Ref. Ares(2018)2198269 - 25/04/2018

I-MECH Smart Mechatronic Solutions

Doc ID	18040901R07
Doc Creation Date	10JAN18
Doc Revision	Revision 7
Doc Revision Date	06APR18
Doc Status	FINAL

Workpackage	Deliverable ID
WP5, System Behaviour Layer Design and Interfaces	**D 5.1 System behavior layer integration and connectivity requirements and specification (first iteration)**

Summary

This deliverable compiles the initial requirements and specifications for the system behavior layer interfaces used in the Pilots, Demonstrators and Use-Cases. The focus for Layer 3, the system behaviour layer, for I-MECH is on the interfaces between sensors and actuators as part of the instrumentation layer and the motion control layer.

The requirements outlined show a desire for a comprehensive, platform independent interface such as OPC UA, capable of connecting hardware and software from multiple vendors in order to facilitate seamless information communication in a functional, and secure way.

Real time condition monitoring requires a standardised digital interface for the communication of data between all layers of the system. This real time platform is required to have the functionality to extrapolate or predict the condition and performance of the system for maintenance purposes. It is also required to create a condition history with the data set so that the real time information can be logged to chart past trends.

Control layer performance and self-commissioning require significant data transfer between Layer 2 and Layer 3 with the facility to log and monitor identification, auto tuning and validation steps as well as control layer performance. It is also required for the self-commissioning system have the capability to set threshold alarms to notify deviation from expected norms.

The requirements featured will be further processed and finalized in follow up deliverable D5.2 and will contribute to the development of BB3 Condition Monitoring and BB6 Control Layer Self Commissioning. Common requirement on deliverables describing building blocks is that they shall provide a generic approach that could be then integrated into the pilot/demo/use-case scenarios in WP6 and WP7.

There is no activity from Pilot 4 in this deliverable because Correa decided to skip activities in WP5 and to move their effort to other WPs. This shift of activities is actually in the process of preparing an amendment which will be soon sent for the approval. Pilot 4 does find it useful to use results of BB3 and BB6 which are mainly prepared in WP5.

This report builds on the initial system architecture developed in D2.3 [Kampschreur et al., 2018] and forms part of the revision (D2.4).

Author Séamus Hickey (VIS)

Keywords
acoustic emission, auto-tuning, cloud, condition monitoring, diagnostics, feedback, feedforward, interfaces, motion control, OPC UA, performance monitoring, predictive maintenance, real-time, SCADA, self-commissioning, stability, system behaviour layer, temperature diagnostics, vibro-diagnostic, wireless sensor

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

Table of contents

Document Revision History

Revision	Status	Date	Author	Description of changes	IAL ID / Review ID
R01	Prewriting	15-DEC-17	Seamus Hickey	Preparation of document structure	
R02	Draft	13-JAN-18	Seamus Hickey	Initial contributions from partners	
R03	Draft 2	12-FEB-18	Seamus Hickey	Integration of second iteration inputs from partners	
R04	Revision	01-MAR-18	Petr Blaha	Numbering of requirements, sorting them in the tables	
R05	Revision 2	19-MAR-18	All contributors	Adding type and way of validation of requirements, linking with task numbers	
R06	Internal review ready	03-APR-18	Seamus Hickey	Proofreading	
R07	Final	24-APR-18	Petr Blaha	Integration of remarks from internal review	

Contributors

Revision	Affiliation	Contributor	Description of work
R01	J&J	Seamus Hickey	preparation of document structure
R02-R05	J&J	Seamus Hickey	chapter 1, 2, 3, 4.1
	BUT	Petr Blaha, Zdenek Havranek	chapter 4.2.1, 2.1 (inspiration taken from D2.3)
	Gefran UniBs	Davide Colombo Antonio Visioli	chapter 4.2.2, 4.2.3
	Sioux CCM	Thomas Lembrechts	chapter 5.1
	Nexperia	Gijs van der Veen	chapter 5.2
	IMA	Giacomo Collepalumbo	chapter 5.3
	Philips	Harry Mansvelt	chapter 5.4
	J&J	Seamus Hickey	chapter 6.1
	ECS	Gabriel Ribeiro	chapter 6.2
	Gefran	Davide Colombo	chapter 7.2
	Fagor	Carlos Yurre	chapter 7.3
R06	J&J	Seamus Hickey	chapter 8, formatting the document
	BUT	Petr Blaha	formatting and reviewing the document

Document control

		Prewriting	Draft	Draft 2	Revision	Revision 2	Internal review ready	Integra-tion of remarks
	Status							
	Revision	01	02	03	04	05	06	07
Reviewer Name	**Role**	**Selection**						
Arend-Jan Beltman	Coordinator	X						

Petr Blaha (BUT)	Work package 5 leader	X	X	X	X	X	X	X
Harry Mansvelt (Philips)	Reviewer 1						X	
Davide Colombo (Gefran)	Reviewer 2						X	
(BUT)	Reviewer 3 (optional)							

File Locations

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
D5.1R	Partner Zone > Project Breakdown > WP5(BUT) > D5.1 System ... > Deliverable 5.1 report.docx	26-MAR-2018
D5.1TT	Partner Zone > Project Breakdown > WP5(BUT) > D5.1 System ... > Deliverable 5.1 timetable	26-MAR-2018

Abbreviations & Definitions

Abbreviation	Description
AMPQ	Advanced Message Queuing Protocol
BB	Building Block
CAN	Controller Area Network
CNC	Computer Numerical Control
CPU	Central Processing Unit
DSP	Digital Signal Processor
ERP	Enterprise Resource Planning
GSC	Generic Substrate Carrier
GSM	Global System for Mobile communication
HMI	Human Machine Interface
HTTPS	Hypertext Transfer Protocol for Secure communication
ICP	PCB's piezoelectric sensors with built-in microelectronic amplifiers. (ICP® is a registered trademark of PCB Group, Inc.)
IO	Input Output
ISO	International Organization for Standardization
KPI	Key Performance Indicator
LAN	Local Area Network
MEMS	Micro Electro Mechanical Systems
MES	Manufacturing Execution Systems
MIMO	Multiple Input Multiple Output
MQTT	Message Queuing Telemetry Transport
OPC	Open Platform Communication
OPC UA	Open Platform Communication Unified Architecture
OSI	Open Systems Interconnection
PC	Personal Computer
PDM	Pulse-Density Modulation

PID	Proportional Integral Derivative (controller)
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RMS	Root Mean Square
RTD	Resistance Temperature Detectors
RTOS	Real Time Operating System
SCADA	Supervisory Control And Data Acquisition
SECS/GEM	SEMI Equipment Communications Standard/(Generic Equipment Model
SISO	Single Input Single Output
SOAP	Simple Object Access Protocol
SPI	Serial Peripheral Interface
TDM	Time Division Multiplex
WAN	Wide Area Network
WP	Work Package

About this Document

This deliverable is related to Task 5.1 and describes the initial system behaviour layer connectivity and interface requirements for the I-MECH platform. It carries on from the general requirements gathered in Tasks 2.1, 2.2 and 2.3, which have been reported in the I-MECH requirements table, in Deliverable 2.1 "I-MECH State-of-the-art & Requirements" and in Deliverable 2.3 "Overall requirements on I-MECH reference platform".
This document particularly focuses on system behaviour layer requirements and presents the current industrial control protocols for communications and interfaces for automated systems. Special emphasis is placed on the pilot plant applications of the I-MECH project.
These protocols provide platform independent service-oriented architecture that integrates all the functionality of the individual subsystems and are a key factor in integrating disparate control and instrumentation layers being investigated as part of the ECSEL JU.
This deliverable serves therefore as a reference for the activity of WP5 "Behaviour layer design and interfaces", which aims at developing communications and interface strategies between the industrial management systems and the production layer for mechatronic systems in I-MECH. WP5 provides input for several building blocks of the I-MECH project. In particular, BB3 "Robust condition monitoring and predictive diagnostics" and BB6 "Self-commissioning velocity and position control loops" are directly connected to this deliverable as they deal with operations that span both the control layer and the system behaviour layer. All BBs related to the topics discussed in this report are referenced as a fundamental focus for efforts within I-MECH.

1 Introduction

Work package 5 "System Behavior Layer design and interfaces" is dedicated to tools and building blocks which gather data representing the actual status of mechatronic systems, in particular its control and instrumentation part. It provides tools for condition monitoring and control system performance assessment. More specifically, the developed buildings blocks allow automatic commissioning of motion control loops based on the experimental identification data acquired from the instrumentation layer and advanced diagnostics / condition monitoring of electrical drive systems. Furthermore, it deals with the design and specification of data and interfaces to System Behavior Layer tools, namely MES, predictive maintenance and system monitoring. It provides tools for combining information gathered from various sensors (information fusion), pre-processing, analysis and providing proper KPIs to the MES/ERP layer for final decision making.

1.1 Requirement coding scheme

The coding scheme for requirements was retaken from D2.3 [Kampschreur et al., 2018] . Each requirement ID is prefixed with rq- (for requirement), the deliverable ID (in this case D5.1) and the abbreviation of the domain:

- rq-D5.1-XXX: XXX abbreviation of the domain
- rq-D5.1-BBx: building block x
- rq-D5.1-Dx: deliverable x
- rq-D5.1-UCx: use case x

We thus joined the initiative proposed in D2.3 [Kampschreur et al., 2018] to 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 System behaviour layer

Layer 3 (System Behavior Layer), defines a system behavior in terms of the desired motion trajectory. It addresses the fundamental demands which originate from the management layers of production systems. In addition, functionality such as user interaction, sequence and/or exception management can also be found in Layer 3.

3 Types of industrial control system protocols

An industrial communication network is the backbone for any automation system architecture as it provides a powerful means of data exchange, data controllability, and flexibility to connect various devices. The use of proprietary digital communication networks in industries over the past decade has led to improved end-to-end digital signal accuracy and integrity.

These networks, which can be either LAN (Local Area Network, which is used in a limited area) or WAN (Wide Area Network which is used as global system) enabled to communicate vast amounts of data using a limited number of channels. Industrial networking has also led to the implementation of various communication protocols between digital controllers, field devices, various automation related software tools and also to external systems.

As the industrial automation systems become more complex and large with more automation devices on the control floor today, the trend is toward Open Systems Interconnection (OSI) standards that permit to interconnect and communicate any pair of automation devices reliably irrespective of the manufacturer.

With the advancements in digital technology, fieldbus technology is now ruling the automation field as it provides multidrop communication facility that results in cost effective and cable saving communication. The following is an overview of some open industrial control protocols that play a significant role in today’s industrial control systems.

3.1 MTConnect

MTConnect [MTConnect, 2018] is a protocol designed for the exchange of data between shop floor equipment and software applications used for monitoring and data analysis. MTConnect is referred to as a read-only standard, meaning that it only defines the extraction (reading) of data from control devices, not the writing of data to a control device. Freely available, open standards are used for all aspects of MTConnect.

MTConnect takes an incremental approach to defining the requirements for manufacturing device communications. It does not define every possible piece of data an application can collect from a manufacturing device, but works forward from business and research objectives to define the required elements to meet those needs. The standard catalogues important components and data items for metal cutting devices. MTConnect provides an extensible XML schema to allow implementers to add custom data to meet their specific needs, while providing as much commonality as possible.

3.2 OPC

OPC, or Open Platform Communications, is a standard for the secure and reliable exchange of data in the industrial automation space and in other industries. The platform enables the transfer of data between devices of various vendors and between the multiple layers of industrial control systems. The standards themselves were developed in a collaborative way by industry vendors, end-users and software developers.

The standard defines the interfaces between OPC clients and servers. The OPC client makes requests of the OPC server which fulfills the request by providing information to the client in a standardised manner. Examples of these standard interfaces in an industrial environment are the data communications generated between the PLC and HMI, DCS and SCADA.

The OPC specifications define access to real time information, alarm and event monitoring as well as historical data. These are critical aspects of the I-MECH project, the utilisation of real-time information to extrapolate and predict events is part of the condition monitoring and predictive maintenance topics (BB3).

3.3 OPC UA

OPC Unified Architecture (UA) [OPC UA, 2018] is a platform independent service-oriented architecture that integrates the functionality of individual OPC specifications into one expandable framework. As part of I-MECH there is an ambition to utilise OPC UA protocols for the data communication between layers, with a particular focus on sensors and actuator data (see Figure 1).

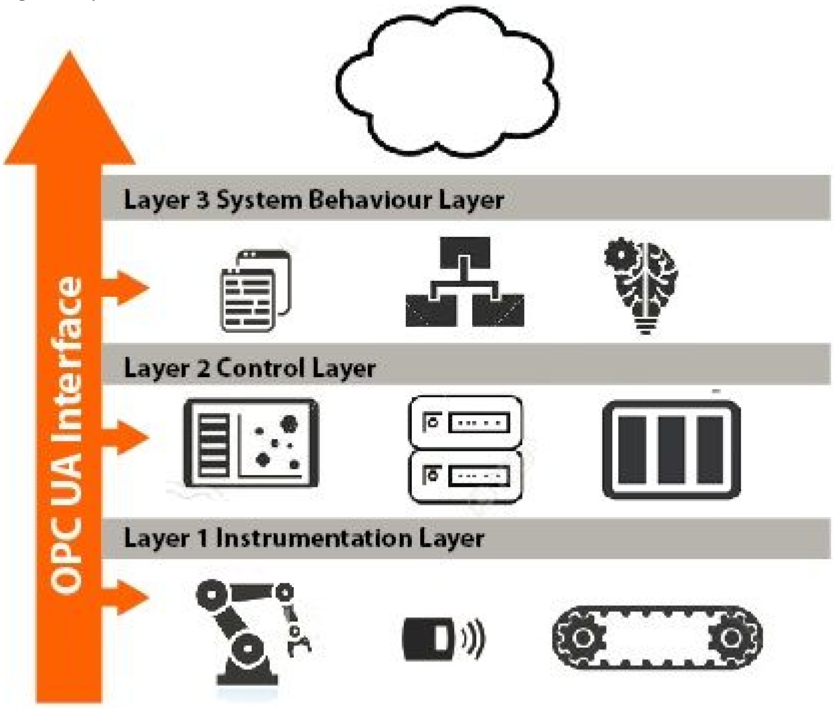


Figure 1 OPC UA interface for I-MECH

OPC UA is constructed on the design specification goals of:

- Functional equivalence - where all OPC Classic specifications are mapped to a unified architecture.

- Platform independence - agnostic to device vendor and suitable for multilayer interfacing. It functions on hardware platforms from traditional PC hardware and cloud-based servers, to PLCs and microcontrollers. It has the added benefit of functioning on multiple operating systems including Windows, Linux, Apple OSX and Android which was previously unattainable on OPC. OPC UA provides the infrastructure for interoperability across the enterprise, from machine-to-machine, machine-to-enterprise and in-between.
- Security - the ability to encrypt information, communications are transmitted with between 128 and 256 bit encryption. The user can also provide authentication on request and all user and process interactions can be audited via an actions log for traceability.
- Expandable - the ability to add new features while not altering existing applications. The idea of OPC UA is to future proof the industrial communications architecture by enabling the integration of newly developed protocols, security algorithms, encoding standards, or application-services.
- Comprehensive information modeling - to be able to define complex information and systems while maintaining standard protocols for communication between complex process steps. The OPC UA information modeling framework turns data into information with data-types and structures defined within the profiles used.

Some of the additional capabilities of OPC UA compared to OPC include the ability to discover OPC servers on a network or on local PC's, enabling the expansion of control systems to include any compatible server on the network. The stratified data structure of OPC UA uses files and folders to allow OPC clients discover structures regardless of complexity, it then can monitor the data and report deviations from specified client criteria, important for condition monitoring purposes within I-MECH and BB3. There is also the capability to read/write information sets based on predetermined access permissions providing flexibility for the design of interfaces and allowing on-demand access to specified process points.

4 Specifications for behaviour layer interface and connectivity

This section summarizes the system behavior layer requirements specific to the pilot, demonstrator and use case plants and will provide the description of general functional blocks, software components, interactions with the motion control layer as well as its interconnection with lower instrumentation level and higher motion planning level. The requirements are divided into functional and non-functional requirements to cover both the motion system behaviour and operation respectively.

4.1 Functional requirements

Functional requirements define specific behavior or functions of the system, they describe what a system is supposed to accomplish. The functionality of the behaviour layer in the I-MECH context focuses on the interfaces between the Layer 2 control layer and Layer 3 system behaviour layer and not the hardware associated with information communication and storage. The following subsections detail the communication and interface, data processing and management, and software requirements for the I-MECH system.

4.1.1 Communication and interface requirements

ID	Requirement	Type	Validation	Source	Task
	Communication and interface requirements				
rq-D5.1-IC.C1	Multi client interface with parallel access for scripting/gui	R	T	WP5	5.2

rq-D5.1-IC.C2	OPC UA	R	T	WP5	5.2
rq-D5.1-IC.C3	Scripting	R	T	WP5	5.2
rq-D5.1-IC.C4	ISA 95 standard for automated interface between enterprise and control systems	R	T	WP5	5.2
rq-D5.1-IC.C5	Ethernet IP/EtherCAT - Have enough bandwidth in case of condition monitoring data logging at upper layer, e.g. 500 kbit/s up to 20 Mbit/s.	R	T	WP5	5.2
rq-D5.1-IC.C6	OPC UA Wireless sensors connection to PLC or upwards to Level 3 or above wireless sensor data transmission to PLC	R	T	WP5	5.2
rq-D5.1-IC.C7	Compatibility with communications protocols and wireless frequency (x Hz)	R	T	WP5	5.2

4.1.2 Data processing and management requirements

ID	Requirement	Type	Validation	Source	Task
	Data processing and management requirements				
rq-D5.1-IC.DP1	Data logging - Enough bandwidth and storage capacity to be able to transfer and store long term (historical) diagnostic data.	R	T	WP5	5.2
rq-D5.1-IC.DP2	Real time access to parameters in the control layer	R	T	WP5	5.2
rq-D5.1-IC.DP3	Process data of raw data outside the condition monitoring measurement module - High performance requirements to be able to do signal pre-processing, data reduction, features extraction and trend/history analysis.	R	T	WP5	5.2

4.1.3 Software requirements

ID	Requirement	Type	Validation	Source	Task
	Software requirements				
rq-D5.1-IC.SW1	Compatibility with MATLAB Simulink	R	T	WP5	5.2
rq-D5.1-IC.SW2	Ability to write custom communication layers to interface with factory automation	R	T	WP5	5.2

4.2 Non-functional requirements

In systems engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than the functional requirements which are specific to system behaviors. The following subsections detail the condition monitoring, novel autonomous instrumentation, control layer self commissioning and control layer performance requirements for the I-MECH system.

4.2.1 Condition monitoring

Condition monitoring in mechatronic systems is an upper level of machine diagnostics which is usually based on regular inspection of small installations by qualified personnel only. Usage of on-line condition monitoring can provide important information about actual machine status with respect to health of its mechanical components, but could also allow the estimation of remaining device lifetime, thus predicting the failure of machine components in the future. This approach can lead to predictive maintenance which allows an effective planning of the maintenance time in advance, saving significant operational cost and minimizing unplanned shutdowns.

International Organization for Standardization deals with condition monitoring programme for machines in following ISO standards. Their aim is to provide general guidelines for condition monitoring to identify and avoid principle cause of failures [ISO 17359] .

- ISO 17359, Condition monitoring and diagnostics of machines — General guidelines
- ISO 13373, Mechanical vibration and shock - Vibration condition monitoring of machines
- ISO 13379, Data interpretation and diagnostic techniques which use information and data related to the condition of a machine.
- ISO 13381, Condition monitoring and diagnostics of machines - Prognostics

BB3 with its condition monitoring should rely on these standards. BB3 will play a key role in Pilots 1, 2, 3 & 5, Demonstrator 1 & 2, as well as Use Cases 1 & 2.

Condition monitoring can be located in Layer 1 and also in Layer 2 of I-MECH platform. In Layer 1, it can act just as an ordinary sensor with a communication channel to some upper layer or it can be also equipped with some signal processing with data reduction. The first option usually requires a communication channel with high data throughput. The second option brings the advantage that the analog signal wiring is very short and therefore immune to noises. It often works as an intelligent slave which is capable of providing the status on request and therefore the requirements on communication bandwidth are much lower. In Layer 2, condition monitoring can be located on different places. Some intermediate results from the current, speed and position loops or path planning can be useful to monitor in Layer 3. Common sampling rates in motor controlled systems are high, usually >10 kHz. This leads to high data throughput requirements.

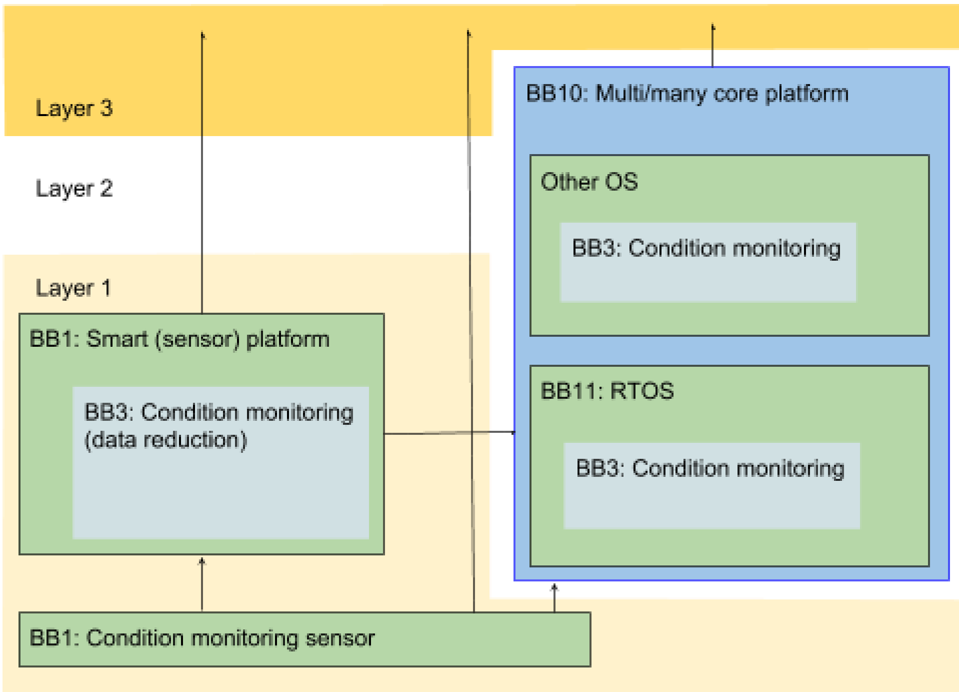


Figure 2 Condition monitoring in I-MECH platform

General requirements on condition monitoring are summarized in following table.

ID	Requirement	Type	Validation	Source	Task
	General expected functionality of the condition monitoring module				
rq-D5.1-CM.G1	Functionality to set thresholds/alarms	R	T	WP5	5.3
rq-D5.1-CM.G2	Possibility to calculate KPIs based on multiple variables & history	R	T	WP5	5.3
rq-D5.1-CM.G3	Functionality to predict/extrapolate	O	T	WP5	5.3
rq-D5.1-CM.G4	Database with standard functions/fingerprints to detect failure of commonly used systems	R	T	WP5	5.3
rq-D5.1-CM.G5	Use of standardized interface to interface with condition monitoring tooling (possibly via a standardized OPC UA information model)	R	T	WP5	5.3
rq-D5.1-CM.G6	Trend analysis on selected features (e.g. RMS value) will be the main procedure for continuous on-line condition monitoring of drive mechanics and analysis of additional features will be considered in specific application cases.	R	T	WP5	5.3
	Condition monitoring of electrical drive mechanics and electronics				
rq-D5.1-CM.D1	Monitoring of mechanical unbalance and shaft misalignment	R	T	WP5	5.3
rq-D5.1-CM.D2	Monitoring of bearings failure or wear (early detection)	O	T	WP5	5.3
rq-D5.1-CM.D3	Monitoring of gearbox teeth wear	O	T	WP5	5.3
rq-D5.1-CM.D4	Monitoring of winding electrical failure	O	T	WP5	5.3
	Required interfaces to sensorics and signal processing modules.				
	Sampling frequency and measurement strategy strongly depends on application and selected features which is plan to be monitored, where early detection using high frequency acoustic emission components need high sampling frequency (>500 kHz). For other vibrodiagnostic measurements lower sampling frequency is sufficient (~5 kHz) and for temperature measurements even lower sampling is expected (~1Hz). In case of signal processing and analysis in dedicated measurement module, data rate can be highly reduced to transfer only warnings, failures or low frequency diagnostic information (e.g. vibration severity ~ effective values).				
rq-D5.1-CM.S1	Low level digital interface for modern vibration and acoustic sensors (TDM, PDM)	R	T	WP5	5.3, 5.2
rq-D5.1-CM.S2	Digital interface for modern vibration and acoustic sensors (SPI)	R	T	WP5	5.3, 5.2
rq-D5.1-CM.S3	Low level analog interface for classical vibration and acoustic sensors (direct voltage, ICP)	R	T	WP5	5.3, 5.2

rq-D5.1-CM.S4	High bandwidth analog interface for acoustic (ultrasonic) emission sensors	O	T	WP5	5.3
rq-D5.1-CM.S5	Digital interface for digital incremental encoders	R	T	WP5	5.3, 5.2
rq-D5.1-CM.S6	Analog interfaces for analog resolver	R	T	WP5	5.3, 5.2
rq-D5.1-CM.S7	Digital interface to smart sensors with integrated electronics	R	T	WP5	5.3, 5.2
rq-D5.1-CM.S8	High level communication interface (process bus) to signal processing and analysis module	R	T	WP5	5.3, 5.2
	Appropriate measurements and analysis methods needed for drive mechanics condition monitoring				
rq-D5.1-CM.M1	*Measurement of speed and position of rotation -* General requirement for vibration diagnostics and mechanical system condition monitoring. Rotational speed and shaft position information	R	T	WP5	5.3
rq-D5.1-CM.M2	*Vibration severity measurement – overall fault detection* One low noise acceleration sensor (MEMS or miniature piezo) with 3-axis, sensor self-noise less than 40 ug/Hz and resonant frequency higher than 5 kHz, acceleration range up to 20 g, sampling rate of at least 4 kS/s, measurement in extended ISO bandwidth (1/10 Hz up to 1 kHz), analog or preferably digital interface to the signal pre-processing and effective value calculation.	R	T	WP5	5.3, 3.2
rq-D5.1-CM.M3	*Vibration diagnostics - unbalance* One low noise acceleration sensor (MEMS or miniature piezo) with 3-axis per bearing (typ. two sensors needed), self-noise less than 40 ug/Hz, sampling rate of at least 4 kS/s, frequency range higher than 10 Hz, depending on shaft speed, analog or preferably digital interface to the signal pre-processing and following analysis. Filtering on rotational frequency with analog or digital tracking filter with quality factor better than 20. Measurement of rotation rate and shaft position with one speed sensor.	R	T	WP5	5.3, 3.2
rq-D5.1-CM.M4	*Vibration analysis – gears and gearboxes faults* Acceleration sensor (MEMS or piezo), one sensor per gearbox shaft, sampling rate of at least 4 kS/s, measurement bandwidth >1 kHz, depending on gear type. Analog or preferably digital interface to the signal pre-processing and following features extraction.	O	T	WP5	5.3, 3.2
rq-D5.1-CM.M5	*Vibration diagnostics of bearings/lubrication - early fault detection/prediction* High frequency accelerometer, AE sensor (ultrasonic emission), one sensor per bearing, one sensor per system (with advanced DSP), resonance sensors or broadband detection sensors, measurement bandwidth up to 1 MHz (depending on bearing type), typical	O	T	WP5	5.3, 3.2, 3.3

	sampling rate higher than 1 MS/s. High performance preprocessing is needed.				
rq-D5.1-CM.M6	*Acoustic diagnostics - noise emission, squeak and rattle* One MEMS microphone per system (with advanced DSP), measurement bandwidth up to 20 kHz, sampling rate of 50 kHz, digital interface with SPI/PDM.	O	T	WP5	5.3, 3.2, 3.3
rq-D5.1-CM.M7	*Temperature diagnostics of power electronics* One or more temperature sensors (RTD, semiconductor, thermocouple), virtualization or model based temperature distribution approach in case of one sensor, sampling rate of 10 S/s model based temperature distribution.	R	T	WP5	5.3, 3.2
rq-D5.1-CM.M8	*Analysis during transition states* The system must allow measurement and analysis in the intermediate transition states of the rotating mechanical system during its normal operation, where the typical transition states are run-up and slow-down. These transition states could be also initiated by diagnostic unit in case of serious suspicion of mechanical failure.	R	T	WP5	5.3
	Condition monitoring could be executed in:				
rq-D5.1-CM.E1	Dedicated condition monitoring module with close connection to sensorics and interfaced to the main control system where time domain data can be easily transferred,	R	T	WP5	5.3, 3.3
rq-D5.1-CM.E2	On main control system with reduced diagnostic data throughput and loading, more difficult connection to diagnostic sensorics in the presence of high electrical noise environment should be designed, possible use of smart sensorics could reduce additional loading of main control system.	R	T	WP5	5.3, 3.7

4.2.2 Control layer self commissioning

Self commissioning often belongs to the Control layer which, in I-MECH platform, is called Layer 2. It could be also be part of Layer 3 but this possibility is out of the scope of the I-MECH project and transforms then to the movement of required data from Layer 2 to Layer 3 and designed controller parameters back to Layer 2. As it can be seen in Figure 2 below, self commissioning of velocity and position control loop is a component of building block 6 (BB6). It can be either implemented as a bare metal application or as a routine of the real time operating system. The way of implementation is usually given by realization of the inverter controller.

Self commissioning has become a part of modern inverter controllers [Kania et al., 2011]. It is composed of several steps, namely the process of identification [Pacas et al., 2010], auto tuning and validation. Identification experiment is usually limited with the range of experimental input signals and with the band of operation of output signals. These parameters depend on actual application and therefore they must be made tunable. Self commissioning is often supported either with scripting language or with graphical user interface which enables the influence of its progress. Based on results from identification, controller parameters can be tuned. Feedback speed and position controllers must be tuned with care since they can destabilize the closed loop system. Additional compensation strategies which are working in feedforward can be tuned as well.

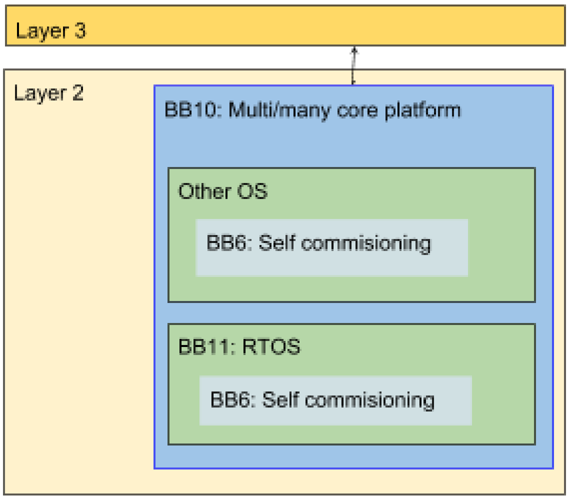


Figure 2 Self commissioning in I-MECH platform

General requirements on self commissioning are summarized in the following table.

ID	Requirement	Type	Validation	Source	Task
	Tracing				
rq-D5.1-SC.T1	Functionality to set thresholds/alarms	O	T	WP5/6	5.4
rq-D5.1-SC.T2	Live tracing in time and frequency domain (bode & nyquist) of all parameters within control system (for manual commissioning and troubleshooting)	O	T	WP5/6	5.4
rq-D5.1-SC.T3	Triggering functionality for traces (like real scope)	O	T	WP5/6	5.4
rq-D5.1-SC.T4	Autotuning approach should allow the use of “proprietary” instrument for tracing (like scope) that are generally available in commercial drives. The tuning method through “standard” language will define the approach for identification steps and signals that can be used to trigger scope system. The tracing instrument should provide the data collected through open-language (like txt, xml)	R	T	WP5	5.4
	Auto tuning / Control architecture				
rq-D5.1-SC.AT1	Auto tuning application/function/library/interface that works with a wide range of control algorithms, both feedforward and feedback (not only PID). E.g. I-MECH control algorithms of BB7,8,9 should specify their tunable parameters and tuning strategy to the auto tuning application/function. The auto-tuning function should provide generic optimization functions, parameter estimation functions, noise generation functions,... <u>Task 5.4 will be used as collector for auto-tuning approaches that have to be developed on BB7,BB8,BB9 (Task 4.4,4.5,4.6)</u>	R	T	WP5/4/3	5.4 4.4 4.5 4.6 3.6
	System limits				

rq-D5.1-SC.SL1	Auto tuning functionality should respect configurable torque, position & velocity limits. These limits are dependent on the I-Mech application, and it should be defined at the beginning of the identification steps.	R	T	WP5	5.4
rq-D5.1-SC.SL2	Limits should be configurable for all axes. Generally approach will be based on SISO system, but compatible also for MIMO system.	R	T	WP5	5.4
rq-D5.1-SC.SL3	A sequence should be specifiable (optionally via scripting interface) to get system in right configuration for auto-tuning.	R	T	WP5	5.4
	System identification				
rq-D5.1-SC.SI1	Tools should be suited for system identification & parameter estimation purposes Expected contributions from Task 4.3	R	T	WP5/4	5.4 4.3
	Feedforward				
rq-D5.1-SC.FF1	The self commissioning function should be able to fit basic model to identified system and be able to commission feedforward controllers based on fitted model for: -friction compensation -mass compensation -spring compensation -gravity compensation -passive vibration compensation (input shaping) Expected contributions from Task 4.3 and 4.4	R	T	WP5/4	5.4 4.3 4.4
	Scripting				
rq-D5.1-SC.SC1	It should be possible to use self-commissioning functionality in combination with scripting (compatibility with standard language like MATLAB, Python).	R	T	WP5/6	5.4
	Validation				
rq-D5.1-SC.V1	Validation signals and approach should be defined in order to check the performance of the autotuning parameters	R	T	WP5	5.4

4.2.3 Control layer performance

The placement of control layer performance monitoring in a view of the I-MECH concept is similar to self commissioning. It also belongs to the Control layer or Layer 2. It serves as a data reduction (data pre processing) algorithm which passes to the upper level only some indicators or classifiers.

ID	Requirement	Type	Validation	Source	Task
	Stability				

rq-D5.1-CLP.S1	Monitoring tooling should have functionality to calculate and monitor stability and robustness margins of feedback loops as well as self learning algorithms from BB9	R	T	WP5	5.3 4.5
	Performance				
rq-D5.1-CLP.P1	Monitoring tooling should have functionality to monitor: -settling time -overshoot -following error as a function of time (e.g. deterioration over time)	R	T	WP5	5.3 5.4

5 I-MECH Pilots – Requirements specification

5.1 Industrial printing - Generic substrate carrier (GSC) (Pilot 1- Sioux CCM)

The requirements listed below are a selection of the pilot 1 requirements listed in D7.1 applicable to work package 5 activities. A description of the pilot system and the context of the listed requirements can also be found in D7.1. D7.1 is considered as the leading document for the latest revision of all requirements and information related to pilot 1.

ID	Requirement	Type	Validation	Source	Task
	Communication and interface requirements				
rq-D5.1-P1-L3.1	The platform shall support Ethernet communication (for communication between layer 3 and rest of the world)	R	I	D7.1	5.2
rq-D5.1-P1-L3.2	Multi-client interface with parallel access between layer 3 and rest of the world	R	I	D7.1	5.2
rq-D5.1-P1-L3.3	Automation programming language. The controller architecture should provide possibility for scripting automated sequences in an interpreted high-level programming language like Python, which allows for easily changing the scripts.	R	I	D7.1	5.2
rq-D5.1-P1-L3.4	OPC UA server: ● Easy interface to configure server ● Preferably with automatic code generation to make variables and parameters used in underlying control/instrumentation layer available on the server and create methods/alarms/…. automatically ● With encryption/user rights management	O	I	D7.1	5.2

<table>
<tr><td></td><td>Data processing and management requirements</td><td></td><td></td><td></td><td></td></tr>
<tr><td>rq-D5.1-P1-BB3.1</td><td>Automatic logging (tracing) of long term effects at low frequency sampling (1Hz) which are saved on host PC. Examples of long term effect signals: forces, temperatures, pump speeds, number of revolutions</td><td>R</td><td>I</td><td>D7.1</td><td>5.2/
5.3</td></tr>
<tr><td>rq-D5.1-P1-BB3.2</td><td>Automatically save relevant trace data the of last 10 seconds when a system error occurs (e.g. maximum tracking error exceeded). Examples of relevant trace signals: setpoints/tracking errors/forces.</td><td>R</td><td>I</td><td>D7.1</td><td>5.2/
5.3</td></tr>
<tr><td>rq-D5.1-P1-BB3.3</td><td>Remote diagnostics & logging. Controller architecture should provide possibility for remote access/control.</td><td>R</td><td>I</td><td>D7.1</td><td>5.2/
5.3</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Condition monitoring</td><td></td><td></td><td></td><td></td></tr>
<tr><td>rq-D5.1-P1-BB3.10</td><td>Detect deviations from expected behaviour, preferably using sensor and controller signals already available in the system, to detect pollution, wear, trends.
Typical GSC KPI’s to monitor:
<table>
<tr><th>KPI to monitor</th><th>Issue to detect</th></tr>
<tr><td>Average print overlay error per second (based on position error)</td><td>Decrease in quality of printer output</td></tr>
<tr><td>Motor torque vs VSU pressure</td><td>Ink spill / contamination below belt

Sliding bearing of belt damaged / moved</td></tr>
<tr><td>VSU motor speed vs VSU pressure</td><td>Obstruction/leakage of vacuum system</td></tr>
<tr><td>Motor temperature vs motor torque</td><td>Failure of motor cooling</td></tr>
<tr><td>Total number of revolutions</td><td>Time for periodic maintenance</td></tr>
<tr><td>Set/Reset forces of reluctance actuators</td><td>Sensor/actuator failure

System failure/modification</td></tr>
<tr><td>Standard deviation of X, Y, Rz controller output</td><td>Instability of controllers</td></tr>
<tr><td>Peak of X, Y, Rz controller output</td><td>Infeed of new products on belt</td></tr>
<tr><td>Periodic errors of belt edge sensors</td><td>Contamination of belt edge</td></tr>
<tr><td>Encoder signal quality</td><td>Errors in encoder alignment</td></tr>
</table></td><td>R</td><td>T</td><td>D7.1</td><td>5.2/
5.3</td></tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr><td></td><td>Contamination of encoder</td></tr>
<tr><td>Average Rz torque</td><td>Damaged actuator

Obstruction of sensors due to expansion of sliding bearing of belt</td></tr>
<tr><td>Controller tracking error</td><td>Deterioration of performance</td></tr>
</table>
</td>
<td></td><td></td><td></td><td></td>
</tr>
<tr>
<td>rq-D5.1-P1-BB3.11</td>
<td>System self-tests at startup / check-up on demand
This is an extensive checkup, a sequence that checks relevant dynamics, sensors, actuators, accuracy etc. This function is to be used when there is a suspicion that the system is not working correctly.
Controller architecture should provide possibility for scripting this sequence in an interpreted high-level programming language like Python, which allows for easily changing the scripts.
Typical GSC self-tests:
<table>
<tr><th>Test</th><th>Issue to detect</th></tr>
<tr><td>Verify gap sensor output at rest</td><td>Gap sensors broken?</td></tr>
<tr><td>Verify noise on belt edge sensor at rest</td><td>Belt edge sensor light cover present?</td></tr>
<tr><td>Verify system dynamics (e.g. via system identification and parameter estimation)</td><td>Mechanical changes to system?</td></tr>
<tr><td>Measure stiffness AMSR leafsprings</td><td>Gap sensor/leafspring broken?</td></tr>
</table>

Generic self-tests:
<table>
<tr><th>Test</th><th>Issue to detect</th></tr>
<tr><td>Verify stability and robustness margins of feedback loops and learning feedforward algorithms</td><td>System component failure

System modifications

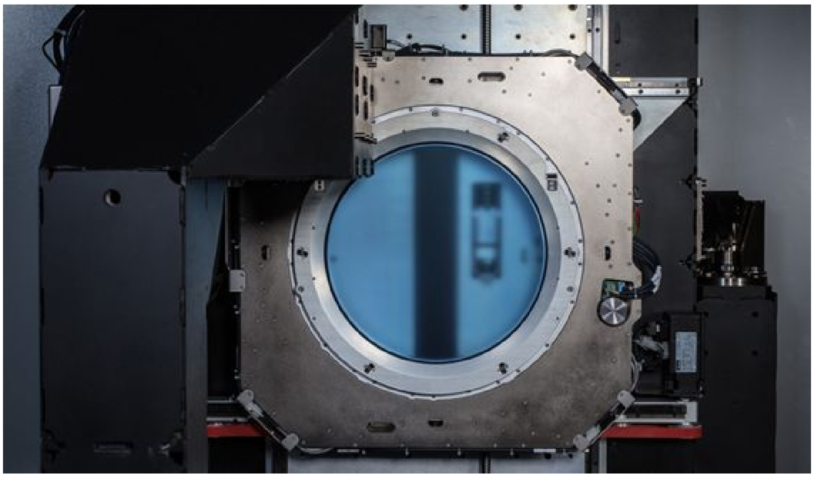
System environment modifications</td></tr>
</table>
</td>
<td>R</td><td>T</td><td>D7.1</td><td>5.2/ 5.3</td>
</tr>
<tr>
<td>rq-D5.1-P1-BB3.12</td>
<td>BBs (e.g. BB5) should have a self-test and be able to report their status.</td>
<td>R</td><td>I</td><td>D7.1</td><td>5.2/ 5.3</td>
</tr>
<tr>
<td>rq-D5.1-P1-BB3.13</td>
<td>Report deviations & warnings to operator such that operator is warned in time to perform maintenance</td>
<td>R</td><td>I</td><td>D7.1</td><td>5.2/ 5.3</td>
</tr>
</table>

rq-D5.1-P1-BB3.14	A 'toolbox'[1] with functions for automated performance assessment of the system shall be available containing at least the following functionality: · Measurement of frequency responses (including cross-couplings for MIMO systems) o It shall be possible to compare identified plant models w.r.t. to reference plant models to identify unexpected deviations. · Identification of (changes in) feedforward terms, including friction · Assessment of: o Bandwidth o Sensitivity o Stability margins o Settle time, rise time, overshoot o Noise/disturbance levels · Functions to easily design an IO-test/self-test Functions shall be accessible via a scripting interface [1] Building block 3 and 6 should preferably use the same toolbox	R	I	D7.1	5.2/ 5.3
rq-D5.1-P1-BB3.15	Functionality to set thresholds/alarms[1] [1] Prevention of false errors due to changes in setpoints, etc.	R	I	D7.1	5.2/ 5.3
rq-D5.1-P1-BB3.16	Possibility to calculate KPIs based on multiple variables & history	R	I	D7.1	5.2/ 5.3
rq-D5.1-P1-BB3.17	Functionality to predict/extrapolate KPIs	R	T	D7.1	5.2/ 5.3
rq-D5.1-P1-BB3.18	Database with standard functions/fingerprints to detect failure of commonly used systems	O	T	D7.1	5.2/ 5.3
	Control layer self-commissioning				
rq-D5.1-P1-BB6.1	Engineering Programming Environment[1] This environment should provide possibility to create and run (interpret) text editable recipes for scripting automated sequences. Preferably in an engineering programming language like Python or MATLAB. This allows for easy creation and modification of sequence scripts by engineers without having to re-compile any software.	R	I	D7.1	5.4

	[1] E.g. AxChange python scripting interface				
rq-D5.1-P1-BB6.2	Feedback control loop commissioning for the following control loops: ● 3x Position control ● 2x Velocity control ● 1x Pressure control (very slow system dynamics) Fully automated identification and commissioning of at least the following feedback structure parameters: PID + 3 generic purpose filters	R	T	D7.1	5.4
rq-D5.1-P1-BB6.3	Feed forward commissioning - Fully automated identification and commissioning of at least the following setpoint feed forward parameters: ● Acceleration ● Static friction ● Viscous friction	R	T	D7.1	5.4
rq-D5.1-P1-BB6.4	Automatic controller commissioning functions/toolbox Application optimization criteria and constraints shall be configurable. ● Application optimization examples: ○ Accuracy ○ Bandwidth ○ Sensitivity… ● Application constraints examples: ○ Hardware limits ○ Position, velocity, torque limits *Wish:* *Should be applicable to a wide range of control algorithms, both (learning) feedforward and feedback (not only PID). The automatic controller commissioning function should therefore provide generic optimization functions, system identification and parameter estimation functions, noise generation functions, etc.*	R	I	D7.1	5.4
rq-D5.1-P1-BB6.11	Roller reluctance actuators / sensors calibration - Fully automated measurement, parameter calculation and verification sequence	R	T	D7.1	5.4
rq-D5.1-P1-BB6.12	Roller encoder eccentricity calibration - Fully automated measurement, parameter calculation and verification sequence	R	T	D7.1	5.4
rq-D5.1-P1-BB6.13	Roller motor position dependent force calibration - Fully automated measurement, parameter calculation and verification sequence	R	T	D7.1	5.4
rq-D5.1-P1-BB6.14	Roller unroundness calibration - Fully automated measurement, parameter calculation and verification sequence	R	T	D7.1	5.4

rq-D5.1-P1-BB6.15	Belt edge calibration - Fully automated measurement, parameter calculation and verification sequence	R	T	D7.1	5.4

5.2 Semiconductor production - 12 inch wafer stage (Pilot 2 – Nexperia)



ID	Requirement	Type	Validation	Source	Task
	Communication and interface requirements				
rq-D5.1-P2.CI1	Multi-client - Multi client interface with parallel access for scripting/gui etc. -> via OPC UA?	R	T	WP5	5.2
	Data processing and management requirements				
rq-D5.1-P2.D1	Logging - Data logging functionality, log resolution should be the same as update rate of control system	R	T	WP5	5.2
rq-D5.1-P2.D2	Scope - Real time access to all parameters in control layer/instrumentation layer with scope functionality with nice GUI	R	T	WP5	5.2
	Software requirements				
rq-D5.1-P2.SW1	Compatible with control layer written/modeled in MATLAB Simulink	R	T	WP5	5.2
rq-D5.1-P2.SW2	Ability to write custom communication layers (e.g. a SECS/GEM interface to factory automation)	R	T	WP5	5.2
	Condition monitoring				
rq-D5.1-P2.CM1	Alarms - Functionality to set thresholds/alarms	R	T	WP5	5.3
rq-D5.1-P2.CM2	KPIs - Possibility to calculate KPIs based on multiple variables & history	R	T	WP5	5.3
rq-D5.1-P2.CM3	Standardized interface - Use of standardized interface to interface with condition monitoring tooling (possibly via a standardized OPC UA information model).	R	T	WP5	5.3
	Control layer self-commissioning				
rq-D5.1-P2.SC1	Tracing - Live tracing in time and frequency domain (bode & nyquist) of all parameters within control system (for manual commissioning and troubleshooting) Triggering functionality for traces (like real scope)	R	T	WP5	5.4

rq-D5.1-P2.SC2	Auto tuning - Auto tuning application/function/library/interface that works with a wide range of control algorithms, both feedforward and feedback (not only PID). E.g. I-MECH control algorithms of BB7,8,9 should specify their tunable parameters and tuning strategy to the auto tuning application/function. The auto-tuning function should provide generic optimization functions, parameter estimation functions, noise generation functions,...	R	T	WP5	5.4
rq-D5.1-P2.SC3	System limits - Auto tuning functionality should respect configurable torque, position & velocity limits. Limits should be configurable for all axes. A sequence should be specifiable (optionally via scripting interface) to get system in right configuration for auto-tuning.	R	T	WP5	5.4
rq-D5.1-P1.SC4	System identification - Tooling should be suited for system identification & parameter estimation purposes	R	T	WP5	5.4
rq-D5.1-P2.SC5	Feedforward - The self commissioning function should be able to fit basic model to identified system and be able to commission feedforward controllers based on fitted model for: ● friction compensation ● mass compensation ● spring compensation ● gravity compensation	R	T	WP5	5.4
rq-D5.1-P2.SC6	Scripting - It should be possible to use self-commissioning functionality in combination with scripting (e.g. using Python).	R	T	WP5	5.4
	Control layer performance				
rq-D5.1-P2.CP1	Stability Monitoring tooling should have functionality to calculate and monitor stability and robustness margins of feedback loops as well as self learning algorithms from BB9	R	T	WP5	5.4
rq-D5.1-P2.CP2	Performance - Monitoring tooling should have functionality to monitor: ● settling time ● overshoot ● following error as a function of time (e.g. deterioration over time)	R	T	WP5	5.4

5.3 High speed packaging - In-line filling & stoppering machine, Tea bag machine (Pilot 3 – IMA)

ID	Requirement	Type	Validation	Source	Task
	Communication and interface requirements				
rq-D5.1-P3.CI1	OPC-UA, OPC-DA, ModbusTCP for HMI communication	R	T	WP5	5.2
rq-D5.1-P3.CI2	Ethercat, Powerlink, SercosIII as fieldbuses for IO and Motion	R	T	WP5	5.2
rq-D5.1-P3.CI3	MQTT, AMPQ for machine to cloud communication	R	T	WP5	5.2
rq-D5.1-P3.CI4	WebServer for online access to relevant variables and parameters	R	T	WP5	5.2
	Data processing and management requirements				
rq-D5.1-P3.D1	Detailed and fine grained logging	R	T	WP5	5.2, 5.3
rq-D5.1-P3.D2	Online access to relevant variables and parameters (e.g. Motion control parametrization and diagnostic)	R	T	WP5	5.3
	Software requirements				
rq-D5.1-P3.SW1	Designed to match Hard Real-Time requirements	R	T	WP5	5.2, 5.3, 5.4
rq-D5.1-P3.SW2	Designed to handle multiprocessor systems with Hard Real-Time requirements in mind	R	T	WP5	5.2, 5.3, 5.4
rq-D5.1-P3.SW3	VxWorks 6.9.x and 7 compatible	R	T	WP5	5.2, 5.3, 5.4

	Control layer performance				
rq-D5.1-P3.CP1	Performance monitoring tool, for monitoring task-level performance (actual cycle time, jitter, response latency…).	R	T	WP5	5.3
rq-D5.1-P3.CP2	50 us minimum scheduling cycle time (not related to minimum Motion Control cycle time, which is 500 us) with 1% jitter.	R	T	WP5	5.3

5.4 Healthcare robotics - Medical manipulator (Pilot 5 - PHI)

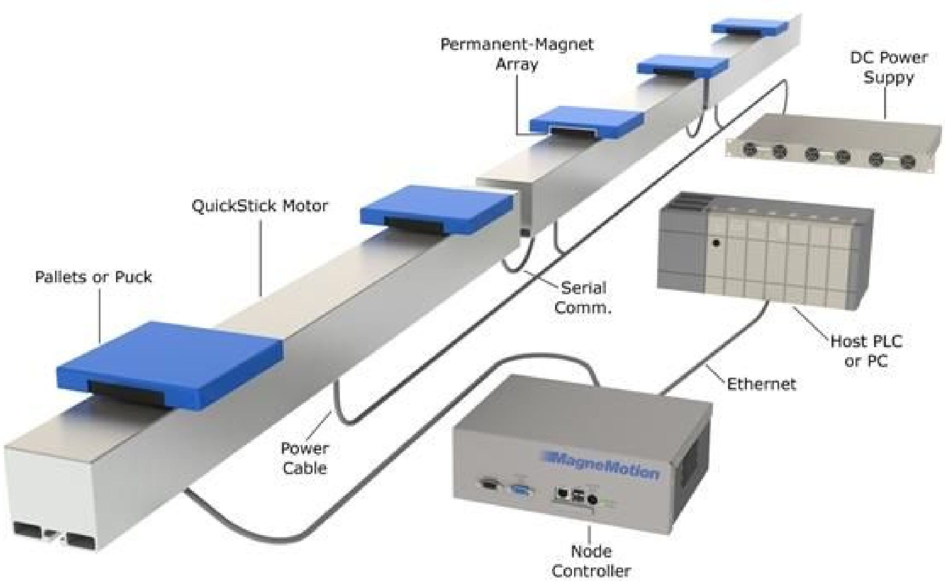
ID	Requirement	Type	Validation	Source	Task
	Communication and interface requirements				
rq-D5.1-P5.CI1	Multi client interface with parallel access for scripting/gui etc.	R	T	WP5	5.2
rq-D5.1-P5.CI2	All output is in MATLAB Simulink code or DS402 protocol messages	R	T	WP5	5.2
	Data processing and management requirements				
rq-D5.1-P5.BB3.1	Logging - Data logging functionality, maximum log resolution is equal to the update rate of the position loop. Triggering functionality for traces.	R	T	WP5	5.3

rq-D5.1-P5.D2	Scope - Real time access to all parameters in control layer/instrumentation layer with scope functionality with functional GUI.	R	T	WP5	5.3
	Software requirements				
rq-D5.1-P5.SW1	Communication between the auto-tuner and the application is in MATLAB or Simulink	R	T	WP5	5.2, 5.4
	Condition monitoring				
rq-D5.1-P5.BB3.2	Functionality to set multiple threshold/alarm levels	R	T	WP5	5.3
rq-D5.1-P5.BB3.3	Generate event when threshold is surpassed	R	T	WP5	5.3
rq-D5.1-P5.BB3.4	Handshake mechanism on event generation and event reset	R	T	WP5	5.3
	Self Commissioning of controller				
rq-D5.1-P5.BB6.1	Frequency analysis (Bode/Nyquist), including coherence output of predefined transfer functions, based on output as provided by rq-D5.1-P5.BB3.1	R	T	WP5	5.4
rq-D5.1-P5.SC1	Auto-tuning works for a predefined feedback control structure (see *partner_zone/project_breakdown/ WP1/Task1.3/Meetings/F2F Alignment 20feb2018/ OutputMeeting/Questionaire Secretaries-BB6.pptx*). Auto-tuning of the current loop (excluding commutation) Feedforward auto-tuning output sets the parameters for effects of viscous friction, coulomb friction, hysteresis, motor efficiency, gravity and inertia. The auto-tuning function must provide optimal parameter settings based on initial performance criteria.	R	T	WP5	5.4
rq-D5.1-P5-L1	Auto-tuning signals are generated and transferred to control loop. Signals include: (swept) sinewave, white noise etc. with filter possibilities Hanning, Hamming and Gaussian and measurement averaging.	R	T	WP5	5.4
rq-D5.1-P5-L2	Generated Auto-tuning signals can be superimposed on application movement commands	R	T	WP5	5.4
rq-D5.1-P5-L3	Signal injection can only be performed on fixed entry points (to be defined)	R	T	WP5	5.4
rq-D5.1-P5.SC2	Auto-tuning output includes optimization of output of various auto-tuning runs under varying geometrical conditions of the application.	R	T	WP5	5.4
rq-D5.1-P5.SC3	It should be possible to use self-commissioning functionality in combination with scripting with MATLAB	R	T	WP5	5.4
	Control Performance				
rq-D5.2-P5.CP1	Monitoring tooling must be present and capable of measuring: ● Position settling time	R	T	WP5	5.3

	● Position overshoot ● Position- and Speed following error ● Motor current and feed forward current based on output from rq-D5.1-P5.BB3.1				
	Safety				
rq-D5.1-P5.S1	Auto tuning functionality must respect configurable torque-, position- & velocity limits. Limits should be configurable for all axes. A sequence should be specifiable (optionally via scripting interface) to get the system in right configuration for auto-tuning.	R	T	WP5	5.4
Rq-D5.1-P5.S2	The auto tuning process can be terminated manually	R	T	WP5	5.4

6 I-MECH Demonstrators – Requirements specification

6.1 Contact lens automated transport layer (Demonstrator 1 – VIS)



ID	Requirement	Type	Validation	Source	Task
	Communication and interface requirements				
rq-D5.1-D1.CI1	Ethernet IP Rockwell automation - Communications as to Rockwell Magnemotion control protocols	R	T	WP5	5.2

rq-D5.1-D1.CI2	OPC UA Wireless sensors connection to PLC or upwards to Level 3 or above Ambition: wireless sensor data transmission to PLC	O	T	WP5	5.2
rq-D5.1-D1.CI3	Wireless sensors - Must be compatible with communications protocols and wireless frequency (x Hz)	R	T	WP5	5.2
	Data processing and management requirements	R	T	WP5	5.2
rq-D5.1-D1.D1	Cloud and Edge cloud - Potential for both as part of I-MECH	R	T	WP5	5.2
rq-D5.1-D1.D2	Mining data from level 2 SCADA system (Rockwell Factory Talk) - connect to internal edge cloud direct to level 2 systems (SCADA)	R	T	WP5	5.2
rq-D5.1-D1.D3	Mining data from level 2 SCADA system (Rockwell Factory Talk) - connect to external cloud to level 2 systems	R	T	WP5	5.2
rq-D5.1-D1.D4	Cyber security requirements are significant for any data leaving J&J networks (current standard PC Duo probably not suitable for cloud system)	R	T	WP5	5.2
	Software requirements				
rq-D5.1-D1.SW1	Rockwell SCADA Factory Talk - Interfaces to PLC, SCADA databases	R	T	WP5	5.2
rq-D5.1-D1.SW2	Ethernet IP - Sensors to PLC. SCADA to cloud potentially	R	T	WP5	5.2
rq-D5.1-D1.SW3	OPC UA - Sensors to PLC (preferred Ethernet IP) SCADA to cloud potentially	O	T	WP5	5.2
rq-D5.1-D1.SW4	Direct I/O (digital, analog) to PLC - To Rocwell AB PLC spec	R	T	WP5	5.2
rq-D5.1-D1.SW5	RFID - Magnemotion carrier identification to PLCX	R	T	WP5	5.2
	Condition monitoring				
rq-D5.1-D1.CM1	Functionality to set multiple threshold/alarms	R	T	WP5	5.3
rq-D5.1-D1.CM1	Predictive capabilities to identify component degradation before an event.	R	T	WP5	5.3
rq-D5.1-D1.CM1	Condition history for product traceability	R	T	WP5	5.3

6.2 Injection mould tool (Demonstrator 2 – ECS)

ID	Requirement	Type	Validation	Source	Task
	Communication and interface requirements				
rq-D5.1-D2.CI1	Wireless communication	R	T	WP5	5.2
rq-D5.1-D2.CI2	GUI to visualizing status	O	T	WP5	5.2
rq-D5.1-D2.CI3	Sending GSM messages (probably)	O	T	WP5	5.2
	Data processing and management requirements				
rq-D5.1-D2.D1	Real-time viewing	R	T	WP5	5.3
rq-D5.1-D2.D2	Data acquisition frequency dependent on the application.	R	T	WP5	5.3
	Condition monitoring				
rq-D5.1-D2.CM1	Alarms – On deviation from normal parameters	R	T	WP5	5.3
rq-D5.1-D2.CM2	Data recording to create a database (probably)	R	T	WP5	5.3

7 I-MECH Use Cases – Requirements specification

7.1 Power electronic for hoist and crane sector (Use case 1.1 – GEF)

ID	Requirement	Type	Validation	Source	Task
	Software				
rq-D5.1-UC1.SW 1	Script should be implementable in IEC 61131-3 in alternative the algorithm for Self-commissioning or Condition monitoring could be developed through field-bus connection like (CAN-BUS or Ethercat).	R	T	WP5	5.2

	Condition Monitoring				
rq-D5.1-UC1.CM1	Algorithm should be define to evaluate the general status of the application and collect information about the process Performance of the inverter should be monitored in order to check the performance of the inverter and to compare the real situation with that was foreseen. The collected information should be used to improve the performance of the control algorithm and to check the sizing of the inverter.	R	T	WP5	5.3
rq-D5.1-UC1.CM2	Monitoring of the drive should help to prevent unwanted situation and to create the basis for a predictive diagnostic and maintenance;	R	T	WP5	5.3
rq-D5.1-UC1.CM3	Normally the main operation of the industrial application is managed by a PLC system. The developed algorithm should be able to transmit data from drive to PLC through standard fieldbus connection.	R	T	WP5	5.3
rq-D5.1-UC1.CM4	Algorithm to detect the unwanted touch of the load or excessive oscillation of the load should be implemented and should be monitored. In this situation the control should be able to compensate or to limitate the effect i.e. adapting the control gains	R	T	WP5/6	5.3
	Self-commissioning				
rq-D5.1-UC1.SC1	Position limit - Auto-tuning approaches has to consider applications limit that allows: Max. and min. position limit; Max. and min allowed speed; Max and min allowed acceleration; Max and min allowed Jerk.	R	T	WP5	5.4
rq-D5.1-UC1.SC2	Actuator limit - According to the type of installation, motor and system auto-tuning has to consider: Max and min Torque limit; PWM (actuator) frequency; Control loop sampling time.	R	T	WP5	5.4
rq-D5.1-UC1.SC3	Application model - The following application characteristics will be taken in account: Inertia (constant or variable); Resonances - Low frequency (oscillation), High frequency (vibration); Friction; Back-lash; Unbalance / Cam; Repetitive noise.	R	T	WP5	5.4
rq-D5.1-UC1.SC4	The identification method should have the minimum duration of the experiment and energy consumption.	R	T	WP5	5.4
rq-D5.1-UC1.SC5	Performance should be evaluated in order to validate the tuning algorithm. Metrics like response time; under and over shoot should be evaluate.	R	T	WP5	5.4
rq-D5.1-UC1.SC6	Interface - Limit of controller like: ● Memory usage; ● Computability	R	T	WP5	5.4, 3.7

	should be taken in account.				

7.2 Compact control + HMI unit for CNC machines - Fagor Aotek controllers (Use case 1.2 – FAG)

ID	Requirement	Type	Validation	Source	Task
	Condition Monitoring				
rq-D5.1-UC2.CM1	The CNC must provide all the information that it can access from the drives connected to the motion buses (CanOpen, Sercos, sercos3, ethercat), mainly data related to torque, power, speed…	R	T	WP5	5.2, 5.3
rq-D5.1-UC2.CM1	The CNC must provide all the errors, warnings and messages coming from those drives or form the CNC and PLC programs.	R	T	WP5	5.2, 5.3
rq-D5.1-UC2.CM1	The system should be able to monitor high frequency signals in short periods of time and eventually store them (oscilloscope) for diagnosis.	R	T	WP5	5.2
rq-D5.1-UC2.CM1	The system should be able to store specific information from data and events during the running time (at lower rates) for condition monitoring and diagnosis.	R	T	WP5	5.2
rq-D5.1-UC2.CM1	The PLC must be able to provide information from all relevant information coming from devices connected to the I/O bus or local bus (for instance, power monitors connected to a CanOpen Bus, intelligent sensors, analog inputs…)	R	T	WP5	5.2
rq-D5.1-UC2.CM1	The CNC must provide all the information related with the operational state of the program running on channels (G-function codes present, program running, subroutines, line…) for diagnosis purposes.	R	T	WP5	5.2, 5.3
rq-D5.1-UC2.CM1	The CNC and PLC must provide information on the state of peripherals and, specifically, on the tool used and related information.	R	T	WP5	5.2, 5.3
rq-D5.1-UC2.CM1	Events could be time stamped for better precision or polling must be fast enough (compromise) .	R	T	WP5	5.3
rq-D5.1-UC2.CM1	When available, all internal relevant data should be gathered by the cnc or the plc and/or collected by a logger and stored or sent. (more specifically temperature of motors, drives, encoders, spindle head…) that are usually transmitted on the drive bus.	R	T	WP5	5.3
	Self-commissioning				

rq-D5.1-UC2.SC1	The CNC software (whether included or external) must have at least an integrated multichannel oscilloscope to plot the tuning relevant signals (torque/current, speed, position, acceleration…) both for command and feedback paths.	R	T	WP5	5.3
rq-D5.1-UC2.SC2	The CNC software should have means to calculate and plot the relevant frequency transfer functions (Bode) relevant for tuning purposes. There are several approaches, being “white” noise over a constant command signal one of the most used, but also chirp signals, etc… can be used	R	T	WP5	5.4
rq-D5.1-UC2.SC3	It is highly desirable that the system has an application to (semi)automatically calculate the control loop constants, as well as the non-linear compensators parameters (backlash, friction…)	R	T	WP5	5.4
rq-D5.1-UC2.SC4	Ideally, every loop, compensation, observer… should have a corresponding way of autotune it integrated in the tuning application.	R	T	WP5	5.4
rq-D5.1-UC2.SC5	It is desirable that the system, included digital drives and fieldbus, can automatically detect connected devices and configure topology.	O	T	WP5	5.4
rq-D5.1-UC2.SC6	It is desirable that the CNC-PLC system has meanings to issue messages, warnings and errors (as different levels of severity on anomalies). It is also desirable that this alarm system be able to work with complex conditions (if (a>b) && c then warning). And that warnings and errors could trigger some messages to a customer (e-mail, sms, application)	O	T	WP5	5.3
	Data fusion and integration				
rq-D5.1-UC2.CM1	It is highly desirable to have a standardization of data coming from sensors through different buses and software stacks to make it available at different control levels (from mechatronics to process control to condition monitoring, loggers, etc…	R	T	WP5	5.2, 5.3
rq-D5.1-UC2.CM1	Specifically, in the CNC-PLC system, an integrated approach to access data from different sources like spindle drives, acceleration or vibration sensors, etc… should lead to better condition monitoring algorithms. The information should be ideally timestamped.	R	T	WP5	5.3
rq-D5.1-UC2.CM1	It is highly desirable that the CNC systems allows the execution in real time of third party control loops or algorithms with access to all the gathered information	R	T	WP5	5.2

	(provided these algorithms are well-behaved, i.e. don't get stuck and hung the CPU...)				
rq-D5.1-UC2.CM1	The information should be exported or published in standard formats. These standards are, for a machine tool, OPC-UA with the machine tool companion standard (VDW, 7-2017) or MT-Connect.	R	T	WP5	5.2
rq-D5.1-UC2.CM1	Where justified by performance reasons, proprietary formats could be preferred (for instance for local resident algorithms as described just before).	R	T	WP5	5.2
rq-D5.1-UC2.CM1	The non-standard information must be published to the client applications.	R	T	WP5	5.2
rq-D5.1-UC2.CM1	For real time integrated algorithms (usually process control), the calling format and application structure will be that of the Simulink environment (inputs, outputs, step (cycle) and integrator updating).	R	T	WP5	5.4
rq-D5.1-UC2.CM1	It is desirable that the communication from the CNC could be served to several clients under several protocols at the same time.	R	T	WP5	5.2
rq-D5.1-UC2.CM1	The CNC must be able to upload information to the company net or to the cloud and be provided with state of the art security software (and/or hardware)	R	T	WP5	5.2
rq-D5.1-UC2.CM1	It is desirable that the CNC system allows for communication protocols included by the customer (for instance MQTT protocols, or proprietary modules)	R	T	WP5	5.2

References

[ISO 17359] ISO 17359: Condition monitoring and diagnostics of machines - General guidelines, second edition, 2011.

[Kampschreur et al., 2018] Kampschreur, T., van der Veen, G. and Lembrechts, T. : Overall requirements on I-MECH reference platform, Deliverable D2.3 of I-MECH project. January, 2018.

[Kania et al., 2011] Kania, J., Panchal, T. H., Patel, V. and Patel, V.: Self commissioning: a unique feature of inverter-fed induction motor Drives. Nirma University International Conference on Engineering pp. 1-6 2011.

[MTConnect, 2018] MTConnect Standard, version 1.4.0, [online document], 2018. Available: MTConnect Institute Online,http://www.mtconnect.org/standard-documents?terms=on [Accessed: March 24, 2018].

[OPC UA, 2018] Unified Architecture, [online document] https://opcfoundation.org/about/opc-technologies/opc-ua/ [Accessed: March 13, 2018].

[Pacas et al., 2010] Pacas, M., Villwock, S., Szczupak, P. and Zoubek, H.: Methods for commissioning and identification in drives, COMPEL - The international journal for computation and mathematics in electrical and electronic engineering, Vol. 29 Issue: 1, pp.53-71, 2010.

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