Chapter 9
Industry 4.0 in Robotics

Irene Fassi, Claudia Pagano and Serena Ruggeri

9.1 Introduction

The so-called 4th industrial revolution is going to radically change the common idea of the factory, and a great industrial research effort is dedicated to the definition of the "Factory of the Future" models.

The fourth industrial revolution is characterised by the word *automation*, which is achieved in industrial production by addressing four challenges:

1. Intelligent machines control the production in smart factories in which the presence of human is enormously reduced.
2. Intelligent machines optimize the capacity of the production facility, coordinating the movement of the material, analysing the status of the production chain and the stock, and zeroing the downtime.
3. Intelligent machines are essentially self-organized; material planning and the handling of orders are fully automated.
4. Intelligent machines can autonomously reconfigure the production line to respond in a very short time to the personalized customer request.

Intelligent machines are based on a complex cyber part and artificial intelligence algorithm. Indeed, Industry 4.0 mainly has two foundations: the hardware layer, which includes all the physical elements, and a software layer, the cyber part. Robot sensing and actuation, which belong to the hardware layer, were discussed in the previous chapter. In this chapter the cyber part is discussed, focusing on programming, control and artificial intelligence. Moreover, a brief overview of the

---

I. Fassi, C. Pagano(✉), S. Ruggeri
Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council, Milan, Italy
e-mail: claudia.pagano@stiima.cnr.it

J.C. Chaplin et al (eds), *Digital Manufacturing for SMEs*
DOI: 10.17639/dt0m-a649

ethical issues related with the intelligence implemented in the machines is provided at the end of the chapter.

9.2 Cyber Physical System

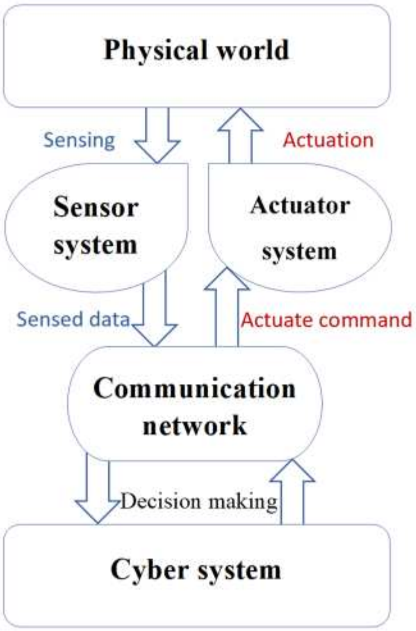


Figure 9.2-1 CPS holistic view.

Smart factories require the synergic deployment of several technologies, and among these the *Cyber Physical System* (CPS) is essential.

The key element for CPS is the *interaction* between physical (real world) and cyber (virtual world) elements, which are cooperating systems, having autonomous behaviours, context awareness through enhanced sensing capabilities, and storage and data processing capabilities from the sensors in the network (Figure 9.2-1).

Some of the practical examples that have already emerged include advanced robotics (e.g. autonomous cars, automatic pilot avionics, micro-robotics, robot-assisted surgery, implanted medical devices), intelligent buildings and the smart electric grid.

Cyber-Physical Systems (CPS) create a link between *physical and digital systems* in order to generate a common infrastructure with advanced capability. They allow integration of the dynamics of the physical processes with those of the software and networking, so that they can be handled as a single entity.

A CPS performs two main tasks:

- *Intense Connection:* In order to guarantee continuous data acquisition from the physical world and information feedback from the cyber space.
- Data Management, Analysis and Computation: In order to create the cyber space.

To fulfil these tasks a CPS is composed of "*collaborating computational entities that connect the cyber world with the surrounding physical environments or processes in an internet environment*" [1]. A CPS includes "*embedded systems (such as equipment, buildings, means of transportation, and medical devices), internet services, logistic, coordination, and management processes*" [1]. A broad range of sensors and actuators are used to connect the elements of the CPS and allow human-machine communication. The CPS stores and processes all the data received from the sensors and communication systems, and controls the physical systems using the actuators.

Research and development in this area is tackling a wide range of issues including inference from empirical data (i.e. elaborate empirical data in order to draw conclusions), sensing and perception, motor learning and control to adapt to different contexts, and the design, implementation, and verification of safe and well-functioning CPS.

A simplified scheme of a CPS and the interaction between humans and machines is shown in Figure 9.2-2.

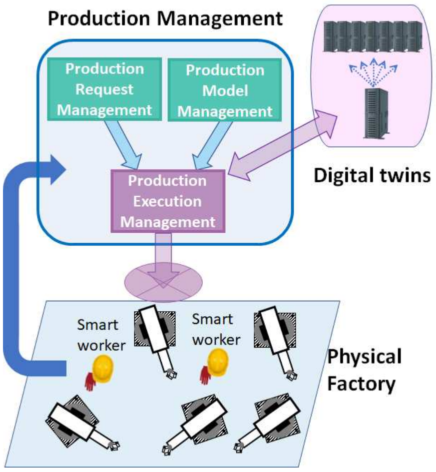


Figure 9.2-2 CPS scheme with interaction connections.

Cyber Physical Production Systems (CPPS) are Cyber Physical Systems for production. They enable and support communication between humans, machines and products. CPPS consist of autonomous and cooperative elements – related to processes, machines, production and logistics – connected across all levels of production, and an information system so that their operation and cooperation activities can be modelled. Thanks to the continuous exchange of data between and within the systems, stored and analysed in real time, the model is always synchronized with the status of the factory and the behaviour of the entire system can be forecast based on past and present situations. This can help in various decision-making processes so that the most appropriate actions can be quickly implemented, improving the productivity of the smart factory [2].

The main characteristics of CPPS are:

- *Intelligence:* The system components are able to acquire information from the environment and act autonomously.
- *Connectedness:* The system elements are able to cooperate and collaborate with each other and are connected to the knowledge and services on the Internet.
- *Responsiveness:* The system is able to react to internal and external changes.

The CPPS concept is strongly linked to other innovative concepts such as Internet of Things (IoT), big data and digital twins. The *IoT* refers to a wireless communication capability integrated with sensors and computing that allows the collection of data related to uniquely identifiable objects through the Internet. *Big data* refers to a new computing paradigm that allows the collection, processing and analysis of massive amounts of data [3].

The Digital Twin (DT) is another technology strictly related with Industry 4.0 and smart manufacturing. As with CPS, the digital twin is associated with the integration between the cyber and physical worlds. By definition a DT creates a virtual model of a physical system in order to predict real behaviour based on the simulation in real time of the system. Therefore, the two concepts have similar goals and approaches. However, while the CPS is implemented with sensors and actuators, which enable the interaction between physical and cyber world, DT is based on models and data. Digital twins are digital replicas of physical systems; they represent evolving digital profiles which are designed using the data collected from the past and present behaviour of a physical object or process. They are mainly applied in tasks such as monitoring, predictive maintenance and optimization of their physical counterparts.

9.2.1 Collaborative CPS

Collaborative CPS (CCPS) are even more complex production systems with respect to other CPS: the concept of “integrality” represents the level of integration of the different modules, aiming at self-learning and reconfiguration properties. In order to improve this integration, that concerns also the Human-Machine Interface (HMI), the CCPS must present a high level of “sociability”, meaning the communication capabilities with the environment, including other CPS [4].

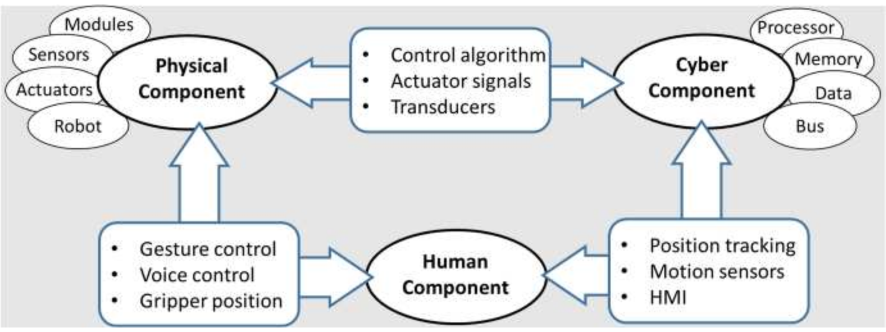


Figure 9.2-3 Structure of a Collaborative Robotics Cyber-Physical System.

CCPS include three main entities: the human component (HC), the physical component (PC) and the computational component (CC), which are interconnected through a variety of technologies (Figure 9.2-3). Being a part of a cyber-physical system, a human and a robot can interact and perform self-organizing tasks. In a future scenario, the integrated sensor network and communication technologies of CPS make the interaction reliable, safe and secure. Several sensors allow the robot to be aware of the human presence and react consequentially, based on the implemented human avoidance scheme. For the interaction to be safe a real-time and accurate human position tracking system is essential. A vision system connected with the robot and monitoring its environment can provide information on the worker location, used as input for speed reduction schemes for the robot. At the same time the monitoring system integrated with a gesture recognition system can be used by the worker to control the robot, together with a voice control. Moreover, force sensors can be integrated in the system to enhance the human robot interaction level. Combining the data from the force sensors and the vision system the robot speed and acceleration can be modulated according to the part of the worker’s body in proximity of the end effector. The force sensors can provide additional feature permitting a contact between the human and the robot in order for the worker to train the robot using the hand.

9.2.2 Swarm Robotics and CPS

An early application of the concept of CPS can be found in swarm robotics, a topic largely studied in the last 20 years. Swarm robotics is inspired by nature and particularly the behaviour of swarms able to perform tasks that would be beyond the capabilities of the individuals. It concerns the coordination of distributed robotic teams, interconnected, which can be, thus, considered as simplified CPS.

Several works have been carried out [5], studying and improving the communication among robots and the algorithm managing the group strategy. For practical reason (the available space), analogy with nature (swarm of insects), but also technical reasons, this approach is extensively studied on large numbers of mini and micro robots; indeed, due to space and energy consumption limitation, miniaturized robots have typically a reduced degree of complexity. Therefore, the group strategy is very suitable for such small and simple robots.

A very successful example of swarm robotics has been developed around the e-puck, a mobile minirobot developed mainly for educational purpose [6], and a variety of works to study the group coordination and synchronization [7][8][9] .

The e-puck [10] has a circular section of 75 mm of diameter and variable height, depending on the extensions provided (Figure 9.2-4).

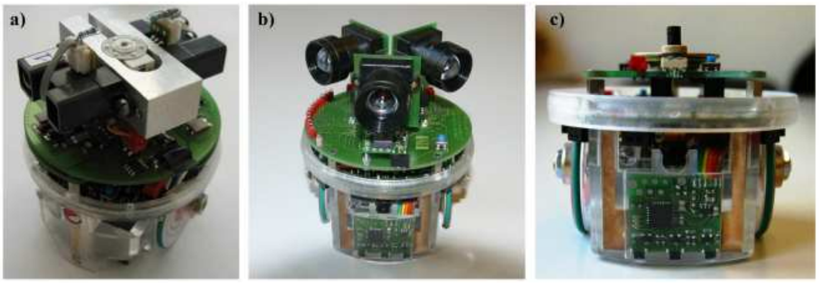


Figure 9.2-4 E-puck robot configuration provided with: a) a IR distance scanner, b) large field of view linear camera and c) ground colour measurement sensor [6].

Among these works some have been focused on the strategy to make a suitable number of e-puck able to move a box and place it in a chosen position. Although the design of the robot allows its customization providing it with the most suitable sensors and features required for the task (Figure 9.2-5), one of the aims of the works is to test several algorithms, not only to achieve the task, but also to do it with the easiest configuration of the robots as possible.

The objects to transport were chosen to be heavy enough to require the efforts of all the group, so that cooperation was essential. The robots are equipped with several IR sensors to detect the obstacles and find the target position, a camera to identify the obstacles (the objects and the other robots) and an optical camera with low image resolution underneath the robot. This camera gives the individual robots the perception of the direction of movement of the object, according to this the robots change the point of application of their pushing, avoiding to neutralize another robot’s force. Eventually they are all aligned to successfully transport the object [7].

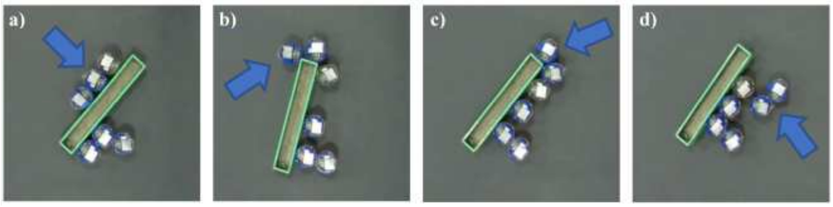


Figure 9.2-5 Alignment of the e-puck robots to successfully transport the object [11].

A more practical application of the typical algorithms of swarm robotics, and specifically path planning and coordination of distributed robotic teams in a CPS, is represented by automated warehouses. As an example, in the warehouses of Amazon and Ocado, the entire system, from the order to the delivery, is automated. A large number of autonomous mobile robots (Figure 9.2-6) are connected among each other and with the system where the orders are placed and an algorithm optimizes the path of the robots according to the positions of the ordered goods so that the displacement of the robots is scheduled to move the goods and pack the order. The connection and coordination among the robots is essential to avoid possible accidents.

In the Amazon warehouse [12] the robots move the shelves containing the required goods toward a worker, who picks the correct goods without the tiring walk amongst the shelves.

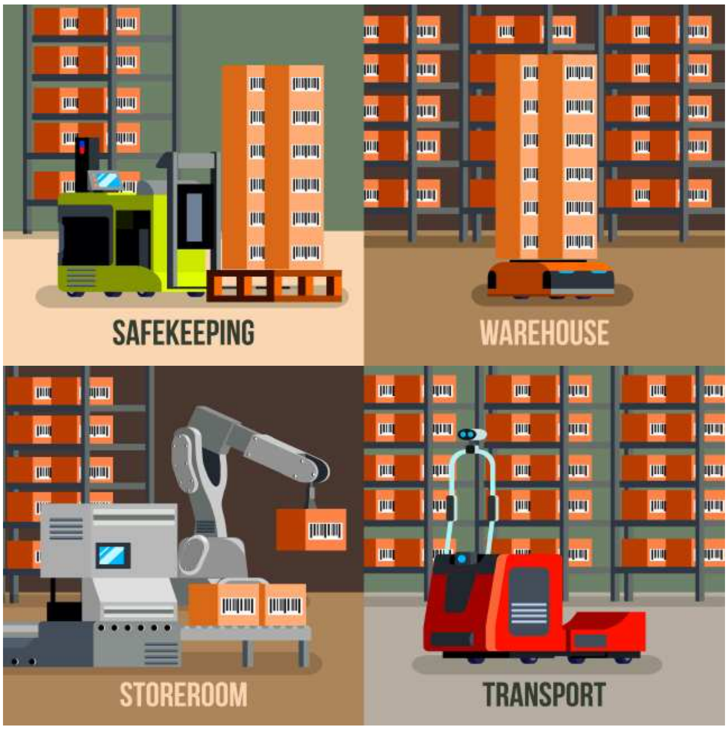


Figure 9.2-6 Automated warehouse*.

In Ocado's warehouse [13] the robots displace crates and pick goods from them moving along a grid system.

---

* Image designed by Freepik

9.3 Robot Programming

The execution of a complete task in a robotic application requires the execution of one or more operations by the robot. The robot has to move according to a specific motion that is completely or partially defined by the motion law imposed on its end-effector. The desired motion results from suitable *motion planning and programming of the robot task* and is achieved by a set of commands sent to the robot's actuators in accordance with a predefined control strategy. For the correct execution of the task, it is important to know the kinematic structure of the robot, including the type and the location of the actuators (the motors), the joints, the transmissions, and the dimensions of the mechanical structure. Such knowledge is completed by that of the electronics controlling the actuators and of the electrical and mechanical limits of the components.

9.3.1 Motion Planning

The position and movement of the end-effector or the force it exerts depends on the position, movement and actions that the different parts of the robot exert by means of its actuators. Therefore, the analysis of the robot and its modelling are preliminary steps in the development of a robotic application.

The problem with planning a trajectory for a robot can be broken down into finding a *path* and defining a *timing law* on the path. Indeed, the path identifies the locus of the points that the robot has to follow to perform the desired motion (Figure 9.3-1), whereas the "*trajectory is a path on which a timing law is specified*" [14], e.g. in terms of velocities at each point.

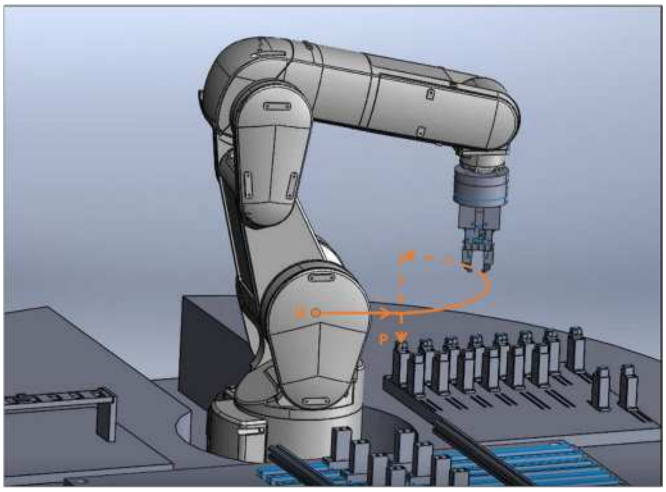


Figure 9.3-1 Example of a robot in simulated motion, moving from an initial point (H – home position) towards point P to pick a component on the table. Image rights: Authors.

Thus, *motion planning* concerns the generation of the timing laws for the coordinates (at the joints or at the end-effector), therefore the inputs to the motion control system [14].

Depending on the operations the robot is required to perform, the robot can adopt one of the following motion strategies [15]: from an initial point to a final point (*point-to-point motion*) with an arbitrary trajectory; through a finite sequence of points assigned along the path (*motion through intermediate points*); or according to a *specific trajectory*.

When defining the timing law, several aspects should be considered, including [14][15]:

- The *constraints* deriving from the type of application.
- The mechanical structure of the robot.
- The *joint forces and torques* should respect the limits of the actuators, the drives and the controller.
- The *trajectories* that interpolate the path points should be *smooth* to minimize undesired effects such as vibrations, the trajectories should not imply a very high *computational load.*
- The *positions and velocities at the joints* should be continuous functions of time (continuity of accelerations is not mandatory although generally welcome).

9.3.2 Programming

Programming a robot means instructing it on the task(s) it has to perform. A programming environment and suitable programming language(s) have to be defined.

Existing *programming methods* include [16][17][18] different approaches:

- *Guiding the robot to the positions of interest* or along the desired paths (teach programming) by using a teach pendant, manually holding the end-effector, or in tele-operation (using a master-slave system, so that the operator moves the master robot and the actual “slave” robot moves accordingly).
- *Writing a program* in a language that the robot can interpret and execute (proprietary language or robotic libraries supporting standard programming languages).
- *Using interactive graphical interfaces*, that is CAD (Computer Aided Design) modelling and system simulation tools to define the task and generate the program automatically.
- *Task-level programming* consists of the user telling the robot what should be done and the robot knows how to do it. This is achieved thorough modelling and sensing of the environment and machine intelligence. Therefore, task-level programming is difficult to achieve at the present time, but future innovative applications may possibly be developed.

Generating robot programs from CAD/CAM software packages is frequently used, since they are very common among manufacturing companies [16]. In the context of the digital factory, digital solutions for the 3D representation and simulation of an individual robot or robotic lines effectively support the design and validation of the manufacturing process. Other approaches to instruct a robot include, for example, using pointing devices or choreographing the task movements [16][18].

Robot programming methods can be separated into *online programming* and *offline programming.*

With *online programming*, the physical robot is used and the operator directly acts on the robot controller. Programming by teaching belongs to this category. The robot is taught or guided as needed and the actions are recorded in the robot's controller memory. The robot then executes the movements in a repetitive way.

Online programming is rather easy and does not require specialist operator skills. It is not sensitive to accuracy errors. However, it does require access to the robot meaning production is shut down during the programming phase. Moreover, the editing of the program (e.g. to add some parameters, correct a line, or add interactions with other devices) can be limited meaning complex tasks can be difficult to set up. Finally, the robot programs are only stored in the robot memory so can be hard to access. This method can be useful if the robot is required to repeatedly execute a task for a long production period. Programming by language online is also possible: in this case the operator writes the commands directly on the teach pendant, but it can be difficult to realise at that moment all the effects, such as possible collisions.

In *offline programming,* software tools (programming by language or interactive graphical interfaces) are used to generate the program (without the physical involvement of the robot) which is then deployed on the robot controller. With offline programming, editing possibilities increase, the real robot can work during the programming phase so production is not shut down, and start-up and product changeover are faster. However, the robot should be accurate (need for a robot calibration) and specialist operator training is required.

A combination of online and offline programming is also possible.

9.3.3 Programming Languages and Environments

There are many *programming languages* used in robotics that have been adopted over the years. Some examples of the first programming languages used were BASIC, Pascal and LISP. While these languages are now outdated, they provided the basis for more recent industrial robotic languages. Other common scripting languages include C, C++, C#, .NET, Python, Matlab and Java, while an example of a visual programming language is Labview.

Moreover, *many proprietary languages* developed by different manufacturers exist and even for similar robots they can be very different. Currently, almost every robot manufacturer has its own proprietary programming language. Some examples are: RAPID by ABB, MELFA-BASIC by Mitsubishi, KRL by Kuka, AS by Kawasaki, VAL3 by Stäubli, PDL2 by Comau, Karel by Fanuc, Inform by Yaskawa and URScript by Universal Robots.

Manufacturers also provide *programming environments for simulation and offline programming*. The robot and the whole work-cell can be represented as a 3D CAD model. The robot movements, arm collision and tool actions can be simulated and the task defined and optimized. Some examples of manufacturers' programming environments are RobotStudio [19] by ABB and RT Toolbox3 Pro [20] by Mitsubishi.

It would clearly be desirable "*to develop either robotic libraries to be used in the context of consolidated standards or new general-purpose languages for industrial automation applications*" [17]. Indeed, one of the current research and development activities aimed at supporting the delivery of Industry 4.0 is the development or use of standards related to programming tools, modelling and simulation, communication protocols and interfaces.

In recent years, third-party offline programming software has been developed that supports all major robotic brands to provide a common framework useful for simulation and programming of different robots in many applications with a single tool. Some examples include RoboDK [21], Robotmaster [22], Delfoi Robotics [23], FASTSUITE [24] and OCTOPUZ [25].

In addition to commercial products, some open-source solutions exist. Among these, *ROS* (Robot Operating System) is becoming very common. ROS provides a common framework for robotics applications. It is a meta-operating system that provides the services expected from an operating system, "*including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides packages of tools and libraries for obtaining, building, writing, and running code across multiple computers [26]*."

"ROS-Industrial *is an open-source ROS module that extends the advanced capabilities of ROS to manufacturing automation and robotics. The ROS-Industrial repository includes interfaces for common industrial robots, grippers, sensors, and device networks [27]*." It provides software libraries for e.g. automatic 2D/3D sensor calibration and motion planning.

9.4 Robot Control

At the robot level, a control and supervision architecture needs to be implemented, able to interact with the external environment within the CPPS and transform the external stimuli into actuators commands. Moreover, a device is needed to regulate the actions generated by the actuators so that the robot behaves as desired: this device is called a *controller*. It receives the desired robot behaviour as input and, in order to determine the actions of the actuators, it often requires information about the robot status (position, velocity, exchanged force, etc.) that is obtained by reading the data provided by suitable sensors. When this monitoring function is present, this type of control is called *closed-loop control*, otherwise it is called *open-loop control*.

The control system is needed to determine the time history of the forces or torques to be developed by the joint actuators to guarantee execution of the commanded task [28]. In simple terms, for example, if y is the actual position of the robot, y_0 the desired robot position, and u the force or torque generated by the robot motor: the controller has to determine u in order to make the robot move so that y matches y_0.

To better comprehend the subject, one can firstly identify the main elements of a control scheme: the "system" to be controlled and the "controller" that provides a control action to the system. The system to be controlled can be seen as a box that responding to an input $u(t)$ produces an output $y(t)$. The objective is to make the system produce an output as the desired controller input $y_0(t)$ (e.g. to make an axis of a robot move in a desired way).

There is an important distinction in control theory between open-loop control, and closed-loop control i.e. control with a feedback loop. In the open-loop control (Figure 9.4-1), the controller calculates the proper control action $u(t)$ to produce an output $y(t)$ equal to $y_0(t)$ on the basis of the known relation between the input and the output of the system.

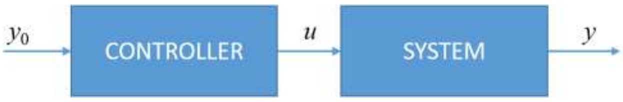


Figure 9.4-1 General scheme of open-loop control.

However, a proper knowledge of the system is often missing, and unpredictable or partially known disturbances (e.g. coupling effects among the joints) can act on the system. Therefore, a common approach consists of closing the control loop with the feedback information on the actual value of the output coming from sensors (Figure 9.4-2). In this way, the controller computes the input to the system in order to reduce as much as possible the error between the desired and actual outputs (e.g. desired position and actual position of the motor).

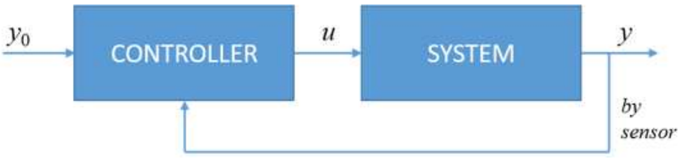


Figure 9.4-2 General scheme of closed-loop control.

In other words, the desired output is compared to the feedback variable provided by a suitable sensor (e.g. encoder), then the controller defines the action on the basis of the error between them. The loop closure allows a better action, such as robustness and reduction of disturbances.

Moving the robot requires the control of different motors (a robot usually has more than one degree of freedom), therefore multi-axis control methods are needed. Depending on the way the different axes are considered and controlled, control strategies can be divided into [28][29]:

- *Decentralized Control*: Each joint is considered independent from other joints and controlled by an independent control loop (Figure 9.4-3).
- *Centralized Control*: In this case the robot is considered as a multivariable system. The dynamic model of the robot is often taken into account (Figure 9.4-4).

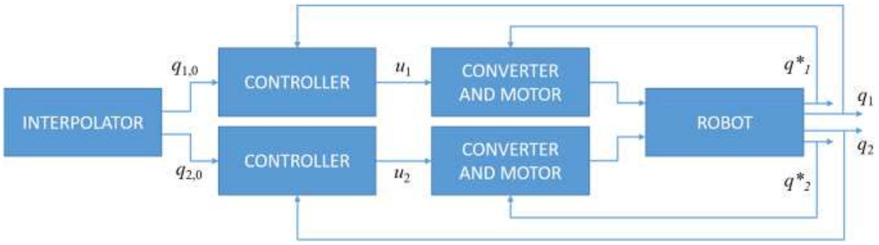


Figure 9.4-3 Decentralized control scheme: example with 2 axes, where q_i (i = 1, 2) represents the joint position and q^*_i the joint velocity. The interpolator block calculates the desired movements required to each axis.

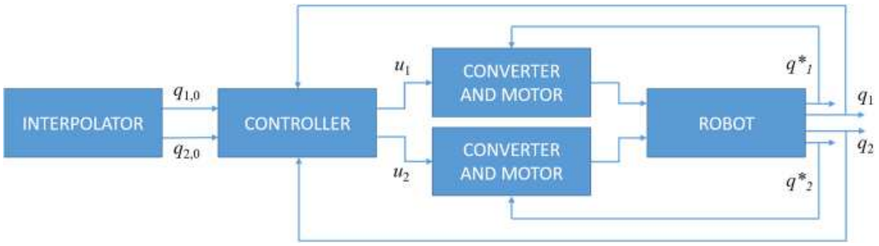


Figure 9.4-4 Centralized control scheme: example with 2 axes, where q_i (i = 1, 2) represents the joint position and q^*_i the joint velocity. The interpolator block calculates the desired movements required to each axis.

In the former case, the actions of each actuator only depend on the actual and desired movements of *that* actuator, while in the case of centralized control the actions of each actuator depend on the desired and actual movements of *all* the actuators.

This also distinguishes the way the dynamic interaction and coupling effects between the joints are considered. In the former case, the dynamic interactions are treated as disturbances, whereas in the latter the coupling effects can be actually taken into account.

Multi-axis control methods often rest on single-axis controllers controlling a motor each, which allows for good performance in many applications. Centralized control strategies are generally implemented when mutual effects between the joints are significant and need to be considered.

Among centralized controllers, different approaches can be considered, such as the *computed torque control* and the *inverse dynamics control*. These controllers are based on the dynamic model of the robot. If this model is not well-known, the performance of the control can scarcely improve. For this reason, a good estimation of the values of the dynamic parameters of the system (e.g. masses, inertias, etc.) is necessary. Such estimation can be achieved by robot dynamic calibration.

Moreover, it is important to note that real robots differ from ideal robots due to the presence of different phenomena including variable inertia, static friction and compliance that have to be taken into account when designing the control scheme.

9.4.1 Force and Vision-based Control

There are cases where a robot is required to exchange forces with the environment, e.g. for pressing a workpiece, for inserting a pin in a hole, or for deburring a workpiece. In these cases, controlling the contact force is more convenient than controlling the motion. Another branch of robotic control is then constituted by *force control* [30][31].

Force control can be divided into:

- *Direct Control*: The force is controlled by closing the force feedback loop. The general scheme of a force controller is illustrated in Figure 9.4-5.
- *Indirect Control*: The force control is achieved by means of motion control (control of a deformation). Impedance control belongs to this category.

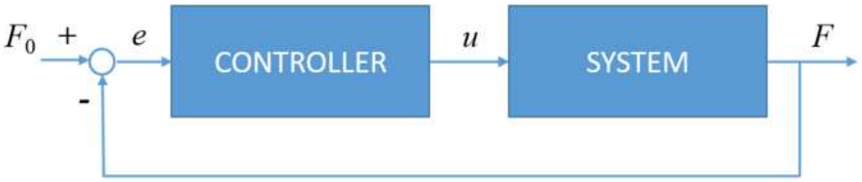
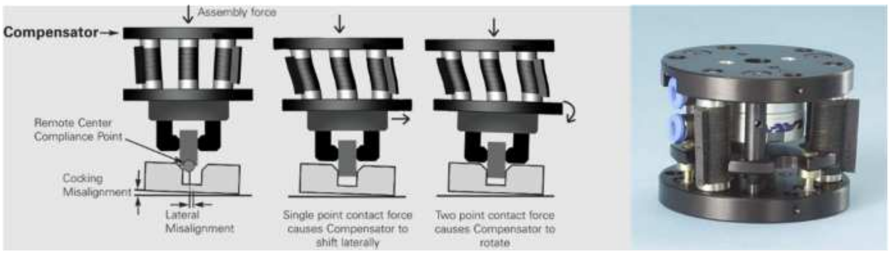


Figure 9.4-5 Direct force control general scheme.

In direct control, the force feedback is provided by a force transducer at the end-effector, at the joints, or at the fingertips of the robot hand when a suitable force is required for grasping and manipulating an object.

It is important to note that it is not possible to control both the motion and the force in the same direction. When the interaction task identifies directions along which the motion has to be controlled and other directions where the force is the controlled variable, *hybrid force/motion control schemes* are used [31][32].

Due to their complexity, sometimes simpler position controllers are combined with a passive compliance on the end-effector, so that small position errors can be compensated by elastic elements. An example is the Remote Center of Compliance (RCC), that is a mechanical device which is commonly mounted between the gripper and the robot's wrist (Figure 9.4-6).



a) b)

Figure 9.4-6 A RCC compensator by ATI Industrial Automation: a) Conceptual scheme[†]; b) Picture of the device[‡]. Image rights: ATI Industrial Automation.

The force feedback is also useful for the *teleoperation* of the robot. In teleoperated mode, the robot is remotely commanded by the operator using a master/slave approach. The idea is that the operator moves a device (the master, e.g. a joystick) and the robot (the slave) acts as in the same way. In this case, *haptic interfaces* can be used to provide the operator with the force/torque feedback [33][34].

Nowadays, vision capabilities are often integrated into a robotic work-cell. One or more vision sensors (cameras) can be mounted on the robot (*eye-in-hand configuration*) or in the work-cell (*eye-to-hand configuration*) to make measurements and supervise the scene (Figure 9.4-7). Therefore, they can provide information about the objects to be processed and the surrounding devices and environment.

[†] Source: https://www.ati-ia.com/Products/Compliance/Compensator_product_desc.aspx

[‡] Source: https://www.ati-ia.com/products/compliance/Compensator_ModelDetails.aspx?id=9116-001-A

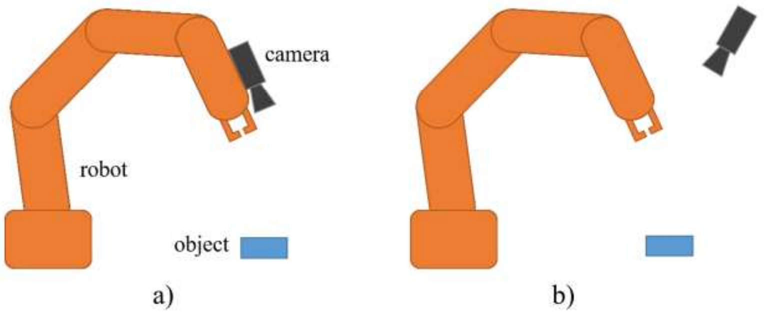


Figure 9.4-7 Configurations of the vision system (simple case of a single camera): a) eye-in-hand; b) eye-to-hand. Image rights: Authors.

In the simplest approach, the robot is commanded in motion control but exploiting the information provided by the camera, implementing the so-called *look-and-move strategy*. As an example, consider a robot that has to pick an object in the working area whose view is provided by a camera. The camera identifies the object and detects its pose which, by means of a proper calibration, can be expressed in the robot coordinate system. This information is then sent to the robot controller that commands the robot to move and pick the object.

When the visual information is used to close the control loop, such a control based on visual feedback is referred to *visual servoing* [35][36]. In other words, "*visual servo control refers to the use of computer vision data to control the motion of a robot*" [35]

The two main approaches to visual servoing are *position-based visual servoing* and *image-based visual servoing* (or hybrid approaches). In both cases the challenge is to move the robot tool into a specific pose relative to the target object. With position-based visual servoing, visual measurements are used to reconstruct the relative pose of the object with respect to the robot, while image-based visual servoing is based on the comparison of the feature parameters of the image of the object between the current and the desired pose [36].

To better understand the two approaches, consider the case of the eye-in-hand configuration, where the robot tool is at a fixed relative pose to the camera and the problem concerns moving the camera.

In position-based visual servoing, a geometric model of the object and camera parameters are used to estimate the object pose relative to the camera. It is then possible to compute the pose error and command the robot to move to correct the pose.

In image-based visual servoing, a model of the object is not needed but it is necessary to detect specific features (e.g. coordinates of points) in the image of the object acquired by the camera. The position of the points in the image depends on the camera pose relative to the object, so that if the camera is moved, the points move in the image. When the camera is in the desired pose, the points appear in known positions in the image, otherwise they appear in different positions. The image-based visual servoing compares the desired and current point features and computes the desired velocity of the camera to move the points where required in the image, therefore the camera to the desired pose.

Image processing algorithms are then essential to extract the image features necessary for controlling the robot.

9.5 Artificial Intelligence

Robotics is one of the fastest growing high technology markets, mainly due to the pervasive introduction of robots in production, and to the growing service and home-care sectors. The actual robotic devices have reached a high level of maturity and reliability in terms of mechanical structures, sensors and actuators, and control techniques to solve basic perception, navigation, and manipulation tasks when working in structured and well-defined environments.

However, to fulfil the I4.0 paradigms robots will need to be more and more versatile, addressing a range of tasks in unstructured and open environments. Therefore, they will need to be equipped with increased cognitive abilities such as knowledge representation, planning, learning, adaptation, and natural human–robot interaction.

Artificial intelligence is a branch of computer science related to the programming and design of hardware and software systems, allowing the machines to have abilities typical of human and animal beings, such as visual, space-time and decisional perceptions.

The concept of intelligence is therefore defined not only as the ability to compute abstract data, but it includes also all those different forms recognized by the Gardner's theory of multiple intelligences [37][36], ranging from spatial to social intelligence, from kinesthetic to introspective.

An intelligent system is created by trying to recreate one or more of these different forms of intelligence. Although often defined as exclusively human or natural, they can be simplified to particular behaviors reproducible by some machines.

Artificial intelligence is already largely exploited in everyday life. For example, the various voice recognition tools regularly used in many applications from smartphones to security systems, are based on machine learning algorithms, which are a form of AI.

Another popular application of machine learning and artificial intelligence paradigm is in the automotive sector, where advanced speed change systems in semi-autonomous driving cars make use of AI algorithms based on fuzzy logic [38]. Moreover, AI is playing a key role in the development of autonomous driving cars, although many technical and ethical challenges still need to be addressed in order to have a reliable and safe self-driving vehicle.

In the medical field, artificial intelligence mainly uses neural networks, with applications in the analysis of the heartbeat, in the diagnosis of some cancer forms, and in the creation of new therapeutic drugs. Recently, AI techniques have been exploited to develop tools for predicting the coronavirus pandemic, and the early detection and diagnosis of infections [39].

Further sectors in which artificial intelligence is used on a regular basis are the stock market, image recognition (including facial recognition), social media marketing, and robotics. In addition, intelligent systems are also used to further improve many sectors of information technology itself, including proving the correctness of algorithms and creating new learning methods.

9.5.1 Historical Overview

In 1954, George Devol built the first programmable robot, called Unimate. It was a hydraulic manipulator arm that could perform repetitive tasks. It was bought by General Motors in 1961 for use in automobile assembly lines to automate metalworking and welding processes. Although in light of today’s AI performance its degree of intelligence is very rudimentary, at the time of its invention it was considered by many as one of the few AI products with commercial value.

Since these early developments, innovative works have been done, specifically improving the robotic capabilities for sensing and perception (mostly through machine vision, but also adding force feedback, tactile abilities, proximity sensors and so on) thus increasing the capabilities of the machines to interact with the surrounding environment. Already in the late 1970s humanoid robots were developed that could stack blocks. The further improvements in ICT hardware and AI technologies boosted the development of more and more advanced robotic systems.

From the coining of the term AI in 1956 until the early1970s, the enthusiasm about AI gave birth to several research subfields, which constitute the foundation of the modern theory of AI: rules-based systems, machine learning, Natural Language Processing (NLP), shallow and deep neural networks, image processing and computer vision, Speaker Recognition and Speech to Text Processing. However, practical applications of AI based programs were still very uncommon and largely limited to solving rudimentary problems, mainly due to limitations in the available computing power.

Moreover, most of the early predictions about the future of strong artificially intelligent machines were not yet realized, making investors skeptical and reluctant to invest more funding. This resulted in a bust phase of AI. Most of the limited number of inventions from this period, such as backpropagation and recurrent neural networks, went largely overlooked and substantial effort was spent to rediscover them in the subsequent decades.

The recent advancements in information and communication technologies gave new life to research into AI, as hardware and network connectivity became cheaper and faster; parallel and distributed systems became practical, and huge amounts of data (Big Data) were easily stored and then available for training AI systems.

9.5.2 Techniques

The use of algorithms capable of reproducing the reasoning of human beings or animals in different situations allowed intelligent systems to further improve their behavioral skillset. Research efforts focused on the development of algorithms which could imitate different behaviors depending on environmental stimuli (the skill often called context-awareness).

These complex algorithms implemented within intelligent systems are therefore able to make decisions or choices according to the context where they are applied. In the case of intelligent vehicle systems, for example, a car without a driver can decide in the case of danger whether to steer or brake according to the situation. For example, depending on whether the information sent by the various sensors such as car speed and direction, size and position of other objects, and ground and weather conditions, it calculates if braking or steering has a higher percentage of safety for the driver, passengers, and pedestrians. The basic knowledge originally embedded in the system is than expanded, created through experience.

A specific sector has grown, focused on the representation of knowledge, which studies all the mechanisms of human reasoning and defines methods to make this knowledge comprehensible to machines, through language and increasingly precise and detailed controls. Transferring existing human knowledge to machines requires transferring not only domain knowledge, but also experiences. This adds the possibility of understanding new information by exploiting the knowledge already present in the starting system.

This information is provided to the machine in various ways, the most important of which are those based on the *Theory of Formal Languages* (which studies the syntactical aspects of structured languages), and the *Theory of Decision*, to identify optimal decisions under specific circumstances.

One of the main steps forward in AI development was made when algorithms were created that were capable of learning through experience just like humans. Learning makes the system capable of improving the ability of the machine to act and make decisions.

Developing algorithms that can learn from their experiences and mistakes is essential to creating intelligent systems that operate in situations for which programmers cannot foresee all the future possibilities.

Through *machine learning,* also discussed in Chapter 5, a machine can learn to perform a certain action even if this action has never been programmed among the possible actions. For example, a robot can perform a peg in hole assembly operation even if the shape and position of the hole differ from the ones it was trained with, or can recognize an object to pick even if that exact image was not previously stored in its memory.

Machine learning is probably the most common branch of artificial intelligence considered in mainstream culture, and constituted the basis of several successful literary works. However, the research is still far from the scenarios represented in science fiction movies and literature, and develops along both theoretical and practical paths in computational learning theory and pattern recognition.

Machine learning was enabled by the development of *artificial neural networks*, i.e. a particular mathematical model inspired by brains and neurons. The name *neural network* derives from the fact that this mathematical model is characterized by a series of interconnections between artificial neurons, and these connections are necessary for the different calculations. Just like biological neural networks, an artificial neural network also has the characteristic of being adaptive to the varying needs deriving from the different information obtained in the different learning phases.

9.6 Ethical Issues

As the degree of autonomy and decision making capability of automatic systems increases, it is essential to define a code of conduct, a list of rules that have to be respected in order to avoid unpredicted behaviours. A typical case is that of autonomous guided vehicles, such as industrial AGVs used for logistic applications, drones and autonomous cars. It is very likely that, in a near future, these concepts will be extended also to robots acting in the industrial environment.

Robot ethics is defined as the set of rules and the code of conduct which is embedded in the robot control system by the designers.

The first set of rules (the well-known *three laws of robotics*) was defined by the science fiction writer Isaac Asimov [40] and the scientific community considers that they are still valid today:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

He later added also a *Law 0*: No robot may harm humanity or, through inaction, allow humanity to come to harm.

However, these laws are not sufficient and there might be scenarios where they cannot be fully obeyed.

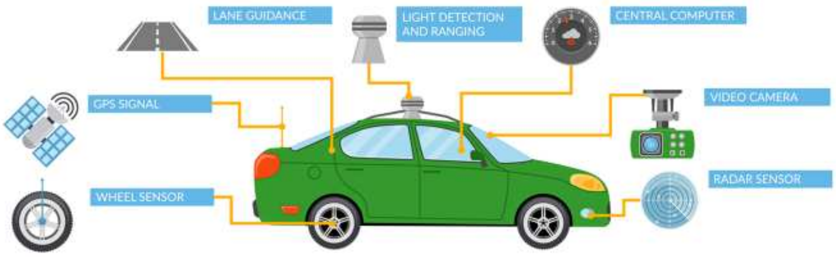


Figure 9.6-1 Autonomous car set-up[§].

The three laws imply that a robot has sufficient intelligence (in terms of perception and cognitive abilities) to also make moral decisions in complex situations, which is still not the case. Therefore, they cannot provide a practical basis for writing a robot code of conduct. An example of a set of rules easily embeddable in a robotic system which was proposed by robo-ethical scientists is:

1. Do not kill
2. Do not cause pain
3. Do not disable
4. Do not deprive of freedom
5. Do not deceive
6. Keep your promise
7. Do not cheat
8. Obey the law
9. Do your duty

On the ethics side, several moral dilemmas need be addressed.

Figure 9.6-2 shows a typical moral dilemma, which could be hardly embedded in the machine’s software code: knowing that there are only two possible events – crashing into the people crossing the street and killing three of them with two injured or crashing the car into the barrier and killing four passengers – which alternative should be chosen? Obviously, this would be a hard decision also for a human driver, with many different factors to be considered.

Several research projects are on-going in the field and many other examples of moral dilemmas can be found at http://moralmachine.mit.edu/, an online experimental platform designed to explore the moral dilemmas faced by autonomous vehicles and to collect human opinions on how machines should make decisions when faced with these moral dilemmas.

[§] Image design by Freepik.

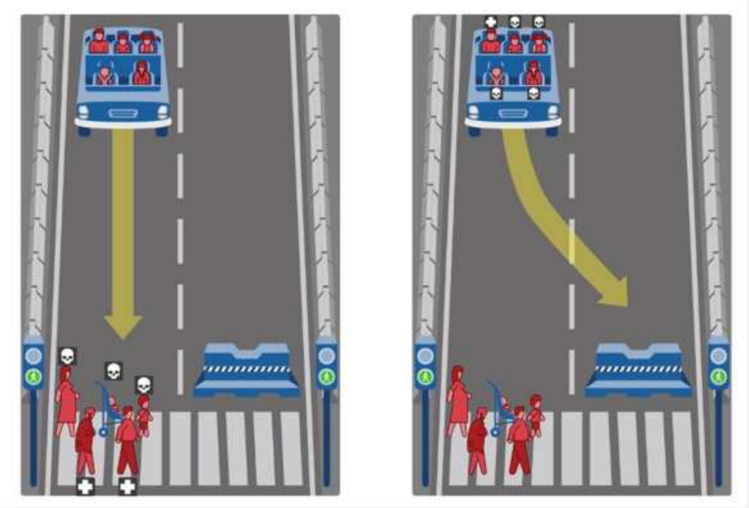


Figure 9.6-2 Driverless car moral dilemma [41].

These topics have also been used as a background principle in the context of the study commissioned by the European Parliament’s Legal Affairs Committee “*to evaluate and analyse, from a legal and ethical perspective, a number of future European civil law rules in robotics*”. In February 2017 the subsequent European Parliament Resolution “Civil Laws in Robotics” was adopted, covering a wide range of different areas, including liability rules, ethical questions, standardisation, safety and security, data protection and intellectual property rights, autonomous vehicles, drones, care robots, medical robots, human enhancement and education and employment and a code of ethical conduct for robotics engineers and a code for research ethics committees. Moreover, it proposes an EU Agency for Robotics and Artificial Intelligence.

The reaction from the robot manufacturing companies was quite fast, showing anxiety that this regulation could hamper the progress and threaten the competitiveness of the entire manufacturing industry. Several social studies have also been carried out to assess the impact of these rapidly evolving technologies on the job market, and a strong debate is still ongoing in newspapers, social media and international organizations.

Indeed, on May 22, 2019 the Organisation for Economic Cooperation and Development adopted the OECD Council Recommendation on Artificial Intelligence [42], which sets principles for responsible stewardship of trustworthy Artificial Intelligence systems and recommendations to national governments for national policies and international cooperation. These principles are based on: inclusive growth, sustainable development and well-being, human centred values and fairness, transparency and explainability, robustness, security and safety; accountability.

9.7 Conclusions

The innovative technologies of Industry 4.0 enable a new generation of factories where industrial robots perform autonomously all tasks while the workers have mainly a supervisory role. Thanks to the improved sensing capabilities and real time control the cooperation between human and robot become easier and safer, enhancing the productivity, production quality and flexibility. The cyber physical systems provided with artificial intelligence not only monitor the entire production process, but make predictions and decisions in order to improve the efficiency of the production.

A lot of progress has been achieved in the last decades, but several challenges have still to be faced both in the physical (hardware) and cyber (software) world, improving sensors, actuators, controllers, data storage and analysis, and artificial intelligence algorithms in order to achieve the new generation of factories envisioned by the Industry 4.0 paradigm. The interconnection requires a *standard communication system*, proprietary languages, thus, are proposed to be replaced by open source solutions; human-robot collaboration needs *advanced perception skills* to work safely in unstructured spaces and *psycho-social aspects* of this collaboration from the point of view of the workers have to be wisely considered; intelligent machines able to make decisions introduce serious *ethical issues*, so appropriate laws have to be defined and implemented in the robots; CPSs are exposed to security threats and the transmission and sharing of data necessitates new levels of *cyber security and privacy*; the modularity and configurability needed for fully customized products requires *new business models*; all the necessary transformations require corresponding changes in workforce skills and organizational structures, which can be achieved with a transformation in the *educational system,* offering not only new contents but also new methodologies for the new generation of employees. Industry 4.0 is an industrial revolution evolving at exponential pace with a huge impact on the quality of life for populations all over the world. A conscious awareness of such impact is essential to benefit the most from this breakthrough since it could also yield to greater inequality. The immediate benefits will be mainly for innovators and investors, the job market will be strongly affected increasing the demand of highly skilled workers and providing few offers to workers with less education and lower skills. Digital interconnections will also shorten the distances between people providing new opportunities for cross-cultural exchanges, improving understanding and cohesion. However, also unrealistic expectation and extreme ideologies will spread faster and with no control accessing more vulnerable people, who could also be victims of privacy violations and stolen data. Therefore, the opportunity offered by the fourth industrial revolution has to be globally shared and shaped toward a future with common objectives and values, to benefit powerless people rather than increasing the gap between the social classes.

9.8 References

[1] N. Jazdi, "Cyber physical systems in the context of Industry 4.0", in Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, 22–24 May, pp. 1- 4, 2014.

[2] E. Geisberger, & M. Broy, (Eds.). agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems Vol. 1. Springer-Verlag.; ISBN 978-3-642-29098-5, 2012.

[3] D. Zuehlke, "SmartFactory Towards a factory-of-things", *Annual Reviews in Control*, vol.34, pp.129-138, 2010.

[4] A. Khalid et al. "A methodology to develop collaborative robotic cyber-physical systems for production environments", *Logistic Research* 9:23, 2016.

[5] I. Navarro and F. Matía, "An introduction to swarm robotics". *International Scholarly Research Notices*, 2013.

[6] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering", in Proceedings of the 9th conference on autonomous robot systems and competitions Vol. 1, No. CONF, pp. 59-65, IPCB: Instituto Politécnico de Castelo Branco, 2009.

[7] M.H.M. Alkilabi, A. Narayan, C. Lu, and E. Tuci, Evolving group transport strategies for e-puck robots: moving objects towards a target area. In Distributed Autonomous Robotic Systems, pp. 503-516 Springer, Cham. 2018.

[8] M.H.M. Alkilabi, A. Narayan, and E. Tuci, "Cooperative object transport with a swarm of e-puck robots: robustness and scalability of evolved collective strategies", *Swarm intelligence*, 11(3-4), pp.185-209, 2017.

[9] F. Perez-Diaz, R. Zillmer, and R. Groß, "Control of synchronization regimes in networks of mobile interacting agents",. *Physical Review Applied*, 7(5), p.054002, 2017.

[10] http://www.e-puck.org/

[11] https://www.youtube.com/watch?v=t8wbicHdRFk

[12] https://www.youtube.com/watch?v=E7jRYvEFytE

[13] https://www.youtube.com/watch?v=4DKrcpa8Z_E

[14] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics - Modelling, Planning and Control, Chapter 4, Springer-Verlag London Limited DOI 10.1007/978-1-84628-642-1, 2010.

[15] G. Legnani, I. Fassi, Robotica Industriale, Chapter 9, CittàStudi, 2019.

[16] M. Hägele, K. Nilsson, J.N. Pires, R. Bishoff, Industrial Robotics, in Springer Handbook of Robotics, Chapters 54, Siciliano, B. Khatib, O. (Eds.), 2nd edition, DOI 10.1007/978-3-319-32552-1, 2016.

[17] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics - Modelling, Planning and Control, Chapter 6, Springer-Verlag London Limited DOI 10.1007/978-1-84628-642-1, 2010.

[18] G. Legnani, I. Fassi, Robotica Industriale, Chapter 11, CittàStudi, 2019.

[19] https://new.abb.com/products/robotics/robotstudio

[20] https://us.mitsubishielectric.com/fa/en/products/industrial-robots-melfa/engineering-software/rt-toolbox3-pro/rt-toolbox3#

[21] https://robodk.com/

[22] https://www.robotmaster.com/en/

[23] https://www.delfoi.com/web/products/delfoirobotics/en_GB/off-line_ENG/

[24] http://www.fastsuite.com/en_EN/fastsuite/why-fastsuite.html

[25] http://octopuz.com/

[26] http://wiki.ros.org/ROS/Introduction

[27] https://rosindustrial.org/about/description/

[28] B. Siciliano, et al. *Robotics - Modelling, Planning and Control*, Chapter 8, Springer-Verlag London Limited 2010. DOI 10.1007/978-1-84628-642-1

[29] G. Legnani, I. Fassi, Robotica Industriale, Chapter 10, CittàStudi, 2019.

[30] L. Villani, et al. *Force Control*, Chapter 9, in *Springer Handbook of Robotics*, Siciliano, B. Khatib, O. (Eds.), 2nd edition, DOI 10.1007/978-3-319-32552-1, 2016.

[31] B. Siciliano, et al., Robotics - Modelling, Planning and Control, Chapter 9, Springer-Verlag London Limited DOI 10.1007/978-1-84628-642-1, 2010.

[32] Kevin M. Lynch and Frank C. Park, Modern Robotics: Mechanics, Planning, and Control, Chapter 11, Cambridge University Press, ISBN 9781107156302, 2017.

[33] B. Hannaford et al., Haptics, Chapter 42, in Springer Handbook of Robotics, Siciliano, B. Khatib, O. (Eds.), 2nd edition, DOI 10.1007/978-3-319-32552-1, 2016.

[34] G. Niemeyer et al. Telerobotics, Chapter 43, in Springer Handbook of Robotics, Siciliano, B. Khatib, O. (Eds.), 2nd edition, DOI 10.1007/978-3-319-32552-1, 2016.

[35] F. Chaumette et al. Visual Servoing, Chapter 34, in Springer Handbook of Robotics, Siciliano, B. Khatib, O. (Eds.), 2nd edition, DOI 10.1007/978-3-319-32552-1, 2016.

[36] B. Siciliano, et al. Robotics - Modelling, Planning and Control, Chapter 10, Springer-Verlag London Limited DOI 10.1007/978-1-84628-642-1, 2010.

[37] H. Gardner, Frames of mind: The Theory of Multiple Intelligences, 1983.

[38] Fuzzy Logic (FL) is a method of reasoning that resembles human decision making process, working on the levels of possibilities of input to achieve the definite output, helping to deal with uncertainties in engineering. It can be implemented at both hardware and software level.

[39] C. Menni, et al., “Real-time tracking of self-reported symptoms to predict potential COVID-19”, Nature *medicine*, *26*(7), 1037-1040. doi.org/10.1038/s41591-020-0916-2, 2020.

[40] I. Asimov, Runaround, 1942.

[41] E. Awad, S. Dsouza, R. Kim, J. Schulz, J. Henrich, A. Shariff, J.-F. Bonnefon, I. Rahwan, “The Moral Machine experiment”, *Nature*, 563, pp.59–64, 2018.

[42] OECD, Recommendation of the Council on Artificial Intelligence, OECD/LEGAL/0449

Digit-T
DIGITAL MANUFACTURING TRAINING

Co-funded by the
Erasmus+ Programme
of the European Union

Digital Manufacturing Training System for SMEs (Digit-T)
Project ref: 2017-1-UK01-KA202-036807