Multimedia Contents

# 54. Industrial Robotics

Martin Hägele, Klas Nilsson, J. Norberto Pires, Rainer Bischoff

Much of the technology that makes robots reliable, human friendly, and adaptable for numerous applications has emerged from manufacturers of industrial robots. With an estimated installation base in 2014 of about 1.5 million units, some 171 000 new installations in that year and an annual turnover of the robotics industry estimated to be US$ 32 billion, industrial robots are by far the largest commercial application of robotics technology today.

The foundations for robot motion planning and control were initially developed with industrial applications in mind. These applications deserve special attention in order to understand the origin of robotics science and to appreciate the many unsolved problems that still prevent the wider use of robots in today's agile manufacturing environments. In this chapter, we present a brief history and descriptions of typical industrial robotics applications and at the same time we address current critical state-of-the-art technological developments. We show how robots with different mechanisms fit different applications and how applications are further enabled by latest technologies, often adopted from technological fields outside manufacturing automation.

We will first present a brief historical introduction to industrial robotics with a selection of contemporary application examples which at the same time refer to a critical key technology. Then, the basic principles that are used in industrial robotics and a review of programming methods will be outlined. We will also introduce the topic of system integration particularly from a data integration point of view. The chapter will be closed with an outlook based on a presentation of some unsolved problems that currently inhibit wider use of industrial robots.

## 54.1 Industrial Robotics: The Main Driver for Robotics Research and Application

Even though robots are considered a cornerstone of today's competitive manufacturing particularly in automobile and related component assembly, there are still challenges to solve for manufacturing to efficiently respond to changing consumer behavior and global shifts in competitiveness. Furthermore, high-growth industries (in electronics, food, logistics, and life-sciences) and emerging manufacturing processes (gluing, coating, laser-based processes, precision assembly, fiber material processing) as well as fulfilling sustainability regulations will increasingly depend on advanced robot technology [54.1]. Additionally, the range of feasible applications could significantly increase if robots were easier to install, to integrate with other manufacturing processes, and to program, particularly with adaptive sensing and automatic error recovery. Further challenges result from the integration of various types of controls (programmable logic controller (PLC), computer numerical control (CNC) sensors) with the robot controller, from close human–robot collaboration and fenceless production with both lightweight and heavy duty robots, and from an increasing need to save energy.

Design and production of industrial robots on the one hand, and the planning, integration, and operation of robot work cells on the other hand are largely independent engineering tasks. In order to be produced in sufficiently large quantities, a robot design should meet the requirements for the widest set of potential applications. As this is difficult to achieve in practice, various classes of robot designs regarding payload capacity, number of robot axes, and workspace volume have emerged for application categories such as assembly, palletizing, painting, welding, machining, and general handling tasks.

Generally, a robot workcell consists of one or more robots with controllers and so-called robot peripherals, e.g., grippers or tools, safety devices, sensors, and material transfer components for moving and presenting parts. Typically, the cost of a complete robot workcell is four to five times the cost of the robots alone; however, efforts are underway to drastically reduce these costs through use of increased robot functionality and artificial intelligence [54.2]. A robot workcell is usually the result of customized planning, integration, programming, and configuration, requiring significant engineering expertise. Standardized engineering methods, tools, and best-practice examples for specifying and designing robot workcells have become available to provide predictable performance and to secure investments [54.3].

Today's industrial robots are mainly rooted in the requirements of capital-intensive large-volume manufacturing, typically defined by the automotive, electronics, and electrical goods industries which make up 80% of all robot installations. Future industrial robots will not be a mere extrapolation of today's designs, but will rather follow new design principles addressing a much wider range of application areas and industries. At the same time, new technologies, particularly from the information technology (IT) or the consumer domain will have an increasing impact on the design, performance, use and cost of future industrial robots.

International and national standards now help to quantify robot performance and define safety precautions, geometry, and media interfaces [54.4]. Most robots operate behind secure barriers to keep people at a safe distance. Recently, improved safety standards have allowed direct human–robot collaboration, permitting robots and human factory workers to share the same workspace [54.5, 6].

## 54.2 A Short History of Industrial Robots

The invention of the industrial robot dates back to 1954 when inventor George Devol filed a patent on a *programmed article transfer* (Fig. 54.1). After teaming up with young engineer and entrepreneur Joseph Engelberger, the first robot company, Unimation, was founded. It put the first robot into service at a General Motors plant in 1961 for extracting parts from a die-casting machine. Most of the hydraulically actuated Unimates were sold through the following years for workpiece handling and for spot-welding of car bodies [54.7]. Soon, many other companies started to develop and manufacture industrial robots in many industrial nations; an innovation-driven industry was born [54.8]. The first International Symposium on Industrial Robotics (now ISR) took place in Chicago in 1970 and proved that robotics had become the field of activity of a vibrant research community.

The breakthrough Stanford Arm was designed as a research prototype in 1969 by Victor Scheinman (Chap. 4). The six-degree-of-freedom (6-DOF) all-

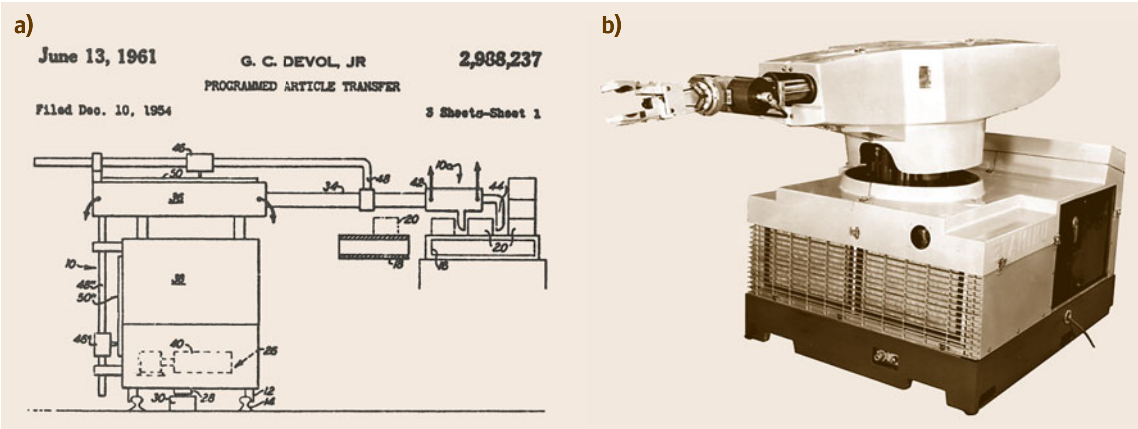



**Fig. 54.1a,b** The invention of the industrial robot. **(a)** This patent was the start of a joint effort between G. Devol and J. Engelberger to form the first robot company, Unimation, a fusion of the terms *universal* and *automation*. The company was acquired by Westinghouse in the late 1980s and subsequently taken up by Stäubli in 1988. **(b)** The first Unimation performed a rather simple handling task in 1961 at a General Motors plant; other car manufacturers followed (courtesy of Smithsonian Institution Archives, Washington DC)

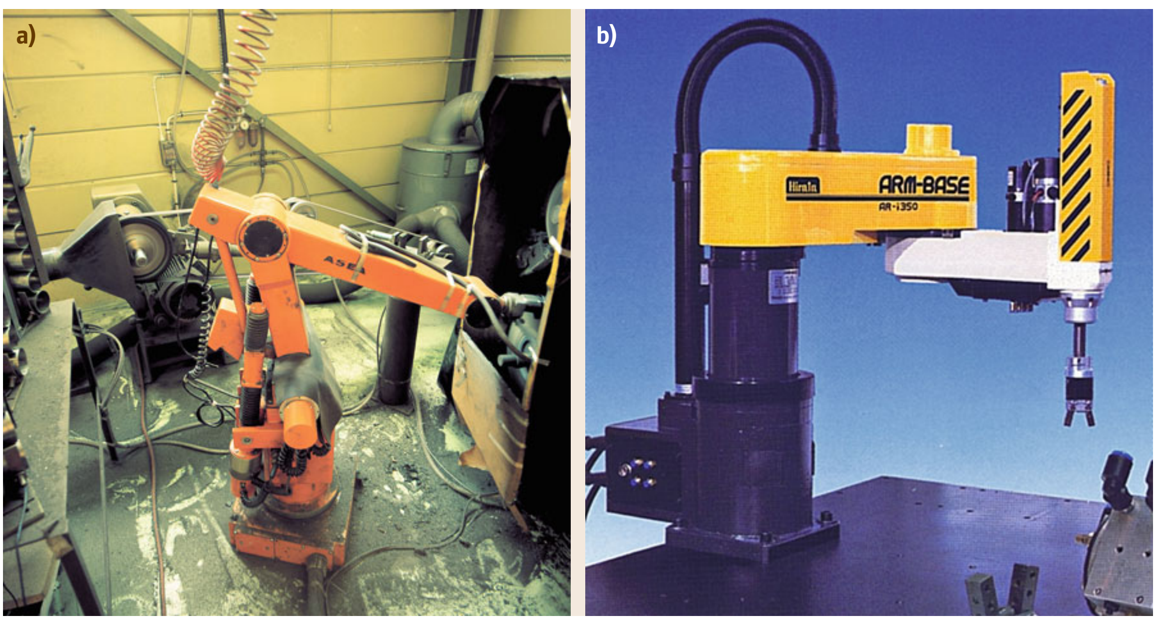



**Fig. 54.2a,b** The all-electric **(a)** IRB-6 and **(b)** a SCARA-type kinematic. **(a)** First introduced in 1973, the IRB-6 has been a breakthrough development as it was the first serially produced robot product, which combined all-electric-drives technology and a microcomputer for motion control and programming. The robot proved very robust, and life-times of more than 25 years in harsh productions were reported (courtesy of ABB Automation). **(b)** The selective compliance assembly robot arm (SCARA) is particularly suited for assembly tasks as it combines rigidity in the vertical axis and compliance in the horizontal axis. In 1978, the first Hirata AR-300 was put together. Depicted is the successor design, the AR-i350 (courtesy of HIRATA Robotics, Mainz)

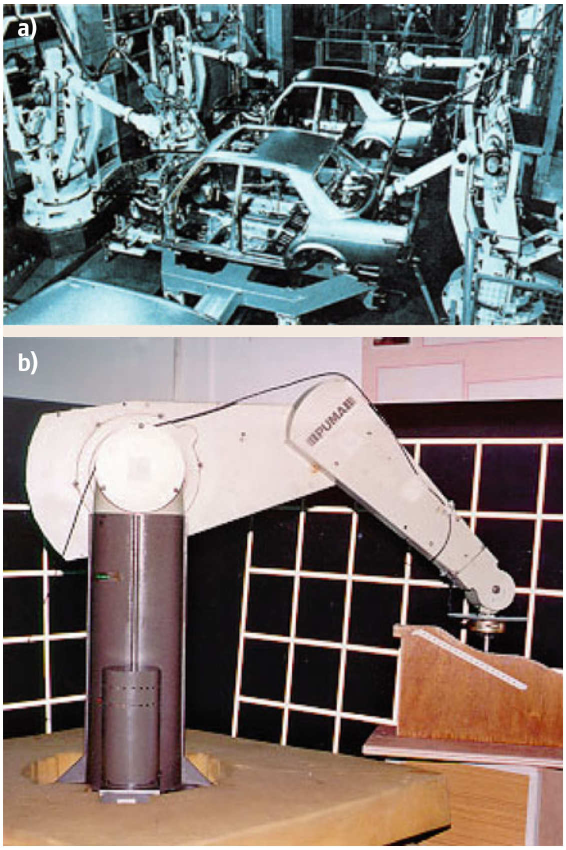



**Fig. 54.3a,b** The KUKA IR 601/60 **(a)** and the Unimation PUMA (programmable universal machine for assembly) 560 **(b)**. **(a)** In 1978, the novel 6 DOF KUKA robot featured a parallel linkage for its second and third axes. At almost two tons of own weight, it could handle payloads of some 60 kg at maximum operating speed. The robot quickly became a workhorse for the automotive industry. **(b)** The six axis PUMA was inspired by the dexterity of a human arm. After its launch in 1979 by Unimation, it became one of the most popular arms and was used, due to its versatility and ease of use, as a reference in robotics research for many years

electric manipulator was controlled by a state-of-the-art computer of the time, a DEC PDP-6. The nonanthropomorphic kinematic configuration with one prismatic and five rotational joints was configured such that the equations for solving the robot kinematics were simple enough to speed up computations. Drives consisted of direct-current (DC) electric motors, harmonic drive and spur gear reducers, potentiometers and tachometers for position and velocity feedback [54.9]. Subsequent robot designs were strongly influenced by Scheinman's concepts (Figs. 54.2 and 54.3).

In 1973, the company ASEA (now ABB) introduced the first microcomputer-controlled all-electric industrial robot, the IRB-6, which allowed continuous path motion (CP), a precondition for many applications such as arc welding or material removal (Fig. 54.2). In the 1970s, intense diffusion of robots into car manufacturing set in mostly for (spot-)welding and handling applications (Fig. 54.3) [54.10].

In 1978, the selective compliance assembly robot arm (SCARA) was invented by *Makino* of Yamanashi University, Japan [54.11]. The ground-breaking four-axis low-cost design was perfectly suited for small parts assembly as the kinematic configuration allows fast and compliant arm motion (Fig. 54.4). Flexible assembly systems based on the SCARA robot in conjunction with compatible product designs (DFA) have contributed significantly to creating a boom in high-volume electronics production and consumer products [54.12]. Further optimization of robot dynamics and accuracy led to the first direct-drive SCARA robot, the AdeptOne in 1984 [54.13].

Requirements regarding a robot's speed, accuracy and weight have led to novel kinematic and transmission designs. An approach toward lightweight and stiff structures has been pursued since the 1980s by developing parallel kinematic machines (PKM) which connect the machine's basis with its end-effector by three to six parallel struts, see also Fig. 54.5. These so-called parallel robots (Chap. 4 and 18) are particularly suited to achieve short cycle times (e.g., for picking), precision (e.g., for material removal), or handling high workloads (Fig. 54.5) and have found their niches in advanced manufacturing [54.14]. However, workspace volumes tend to be significantly smaller than those of serial or open kinematic chain robots which are comparable in size.

Efforts of reducing mass and inertia of serial robot structures have been a primary research target, where the human arm with a weight-to-load ratio better than 1 : 1 was considered the ultimate benchmark. In 2006, robot manufacturer KUKA introduced their LBR lightweight prototype robot, a compact 7-DOF robot arm with advanced torque-control capabilities which has recently been introduced in high-performance industrial applications [54.15]. An obvious next step in approaching human dexterity is the recent introduction of two-armed robot designs with some recent developments being depicted in Fig. 54.7 [54.16]. In conjunction with a robot's capability to support safe human–robot collaboration, new manufacturing concepts can be implemented which expand capabilities, productivity, and ergonomic quality to manual workplaces [54.17].

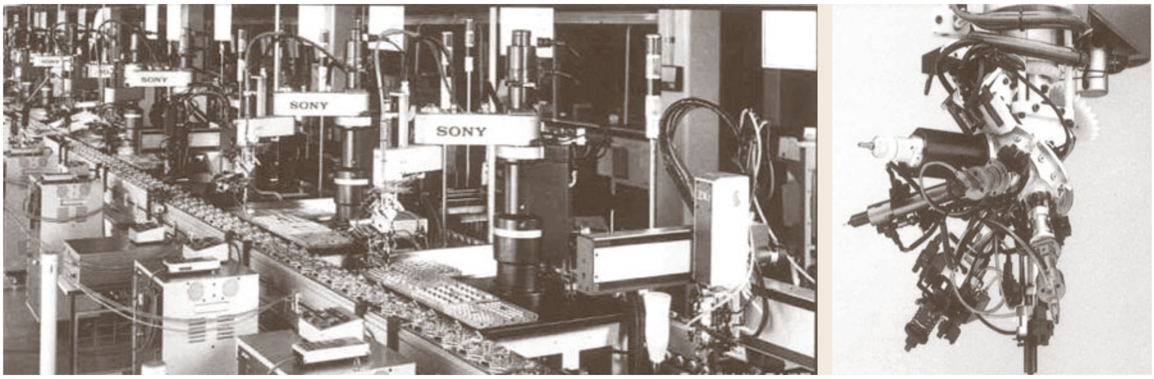



**Fig. 54.4** An automated video cassette recorder (VCR) assembly line (about 1989) with SCARAs carrying a turret with multigripper tools. Typically five parts are added by one robot before the VCR is moved to the next station of the automated assembly line

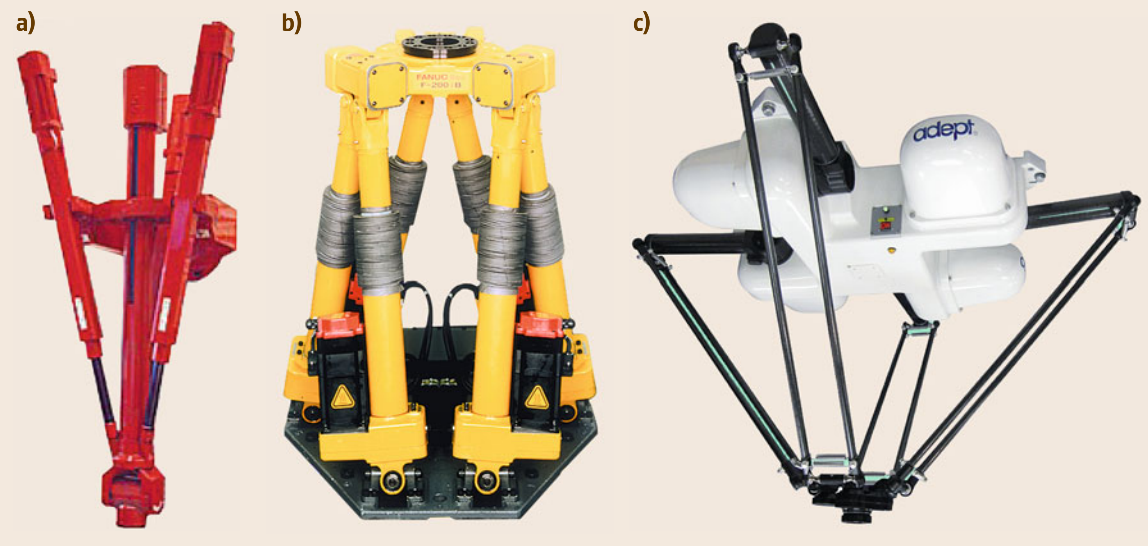



**Fig.54.5a–c** Parallel robots are slowly diffusing into various fields of industrial application: **(a)** the Neos Tricept 600, **(b)** Fanuc F-200iB. **(c)** Adept Quattro. **(a)** In 1992, Neos Robotics represented with their Tricept robot range a concept to combine the stiffness of machine tools with the dexterity of a robot for heavy-duty applications such as in friction stir welding (FSW) or machining of aluminum for the aircraft industry. **(b)** The Fanuc F-200iB introduced in 2002 is a 6-DOF parallel robot particularly designed for welding gun handling, deflashing, or for assembly tasks (100 kg payload, $\pm 0.1$ mm accuracy) in automotive assembly processes; **(c)** the Adept Quattro (introduced in 2007, following the ABB FlexPicker in 1998) is suited for high-speed applications in packaging, manufacturing, assembly, and material handling. The quad dual-link arm design forms an over-determined kinematic linkage, which in the wrist is converted to the forth axis of end-effector rotation by means of an internal transmission in the wrist

In parallel to industrial robots, automated guided vehicles (AGV) have emerged. These mobile robots are used for moving workpieces or loading equipment following a predetermined or virtual path in industrial environments. Within the concept of automated flexible manufacturing systems (FMS) AGVs have become an important part of their routing flexibility. Initially, AGVs relied on prepared floors such as embedded wires, magnets, or other tags for motion guidance. Meanwhile, freely navigating AGVs along virtual trajectories are entering large-scale manufacturing and logistics. Usually, their navigation is based on laser scanners that provide an accurate two- or even three-dimensional map of the actual environment for self-localization and obstacle avoidance. Early on, combinations of AGVs and robot arms were sug-

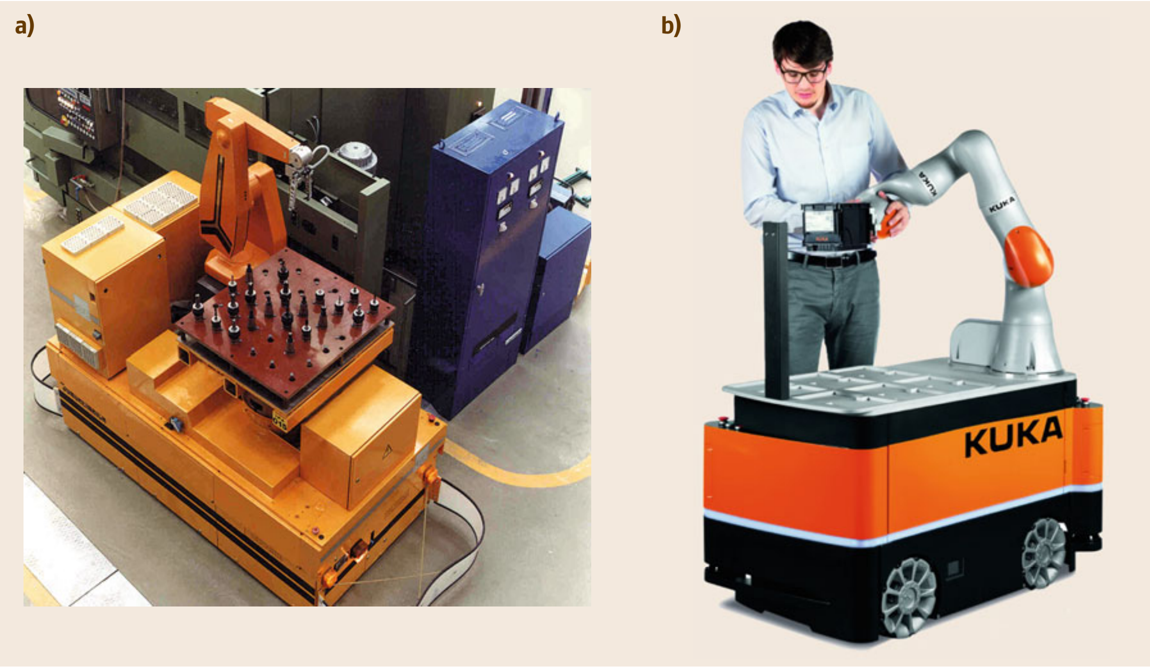



**Fig.54.6a,b** Mobile robots were introduced in the early 1980s for increased flexibility and reliability in factory logistics. **(a)** The MORO (1984) developed at Fraunhofer IPA was one of the first prototypes to combine a robot arm on a wire-bound mobile platform which follows a wire buried in the floor. **(b)** The KUKA omniRob features an omnidirectional platform and the LBR iiwa lightweight arm which form a highly kinematically redundant robot system (courtesy of KUKA)

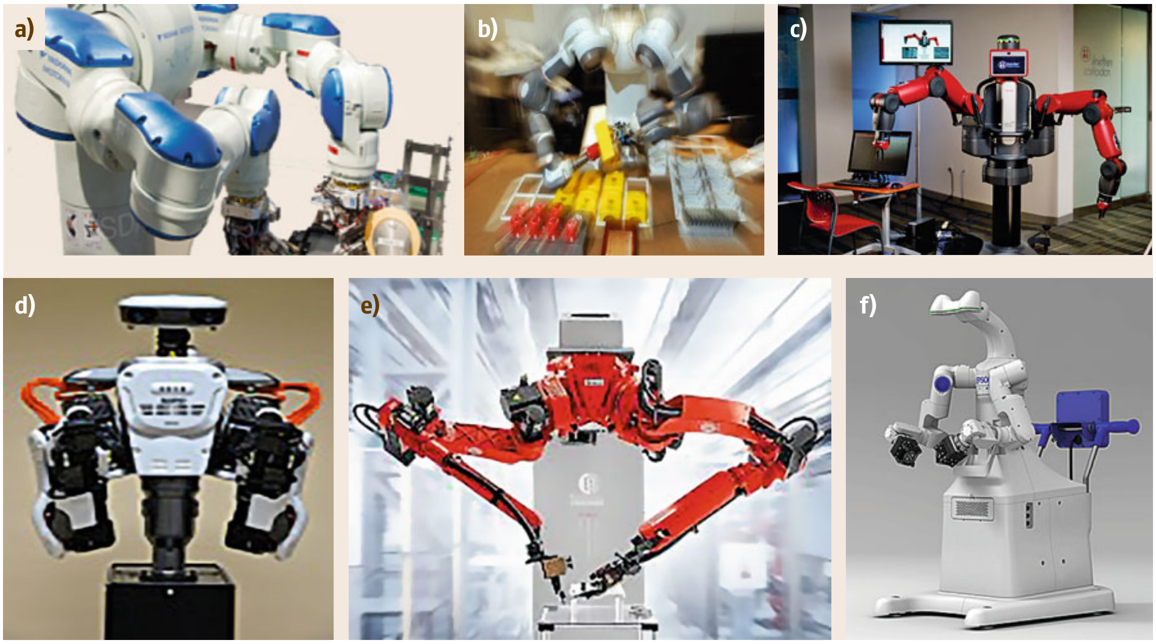



**Fig.54.7a–f** Examples of different designs of dual-arm robots (courtesy of **(a)** Motoman, **(b)** ABB, **(c)** Rethink Robotics, **(d)** Kawada Industries, **(e)** COMAU, **(f)** Seiko Epson)

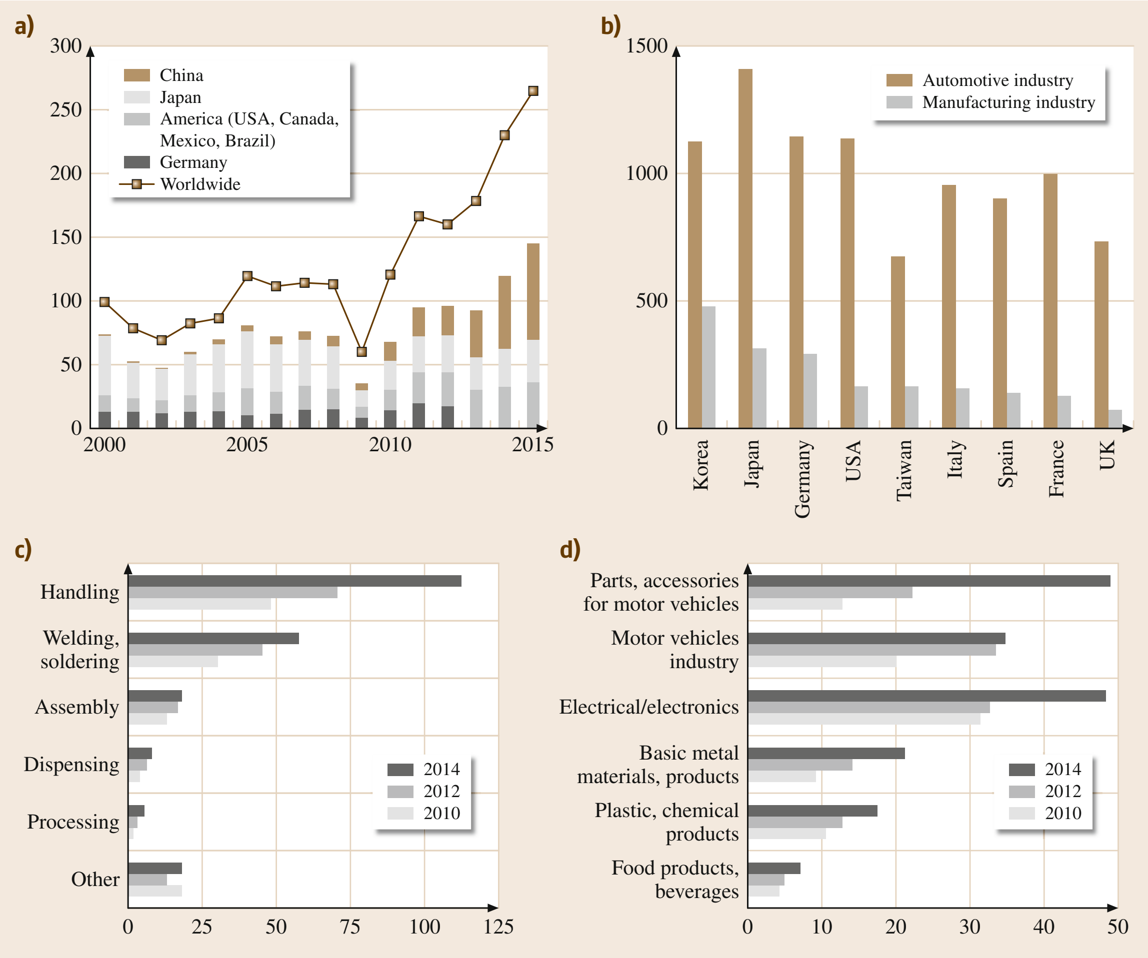



**Fig.54.8a–d** Statistics of worldwide industrial robotics use (after [54.1]). **(a)** Estimated annual robot installations in selected countries (1000 units, estimate for 2015), **(b)** Number of multipurpose industrial robots (all types) per 10 000 employees in the automotive and in manufacturing industries 2014. **(c)** Estimated worldwide annual shipments of industrial robots in main application areas. **(d)** Estimated worldwide annual shipments of industrial robots in main industrial branches

gested to automatically load and unload machine tools (Fig. 54.6). Safety and power supply have been an obstacle to these system's diffusion in industrial practice. Currently, first solutions for mobile manipulation appear [54.18].

The ability to use human and robot workers either interchangeably or in workspace sharing/collaboration scenarios in human workplaces motivated the design of anthropomorphic dual-arm robots (Fig. 54.7. Even though industrial acceptance initially has been low, advances in programming comfort, securing safe human–robot coexistence/collaboration and system cost have led to significant interest in using dual arms in agile manufacturing concepts, particularly in assembly and handling applications [54.19]. The dual-arm systems suggest a new way of using powerful and lean type of robot which is easy to install by the manufacturing end-user with little adaptation of manual workplaces.

Today, industrial robotics is seen as a central pillar to future manufacturing competitiveness and economic growth:

- The International Federation of Robotics (IFR) estimates that between 2000 and 2008 the robotics industry had created 8–10 million highly qualified jobs, either directly or indirectly. The prediction is that between 2012–2020 another 4 million jobs will be created in the *robot ecosystem* [54.20]. The extent of job creation by robotics has been discussed controversially. It is undisputed, however, that a wider use of robots in manufacturing is able to significantly strengthen a competitive position of a company or an industrial sector [54.21]. Economically, manufacturing productivity gains are particularly effective for economic growth. There is no sustainable product innovation without manufacturing competence which includes knowledge and practice of planning, designing, and operating advanced robotic systems [54.22] (Fig. 54.8).

- The average price for a robot in 2014 was in the order of US$ 46 800, which is about one-third of its equivalent price in 1990. At the same time, robot performance parameters such as speed, load capacity, and mean time between failures (MTBF) have dramatically improved. This means that automation has become more affordable, providing a faster return on investment [54.1].
- Traditionally, robot automation has not played a significant role in the implementation of lean manufacturing strategies. However, efforts are underway to introduce industrial robots to lean, agile manufacturing. Characteristics are robot solutions that can be flexibly added to manufacturing systems *on demand*, that are significantly less expensive on a life-cycle-costing (LCC) basis than today's systems due to reduced peripherals and systems integration (*system out of the box*) [54.23]. With robots becoming commodities in manufacturing they might be used as intuitively and naturally as a handheld power tool today. This would imply intuitive and safe human–robot collaboration and versatility due to advanced sensing, control, and embedding the robot set-up and operation in an IT infrastructure.
- Factories of the future will represent a network of self-organizing cyber physical systems (CPS). As part of this industrial internet CPSs embed computation, networking, and physical processes and can either represent manufacturing equipment such as machine tools, fixtures, trays, conveyors, tools, etc. or the workpiece which controls and memorizes its production. Robots are considered the centerpiece of future smart factories which combine manufacturing agility, profitability, human ergonomics, and minimized resource consumption [54.24].
- Robots in assembly have not reached their predicted installation potential mainly due to cheap labor cost and the lean assembly work systems which support highest flexibility and productivity at minimal waste. A reason is partly seen in the slow advances in dexterous manipulation for assembly tasks with industrial robustness. Here, torque-controlled lightweight robots [54.25] and two-armed robot systems have been proposed to imitate human ergonomics and task execution [54.26].
- New financing models such as leasing, *pay by service* will allow end-users to use robots on demand or to have manufacturing service providers to operate manufacturing lines on a *pay-on-production* basis [54.27].

Figure 54.8 depicts some key figures on the recent extent of industrial robot diffusion into manufacturing.

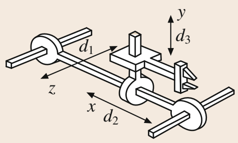
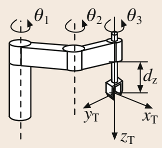
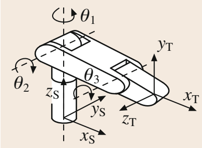
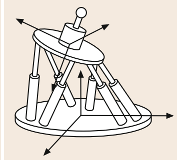
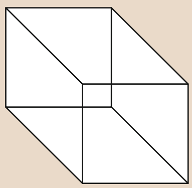
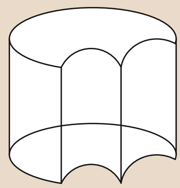
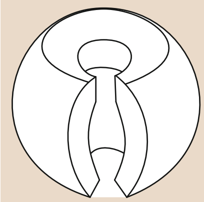
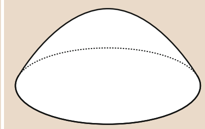
## 54.3 Industrial Robot Kinematics

By definition, an industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications [54.28]. Robots can be categorized according to their number of independent kinematic axes, and their mechanical structure which affect most of the robots' kinematic properties, the computation methods used to determine joint motions, and the form and size of the robot workspace. Robot mechanical structures are composed of links that are rigid bodies connecting neighboring (prismatic, rotary, cylindrical or spherical) joints. The diagrams in Table 54.1 show several common types of robot mechanical structures. Of course, the workspace of industrial robots can be significantly expanded by placing the robot arm on an additional linear axis, sometimes reaching a length of more than 50 m, or even on mobile platforms. Furthermore, robot mechanical structures can be composed by joint modules which are connected by links to form task-specific designs.

With advances in the state of the art in motion control and computer hardware processing capabilities, computation is much less a constraint on mechanism choice than it was for early robot designers. The choice of mechanical structure of the robot depends mostly on fundamental mechanical requirements such as payload and workspace size. Considering a given level of cost, there is usually a trade-off between workspace size and stiffness. To enable the robot to reach inside or around obstacles it is clearly advantageous to use an articulated mechanical design.

Considering also the stiffness and accuracy (in a practical sense considering what is reasonable to build), the picture is more complex. Each of the first three types in Table 54.1 we refer to as serial kinematic

**Table 54.1** Main categories of mechanical structures of industrial robots: Gantry is what a Cartesian coordinate robot is typically called with three prismatic joints, whose axes are coincident with a Cartesian coordinate system. The SCARA or selective compliance assembly robot arm has two parallel rotary joints to provide compliance in a selected plane. The articulated robot has three or more, typically six, rotary joints placed in series with their interconnecting links. The parallel link robot is characterized by links that form closed loop structures shown with prismatic joints, but can also have revolute joints such as the Delta robot (Fig. 54.5) (pictures courtesy of Güdel, ADEPT, ABB, PI Physik Instrumente)

| Category | Gantry (or Cartesian) | SCARA | Articulate | Parallel |
|---|---|---|---|---|
| Robot main axes structure |   3 prismatic joints |   1 prismatic and 2(3) revolute joints |   3 revolute joints |  Typically with 3, 4 or 6 prismatic axis |
| Workspace shape |  |  |  |  |
| Technical example |  |  |  |  |

machines (SKMs), while the last is a parallel kinematic machine (PKM). To obtain maximum stiffness, again for a certain minimum level of cost, the end-effector is better supported from different directions, and here the PKM has significant advantages. On the other hand, if high stiffness (but not low weight and high dexterity) is the main concern, a typical computerized numerical control (CNC) machine (e.g., for milling) is identical in principle to the gantry mechanism. There are also modular systems with servo-controlled actuators that can be used to build both robots with purpose-designed mechanisms.

## 54.4 Typical Industrial Robot Applications

Out of the many possible uses of industrial robots selected case studies on high-potential robot applications will be briefly described. Typical associated enabling technologies will be depicted.

### 54.4.1 Handling

Handling in robotics comprises numerous processes such as grasping, transporting, packaging, palletizing, and picking. As seen in Fig. 54.8 handling is the largest robot application field which is found in all branches of manufacturing and logistics. A central feature and major challenge in the engineering of robotic handling systems is the design of the gripper and associated grasping strategies given the physical properties of the workpiece, throughputs, and uncertainties regarding object geometry and location. Current high-potential application of robot handling systems are: tending of CNC machines for workerless shifts [54.29], palletizing, and lifting of objects for ergonomic reasons or

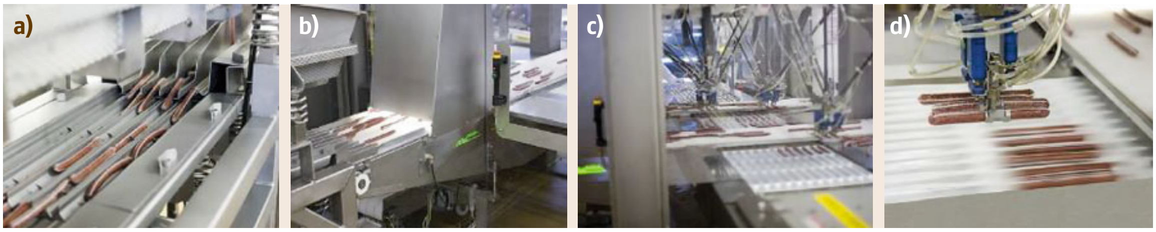



**Fig.54.9a–d** Units of sausage are cut from strings, then placed into the thermo-formed cavities before applying lidding. The coordination of the robots and the optimization of the picking frequency require a selection of the best path for each robot. Missed sausages are fed back on the conveyor for another try. The shown 4-DOF parallel robot reaches cycle times of 1–3 Hz and can move payloads of up to 8 kg (courtesy of robomotion, Germany)

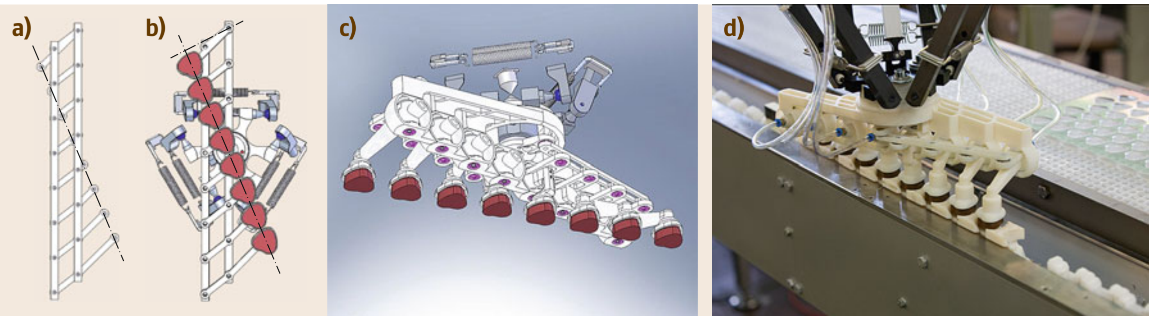



**Fig.54.10a–d** Lightweight customized vacuum gripper (0.75 kg mass) through additive manufacturing. Cookies are delivered continuously on a belt, grasped from the belt, and batches of eight are put on a blister matrix before final packaging. The gripper's spacing is pneumatically actuated and its rotation through the parallel robot's central rotational axis (courtesy of robomotion, Germany)

when limitations specified in load handling regulations are exceeded [54.30], for reasons of cleanliness as is typical in the food, pharmaceutical, and semiconductor industries [54.31–33], avoiding monotonous work and psychological strain, and ensure logistics quality through workpiece or object tracking [54.34]. In the following, two use-cases of material handling will be highlighted, each in a different industrial domain, and based on specific industrial robot type and enabling technologies.

#### Food Handling

The food sector is claimed to have significant potential for the application of robots as fundamental change to productivity, product quality, and worker ergonomics can be achieved [54.35]. In food automation, *untouched by human hand* entails critical requirements for robot automation such as the need for hygienic design, operational speed, ease of programming, and cost. In the past, these requirements had been difficult to achieve due to high throughputs, therefore requiring rapid grasps and fast robot motion, robust sensing for detecting object locations on conveyor belts. High speed at high flexibility apparently is a key in the individual handling of food objects. Therefore, fast SCARA and parallel robots have found wide acceptance in this field.

An example of a packaging line in food production is depicted in Fig. 54.9 where cut mini salamis are delivered in four streams per conveyor belt in random sequence. The positions of the sausages on the translucent belt are determined by a computer vision system. The robot picks the sausage from the measured position sequentially until the gripper holds three sausages which are then placed into cavities. With four parallel robots a maximum pick rate of 600 sausages per minute can be processed. Key of the application is its high-speed 2-D (two-dimensional) computer vision system which feeds the robot's path planning for collision-free picks at a minimum loss rate of unpicked salami [54.37].

Recent efforts have led to customized designs of gripper systems through 3-D (three-dimensional) printing (additive manufacturing) which for instance includes actuation through pneumatically driven bellows and low-wear metal joints. An example of a highly actuated gripper based on 3-D-printing for use in food handling is depicted in Fig. 54.10. Additive manufac-

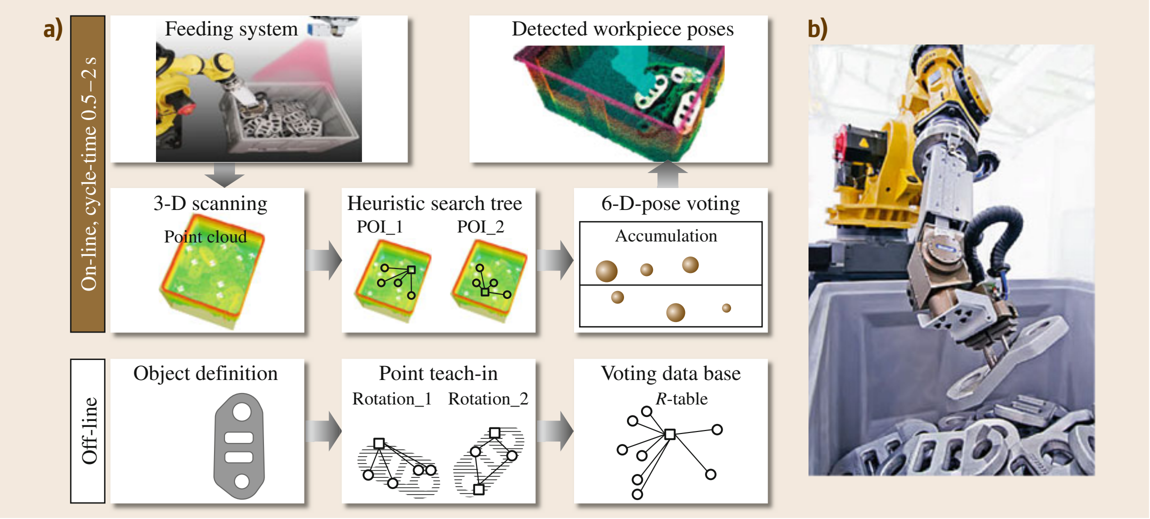



**Fig. 54.11a,b** Procedure of a bin-picking method [54.36] **(a)** and gripper with additional degree of freedom for reaching deep into bins **(b)**. Depicted is a 2-D laser scanner on a swiveling unit for acquiring the point cloud in parallel to the robot's motion. The object detection itself consumes 0.5–2 s and is less time critical than the robot's motion and grasps

turing processes seem to be perfectly suited to achieve higher flexibility in manufacturing automation [54.38]. Numerous materials have become available for different additive manufacturing processes so that even specific manufacturing requirements can be matched. Initial doubts about gripper durability have been dispelled: lifetimes of more than 10 million load cycles for robot grippers manufactured on the basis of laser sintered polyamide have been reported.

### Bin-Picking

Generally, industrial practice in robot workcell planning aims at finding a compromise between reducing the variation of the workpiece location and the cost of sensor systems to compensate for residual variation or uncertainties. Today, nearly all parts arrive at robot workcells in a repeatable manner, either being stored in special carriers or magazines, or by being transported and oriented by vibrating devices that allow the parts to settle into a predictable orientation for proper robot grasping. However, cost and flexibility requirements in manufacturing automation will result in reducing customized parts magazines to more universal carriers, containers, or conveyor belts. If randomly oriented on a conveyor belt or in a carrier, parts have to be properly identified and located so that the robot can produce an collision-free grasp.

The challenge of grasping partly or randomly ordered parts by robot has been referred to as bin-picking and has been investigated by numerous researchers since the mid-1980s [54.39]. Even though an abundance of approaches has been presented, only recently bin-picking installations have found their way into daily manufacturing in significant numbers. Bin-picking algorithms follow a typical sequence of steps: initial point cloud data acquisition, object detection, pose estimation, collision-free path and grasp planning, object grasping, and object placement. Most methods in bin-picking assume known geometrical representations (a computer-aided design CAD-model) of the workpiece in question including the specification of admissible grasps for applying template-matching methods [54.40, 41]. Figure 54.11 depicts a variant of a fast template-matching method, which encompasses the following steps for detecting object poses:

- For detecting the scene (e.g., a carrier or box filled randomly with workpieces) typically laser-based sensor systems are used for acquiring a sufficiently dense point cloud. The object pose detection then is considered as a combinatorial optimization problem for which a construction heuristic is applied. For this heuristic tree search, a finite set of possible workpiece poses is initially derived from the search space.
- In order to use a decision tree, the elements of the search set are split into two components: The first component describes a point of interest (POI) in the search space which is part of the workpiece

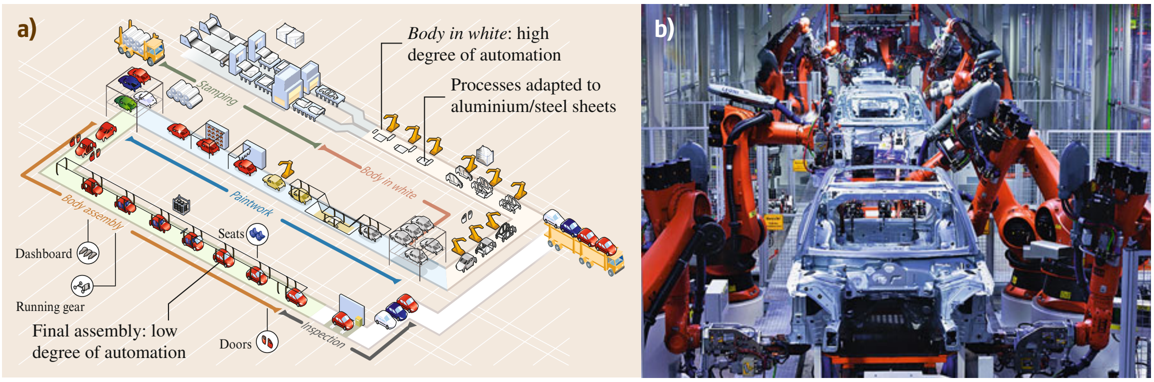



**Fig.54.12a,b** Car production **(a)** usually follows the illustrated steps along the assembly line: Stamping of the metal sheet into plates, fixing and alignment of the plates on trays, spot welding, painting the car body, and final assembly of the car body (doors, dashboard, windscreens, power-train seats, and tires). Car factories can host over 1000 industrial robots working typically three shifts per day (courtesy of PSA Peugeot Citroën, Paris and Art Movies, Paris). The Audi plant in Ingolstadt Germany **(b)** is highly automated. The picture shows spot welding robots along the *body-in-white* transfer line. Trays carrying car bodies pass through the *robot garden*

surface. The second component describes possible workpiece poses relative to a POI. The partial search quantities obtained thereby have a significant lower complexity compared to the original search set, since the points of interest can provide a constraint on relative workpiece poses, thus restricting their assumed freedom of movement.

- Typical tree search strategies such as best-first search can be used. In that case best-first search explores the search tree by expanding the most promising nodes first. These nodes are chosen according to a heuristic evaluation score, representing the estimated distance from the node to a solution.
- Final evaluation of the workpiece poses is provided by a six-dimensional (6-D) Hough voting procedure, i. e., a generalized Hough transform. The features used for Hough voting are sensor measurements located relative to a POI. For all possible constellations of a sensor measurement relative to a point of interest, a probabilistic statement about possible workpiece poses can be made. Through the superposition of all probability statements, solution candidates can be formed, which are subjected to a statistical test based on a quality rating. The obtained quality rating along with a given level of significance is used in order to decide about the acceptance of a workpiece pose.

The method is able to locate three to four workpieces on average within 0.5–2 s, using a standard desktop computer. Moreover, robot grippers are equipped with a seventh axis to allow grasping parts from corners of the bin. Furthermore, grippers should be formed in such a way that they may reach deep into the bin so that they offer only minimal collision volumes.

## 54.4.2 Welding

Welding is a manufacturing process that joins materials by applying heat, sometimes with pressure. Usually, workpiece material is melted at the process location often with additional filler material. Typical robot-based welding processes are spot welding, particularly in car body assembly, and gas-shielded metal arc welding (GMAW). With increasing compactness of laser sources and robot motion accuracy, laser welding is in the rise.

Manual welding requires skilled workers, as small imperfections in the weld can lead to severe consequences. Furthermore, welders are exposed to hazardous working conditions (fumes, problematic ergonomic working positions, heat, and noise) so that the use of robots has become beneficial in GMAW processes even for the smallest lot-sizes. Commonly, the automatic arc-welding process is based on a consumable wire electrode and a shielding gas that is fed through a welding gun. Modern welding robots are particularly suited through the following characteristics:

- Computer control allows efficient programming of task sequences, robot motions, external actuators, sensors, and communication with external devices such as welding sources.

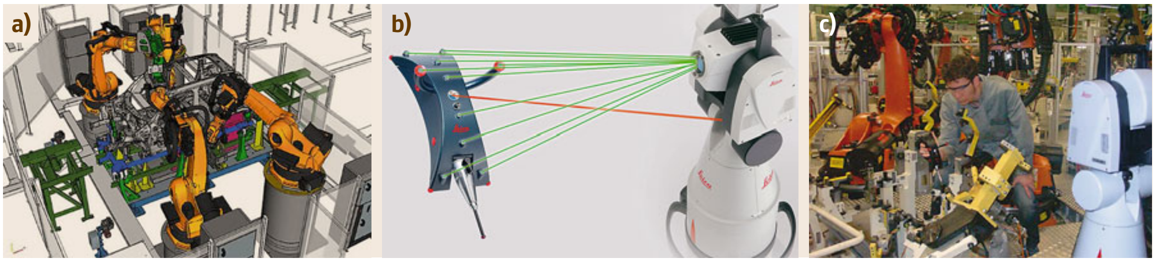



**Fig. 54.13a–c** Offline programming of a spot welding workcell. **(a)** The robot workcell and the task execution are modeled on the basis of realistic robot models (geometry, kinematics, kinetics). **(b)** The shown laser tracker is a portable measurement system that relies on laser beams to accurately measure in a radial volume (accuracy of $\pm 10\,\text{ppm} = 10\,\mu\text{m/m}$, up to 80 m in diameter, measuring rate up to 3000 points/s). If measured objects cannot be equipped with reflecting targets or reached by the tracker, handheld probes are tracked instead. **(c)** A tracker in use for interactively measuring the geometry of the robot workcell (courtesy of Leica, now Hexagon MI)

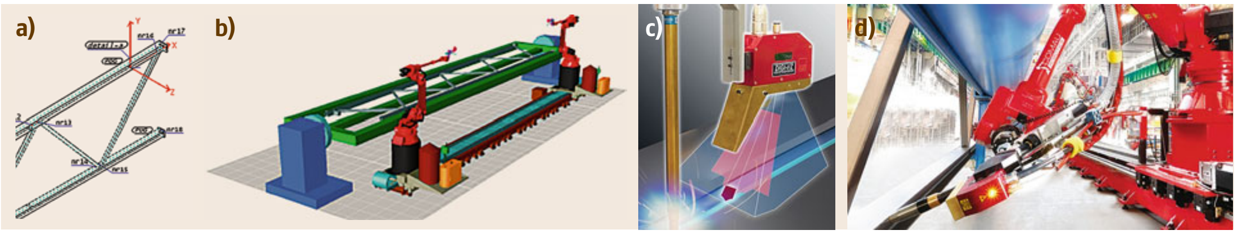



**Fig. 54.14a–d** GMAW welding of building trusses in lot-size one by robot. Illustration **(a)** shows the CAD drawing of a steel truss with relevant information for welding process, **(b)** one-half of the welding workcell with two welding robots working simultaneously on the truss when the neighboring truss on the other half is loaded or unloaded, **(c)** the laser-based seam finding and tracking sensors and **(d)** the welding robot (courtesy of Servo Robot, Canada; Goldbeck, Germany)

- Free definition and parameterization of robot positions or orientations, reference frames, and paths.
- High repeatability and positioning accuracy of paths. Typically repeatability is some $\pm 0.05\,\text{mm}$ and positioning accuracy is better than $\pm 1.0\,\text{mm}$. These values can be significantly improved through modern robot calibration methods [54.42].
- High speeds of the end-effector of up to 8 m/s for quick approach and depart motions.
- Typically, articulated robots have six DOF so that commanded orientations and positions in their workspace can be reached, which in the welding case means there is one DOF free for rotation around a rotational-symmetric welding tool. Additionally, workspace extensions by mounting the robot on a linear axis (seventh DOF) or even on mobile bases are common, especially for welding of large structures.
- Typical payloads range 6–150 kg. Higher load capacities are required for spot-welding guns (typically $> 50\,\text{kg}$) and their cable package.
- Programmable logic controller (PLC) capabilities such that fast input/output control and synchronizing actions within the robot workcell are accomplished.
- Interfacing to high-level factory control through factory communication networks.

Electric current sources, torches, and peripheral devices for automatic cleaning and maintaining the GMAW torch (anti-splatter, wire-cutting, tool changer, etc.) are offered by specialized companies. Often sensors are used to track welding gaps and measure weld seams either before or synchronously with the welding process, thus adapting the robot's trajectory in the presence of workpiece variation and distortion. Also, collaborating robots have been introduced where one robot fixes and moves the workpiece in synchronization with another robot carrying a welding tool so that the weld can be performed with the pool of molten metal horizontal.

### Spot Welding in the Automotive Industry with Offline Programming

Car manufacturing has been one of the key drivers in the technical development of industrial robots as the precision handling of spot-welding guns was one of the first breakthrough use cases (Fig. 54.12) *body-in-white* (i. e., unpainted car body) assembly is mostly done by robots, very much in contrast to the final assembly which is

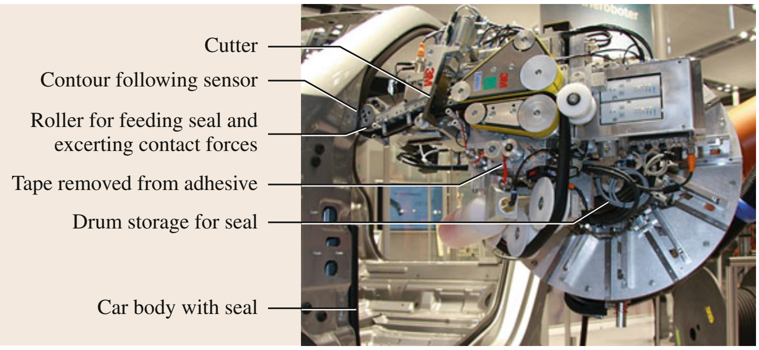



**Fig. 54.15** A robot-guided tool for handling and processing limb material, in this case a self-adhesive seal for car bodies

dominated by manual work. Demands for faster cycle times have led to a concurrent and coordinated motion of the spot-welding gun and robot: the robot continues to move while the weld gun is simultaneously rotated about the electrode axis during welding [54.43].

Most of a spot welding robot's programming is done using offline programming (OLP) packages (Fig. 54.13. A library of robots, devices, and advanced CAD capabilities helps plan, program, visualize, and optimize layouts and complete production cycles under assumed manufacturing conditions. Robot programs can be generated and downloaded to robots workcells. A critical step is the calibration of the robot workcell with respect to the simulations [54.44].

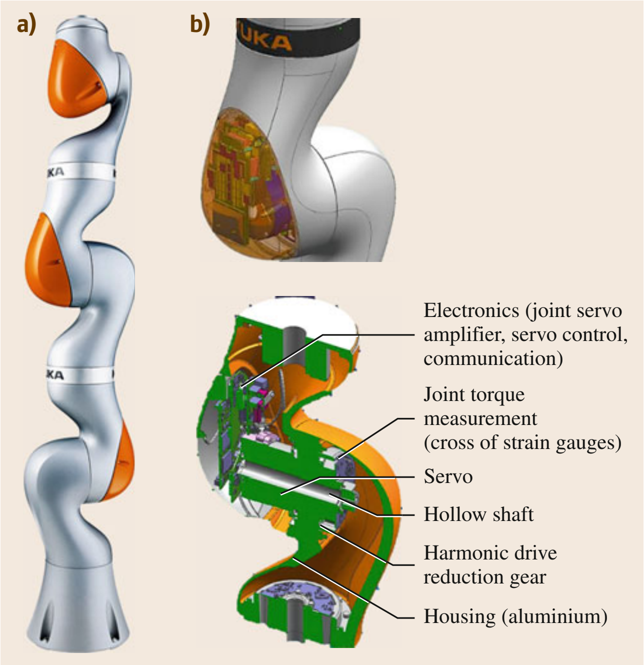



**Fig.54.16a,b** Design of the KUKA iiwa: **(a)** Shape and **(b)** integration of joint mechatronics

#### Arc Welding in Metal Construction

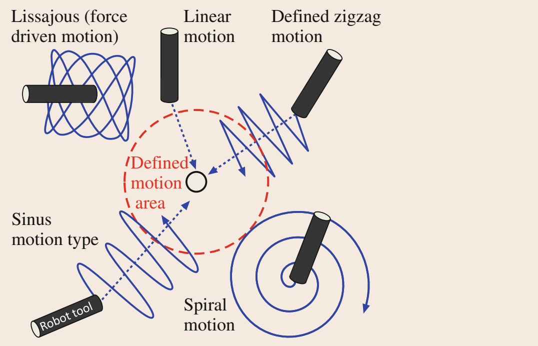
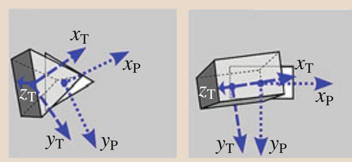
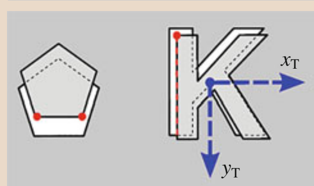
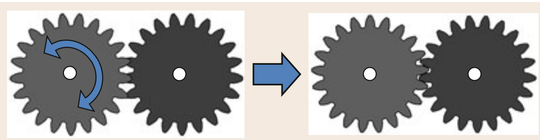
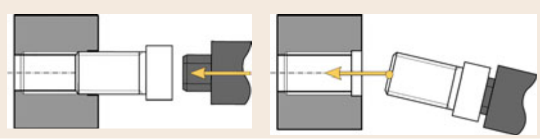
Normally steel constructions are designed using CAD programs that offer functions for GMAW-task definitions such as welding parameters, multipass seams, weld beads sequencing, etc. This information may be used for automatically generating welding robot programs, even in the case of lot-size one jobs.

As an example, the generation principle of a welding robot program is depicted in Fig. 54.14. Large-scale trusses of up to 15 m for large halls are welded-to-measure. The robot program is generated from the CAD drawing with process relevant information. Workpiece tolerances for example, induced by placing the steel components into the fixtures, by bending of the material under its own weight are compensated through active measurement. The robot-mounted sensor locates the weld seam by laser-based vision for shifting the generated programs in such a way that they match the actual weld seams. This calibration is automatically performed if expected and actual bead locations are within a range of $\pm 2.5$ cm.

### 54.4.3 Assembly

Assembly in manufacturing describes the combination of subsystems or components to systems of a higher complexity through joining. Assembly in manufacturing comprises four process groups: joining, handling, controlling, and auxiliary processes (cleaning, adjustment, marking, etc.) [54.45]. The composition of these four functions may vary depending on batch size, product, and throughput: from assembly workcells to high-throughput assembly lines. Assembly processes form up to 80% of a product's manufacturing cost and this is where the greatest competitive advantage can be gained [54.46]. Therefore, optimization in assembly includes tightly interweaved aspects: Design

**Table 54.2** Assembly subprocesses or *modules* and their implementation on the KUKA LBR iiwa

| Assembly subprocess | Characterization | Principle |
|---|---|---|
| *search*: search module which supports several search motions or search strategies | Search motion type examples:<br>• Linear<br>• Zigzag<br>• Spiral<br>• Sinus<br>• Lissajous<br>Commanding the search motion generation:<br>• Position-based trajectory<br>• Force-based trajectory<br>• Combination of position and force-based trajectory |   |
| *peg in hole*: execution of typical part joining motions | Reduction of arbitrary joining part types to three abstract planar types:<br>• Round (arbitrary axial orientation)<br>• Triangular (defined axial orientation)<br>• Rectangular (additionally defined workpiece coordination system)<br>Strategy of triangular types is common for numerous workpiece contours:<br>• Orientation is given, tipping is executed in one step; object pivots around one corner |   Parts T and P of joining motion with local coordinate systems     • *Red dot*: Corner which enters first • *Red dotted line*: edge which follows in tipping |
| *gear*, *search*: toothed gear joining motion and *screw-in* motion | • Meshing in toothed gears: Torque oscillations about gear axis and linear forward motion<br>• Screw-in motion: fixed torque for torque-based screwing<br>• Angle-controlled tightening required in most screwing process applications: command fixed torque, then turn by a defined angular increment. |   |

for Assembly (DFA), workcell and assembly line design as well as logistics and manufacturing organization [54.12]. Early on, industrial robots were used in assembly automation, particularly in high-throughput manufacturing lines (Fig. 54.4). However, robots are increasingly used in highly flexible workcells and will enter agile lean manufacturing workplaces as versatile tools at the hands of the human worker. In the following, selective use cases of robots in assembly will be described by detailing on specific enabling technologies.

### Assembly of Limb Material

Numerous assembly processes include handling of limb materials such as rubber hoses, wire harnesses, etc. that have to be fixed in position in order to be joined (Fig. 54.15). Obviously stabilizing the material and securing process quality often result in ingenious grip-

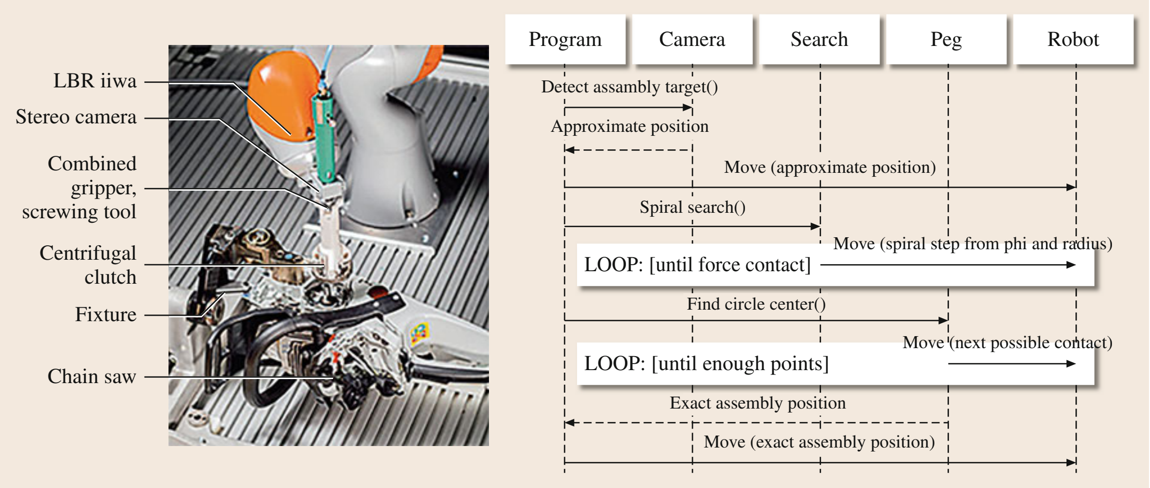



**Fig. 54.17** Set up and implementation of a centrifugal clutch assembly for a chain saw with a sequence diagram depicting the consecutive steps until tightening the clutch. Through the robot's torque sensors in each of its links and an appropriate kinematic and dynamic model, the resulting forces at the tool tip are controlled

per designs involving additional actuation and sensing functionalities.

An example is the automatic application of self-adhesive seals as they easily lose their shape and can be stretched or compressed. Since manual application of adhesive seals to vehicle bodies or doors is sensitive and ergonomically problematic a robot-guided tool has to secure bonding of the material's surface to the car body. The seal material is fed from a roll under correct tension and the tape, which covers the adhesive, is removed and stored in a small tank. At the tip of the tool a laser sensor follows the car body or door contour and an actuated roller produces a continuous normal force on the seal. Both the laser sensor and roller's motion are translated into a tension free motion of the robot. In addition, a magazine on a flange ensures that the seal is correctly tensioned and a material reserve for one car door is provided [54.47].

Here, the robot acts as slave to guide a tool which acts as both measuring unit and precision actuator with own master controller. Further efforts aim at embedding rich sensor and control modalities in the robot to account for complex process control based on tactile and geometric information.

### Advanced Robot-Based Assembly Process Control

Automation of advanced assembly processes depends on physical contact between the joined workpieces. For this contact formation to be controlled a robot should offer compliant motion control which is a control method that modulates robot position and velocity based on measured or estimated joint torques or contact forces [54.48]. Subject of intense research efforts for a long time application packages for compliant force control in industrial robots which fulfill requirements regarding versatility, robustness, and ease of use in programming have become available during the last ten years [54.49]. The solutions are commonly based on a 6-DOF force–torque sensor which is attached to the robot flange.

The fully torque-controlled DLR (Deutsches Zentrum für Luft- und Raumfahrt) lightweight robot broke new grounds as its 7-DOF redundant kinematic structure, torque sensing in every joint and a variety of compliance modes allowed difficult assembly tasks with complex contact formation during the joining process [54.15]. DLR and KUKA managed to successfully go the strenuous road from the original LBR invention, an idea made manifest in 1991, to a product, first applied in research and predevelopment of new industrial manufacturing concepts in a series of development steps: KUKA LBR3 (2006), LBR4 (2008), and LBR iiwa (2012) [54.25]. Figure 54.16 depicts the integrated mechatronic design of a joint with its unique joint-torque measurement.

To simplify the programming of complex joining processes several assembly subprocess modules were developed of which three are depicted in Table 54.2: the search for contact formation and the execution of two typical joining motions (peg-in-hole, toothed gear joining).

Figure 54.17 depicts a scenario of a force-based centrifugal clutch assembly for a chain saw. The robot detects and locates the clutch workpieces on a tray by vision sensors. Now the grasped workpiece has to be joined and tightly screwed onto the shaft. In this case the robot's tactile capabilities and compliant behavior are used to achieve a robust and quick assembly. Once a rough shaft center position estimate is acquired by the robot's vision sensor, the robot approaches to this position and starts performing a search pattern until contact detection. The peg-in-hole process is followed by a screw-in motion for tightening the clutch. Alternative designs may be based on 6-DOF industrial robots with a wrist-mounted force–torque sensor to measure process-induced forces on the robot, or simply estimate the forces from the motor torque control [54.50].

The sequence diagram shows a simplified execution trace of a spiral search of a circular shape. Listing 54.1 lists part of the Java code in the KUKA Sunrise control system which identifies the three parts of the compliant assembly: main function, spiral search until collision, and border walk.

*Listing 54.1 Extract from the code controlling the sequence of the centrifugal clutch assembly*

Main Program:

```
Frame rough = camera.detectAssemblyTarget(); // Retrieve rough position from stereo vision
// Move robot at 30 mm/s. Linear motion. Stop if force > 20 Newton
ForceCondition forceCond = new ForceCondition(robot.getDefaultMotionFrame(), 20);
robot.move(lin(rough).setCartVelocity(30).breakWhen(forceCond)); // First approximation
Frame target = spiralSearch(rough,forceCond); // Obtain the assembly target
assemble(Target); // Assembles the shaft at the given position
```

Search Module. Returns the assembly target:

```
Frame spiralSearch(Frame rough, ForceCondition forceCond){ // Spiral search until collision
 for (double phi = 0; phi < 10.0 * pi; phi +=pi / 90.0){ // 5 loops, 2° step
  Frame spiral = rough.copy();
  Double radius = max_radius * phi / (10.0 * pi);
  spiral.setX(rough.getX + Math.cos(phi) * radius);
  spiral.setY(rough.getY + Math.sin(phi) * radius);
  robot.move(lin(spiral).setCartVelocity(30).breakWhen(forceCond));
  if (motion.getFiredBreakConditionInfo())return findCircleCenter(); // It collides
 }
 return null; // Target not found}
```

Border walk to find shaft center:

```
Frame findCircleCenter()\textbraceleft // Calculates the center of a circular shape
 ArrayList contact = new ArrayList<Frame> (); // Stores frames where collisions take place
 do{
   Frame next = getNextFrame(robot.getForce());
   // Force feedback combined with search pattern such as a grid produces next frame

   robot.move(lin(next).breakWhen(forceCond)); // Move to next position, if possible
   if (motion.getFiredBreakConditionInfo()) contact.add(robot.getDefaultMotionFrame());
  } while (!areEnoughContacts(contact)); // Until enough contact points to identify center
  return getCircleCenter(contact); // Circle center can be determined with three points
}
```

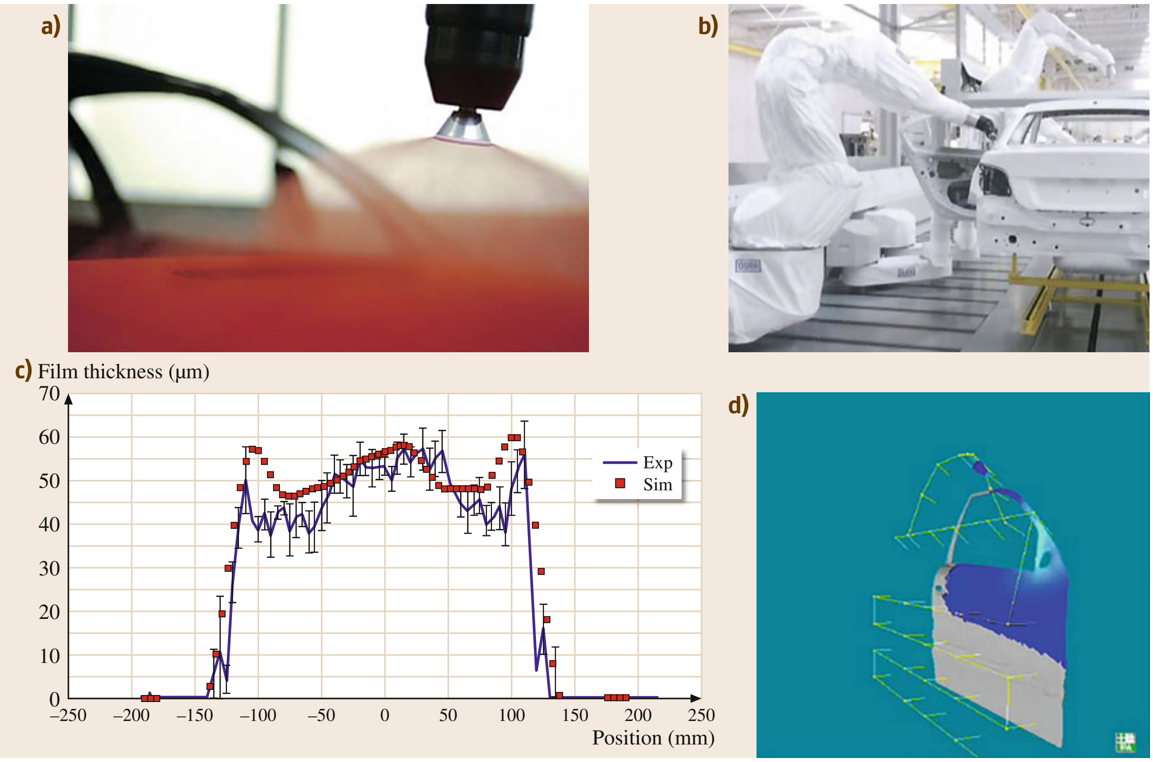



**Fig.54.18a–d** High-speed rotating atomizer and a multirobot workcell for car body painting. **(a)** A Dürr EcoBell2 paint gun which atomizes the paint material at the edge of the rotating bell disk by centrifugal forces. All current paint materials such as solvent-based or water-borne paints can be used in car production. **(b)** Multiple robots work in parallel for optimal throughput and accessibility of the car body. Most of the programming which includes the synchronization of the robots is performed offline, also with respect to optimal coverage of the painted surface. **(c)** Simulation of the painting process is critical for achieving highest yields (e.g., minimal overspray, uniform paint deposition, etc.) **(d)** Results from simulations for optimized program generation are shown (courtesy of Dürr, Fraunhofer IPA, Stuttgart)

### 54.4.4 Painting

Hazardous working conditions for human operators motivated Trallfa, a Norwegian company, to develop simple spray-painting robots in 1969, particularly for spraying bumpers and other plastic parts in the automotive industry. Initially pneumatically driven for anti-explosion reasons, today's robot designs are fully electric. They also have hooks and grippers to open hoods and doors during the painting task. Hollow wrists that house gas and paint hoses allow fast and agile motions. Spray guns for robots have evolved dramatically for delivering uniform quality using as little paint and solvent as possible and also for switching between different paint colors. Originally spray-painting robots replicated movements copied from human workers. Most of the programming for robot painting today is done offline as state-of-the-art programming systems offer integrated process simulations to optimize paint deposition, thickness, and coverage (Fig. 54.18). The simulation of the process is quite complex as different effects are taken into account such as turbulent flow field between atomizer and target, static electrical field between rotating bell disk and target, charging of the paint droplets at the bell, space charge effects due to the flow of charged paint droplets, and Coulomb forces acting on the droplets [54.52].

### 54.4.5 Processing

Material removal processes such as grinding, deburring, milling, and drilling are increasingly carried out by industrial robots with serial kinematics as they combine dexterity, versatility, and cost-effectiveness. The employed process tools are often combined with passive compliance or active force control as the workpiece geometry commonly exhibits tolerances in geometrical or material properties [54.53]. However, robot accuracy (±0.5 mm range) compares poorly to values in the ±0.01 mm range of typical machine

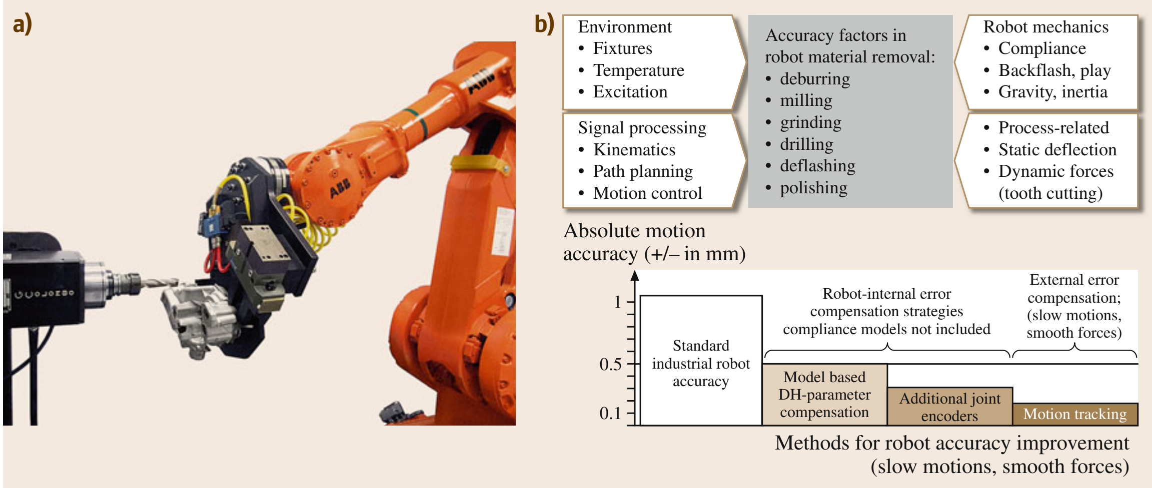



**Fig.54.19a–d** For small-to-medium-sized parts, a preferred configuration is to have the part grasped and guided by the robot relative to the fixed spindle. **(a)** The robot's gripper is mounted on a force–torque sensor to measure and limit process forces. For machining the edges of the workpiece (deflashing) the machining software package provides a machine-learning technique for optimising the motions [54.51]. **(b)** Influences on accuracy in robotic machining and methods for robot accuracy improvement. Still typical machine tool motion accuracies are in the order of $\pm 0.01$ mm or better (courtesy of ABB, Fraunhofer IPA)

tools [54.54]. The lower eigenfrequencies and damping coefficients of mechanical structures should be as high as possible for precision: The lower eigenfrequencies of milling machines are in the range of 50–100 Hz as compared to 20–30 Hz for typical industrial robots [54.55, 56]. Figure 54.19 depicts the factors that affect the robot's accuracy in typical material removal applications. Figure 54.20 shows first eigenfrequencies, damping, and stiffness in Cartesian space. These characteristics are essential for the design of machining processes and resulting workpiece quality. More robot-guided processes such as laser welding and laser cutting depend on achieving similar accuracies.

Robot position accuracy results from geometric error sources (deviations between actual robot structure and assumed Denavit–Hartenberg (DH) parameters) and nongeometric parameters (compliance of the me-

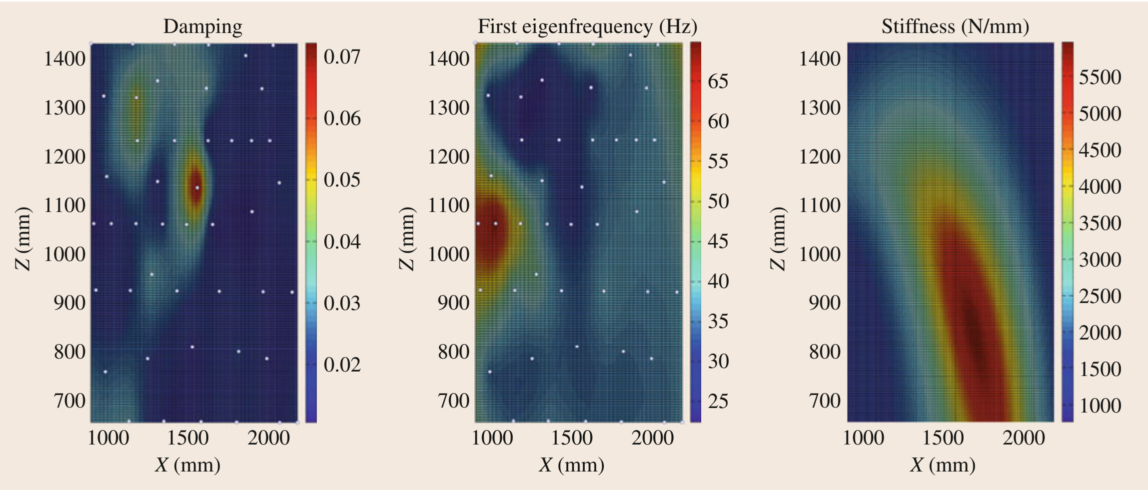



**Fig.54.20a–c** Robot dynamics in Cartesian space: **(a)** Damping and **(b)** first eigenfrequencies for a KUKA KR60 and **(c)** stiffness for a KR125 in a typical XZ-process plane (measured from the robot's first axis) of the robot's workspace (courtesy of ISG Stuttgart, Fraunhofer IPA)

**Table 54.3** Characteristics of computerized numerical control (CNC) and robot controller (RC)

| Category | CNC | RC | Interpretation |
|---|---|---|---|
| Targeted application | Machining, material removal | Handling, assembly | A CNC machine tool is single purpose; generally robots are universal machines |
| Motions | Path-based, complex contour-oriented | Point- or path-based and motion time-oriented | Extended look-ahead of CNC controllers allows detailed path description and adaption on ≫ 100 via points (< 10 via points in most robot controllers) |
| Programming | On-site programming based on standardized programming language (G-code), ISO (International Organization for Standardization) 6983, use of computer aided manufacturing (CAM) tools | On-site teaching (teach pendant, editor) based on supplier specific languages, use of typical robot simulation environments | Whereas robots are traditionally programmed manually on site, CNC controllers use CAM technology to generate complex paths automatically based on CAD data |
| Command reading | Online interpreter, continuous loading of instruction | Initial loading of programs, which are usually interpreted, sometimes compiled) | Program size in robot controllers limited by memory, CNC interprets programs online, may execute an unlimited number of commands |

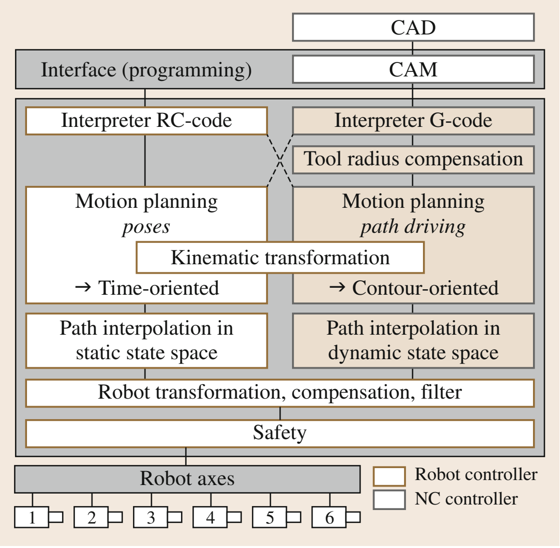



**Fig. 54.21** Structure of an NC-kernel integrated into a robot controller (courtesy of ISG Stuttgart)

chanical structure and gear play). In order to reduce the impact of the nongeometric parameters several stepwise approaches can be implemented:

- In combination with force prediction or online measurements robot compliance can be compensated by means of joint stiffness models [54.57]. In drilling applications additional encoders are sometimes mounted on the arm side of robotic joints in order to measure gear-induced joint compliance and backlash, or those effects can be estimated based on determined joint properties [54.58], so that a compensation can be achieved.
- In combination with the geometric error calibration accuracies of better than $\pm 0.2$ mm for typical robots have been realized in larger robot workspaces of $3 \times 3 \times 2\,m^3$ [54.59].
- For higher accuracies further sensor and actuators systems have been introduced. Using optical tracking deviations of $\pm 0.2$ mm have been demonstrated in steel [54.60]. With additional actuation deviations could be reduced to $\pm 0.1$ mm [54.61].

Robot and CNC machine tool controls may have similar origins, but have taken different paths in developments over the years as depicted in the following Table 54.3.

Increasingly CNC controllers are used for robots in material removal applications for taking advantage of the well-established off-line programming tools in the CNC world and for improving motion accuracy of the robot for complex 3-D contours (Figs. 54.21 and 54.22. Modern robot controllers integrate so-called numerical control (NC) kernels which share components of the robot controller such as user interfaces, kinematic transformations, and safety functions.

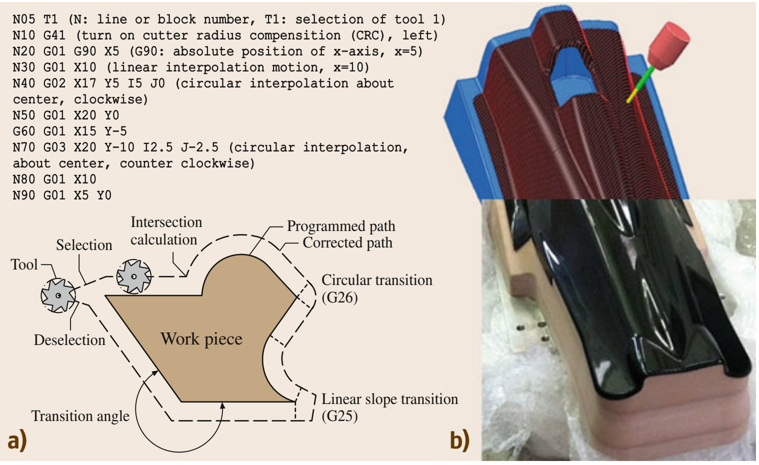



**Fig. 54.22** **(a)** A simple CNC program (G-code, ISO 6983). A CNC program normally is machine independent and orients itself on the workpiece contour after processing. **(b)** Most CNC programs are generated automatically through CAM tools that transfer geometric information into executable G-code. The example shows generated tool trajectory generation for precision milling of a carbon fiber (CF) part and the finished workpiece (courtesy of Delcam, Birmingham; ISG, Stuttgart)

## 54.5 Safe Human–Robot Collaboration

Human–robot collaboration allows the combination of typical strengths of robots with some of the numerous strengths of humans. Typical strengths of industrial robots are high stamina, high payload capacity, precision, and repeatability. Strengths of human workers that are unmatched by any machinery comprise flexibility for new production tasks, creative problem-solving skills, and the ability to react to unforeseen situations.

However, industrial robots have a significant potential to harm humans. Therefore, standards for designing and operating industrial robot automation systems have been introduced and found international acceptance. Since 1999, efforts have been made to define measures, rules, and examples specifically for robots in collaborative modes.

### 54.5.1 Overview of Basic Robot Safety Standards

Machine safety standards provide guidelines for designing and operating any type of machinery. Their consideration is optional, but exhibits the straightforward way of verifying the fulfillment of the fundamental health and safety prerequisites of the Machinery Directive [54.63, 64]. Generally, safety standards are classified into three categories (Fig. 54.23):

- Type A standards: basic standards that define fundamental principles to achieve machine safety.
- Type B standards: generic standards that regulate either a specific safety aspect, e.g., separating dis-

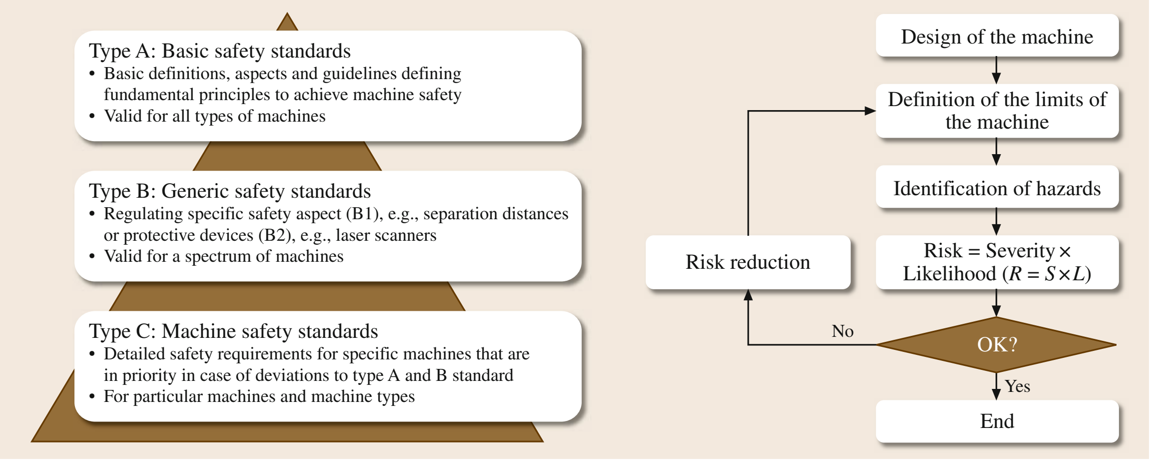



**Fig. 54.23** Different types of standards and a simplified iterative risk assessment procedure. See ISO 12100 [54.62, Fig. 1 and 2] for a more detailed schematic representation of risk reduction process

tances, or specific protective devices, e.g., optoelectronic protective equipment like laser scanners or emergency stop (ISO 13850), and are valid for a spectrum of machines.

- Type C standards: machine standards that list detailed safety requirements for a specific type of machine, e.g., industrial robots.

For robotic safety the following standards are of highest importance:

- ISO 12100 [54.62]: Type A standard listing general principles of machine safety, in particular the risk assessment process.
- ISO 13849 [54.65] and IEC 62061 [54.66, 67]: Type B safety standards governing the design of control systems with safety functions.
- ISO 13855 and ISO 13857 [54.68, 69]: Safety distances for separating and nonseparating safety equipment, e.g., fences, light curtains, and laser scanners.
- ISO 10218-1/-2 [54.5, 6]: Type C standards specifically covering safety of industrial robots and robot system integration.

For the setup of any robot installation, an iterative risk assessment process as defined by ISO 12100 has to be conducted. Its workflow as depicted in Fig. 54.23 starts with a functional and geometric design of the machine followed by the definition of the machine limits including spatial boundaries and boundaries regarding usage, e.g., typical tasks, operator qualification, and environmental conditions. The process continues with an identification of the tasks which are consecutively assessed on its risks. Methods for risk estimation are, for example, risk trees as presented in Fig. 54.24. Thereby, any individual hazard is rated on its risk level by quantitatively considering the potential injury severity, the exposition to the hazard, and the possibility to avoid it. The risk estimation leads to a decision if all hazards have been adequately addressed in the machine design. If this is not the case, the machine design is modified to reduce the specific risk and the risk assessment process is repeated.

According to ISO 12100, measures to reduce risks need to be carried out according to the following priorities:

- Risk reduction through inherently safe design
- Risk reduction through safeguards and protective devices
- Risk reduction through information for use (e.g., work instructions, instructions to wear protective equipment).

The central safety standard family for robot safety is ISO 10218. Part 1 addresses the safety requirements of the robots, complemented by part 2, which focuses on the robot system, i. e., the complete integrated machine performing a production task. The second revision of part 1 and the first revision of part 2 were released in 2011 and replace any former robot safety standard, such as EN 775. These standards for the first time define *human–robot-collaboration* as a specific form of a robotic application in an industrial setting and provide guidelines for setting up such collaborative robot systems.

### 54.5.2 Types and Requirements of Human–Robot Collaboration

ISO 10218-1:2011 contains specific requirements on human–robot-collaboration. It defines four types of collaborative operations in which a human can collaborate with a robot in automatic mode, as depicted in Fig. 54.25. All controller functions for safe human–

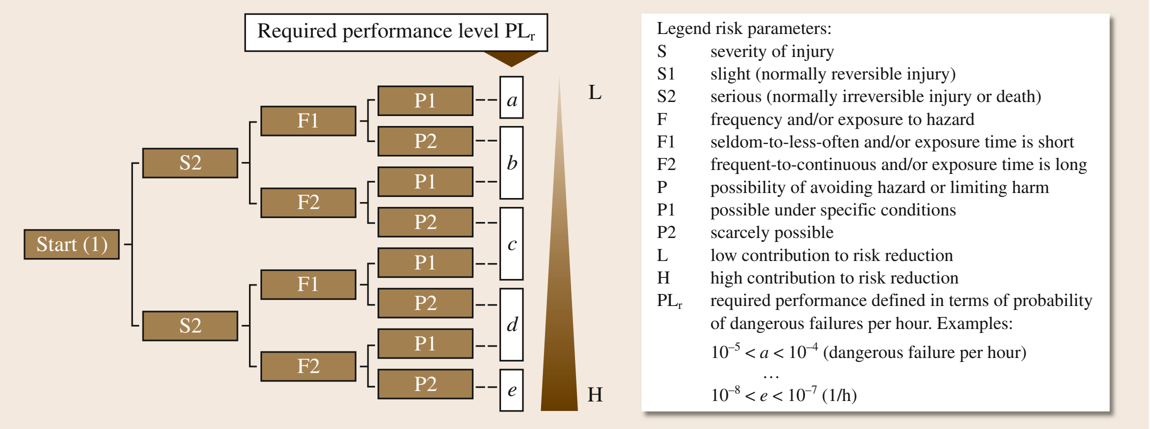



**Fig. 54.24** Risk graph for determining required $PL_r$ for safety function according to EN ISO 13849-1 (after [54.65])

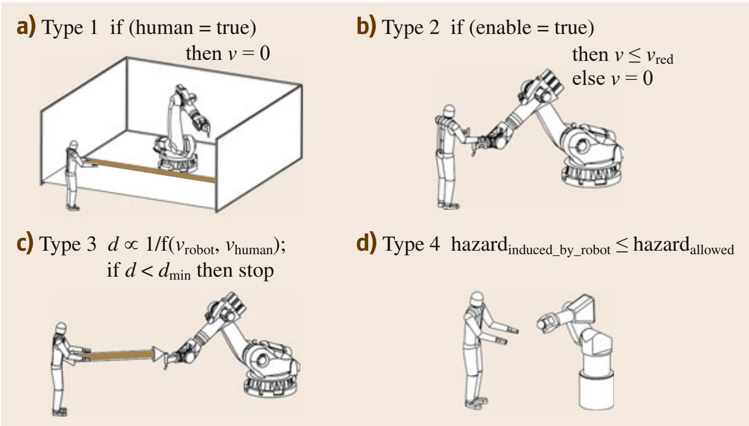



**Fig. 54.25a–d** Modes of human–robot collaboration according to ISO 10218. **(a)** Stop on access with automatic task resumption, **(b)** hand-guiding **(c)** separation and speed reduction, **(d)** monitoring and power and force limiting



robot collaboration have to comply with performance level $\mathrm{PL}_{\mathrm{d}}$ with category 3 structures (ISO 13849-1) or SIL 2 (IEC 62061) if the safety assessment does not yield a differing requirement:

- Safety-rated monitored stop (Fig. 54.25 Type 1): The robot is stopped upon access of the human to the collaborative workspace with the robot drives still in control. A so-called safety controller, now offered by most robot manufacturers, assures the standstill of the robot. The robot task can be automatically resumed as soon as the human operator has left the collaborative workspace. Human and robot share the same workspace, but the robot does not move while the human is present.
- Hand-guiding (Fig. 54.25 Type 2): This type of operation implies a direct physical interaction between human and robot with full control of the human over the robot movement. The human guides the robot through a direct input device (e.g., a handle) at or near the end-effector while activating a three-position enabling device (three-position hold-to run device). Thereby the position of the worker within the collaborative workspace is defined. A safety controller for delimiting the robot speed to a specified threshold is required. Hand-guiding in combination with graphic support through icons or 3-D simulation is in particular suitable for intuitive programming of industrial robots [54.70] during automatic mode of the robot.
- Speed and separation monitoring (Type 3): The relative speed between robot and human as well as their distance are actively monitored. If the human is present, the robot has to maintain a safe combination of speed and distance to the human to be able to stop any hazardous motion before a contact with the human may occur. Again safety controllers in conjunction with safe surveillance sensors (including safe sensor data processing) are needed to supervise speed and position of the robot for human

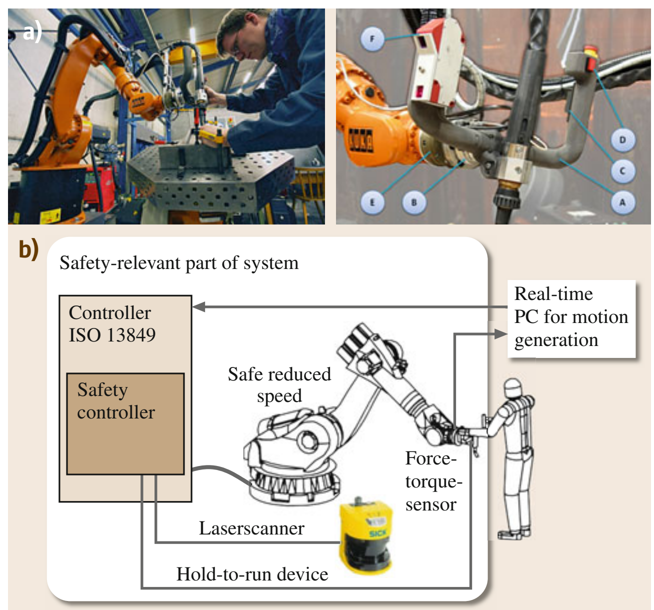



**Fig. 54.26a,b** Intuitive instruction of a welding robot by lead-through programming: **(a)** the worker guides the robot by a flange-mounted (B) handle (A) while pressing safety switches (three-position hold-to run button) (C) on either side. The force exerted on the handle is measured by a force–torque sensor (E) and translated into robot motion. The seam tracking sensor (F) simultaneously measures the workpiece contour for precisely localizing and extracting weld seams. The recorded robot motion is visualized and edited in a simulation environment before task execution. **(b)** The safety relevant part of the system contains a three-position hold-to-run device, a laser scanner for safety monitoring during automatic program execution and a speed monitoring function during hand-guiding is an operational feature which does not have to be realized with a specific safety integrity level (courtesy of Fraunhofer IPA)

motion capturing. Currently, relatively few sensors in industrial automation offer this capability with the required safety integrity level. However, such systems will be available in the future, then being able to feature novel safety strategies for dynamic distance control [54.71].

- Power and force limiting (Type 4): The mechanical hazard potential of the robot is sufficiently reduced to allow a direct, physical interaction of human and robot without an additional safety controller. This is achieved by appropriately limiting collision forces through the design of the robot system such that in the event of a contact between the human and a robot biomechanical tolerance limits are not exceeded.

The ISO 10218 standards in their latest revision as of 2011 explicitly demand that speeds, distances, powers and forces are to be sufficiently limited, but does not give precise threshold values for these limitations. These threshold values need to be determined through risk assessment for a specific application that is foreseen with the robot system. Currently, the technical committee ISO/TC184/SC2 *Robots and robotic devices* develops a technical specification ISO/TS 15066 (in *committee draft* (CD) status of December 2015). It has been drafted to offer more guidance on the risk assessment for collaborative robots and will be released in the near future [54.72]. The procedure to achieve safety outlined in TS 15066 is new for machine safety (not only for robot safety) and is expected to make its way into further standards, especially into the second revision of ISO 10218-2.

For the first time, ISO/TS 15066 introduces tolerance values for the physical strain of the human body. These strain thresholds include maximum forces and maximum pressures for different body parts that can be sustained without suffering from pain or even injury. The risk assessment process involves determining body regions with risk of contact depending on the specific applications and workflows. Based on this information, it can be proved experimentally or in simulation that the given severity threshold values are not exceeded due to limited mechanical or robot control parameters [54.73, 74]. The technical specification aims at transforming highly complex biomechanical injury thresholds into controllable robot performance limits. One of the principles used for dynamic collision analysis is based on the fundamental robot dynamics theory which enables the representation of the reflected inertia of a robot system at any point along the robot structure. Drop-test results from medical injury assessment have been made available for designing robot controllers [54.75].

Part 2 of the ISO 10218 standard provides guidelines regarding the aspects that need to be taken into account during robot workcell setup and system integration. It states the need not only to assess the robot itself, but to take into account the complete application, particularly the end-effector, process, environment, and typical work tasks. This is of special importance for collaborative human–robot operation. As the application always needs to be taken into account, it is not possible to design a *safe robot*, but only to provide robots that are equipped with safety features for setting up a safe collaborative operation.

### 54.5.3 Examples of Human–Robot Collaboration

In the following, two examples for human–robot collaboration are given.

#### Intuitive Instruction of a Welding Robot by Lead-Through Programming

Figure 54.26 presents a robot welding workcell which is designed for small batch size production. An intuitive teaching process implementing *hand-guiding* according to ISO 10218 (Type 2 in Fig. 54.25) significantly speeds up robot programming time. The presence of the human during the collaborative operation is monitored via a laser-scanner activating safety zones and safe speed reduction of the robot safety controller during the hand-guiding operation. The recorded motion can be executed automatically after switchover as soon as the human is out of the collaborative workspace.

#### Collaborative Assembly of Battery-Cases

In this case, sensitive tasks are carried out by the human, while strenuous tasks are executed automatically by a small payload robot (Fig. 54.27 for the cell design). Thereby, the safety concept comprises the requirements for *speed and separation monitoring* (Type 1 in Fig. 54.25) from ISO 10218 and involves two light curtains with a signal processing on a safety PLC and a robot with a safety controller. Switchover between the safety zones is activated upon detection of human presence in a collaborative workplace area. Each safety zone statically monitors the minimum possible distance to the robot's active workspace thereof deriving the maximum possible robot speed. Such a system reduces space requirements for a robot installation while enhancing the flexibility of the assembly process due to the collaborative nature of this assembly process.

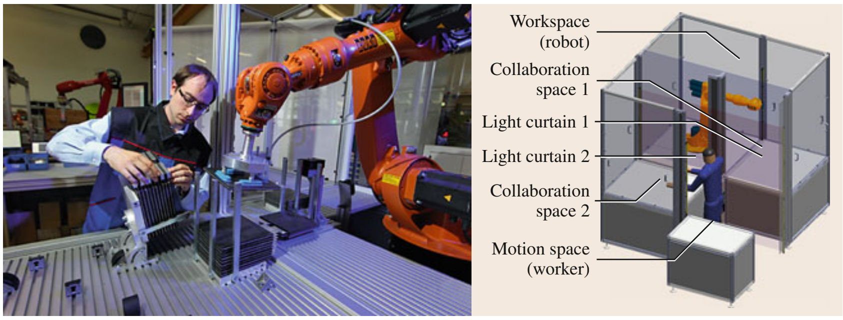



**Fig. 54.27** The robot's workcell is divided into three segments by two light curtains which trigger in which area the robot works at lower velocity when collaborating with the worker in the collaboration spaces

## 54.6 Task Descriptions – Teaching and Programming

Although desirable, a robot cannot be instructed in the same way that one would instruct a skilled human worker how to carry out a task. With skilled human workers knowing the applications, devices, processes, and the general requirements on the product to be manufactured, we would only need to summarize what needs to be done. Humans have, with or without awareness thereof, extensive knowledge about motions, physical effects, cause–effect relationships, learned procedures, etc., and they maintain such knowledge for their own good. A human is skilled because physical capabilities are combined with reuse of that learned/maintained knowledge, and the purpose of the manufacturing operation can be explained and understood.

Robots are not capable of performing such knowledge-based behaviors in a productive manner. Instead, instructions have to be quite explicit and motion oriented, and even motion planning is rarely used since it is difficult to encode much of the required background knowledge. We could aim for programming principles that resemble instructing a (totally) unskilled worker, telling precisely how every aspect of the task is to be performed. Such an explicit way of instructing the robot then should be human friendly, but the performance requirements motivated by productivity needs imply that methods for defining the task need to reflect machine/robot properties considering product data and production processes. Existing approaches include:

- Manually guiding the robot to the positions of interest, or even along the desired paths or trajectories if human accuracy is enough (Fig. 54.26).
- Having simple ways to make use of CAD data whenever available (Fig. 54.13).
- Using different complementary modalities (paths of communication between the human and the robot), such as pointing devices and 3-D graphics.
- Choreographing the task movements, for instance loops and conditions, without requiring extensive programming competencies.
- Means of describing acceptable variation, e.g., acceptable deviations from the nominal path or POI.
- Specification of how external sensing should be used for path adaptations or for handling unknown variations.

Most of these items still mean that the robot is rather dumb, and the knowledge of solutions is maintained among human experts. Consequently, robots are time-consuming and hard to instruct for productive industrial tasks, and the created solutions are not reusable since the knowledge is not explicitly represented in a way that is useful for the robot. To embody knowledge in the robot system is the way to make task definition more efficient, which can be better understood if we review the origins and the progress so far. Initially, mainly during the 1970s and 1980s, there were some painting robots that could be programmed by manual guidance. This was possible due to the following abilities:

- Applications such as painting permitted the use of a lightweight arm, including the end-effector, possible balanced with respect to gravity if needed.
- The accuracy requirements were (compared to today) modest so it was possible to use back-drivable drive trains and actuators, and the definition of the motions could then be done manually by the operator moving the end-effector along the path. The recorded poses, including the timing/speed information, defined the programmed motion.
- Since no inverse kinematics was needed during programming or real-time operation, it was not a problem from a computing point of view to use arm kinematics without singularities in the workspace.

- The requirements of optimality of painting motions were also modest, compared to recent years when environmental conditions (for nature and workers) called for minimal use of paint.

It is often referred to as an inevitable problem that there are singularities to be handled within the workspace. However, to simplify the kinematics and its inverse from a software point of view (e.g., during the 1980s considering the power of the microprocessors and algorithms available at that time), robots were actually designed to have simple (to compute) inverse kinematics. For instance, the wrist orientation was decoupled from the translation by the arm by using wrist axes that intersected with the arm axes. The resulting singularities within the workspace could be managed by restrictions on wrist orientations, but an unfortunate implication was that robot arms were no longer back-drivable (close to the singularities) when designs were (due to engineering and repeatability requirements) adopted to standard industrial controllers. Then with the development of microprocessor-based industrial controllers and the definition of motions based on jogging (manual moves by for instance using a joystick) and CAD data, the means of robot programming became closer to computer programming (extended with motion primitives).

Robot programming languages and environments have traditionally been separated into online programming (using the actual robot in situ) and OLP (using software tools without occupying the robot). With the increasing power of OLP tools, their emerging ability to connect to the physical robot, and the increasing level of software functions that are embedded into the robot control system, online programming is now unusual, except to verify and manually adjust programs generated offline. Of course, it is economically important to minimize downtime for robot programming, and advanced sensor-based applications may be too hard to develop without access to the true dynamics of the physical workcell for fine-tuning. Nevertheless, robot languages and software tools must provide for both methods of programming.

Even though robot languages from different manufacturers look similar, there are semantic differences that have to do with both the meaning when programs are running (the robot performing its operations) and the way the robot is instructed. The need to ensure that existing robot programs can operate with replacement robots and controllers, and also to make use of existing knowledge in robot programmers and incorporated into OLP software, requires manufacturers to continue supporting their original proprietary languages. Features such as backward execution (at least of motion statements) and interactive editing during interpretation by the robot (in combination with restrictions to make the programming simpler) also make robot languages different from conventional computer programming languages.

During the last decade and in current developments, the trend has been to focus on the tool (robot end-effector) and on the process knowledge needed for the manufacturing process, and to let the operator express the robot task in such terms. This development results in a need for an increased level of abstraction to simplify the programming, reflecting the fact that the so-called robot programmer knows the production process very well, but has quite limited programming skills. To understand why such a high level of abstraction has not come into widespread usage, we may compare this with the early days of industrial robotics when there was no kinematics software built into the controllers, and hence the robots were programmed via joint-space motions. (Kinematics here deals with the relation between robot motors/joints and the end-effector motions.)

Built-in inverse kinematics permits tool motion to be specified in Cartesian coordinates, which is clearly a great simplification in many applications. That is, the robot user could focus more on the work to be carried out by the robot and less on the robot itself. However, robot properties such as joint limits cannot be neglected. Until the beginning of the 1990s, robots did not follow programmed trajectories very well at high speeds or accelerations, and full accuracy could only be achieved at low speed. Modern robot systems with high-performance model-based motion control perform their tasks with much greater accuracy at high speed due to model-based control features (Chap. 6).

There are increasing opportunities to raise the level of abstraction to simplify the use of robots even more by encoding more knowledge about the robot, tools, the process, and workcells into control systems. This is a gradual development with possibilities depending on the application; when knowledge is not encoded the robot programmer needs to be more aware of robot properties and how to handle them. Reasons for not including such knowledge typically lie in the fact that it is simply too difficult to do so or because the robot programmer does neither have the competence nor the tools nor reasons to do so. With reference to Table 54.4, the following example explains that type of trade-off and what it means in terms of centricity.

A machine part consisting of steel plates and pipes is to be manufactured by means of welding, and there is robot and process equipment available for production of this (and other) types of workpieces:

- A product-centric system would generate configurations and robot programs, and instruct the oper-

**Table 54.4** Robot-related centricity, ranging from a high-level view of the work to be accomplished in terms of the product to manufacture, to a low-level robot motion view that, in practice, constraints what manufacturing operations can be performed

| User application (↑) | Robot constraint (↓) | | |
|---|---|---|---|
| | | Product | Description of the final shape and assemblies of the workpieces, in terms of that product; the robot system plans the operations |
| | | Process | Based on known sequences of specific manufacturing operations, each of these is specified in terms of their processes parameters |
| | | Tool | The motion of the robot-held tool is specified in terms of programs or manual guidance; the user knows the process it accomplishes |
| | | Arm | The robot arm and its end-plate for tool mounting is programmed how to move in Cartesian space |
| | | Joint | For each specified location, the joint angles are specified, so straight-line motions are difficult; the robot provides coordinated servo control |

ator for whatever manual assistance that might be needed (e.g. clamping and fixturing). The system would determine the welding data such as the type of welding and how many passes for each seam.

- A process-centric system would instead accept input in terms of the welding parameters to use, including the order of the welds. The system would select input signals to the process equipment (such as what output voltage the robot controller should set such that the welding will be done with a certain current), and robot programs specifying the motions of the welding torch would be generated.
- A tool-centric way of programming would require the operator to set up the process manually by configuring the equipment in terms of their native settings, and appropriate tool data would be configured such that the robot controller can accomplish the programmed motions, which are specified by giving coordinates and motions data referring to the end-effector.
- An arm-centric system is similar to the tool-centric approach in that Cartesian and straight-line motions can be (and need to be) explicitly specified/programmed by the programmer, but extra work is needed since the robot does not support a general tool frame.
- A joint-centric style would be needed if one of the very early robot systems is being used, requiring a straight line to be programmed by lots of joint-space poses close to each other.

Thus, the question is, are you programming robot joint servos to make the robot provide a service for you, are you programming/commanding an arm how to move a tool, is it the tool that is made programmable by means of the robot, is it manufacturing services that are ordered by specifying the desired process parameters, or is it an intelligent system that simply can produce your product? Still today in industry, there are reasons to think and program considering all five levels, although the tool-centric alternative is kind of a baseline. Hence tool-data needs to be maintained in the robot program, as in the next example below.

As a final goal related to the product-centric view, so-called task-level programming is desirable. This has been a goal since the 1980s, and implicitly also since the very beginning of robotics. It would mean that the user simply tells the robot what should be done in common terms and the robot would know how to do it, but it would require extensive knowledge about the environment and so-called *machine intelligence*. The need for extensive modeling of the environment of the robot is well known. Sensing of the environment is costly, but in an industrial environment it should only be needed occasionally. Modeling has to encompass full component dynamics and the limitations of the manufacturing process. With these difficulties, task-level programming is not yet achievable in practice. If we limit our domain, however, such as to machining where motion planning is already used for so-called machine tools, generating the robot programs from product descriptions is feasible since the process (with tolerances due to robot compliance) is modeled.

As indicated in the application examples, it is now common practice to generate robot programs from geometric data in CAD files. That is, the CAD application could be the environment used for specifying how the robot should perform the required operations on the specified parts, which is the CAM part of CAD/CAM (Computer Aided Manufacturing, often in term of a back-end to the CAD system, and hence usually not explicitly referred to). CAM is not quite task-level programming since human operators do the overall planning. CAD software packages are powerful 3-D tools that are very common among manufacturing companies. Consequently, using these tools for robot programming is desirable since the operator may start the OLP of the necessary manufacturing operations using the 3-D model of the product, even before selecting a robot.

There is an ongoing competition along the so-called vertical integration from product data down to machine operations, including at least the following three approaches:

- CAD providers increasingly offer advanced robot-aware functions, thereby expanding their field of service and expertise from a higher (CAD) to a lower (robot) level. An example of a robot-aware CAM system is depicted in Fig. 54.22.
- Robot providers have developed programming tools with CAD data import and various motions planning modules, and thus, expanded their offerings to higher levels. One such software is shown in Fig. 54.29.
- A further development is to include a CNC kernel into the robot controller, as suggested in Fig. 54.21, making the robot behave as a CNC machine.

These approaches as such are more complementary than competing, with the competing stakeholders presently exploring what benefits can be gained for customers and business in each type of application.

*Listing 54.2 Example pick-and-place operation in the ABB robot programming language Rapid*

```
PROC PickInPallet()
 MoveJ Offs(pPickInPallet,-500,0,500),v2000,z100,tGripper\WObj:=wobjPalletStatic;
 MoveL RelTool(pPickInPallet,0,0,-100),v2000,z100,tGripper\WObj:=wobjPallet;
 MoveL RelTool(pPickInPallet,0,0,0),v100,fine,tGripper\WObj:=wobjPallet;
 Grip;
 MoveL RelTool(pPickInPallet,0,0,-100),v500,z10,tGripper\WObj:=wobjPallet;
 MoveJDO Offs(pPickInPallet,-500,0,500),v2000,z100,tGripper\WObj:=wobjPalletStatic,
 doNewObject,1;
ENDPROC

PROC PlaceAtOutFeeder()
 MoveJ Offs(pDropOutFeeder,0,0,200),v2000,z10,tGripper\WObj:=wobjOutFeeder;
 WaitDI diOutFeederReady,1;
 MoveL Offs(pDropOutFeeder,0,0,0),v200,fine,tGripper\WObj:=wobjOutFeeder;
 Release;
 MoveLDO Offs(pDropOutFeeder,0,0,200),v500,z10,
 tGripper\WObj:=wobjOutFeeder,doStartOutFeeder,1;
 MoveL pFromOutFeeder,v2000,z10,tGripper\WObj:=wobjMachine;
ENDPROC

PROC Grip()
 SetDO doGrip,1;
 WaitDI diGripped,1;
ENDPROC
PROC Release()
 SetDO doGrip,0;
 WaitDI diGripped,0;
ENDPROC
```

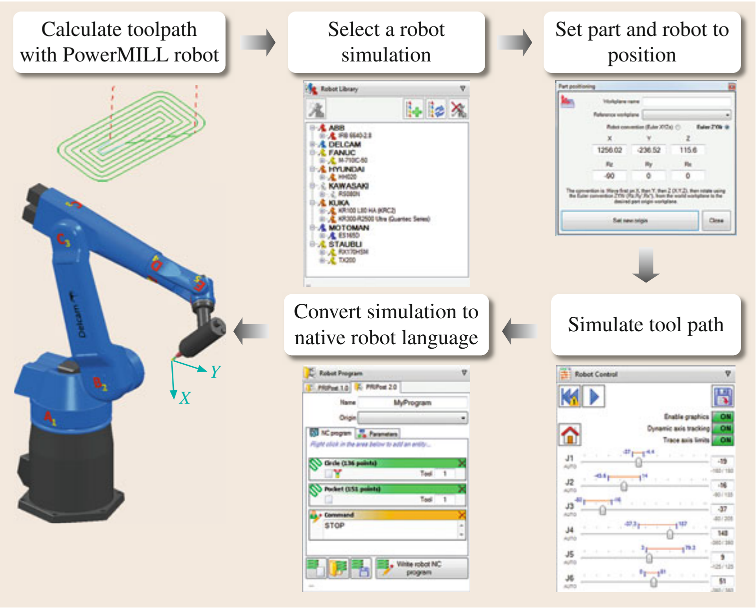



**Fig. 54.28** Workflow according to the PowerMill software from Delcam with its robot module. Note that step 4 includes also simulation of robot properties such as closeness to singularities

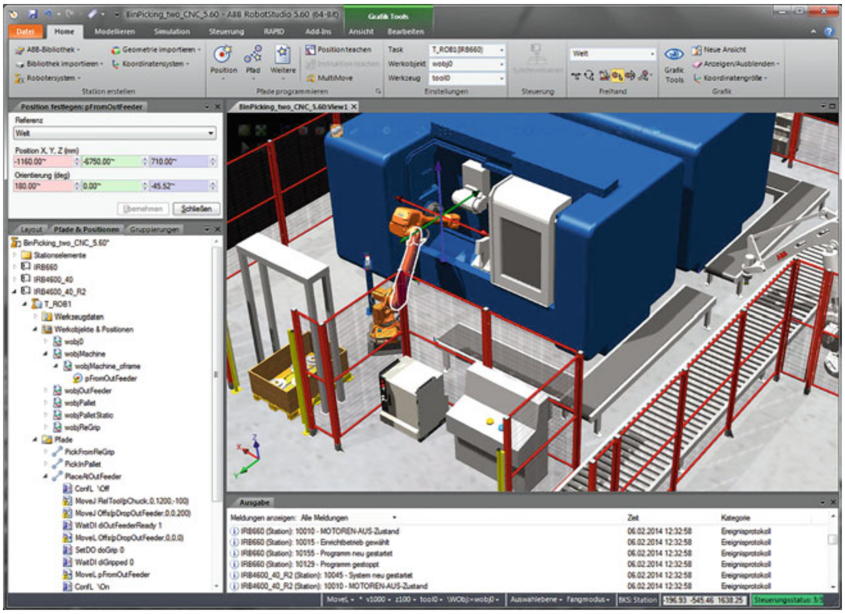


**Fig. 54.29** Robot programming environment, with a virtual controller (console to the lower right) for each robot including the model-based embedded control software compiled for the personal computer (PC). To the left there are the instances of all objects depicted as a tree structure, ranging from complete machines down to the bits of the I/O signals

At step 5 in Fig. 54.28, the motion specification is generated and expressed in the native language of the robot. On the other hand, the tool path could be expressed in machine-independent, standardized G-code. A further step would be to integrate a machine tool interface and control into the robot controller which is depicted in Fig. 54.22. What approach to take is an area of current research and development (R&D) and business decisions, but most robots are programmed in their own language only, since the needed system functions as a consequence of Table 54.4.

To exemplify task definitions based on a native robot language, consider the application depicted in Fig. 54.29, which includes bin-picking, handling, machine tending, conveyer tracking, and palletizing. Listing 54.2 lists a sample program with some of the functions from that application programmed in ABB's proprietary robot language Rapid. The program being tool-centric shows in the tool argument of the Move statements, the *tGripper* that is a reference to the tool data including the tool frame. At the same time, it is arm-centric in the way that the execution of motions is performed per arm (with support for synchronized execution for multiarm robots), as expressed by the *MoveL* statements. Correspondingly, the *MoveJ* performs a joint-space motion.

Note that `Grip` and `Release` are in separate procedures where execution waits until a hardware acknowledged signal is obtained. For `Move` statements `L` denote `Linear`, `J` means `Jointspace`, functions `Offs` and `RelTool` compute target poses based on other determined poses (i. e., frames). The $v$-constants are velocities in mm $s^{-1}$, and $z$-constants (or fine) specify how close to the target pose the motion should be. `DI` refers to digital input and `DO` to digital output. The `WObj` is the work object, i. e., the specification of the base frame of the motions.

When programming motion behaviors like a search strategy, which could be considered being a motion primitive from an application point of view, use of the end-user language of that robot might not be the best option. Instead a computer programming language can be more appropriate as shown in Listing 54.1. In the ABB case, the strategies are implemented as part of an application package as for assembly, and the code is internally written in C (cannot be viewed or changed by the user, but has optimized real-time performance) while the functionality is exposed to the robot user via special statements like FCRefSpiral [54.51] and more to accomplish the principles of Table 54.2. A variety of solutions are available for different robot brands [54.76].

Considering a complete setup with multiple robots, either with a few robots and some peripherals like in Fig. 54.9 or a complete manufacturing line as in Fig. 54.12, there are several robot programs that need to be programmed to work together and hence there is a coordination and complexity problem. A useful approach is to use a service-oriented approach, considering robots as servers providing services according to a set of programs that are exposed on the factory or production-cell network which means programming is done by:

- Building services (using several technologies) which are available, discoverable or not, remotely accessible by the application programmer
- Building applications that coordinate and use those services.

For such system to be programmable, however, the equipment (typically from different suppliers) has to be configured and interconnected according to the sometimes overlooked art of robot system integration.

## 54.7 System Integration

It is interesting to note that connecting different workcell devices with each other, and integrating them into a working system, is hardly mentioned in the robotics literature. Nevertheless, in actual nontrivial installations, this part typically represents, apart from the cost for peripheral device, roughly half of the overall installation costs. The automation scenario includes all the problems of integrating computers and their peripherals, plus additional issues that have to do with the variety of (electrically and mechanically incompatible) devices and their interaction with the physical environment (including their inaccuracies, tolerances, and unmodeled physical effects such as backlash and friction). The number of variations is enormous so it is often not possible to create reusable solutions. In total, this results in a need for extensive engineering to put a robot to work (Table 54.5). This engineering is what we call system integration.

Carrying out system integration according to current practice is not a scientific problem as such (although how to improve the situation is), but the obstacles it comprises form a barrier to applying advanced sensor-based control for improved flexibility, as

**Table 54.5** Stages of system integration, typically carried out in the order listed

| | |
|---|---|
| Physical | Selecting equipment based on dimensioning for mechanical size, load, and stress |
| | Mechanical interfacing (locations, adapter plates, etc.) |
| | Electrical power supply (voltages and currents for robots, effectors, feeders, etc.) |
| | Connections for analog signals (shielding, scaling, currents, binary levels, etc.) |
| | Safety design and risk assessment |
| Communication | Interconnections for single-bit digital I/O |
| | Byte-wise data communication, including latencies and bit rates |
| | Transfer of byte sequences |
| Configuration | Configuration of messages between interacting devices |
| | Establishment of services |
| | Tuning for performance and resource utilization |
| Application | Definition of application-level functions/services |
| Task | Application programming, using the application-level services |

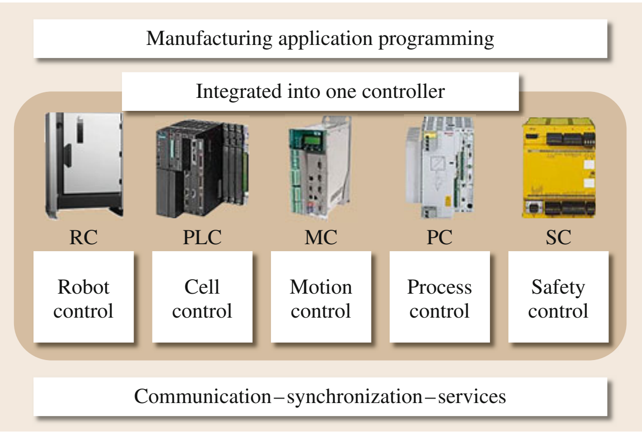



**Fig. 54.30** Manufacturing cell with a multicontroller architecture with dedicated controllers (robot motions, production sequences, positioning systems, processes and worker safety). In modern robot controllers, such as the KUKA KR C4, these controllers are replaced by software tasks of a single multifunctional robot controller, thus reducing investment costs on controllers and communication, simplifying engineering, programming and diagnosis, and enhancing process quality and shortening cycle time ►

is needed in small series production. In particular, in future types of applications using external sensing and high-performance feedback control within the workcell, system integration will be an even bigger problem since it includes tuning of the feedback too.

For large series production (e.g. in the automotive industry), the engineering cost of system integration is less of a problem since its cost per manufactured part is small. On the other hand, the trend toward smaller series

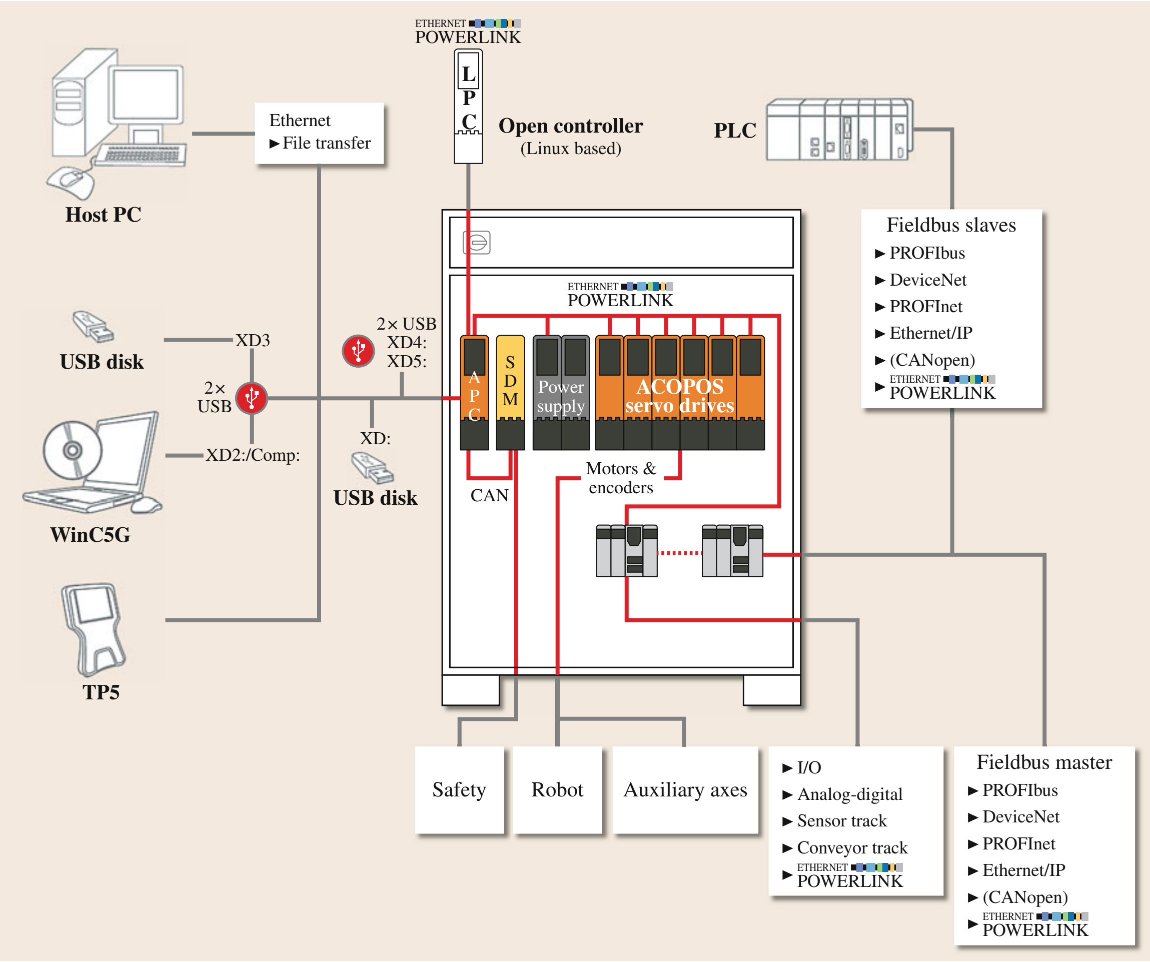



**Fig. 54.31** Control modules and interfaces in a typical (but open) robot controller. Several of the named network technologies are based on Ethernet, ranging from normal LAN to different real-time protocols (courtesy of COMAU)

of customized products, or products with many variants that are not kept in stock, calls for high flexibility and short changeover times. Flexibility in this context refers to variable product variants, batch sizes, and process parameters. In particular, this is a problem in small and medium-sized productions, but the trend is similar for larger enterprises as flexibility requirements are continuously on the rise. One may think that simply by using standards for input/output (I/O) and well-defined interfaces, integration should be just a matter of connecting things together and run the system.

Of course, the object-oriented software solution would be to have a gripper class with operations grasp and release. That would be appropriate for a robot simulator in pure software, but for integration of real systems such encapsulation of data (abstract data types) would introduce practical difficulties because:

- Values are explicit and accomplished by external (in this case) hardware, and for testing and debugging we need to access and measure them.
- Online operator interfaces permit direct manipulation of values, including reading of output values. The use of functions of object methods (so-called set:ers and get:ers) would only complicate the picture; maintaining consistency with the external devices would be no simpler.

Already in small installations, the complexity of system integration becomes apparent. So-called vertical integration means integration of low-level devices with a high-level factory control system, whereas integration of peripherals on (for instance) workcell level is called horizontal integration (Fig. 54.30).

Lack of self-descriptive and self-contained data descriptions that are also useful at the real-time level further increases the integration effort since data interfaces/conversions typically have to be manually written. In some cases, as illustrated in Fig. 54.31, there are powerful software tools available for the integration of the robot user level and the engineering level [54.77]. The fully (on all levels) integrated and nonproprietary system such as the recently discussed Industry4.0 or industrial Internet initiatives is still is a challenge, particularly for small and medium-sized productions [54.24].

## 54.8 Outlook and Long-Term Challenges

The widespread use of industrial robots in standard, large-scale production such as the automotive industry, where robots perform repetitive tasks in well-known environments, resulted in the common opinion that industrial robotics is a solved problem. This opinion was underpinned by the robot systems' impressive automatic performance, based on advanced semiautomatic programming and resulting in an unbeatable product quality when compared to manual labor. However, large-scale production comprises only a minor part of the work needed on an industrial scale in any wealthy society, especially considering the number of companies and the variety of applications and processes that could and should be automated for productivity, health and sustainability reasons.

Global prosperity and wealth requires resource-efficient and human-assistive robots. The challenges today are to recognize and overcome the barriers that are currently preventing robots from being more widely used, especially in small and medium-sized manufacturing.

Taking a closer look at the scientific and technological barriers, we find the following challenges:

- *Human-friendly task specification*, including intuitive ways of expressing permitted/normal/expected variations and errors. There are numerous upcoming and promising techniques for user-friendly human–robot interaction (such as speech, gestures, manual guidance, and so on), but the focus is still on specification of the nominal task rather than the complete task which is capable of handling foreseen deviations from the nominal case (Part G, Robots and Humans). Taking care of these expected deviations can account for up to 80% of the total programming time.
- *Intuitive human–robot interaction* The foreseen variations, and the unforeseen variations experienced during robot work, are difficult to manage. When instructing a human, he/she has an extensive and typically implicit knowledge about the work and the involved processes. To teach a robot, it is an issue both how to realize what the robot does not know, and how to convey the missing information efficiently. Furthermore, a person should feel comfortable and familiar with the robot's functionality and operation during all operational modes including maintenance and error recovery. Research and development toward industrial robotics usability and ergonomics for increasing acceptance and operator efficiency has been of surprisingly low activity in the past.
- *Efficient mobile manipulation*. Successful implementations and systems are available for both mobility and for manipulation, but accomplished

in different systems and using different types of (typically incompatible) platforms. A first step would be to accomplish mobile manipulation at all, combining all degrees of freedom of both the mobile base and arm(s), intelligently exploiting system redundancies to achieve a given task. A second step providing truly dependable autonomous navigation (with adaptive, but predictable understanding of constraints, such as frequent environmental changes and typical shop-floor dynamics, and appropriate sensor-based reactions) dexterous manipulation with multifingered grippers, and robust force/torque interaction with environments (that have unknown stiffness). As a concluding step, all this needs to be done with decent performance using reasonably priced hardware, and with human–robot interfaces according to the previous items.

- *Low-cost components including low-cost actuation*. Actuation of high-performance robots represent about two-third of the overall robot cost, and improved modularity often results in a higher total hardware cost (due to less opportunities for mechatronic optimization). On the other hand, cost-optimized (with respect to certain applications) systems result in more-specialized components and smaller volumes, with higher costs for small series production of those components. Since future robotics and automation solutions might provide the needed cost efficiency for small series customized components, we can interpret this as a boot-strapping problem, involving both technical and business aspects. The starting point is probably new core components that can fit into many types of systems and applications, calling for more mechatronics research and synergies with other products.
- *Composition of subsystems*. In most successful fields of engineering, the principle of superposition holds, meaning that problems can be divided into subproblems and that the solutions can then be superimposed (added/combined) onto each other such that the total solution comprises a solution to the overall problem. These principles are of key importance in physics and mathematics, and within engineering some examples are solid-state mechanics, thermal dynamics, civil engineering, and electronics. However, there is no such thing for software, and therefore not for mechatronics (which includes software) or robotics (programmable mechatronics) either. Thus, composition of un-encapsulated subsystems is costly in terms of engineering effort.
  Even worse, the same applies to encapsulated software modules and subsystems. For efficiency, system interconnections should go directly to known (and hopefully standardized) interfaces, to avoid the indirections and extra load (weight, maintenance, etc.) of intermediate adapters (applying to both mechanics for end-effector mounting and to software). Interfaces can be agreed upon, but the development of new versions typically maintains backward compatibility (newer devices can be connected to old controller), while including the reuse of devices calls for mechanisms for forward compatibility (automatic upgrade based on meta information of new interfaces) to cover the case that a device is connected to a robot that is not equipped with all the legacy or vendor-specific code.
- *Embodiment of engineering and research results*. Use or deployment of new technical solutions today still starts from scratch, including analysis, understanding, implementation, testing, and so on. This is the same as for many other technical areas, but the exceptional wide variety of technologies involved with robotics and the need for flexibility and upgrading makes it especially important in this field. Embodiment into components is one approach, but knowledge can be applicable to engineering, deployment, and operation, so the representation and the principle of usage are two important issues. Improved methods are less useful if they are overly domain specific or if engineering is experienced to be significantly more complicated. Software is imperative, as well as platform and context dependent, while know-how is more declarative and symbolic. Thus, there is still a long way to go for efficient robotics engineering and reuse of know-how.
- *Open dependable systems*. Systems need to be open to permit extensions by third parties, since there is no way for system providers to foresee all upcoming needs in a variety of new application areas. On the other hand, systems need to be closed such that the correctness of certain functions can be ensured. Extensive modularization in terms of hardware and supervisory software make systems more expensive and less flexible (contrary to the needs of openness). Highly restrictive frameworks and means of programming will not be accepted for widespread use within short-time-to-market development. Most software modules do not come with formal specification, and there is less understanding of such needs. Thus, systems engineering is a key problem.
- *Sustainable manufacturing*. Manufacturing is about transformation of resources into products, and productivity (low cost and high performance) is a must. One aspect of sustainability deals with energy-efficiency of the devices that are used for production. Saving energy can constrain the path planning of an

individual and a whole fleet of robots, and even shift the operation of energy-consuming tasks in periods of the day where energy cost are low. A second aspect of sustainability deals with recycling of scarce resources in terms of materials and noble earths. In most cases, this can be achieved by crushing the product and sorting the materials, but in some cases disassembly and automatic sorting of specific parts are needed. There is, therefore, a need for robots in recycling and de-manufacturing. Based on future solutions to the above items, this is then a robot application challenge.

An overall issue is how both industry and academia can combine their efforts such that sound business can be combined with scientific research so that future developments overcome the barriers that are formed by the above challenges.

## Video-References

VIDEO 260 SMErobotics project video
available from http://handbookofrobotics.org/view-chapter/54/videodetails/260

VIDEO 261 SMErobot video coffee break (English)
available from http://handbookofrobotics.org/view-chapter/54/videodetails/261

VIDEO 262 SMErobot final project video
available from http://handbookofrobotics.org/view-chapter/54/videodetails/262

VIDEO 265 SMErobot – New parallel kinematic with unique concepts for demanding handling and process applications
available from http://handbookofrobotics.org/view-chapter/54/videodetails/265

VIDEO 266 SMErobot D4 *The woodworking assistant*
available from http://handbookofrobotics.org/view-chapter/54/videodetails/266

VIDEO 380 SMErobotics demonstrator D1 assembly with dual-arm industrial manipulators
available from http://handbookofrobotics.org/view-chapter/54/videodetails/380

VIDEO 381 SMErobotics demonstrator D2 human–robot cooperation in wooden house production
available from http://handbookofrobotics.org/view-chapter/54/videodetails/381

VIDEO 382 SMErobotics demonstrator D3 assembly with sensitive compliant robot arms
available from http://handbookofrobotics.org/view-chapter/54/videodetails/382

VIDEO 383 SMErobotics demonstrator D4 welding robot assistant
available from http://handbookofrobotics.org/view-chapter/54/videodetails/383

## References


54.1 The International Federation of Robotics (IFR): *World Robotics 2015. Statistics, Market Analysis, Forecasts, Case Studies and Profitability of Robot Investments* (International Federation of Robotics, Frankfurt 2015)

54.2 D. Bourne: My Boss the Robot, Scientific American **308**(5), 37–41 (2013)

54.3 M.P. Groover: *Automation, Production Systems, and Computer-Integrated Manufacturing*, 3rd edn. (Prentice-Hall, Upper Saddle River 2007)

54.4 International Organization for Standardization: Standards catalogue, 25.040.30: Industrial robots. Manipulators, http://www.iso.org/iso/home/store/catalogue_ics/catalogue_ics_browse.htm?ICS1=25&ICS2=40&ICS3=30 (2014)

54.5 International Organization for Standardization: ISO 10218-1:2011. Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots, http://www.iso.org/iso/home/store/catalogue_detail.html?csnumber=51330 (2011)

54.6 International Organization for Standardization: ISO 10218-2:2011. Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration, http://www.iso.org/iso/home/store/catalogue_detail.html?csnumber=41571 (2011)

54.7 L. Westerlund: *The Extended Arm of Man. A History of the Industrial Robot* (Informationsförlaget, Stockholm 2000)

54.8 D. Hunt: *Industrial Robotics Handbook* (Industrial, New York 1983)

54.9 V.D. Scheinman: Design of a Computer Controlled Manipulator, Ph.D. Thesis (Stanford Univ., Stanford 1969)

54.10 International Federation of Robotics (IFR): History of Industrial Robots. From the first installation until today (International Federation of Robotics, Frankfurt 2012), http://www.ifr.org/uploads/media/History_of_Industrial_Robots_online_brochure_by_IFR_2012.pdf

54.11 H. Makino: Assembly robot, US4341502 A (1982)

54.12 G. Boothroyd, P. Dewhurst, W. Knight: *Product Design for Manufacture and Assembly*, 3rd edn. (CRC, Boca Raton 2011)

54.13 R. Curran, G. Mayer: The architecture of the AdeptOne® direct-drive robot, Am. Control Conf., Boston (1985)
54.14 X.-J. Liu, J. Wang: Performance evaluation of parallel mechanisms. In: *Parallel Kinematics*, Springer Tracts in Mechanical Engineering, (Springer, Berlin, Heidelberg 2014) pp. 185–238
54.15 S. Haddadin, S. Parusel, R. Belder, A. Albu-Schaeffer: It is (almost) all about human safety: A novel paradigm for robot design control and planning, Lect. Notes Comput. Sci. **8153**, 202–215 (2013)
54.16 R. Bloss: Innovations like two arms, cheaper prices, easier programming, autonomous and collaborative operation are driving automation deployment in manufacturing and elsewhere, Assem. Autom. **33**(4), 312–316 (2013)
54.17 S. Kock, T. Vitor, M. Björn, H. Jerregard, M. Källmann, I. Lundberg: Robot concept for scalable, flexible assembly automation, Proc. IEEE Int. Symp. Assem. Manuf. (ISAM), Tampere (2011)
54.18 S. Bogh, M. Hvilshoj, M. Kristiansen, O. Madsen: Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (AIMM), Int. J. Adv. Manuf. Technol. **61**(5-8), 713–726 (2012)
54.19 C.Y.K. Smith, L. Nalpantidis, X. Gratal, P. Qi, D. Dimarogonas, D. Kragic: Dual arm manipulation – a survey, Robot. and Auton. Syst. **60**(10), 1340–1353 (2012)
54.20 P. Gorle, A. Clive: *Positive Impact of Industrial Impacts on Employment* (International Federation of Robotics, Frankfurt 2013)
54.21 J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, A. Marrs: Disruptive technologies: Advances that will transform life, business, and the global economy, http://www.mckinsey.com/insights/business_technology/disruptive_technologies (2013)
54.22 E. Westkämper: *Towards the Re-Industrialization of Europe: A Concept for Manufacturing for 2030* (Springer, Berlin, Heidelberg, 2014)
54.23 M. Hedelind, M. Jackson: How to improve the use of industrial robots in lean manufacturing systems, J. Manuf. Technol. Manag. **22**(7), 891–905 (2011)
54.24 H. Kagermann, W. Wahlster, J. Helbig: *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0*. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf (Forschungsunion/acatech, Frankfurt 2013)
54.25 R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe: The KUKA-DLR lightweight robot arm – a new reference platform for robotics research and manufacturing, 41st Int. Symp. Robot. 6th Ger. Conf. Robot. (ROBOTIK), Munich (2010)
54.26 M. Hedelind, S. Kock: Requirements on flexible robot systems for small parts assembly: A case study, Proc. 2011 IEEE Int. Symp. Assem. Manuf. (ISAM), Tampere (2011) pp. 1–7
54.27 S. Kinkel, E. Kirner, H. Armbruster, A. Jäger: Relevance and innovation of production-related services in manufacturing industry, Int. J. Technol. Manag. **55**(3/4), 263–273 (2011)
54.28 International Organization for Standardization: ISO 8373. Robots and robotic devices – Vocabulary, http://www.iso.org/iso/home/store/catalogue_detail.html?csnumber=53830 (2012)
54.29 R. Bloss: Review of manufacturing cells as they achieve high levels of autonomy and flexibility, Assem. Autom. **33**(2), 112–116 (2013)
54.30 A. Kahn: *The Encyclopedia of Work-related Illnesses, Injuries, and Health Issues* (Facts On File, New York 2004)
54.31 M. Wilson: Developments in robot applications for food manufacturing, Ind. Robot Int. J. **37**(6), 498–502 (2010)
54.32 R. Moreno Masey, J. Gray, T. Dodd, D. Caldwell: Guidelines for the design of low-cost robots for the food industry, Ind. Robot Int. J. **37**(6), 509–517 (2010)
54.33 K. Mathia: *Principles and Applications in Cleanroom Automation* (Cambridge Univ. Press, Cambridge 2010)
54.34 D. Rossi, E. Bertolini, M. Fenaroli, F. Marciano, M. Alberti: A multi-criteria ergonomic and performance methodology for evaluating alternatives in "manuable" material handling, Int. J. Ind. Ergon. **43**(4), 314–327 (2013)
54.35 D. Caldwell: *Robotics and Automation in the Food Industry: Current and Future Technologies* (Woodhead, Cambridge 2012)
54.36 M. Palzkill, A. Verl: Object pose detection in industrial environment, Proc. 7th Ger. Conf. Robot. (ROBOTIK), Munich (2012)
54.37 R. Mattone, M. Divona, A. Wolf: Sorting of items on a moving conveyor belt. Part 2: Performance evaluation and optimization of pick-and-place operations, Robot. Comput.-Integr. Manuf. **16**(2/3), 81–90 (2000)
54.38 A. Grzesiak, R. Becker, A. Verl: The bionic handling assistant: A success story of additive manufacturing, Assem. Autom. **4**, 329–333 (2011)
54.39 K. Ikeuchi, B. Horn, S. Nagata, T. Callahan, O. Fein: Picking up an object from a pile of objects, A.I. Memo No. 726 (Artif. Intell. Lab./MIT, Cambridge 1983)
54.40 M.-Y. Liu, O. Tuzel, A. Veeraraghavan, T. Marks, R. Chellappa: Fast object localization and pose estimation in heavy clutter for robotic bin picking, Int. J. Robotics Res. **31**(8), 951–973 (2012)
54.41 J.J. Rodrigues, J.-S. Kim, M. Furukawa, J. Xavier, P. Aguiar, T. Kanade: 6D pose estimation of textureless shiny objects using random ferns for bin picking, IEEE/RSJ Int. Conf. Intell. Robots Syst., Vilamoura (2012)
54.42 Z. Pan, J. Polden, N. Larkin, S. Van Duin, J. Norrish: Recent progress on programming methods for industrial robots, Robot. Comput.-Integr. Manuf. **28**(2), 87–94 (2012)
54.43 P. Klüger: Future body-in-white concepts. Increased sustainability with reduced investment and life-cycle-cost, Automot. Manuf. Solut., Shanghai (2013)

54.44 M. Shah, R.D. Eastman, T. Hong: An overview of robot-sensor calibration methods for evaluation of perception systems, Workshop Perform. Metr. Intell. Syst., New York (2012)

54.45 Deutsches Institut für Normung: DIN8593-0:2003-09: Manufacturing processes joining – Part 0: General; classification, subdivision, terms and definitions, http://www.beuth.de/de/norm/din-8593-0/65031206 (2003)

54.46 N. Lohse, H. Hirani, S. Ratchev, M. Turitto: An ontology for the definition and validation of assembly processes for evolvable assembly systems, Proc. 6th IEEE Int. Symp. Assem. Task Plan. (2005) pp. 242–247

54.47 K. Malecki: Vehicle body mounted door seals on a roll, Adhesion Adhesives Sealants **6**(1), 23–25 (2009)

54.48 S. Kock, T. Vittor, B. Matthias: Robot concept for scalable, flexible assembly automation, Proc. IEEE Int. Symp. Assem. Manuf. (ISAM), Tampere (2011)

54.49 H. Chen, J. Wang, G. Zhang, T. Fuhlbrigge, S. Kock: High-precision assembly automation based on robot compliance, Int. J. Adv. Manuf. Technol. **45**(9/10), 999–1006 (2009)

54.50 A. Stolt, M. Linderoth, A. Robertsson, R. Johansson: Force controlled robotic assembly without a force sensor, Proc. IEEE Int. Conf. Robotics Autom. (ICRA), Saint Paul (2012)

54.51 ABB Robotics: Application manual – force control for machining, http://new.abb.com/products/robotics/application-equipment-and-accessories/flexfinishing/function-package-for-force-control (2013)

54.52 J. Domnick, Z. Yang, Q. Ye: Simulation of the film formation at a high-speed rotary bell atomizer used in automotive spray painting processes, Proc. 22nd Eur. Conf. Liq. At. Spray Syst. ILASS Eur., Como (2008)

54.53 M. Jonsson, A. Stolt, A. Robertsson, S. von Gegerfelt, K. Nilsson: On force control for assembly and deburring of castings, Prod. Eng. **7**(4), 351–360 (2013)

54.54 Z. Pan, H. Zhanh, Z. Zhu, J. Wang: Chatter analysis of robotic machining process, J. Mater. Process. Technol. **173**(3), 301–309 (2006)

54.55 M. Weck, C. Brecher: *Werkzeugmaschinen 2 – Konstruktion und Berechnung* (Springer, Berlin, Heidelberg 2006)

54.56 U. Schneider, M. Ansaloni, M. Drust, F. Leali, A. Verl: Experimental investigation of sources of error in robot machining, Int. Workshop Robot. Smart Manuf., Porto (2013)

54.57 J. Wang, H. Zhang, T. Fuhlbrigge: Improving machining accuracy with robot deformation compensation, Int. Conf. Intell. Robots Syst., St. Louis (2009)

54.58 C. Lehmann, B. Olofsson, K. Nilsson, M. Halbauer, M. Haage, A. Robertsson, O. Sornmo, U. Berger: Robot joint modeling and parameter identification using the clamping method, IFAC Conf. Manuf. Modell. Manag. Control, Saint Petersburg (2013)

54.59 B. Saund, R. DeVlieg: High accuracy articulated robots with CNC control system, SAE Int. J. Aerosp. **6**(2), 780–784 (2013)

54.60 U. Schneider, J.R. Diaz Posada, M. Drust, A. Verl: Position control of an industrial robot using an optical measurement system for machining purposes, Proc. 11th Int. Conf. Manuf. Res. (2013) pp. 307–312

54.61 U. Schneider, B. Olofsson, O. Sörnmo, M. Drust, A. Robertsson, M. Hägele, R. Johansson: Integrated approach to robotic machining with macro/micro actuation, Robotics Comput.-Integr. Manuf. **30**(6), 636–647 (2014)

54.62 International Organization for Standardization: ISO 12100:2010. Safety of machinery – General principles for design – Risk assessment and risk reduction, http://www.iso.org/iso/home/store/catalogue_detail.html?csnumber=51528 (2010)

54.63 European Parliament: Directive 2006/42/EC of the European Parliament and the Council of 17 May 2006 on machinery, and amending Directive 95/16/EC (recast), http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:157:0024:0086:EN:PDF (2006)

54.64 Siemens: *Easy Implementation of the European Machinery Directive, Functional Safety of Machines and Systems* (Siemens, Fürth 2012)

54.65 International Organization for Standardization: ISO 13849-1:2006. Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design, http://www.iso.org/iso/home/store/catalogue_detail.html?csnumber=3491 (2006)

54.66 International Electrotechnical Commission: IEC 62061 ed1.0. Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems: http://webstore.iec.ch/webstore/webstore.nsf/artnum/033604!opendocument (2005)

54.67 International Electrotechnical Commission: IEC 61508 ed2.0. Functional safety of electrical/electronic/programmable electronic safety-related systems, http://www.iec.ch/functionalsafety/standards (2010)

54.68 International Organization for Standardization: ISO 13855:2010. Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body, http://www.iso.org/iso/home/store/catalogue_detail.html?csnumber=42845 (2010)

54.69 International Organization for Standardization: ISO 13857:2008. Safety of machinery – Safety distances to prevent hazard zones being reached by upper and lower limbs, http://www.iso.org/iso/home/store/catalogue_detail.html?csnumber=39255 (2008)

54.70 R. Hollmann, A. Rost, M. Hägele, A. Verl: A HMM-based approach to learning probability models of programming strategies for industrial robots, 2010 IEEE Int. Conf. Robotics Autom., Anchorage (2010)

54.71 B. Lacevic, P. Rocco, A. Zanchettin: Safety assessment and control of robotic manipulators using danger field, IEEE Trans. Robotics **29**(5), 1257–1270 (2013)

54.72 International Organization for Standardization: ISO/TC 184/SC 2. Robots and robotic devices, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=54138&development=on (2014)

54.73 S. Oberer-Treitz, A. Puzik, A. Verl: Measuring the collision potential of industrial robots, 41st Int. Symp. Robot. 6th Germ. Conf. Robot. (2010)

54.74 R. Behrens, N. Elkmann: Study on meaningful and verified thresholds for minimizing the consequences of human-robot collisions, IEEE Int. Conf. Robotics Autom. (ICRA) (2014) pp. 3378–3383

54.75 S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, A. Albu-Schaeffer: On making robots understand safety: Embedding injury knowledge into control, Int. J. Robotics Res. **31**(13), 1578–1602 (2012)

54.76 J. Marvel, J. Falco: *Best practices and performance metrics using force control for robot assembly* (Nat. Inst. Stand. Technol., Gaithersburg 2012)

54.77 R. Zurawski: *Integration Technologies for Industrial Automated Systems* (CRC, Boca Raton 2006)