a smart actuator or smart module with control loops decentralized with respect to control loops executed on BB11. In such a case BB1 could be combined with BB5 (high performance servo amplifier) via a piggy-back construction on the BB1 PCB used for the smart sensor/actuator/module platform board. Such a decentralized control approach could be necessary to obtain the required update rates and bandwidths of position loops. The next face-to-face meeting will be used to discuss how to facilitate such an approach and whether additional requirements are necessary (e.g. whether it should be possible to execute the I-MECH control & simulation framework on BB1). Decisions will be published in a new revision of D2.3.

Table 6: Smart sensor platform specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-BB1.1	Support for flexible, reconfigurable and real-time data processing algorithms, e.g.: ● Decimation, filtering and signal processing ● Complex mathematical models and estimation/fusion ● Sensor fusion ● Transformation from physical to virtual axes and vice versa	R	T/I	DoA	3.1, 3.3
rq-D2.3-BB1.2	FPGA for real-time, high-speed and parallel processing of critical data	O	T	DoA	3.1, 3.3
rq-D2.3-BB1.3	Encoder: ● Protocols: ○ SINCOS ○ SSI ○ S0S90 quadrature ○ EnDat 2.x ○ BISS-C ○ Hiperface ● Absolute and relative ● Interpolation, encoder correction ● Resolution depends on encoder	R[2]	I/T	Table	3.1, 3.3
rq-D2.3-BB1.4	High-speed ADC support: ● Range: +/- 10V, 0-5V, 0-10V[3] ● 2 - 16 channels ● 12 - 24 bit ● at least 8kHz	R	T	Table	3.1, 3.3
rq-D2.3-BB1.5	Analog outputs support: ● Range: +/- 10V, 0-5V, 0-10V ● 4 - 6 channels ● 16 - 18 bit ● at least 8kHz	R	T	Table	3.1, 3.3
rq-D2.3-BB1.6	Digital IO support: ● type: TTL, 24V tolerant, PNP ● 5-48 channels	R	T	Table	3.1, 3.3
rq-D2.3-BB1.7	Camera input(s): see BB4 (USB 3.x, 1Gbps ethernet)	O	T	Table	3.1, 3.5
rq-D2.3-BB1.8	Wireless input(s): see BB2	O	T	Table	3.1, 3.4

[2] Not all encoder protocols may need to be supported.
[3] Specific interfaces such as 4-20mA and IEPE can be provided by separate COTS modules

2.3.3 BB2: Real-time wireless sensors

This building block should facilitate the reliable inclusion of wireless sensors in real-time feedback control. Key aspects are reliability of the wireless data, low energy consumption of the wireless nodes and high update rates / low latency. For certain applications, e.g. a proximity sensor or environmental sensors, the real-time performance may be relaxed. This building block on the transmitter/sensor side will be a low-power variant of BB1 (a low-power platform for smart sensors).

Table 7: Real-time wireless sensors specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-BB2.1	Wireless power transfer	O	T	Table	3.1, 3.3, 3.4
rq-D2.3-BB2.2	Energy harvesting	O	T	Table	3.1, 3.3, 3.4
rq-D2.3-BB2.3	Bluetooth low-energy, Zigbee, WiFi, proprietary UWB *Note that WiFi and UWB are mutually exclusive since they share the same band*	R	T	Table	3.1, 3.4
rq-D2.3-BB2.4	Bandwidth: depends on data transfer	O	T	Table	3.1, 3.4
rq-D2.3-BB2.5	For wireless sensors a bandwidth cannot be guaranteed. In real time control applications a method must be specified to fail gracefully when the bandwidth is insufficient.	R	T	WP2 Team	3.1, 3.4
rq-D2.3-BB2.6	Update rate: ● At least 1-2 kHz (1ms-500μs) for real-time control ● Lower (how fast?) for other purposes	R	T	Table	3.1, 3.4
rq-D2.3-BB2.7	Latency: ● Preferably less than half a sample period to avoid excessive phase loss for control applications	R	T	Table	3.1, 3.4
rq-D2.3-BB2.8	Secure communication: to be determined	O	I	Table	3.1, 3.4
rq-D2.3-BB2.9	Environmental: nodes must frequently operate in harsh environments	O	T	Table	3.1, 3.4
rq-D2.3-BB2.10	Sensor types: ● Encoders ● Proximity sensors ● Temperature ● Acceleration ● Vibration ● Mote layers (see Tyndall link)	O	T	Table	3.1, 3.4
rq-D2.3-BB2.11	Battery life (if applicable): 6 months - 1 year (depends on application criticality)	O	T	Table	3.1, 3.4

2.3.4 BB4: High-speed vision

The vision system in layer 1 may take several forms, ranging in complexity from e.g., in-line inspection for adjustment of process parameters to real time vision-in-the-loop applications, placing stringent demands on performance.

Table 8: High-speed vision specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-BB4.1	Update rate (per camera) ● Vision-in-the-loop: 5kHz ● Inspection: 100Hz	R	T	Table	3.1, 3.5

rq-D2.3-BB4.2	Integrated image processing (e.g. feature detection and extraction of 'smart metrics')	R	T	Table	3.1, 3.3, 3.5
rq-D2.3-BB4.3	Integrated image pre-processing (e.g. data reduction)	R	T	Table	3.1, 3.3, 3.5
rq-D2.3-BB4.4	(Multiple) region of interest (ROI) extraction	R	T	Table	3.1, 3.3, 3.5
rq-D2.3-BB4.5	Pixel binning (e.g., 2x2, 3x3, …)	R	T	Table	3.1, 3.5
rq-D2.3-BB4.6	Number of cameras: flexible (3-12)	O	T	Table	3.1, 3.5
rq-D2.3-BB4.7	Camera interfaces: Gigabit ethernet, USB 3.x, SpaceFibre	O	T	Table	3.1, 3.5
rq-D2.3-BB4.8	Camera bandwidth: 800Mbps - 5 Gbps (Depends on application/interface)	R	T	Table	3.1, 3.5
rq-D2.3-BB4.9	Resolution: 5-12 MPix (depends on application)	O	T	Table	3.1, 3.5
rq-D2.3-BB4.10	Applied standards: GigE Vision 2.0 (applies to ethernet only, for PTP time synchronization)	R	T	WP2 team	3.1, 3.5

2.3.5 BB5: High-performance servo amplifier

The high-performance servo amplifier building block should support 1-phase and 3-phase motors, including voice coils, brushless and brushed DC/AC motors, reluctance motors and inductance motors. Note that stepper motors are outside the scope of BB5. For efficiency, cost and performance reasons, amplifiers will be of the class D current amplifier type (i.e., using pulse width modulation (PWM)). The nature of these amplifiers demands proper EMC design.

The table below shows the ranges (e.g., power, current, …) that have been observed in the requirements table and do not imply that a single amplifier should be able to fulfill all requirements. Requirements for amplifiers suited for specific actuators of pilots will be specified in detail for each pilot that will use this block as part of task T3.1. A few requirements, mainly related to interfacing and smart functions of the amplifier are relevant for multiple I-MECH applications.

Performance characteristics are summarised below.

Table 9: High-performance amplifier specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-BB5.1	Voltage/PWM feedforward (also known as jerk feedforward)	R	T	Table	3.1, 3.6, 4.1, 5.1, 5.3
rq-D2.3-BB5.2	Correction for bus voltage fluctuations	R	T	Table	3.1, 3.6, 4.1, 5.1, 5.3
rq-D2.3-BB5.3	Automatic tuning of current loop	R	T	Table	3.1, 3.6, 4.1, 5.1, 5.3
rq-D2.3-BB5.4	Manual tuning of current loop in frequency and time domain	R	T	Table	3.1, 3.6, 4.1, 5.1, 5.3
rq-D2.3-BB5.5	Adjustable firmware/control algorithm	R	T	Table	3.1, 3.6, 4.1, 5.1, 5.3
rq-D2.3-BB5.6	Active EMI cancellation concepts	O	T	Table	3.1, 3.6, 4.1, 5.1, 5.3
rq-D2.3-BB5.7	Amplifier condition monitoring and error feedback	R	T	Table	3.1, 3.6, 4.1, 5.1, 5.3
rq-D2.3-BB5.8	Peak power rating: 50W - 10kW (application-dependent)	O	T	Table	3.1, 3.6
rq-D2.3-BB5.9	Peak current rating: 6A - 30A (application-dependent)	O	T	Table	3.1, 3.6

rq-D2.3-BB5.10	RMS current rating: 2A - 10A (application-dependent)	O	T	Table	3.1, 3.6
rq-D2.3-BB5.11	Bus voltage supply: 24VDC - 400VAC (application-dependent)	R	T	Table	3.1, 3.6
rq-D2.3-BB5.12	Signal-to-noise ratio: 90 dB (matches resolution)	O	T	Table	3.1, 3.6
rq-D2.3-BB5.13	Current resolution: 16 bit (typ. 1 mA at 30A full-scale)	R	T	Table	3.1, 3.6
rq-D2.3-BB5.14	Current loop bandwidth: at least 10kHz (application-dependent)	R	T	Table	3.1, 3.6
rq-D2.3-BB5.15	Current loop sample rate: up to 384 kHz (application-dependent) (e.g., a high rate may be necessary for active EMC suppression techniques)	O	T	Table	3.1, 3.6
rq-D2.3-BB5.16	PWM frequency: Depends on amplifier architecture, synchronized with current loop and position loop (i.e. all switching and control frequencies synchronized)	O	T	Table	3.1, 3.6
rq-D2.3-BB5.17	Commutation methods: ● Field-oriented control using encoder feedback ● Commutation using back-EMF measurement	R	T	Table	3.1, 3.6
rq-D2.3-BB5.18	Protection: ● full protection ● OVP, UVP, Imax, I2T, Tmax	R	T	Table	3.1, 3.6
rq-D2.3-BB5.19	Safety: ● support PLe and all features for medical purposes (STO, SS1/2/3, …) ● Configurable per axis	R	T	Table	3.1, 3.6
rq-D2.3-BB5.20	Size: As small as possible while satisfying EMC requirements, depends on application	R	T	Table	3.1, 3.6

2.3.6 Common control building block requirements

The layer 2 building blocks related to control algorithms share some functional requirements. These are first summarized below, after which specific functional requirement are listed in the subsequent chapters.

Table 10: General control software specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-CB.1	Layer 2 control blocks shall support online controller tuning and parameter adaptation (e.g., for tuning, to account for varying dynamics, for gain-scheduling control)	R	T	DoA	4.1, 4.2, 4.7, 5.1
rq-D2.3-CB.2	Support for (model-based) feedforward of (a.o.) ● position ● velocity ● acceleration ● jerk ● friction	R	T	WP2 team	4.1, 4.2

rq-D2.3-CB.3	Layer 3 software algorithm building blocks shall use a standardized API to communicate with the I-MECH control & simulation framework. This API should offer at least the following functionality: ● exchange of signals, metadata and parameters with the I-MECH control & simulation framework ● exposing the list of accessible signals, configurable parameters and data-objects used within the building block to the I-MECH control & simulation framework.	R	I/T	WP2 team	4.1, 4.4, 4.5, 4.6, 4.7

2.3.7 BB6: Self-commissioning velocity and position control loops

Self-commissioning of control loops is an important building block to facilitate the commissioning of motion axes in motion systems. In many motion control systems there are at least two control loops: a current control loop (often in the current amplifiers; see BB5) and a position control loop. Sometimes, a velocity control loop is inserted between the former two.

Table 11: Self-commissioning velocity and position control loop specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-BB6.1	Self commissioning of current, velocity and position controllers	R	T	DoA	4.1, 5.4
rq-D2.3-BB6.2	Automatic tuning of current, velocity and position control loops ● I-MECH (PID or more advanced) control algorithms should have a corresponding tuning algorithm or parameterization which lends itself to auto-tuning ● Data exchange between layer 2 and tuning algorithm (controller model, FRF data, etc.) should be standardized	R	T	DoA	4.1, 4.2, 4.7, 5.4
rq-D2.3-BB6.3	Automated feedforward commissioning and periodic or ad-hoc fine-tuning to compensate for changes		T	DoA	4.1, 5.4
rq-D2.3-BB6.4	System identification tools to estimate time-domain or frequency-domain models. ● To aid controller tuning ● To aid feedforward tuning ● To aid robust multivariable controller design A choice of excitation signals (e.g. multisine, (filtered) noise, sweep, step, impulse) and possibilities to inject these signals at various places in the control loop.	R	T	DoA	T
rq-D2.3-BB6.5	Support for model-based control and/or load-side auxiliary feedback (see also BB8)	R	T	DoA	4.1, 4.5, 5.4

2.3.8 BB7: Vibration control module

The vibration control module provides functionality to alleviate vibrations in flexible motion systems. It can be seen as a self-supported building block, but also interfacing with other building blocks (BB6, BB8 and BB9) in terms of feedforward signal shaping and control loop augmentation using load-side measurements.

Table 12: Vibration control module specifications

ID	Requirement	Type	Validation	Source	Task

rq-D2.3-BB7.1	Load-side feedback to augment existing control loops	R	T	DoA	4.1, 4.4
rq-D2.3-BB7.2	Load detection to deal with varying or unknown loads	R	T	DoA	4.1, 4.4
rq-D2.3-BB7.3	Passive vibration reduction; e.g., automatic detection and estimation of (lightly damped) resonance modes and appropriate controller adaptation (avoiding excitation of resonances)	R	T	DoA	4.1, 4.4
rq-D2.3-BB7.4	Active vibration reduction; e.g., using feedback	R	T	DoA	4.1, 4.4
rq-D2.3-BB7.5	Physically intuitive software tools to aid commissioning of BB7 technologies in industrial practice	R	T	DoA	4.1, 4.2, 4.4, 4.7

2.3.9 BB8: Robust model-based multivariable control

The robust model-based multivariable control building block will contribute to a performance increase, by relying on a physical model of the controlled process. If such models are uncertain or too complicated, identification of system dynamics offers an alternative.

Table 13: Robust model-based multivariable control specifications

Specification	Requirement	Type	Validation	Source	Task
rq-D2.3-BB8.1	Multivariable control which takes into account mutual interactions and coupling (MIMO)	R	I	DoA	4.1, 4.2, 4.3, 4.5
rq-D2.3-BB8.2	Robust control strategies which allow quantification of model uncertainty and appropriate adjustment of the controller to cope with these uncertainties (in collaboration with system identification in BB6)	R	T	DoA	4.1, 4.2, 4.3, 4.5
rq-D2.3-BB8.3	System identification tools suited to industrial practice	R	T	DoA	4.1, 4.2, 4.3, 4.5

2.3.10 BB9: Iterative and repetitive control module

BB9 offers the functionality to improve motion system performance for repetitive tasks which are similar in nature.

Table 14: Iterative and repetitive control module specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-BB9.1	Implementation(s) of iterative learning control for repetitive tasks, sufficiently robust for industrial application	R	T	DoA	4.1, 4.6
rq-D2.3-BB9.2	Implementation(s) of repetitive control for repetitive disturbances.	R	T	DoA	4.1, 4.6
rq-D2.3-BB9.3	Robust learning control: • Robustness to process variations (e.g., dealing with non-repetitive but similar tasks) • Robustness to system variations (e.g., sensitivity to variations in system dynamics)	R	T	DoA	4.1, 4.6

2.3.11 BB10: Control specific multi/many core platform

BB10 will provide the backbone of the I-MECH motion platform, coordinating all tasks, ranging from time-critical tasks like computing the control loops and safety monitoring to less time-critical tasks such as data processing and logging. Requirements for BB10 are listed in the table below.

Table 15: Control-specific multi-many core platform specifications

ID	Requirement	Type	Validation	Source	Task
rq-D2.3-BB10.1	Multi-many core for parallel processing and task scheduling At least 2 cores	R	I	Table	3.1, 3.7
rq-D2.3-BB10.2	The hardware platform shall have a precise clock used for deterministic and predictable timing of critical tasks	R	T	Table	3.1, 3.7
rq-D2.3-BB10.3	The platform shall have minimal latency in the real time control loop (i.e latency between encoder reading and next current setpoint)	R	T	Table	3.1, 3.7
rq-D2.3-BB10.4	Control loop sample rates: • 4kHz - 20kHz (250μs - 50μs sample time) Fully synchronous with PWM and all other clocks	R	T	Table	3.1, 3.7, 4.1, 4.2
rq-D2.3-BB10.5	Control loops fully synchronous with PWM and all other clocks	R	T	WP2 team	3.1, 3.7, 4.1, 4.2
rq-D2.3-BB10.6	Instruction set: • Intel x86:64-bit instruction set VMX, VT-D, EPT (extended page tables), unrestricted guest mode, preemption timer • ARM SoC: ARMv7 with virtualization extensions or ARMv8	R	I	Table	3.1, 3.7
rq-D2.3-BB10.7	Realtime exclusive access to peripherals (PCI/PCIe)	R	T	Table	3.1, 3.7
rq-D2.3-BB10.8	Realtime exclusive access to I/O sub-system Direct memory access	R	T	Table	3.1, 3.7
rq-D2.3-BB10.9	Hypervisor support	R	I	Table	3.1, 3.7

2.3.12 BB11: RTOS for multi/many core platform

Requirements for BB11 are summarized in the table below. An important requirement to support the control algorithms (BB6-BB9) and to support flexibility, user-configurability and long term software maintainability and sustainability is the possibility to implement algorithms in a high-level language.

Table 16: RTOS for multi-many core platform specifications

Specification	Requirement	Type	Validation	Source	Task
rq-D2.3-BB11.1	Programming languages (control algorithm): • C/C++ • Python (only for scripting?) • MATLAB/Simulink • IEC 61131-3 • Support for GCC languages	R	I	Table	3.7, 4.1, 4.7, 5.1
rq-D2.3-BB11.2	General-purpose OS (Hypervisor should allow execution of general purpose OS besides RTOS): • Linux • Windows	O	I	Table	3.7, 4.1, 4.7
rq-D2.3-BB11.3	Hypervisor for virtual machine management	R	I	Table	3.7, 4.1, 4.7
rq-D2.3-BB11.4	Certified RTOS for time-critical tasks	R	I	Table	3.7, 4.1, 4.7

rq-D2.3-BB11.5	The motion control specific RTOS shall offer an I-MECH control & simulation framework that shall have at least the following functionality: ● manage address space with addresses of all available signals, parameters and data-objects within layer 2 ● manage access to items in address space ● facilitate scheduling of algorithms connected with the I-MECH API to the control & simulation framework ● CPU assignment of algorithms connected with the I-MECH api to the control & simulation framework	R	I	WP 2 Team	3.7, 4.1, 4.7
rq-D2.3-BB11.6	An RTOS abstraction layer shall provide abstraction of OS-specific functions required by the I-MECH control & simulation framework.	O	I	WP 2 Team	3.7, 4.1, 4.7
rq-D2.3-BB11.7	A communication abstraction layer shall provide abstraction of communication protocols used to communicate with the control & simulation framework. ● it shall be possible to extend the abstraction layer to support new communication protocols without modifying the interface of the abstraction layer with the I-MECH control & simulation framework. ● the abstraction layer shall implement an OPC UA server to allow access to signals, parameters and metadata stored in the address space of the I-MECH control & simulation framework.	R	I	WP 2 Team	3.7, 4.1, 4.7

2.4 Operability requirements

2.4.1 Operating conditions

Required operating conditions are highly application-specific. The I-MECH reference platform will not adhere to a particular class of environmental conditions. The conditions which apply (e.g.. following from the individual hardware components) shall be specified. These could for instance be:

- +0°C to +70°C (commercial)
- +20°C to +24°C (cleanroom)
- -25°C to +85°C (industrial)
- International Protection Marking IP20

2.4.2 Maintainability

2.4.2.1 BB3: Robust condition monitoring and predictive diagnostics

BB3 will gather performance and diagnostic data from dedicated sensors as well as data provided directly by other building blocks. This data allows for direct monitoring of overall system performance, predictive maintenance as well as gathering long-term statistics. Requirements on BB3 are summarized below.

Table 17: Robust condition monitoring and predictive diagnostics requirements

ID	Requirement	Type	Validation	Source	Task

rq-D2.3-BB3.1	Model-based fault detection and diagnostics	R	T	DoA	5.1, 5.2, 5.3
rq-D2.3-BB3.2	Machine (tool) diagnostics using miniaturized MEMS sensors	R	T	DoA	5.1, 5.3
rq-D2.3-BB3.3	Automated diagnostics of controller performance to signal degradation due to system changes (e.g., wear, friction, process changes)	R	T	DoA	5.1, 5.2, 5.3
rq-D2.3-BB3.4	Consolidation of data and processing into intuitive error/diagnostic messages to the user and/or factory	R	T	WP2 team	5.1, 5.2, 5.3
rq-D2.3-BB3.5	Smart sensors with integrated processing to monitor: ● Bearing degradation and wear ● Temperatures (heat sinks, power electronics, oils, lubricants, components, …) ● Noise emission and vibration (bearings, unbalance, transmissions, …)	R	T	DoA	3.3, 5.1, 5.3
rq-D2.3-BB3.6	Processing of condition/diagnostics data from all connected building blocks	R	T	DoA	5.1, 5.2, 5.3
rq-D2.3-BB3.7	Streaming of unprocessed (raw) performance and diagnostic data	R	T	WP2 team	5.1, 5.2

2.4.3 Safety

The control platform must be able to support a range of safety features, related to motion systems operating in proximity to humans, electrical safety, or protecting hardware. While many safety features can be implemented in software, some of them must be implemented in hardware to allow safety features to engage without delay.

2.4.3.1 Motion system safety requirements

This section details the standards we wish to be able to adhere to. A number of (application-dependent) safety features are required by consortium members. Safety requirements (e.g. PIL/SIL level) will apply to a complete system in which the I-MECH motion control platform will be integrated. It is not desirable that all components of the I-MECH platform should be certified for these safety requirements, as the amount of software code to be verified will be too large. I-MECH components should therefore be designed such that they will not prevent a system integrator applying the I-MECH platform from achieving the safety requirements listed below. Generic interfaces should facilitate adhering to standard and BBs should state the standard adhered to.

Table 18: Required safety features

ID	Feature/Standard	Sub-feature/Specification	Type	Validation	Source	Task
rq-D2.3-S.1	Granularity of safety system	Safety features should be configurable per drive (e.g. other drives may remain active during a safety event)	R	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.2	Safe switch-off	Safe Torque Off (STO): Safe disabling of the torque by means of immediate switching off of the energy supply. Safe Brake Control (SBC): Safe activation of brake.	R	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.3	Safe standstil	SS1: Safe monitored standstill, followed by disabling the torque of the drive. SS2: Safe monitored standstill followed by standstill monitoring while torque remains enabled.	O	I	Table	3.1, 4.1, 6.2, 6.3, 6.4

rq-D2.3-S.4	Safe monitoring	SSM: Safe Speed Monitoring SCA: Safe Cam SLT: Safely Limited Torque SMT: Safe Motor Temperature	O	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.5	Safe positioning	SLP: Safely-Limited Position SLI: Safely-Limited increment	O	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.6	Safe motion	SLS: Safely-Limited-Speed SSR-Safe Speed Range SDI-Safe Direction SLA-Safely Limited Acceleration SAR-Safe Acceleration Range Collision detection sensor Safe motion based on 3D model	O	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.7	Other safety features	Watchdog timers on I/O interfaces SMS: Safe Maximum Speed RSP: Remanent Safe Position Consider FSoE (Failsafe over etherCAT) support	O	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.8	Safety level (PL/SIL, see ref)	PLe (IEC 62061): Performance level e SIL3 (ISO 13849): Safety integrity level 3 see following references: robotics.org abb.com	O	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.9	Machine directive	2006/42/EC	R	I	Table	3.1, 4.1, 6.2, 6.3, 6.4
rq-D2.3-S.10	Building block safety	Building blocks shall state the standards adhered to, or specify the safety features facilitated by means of the interfaces	R	I	WP2 team	3.1, 4.1, 6.2, 6.3, 6.4

2.4.3.2 Electrical safety requirements

This section requires detailing to determine the standards we wish to adhere to. Generic interfaces should facilitate adhering to standard and BBs should state the standard adhered to.

Table 19: Electrical safety requirements

ID	Aspect	Requirement	Type	Validation	Source	Task
rq-D2.3-E.1	Electrical safety	IEC 60601 (medical equipment, depending on application) IEC 60204-1 (electrical safety of machines) 2006/42/EC (Machine directive) 2014/35/EU (low-voltage directive)	O (depending on application)	I	Table	3.1, 6.3
rq-D2.3-E.2	IP Class	IEC 60529, class IP20	O	I	Table	3.1, 6.3
rq-D2.3-E.3	Component electrical safety	Electrical safety standards adhered to shall be stated for all I-MECH platform components	R	I	WP2 team	3.1, 6.3

2.4.3.3 Electromagnetic compatibility and emission requirements

This section requires significant further elaboration: e.g. which directives do we wish to adhere to? Probably at least the ones to obtain CE approval. Besides, what must be done to ensure EMF does not harm our own system, e.g. from PWM to sensors?

Table 20: Electromagnetic compatibility and emission requirements

ID	Aspect	Requirement	Type	Validation	Source	Task
rq-D2.3-E.4	Electromagnetic compatibility (EMC)	2014/30/EU (EMC directive) IEC/EN 55011/61326	R	T	Table	3.1, 6.3
rq-D2.3-E.5	Electromagnetic fields (EMF)	2013/35/EU (EMC health and safety)	R	T	Table	3.1, 6.3
rq-D2.3-E.6	Radio frequency emissions	2014/53/EU (Radio Equipment Directive) IEC 61000-6-2 EMF immunity (industrial) IEC 61000-6-4 EMF emission (industrial)	R	T	Table	3.1, 6.3
rq-D2.3-E.7	Platform (electrical) design	Cabling requirements Shielding requirements (Active) filtering requirements Active EMI noise cancelling concepts Noise immunity Ground loops Bandwidth-limited inputs Input signal filtering (e.g. could be digital)				3.1, 6.3
rq-D2.3-E.8	Component EMC specifications	EMC standards adhered to shall be stated for all I-MECH platform components	R	I	WP2 team	3.1, 6.3

2.5 Storage, Transportability, Installation & Disposal

- RoHS compliance
- Waste Electrical and Electronic Equipment (WEEE) Compliance

Acknowledgement

This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737453. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program and Netherlands, Czech Republic, Latvia, Spain, Greece, Portugal, Belgium, Italy, France, Ireland

1. IEEE. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. In: ieee.org [Internet]. [cited 27 Nov 2017]. Available: https://standards.ieee.org/findstds/standard/1588-2008.html