Doc ID	18032701R04
Doc Creation Date	06 NOV 2017
Doc Revision	04
Doc Revision Date	28 MAR 2018
Doc Status	Final

Workpackage	Deliverable ID
WP4, Control layer design and development	**D4.1, Motion control requirements and specification (first iteration)**

Summary

This deliverable provides a summary of initial requirements analysis and specification for motion control applications utilizing centralized and decentralized control solutions.
The report summarizes the motion control layer requirements specific for the pilot plants and provides the description of general functional blocks, software components, interactions in the motion control layer as well as its interconnection with lower instrumentation level and higher motion planning level.
This report also serves for system architecture revision (D2.4).

(Main)Authors UNIBS Team (Antonio Visioli)

Keywords

Control, Drives, Dynamics, Vibrations, Residual Oscillations, Bandwidth, Iterative Learning Control, Repetitive Control, Robustness, Trajectory Planning, Observers, Filters, Vision

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

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	Due Date	Owner	IAL ID

Document Revision History

Revision	Status	Date	Deliverable leader	Description of changes	IAL ID / Review ID
R01	Draft	01-10-17	Visioli	Initiate	
R01	Draft	05-12-17	Visioli	First draft completed	
R02	Draft	09-01-18	Visioli	Second draft reviewed	
R03	Draft	31-01-18	Visioli	Third draft completed	
R04	Final	17-03-18	Visioli	Final version completed based on reviewers' feedback	

Contributors

Revision	Acronym of Partner	Contributor	Description of work
R03	Nexperia	Van der Veen	Pilot 2 requirements (Section 5.2) and various contributions
	Sioux CCM	Lembrechts	Pilot 1 requirements (Section 5.1) and various contributions
	GMV A&D	Rodrigo	Use case 2.1 requirements (Section 7.4)
	FAGOR	Rodriguez de Yurre	Use case 1.2 requirements (Section 7.2)
	TU/e	Mooren	Various contributions
	EVI	Scordino	Contribution to Section 4.4
	GEFRAN	Colombo	Use case 1.1 requirements (Section 7.1)
	IMA	Collepalumbo	Pilot 3 requirements (Section 5.3)
	PHI	Smeets	Pilot 5 requirements (Section 5.5)
	EDILASIO	Ribeiro	Demonstrator 2 requirements (Section 6.2)
	J&J Vistakon	Hickey	Demonstrator 1 requirements (Section 6.1)
	ZAPUNI	Cech	Use Cases 1.3 and 2.2 (Sections 7.3 and 7.5)

Document control

	Status	Draft						
	Revision	03						
Reviewer Name	**Role**	**Selection**						
Arend-Jan Beltman	Coordinator	X						
Tekniker	Reviewer	X						
ZAPUNI	Reviewer	X						
Gefran	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

Abbreviations & Definitions

Abbreviation	Description
SISO	Single-input-single-output
MIMO	Multiple-input-multiple-output
P	Proportional
PI	Proportional-integral
PID	Proportional-integral-derivative
MPC	Model predictive control
DMC	Dynamic matrix control
GPC	Generalized predictive control
EPSAC	Extended prediction self-adaptive control
PFC	Predictive functional control
UPC	Unified predictive control
LQ	Linear quadratic
SMC	Sliding mode control
RC	Repetitive control
HORC	High order repetitive control
RRC	Robust repetitive control
ILC	Iterative learning control
IP	International Protection
GSC	Generic substrate carrier
AMSR	Axial movable Segment Rolls
PLC	Programmable logic controller

Definition	Description

About this document

This deliverable is related to the Task 4.1 and describes the control system requirements and specifications.
It is based on the analysis performed 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".
In particular, this document focuses on control requirements and summarizes the state-of-the-art of control algorithms applied to mechatronic systems and the future requirements of control systems, with a special emphasis in the pilot plant applications of the I-MECH project.
In fact, the improvement of control systems is a key factor in the improvement of systems engineering in general and of electronics components employed for industrial production, which is one of the main goal of the ECSEL JU.
This deliverable serves therefore as a reference for the activity of WP4 "Control Layer design and development", which aims at developing centralized and decentralized motion control strategies for mechatronic systems. Note that WP4 involves virtually all the building blocks of the I-MECH project. In particular, BB6 "Self-commissioning velocity and position control loops", BB7 "Vibration control module", BB8 "Robust model-based multivariable control" and BB9 "Iterative and repetitive control module" are directly connected to this deliverable as they deal explicitly with control methodologies. BB10 "Control Specific Multi-many core Platform" and BB11 "RTOS for multi-many core platform" are related to the HW/SW platforms which allows the implementation of advanced (possibly model-based) control solutions.
In any case, for each topic described in the following sections the related BBs will be referenced.

1 Introduction

In order to control a generic mechatronic system, manual regulation has left the place to more performing automatic control during the years. For different reasons of safety and advanced performance requirements, the human being has been replaced in the control of generic mechatronic system by an automatic controller.
The controller is the component that allows the plant to obtain the desired performance in terms of desired outputs. It is possible, in fact, to consider the controller as the brain of a generic mechatronic system. The purpose of the controller is to give to the plant the desired inputs in order to obtain from the plant the desired outputs.
If a generic mechatronic system is considered as the plant to control, the controller can work in two different ways: in open-loop or in closed-loop [1].

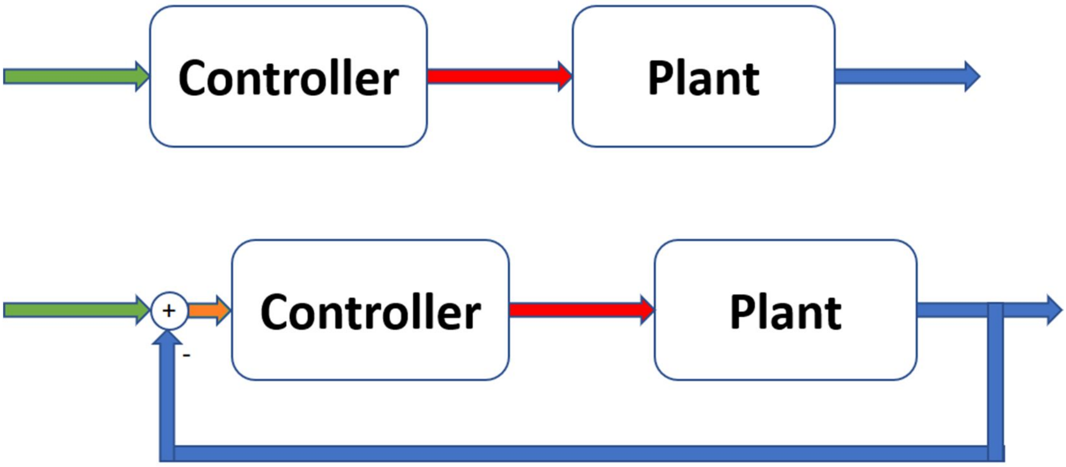


Figure 1 Open loop control and closed loop control

When a controller works in open-loop, the desired output (that is the reference signal, green arrow) is converted by the controller in the proper control action (red arrow) to give to the plant in order to obtain the real output (blue arrow). This control structure can work only if the plant is perfectly modelled in all its parts, if the controller is designed perfectly and if there are not disturbances or modelling errors (this situation never happens in practice).
In order to control the plant in a more robust way, the closed-loop controller strategy has to be applied. In this scheme the controller tries to give to the plant the appropriate control action (red arrow) in order to obtain the desired output (green arrow). The control action is computed starting from the error (orange arrow) between the desired output and the real one (blue arrow).
Depending on the mechatronic system that we face with, several different closed loop control schemes can be used. All of them can be implemented in order to obtain the best possible performance from the mechatronic system.
It is worth stressing that the primary concern of a control architecture is to ensure the safety of the system and this has to be considered at different levels (e.g., the system output must be always inside the required range, as well as the output of the controller).

2 Types of controller structures

The control strategies that can be used to control completely different mechatronic systems are not always the same. Hereafter some of the control structures that can be implemented to control different mechatronic systems are shown. All of them are based on the closed-loop control strategy because, as mentioned before, in case of measurement noise, disturbances, modelling errors, nonlinearities and other problems the system can be controlled [2].
The control structures described hereafter (or combinations of the control structures) are a part of the ones that can be used for the development of the I-MECH platform, and they are:

1- Single loop control
2- Single loop with gain scheduling control
3- Single loop plus feedforward control
4- Cascade control
5- Cascade control with gain scheduling
6- Cascade plus feedforward control
7- Coupled control.

2.1 Single-loop control

The single-loop control structure is the simplest control structure that can be used to control a general mechatronic system.

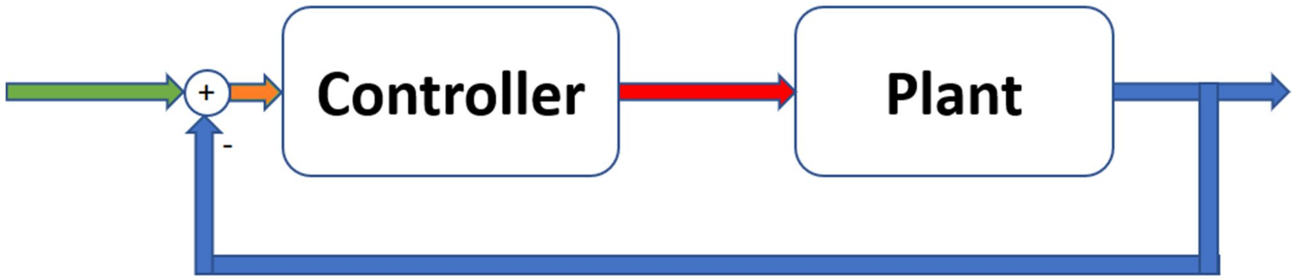


Figure 2 Single-loop control structure

In this scheme the controller gives to the plant the appropriate control action (red arrow) in order to obtain the desired output (green arrow). The control action is computed starting from the error (orange arrow) between the desired output and the real one (blue arrow), see Figure 2.
This structure is able to manage possible nonlinearities, modelling errors and measurement noise and to control the plant in order to obtain the desired performance. This structure can be used to control most of the mechatronic system that we face with, but in contrast to its simplicity, the performance that can be reached by means of this strategy usually are not the best that can be obtained.

2.2 Single-loop with gain scheduling control

In order to improve the performance of the single-loop control structure, especially when the plant is nonlinear, it is possible to use a gain scheduling strategy. The gain scheduling strategy allows the selection between different predefined controller parameters, in order to obtain the best performance close to the working point of the plant. A bumpless switching strategy has to be implemented in order to switch properly between the predefined sets of controller parameters. The single loop with gain scheduling control structure is shown in Figure 3.

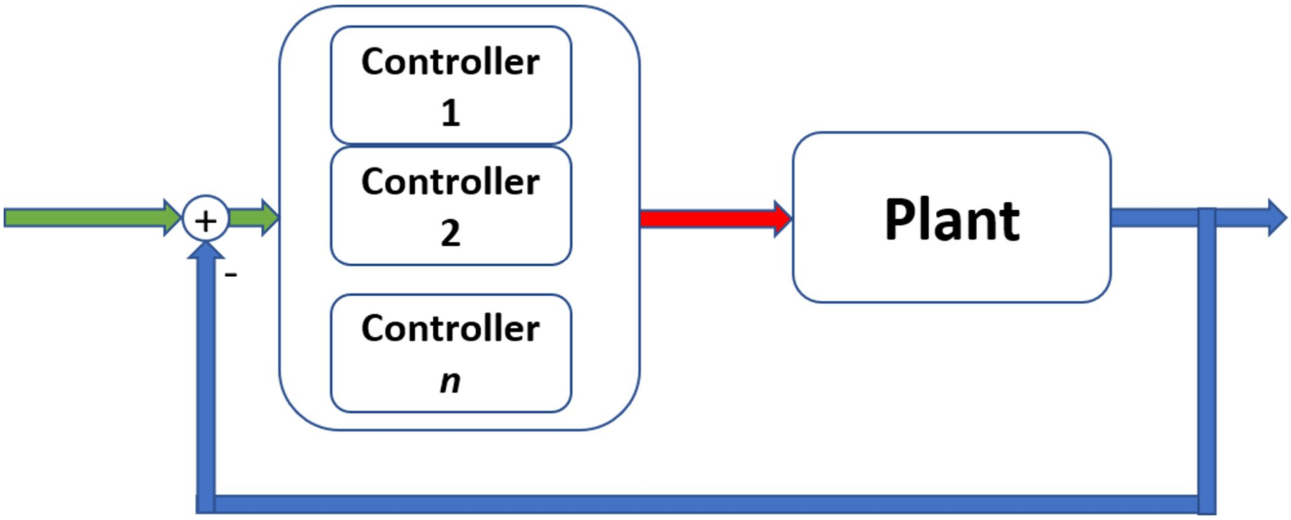


Figure 3 Single-loop with gain scheduling structure

2.3 Single-loop plus feedforward control

Sometimes, in order to increase the dynamic performance of a mechatronic system controlled with a single loop structure, a feedforward action can be added, for example, to compensate some disturbances or to improve the trajectory tracking of the system.

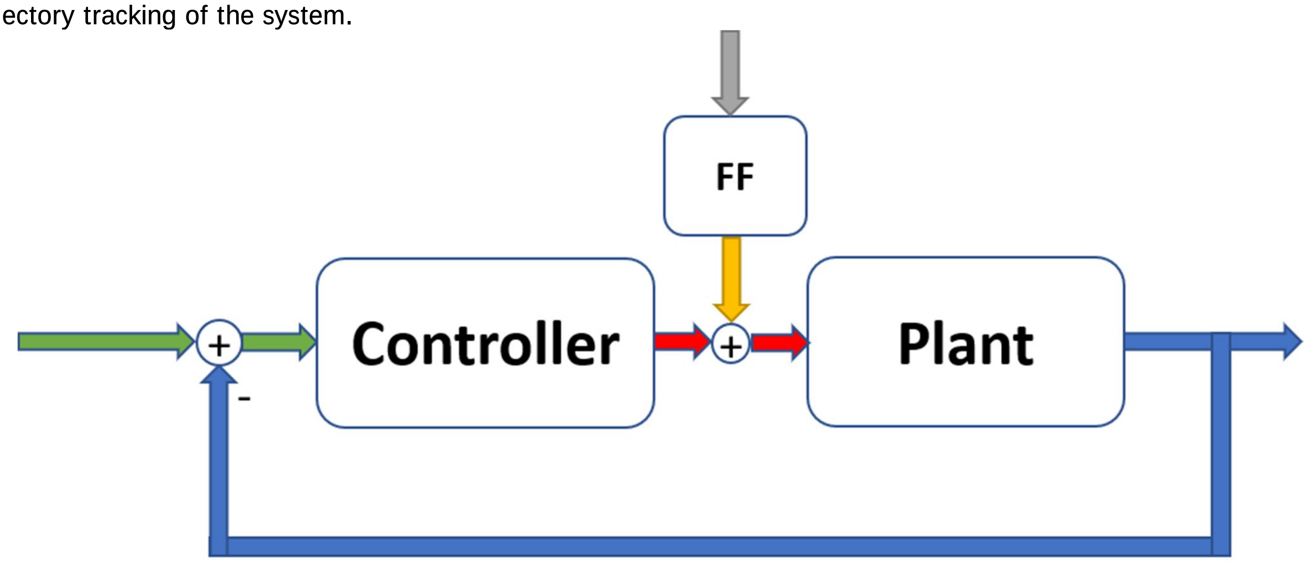


Figure 4 Single-loop plus feedforward control structure

The feedforward block (FF) can be designed in order to perform different tasks such as inertia and/or friction compensation, trajectory tracking, disturbance rejection (see Figure 4). Often the FF block is based on a model of the system (Tasks 4.2 and 4.3).

2.4 Cascade control

In motion control, most of the times, a single-loop control structure does not allow the achievement of a satisfactory performance of the system. To improve the machine performance, multi-loop cascade control structures can be used. Multi-loop cascade structures can be used every time that a plant is composed by the sequence of parts that have different bandwidth. An electric motor, for example, can be seen as composed by three parts: the electric part with a very high bandwidth, a velocity part with a medium bandwidth and a position part with a low bandwidth. An example of a two-loop cascade control structure is the one shown in Figure 5.

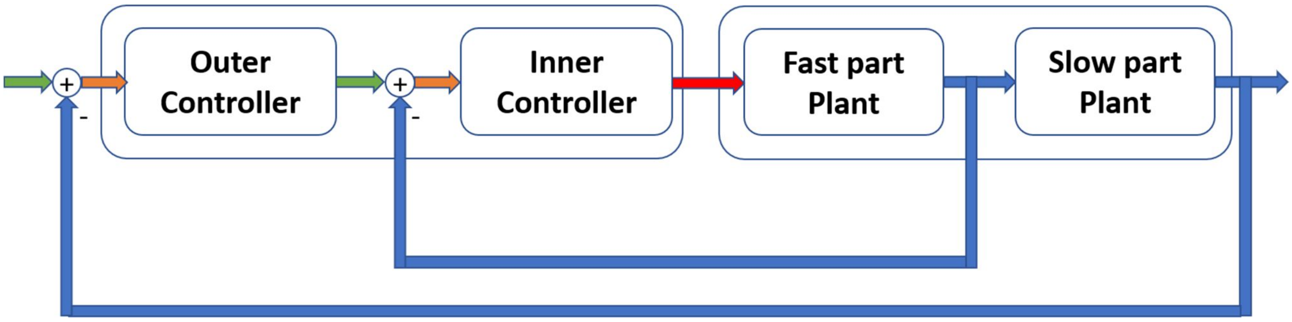


Figure 5 Cascade control structure

In the cascade control structure, each controller is in charge to control a part of the plant. The output of the outer controller becomes the reference signal that the inner controllers have to follow with their parts of the plant. This control structure can bring the plant to obtain better results in terms of trajectory tracking and disturbance rejection. The cascade control structure can be considered as the basic structure for BB6 (Task 5.4) where the two controllers have to be tuned automatically.

2.5 Cascade control with gain scheduling

The gain scheduling strategy described above can be employed also in the cascade control structure to improve the performance that can be obtained from nonlinear plants. The gain scheduling strategy can be used in all the loops of the cascade structure without limitations.

2.6 Cascade plus feedforward control

The feedforward strategy can be adopted also in the cascade structure. Generally, it is possible to use the feedforward to better control the velocity and the torque of a mechatronic system. The feedforward actions can help the cascade structure in the set point tracking and in the disturbances rejection tasks.

2.7 Coupled control

Sometimes, if the mechatronic system is a multiple-input-multiple-output (MIMO) system, it can have a coupled dynamics between the different axes (consider, for example, a standard robot manipulator). In general, for this kind of systems, a MIMO controller has to be used (see Figure 6).

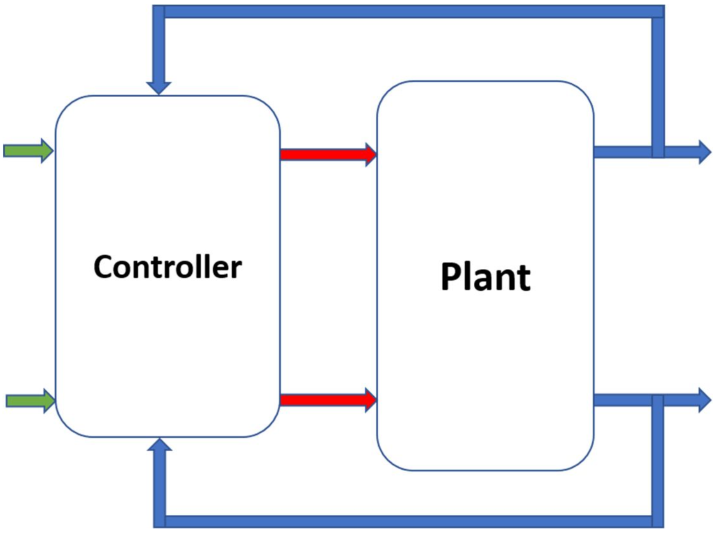


Figure 6 MIMO control system

In particular, in order to obtain a good performance from these systems, a decoupling control can be used. Generally, a decoupling control is used to generate the reference signals for the different actuators in order to obtain the best possible performance from the mechatronic system.
In generic MIMO systems, the control can be centralized or decentralized: if the control is built in a decentralized way, the coupling between the variables can be seen as a disturbance.

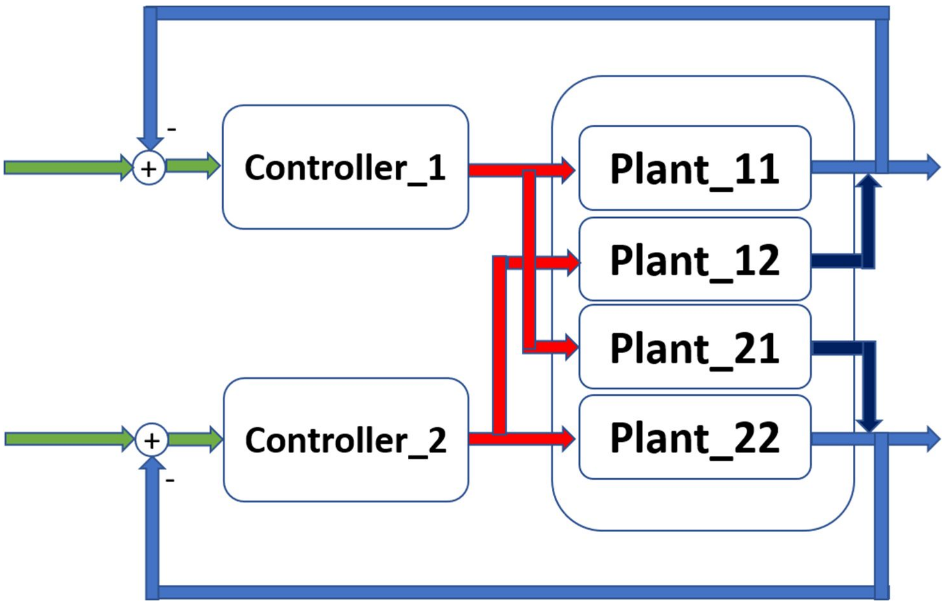


Figure 7 Decentralized control structure

If the control is centralized, different decoupling techniques can be used in order to decouple the dynamics: ideal decoupling (Figure 8), simplified decoupling (Figure 9) and inverse decoupling (Figure 10). The decoupler is usually designed based on a model of the system (BB8 and Task 4.3).

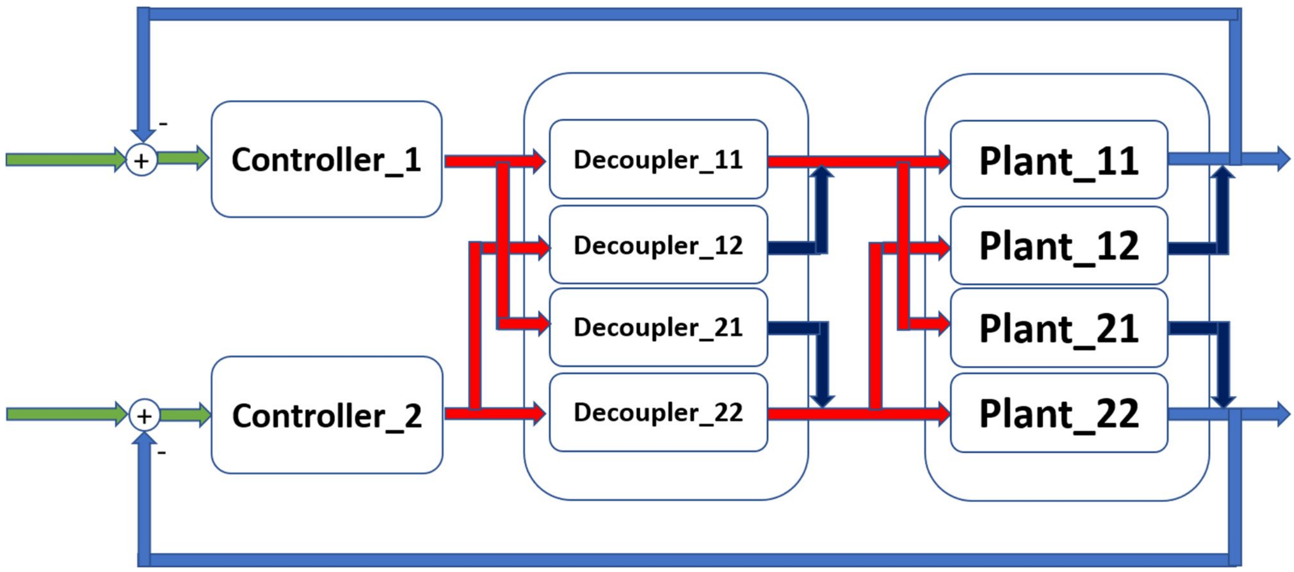


Figure 8 Ideal decoupling structure

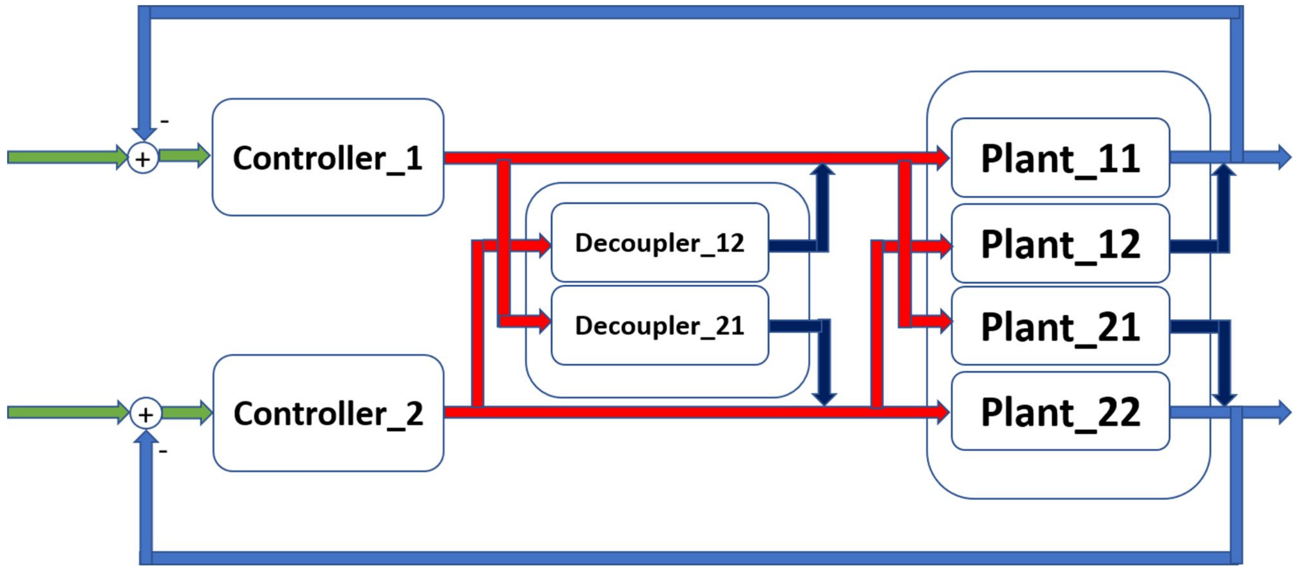


Figure 9 Simplified decoupling structure

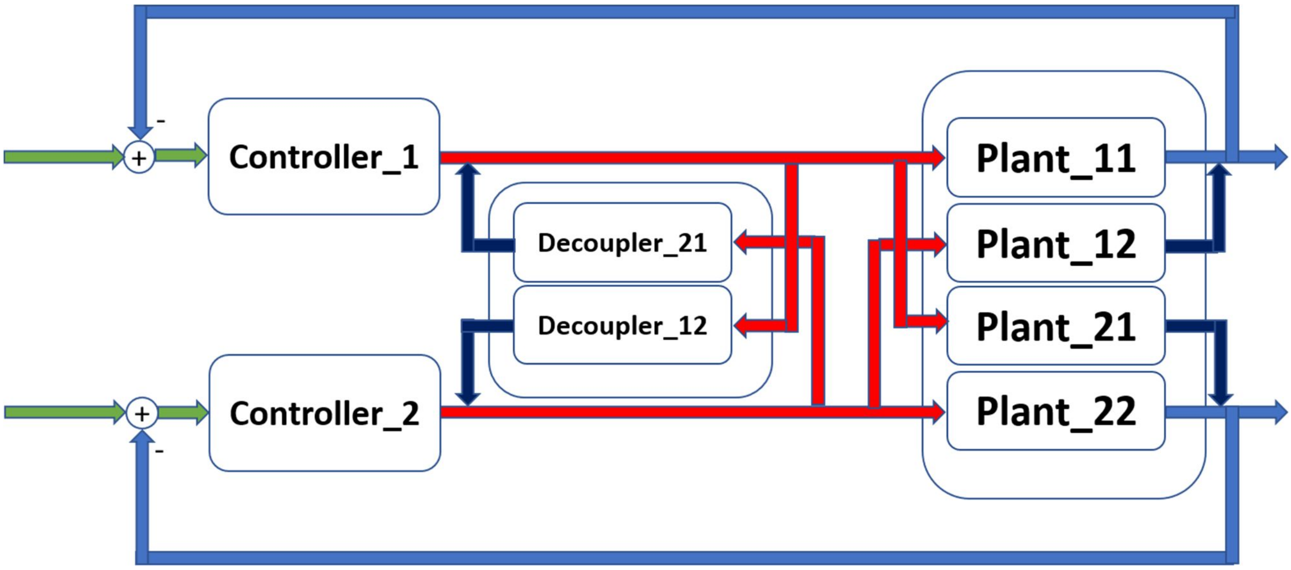


Figure 10 Inverse decoupling structure

2.8 Digital control

Nowadays, controllers for mechatronic systems, therefore, all the previously described control structures, are implemented in a discrete time framework. The use of advanced programmable logic controller (PLC) and industrial computers brought the control of mechatronic systems in the discrete time world. In practice, controllers are nowadays almost exclusively implemented digitally. This means that the controller operates in discrete time, although the controlled systems usually operate in continuous time. Therefore, the digital controller has to be connected to the system by interfaces which:

- transform the continuous-time system output $y(t)$ to a discrete sequence y_k, which can be processed by the digital controller, and
- transform the digital control sequence u_k to a continuous time control signal $u(t)$ sent to the system [3] [4] [5].

The discretization method used for the controller must be appropriate but, most of all, in order to control a system with an acceptable performance, the sampling period of the system has to be sufficiently small. This implies that the computational capability of the control system might have to be improved to satisfy the control requirements of the mechatronic system. The design of a HW/SW platform which allows the implementation of advanced motion controllers is addressed in Task 4.7 and is the specific topic of BB10 and BB11.

3 Motion control systems

In general, different controllers can be used in motion control systems. Some of them comes from several years of experience while others have been recently developed. The following list represents some of the controllers that can be implemented to control a mechatronic system:

- Proportional-integral-derivative controller (PID);
- Predictive controller;
- Optimal controller;
- Adaptive controller;
- H2 controller;
- H∞ controller;
- Observer based controller;
- Sliding model controller;
- etc.

3.1 Proportional-integral-derivative controller (PID)

The proportional-integral-derivative controller (PID) is the most well-known controller that can be used in practice [6] [7] [8] [9] [10]. The control action that the proportional-integral-derivative controller determines is the sum between three parts:

- one proportional to the error between the desired output $Y_{sp}(s)$ and the system output $Y(s)$;
- one proportional to the integral of the error between the desired output $Y_{sp}(s)$ and the system output $Y(s)$;
- one proportional to the first time-derivative of the error between the desired output $Y_{sp}(s)$ and the system output $Y(s)$.

The mathematical law (by using the Laplace transform) of the control action U is

$$U(s) = K_p\left(sT_d + \frac{1}{s\,T_i} + 1\right)E(s)$$

where E is the control error defined as

$$E(s) = Y_{sp(s)} - Y(s)$$

K_p is the proportional gain, T_i is the integral time constant and T_d is the derivative time constant.
The PID controller can be implemented in three different main forms:

- the Ideal (non-interacting) form
- the Series (interacting) form
- the Parallel form.

The transfer function of the PID controller in the Ideal form

$$C(s) = K_p\left(sT_d + \frac{1}{s\,T_i} + 1\right),$$

the transfer function of the PID controller in the Series form is

$$C(s) = K_p'\left(\frac{1}{s\,T_i'} + 1\right)(sT_d' + 1)$$

while the transfer function of the PID controller in the Parallel form is

$$C(s) = \left(K_p^* + \frac{K_i^*}{s} + K_d^* s\right).$$

Each form has its own advantages and disadvantages and can be preferred in a given application.
Of course, formulae that allow the conversion from a PID form to another one can be used [7].
In order to obtain a proper controller transfer function and in order to avoid the amplification of the high-frequency noise, it is necessary to filter the derivative action (i.e., the error the derivative action). Considering that, the derivative part of the PID controller becomes

$$D(s) = \frac{sT_d}{1 + s\,\frac{T_d}{N}}$$

where N is a parameter that allow to tune the filtering action (usually $N = 5 \div 20$).
So, the PID controller forms become:

- the Ideal form

$$C(s) = K_p \left(\frac{sT_d}{1 + s\,\frac{T_d}{N}} + \frac{1}{s\,T_i} + 1 \right),$$

- the Series form

$$C(s) = K_p' \left(\frac{1}{s\,T_i'} + 1 \right) \left(\frac{sT_d'}{1 + s\,\frac{T_d'}{N}} + 1 \right),$$

- and the Parallel form

$$C(s) = \left(K_p^* + \frac{K_i^*}{s} + \frac{sK_d^*}{1 + s\,\frac{K_d^*}{N}} \right).$$

Alternatively, the whole control variable can be filtered (output-filtered PID controller).

In order to avoid the 'derivative kick', which would yield the actuator to saturation, when a step signal is applied to the setpoint, it is convenient to apply the derivative action to the system output variable instead of to the control error. Considering that, the derivative part of the control action becomes

$$U_D(s) = K_p \left(\frac{sT_d}{1 + s\,\frac{T_d}{N}} \right) Y(s)$$

thus, the PID control action (by considering the PID written in the Ideal form) becomes

$$U(s) = K_p \left[\left(\frac{1}{s\,T_i} + 1 \right) E(s) + \frac{sT_d}{1 + s\,\frac{T_d}{N}} Y(s) \right]$$

In order to avoid an excessive overshoot of the system outputs, the set-point value can be weighted just for the proportional action. In this way it is possible to obtain a two-degree-of-freedom controller where the set-point tracking and load disturbance rejections tasks are decoupled.
Considering that, the proportional part of the control action becomes

$$U_{p(s)} = K_p \big(b\,Y_{sp}(s) - Y(s) \big) \quad \text{with } 0 \leq b < 1.$$

In this formulation the parameter b is used to weight the set point.

The PID controller can be finally written in the ISA form, where the set-point weight and the possibility to compute the derivative action on the output variables are taken into account.
The PID control action of a PID written in the ISA form is:

$$U(s) = K_p \left[\big(b\,Y_{sp}(s) - Y(s) \big) + \frac{1}{s\,T_i} E(s) + \frac{sT_d}{1 + s\,\frac{T_d}{N}} \big(c\,Y_{sp}(s) - Y(s) \big) \right] \quad \text{with } \begin{array}{c} 0 \leq b < 1 \\ c = 0 \vee c = 1 \end{array}.$$

When the control variable saturates (because of actuator constraints), the control system is in open-loop. When the error is big for a long time-interval, the integral term can achieve a very high value and therefore the control error should have the opposite sign for another long time-interval in order for the actuator to leave the saturation value. To reduce this problem, different methods for anti-windup can be implemented [11]. Some of them are listed here:

- conditional integration
- PID in automatic reset form
- preload
- back calculation
- etc.

The conditional integration works as follows: the integral term is switched-off (frozen) when the actuator saturates, and the integral term increases the saturation, that is, when control variable and control error have the same sign.
The PID in automatic reset form works as follows: the integral term is computed directly by considering the saturations of the control variables.

The preload works as follows: when the actuator saturates, the integral term is set to a predefined value (usually chosen slightly less than the steady-state value in order to avoid overshoots).
The back calculation works as follows: the integral term is recalculated when the actuator saturates in the following way

$$U_I(s) = \frac{K_p}{sT_i}E(s) + \frac{1}{sT_t}(U_{SAT}(s) - U(s))$$

where $U_{SAT}(s)$ is the saturated value of the control action $U(s)$ and T_t is the desaturation time constant.
The PID controller can be discretized in different ways. Considering for example a PID controller in ideal form, the corresponding continuos-time control law is written as

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(v)dv + T_d\frac{de(t)}{dt}\right)$$

Defining a sampling interval Δt and by using backward finite differences, the integral term can be approximated as

$$\int_0^{t_k} e(v)dv = \sum_{i=1}^{k} e(t_i)\Delta t$$

and the derivative term as

$$\frac{de(t_k)}{dt} = \frac{e(t_k) - e(t_{k-1})}{\Delta t}$$

Then, the discrete-time control law becomes

$$u(t_k) = K_p\left(e(t_k) + \frac{\Delta t}{T_i}\sum_{i=1}^{k} e(t_i) + \frac{T_d}{\Delta t}\left(e(t_k) - e(t_{k-1})\right)\right)$$

In this way the control variable is computed directly (positional algorithm). Alternatively, the control variable at time instant t_k can be calculated based on $u(t_k\text{-}1)$. By subtracting the expression of $u(t_k\text{-}1)$ from that of $u(t_k)$ we obtain

$$u(t_k) = u(t_{k-1}) + K_p\left(\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t}\right)e(t_k) + \left(-1 - \frac{2T_d}{\Delta t}\right)e(t_{k-1}) + \frac{T_d}{\Delta t}e(t_{k-2})\right)$$

which is the incremental or velocity algorithm. It is worth noting that there is no accumulation of error in the velocity form. Thus, the anti-windup can be simply implemented by avoiding to increase the control variable when the actuator saturates.
The velocity form expression can be written more compactly as

$$u(t_k) - u(t_{k-1}) = K_1 e(t_k) + K_2 e(t_{k-1}) + K_3 e(t_{k-2})$$

where

$$K_1 = K_p\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t}\right) \qquad K_2 = -K_p\left(1 + \frac{2T_d}{\Delta t}\right) \qquad K_3 = K_p\frac{T_d}{\Delta t}$$

By defining z^{-1} as the backward shift operator

$$z^{-1}u(t_k) = u(t_{k-1})$$

The transfer function of the discretized PID controller in velocity form is

$$C(z^{-1}) = \frac{K_1 + K_2 z^{-1} + K_3 z^{-2}}{1 - z^{-1}}$$

where K_1, K_2 and K_3 are tuning parameters.

3.2 Model Predictive Control

Model predictive control (MPC) is typically used in the process industry because it requires the solution of complex optimization problems, but nowadays, because of the increased computational power of microprocessors, has become more and more important also in mechatronic applications [12].

A control predictive action can be understood as a control signal that is generated in advance to provide an optimal behaviour of the system. The idea behind a predictive controller is to react now based on the future behaviour of the system [13]. Generally, predictive control does not designate a specific control strategy, but rather an ample range of control methods with similar features: in particular, the use of a model to obtain the control signal by minimizing an objective function.

The design methods share the same basic ideas:

- the explicit use of a model to predict the system output at future instants of time (prediction horizon);
- the calculation of a control sequence by minimizing an objective function;
- the use of a receding strategy, only the first resulting control signal is used at each step and the horizon is displaced towards the future.

The pros of the use of a model predictive control are the following one:

- it is attractive to staff with only a limited knowledge of control (intuitive concepts and easy tuning);
- the extension to multivariable case is straightforward;
- an implicit feedforward action can be used to deal with measurable disturbances;
- the constraints are included systematically;
- it is very powerful when future references are known;

while the main drawbacks are the following ones:

- it is a strongly model-dependent controller;
- the optimization problem is not always easy to manage due to possible long computational time.

In more detail, the model predictive control systems are designed based on a mathematical model of the plant. The model to be used in the control system design can be a transfer function or a state-space model.

By using a state-space model, the current information required for predicting ahead is represented by the state variable at the current time.

Considering the continuous or discrete state space representation of the system (see Section 3.2).

At each iteration the predictive controller is able to compute the future control actions as explained in [14].

The objective function to minimize can be related to the future tracking errors and to the control effort, an example can be the following objective function

$$J = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2$$

where N_1 and N_2 are the minimum and maximum prediction horizons, N_u is the control horizon (typically $N_u = N_2$) and $\delta(j)$ and $\lambda(j)$ are the weighting factors for future errors and control effort (typically $\delta(j) = \delta$ and $\lambda(j) = \lambda$).

In the minimization of the cost function, the reference trajectory (if known in advance) must not be necessarily the same as the desired one: usually it is an approximation.

The cost function can be usually a constrained function.

In order to obtain the future control values, $u(t+k|t)$, it is necessary to minimize the functional j by using the prediction model:

- the predicted outputs, $\hat{y}(t+k|t)$, are calculated as a function of past values of inputs and outputs and future control signals, making use of the prediction model.
- Substitute the compact prediction values in the cost function, J.
- Minimize J with respect to Δu where an analytical solution can be obtained for the quadratic criterion when a linear model is considered and there are not constraints.
- Non-trivial solution, $N_2 - N_1 + 1$ independent variables.
- The structure of the control law depends on the control horizon N_u: $\Delta u(t+j-1|t)$ for $j > N_u$.
- The receding horizon concept is then applied at each sampling period.

Several model predictive control algorithms can be used, example of these algorithms are:

- (DMC) dynamic matrix control [15] [16];
- (GPC) generalized predictive control [17] [18] [19];
- (EPSAC) extended prediction self-adaptive control [20];
- (PFC) predictive functional control [21];
- (UPC) unified predictive control [22];
- Explicit MPC [23];
- Nonlinear MPC [24];

- etc.

The great advantage of MPC is to fully exploit the constraints of the system with a great economic benefit. The extension to the MIMO case is, in principle, not difficult, however, the tuning of the overall control system is not trivial (in addition to the problem to obtain an accurate model of the plant).
The evaluation of MPC as an alternative to PID based control scheme can be done in Task 5.4.

3.3 Optimal/Robust controller

While optimality and robustness are features that should typically present in any control system, when these features are explicitly considered in the design of the controller we denote this approach as optimal/robust control.
In particular, optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved.
While robust control is an approach to controller design that explicitly deals with uncertainties. Robust methods aim to achieve robust performance and/or stability in the presence of bounded modelling errors.

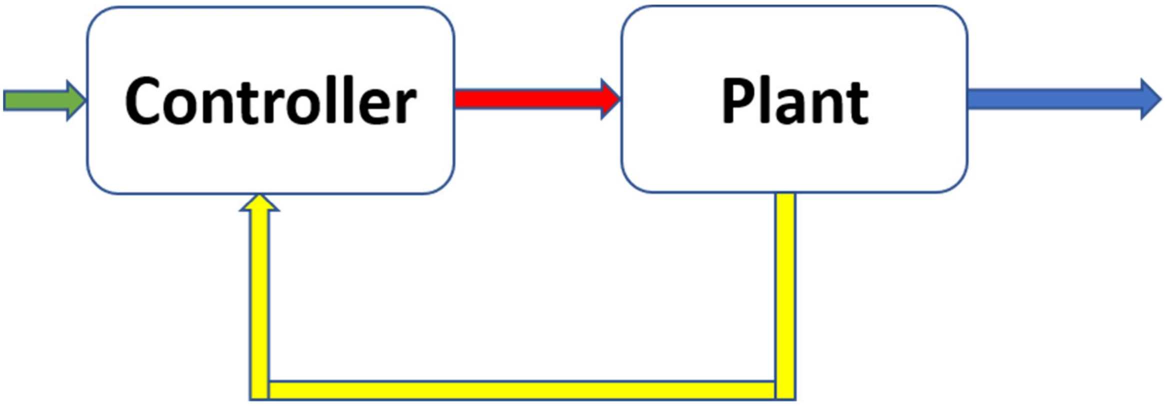


Figure 11 Optimal control

In an optimal control problem (see Figure 11), the feedback variable is usually the state of the system (yellow arrow).
A control problem includes a cost functional that is a function of state and control variables. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost function. The optimal control can be derived [25] using Pontryagin's maximum principle, or by solving the Hamilton–Jacobi–Bellman equation.
Control problems usually include different constraints on both system inputs and/or system outputs.
A more abstract framework goes as follows. Minimize the continuous-time cost functional

$$J = \Phi\left[x(t_0), t_0, x\left(t_f\right), t_f\right] + \int_{t_0}^{t_f} \mathcal{L}\left[x(t), u(t), t\right] dt$$

subject to the first-order dynamic constraints (the state equation)

$$\dot{x} = a[x(t), u(t), t]$$

the algebraic path constraints

$$b[x(t), u(t), t] \leq 0$$

and the boundary conditions

$$\Phi\left[x(t_0), t_0, x\left(t_f\right), t_f\right] = 0$$

where $x(t)$ is the state, $u(t)$ is the control action, t is the independent variable (generally speaking, time), t_0 is the initial time, and t_f is the terminal time. The terms Φ and $\mathcal{L}$ are called the endpoint cost and Lagrangian, respectively. Furthermore, it is noted that the path constraints are in general inequality constraints and thus may not be active (i.e., equal to zero) at the optimal solution. It is also noted that the optimal control problem as stated above may have multiple solutions (i.e., the solution may not be unique).
The optimal controller can be designed by considering the continuous or discrete state space representation of the system.
The feedback system can be represented in such a way exogenous inputs and error variables are highlighted together with manipulated and measured variables. Alternatively, the modelling uncertainties can be also highlighted explicitly in

the control scheme. Then, the norm between the exogenous inputs and the error or the effect of modelling uncertainties can be minimized [26].
Special cases of the optimal and/or robust controllers are:

- the linear quadratic (LQ) optimal controller [27];
- the H2 control method [28] [29] [30];
- the H∞ control method [30] [31] [32].

Optimal/robust control methods that are suitable to be employed in motion control systems will be implemented in BB8 (within the task 4.5).

3.4 Adaptive controller

An adaptive controller is a controller that is able to adapt its parameters in order to increase the performance of a mechatronic system, especially when the system is highly nonlinear, without needing a priori information about the system itself [33] [34].
There are two main types of adaptive controllers, depending on the scheme they use:

- Feedforward adaptive control;
- Feedback adaptive control.

Feedforward adaptive controllers, as the standard feedforward controllers, act in open loop while feedback adaptive controllers act, as the standard feedback controllers, in closed loop.
In the adaptive control strategies, the controller parameters can be computed directly from an estimation of the system parameters, from the control law, or from both [35] [36] [37].

3.5 Inferential control

The so-called inferential control can be used when the controller has to be designed for an unmeasured performance variable [38]. The most employed control strategies applied when there is not the possibility to add some sensors, or when the sensors are not precise enough, are those based on an observer, which “simulates” the presence of a sensor or it is used to obtain a better information from the already present sensors without the implementation of filters. For example, an observer can be used to estimate the position of the load in case of the presence of an elastic transmission (BB7).

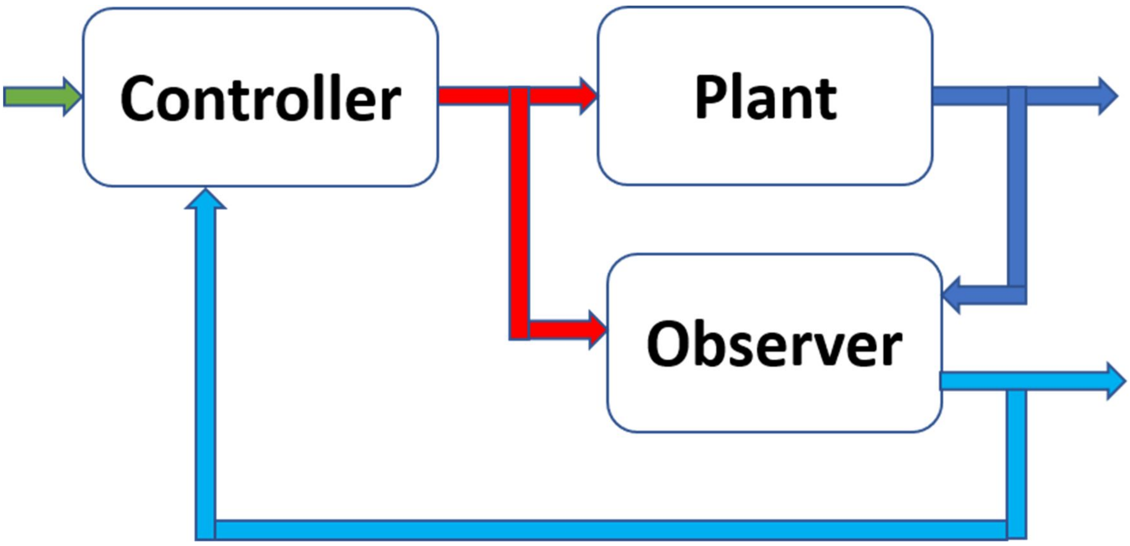


Figure 12 Observer based control

The observer contains the mathematical model of the plant and its outputs or its states can be used as feedback for the controller (see Figure 12).
To keep the observer state as close as possible to the real plant state, an innovation strategy can be used. This innovation strategy allows the observer to follow the real plant during the work.
The linear observer can be expressed by considering the continuous state space representation of the system

$$\dot{x}(t) = \bar{A}x(t) + \bar{B}u(t)$$
$$y(t) = \bar{C}x(t) + \bar{D}u(t)$$

or the discrete one

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k + Du_k.$$

Starting from these formulations, the observer strategies can be

$$\dot{\hat{x}}(t) = \bar{A}\hat{x}(t) + \bar{B}u(t) + \bar{L}(y(t) - \hat{y}(t))$$
$$\hat{y}(t) = \bar{C}\hat{x}(t) + \bar{D}u(t)$$

for the continuous time formulation, and

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y - \hat{y})$$
$$\hat{y}_k = C\hat{x}_k + Du_k$$

for the discrete time formulation.
In the observer formulations, $\hat{x}$ represents the estimated state, $\hat{y}$ represents the estimated outputs, y the real system outputs, u the real system inputs and L is the innovation gain. L allows a fast or a slow system output tracking.

3.6 Sliding mode controller

Sliding mode control (SMC) is a nonlinear variable structure control strategy that uses discontinuous control actions to control generic nonlinear systems [39] [40] [41]. It is usually based on a state-feedback control scheme, and it is generally used as an alternative of the standard cascade control structure. The main characteristics of the sliding mode controller is that a discontinuous control action forces the trajectory of the system to 'slide' on a specific surface which represents the desired system dynamics. In mechatronic systems the use of switching control action usually yields to chattering, which can be detrimental for the system and for this reason appropriate modifications have to be investigated.

3.7 Iterative learning controller

In control systems, Iterative Learning Control (ILC), can be used every time that the machine or a plant has to perform repetitive tasks or every time that a machine or a plant is affected by repetitive disturbances.
Specifically, iterative learning control is a technique for improving the transient response and tracking performance of processes, machines, equipment or systems that execute the same trajectory, motion, or operation over and over. The standard scheme of an iterative learning control structure is the one in Figure 13.

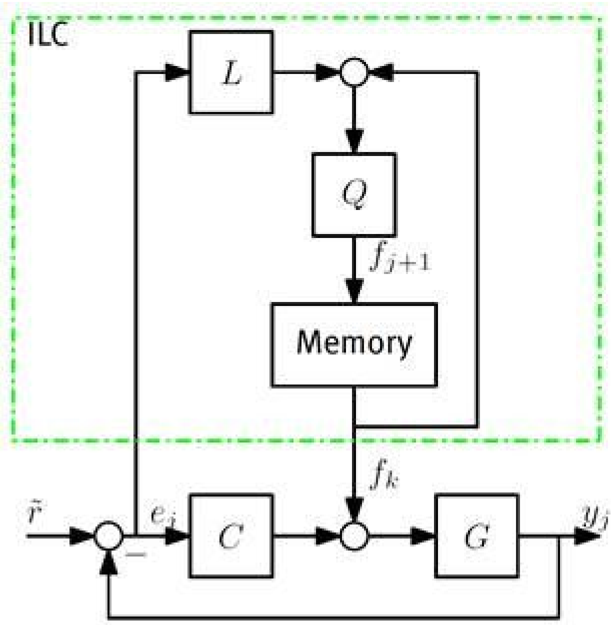


Figure 13 Standard Iterative Learning Control structure

The goal of the learning control algorithm is to compute the future repetition control action by considering the previous repetition information regarding the control error.
In particular, previous trials are used to update a learning signal *f*, based on a learning filter *L* and a robustness filter *Q*. If these are properly designed, this results in attenuation of re-occurring errors, however it can amplify non-repeating errors known as the waterbed-effect.
The memory blocks are necessary to keep the values of the desired signals during each repetition.
With the ILC scheme, the obtained output becomes closer to the desired output repetition after repetition.
A survey of ILC algorithms can be found in [42] or [43].
Note that, if suitably employed, ILC schemes can be also used for the compensation of periodic disturbances.
ILC methods that are suitable to be employed in motion control systems (namely, they have to be computationally efficient, ensure a monotonic decrement of the tracking error, have an automatic tuning function for the design parameters) will be implemented in BB9 (within the Task 4.6).

3.8 Repetitive control

Periodic disturbances can be compensated through the use of appropriate control schemes, mainly known as repetitive controllers.
Some of the methods that can be used are listed hereafter:

- (RC) repetitive control [44] [45] [46] (see, for example, Figure 14);
- (HORC) high order repetitive control [47];
- (RRC) robust repetitive control [48];
- etc.

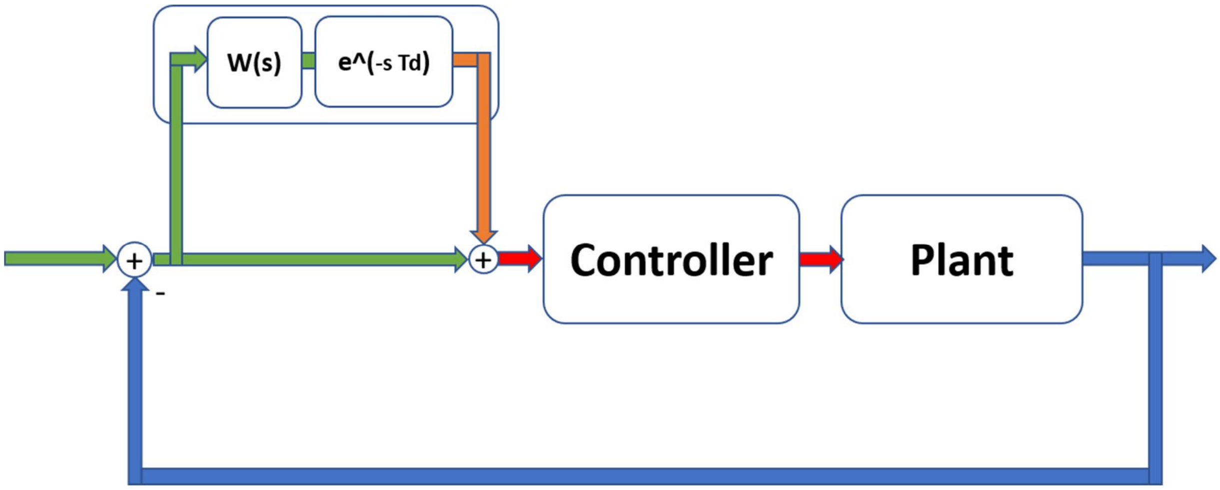


Figure 14 Example of repetitive controller

By considering Figure 14, it is possible to see an example of standard repetitive control (RC) structure. In this structure, the repetitive module is just added to the predefined closed loop control structure. In the RC module, $W(s)$ represents the inverse of the closed loop transfer function between the error signal and the obtained output, while e^{-sT_d} represents the delay between each disturbance action (T_d is, in fact, the period of the disturbances).

Actually, also (ILC) iterative learning control (see Figure 13) can be used to compensate for periodic disturbances [49] [50] [51] [52] [53].

Note that RC methods require the model of the plant while ILC methods do not.

RC methods that are suitable to be employed in motion control systems will be implemented in BB9 (within the Task 4.6).

4 Specifications of a control system

A general control system has to fulfil different specifications depending on the plant to control and on the performance required. A lot of different specifications can be defined before starting the implementation of the controller, but all of them can be clustered in three main groups:

1- Performance specifications;
2- Technical (functional) specifications;
3- Realization specifications.

The performance specification and/or requirements can be seen as the specifications given in the field of the control theory (stability, phase margin, set-point tracking etc.).
The technical specifications and/or requirements are the specifications given in terms of operations (friction and inertia compensation, oscillation compensation etc.).
Finally, the realization specifications and/or requirements can be seen as the specifications to fulfil when the control has to be practically designed and implemented (rapid control prototyping, automatic tuning, connectivity, fieldbus transmission etc.).
In addition to the listed specifications and or requirements, the general safety requirement has always to be taken into account. Indeed, the safety specifications have the priority over all the other specifications.

4.1 Performance specifications

The performance specification and/or requirements can be seen as the specifications given in the field of the control theory [54] [55].
Usually these specifications and/or requirements comes from a control system engineer or an expert in the field of automatic control.
Examples of these specifications are given in the following list:

- stability specifications
- overshoot specifications
- set point tracking specifications
- disturbance rejection specifications
- robustness specifications
- etc.

4.1.1 Stability specifications

The stability specifications and/or requirements are related to the design of a controller that allows the maintenance of the system stability. Usually, in order to assure the stability of the system to control, it is possible to define some parameters that are directly related to the stability. Such parameters can be for example the phase margin, the gain margin, or the maximum sensitivity if a SISO system is considered.
In case of MIMO systems see, for example, [56].
For the case of SISO systems, the phase margin is defined as

$$\varphi_m = 180° - |\varphi_c|$$

where $|\varphi_c|$ is the absolute value of the phase displacement of the loop transfer function at the gain crossover frequency, while the gain margin is defined as

$$K_m = \frac{1}{|L(j\omega_\pi)|}$$

where $|L(j\omega_\pi)|$ is the modulus of the loop transfer function when $\varphi_c = -180°$.
To ensure the robust stability of the system the phase margin or the gain margin must have proper values.
Another index that describes the stability specification can be related to the sensitivity transfer function.
The maximum sensitivity, in fact, can be employed as an effective measure of the system robustness. The formulation of the maximum sensitivity is

$$M_s = \max_{0\le\omega<\infty}\left|\frac{1}{1 + C(j\omega)P(j\omega)}\right| = \max_{0\le\omega<\infty}|S(j\omega)|$$

The less is the value of M_s, the more the system is damped (namely, far from the instability) and vice versa. Typical appropriate values of M_s are in the range $1.2 \div 2$.

The following relationships with the phase margin φ_m and with the amplitude margin K_m hold:

$$K_m > \frac{M_s}{M_s - 1}$$

$$\varphi_m > 2\,\mathrm{asin}\left(\frac{1}{2M_s}\right).$$

The representation of the maximum sensitivity, the gain margin and the phase margin on the Nyquist diagram are shown in Figure 15.

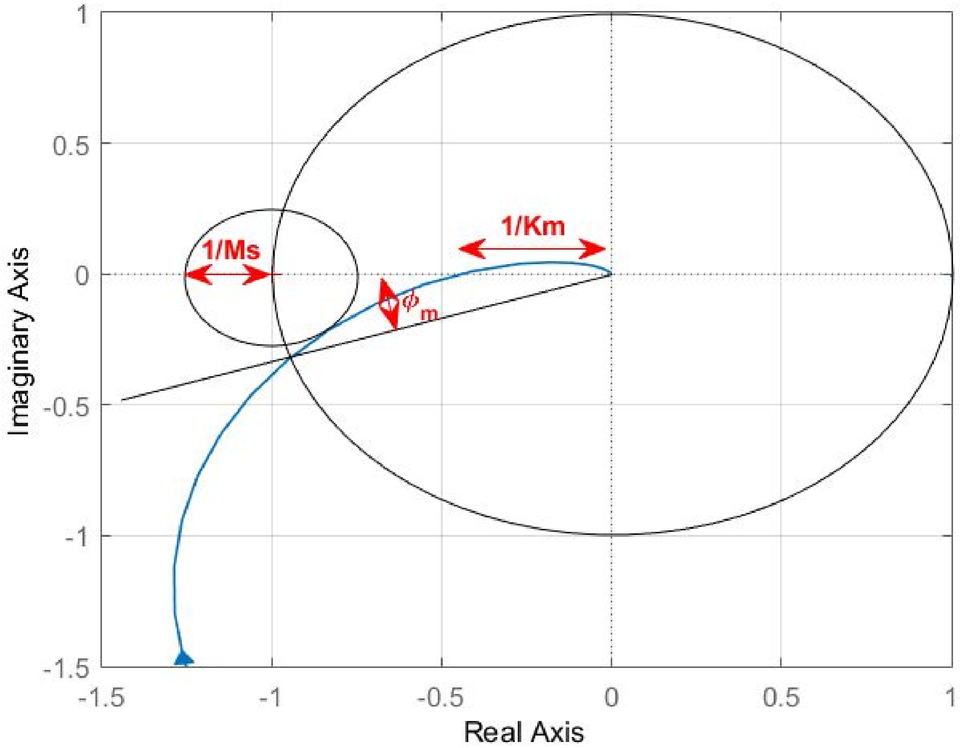


Figure 15 Maximum Sensitivity, Gain Margin and Phase Margin on the Nyquist diagram

Sometimes, in order to obtain a more robust control, some robustness specifications and/or requirements can be expressed. Robustness specifications, usually, can be given by specifying threshold values of some robustness indexes like the phase margin and/or the gain margin described above. Generally, for robustness purposes, the phase margin has to be kept under a certain limit while the gain margin must be kept over a lower limit (see Figure 16). In other cases, some limits on the maximum sensitivity transfer function can be given to ensure the robustness of the controlled system.

Robustness can be referred to parametric (structured) uncertainties, namely, the control performance should be achieved in spite of variations of the system parameters, or it can be referred to non-parametric (unstructured) uncertainties, namely, the control performance should be achieved in spite of the presence of an unmodelled dynamics of the system (for example, high frequency poles).

The design of robust control systems is specifically considered in Task 4.5 and in BB8.

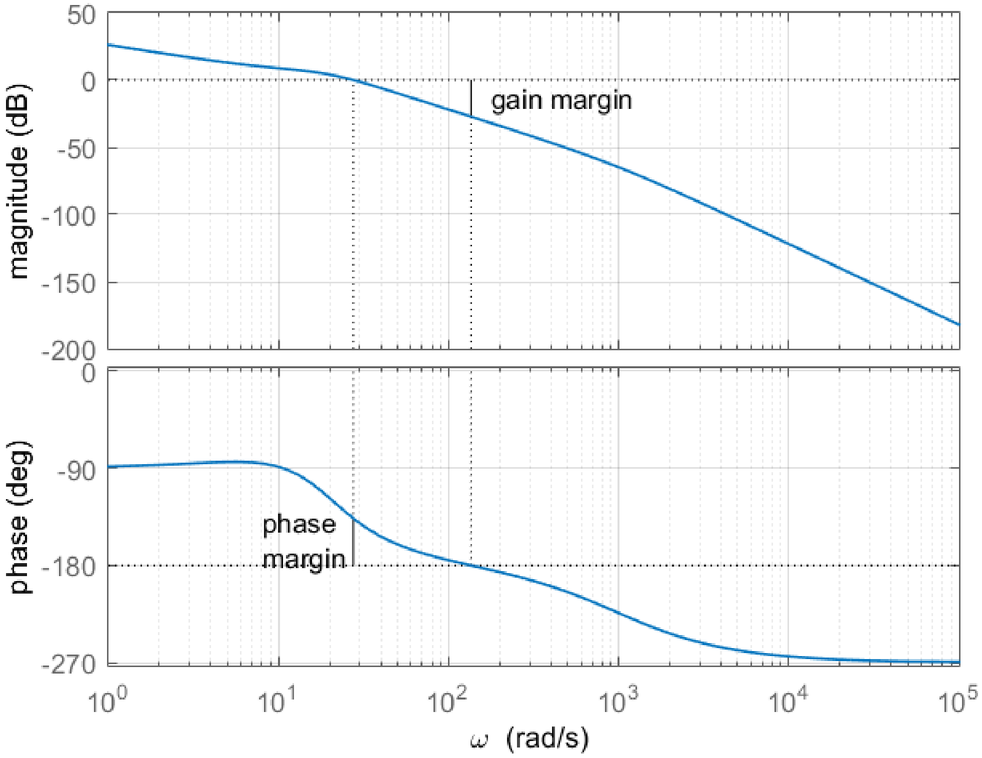


Figure 16 Robustness indexes, gain margin and phase margin

4.1.2 Overshoot specifications

Another technical specification can be related to the maximum overshoot that the system outputs can have during a transient response. The overshoot is the maximum positive difference between the real output of the system and the desired one (see Figure 17). If a big overshoot occurs, the performance of a generic mechatronic system decreases and machine damages can happen, so, generally speaking, a small overshoot is always desirable.

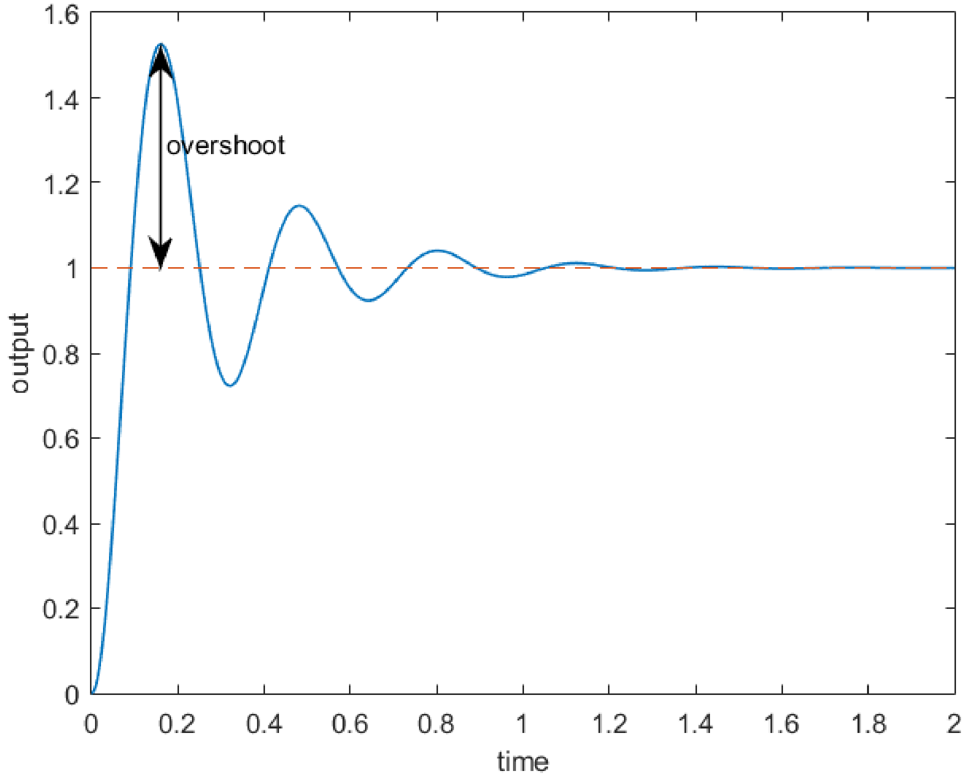


Figure 17 Output with overshoot

4.1.3 Set point tracking specifications

The technical specification and/or requirement that is related to the set-point tracking is usually an important specification when we face with mechatronic systems that have to move very precisely.
The set-point tracking specification usually is given by defining the maximum possible position and/or velocity error during all the trajectory that the machine has to perform.

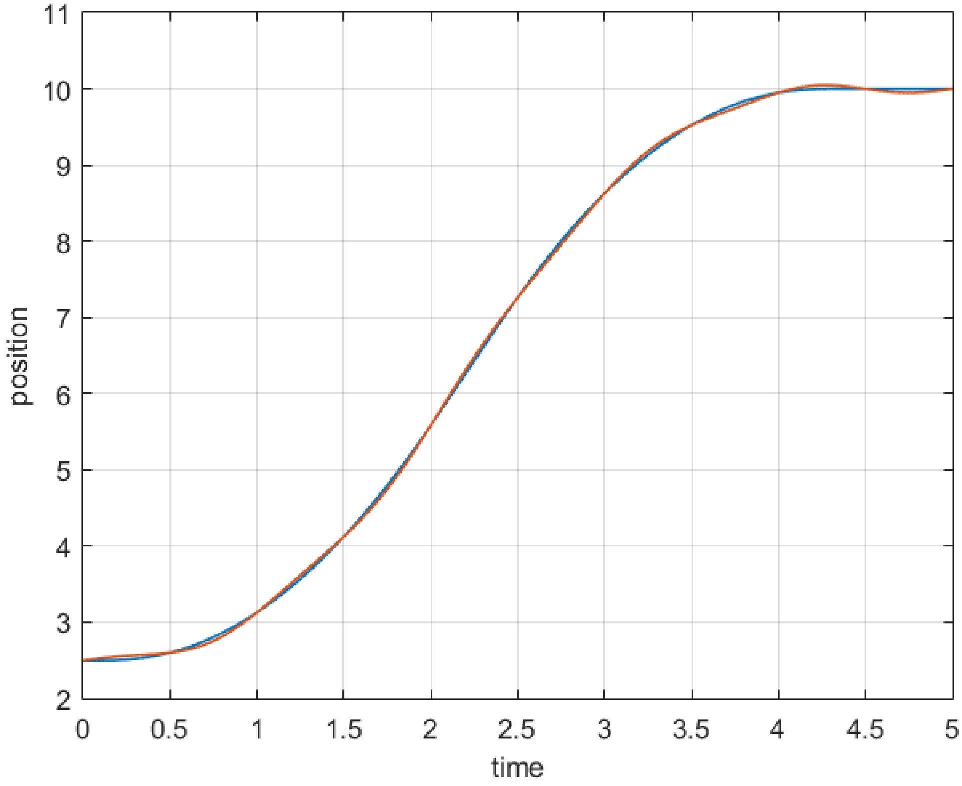


Figure 18 Set-point tracking

4.1.4 Disturbance rejection specifications

Another technical specification is related to the specification of the maximum time for the disturbance rejection. If a disturbance occurs during the system operations, the control strategy that has been implemented must reject the disturbance in a maximum predefined time. The disturbances can act on the system inputs or on the system outputs, and different specifications can be given depending on the type of disturbance (see Figure 19).

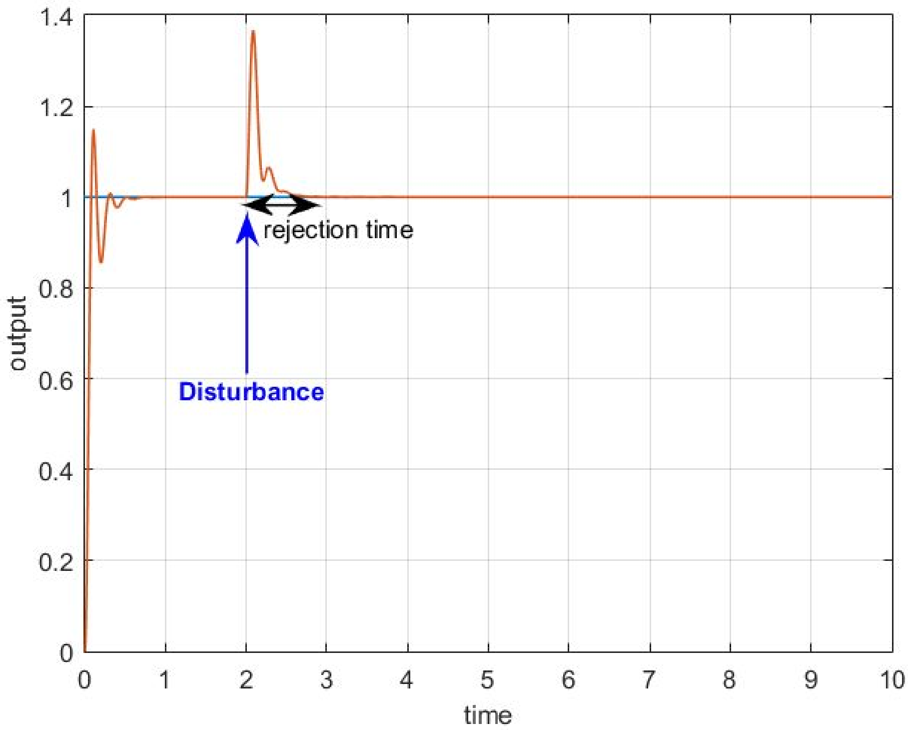


Figure 19 Disturbance rejection

4.1.5 Disturbance attenuation

Another technical specification and/or requirement of a control system can be related to the attenuation of certain particular disturbance acting on particular frequencies. A control strategy, in fact, can be implemented in order to attenuate certain particular frequencies of a predefined order of magnitude. If, for example (see Figure 20), a disturbance with a certain frequency (e.g. 1000 rad/s) acts on the system, the control strategy must reduce its magnitude of a certain value (e.g. 60 dB).

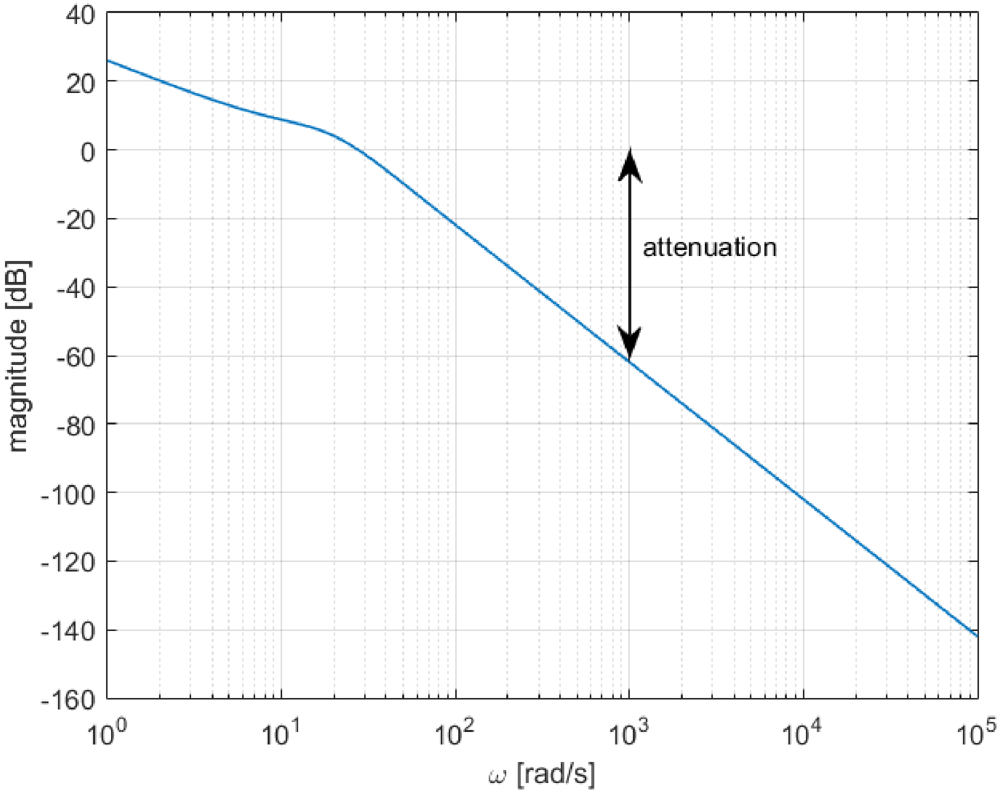


Figure 20 Disturbance attenuation

4.1.6 Bandwidth specifications

If a desired speed of the plant outputs is required, some specifications can be given on the bandwidth of the system. The speed of the system response in the time domain is, in fact, strictly related to the bandwidth of the system in the frequency domain. As mentioned, the specifications can be give on the frequency response of the loop transfer function of the system (see Figure 21), that is on the product between the controller transfer function and the system one.

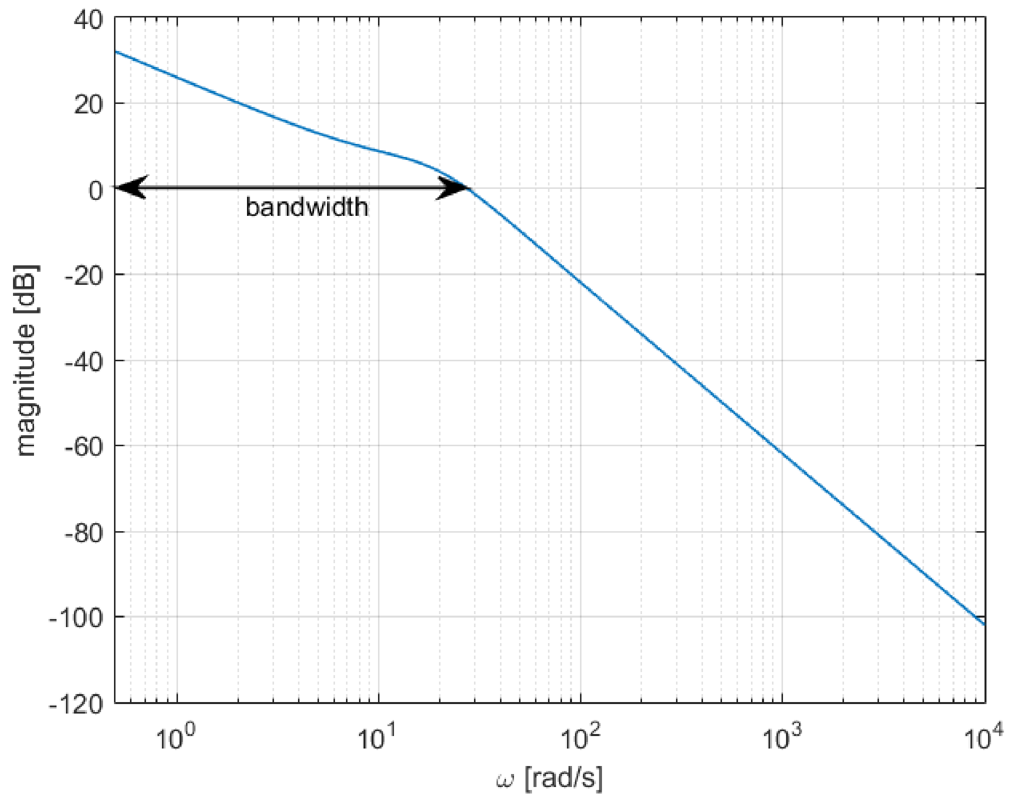


Figure 21 Bandwidth

4.2 Technical specifications of a control system

Other specifications for a control system can be related to the required operations. The technical specifications and/or requirements are, in fact, the specifications given in terms of friction and inertia compensation, oscillation compensation, etc.

Examples of the specifications that engineers formulate after having taken into account the customer specifications and/or requirements are given in the following list:

- Signal filtering
- Friction estimation and compensation
- Inertia compensation and acceleration estimation
- Periodic disturbances compensation
- Repetitive disturbances compensation
- Oscillations compensation
- Vibration compensation
- Robustness to dynamic variations
- Set point shaping block
- Auto-tuning block
- Fault detection
- etc.

4.2.1 Signal filtering

In general, measurement signals are corrupted by noise and for this reason they have to be properly filtered before being exploited by the control algorithm. Standard (low-pass) filtering techniques or more advanced (e.g. Kalman filtering) methods can be used but the design of this software components should be considered as part of the overall control system design, that is, of BBs where signal from the instrumentation layer are used.

4.2.2 Friction estimation and compensation

A method for friction compensation can be used every time that a motion control needs some improvements in positioning precision. Friction, in fact, decreases the positioning precision of a mechatronic system, especially when the speed of motion is close to zero. In order to apply a friction compensation method, it is mandatory to have a reliable friction model of the system to control. Many models proposed in the literature can be used (see, for example [57] [58]), and also temperature-based friction model can be identified and applied [59].

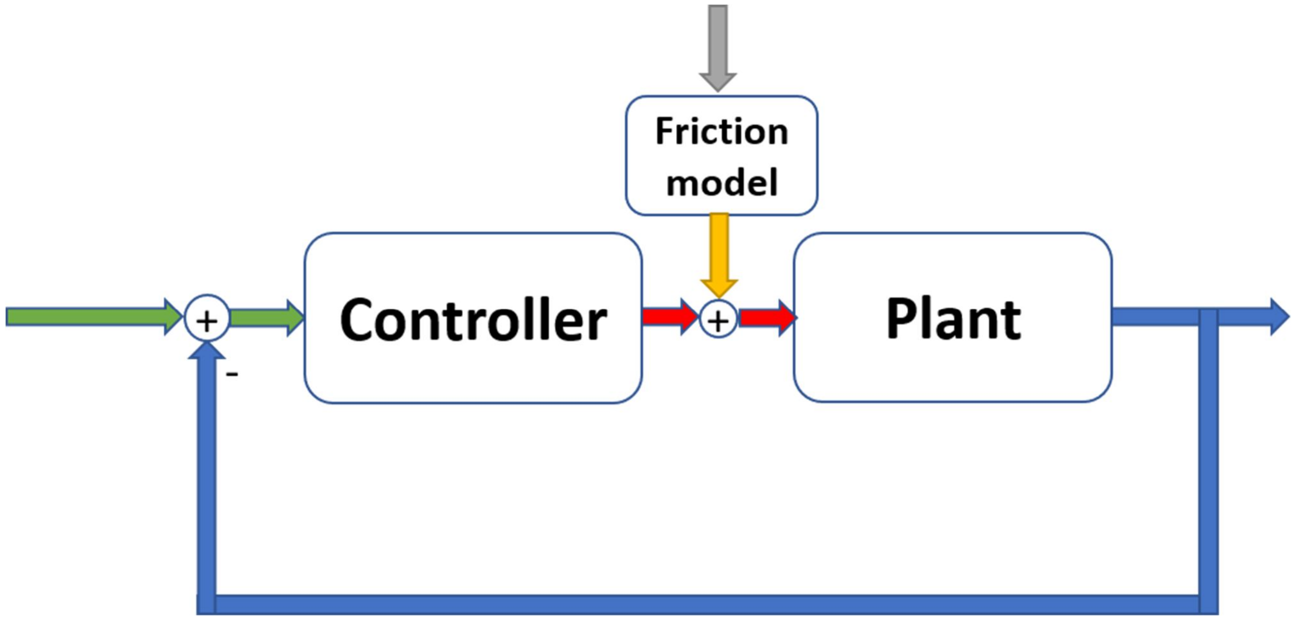


Figure 22 Friction compensation method

In Figure 22, a feedforward method is used to compensate for the friction forces. The feedforward strategy is used to add the estimation of the friction force to the current control action in order to compensate the real friction force.
Several friction compensation methods can be used for mechatronic systems, see for example [58] [60].
A friction compensation block can be used in order to identify the friction model, to estimate the friction model parameters and to compensate for the friction effects (BB6, BB8).

4.2.3 Inertia compensation and acceleration estimation

A method for inertia compensation can be used to improve the set point tracking of mechatronic systems in high dynamics conditions, that is, when the accelerations are relevant. The inertia compensation method, in fact, can reduce the position error when the inertial forces are important. In order to obtain a good inertia compensation strategy, it is required an accurate acceleration estimation, that means: without noise and as close as possible to the real acceleration.
To perform a good acceleration estimation [61] [62], it is possible to use observer based methods [63], Kalman filter based methods, filter based methods [64], etc.
A scheme that can be used for the inertia compensation is in Figure 23.

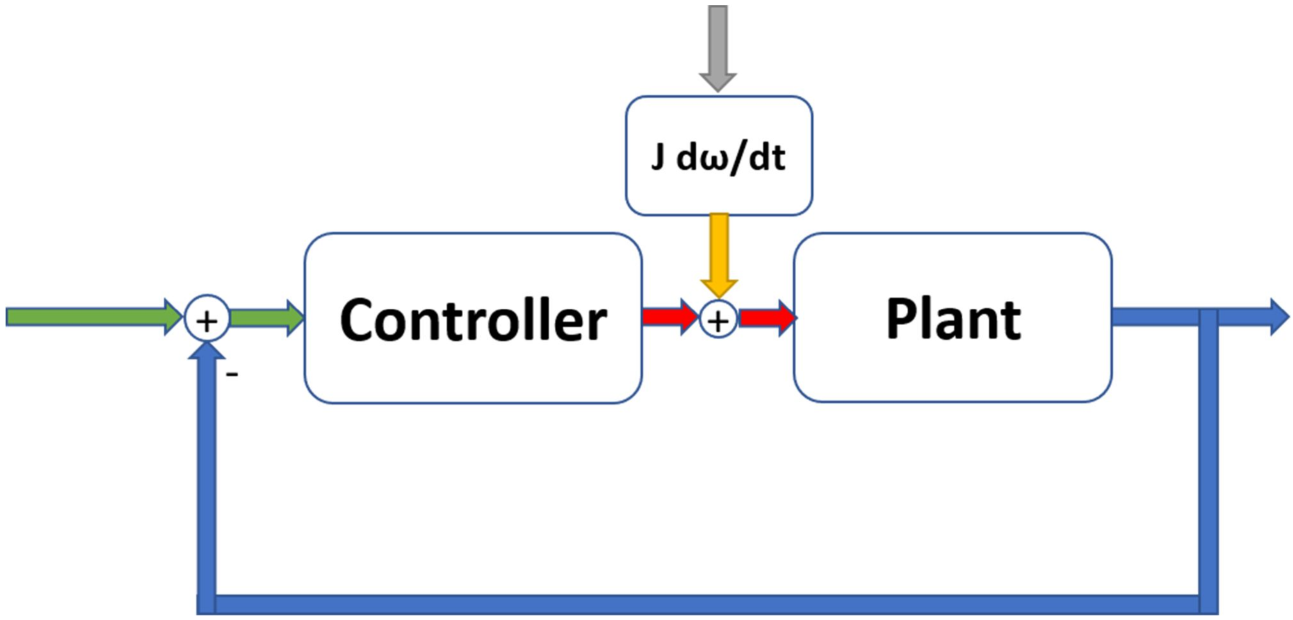


Figure 23 Inertia compensation method

In this scheme, a feedforward method is used to compensate for the inertial forces. The feedforward strategy is used to add the estimation of the inertial force to the current control action in order to mathematically erase the real inertial force.
An inertia compensation block can be added to the I-MECH platform in order to estimate the inertial model and to compensate for the inertial effects (BB6, BB8).

4.2.4 Periodic harmonic disturbances compensation

A harmonic disturbance is a disturbance that is composed by a sinusoid with a certain period and a certain amplitude.
A method for periodic harmonic disturbance compensation can be used in order to decrease the outputs error related to periodic disturbances. Periodic harmonic disturbances, in fact, could not be properly compensated just with a standard feedback controller. In order to compensate for periodic harmonic disturbances, some different techniques can be used. Examples of these techniques are:

- standard feedforward technique (if the disturbance is well known and if the model of the plant to control is also known);
- adaptive feedforward techniques [65] [66];
- filter based techniques (notch filtering, if the disturbance is inserted directly by the controller) [67] [68];
- etc.

A periodic harmonic disturbance compensation block can be added to the I-MECH platform in order to simplify the compensation of periodic disturbances on the different mechatronic systems (BB9).

4.2.5 Repetitive disturbances compensation

Repetitive disturbances, that are different from periodic harmonic disturbances because they are not composed by just a sinusoid, cannot be compensated just with a standard feedback controller.
As for harmonic disturbances, also periodic disturbances can be compensated through the use of appropriate methods like the ILC or repetitive control methods described in Sections 3.9 and 3.10.
RC and ILC methods are specifically addressed in Task 4.6 and results will be implemented in BB9.

4.2.6 (Low-frequency) Oscillations compensation

We denote as oscillations the low frequency phenomenon that appears every time that the load is not rigidly connected to the motor. Examples of mechatronic systems with oscillations are overhead cranes, elevators, tendon driven robots etc.
Methods for oscillations compensation can be implemented to reduce the risks of undesired motions around the set point.
Different strategies for oscillations compensation can be used, depending on the type of motion that the mechatronic system has to perform. Oscillations compensation strategies can be, for example, in open loop, in closed-loop, sensor based, sensor less, etc.
Some of the different methods of oscillations compensation that can be use in practice are listed here:

- input shaping based methods [69] [70] (see, for example, Figure 24);
- inverse dynamics based methods [71] [72];
- fuzzy logic based methods [73];
- observer based methods [74];
- etc. [75]

The creation of an oscillation compensation block in the I-MECH platform can help to easily compensate for undesired oscillations acting on the plant.

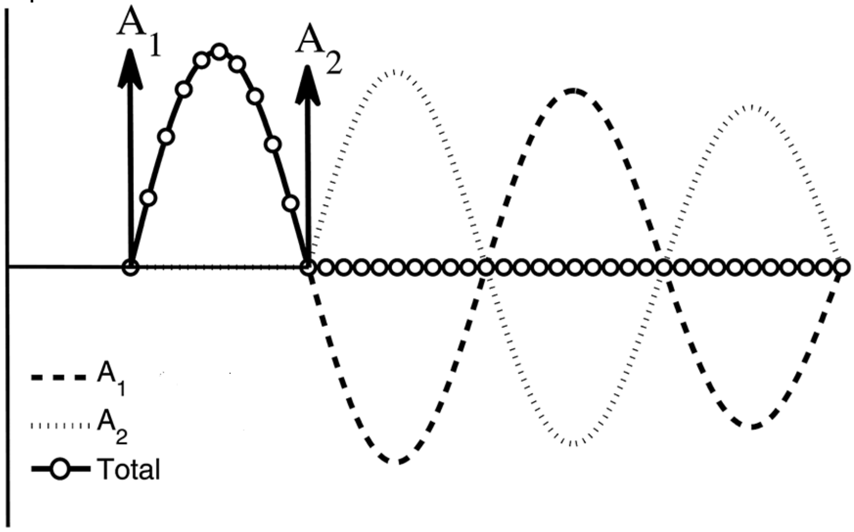


Figure 24 Example of input shaping principle for oscillation reduction

In Figure 24 it is presented the working principle of the input shaping technique used to avoid oscillations of the load. The figure represents the working principle of the two-impulse input shaping, also called Zero-Vibration (ZV) shaper. Here, two impulses are given as inputs to the system. If the impulses are computed with a proper amplitude (namely A_1 and A_2), and they are applied with a proper time delay (equal to half of the period of oscillation), they generate two oscillatory responses which cancel each other. The mathematical convolution between the desired input and the properly computed impulses generates an input signal that allows to obtain a zero-oscillation steady state.
Oscillation compensation methods are specifically addressed in Task 4.4 and results will be implemented in BB7.

4.2.7 (High-frequency) Oscillation compensation

We denote as vibrations the high frequency phenomenon that might affects a mechatronic system.
Methods for vibrations compensation can be implemented to reduce the risks of undesired motions around the set point.
Different strategies for vibrations compensation can be used, depending on the type of motion that the mechatronic system has to perform. Vibration compensation strategies can be, for example, in open loop, in closed loop, sensor based, sensorless, etc.
Some of the different methods for vibration compensation that can be use in practice are listed here:

- Kalman filter-based methods [76];
- Repetitive controller-based methods [77];

- Input shaping-based methods [78];
- others [79] [80] [81] [82].

The creation of a vibration compensation block in the I-MECH platform can help to easily compensate for undesired vibrations acting on the plant.
Vibration compensation methods are specifically addressed in Task 4.4 and results will be implemented in BB7.

4.2.8 Robustness to dynamic variations

Mechatronic systems, in general, change their behaviours during their working time. This phenomenon is common in all the mechatronic systems and it can be related to different causes. For example, friction force changes its value depending on the temperature of the mechanical transmissions of the mechatronic systems.
In order to maintain the performance of the mechatronic system during the working time, a robust controller must be implemented. The robustness of the controller is necessary in order to take into account the dynamics variation of the mechatronic system during the working time. This topic is specifically considered in Task 4.5 and in BB8.

4.2.9 Set-point shaping block

Mechatronic systems usually require set points that depend on the mechanical components of the system to control. Un-appropriate set point signals can introduce, for example, undesired vibrations and can usually bring the actuators to the saturation. Furthermore, un-appropriate set points can produce overshoots and oscillations, with the possible risk of plant damages.
If, for example, a motor needs to be controlled, a step position set point is the worst set point that it is possible to use. Set point shaping blocks can be used in the I-MECH platform to allow a smooth control of the motors and, in particular conditions, they can be used also to help the implementation of some strategies like input shaping and/or others.
In motion control systems, the trajectories that can be used to reach the final position set point can be different. Some of these types of trajectories are, for example:

- Three-traits velocity profile;
- S-curve velocity profile;
- Polynomial profile;
- Cycloidal trajectories;
- etc.

The parameters of all these types of trajectories can be computed in some optimal way, for example by minimizing the jerk, by minimizing the total motion time under some constraints,
Various feedback or reference shaping filters may be used to improve measurement noise attenuation and excitation of high-frequency dynamics of the mechanical loads.
Furthermore, on-line trajectory re-planning can be implemented in the I-MECH platform in order to have an on-line re-planning of the trajectory to apply on several different mechatronic systems.

4.2.10 Auto-tuning block

The tuning of the controller parameters is usually one of the most difficult tasks for a control system engineer, especially when the system to control is nonlinear and/or it has several control loops. The tuning of the controller is not just related to the controller parameters themselves, but also to the parameters of the filters applied to the controller, that are, the filter applied to the output of the system (if necessary), the filter applied to the controller output (if necessary), and all the filters necessary to provide a satisfactory performance. Furthermore, the control action might include also the feedforward controllers for, e.g., the inertia compensation and/or the friction compensation.
In order to have the most performing control action on a mechatronic system, an auto-tuning block can be used to tune all the parameters that are involved in the motion control of the machine.
Some tuning rules exists for the control of industrial processes (e.g. the Ziegler-Nichols method, the Haalman method, the Kappa-Tau method, the AMIGO method and others), but auto-tuning procedures for mechatronic systems are not very well-known and used, especially for industrial drives [83].
The auto-tuning block in th I-MECH platform has to identify the mechatronic system, to tune the control parameters without the help of the operator. Of course, some limits need to be respected during the auto-tuning procedure. If, for example, the system must stay almost still during the auto-tuning procedure, some velocity limits must be inserted.

An effective design of the repetitive controller with auto-tuning capability applicable both for single and multivariable systems with enhanced disturbance rejection optimizable for multiple performance criteria can be created.
Iterative methods for combined experimental identification and robust control design can be developed.
The automatic controller design method has to consider the compliance in the attached mechanical load based on the plant model acquired from the identification procedure.
BB6 is specifically related to the automatic commissioning (for which tuning is the most relevant part) of position and velocity control loops. Although this feature is part of WP5 (Task 5.4), there is no doubt that the control requirements play a significant role in this context.

4.2.11 Fault detection

The detection of faults is, nowadays, more and more important. In fact, it can help engineers to find the problems acting on the mechatronic system and, at the same time, can reduce the possible damages that can occur after a fault of some components. Thus, fault detection procedures play a key role in the ensuring the safety of the system.
In order to have a good fault detection block, different strategies can be used. One of this strategy is, for example, to introduce additional sensors on the mechatronic system and to exploit signals related to the control system.
Regarding the electrical drives, fault detection algorithms based on sensor fusion methods can be implemented. Otherwise, the development of a set of models of drives with normal operation and fault profiles for fault detection and diagnosis using Bayesian approaches which can be integrated into the drive control systems. Another method can consist of using algorithms and sensors for combined vibration and acoustic noise analysis for servo drives, used for on-line diagnostics and predictive maintenance. The enhancement of condition monitoring and fault detection by means of vibration, acoustic and ultrasonic sensing and analysis including utilization of miniature (e.g. MEMS) vibration and acoustic sensors and advanced signal processing techniques can be used.
This topic is part of WP5 (Task 5.3) but the connection with the control requirements is quite strict as an effective control system might hide the occurrence of a fault (for example, a dramatic increment of the friction because of a damage in a transmission might be correctly compensated by the friction estimation and compensation block).
BB3 is specifically related to the robust condition monitoring and predictive diagnostics, and therefore to the fault detection.

4.2.12 Homing strategies

In many applications homing routines are necessary to provide the servo drive with the absolute position of the motor with respect to the machine. Depending on the available sensors, a suitable homing procedure has to be performed. This should be done in any case by taking into account speed and position limits, for example in order to avoid overtravelling. In a general motion control system, the homing strategy should be therefore reconfigurable to suitably deal with a particular application.

4.3 Design specifications

Control requirements involve also requirements on the control system design phase. In order to provide a framework for the I-MECH platform for the development of the control related BBs, a suitable software environment has to be defined by taking into account the following aspects.

4.3.1 Model-based design

The development of (advanced) control functionalities as those described in the previous sections requires in most of the cases the development of a model-based approach, for which the development of a model of the system is of course essential. For this purpose, suitable general modelling strategies should be considered and this is the specific topic of Task 4.2. The model should take into account all the different issues arising in mechatronic systems (friction, inertias, elasticities, etc.) and should consider both SISO and MIMO systems. It has therefore to be sufficiently complex in order to consider all the different physical aspects that are relevant in a given application. However, suitable model reduction strategies should be developed for the purpose of designing the controller in a simple and possibly standardized way.

4.3.2 Identification

Once a model has been elaborated, its parameters have to be estimated by means of a suitably developed identification method (Task 4.3). In order to be general, the identification experiment should take into account the kinematic and dynamic limits of the actuator(s) and of the application. Then, the experiment should not be too time consuming and should be performed in such a way to avoid any risk of failure or damage for the machine. Suitable general identification routine should be therefore developed for the purpose of obtaining the complete model of the system.

4.3.3 Automatic tuning and adjustability

The identification procedure is often used together with a controller tuning procedure in order to obtain a fully automatic tuning procedure, which is actually the topic of Task 5.4 but it has a clear relationship with control requirements (BB6). The control platform should implement this functionality in order to allow the fast commissioning of the control system and, in general, should allow the easy adjustability of the control parameters when the system dynamics or the control specifications change. Automatic tuning should be applicable also to those design parameters involved in BB7, BB8 and BB9 in order for this BBs to be exploited in a wide range of application.

4.3.4 Control prototyping

The overall controller design (i.e., the selection of all the parameters of the algorithm) should consists of different steps according to the V-model framework. This implies that a model of the system should be used (with different level of complexity) both for the design of the control algorithm and for its testing. In this latter case software-in-the-loop (SIL) and hardware-in-the-loop (HIL) simulations could be performed and for this purpose suitable software (e.g. Matlab/Simulink or Amesim) and hardware tools that are capable to reproduce the system dynamics have to be available (Task 4.2). In addition, software tools should be available for the fast and easy deployment of the control algorithm into the processor of the drive. In other words, the devised building blocks related to control have to be implemented with software tools (e.g. Simulink) so that

- they can be easily integrated in the overall control system design;
- the parameters can be easily adjusted by the user (if it is not possible in automatic mode);
- the controller can be easily tested on an accurate model of the system (SIL and/or HIL);
- the code of the control algorithm can be easily generated and deployed into the specific hardware. In this phase, it has to be ensured that the generated code is 100% reliable in order to avoid to compromise the safety of the system.

4.4 Realization of the control system

The control algorithm can be seen as a mathematical operation between certain inputs and outputs, however in order to implement all the required functionalities of a control system, a suitable HW/SW platform has to be available. In particular, the following items have to be considered:

- specific timing requirements (i.e., small sampling period, low jitter and delays) should be ensured;
- the available memory should be sufficient for the implementation of the control algorithms, especially those that require look-up tables;
- SISO as well as MIMO controllers could be implemented, which requires several sensors to be read (perhaps from several distinct sensor boards) and several amplifiers to be addressed simultaneously;
- it should be possible to probe and record and modify all controller signals, settings, internal states, etc… online, to analyse the control system during development, commissioning and perhaps during operation;
- the connection between the control layer and the instrumentation layer (WP3) should be made transparent to the control algorithm by implementing specific protocols. In fact, when a controller operates, the transmission of data between the controller itself and the different peripherals is one of the most important activity. The transmission of data can be real time or not, depending on the type of data to transmit. One of the many kinds of fieldbus is usually employed in this context. Note that, nowadays, also wireless technologies can be adopted to communicate on the field. They have the clear advantage of avoiding cablings but, on another side, the battery consumption and the transmission bandwidth are issues that have to be carefully considered in the design of the control system.

In this context, within WP4, Task 4.7 (related to BB11) aims at developing a many-/multi-core platform for mixed criticality systems. In particular, it aims at the concurrent execution of multiple operating systems with different criticality levels on the same hardware platform.
These can be, for example, a real-time operating system (RTOS) like ERIKA Enterprise (http://www.erika-enterprise.com/) handling time-sensitive or safety critical tasks and a general-purpose operating system like Linux for non-critical tasks (e.g., logging, networking, HMI, etc.).
The main benefit of such an approach is the reduction of both design costs and time-to-market of complex mechatronic systems.

From this point of view, the upper layers of the I-MECH platform must provide proper communication mechanisms between the different operating systems running on the system.
The communication must preserve the real-time characteristics of the critical part (e.g., through non-blocking primitives or lock-free data structures) as well as meet non-functional requirements such as low communication latency, reliability and security.

The realization of the control system should ensure that the safety specifications are satisfied at all levels. Example of safety specifications can be related to:

- safe controller output;
- safe code generation;
- safe system operation;
- safe tooling use;
- safe sensoring;
- safe system procedures (auto-tuning, homing, error handling, etc.);
- etc.

5 I-MECH Pilot Plants requirements and specifications

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

The GSC (Generic Substrate Carrier), developed by Sioux CCM, is a stainless-steel conveyer belt for very accurate transport of substrates. It is capable of transporting practically any kind of substrate, for example paper, cardboard or foils, but also wooden or glass panels. The system is particularly interesting for the industrial inkjet market, which is expected to exploit the GSC for the emerging 1200x1200 DPI registration challenge. Manufacturing equipment for single-pass digital printing faces an interesting challenge to deal with the growing productivity demand in combination with rising droplet registration accuracy. Although the administration speed of print-heads is still increasing, high throughput speeds and/or increased print resolution can only be achieved by using multiple heads in series. When using at least 4 different inks the distance between the first and last print-head can become more than 1 meter. The GSC can meet the challenge when the relative registration accuracy over such a distance must be less than 10 micrometer.
The GSC utilizes a stainless-steel conveyor belt in order to eliminate the mechanical (e.g. elasticity) properties of the substrate and prevent deformation of the substrate during transport. The substrate is clamped on the conveyor belt using vacuum technology. Many conventional steel conveyor belts and their steering systems cannot reach the previously mentioned accuracy target. During rotation, a belt will (always) translate axially with respect to the rolls of the conveyor due to limitations in belt and/or roll manufacturing (e.g. accuracy of the weld perpendicularity). The GSC uses movable segmented rollers to actively control and correct the position of the belt without deformation of the belt. This allows accurate control of transported substrates without deformation of the substrate over a large distance, which is ideal for the digital single-pass inkjet industry.

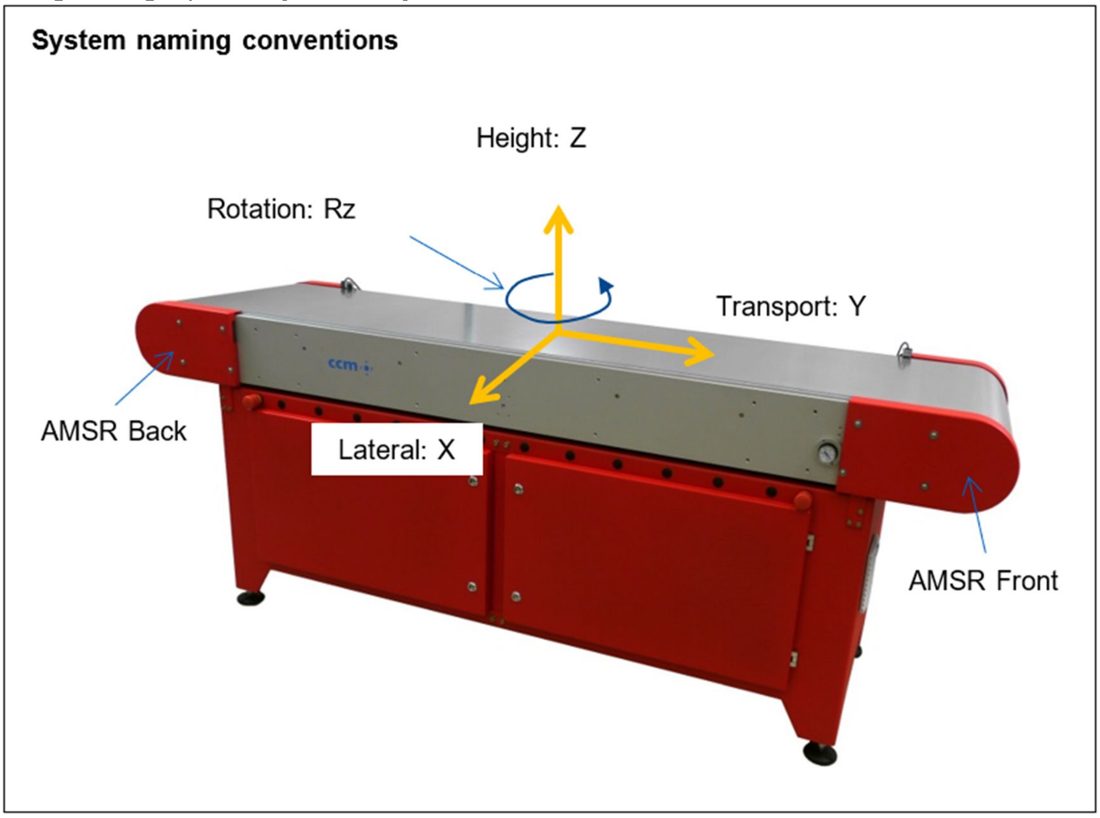


Figure 25 Pilot plant 1

The segmented rollers, called “Axially Movable Segments Rolls” (AMSR), allow continuous control of the 3 degrees of freedom of the belt (Y, X & Rz). The belt position in transport direction is controlled by means of rotation of one of the AMSRs using a brushless AC motor. The belt position in X and Rz direction is corrected by moving individual segments of the rolls in axial direction of the roll based on the belt position measured with 2 “belt edge sensors”. Reluctance actuators are used to position the axially movable segments of the rolls. Each roll has two pairs of reluctance force actuators for segment manipulation. One pair is used to position the segments of the AMSR in contact with the belt. The other pair of reluctance actuators is used to actively position an element back to its 'centre' position once per revolution when the segment is not in contact with the belt.

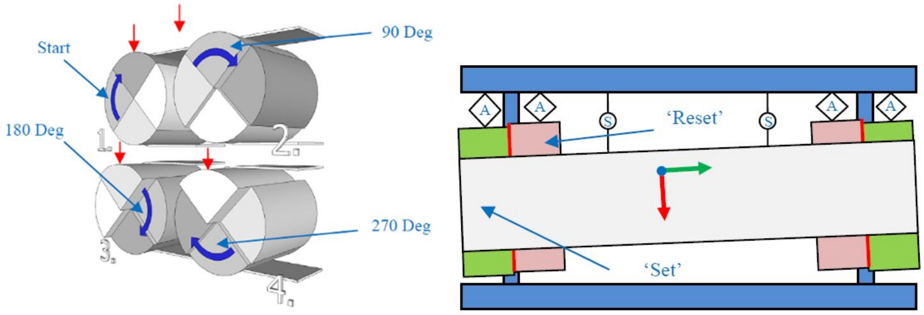


Figure 26 Principle of steering with AMSRs. "A" represents a pair of reluctance actuators, "S" represents a belt edge sensor.

The relation between force, distance and current is strongly nonlinear for reluctance actuators. Contactless sensors are therefore implemented to measure the gap between the reluctance actuators and the actuated segments. These measurements are fed back to a controller to calculate the current required to exert the desired force on a segment as function of the measured distance. This requires an extremely responsive off-the-shelf current amplifier, especially at higher belt speeds, which does not currently exist in the market. Thus, the performance of the GSC is limited. Furthermore, the existing computing platform (only) employs a single core of the multiple cores available. This limits the sample-time (it is a discrete system) and therefore the responsiveness and performance.

<u>Properties of the GSC (pre-I-MECH):</u>

- Total degrees of freedom to control: 8
  - Belt position X
  - Belt rotation Rz
  - Belt position Y
  - 4x AMSR element position
  - Vacuum pressure
- Total number of actuators to control DoFs: 10
  - 2 pairs of reluctance actuators per AMSR, 2 AMSRs per GSC
    - Used for belt steering in X and Rz
    - Each pair of reluctance actuators forms a single SISO axis, because a single reluctance actuator can only exert force in one direction
    - Each reluctance actuator is driven by a 1-phase amplifier
  - 1 Brushless AC motor inside one of the two AMSRs
    - For belt movement in Y (transport direction)
    - Brushless AC motor is driven by a 3-phase amplifier
  - 1 AC induction motor
    - For vacuum generation
    - Frequency controlled , based on vacuum measurement
    - Induction motor is driven by a 3-phase amplifier
- Total number of encoders/position sensors + vacuum sensor: 12
  - Belt position measurement in transport direction – Y:
    - 2 SinCos encoders etched on the belt (fused into 1 encoder signal)
    - 1 SinCos encoder on the motor axis used for commutation of the brushless AC motor
  - Belt position measurement in lateral direction – X & Rz:
    - 2 belt edge sensors, possibly replaced by 2 high speed belt edge camera's as part of the I-MECH project
  - Distance between AMSR segments and actuator:
    - 2 gap sensors per AMSR are used to measure position of the AMSR elements in contact with the belt

        - 1 gap sensor per AMSR is used to measure the position of the AMSR element not in contact with the belt
    - Vacuum pressure
        - 1 vacuum sensor
    - All 9 sensors are connected to analog differential inputs sampled at the position loop sample rate
    - The encoder outputs are sampled at very high speed (400 kHz) and extrapolated to ~20 MHz for usage by the industrial printing systems in which the GSC is integrated.

Requirements:
Functional requirements: Requirements I-MECH control & simulation framework
- It shall be possible to setup simultaneous multi-client connections with the control layer (e.g. client 1: control of GSC via a scripting interface, client 2: user interface, client 3: start/stop logging/tracing for diagnostics).
- It shall be possible to trace all signals and parameters (including signals from the current loop) and communicate traces/signal/parameter values semi-real-time to upper layers of the control system.
    - Decimation of tracing shall be configurable.
    - It shall be possible to easily configure which signals/parameters are available for tracing/observable from upper layers of the control system based on access level (e.g. operator/service/developer).
    - It shall be possible to trigger tracing of signals based on a selectable signal level/rising edge/falling edge/external trigger
    - It shall be possible to trace at least 20 signals simultaneously at 10 kHz
    - It shall be possible to acquire data continuously (data acquisition in chunks)
    - Multiple, synchronously sampled signals shall be guaranteed to be from the same controller time step
- All controller related parameters shall be changeable online.
    - It shall be possible to configure which parameters are changeable online.
    - It shall be possible to define parameter access levels (e.g. operator/service/developer) and define visibility of parameters depending on access level
    - It shall be possible to define parameter limits (min/max)
- A toolbox 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)
        - 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:
        - Bandwidth
        - Sensitivity
        - Stability margins
        - Settle time, rise time, overshoot
        - Noise/disturbance levels
    - Functions to easily design an IO-test/self-test
    - Functions shall be accessible via a scripting interface
- Tools shall be available to assess robustness of controllers and optimize the robustness of controllers:
    - Based on (identified) plant models
    - The assessment shall take time/position varying behaviour into account
        - Control algorithms shall be able to deal with load variation (e.g. large changes in friction of the belt and number of segments of ASMRs in contact with the belt)
- It shall be possible to implement custom homing/commutation strategies
- The control framework shall support the use of look up tables (e.g. for calibration of the reluctance actuators)
- The framework shall support MIL/SIL/PIL/HIL testing:
  - The framework shall offer a communication interface extensible to support simulation environments for both 1D as well as 3D simulation of plant physics and simulation of logic behaviour. E.g. support for simulation via Simulink, 20-sim and Siemens NX/Simcenter.
  - It shall be configurable which modules are simulated and which modules are real hardware during HIL testing.

    - Real time execution of models shall be possible during PIL/HIL testing
- It shall be possible to implement state machines to control behaviour of axes and the system (e.g. disabling & enabling, homing, etc.).
- Logging and reporting of errors and warnings shall be available, including errors and warnings from the OS & communication abstraction layer
    - Logging shall be readable (usage of enumerates instead of error numbers, so no “error 3” messages)
    - Logging shall be detailed, including: time stamp (when), module (who), event (what), cause (why)
- The I-MECH control and simulation framework shall support multi-core computing (for scalability).
    - It shall be possible to assign execution of algorithms to specific cores.
    - It shall be possible to assign algorithms to specific cores based on priority (e.g. to separate high priority control loops from low priority communication tasks).
- The framework shall be able to deal with algorithms that pre-process signals being exchanged with control algorithms/control building blocks, for example to:
    - Linearize the response of sensors & actuators (e.g. behaviour of reluctance actuators)
    - Optimize encoder signals

<u>Functional requirements: Requirements control algorithm building blocks</u>

- I-MECH control algorithms shall be suited for automatic self-commissioning (including auto-tuning).
    - Auto-tuning shall be available for both feedback loops as well as feedforwards.
        - Tuning shall be based on identified plant models/analytic models.
        - System identification and auto-tuning of MIMO systems shall also be possible.
    - Auto-tuning functionality shall be able to optimize both for robustness and performance.
        - Optimization criteria shall be configurable.
    - It shall be possible to configure at least position, velocity, acceleration/force/torque/current limits + tuning order for auto-tuning algorithms (applicable to multiple axes)
        - Implementation of automatic self-commissioning could be a toolbox with auto-tuning functions that can be called via a scripting interface
- Control algorithms shall offer the capability for gain scheduling
- The platform shall offer a trajectory generator algorithm, capable of:
    - Electronic gearing
    - Inverse kinematics
    - 3rd order, polynomial & sinusoidal setpoint generators
    - On the fly (smooth) trajectory updates
    - Position/velocity/acceleration/force setpoint modes
    - Jogging mode
    - Multi-axis/multidimensional setpoints
        - Synchronization mechanisms between setpoint generators
- The platform shall offer multiple methods for feedforward control:
    - Options for static/position/velocity/acceleration/jerk/friction feedforward control
    - Capability for model-based feedforward using model inversion
    - Feedforward control by taking into account cross couplings
    - Wish: Input shaping
    - Repetitive control / iterative learning control algorithms.
        - Algorithms shall include methods to quantify the performance of the algorithm (preferably in the frequency domain).
        - Algorithms shall be able to compensate for disturbances with frequencies above the bandwidth of the feedback controller.
        - Algorithms shall be able to cope with online changes of feedback controller settings.
        - It shall be possible to identify plant models used for learning control automatically when required.
        - The implementations shall be able to deal with variations in belt speed without requiring to re-learn all periodic disturbances.
        - Algorithms shall include a method for optimization of the learning gain/learning rate.
        - Relevant parameters shall be available for condition monitoring functionality.

    - Wish: a learning algorithm that takes changing plant dynamics into account (due to changes in number of rolls in contact with the belt)
    - Wish: a learning control taking MIMO cross coupling terms into account
- The platform shall offer multiple feedback control algorithms, including:
  - 'Standard' PID control algorithms for SISO axes, implementing:
    - Anti-windup
    - Integrator reset
    - Notch/biquad/lowpass/highpass filters
  - MIMO feedback control algorithms
    - A method shall be available to translate a MIMO controller to individual PID controllers for all axes and cross-couplings, such that manual modification of the controllers after controller synthesis using optimization algorithms remains possible via well recognized control structures
- Wish: Observers

Performance requirements

- Current loop
  - Current loop control bandwidth > 500 Hz
  - Current loop sample rate > 32 kHz
- Position control loop
  - Position loop sample rate > 10 kHz
  - Jitter (sample time variations) < 5 µs
  - It shall be configurable which clock will be used as "master clock" to which all other clocks in the system are synchronized (e.g. the clock of the central industrial PC, the clock of one of the EtherCAT slaves, the clock of a camera system, other external clock signal…)
- Communication with layer 1 sample rate> 10 kHz
- Desired accuracy improvement at a belt speed of 2.5 m/s > 30%
  - Current performance at 2.5 m/s:
    - Lateral tracking precision < 15 um (3σ)
    - Belt velocity stability in transport direction < 8 mm/s (3σ)

Implementation requirements

- Control algorithms shall be designed & customizable in Matlab Simulink
- Centralized control loops/I-MECH control framework shall be executed either on an X86 platform or on an SoC platform equipped with an FPGA (e.g. a Xilinx Ultrascale platform).
  - The platform shall support an EtherCAT fieldbus (for communication with layer 1 devices)
  - The platform shall support Ethernet communication (for communication between layer 3 and rest of the world)
- The control platform shall implement a hypervisor-like solution to allow execution of a non-real-time operating system besides the real-time operating system on the same hardware platform (e.g. for user interface, offline data processing, implementation of the system behaviour layer, condition monitoring and logging for predictive maintenance, etc.).

Other requirements to be evaluated:

- Both centralized and decentralized control structures;
- Residual oscillation compensation block;
- Vibration compensation block;
- Gain scheduling block;
- Observer block;
- MIMO axes management block;
- On the fly trajectory update block;
- Input shaping block;
- Jerk feedforward block;
- Advanced feedforward using inverse model block;

- Advanced feedforward that takes into account cross couplings;
- Iterative learning control block;
- Predictive learning control block;
- Support for tracing all the signals (included current)
- Auto-tuning block.

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

The 12-inch wafer stage is part of the Nexperia ADAT3 pick-and-place platform, used in the assembly of semiconductors. The wafer stage positions a diced (sawn) silicon wafer with semiconductor products (dies). These dies are picked up by the machine and transferred to a package, tape or other carrier.

Pick-and-place in semiconductor assembly requires an accurate alignment of the semiconductor component and pick-up tool at a fixed pick-up position. For cost reasons, the system architecture is classical where moving modules are mounted onto a rigid machine frame. The frame serves as metrological reference as well as the force frame absorbing the reaction forces. From accuracy point of view, this set-up leads to crosstalk between servos axes which requires careful dynamical design to assure individual axes are performing within tolerance budget. The motion control of the 12 inch wafer stage must take into account the internal dynamics of the stage and dynamics from the surroundings in order to reach micro-meter accuracy and repeatability.

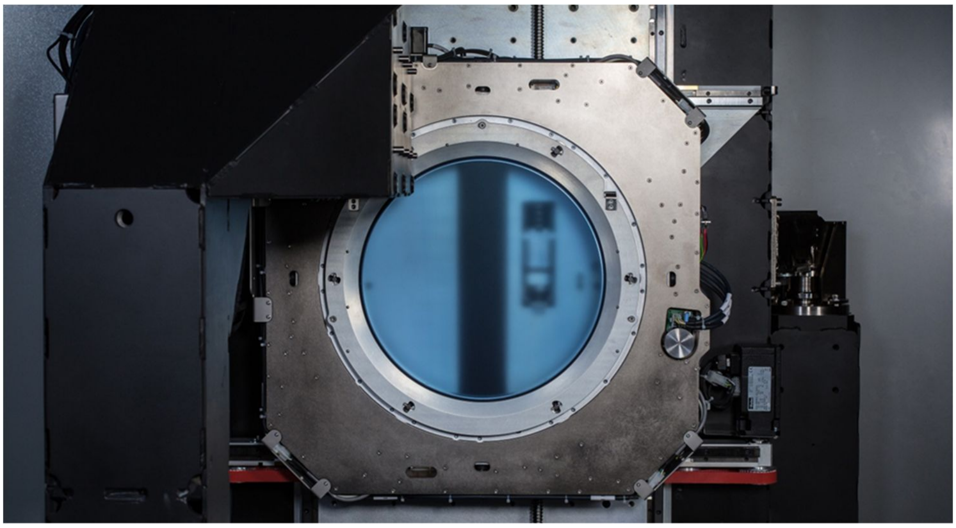
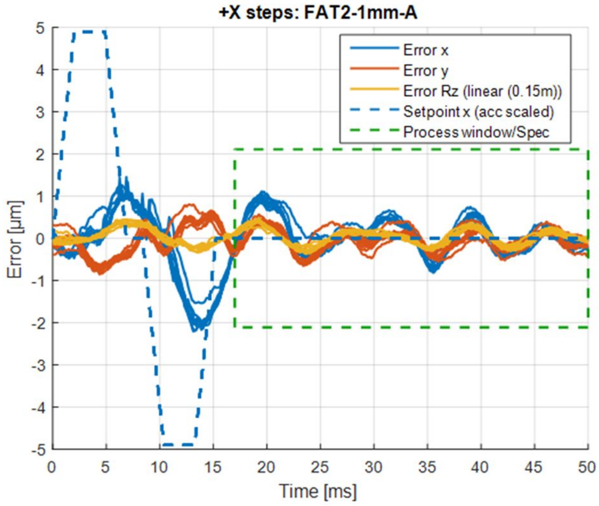


Figure 27 Pilot plant 2

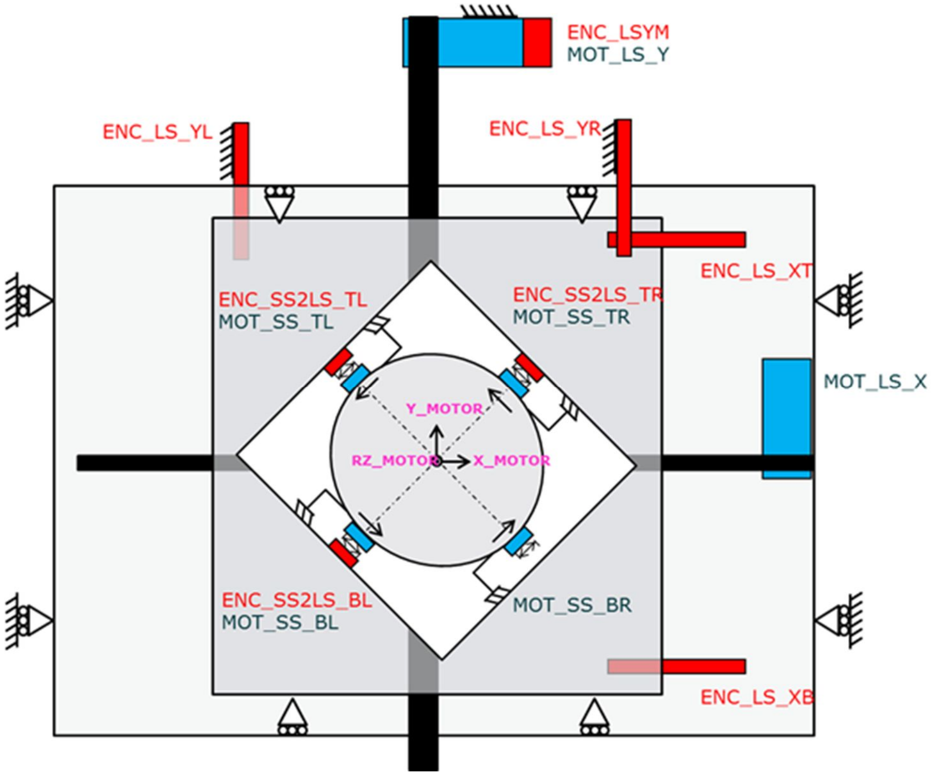
The 12 inch wafer stage is designed to operate within a 50ms machine cycle (corresponding to 72000 products per hour). The system features a MIMO-controlled short-stroke-long-stroke stage with a patented drive solution designed to achieve high-accuracy positioning (+/-2um) of a 9kg wafer stage in extremely short (1mm) set-points of 17 ms. The figure below shows typical setpoints, settling to within 2um within 17ms.



A well-tuned feedforward controller is required in order to be in position in time and to prevent settling. Tuning the feedforward parameters of this stage, and maintaining tune over time, requires adaptive control strategies to cope with machine to machine variations and time dependent behaviour, e.g. friction, motor constants, wear, temperature fluctuations.

The following schematic gives some insight into the encoders and actuators used for the 12 inch wafer stage. Some features are:

- The long-stroke (LS) stage of ~110 kgs is positioned by means of X and Y drives (a spindle drive MOT_LS_Y and a belt drive MOT_LS_X). It moves on X and Y linear guides and its position is measured by 4 encoders in the X-Y plane (ENC_LS_Y and ENC_LS_X). Two encoders are used per direction in order to ignore the rotation of the stage.
- The short-stroke (SS) stage of ~9kgs is suspended in the LS stage on leaf springs and is positioned by means of 4 voice coil motors (MOT_SS_). The four actuators are one more than strictly necessary, allowing over-actuation to be used to avoid exciting SS internal dynamics. The LS position is measured by 3 encoders in the X-Y plane (ENC_SS2LS_) to determine X, Y locations and rotation around the Z axis.



Requirement summary:

Performance

- Setpoint distance: 1mm (third order setpoint with acceleration and jerk limits)
- Setpoint duration: 17ms including settling time
- Setpoint settling: +/-2um
- 20 kHz control sample rate
- Feedback from 8 encoders and 6 actuators

Functional

- Adjustable control algorithm
- 3rd order setpoint generation with on-the-fly trajectory updates
- Jerk/Voltage feedforward in the amplifier
- Iterative learning control to achieve perfect setpoint tracking
- Condition monitoring: with adaptive feedforward (e.g., ILC) the system may keep performing very well until something fails catastrophically. Hence, changes in feedforward parameters may need to be monitored for predictive maintenance.
- Automatic tuning of controllers and feedforwards
- Support for MIMO control (feedback and feedforward) taking into account coupling, position-dependent dynamics and thermal effects
- Gain scheduling for position-dependent dynamics
- Model-based methods for design, diagnosis and system identification (possibly models for unpredictable wafer friction and relaxation), including robustness assessement of controllers on the basis of identified models.
- Drive condition monitoring
- Support for tracing all signals, including current-loop, in real-time and for on-line changing of parameters
- Predictive modelling: optimizing the performance of the long-stroke-short-stroke stage for any point-to-point move (subject to short-stroke limits)

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

A significant part of IMA's business is designing and producing packaging machinery for the food and pharmaceutical industry. IMA's top-level machines are best worldwide when it comes to speed, flexibility and product quality. In order to achieve such an outstanding result IMA has developed, over the last 30 years, its own control techniques and strategies. Among them, the most advanced is the control system adopted for both the machines producing tea bags and the ones filling vials for the pharmaceutical industry.

The control system is made of a number of different layers, each responsible for a specific task.
At the bottom lies an x86-based Industrial PC with PCI and PCI-e peripherals used for hosting fieldbus masters (e.g. Ethercat, Powerlink, CanOpen). On top of that, a hard real-time OS ensures that latency and cycle-times requirements are met, thus allowing more precise product handling and more comprehensive quality control checks. The top layers are the motion control, responsible for generating the motor setpoints and the verification of the encoders feedback, and the machine controller which orchestrates the overall machine subsystems.

Requirement summary:
Performance
- Cycle time: 500µs
- Jitter: 1% of cycle time
- At least 100 Axes

Functional
- Intel x86 platform, in order to be able to re-use the existing software layers
- VxWorks 6.9.x support
- In Multicore platforms, minimize ad-hoc source code creation
- In Multicore platforms, efficient inter-core communication mechanisms

5.4 Big CNC - Smart machining tools and milling machines (Pilot 4 - CORREA)

This Pilot is not involved in WP4.

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

Medical manipulators, especially the ones used for interventional treatment of patients generally do not belong to the realm of highly dynamic systems. The Philips manipulator, used in this pilot is no exception to that. The medical field does, however, have control challenges. These are sourced by the following main three system features:

1. Repeatability: Certain imaging procedures require multiple movements where the repeatability is of prime importance.
2. System variations: A medical manipulator is by nature a rather voluminous construction where large masses are propelled by small motors and serious gearing. These components introduce variations in system behaviour, often with a non-linear connotation like friction, backlash and play. Because the servo systems in such a manipulator are used intermittently during a medical intervention (typically, a motor stands still for 95% of the time), the system actually never reaches an operating temperature that is normal for drive train components like motor and gearbox. This leads to additional non-linear behaviour of e.g. lubricants, seals and to variations in the force constant in motors. Fretting corrosion is a point of attention.
3. Safety: The majority of (complex) position servo applications are employed in manufacturing equipment of goods (e.g. welding robot, pick & place machine), or in goods themselves (e.g. DVD player). Servo system failure in manufacturing equipment mainly leads to reduced productivity, thus having mainly financial

consequences. Safety aspects in those environments are mostly limited to ensuring that operators are not negatively affected. Servo failure in mass produced goods themselves generally do not have serious consequences. Many servo systems in the medical field (but also e.g. in the military field), however, act on human beings (e.g. X-ray manipulator, surgical robot). Medical system servo system failure may thus immediately influence the health of a human and also of the operator of the device. More explicitly, this concerns patients and medical personnel and this is a sensitive environment when it comes to safety. The medical working environment therefore imposes measures that stringently enforce product safety. This is also reflected in international rules and laws.

A common approach in the total safety concept in a medical device is to detect position- or force violation and stopping the device along a predefined path. This requires an accurate knowledge of the (dynamic) behaviour of the medical device. Whereas feed forward techniques in non-medical servo applications are used to improve servo performance, in the Philips manipulator they are used to generate a kind of nominal force for moving the mechanics (and thus leading to a nominal trajectory error). Deviations of that nominal force are then used as an indicator for abnormal conditions (e.g. collision with the environment, equipment failure). This approach only works when the model is accurate enough (despite production variations) and the feed forward compensation is sufficiently dimensioned.

So, although not highly dynamic, medical manipulators are continuously growing in complexity, nowadays requiring control strategies historically only found in the industrial robot realm. These include kinematic solvers and extensive parameter constraint handling. The absence of sensors in the location the repeatability is sought, adds complexity.
The combination of all this points into the direction of model-based control, predictive control, robust control, sliding mode control, ILC, adaptive control and repetitive control. Of these, model-based control suits the Philips medical manipulator best and this is the main focus for pilot 5.

Control environment implementation.

Next to the control challenge itself, the control environment implementation in medical systems requires special attention that differs from normal industrial applications. Apart from stringent EMC and safety requirements (not part of I-Mech Pilot 5 but nonetheless a matter that inflicts constraints to the solution space), the use of software and software tooling is also strict. We can roughly discern three levels of control environment implementations:

1. Tailored to fast control adaptation. This relates to environments where control algorithms are changed frequently. One may typically think of the environments featuring xPC or dSPACE type of implementations often with rapid prototyping platform support. Many of these closely relate to Mathworks products for algorithm development and analysis. These systems are great for exploring and testing because of their flexible architecture, but are not suited to be shipped as part of a medical product, produced in quantities over hundreds per year, each having a fixed set of control settings.

2. Tailored to flawless functional integration. This refers to environments where code is automatically generated by modelling environments. The code is then used as part of the total system application software code. In a medical environment it is required that proof is supplied that such auto-generated code is safe, and that the tooling that generates it is validated. Using code generation is attractive in complex systems but using it in medical manipulators is somewhat new, including the uncharted approach to this matter of legislative- and international compliance bodies. Moreover, it is preferred to have a testing environment that is the same as, or at least highly comparable to, the implementation in the final product itself.

3. Tailored to the application. This relates to the software that is created specifically for the application and actually is run during operation of the system.

The third environment is always present as the product would not exist if it weren't. The first environment is not suited for the Philips medical manipulator architecture and business model. The second control environment implementation is developed in Pilot 5 (see Figure 27)

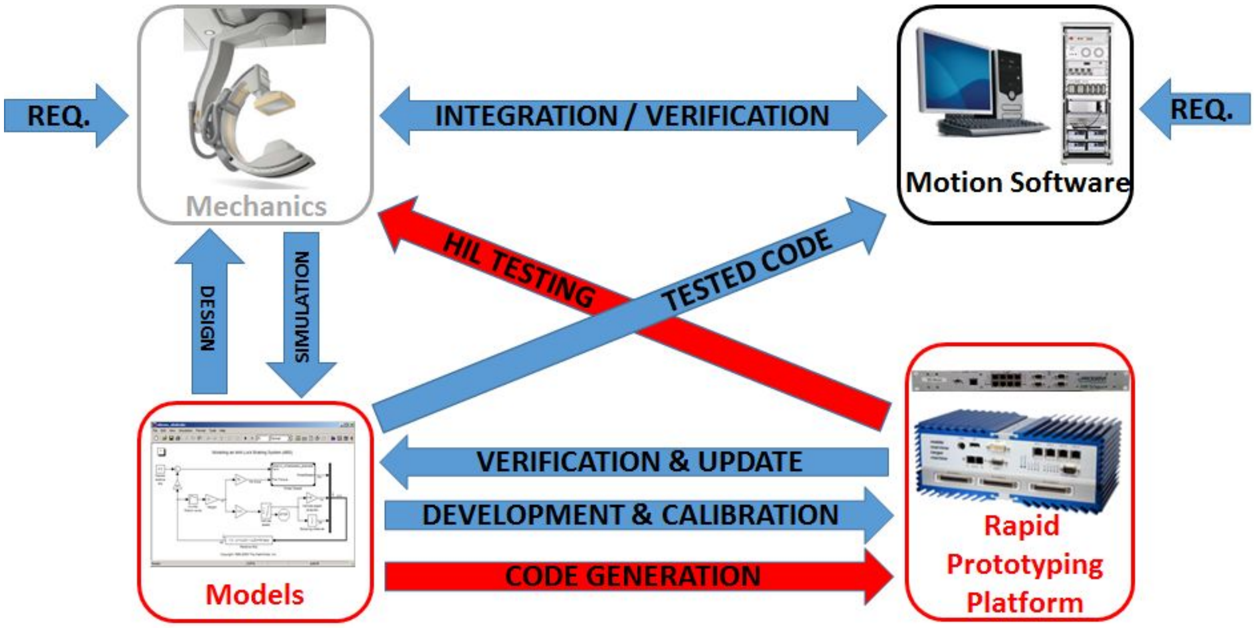


Figure 27 Main approach in Model Based Design. Elements in red are part of Pilot 5

Pilot 5 thus will demonstrate how a model subsystem is controlled, using a rapid prototyping environment that closely matches with the environment that controls the medical manipulator. For this purpose, the intention is to have the following developed:

1. A model of a subsystem of the medical manipulator
2. Extraction of a controller from the model as part of a feedforward control strategy
3. Building the algorithm in a prototyping environment and running it in specific rapid prototyping environment, using automated code generation whilst honouring constraints set by the environment and the application
4. Subsystem parameter tuning or calibration
5. Assessment of adaptive control needs
6. Testing the result and comparing it with current methods used

<u>Environment constraints.</u>
Model calculation time: To be determined
Fieldbus speed : < 1 [kHz]
Control loop bandwidth: < 100 [Hz]

<u>Requirements</u>:

- A model of a subsystem of the medical manipulator
- Extraction of a controller from the model as part of a feedforward control strategy
- Building the algorithm in a prototyping environment and running it in specific rapid prototyping environment, using automated code generation whilst honouring constraints set by the environment and the application
- Subsystem parameter tuning or calibration
- Assessment of adaptive control needs
- Testing the result and comparing it with current methods used

- Other requirements:
    - Feedback + feedforward adjustable control algorithms;
    - 5-16 axes control;
    - Iterative learning control block;

- Repetitive control block;
- Runt to run control block;
- Auto-tuning block;
- Time + frequency real time domains analyses block.

6 I-MECH Demonstrators requirements and specifications

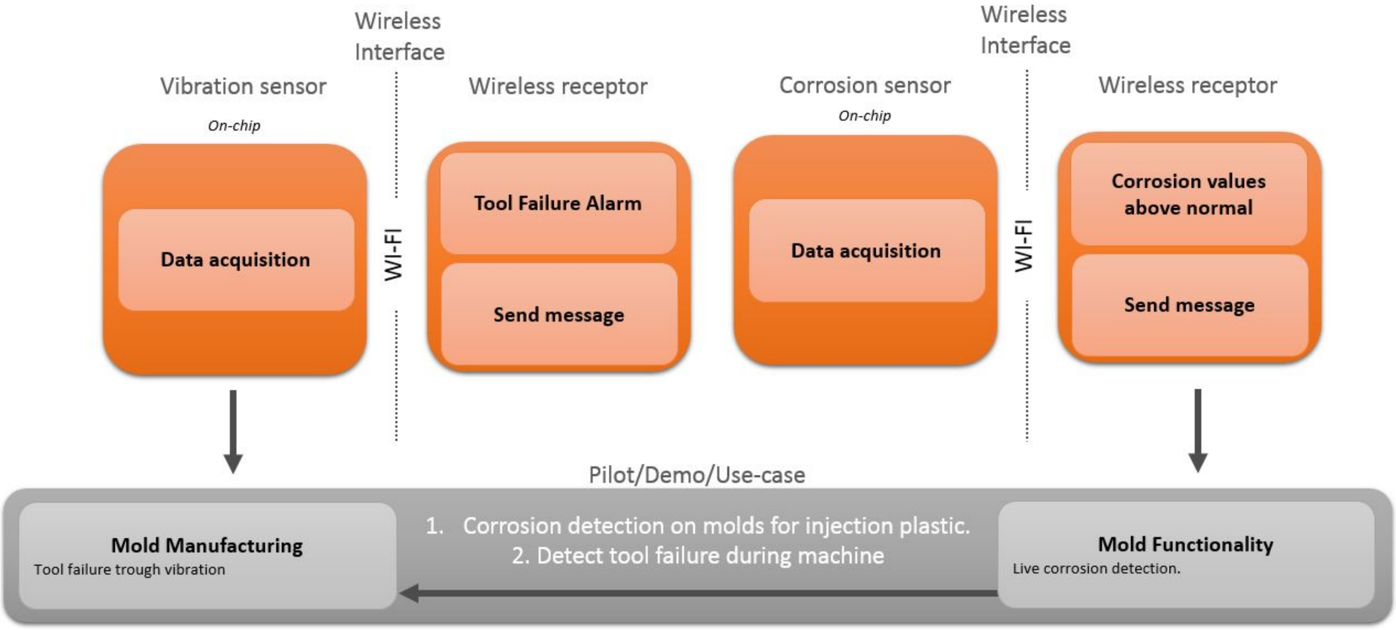
6.1 Insulin Delivery System (Demonstrator 1 – J&J Vistakon)

Johnson & Johnson Vision Care (JJVC), a division of Johnson & Johnson Medical Limited, produce solutions to improve vision and vision care, including the manufacture and distribution of contact lenses. JJVC will provide a clean production environment to demonstrate the I-MECH platform.
The I-MECH demonstrator will focus on the MagneMotion Quickstick linear synchronous motor (LSM) product transfer layer used in JJVC. LSM provides an intelligent and highly controlled product transfer layer as part of the contact lens manufacturing processes.
The demonstrator will apply smart sensing and condition monitoring developed in I-MECH to improve product transfer system performance and implement a predictive maintenance platform to monitor vehicle condition and to address deterioration before failure. This will be achieved with the implementation of wireless sensors to monitor electro-mechanical parameters and the use of a data analysis platform, developed within I-MECH, to contextualise the data collected.
There is also an opportunity within I-MECH to create a digital twin of the system to model and analyze the existing process flow and system design in order to optimize output and provide a digital reference for the real time performance.
Finally, as part of I-MECH, an energy recovery solution will be developed to harvest electro-magnetic energy produced in the repeated acceleration and deceleration of the LSM vehicles to decrease the energy footprint of the manufacturing system.

Requirements:
- dual loop feedback
- monitoring of MagneMotion carrier parameters
- monitoring of carrier degradation
- monitoring of energy consumption
- data for predictive maintenance
- data for overall system efficiency optimization
- CPS for remote optimization of the performance of the line
- compatible with MagneMotion /AB PLC Control
- centralised control
- ethernet IP communication protocols

6.2 Injection mould tool (Demonstrator 2 - EDILASIO)

This demonstrator will be used to evaluate wireless sensing and smart sensing building blocks, amongst others, by creating a concept mold with these kind of smart functions, giving to the molding tool a bigger role in terms of functionalities.
It will be divided into 2 parts: Mold manufacturing, and Mold functionality. The first one will be focused on implementation of sensors capable of giving a response during the manufacturing of mold, such as tool failure. The second one will be focused on the functionality of mold, were it will be implemented sensors/actuators capable of interact and acquire data from the tool. In this case, a corrosion sensor that will give me a response on when maintenance should be done. The next diagram explains the main interactions.



<u>Requirements:</u>

Performance
- Depending on the application (still on study)

Functional
- Vibration data acquisition and feedback
- Corrosion data acquisition from mold tool and their predictive maintenance.

7 I-MECH Use Cases requirements and specifications

7.1 Drive for industrial applications (Use case 1.1 - GEFRAN)

Use case 1.1 is composed by two drives that control two independent axes of a portal crane.
Drives are part of ADV200 family and they are specialized for hoist and crane applications (ADV200HC). The GEFRAN ADV200 HC inverter offers, in a unique technological solution, all the basic requirements demanded by the most modern industrial hoisting systems. ADV200 HC can manage asynchronous motors with or without speed/position sensors (encoder) and it can control all the system movements, both hoisting and travellers (hoist, gantry, trolley). In addition, these drives allow the possibility to develop SW for industrial application by using commercial language based on IEC-61131-3 (PLC-OPEN) and to be connected with several field-bus systems (ProfiBus, CANopen, DeviceNet, EtherNet real-time, GDNet, EtherCAT, EtherNet/IP, ProfiNet).
To improve the performance of the hoist and crane applications, during the WP4 it is desired to develop:

- anti-sway strategies that can help the operators during the motion. The anti-sway could be available with or without additional sensors. External sensors can be accelerometers and/or gyroscopes. The anti-sway strategies can be active or passive and they can be implemented directly on the Gefran drives.
    - Passive antisway: based on the available information of drive, motor
    - Active antisway: with addition information with external sensor (accelerometer or gyroscope)
- an automatic tuning approach for both the position and the velocity control loops of the two drives. The automatic tuning approach should take into account the position, velocity and torque constraints superimposed by the user for the tuning process.
- a method to tune batteries of filters (notch, bi-quadratic, low pass, band pass, high pass, adaptive…) that can improve the control action of the drives.
- a method to monitor the information of the system.

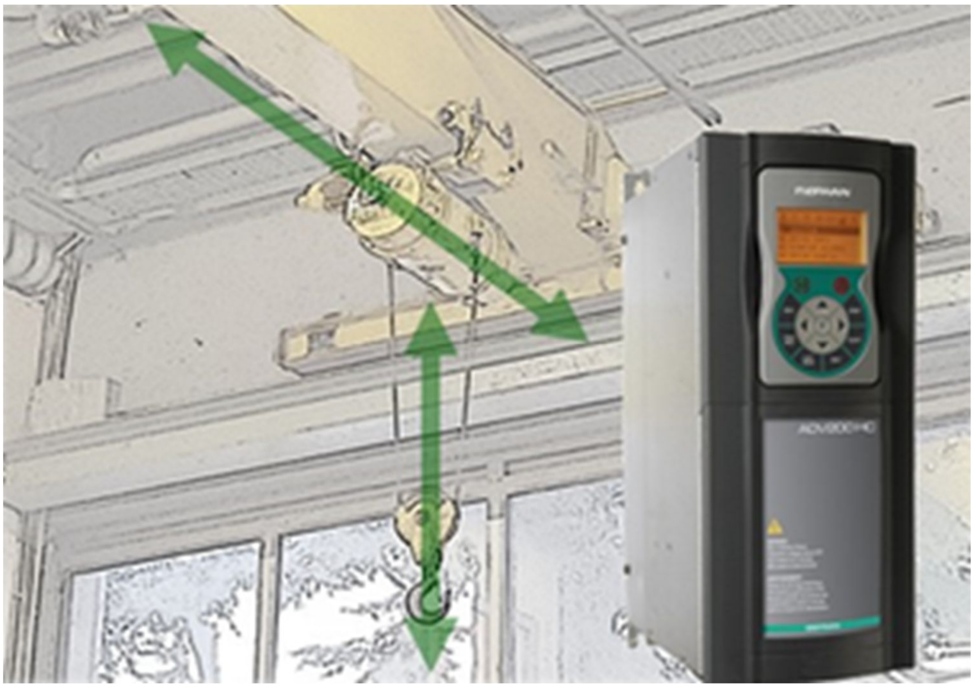


Figure 28 Use case 1.1

A list of generic requirements that can be added to the previous one is exposed hereafter.
Requirements:

- dual loop feedback block;
- online controller parameters change block;
- residual oscillation compensation block;
- vibration compensation block;

- multiple notch, adaptive filter block;
- anti-windup block;
- observer block;
- input shaping and/or model inversion block;
- advanced feedforward using inverse model block;
- iterative learning control block;
- repetitive control block;
- predictive learning control block;
- auto tuning block.

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

As fixed in the requirements table (in D 2.1) Fagor CNC already has some i-mech BBs implemented in proprietary means (mainly C with intrinsic functions for x86). The CNC can control at the same time up to four interpolation paths (channels). Independent axis can be at the same time commanded by the integrated PLC. The mechatronic problems described in the i-mech project are all applicable to machine tools. Some of the more common problems deal with BB7 and BB8, due to big steel masses moved at high accelerations that have usually low frequency structural vibration modes, with very low damping (steel), and varying frequency depending on axis positions.
The Fagor CNC is based on an x86 CPU architecture and this will be a requirement. Improvements of the actual hardware and software platforms as described in BB10 and BB1 are a natural result of the i-mech project.
But, as a real machine tool is usually not available for demonstration, some of the partners of the Spanish consortium will build a prototype where the important concepts of the BBs will be demonstrated under the specific requirements of this document.
For this, Use Case 1.2 will consist of a complete mechanical + hardware + software platform according to i-mech requirements. The complete system will be integrated by Rovimatica and consist at least of:
- Scara robot arm (4 degrees of freedom) with suitable motors and encoders.
- Motor drives developed and built by Ingenia as described in BB5 (D 2.3.5) with ethercat interface and at least 8KHz sampling rate (16KHz desired) and commanded in current/torque.
- Fagor CNC based in a COTS congatech x86 module with up to 4 cores and virtualization hardware according to BB10. (D 2.3.11).
- Accelerometers put near the effector end and connected to the CNC (desirable through ethercat bus at the same rate)
- Hypervisor running on that platform as required in BB11 (D 2.3.12) where at least a core will run windows 7 embedded, at least another core will run Fagor CNC kernel software and at least another could run control loops.
- This platform and hardware is chosen, (other than its availability) because it fulfils many of the problems experienced in mechatronic systems as explained in deliverable. 2.1:

  - variable inertia seen from the first axis depending on the position of the second axis, what calls for mimo and/or robust control, BB8 (D 2.3.9), depending on the approach taken to the problem.

  - kinematics transform between command (cartesian) and actuator (joint), and between actuator and acceleration feedback(cartesian or rotated by twist axis depending on configuration).

  - elasticity due to harmonic drive reductions. This calls for BB7 (D 2.3.8) in the form of feedback control both with observer or direct acceleration feedback. Working with observer, estimation of load can be done at the same time relating again with robust controller as required in BB8 (D 2.3.9).

  - depending on the final mechanics, different types of motors (standard, linear) with and without reduction. For standard PI loops, on demand auto-tuning of position and speed loop as of BB6 (D 2.3.7) will be demonstrated.

- BB9, iterative and repetitive control module as described in D 2.3.10 will be demonstrated in 2 forms: first as cogging feedforward controller, and second, for instance, as a specific implementation of a pick&place application.

Requirements referred to D2.3 other than general are:

2.3.5 BB5: High-performance servo amplifier:

- Current loop bandwidth > 2kHz (the above application)
- Current loop sample rate : >= 16khz.
- Commutation method: FOC with encoder, motor position sent to CNC in ethercat,
- Full synchronization of drive and CNC loops through ethercat

2.3.7 BB6: Self-commissioning velocity and position control loops
- At least for standard PI controllers.

2.3.8 BB7: Vibration Control Module:
- Load side feedback (to be decided in application) with its own kinematics
- Load detection (estimation algorithms based on model needed)
- Active vibration reduction with observer
- Active vibration reduction with load acceleration feedback (with its own kinematics)
- Automatic or semiautomatic tuning of model parameters.

2.3.9 BB8: Robust model-based multivariable control
- Multivariable control: at least a 2x2 MIMO matrix for axis q1 and q2
- Robust control based on model as in BB7

2.3.10 BB9: Iterative and repetitive control module
- Repetitive control, at least for cogging of motors
- Learning control, for pick and place or similar application

2.3.11 BB10: Control specific multi/many core platform:
- 2 cores minimum, 4 cores highly desirable
- Sample rate: 8Khz (16KHz desirable) in at least one core
- Sample rate: 1KHz (4KHz desirable) in another core
- x86 platform with hypervisor support

2.3.12 BB11: RTOS for multi/many core platform
- Program language: C (with intrinsic functions)
- General Purpose OS: Linux and Windows 7 embedded (not necessary at the same time)
- Hypervisor as required for “near 0” latency and the sample rate for BB10

Platform requirements
Requirements referred to implementation depend heavily on the control approach taken and the hardware platform decided.
- The system will be highly centralized. All the loops will run in the x86 platform (except torque/current and below).
- The connection to the drives will be Ethercat.
- System fully synchronized, all the loops and also between drive loops (pwm, current…) and speed, position and other loops.
- The BBs will be implemented as C routines for x86 (use of specific functions available, for instance SIMD instructions) in the execution platform.
- There will be a twin for every algorithm of the BBs in Simulink (desirable with the exact behaviour relating format, etc…)
- Frequency and execution order must be explicit for loops and blocks.
- This applies also to reading inputs, output application and updating of integrators. All these parts of control loops should be callable independently.

7.3 PAC based modular HW for machinery (Use case 1.3 - TECO)

Teco aim at producing upgraded motion control HW and bring it to the market after the project end. Specific control layer requirements are:

Specific Layer 4 control requirement	
Requirements	**Technical specification**
Trajectory generators	Single axis, Multi axis, Coordinated motion (according to PLCOpen standard), G code
Regulator structure	Configurable – default: Cascade (position, velocity, current), advanced: velocity, current feedforward, state controller, custom specified filters, centralized controller (robot dynamic compensation)
USB comm	For displays, disks or other peripherals
Wireless comms	For wireless sensors
EtherCAT communication period	8 kHz, configurable input shift time, in/out processing time <20 us in total at 64B per axis
MQTT	Development and possible integration of MQTT protocol
Processor board	Qualcomm Snapdragon, 4x64 bit core, 1.2 GHz, 1 GB DDR3 RAM, 8GB eMMC
Computer Vision	
Integration of High speed cameras	Measurement of object size and orientation, camera as a motion sensor
Applications	
Assumed applications	Assembly manipulators, CNC machines

Figure 29 Design of upgraded motion control HW produced by Teco

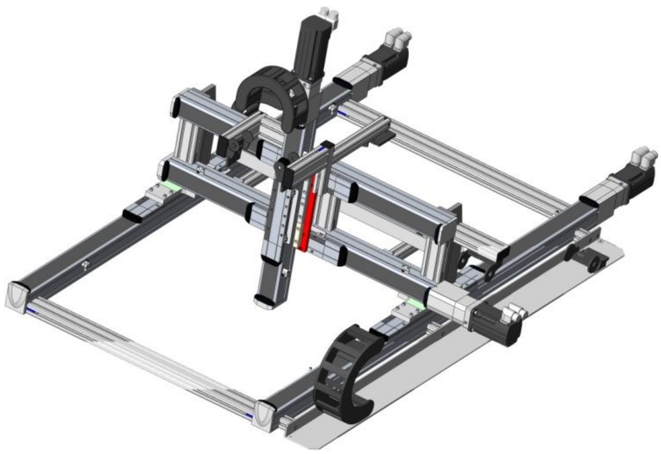


Figure 30 Robotic manipulator for testing TECO PACs

7.4 Space GNC systems through the use of robotic devices (Use case 2.1 – GMV A&D)

The use case defined for I-MECH is platform-art©, which is used for validation of space GNC systems at GMV. Platform-art© is composed of the following main subsystems

- Target: A mockup of the space vehicle mounted on a manipulator robot
- Chaser: A gripper mechanism simulating the active, chaser vehicle, mounted on another robot in turn mounted on a linear track. It can be equipped with a camera and/or a gripper device.
- Control system based on manipulators controllers (Kuka), DSpace (for simulation of dynamic behavior in space) and RTOS motion controller (for real time control of the system).

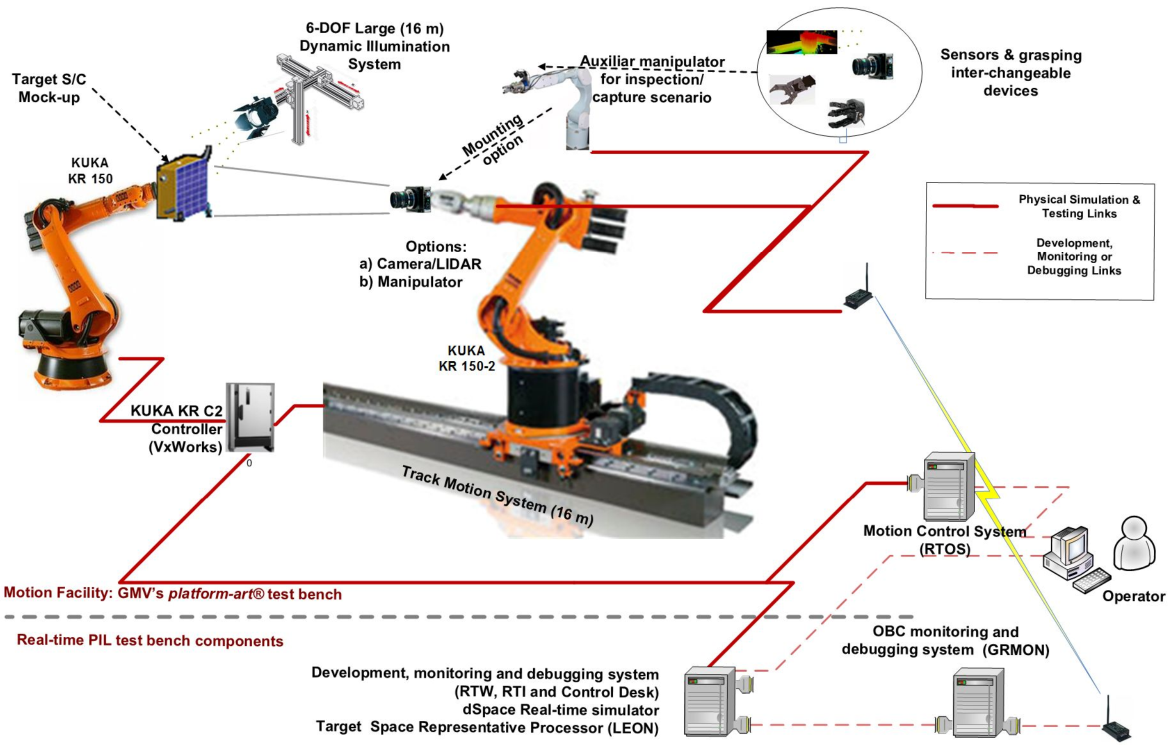


Figure 31 Use case 2.1 - platform-art© setup

The use case can be decomposed in two parts.

The most general use of platform-art© (the use case platform defined in I-MECH) is the simulation of activities that include the precise approximation between space vehicles or between a space vehicle and other bodies in tasks like rendez-vous, space debris removal, automatic assembly of large structures, etc. In order to simulate these tasks, a mockup of a space vehicle, asteroid, etc. is installed in an industrial manipulator (simulating the target) and a camera is installed in another similar manipulator (simulating the active, chaser vehicle). The final approximation in these applications requires a controller capable of implementing visual servoing.

Another use of platform-art© is the simulation of activities that imply contact between space vehicles. In order to simulate the dynamic behavior of the vehicles, a mockup of the space vehicle is installed in an industrial manipulator (simulating the target) and a gripper mechanism is installed in another similar manipulator (simulating the active, chaser vehicle). In these applications a controller capable of implementing 6-axis force control in the gripper (which is equipped with a 6axis load-cell) is required.

Current setup has two main lacks:

- Delays in the control of manipulators (due to Kuka controller) prevent accurate force control and put in risk the integrity of the manipulators and mockups when the kinematic chain is closed during contact operations
- Visual servoing is currently not available

In order to overcome these lacks, the following solution is proposed:

- A manipulator equipped with a gripper is installed on top of the platform-art© chaser manipulator, allowing full access to the parameters configuration and path planning
- Force control is implemented in this manipulator, allowing precise, fast control of the forces exerted on the target during contact.
- Visual servoing is implemented in order to allow precise approximation of the gripper to the target
- Both the platform-art© chaser manipulator and the force/vision-controlled manipulator are controlled jointly to allow for precise force control and visual servoing throughout the overall platform workspace.

Specific requirements

General control requirement	Possible related application	Bibliographic references
6 Axes Visual servoing	**platform-art©,** Approximation case	https://www.gmv.com/export/sites/gmv/DocumentosPDF/platform_art/PLATFROM__ENG.pdf
6 Axes Force control	**platform-art©,** Grasping case	https://www.gmv.com/export/sites/gmv/DocumentosPDF/platform_art/PLATFROM__ENG.pdf

The following table shows the requirements regarding visual servoing and force control.

Specific control requirement	Technical specification
Vision	
Interface (e.g., Cameralikn, USB3, GiGE, Firewire 800..)	Actual: SpaceWire, GiGE, USB3.0, Camera-Link Desired: + SpaceFibre and TSN/TTEthernet
Bandwidth per camera	Actual: Nominal 150-200 Mbps Desirable up to 400 Mbps Desired: 800 Mbps
Number of realtime streaming cameras	Actual: 1 Desired: 2
Resolution	Actual: 2048x2048 10-12bits and 1024x1024 10-12bits Desired: 2048x2048 10-12bits and 1024x1024 10-12bits
Max update rate	Actual: 5-10 Hz Desired: 10-50 Hz
High speed vision for quality inspection	No
High speed vision for visual servoing	Yes
Distance range:	< 2m
Speed range:	<1cm/s
Vision control tolerance	aprox 1cm (at 0.5m)
Vision control loop:	< 20ms (50Hz)
Force control	
6 Axis force control	Force control, Impedance control loop in a 6 DOF manipulator.
Sampling frequency	>200Hz 6 axis force control loop
Trajectory generation, inverse kinematics	Yes, 6 DOF manipulator
Learning control	YES (TBC), Predictive learning control, (TBC)
MIMO axes	6
Sensor data	6 axis load cell, motor current sensing
Force range:	<10N
Force control tolerance	<5%

Implementation requirements

The motion control for the force/vision-controlled device shall be implemented in a separate hardware allowing an easy and fast deployment and testing on the platform-art©.

<u>Other Requirements</u>:

- Current loop bandwidth > 2kHz.
- Current loop sample rate >= 16khz.

7.5 Open modular robotic arm (Use case 2.2 - ZAPUNI)

Open robotic modular arm will be designed as redundant 7DoF serial manipulator with advanced collaborative functions. The robotic arm is considered as fully open architecture regarding the design of the control system (robot controller) as well as the design of the compact actuators forming robot joints with own servo-drives. The main reasons for developing a fully open robot architecture in comparison with using standard robot architectures (industrial and collaborative robots) are as follows:

- Possibility to introduce non-standard kinematics, e.g. the robot under consideration will have 7 independent revolute axes – this concept makes possible to use benefits of redundant robot (dexterity, reduced footprint, etc.)
- The auxiliary sensors can be easily added (wide range of communication protocols can be managed)
- The low-level servo-drive control system can be set and tuned without restrictions steaming from standard servo-control drives (e.g. given regulator structure and tunable parameters).
- New advanced motion control algorithms can be included directly into the low-level control scheme with a short sampling period (e.g. feedforward/feedback algorithms for vibrations damping, e.g.)
- Advanced motion control algorithms, especially for collaborative robot control, safety system and intuitive robot motion learning can be integrated to fully open superior robot motion control system (robot controller)

The following table shows the overall robot requirements and regarding specific motion control system requirements:

Overall requirement	
Requirements	**Technical specification**
Payload	1 kg
Reach	< 1 m
Weight	< 20kg
Applications	Pick & Place, Gluing, Small part assembly, screwing, etc.
Human Guidance learning	YES
Force control	YES
Compliance behavior (impedance)	YES

Specific control requirement	
Requirements	**Technical specification**
Joint actuator low-level control	
Regulator structure	Configurable – default: Cascade (position, velocity, current), advanced: velocity, current feedforward, state controller, custom specified filters, centralized controller (robot dynamic compensation)
Sampling period of current controller	32 - 64 kHz
Sampling period of velocity/position controller	8 - 16 kHz
EtherCAT communication period	8 kHz, configurable input shift time, in/out processing time <20 us in total at 64B per axis
Trace and diagnostics	Ability to trace all signals in control loops over EtherCAT at full sampling rate (oversampling)
Robot kinematic control	
Number of axes	7
Actuator types	Revolute (geared motor)

Inverse kinematics algorithms	Default: direct control of redundant axis, advanced: joint limits/velocities limitation, obstacle overcoming, singularity overcoming
Force control / impedance control	
Force control	6 DoF (end effector F/T sensor), sensitivity < 1 N
Impedance control	End-effector dynamics can be prescribed (compliance regarding collaborative tasks)
Safety subsystem	
Collisions detection	Joint F/T sensors (own design), resolution < 2 N F/T sensor (commercial solution) in the robot base, resolution < 5 N
Vision	
Parts/workcells initialization	aRUco markers for robot homing in a technology process, accuracy: < 3 mm
Intuitive programming	
SpaceMouse end-effector guidance	Recorded data postprocessing (interpolation/approximation, velocity profile definition, etc.)
F/T sensor end-effector guidance	See Impedance control

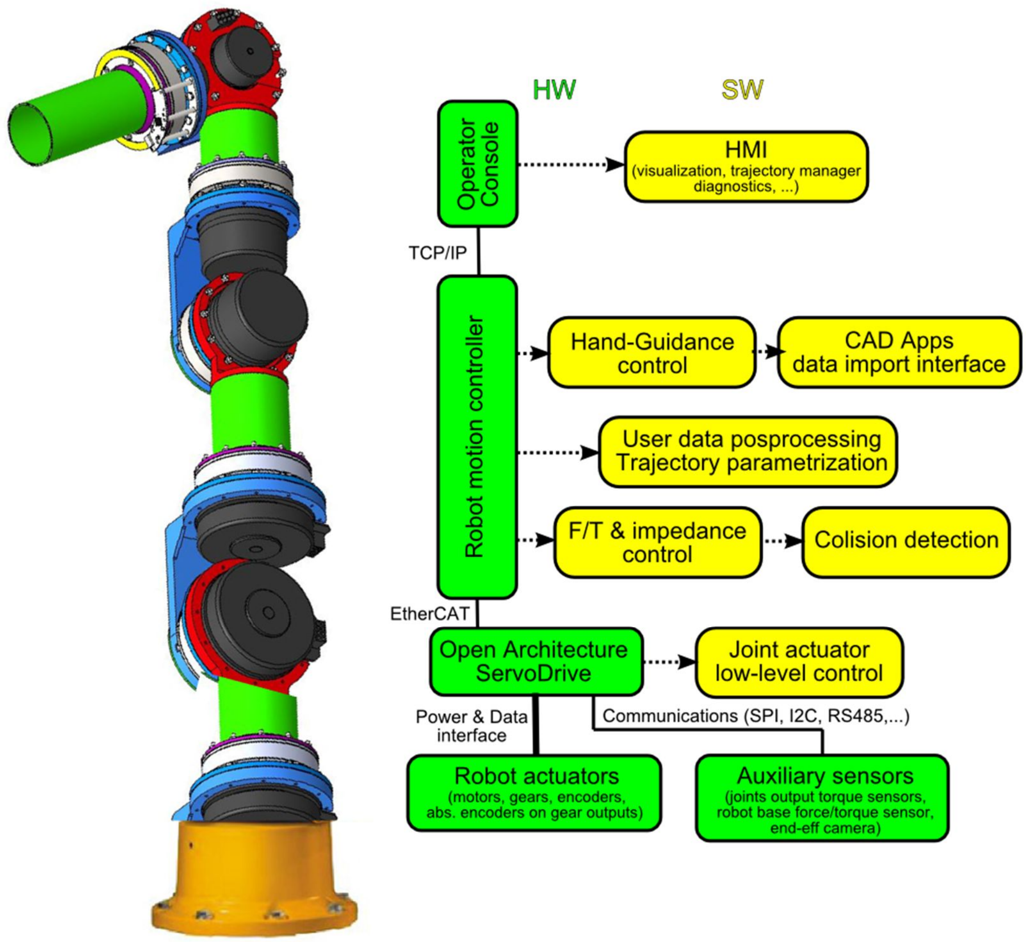


Figure 32 Use Case 2.2

8 Summary tables

In order to summarize the control requirements for the different pilots, demonstrators and use cases, and in order to clarify their relationships with the I-MECH Tasks and Building Blocks, the structured approach already followed in D2.3 is used in the following table.

The employed code legend is as follows (each requirement ID is prefixed with rq- (for requirement), the deliverable ID, in this case D4.1, and the kind of requirement):

- rq-D4.1-P: Performance requirements
- rq-D4.1-T: Technical requirements
- rq-D4.1-R: Realization requirements

Code	Description	Section in this deliverable
rq-D4.1-P.01	Stability	4.1.1
rq-D4.1-P.02	Overshoot	4.1.2
rq-D4.1-P.03	Set point tracking	4.1.3
rq-D4.1-P.04	Disturbance rejection	4.1.4
rq-D4.1-P.05	Noise attenuation	4.1.5
rq-D4.1-P.06	Bandwidth and sampling frequency	4.1.6
rq-D4.1-T.01	Signal filtering	4.2.1
rq-D4.1-T.02	Friction and inertia estimation	4.2.2 – 4.2.3
rq-D4.1-T.03	Feedforward compensation	2.3 – 2.6
rq-D4.1-T.04	Periodic harmonic disturbance compensation	4.2.4
rq-D4.1-T.05	Repetitive disturbance compensation	4.2.4
rq-D4.1-T.06	Oscillation compensation	4.2.6
rq-D4.1-T.07	Vibration compensation	4.2.7
rq-D4.1-T.08	Robustness to dynamic variations	4.2.8
rq-D4.1-T.09	Set point shaping	4.2.9
rq-D4.1-T.10	Decoupling	2.7
rq-D4.1-T.11	Performance assessment and fault detection	4.2.11
rq-D4.1-T.12	Homing strategies	4.2.12
rq-D4.1-R.01	Model based design	4.3.1
rq-D4.1-R.02	Identification	4.3.2
rq-D4.1-R.03	Automatic tuning and adjustability	4.2.10 – 4.3.3
rq-D4.1-R.04	Control prototyping	4.3.4
rq-D4.1-R.05	Implementation	4.4

Code	Pilots, Use cases, Demonstrators	Task	Building Block
rq-D4.1-P.01	P1, P2, P3, P5 D1, D2, UC1.1, UC1.2, UC1.3, UC2.1, UC2.2		BB6
rq-D4.1-P.02	UC1.1,		BB6
rq-D4.1-P.03	P1, P2 UC2.1, UC2.2		BB5, BB6, BB8
rq-D4.1-P.04	P1,		BB6
rq-D4.1-P.05	P1,		BB6
rq-D4.1-P.06	P1, P2, P3, P5, UC1.2, UC2.1, UC2.2		BB4, BB6
rq-D4.1-T.01	UC1.3		BB6, BB8
rq-D4.1-T.02	P1, P2	T4.3, T4.4, T4.5, T4.6	BB6
rq-D4.1-T.03	P1, P2, P5	T4.4,	BB6
rq-D4.1-T.04	P1	T4.4,	BB9
rq-D4.1-T.05	P1, P5 UC1.2	T4.6,	BB9
rq-D4.1-T.06	UC1.1	T4.4,	BB7
rq-D4.1-T.07	P1, UC1.1, UC1.2,	T4.4,	BB7
rq-D4.1-T.08	P1, P2	T4.5,	BB6, BB8
rq-D4.1-T.09	P1, P2, UC1.1, UC1.2, UC1.3, UC2.1	T4.4,	BB7, BB8, BB9
rq-D4.1-T.10	P1, P2, P5 UC1.2, UC1.3, UC2.1, UC2.2	T4.3, T4.4, T4.5, T4.6	BB8
rq-D4.1-T.11	P1, P2 D1 UC2.2		BB3
rq-D4.1-T.12	P1, P2 UC2.2		
rq-D4.1-R.01	P1, P2, P5, D1 UC1.1, UC2.2	T4.2	BB6, BB8
rq-D4.1-R.02	P1, P2, UC1.1, UC1.2		BB6, BB7, BB8, BB9
rq-D4.1-R.03	P1, P2, P5, UC1.1, UC1.2, UC1.3, UC2.2	T4.3, T4.4, T4.5, T4.6	BB6
rq-D4.1-R.04	P1, P5 UC2.1		BB6
rq-D4.1-R.05	P1, P2, P3, P5 D1, D2, UC1.3 UC1.1, UC1.2, UC1.3, UC2.1, UC2.2	T4.7	BB4, BB10, BB11

Task Leaders:
T4.2: SIEMENS PLM. T4.3: REDEN. T4.4: TEK. T4.5: TU/e. T4.6 TEK. T4.7: TU/e.

Pilots, Use cases, Demonstrators	Code	Description
Pilot 1 (Sioux CCM)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	- by means of RC and ILC strategies. - lateral tracking precision < 10 um (3σ) - belt velocity stability in transport direction < 8 mm/s (3σ)
	rq-D4.1-P.04	
	rq-D4.1-P.05	compensation of disturbances above the bandwidth of the feedback controller
	rq-D4.1-P.06	- current loop control bandwidth > 500 Hz - current loop sample rate > 32 kHz - position loop sample rate > 10 kHz - jitter < 5 µs - communication with layer 1 sample rate> 10 kHz
	rq-D4.1-T.01	
	rq-D4.1-T.02	friction and inertia compensation
	rq-D4.1-T.03	- static/position/velocity/ acceleration/jerk/friction feedforward control - model-based feedforward
	rq-D4.1-T.04	capability of dealing with variations in belt speed
	rq-D4.1-T.05	dealing with variations in belt speed without requiring to re-learn all periodic disturbances.
	rq-D4.1-T.06	
	rq-D4.1-T.07	notch/biquadratic/lowpass/highpass filters
	rq-D4.1-T.08	dealing with time/position varying behaviour and load variation
	rq-D4.1-T.09	trajectory generator algorithm capable of: - electronic gearing - inverse kinematics - 3rd order, polynomial & sinusoidal setpoint generators - on the fly (smooth) trajectory updates - position/velocity/acceleration/force setpoint modes - jogging mode - multi-axis/multidimensional setpoints (with synchronization) input shaping
	rq-D4.1-T.10	decoupler in order to implement decentralized PID controllers
	rq-D4.1-T.11	- measurement of frequency responses (including cross-couplings for MIMO systems): - identification of unexpected deviations (feedforward terms, friction) - assessment of bandwidth, sensitivity, stability margins, settling time, rise time, overshoot, noise/disturbance levels - functions to easily design an IO-test/ self-test - functions shall be accessible via a scripting interface. - assessment of performance obtained by RC and ILC algorithms - user-friendly interface
	rq-D4.1-T.12	custom homing/commutation strategies
	rq-D4.1-R.01	
	rq-D4.1-R.02	identification procedure for learning control
	rq-D4.1-R.03	- gain scheduling. - on line adjusting procedure for selected parameters

		- auto tuning for feedback and feedforward terms and for RC and ILC algorithms - configurable performance requirements in auto tuning - acceleration/force/ torque/current limits should be considered - user-friendly interface
	rq-D4.1-R.04	- MIL/SIL/ PIL/HIL testing with different modelling software tools
	rq-D4.1-R.05	- multi-client interface with configurable user access levels -configurable min/max values of parameters - tracing of all signals and parameters (at least 20 signals @ 10 kHz continuously) - configurable decimation of tracing - triggering of traces - possible use of look up tables - possible use of state machines - supporting multi-core computing (for scalability). - control algorithms designed and customizable in Matlab Simulink. - centralized control loops executed on an X86 platform or SoC platform with FPGA (e.g. Xilinx Ultrascale). - supporting EtherCAT fieldbus (for communication with layer 1 devices) - Master clock for timing configurable - supporting Ethernet communication - implementation of a hypervisor-like solution
Pilot 2 (Nexperia)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	Setpoint distance: 1mm Setpoint duration: 17ms including settling time Setpoint settling: +/-2um
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	20 kHz control sample rate
	rq-D4.1-T.01	
	rq-D4.1-T.02	models for unpredictable wafer friction and relaxation
	rq-D4.1-T.03	- ILC to achieve perfect setpoint tracking. - jerk/voltage feedforward in the amplifier
	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	
	rq-D4.1-T.08	
	rq-D4.1-T.09	
	rq-D4.1-T.10	decoupling considering position-dependent dynamics and thermal effects
	rq-D4.1-T.11	- system condition monitoring - drive condition monitoring
	rq-D4.1-T.12	
	rq-D4.1-R.01	model-based methods for design, diagnosis and system identification
	rq-D4.1-R.02	
	rq-D4.1-R.03	- adjustable control algorithm - automatic tuning of controllers and feedforward terms

		- gain scheduling for position-dependent dynamics.
	rq-D4.1-R.04	
	rq-D4.1-R.05	- feedback from 8 encoders and 6 actuators. - support for tracing all signals, including current-loop, in real-time and for on-line changing of parameters
Pilot 3 (IMA)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	- cycle time: 500µs - jitter: 1% of cycle time
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	
	rq-D4.1-T.08	
	rq-D4.1-T.09	
	rq-D4.1-T.10	
	rq-D4.1-T.11	
	rq-D4.1-T.12	
	rq-D4.1-R.01	
	rq-D4.1-R.02	
	rq-D4.1-R.03	
	rq-D4.1-R.04	
	rq-D4.1-R.05	- at least 100 axes - Intel x86 platform - VxWorks 6.9.x support - minimize ad-hoc source code creation in multicore platforms - efficient inter-core communication mechanisms in multicore platforms
Pilot 4 (CORREA)		Pilot not involved in WP4.
Pilot 5 (PHI)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	- model calculation time: to be determined - fieldbus speed : < 1 [kHz] - control loop bandwidth: < 100 [Hz]
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	model-based feedforward control

	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	
	rq-D4.1-T.08	
	rq-D4.1-T.09	
	rq-D4.1-T.10	achieved by means of model-based feedforward control
	rq-D4.1-T.11	safety always guaranteed at all levels
	rq-D4.1-T.12	
	rq-D4.1-R.01	model of a subsystem of the medical manipulator
	rq-D4.1-R.02	
	rq-D4.1-R.03	- parameter tuning or calibration - assessment of adaptive control needs
	rq-D4.1-R.04	use of automated code generation taking into account constraints set by the environment and the application.
	rq-D4.1-R.05	
Demonstrator 1 (J&J Vistakon)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	
	rq-D4.1-T.08	
	rq-D4.1-T.09	
	rq-D4.1-T.10	Centralised control.
	rq-D4.1-T.11	Smart sensing and condition monitoring developed in I-MECH to improve product transfer system performance and implement a predictive maintenance platform to monitor vehicle condition and to address deterioration before failure. Wireless sensors to monitor electro-mechanical parameters and the use of a data analysis platform. Monitoring of MagneMotion carrier parameters Monitoring of carrier degradation Monitoring of energy consumption Data for predictive maintenance
	rq-D4.1-T.12	
	rq-D4.1-R.01	Create a digital twin of the system to model and analyze the existing process flow and system design in order to optimize output and provide a digital reference for the real time performance.
	rq-D4.1-R.02	
	rq-D4.1-R.03	
	rq-D4.1-R.04	

	rq-D4.1-R.05	Energy recovery solution will be developed to harvest electro-magnetic energy produced in the repeated acceleration and deceleration of the LSM vehicles to decrease the energy footprint of the manufacturing system. Data for overall system efficiency optimization. CPS for remote optimization of the performance of the line. Compatible with MagneMotion /AB PLC Control. Ethernet IP communication protocols. Dual loop feedback.
Demonstrator 2 (EDILASIO)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	vibration data acquisition and feedback
	rq-D4.1-T.08	
	rq-D4.1-T.09	
	rq-D4.1-T.10	
	rq-D4.1-T.11	corrosion data acquisition from mould tool and predictive maintenance
	rq-D4.1-T.12	
	rq-D4.1-R.01	
	rq-D4.1-R.02	
	rq-D4.1-R.03	
	rq-D4.1-R.04	
	rq-D4.1-R.05	
Use case 1.1 (GEFRAN)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	implementation of repetitive control
	rq-D4.1-T.06	passive and active anti-sway strategies with or without additional sensors
	rq-D4.1-T.07	iterative learning control block for vibration compensation
	rq-D4.1-T.08	

	rq-D4.1-T.09	input shaping and/or model inversion block
	rq-D4.1-T.10	
	rq-D4.1-T.11	
	rq-D4.1-T.12	
	rq-D4.1-R.01	
	rq-D4.1-R.02	
	rq-D4.1-R.03	- automatic tuning for all the control parameters - online controller parameters change block - predictive learning block
	rq-D4.1-R.04	
	rq-D4.1-R.05	availability of all signals
Use case 1.2 (FAGOR)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	current loop bandwidth > 2kHz. current loop sample rate >= 16khz. sample rate: 8khz (16kHz desirable) in at least one core sample rate: 1kHz (4kHz desirable) in another core
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	repetitive control for cogging of motors
	rq-D4.1-T.06	
	rq-D4.1-T.07	active vibration compensation with observer and load acceleration feedback
	rq-D4.1-T.08	model-based robust control
	rq-D4.1-T.09	learning control for pick and place or similar application
	rq-D4.1-T.10	multivariable control at least for axis q1 and q2.
	rq-D4.1-T.11	
	rq-D4.1-T.12	
	rq-D4.1-R.01	Use of Simulink
	rq-D4.1-R.02	
	rq-D4.1-R.03	self-commissioning velocity and position control loops
	rq-D4.1-R.04	
	rq-D4.1-R.05	- commutation method: FOC with encoder, motor position sent to CNC in ethercat. - full synchronization of drive and CNC loops through ethercat. - control specific multi/many core platform (2 cores minimum, 4 cores highly desirable, x86 platform with hypervisor support) - RTOS for multi/many core platform - program language: C (with intrinsic functions) - general Purpose OS: Linux and Windows 7 embedded (not necessary at the same time) - hypervisor as required for “near 0” latency and the sample rate for BB10.

		- full synchronization of the system loops - all the parts of control loops should be callable independently.
Use case 1.3 (TECO)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	EtherCAT communication period 8 kHz, configurable input shift time, in/out processing time <20 us in total at 64B per axis
	rq-D4.1-T.01	custom specified filters
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	
	rq-D4.1-T.08	
	rq-D4.1-T.09	Trajectory generators: Single axis, Multi axis, Coordinated motion (according to PLCOpen standard), G code
	rq-D4.1-T.10	MIMO Centralized controller (robot dynamic compensation)
	rq-D4.1-T.11	
	rq-D4.1-T.12	
	rq-D4.1-R.01	
	rq-D4.1-R.02	
	rq-D4.1-R.03	Configurable: - default: Cascade (position, velocity, current) - advanced: velocity, current feedforward, state controller, custom specified filters, centralized controller
	rq-D4.1-R.04	
	rq-D4.1-R.05	- USB communication for displays, disks or other peripherals - Wireless communication for wireless sensors - Development and possible integration of MQTT protocol - Processor board Qualcomm Snapdragon, 4x64 bit core, 1.2 GHz, 1 GB DDR3 RAM, 8GB eMMC - Integration of High speed cameras for Measurement of object size and orientation, camera as a motion sensor
Use case 2.1 (GMV A&D)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	- force range < 10N - force control tolerance < 5%
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	current loop bandwidth > 2kHz. current loop sample rate >= 16khz. Bandwidth per camera: - actual: nominal 150-200 Mbps - desirable up to 400 Mbps

		- desired: 800 Mbps Max update rate: - actual: 5-10 Hz - desired: 10-50 Hz Vision control loop < 20ms (50Hz) Sampling frequency >200Hz for the 6 axes force control loop
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	
	rq-D4.1-T.08	
	rq-D4.1-T.09	- trajectory generation, inverse kinematics - learning control.
	rq-D4.1-T.10	6 MIMO axes
	rq-D4.1-T.11	
	rq-D4.1-T.12	
	rq-D4.1-R.01	
	rq-D4.1-R.02	
	rq-D4.1-R.03	
	rq-D4.1-R.04	Implementation of the motion control for the force/vision-controlled device in a separate hardware allowing an easy and fast deployment and testing on the platform-art©
	rq-D4.1-R.05	Interface (e.g., Cameralikn, USB3, GiGE, Firewire 800..): - Actual: SpaceWire, GiGE, USB3.0, Camera-Link - Desired: + SpaceFibre and TSN/TTEthernet Number of realtime streaming cameras: - Actual: 1 - Desired: 2 Resolution: - Actual: 2048x2048 10-12bits and 1024x1024 10-12bits - Desired: 2048x2048 10-12bits and 1024x1024 10-12bits High speed vision for visual servoing. Distance range < 2m Speed range < 1cm/s Vision control tolerance: - approx 1cm (at 0.5m)
Use case 2.2 (ZAPUNI)	rq-D4.1-P.01	
	rq-D4.1-P.02	
	rq-D4.1-P.03	Force control: 6 DoF (end effector F/T sensor), sensitivity < 1 N Impedance control: End-effector dynamics can be prescribed (compliance regarding collaborative tasks)
	rq-D4.1-P.04	
	rq-D4.1-P.05	
	rq-D4.1-P.06	Sampling period of current controller: 32 - 64 kHz Sampling period of velocity/position controller: 8 - 16 kHz

		EtherCAT communication period 8 kHz, configurable input shift time, in/out processing time <20 us in total at 64B per axis
	rq-D4.1-T.01	
	rq-D4.1-T.02	
	rq-D4.1-T.03	
	rq-D4.1-T.04	
	rq-D4.1-T.05	
	rq-D4.1-T.06	
	rq-D4.1-T.07	
	rq-D4.1-T.08	
	rq-D4.1-T.09	
	rq-D4.1-T.10	MIMO 7 axes Inverse kinematics algorithms: - Default: direct control of redundant axis: - advanced: joint limits/velocities limitation, obstacle overcoming, singularity overcoming
	rq-D4.1-T.11	Ability to trace all signals in control loops over EtherCAT at full sampling rate (oversampling)
	rq-D4.1-T.12	aRUco markers for robot homing in a technology process, accuracy: < 3 mm
	rq-D4.1-R.01	Joint F/T sensors (own design), resolution < 2 N F/T sensor (commercial) in the robot base, resolution < 5 N
	rq-D4.1-R.02	
	rq-D4.1-R.03	Configurable: - default: Cascade (position, velocity, current) - advanced: velocity, current feedforward, state controller, custom specified filters, centralized controller (robot dynamic compensation)
	rq-D4.1-R.04	
	rq-D4.1-R.05	Payload 1 kg Reach < 1 m Weight < 20kg Applications: Pick & Place, Gluing, Small part assembly, screwing, etc. Human Guidance learning: YES Force control: YES Compliance behavior (impedance): YES SpaceMouse end-effector guidance: Recorded data postprocessing (interpolation/approximation, velocity profile definition, etc.) F/T sensor end-effector guidance: See Impedance control

Building block	Pilots, Use cases, Demonstrators	Description
BB1 - Platform for Smart Sensors with Advanced Data Processing (ZAPUNI)	Pilot 1	
	Pilot 2	
	Pilot 3	
	Pilot 4	
	Pilot 5	
	Demonstrator 1	
	Demonstrator 2	
	Use case 1.1	
	Use case 1.2	
	Use case 1.3	
	Use case 2.1	
	Use case 2.2	
BB2 - Real-time wireless sensors (EDI)	Pilot 1	
	Pilot 2	
	Pilot 3	
	Pilot 4	
	Pilot 5	
	Demonstrator 1	
	Demonstrator 2	Wireless sensors to monitor electro-mechanical parameters and use of a data analysis platform to contextualise the data collected.
	Use case 1.1	
	Use case 1.2	
	Use case 1.3	Wireless communication for wireless sensors
	Use case 2.1	
	Use case 2.2	
BB3 - Robust condition monitoring and predictive diagnostics (BUT)	Pilot 1	- configurable tracing of all signals and parameters (including signals from the current loop) and semi-real-time communication of the traces/signal/parameter values to upper layers of the control system. - configurable decimation of tracing
	Pilot 2	- monitoring of changes in feedforward parameters - drive condition monitoring - configurable tracing of all signals and parameters
	Pilot 3	
	Pilot 4	
	Pilot 5	
	Demonstrator 1	Smart sensing and condition monitoring developed in I-MECH to improve product transfer system performance and implement a predictive maintenance platform to monitor vehicle condition and to address deterioration before failure. Monitoring of MagneMotion carrier parameters Monitoring of carrier degradation Monitoring of energy consumption Data for predictive maintenance
	Demonstrator 2	- vibration data acquisition and feedback. - corrosion data acquisition from mould tool predictive maintenance.

	Use case 1.1	general system monitoring
	Use case 1.2	
	Use case 1.3	
	Use case 2.1	
	Use case 2.2	Ability to trace all signals in control loops over EtherCAT at full sampling rate (oversampling)
BB4 - High speed vision (TNO)	Pilot 1	
	Pilot 2	
	Pilot 3	
	Pilot 4	
	Pilot 5	
	Demonstrator 1	
	Demonstrator 2	
	Use case 1.1	
	Use case 1.2	
	Use case 1.3	Integration of High speed cameras for the measurement of object size and orientation, camera as a motion sensor
	Use case 2.1	Interface (e.g., Cameralikn, USB3, GiGE, Firewire 800..) - Actual: SpaceWire, GiGE, USB3.0, Camera-Link - Desired: + SpaceFibre and TSN/TTEthernet Bandwidth per camera - Actual: Nominal 150-200 Mbps - Desirable up to 400 Mbps - Desired: 800 Mbps Number of realtime streaming cameras - Actual: 1 - Desired: 2 Resolution - Actual: 2048x2048 10-12bits and 1024x1024 10-12bits - Desired: 2048x2048 10-12bits and 1024x1024 10-12bits Max update rate - Actual: 5-10 Hz - Desired: 10-50 Hz High speed vision for visual servoing. Distance range < 2m Speed range < 1cm/s Vision control tolerance approximatively 1cm (at 0.5m) Vision control loop < 20ms (50Hz)
	Use case 2.2	aRUco markers for robot homing in a technology process, accuracy: < 3 mm
BB5 - High performance servo amplifier (INGENIA)	Pilot 1	current loop control bandwidth > 500 Hz
	Pilot 2	jerk/voltage feedforward in the amplifier
	Pilot 3	
	Pilot 4	
	Pilot 5	
	Demonstrator 1	
	Demonstrator 2	
	Use case 1.1	
	Use case 1.2	Current loop bandwidth > 2kHz Current loop sample rate >= 16khz
	Use case 1.3	

	Use case 2.1	Current loop bandwidth > 2kHz Current loop sample rate >= 16khz
	Use case 2.2	Sampling period of current controller: 32 - 64 kHz
BB6 - Self-commissioning velocity and position control loops (GEFRAN)	Pilot 1	- gain scheduling. - on line adjusting procedure for selected parameters - autotuning for feedback and feedforward terms and for RC and ILC algorithms - configurable performance requirements in autotuning - acceleration/force/ torque/current limits should be considered - user-friendly interface
	Pilot 2	- automatic tuning of controllers and feedforwards - gain scheduling for position-dependent dynamics
	Pilot 3	
	Pilot 4	
	Pilot 5	- subsystem parameter tuning or calibration - assessment of adaptive control needs
	Demonstrator 1	dual loop feedback
	Demonstrator 2	
	Use case 1.1	- automatic tuning for all the control parameters - online controller parameters change block - predictive learning block
	Use case 1.2	self-commissioning velocity and position control loops, at least for standard PI controllers.
	Use case 1.3	Configurable control structure: - default: Cascade (position, velocity, current) - advanced: velocity, current feedforward, state controller, custom specified filters, centralized controller (robot dynamic compensation)
	Use case 2.1	
	Use case 2.2	Configurable control structure: - default: Cascade (position, velocity, current) - advanced: velocity, current feedforward, state controller, custom specified filters, centralized controller (robot dynamic compensation)
BB7 - Vibration control module (TEK)	Pilot 1	Vibration compensation block
	Pilot 2	
	Pilot 3	
	Pilot 4	
	Pilot 5	
	Demonstrator 1	
	Demonstrator 2	
	Use case 1.1	passive and active anti-sway strategies with or without additional sensors
	Use case 1.2	active vibration compensation with observer and load acceleration feedback
	Use case 1.3	
	Use case 2.1	
	Use case 2.2	
	Pilot 1	decoupling to implement decentralized PID controllers

BB8 - Robust model-based multivariable control (FAGOR)	Pilot 2	decoupling considering position-dependent dynamics and thermal effects
	Pilot 3	at least 100 axes
	Pilot 4	
	Pilot 5	model-based feedforward control (5-16 axes)
	Demonstrator 1	centralised control
	Demonstrator 2	
	Use case 1.1	
	Use case 1.2	multivariable control at least for axis q1 and q2
	Use case 1.3	MIMO control and trajectory generator Robot dynamic compensation
	Use case 2.1	force and impedance control in a 6 DOF manipulator (sampling frequency >200Hz, trajectory generation, inverse kinematics) sensor data: 6 axis load cell, motor current sensing force range < 10N force control tolerance < 5%
	Use case 2.2	Inverse kinematics algorithms: - Default: direct control of redundant axis - advanced: joint limits/velocities limitation, obstacle overcoming, singularity overcoming Collisions detection: - Joint F/T sensors (own design), resolution < 2 N - F/T sensor (commercial solution) in the robot base, resolution < 5 N
BB9 - Iterative and repetitive control module (TEK)	Pilot 1	- compensation of disturbances with frequencies above the bandwidth of the feedback controller - automatic tuning of the algorithm - condition monitoring functionality - decoupling functionality
	Pilot 2	Implementation for perfect setpoint tracking
	Pilot 3	
	Pilot 4	
	Pilot 5	implementation of ILC and RC algorithms
	Demonstrator 1	
	Demonstrator 2	
	Use case 1.1	implementation of ILC and RC algorithms
	Use case 1.2	Repetitive control, at least for cogging of motors. Learning control, for pick and place or similar application.
	Use case 1.3	
	Use case 2.1	implementation of ILC and RC algorithms
	Use case 2.2	
BB10 - Control Specific Multi-many core Platform (TU/e)	Pilot 1	Control loops framework shall be executed on an X86 platform.
	Pilot 2	
	Pilot 3	Intel x86 platform, in order to be able to re-use the existing software layers
	Pilot 4	
	Pilot 5	
	Demonstrator 1	
	Demonstrator 2	
	Use case 1.1	
	Use case 1.2	- 2 cores minimum, 4 cores highly desirable.

		- sample rate: 8Khz (16KHz desirable) in at least one core. - sample rate: 1KHz (4KHz desirable) in another core. - x86 platform with hypervisor support
	Use case 1.3	Qualcomm Snapdragon, 4x64 bit core, 1.2 GHz, 1 GB DDR3 RAM, 8GB eMMC
	Use case 2.1	
	Use case 2.2	
BB11 - RTOS for multi-many core platform (UNIMORE)	Pilot 1	The X86 control platform shall implement a hypervisor-like solution to allow execution of a non-real-time operating system besides the real-time operating system on the same hardware platform
	Pilot 2	
	Pilot 3	- VxWorks 6.9.x support - minimize ad-hoc source code creation in multicore platforms - efficient inter-core communication mechanisms in multicore platforms
	Pilot 4	
	Pilot 5	
	Demonstrator 1	
	Demonstrator 2	
	Use case 1.1	
	Use case 1.2	Program language: C (with intrinsic functions) General Purpose OS: Linux and Windows 7 embedded (not necessary at the same time) Hypervisor as required for “near 0” latency and the sample rate for BB10
	Use case 1.3	
	Use case 2.1	
	Use case 2.2	

9 Conclusions

In this deliverable, initial requirements and specifications for I-MECH motion control applications utilizing centralized and decentralized control solutions have been analyzed.

From the considered requirements, it can be highlighted that

- BB1 is not directly connected to WP4;
- the wireless sensors provided by BB2 can be used directly from Demonstrator 2 to monitor electro-mechanical parameters and the use of a data analysis platform.
- several Pilots, Use Cases and Demonstrators will use the outputs of BB3. In particular, predictive maintenance as well as signal tracing, is a key point for the involved systems.
- BB4 is relevant for Use Case 2.1, as Use Case 2.1 and 2.2 needs to implement a control strategy which uses cameras.
- Regarding the connection between BB5 - High performance servo amplifier (ING) and WP4, in this deliverable it is shown that only Pilot 2 and Use Case 1.2, 2.1 and 2.2 need the development of High performance servo amplifier from BB5 for the implementation of jerk/voltage feedforward techniques and for increased current control bandwidth.
- Regarding the connection between BB6 - Self-commissioning velocity and position control loops (GEF) and WP4, it is possible to say that most of the Pilots, Use Cases and Demonstrators involved in the project need the direct work from BB6. In particular, most of them are interested in the self-commissioning of velocity and position control loops.
- Regarding the connection between BB7 - Vibration control module (TEK) and WP4, only Use case 1.1 and 1.2 need a direct work from BB7. In the specific case, they need passive and active anti-sway and vibration compensation blocks.
- Regarding the connection between BB8 - Robust model-based multivariable control (FAG) and WP4, it is possible to notice that most of the Pilots, Use Cases and Demonstrators involved in the project need the direct work from BB8. In particular, the need of a decoupling strategy is expressed.
- Regarding the connection between BB9 - Iterative and repetitive control module (TEK) and WP4, this deliverable shows that several Pilots and Use Cases need the work from BB9. In this context, RC and ILC modules are needed to compensate for periodic disturbances and to increment the set point tracking precision.
- Regarding the connection between BB10 - Control Specific Multi-many core Platform (TUE) and WP4, in this deliverable only Use Case 1.2 and 1.3 express the needing of the work from BB10.
- Regarding the connection between BB11 - RTOS for multi-many core platform (UMO) and WP4, in the deliverable only Pilot 3 Use Case 1.2 express the direct needing of the work from BB11.

It is worth stressing that for a more complete overview of the link between WP4 and BBs it is necessary to take a look to the summary tables in Chapter 8.

Bibliography

[1] B. C. Kuo, Automatic control systems, Prentice Hall PTR, 1987.

[2] G. C. Goodwin, S. F. Graebe and M. E. Salgado, Control system design, Upper Saddle River, 2001.

[3] G. F. Franklin, J. D. Powell and M. L. Workman, Digital control of dynamic systems, Menlo Park, CA: Addison-wesley, 1998.

[4] R. C. Dorf and R. H. Bishop, Modern control systems, Pearson, 2011.

[5] R. Isermann, Digital control systems, Springer Science & Business Media, 2013.

[6] K. J. Åström and T. Hägglund, PID controllers: theory, design, and tuning, Isa Research Triangle Park, NC, 1995.

[7] A. Visioli, Practical PID control, Springer Science & Business Media, 2006.

[8] K. J. Åström and T. Hägglund, Advanced PID control., ISA-The Instrumentation, Systems and Automation Society, 2006.

[9] M. A. Johnson and M. H. Moradi, PID control, Springer-Verlag London Limited, 2005.

[10] K. K. Tan, Q.-G. Wang and C. C. Hang, Advances in PID control, Springer Science & Business Media, 2012.

[11] A. Visioli, "Modified anti-windup scheme for PID controllers," *IEE Proceedings-Control Theory and Applications,* vol. 150, no. 1, pp. 49-54, 2003.

[12] B. Käpernick and K. Graichen, "Model predictive control of an overhead crane using constraint substitution.," in *American Control Conference (ACC)*, Washington, DC, USA , 2013.

[13] E. F. Camacho and C. B. Alba, Model predictive control, Springer Science & Business Media, 2013.

[14] L. Wang, Model Predictive Control System Design and Implementation Using MATLAB, Melbourne: Springer-Verlag London Limited, 2009.

[15] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control?? a computer control algorithm," *Joint automatic control conference,* vol. 17, p. 72, 1980.

[16] C. R. Cutler, "Dynamic matrix control: an optimal multivariable control algorithm with constraints," 1983.

[17] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized predictive control-Part I. The basic algorithm," *Automatica,* vol. 23, no. 2, pp. 137-148, 1987.

[18] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized predictive control-part II. Extension and Interpretations," *Automatica,* vol. 23, no. 2, pp. 149-160, 1987.

[19] E. F. Camacho, "Constrained generalized predictive control," *IEEE transactions on automatic control,* vol. 38, no. 2, pp. 327-332, 1993.

[20] R. M. C. De Keyser and A. R. Van Cauwenberghe, "Extended prediction self-adaptive control," *IFAC Proceedings Volumes,* vol. 18, no. 5, pp. 1255-1260, 1985.

[21] J. APA and D. O'Donovan, Predictive functional control: principles and industrial applications, Springer Science & Business Media, 2009.

[22] R. Soeterboek, Predictive control: a unified approach, Prentice-Hall, Inc., 1992.

[23] A. Bemporad, F. Borrelli and M. Morari, "Model predictive control based on linear programming~ the explicit solution," *IEEE Transactions on Automatic Control,* vol. 47, no. 12, pp. 1974-1985, 2002.

[24] F. Allgöwer and A. Zheng, Nonlinear model predictive control. Birkhäuser, 2012., Birkhäuser, 2012.

[25] D. E. Kirk, Optimal control theory: an introduction, Courier Corporation, 2012.

[26] K. Zhou and J. Doyle, Essentials of Robust Control, Prentice Hall, 1998.

[27] P. Grabowski, “The LQ controller synthesis problem,” *IMA Journal of Mathematical Control and Information,* vol. 10, no. 2, pp. 131-148, 1993.

[28] M. A. Rotea, “The generalized H2 control problem,” *Automatica,* vol. 29, no. 2, pp. 373-385, 1993.

[29] A. A. Stoorvogel, “The singular H2 control problem,” *Automatica,* vol. 28, no. 3, pp. 627-631, 1992.

[30] J. B. Burl, Linear optimal control: H (2) and H (Infinity) methods, Addison-Wesley Longman Publishing Co., Inc., 1998.

[31] T. Başar and P. Bernhard, H-infinity optimal control and related minimax design problems: a dynamic game approach, Springer Science & Business Media, 2008.

[32] K. Zhou and J. C. Doyle, Essentials of robust control, Upper Saddle River,, NJ: Prentice hall, 1998.

[33] K. J. Åström and B. Wittenmark, Adaptive control, Courier Corporation, 2013.

[34] S. Sastry and M. Bodson, Adaptive control: stability, convergence and robustness, Courier Corporation, 2011.

[35] H. Kaufman, I. Barkana and K. Sobel, Direct adaptive control algorithms: theory and applications, Springer Science & Business Media, 2012.

[36] I. D. Landau, R. Lozano, M. M’Saad and A. Karimi, “Indirect adaptive control,” in *Adaptive Control*, Springer London, 2011, pp. 409-456.

[37] D. D. Sworder, “Hybrid adaptive control,” *Applied mathematics and computation,* vol. 45, no. 2, pp. 173-192, 1991.

[38] T. Oomen, E. Grassens and F. Hendriks, “Inferential Motion Control: Identification and Robust Control Framework for Positioning an Unmeasurable Point of Interest,” *IEEE Transactions on Control Systems Technologies,* vol. 23, no. 4, pp. 1602-1610, 2015.

[39] A. Levant, “Sliding order and sliding accuracy in sliding mode control,” *International journal of control,* vol. 58, no. 6, pp. 1247-1263, 1993.

[40] V. Utkin, “Sliding mode control,” in *Control systems, robotics and automation-Volume XIII: Nonlinear, Distributed, and Time Delay Systems-II*, Oxford, Heinz Unbehauen, 2009, p. 130.

[41] A. Bartoszewicz and R. J. Patton, “Sliding mode control., 2007, 21.8-9: .,” *International Journal of Adaptive Control and Signal Processing,* vol. 21, no. 8-9, pp. 635-637, 2007.

[42] H.-S. Ahn, Y. Chen and K. L. Moore, “Iterative learning control: Brief survey and categorization,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 37, no. 6, pp. 1099-1121, 2007.

[43] K. L. Moore, Iterative learning control for deterministic systems, Springer Science & Business Media, 2012.

[44] C. Kempf, W. C. Messner, M. Tomizuka and R. Horowitz, “Comparison of four discrete-time repetitive control algorithms,” *IEEE Control Systems,* vol. 13, no. 6, pp. 48-54, 1993.

[45] K. Zhang, Y. Kang, J. Xiong and J. Chen, “Direct repetitive control of SPWM inverter for UPS purpose,” *IEEE Transactions on Power Electronics,* vol. 18, no. 3, pp. 784-792, 2003.

[46] R. Grino, R. Cardoner, R. Costa-Castelló and E. Fossas, "Digital repetitive control of a three-phase four-wire shunt active filter," *IEEE Transactions on Industrial Electronics,* vol. 54, no. 3, pp. 1495-1503, 2007.

[47] M. Steinbuch, S. Weiland and T. Singh, "Design of noise and period-time robust high-order repetitive control, with application to optical storage," *Automatica,* vol. 43, no. 12, pp. 2086-2095, 2007.

[48] Y. Li and Q. Xu, "Design and robust repetitive control of a new parallel-kinematic XY piezostage for micro/nanomanipulation," *IEEE/ASME Transactions on Mechatronics,* vol. 17, no. 6, pp. 1120-1132, 2012.

[49] Y. Chen, K. L. Moore and H.-S. Ahn, "Iterative learning control," *Encyclopedia of the Sciences of Learning. Springer US,* pp. 1648-1652, 2012.

[50] D. A. Bristow, M. Tharayil and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems,* vol. 26, no. 3, pp. 96-114, 2006.

[51] H.-S. Ahn, Y. Chen and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 37, no. 6, pp. 1099-1121, 2007.

[52] J. Madariaga, L. G. Uriarte, I. Ruíz de Argandoña, J. L. Azpeitia and J. C. Rodríguez de Yurre, "Machine-Tool Tracking Error Reduction In Complex Trajectories Through Anticipatory ILC," *Spring ASPE Topical Meeting: Control of Precision Systems,* 2010.

[53] A. Visioli, G. Ziliani and G. Legnani, "Iterative-Learning Hybrid Force/Velocity Control forContour Tracking," *IEEE Transaction on Robotics,* vol. 26, no. 2, pp. 388-393, 2010.

[54] W. J. Rugh, Linear system theory, Upper Saddle River, NJ: Prentice Hall, 1996.

[55] W. J. Rugh, Nonlinear system theory, Baltimore: Johns Hopkins University Press, 1981.

[56] M. G. Safonov, Stability and robustness of multivariable feedback systems, MIT press, 1980.

[57] V. Van Geffen, "A study of friction models and friction compensation," *DCT,* vol. 118, p. 24, 2009.

[58] H. Olsson, K. J. Åström, C. C. De Wit, M. Gäfvert and P. Lischinsky, "Friction models and friction compensation," *European journal of control,* vol. 4, no. 3, pp. 176-195, 1998.

[59] L. Simoni, M. Beschi, G. Legnani and A. Visioli, "Modelling the temperature in joint friction of industrial manipulators," *Robotica,* pp. 1-22, 2017.

[60] C. Canudas, K. Åström and K. Braun, "Adaptive friction compensation in DC-motor drives," *IEEE Journal on Robotics and Automation,* vol. 3, no. 6, pp. 681-685, 1987.

[61] R. J. E. Merry, M. J. G. Van de Molengraft and M. Steinbuch, "Velocity and acceleration estimation for optical incremental encoders," *Mechatronics,* vol. 20, no. 1, pp. 20-26, 2010.

[62] J. D. Han, Y. Q. He and W. L. Xu, "Angular acceleration estimation and feedback control: An experimental investigation," *Mechatronics,* vol. 17, no. 9, pp. 524-532, 2007.

[63] P. B. Schmidt and R. D. Lorenz, "Design principles and implementation of acceleration feedback to improve performance of dc drives," *IEEE Transactions on Industry Applications,* vol. 28, no. 3, pp. 594-599, 1992.

[64] S. G. H. Hepner, "Adaptive two-time-scale tracking filter for target acceleration estimation," *Journal of guidance, control, and dynamics,* vol. 14, no. 3, pp. 581-588, 1991.

[65] J.-J. E. Slotine and W. Li, Applied nonlinear control, Englewood Cliffs, NJ: Prentice hall, 1991.

[66] H.-S. Na and Y. Park, "An adaptive feedforward controller for rejection of periodic disturbances," *Journal of Sound and Vibration,* vol. 201, no. 4, pp. 427-435, 1997.

[67] R. T. Burt and J. Y. Zhang, “Notch filter for ripple reduction in chopper stabilized amplifiers”. U.S. Patent 7,292,095, 2007.

[68] R. Burt and J. Zhang, “A micropower chopper-stabilized operational amplifier using a SC notch filter with synchronous integration inside the continuous-time signal path,” *IEEE Journal of solid-state circuits,* vol. 41, no. 12, pp. 2729-2736, 2006.

[69] W. Singhose, D. Kim and M. Kenison, “Input shaping control of double-pendulum bridge crane oscillations,” *Journal of Dynamic Systems, Measurement, and Control,* vol. 130, no. 3, p. 034504, 2008.

[70] S. Garrido, M. Abderrahim, A. Giménez, R. Diez and C. Balaguer, “Anti-swinging input shaping control of an automatic construction crane,” *IEEE Transactions on Automation Science and Engineering,* vol. 5, no. 3, pp. 549-557, 2008.

[71] A. Piazzi and A. Visioli, “Optimal dynamic-inversion-based control of an overhead crane,” *IEE Proceedings-Control Theory and Applications,* vol. 149, no. 5, pp. 405-411, 2002.

[72] F. Padula, A. Visioli, D. Facchinetti and A. Saleri, “A dynamic inversion approach for oscillation-free control of overhead cranes,” *IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA),* pp. 1-6, 2015.

[73] M. Mahfouf, C. H. Kee, M. F. Abbod and D. A. Linkens, “Fuzzy logic-based anti-sway control design for overhead cranes,” *Neural Computing & Applications,* vol. 9, no. 1, pp. 38-43, 2000.

[74] Y.-S. Kim, K.-S. Hong and S.-K. Sul, “Anti-sway control of container cranes: inclinometer, observer, and state feedback,” *International Journal of Control, Automation, and Systems,* vol. 2, no. 4, pp. 435-449, 2004.

[75] Q. H. Ngo and K.-S. Hong, “Sliding-mode antisway control of an offshore container crane,” *IEEE/ASME transactions on mechatronics,* vol. 17, no. 2, pp. 201-209, 2012.

[76] J.-K. Ji and S.-K. Sul, “Kalman filter and LQ based speed controller for torsional vibration suppression in a 2-mass motor drive system,” *IEEE Transactions on industrial electronics,* vol. 42, no. 6, pp. 564-571, 1995.

[77] T. Su, S. Hattori, M. Ishida and T. Hori, “Suppression control method for torque vibration of AC motor utilizing repetitive controller with Fourier transform,” *IEEE Transactions on Industry Applications,* vol. 38, no. 5, pp. 1316-1325, 2002.

[78] S. M. N., “Input shaping for vibration-free positioning of flexible systems,” *Proceedings of the Institution of Mechanical Engineers Part I Journal of Systems and Control Engineering,* vol. 215, no. 5, pp. 467-481, 2001.

[79] I. Ruiz de Argandona, M. Zatarain, A. Illarramendi and J. L. Azpeitia, “Improvement of the performance in Machine Tools by Means of State Space Control Strategies,” *44th IEEE Conference on Decision and Control and European Control Conference,* 2005.

[80] D. Croft and S. Devasia, “Vibration compensation for high speed scanning tunneling microscopy,” *Review of Scientific Instruments,* vol. 70, no. 12, pp. 4600-4605, 1999.

[81] R. Horowitz, Y. Li, K. Oldham, S. Kon and X. Huang, “Dual-stage servo systems and vibration compensation in computer hard disk drives,” *Control Engineering Practice,* vol. 15, no. 3, pp. 291-305, 2007.

[82] I. D. Landau, M. Alma and T.-B. Airimitoaie, “Adaptive feedforward compensation algorithms for active vibration control with mechanical coupling,” *Automatica,* vol. 47, no. 10, pp. 2185-2196, 2011.

[83] O. Ozen, Sariyildiz, Y. H. E., K. Ogawa, K. Ohnishi and A. Sabanovic, "Practical PID controller tuning for motion control," *IEEE International Conference on Mechatronics (ICM),* pp. 240-245, 2015.

[84] "platform_art," [Online]. Available: https://www.gmv.com/export/sites/gmv/DocumentosPDF/platform_art/PLATFROM__ENG.pdf.

[85] H. Olsson, K. J. Astrom, C. C. De Wit, M. Gafvert and P. Lischinsky, "Friction models and friction compensation," *European Journal of Control,* vol. 4, no. 3, pp. 176-195, 1998.

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.