



3

AI and ML for Human-Robot Cooperation in Intelligent and Flexible Manufacturing

Manuel A. Ruiz Garcia, Erwin Rauch, Renato Vidoni,
and Dominik T. Matt

3.1 Introduction

The last decades were marked by the further development of modern production systems and the introduction of Industry 4.0 in manufacturing. The concepts and technologies of Industry 4.0 are mostly aimed at the networking of production and the efficient design of production systems. The logic of Industry 4.0 foresees humans and

M. A. Ruiz Garcia · E. Rauch (✉) · R. Vidoni · D. T. Matt
Industrial Engineering and Automation (IEA), Faculty of Science and
Technology, Free University of Bozen-Bolzano, Piazza Università, Bolzano,
Italy
e-mail: Erwin.Rauch@unibz.it

M. A. Ruiz Garcia
e-mail: ManuelAlejandro.RuizGarcia@unibz.it

R. Vidoni
e-mail: Renato.Vidoni@unibz.it

D. T. Matt
e-mail: Dominik.Matt@unibz.it

robots as indistinguishable parts of a larger heterogeneous body of distributed autonomous and cooperative entities. Under such a perspective, robots are endowed with self and environment awareness and are able to smartly interact with both humans and other machines (Ruiz Garcia et al. 2019). Consequently, and in contrast to the third industrial revolution, machines are not intended to substitute humans in industry, but to work with them in synergy.

In collaborative industrial scenarios, safety greatly depends on the reciprocal understanding between the human operator and the robotic system. In particular, the most dangerous risk specific to robots is the unexpected collisions between the robot and the environment (Siciliano and Khatib 2016). When an unexpected exertion occurs between a collaborative robot and its surrounding environment, impact forces are eased thanks to their lightweight design and compliant mechanisms and control. However, avoiding unexpected force exertions implies foreseeing dangerous situations, and thus it relies on sensing, situational awareness, planning and decision-making capabilities. Therefore, without suitable exteroceptive sensing a collaborative robot cannot be considered as a safe companion in the context of human-robot cooperation (HRC). In other words, to safely interact with a human operator and the environment, a collaborative robot must predict and prevent any risky circumstances based on its own situational awareness. That is, the robot must identify, understand and forecast operator's actions and environmental changes to promptly react and safely adapt to either expected or unexpected operative conditions. On the other hand, the operator needs to be aware of the collaborative robot's motion to guarantee him or her own safety. Therefore, in the context of HRC, beyond the sensing capabilities a collaborative robot also needs to be endowed with suitable means of interaction so to constantly inform the human operator about what are the current and future goals and actions to be reached and performed, respectively, on a finite time horizon.

D. T. Matt

Innovation Engineering Center (IEC), Fraunhofer Research Italia s.c.a.r.l.,
Bolzano, Italy

Another important aspect is that Industry 4.0 is also seen as an enabler for the flexibilization of production systems and, thus, it potentially represents an important milestone for multi-variant manufacturing. In this concern, mass customization can be defined as the capability to deliver products and services that best meet individual customers' needs with near mass production efficiency (Tseng et al. 1996). Such a diversification in production requires to manage not only the inner product variety, but also the induced process variety due to differences in assembly sequence and the necessary changes of the manufacturing system required to handle them. A natural way to achieve this goal is through the flexibilization of manufacturing systems, such to allow changing from one product to the other without the need to stop the production for a changeover nor including other manual adaptations of the manufacturing system. It is worth noticing that such an automated adaptation in the context of collaborative manufacturing greatly resembles the ones required by a collaborative robot to establish a safe HRC. Indeed, the understanding and the forecasting of the operator's actions and environmental changes, in terms of the current product variant, provide all the necessary information required for the definition of such an adaptation. On the other hand, the automation of such an adaptation relies on planning and decision-making capabilities.

Therefore, in abstract terms, the definition of a safe HRC and the automated adaptation of a multi-variant collaborative manufacturing system represent two particular instances of a general problem. This chapter is devoted to the deconstruction of such a general problem in terms of three smaller perceptive and cognitive issues: *scene monitoring*, *task modelling* and *planning*.

3.2 Artificial Intelligence and Machine Learning

Since the beginning of the twenty-first century, there has been a widespread use of ML techniques, specially Deep Learning (DL) ones, in the analysis of large amounts of data so to automatically drawn conclusions from it. Since then ML and DL, together with AI, are now terms

belonging to the common imagination. However, there seems to be a common believe that AI, ML and DL refer to the same—or nearly the same—concept. In some particular rhetorical circumstances, this could be the case, but in general terms such a concept overlap is totally misleading. The aim of this section is twofold. On the one hand, to briefly clarify what is the scope of each research field and to highlight the relationships between them. On the other, to identify the key general problems that such techniques can potentially solve in the context of collaborative manufacturing.

3.2.1 What's Artificial Intelligence?

As a starting point, one can state that DL is a subset of ML, and that at the same time ML *seems to be* subset of AI. Therefore, it comes natural to start with the definition of AI. However, due historical reasons that fall beyond the scope of the present chapter, it is not possible to provide a “gold-standard” definition of AI unless one assumes that some background on the field is already known. So let us start instead with a brief digression on what an artificial system should do to be considered as *intelligent*. First of all, it is worth noticing that intelligence can be conceived either in terms of reasoning (thinking) or behaviour (acting). On the other hand, one can build a comparison metric of intelligence with respect to the human performance or with respect to an ideal model of intelligence, commonly known as *rationality*. Therefore, an artificial system can be considered as intelligent if it (Russell and Norvig 2010):

- 1 *Acts like a human* (Turing test approach). An artificial system acting like a human should be able to fool a human interrogator, who cannot distinguish if the answers are being provided by a computer or by a human. However, this evaluation mechanism implicitly assumes that the artificial system is already equipped with all the necessary means for communicating naturally and understanding the interrogator questions. Clearly, this approach doesn't logically scale, since providing such necessary means would require solving some general AI problems beforehand.

- 2 *Thinks like a human* (cognitive modelling approach). Whether an artificial system is able to think or not like a human, depends on the availability of an accurate theory or model of the mind, which can only be defined by experimental evaluation and validation either with human or animals. Although closely related to AI, all of such cognitive research efforts are totally out of scope.
- 3 *Thinks rationally* (laws of thought approach). To understand if an artificial system thinks rationally, an irrefutable reasoning process needs to be known. In this regard, the *formal logic* was introduced to study the inference in abstract (or formal) content. Based on such theories, the classical AI approach assumes that intelligent systems can be built on top of computer programs that search without exhaustion for a solution of given a set of problems stated in logical notation. Unfortunately, one key limitation of this approach is that it is difficult to model knowledge uncertainty, thus reality. On the other hand, computational resources can be easily exhausted when performing some (general) reasoning steps.
- 4 *Acts rationally* (rational agent approach). An artificial systems act rationally when focused on achieving a goal given a set of beliefs. Therefore, acting rationally implies perceiving, then acting, or equivalently, it implies mapping perceptual inputs or *percepts* into *actions*. Any artificial system able to perceive and act is what is called an *agent*. Here rationality is concerned with a success expectation in terms of what has been perceived—in contrast to the laws of thought approach, where rationality implies making correct inferences. As a result, a *rational agent* performs actions that are expected to maximize a *performance measure*, given a designated goal, a sequence of percepts and whatever *built-in knowledge* it may have. We observe that *causality* is a necessary condition for rationality.

Based on the latter approach, AI can be defined as the branch of computer science concerned with the study and development of rational agents. In particular, AI deals with the different ways to represent and implement how a rational agent maps percepts into actions. Consequently, AI aims to develop algorithms that, given the properties of the environment and the agent's structure, produce rational behaviours. In

the rest of the section, *agent* will always refer to a *rational agent* unless stated otherwise. The environmental properties can be summarized as follow:

- **Observability:** an environment is *fully observable* if the agent's sensors allow reconstructing the whole state of the environment at each time instant; *partially observable* if the only part of the state can be reconstructed; *unobservable* if the agent has no sensors.
- **Predictability:** an environment is said to be *deterministic* if its next state can be uniquely determined in terms of its current state and the executed action by the agent; *nondeterministic* otherwise. One particular case of nondeterministic environments is the *stochastic* one, where the possible outcomes of actions are characterized by probabilities. In most practical scenarios partially observable environments are treated as stochastic ones. Therefore, an environment is *uncertain* if it is either partially observable or non-deterministic.
- **Staticness:** an environment is said to be *dynamic* if it changes while the agent is deliberating; *static* otherwise. It is worth noticing that a dynamic the environment may change either autonomously (time-variant) or due to the actions executed by the agent. If the environment changes only due to the agent's actions, then it is said to be *semidynamic*.
- **Discreteness:** the environment's state evolution can be either *continuous* or *discrete*. In total analogy, also the agent's percepts and actions can be of either type.
- **Knowledge:** in a *known* environment, the consequences of executing an action (either the outcomes itself or the outcomes' probabilities) are well understood by the agent. That is, in a known environment the agent understands the "laws" governing the environment's evolution. When those "laws" are missing from the agent's knowledge, the environment is said to be *unknown*. The environment's knowledge property is independent from its observability. It is worth noticing that in the case of an unknown environment the agent must *learn* the way it works to be able to make decisions.
- **Episodicness:** an *episodic* environment does not depend on the actions taken previously; such an environment can be split as a series of

independent one-shot actions overtime. In contrast, a *sequential* environment depends on previous actions. That is, its current state is determined by past actions.

- **Agency:** an environment can be single agent or multi-agent. In the latter case, agents can either *cooperate* to reach a common goal or *compete* to conclude their individual goals or a mix of both.

As an example, an autonomous driving agent deals with a partial observable (it is not always possible to fully observe all pedestrians, vehicles or other entities on the road), stochastic (it is not possible to fully predict how such entities are going to move next), dynamic (entities' states evolve in time), continuous (likewise the rest of our world), known (pedestrians or other vehicles are not expected to fly), sequential (as a result of its continuity) and multi-agent (entities act on their own free will) where agents follow both common (e.g. avoid collisions) and individual goals (e.g. reach home on time).

The agent structure is defined by the way percepts are mapped into actions in order to achieve a goal. In particular, one can identify:

- **Reflex agents:** this type of agents execute one single action a time, given either the current percept or the whole percepts sequence. When the reflex agent relies only on the current percept to make a decision, the agent's structure is defined by a set of *condition-action rules*. On the other hand, when the agent deliberates what to do next based on the whole (or partial) percept sequence, its structure is given by an *internal model* representation of the environment together with a set of condition-action rules. Therefore, reflex agents are not concerned with the implications of their actions, they simply act as prescribed by their built-in rules. In such a sense, the goal of a reflex agent is implicit and uniquely determined for each environmental state.
- **Goal-oriented agents:** agents of this type are provided with some extra information, specifying what's the expected final or target configuration of the environment. Therefore, goal-oriented agents cannot rely on a set of condition-action rules to make decisions. On the other hand, they necessarily need to be aware about the implications each of their actions could lead. Also, they may need to execute more than one

action to actually achieve one particular goal. In general, however, it is not possible to guarantee that a goal-based agent will succeed with all given goals, even through the execution of an infinite number of actions. First, some goals may not be *reachable* from the current environmental state (unfeasible or due uncertainty). Second, goals may be conflicting in between. Also, when multiple action sequences allow to reach the same target, the goal based agent lacks a rational way to decide which sequence to execute.

- **Utility-based agents:** in the aforementioned cases where a goal based agent fails to succeed, the agent can, instead of exactly achieving a set of goals, try to execute the set of actions that maximize a given *utility* function, which specifies the appropriate trade-offs between them. Such an utility function represents the agent's internalization of the rationality's performance measure. It is worth noticing that in the case when different sets of action sequences allow to reach the same result, the utility function can be used to discern what's the best sequence among them.

Not all agent structures are appropriate for dealing with all types of environments. On the other hand, not always an utility-based agent will perform better than another agent with a simpler internal structure. This will depend mostly on environmental properties and the agent's adaptation to the environmental changes—in practice, it is impossible to have a perfect built-in knowledge of the environment. For example: modern collaborative robots implement a reflex agent to suddenly stop the robot motion when the external force exertions are above a predefined threshold to guarantee a safe physical interaction; in this applicative context the reflex agent guarantees the smallest decisional latency, thus minimizing the risk of damage to the environment or robot. In contrast, a trajectory planner implementing a reflex agent based on artificial potential fields may fail to reach the desired goal when getting trapped on a local minima.

Agent structures and environment states can be decomposed into a finite set of fundamental units or blocks. For example, one can encapsulate all perceptive aspects of an agent into a sensing unit. Each such an unit can be seen either as black box (*atomic representation*) or as a set

of variables and attributes (*factored representation*) or as a set of interacting objects (*structured representation*). Based on the environmental properties, agent's structure and the ways of representing them, it is now possible to identify what are the basic AI problems and the algorithms and techniques to solve them.

Planning agents seek to identify and execute a sequence of actions to reach their objectives. In terms of the environmental properties, we can identify four major categories of planning agents:

- *Problem-solving agents*: use an atomic or a domain dependent factored representation of the environment. This kind of agents rely on general searching algorithms: depending on the environmental properties, the agent can use blind search, heuristic search, local search or adversarial search; in the case of factored representations, the problem-solving agent can take advantage of constraint satisfaction search. A clear limitation of atomic representations is that the searching algorithm cannot exploit any knowledge contained on atomic black boxes, that is, there's no room for *inference*. Example of problems that can be solved with this type of agents is the VLSI layout design and the classical travelling salesman.
- *Logical agents*: take advantage of a domain-independent structured representation of the environment. This allows to split the agent's structure into a representation unit (*knowledge base*) and a reasoning unit (*inference engine*). The knowledge base (KB) contains all domain-specific content, but it is stored as a set of formal (abstract, logical) *sentences* or statements expressed according to the syntax of a *representation language*. Each sentence can result either true or false, depending on the *model* used to evaluate it. Models are the mathematical abstraction of any possible environmental state. The inference engine allows to derive new sentences from the old ones in terms of *logical entailment*, that is, new sentences logically follow from the old ones. Such a logical reasoning can be done either in terms of *model checking* or *theorem proving*.
- *Classical planning agents*: in contrast to logical agents, which rely on a structured variable-free representation, planning agents use a factored representation of the environment in terms of *state variables*. This

leads to a more flexible and succinct representation for actions, goals and plans, through the introduction of specific planning languages for representing the KB. This kind of agents relies on specific searching algorithms that, depending on the environmental properties, can be state-space search, planning graphs or hierarchical search.

- *Rational planning agents*: when dealing with uncertain environments, all previously described agents keep track of what is called the *belief state*, that is, the set of all possible environmental states logically explaining the observations. In turns, solving a planning task on an uncertain environment implies considering all possible explanations, no matter how *unlikely* they might be. Clearly, finding solutions on large search spaces becomes unfeasible with such agents. Another important limitation is given by the *qualification problem*: in logical terms it is not possible to specify all preconditions required for an action to succeed. In other words, it cannot be deduced whether an unexpected exception happens or not and, when such an exception happens, the plan's outcome cannot be inferred. Therefore, a rational decision must take into account both, the relative significance between goals (*utility*) and the prospect whether they will be achieved or not (*probability*). In particular, rational decisions maximize the expected utility when averaged over all of the possible outcomes of the action. These represent the bases of the *probabilistic reasoning*. Basic algorithmic approaches for implementing such type of reasoning are Bayesian networks, sampling-based methods for approximate inference and fuzzy logic. In case of partial observability, one can take advantage of hidden Markov models, Kalman filter or dynamic Bayesian networks to reconstruct the current environmental state. Rational agents immerse in episodic environment can make use of decision networks or their dynamic extension in case of sequential environments, which are modelled as (partially observable) Markov decision processes.

Perception is the process of extracting information about the environment from the sensors data. Although there's a large variety of sensing technologies providing sensory modalities, the most of the AI research efforts have been focused on *vision* (computer vision) and *speech* (natural

language processing). Agents require perception to improve their knowledge of the environment and thus to achieve their goals; perception is not an end by itself. In general terms, an agent needs to identify what aspects of the perceptual stimulus actually bear or not relevant information. In general, there are three different approaches that can lead to this identification:

- *Feature extraction*: feature extraction refers to the process where raw data measurements are converted into a low-dimensional vector of numerical values, bearing the same informative content of the original measurements. Due to the dimensionality reduction, features are intended to be not only informative but also non-redundant. Nowadays manual or hand-crafted feature extraction is no longer a common practice in applied sciences, due to the advancements of machine learning algorithms (some of them listed on Sect. 3.2.2) together with the availability of large public datasets. Classical examples of feature extraction procedures could be the identification of the principal axes of a data cluster and the computation of the intensity histogram of an image.
- *Pattern recognition*: implies the automatic identification of regularities on data that are representative of some properties of the environment. Depending on the application context and nature of the perceptual information, a pattern recognition strategy can be applied directly to the raw measurements or to the features representations. As in the case of features extraction, nowadays pattern recognition problems are solved by means of machine learning algorithms. Some common examples of pattern recognition applications include automatic tumour identification from medical images, speech recognition, spam filtering and face detection.
- *Reconstruction*: refers to the direct inference of physical properties of the environment in terms of the measured data. For example, in the case of images, a reconstruction problem could be to infer the depth of each pixel. In the case of audio signals, to localize the source given a distributed array of measurements. Also the agent's velocity estimation given a sequence of range scans is a particular instance of a reconstruction problem (*state estimation*). In general, reconstruction

problems require specific algorithms to be solved. Despite, there are many successful application of machine learning algorithms on specific reconstruction tasks.

Natural language processing (NLP) deals with structured representations of the language and aims either, to acquire knowledge from data (audio or text) given in natural language, or to naturally communicate with humans or other agents. *Information-seeking tasks* rely on a *language model* (*n-gram*) based on characters or words, to predict the probability distribution of the language expressions. *Categorization* of documents can be effectively implemented using naive Bayes *n-gram* models or general classification algorithms (some of them listed on Sect. 3.2.2). *Information retrieval* is the task of finding documents that are relevant to a given information query and can be effectively achieved with a bags of words modelling. *Information extraction* consists of the automatic knowledge acquisition from documents; using a primitive notion of language's syntax and semantics, successful information extraction systems have been implemented using finite-state machine, hidden Markov model and conditional random fields. *Natural communication*: require more complex *grammatical models* and reasoning algorithms that takes into account the syntax, semantics and pragmatics of the language. Machine translation and speech recognition represent the most outstanding achievements of NLP in natural communication.

Robotics represents one of the most active and successful fields of AI research. Robots are complex physical agents that perform tasks on the physical world. Robotic system can exhibit distinct levels of autonomy depending on its learning and deliberating capabilities. In particular, AI methods are widely used the highest planning levels, that is, *action planning* and *path planning*. Action planning refers to the identification of a sequence of actions aimed to satisfy a given goal; task that can be addressed with any of the previously described methods for classical and stochastic planning agents. Path planning aims to identify a sequence of collision-free configurations that allow reaching a destination pose in the environment; this task can be solved by geometric algorithms, Markov decision process, sampling-based search, artificial potential fields, rapidly-exploring random trees, among others. Other

low-level aspects affecting the behaviour, like for example trajectory planning, motion planning, trajectory following and motion control can be tackled either by classical methods and techniques found on the automation and control systems literature or through the application of machine learning techniques.

Knowledge representation studies what information or facts about the world should be included on the KB and how such information should be represented. The knowledge abstraction is built in terms of a *conceptualization* of the individuals and their relations in the environment, that is, a map that assigns to each one of them a symbol or a set of symbols in a computer program (the set of symbols is commonly known as *vocabulary*). The *ontology* provides the specification of a conceptualization (Poole and Mackworth 2017). In other words, an ontology specifies the meanings of symbols in terms of the environment under study. The specification provided by the ontology includes what entities can be modelled (*categories*), their properties, relationships (*hierarchy*) and clarifications (restrictions) on the meanings of some of the symbols in the form of axioms. Considering the central role of categories in any large-scale KB, algorithms for reasoning with categories has been also developed: semantic networks and description logics.

As already mentioned, together with the perceptual stimuli, an agent also relies on its built-in or prior knowledge of the environment. **Learning** refers to the ability of an agent to update, upgrade or deprecate any prior knowledge based on its own percepts sequence. Therefore, the behaviour of a learning agent can become effectively independent of its prior knowledge after sufficient experience. As a consequence, any learning agent is inherently *autonomous*: modifying its own beliefs with respect to experience, implies a behavioural evolution on time. It is worth noticing that learning implies *adaptation*, but not the other way around. Regardless of the internal structure, any agent can take advantage of learning to increase its own levels of autonomy. In general, there are two learning strategies that an agent can try: *tuning* its own beliefs based on a direct feedback of the executed actions and expanding its knowledge by *exploration*, that is, by executing actions leading to new experiences.

The branch of computer science focused on the study and implementation of algorithms that improve through experience is known as **machine learning**. The following section introduces ML in detail.

3.2.2 What's Machine Learning?

We have already mentioned that ML deals with algorithms that improve with experience. However, some clarifications are needed. On the one hand, *experience* refers to collecting evidence about the relation that must hold between the inputs and outputs of the algorithm. Evidence is given in the form of data samples, that is a collection of observations-outcomes pair. It is worth noticing that often the observations-outcomes pair corresponds to the inputs-outputs pair of the algorithm. However, in general, such a correspondence may depend on the problem under study and the algorithm itself (i.e. the learning strategy). Most of the ML algorithms rely on a factored representation, where both inputs and outputs are given as N-dimensional vectors of either discrete or continuous numerical values. On the other hand, *to improve* means lessening the uncertainty regarding the nature of the inputs-outputs relationship. In view of this, ML algorithms reach their objective by generalizing (or extrapolating) from specific evidence to general rules. That is, they follow an *inductive reasoning* (bottom-up paradigm). And as such, the predictions of any ML algorithm strongly depend on the evidence supplied to it: no ML algorithm is able to generalize beyond the domain of support induced by the known evidence.

Assuming that the input–output relationship can assume a functional representation, then reducing the uncertainty implies finding a better approximation, or *hypothesis*, to it. In general, different hypotheses may be consistent with the evidence, and one fundamental problem is how to select the best hypothesis among them. Based on the Ockham's razor (Mitchell 1997), the simplest consistent hypothesis should be preferred. However, in general, there should be a trade-off between the consistency and complexity of a hypothesis. Indeed, increasing the complexity reduces the *aleatoric uncertainty* (improves robustness), but at the same it

increases the *epistemic uncertainty*, since generalizing becomes more difficult and requires more evidence to deal with sparsity (Hüllermeier and Waegeman 2019). Therefore, it is common to set up the quest for the best hypothesis in two steps. The first, known as *model selection*, defining the *hypothesis space*. The second, in terms of optimization to determine the best hypothesis in such a space. A learning model assuming that a finite number of parameters suffices to capture everything about the data is called *parametric*. Although, such an assumption notably restricts flexibility, the complexity of parametric models is bounded, no matter if the amount of available evidence is unbounded. In contrast, *non-parametric* assume that it is not possible to capture the data distribution in terms of a finite set of parameters. This makes such models way more flexible than the parametric ones, but their complexity increases with the amount of data provided.

Based on the information provided by the observations-outcomes samples defining the available evidence, distinct forms of learning can be identified (Bishop 2006):

1. *Supervised learning*: in this case the evidence is composed by samples of inputs-outputs pairs. Then, the learning objective is to generate the best hypothesis approximating the function that maps inputs into outputs. The best hypothesis is obtained through optimization and corresponds to the one minimizing a *loss function*, measuring the amount of utility lost between the prediction and the true output value. When the output of the algorithm corresponds to a finite number of discrete categories, or *labels*, the learning problem is called *classification*, otherwise *regression*. Many algorithms have been developed to solve this kind of problems, to name a few: decision tree learning, naive Bayes classifier, k-nearest neighbour (k-NN), metric learning, support vector machines (SVM), random forests, artificial neural network (ANN), ensembles of classifiers and Gaussian process regression.
2. *Unsupervised learning*: the evidence consists of samples containing only the inputs of the algorithm. The learning objective could be: to discover groups of samples having similar attributes (*clustering*), to project the samples into a low-dimensional space while preserving

- some of their meaningful properties (*features extraction, dimensionality reduction*) or to determine the data distribution within the space (*density estimation*). Algorithm for solving unsupervised learning problems is special ANN architectures (auto-encoders, self-organizing map), k-means, DBSCAN, hierarchical clustering, principal component analysis (PCA), mixture models and Gaussian processes.
3. *Reinforcement learning* (RL): evidence is composed by a collection of samples of the form observations-reinforcements, where each observation is a state-action pair and the reinforcements can be either a *reward* or a *punishment*. The learning objective is to determine the optimal *policy* maximizing the overall total reward. In RL, a *policy* is the mapping from every possible state to the best action in that state. In practical applications, there's no prior evidence; it is obtained during the learning process by trial and error. Actions are executed based on a trade-off between *exploitation* of known state-actions pair generating high rewards and *exploration* to discover new ones. Most of the RL algorithms that can be found on literature are variants either of the policy gradient or the Q-learning methods.

It is worth mentioning that nowadays there are *semi-supervised* forms of learning dealing with evidence having a large number of data samples with uncertain or missing information about the outcomes.

In general terms, deep learning (DL) refers to the principle that learning with multiple levels of composition (hierarchy) allows to improve the learning outcomes when sufficient evidence is provided. Such a principle can be potentially applied to any ML algorithm (Deng and Yu 2014). However, in practice, due the contemporary real-world impact of deep neural network (DNN) on the fields of computer vision and natural language processing, DL is widely understood as a synonym of DNN. From this standpoint, DL (DNN) is a special type of ANN having a very large number of hidden layers. With respect to the 1980s, today we have the enough computational power (GPGPU) and the sufficiently large datasets that such complex ANN models require to succeed: the only way to deal with the intrinsic epistemological uncertainty of a complex model is to feed it with sufficient amounts of (non-redundant)

data. Moreover, although there were no significant theoretical contributions to the field of ANN since then, the use of convolutional neural networks (CNN) allows to dramatically reduce the number weights and thus to speed-up the learning algorithm. As a last remark, it is worth mentioning the technique known as *transfer learning*. In brief, the technique consists of exploiting the available knowledge for solving one task and applying it for solving a different one (Goodfellow et al. 2016). This technique is widely used on DL applications, in particular through *fine tuning*.

3.2.3 What's the Relation Between Artificial Intelligence and Machine Learning?

As a first approximation, one can say that ML *seems to be* a branch of AI. However, in analogy with the perception case, agents require learning to improve their knowledge of the environment and, consequently, to further their own goals. Therefore, learning in AI is not an end to itself, but a necessary constituent to build intelligent machines. *It follows that, although AI and ML are highly related, they pursue two different avenues.* The distinction between the two research fields can be also traced through a historical perspective.

In the early days of AI, some researchers were experimenting the ways machines can learn from data. Different approaches were developed to achieve such a goal. In particular, nowadays ANN is the most widely known. However, due to the strong emphasis that settle the AI community on the KB logical approach, by 1980 the data driven and the statistical ones were already ignored by the AI community. The latter approaches continued their way on the fields of pattern recognition and information retrieval, while the ANN enthusiast continued the research as part of the connectionism line of thought. After the reinvention by them of the back-propagation algorithm, ML started to gain attention as a separate field in the 1990s. The focus of ML was no longer to achieve AI but to solve practical problems based on statistical and probabilistic methods and models.

3.3 Human–Robot Cooperation for Smart Manufacturing

Industry 4.0 foresees humans and CPS as cooperative entities. Under such a perspective, CPS need to aware not of its inner state, but also of the environmental ones, including any other entity on its surroundings. Moreover, CPS are required to smartly interact with both humans and other machines. Such a rich interaction between humans and CPS requires safe physical human-machine interaction (pHRI), unambiguous and resilient information flows, autonomous information processing and real-time decision-making capabilities. The first requirement is automatically satisfied in the context of collaborative robotics. The second deals exclusively with the Internet of Things (IoT) infrastructure. The last two are, in general, open research problems. The goal of this section is to highlight the potential of AI and ML approaches to tackle such problems in the context of human-robot cooperation in assembly.

3.3.1 CPS and Safety

Cyber-physical systems (CPS) represent one of the fundamental key enabling technologies for Industry 4.0. Although CPS are still in the making, it has been conjectured that their introduction in industry will dramatically change the way value is created along all the digitization axes of the manufacturing sector: smart product, smart manufacturing and business model. Based on the 5C architecture (Lee et al. 2015), implementing a CPS comprises the following levels:

- *Smart connection level*: is concerned with the sensing and transduction technologies and the IoT infrastructure for real-time, seamless and resilient data exchange between all parties.
- *Data-to-information conversion level*: incorporate all information retrieval methodologies aimed to understand the state of the machine and its components. In other words, this level deals with the implementation of the single machine self-awareness.

- *Cyber level*: represents a central information hub between all machines. Through the data aggregation and subsequent analysis it could be possible to compare the performance between different machines and to predict the future behaviour of each.
- *Cognition*: includes a set of decision support systems that implement preliminary data analysis and valuable means for data visualization, aimed to transfer efficiently the inferred knowledge to the human experts.
- *Configuration level*: refers to the actuation mechanism aimed to apply any corrective or preventive decision taken at the cognition level to the physical space.

The 5C architecture is thus defined as a human-in-the-loop (HiTL) scheme where human experts, aided by decision support systems, take all decisions regarding how to improve the manufacturing process. It is worth noticing that the applicative context of this architecture is limited to classical manufacturing processes. Indeed, it doesn't account for possible interactions with the environment (safety) and it lacks of a proper design for distributed processing capabilities. Therefore, the 5C architecture is not well suited for modern robotic assembly workstations, specially for those having shared collaborative environments. Another key concept in Industry 4.0 not captured by the 5C architecture is that CPS should be able to cooperate with humans and other CPS. Cooperation implies two fundamental objectives. The first, to ensure safety; a constraint that cannot be violated by any means. The second, to conclude the assigned task; whose achievement can be only guaranteed in safe operative conditions. With regard to safety, CPS must be able to build their own knowledge not only in terms of self-awareness but also in terms of *situational-awareness*, including both, the state of the physical environment and the state of the current assembly cycle. With regard to the task completion, CPS must manifest some degrees of decision-making capabilities. In other words, they must be able to learn how to interact with the environment, including other entities, based on their beliefs about the environmental state.

With this idea in mind, let's rephrase the above considerations in AI jargon. We start by observing CPS are able to perceive and act, thus,

from the very basic definition, it follows that CPS are indeed rational agents. In particular, CPS belong to the class of model-based agents: they must keep an internal representation of their physical counterpart and of their environment, including the state of the manufacturing process (self- and situational-awareness). Moreover, they should achieve multiple goals at the same time based on the current beliefs: an utility measure is required to define the proper trade-off. Consequently, CPS should plan their actions so to maximize the expected utility when averaged among all possible outcomes that can result from their actions. Furthermore, CPS must cooperate between them considering that the overall goal is to improve production; still, competing CPS willing to reach the highest performance can be desirable in a manufacturing context (paradigm defined as “self-compete” in the 5C architecture). Finally, CPS must deal with both aleatoric and epistemic uncertainty, specially on workspaces share with human beings. Nevertheless, there are some key different that makes a CPS something more *tangible* than an abstract agent. On the one hand, CPS are always associated to a physical counterpart and a concrete implementation. On the other hand, a CPS may exhibit degrees of complexity that are difficult to express or implement in terms of a single rational agent.

Based on the above considerations, we identify a structured representation for a machine or robot to be considered as a safety-aware CPS (SA-CPS), defined in terms of four interacting components (see Fig. 3.1):

1. *Safety monitor*: based on the percepts sequence and current beliefs, the aim of this block is to monitor the operative conditions of the CPS and to trigger an alarm when safety is unexpectedly lost or when it can be potentially lost in a finite time horizon. Therefore, this unit relies on an internal model to predict potential risky circumstances and to decide when to notify the other components of the CPS. This block is always active and runs in parallel with any of the other three units.
2. *Safety reflexes*: the aim of this block is to promptly react when an alarm is triggered by the safety monitor. The set of actions executed by this block seek to quickly restore the save operative conditions despite the current operative state. In terms of the AI agents taxonomy, this unit

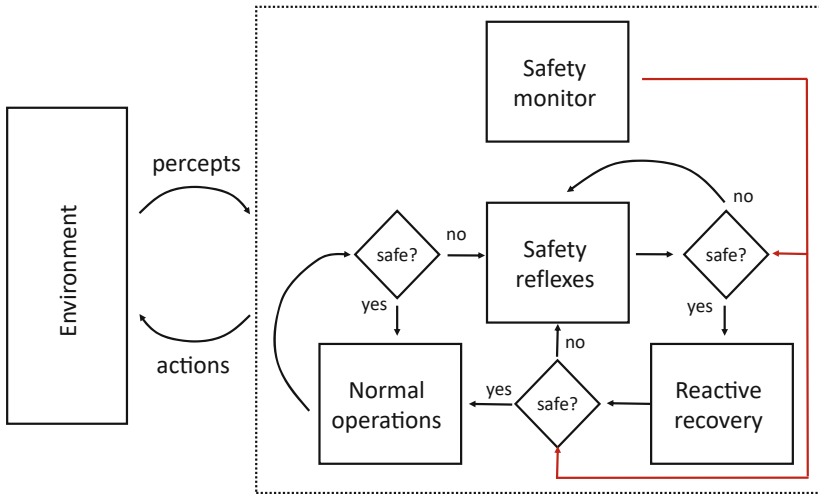


Fig. 3.1 Structured representation of the abstract safety CPS

together with the safety monitor one can be considered as a model-based reflex agent.

3. *Reactive recovery*: this block aims to restore a pre-empted operative state when the safe operative conditions are recovered again. Therefore, the goal of this component is to plan a sequence of actions allowing to ensure that the normal operations can be restarted just after a risky circumstance has been mitigated. When this component is active, normal operations are on hold.
4. *Normal operations*: this block incorporates all the functionalities required to reach the CPS goals. It can pre-empted at any time by the safety reflexes and can only restart operations after suitable recovery actions had taken place. This component can be seen as an utility-based agent, focused on the completion of the manufacturing task assigned to the CPS.

It is worth noticing that Fig. 3.1 only captures the logical relation between the four components. However, the interactions between them are in general richer and complex. As a last remark, modern collaborative robots have a similar internal structure. In particular, safety is defined

in terms of physical interaction; prompt reactions imply stopping the current motion and blocking the motor actuators; and recovery actions consist of unlocking again the motors and restarting the pre-empted motion.

3.3.2 Human–Robot Cooperation in Assembly

The most dangerous risk specific to robots is the unexpected collisions between the robot and the environment (Siciliano et al. 2010). When an unexpected exertion occurs between a collaborative robot and its surrounding environment, impact forces are eased thanks to their lightweight design and compliant mechanisms and control. However, avoiding unexpected force exertions implies foreseeing dangerous situations, and thus it relies on sensing, situational awareness, planning and decision-making capabilities. *Therefore, without suitable exteroceptive sensing a collaborative robot cannot be considered as a safe companion in the context of human–robot cooperation.* Indeed, to safely interact with a human operator a collaborative robot must predict and prevent any risky circumstances based on its own situational awareness. To this end, it is required to associate to the human operator and the environment a set of meaningful spatio-temporal features that allows—with some degree of accuracy, within a finite time horizon—to model and predict the operator's behaviour and the environmental changes. In terms of safety, it is required to sense and predict the operator's motion. In terms of cooperation, it is required to understand and predict the operator's actions and intentions.

We identify three major synergic elements (see Fig. 3.2) required for a collaborative robot to be considered as a safe companion in the context of human-robot cooperation: (i) scene monitoring, (ii) tasks modelling and (iii) planning. Although these general problems can be unreasonable complex, within the context of cooperative assembly workstations where different constraints are imposed to the environment and due to the cyclic nature of the assembly process, the analysis of each element can be greatly simplified. In particular, we introduce the following simplifying assumptions:

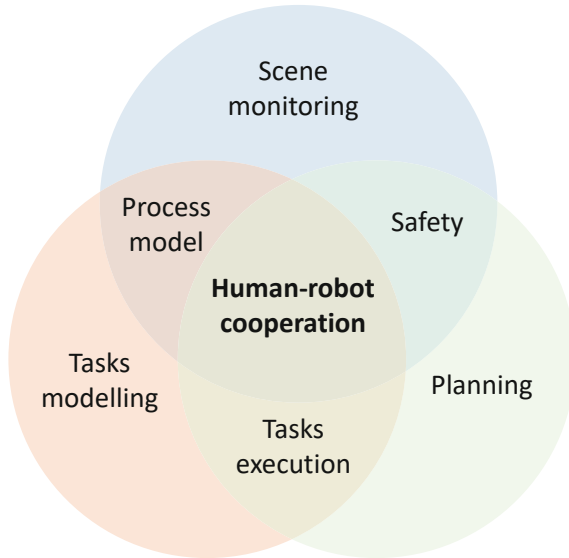


Fig. 3.2 Key enabling technologies for human-robot cooperation

- i. The environment is limited to the collaborative workstation and its assembly process.
- ii. There's only one human operator and one collaborative robot active a time in the workstation.
- iii. The state transitions on the environment are triggered only by events.
- iv. There's a finite number of sequences of state transitions that allow reaching the final state.
- v. There's a finite number of environmental states.

The first and second allow us to focus on the human-robot cooperation by ignoring the interactions with the rest of the assembly line. Therefore, we assume that the inputs of the assembly process are always available and that the outputs of the assembly process are being gathered autonomously by an external entity without affecting the assembly process. The third, to limit how deep the robot's understanding of the environment should be. For example, when a human operator is finishing one part, it is not always possible to know what specific

finishing touch is being performed, what are the missing ones or what were already performed. At a higher level, however, the part is being finished. Thus, we implicitly assume that in terms of cooperation it is not required to reconstruct the whole product state, but only up to the process state. The fourth accounts for the inner variability inside the assembly process. The fifth implies that the assembly process can be split in a finite sequence of tasks. Based on these assumptions, the environment results:

- Fully observable, all state transitions are distinguishable with suitable sensing and perceptive capabilities.
- Stochastic. On the one hand, there's not an unique combination of state transitions allowing to complete the assembly task. On the other, the time between successive state transitions can (greatly) vary between different assembly cycles.
- Static, mainly due to assumptions (i) and (ii). However, in the context of flexible manufacturing some clarifications are required. In case of multi-variant or multi-product lines both, the assembly cycle and the workstation layout may require some adjustments. However, such adjustments do not occur whitening the assembly cycle. Indeed, the current product under manufacture must be completed, aborted or pre-empted before switching the assembly goal. In other words, any environmental change required for a flexible assembly line will be triggered by an event (in analogy to assumption [iii]). Moreover, all possible environmental changes are necessarily countable and finite. Therefore, without loss of generality, we can assume that in the context of flexible manufacturing there exists a finite set of static environments and that each of them can be handled independently from the other.
- Discrete, by assumption (v).
- Known. All possible state transitions are well understood, in terms of the expected outcomes of the assembly process. This is also enforced by assumption (iv).
- Sequential, as the assembly process.
- Defining the environment's agency is rather ambiguous, considering that under our modelling assumptions a single CPS may be defined in terms of several interacting rational agents. However, due to the

restriction in assumption (i) and (ii), we assume that there's only one CPS, given by the collaborative robot and its associated sensing and processing capabilities.

Based on the properties of the environment, we can introduce the key enabling technologies for a safe human-robot cooperation in collaborative assembly.

Scene monitoring refers to the real-time reconstruction of the state of both the operator and the manufacturing parts and products along the whole assembly processes. Here the objective is not to reconstruct the state of the assembly process, but to increase the CPS's awareness about where the objects and operator are in physical terms (pose, motion, etc.). In other words, the goal of the scene monitoring unit is twofold. On the one hand, to extract from the percepts sequence the information required to evaluate and guarantee the safe operative conditions at every time instant. On the other hand, to extract from the percepts sequence the required information to allow further inference regarding the current and future operator's activities, and the current and future state of the ongoing assembly cycle. Therefore, the scene monitoring problem can be analysed in terms of both, the recognition and tracking of assembly parts and products, and the operator's motion tracking.

- *Object's recognition and tracking*: there are different technologies that can be used to efficiently recognize and track the pose, motion and manufacturing state of objects. To name a few, one can identify 2D/3D vision systems, RF systems, range finder, sonar, mmWave, etc. A throughout treatment of the problem of object recognition in smart manufacturing is found in (Riordan et al. 2019).
- *Operator's motion tracking*: due to the stochastic nature of the operator's body, head, arms, etc., movements while executing an assembly task, the problem of monitoring and predicting the operator's motion can be considered as a particular instance of a filtering problem (Tan and Arai 2011). That is, based on a set of past possibly noisy observations of the operator's pose determine the best estimate of the current operator's motion. Today on research and industry exist a wide range

of different sensing technologies that can be used to measure the operator's pose. The current technological trend points towards multiple networking range sensing devices or computer vision-based systems providing analogous measurements (Ferrari et al. 2018; Gkournelos et al. 2018; Agethen et al. 2016). The use of multiple sensors not only ensures a better accuracy of the estimation but also accounts for the decreasing point-density at far distances of a single sensors. Moreover, different view points are required for a reliable identification of features or markers. However, state-of-the-art deep learning models for pose estimation in RGB images (Cao et al. 2017) and 2D lidar data (Weinrich et al. 2014) allows to reach high-levels of accuracy. Indeed, the larger field of view of lidar sensors allows to track the operator beyond the field of view of the RGB-D sensor.

Tasks modelling aims to understand what are the current and future operator's activities, and the current and future state of the ongoing assembly cycle. However, considering that only the operator and robot actions can cause a process state transition, the tasks modelling problem can be restricted to the recognition and prediction of the actions executed by the operator.

- *Operator's intentions prediction*: in industrial manufacturing scenarios, the problem of task prediction is greatly simplified by the cyclic nature of the operator's work. Any manufacturing cycle is indeed defined by a finite set of atomic tasks. However, the order in which such atomic tasks are performed by the operator to conclude the cycle, in general, is not uniquely defined. Therefore, for a machine to be aware of the current state of an assembly cycle, it is required to recognize any atomic task executed by the operator and to model the transitions between them (Alati et al. 2019b). Identifying a task implies understanding the actions being performed by the operator, while understanding the transitions between tasks implies predicting the operator's intentions. Based on this idea, the prediction of intentions problem can be analysed in terms of two distinct processes: (i) action recognition, and (ii) action prediction. Action recognition refers to the prompt identification of the current task executed by

the operator. The aim of this process is to continuously monitoring the operator's actions on real-time. The action identification can be driven by different cues, like gestures (Carrasco and Clady 2010), scene objects being manipulated (Koppula and Saxena 2015) or environmental information (Casalino et al. 2018). Action recognition has been also extensively studied in terms of whole body motion tracking and segmentation (Natola et al. 2015; Tome et al. 2017). Although there are no specific manufacturing datasets for the evaluation of action recognition models, in recent years different deep network architectures had been demonstrated high levels of accuracy on totally unrelated but similar manipulation tasks, like the one proposed by the Epic Kitchens challenge (Damen, et al. 2018; Wang et al. 2018). Action prediction refers to the total or partial reconstruction of the possible sequence of actions that the operator would execute just after concluding the current task. Consequently, this process implies the generation and constant refinement of an action transition model (Zanchettin and Rocco 2017; Zanchettin et al. 2018). In general, it can be also assumed that all operator's states and actions are fully observable and that the operator can only execute one action a time.

In general, **planning** actions in collaborative workstations requires finding a suitable and safe plan to complete a manipulation or a mobility task assigned to the CPS. However, we will restrict our attention to the manipulation case, since in most collaborative workstation the CPS is defined on top of a robot manipulator with a fixed inertial base. The objective here is to superimpose the robot's state on top of the assembly process model, such that to allow the real-time analysis and generation of the robot plan. In other words, the robot's collaborative behaviour is achieved by dynamically allocating its tasks, in terms of the predicted operator's actions and the relative action transition model (Alati et al. 2019a). As a result, the objective of the action planning is to reach a designated assembly process goal state. This implies that any goal state includes both the operator's and robot's states. Therefore, it is expected that one particular goal can be reached from a finite set of initial candidate states, each one depending on the particular sequence of actions performed by the operator. Consequently, in

human–robot collaborative environments, the action planning process deals with the robot behaviour adaptation (Mitsunaga et al. 2008) to the time-varying set of constraints imposed the operators' actions. In turns, imposed by the customer requirements, diversity of the available manufacturing variants and operator's task execution preferences (Munzer et al. 2017). Therefore, the robot adaptation should provide a proactive (anticipatory) collaborative behaviour driven by the different forms of human-robot interaction associated to each target goal (Mason and Lopes 2011). In other words, robots working alongside humans should model how to anticipate a belief about possible future human actions (Koppula et al. 2016). In complete analogy to the operator's case, the cyclic nature of the assembly process implies that there's a predefined number of goals that the robot can reach, a finite set of deterministic actions that it can perform and a finite set of states that it can have. Moreover, it can be also assumed that the robot can only execute one action a time. However, in general, the execution time of any planned cannot be defined in advance since it also depends on the current state of the assembly process. Specially in the cases when the action execution requires explicit synchronization with the operator.

We observe that each key enabling technology comprises different perceptive or cognitive processes that, based on the structured representation of the SA-CPS, can be mapped to one or more of its building blocks. In particular, the scene monitoring greatly overlaps with the safety monitor. However, the scope of the former is not only to evaluate risks but also to understand the current process state and its evolution in the near future, which belongs to the normal operations block. Planning is required for safety reflexes, reactive recovery and normal operations. Finally, task modelling belongs mainly to the normal operations blocks. However, understanding the assembly sequence provide useful hints on the prediction of risky circumstances.

3.4 Conclusions

Industry 4.0 foresees humans and CPS as cooperative entities. Under such a perspective, CPS need to aware not of its inner state, but also of the environmental ones, including any other entity on its surroundings. To smartly interact with both humans and other machines, CPS must be endowed with real-time decision-making capabilities. Although still today there are many open problems on the field, different AI and ML techniques can be combined together to provide feasible solutions to real-world problems, specially on the fields of HRC and automated adaptation of a multi-variant collaborative manufacturing system.

Within these applicative context, it is required to provide a strong emphasis on safety, concept that to our knowledge has not being taken into account on any formalization of the concept of CPS. A safety-aware CPS is composed at least by four fundamental blocks:

- A constantly running safety monitor system to evaluate the safety status independently of any other functionalities of the CPS.
- A safety reflexes block to be activated when a risky circumstance has been detected.
- A reactive recovery unit to restore safe operative conditions just after the safety has been guarantee by the prompt actions of the safety reflexes unit.
- A normal operations module, which normally runs unless pre-empted due to safety issues.

HRC can be effectively implemented through the exploitation of three key enabling technologies, namely: scene monitoring, task modelling and planning. Different state-of-the-art AI and ML algorithms can deal with deferent aspect of one or more of these technologies. The research in this area is still in an early stage, so this contribution aims to motivate other researchers to do further research and practitioners to collaborate with research institutions for conducting tests on practical applications in real case studies.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734713.



References

- Agethen, P., M. Otto, S. Mengel, and E. Rukzio. 2016. Using marker-less motion capture systems for walk path analysis in paced assembly flow lines. *Procedia CIRP* 54: 152–157. <https://doi.org/10.1016/j.procir.2016.04.125>.
- Alati, E., L. Mauro, V. Ntouskos, and F. Pirri. 2019a. Anticipating next goal for robot plan prediction. *SAI Intelligent Systems Conference*, 792–809. https://doi.org/10.1007/978-3-030-29516-5_60.
- Alati, E., L. Mauro, V. Ntouskos, and F. Pirri. 2019b. Help by predicting what to do. *IEEE International Conference on Image Processing*, 1930–1934. <https://doi.org/10.1109/ICIP.2019.8803155>.
- Bishop, C.M. 2006. *Pattern recognition and machine learning*. Berlin: Springer-Verlag.
- Cao, Z., T. Simon, S.E. Wei, and Y. Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Conference on Computer Vision and Pattern Recognition*, 7291–7299. <https://doi.org/10.1109/CVPR.2017.143>.
- Carrasco, M., and X. Clady. 2010. Prediction of user's grasping intentions based on eye-hand coordination. *IEEE/RSJ International Conference on Intelligent Robots and System*, 4631–4637. <https://doi.org/10.1109/IROS.2010.5650024>.
- Casalino, A., C. Messeri, M. Pozzi, A.M. Zanchettin, P. Rocco, and D. Praticchizzo. 2018. Operator awareness in human-robot collaboration through wearable vibrotactile feedback. *IEEE Robotics and Automation Letters* 3 (4): 4289–4296. <https://doi.org/10.1109/LRA.2018.2865034>.

- Damen, D. et al. 2018. Scaling egocentric vision: The Epic-Kitchens dataset. *European Conference on Computer Vision*. arXiv:1804.02748 [cs.CV]. Available at <https://arxiv.org/abs/1804.02748>.
- Deng, L., and D. Yu. 2014. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing* 7 (3–4): 1–199. <https://doi.org/10.1561/20000000039>.
- Ferrari, E., M. Gamberi, F. Pilati, and A. Regattieri. 2018. Motion analysis system for the digitalization and assessment of manual manufacturing and assembly processes. *IFAC-PapersOnLine* 51 (11): 411–416. <https://doi.org/10.1016/j.ifacol.2018.08.329>.
- Gkournelos, C., P. Karagiannis, N. Kousi, G. Michalos, S. Koukas, and S. Makris. 2018. Application of wearable devices for supporting operators in human-robot cooperative assembly tasks. *Procedia CIRP* 76: 177–182. <https://doi.org/10.1016/j.procir.2018.01.019>.
- Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep learning*. MIT Press.
- Hüllermeier, E., and W. Waegeman. 2019. Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction to concepts and methods. ArXiv:1910.09457 [cs.LG]. Available at <https://arxiv.org/abs/1910.09457>.
- Koppula, H.S., and A. Saxena. 2015. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (1): 14–29. <https://doi.org/10.1109/TPAMI.2015.2430335>.
- Koppula, H.S., A. Jain, and A. Saxena. 2016. Anticipatory planning for human-robot teams. In *Experimental robotics. Springer tracts in advanced robotics*, ed. M. Hsieh, O. Khatib, and V. Kumar, 109. https://doi.org/10.1007/978-3-319-23778-7_30.
- Lee, J., B. Bagheri, and H. Kao. 2015. A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters* 3: 18–23. <https://doi.org/10.1016/j.mfglet.2014.12.001>.
- Mason, M., and M. Lopes. 2011. Robot self-initiative and personalization by learning through repeated interactions. *ACM/IEEE International Conference on Human-Robot Interaction*, 433–440. <https://doi.org/10.1145/1957656.1957814>.
- Mitchell, T.M. 1997. *Machine Learning*. New York: McGraw-Hill.
- Mitsunaga, N., C. Smith, T. Kanda, H. Ishiguro, and N. Hagita. 2008. Adapting robot behavior for human-robot interaction. *IEEE Transactions on Robotics* 24 (4): 911–916. <https://doi.org/10.1109/TRO.2008.926867>.
- Munzer, T., M. Toussaint, and M. Lopes. 2017. Preference learning on the execution of collaborative human-robot tasks. *IEEE International Conference*

- on *Robotics and Automation*, 879–885. <https://doi.org/10.1109/ICRA.2017.7989108>.
- Natola, F., V. Ntouskos, M. Sanzari, and F. Pirri. 2015. Bayesian non-parametric inference for manifold based MoCap representation. *IEEE International Conference on Computer Vision*, 4606–4614. <https://doi.org/10.1109/ICCV.2015.523>.
- Poole, D., and A. Mackworth. 2017. *Artificial intelligence: Foundations of computational agents*. New York: Cambridge University Press.
- Riordan, A., D. Toal, T. Newe, and G. Dooly. 2019. Object recognition within smart manufacturing. *Procedia Manufacturing* 38: 408–414. <https://doi.org/10.1016/j.promfg.2020.01.052>.
- Ruiz Garcia, M.A., R.A. Rojas, L. Gualtieri, E. Rauch, and D. Matt. 2019. A human-in-the-loop cyber-physical system for collaborative assembly in smart manufacturing. *Procedia CIRP* 81: 600–605. <https://doi.org/10.1016/j.procir.2019.03.162>.
- Russell, S., and P. Norvig. 2010. *Artificial intelligence: A modern approach*. Prentice Hall.
- Siciliano, B., and O. Khatib. 2016. *Springer handbook of robotics*. Springer International Publishing.
- Siciliano, B., L. Sciavicco, L. Villani, and G. Oriolo. 2010. *Robotics: Modelling, planning and control*. London: Springer.
- Tan, J.T.C., and T. Arai. 2011. Triple stereo vision system for safety monitoring of human-robot collaboration in cellular manufacturing. *International Symposium on Assembly and Manufacturing*, 1–6. <https://doi.org/10.1109/ISAM.2011.5942335>.
- Tome, D., C. Russell, and L. Agapito. 2017. Lifting from the deep: Convolutional 3d pose estimation from a single image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2500–2509. <https://doi.org/10.1109/CVPR.2017.603>.
- Tseng, M.M., J. Jiao, and M.E. Merchant. 1996. Design for mass customization. *CIRP Annals* 45 (1): 153–156. [https://doi.org/10.1016/S0007-8506\(07\)63036-4](https://doi.org/10.1016/S0007-8506(07)63036-4).
- Wang, L., et al. 2018. Temporal segment networks for action recognition in Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (11): 2740–2755. <https://doi.org/10.1109/TPAMI.2018.2868668>.
- Weinrich, C., T. Wengefeld, C. Schroeter, and H. Gross. 2014. People detection and distinction of their walking aids in 2D laser range data based on generic distance-invariant features. *IEEE International Conference on Robot*

and *Human Interactive Communication*, 767–773. <https://doi.org/10.1109/ROMAN.2014.6926346>.

Zanchettin, A.M., A. Casalino, L. Piroddi, and P. Rocco. 2018. Prediction of human activity patterns for human–robot collaborative assembly tasks. *IEEE Transactions on Industrial Informatics* 15 (7): 3934–3942. <https://doi.org/10.1109/TII.2018.2882741>.

Zanchettin, A.M., and P. Rocco. 2017. Probabilistic inference of human arm reaching target for effective human-robot collaboration. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 6595–6600. <https://doi.org/10.1109/IROS.2017.8206572>.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

