5
Manipulation and Control

Q1 Q2 *José de Gea Fernández, Elie Allouis, Karol Seweryn, Frank Kirchner, and Yang Gao*

5.1
Introduction

The use of robotic systems for planetary exploration has been successfully shown in recent years with outstanding examples such as the Mars rovers Opportunity, Spirit, and Curiosity. While the first phase of planetary exploration tends to deal with "passive" *in situ* exploration of unknown areas by using cameras, measuring values of the atmosphere by using onboard instruments, and so on, the relevant scientific experiments can be performed as soon as samples (such as soil or rocks) can be analyzed either by bringing them back to Earth or by analyzing them *in situ*. In either case, the rover is required to have the ability to gather samples, drill into the rocks, or bring the samples to the rover's onboard scientific instruments in order to get them analyzed; hence, robotic manipulation is an importance functionality in these missions. In case of advanced missions such as to establish extraterrestrial outposts on the Moon, the robots need to be equipped with one or several manipulators to be able to grasp, transport, and assemble infrastructure.

This section reviews current and existing robotic manipulation systems for planetary exploration. The rest of the chapter discusses relevant design requirements, specifications, and procedures, describes underlining technologies such as dynamical and motion control of robotic arms, and presents various future needs and directions in this area.

5.1.1
Review of Planetary Robotic Arms

As explained in Chapter 1, planetary robotic systems belong to a class of their own where robustness and accuracy need to be achieved with limited mass, power, and processing capabilities. Hence, the design and operation of a robotic arm for space should be challenging and in this case rather unique too. Up to date, a fewer number of planetary robotic arms have been designed, flown, or operated successfully compared with the total number of successful missions that have landed on the extraterrestrial bodies. This section reviews these existing developments

Contemporary Planetary Robotics: An Approach Toward Autonomous Systems, First Edition.
Edited by Yang Gao.

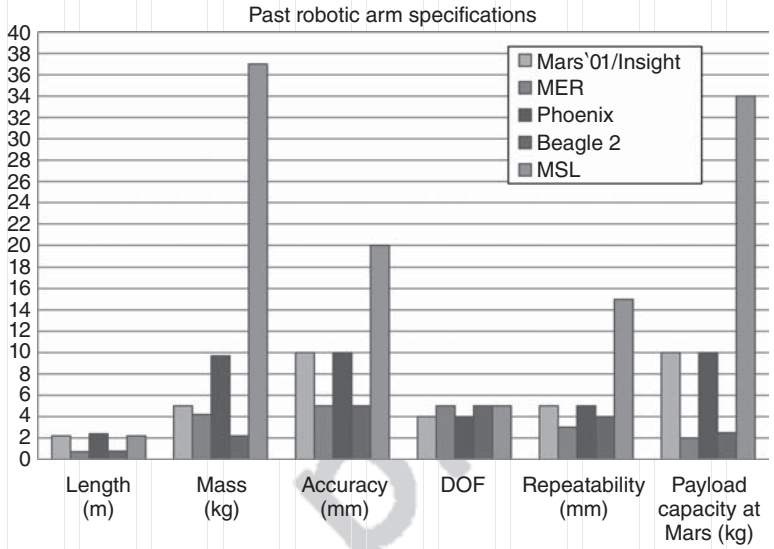


Figure 5.1 Specifications of existing planetary robotic arm systems.

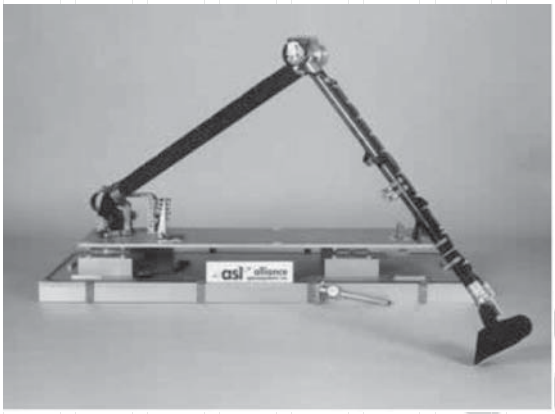
Table 5.1 Overview of existing planetary arms.

Arms	Length (m)	Mass (kg)	Accuracy (mm)	DOF (–)	Repeatability (mm)	Payload (kg)	Materials capacity	Max speed (deg/s)
Beagle 2	0.75	2.2	5	5	4	2.5	Ti joints-CFRP	0.5
Mars'01	2.2	4.1	10	4	5	10	Graphite epoxy	2–6
MER	0.7	4.2	5	5	3	2		
Phoenix	2.4	9.7	10	4	5	≈10	Alu/Ti	
MSL	2.2	37	20	5	10	34		3.2–5.2

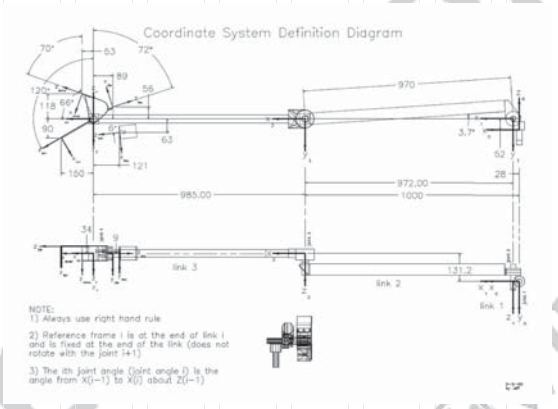
in real-world missions and the state-of-the-art planetary robotics they represent. Figure 5.1 and Table 5.1 provide a summary and comparison of properties of these robotic arm systems.

5.1.1.1 Mars Surveyor '98/'01

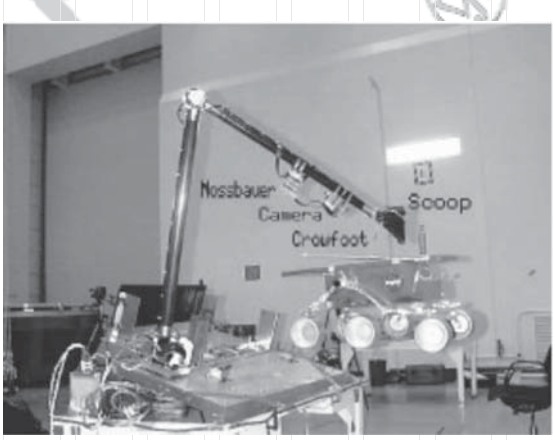
The Mars Surveyor 2001 robotic arm as shown in Figure 5.2 is a low-mass 4 degree-of-freedom (DOF) manipulator with a backhoe design inherited from the Mars Surveyor '98 robotic arm [1, 2]. The end-effector consists of a scoop for digging and soil sample acquisition, secondary blades for scraping, an electrometer for measuring tribo-electric charge and atmospheric ionization, and a crowfoot for deploying the rover from the lander to the surface. Control of the arm was achieved by a combination of software executing on the lander computer and



(a)



(b)



(c)

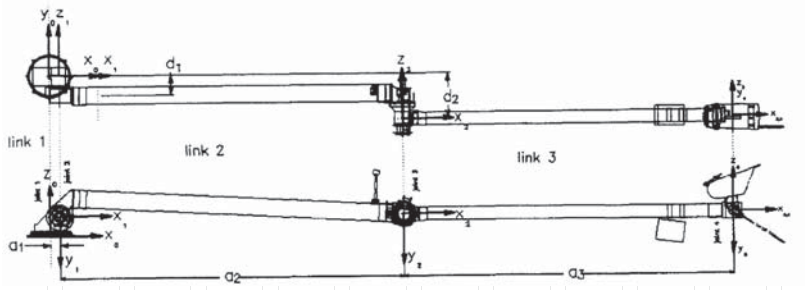
Figure 5.2 MARS'01 arm [2, 3]. (a) Photo, (b) kinematics, (c) in operation for rover deployment. (Courtesy NASA/JPL-Caltech.)

firmware resident in the robotic arm's electronics. The arm was an essential instrument in achieving the scientific goals of the Mars Surveyor '01 Mission by providing support to the other Mars Surveyor 2001 science instruments as well as conducting arm-specific soil mechanics experiments. While the arm was not flown, it was space qualified to be launch on the canceled Mars'01 mission. In addition to the acquisition of samples, it was designed to deploy a small 10 kg rover the size of the Sojourner rover. At 2.2 m long its graphite/epoxy construction made it a lightweight system with good accuracy for its purpose.

5.1.1.2 **Phoenix**

Building on the Mars'01 design, the Phoenix Mars Lander robotic arm as shown in Figure 5.3 is a 2.4 m arm with a aluminum/titanium construction, pushing its mass to a recorded mass of 9.7 kg [4]. It operated for 149 sols after landing on May 25, 2008. During its mission, it dug numerous trenches in the Martian regolith and acquired samples of Martian dry and icy soil.

(a)

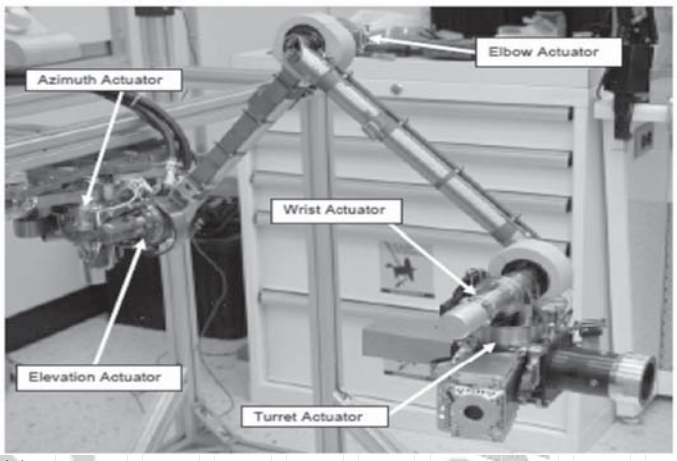


(b)

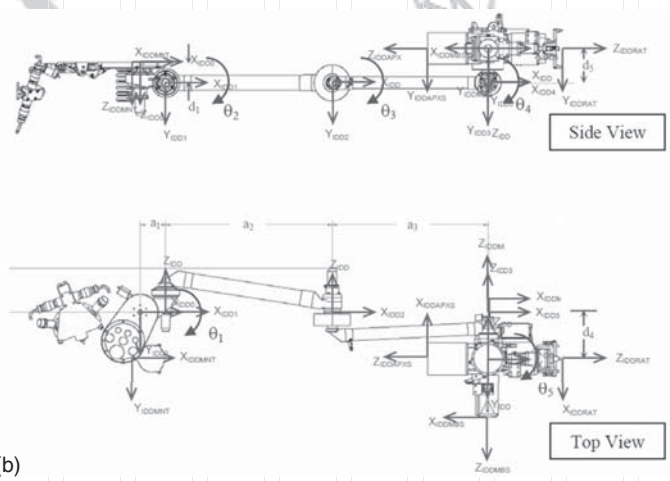
Figure 5.3 Phoenix arm [4]. (a) Calibration and testing, (b) kinematics. (Courtesy NASA/JPL-Caltech.)

5.1.1.3 MARS Exploration Rovers (MERs)

The successful instrument deployment device (IDD) as shown in Figure 5.4 has been used in the two MERs and has accommodated 5 DOFs with an overall length of 0.7 m and a mass of 4.2 kg [5]. It accommodated a range of payload and was tasked to deploy them directly to target locations onto rocks and the surface. The driving system requirements for the IDD are primarily concerned with the absolute and relative positioning performance associated with the placement of the instruments on targets of interest including rock and soil targets as well as rover-mounted targets. The absolute positioning requirement stated that each *in situ* instrument should be positioned to within 10 mm in position and 10° with respect



(a)



(b)

Figure 5.4 MER IDD [5]. (a) Breadboard, (b) kinematics. (NASA/JPL-Caltech.)

to the surface normal of a science target that has not been previously contacted by another *in situ* instrument. This requirement was then broken down into two error budgets associated with the ability of the IDD to achieve a certain instrument position and orientation and the ability of the front HazCam stereo camera pair to resolve the 3D position and surface normal of a science target. Therefore, the overall absolute positioning and orientation error requirements were split equally into two error budgets. The IDD was required to be capable of achieving a position accuracy of 5 mm and an angular accuracy of 5° in free space within the dexterous workspace of the IDD. Factors that affect the ability of the IDD to meet this requirement include knowledge of the IDD kinematics (link lengths, link offsets, etc.), knowledge of the location of actuator hardstops used to home the actuators, actuator backlash effects, closed-loop motion controller resolution, and knowledge of IDD stiffness parameters. A calibration procedure was utilized to experimentally determine the parameters that affect the IDD positioning performance. The remaining half of the error budget was assigned to the front HazCam stereo pair such that the vision system was required to determine the location of the science target with a position accuracy of 5 mm and the angular accuracy was 5° with respect to the target's surface normal. Factors that affect the ability of the stereo camera pair to meet this requirement include camera calibration errors, stereo correlation errors, and image resolution issues. For RAT grinding operations, the IDD is required to place and hold the RAT on the rock target with a specified preload. The IDD is required to provide the RAT with a preload of at least 10 N within 90% of the reachable science target workspace. As mentioned previously, each instrument carried proximity sensors to detect contact between the instrument and the target surface.

5.1.1.4 **Beagle 2**

The Beagle 2 arm [6] as shown in Figure 5.5 was designed, built, and space qualified for the Beagle 2 mission and is to date the only European planetary robotic arm to be designed up to flight level, including planetary protection qualification. It had a length of 0.75 m and 5 DOFs to deploy an instrument workbench of 2.5 kg. The arm was particularly compact and at 2.5 kg and the lightest arm sent to Mars with the highest mass to payload ratio of about 1 : 1. Each joint comprises a Maxon DC brushed motor and high-ratio planetary gearbox, driving through a 100 : 1 harmonic drive gearbox. Joint position is detected by a potentiometer mounted directly to the output shaft. All structural items were manufactured from titanium in order to closely match the thermal expansion of the bearings, while minimizing the mass. It is also the best choice where the carbon fiber arm tubes are bonded to the end fittings. The performance of the arm was driven by the accuracy of the stereo camera pair on the position adjustable workbench (PAW) of ±2 mm and the resolution of the potentiometer used to measure joint position (±0.2 mm). A total positional accuracy of ±5.34 mm was achieved. All the instruments on the PAW used to examine rocks required a contact force to be applied. The arm was, therefore, capable of generating the 5 N required (Table 5.2).

Q3

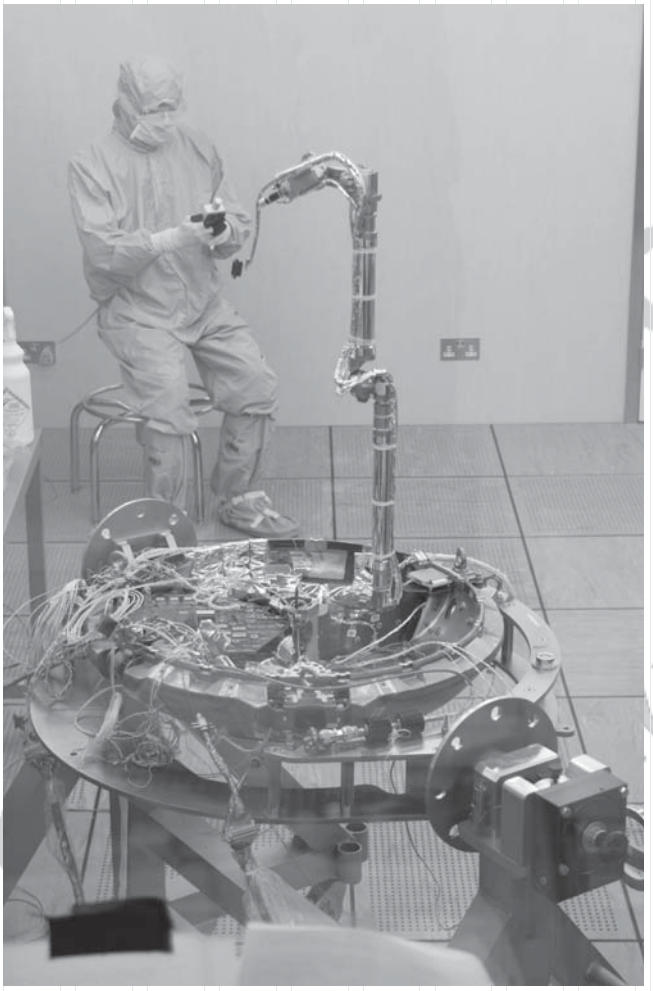


Figure 5.5 Beagle 2 arm. (Courtesy Beagle 2 Team.)

5.1.1.5 Mars Science Laboratory

The Mars Science Laboratory robotic arm as shown in Figure 5.6 is a critical, single fault-tolerant mechanism in the MSL science mission that must deliver 5 out of the rover's 12 science instruments to the Martian surface [7].

The main attributes of the MSL robotic arm are

- 5 DOFs;
- 2.2 m outstretched length from base to center of instrument turret;
- 67 kg mass without turret instruments;
- 5 turret instruments with mass of 34 kg;
- electrical cabling system with 920 signals traversing the length of the arm;

Table 5.2 Beagle 2 arm performance.

Characteristics	Performance achieved
Mass (kg)	2.2
Maximum reach (m)	0.709
Max output torque (N m)	25
Back-driving torque (N m)	21
Max rotational speed (deg/s)	0.5
Supply voltage (V)	12
Drive current (mA)	100 (Max)
Position feedback	10 kΩ, 0.1% Linearity potentiometer
Operating temperature range (°C)	−40 to +30
Nonoperating temperature range (°C)	−100 to +125

- two dual-use caging mechanisms capable of surviving landing loads of over 20 g, passively re-stowing the RA after deployment, and surviving rover driving loads of 8 g;
- capable of surviving temperature range of −128 to +50 °C and operating within a temperature range of −110 and +50 °C.

The primary function of the arm is to position the turret-mounted instruments and tools with respect to Mars surface or rover-mounted targets [8]. The key requirement levied against absolute positioning accuracy is the arm shall be placed with an accuracy of 20 mm in position and 10° in orientation relative to the surface normal of a target selected in stereo imagery. Positioning accuracy is to be 15 mm for instruments that can sense contact with the target (lateral accuracy). The arm is required to have a repeatability of 10 mm. A key contact science requirement is for the system to be capable of deploying and placing an instrument on a surface target determined from stereo imagery, retracting the instrument and placing another instrument or tool within a single command cycle. All requirements related to contact science are applicable to rover tilts up to 30°. The requirements regarding science target operations are specified with respect to the arm's primary workspace, which is a 1 m tall vertically oriented cylinder with a radius of 800 mm as shown in Figure 5.6. It is located 1.1 m in front of the rovers front panel and extends 200 mm below the level of the rovers front wheels. The system is required to be capable of sample acquisition at 90% of the reachable targets within the primary workspace. In addition, the system is required to be capable of acquiring, processing, and delivering samples to science instruments when the rover has a tilt up to 20°. Each of the five arm joints is driven by an actuator consisting of a brushless DC motor integrated with an encoder, a planetary gearhead, a resolver, a brake, and hard stop hardware. Motor angular position is measured by the incremental encoder. Actuator output angle (joint angle) is inferred from motor position through the gearhead reduction ratio. The joint angle is also measured directly by the resolver. After joint motions are complete, the brakes are engaged to hold the

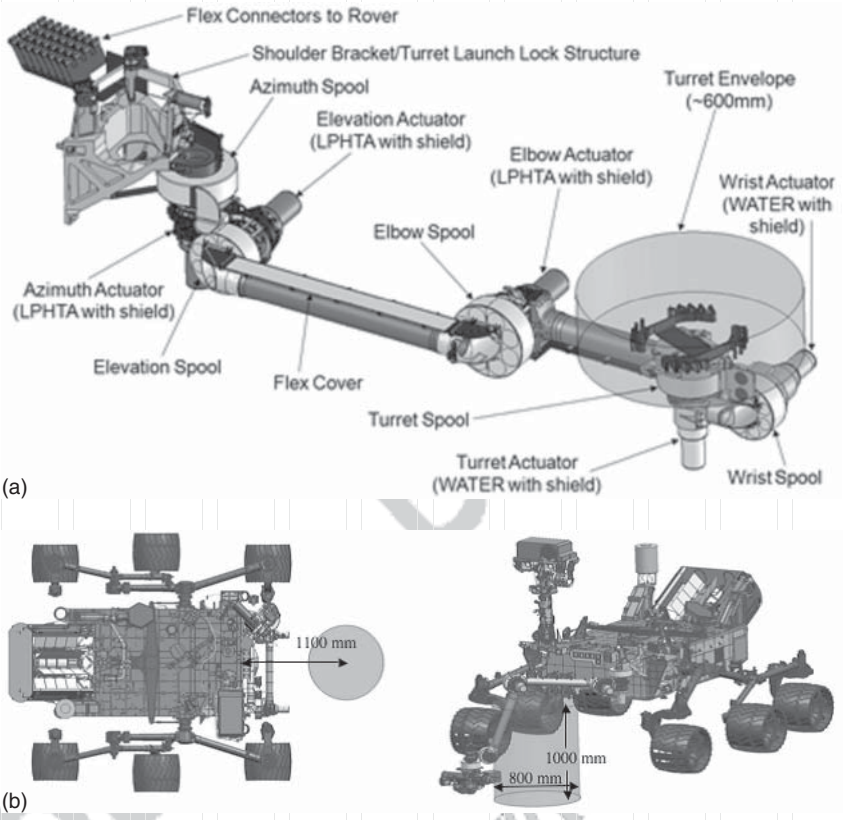


Figure 5.6 MSL robotic arm [8]. (a) Drawing, (b) primary workspace. (Courtesy NASA/JPL-Caltech.)

motor position without servoing (and without power). The two shoulder joints and the elbow use low-power high-torque actuators (LPHTAs). The remaining joints use wrist and turret actuators (WATERs). A summary of the capabilities of these actuators is shown in Table 5.3.

Q4

5.2 Robotic Arm System Design

5.2.1 Specifications and Requirements

The design of a planetary robotic arm system needs to first investigate the performance requirements (such as accuracy and repeatability), environmental requirements (such as lighting, dust, and thermal), as well as the design requirements

Table 5.3 Curiosity arm parameters.

Parameters	Units	LPHTA	WATER
Gear ratios	None	7520	4624
Max output torque	N m	1143	259
Max current limit	A	5	3
Max speed	RPM	0.532	0.865
Backlash	mrad	3.64	4.36
Brake holding torque	N m	1313	517
Mass	kg	7.8	4.24

(such as size, mass, power, speed, payload, reach, number of DOFs, redundancy, workspace or working envelope, and autonomous capabilities).

The requirements of a planetary manipulator design are translated from the tasks that the manipulator is expected to perform. For instance, for a simple pick-and-place operation in the manufacture industry, a robotic arm with 3 DOFs is deemed sufficient. Similarly, payload of the robotic arm depends on the anticipated target objects to be handled. Ideally, the manipulator should be designed with extra flexibility so that it can potentially perform a variety of tasks. But the space system design constraints (typically with regard to the mass) are expected to impose restrictions to the design (e.g., end up limiting the DOF to reduce the mass).

5.2.1.1 Performance Requirements

The performance of a manipulator is primarily determined by the chosen kinematic structure, by the actuator technology used and by the control algorithms in place. Those decisions can influence a number of performance measures but basically related to the following:

- Accuracy: This measure indicates how good a manipulator can position itself in a certain location in space and is an important figure especially if the manipulator is used for nonrepetitive movements. The accuracy of the manipulator is in turn influenced by the accuracy of the arm kinematic model available, which usually needs to be enhanced with experimental calibration procedures. Typical required accuracy for the industrial manipulator is around tenths of millimeter.
- Repeatability: This measure indicates the ability of the manipulator to return repeatedly to the exact same location when given the same target position and is an important figure for a reliable manipulator, in this case, especially for repetitive tasks with preprogrammed positions. The magnitude of the required repeatability depends on the task. For example, typical industrial robots are below 1 or 2 mm, that is, an order of magnitude higher than the accuracy.

5.2.1.2 Design Specifications

- Workspace: The workspace is the spatial volume in which the manipulator can work; in other words, the position and orientations in space that the robot can

reach. It is defined by three characteristics of the mechanical design, namely the length of each link, the range of movement of each joint, and the type of joint (revolute or prismatic) used. The design should try to maximize the so-called "dexterous workspace" (set of positions in space that can be reached with multiple orientations of the robot's end-effector). The "dexterous workspace" is not only interesting in the phase of the mechanical design, but can also be computed and used during the execution of a manipulation movement in order to relocate the rover or the arm to manipulate in the most convenient area.

- Reach: The reach can be seen as the maximum extent of the "reachable" workspace ("reachable workspace" is the workspace defined by the positions, which can be reached at least with one orientation). Its value depends on the tasks that the robot needs to fulfill and the usual location of the objects to grasp (in front of the rover, on the sides, behind) with respect to the location of the robotic system on the rover or lander.
- Size: The size (length) of the manipulator is determined primarily by the requirements on the robot's workspace: that is, how far should it reach. The size, in terms of how "bulky" the manipulator is, is primarily given by the requirements on the payload (the more payload, the larger motors are required, and consequently, the more weight of the structure) and the required precision (the more precision, the stiffer the robot, that is, usually thicker and thus heavier links).
- Payload: The payload is the maximum load that the manipulator is able to lift (excluding its own weight). Obviously, payload is related to the acceleration or speed at which the object being held needs to be moved. The greater the acceleration, the lower the payload than can be moved for the same robot. In extraterrestrial bodies, the fact of usually dealing with low gravity and requiring very low motion speeds allows for relative large payloads without requiring large torques (thus not requiring heavy motors).
- Mass: The mass of the robotic system depends on the previous points related to the size of the system. Its maximum value is likely given as a mission constraint on the robot's maximum weight, which in turn influences size and structure of the robot.
- Power: Similarly, the power required to drive the robotic manipulator depends on the size, payload, and mass of the manipulator. In addition, the environmental conditions (duration of the day and the nights) of the planetary body influence the size of the energy system. The robot needs to be able to work during the day and keep enough energy to withstand the cold night during which probably some energy is used to heat up some subsystems.
- Number of DOFs: The number of degrees of freedom (which equals to the number of joints in a usual serial open chain manipulator) is determined by the demand of the task. Six DOFs would be required to place the end-effector at any arbitrary position and orientation within the "dexterous workspace" of the robot. However, many tasks can be accomplished with fewer DOFs as they might not need arbitrary orientations. For instance, a simple pick-and-place

operation could be achieved with only 3 DOFs if the object can be always picked up vertically from above.

- Redundancy: Redundancy in terms of the manipulator kinematics refers to the fact of possessing more DOFs than the task requires. For instance, if the task requires 3 DOFs (to position the robot at a certain 3D position regardless of its orientation), a robot with 4 or more DOFs is called redundant. However, a classical redundant robot is the one having 7 or more DOFs, because generally a robot manipulator needs to be able to reach a given 3D position and orientation (thus, requiring 6 DOFs). One benefit of having the redundancy is to allow additional criteria for the motion of the robot without modifying the end-effector position and orientation. A typical example is for a manipulator holding an object in a 3D space, keeping fixed position and orientation of the end-effector, while at the same time being able to move its "shoulder" (or any other) joint to avoid obstacles or to reconfigure the arm posture, and so on. On the other hand, the control algorithms are reasonably more complex, as they need to use some type of redundancy resolution method.
- Autonomous capabilities: An additional requirement can be given by the level of autonomy (LoA) that the robot needs to possess, ranging from a manually teleoperated system to a fully autonomous system (as introduced in Section 1.2 and explained in Section 2.6.2 from system design point of view). The chosen LoA drives various hardware systems design; for example, in order to achieve full autonomy, the robot needs to be equipped with powerful onboard computational resources and dedicated sensor devices. A thorough discussion on autonomy for planetary robots is detailed in Chapter 6.
- On- or off-board processing: Depending on the required LoA, more or less onboard processing power need to be integrated within the robot. For a teleoperated system, the robot requires minimum onboard computational power to implement low-level controllers, which move the robot joints according to the commands received via a remote connection. But if the robot needs to be able to plan collision-free trajectories for the arm, the requirements for the onboard processing increase substantially.
- Software control algorithms: Low-level (joint level) control algorithms is the minimum requirement for the arm in order to drive the joints, which refers to feedback controllers that send commands to the joints (positions or velocities) and use sensors to monitor the accomplishment of the commands. In addition, feedforward signals might be used to compensate for dynamics effects. In case of using lightweight materials, which might introduce vibrations or inaccuracies on the positioning due to bending of the mechanical arm links, vibration suppression algorithms is also required onboard. High-level control algorithms (i.e., output references to be followed by the low-level controllers) are additional demands, which can be implemented either on board the robot at the cost of computational power or on ground at the cost of uplink communication latency.

One can take the specifications of the robotic arm mounted on the Mars Science Laboratory rover (Curiosity) as an example [8]. The Curiosity rover was required

Q6

to gather soil samples and drill rocks in order to provide samples to the instruments on board the rover. Moreover, the manipulator had to be able to place science instruments over interesting surface targets to study their chemical and mineral composition. As such, the manipulator had four primary requirements to achieve those tasks [8]: (i) to be able to place instruments over surface targets (rocks and regolith), which requires stereo camera systems to extract the 3D location of the targets, (ii) to be able to place the arm to acquire soil samples or position the drill over rock targets, (iii) to be able to support the processing of samples by shaking the onboard instruments, and (iv) to be able to drop off samples to onboard instruments. Basically, the manipulator had to be able to position its tools relative to desired targets over the Mars surface or over onboard instruments carried by the rover. The required accuracy for successfully achieving those tasks was defined as to be lower than 20 mm in position and 10° in orientation and a repeatability lower than 10 mm. The workspace was defined as a vertically placed cylinder of 1 m in height, with radius 800 m and located 1.1 m in front of the rover. The final design selected a configuration with 5 DOFs and a 30-kg payload with a reach of around 2 m. The joints are driven by brushless DC motors equipped with incremental encoders, planetary gears, and brakes. What is called the low-level arm control includes forward and inverse kinematics computations, trajectory generation, and deflection compensation among others. The latter is required to compensate for the deflection of the arm links due to the long reach of the arm (over 2 m) and the heavy payload (30 kg). The forward and inverse kinematics use a rigid-body assumption, which is compensated on board by using a model of the stiffness of the arm to compute its deflection depending on the current pose. In this way, the positioning error can achieve the required specifications. High-level behaviors are a series of sequences of low-level behaviors to ease the use of the rover for recurring operations. For example, the behavior ARM_PLACE_TOOL would place the tool in the specified target position by sequentially calling a series of low-level behaviors.

5.2.1.3 Environmental Design Considerations

For a planetary robot hence the manipulator, environmental conditions of the specific extraterrestrial body in many respects drive the robot design and selection of components. A thorough study on environment-driven design considerations is presented in Chapter 2.4. Here, some of the design factors are re-enforced with regard to the robotic manipulator system.

- Light: One of the most interesting areas to yet be explored on extraterrestrial bodies are craters, where the use of a manipulator to gather samples from inside the crater can be of high scientific interests. Within the craters, the areas with "eternal darkness" are most interesting for science, for example, the famous Shackleton crater at the South Pole on the Moon where measurements from the Lunar Reconnaissance Orbiter/LCROSS indicate the presence of water ice. However, the eternal darkness needs to be considered while designing the camera systems, which would be used by a manipulator to recognize the targets.

- Dust: Dust storms have been a major problem for missions on Mars. In 2007, dust storms covered the solar panels of the rovers Spirit and Opportunity preventing sunlight hence power generation. In addition, dust on the lens of a microscopic imager mounted on Spirit reduced the quality of the images delivered by that camera. The abrasive nature of the dust allows it to seep into mechanisms through seals and reduce the efficiency of the actuators, leading to jamming and ultimately failure. For a planetary manipulator, the dust problems need to be taken into consideration, either by having some way of self-cleaning or by designing the system such that it can work with reduced performance, energy, or image quality.
- Gravity: In a low-gravity environment compared with Earth, the weight of the manipulator will be lower, which in turn requires lower torques to drive its joints, and vice versa. For example, the working Canadarm in space can have trouble to work on Earth as the actuators of the manipulator are likely unable to lift its own weight under Earth surface gravity. Gravity (lower or higher) also affects the dynamic properties of the robot. For example, the manipulator is designed to use the computed-torque control (as described in Section 5.3.1.3) that requires identification of the dynamic model of the robot manipulator, then the dynamic model needs to be reidentified (or at least adapted) if the control needs to be used on another planetary body.
- Temperature: It is known that the temperatures on other planetary bodies can vary abruptly. On Mars, for instance, temperatures might reach 20 °C at noon on the equator, whereas at the poles the temperatures can be as low as −150 °C. As mentioned earlier, the polar regions (and their craters) are usually most scientifically interesting for sample collection and manipulation. Consideration needs to be taken for the design of the electronics and mechanical parts to withstand such temperature range and extremes.

5.2.2 Design Trade-Offs

This section presents design trade-offs for developing the planetary robotic manipulator systems given the requirements discussed in the previous section, including selection of the kinematics (i.e., singularities, manipulability, etc.), the structure and material, and the required sensors.

5.2.2.1 Arm Kinematics

- Singularities: The selection of a certain mechanical manipulator structure determines a certain number of configurations in which the robot enters a "singularity," that is, a configuration in which the robot is not able to generate end-effector velocities (or forces) in a certain direction. Near a singularity, the joint velocity required to achieve a certain Cartesian velocity can be extremely large. In other words, the robot "loses" DOF: mathematically stated, the Jacobian of the robot manipulator loses rank, that is, two or more columns of the Jacobian matrix become linearly dependent. Given the fact that the number

of degrees of a manipulator is usually carefully selected in order to achieve a certain task, the fact of entering singularities and thus losing DOF is a situation needs to be carefully studied. On the one side, the mechanical design can try to minimize the number and type of singularites (by choosing the mechanical structure) and on the other side, the control algorithms can and need to be aware of singularities in order to avoid approaching or working close to them.

- Manipulability: In order to assess how close (or how far) is the manipulator from a singularity, the concept of "manipulability" has been defined. It describes the ability of the robot to move freely in all directions within its workspace (thus, it is closely related to the robot singularities). The manipulability can refer to (i) the ability to reach a certain position (related to the workspace of the robot, thus a global measure) or (ii) to the ability of changing the position or orientation given a certain robot configuration (a small movement around the current configuration, thus, it is a local measure). To study manipulability, one studies the Jacobian of the manipulator, to relate infinitesimal joint movements to infinitesimal Cartesian movements. To this aim, many different manipulability measures have been proposed in the literature, introduced initially by Yoshikawa [9]. By using this information, the manipulator can adjust its configuration (if available, by making use of the robot redundancy) to maximize the manipulability and thus maximize the probability of succeeding on the execution of the manipulation task. The original work from Ref. [9] describes the manipulability index as a distance to singularities computed as

$$w = \sqrt{det(J(q) \cdot J^T(q))} \tag{5.1}$$

where J is the Jacobian of the robot manipulator and q the joint positions. Higher values of manipulability w indicates higher ability to move in all directions around the current configuration of the robot.

5.2.2.2 Structure and Material

Usually, robot manipulators are being designed as rigid as possible so as to minimize the deflection of the links while moving its own weight or a payload. The stiffer the robot, the more accurate the positioning can be achieved without requiring complex control algorithms. However, stiff rigid links generally mean more weight of the mechanical structure, which in space applications is of main concern. For this reason, lighter materials (i.e., usually more flexible materials such as carbon and glass-fiber composites) can be used to build the manipulator at the price of possibly reducing the robot positioning accuracy, increasing the risk of vibrations during the movement, and thus, requiring more complex algorithms to deal with those issues.

5.2.2.3 Sensors

Apparently from the previous section, a decision needs to be made about the choice of sensors on board the manipulator, and the need to consider saving the mass by using the minimum set of sensors required to successfully accomplish the assigned tasks.

- Position and velocity sensors: Joint positions (and velocities) are one of the basic manipulator sensors and of crucial importance. There are different technologies: from simple and low-resolution potentiometers to optical or magnetic encoders providing high resolution. Similarly, they can be found as incremental or as absolute encoders. The first type starts counting revolutions from the current position of the motor after power-up; thus, before having passed through a reference point, there would be no way of knowing in which absolution position the motor is in. This usually requires some sort of homing procedure (joints move until finding its "zero" position) at every start-up. On the other hand, as the name already indicates, absolute encoders signal the absolute position of the motor within a rotation.
- Force or torque sensors: The typical force sensor is attached to the wrist of the manipulator and measures forces in the three Cartesian directions and torques in the three axes of rotation of the end-effector. These measurements might be interesting for accurately controlling the forces exerted by the robot. Torque sensors on each joint are very rarely used, although they might be interesting for several purposes: for implementing computed torque control or vibration control algorithms, especially in the case of joints presenting some flexibility.
- Cameras: The use of cameras is one of the basic exteroceptive sensors that a planetary arm uses. On the one side, there is the need of recognizing the location of the objects to be grasped, which is usually accomplished by using a pair of cameras mounted as a stereo vision system. On the other side, the manipulator is probably designed using lightweight materials due to the constraints on the mission's payload, thus making the manipulator less accurate than a rigid one. In this case, a precise calibration can be performed to improve the accuracy of the arm but usually combined with the use of the camera information to drive the motion of the manipulator toward the object (what is known as "visual servoing").

5.3 Robotic Arm Control

Manipulation refers to the process of moving or rearranging objects in the environment. In order to control a robotic manipulator system to perform a desired motion, several control components are required. Those components can be grouped into three functions as shown in Figure 5.7.

1) High-level control: The highest manipulator control layer, planning the path for the end-effector and implementing the LoA for the robot, which can range from the teleoperation to the fully autonomous operation with varying LoA in between.
2) Trajectory generation: Components that provide references for the joint controllers, basically acting as interface between the high-level commands and the low-level control commands.

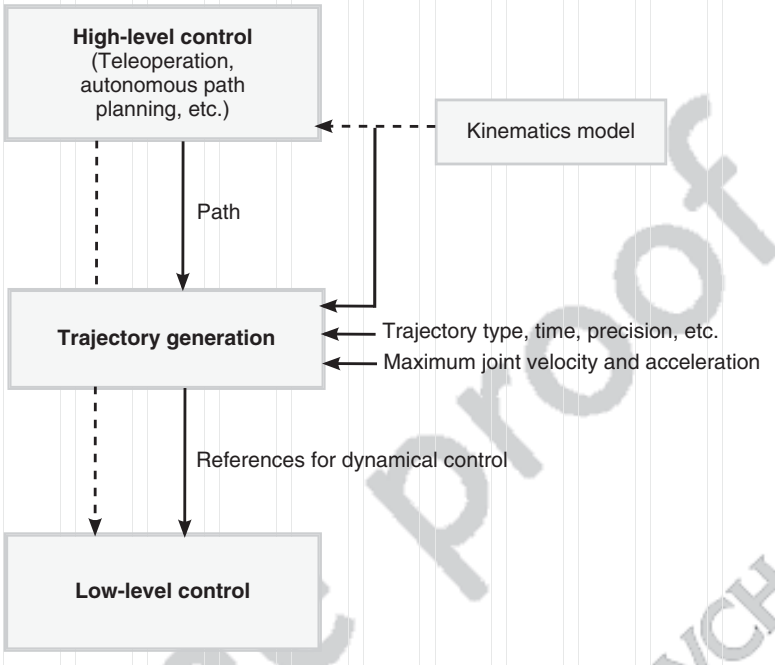


Figure 5.7 Layered structure of the motion control for a robot manipulator.

3) Low-level control: The control strategies dealing with the commands at joint level to achieve the desired motion references dictated by the high-level control (usually given in Cartesian space).

In addition to the three-level control, collision avoidance is often regarded explicitly where the control strategies need to avoid collisions of the robot with the environment and with itself (self-collision), particularly true for complex and redundant robots. In the following sections, the basic concepts and state-of-the-art design examples for each of these functions are described.

5.3.1 Low-Level Control Strategies

5.3.1.1 Position Control

Position (or velocity) trajectories, which ideally the robot manipulator should be able to follow according to its kinematic constraints, are received by the robot joints, which will try to follow the trajectories as close as possible. This step is usually known as kinematic control.

In practice though, those trajectories are not always achieved due to the dynamical characteristics of the robot (such as inertia, frictions). The dynamical model of a robot is highly nonlinear, multivariate, coupled, and of time-varying parameters, which makes the dynamical control usually a hard problem. For that

reason, in practice usually some assumptions are taken, which simplify the control scheme. Factors such as the gear ratio or the viscous friction can actually help to simplify some dynamics. For instance, a high gear ratio as commonly found in industrial robotics allows to consider the links of the robot as decoupled. In those conditions, a set independent proportional and derivative (PD) controllers for each joint might be a more than suitable solution. Note that, however, a high gear ratio does not come for free: it also adds undesired viscous friction, joint play, and joint elasticity.

This section shows the two typical independent joint feedback control schemes for stiff robots (high-gear ratios), which assume the manipulator as a decoupled system (a joint motion has no influence on the other joints' motion). In the rest of the section, the following symbols are used:

- The end-effector position and orientation is denoted as X. This 6×1 vector is defined as $X = (p^T \varphi^T)^T$, where the vector p describes the end-effector position and φ is a set of Euler angles from the rotation matrix describing the orientation of the end-effector.
- The three-dimensional force f as well as the three-dimensional moment m exerted by the robot's end-effector are the components of the wrench $h = (f^T m^T)^T$.
- The joint positions are denoted by q.
- The commanded torques are denoted by τ.
- The subscript d denotes a "desired" reference value for the specific magnitude (either force or position/orientation in Cartesian or joint space).

Feedback Position Control with Position Reference The simplest and most common form of robot joint control is independent PD control, described as

$$e_p = q_d - q \tag{5.2}$$

$$\tau = K_P \cdot e_p + K_D \cdot \dot{e}_p \tag{5.3}$$

The error signal is formed by comparing the joint position reference with the current joint position measurement. This error is then used by the proportional part (P) and the derivative part (D) of the controller to eliminate (or minimize) the error signal. The selection of the controller parameters K_P and K_D determines the behavior of the controller. The controller scheme can be seen in Figure 5.8.

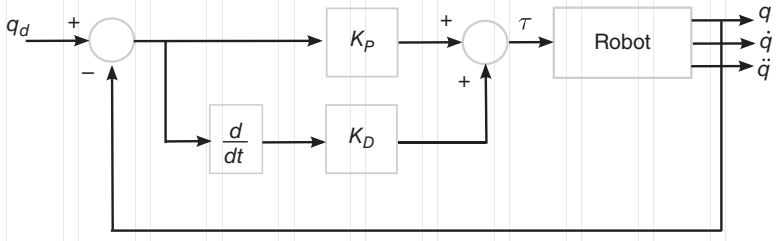


Figure 5.8 Feedback controller with position reference.

Feedback Position Control with Position and Velocity Reference A slightly better performance is obtained by using also a velocity reference in addition to the position reference, as described by

$$e_p = q_d - q \tag{5.4}$$
$$e_d = \dot{q}_d - \dot{q} \tag{5.5}$$
$$\tau = K_P \cdot e_p + K_D \cdot e_d \tag{5.6}$$

The controller scheme can be seen in Figure 5.9.

5.3.1.2 Force Control

Position control is the best solution as long as no (or little) contact with the environment is necessary. However, in case of unstructured environments such a planetary body, where the knowledge of the environment is imprecise, or in complex motion-constrained tasks, some kind of force control is required in order for the robot to attain its goal. More specifically, robots need to sense contacts with the environment and, by sensing, comply with the current environment, regardless of its unknown nature.

One of the required components for a robust and adaptive robot is the ability to adapt the contact interaction forces when manipulating an object or contacting a surface (in this area, any object/surface with which the robot makes contact is generally defined as "environment"). This problem domain is generally known as compliance control and tries to guarantee that the robot accommodates the interaction forces rather than resisting to the constraints posed by the contact with the environment.

Active compliance control methods use measurements from the contact forces (and moments) and robot's motion, which are fed back to the robot's controller in order to generate appropriate motion commands according to a desired robot's behavior. The research on active methods to provide robust force control in any of its flavors has been gaining importance in the last three decades and a great deal of research papers is available. A first description of the state of the art in the 1980s can be found at [10]. Similarly, [11] reviews the state of the art in the 1990s. Some books also appeared at that time that focused exclusively on force control [12]. More recently, Lefebvre *et* al. [13] surveyed the state-of-the-art of the required subcomponents for active compliant systems. The recent "Handbook of

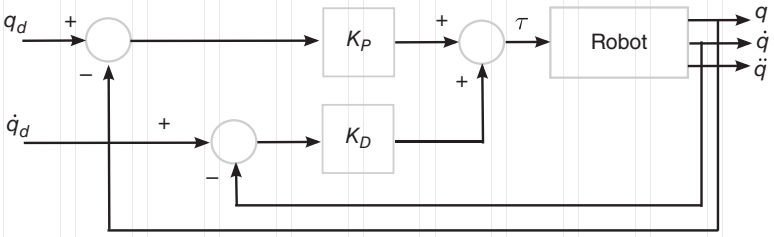


Figure 5.9 Feedback controller with position and velocity reference.

Robotics" [14] includes also a chapter that reviews the current state of robot force control. In the area of planetary rovers, the robotic arm mounted on the Mars Science Laboratory rover (Curiosity) includes the capability to autonomous drill a rock, being its main component a force feedback loop that controls the forces during the execution of the drill. The algorithms to achieve those operations are described in Ref. [15]. The drill on the rover Curiosity is the first autonomous extraterrestrial drill into a rock.

In general, any active control method for compliance control tries to somehow add or combine motion and force errors, and use a controller or set of controllers to send the most proper commands to the robot's joint actuators. The way these errors are combined is what creates the basic distinction between direct and indirect compliance control methods.

- Direct force control methods are those where the controller directly regulates the contact force to a desired reference. A classical force control strategy or a hybrid force/motion control belongs to this category.
- Indirect force control methods are those where the force is controlled indirectly via motion control. Impedance control in its different "flavors" belongs to this category.

Both approaches differ in the way to specify the interaction task. For instance, in the hybrid force/motion control (or direct method), the task is specified in the geometric space, as can be seen later that the user defines which directions are controlled on force and which ones are controlled on position. In the case of impedance control (indirect method), the designer can define a dynamic relationship between force and motion. In the end, the set of impedance parameters defined will determine the robot's behavior. When comparing the design methodologies for compliance control in a practical sense, both are quite similar: where a designer would identify a constrained direction in hybrid force/motion control, he would probably define a more compliant behavior when using an impedance controller; where a designer would identify an unconstrained direction in hybrid force/motion control, he would probably define a stiffer behavior when using an impedance controller. Interaction control methods might also be classified according to the static or dynamic performance of the control method:

- Dynamic model-based control methods are those that are concerned with the transient, that is, with the dynamical response of the system. To this category belong impedance and admittance control schemes, as well as hybrid and parallel force/motion strategies. For any of them, a complete dynamic model of the robot is required, thus being more complex, both to be designed and implemented. Furthermore, force measurements are necessary in order to obtain a decoupled and linear interaction model that is easily tractable.
- Static model-based control methods are those that are only concerned with the steady-state response of the system. In the case of impedance control, the static case is called stiffness control. In the case of admittance control, the static case is known as compliance control. That is, compliance and stiffness control are subsets of admittance and impedance control, respectively. These methods are thus

easier to implement, as they do not require dynamic models but only knowledge about the gravity terms.

A review of several interaction control methods, divided into dynamic and static model-based methods, is given in Ref. [16], which also includes experimental evaluation.

Hybrid Force/Motion Control This method [17] treats the contact interaction as a geometric problem, where there are a set of geometric constraints to be taken into account. After careful examination of the interaction task, a number of robot's DOFs are regarded as "force-controlled," whereas the rest are considered to be "motion-controlled." That means that the hybrid controller controls exclusively motion along unconstrained, "motion-controlled," directions, and force/moment along constrained, "force-controlled," directions. This approach is based on the assumption that for most of the usual constrained robotic tasks, it is possible to split the task into two mutually independent subspaces, one controlling contact forces and one controlling robot's motion. Thus, making use of different subspaces, motion, and force can be controlled simultaneously. Figure 5.10a shows a diagram of a hybrid force/motion controller.

Parallel Force/Motion Control The parallel approach [18] as shown in Figure 5.10b is classified as a direct control method as it starts with the geometric constraints also used for hybrid force/motion control.

The difference with a hybrid force/motion approach is that parallel force/motion control does not use different control subspaces for force and motion, but combines and weights the contributions of motion and force controllers into one controller using a single matrix. In this case, the controller gives priority to force errors, which dominate the controller's response. Thus, a position error would be "tolerated" along a constrained direction in order to ensure proper force tracking.

Impedance Control The general term impedance control is commonly used indistinctly when referring to either impedance or admittance control. Both pursue the same goal – active modification of the mechanical impedance of the robot – but they do it from different perspectives. Impedance control basically works by measuring position and outputting force, whereas admittance control works by measuring force and outputting position. Due to the different ways of solving the control problem, the accuracy of the compliance method lies on different factors. The impedance control depends on the accuracy of the position sensors and the bandwidth and accuracy of the force-controlled actuators, whereas in admittance control the accuracy depends on the force sensors used and the bandwidth and accuracy of the position-controlled actuators.

Active impedance control as shown in Figure 5.11a indirectly regulates the contact forces by generating an appropriate motion that ends up in a desired dynamic relationship between the robot and the environment. In contrast to hybrid force/control methods, impedance control uses a single control law to regulate simultaneously both position and force by specifying a target dynamic relationship

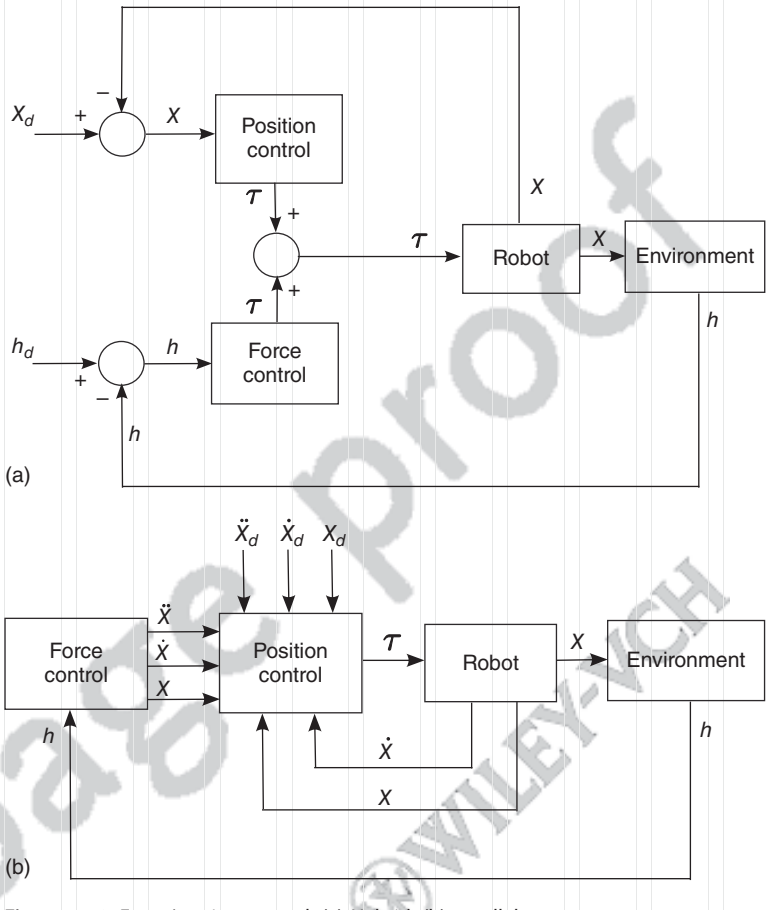


Figure 5.10 Force/motion control. (a) Hybrid, (b) parallel.

between them. In other words, it weighs the contributions of the force and motion controllers using a set of weighing matrices. In contrast to hybrid force/control where there are also weighing matrices, the matrices used in active impedance control have physical dimensions of impedance: that is, stiffness, damping, and inertia parameters. These parameters thus shape the dynamical behavior of the robot as if mechanical springs, dampers, and extra inertia were included into the robot's end-effector. The design of the target impedance that ensures a proper behavior is though not an easy task. It is clear to see that the behavior of the robot when contacting an environment needs to be different than when moving freely, especially when a good position tracking is desired in free space (remember that a stiff robot is a good position tracker, whereas by definition a stiff robot is not compliant when contacting the environment). Moreover, even a well-defined target impedance depends finally on the dynamics of the environment, which needs to be estimated as good as necessary to ensure not only stability on a first level but also the necessary robot's performance.

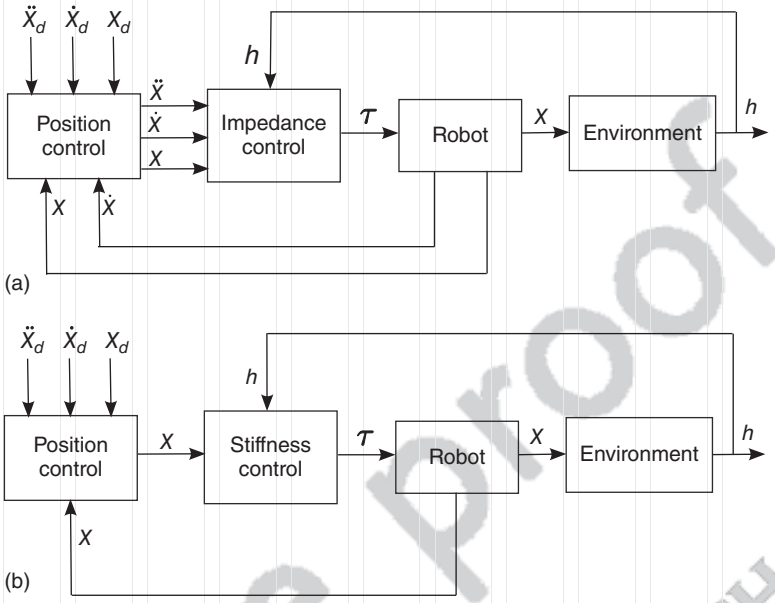


Figure 5.11 Control strategies. (a) Impedance control, (b) stiffness control.

The work from Hogan [19] is considered the benchmark for impedance control as it is the first to describe the concept of a virtual mechanical impedance for controlling robot-environment interaction forces. However, it was not the first idea on the use of virtual mechanical elements to control forces: in Ref. [20], a generalized spring and damper system for force control is presented that is later implemented in Ref. [21]. One of the best current examples of the use of active impedance control are the LWR lightweight arms from DLR [22]. Ever since a vast literature has been published around the topic in order to cope with pitfalls of the original description or deal with implementation issues. For instance, Hogan itself [23] and later other authors [24] analyzed the stability of the impedance control law or the instability originated after contacting stiff environments [25]. In order to tackle the problem of the uncertainty on the parameters of the environment model as well as of the robot, some works propose adaptive [26–29] or robust [30, 31] impedance control strategies that would deal with uncertainties. Moreover, as the original description lacks of force tracking capabilities, some works propose methods for enhancing the controller to track forces [32, 33]. The impedance control area has seen as well the use of neural network implementations of the controller [34, 35], in most cases used as a method to minimize the problems arising from the model's uncertainties. Learning algorithms have also been proposed [36, 37], as well as impedance control methods have been not only applied to single-arm systems but also to multiple-arm systems [38–40].

Stiffness Control Stiffness control is only concerned with the steady state of the end-effector, thus this is a simplification of a complete impedance control. Only

a proportional action (a single matrix) is necessary to define the behavior of the controller, which controls the behavior of the robot in such a way that behaves as a 6-DOF spring with respect to the forces (and moments) applied to the robot's end-effector. Since this method focuses specifically on the steady-state response, it does not require knowledge about the complete robot's dynamics. In this case, only the gravity terms of the Lagrangian equation representing the dynamics of the system is necessary. In brief, the stiffness control regulates the static relationship between the forces exerted on the environment and the deviation (if and as much as necessary) of the position and orientation from the desired values. Figure 5.11b shows a block diagram on the stiffness control that only receives information about position, since it does not deal with higher derivatives of this variable.

Admittance Control Admittance control holds basically the same working principle as impedance control. However, it separates explicitly the position control from the impedance control. The position controller is an inner loop, as in industrial robots, designed to be stiff to reject position disturbances robustly. The trick is that the position controller does not receive directly the desired motion but the output of the "impedance controller." Figure 5.12a shows a diagram sketching the admittance control. As it can be seen, the "admittance controller" to be more correct, since the input is force (wrench) and the output is position, receives two inputs: the desired position and the end-effector wrench (force and moment). A proper target impedance can generate an output that is a suitable position/orientation for the robot that maintains the desired dynamical relationship between force and motion.

Note that the admittance control diagram shows similarity to the previous parallel force/motion control, which requires knowledge of the interaction task in order to define the geometric constraints. The difference with admittance control is the criteria to define the force controller (physical impedance parameters vs force-controlled task directions).

Compliance Control As stiffness control is for impedance control, compliance control is a subclass of admittance control, where the controller is only concerned with the static relationship between the forces exerted by the robot and the deviation of the position/orientation from the desired values. Figure 5.12b shows the block diagram of the compliance control, where the controller only generates a reference position (no higher derivatives) for the position controller.

5.3.1.3 Dynamic Control

Dynamics studies the relations between forces acting on a body and its resulting motion. For that reason, the purpose of a robot dynamical model is to obtain the relation between the movement of the robot and the forces acting on it. The dynamical model can be used to achieve several objectives:

- simulation of the robot motion,
- design and evaluation of the mechanical structure of the robot,

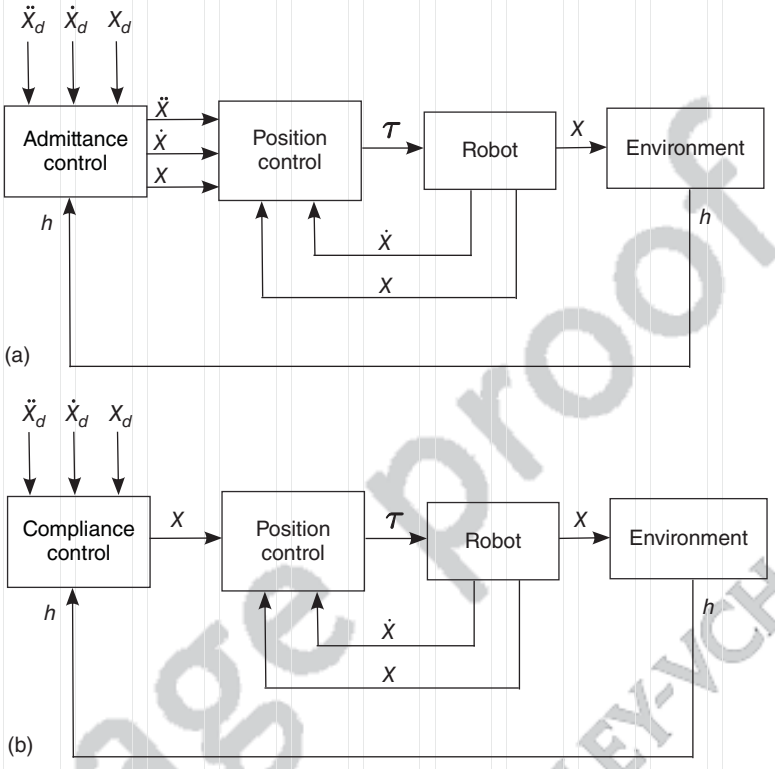


Figure 5.12 Control strategies. (a) Admittance control, (b) compliance control.

- selection of the required actuators for the robot,
- design and evaluation of the dynamical control of the robot.

This last point is of great significance, since the quality of the dynamical control determines the robot's precision and speed of movement. In other words, the use of dynamic control strategies are used to ensure that the trajectories followed by the robot are as close as possible to the ones proposed by the kinematic control.

However, the high complexity associated with the formulation of a dynamical model obliges in most cases to assume simplifications on the model. Needless to say, a complete dynamical model of a robot not only includes the dynamics of its links but also of the transmission system, the actuators, and the power/control electronics. Those elements include new inertia, friction, and saturation effects, which even make the dynamical model more complex.

The purpose of a dynamic control is to compute the torques τ, which make possible to obtain the desired joint positions q (possibly also joint velocities and accelerations). Obviously, torques are not directly sent to the motors, but the computer generates some torque-related "quantities," which in turn are transformed into analog voltages by the PWM generator. The motor then produces torques

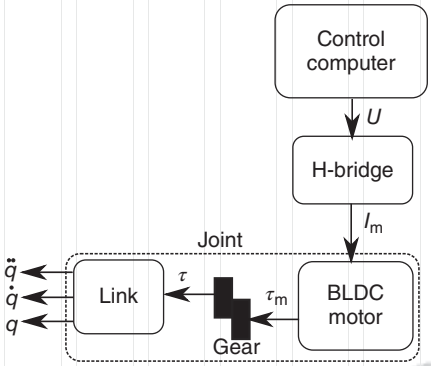


Figure 5.13 Overview of a torque control scheme of a brushless DC motor-based joint.

that provide the actuating torques τ at the joint through the gear ratios as seen in Figure 5.13.

The dynamic equation relating τ and q is well-known:

$$\tau = M\ddot{q} + C\dot{q} + G + \tau_f = M\ddot{q} + H \tag{5.7}$$

where M is the inertia matrix ($n \times n$) of the link, C are of Coriolis forces/torques ($n \times 1$), G is the vector of gravity forces/torques ($n \times 1$), and τ_f is the vector of friction forces/torques ($n \times 1$).

Mechanical Motor Model The simplified (without regard to Coriolis forces) dynamic equation of the mechanical part of the motor can be described as

$$\tau_m = J_m\ddot{q}_m + f_m\dot{q}_m + C_m \tag{5.8}$$

where τ_m is the motor torques, J_m is the inertia matrix of the motor, $\ddot{q}_m$ is the motor acceleration, $\dot{q}_m$ is the motor velocity, f_m is the motor viscous friction, and C_m is the resisting torque at the motor. We can include the gear ratio (N) to relate motor and link velocities/angles as $\dot{q}_m = N\dot{q}$. Thus, the resisting torque at the motor side is related with the link torque as

$$C_m = N^{-1}\tau \tag{5.9}$$

By using the gear ratio (N) and Eq. (5.9) in Eq. (5.8) and replacing τ by the expression in Eq. (5.7), we obtain

$$\tau_m = M'\ddot{q} + H' \tag{5.10}$$

where

$$M' = J_m N + N^{-1}M \tag{5.11}$$

and

$$H' = f_m N\dot{q} + N^{-1}H \tag{5.12}$$

Electrical Motor Model Regarding the electrical part of the model, a DC motor model is used as an example.[1)] The voltage vector at the motor input can be defined as

$$U_m = r_m I_m + L_m \frac{dI_m}{dt} + E_m \tag{5.13}$$

where r_m is the motor armature resistance, I_m is the motor's armature current, L_m is the inductance of the motor, and E_m is the motor back-emf (counter-electromotive) voltage. For DC motors, we additionally have that

$$E_m = K_e \dot{q}_m \tag{5.14}$$

and

$$\tau_m = K_c I_m \tag{5.15}$$

What it is finally missing is the power electronics (H-bridge) model, which can be assumed as a proportional relation between input voltages U and I_m motor currents as

$$U = K_v I_m \tag{5.16}$$

Feedforward Plus Feedback Control The simplest dynamic controller is implemented making use of the inverse model as feedforward signal. This allows its computation directly (and exclusively) from reference signals of position/velocity/acceleration, thus it can be computed offline. The controller is described by

$$e_p = q_d - q \tag{5.17}$$
$$e_d = \dot{q}_d - \dot{q} \tag{5.18}$$
$$\tau_{FB} = K_P \cdot e_p + K_D \cdot e_d \tag{5.19}$$
$$\tau_{FF} = \hat{M}(q_d)\ddot{q}_d + \hat{C}(\dot{q}_d, q_d)\dot{q}_d + \hat{G}(q_d) \tag{5.20}$$
$$\tau = \tau_{FB} + \tau_{FF} \tag{5.21}$$

The controller scheme can be seen in Figure 5.14. In this scheme, it might be necessary to include a zero-order delay after the computation of the inverse model since we are feeding forward a "future" signal into the inner feedback loop.

Computed Torque Control. In this control strategy, an inverse model is used in the feedback loop of the controller, that is, it needs to be computed online. In addition, there is a PD controller, which works against a simple linear model for the acceleration. Given the knowledge about Eq. (5.7), a nonlinear decoupling controller can be written as $\tau = \alpha \cdot \tau' + \beta$, and choosing $\alpha = M$ and $\beta = H$. This can decouple the system and gives $\tau' = \ddot{q}$, which means that the PD controller needs to deal only

1) Assumption is valid for a BLDC motor using a vector control strategy, which mathematically transforms the BLDC motor into a "pure" DC motor.

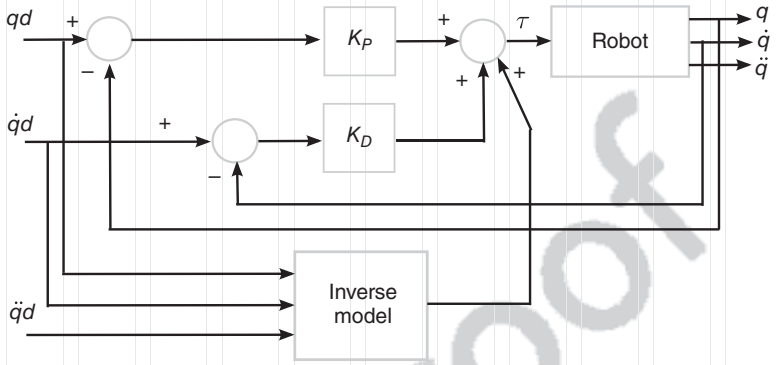


Figure 5.14 Feedforward (inverse model) plus feedback controller.

with a linear system. In summary, the controller is described by

$$e_p = q_d - q \tag{5.22}$$
$$e_d = \dot{q}_d - \dot{q} \tag{5.23}$$
$$\tau' = K_P \cdot e_p + K_D \cdot e_d + \ddot{q}_d \tag{5.24}$$
$$\tau = \hat{M} \cdot \tau' + \hat{H} \tag{5.25}$$

The schematic use of the inverse model (and its components) within the computed torque control scheme can be seen in Figure 5.15. The complete controller scheme can be seen in Figure 5.16.

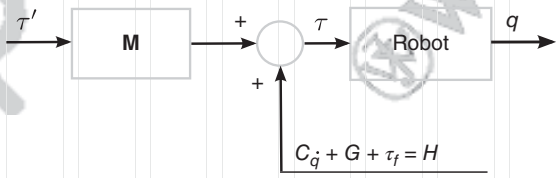


Figure 5.15 Inverse model for decoupling robot model.

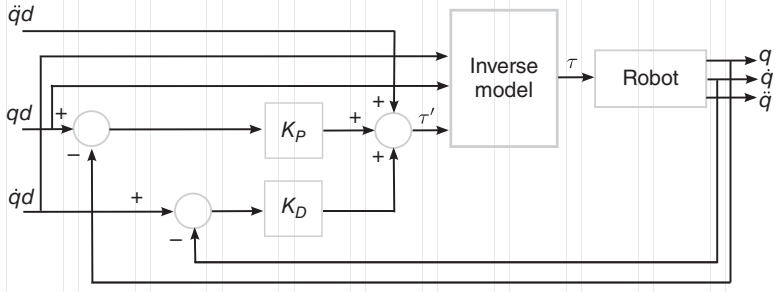


Figure 5.16 Computed torque control.

5.3.1.4 Visual Servoing

When a robot operates in an unstructured environment, the use of sensor-based control strategies for ensuring the correct execution of the manipulation tasks are unavoidable. Position or velocity control is used in most robot systems, as we just saw in the previous sections. Force control and other compliance control methods are also of interest when operating in nondeterministic scenarios as previously seen. Another source of information is vision; the use of the information from a camera as feedback for controlling the motion of a manipulator is in general known as *Visual Servoing*. The method allows to deal with real-time changes of the relative pose of the object with respect to the robot and has the advantage of possessing great accuracy and robustness. In contrast to the methods described earlier, the visual servoing does not need a explicit reference signal but always tries to reactively move toward the object by minimizing the error between the pose of the object and the pose of the gripper.

Hill and Park [41] introduced the term *Visual Servoing* in order to differentiate with previous works. In this approach, visual features from the camera image are used in order to orient, for instance, the robot gripper with the object to be grasped. The camera can be placed on a fixed position with respect to the manipulator (the so-called *eye-to-hand* approach) or can be directly be attached to the manipulator's end-effector (the so-called *eye-in-hand* approach). The general advantage of the eye-to-hand approach is that the camera is not moving and usually has a better overview over the workspace and the object/gripper. On the other side, in this case, the robot needs to track both the object and the robot's end-effector. In the eye-in-hand approach, since the camera is mounted on the gripper, the pose of the gripper can be extracted directly from the pose of the camera.

One of the first eye-to-hand systems was shown in 1973 by Shirai and Inoue [42]. In Reference [43], a static camera was used to grasp an object that is moving on a plane. Kragic [44] used a motion tracking system in connection to a grasp simulator in order to plan the grasping motion. However, the eye-in-hand configuration is used more frequently, see Refs [45, 46]. By using two cameras, complete 3D information of the object can be retrieved, even when the object is unknown. One of the first stereo visual servoing systems was implemented using a eye-to-hand configuration [47].

Apart from the location and number of cameras, visual servoing can be also categorized according to the control strategy used, namely imaged-based, position-based, or hybrid strategies [48]. In the image-based visual servoing (see Figure 5.17), the position of the gripper is extracted directly from features of the image, with a so-called feature sensitivity matrix is computed [49, 50]. By using this matrix, the motion required to move the manipulator toward the object to be grasped can be estimated. In the position-based visual servoing (see Figure 5.18), the complete 3D position and orientation (pose) of the target object are extracted from the camera information and the error between the pose of the end-effector and the object pose is brought to zero [51, 52]. The advantage of image-based method is that they do not require stereo camera system although it has to deal with the problems related to tracking features are lost from the image. When

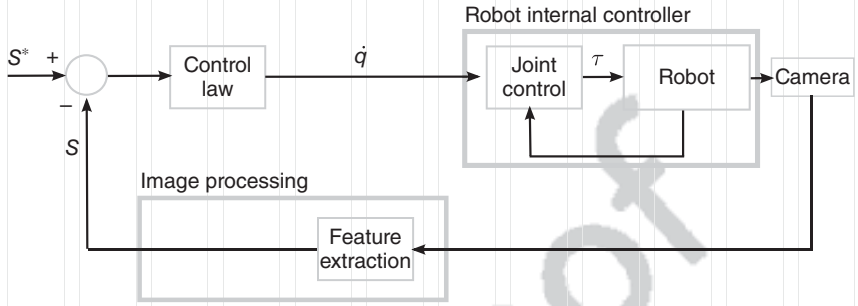


Figure 5.17 Image-based visual servoing control scheme.

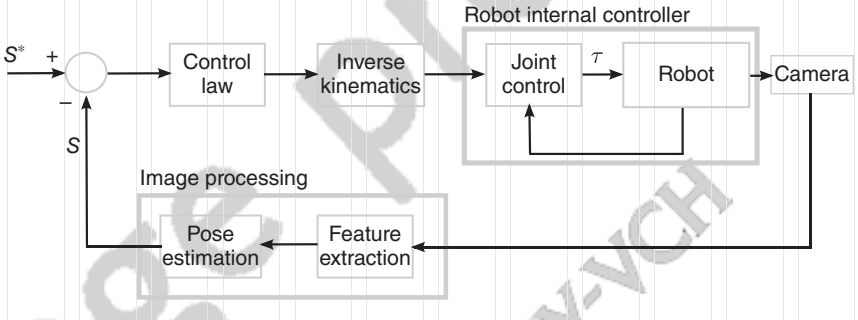


Figure 5.18 Position-based visual servoing control scheme.

using position-based methods, and since the control is done in Cartesian space, possible constraints on the path can be taken into account, for instance, to avoid collisions. However, those methods require a higher computational effort and 3D information about the object.

5.3.2 Manipulator Trajectory Generation

This section focuses on how to allow a robot manipulator moves from some initial Cartesian position X_0 to some desired final position X_f. Usually, the path to follow is given by the high-level control (for instance, by an automatic path planning module). The path consists of the initial, final, and via points that the robot is going through. A trajectory also includes the time information at which each joint reaches a certain position, velocity, and acceleration. In other words, the trajectory generation involves time-parameterizing the path given by a higher level controller. Thus, the trajectory generation module establishes the trajectories to be followed by each joint with respect to time and in order to meet the user's requirements, for example, target position, type of trajectory, and time used for the movement, and so on.

5.3.2.1 Trajectory Interpolation

In most cases, paths are given and formulated in Cartesian (task) space in which it is easier to define tasks and the motion of the manipulator. Furthermore, the generation of a collision-free trajectory is also simplified when working in task space. On the other side, working on joint space can avoid problems with singularities and requires less computation. Either case, before sending the commands to the manipulator joints, the Cartesian trajectory needs to be converted to joint space.

In order to translate a certain Cartesian trajectory into a joint space trajectory, the inverse kinematics of the manipulator is used. This topic is out of the scope of this chapter, but two important things need to be kept in mind: the mapping between a Cartesian trajectory and the joint space trajectory is not unique and depends on the number of DOFs of the manipulator. That is, several joint space trajectories might yield the same Cartesian trajectory. In addition, there might be no solution to the inverse kinematics problem. That might happen because of singularities of the manipulator or because the target positions are out of its reachable workspace.

Given either a Cartesian trajectory or a joint space trajectory, it is not possible to store all the trajectory points in memory and it might not be possible to find an analytical expression to describe it. The usual procedure is then to store the initial, final, and intermediate via points of the trajectory through which the manipulator needs to pass. The conversion between this set of discrete points and the continuous trajectory points to be sent to the robot is what is called *trajectory interpolation*.

The simplest case is the use of a linear interpolator. Given two known points of the trajectory (and its associated time stamps), a linear interpolator draws a straight line between them. That function can help find intermediate points between the two initial known points. This is a very simple method to compute and ensure continuity in the position. However, it does not come for free: the velocity is kept constant between two successive points, which might create abrupt changes on the velocities between sets of adjacent points and, in theory, it might require infinite values for the acceleration on those changes. This is one of the common reasons for jerky robot movements.

In order to avoid those problems, the solution can be to use higher-order polynomial functions to represent the trajectory. A usual choice is a cubic interpolator, which uses a third-order polynomial; that is, four coefficients are available to describe two adjacent points. In other words, by using a cubic interpolator, there is the possibility to specify four constraints that determine the cubic trajectory which are usually the initial and final positions (as the linear interpolator) plus the initial and final velocities. Equation (5.26) describes a cubic polynomial:

$$\theta(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \tag{5.26}$$

where a_0, a_1, a_2, and a_3 are the four coefficients that specify the cubic trajectory. As an example, given boundary conditions set to zero, that is, initial position and velocity ($\theta(0)$ and $\dot{\theta}(0)$) as well as final position and velocity ($\theta(t_f)$ and $\theta(\dot{t}_f)(0)$) defined as zero, and solving Eq. (5.26) with the previous boundary conditions, we

obtain the values for the four unknown coefficients as

$$a_0 = \theta_0 \qquad a_1 = 0$$
$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) \qquad a_3 = \frac{2}{t_f^3}(\theta_f - \theta_0)$$

Thus, the cubic trajectory at time t is defined as

$$\theta(t) = \theta_0 + \frac{3}{t_f^2}(\theta_f - \theta_0) \cdot t^2 + \frac{2}{t_f^3}(\theta_f - \theta_0) \cdot t^3 \tag{5.27}$$

Figure 5.19 shows an example of using Eq. (5.27) with the input parameters: initial position $\theta(0) = 0$, initial velocity $\dot{\theta}(0) = 0$, final position $\theta(t_f) = 10$ (degrees), and final velocity $\theta(t_f) = 0$ in a total time of 5 s. Notice that the slope at the beginning and end of the curve is zero, corresponding to the given initial and final velocities as zero.

By using such a cubic trajectory, the acceleration between sets of two adjacent points is kept linear. This might cause problems in some cases, as real robots have upper limits on the maximum accelerations (i.e., torques) that can achieved. In that case, other interpolators can be used that keep the acceleration constant or under a certain limit. For instance, a linear interpolator with parabolic blends can be used. This interpolator is divided in three parts: a linear part (middle part of the trajectory) and quadratic functions that defined the beginning and the end of the trajectory. The interpolator thus *blends* from quadratic to linear and from linear to quadratic to keep the acceleration constant. Similarly, a minimum-time trajectory interpolator can be used, which limits the maximum acceleration to a desired value and in addition create a trajectory from one point to the next in the minimum time possible [53].

5.3.2.2 On-Line Trajectory Generation

The aforementioned interpolators are *offline* computed trajectories, that is, they cannot be easily modified during its execution. Sometimes though, it is interesting to have trajectories that can be slightly (or majorly) adapted during the robot's

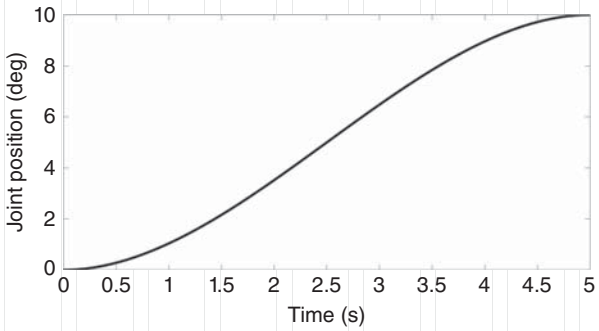


Figure 5.19 Example of a trajectory interpolation between two points using a cubic polynomial.

motion to deal with a dynamic environment or react to a certain sensor event [54, 55].

One recent example of such online trajectory generation algorithms is the *Reflexxes* software library [56]. This library can compute jerk-limited motions within one control cycle so that robots can almost instantaneously react to (unexpected) sensor events. Since the computations are performed very fast (at around 1 ms), a new time-optimal state of motion can be calculated for the same control cycle given an arbitrary initial state of motion and the kinematic constraints in order to reach a desired target state of motion. The output values can be directly sent to the low-level controllers thus achieving very reactive motions.

5.3.3 Collision Avoidance

Collision avoidance has been traditionally associated with high-level control, especially in the area of path planning, which aims at finding collision-free trajectories (i.e., guarantee that the robot manipulator reach the target pose without colliding with the environment). The low-level control described in Section 5.3.4.1 executes those collision-free trajectories.

However, it is obvious that, in dynamic environments, objects might change their position or obstacles might come into the path of the manipulator between the time the robot is planning a trajectory and the moment that trajectory is finally being executed. Ideally then, the low-level control layer should also be able to react in real-time to possible collisions. One option has been already seen in the previous Section 5.3.2.2: the connection between sensor feedback and low-level controllers allows to react instantaneously to sensor events. In this case, that could be, for example, a certain unexpected force measured on the end-effector.

The seminal work in this area was presented in Ref. [57]. In this article, real-time obstacle avoidance at low-level control is described based on the concept of artificial potential fields. The idea behind the potential field forces is that the target position is an attractive pole for the end-effector and the obstacles on its way act as repulsive poles. In this case, the low-level control is able to avoid in real-time known obstacles coming into the path of the moving manipulator. However, as a local method, it might fall into a local minima, and it did not consider collisions with the robot itself, apart of the possibility of including joint limits. In Reference [58], a real-time collision avoidance method was also presented which used a spring-damper model to generate virtual forces when the arm entered an non-allowed area (i.e., obstacle). An outer force–control loop can keep those forces at zero by modifying the Cartesian position of the end-effector, thus *escaping* the collision. In Reference [59], a method for detecting and reacting in real-time to collisions was shown, which used the total energy and the generalized momentum of the robot to detect anomalies (i.e., collisions) and react fast by moving the robot away from the collision area. In Reference [60], a fast method to detect collisions between a robot and moving obstacles was presented. The method evaluates distances between the robot and obstacle directly from depth data of a Microsoft

Kinect sensor to generate repulsive vectors that disturb the trajectory being executed. The work emphasizes on fast perception of the distances to the obstacle to achieve a reactive system at the risk of being not necessarily accurate, because the rationale is control accuracy can be compensated by a fast reaction to an incoming collision.

5.3.3.1 Self-Collision Avoidance

The previous methods mostly aim at avoiding collisions between a robot and an obstacle, either static or moving. However, and especially for manipulators with many DOFs, mounted on an articulated torso, it is also necessary to provide with some methods that avoid that one or more of the links of the robot collide with parts of itself. One of the first works on this area was presented in Ref. [61], in which a geometrical approach using minimum distance measurements for convex hulls is used to detect interference between links. In Reference [62], the focus is on self-collision method for robots cooperating with humans. In this case, the self-collision method is based on representing the robot as elastic elements and adding two priority functions to also consider both task and environmental constraints during the self-collision avoidance motion. The detection of the contact between the elastic elements generates a virtual reaction force, which is used to control the robot motion. An interesting method is shown in Ref. [63], the so-called *skeleton* method for real-time self-collision avoidance. In this approach, a model of the whole robot (the skeleton) is used for analytical computation of the closest points to a collision along the skeleton, generating repulsing forces via potential fields, and computing the required torque commands via the transpose Jacobian which are added to the driving joint torques. In Reference [64], a new repulsion method based on potential fields for self-collision avoidance is presented, which extends the previous work [63]. In this case, a damping is added to avoid critical situations that caused instabilities and oscillations in the previous method. Moreover, a mechanism is also included that determines whether the collision is avoidable via the repulsion method or, instead, if the braking should be activated. The collision model is based on the concept of swept volumes as referenced later in this paragraph [65]. In Reference [66], a self-collision velocity-based controller is proposed with a new distance computation making use of patches of spheres and toruses, which aim at ensuring a continuous gradient, that is, a smooth commanded trajectory. Another interesting method is also presented in Ref. [65]. The algorithm checks pairwise for collisions making use of swept volumes of all body parts. These swept volumes are represented as usual as convex hulls but, in this case, being extended by a buffer radius for an efficient and numerically stable computation, which could run in hard real-time conditions. The algorithm takes joint angles and velocities as input and computes whether (and when) a braking action should start in order to have time to stop the robot before a collision occurs. The algorithm requires two models: a kinematic model of the robot and a geometrical model with the robot rigid bodies (collision model). In References [5, 67], the self-collision mechanism used at the Mars Rovers Opportunity and Spirit to avoid collisions between the manipulator and the rover is described.

Q8

Those robot arms receive a sequence of planned commands from Earth and prior to the execution of the movement, each via point both in Cartesian and joint space are checked for collisions. The algorithm determines whether there are intersections between the geometric models of the arm and the rover (and possible payload). The collision checking is also used as part of the ground validation of the commands that are sent to the rover. In addition, the ground validation also includes the detection of collisions with the terrain.

5.3.4 High-Level Control Strategies

The high-level control is to plan the path for the end-effector of the manipulator, whereby different strategies refer to the LoA of the manipulator operation. As introduced in Section 1.2, the LoA for any spacecraft or space system's onboard operation has been defined by the European Cooperation for Space Standardization (ECSS) in four levels (E1–E4). A planetary robotic manipulator (being a space system) can, therefore, be designed to operate in all four levels, for example, under complete user control (or teleoperation) known as the E1 LoA, under semi-autonomy known as the E2 or E3 LoA, and full autonomy known as the E4 LoA.

5.3.4.1 Path Planning

In general, path planning determines a path, connecting an initial and final desired configuration of the manipulator, while avoiding the possible obstacles on the way. Depending on the LoA being used, the robotic manipulator has corresponding level of onboard capability to find and choose which trajectories to use in order to reach the target end-effector positions.

The planning of a motion for manipulation has been historically divided into a phase to reason about the manipulator path at a global scale (or called *gross motion planning*), and a phase to deal with uncertainties, forces, friction, and alike that appear when the robot contacts the environment (or called *fine planning*) [14]. Automatic motion planning is realized by exploring the configuration space using probabilistic algorithms to search for a collision-free path that would connect start and goal configurations [68]. Within that area, sampling-based planners are a group of widely used methods, which use collision avoidance algorithms to search for robot configurations that are collision-free. Such planners are not complete as they cannot guarantee a solution in a finite amount of time, but provide a weaker form of completeness: given enough time and if a solution exists, the planner will eventually find it. A typical classification sorts the algorithms into multi-query approaches and single-query approaches. The algorithms in the first group construct a roadmap once that can subsequently accept multiple queries. The best examples are probabilistic roadmap methods (PRM) [69]. The algorithms of the second group are built online each time a query is made. The best proponents of this group are rapidly exploring random tree (RRT) algorithms [70]. Alternatively, the *potential field*-based techniques have been proposed to solve specific problems in an efficient way [71]. This methodology is derived from methods used in

Q9

obstacle avoidance in which there is no explicit roadmap to be constructed but a real-valued function represented by potential fields that can guide the robot to the goal configuration. The potential field is constructed based on the *attraction primitive field* that attracts the robot to its goal position and *repulsive primitive field* that pushes the robot away from obstacles.

There are also methods that do not use automatic motion planning. One example is the central pattern generators (CPGs), which are used for generating rhythmic motion. These rhythmic patterns have been used to perform manipulation tasks such as hammering, sawing, or playing drums [72, 73] or for multifingered grasping [74]. Learning approaches have also been developed for manipulation and grasping [75]. In this area, learning by demonstration [76, 77] and imitation learning [78] are regarded as appropriate methods for learning, as humanoid robots have similar kinematics to humans and are primarily designed to solve human tasks.

The MERs (Spirit and Opportunity) combined ground-based motion planning sequences (i.e., LoA E1) with autonomous onboard motion planning (i.e., LoA E4) [5, 79, 80]. The E4 motion planning is only used for the rover navigation on board, while the robotic arm motion planning is performed on ground at E1 and the onboard software only checks for collisions between arm and rover before executing the trajectories. For the Mars Science Laboratory rover (Curiosity)'s first 7 months' operation of the robotic, the creation and validation of the arm motion control sequences are responsibility of the rover planners after a plan has been approved by the ground control station [81], which corresponds to the LoA E2 or E3.

5.3.4.2 Telemanipulation

The case of controlling a manipulator through teleoperation by human operators is often called *telemanipulation*. In telemanipulation, the human operator controls all the movements of the manipulator remotely. In the case that the manipulator is programmable and able to adapt to the environment by using its own sensors so that the human operator only needs to communicate intermittently with the manipulator to supervise it, the operation is also called *telerobotics*.

In teleoperation, one key element is feeding back the sensor information from the remote system back to the human operator. The simplest way of achieving this is to deploy a number of cameras in the remote area. In fine manipulation tasks, the availability of force feedback from the remote manipulator is useful to simplify the job of the human operator. In terms of the control, the manipulator in telemanipulation is known as the "slave" system and the device used by the human operator is known as the "master" system.

Unilateral Control In the simplest teleoperation, the control strategy used is called unilateral control whereby there is no feedback from the "slave" back to the "master" and thus the "master" does not require to be equipped with motor units but just with encoders to measure the joint positions. Such a "master" is usually called a "passive" master. The name unilateral control comes from the fact that control

is only performed in one direction, that is, from the master to the slave. The master generates the reference signals (position or velocity), which are sent to the low-level controllers of the "slave." The "slave" or the manipulator receives the reference signals from a remote location instead of from a local control computer. The control of a teleoperated slave manipulator can be much simpler than the control of a typical industrial manipulator since there is no need for computation of inverse kinematics or dynamics.

Bilateral Control In order to increase the performance of the teleoperated system and ease the job of the human operator, it is beneficial to send force sensor information from the remote manipulator back to the operator. Such systems allow bilateral control since the reference signals flow in two directions, that is, the "master" sends position or velocity reference signals to the "slave," and the "slave" transmits forces to the operator that are applied to his/her hand or the arm. To achieve the bilateral control, the "master" needs to be equipped with actuators.

Basically, there are two bilateral control schemes based on the variables used for the manipulator control:

- Position–position control: This control scheme was first proposed and applied by Goertz and Thompson [82] and has been extensively used since then. The idea behind is to control the master system in a local position control loop whose position reference is the current position of the slave system as shown in Figure 5.20a. The slave system is also in position control mode, with its reference signal being the current position of the master system. Thus, this scheme is completely symmetric in terms of the control.
- Force–position control: This is probably the most popular control scheme in telemanipulation. Similar to the position–position scheme, the slave is controlled in position by taking as reference signal the current position of the master. However, the master receives a sensory feedback from the force sensors on the slave, which allows the operator to feel the external forces acting on the slave as shown in Figure 5.20b.

Performance Parameters There are a number of parameters that allow to assess the performance of the teleoperation system, as described earlier:

- Stability: probably the most important criterion to take into account. Bilateral control schemes tend to be unstable, especially when the slave is in contact with stiff environments; for example, a small change in position might translate into large contact forces. Another major source of instability is due to the delays in the communication between the master and slave.
- Error: in position and force between the master and slave.
- Force feedback ratio: defined as the relation between the reaction force between the slave and the environment and the force generated by the master in order to reflect that force to the operator. The ratio needs to be selected so that it matches the capabilities of the human operator to avoid fatigue.

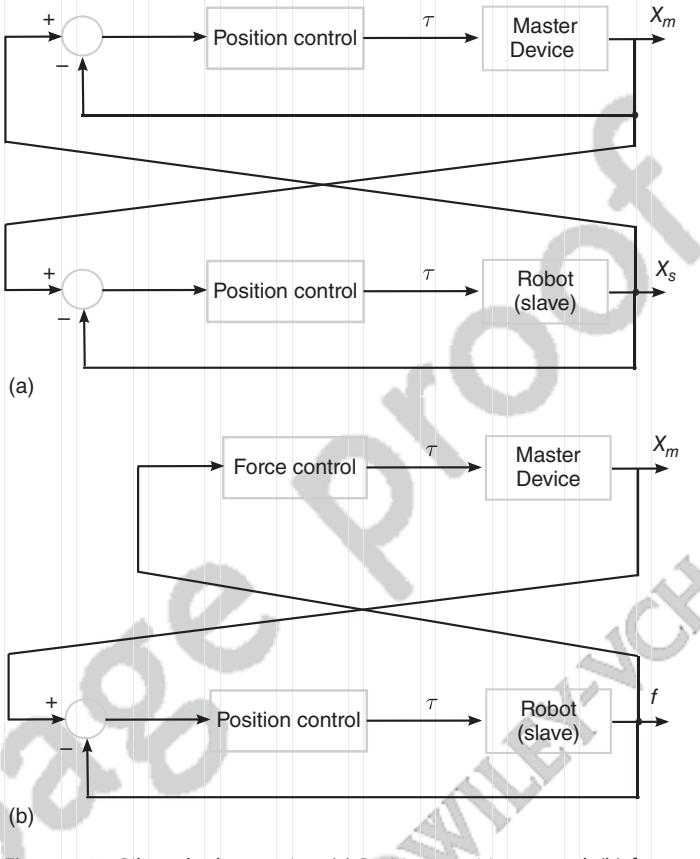


Figure 5.20 Bilateral teleoperation. (a) Position–position control, (b) force–position control.

- Transparency: to achieve the goal of transmitting position and forces between master and slave in such a way that the operator feels perfectly coupled with the environment. In other words, the teleoperation system (master, control, and slave) behaves "transparently" between the operator and the remote area.

The first teleoperated system in space was the experiment ROTEX (RObot Technology EXperiment) [83], developed by the German Aerospace Center (DLR) and performed on board the International Space Station (ISS) in 1993. The robotic "slave" system was a small 6-DOF robot with a length of around 1 m equipped with 6-DOF force/torque wrist sensor, tactile sensors, and stereo cameras, among others. The robot was controlled in several modes: teleoperated from the ISS by the astronauts using a TV monitor (i.e., E1), teleoperated from the ground (i.e., E1), or follow preprogrammed control commands given by the ground (i.e., E2). The highlight of the control was a multi-local sensory feedback allowing higher LoA for the robot, as well as the use of predictive "displays" or

simulations to cope with the transmission delays of up to 7 s when using the teleoperation from the ground [84]. In a subsequent mission called ROKVISS [85], a two-joint robotic manipulator was installed outside the ISS in January 2005, which could be teleoperated from the ground via a direct communication link. Because of the direct link, the communication delay was only around 20 ms, which allowed the use of a bilateral control strategy using force feedback from Earth. Due to the fact that the communication window with the robotic arm was limited to a maximum of 7 min, the system also had an E2 operation mode. In January 2015, a NASA astronaut teleoperated from the ISS a device located at the ESA Telerobotics Laboratory on Earth by using a body-mounted joystick with force feedback capabilities. The focus of the experiment was to understand the human capabilities for manipulation and to apply forces in a weightless environment [86].

5.3.4.3 Higher Autonomy (E2–E4)

In case of planetary missions such as on Mars, the typical time delay of communication often impedes the usage of a suitable direct teleoperation of the robot. This requires the robots to have higher LoA beyond E1 to complete tasks on their own. In the literature of manipulator high-level control, various strategies (particularly representing LoA E2 and E3) have been developed, named as, for example, the supervisory, shared, or traded control. In all cases, the mission is completed by combining teleoperation and onboard sensory feedback of the manipulator hence involving operations of different LoA.

Supervisory Control or E2 LoA In the supervisory control [87], the operator specifies the high-level plans and initiates them, whereas the robot carries out the tasks in an automated fashion. As aforementioned, in various Mars missions (containing the Spirit, Opportunity, and Curiosity rovers) the motion planning for the onboard robotic arm was performed on ground, and validated and executed by the arm on itself [81, 88, 89].

Q11

Shared Control or E3 LoA In the shared control [90, 91], the robotic system and the human operator work on the same task together. In this scheme, the human operator generates gross motions, which then get rectified by the robot using local sensory feedback. In case of no delay, this feedback gets sent to the operator as well. In case of major delays, the operator and robot work simultaneously but on different parts of the tasks. An example is the concept of virtual fixtures, that is, virtual surfaces or objects that are superimposed to the visual and force feedback of the operator. They act as "active constraints" in the sense that they limit the motion of the robot in order to ease the task and avoid false or dangerous movements. Imagine the analogy of controlling the mouse pointer on the computer screen: the remote system (pointer) is controlled by the master (mouse) anywhere inside the computer screen but cannot leave the limits imposed by the screen frame (virtual fixture). Thus, the user can focus on the task without worrying about the pointer going outside the limits or into restricted areas.

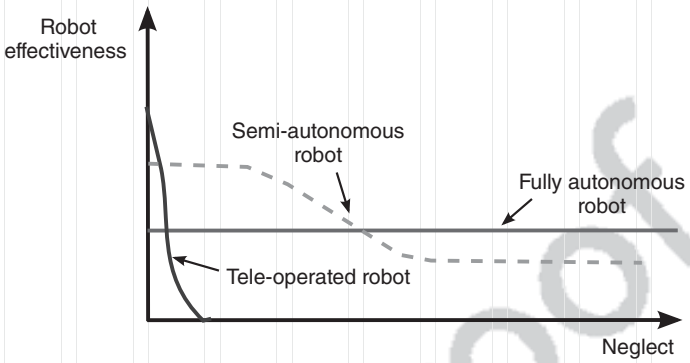


Figure 5.21 The neglect curve. (Reproduced from Ref. [95].)

Traded Control or E4 LoA In the traded control [92, 93], the human operator and the robotic system also work on the same task but at different times. In this control scheme, the robot mostly performs the task autonomously but at certain times the human can intervene to complete a specific part of the task, often requested by the robot itself (e.g., after the robot identifying a unknown situation or task).

Designing LoA It is clear that a robotic manipulator can be operated in a range or a combination of LoA based on mission constraints and task requirements. The following system-level design of LoA has been investigated:

- adaptive autonomy, if the robotic system has exclusive control over the determination of the LoA;
- adjustable autonomy, if the user can adjust the LoA;
- mixed initiative, where both user and robot have equal responsibility on the decision of choosing the LoA [94].

There exists a certain relation between the robot's effectiveness of performing a task and the level of attention or stress of the operator (indirectly related to the LoA) [95], as shown by the so-called neglect curve in Figure 5.21. While teleoperated systems achieve higher level of effectiveness if the user is entirely focused on the task, the effectiveness abruptly drops if the operator loses attention (i.e., "neglects" the system). On the other hand, a fully autonomous robotic system might have lower effectiveness but that level can remain constant independent of the level of attention of the operator. A semi-autonomous system is expected to perform somewhere in the midway, with relative high level of effectiveness even when the level of attention from the operator is relatively low.

5.4 Testing and Validation

Testing and validation are critical to the design and implementation of a planetary robotic arm. They represent important steps in the design process to finalize the

control scheme, to test the operation of the arm in the representative environmental conditions and to validate the design performance. The arm design and testing activities are inherently linked and form part of the underlying development and testing strategy established at the beginning of the project, balancing the need for a realistic yet implementable test scenario.

5.4.1 Testing Strategies

As detailed in Section 2.4, the robotic systems for planetary exploration are expected to face a variety of environmental conditions, for example, air or airless extraterrestrial bodies, extreme cold or hot temperature. The robotic systems can also experience a range of gravity, from low-gravity found on asteroids or small moons to major gravity fields around planets such as Mars and Venus. The robotic manipulator system considers gravity as a key environmental factor that influences its design and testing philosophy. Three testing strategies are available to count for the gravity effect, each presenting its own engineering and system challenges:

- Design to target gravity: This strategy involves design of the arm to sustain in the gravity field of the target planet, which is likely to provide the smallest mass for the arm. However, this approach makes testing more challenging due to the need to offload the arm during the validation process. The offloading setup can use a range of configurations such as a mass pulley system or balloons that offset the weight of the arm to mimic the target gravity. However, such setups inherently constrain some aspects of the testing, mainly because the arm goes through a range of poses that may not be compatible with the offloading mechanism.
- Self-supporting arm with scaled mass payload: This strategy involves the design of an arm that can be tested on normal Earth conditions (or at 1 g) but carries a payload mass scaled to the target gravity. For example, an arm designed to work on Mars can be tested on Earth with a payload mock-up a third of its mass. This approach provides a higher mass solution compared with the previous strategy, but allows a wider range of testing solutions, thanks to the elimination of the gravity offloading rig. The use of typical design margins can to some extent bridge the gap between the two gravity environments. For example, a Martian arm could be designed with a motorization actor of 2 (i.e., a joint torque capability is doubled to account for various inefficiencies and contingencies in the system) can be allowed to work on Earth without the margin, limiting the impact on the arm design (especially the resulting mass).
- Full arm mass and payload: This strategy sees the design of the arm to be fully compatible with terrestrial testing with its actual full mass payload. This approach produces the highest mass by sizing the arm and its payload for the Earth

gravity field (not for the target gravity), resulting in significantly oversized joints. The operation of the actual payload would provide the mission operators a powerful tool to validate the operation and enable realistic rehearsal of the payload activities. However, the tuning of the arm control loop will be significantly affected by the dynamics of the arm. The higher gravity on Earth does not only have a damping influence on arm and joint dynamics but also has an effect on the flexure of the arm and joints, unlike what is expected to be experienced in the actual extraterrestrial environment.

Each strategy presented earlier has its pros and cons, affecting mainly the overall system mass as well as the realism of the testing. The "self-supporting arm with scaled payload" is one that provides a good middle ground, providing enough scope or testing on Earth as well as minimizing the extra mass required. Most planetary missions that used a robotic arm had selected this strategy, from the small arm on Beagle 2 lander to the long arms of the Phoenix and InSight lander (see Figure 5.22) where the benefits are even greater.

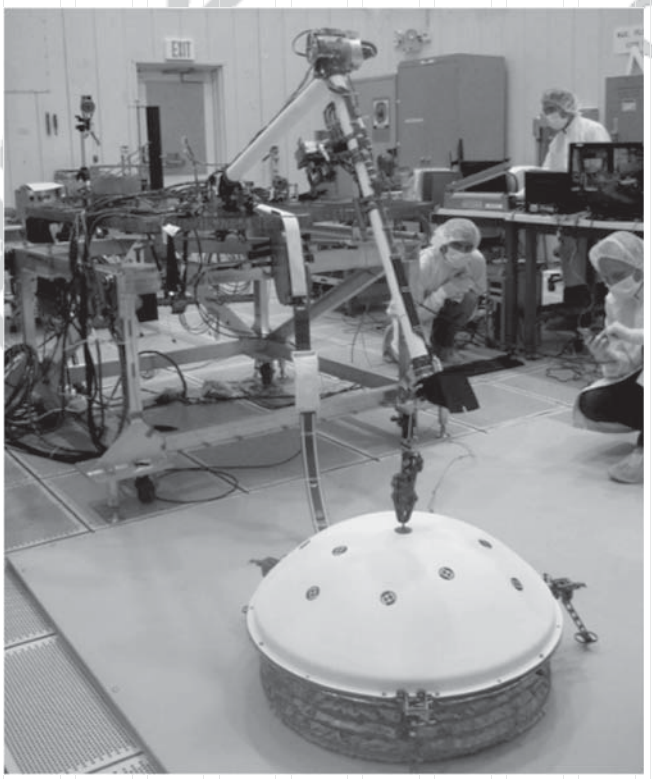


Figure 5.22 InSight testing with scaled-mass payload.

5.4.2 Scope of Testing Activities

The purpose of a planetary arm can vary from one mission to the next; however, it will invariably involve the precise deployment of the end-effector at specific locations. Whether it needs to place instruments or tools to the target locations or retrieve samples from the surface before delivering them precisely to the onboard instruments for processing, the robot must be able to reliably and constantly perform its tasks.

The purpose of the testing is twofold: (i) to physically characterize the manipulator to perform a calibration of its control and (ii) to assess the performance to execute the task with the appropriate resolution, repeatability, and accuracy which are defined as follows:

- *Resolution* is the smallest incremental move the manipulator can produce.
- *Repeatability* is the measure of the ability that the robot can move back to a pose (position and orientation).
- *Accuracy* is the ability of the manipulator to precisely reach a pose in 3D space.
- In addition, with the motion of the arm comes the concept of *dynamic accuracy and repeatability* that capture the ability of the manipulator to follow a prescribed trajectory.

As discussed by Roth and Mooring [96, 97], three levels of calibration for robotic arms are used, namely the joint level, kinematic model level, and dynamic model level.

- The joint calibration focuses on the determination of the correct relationship between the signal of the joint sensors and the actual joint displacement.
- The kinematic calibration focuses on the physical characterization of the arm to minimize the impact of geometric errors. These are the errors in the parameters that define the geometric relationship between the axes of motion. These stem from the manufacturing and machining errors resulting from machining tolerances, link length, and joint orientation. Due to the machining techniques, these can be minimized, but are intrinsically unavoidable. The testing will, therefore, result in a basic kinematic model on the arm with the correct joint-angle relationships.
- The dynamic calibration finally focuses on the remaining nongeometric (nonkinematic) errors introduced by the inherent joint compliance, gear backlash, friction, and link flexion.

While the joint calibration can be performed with a minimal setup, the kinematic and dynamic calibrations require more extensive testing scheme. The calibration of manipulators is a complex operation and a topic investigated extensively for industrial manipulators. References such as [97] provide thorough descriptions of the process, both functionally and mathematically. Given the significance of kinematic calibration to the planetary robotic arm, it is discussed further in this section.

5.4.2.1 Kinematic Calibration

Once the joint calibration has been performed, the kinematic calibration is followed and concerned with the minimization of the error between the actual pose and its intended target pose. Once built the arm physical implementation is often fixed and cannot be changed, but these errors can be mitigated through software. The process for a kinematic model-based calibration is implemented in four discrete steps [97]:

1) Modeling: A kinematic model is generated to provide a mathematical representation of a perfect manipulator. The Denavit–Hartenberg or other methods can be used to describe the properties and relationship between the arm links and joints.
2) Measurement: Moving onto the real hardware, this activity focuses on the gathering of the necessary data that will be used to measure errors and ultimately help with their compensation. To this end, the manipulator is exercised across its workspace and the actual position of the tool is compared with the position predicted by the theoretical model. To perform accurate measurement of the end-effector position, a number of metrology methods can be used including single-point or complete pose measurements. Single-point measurement, as the term indicates, measures precisely the location of a single point and can implement theodolites, laser interferometers, acoustic sensors, or coordinate measurement machines (CMMs). The measurement process is lengthy and time consuming to change the pose of the arm and measure its outcome. In recent years, complete pose measurements have been made easier though the use of optical systems such as a marker/camera setup that can resolve the position of the markers precisely. These systems can be implemented in a number of ways including, the use of typical cameras and fiducial markers, IR cameras and active IR illumination of reflective markers, or IR cameras and active IR markers. These systems have seen significant development and improvement in recent years both in terms of frequency and accuracy driven by the needs of the film and entertainment industries as a motion capture (MoCap) system. By capturing the position of a wide range of points accurately at an appropriate frequency, these systems provide not only valuable data for the kinematic calibration but also to the dynamic calibration, allowing the recording of dynamic events such as trajectories, vibration, and the deflections of limbs.
3) Parameter identification: To bridge the gap between an idealized model and the test data, parameter identification focuses on the quantification of the kinematic parameters that cause the computed pose to match as closely as possible the test data. It is in essence an optimization problem that can involve a number of methods, pending on the nature of the errors or the model (e.g., deterministic/stochastic, linear/nonlinear), and typically involve the use of least mean square techniques. (See Ref. [97] for more in-depth discussion.)
4) Implementation of compensation: Once the kinematic parameters have been identified, the accuracy of the manipulator is improved by the implementation of the new model into the controller. The previous steps of modeling,

measurement and parameters identification address the propagation of errors from joint to end-effector pose in the forward model, addressing therefore the forward calibration of the arm. The implementation phase is, therefore, critical to improve the calibration of the inverse model and, therefore, provide inverse calibration.

5.4.2.2 Beyond Calibration

Calibration in a laboratory environment can provide a good model for a newly built robotic arm as it is then fit onto its deployment platform. Nevertheless, a range of environmental and mechanical aspects need to be taken into consideration throughout the lifetime of the robot.

- Launch and landing: During launch and landing, the system is subjected to a range of violent mechanical loads including vibration shocks through pyro actuation, parachute opening, and landing, and so on. These may affect the mechanical alignment of the structure or elements of the joints.
- Operation: Operation on a robotic platform (such as a rover) may see a range of shocks due to the platform overcoming rocks and falling from them. Across its lifetime, wear of the gears and bearing, and possibly the seepage of dust into the mechanisms can affect the properties of each joint in a different way, affecting the internal friction or the accuracy of the positioning.
- Thermal environment: The operation of the arm across a wide temperature range needs to take into account the coefficient of thermal expansion (CTE) of its constituting elements. As materials get warmed up and cooled down, they contract and expand, which potentially affect the accuracy of the end-effector placement through displacement or twisting. To mitigate these aspects, the CTE of various key elements would be matched to minimize local stresses and deformations across the temperature range.

To monitor and compensate for these errors over the mission lifetime, manipulators can be dispatched to dedicated known locations on the body of the lander or rover to assess the errors in the system. Through the recording of the final end-effector pose, an *in situ* measurement step, the kinematic parameters can be updated to correct these errors and improve the overall accuracy of the manipulator. The future missions are envisaged to implement some level of visual servoing, making the precise calibration of the arm not as critical as before because the arm is controlled either by a camera overseeing the operation or placed on the end-effector (i.e., eye in hand).

5.4.3 Validation Methods

Lack of detailed measurement data and operational experience directly from planetary environment stays behind the difficulties in defining the key parameters of robots that should be evaluated. Moreover, it is even in some cases difficult to answer the question of where these parameters should be tested: here on Earth

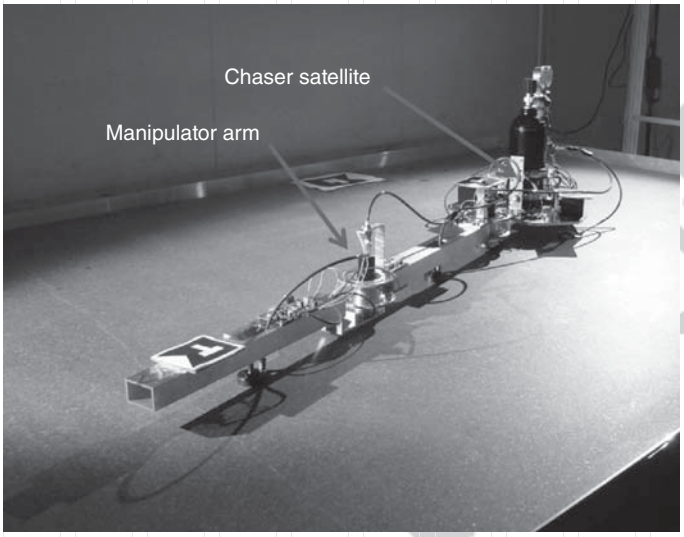
Trim Size: 170mm x 244mm Gao c05.tex V1 - March 30, 2016 8:41pm Page 300

or rather at the destination? Having in mind these uncertainties and based on the current knowledge, the following possible phenomena need to be measured before the operations of the robotic arm are defined:

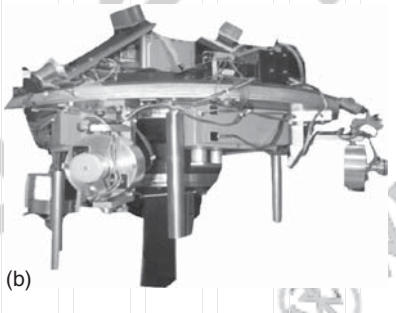
1) Due to reduced gravity, the lander or rover is weakly connected to the ground. The manipulator arm motion can generate reactions that are too high to be transferred to the ground. This effect might be destructive to the lander or rover's stability.
2) Due to relatively low stiffness on both the structure and the joints of the arm, the frequency characteristics might be a good starting point to evaluate the potential influence of robotic arms on other subsystems (e.g., solar panels or lander gear).
3) In low-gravity environment, the predominant load of manipulator joints comes from the dynamics of motion. Therefore a situation, where the torque crosses zero point and backlash significantly affects the arm operations, may often occur. This effect can be enhanced by specific tribological behavior of mechanisms in vacuum.
4) Exploration of planets often includes regolith sampling processes, which is typically a combined task of manipulator arm and sampling mechanism. Due to unknown regolith parameters *a priori*, the interactions between the robotic arm and the regolith should be tested.

Various methods can be used to simulate some of the effects indicated earlier:

- One possible approach to simulate near-zero gravity conditions is to perform experiments during *parabolic flights*. During such a flight, an aircraft flies a trajectory that provides about 25 s of free fall.
- The other possibility is to perform tests using *drop towers* where the maximal time of experiment is much shorter (in most facilities free fall lasts only up to 10 s, and residual gravity acceleration is around $10^{-3} - 10^{-6}$.
- In case of reduced but nonzero gravity conditions, the use of planar *air-bearing tables (ABTs)* with movable platform might be a possible choice. In such an approach, the test objects (e.g., a manipulator intended for sampling operations) are mounted on a planar ABT that allows for almost frictionless motion on a flat surface that is inclined to simulate a certain level of gravity. In effect, the reduced gravity conditions act properly on the mechanisms in two transnational and one rotational DOF, and the residual gravity acceleration is around $10^{-3} - 10^{-5}$ depending on the type of table's surface and other parameters. Therefore, disturbances experienced using ABT are smaller compared with using parabolic flights and most drop towers. Depending on the size of the air container, duration of experiments can last at least several minutes and even up to many tens of minutes, thus significantly longer than the previous two options. The main disadvantage of using ABT is the limited number of DOFs. As indicated, the motion can be performed without disturbances in one plane only with two translational DOFs and one rotational DOF. Special designs allow additional rotational DOF by taking advantage of spherical air-bearings, but the residual



(a)



(b)

Figure 5.23 Two types of ABT configurations. (a) Flat ABT testbed simulating the chaser spacecraft during capture maneuver with a robotic arm. (Courtesy CBK PAN.) (b) Round ABT testbed simulating frictionless motion around the roll, yaw, and pitch axis. (Courtesy Surrey Space Centre.)

gravity impact is then higher and only the first-order influence can be analyzed. An example of a flat ABT, developed originally to test orbital robots working in space, is illustrated in Figure 5.23a [98, 99]. The size of table (2×3 m) allows for performing complicated maneuvers such as rendezvous maneuvers, final docking, or detumbling of a gathered target [100]. An example of a round ABT, developed typically for testing spacecraft attitude control, is illustrated in Figure 5.23b [101].

5.4.3.1 Use of ABTs

The low-gravity planetary application of the ABTs can be achieved by inclining the table by different degrees depending on the target bodies (e.g., for the Moon is 9°, and for Mars' moon Phobos is just 0.05°). To simulate and analyze the

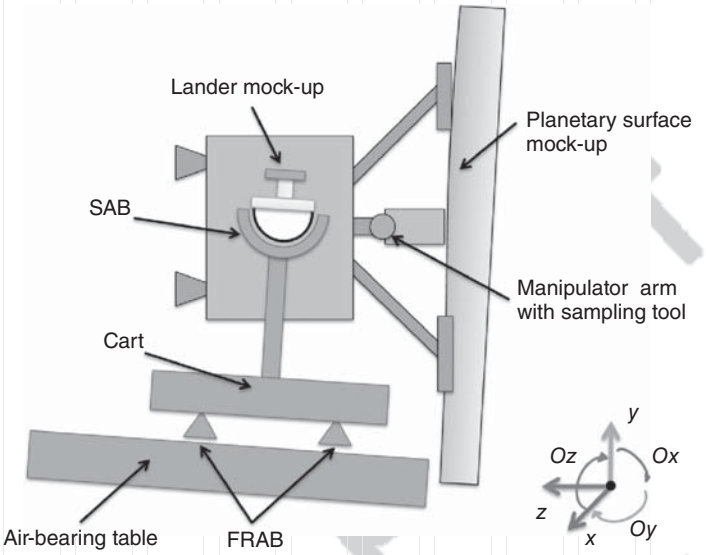


Figure 5.24 A lander mock-up installed through a spherical air bearing (SAB) on a cart moving on an inclined flat granite table with air bearings (FRAB), and the lander legs are touching a vertically oriented planetary surface mock-up. (Courtesy CBK PAN.)

manipulator's behaviors in such planetary condition, a planar ABT testbed can be set up using the granite surface as illustrated in Figure 5.24. The lander or rover mock-up can stand on the cart, which produces air against the granite table for frictionless movements. The robotic arm on board the lander or rover can be installed on the side. The arm can be tested to measure the manipulator and lander interactions including forces, momentum transfer, or induced lander motion. Given different mission scenarios, additional payloads can be added for simulation, for example, penetrometers, sampling tools or a regolith surface mock-up [102].

Analysis of similarities and differences between the ABT and real environment on a low-gravity body is crucial since not all measured parameters are realistically reproduced by the testbed. In low-gravity planetary conditions, an unanchored lander is free to move with 3 DOFs and rotate with 3 DOFs. In this description as shown in Figure 5.25, the gravity vector is along $-z$ axis and the regolith local normal to the surface is along +z axis. The lander motion in $-z$ axis is limited by interactions with the planetary surface. The loads acting on the lander are generated by manipulator dynamics and may create lander's motion on the surface, and in some specific cases the lander rebounds or even escapes from planetary surface. If the manipulator arm's end-effector interacts with the regolith, the static or quasistatic reactions may appear in all directions.

It is worth mentioning that interactions between sampling tools, drills or penetrometers with the regolith can create forces and torques through the manipulator arm, which in turn leads to the lander motion with respect to the sampling

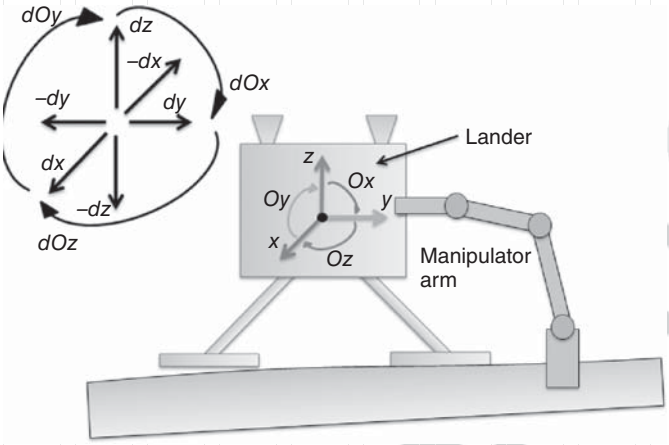


Figure 5.25 A possible configuration of a lander and a manipulator arm on planetary surface. (Courtesy CBK PAN.)

tool. This can have major impact on the manipulator itself, since its last joint will be the first one with all related consequences such as loads increase and control issues.

As aforementioned, the motion of the cart with lander mock-up shown in Figure 5.24 is limited by the flat configuration of the ABT, where both main motion of subsystems and generated forces/torques also have flat nature. For example, the test configuration of mock-up lander, manipulator arm, or other devices should be chosen in such a way that the force generated creates a motion not affected by the Earth gravity. As shown in Figure 5.26, the forces F_x and F_z generate lander's motion in x, z and around Oy direction, all of which are not affected by the Earth gravity. In another case where sampling tool generates the torque M_z and force F_z, the induced motion for the lander mock-up generates linear motion in x and z direction as well as rotation around the Oz axis, whose impact can be analyzed taking advantage of a spherical air bearing (SAB). However, that type of motion is limited to a couple of degrees hence only first order effects can be analyzed.

In summary, the major advantage of using the ABT as a testbed to validate the planetary manipulator arm is related to the natural conservation of momentum, which is helpful to study the interactions between the robotic arm and its housing platform. The limitation of the flat ABT is the two dimensions used and the fact it does not allow a full study of elasticity or vibration phenomena and joint friction, which is significantly reduced in the 2D "world."

The ABT testbed is typically used for testing spacecraft's attitude control systems or for validating dynamical models and numerical simulations. In addition to testing algorithms and concepts, flight hardware and engineering models of sensors and instruments can also be performed. Typical applications that these experiments are related to include rendezvous maneuvers, contact dynamics during

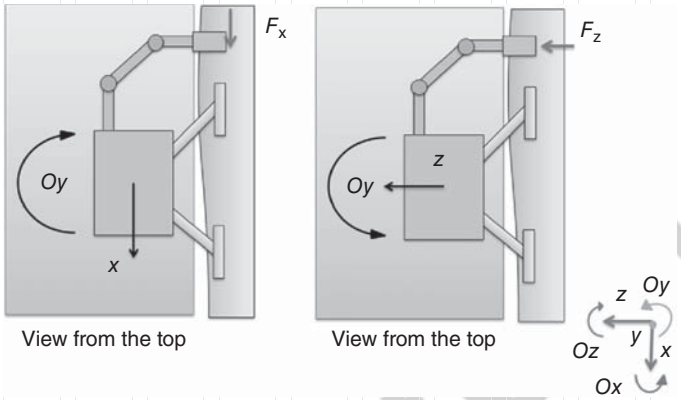


Figure 5.26 An ABT testbed allowing tests of sampling tool interactions where all major components of motion induced by forces (F_x, F_z) are flat. (Courtesy CBK PAN.)

Q12 docking, proximity navigation, guidance and control, and landing for low-gravity bodies. It is generally a good practice or advisable to perform as many tests on Earth as possible through testbeds such as the ABTs, and if possible to perform further tests on orbit in genuine space environment (e.g., at the International Space Station or ISS).

5.5 Future Trends

This section aims to shed some light on potential capabilities of planetary manipulation robots in the future, hence has chosen to present a number of example technologies in areas such as the dual-arm manipulation, whole-body control, and mobile manipulation. Some of these named systems or technologies have been investigated on Earth for some years to support different terrestrial applications; for example, the mobile manipulation has recently emerged and been recognized as a research field of its own merit.

5.5.1 Dual-Arm Manipulation

Dual-arm manipulation is an increasingly important research area where the manipulation robot is equipped with two arms that cooperate with each other to perform tasks with dexterity similar to that of a human. Driven by the interests of making more humanoid robots, the two-arm manipulators are also deemed more effective while performing complex assembly tasks. This can be extremely relevant to future planetary missions whereby building infrastructure such as the lunar or Martian outposts for human permanent presence/habitation is envisaged.

When comes to planning and executing tasks using the dual-arm manipulation, coordination between the arms is crucial. For example, holding and transporting an object with both arms requires maintaining the relative pose (position and orientation) between the arms, which is usually known as *planning with constraints*. One of the first approaches for controlling the dual-arm system used a master–slave configuration by minimizing the error between the relative poses of both arms [103]. Another approach used a similar concept [104] with one robot as a leader and another as a follower. The follower tracks the motion of the leader while taking into account the constraints imposed by the relative pose to be maintained. Work by Hayati [105] is one of the first that tackled the challenge to simultaneously control the motion and force of a multiarm system, where the arm forces are split between the arms. The study in Ref. [106] developed a closed-chain dynamical model for a dual-arm system. More recent developments in this subject include [107] where a cooperative control scheme for a dual-arm system capable of transporting rigid objects was described. The proposed control scheme integrated force and position control as well as vibration minimization. Another relevant work presented in Ref. [108] proposed a decentralized control scheme, which allows a multiarm system to move a single object in a coordinated way without using geometrical relations between the arms. There have been some well-known developments focused on dual-arm robotic platforms, to name a few, the DLR's Justin robot as shown in Figure 5.27 where a framework to parallelize the planning and execution of bimanual operations has been proposed [109], or another German institute Karlsruhe Institute of Technology (KIT)'s humanoid ARMAR-III robot that offers a redundant two-arm manipulation system [110] and a solution for efficient motion planning and especially adaptation of those plans under consideration of the redundancy of the robot [111].

The dual-arm manipulation systems can be combined with additional mobility (such as on wheels or legs), hence leading to more powerful holistic control strategies. Such whole-body control systems represent promising technologies for future planetary missions following the footsteps of their terrestrial counterparts.

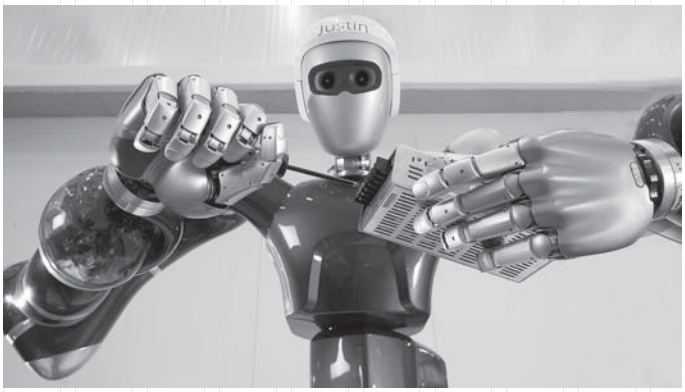


Figure 5.27 Justin robot. (Courtesy DLR, Creative Commons BY 3.0.)

5.5.2 Whole-Body Motion Control

Future, complex, and highly redundant robotic systems are expected to possess mobility (with locomotion of the wheels or legs, etc.) and manipulation capabilities, whereby these different capabilities should not be treated independently from each other. Hence, the holistic approach that considers control of a complex robotic system as a whole has been proposed and most popular in recent years. This approach is also known as the *whole-body control*. To give an example: in the case of a humanoid robot walking on two legs, balance has to be guaranteed at all times especially when the arms manipulate or contact the environment; the option of simply blocking the joints of the legs (and torso) to keep a fixed posture while manipulating would be a valid solution but a rather inefficient one.

Whole-body control frameworks operate in between single-joint controllers and possible high-level planners. These frameworks can take care of multiple and simultaneous control objectives, which is especially relevant for highly redundant robots as humanoids or mobile manipulators. Moreover, they make use of real-time feedback to control the robot. As a result, robots using whole-body control approaches are more adaptive and can react promptly to unexpected sensory feedback signals, resolving at run time for an optimal use of all the available DOFs.

The origin of whole-body motion generation comes from the generation of walking for humanoid robots while trying to ensure balance of the system, not yet considering manipulation or multiple-contacts with the environment other than those at the feet of the humanoid robot. In early days, a common approach was to divide the problem into three independent stages [112]: The *first stage* is to compute a coarse movement for all DOFs. This is usually done by using a motion tracking system to observe humans walking or performing some actions [113, 114]. It is also possible to use probabilistic path planners, which can generate automatically a path according to certain constraints and goals [68, 69]. Kinematic models of humans and humanoid robots might be similar, but their dynamic models are never the same. For this reason, the *second stage* is to compute a physically feasible motion from the observed or computed coarse movement [115]. The *final stage* is then to provide online stabilization via sensory feedback during the movement execution in order to deal with unforeseen or unmodeled dynamics. The first approach for generating full-body motion was presented in Ref. [116] where the authors used a discrete set of foot steps selected by vision for the locomotion and automatic path planning for the manipulation. Work using online modification of the zero moment point (ZMP) trajectory during the execution of the motion was also documented in Ref. [117]. More recently proposed approaches have focused more on generating real-time whole-body motions.

In References [118, 119], the term "whole-body control" was used for the first time to refer to a floating-base task-oriented dynamic control and prioritization

framework that enables a humanoid robot to fulfill simultaneous real-time control objectives. Prioritization and coordination of several controllers is achieved using a hierarchy that handles conflicts and selects the one with highest priority. This approach, in comparison to previous ones, was the first one that focuses not only on the walking but on the manipulation tasks while on two legs. The most recent work of this team group has been renamed to "whole-body operational space control" (WBOSC) to avoid confusion from the more broadly defined term "whole-body control" used nowadays [120]. The system prototype of WBOSC was also extended to include the possibility of internal force control and is available as an open-source software named ControlIt! [121]. Figure 5.28 shows the overall control diagram proposed by WBOSC as well as the joint torque controllers. In essence, it is a distributed control system in which the joint controllers take care of the actuator dynamics while the central control or WBOSC remains responsible for the overall robot dynamics.

A recent work in Ref. [122] presents a generalized hierarchical control, which is able to deal with both strict and nonstrict priorities in an arbitrary number of tasks. In dynamical environments, a certain nonstrict priority might become a strict one and the presented approach enables the possibility of switching task priorities to deal with those situations. Another prominent approach for whole-body control is based on controlling the robot's linear and angular momentum at

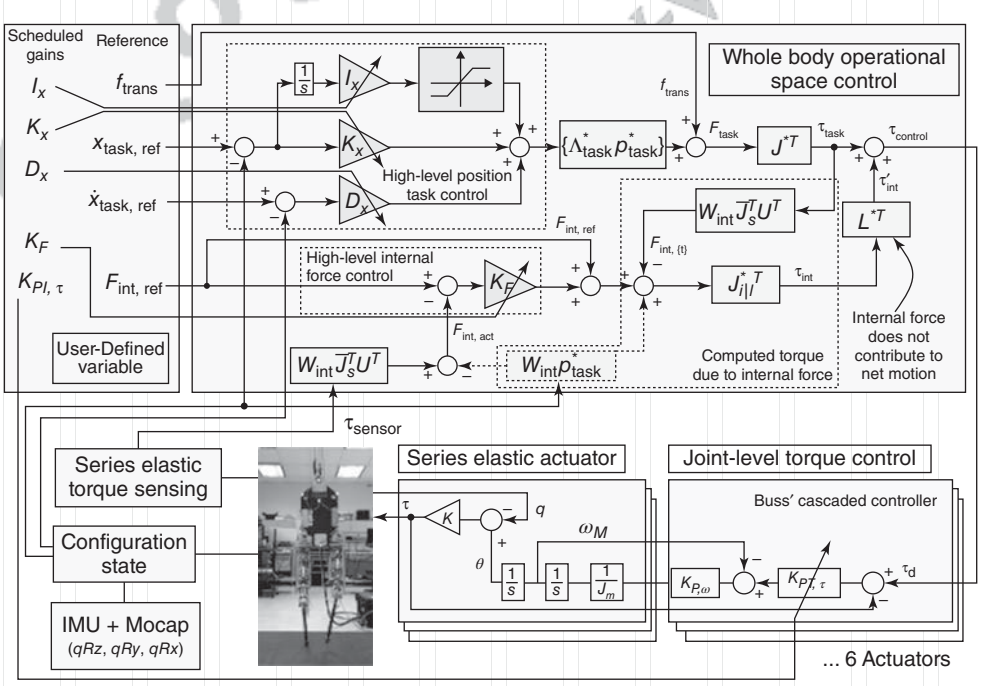


Figure 5.28 Whole-body operational space control (WBOSC) diagram. (Courtesy Luis Sentis, HCRL, University of Texas at Austin.)

each point in time [123], the so-called resolved momentum control (RMC). The approach taken in Ref. [124] is to use a set of *a priori* typical basic postures, which are combined to perform the current task so that it is dynamically balanced and natural. The software framework iTaSC can also be used to generate whole-body motions by specifying constraints between parts of the robot, and between the robot and the environment, allowing the specification and sensor-based reactive control of complex robotic tasks, which require multiple and simultaneous subtasks [125]. An example of its use can be seen in Figure 5.29 [126], in which the robot AILA makes use of a whole-body reactive control for performing a task at a mock-up ISS. Similarly in Ref. [127], the framework controls the robot by using sensor-based control tasks, which are simultaneously executed and synchronized by a so-called "stack of tasks." Another framework proposed in Ref. [128] is implemented as a teleoperation system that allows the operator to select and control only the necessary points of the robot's body for manipulation. A switching method allows the operator to select specific point among the body parts and whole-body motions are then automatically generated based on RMC to maintain stability and maximize the reaching workspace. The methodology in Ref. [129] is to observe humans with a MoCap system and generate dynamically stable and physically plausible whole-body motions from the captured data.

The recent frameworks on whole-body control are mostly used in mobile manipulation (discussed in the following section) and humanoid robots. As long as manipulation is involved, contacts with the environment are desirable and not treated as disturbances. The complex robotic systems need to deal with simultaneous multicontact forces (such as between feet or mobile base and the ground, or between manipulator(s) and the objects being manipulated), with the aim of keeping balance or an optimal posture. This requires efficient and online control

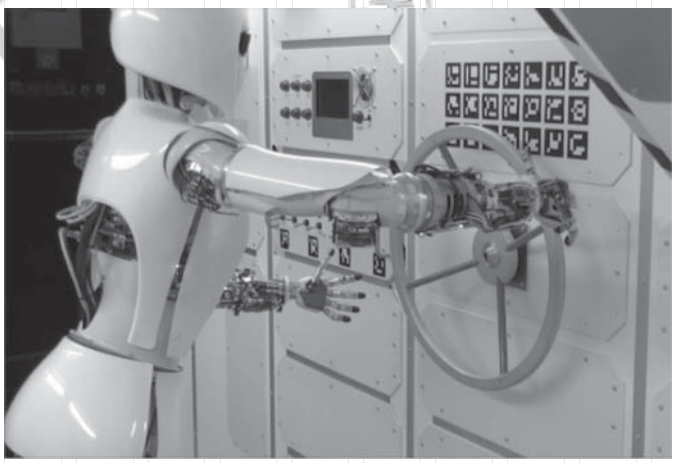


Figure 5.29 Robot AILA using whole-body control to perform some tasks within a mock-up International Space Station. (Courtesy DFKI GmbH.)

strategies based on real-time feedback that can make optimal usage of the redundancy within such robotic systems.

5.5.3 Mobile Manipulation

The field of mobile manipulation overlaps with the field of humanoid robotics both of which are closely related to the formerly described "whole-body control." These fields of robotics focus on manipulation systems mounted on mobile platforms (either on wheels or legs, with or without human morphology). Unlike whole-body control that primarily focuses on the optimal selection of the robot's DOF and the prioritization and execution of simultaneous subtasks, the mobile manipulation focuses on the mobile manipulators, which are able to solve complex tasks in unknown, unstructured, and changing environments (such as in outer space). Thus, mobile manipulation will likely use the whole-body control concepts as a building block of the mobile manipulator system.

In recent years, autonomous mobile manipulation has emerged to a new research topic in robotics and been identified as critical to future robotic applications [130]. Since the research focus of mobile manipulation lies on producing a new category of robots beyond the current state of the art in order to be able to solve complex tasks in unstructured and changing environments, the work needs to deal with the execution of complex manipulation tasks that might require mobility (such as moving from one location to another) in such challenging environments. Many robotic solutions have been deployed in well-known static environments such as factories where uncertainties and unexpected events are minimized. Future robots that benefit from having mobile manipulation are expected to be reliable and deployable in the household environment, during man-made or natural disasters, or for extraterrestrial exploration, and so on. These robots then need to be able to perform a large variety of tasks in completely or partially unknown scenarios by acquiring and reusing generic skills which can be adapted to novel or unexpected situations.

Mobile manipulation aims at maximizing task generality, that is, increasing the variety of tasks that the robots can autonomously accomplish and, at the same time, minimizing the dependency on prior information about the environments in which robots are deployed. The main challenge similar to most robotics research is about how to integrate various subsystems together (including the perception, manipulation, planning, control, cognition, artificial intelligence, and mobility, etc.) so that the mobile manipulation system is able to cope with a wide range of real-world situations.

5.5.3.1 Mobile Manipulators as Research Platforms

Some notable examples of existing terrestrial mobile manipulators include (but not limited to) the following:

Q13

- The PR-2 robot developed by Willow Garage [131]. This is a dual-arm robot with an omni-directional mobile base. It includes a variety of sensors such as tilting laser scanner on the head, laser scanner on the mobile base, two pairs of stereo cameras, and an IMU located inside the body. Currently, the PR-2 is probably the most advanced autonomous mobile manipulator capable of performing complex manipulation and navigation tasks.
- The "butler" robot HERB developed at Intel Research Labs in collaboration with Carnegie Mellon University [132]. This is an autonomous mobile manipulator that was designed to perform complex manipulation tasks in the home environment. The robot is able to search, recognize, store new objects as well as manipulate door handles and objects, and navigate in cluttered environments.
- The Rollin' Justin developed at the DLR Institute of Robotics and Mechatronics [133]. Justin is a progressive development whose two arms were initially built on the lightweight arms from DLR (LWR-III) and was later equipped with a mobile base to enhance the robot's field of work.
- The UMan from the Robotics and Biology Lab at the University of Massachusetts Amherst [134]. This robot consists of a modified Nomadic XR4000 holonomic mobile base with 3 DOFs, a WAM 7-DOF manipulator arm, and a 4-DOF hand from Barret Technologies.
- The robot AILA developed at DFKI Robotics Innovation Center [135]. This robot is a mobile dual-arm system developed as a research platform for mobile manipulation. AILA has 32 DOFs, including 7-DOF arms, 4-DOF torso, 2-DOF head, and a mobile base equipped with 6 wheels, each of which has 2 DOFs.

5.5.3.2 DARPA Robotics Challenge (DRC)

The DRC and its latest competition in 2015 set the benchmark of industrial robotic platforms for mobile manipulation [136]. The competition has been organized by the US Defense Advanced Research Projects Agency (DARPA) and aims to promote and foster development of semi-autonomous robots to perform complex tasks in realistic disaster scenarios. The involved tasks include driving a vehicle, walking among rubble, removing debris blocking the way, opening doors, climbing a ladder, using tools to drill a hole on the wall and rotate a valve. The competition was initiated in 2012, had a first virtual challenge that took place in 2013, and later on two live demonstrations including the trials in December 2013 and the finals in June 2015. For the trials, most of the teams used teleoperated systems for solving most of the tasks. At the finals, 25 robotic systems from different countries participated among which some robots applied certain LoA. In this final competition, three teams scored the maximal attainable number of points (see photos in Figure 5.30), including the following:

- The KAIST team from South Korea and their robot Hubo, which has been developed since 2002 and whose complete and more powerful version was specifically designed for the DRC competition. The biped robot has a total height of 180 cm and a mass of 80 kg.

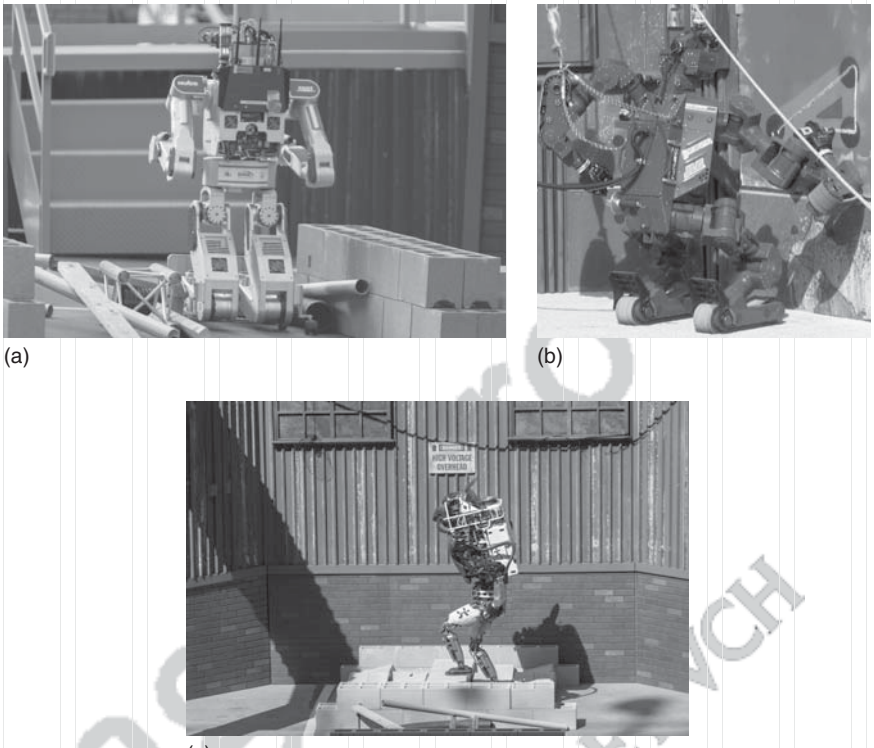


Figure 5.30 DRC robots in 2015 finals.

- The Tartan Rescue team from United States with their robot Chimp, which was developed at Carnegie Mellon University, is 150 cm in height and weighs 201 kg. Chimp combined high-level operator commands with certain low-level autonomy (e.g., autonomously plan and execute joint and limb movements or grasps). The robot rolls on four legs using rubber tracks such as a tank. In order to manipulate, the robot then stands up on two legs and manipulates with the front limbs.
- The Florida Institute of Human Machine Cognition (IHMC) Robotics team from United States with their Atlas robot Running Man, which was developed by Boston Dynamics and made available for the IHMC team participating at the DRC. The biped robot is 190 cm in height and weights 175 kg.

The final decision of the winner was judged by which robot used the least time to perform all tasks; hence, the first place went to the KAIST team.

5.5.3.3 Mobile Manipulators for Space

NASA's Johnson Space Center (JSC) has built a humanoid robot (initially named Valkyrie, latest prototypes known as "R5," see Figure 5.31). During the DRC competition, JSC collaborated with the University of Edinburgh and the IHMC on

Figure 5.31 NASA's R5 or Valkyrie robot. (Courtesy NASA.)

its control. However, the status of R5's development did not allow the robot to show its full capabilities during the competitions (in fact, the robot only participated without much success on the trials). In November 2015, NASA awarded two Valkyrie robots to two university teams (i.e., MIT in Cambridge and Northeastern University in Boston) in preparation for an upcoming NASA's Space Robotics Challenge (SRC) that aims to explore the technology readiness level of sending humanoid robots into space, specifically to Mars. As reported in a press release, NASA's interest in humanoid robots is driven from the observation that they have huge potential to efficiently operate in human-built environments and effectively cooperate with astronauts. The robots developed for the DRC demonstrate commonalities between robots designed to work in a disaster scenario and those for the extreme environments such as the planetary bodies. The tasks that the Valkyrie robots need to fulfill at the SRC are not yet released at the time of writing this book, though initial ideas have been made public. It is envisaged that in 2017 the two robots need to compete against each other by performing various tasks such

as exiting a habitat and using a ladder, removing cables from one storage and attaching them to another location while traversing irregular terrain, repairing of components such as a broken valve or a tire, and collecting samples of soil or rocks as typically seen in planetary missions.

References

1. Bonitz, R.G. (1997) Mars surveyor '98 lander MVACS robotic arm control system design concepts. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, USA, April 20-25, 1997*, pp. 2465–2470.
2. Bonitz, R.G., Nguyen, T.T., and Kim, W.S. (2000) The Mars surveyor '01 rover and robotic arm. Aerospace Conference Proceedings, 2000 IEEE, Vol. 7, pp. 235–246.
3. Barnes, D., Phillips, N., and Paar, G. (2003) Beagle 2 simulation and calibration for ground segment operations. *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
4. Shiraishi, L. and Bonitz, R.G. (2008) NASA Mars 2007 Phoenix lander robotic arm and icy soil acquisition device. *Journal of Geophysical Research*, **113**.
5. Baumgartner, E.T., Bonitz, R.G., Melko, J.P., Shiraishi, L.R., and Leger, P.C. (2005) The Mars exploration rover instrument positioning system. *Aerospace Conference, 2005 IEEE*, pp. 1–19.
6. Phillips, N. (2001) Mechanisms for the Beagle 2 lander. *Proceedings of the 9th European Space Mechanisms and Tribology Symposium*, pp. 25–32.
7. Fleischner, R. and Billing, R. (2011) Mars science laboratory robotic arm. *ESMATS*.
8. Robinson, M., Collins, C., Leger, P., Kim, W., Carsten, J., Tompkins, V., Trebi-Ollennu, A., and Florow, B. (2013) Test and validation of the Mars Science Laboratory Robotic Arm. *8th International Conference on System of Systems Engineering (SoSE) 2013, Maui, HI, USA, 2-6 June 2013*, pp. 184–189.
9. Yoshikawa, T. (1985) Manipulability of robotic mechanisms. *International Journal of Robotics Research*, **4** (2), 3–9.
10. Whitney, D.E. (1987) Historical perspective and the state-of-the art in robot force control. *International Journal of Robotics Research*, **6**, 3–14.
11. De Schutter, J., Bruyninckx, H., Zhu, W.-H., and Spong, M.W. (1997) Force control: a bird's eye view, in *Control Problems in Robotics and Automation: Future Directions* (ed. B. Siciliano), Springer-Verlag, pp. 1–17.
12. Siciliano, B. and Villani, L. (eds) (1999) *Robot Force Control*, Kluwer Academic Publishers, Boston, MA.
13. Lefebvre, T., Xiao, J., Bruyninckx, H., and De Gersem, G. (2005) Active compliant motion: a survey. *Advanced Robotics*, **19** (5), 479–499.
14. Siciliano, B. and Khatib, O. (eds) (2008) *Springer Handbook of Robotics*, Springer-Verlag, Berlin, Heidelberg.
15. Helmick, D.M., McCloskey, S., Okon, A., Carsten, J., Kim, W.S., and Leger, C. (2013) Mars science laboratory algorithms and flight software for autonomously drilling rocks. *Journal of Field Robotics*, **30** (6), 847–874.
16. Chiaverini, S., Siciliano, B., and Villani, L. (1999) A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Transactions on Mechatronics*, **4** (3), 273–285.
17. Raibert, M.H. and Craig, J.J. (1981) Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, **102**, 126–133.
18. Chiaverini, S. and Sciavicco, L. (1993) The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, **9**, 361–373.
19. Hogan, N. (1985) Impedance control-an approach to manipulation, part I-

theory, part II-implementation, part III- application. *Journal of Dynamics Systems, Measurement, and Control-Transactions of the ASME*, **107** (1), 1–24.

20. Nevins, I. and Whitney, D.E. (1973) The force vector assembler concept. *Proceedings of 1st CSIM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*.
21. Whitney, D.E. (1977) Force feedback control of manipulator fine motions. *ASME Journal of Dynamic Systems, Measurement, and Control*, **99**, 91–97.
22. Albu-Schäffer, A. and Hirzinger, G. (2002) Cartesian impedance control techniques for torque controlled light-weight robots. *ICRA*, pp. 657–663.
23. Hogan, N. (1988) On the stability of manipulators performing contact tasks. *IEEE Journal of Robotics and Automation*, **4**, 677–686.
24. Surdilovic, D. (1996) Contact stability issues in position based impedance control: theory and experiments. *Proceedings of IEEE ICRA96*, pp. 1675–1680.
25. Kazerooni, H. (1990) Contact instability of the direct drive robot when constrained by a rigid environment. *IEEE Transactions on Automation and Control*, **35**, 710–714.
26. Seraji, H. and Colbaugh, R. (1993) Adaptive force-based impedance control. Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1537–1544.
27. Colbaugh, R., Seraji, H., and Glass, K. (1993) Direct adaptive impedance control of robot manipulators. *Journal of Robotic Systems*, **10** (2), 217–248.
28. Singh, S.K. and Popa, D.O. (1995) An analysis of some fundamental problems in adaptive control of force and impedance behavior, theory and experiments. *IEEE Transactions on Robotics and Automation*, **11**, 223–228.
29. Matko, D., Kamnik, R., and Badj, T. (1999) Adaptive impedance force control of an industrial manipulator. *Proceedings of IEEE International Symposium on Industrial Electronics*, pp. 129–133.
30. Lu, A.A. and Goldenberg, Z. (1995) Robust impedance control and force regulation: theory and experiments. *International Journal of Robotics Research*, **14**, 225–254.
31. Jung, S. and Hsia, T.C. (2000) Robust neural force control scheme under uncertainties in robot dynamics and unknown environment. *IEEE Transactions on Industrial Electronics*, **47** (2), 403–412.
32. Seraji, H. and Colbaugh, R. (1997) Force tracking in impedance control. *International Journal of Robotics Research*, **16**, 97–117.
33. Jung, S., Hsia, T.C., and Bonitz, R.G. (2001) Force tracking impedance control for robot manipulators with an unknown environment: theory, simulation, and experiment. *International Journal of Robotics Research*, **20** (9), 765–774.
34. Jung, S. and Hsia, T.C. (1998) Neural network impedance force control of robot manipulator. IEEE Transactions on Industrial Electronics, pp. 451–461.
35. Jung, S., Yim, S.B., and Hsia, T.C. (2001) Experimental studies of neural network impedance force control for robot manipulators. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2001, vol. 4,* pp. 3453–3458.
36. Cheah, C.C. and Wang, D. (1995) Learning impedance control for robotic manipulators. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), vol. 2,* pp. 2150–2155.
37. Cohen, M. and Flash, T. (1991) Learning impedance parameters for robot control using an associative search network. *IEEE Transactions on Robotics and Automation*, **7** (3), 382–390.
38. Bonitz, R.G. and Hsia, T.C. (1996) Robust internal-force based impedance control for coordinating manipulators-theory and experiments. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), vol. 1,* pp. 622–628.
39. Lin, S. and Tsai, H. (1997) Impedance control with on-line neural network

compensator for dual-arm robots. *Journal of Intelligent Robotics Systems*, **18** (1), 87–104.

40. Moosavian, S.A.A. and Papadopoulos, E. (1998) Multiple impedance control for object manipulation. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 1,* pp. 461–466.
41. Hill, J. and Park, W. (1979) Real-time control of a robot with a mobile camera. Proceedings of the 9th International Symposium on Industrial Robots, pp. 233–246.
42. Shirai, Y. and Inoue, H. (1973) Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, **5** (2), 99–106.
43. Buttazzo, G.C., Allotta, B., and Fanizza, F.P. (1994) Mousebuster: a robot for real-time catching. *IEEE Control Systems Magazine*, **14** (1), 49–56.
44. Kragic, D. (2001) *Visual servoing for manipulation: robustness and integration issues*. PhD thesis. KTH, Numerical Analysis and Computer Science, NADA.
45. Espiau, B., Chaumette, F., and Rives, P. (1992) A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, **8**, 313–326.
46. Hashimoto, H., Ogawa, H., Umeda, T., Obama, M., and Tatsuno, K. (1995) An unilateral master-slave hand system with a force-controlled slave hand. ICRA'95, pp. 956–961.
47. Andersson, R.L. (1989) Dynamic sensing in a ping-pong playing robot. *IEEE Transactions on Robotics and Automation*, **5** (6), 728–739.
48. Vahrenkamp, N., Wieland, S., Azad, P., Gonzalez, D., Asfour, T., and Dillmann, R. (2008) Visual servoing for humanoid grasping and manipulation tasks. Humanoid Robots, 2008. Humanoids 2008. *8th IEEE-RAS International Conference on,* pp. 406–412.
49. Weiss, L., Sanderson, A., and Neuman, C. (1987) Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, **3** (5), 404–417.
50. Hosoda, K. and Asada, M. (1994) Versatile visual servoing without knowledge of true Jacobian. Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. *Proceedings of the IEEE/RSJ/GI International Conference on, vol. 1,* pp. 186–193.
51. Wilson, W.J., Williams Hulls, C.C., and Bell, G.S. (1996) Relative end-effector control using Cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, **12** (5), 684–696.
52. Martinet, P. and Gallice, J. (1999) Position based visual servoing using a non-linear approach. Intelligent Robots and Systems, 1999. IROS '99. *Proceedings. 1999 IEEE/RSJ International Conference on, vol. 1,* pp. 531–536.
53. Rajan, V. (1985) Minimum time trajectory planning. Robotics and Automation. Proceedings. 1985 IEEE International Conference on, vol. 2, pp. 759–764.
54. Biagiotti, L. and Melchiorri, C. (2008) *Trajectory Planning for Automatic Machines and Robots*, Springer-Verlag, Berlin Heidelberg.
55. Kröger, T. (2010) *On-Line Trajectory Generation in Robotic Systems*, Springer Tracts in Advanced Robotics, vol. 58, Springer-Verlag, Berlin, Heidelberg, Germany.
56. Kröger, T. (2011) Opening the door to new sensor-based robot applications – The reflexxes motion libraries. *Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China.*
57. Khatib, O. (1986) Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, **5** (1), 90–98.
58. Seraji, H. and Bon, B. (1999) Real-time collision avoidance for position-controlled manipulators. *IEEE Transactions on Robotics and Automation*, **15** (4), 670–677.
59. De Luca, A., Albu-Schaffer, A., Haddadin, S., and Hirzinger, G. (2006) Collision detection and safe reaction with the DLR-III lightweight manipulator arm. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 1623–1630–.

60. Flacco, F., Kroger, T., De Luca, A., and Khatib, O. (2012) A depth space approach to human-robot collision avoidance. *Robotics and Automation (ICRA), 2012 IEEE International Conference on,* pp. 338–345.
61. Kuffner, J., Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., and Inoue, H. (2002) Self-collision detection and prevention for humanoid robots. *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on,* vol. 3, pp. 2265–2270.
62. Seto, F., Kosuge, K., and Hirata, Y. (2005) Self-collision avoidance motion control for human robot cooperation system using robe. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on,* pp. 3143–3148.
63. De Santis, A., Albu-Schaeffer, A., Ott, C., Siciliano, B., and Hirzinger, G. (2007) The skeleton algorithm for self-collision avoidance of a humanoid manipulator. *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on,* pp. 1–6.
64. Dietrich, A., Wimbock, T., Taubig, H., Albu-Schaeffer, A., and Hirzinger, G. (2011) Extensions to reactive self-collision avoidance for torque and position controlled humanoids. *Robotics and Automation (ICRA), 2011 IEEE International Conference on,* pp. 3455–3462.
65. Taubig, H., Bauml, B., and Frese, U. (2011) Real-time swept volume and distance computation for self collision detection. *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on,* pp. 1585–1592.
66. Stasse, O., Escande, A., Mansard, N., Miossec, S., Evrard, P., and Kheddar, A. (2008) Real-time (self)-collision avoidance task on a HRP-2 humanoid robot. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on,* pp. 3200–3205.
67. Leger, C. (2002) Efficient sensor/model based on-line collision detection for planetary manipulators. *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA* 2002, May 11-15, 2002, Washington, DC, USA, pp. 1697–1703.
68. Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., and Inoue, H. (2005) Motion planning for humanoid robots, in *Robotics Research*, Springer Tracts in Advanced Robotics, vol. 15 (eds P. Dario and R. Chatila), Springer-Verlag, Berlin / Heidelberg, pp. 365–374.
69. Kavraki, L.E., Svestka, P., Latombe, J.C., and Overmars, M.H. (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, **12**, 566–580.
70. Lavalle, S.M. and Kuffner, J.J. (2000) Rapidly-exploring random trees: progress and prospects. Algorithmic and Computational Robotics: New Directions, pp. 293–308.
71. Ge, S.S. and Cui, Y.J. (2002) Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, **13**, 207–222.
72. Williamson, M.M. (1999) *Robot arm control exploiting natural dynamics*. PhD thesis, Massachusetts Institute of Technology.
73. Ijspeert, A.J., Nakanishi, J., and Schaal, S. (2002) Learning rhythmic movements by demonstration using nonlinear oscillators. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 958–963.
74. Kurita, Y., Lim, Y., Ueda, J., Matsumoto, Y., and Ogasawara, T. (2004) CPG-based manipulation: generation of rhythmic finger gaits from human observation. ICRA, pp. 1209–1214.
75. Saxena, A., Driemeyer, J., Kearns, J., Osondu, C., and Ng, A. (2008) Learning to grasp novel objects using vision, in *Experimental Robotics*, Springer Tracts in Advanced Robotics, vol. 39 (O. Khatib, V. Kumar, and D. Rus), Springer-Verlag, Berlin / Heidelberg, pp. 33–42.
76. Zöllner, R., Asfour, T., and Dillmann, R. (2004) Programming by demonstration: dual-arm manipulation tasks for humanoid robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

77. Argall, B.D., Chernova, S., Veloso, M., and Browning, B. (2009) A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, **57** (5), 469–483.
78. Billard, A., Epars, Y., Calinon, S., Schaal, S., and Cheng, G. (2004) Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, **47** (2-3), 69–77.
79. Tunstel, E., Maimone, M.W., Trebi-Ollennu, A., Yen, J., Petras, R., and Willson, R.G. (2005) Mars exploration rover mobility and robotic arm operational performance. *SMC,* IEEE, pp. 1807–1814.
80. Trebi-Ollennu, A., Baumgartner, E.T., Leger, P.C., and Bonitz, R.G. (2005) Robotic arm in-situ operations for the Mars exploration rovers surface mission. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, Hawaii, USA, October 10-12, 2005,* pp. 1799–1806.
81. Robinson, M., Collins, C., Leger, P., Carsten, J., Tompkins, V., Hartman, F., and Yen, J. (2013) In-situ operations and planning for the Mars Science Laboratory Robotic Arm: the first 200 sols. *System of Systems Engineering (SoSE), 2013 8th International Conference on,* pp. 153–158.
82. Goertz, R. and Thompson, R. (1954) Electronically controlled manipulator. *Nucleonics*, **12** (11), 46–47.
83. Hirzinger, G., Brunner, B., Dietrich, J., and Heindl, J. (1993) Sensor-based space robotics-rotex and its telerobotic features. *IEEE Transactions on Robotics and Automation*, **9** (5), 649–663.
84. Hirzinger, G., Heindl, J., and Landzettel, K. (1989) Predictive and knowledge-based telerobotic control concepts. *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on, vol. 3,* pp. 1768–1777.
85. Preusche, C., Reintsema, D., Landzettel, K., and Hirzinger, G. (2006) Robotics component verification on ISS Rokviss - preliminary results for telepresence. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on,* pp. 4595–4601.
86. European Space Agency (2015) First Handshake and Force-Feedback with Space.
87. Ferrell, W.R. and Sheridan, T.B. (1967) Supervisory control of remote manipulation. *IEEE Spectrum*, **4** (10), 81–88.
88. Wright, J., Hartman, F., Cooper, B., Maxwell, S., Yen, J., and Morrison, J. (2006) Driving on Mars with rsvp. *IEEE Robotics Automation Magazine*, **13** (2), 37–45.
89. Wright, J.R., Hartman, F., Maxwell, S., Cooper, B., and Yen, J. (2013) Updates to the rover driving tools for curiosity. *System of Systems Engineering (SoSE), 2013 8th International Conference on,* pp. 147–152.
90. Sheridan, T.B. and Verplank, W.L. (1978) Human and computer control of undersea teleoperators (Man-Machine Systems Laboratory Report).
91. Crandall, J.W. and Goodrich, M.A. (2002) Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation. *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, vol. 2,* pp. 1290–1295.
92. Hayati, S. and Venkataraman, S.T. (1989) Design and implementation of a robot control system with traded and shared control capability. Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on, vol. 3, pp. 1310–1315.
93. Kortenkamp, D., Bonasso, P., Ryan, D., and Schreckenghost, D. (1997) Traded Control with Autonomous Robots as Mixed Initiative Interaction. AAAI Technical Report SS-97-04. Q15
94. Dorais, G.A., Bonasso, R.P., Kortenkamp, D., Pell, B., and Schreckenghost, D. (1999) Adjustable autonomy for human-centered autonomous systems. Proceedings of the 1st International Conference of the Mars Society.
95. Crandall, J.W. and Goodrich, M.A. (2001) Experiments in adjustable autonomy. Systems, Man, and Cybernetics,

2001 IEEE International Conference on, vol. 3, pp. 1624–1629.

96. Roth, Z.S., Mooring, B.W., and Ravani, B. (1987) An overview of robot calibration. *IEEE Journal on Robotics and Automation*, **3** (5), 377–385.

97. Mooring, B., Driels, M., and Roth, Z. (1991) *Fundamentals of Manipulator Calibration*, John Wiley & Sons, Inc., New York.

98. Rybus, T. et al. (2013) New planar air-bearing microgravity simulator for verification of space robotics numerical simulations and control algorithms. *Proceedings of the 12th ESA Symposium on Advanced Space Technologies in Robotics and Automation* (*ASTRA* 2013).

99. Rutkowski, K., Rybus, T., Wawrzaszek, R., Seweryn, K., and Grassmann, K. (2015) Design and development of two manipulators as a key element of a space robot testing facility. *Archive of Mechanical Engineering*, **LXII** (3), doi: 10.1515/meceng-2015-0022.

100. Banaszkiewicz, M. and Seweryn, K. (2008) Optimization of the trajectory of a general free - flying manipulator during the rendezvous maneuver. *AIAA Guidance, Navigation, and Control Conference*.

101. Wu, Y.-H., Gao, Y., Lin, J.-W., Raus, R., Zhang, S.-J., and Watt, M. (2013) Low-cost, high-performance monocular vision system for air bearing table attitude determination. *Journal of Spacecraft and Rockets*, **51** (1), 66–75.

102. Wawrzaszek, R., Banaszkiewicz, M., Rybus, T., Wisniewski, L., Seweryn, K., and Grygorczuk, J. (2014) Low velocity penetrators (LVP) driven by hammering action - definition of the principle of operation based on numerical models and experimental test. *Acta Astronautica*, **99**, 303–317.

103. Alford, C.O. and Belyeu, S.M. (1984) Coordinated control of two robot arms. *Proceedings of IEEE International Conference on Robotics & Automation*, pp. 468–473.

104. Zheng, Y.F. and Luh, J.Y.S. (1986) Joint torques for control of two coordinated moving robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1375–1380.

105. Hayati, S. (1986) Hybrid position/force control of multiarm cooperating robots. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1375–1380.

106. Tarn, T., Bejczy, A., and Yun, X. (1987) Design of dynamic control of two cooperating robot arms: closed chain formulation. *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 7–13.

107. Yamano, M., Kim, J.-S., Konno, A., and Uchiyama, M. (2004) Cooperative control of a 3D dual-flexible-arm robot. *Journal of Intelligent and Robotic Systems*, **39**, 1–15.

108. Wang, Z.-D., Hirata, Y., Takano, Y., and Kosuge, K. (2004) From human to pushing leader robot: leading a decentralized multirobot system for object handling. *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*, pp. 441–446.

109. Zacharias, F., Leidner, D. et al. (2010) Exploiting structure in two-armed manipulation tasks for humanoid robots. *The IEEE International Conference on Intelligent Robots and Systems (IROS)*.

110. Vahrenkamp, N., Do, M., Asfour, T., and Dillmann, R. (2010) Integrated Grasp and motion planning. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, pp. 2883–2888.

111. Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., and Dillmann, R. (2009) Humanoid motion planning for dual-arm manipulation and re-grasping tasks. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09).

112. Azevedo, C., Poignet, P., and Espiau, B. (2004) Artificial locomotion control: from human to robots. *Robotics and Autonomous Systems*, **47** (4), 203–223.

113. Lee, N., Rietdyk, C.S.G., and Naksuk, S. (2005) Whole-body human-to-humanoid motion transfer. *IEEE-RAS International Conference on Humanoid Robots*, pp. 104–109.

114. Mombaur, K. and Sreenivasa, M.N. (2010) HRP-2 plays the yoyo: from

human to humanoid yoyo playing using optimal control. ICRA, pp. 3369–3376.

115. Yamane, K. and Nakamura, Y. (2003) Dynamics filter - concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, **19**, 421–432.

116. Kuffner, J.J., Kagami, S., Nishiwaki, K., Inaba, M., and Inoue, H. (2002) Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, **12**, 105–118.

117. Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., and Inoue, H. (2002) Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2684–2689.

118. Sentis, L. (2007) *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, Stanford University, Stanford, CA.

119. Sentis, L., Petersen, J., and Philippsen, R. (2013) Implementation and stability analysis of prioritized whole-body compliant controllers on a wheeled humanoid robot in uneven terrains. *Autonomous Robots*, **35**, 301–319.

120. Sentis, L., Park, J., and Khatib, O. (2010) Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on Robotics*, **26** (3), 483–501.

121. Fok, C.-L., Johnson, G., Yamokoski, J.D., Mok, A.K., and Sentis, L. (2015) Controlit! - A software framework for whole-body operational space control. CoRR, abs/1506.01075.

122. Liu, M., Tan, Y., and Padois, V. (2015) Generalized hierarchical control. *Autonomous Robots*, **XX** 1–15.

123. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003) Resolved momentum control: humanoid motion planning based on the linear and angular momentum. *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1644–1650.

124. Nishiwaki, K., Kuga, M., Kagami, S., Inaba, M., and Inoue, H. (2005) Whole-body cooperative balanced motion generation for reaching. *International Journal of Humanoid Robotics*, **2** (4), 437–457.

125. Smits, R., De Laet, T., Claes, K., Bruyninckx, H., and De Schutter, J. (2009) iTASC: a tool for multi-sensor integration in robot manipulation, in *Multisensor Fusion and Integration for Intelligent Systems*, Lecture Notes in Electrical Engineering, vol. **35**, Springer-Verlag, pp. 235–254.

126. de Gea Fernandez, J., Mronga, D., Wirkus, M., Bargsten, V., Asadi, B., and Kirchner, F. (2015) Towards describing and deploying whole-body generic manipulation behaviours. Space Robotics Symposium.

127. Mansard, N., Stasse, O., Chaumette, F., and Yokoi, K. (2007) Visually-guided grasping while walking on a humanoid robot. *Proceedings of 2007 IEEE International Conference on Robotics and Automation*, pp. 3042–3047.

128. Neo, E.S., Yokoi, K., Kajita, S., Kanehiro, F., and Tanie, K. (2005) A switching command-based whole-body operation method for humanoid robots. *IEEE/ASME Transactions on Mechatronics*, **10** (5), 546–559.

129. Kim, S., Kim, C.H., You, B.-J., and Oh, S.-R. (2009) Stable whole-body motion generation for humanoid robots to imitate human motions. IROS, pp. 2518–2524.

130. Brock, O. and Grupen, R. (2005) NSF/NASA workshop on autonomous mobile manipulation (Amm), Houston, USA, http://robotics.cs.umass.edu/amm.

131. Chitta, S., Cohen, B., and Likhachev, M. (2010) Planning for autonomous door opening with a mobile manipulator. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.

132. Srinivasa, S.S., Ferguson, D., Helfrich, C.J., Berenson, D., Collet, A., Diankov, R., Gallagher, G., Hollinger, G., Kuffner, J., and Vande Weghe, M. (2009) HERB: a home exploring robotic butler. *Autonomous Robots*, **28** (1), 5–20.