Torque Control: A Study On The iCub Humanoid Robot

Alberto Parmiggiani

DIST

University of Genoa, Italy
Italian Institute of Technology

A thesis submitted for the degree of

PhilosophiæDoctor (PhD)

February 2010

Declaration

The thesis work was conducted from 22^{nd} january 2007 to 18^{th} febuary 2010 under the supervision of Lorenzo Natale and Giorgio Metta at the Italian Insitute of Technology (IIT).

Genoa,

Abstract

Despite the delays and slow dynamics of the sensorimotor apparatus, humans are capable of producing an extraordinarily wide repertoire of motor behaviours. Robots on the other hand are characterized by responsive actuation systems, and fast feedback loops. Nevertheless robots can hardly match the dexterous manipulation capabilities of humans. Moreover the inherent dynamics of today's robots are dominated by actuator inertia and friction thus complicating physically interaction. The work described in this thesis goes in the direction of bridging the gap between the capabilities of humans and robots. The first part of the thesis describes how the back-drivability of a robot can be improved with joint torque control. This was achieved by designing torque sensors, and by implementing joint level torque control on the arm of the iCub robot. The second part describes how back-drivable robotic platforms can be used to perform dynamic tasks. Numerical methods for planning this kind of tasks were implemented and tested in simulation.

a Monica,

alla mia familia

Contents

List of Figures

LIST OF FIGURES

List of Tables

Introduction

Humans are capable of producing an extraordinary wide repertoire of motor behaviours. Understanding how such skilled movements are controlled and coordinated is still an open problem. It is known from kinesiology and motor control studies, that the human motor system tackles this problem with complex hierarchical structures that involve sensory feedback and partly openloop strategies [Kelso, 1982]. Some movements can be viewed as a sort of error nullying activity where error correction is performed on the basis of afferent signals transmitted to the central nervous system (CNS); but processing these sensory signals requires time. There are however cases, e.g. when movements are generated quickly, in which this error detection and correction strategy, based only on sensory input, would fail. Most theories agree on the fact that the CNS adopts an openloop control mechanism that relies on an internal model of the dynamics of the musculoskeletal system. With reference to the field of robotics the motor control problem is often addressed with error feedback controllers whereas the feedforward aspect has in general received less attention. This is probably due the inherent behaviour of today's robots whose dynamics are dominated by inertia and friction and to the difficulties encountered when trying to estimate accurate parameter based dynamic models. We nevertheless consider as very important the study of dynamic tasks and feedforward control schemes in the perspective of tomorrow's robots becoming more and more backdriveable. To study dynamic tasks a suitable robot is needed. Current robots follow the mainstream design paradigm, which is based on the use of rotary electric motors coupled with speed reducing gearboxes. This choice implies several disadvantages among which high reflected inertia, backlash effects, non-linear frictional effects, etc., which contribute altogether to "cancel" forces and torques that arise dynamically. The RobotCub robotic platform [Metta et al., 2005] is also based on this design and is thus also affected by this problem.

INTRODUCTION

The work described in this thesis unfolded on two parellel research directions. The initial part of the project focused on the improvement of the iCub arm back-driveability. Torque sensors for the joints of an arm were designed and constructed. An experimental upper-body prototype with joint-torque feedback was then assembled and tested.

The activity of the second part of the project focused the study of controllers for this experimental platform. The enhanced back-drivability of the robot make it a suitable platform to investigate topics such as dynamic tasks. These tasks are generally challenging and their planning is not straightforward. To this end several algorithms, based on machine learning techniques, were implemented and tested in simulation. This work is intended to lay the bases for future implementations on the real robot. Out of these algorithms the iLQG [Todorov and Li, 2005] performed rather well, and is therefore considered an interesting and promising research topic.

Backdrivable robots

1

The back-drivability problem

It is nowadays widely accepted that back-drivability is a fundamental prerequisite for a robotic system to interact with a surrounding unmodeled environment. Standard industrial robotic manipulators completely lack this property being designed specifically for precise trajectory tracking. For this application domain it is extremely helpful to dispose of stiff robots that minimize unwanted link or joint displacements. Moreover non-back-drivability might be useful in the case of joints that remain stationary most of the time since they can be unpowered but will nonetheless maintain their position. The same considerations however do not hold for robots intended to physically interact with the environment or with humans.

As everyday experience teaches us we continuously interact with the world around us in a stable and robust manner. Moreover if our limbs are perturbed by external forces our musculoskeletal structure is capable of absorbing the momentum caused by this interaction. On the other hand the vast majority of robots, with the exception of those specifically designed for interaction purposes, behave differently. These robot rely on the mainstream design paradigm which makes use of rotational electro-magnetic motors. These devices deliver power efficiently at high speed and low torque regimes. It is thus necessary to couple them with devices capable of converting the power in a useful range. For this purpose gear trains are used to obtain the mechanical advantage to amplify torques and reduce angular velocities to useful regimes.

As it is known, however, interactions with robots actuated by highly reduced motors feel unnatural as these robotic systems tend to be extremely stiff. It is thus often said that it is hard to back-drive them.

These examples broadly describe the concept of back-drivability but do not give a rigorous definition of what it is. Researchers have in the past proposed several definitions. Among the first significant contributions in this field are those by Salisbury and Townsend [Townsend, 1988; Townsend and Salisury, 1993]. The authors argue that the back-drivability is both acceleration and velocity dependent:

> "A mechanism which has good acceleration-dependent back-drivability generates only small inertia-induced contact forces when accelerated by the contact. [...]
>
> Similarly, a mechanism which has good velocity-dependent back-drivability generates small friction-induced forces in response to imposed endtip velocities."

This definition is only qualitative as the authors do not provide any quantitative values for "small inertia-induced contact forces" and "small friction-induced forces".

More recently similar concepts were elucidated by Ishida and Takanishi in their work on the development of improved actuators for the Sony SDR robot [Ishida and Takanishi, 2006]. In this work they define back-drivability as:

> "The level of easiness of the transmission from the output axis to input axis of the movement which is occurred at the output axis by the force which is added to the output axis in case of actuators or power transmission mechanism."

and they propose several methods to improve it. The authors however argue that back-drivability mostly counts at low joint angular velocities, thus viscous frictional torques and inertia torques are generally less relevant; they therefore redefine back-drivability as:

> "The necessary torque value to start up the rotation of the gear from the output axis, in other words, the value of rotational torque which is measured when the gear is rotated from the output axis with very low speed near zero."

The former definition has the great advantage of allowing to quantify numerically the back-drivability of a system, but mostly focuses on the performances of the speed reduction mechanism not taking into account the complex dynamic behaviours caused by the other elements of the system. These definitions suggest how the back-drivability of a robotic system heavily depends on its actuators. Moreover they point out that it is an inherent, open-loop property of a robotic system.

1.1 Efficiency

The problem of defining back-drivability might also be considered from an "energy balance" point of view. Robotic systems are generally forward drivable, that is their actuators can transfer work (e.g. kinetic energy) to the robot links and to the external environment, thus altering their states. On the other hand, back-drivability can be regarded as a measure of how hard it is to transfer energy from the external environment into the robotic system. First of all let us define a force[1] as "driving" if it produces positive work W_d in a machine. Conversely a force shall be defined as "resisting" if it produces negative work W_r. Friction forces cause a passive resistance to motion therefore produce negative work W_l. The energy balance equation can thus be written as:

$$W_d = W_r + W_l \tag{1.1}$$

The loss of energy in a system can be generally quantified with the efficiency index η defined as:

$$\eta = \frac{W_r}{W_d} \tag{1.2}$$

where W_r is the actual work in the machine and W_d is the amount of work in the machine in ideal conditions (i.e. in the absence of frictional forces).

A situation that is likely to happen in a mechanism is a decrease of magnitude of the driving force. In certain situations the resisting force might be high enough to initiate backward movement. From energetical considerations we can discern when motion in a machine can be revered. Let us define the "backward" efficiency η' as the ratio between resisting work in backward motion W_r' and driving work in backward motion W_d'. The amplitude of driving forces in backward motion generally coincides with that of resisting forces in direct motion. This hypothesis can be formulated as:

$$W_r \approx W_d' \tag{1.3}$$

[1]In the present section forces and torques, intended as generalized forces, are going to be referred to as forces.

hence:

$$\eta' \approx \frac{W_r'}{W_r} \tag{1.4}$$

Being W_l and W_l' the works dissipated in direct and reverse motion respectively, the loss of efficiency in direct and backward motion can be respectively defined as:

$$1 - \eta = \frac{W_l}{W_d} \tag{1.5}$$

$$1 - \eta' \approx \frac{W_l'}{W_r} \tag{1.6}$$

If the two are compared:

$$\frac{1-\eta'}{1-\eta} \approx \frac{W_l'}{W_l}\frac{W_d}{W_r} \tag{1.7}$$

results in

$$\frac{1-\eta'}{1-\eta} \approx \frac{W_l'}{W_l}\frac{1}{\eta} \tag{1.8}$$

Let us denote with k the ratio W_l'/W_l; Eq.1.8 can be compacted as:

$$\eta' \approx \frac{\eta(1+k) - k}{\eta} \tag{1.9}$$

This allows to draw quantitative conclusions: if $\eta < k/(1+k)$ the reverse motion efficiency becomes negative (i.e. $\eta' < 0$). This, in turn, implies that reverse motion is impossible. As generally k is in the order of 1, reverse motion is possible when the efficiency of direct motion η is at least higher than 0.5. If the condition described by Eq.1.3 holds, these considerations help to estimate under which efficiency values a mechanical system is not back-drivable. In the following sections an alternative perspective on the problem of back-drivability will be presented and discussed with the help of simple examples.

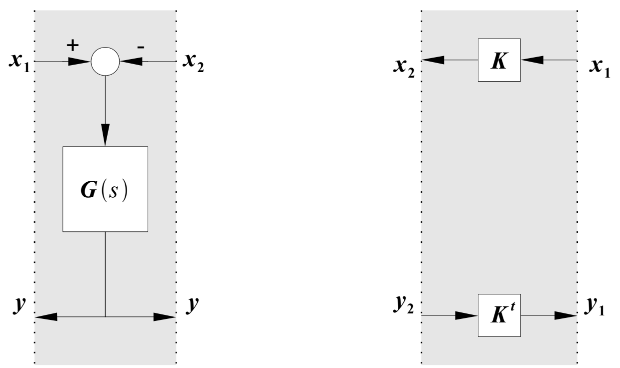
1.2 Power oriented graphs

In the current and the following chapters several examples of physical systems will be presented. To derive the block-diagram representations, state-space equations and transfer functions of these systems the power oriented graphs (POG) formalism will be adopted because of its simplicity and ease of application. This section will thus present a general overview of this technique.

The bond graphs (BG) approach is a fairly know graph-based method for modeling physical systems [Karnopp et al., 2000; Paynter, 1961]. It is based on the concept of power interaction between different systems. In the years, a formal graphical language has also been developed to support physical modeling in different domains. This technique has however some drawbacks:

- it may require more than a dozen different symbols to model a physical system
- it is not easily intelligible
- it requires a classification of the "power variables" into "effort" and "flow"
- its computational implementation is not straight-forward

A less known alternative technique to BG is the power oriented graphs (POG) framework [Zanasi, 1991, 1993, 1994; Zanasi and Salisbury, 1992] (the reader shall refer to these references for mathematical details). As for BG, POG use power interaction between subsystems as basic concept for modeling. POG are constructed by combining in a modular way two basic block components. These two components, named "elaboration block" and "connection block", are respectively represented in Fig.1.1 (a) and (b). The main feature of this representation is the direct correspondence between



(a) (b)

Figure 1.1: Modular POG blocks. The figure represents the basic construction block for a POG; (a) represents a "elaboration block" whereas (b) represents a "connection block". Dotted lines represent "Physical sections".

dual system variables and the corresponding power flow. The inner product of the two variables passing through a "physical section" (e.g. $\boldsymbol{x}$ and $\boldsymbol{y}$), represented in Fig.1.1 as a dotted line, has the physical meaning of the power P $[W]$ flowing through the section:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^t \boldsymbol{y} = P \tag{1.10}$$

Because of this feature POG allow a neat representation of hybrid system, where energy is transformed from a physical domain to a different one (e.g. the electrical to mechanical conversion of a electric motor). The elaboration block is used to represent elements that can store and dissipate energy (e.g. a capacitor, a resistor, a spring or a moving mass), while the connection block represents an energy transformation in the system (e.g. from hydraulic to mechanical). In the general case the variables are not restricted to be scalar: POG are therefore also suitable to represent multi-dimensional systems. Moreover the systems' transfer functions can be easily computed by applying Mason's formula [Chen, 1997] which allows the systematic calculation of a graph transmittances (see [—, 1960] for details). As POG are directed signal flow graphs, their graph transmittances directly correspond to the transfer function between different system variables. Power oriented graphs thus have the following advantages over bond graphs:

- they require only two elements for the graphical representations
- they do not require a classification of power variables
- they allow the modeling of a wide variety of systems in different energetic domains
- they are easily readable
- their computational implementation is easy (becomes straight-forward with graphical block diagramming tools such as Scicos or Simulink)

Finally if an elaboration block represents a linear transformation, i.e:

$$\boldsymbol{G}(s) = [\boldsymbol{M}s + \boldsymbol{R}]^{-1} \tag{1.11}$$

with $\boldsymbol{M}$ symmetric and positive definite, the energy stored by the elaboration block E_s can be directly computed as:

$$E_s = \frac{1}{2}\boldsymbol{y}^t \boldsymbol{M} \boldsymbol{y} \tag{1.12}$$

and the power dissipated P_d takes the form:

$$P_s = \boldsymbol{y}^t \boldsymbol{R} \boldsymbol{y} \tag{1.13}$$

In case all the elaboration blocks are linear there is a direct correspondence between POG representations and the state space description of the system which can be obtained simply by applying Masons' formula and rewriting the equations in matricial form.

1.3 Problem statement

Let us consider the dynamics of a rigid body rotating on an axis within the POG framework; this simple system would be represented by the block diagram represented in Fig.1.2 where τ_m denotes the actuator torque, τ_e denotes the value of torques applied externally and J_l denotes the moment of inertia of the limb. The dynamic equations

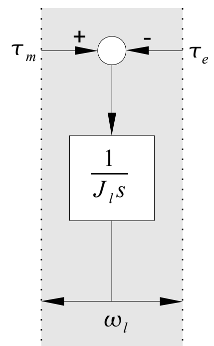


Figure 1.2: Simple joint model. The figure represents a simple joint model with POG.

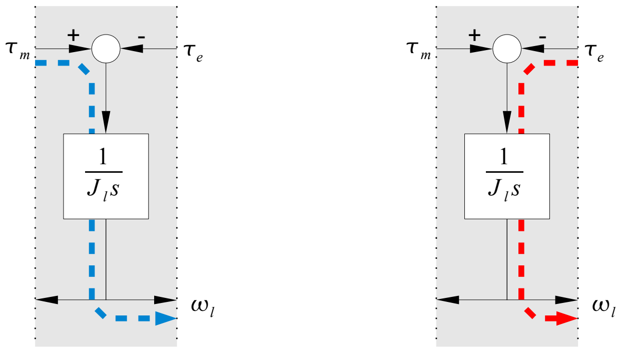
take in this case the simple form:

$$J_l \dot{\omega}_l = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \tau_m \\ \tau_e \end{bmatrix} \tag{1.14}$$

In this context the forward-drivability $F(s)$ of a robotic system is defined as the transfer function from actuator torque τ_m to the driven link angular velocity ω_l (as represented graphically in Fig.1.3(a)). On the other hand, the back-drivability $B(s)$ can be defined as the relation between the external torques τ_e, perturbing the link from the outside, and the resulting link acceleration. In this case:

$$F(s) = B(s) = \frac{1}{J_l s} \tag{1.15}$$

The example given above does not take into account the fact that actuator torques



(a) (b)

Figure 1.3: Forward and backward drivability. The figure gives diagrammatic representation of forward drivability (a) and backward drivability (b).

cannot be generated ideally as in Fig.1.2 but have to be converted from other power sources and the devices doing this have a mechanical admittance of their own.

Human limbs are generally considered a classical example of "back-drivable joints": by analyzing their physical properties we can draw base-line indications against which to compare robotic systems.

Several studies regarding the admittance and impedance characteristics of human limbs are available in literature. Most authors agree that joint stiffness can be approximated as linear function of joint displacement; although this assumption simplifies resulting models it might not always be applicable [McMahon, 1984]. Joint viscosity on the other hand can be modeled with viscous friction models [Oatis, 1993; Prochazka et al., 1997] or with a combination of viscous and Coulomb type friction [Venture et al., 2007]. As an example let us consider the human shoulder joint. Since in general Coulomb friction forces are low (see [Venture et al., 2007]) this joint can be modeled as a linear spring, flywheel damped system: let K_m denote the muscle fiber stiffness, B_m

their viscous damping and being J_l the limbs' inertia the system can be represented with the POG in Fig.1.4. The system state space description can be directly derived

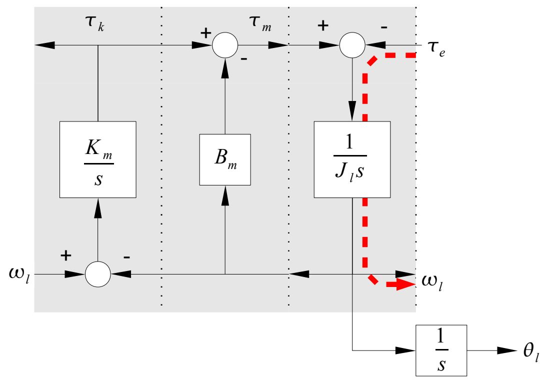


Figure 1.4: Human arm POG. The figure represents the POG block diagram for the damped spring-flywheel system described in the main text. A integration block has been added to derive the systems' angular position θ_l.

from the POG:

$$\begin{bmatrix} 1/K_m & 0 & 0 \\ 0 & J_l & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\tau}_k \\ \dot{\omega}_l \\ \dot{\theta}_l \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & B_m & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tau_k \\ \omega_l \\ \theta_l \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_m \\ \tau_e \end{bmatrix} \tag{1.16}$$

Let the state vector be $\boldsymbol{x} = [\tau_k, \omega_l, \theta_l]^t$ and the vector of inputs $\boldsymbol{u} = [\omega_m, \tau_e]^t$. Eq.1.16 can be compactly rewritten as:

$$\boldsymbol{L}\dot{\boldsymbol{x}} = \widetilde{\boldsymbol{A}}\boldsymbol{x} + \widetilde{\boldsymbol{B}}\boldsymbol{u} \tag{1.17}$$

where L is the so-called "energy matrix". Finally by inverting the diagonal matrix L Eq.1.17 can be transformed into the canonical state space representation:

$$\dot{\boldsymbol{x}} = \boldsymbol{L}^{-1}\widetilde{\boldsymbol{A}}\boldsymbol{x} + \boldsymbol{L}^{-1}\widetilde{\boldsymbol{B}}\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{1.18}$$

The transfer function representing the system back-drivability is then:

$$B(s) = \frac{\omega_l(s)}{\tau_e(s)} = \frac{1}{J_l s^2 + B_m s + K_m} \tag{1.19}$$

Approximate values for J_l, B_m and K_m for the human arm are given in [Chandler et al., 1975; Venture et al., 2007]: these parameters take approximately the values of

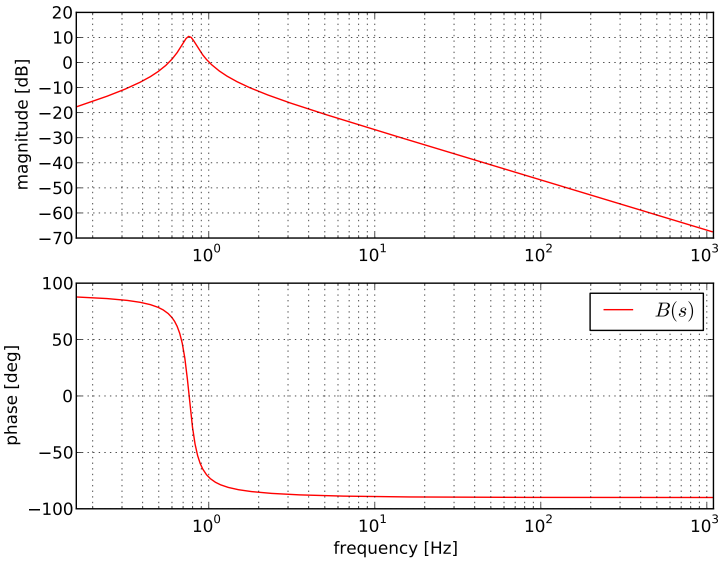


Figure 1.5: Bode diagram of the function $B(s)$. The figure shows an approximation of the Bode diagram of back-drivability characteristic of the human arm (Coulomb frictional effects have been neglected).

$0.35[kg\ m^2]$, $0.3[Nm\ s/rad]$ and $8.0[Nm/rad]$ respectively. Given these values and the equation of the system we can plot the Bode diagram back-drivability transfer function $B(s)$ which is represented in Fig.1.5.

1.4 Conventional robot design

Let us now extend these considerations to robotics: in this case the back-drivability of a robot can also be seen as the mechanical admittance of the system to motions caused by external forces.

As previously mentioned most robots today are actuated by electric motors. These motors have a typical inertia and damping which affect the overall back-drivability of the system. Moreover since electric motors are not powerful enough to directly drive the robots joints, speed reducers have to be employed to transform the motors output power into a useful speed-torque ranges. From the principle of virtual works it can be shown that a speed reduction of r produces a mechanical advantage that allows to multiply output torques by the same factor.

This solution is extremely effective both in terms of implementation complexity and cost; as a result the vast majority of robots today are actuated by electric motors with gears, and this trend is unlikely to change soon. The field of humanoid robotics makes exception: since the early '80 almost most robots have been developed in this way (see [Hirai et al., 1998; Ishida et al., 2003; Kaneko et al., 2008; Metta et al., 2005; Nagakubo et al., 2003; Tellez et al., 2008] to cite a few). Speed reduction devices are however generally affected by several non-linearities such as:

- non-linear elastic effects
- relevant coulomb type friction
- teeth mesh elasticity induced vibrations
- mechanical dead-bands
- non-linear frictional behaviours

but their use is also detrimental for what regards the back-drivability of a robotic system.

Let us consider a simple example of driving the system represented in Fig.1.2 with an electric motor and gearbox combination. For simplicity none of the non-linearities mentioned above will be modeled. Because of the motor inertia (denoted by J_m), the addition of a motor results in an increase of the total inertia of the body to be rotated. If r is the reduction ratio of the speed reduction device, the inertia of the motor will be amplified by a factor of r^2:

$$Jr = r^2 J_m \tag{1.20}$$

The total link inertia J_t will thus be:

$$J_t = J_l + J_r \tag{1.21}$$

It is easy to see that as r increases J_r tends to become significant in J_t. The speed reducer introduces relevant mechanical friction. For simplicity it will be assumed the speed reducer is of the single-stage type and that friction torques can be modeled as viscous, with the coefficient B_m. Let us finally introduce in the model the power dissipation caused by the back-e.m.f., with R_m denoting the motor winding resistance

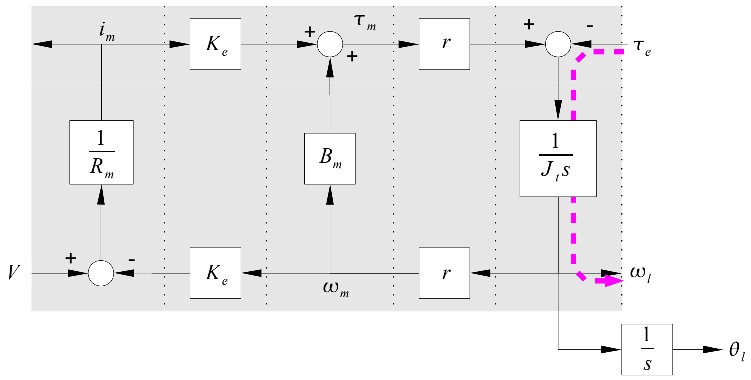


Figure 1.6: Robotic joint POG.The figure shows the POG of a robotic joint driven with a reduced electric motor. In this case both the viscous friction of the speed reducer and the motor power dissipation have been modeled.

and K_e the electric constant of the motor. The resulting robotic joint POG model is shown in Fig.1.6. The equations that describe the system in this case are:

$$\begin{bmatrix} J_l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\omega}_l \\ \dot{\theta}_l \end{bmatrix} = \begin{bmatrix} ({K_e}^2/R_m + B_m)r^2 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_l \\ \theta_l \end{bmatrix} + \begin{bmatrix} -1 & K_e r \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tau_e \\ V \end{bmatrix} \tag{1.22}$$

Eq.1.22 yields the back-drivability transfer function:

$$B(s) = \frac{\omega_l(s)}{\tau_e(s)} = \frac{1}{(J_l + J_r)s + ({K_e}^2/R_m + B_m)r^2} \tag{1.23}$$

Let us imagine we want to drive a robotic joint with the same inertia of the human arm. Reasonable values for a system actuated with a brushless motor and a Harmonic-Drive speed reducer for the various parameters are listed in Table.1.1.

The back-drivability characteristics of the human arm and of the robotic joint can now be plotted and compared. Fig.1.7 shows the linear back-drivability characteristics for a body with the inertia of a human arm. For frequencies above $2Hz$ the human arm characteristic (red curves) roughly coincides with that of a freely-rotating body (gray curves): in normal interactions, the admittance of the muscles does not affect significantly the dynamics of the human arm.

On the other hand the back-drivability characteristic of the robotic joint (magenta curve) is dominated by the effect of frictional forces: in this case the effect of the link inertia comes into play only after $20Hz$. Even after this frequency however the curve

parameter	value	units
J_m	$2.5e^{-5}$	$[kg\ m^2]$
r	100	$[-]$
B_m	0.0025	$[Nm\ s/rad]$
K_e	0.0657	$[Nm/A]$
R_m	1.38	$[\Omega]$

Table 1.1: Robotic joint parameters. The table lists the values of the parameters for the example described in the main text. These values have been taken from manufacturers' datasheet ([HarmonicDrive, 2010; Kollmorgen-DanaherMotion, 2010]) and are typical for a robotic joint actuated with a brushless electric motor, and a Harmonic-Drive speed reducer combination.

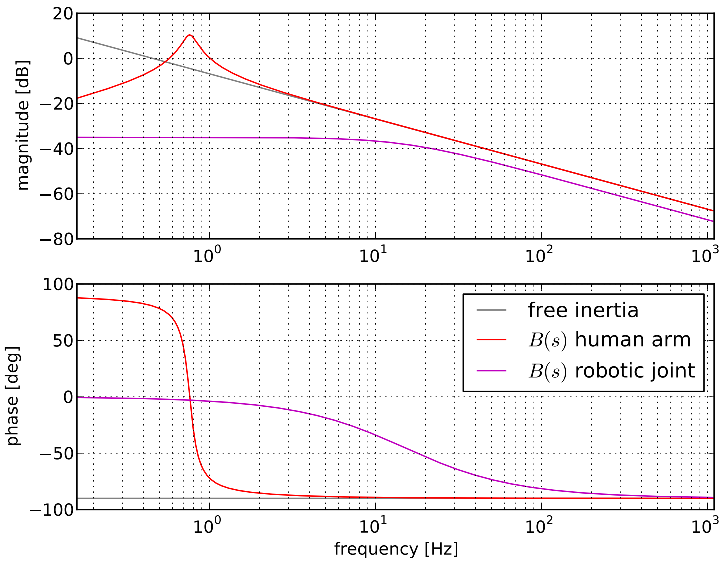


Figure 1.7: Comparison of back-drivability characteristics. The figure shows the torque to angular velocity transfer functions of a body whose inertia is equivalent to that of the human arm. The gray curve is that of a freely rotating body (Eq.1.15), the red curve is that of an inertia driven by a muscle (Eq.1.19), and the magenta curve represents the same inertia driven by a electric motor and gearbox combination (Eq.1.23).

does not tend towards the freely-rotating body characteristic because of the reflected inertia of the motors.

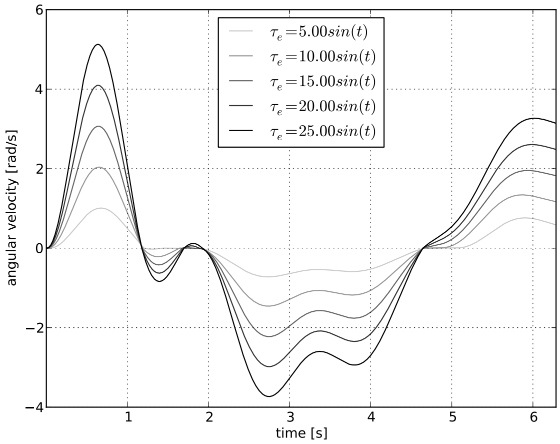
These considerations are restricted to the linear case as the important effects of Coulomb-type friction have not been considered. Speed reducers are affected by significant stiction-type phenomena (in the case of Harmonic-Drives the back-driving starting torques can reach the value of half of the total rated torque of these devices). To evaluate the effect of these torques on the dynamic behaviour of the two systems presented so far let us introduce Coulomb-type frictional torques in Eq.1.16 and Eq.1.22 which become:

$$\begin{cases} \dot{\tau}_k = K_m(\omega_m - \omega_l) \\ \dot{\omega}_l = (\tau_k + B_m\omega_l - \tau_e)/J_l + B_c sign(\omega_l) \\ \dot{\theta}_l = \omega_l \end{cases} \tag{1.24}$$

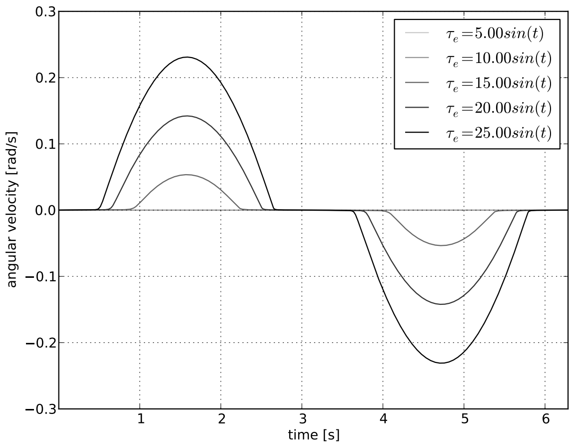
and

$$\begin{cases} \dot{\omega}_l = (({K_e}^2/R_m + B_m)r^2\omega_l - \tau_e)/J_t + B_c sign(\omega_l) \\ \dot{\theta}_l = \omega_l \end{cases} \tag{1.25}$$

The value of the friction coefficient B_c for the human arm, and the robotic joint examples can be taken equal to $0.25[Nm]$ and $20.0[Nm]$ respectively (see [HarmonicDrive, 2010; Venture et al., 2007]). The system's responses to sinusoidal external torques τ_e of amplitudes varying from 5 to $20[Nm]$ are plotted in Fig.1.8. By comparing the scale on the axes it is evident that frictional forces have a dramatic effect on the back-drivability of a robotic system if compared to the human arm. It is therefore important to conceive ways to overcome this drawback.



(a)



(b)

Figure 1.8: Systems responses to sinusoidal torque inputs. The figure shows the velocity profiles caused by a sinusoidal externally applied torque for the human arm system (a) and the robotic joint system (b). As can be seen by comparing the scale of the y-axes, the response of the robotic joint is totally dominated by the effect of Coulomb-type frictional phenomena.

2

State of the art

The present chapter is intended to review the state of the art of back-drivable robots. It will basically focus on the technological approaches that have successfully been implemented in complex (high number of DoF) anthropomorphic robotic platforms. Thus this short review is to be considered far from comprehensive, but should cover most of the relevant technologies conceived to improve the back-drivability of robotic systems. The different approaches will be presented in chronological order to describe the evolution of robotic technologies for physical interaction.

As seen in the introductory chapter, it is fundamental to increase the mechanical admittance of a robot if we want it to operate safely and robustly in unstructured environments, especially if it is to cooperate with humans. As a general approach, people developing robots for the aforementioned applications try to introduce compliance [Van Ham et al., 2009]. Compliance can be introduced passively by designing actuators coupled with spring-like elements or actively. In the latter case an appropriate controller is designed to increase the system's mechanical admittance. Among the first to investigate the problem were Wu and Paul in [Wu and Paul, 1980].

As an example, let us consider the simplified 1DoF low admittance robotic joint described in section 1.4. Let us further detail the model by introducing a joint torque sensor as an elastic element of stiffness K_s. The driven link inertia J_l and the motor's rotor inertia J_m are now decoupled: it is thus necessary to add an additional inertia elaboration block to the POG. This system can be represented with the POG shown in Fig.2.1(a).

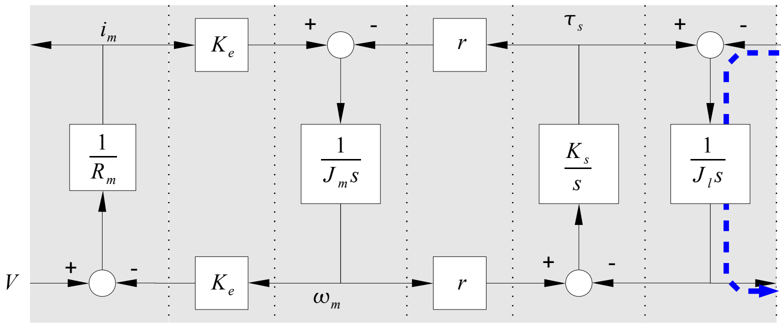


Figure 2.1: Simple 1DoF system example. The figure shows the POG for the robotic joint described in the main text. The back-drivability characteristic is represented as a dashed blue arrow.

The graphic representation yields the system's dynamic equations:

$$\begin{bmatrix} J_m & 0 & 0 \\ 0 & 1/K_s & 0 \\ 0 & 0 & J_l \end{bmatrix} \begin{bmatrix} \dot{\omega}_m \\ \dot{\tau}_s \\ \dot{\omega}_l \end{bmatrix} = \begin{bmatrix} -K_e^2/R_m & -r & 0 \\ r & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_m \\ \tau_s \\ \omega_l \end{bmatrix} + \begin{bmatrix} K_e & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} V \\ \tau_e \end{bmatrix} \tag{2.1}$$

The open-loop back-drivability characteristic (represented as a dashed blue arrow in Fig.2.1) can be computed from the POG by applying Mason's formula:

$$B(s) = \frac{\omega_l(s)}{\tau_e(s)} = \frac{J_m r^2 s^2 + (r^2 K_e^2/R_m)s + K_s}{J_l J_m r^2 s^3 + (J_l r^2 K_e^2/R_m)s^2 + K_s(J_m r^2 + J_l)s + K_e^2 K_s r} \tag{2.2}$$

The torque sensor allows to measure the torques exchanged between the robot and the environment. This signal can be used to design a controller $C(s)$ that closes a feedback loop as shown in Fig.2.2. If properly designed, this controller allows to actively increase the admittance of the system. As an example, let us consider the effect of a simple proportional controller: $C(s) = K_p$. Once again, applying Mason's formula to the POG, yields the back-drivability characteristic:

$$B(s) = \frac{\omega_l(s)}{\tau_e(s)} = \frac{J_m r^2 s^2 + (r^2 K_e^2/R_m)s + K_s(1 - K_p K_e r/R_m)}{J_l J_m r^2 s^3 + (J_l r^2 K_e^2/R_m)s^2 + K_s((J_m r^2 + J_l) - K_p J_l r K_e/R_m)s + K_e^2 K_s r/R_m} \tag{2.3}$$

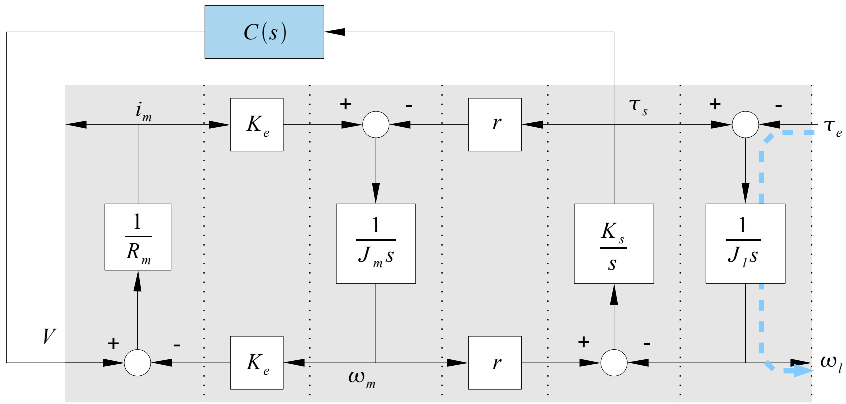


Figure 2.2: Simple 1DoF system with feedback loop example. The figure shows the POG for the robotic joint with a controller closing the joint torque feedback loop.

The different behaviours of the closed-loop system and of the open-loop system can now be compared (see Fig.2.3). The Bode diagrams of the closed loop transfer function show how the admittance of the system can be increased by control. This very basic example outlines the approach that is generally taken to improve the back-drivability of low admittance robots. The embodiment of this principle however varies, mainly depending on the type of actuation scheme selected: different actuators present different characteristics and dynamic behaviour.

This method has however several caveats. First of all the controller will have a limited and not infinite bandwidth: above this bandwidth it will not be possible affect the systems' behaviour. Secondly with standard PID control it is not possible to increase the gains at will, not to surpass the system's stability regions. There are moreover passivity related problems, as shown in [Colgate, 1988; Hogan and Buerger, 2004], that further limit the gains of a hypothetical PID controller. Model based feed-forward control is thus necessary in most cases to obtain high performances.

It is worth to mention that, in the industrial automation field, a standard solution to this problem is to equip robotic manipulators with force-torque sensors located near the end effector instead of integrating joint torque sensors. This so-called sensor "non-colocation" on the joints however introduces the significantly slow dynamics of the manipulator between sensing and actuation. This approach makes it thus difficult to

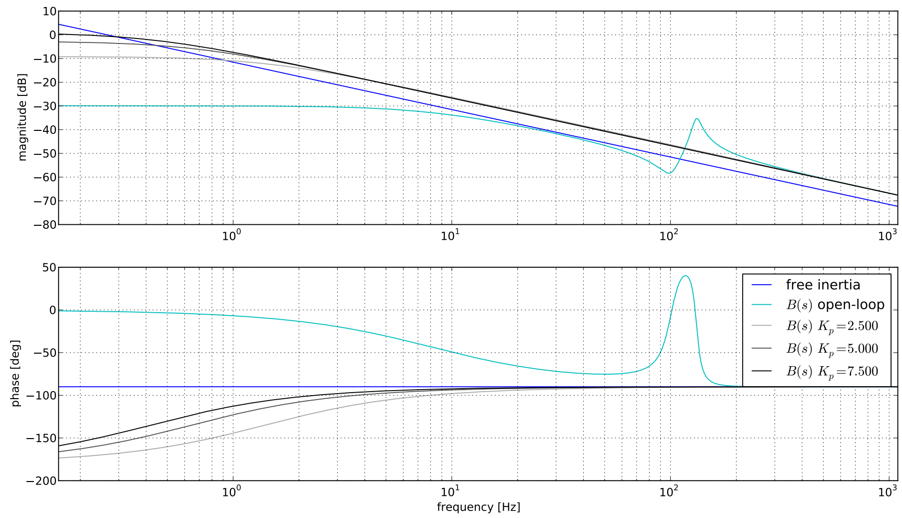


Figure 2.3: Behaviour of a joint controlled with torque feedback. The open-loop freely rotating inertia and damped system characteristics are plotted in blue and cyan respectively. Curves in tones of gray represent the admittance characteristics of the closed-loop system for varying gains.

ensure the dynamic system stability when it is controlled in feedback (see [Eppinger and Seering, 1987]) and to design high performance controllers. Moreover with this approach since forces and torques are not measurable in the arm interactions have to be limited to the end-effector.

2.1 Direct drive robots

As most of the problems described in section 1.4 are caused by speed reducers, a popular approach at the beginning of the '80s was to completely avoid their use. In direct drive (DD) robots the driven links were directly coupled with high-torque, low-speed motors [Asada et al., 1982; Asada and Youcef-Toumi, 1987]. This kind of actuation principle is extremely appealing because:

- it introduces only minimal additional inertia and friction
- it is not affected by backlash problems
- it is not affected by transmission elasticity problems

The benefits of using a transmission however are lost as well; this implies that the actuator has to be on the joint and that torques cannot be amplified. Because of technological limits, the torques provided by electro-magnetic motors are not very high [Hollerbach et al., 1992]. Even with high quality rare earths magnets, the field distribution at the motor airgap B hardly exceeds $0.5[T]$; on the other hand, effective currents per axial length i are in the order of $35000[A/m]$. From the Lorentz equation, the maximum tangential force per unit area at the airgap can be derived as:

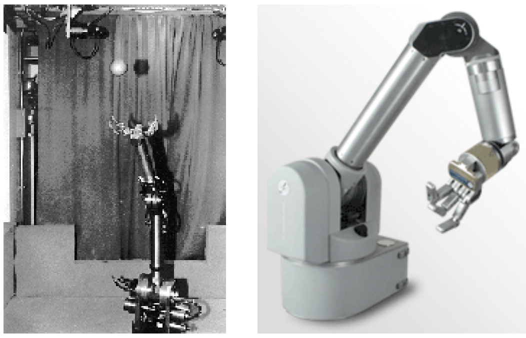
$$\frac{dF}{dA} = iB \approx 175000 \frac{N}{m^2} \tag{2.4}$$

This peak value can only be sustained for short periods of time to avoid overheating that would damage the motor. The constant rated tangential force per unit area is roughly the half of this value: $8000[N/m^2]$. This value is not sufficiently high to construct motors with torque-to-mass ratios suitable for robotic applications. The design of DD robots is thus complicated in several ways. In a serial manipulator configuration, every motor is a load for the previous one: the final weight distribution is so unfavorable that a robot can hardly withstand gravity induced torques. This in turn implies that a robot of this type will have to be designed with a very particular configuration and will only be capable of moving very low payloads (see [Asada and Kanade, 1981; Youcef-Toumi, 1985]). Motor overheating is also a problem since the robot is generally required to exert high torques for prolonged periods of time. Finally, contrary to geared motors, the dynamics of a DD joint are not dominated by frictional phenomena and behave as underdamped systems. Since the effect of inertia variations (which depend on the manipulator configuration) will not be attenuated by relevant frictional forces, it is generally more complex to design stable controllers which ensure closed-loop stability. Standard PID control is not sufficient to this purpose therefore DD robots require model based controllers [An, 1986]. Despite all these drawbacks this technology is still interesting and some research in this field is still being pursued [Aghili et al., 2002, 2007].

2.2 Current controlled robots

Since DD robots were not powerful enough, researchers began, at the end of the '80s and beginning of the '90s, to consider the design of robots with low (less than 50:1) speed reduction ratios. Joints powered by lowly reduced motors have the advantage of exhibiting significantly low reflected friction and inertia; moreover Coulomb-type friction is also less relevant. Since the disturbances caused by the speed reducer become less important, motor current measurements become an effective way to measure the torques at joint level. The whole arm manipulator (WAM) proposed by Salisbury and Townsend, which was specifically designed for physical interaction, was based on this approach (see [Salisbury et al., 1991; Townsend and Salisury, 1993]). This robot (shown in Fig.2.4) first combined two very interesting design solutions:

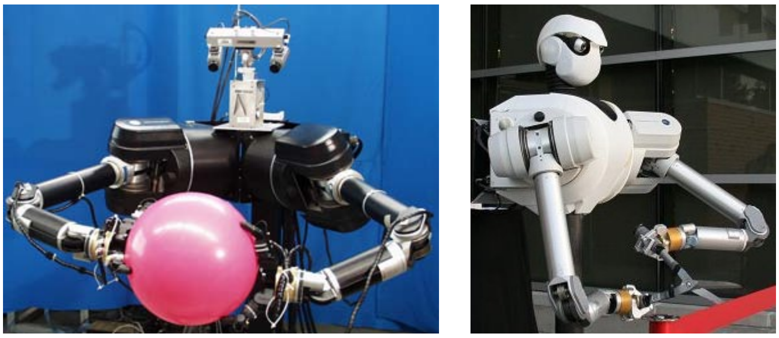
- a cable drive transmission system
- a differentially coupled proximal joint



(a) (b)

Figure 2.4: Whole arm manipulator. The figure shows the prototype WAM that Hong and Slotine used for their ball catching experiment (a), and a photo of the robot commercialized by Barrett Inc. (b).

The cable differential mechanism allows to design extremely compact joint. Cable drives use pinions, driven pulleys and idle pulleys to transmit power between different links. With the use of stepped pulleys, motion can be transmitted also between orthogonal axes. A nice feature of cable drives is the wide range of possible transmission ratios that can be obtained simply by varying the primitive diameters of coupled pulleys. However two-stage reductions are generally necessary to obtain sufficient torques but the resulting mechanism are rather complex and their maintenance is not straight-forward. Unfortunately cable drives are not infinitely stiff; this elasticity poses several problems when cable driven joints are controlled in closed loop. Nevertheless Hong and



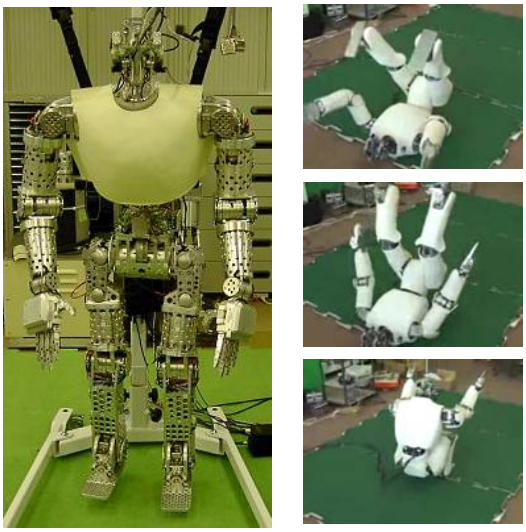
(a) (b)

Figure 2.5: WAM humanoid robots. The figure shows a photo of the Dexter robot (a) and of the Iowa State University robot (b), which both employ WAM commercialized by Barrett Inc. for their arms.

Slotine were able to demonstrate dynamic tasks with a prototype of the MIT WAM such as ball catching and throwing [Hong and Slotine, 1995]. Currently two humanoid robots comprise the commercial version of the WAM for their arms: the Dexter robot at the University of Massachussets Amherst Fig.2.5(a), and the Iowa State University humanoid Fig.2.5(b). Whenever the reduction ratio is low, current feedback is a viable alternative to measure joint torques. It is worth mentioning that this principle was also employed by Nagakubo et al. for the development of the ETL humanoid robot (see Fig.2.6(a)) [Nagakubo et al., 2000, 2003]. This robot was used to demonstrate complex dynamic tasks such as the whole body rising shown in Fig.2.6(b).

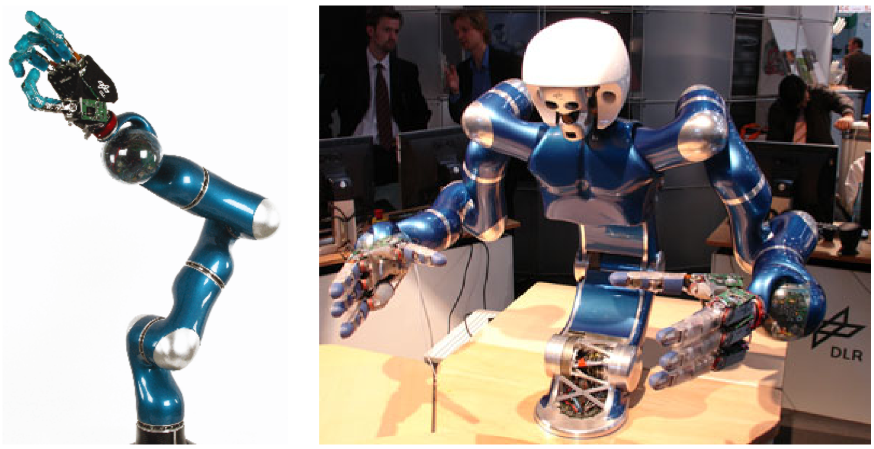
2.3 Joint torque controlled robots

A possible solution to cancel the detrimental effects of friction, reflected inertia and other non-linearities which generally affect robots is to equip them with joint-level torque control. Since actuation bandwidth is inherently limited by the friction and non-linearities in the actuator-transmission system, first prototypes mainly succeeded in reducing joint friction effects [Holmberg et al., 1992; Vischer and Khatib, 1995]. To obtain high performance levels it was necessary to integrate actuator, transmission and sensors with concurrent design methodologies [Hirzinger et al., 2000]. This approach was proven to be effective with the realization of the DLR light-weight robot series (see Fig.2.7) [Hirzinger et al., 2001, 2000, 2002]. A drawback of this approach is that,



(a) (b)

Figure 2.6: The ETL humanoid robot. The figure shows a photo of the full-body ETL humanoid robot (a), and some frames of a whole body rising sequence (b).



(a) (b)

Figure 2.7: DLR robots. The figure shows two photos of robots developed by the German aerospace centre (DLR). A photo of the third version of the light-weight robot series is shown in (a) whereas photo (b) shows the Justin humanoid robot whose arms are two of the manipulators shown in (a).

whereas at low frequencies the manipulator can be controlled to exhibit near-infinite mechanical admittance, above the control bandwidth its dynamic behaviour cannot be changed. In this situation, the response of a torque-controlled manipulator is governed by its open-loop characteristic that in practice does not differ from that of traditionally designed robots. Although it is still difficult to obtain inherent safety, high inherent admittance and good tracking performance over a wide range of frequencies, joint torque control has shown to be a very effective way to improve the back-drivability of robotic manipulators [Albu-Schaffer et al., 2003].

Transmission design, and in particular the choice of speed reducers, is critical in joint torque controlled robots. The selection of multistage planetary gearings is a common option. This type of speed reducers generally have a considerable amount of backlash (specially for gear ratios inferior to 100:1). When, in these devices, the sense of motion is inverted, this backlash generates torque spikes which eventually lead to system instability. Although this problem can be solved by preloading the gears, this often leads to rapid wear, higher friction and potentially shorter mechanism life.

The backlash problem of planetary gear heads can be avoided by using Harmonic-Drive speed reducers. These however are characterized by a low stiffness which introduces a resonant frequency which often falls within the controller's bandwidth. This problem, if not accounted for, also potentially leads to system instabilities.

2.4 Series elastic actuators

An alternative solution to increase the mechanical admittance of robots are series elastic actuators (SEA). In this approach, the dynamics of scarcely back-drivable conventional actuators are decoupled from those of the driven link by interposing between them an elastic element. With this method the high frequency admittance of the robotic joint is limited to the stiffness of the elastic coupling. On the other hand, the behaviour of the system at low frequencies can be steered by a controller that exploits the elastic element deflection as a measure of joint torque: this feed-back loop allows to regulate the mechanical admittance of the joint. This approach was first introduced by MIT researchers Pratt and Williamson which proposed their first prototype in 1995 [Pratt and Williamson, 1995; Williamson, 1995]. The earliest versions of these actuators were developed for linear motions although several "rotational" versions were developed successively. The dynamics of a 1DoF SEA robotic joint can be represented (see Fig.2.8) and studied with the POG formalism. The POG yields the following equations describ-

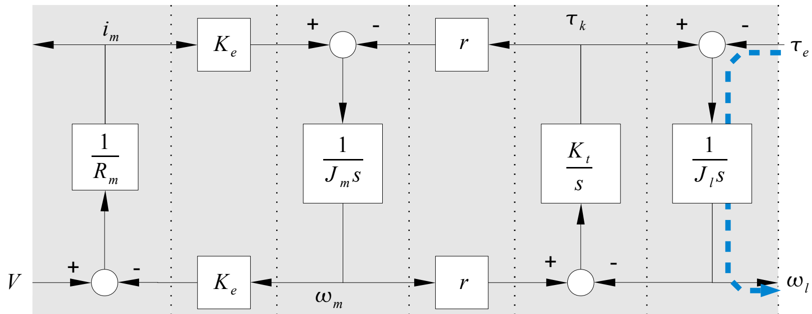


Figure 2.8: POG representation of a SEA joint. The figure represents the simplified block diagram of a SEA joint with the POG formalism.

ing the systems' dynamics:

$$\begin{bmatrix} J_m & 0 & 0 \\ 0 & 1/K_t & 0 \\ 0 & 0 & J_l \end{bmatrix} \begin{bmatrix} \dot{\omega}_m \\ \dot{\tau}_k \\ \dot{\omega}_l \end{bmatrix} = \begin{bmatrix} -K_e^2/R_m & -r & 0 \\ r & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_m \\ \tau_k \\ \omega_l \end{bmatrix} + \begin{bmatrix} K_e & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} V \\ \tau_e \end{bmatrix} \tag{2.5}$$

where R_m, J_m, J_l and K_t represent the motor winding resistance, the motor and link inertias and the transmission stiffness respectively; ω_m, ω_l, θ_l and τ_k represent the motor angular velocity, the driven-link angular velocity, the links' position and the transmitted torque respectively; K_e and r represent the electric constant of the motor and the transmission ratio of the speed reducer V is the motor excitation voltage and τ_e denotes externally applied torques. Interestingly, the SEA and the joint torque control approaches are extremely similar and even the differential equations which describe their dynamics are the same. The back-drivability characteristic of these systems is:

$$B(s) = \frac{\omega_l(s)}{\tau_e(s)} = \frac{J_m r^2 s^2 + (r^2 K_e^2/R_m)s + K_t}{J_l J_m r^2 s^3 + (J_l r^2 K_e^2/R_m)s^2 + K_t(J_m r^2 + J_l)s + K_e^2 K_t r} \tag{2.6}$$

Given Eq.2.6, the open-loop characteristics of a torque controlled joint and of a SEA joint can be compared in Fig.2.9. Parameter values are taken from Table.1.1. In both cases the maximum output torque above the open-loop mode of the system decrease at the rate of $20dB/dec$ regardless of the controller used. In the case of SEA, this

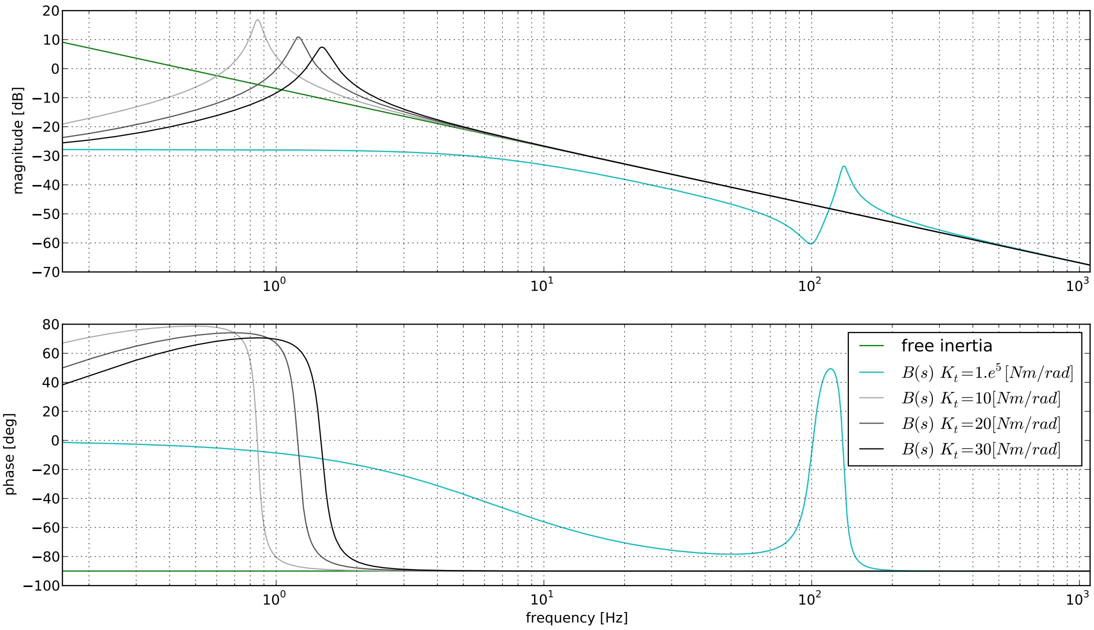
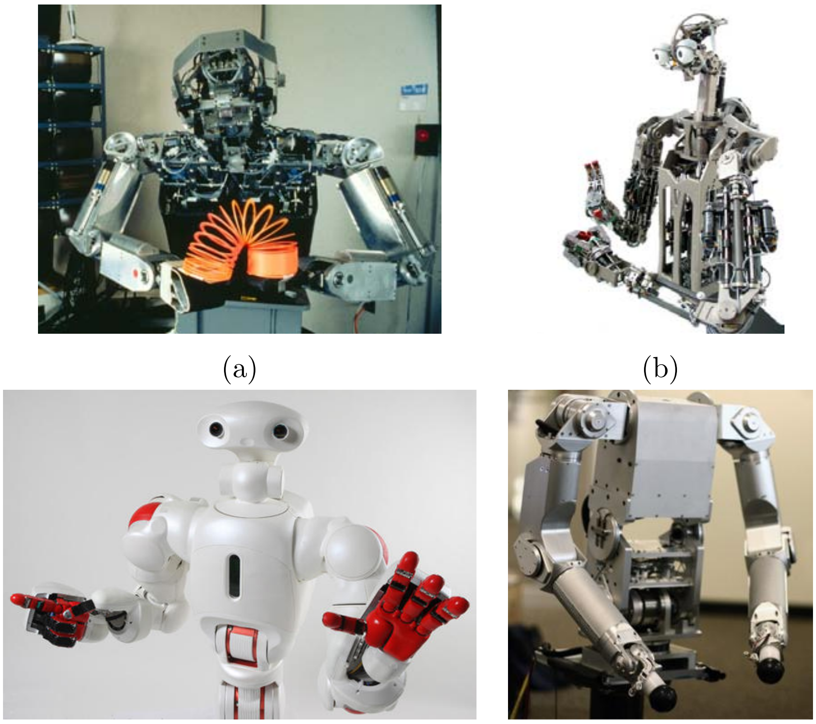


Figure 2.9: Open loop system characteristics. The figure compares the open loop behaviour of joint torque controlled and SEA systems. Joint torque controlled systems have transmission stiffness values in the order of $1.e^5[Nm/rad]$: a typical back-drivability characteristic is plotted in cyan. SEA systems have lower transmission stiffnesses: three curves in tones of grey are plotted for varying values of K_t. For comparision the characteristic of a freely rotating body is also shown (green curves).

presents a fundamental physical limitation of the actuator as the open-loop modal frequencies are rather low. Since the elastic element stiffness determines this frequency, its choice presents a tradeoff between actuator bandwidth and back-drivability. Their main difference between the two methods is the principle on the basis of which the stiffness of the transmission should be chosen. In the joint torque control approach, the transmission stiffness is designed to be as high as possible; compliant behaviours can be introduced by control below the controller bandwidth. On the other hand, SEA joints are designed for open-loop compliance; "stiff" behaviours can be added by control at frequencies below the first modal frequency of the system.

Few years after their first introduction SEA were employed for the humanoid robot COG developed by Brooks et al. [Brooks et al., 1999] at the MIT AI lab. The COG project was among the first to strongly emphasize the concpet of embodiment and incremental learning by interaction.



(a) (b)

(c) (d)

Figure 2.10: SEA humanoid robots. The figure shows photos of four humanoid robots actuated with SEA: COG (a) and DOMO (b) by the MIT AI lab, TWENDY-ONE (c) by Waseda University, and the compliant arms by Meka Robotics Inc. (d).

A group from the same laboratory later developed DOMO [Edsinger-Gonzales and Weber, 2004]. As COG, DOMO is an upper-torso humanoid robot developed to study robot manipulation in unstructured environments.

More recently a group at Waseda University undertook the development of the WENDY robot, and has recently announced the finishing of its improved version TWENDY-ONE [Iwata and Sugano, 2009] which features rotational SEA at every joint.

Another example of SEA robot is the humanoid developed by Meka Robotics, which is now being commercialized as a research platform for embodied cognition.

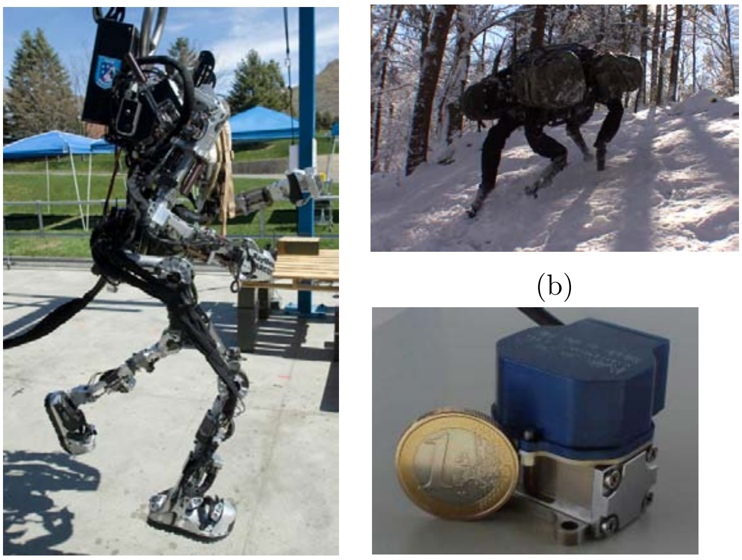
2.5 Servo-hydraulic systems

An alternative to generating torques with electro-magnetic motors is to employ hydraulic power. Hydraulic actuators have the highest torque-to-mass and power-to-mass ratios currently available [Constantinos Mavroidis and Mosley, 1999; Hollerbach et al., 1992]; these characteristics make them appealing for the design of autonomous robots. Controlling accurately the pressure and flow rate in a hydraulic system is however extremely challenging. The fluid in the system can either be controlled by servo-valves (which are generally electrically actuated) or directly by variable-displacement volumetric pumps. Servo-valves however exhibit complex, high order, non-linear behaviours such as:

- hystereses
- deadbands
- Coulomb-type friction
- non-linear spool displacement to flow rate characteristics
- internal leakages

Moreover even high performance servo-valves generally have slow responses and their bandwidths rarely exceed $30[Hz]$.

An exception is constituted by flapper-nozzle servo-valves conceived in the '50s by W.C. Moog. These valves are based on a two-stage design with a nearly frictionless



(b)

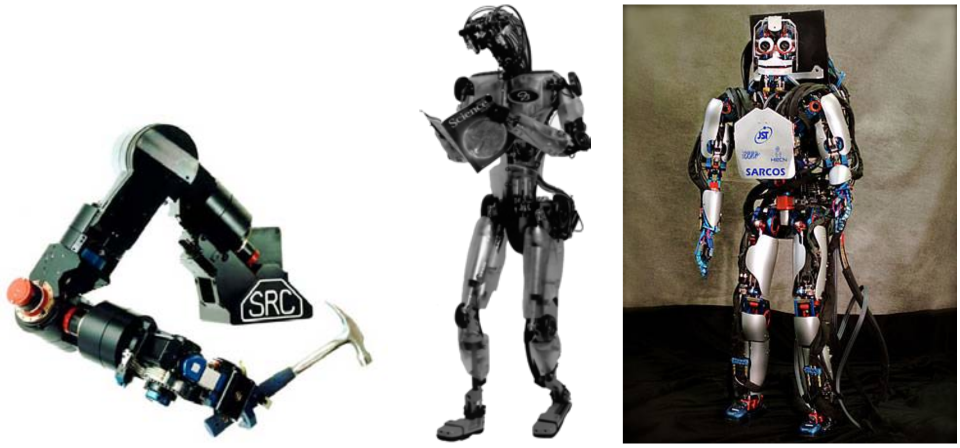
(a) (c)

Figure 2.11: Hydraulic power applications in robotics. The figure shows a photo of the Sarcos exoskeleton (a), a photo of the Boston Dynamics BigDog quadruped (b) and a photo of a compact flow control servo-valve by MOOG (c).

pilot stage. The flapper is driven by a torque motor and regulates the aperture of a variable orifice that drives second-stage spool. The spool position causes the deflection of a spring that restores the torques on the flapper thus closing the feed-back loop. These valves have been engineered into extremely compact and robust products by the U.S. manufacturer MOOG and are now available on the market (see Fig.2.11(c)). The datasheet provided by the manufacturer states that these valves have highly dynamic responses and can achieve bandwidths in the order of $200[Hz]$.

These valves allowed the development of impressive demonstrative robot prototypes such as the BigDog quadruped (Fig.2.11(b)) and the Petman biped by Boston Dynamics, and the Sarcos exoskeleton (Fig.2.11(a)). Unfortunately all these robots are the result of military research projects and most technical informations about them are regarded as strategic and are therefore kept classified.

The Sarcos company (a spin-off of the research group led by Stephen Jacobsen at the university of Utah) gained know-how with the development of animatronic robots for entertainment industry [Hollerbach and Jacobsen, 1996], which in recent years has been applied to develop several humanoid robots for AI research. One of the earliest robots



(a) (b) (c)

Figure 2.12: Sarcos robots. The figure shows photos of the robots developed by Sarcos Inc.: the Sarcos Dextrous arm (a), the DB humanoid robot (b) and the CB humanoid robot (c).

was DB (see Fig.2.12(b)), which is the acronym for Dynamic Brain, developed for the JAIST ERATO brain project [Atkeson et al., 2000]. This full body humanoid robot was not powerful enough to stand on its legs but was nevertheless used to demonstrate the acquisition of motor skill with the help of machine learning. More recently two new robots (see Fig.2.12(c)) based on the same technology have been developed for the computational brain project [Cheng et al., 2007]. Details on the control of the joints of this kind of robots can be found in [Bentivegna et al., 2008].

The examples reported so far suggest that hydraulic actuation, in combination with joint torque sensing, seems a convenient way to obtain high quality force control for robotic applications. Apart from valve non-linearities, this technology has several disadvantages which make it less appealing. First of all such a system requires to be powered by a pump unit which has in general an associated reservoir tank for the return line of the hydraulic circuit. Moreover powerful motors (typically in the order of several $[kW]$) are required to operate the pump. This results into additional bulk which negatively affects the power-to-weight ratio of the robotic platform intended as a whole. While for outdoor applications it is feasible to use a compact internal combustion engine, robots designed for indoor environments generally need to be powered by an off-board power unit which requires the robot to be tethered: these aspects negatively

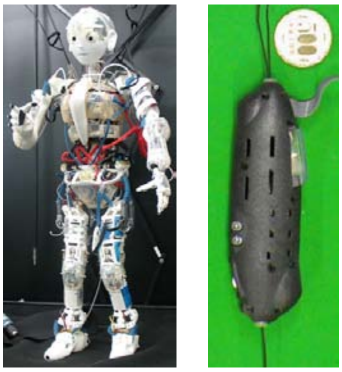
(a) (b)

Figure 2.13: Parallel drive robots. The figure shows photos of the Stanford human friendly robot (a), and of the PR2 humanoid robot by Willow Garage (b).

affect the robots' autonomy. Finally to implement high band-width controllers flapper-nozzle servo-valves are practically the only viable solution. However their extremely high cost (roughly 4000€) hampers the diffusion of hydraulic power as a solution for the actuation of robots.

2.6 Parallel drive robots

Recently, a new class of manipulators has come to integrate the parallel drive concept introduced in [Morrell and Salisbury, 1995; Sharon, 1989]. In these works the authors proposed a parallel and distributed two-stage actuation scheme for robotic joints. The basic assumption underlying this approach is that the torques to be exerted decrease along the frequency spectrum [Zinn et al., 2004]. Therefore a high admittance actuator located at the manipulator base is intended to provide the slow varying component of torques. Instead high frequency torque components are provided by a smaller less powerful and faster actuator located directly on the joint. This approach led to the development of the Stanford human safe robots [Shin et al., 2008] and the PR series personal robots [Wyrobek et al., 2008], which are shown in Fig.2.13.



(a) (b)

Figure 2.14: Musculoskeletal humanoid robot. The figure shows photos of the musculoskeletal humanoid robot Kotaro (a), and a photo of the muscle-unit actuators (b).

2.7 Musculoskeletal humanoids

Finally it is worth citing another type of back-drivable robot which is difficult to assign to any of the previous categories. Since 2002 a group at the University of Tokyo has been developing a musculoskeletal humanoid robot [Mizuuchi et al., 2002]. This robot is actuated by so-called "muscle-units" that comprise an electric motor with a planetary speed reducer, a position sensor and an optical tension sensor [Mizuuchi et al., 2004]. The robot Kotaro [Mizuuchi, 2006] and the actuator are shown in Fig.2.14. This design tackles the back-drivability problem from two sides: firstly the actuation does not reduce significantly the admittance of the robot, and secondly, the actuators can be controlled in closed force loop to further increase the robot's admittance.

3

Setup description

The scope of the current work was focused on the main joints of the arms of the iCub, neglecting for the moment the forearm and wrist joints, which are less critical for our purposes. These joints constitute a 4DoF manipulator with one rotational 3DoF proximal "shoulder" joint and a rotational distal "elbow" joint (see Fig.3.1).

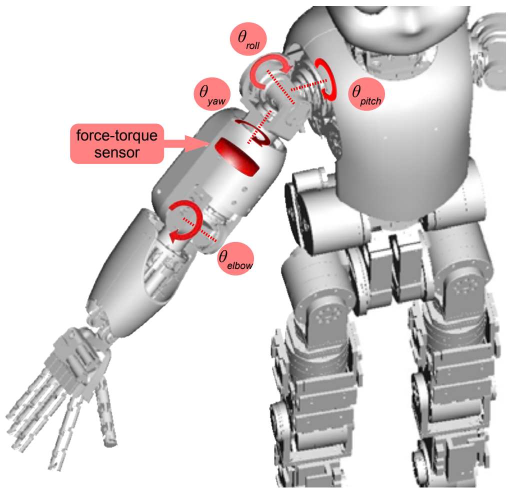


Figure 3.1: The iCub arm. A CAD view of the arm of the iCub robot with superimposed joint labels (θ_{pitch} for shoulder pitch, θ_{roll} for shoulder roll, θ_{yaw} for shoulder roll and θ_{elbow} for the elbow rotation). The figure also represents the approximate position of the six axis force torque sensor integrated in the arm.

The initial specifications for the design of the robot aimed at replicating the physical abilities of a three-year-old baby [Metta et al., 2005]. To enable natural and stable manipulation tasks, and by considering biomechanical models, it was decided, in the design phase, that the iCub arm should have had seven DoF: three at the shoulder level, one at the elbow and two at the wrist. Moreover, the mass distribution and the ranges of motion were designed to be similar to those found in humans (see Tables 3.1 and 3.2).

Joint		range of motion	
		Human	iCub
shoulder	flex/extension	[-8,+200]	[-50,+230]
	ab/adduction	[-85,+200]	[-90,+150]
	rotation/twist	[-54,+127]	[-90,+90]
elbow	flex/extension	[0,+160]	[0,+140]

Table 3.1: Ranges of motion. The table lists the ranges of motion for the various joints of the arm for a three-year-old and for iCub.

Body section	mass [kg]	length [m]
upper arm	1.15	0.15
forearm (hand included)	1.25	0.13

Table 3.2: Mass distribution. The table lists the masses and lengths of the links of the arm.

Having determined a suitable kinematic structure, the design process moved to dynamic performance criteria. An open source software package for three-dimensional rigid body dynamics was used to simulate the behaviour of the robot. More specifically, the torques normally exerted during sinusoidal and crawling movements were measured [Metta et al., 2004]. These data, whose peak values are are listed in Table 3.3, provided a baseline for the selection of the robots' actuators. These values are however rather

Joint		peak torque [Nm]
shoulder	flex/extension	40.5
	ab/adduction	18.1
	rotation/twist	7.9
elbow	flex/extension	18.6

Table 3.3: Peak torques. The table lists for the various arm joints the values of peak torques exerted by the robot to perform a crawling motion.

high: recent tests on the iCub prototype suggest that in normal manipulation tasks (e.g. while performing fast reaching movements), joint torques rarely exceed the value of 8Nm.

3.1 The motor groups

To match the torque requirements listed in Table3.3 three modular motor groups were developed: this allowed their reuse throughout the main joints of the robot. All of them are similar since they all comprise a Kollmorgen-DanaherMotion RBE 012 type brushless frameless motor [Kollmorgen-DanaherMotion, 2010] and a CSD frameless Harmonic Drive flat speed reducer [HarmonicDrive, 2010]. Harmonic Drive speed reducers allow to obtain very high reduction ratios in small spaces, are very light, and have practically no backlash. Brushless motors have have a very good power density and generally outperform conventional brushed DC motors. The use of frameless components allows to further optimize spaces and to remove the unnecessary weight of the housings.

Two different motor groups (shown in Fig.3.2) are used in the arm:

- the high power motor group: capable of delivering 40Nm of torque, it is based on the RBE 01211 motor and a CSD-17-100-2A Harmonic Drive, and has, roughly, a diameter of 60mm and a length of 50mm.

- the medium power motor group: capable of delivering 20Nm of torque, it is based on the RBE 01210 motor and a CSD-14-100-2A Harmonic Drive, and has, roughly, a diameter of 50mm and a length of 50mm.

3.2 Frame structure

The total weight design specification was particularly hard to difficult with: special care had to be taken in the design of structural elements to avoid adding mass. For what concerns the materials, the major part of the structural elements of the robot were fabricated with the Al 6082 aluminum alloy. Its internal grain structure is governed by the addition of large amounts of manganese. With its ultimate tensile strength (UTS) of 310 MPa and roughly the typical density of aluminum 2700 kg/m^3 Al 6082

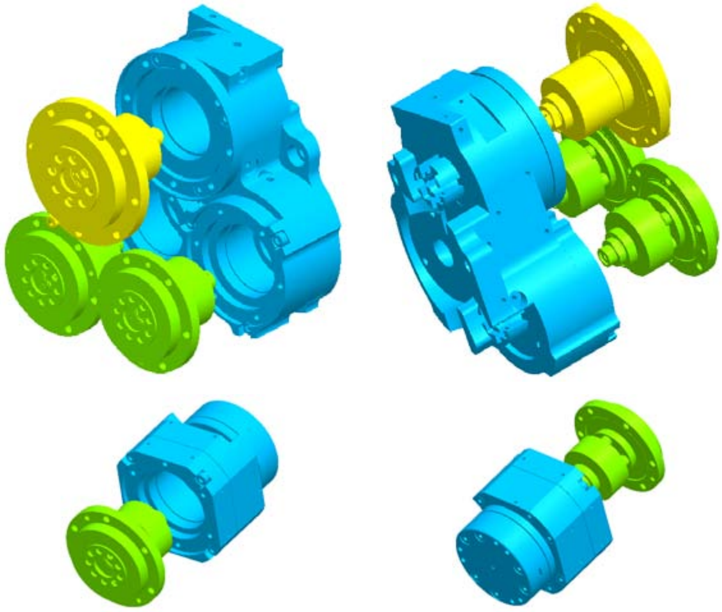


Figure 3.2: Right shoulder motor assemblies. The figure shows exploded CAD views of iCub's right shoulder motor groups (top) and the elbow motor group (bottom) from a front and rear view. The high power motor group is coloured in yellow whereas the medium power motor groups are coloured in green.

is among the best materials in the 6000 alloy series [Matweb, 2010b]. For these reasons it is considered a noteworthy structural material.

Critical components were manufactured with the Al 7075 aluminum alloy. This material is typically employed for aerospace applications because of its excellent strength to weight ratio. The use of zinc as the primary alloying element results in a strong material, with good fatigue strength and average machinability. The density of Al 7075 has a density of 2810 kg/m^3 which is slightly higher than normal aluminum; its UTS of 524 MPa [Matweb, 2010c] is comparable with that of medium quality steels and make it one of the toughest types of aluminum alloys currently available.

Finally, highly stressed parts (such as joint shafts) were obtained from the high resistance stainless steel 39NiCrMo3. This material, known in the AISI standard as AISI 9840 is a nickel-chromium-molybdenum steel, that exhibits a good combination of strength, fatigue resistance, toughness and wear resistance. Its UTS is high, around 1.2 GPa [Matweb, 2010a].

3.3 The shoulder joint

The shoulder joint is based on a cable differential mechanism similar to the one introduced by Salisbury et. al. [Salisbury et al., 1991; Townsend, 1988] and detailed in [Townsend and Salisury, 1993]. The shoulder joint has the peculiarity of having its three axes of rotation intersecting at a single point (which is a typical characteristic of robotic wrist mechanisms). In normal "serial" manipulators all the motors and speed reduction units are mounted directly on the joints, thus increasing the inertial loads on the motors. Instead, the iCub shoulder comprises a "coupled" epicyclic transmission system (shown in Fig.3.3). This mechanism allows the joint to be remotely driven: motors can thus be mounted in the proximity of the joint rather than on the joint itself. The three motors driving the shoulder are in fact housed in the upper-torso of the robot.

A mechanism of this kind has several advantages:

- large workspace,
- compact size, low weight and inertia,,
- "quasi"-spherical motion

but is affected by some drawbacks such as:

- higher mechanical complexity (higher manufacturing costs and longer assembly time),
- lower mechanical stiffness,
- less intrinsic robustness.

Cable drives allow the transmission of power between different bodies with driven pulleys, stepped pulleys, pinions, and idle pulleys and are a good alternative to gears when space is limited.

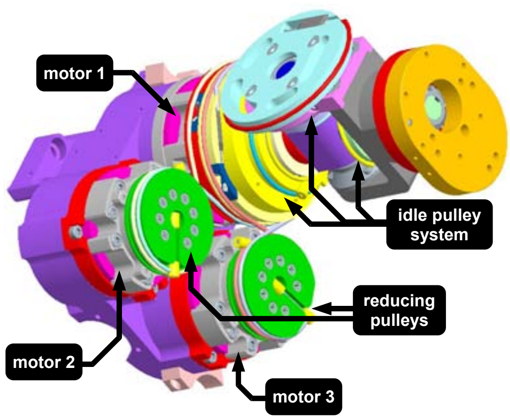


Figure 3.3: The shoulder joint. A CAD view of the shoulder joint mechanism indicating the three motors actuating the joint and the pulley system.

In the current embodiment the first motor actuates directly the first joint whereas the second and third motors actuate two pulleys that are coaxial with the first motor. These pulleys have slightly different primitive diameters thus producing a transmission reduction r equal to the ratio of their diameters:

$$r = 40mm/65mm \approx 0.615385 \qquad\qquad 1/r = 1.625 \tag{3.1}$$

The pulley motion is then transmitted to the shoulder roll and pitch joints through a second set of idle pulleys.

Since the joint is highly coupled the relations between the velocities and torques at motor and joint level are not straightforward. The technique outlined by Tsai in [Tsai,

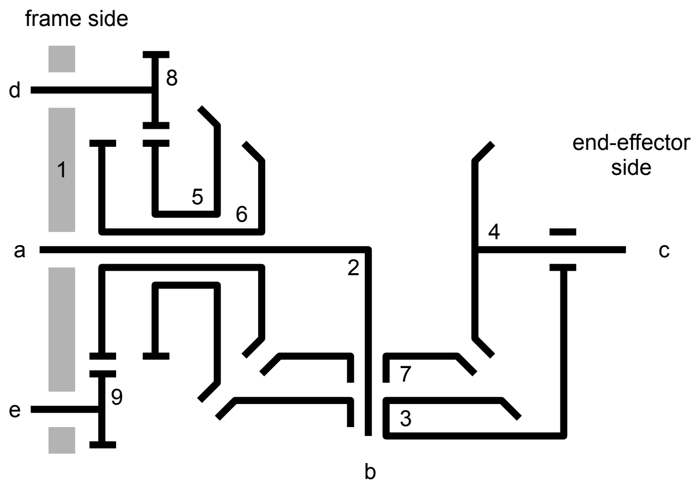


Figure 3.4: Shoulder mechanism. The figure shows the simplified functional schematic representation of the shoulder mechanism.

1999], for robotic wrist mechanisms, will be adopted to thoroughly analyze shoulder mechanism. This method is particularly convenient for the analysis of complex epicyclic transmissions such as the mechanism that constitutes the iCub shoulder. The method firstly requires the derivation of the canonical graph representation of a mechanism. This graph is then used to derive a system of linear equations with the theory of fundamental circuits. Redundant equations are finally eliminated with coaxiality condition equations.

To derive the graph representation of a mechanism it is convenient to start from its simplified representation. The functional schematic representation of the shoulder mechanism is shown in Fig.3.4. As can be seen, the mechanism is very complex. As generally done, the following simplifying assumptions were made to obtain this representation:

1. all parallel and redundant paths are shown as a single path only. Whenever a link is supported by two or more bearings on one shaft only one will be considered,

2. rigidly connected elements are considered as a single link,

3. all joints are assumed to be binary. Multiple joints are replaced with a proper combination of binary joints,

4. all transmission pairs constituted by cable drives are represented as gear pairs.

In Fig.3.4 the motor groups 1, 2, and 3 are represented as links 2, 8 and 9 respectively. For presentation clarity, the shoulder angular displacements θ_{pitch}, θ_{roll} and θ_{yaw} will be renamed as $\theta_{2,1}$, $\theta_{3,2}$ and $\theta_{4,3}$ respectively (the pedices indicate the two links whose relative displacement is being considered). To simplify the description (without any loss of generality) all transmission pairs constituted by cable drives will be represented and referred to as gear pairs.

3.3.1 Kinematic analysis

Any kinematic chain can be represented as a graph. This abstract representation has several advantages:

- graphs allow for a more organized and systematic kinematic and dynamic analysis of the mechanism,
- the topology of a mechanism can be uniquely defined (this also allows to identify similarities between various mechanism embodiments),
- some mathematical graph properties can be applied directly.

In a graph representation the vertices represent the mechanism links and the edges represent its joints. To distinguish between pair connections, the edges can be coloured and labeled. Following the convention described in detail in [Tsai, 1999], gear pairs are drawn as heavy edges, whereas turning pairs are drawn as thin edges. Turning pairs are also labeled according to their axis locations. Once the fixed link of a kinematic chain is assigned, its topological structure is completely defined: this link is generally represented in the graphs with two concentric circles.

Ambiguities in the graph representations may however arise when three or more links share a common axis. Because of the previously described assumptions, the turning pairs among coaxial links can be reconfigured at will without affecting the functionality of the mechanism. These possible alternative mechanism representations are called pseudoisomorphic graphs. To disambiguate between these different possibilities, the kinematic chain has to be derived in an univocal way, thus yielding the canonical mechanism graph. The graph is canonical if all the thin edges paths beginning from the

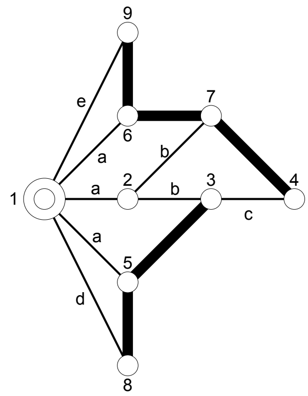


Figure 3.5: Canonical graph. The figure shows the canonical graph of the iCub shoulder mechanism. Turning pairs are represented as labeled thin edges and gear pairs as thick edges.

base link have different edge labels. The functional schematic representation in Fig.3.4 can therefore be reduced to its canonical graph which is represented in Fig.3.5.

If mechanical limits are neglected, an epicyclic gear train should allow unlimited rotations. This in turn implies that there should be no circuits formed exclusively by turning pairs (otherwise unlimited rotations would be impossible). Moreover, each gear in epicyclic transmissions has to have a turning pair on its axis to maintain a center distance between a gear pair. These considerations imply that the subgraph formed by removing all the geared edges from the graph of an epicyclic gear train, is a tree. The tree of the iCub shoulder mechanism is shown in Fig.3.6.

Adding any geared edge back to the tree, forms a circuit which is called fundamental circuit. The number of fundamental circuits in a gear train is equal to the number of gear pairs. Every fundamental circuit has a node such that all the thin edges on one of its sides have identical labels and the edges on the opposite side have different labels: this node is called transfer vertex and it corresponds to the carrier of a gear pair. The fundamental circuits and transfer vertices of the shoulder mechanism are listed in Table 3.4.

Every fundamental circuit has an associated fundamental circuit equation that relates the angular displacements of its links. The fundamental circuit equation can be

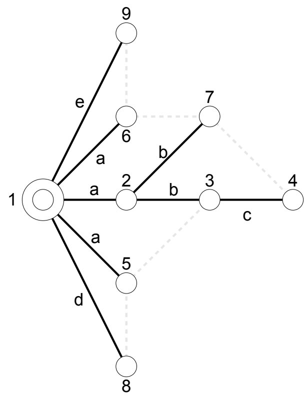


Figure 3.6: Canonical tree. The figure shows the canonical tree of the iCub shoulder mechanism. Geared pair edges have been eliminated from the canonical graph.

fundamental circuit	transfer vertex
1-5-8	1
1-6-9	1
2-3-5	2
6-7-2	2
7-4-3	3

Table 3.4: Fundamental circuits. The table lists the fundamental circuits and transfer vertices of the iCub shoulder mechanism.

expressed as:

$$\theta_{i,k} = \pm N_{j,i}\theta_{j,k} \tag{3.2}$$

where $\theta_{i,k}$ and $\theta_{j,k}$ denote the displacements of gears i and j with respect to the carrier k respectively, and $N_{i,j}$ denotes the reduction ratio of the gear pair. The sign in Eq.3.2 is determined by the sign of the rotation of gear j produced by a rotation of gear i relative to the carrier k. The fundamental circuits of the shoulder mechanism thus yield the following system of linear equations (1:1 reduction ratios are omitted for clearness):

$$\begin{cases} \theta_{5,1} = -N_{5,8}\theta_{8,1} \\ \theta_{6,1} = -N_{6,9}\theta_{9,1} \\ \theta_{3,2} = -\theta_{5,2} \\ \theta_{6,2} = -\theta_{7,2} \\ \theta_{7,3} = -\theta_{4,3} \end{cases} \tag{3.3}$$

To solve the kinematics of epicyclic gear trains, the fundamental circuit equations must be coupled with the coaxiality conditions which relates the angular displacements of coaxial links. Given three coaxial links, i, j, and k the following relation between their relative displacements holds:

$$\theta_{i,k} = \theta_{i,k} - \theta_{j,k} \tag{3.4}$$

Being 2-3-7 and 1-2-5-6 the tuples of coaxial links, the following linear equations can be derived:

$$\begin{cases} \theta_{7,3} = \theta_{7,2} - \theta_{3,2} \\ \theta_{6,2} = \theta_{6,1} - \theta_{2,1} \\ \theta_{5,2} = \theta_{5,1} - \theta_{2,1} \end{cases} \tag{3.5}$$

Combining Eq.3.3 and Eq.3.5, unwanted angular displacements can be eliminated, thus yielding as result the equations describing the relationship between the actuator displacements and the joint angles:

$$\begin{cases} \theta_{8,1} = \dfrac{1}{N_{5,8}}(-\theta_{2,1} + \theta_{3,2}) \\ \theta_{9,1} = \dfrac{1}{N_{6,9}}(-\theta_{2,1} + \theta_{3,2} - \theta_{4,3}) \end{cases} \tag{3.6}$$

Link 2 is directly an input link; the following identity can thus be written:

$$\theta_{2,1} = \theta_{2,1} \tag{3.7}$$

Finally Eq.3.6 and Eq.3.7 can be combined in matricial form:

$$\begin{bmatrix} \theta_{2,1} \\ \theta_{8,1} \\ \theta_{9,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{r} & \frac{1}{r} & 0 \\ -\frac{1}{r} & \frac{1}{r} & -\frac{1}{r} \end{bmatrix} \begin{bmatrix} \theta_{2,1} \\ \theta_{3,2} \\ \theta_{4,3} \end{bmatrix} \tag{3.8}$$

where the reduction ratios $N_{5,8}$ and $N_{6,9}$ have been substituted with their actual value r. Adopting the notation introduced in Tsai [1999], denoting the vector of input angular displacements as $\boldsymbol{\phi} = [\theta_{2,1}, \theta_{8,1}, \theta_{9,1}]^t$ and the vector of joint angular displacements as $\boldsymbol{\theta} = [\theta_{2,1}, \theta_{3,2}, \theta_{4,3}]^t$, Eq.3.3.1 can be compactly rewritten as:

$$\boldsymbol{\phi} = \boldsymbol{T}\boldsymbol{\theta} \tag{3.9}$$

3.3.2 Static analysis

The expression that relates torques generated at the actuator level and the ones effectively applied at joint level, can finally be derived by applying the principle of virtual works. Being $\delta\boldsymbol{\phi} = [\delta\phi_1, \delta\phi_2, \delta\phi_3]$ and $\delta\boldsymbol{\theta} = [\delta\theta_{2,1}, \delta\theta_{3,2}, \delta\theta_{4,3}]^t$ the actuator and joint virtual displacements respectively, let $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]^t$ and $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^t$ denote the torques exerted at actuator and joint level respectively. The virtual work δW produced by the active forces is thus given by:

$$\delta W = \boldsymbol{\xi}^t \delta\boldsymbol{\phi} - \boldsymbol{\tau}^t \delta\boldsymbol{\theta} \tag{3.10}$$

The relation for virtual displacements can be derived from Eq.3.9 as:

$$\delta\boldsymbol{\phi} = \boldsymbol{T}\delta\boldsymbol{\theta} \tag{3.11}$$

which combined with Eq.3.10 yields:

$$\delta W = (\boldsymbol{\xi}^t \boldsymbol{T} - \boldsymbol{\tau}^t)\delta\boldsymbol{\theta} \tag{3.12}$$

The system is in equilibrium if and only if, for any infinitesimal virtual displacement, the virtual work is zero thus:

$$\boldsymbol{\xi}^t \boldsymbol{T} - \boldsymbol{\tau}^t = 0 \tag{3.13}$$

which can be rearranged as:

$$\boldsymbol{\tau} = \boldsymbol{T}^t \boldsymbol{\xi} \tag{3.14}$$

It is interesting to note that this relation is independent of the mechanism orientation and is only a function of its topology. Finally, given a tuple of desired joint torques $\boldsymbol{\phi}$, the motor torques $\boldsymbol{\xi}$ can be derived with:

$$\boldsymbol{\xi} = \boldsymbol{T}^{-t}\boldsymbol{\phi} \tag{3.15}$$

being:

$$\boldsymbol{T}^{-t} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -r & -r \\ 0 & 0 & -r \end{bmatrix} \tag{3.16}$$

3.4 The elbow

The 1 DoF elbow joint is rather simple in its design. The output link is driven through a pulley system which conveys the motion from the motor group. The motor is housed at the centre of the assembly oriented 90deg with respect to the axis of rotation of the elbow. On the upper part of the elbow assembly is mounted the six-axis force-torque sensor.

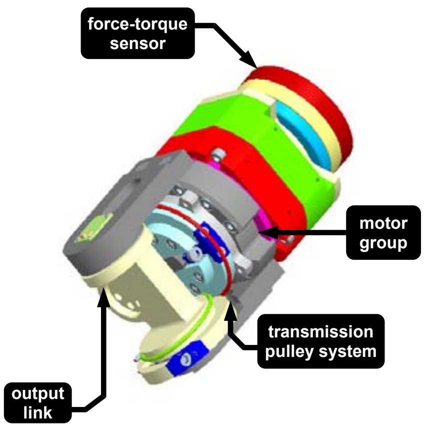


Figure 3.7: The elbow sub-assembly. The figure shows the elements of the elbow sub assembly.

3.5 Sensors and electronics

3.5.1 Joint position sensors

For what concerns position sensing, each actuator units contains three Hall effect sensors integrated in the motor stator that can be used as an incremental rotary position sensor. Moreover every joint angular position is sensed with an absolute 12bit angular encoder (employing the AS5045 microchip from Austria Microsystems).

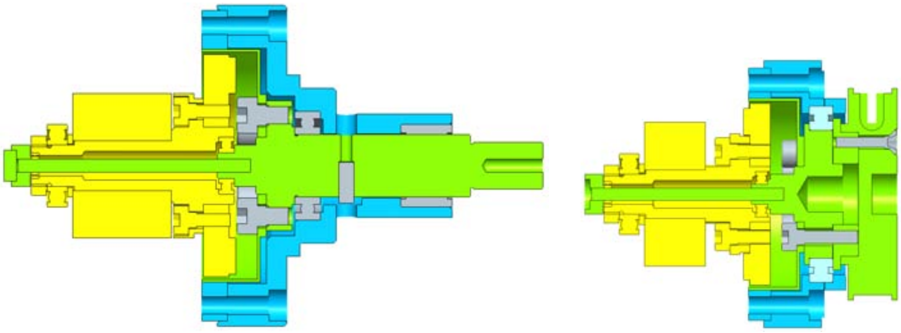
An additional problem is constituted by position sensing in the shoulder joint. Because of space limitations it was unfeasible to integrate an encoder directly on the yaw axis. The missing information is recovered by placing an encoder on the axis of motor 3. The observed positions $\boldsymbol{\theta}_o = [\theta_{pitch}, \theta_{roll}, \theta_{m3}]^t$ are then mapped to the actual joint positions $\boldsymbol{\theta}_j$ with the following transformation matrix:

$$\boldsymbol{\theta}_j = \boldsymbol{O}\boldsymbol{\theta}_o \qquad \boldsymbol{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 1/r \end{bmatrix} \tag{3.17}$$

As can be seen from Fig.3.3 there are no places to fit a position sensor on the external part of motor 1 and 2, as there are no parts in front of them which remain fixed with respect to the frame. For this reason it was necessary to locate the position sensor in the rear of the motor. To do this the “slow” motion of the output link is transmitted through the motors’ rotor hollow shaft with a thin shaft that carries the magnet for the sensor: this particular arrangement is show in Fig.3.8.

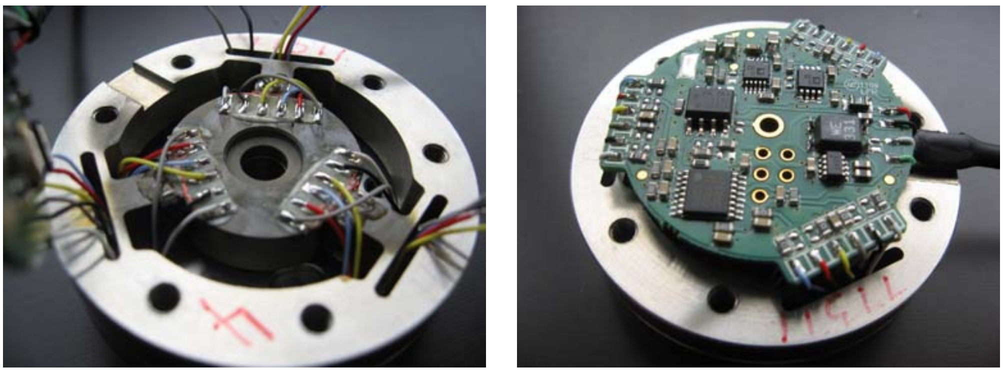
3.5.2 Six-axis force-torque sensor

The iCub arm also comprises a six-axis force-torque sensor [Tsagarakis et al., 2007]. The sensor load cell is based on a three spoke structure machined from stainless steel Fig.3.9(a). On each side of each spoke is mounted a semi-conductor strain-gauge: opposite strain gauges are connected in a half Wheatstone’s’ bridge configuration. The sensor integrates an electronic board for the data acquisition and signal conditioning Fig.3.9(b). The board samples six analog channels with an INA118 instrumentation amplifier: each input is connected to one of the six aforementioned half Wheatstone’s’ bridges. The analog to digital conversion is performed by an AD7685 converter on the multiplexed signals of the six channels. It is besides possible to add an offset by means of a DAC. The board also allows the installation of thermal compensation resistors



(a) (b)

Figure 3.8: Motor groups cross section. The figure shows CAD representations of the motor groups 1 and 2 cross sections ((a) and (b) respectively). Fixed parts are represented in blue, "fast" rotating components in yellow and "slow" ones in green.



(a) (b)

Figure 3.9: Six-axis force-torque sensor. The figure shows photos of the sensors' three-spoke structure (a), and the integrated electronic board (b).

that minimize the thermal drift effects of the semi-conductor strain gauges. All the operations are managed with a 16bit DSP from Microchip (dsPIC30F4013) which also provides digital signal filtering and the linear transformation needed to project the signals of the strain gauges to the force/torque space. The data are finally broadcast through a CAN bus interface at a frequency of $1kHz$.

3.5.3 Control boards

The arms' brushless motors are controlled with the BLL (BrushLess Logic) and the BLP (BrushLess Power) electronic boards shown in Fig.3.10. The BLL board processes the various signals provided by the sensors and generates the control signals that govern the motion of the motors. These signals are then passed to the BLP board which contains the actuator high-power drivers: the voltages applied to the three phases are controlled by the amplifiers with pulse width modulation (PWM). All the electronics are embedded on-board near the motor joint assemblies. Data to and form BLL boards are exchanged through a CAN bus interface.

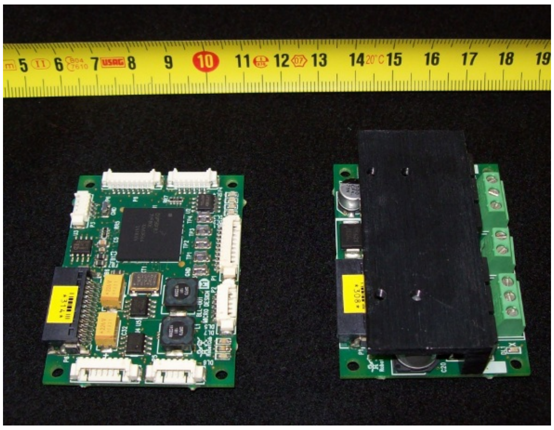


Figure 3.10: BLL and BLP electronic boards.

3.5.4 Communication bus

The BLL boards and the six-axis force-torque sensor communicate on a 1Mbit CAN bus; the bandwidth of the bus allows to pass, roughly, eight 8bit packages every millisecond. All the electronics described so far are connected as shown in Fig.3.11.

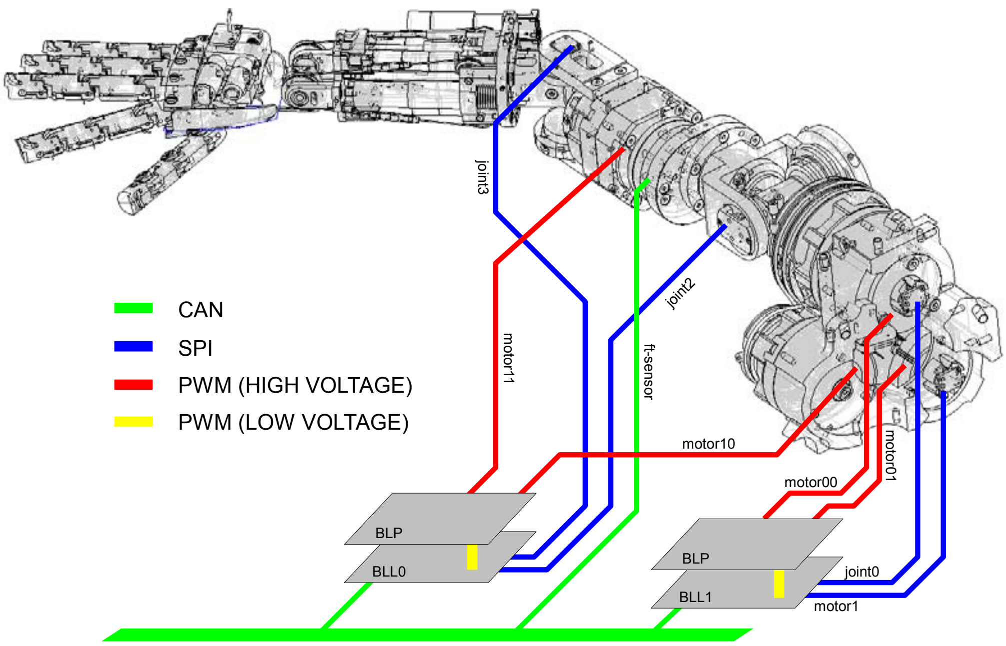


Figure 3.11: Arm connection diagram. The figure shows a diagram of the internal electric and electronic connections of the right arm.

4

Joint torque sensor design

Among all the possible solutions to improve the back-drivability of a robotic manipulator, described in chapter 2, joint torque control was chosen as the most suitable method for the iCub. As described in section 2.3 the implementation of joint torque control requires a measure of joint torques. This problem is not trivial because of the stringent weight and size limitations which characterize robotic manipulators. The problem of joint torque sensor design dates back to the introduction of joint torque control.

In [Hirzinger et al., 1993; Holmberg et al., 1992] the use of inductance based displacement transducers is introduced.

Among the first to consider the problem in a systematic way were Visher and Khatib. In [Vischer and Khatib, 1995] the authors describe the sensor design procedure they followed and present several sensor geometries and torque measurement techniques. They describe the trade-offs in sensor design and different possible sensor placements. The proposed design also employs the alternative measurement principle of magnetic inductance. This design allows to overcome several limitations of strain-gauge sensors. However since inductive transducers are not as compact as strain-gauges this solution has not been widely adopted.

In [Aghili et al., 1997, 2001] Aghili et al. describe the design of a joint torque sensor for the McGill/MIT direct drive joint. The authors propose several different designs and compare them on the basis of their capability of rejecting exogenous force-torque components. These works also nicely describe the trade-off between high mechanical stiffness and high torque sensitivity, normally encountered in torque sensor design. The ideal sensor geometry is, in this case, determined with the help of finite element analyses (FEA).

In [Hirzinger et al., 2000] a new strain-gauge based joint torque sensor is described for the second version of the DLR light weight robot series. The sensor is based on a four spoke structure; eight strain gauges are glued on the side of each spoke. The sensor geometry is optimized with finite element simulations.

Another torque measuring technique has been proposed for robotic joints with Harmonic-Drive speed reducers. Since these speed reducers contain a flexible element several authors proposed to exploit it as a torque transducer by measuring its deformation with strain gauges [Hashimoto, 1989; Hashimoto et al., 2002, 1993; Kazerooni, 1991, 1995; Taghirad and Belanger, 1999]. The advantage of this method is that no additional parts are required. Unfortunately measurements are corrupted by characteristic torque ripples of Harmonic Drives. This problem can be solved with estimator based techniques (e.g. Kalman filtering). An additional problem is constituted by the periodic deformation of the flexspline caused by the wave-generator. These disturbances can be compensated for with a proper arrangement of the sensing elements. Finally the electrical wiring needs to be conducted out of the gearbox for signal processing, but is generally attached to a rotating part. All these drawbacks make this method less appealing and hamper its diffusion.

4.1 Design specifications

The above mentioned works found in the literature were then matched to other project requirements. The design process therefore began with the following set of specifications:

1. high frequency joint level torque feedback: high frequency torque feedback loop is essential for smooth control; $1[kHz]$ loop rate was considered to be sufficient for the intended application

2. high resolution torque feedback: for every sensor 16bits over a dynamic range dictated by the joint peak torques (see Tab.3.3) were chosen as appropriate

3. sensor resistance to exogenous force and torque components, i.e. the sensor shall not respond to forces and torques other than the one which it is intended to measure.

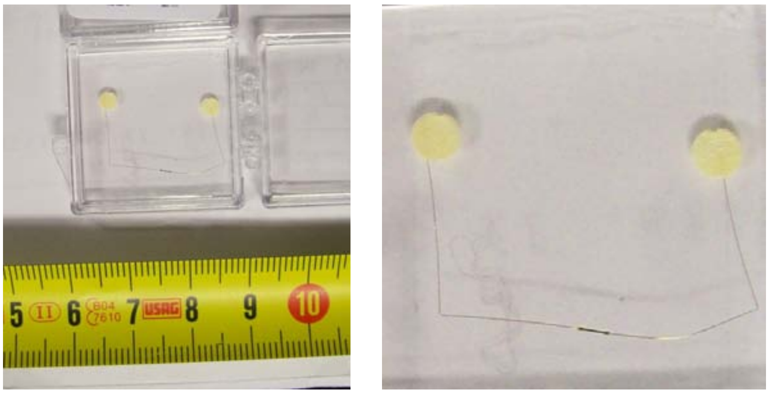
4. full electronic and mechanical retro-compatibility: the hardware upgrade had to be considered as a sort of "plug-in" and was thus required to seamlessly integrate on the current iCub robot version

The fourth requirement was particularly difficult to fulfill because it implied that the addition of joint torque sensors should not have interfered with any of the functional part dimensions: this restricted significantly possibilities at the design stage. An additional requirement for properly measuring the deformations of an elastic element is a local "evenness" of the strain field. This can, in general, be quantified as a 10 to 1 ratio between the absolute values of the first and second principal strain components.

4.2 Semi-conductor strain gauges

To increase the signal to noise ratio and to obtain high resolution it is desirable to design a structure which can generate the highest possible strain. However this generally results in increasing the internal stresses in the part. The sensor design problem is thus complicated by two conflicting requirements: mechanical robustness and torque sensitivity. To measure deformations metal-alloy strain gauges are widely employed: an alternative is constituted by semiconductor strain gauges (SSG). In SSG the change in resistivity depends on piezo-resistive effects of boron doped silicon. The semiconductor bonded strain gauge is a thin slice of silicon substrate with the resistance element diffused into a substrate of silicon (see Fig.4.1). The wafer element usually is not provided with a backing, and bonding it to the strained surface requires great care as only a thin layer of epoxy is used to attach it. Although more expensive, SSG have several advantages over standard metal strain gauges among which higher sensitivity (less deformation is needed to produce the same effect), higher fatigue life, higher output signal.

Since the current application required a very high sensitivity and large signal to noise ratio, SSG were preferred over standard metal strain gauges. A drawback of SSG



(a) (b)

Figure 4.1: Semiconductor strain gauges. The figure shows a photo of a SSG (a) and a detail of the SSG from the same photo magnified (b).

is their attachment process, which is very delicate and requires long curing and settling times.

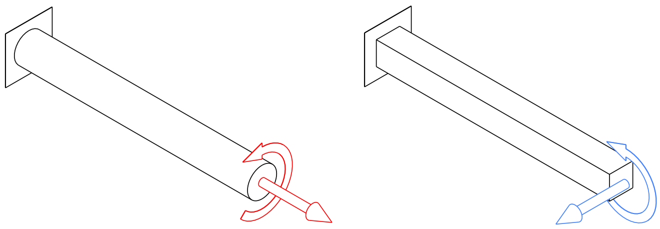
Moreover SSGs are very sensitive to temperature changes: the resistivity of these components drifts up to 10% for a 10°C temperature shift. The standard solution to cope with these temperature-caused resistivity drifts is to arrange four SSGs in a Wheatstone bridge configuration, or two in a half-bridge configuration: provided that the resistivity changes occurring in the different SSGs are similar, the bridge remains balanced.

SSGs maintain a linear strain-resistivity behavior up to $\pm 1000\mu\epsilon$ while the maximum strain they can tolerate is $\pm 5000\mu\epsilon$. Their choice implied therefore an additional design constraint regarding the strain levels in the region of the deformable part where they were to be glued.

4.3 Torque measurements in simple structures

Semi-conductor strain gauges only measure deformations in one linear direction. Measuring forces or torques generally requires using several SSGs appropriately attached to a structure whose deformation under load is known. However, calculating the deformation of a structure is not an easy task. Therefore, to develop torque sensors, one of the simplest and most robust solutions is to employ beam-like structures. Torques applied to slender beam structures can be subdivided into two main classed:

- torsional, i.e. when the moments are coaxial with the beam axis, as in Fig.4.2(a),
- flexional, i.e. when the moments are perpendicular to the beam axis, as in Fig.4.2(b).



(a) (b)

Figure 4.2: Torsion and bending moments in beams. The figure shows the sketches of a torsion moment (a) and a bending moment (b) applied to a slender beam.

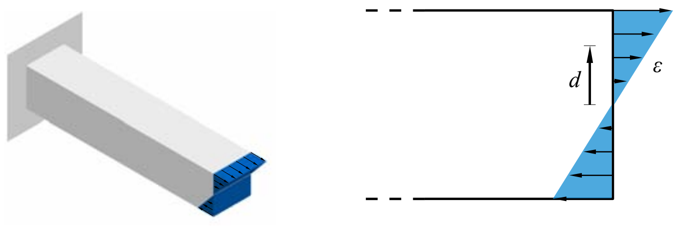
Predicting the deformation of a structure is not trivial as it requires the solution of a partial differential equation (PDE). This PDE allows to calculate the unknown function describing the deformations of the structure from the displacements constraints and applied loads boundary conditions. Closed form solutions have so far only been found for structures with simple geometries: the bending of slender beams and the torsion of beams with circular cross sections fall in this category. For the details on the derivation of the formulas cited in the next sections the reader shall refer to theory of elasticity classics such as [Timoshenko and Goodier, 1970].

4.3.1 Flexion

The Euler-Bernoulli beam theory allows to calculate the displacements of a loaded beam, under the assumption that the beam sections remain plane after loading. It furthermore allows to derive the classic formula for determining the flexional strains in the axial direction ε:

$$\varepsilon = \frac{M_b}{IE} d \tag{4.1}$$

where M_b is the bending moment, I is the moment of inertia of the section calculated with respect to the beam neutral plane, E is the Young's modulus, and d is the distance from the beam neutral plane. This equation is valid for beams of constant cross-sections, made of isotropic, homogeneous, linearly-elastic materials. Fig.4.3 shows the



(a) (b)

Figure 4.3: Beam section deformations under bending. The figure shows a three dimensional view (a) and a diagram (b) of a beam section deformation under a bending moment load.

deformation diagrams of a rectangular beam section. As can be seen, the highest deformations are in the top and bottom faces of the beam. If two strain gauges are attached on these opposite faces (Fig.4.4), the relative difference of the strains they measure provides a good measure of the bending torque. This layout has the advantage of being insensitive to loads applied in other directions (e.g. normal or shear forces and torsion torques).

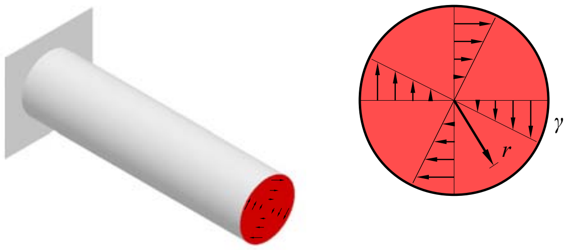
Figure 4.4: Optimal SSGs placement to measure flexion. The figure shows two three dimensional views of the optimal SSGs placement to measure a bending torque applied to a beam of rectangular cross section. SSGs are represented as black patches.

4.3.2 Torsion

The shear strains γ caused by the torsion torque in a circular cross section beam is given by:

$$\gamma = \frac{M_t}{I_p G} r \tag{4.2}$$

where M_t is the torsion moment, I_p is the polar moment of inertia of the section calculated with respect to the beam axis, G is the shear modulus, and r is the distance from the beam axis. In this case the highest shear strains build up on the beam's outer



(a) (b)

Figure 4.5: Beam section deformations under torsion. The figure shows a three dimensional view (a) and a diagram (b) of a beam section deformation under a torsion moment load.

cylindrical surface. However SSGs cannot be directly used to measure shear strains as they can only be employed to measure linear strains. Nevertheless shear strains can be resolved into their principal linear strain components via Mohr's circle analogy. In this case the principal strain components are oriented at a $\pi/2$ helical angle with respect to the beam axis. The principal components have however opposite sign: one is compressive and the other is tensile. Therefore the best solution to measure torsion torques in beams of circular cross sections, is to attach two SSGs at a $\pi/2$ helical angle with respect to the beam axis (see Fig.4.6) and measure the difference of their relative deformations.

Figure 4.6: Optimal SSGs placement to measure torsion. The figure shows a three dimensional view of the optimal SSGs placement to measure a torsion torque applied to a beam of circular cross section. SSGs are represented as black patches.

4.4 Conceptual design and design procedure

As show in [Aghili et al., 1997] there is a tradeoff between a sensor's sensitivity and its mechanical resistance. It would be desirable to design a sensor with the highest possible deformations to increase sensitivity. High deformations however cause high stresses which negatively affect the overall system robustness. As described at the beginning of chapter 3 the iCub motor groups are slightly over-dimensioned. This considerably complicates the design of sensors as they are required to resist the corresponding joint peak torque values and still be sensitive enough to measure lower torques with sufficient resolution. A rather low safety factor, close to one, was therefore adopted to ensure sufficient sensitivity.

The issue of integrating the torque sensing elements onto the arm joints was then considered. To solve this problem there are generally two alternatives: a first option is to redesign and sensorize one of the elements of the transmission chain whereas a second alternative is to insert in the transmission chain an additional controlled deformation transducer. This latter choice is generally easier to implement because it frees the designer of the dimensional constraints posed by the existing parts. Another important decision is whether to place the torque sensors at the joint level or at the motor level in the coupled shoulder joint (described in Sec.3.3). It was decided to place the sensors at the joint level for two reasons: firstly we considered that it is most important to know the torques exerted by the robot rather than to simplify its controller; secondly placing the sensors at the motor level requires dimensional changes incompatible with the third design specification. Moreover introducing the sensors at joint level allows to compensate transmission non-linearities (friction, elasticity) although makes the controller more complicated.

For each sensor the following iterative procedure was followed:

1. possible locations for the sensors in the current structure of the arm were initially identified

2. once a plausible sensor placement was identified, the initial sensor geometry and dimensions were determined with the equations of linear elasticity [Timoshenko and Goodier, 1970]

3. the tentative sensor design was firstly validated with structural finite element analyses (FEAs), performed with the Ansys commercial software package

4. analyses where then iterated several times to optimize critical geometric features

4.5 Sensors placement

4.5.1 Shoulder pitch joint (joint 0)

The complex mechanism of the shoulder joint made the integration of the sensor for the pitch axis a rather difficult design problem. It was in the end decided to position the sensor directly on the motor output shaft. Firstly the shaft cross-section was reduced locally, to obtain strain levels tailored to the SSGs operating range. This particular placement however required to route the sensor wires out of the mechanism to the signal conditioning electronics. This issue was solved by designing a new hollow motor shaft

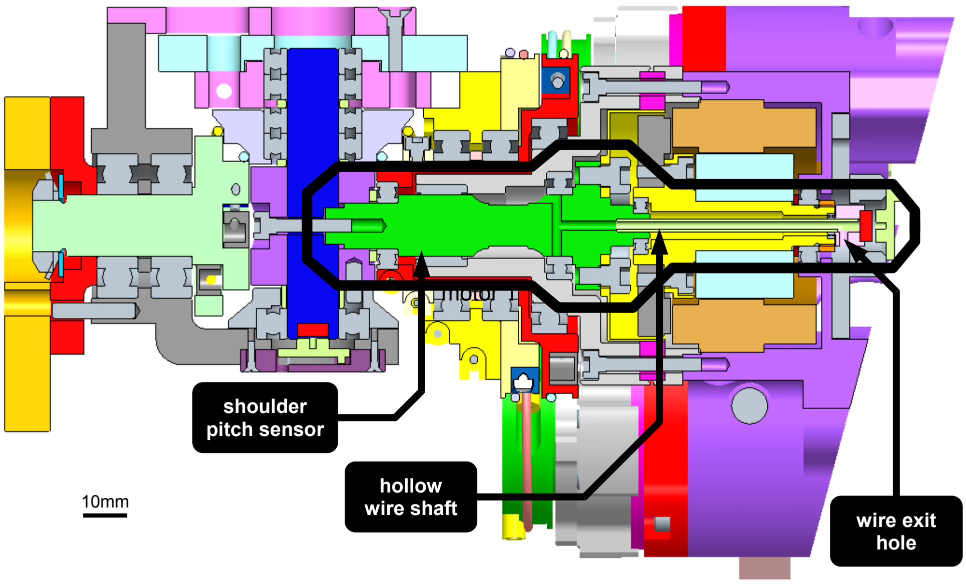
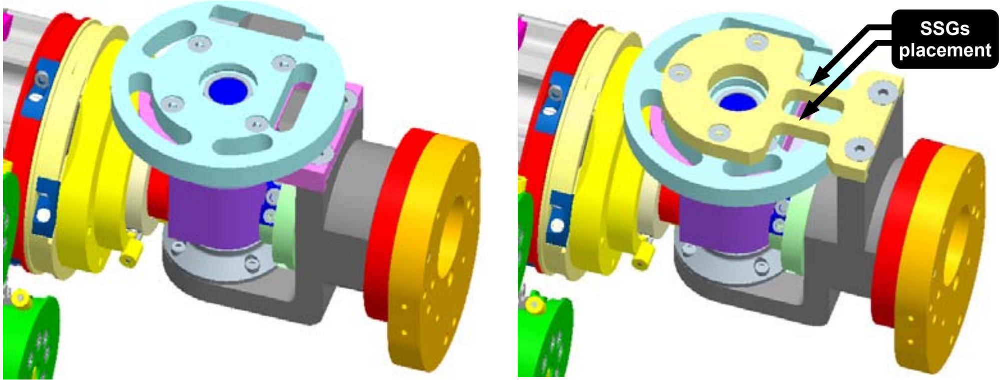


Figure 4.7: Shoulder pitch sensor. The figure shows a CAD cross section of the new shoulder joint: the sensor and the ire shaft are labeled.

that allowed to extract the cable from the rear of the motor (Fig.4.7). Because of the small spaces available and of the strong motor EMI, a special four-wire shielded micro-cable with 1.01mm outer diameter had to be employed. As the mechanical robustness of the sensor was critical 36CrNiMo4 stainless steel was chosen as construction material.

4.5.2 Shoulder roll joint (joint 1)

The sensor for the roll axis was inserted, as an additional part, in the transmission chain conveying the motion to the shoulder roll joint. The terminal part of the set of pulleys driven by the second motor was modified, and a new part, fixed externally, was added (Fig.4.8). The new part has two beam-like structures in its terminal part whose flexion is proportional to the transmitted torque. Ergal7075 was the material chosen for this part: its low Young's modulus implies that less force is required to induce measurable strains, thus increasing the sensitivity of the sensor. Its rather high mechanical resistance is another desirable characteristic.



(a) (b)

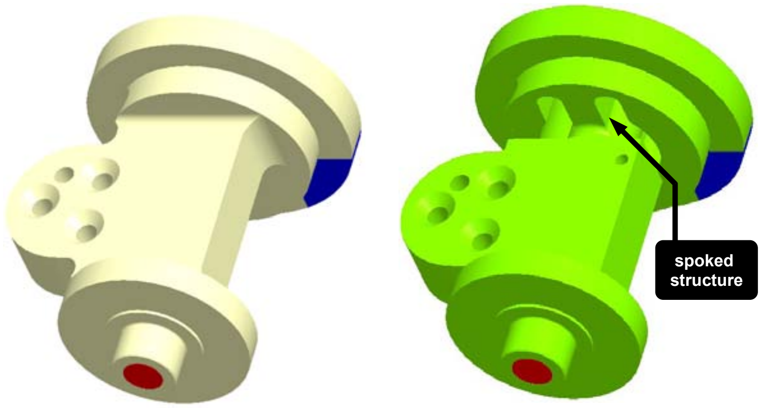
Figure 4.8: Shoulder roll joint. The figure allows the comparison of the previous joint design (a) and its upgraded version (b).

4.5.3 Shoulder yaw joint (joint 2)

No additional sensors were required for the shoulder yaw joint, since its axis coincides z axis of the six-axis force-torque sensor mounted after the shoulder joint.

4.5.4 Elbow joint (joint 3)

The elbow joint was modified by adding two grooves thus creating a spoke-like structure: measuring the flexion of this part allowed to measure the torque exerted by the elbow joint (see Fig.4.9). This modification required however some minor changes of the subassembly bearings and their supports. Because of these changes 36CrNiMo4 stainless steel was chosen for this sensor because of its high mechanical resistance.



(a) (b)

Figure 4.9: Elbow joint upgrade. The figure shows the changes undergone by the elbow joint: the old version is shown in (a) whereas the new part is shown in (b).

4.6 Finite element analyses

As mentioned in section 4.4 sensor designs were validated the help of finite element analyses (FEA). ProE CAD models were processed with the Ansys Educational FEA software. A linear elastic analysis type was considered as the most appropriate since the sensors are not affected by relevant non-linearities and still allowed fast convergence to the solution.

The meshes for the FEA were generated automatically with a patch independent tetrahedron algorithm with refinement conditions in the critical regions to better model the stress/strain gradients.

For what concerns the boundary conditions all the simulations were performed with face loading and restrained displacements The torque loads for each joint were taken equal to the corresponding peak values listed in Tab.3.3. The screws used to fix the parts were modeled with a zero displacement (x, y, and z directions) condition. In the shoulder pitch and elbow sensors bearings were modeled by constraining the radial displacements of the surfaces they were acting on. In the shoulder roll joint, to simulate simply the contact with the subjacent part, the vertical displacements of the base of the sensor were constrained to be zero.

The results of the simulations are shown in Fig.4.10, Fig.4.11 and Fig.4.12. The last iteration of the FEAs allowed to obtain in the region to be sensorized strain levels appropriate to the SSGs operating range ($\pm 500\mu\epsilon$ to $\pm 1000\mu\epsilon$).

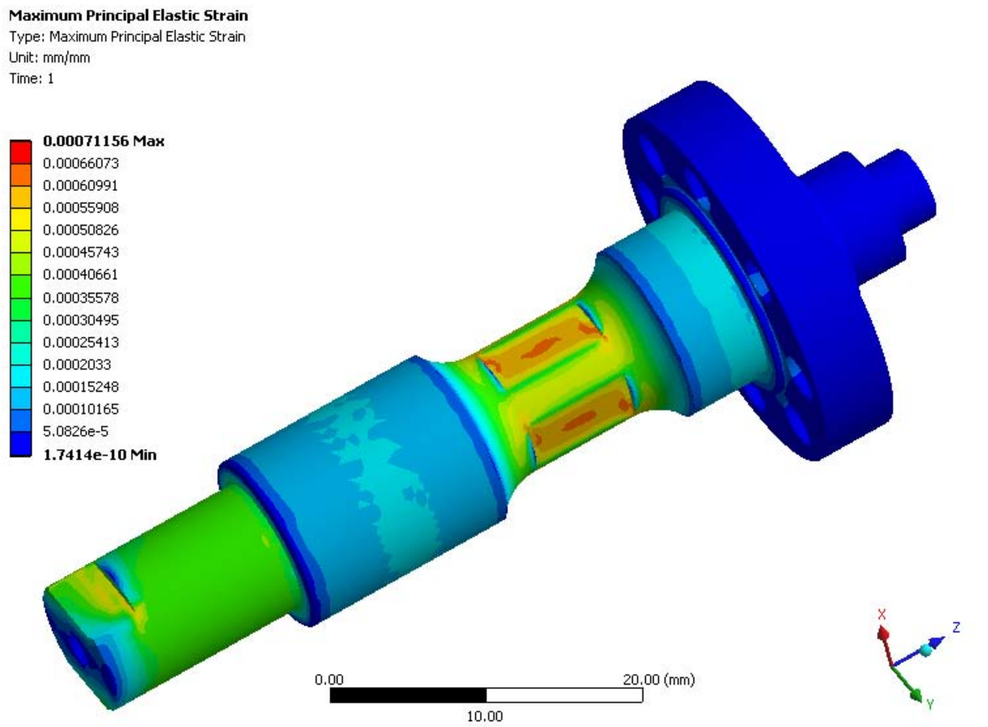


Figure 4.10: Shoulder pitch joint FEA. The figure represents the strain field for the shoulder pitch joint sensor.

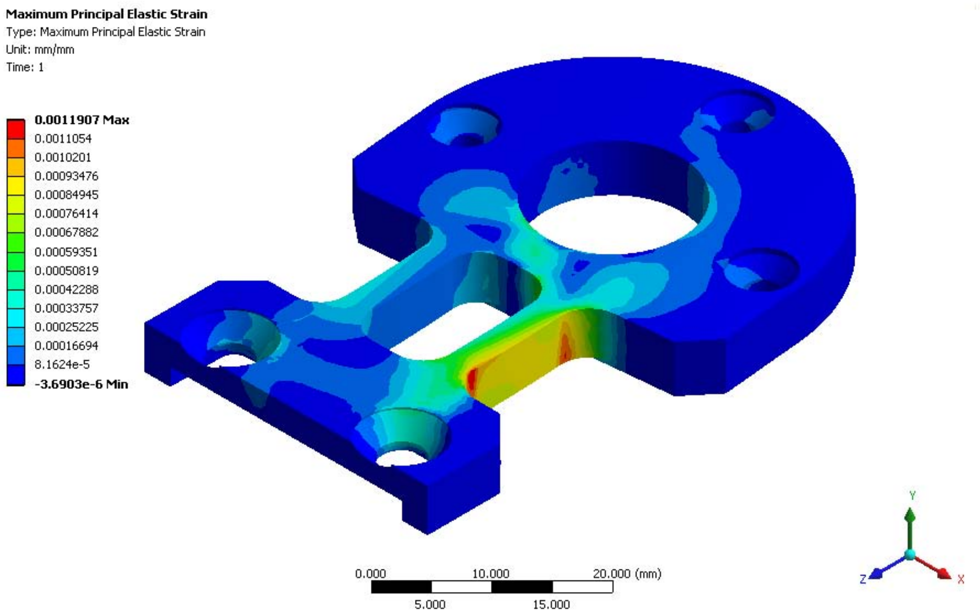


Figure 4.11: Shoulder roll joint FEA. The figure represents the strain field for the shoulder roll joint sensor.

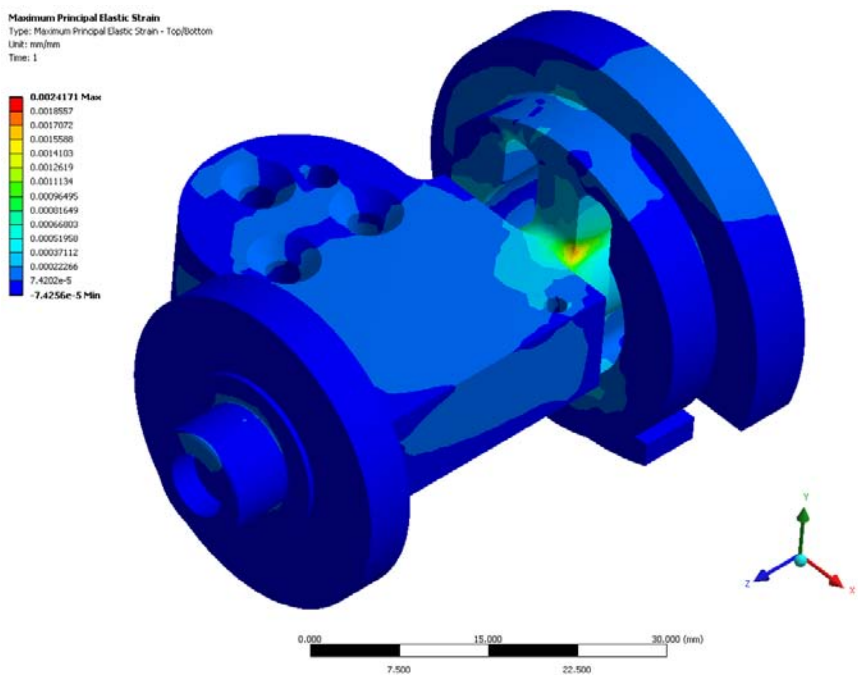
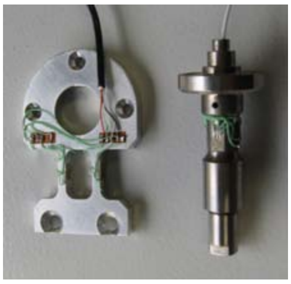


Figure 4.12: Elbow joint FEA. The figure represents the strain field for the elbow joint sensor.

5

Sensor validation

The sensors described in chapter 4 were constructed and instrumented with SSG (see Fig.5.1). It was the necessary to perform some preliminar tuning before they could be



(a)

(b)

Figure 5.1: Joint torque sensors. The figure shows a photo of the shoulder pitch and roll sensors (a) and a photo of the elbow sensor (b).

employed.

5.1 Temperature compensation

Among the first problems to consider was the temperature compensation mechanism of the SSG. SSG are very sensitive to temperature changes: the resistivity of these components drifts up to 10% for a $10°C$ temperature shift. The standard solution to cope with these temperature-caused resistivity drifts is to use up to four SSG in a Wheatstone bridge configuration (see Fig.5.2(a)). In this way, since temperature drifts occur in all the SSG, the bridge remains balanced and the offset shift is somehow corrected. However a more sophisticated strategy can be adopted: two resistors (of appropriate value) of constant value are added in series and in parallel to one of the SSG of the bridge (as in Fig.5.2(b)), in order to keep the resistivity difference of the two SSG of the bridge as low as possible.

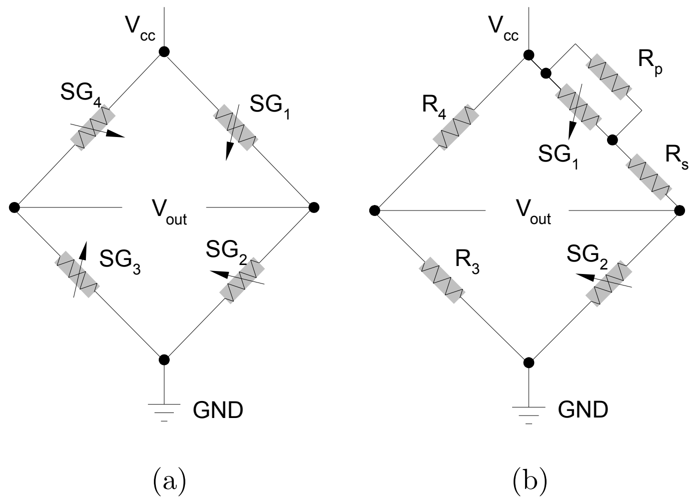


Figure 5.2: Wheatstone bridge temperature compensation circuits. Strain gages and normal precision resistor are indicated with the SG and R letters respectively. The standard 4-gages bridge is show in (a); the alternative stabilizing circuit which we adopted is show in (b).

At this point the value of the two resistors (R_p and R_s) had to be determined. To this end the resistivity changes of the SSG attached on the sensors were measured at varying temperatures. The sensors were placed in a Binder constant climate chamber. From an initial value of $20°C$ the temperature was first increased up to $50°C$ with three $10°C$ steps. The temperature was then reduced back to $20°C$ with the same temperature step size to check if the system exhibited hysteresis. Before measuring the resistivity of the SSG the component was left settle for one hour to allow it to reach temperature steady state. This suitable time value was found empirically by assuming the component behaved as a first order system. To check the behavior of the SSG the linear regressions curves for the data obtained were computed. The residuals of the fitted lines were rather small (i.e. $< 0.1°C$); it was therefore determined that the behavior of the SSG in the $20 \div 50°C$ range can be considered as linear.The appropriate values for the series and parallel resistors for each channel were finally calculated using a non-linear least-squares technique adopting the trust-region algorithm.

5.2 Force torque sensor calibration

The second problem to be addressed was the calibration of the six-axis force-torque sensor (see section.3.5.2). The calibration matrix is a 6×6 matrix that allows to convert the raw voltage data measured with the strain gauges into the real components of forces and torques. Being $\boldsymbol{f}$ the 6×1 forces-torques vector, $\boldsymbol{v}$ the 6×1 measured voltages vector and $\boldsymbol{C}$ the 6×6 calibration matrix, the following equation holds (see Fig.5.3):

$$\boldsymbol{f} = \boldsymbol{C}\boldsymbol{v} \tag{5.1}$$

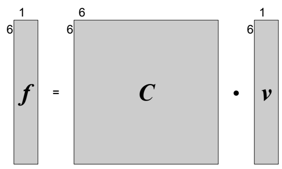


Figure 5.3: Calibration matrix linear transormation.

Given 6 linearly independent load conditions in the form of a 6×6 matrix $\boldsymbol{F}$ and the relative digital data $\boldsymbol{V}$ the calibration matrix $\boldsymbol{C}$ could be derived as:

$$\boldsymbol{C} = \boldsymbol{F}\boldsymbol{V}^{-1} \tag{5.2}$$

Theoretically these 6 conditions would suffice to calibrate the sensor; in practice many more data are used. The squared error between the measured voltages, multiplied by the calibration matrix, and the corresponding force and torque values, is minimized by varying the elements of the calibration matrix. This process yields as result the 36 coefficients of the calibration matrix. Voltage data are gathered (for more than six conditions) in the rectangular matrix $\boldsymbol{V}$ which is then inverted with the Moore-Penrose pseudo-inverse algorithm was used (the process is represented in Fig.5.4).

One of the easiest ways to generate known loads and to apply them to the sensor is to use a truss structure and hang weights to it. Given the weights and the geometry of the truss structure, the value of forces and torques applied to the sensor can be derived. A calibration was designed and constructed: the structure is represented in Fig.5.5. To properly elicit measurements in the six channels of the force-torque sensor a set 24 loading conditions (lited in table 5.1) was empirically chosen: this constitued the

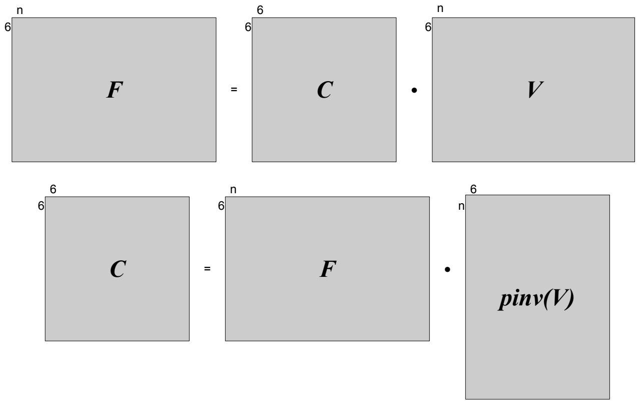


Figure 5.4: Calibration matrix calculation outline diagram.

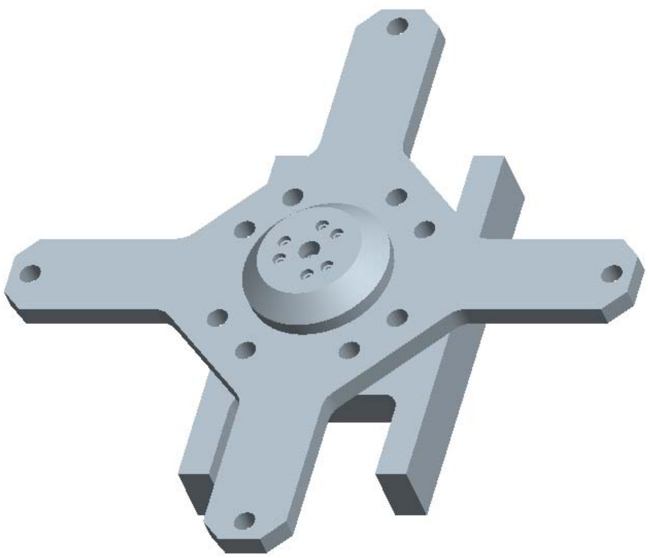


Figure 5.5: Calibration structure. The figure shows a photo of the structure used to calibrate the sensors.

known loads matrix $\boldsymbol{F}$. It is worth noting that the loading conditions do not need to comprise pure forces and torques as long as there are at least six linearly independent conditions (i.e. the rank of matrix $\boldsymbol{F}$ is 6).

These loading conditions are visualized in Fig.5.6; Fig.5.7 shows a detailed view of the residual quadratic errors after the calibration of the force-torque sensor.

5.3 Joint torque sensor calibration

The joint torque sensors were calibrated against the gravity torques generated by the arm. The gravitational torques at the joints were measured and compared with their expected value. The results of this test are shown in Fig.5.8. The curves were traced by measuring static joint torques at various joint postions, spanning the joint range at $5\,deg$ increments. Some mismatches can be appreciated, specially for the elbow joint. Various reasons for these behavoiurs can be conjectured such as unmodeled frictional stiction effects or non-linear elastic characteritics in the steel tendons and the Harmonic-Drive reducers: further analyses in this sense are required. Moreover the shoulder yaw joint yields slightly more noisy measures. The digital signals are in general affected by 2 to 3 bits of noise thus reducing the final resolution to 13 bits. Nevertheless sensor readings generally correlate nicely with the corresponding expected results.

component	F_x	F_y	F_z	τ_x	τ_y	τ_z
units	$[N]$	$[N]$	$[N]$	$[Nm]$	$[Nm]$	$[Nm]$
condition number						
1	0	0	51.01	-7.52	0	0
2	0	0	51.01	0	-7.52	0
3	0	0	51.01	7.52	0	0
4	0	0	51.01	0	7.52	0
5	-51.01	0	0	0	-0.26	-7.14
6	-51.01	0	0	0	-9.44	0
7	-51.01	0	0	0	-0.26	7.14
8	0	-51.01	0	0.26	0	-7.14
9	0	-51.01	0	9.44	0	0
10	0	-51.01	0	0.26	0	7.14
11	51.01	0	0	0	0.26	-7.14
12	51.01	0	0	0	9.44	0
13	51.01	0	0	0	0.26	7.14
14	0	51.01	0	-0.26	0	-7.14
15	0	51.01	0	-9.44	0	0
16	0	51.01	0	-0.26	0	7.14
17	0	0	-247.21	0	0	0
18	0	0	247.21	0	0	0
19	-247.21	0	0	0	-1.24	0
20	247.21	0	0	0	1.24	0
21	123.61	214.09	0	-1.07	0.62	0
22	-123.61	-214.09	0	1.07	-0.62	0
23	123.61	-214.09	0	1.07	0.62	0
24	-123.61	214.09	0	-1.07	-0.62	0

Table 5.1: Loading conditions

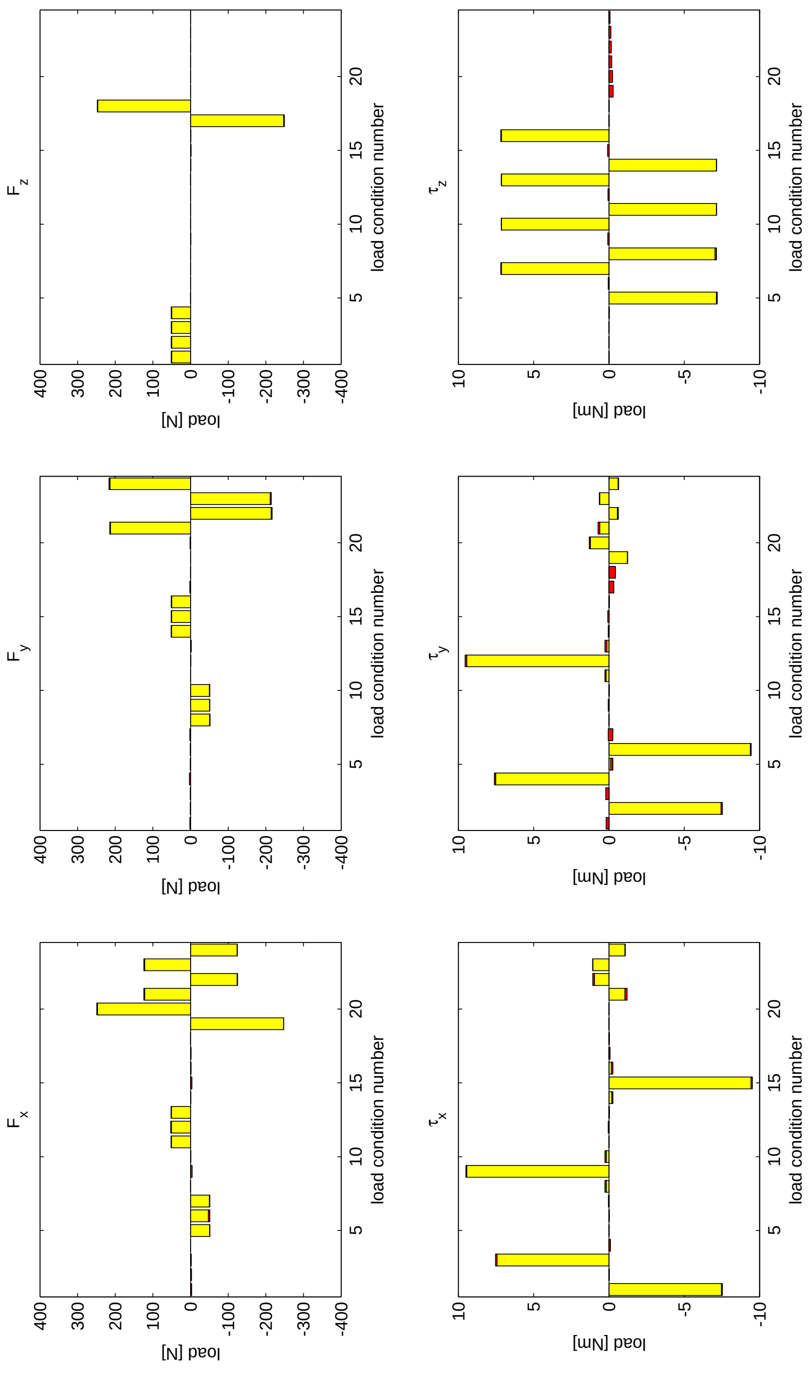


Figure 5.6: Calibration loading conditions. The figure shows the values of the forces as yellow bars. On each component, in every condtion, is stacked the corresponding residual error represented ar a red bar. A more detailed plot of the errors is shown in Fig.5.7.

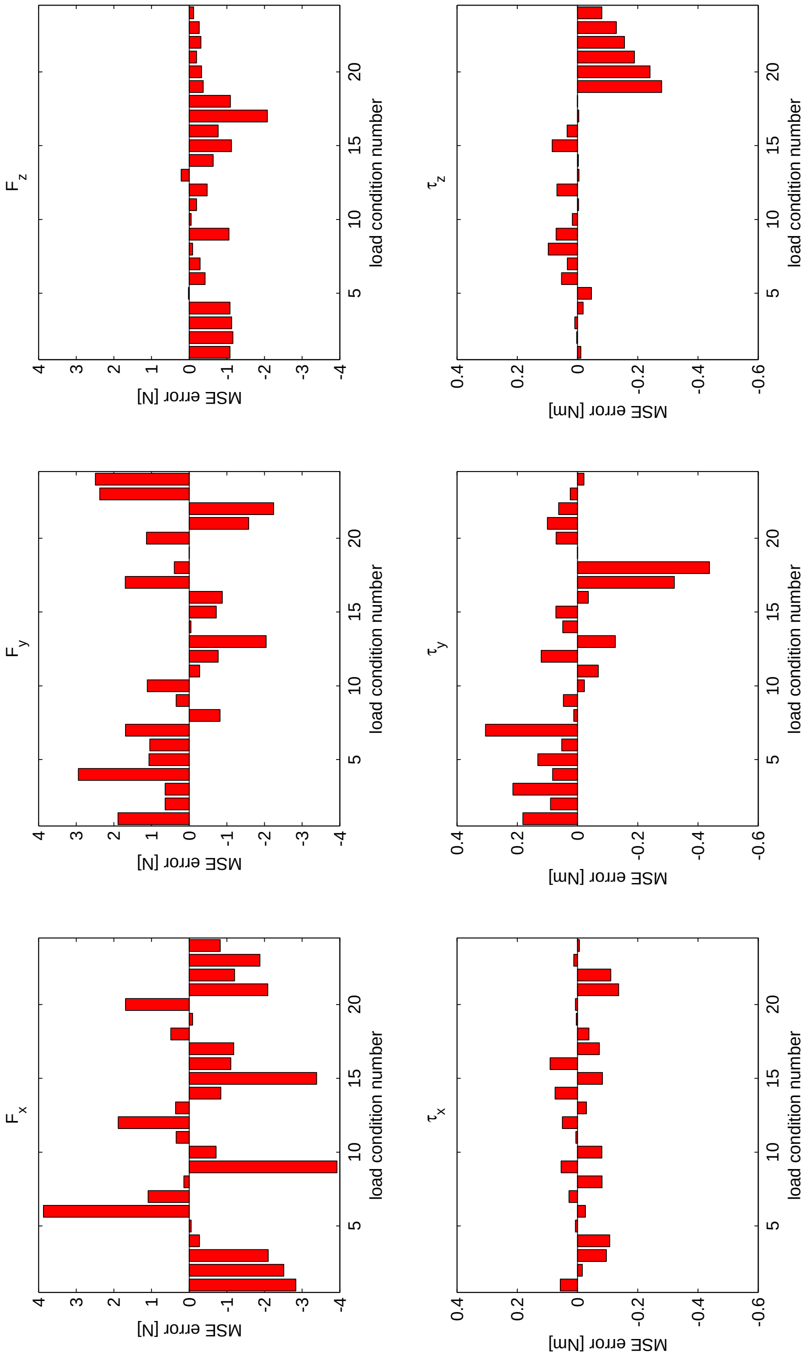
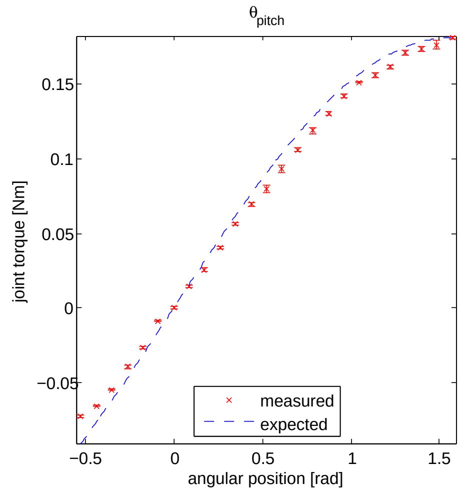
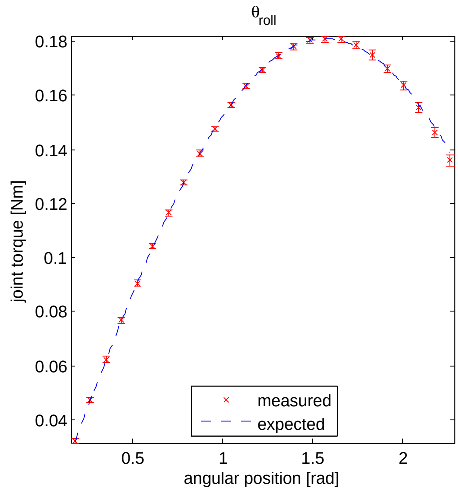


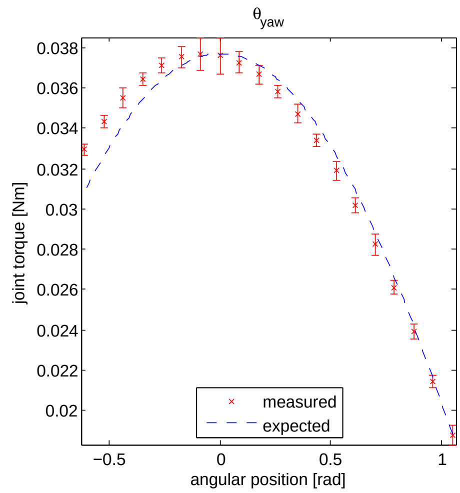
Figure 5.7: Residual quadratic errors for the loading conditions. The figure shows a plot of the residual quadratic errors for each loading condition, and for the six components of forces and torques.



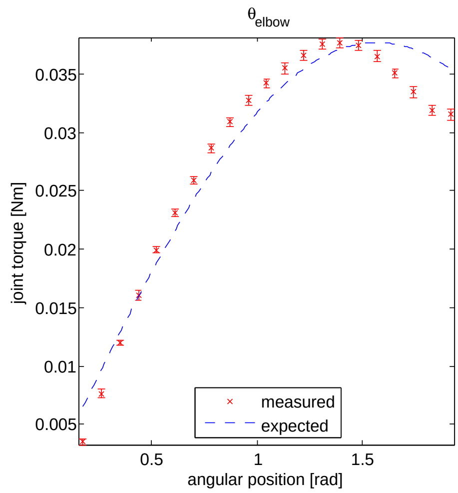
(a)



(b)



(c)



(d)

Figure 5.8: Sensor validation. The joint torque measurements on the shoulder pitch, shoulder roll and elbow joints are shown in (a), (b), and (c) respectively.

Dynamic tasks

6

Motor control, primates and robots

This short chapter intends to present a brief description of how the motor control problem is solved in robots and primates. This introduction will introduce several basic concepts, that will help to elucidate the relevant motivations to study dynamic tasks.

The primate motor system is capable of generating a remarkable variety of movements. If the sensorimotor system is analyzed from a "control systems" point of view [Schaal and Schweighofer, 2005] it can be subdivided into several units that cooperate to achieve a desired movement. The central nervous system (CNS) takes care of the movement planning, and generates a motor command as a set of muscle activation patterns. These, in turn, govern muscular contractions which produce the movements of the limbs. To improve the control of movements the feedback signals of different sensory modalities are collected and sent back to the CNS. Biological muscles however have a relatively limited actuation bandwidth. Moreover sensory feedback is not instantaneous and its transmission might require up to 200$[ms]$ (depending on the modality). Despite all these limitations humans perform with relative ease a wide range of motor skills, from extremely fine manipulation (e.g. sewing, writing, playing musical instruments) to fast, highly coordinated, powerful movements (e.g. weight-lifting, long jumping, juggling). The way in which these movements are planned and controlled is still an open issue.

6.1 Models for biological motor control

6.1.1 Internal models

It is by now widely accepted that the primate brain relies on the computational principle of internal models [Kawato, 1999]. These processing units, probably located in the cerebellum and the primary motor cortices [Wolpert et al., 1998], mimic the input/output characteristics of the motor apparatus. In [Wolpert and Ghahramani, 2000] the sensorimotor loop is subdivided into three main subprocesses which continuously run in parallel(Fig.6.1):

- the inverse model
- the forward dynamic model
- the forward sensory model

The inverse model given a task, a state and a context issues a motor command. The way by which this is done is not yet clear. A rather accredited theory is that the brain contains a map that allows to transform all the points that constitute a given trajectory into motor commands [Kawato et al., 1987]. The forward dynamic model given a state signal and an issued motor command predicts the next state of the system. The forward sensory model given the current state of the system and a motor command allows to predict the sensory feedback it will produce. As mentioned earlier the transduction and transport of sensory signals to the CNS involves considerable delays. Moreover these signals are generally corrupted by noise. In this sense the forward dynamic model and the forward sensory model form a sort of predictive observer capable of reducing the uncertainty in the state estimation caused by noisy and delayed feedback signals. This provides a reliable state signal which can be used to compute the motor command by the inverse model. Moreover the forward dynamic model can be used for mental simulations of intended movements thus providing a support for the planning phase.

6.1.2 Optimal control

An alternative perspective on the motor control problem has been proposed by Todorov in [Todorov, 2004]. In this review several examples show how optimality criteria allow to predict and model complex behaviours. This approach is appealing because the

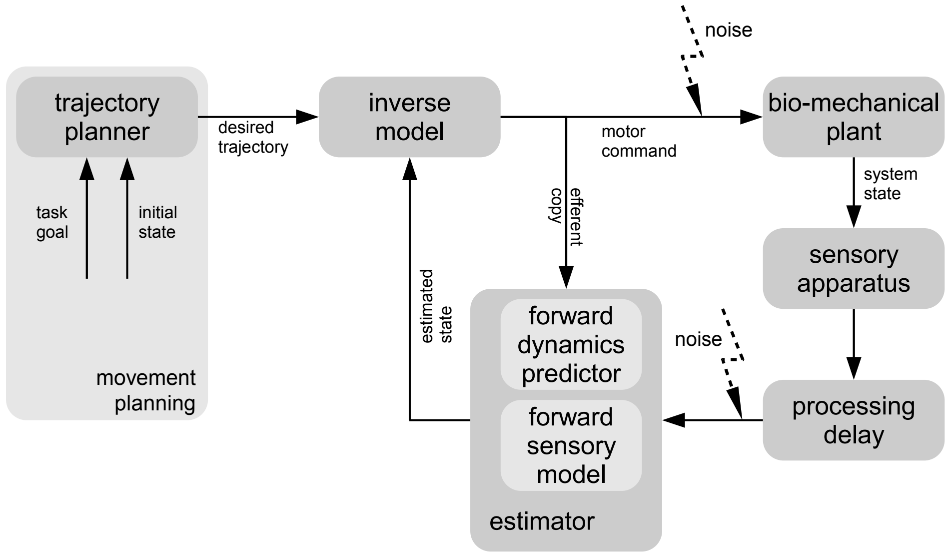


Figure 6.1: Sensorimotor loop diagram. The figure shows a diagram of the human sensorimotor loop (adapted from [Wolpert and Ghahramani, 2000; Wolpert et al., 1998]).

assumption that the sensorimotor system continuously aims at improving behavioural performance and overall efficiency is well justified. The author shows how traditional methods generally require as input a detailed description of how a desired goal is to be accomplished. On the other hand the optimal control framework allows to derive a control strategy simply by specifying a performance criterion. Optimal control methods can be subdivided into two main categories: open-loop and closed-loop, depending on whether the role of on-line sensory feedback is considered or not. In a practical implementation an open-loop optimization would yield a motor plan to be executed by a feed-back controller. In a closed-loop optimization process also the controller can be tuned, thus allowing the plant and the task to shape the controller that best solves the task (see Fig.6.2). Notably this elegant and coherent approach is well in accordance with the results of several fast reaching experiments presented in [Desmurget and Grafton, 2000]. Interestingly this framework is similar to the one presented in the previous section as in both cases an internal model, relying on predicted feed-back, is needed to predict state changes. The author however argues:

> “[...] what are usually called internal forward models as distinguished

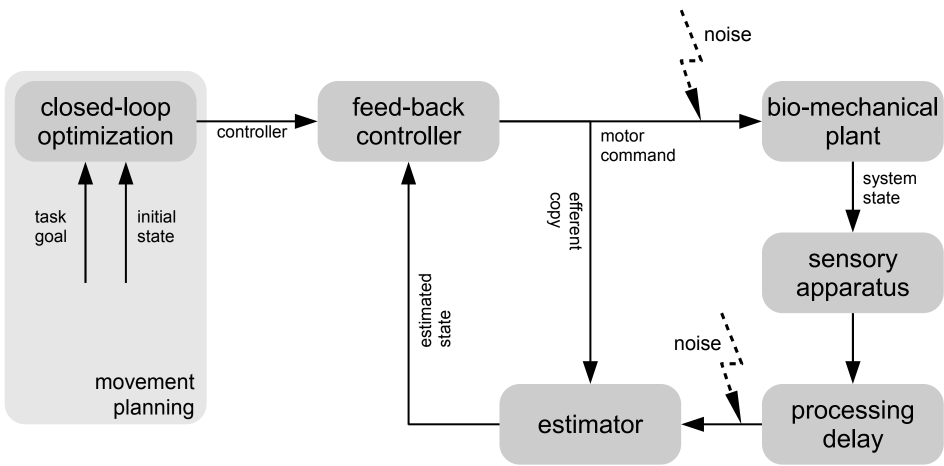


Figure 6.2: Sensorimotor loop diagram. The figure shows a diagram of the human sensorimotor loop (adapted from [Todorov, 2004]) in its closed-loop variant.

> from internal inverse models. The latter are thought to transform task goals into motor commands, but because this is the job of a controller, I believe the "inverse model" terminology should be avoided."

Indeed in the optimal control framework, the motor command is not computed by the pre-programmed inverse model, but by the controller generated by the closed loop optimization procedure. The main difference between the inverse model and closed-loop optimization frameworks is the form of the controller, which among different tasks, remains fixed in the former case and is allowed to change in the former. To solve this limitation of the inverse model approach Wolpert and Kawato proposed a framework based on multiple parallely running inverse models [Wolpert and Kawato, 1998].

A basic problem of optimality based methods is the form of the cost function. What is generally done in optimal control studies in primates is to choose a cost function a priori and to check its validity a posteriori by comparing the theoretical predictions of a model with experimental observations. Several criteria have been proposed, such as the minimum jerk, the minimum torque change, (see [Todorov, 2004] and references therein) but the large body of experimental data that have been gathered seem to suggest that the optimality criterion varies among different tasks. A recent study by Berret et al. shows how the minimum absolute work cost function allows to integrate in the planning both inertial and gravitational forces on the limb to minimize energy expenditure [Berret et al., 2008]. The model allows to describe several features of arm reaching performed in normal conditions and in the absence of gravity. Mathematical tools allowing to infer the cost function from behavioural data would be extremely desirable. Although several mathematical methods for inverse optimal control have so far been proposed (e.g. inverse reinforcement learning) there does not seem to be a well established procedure.

Finally despite the solid theoretical bases of this approach, the computational complexity of optimal control methods makes their application to real world problems extremely difficult.

6.2 Traditional robot control

The way in which robots are normally controlled differs significantly from the sensorimotor processes taking place in the primate brain. This is most probably caused be the inherent "hardware" differences between the human body and conventionally designed robots (as described in chapter 1). These differences can be summarized into three basic points:

1. robots' sensory feedback generally have only minimal delays, typically below few $[ms]$. This lag is about one to two orders of magnitude lower than the corresponding typical values for the human nervous system

2. robotic actuators generally have a rather large bandwidth, once again one to two orders of magnitude superior to that of biological muscles

3. the mechanical admittance of robotic actuators is considerably lower than that of biological muscles (as described in section 1.4)

Out of these points only the third is generally considered as a drawback. The first two points imply that, contrary to the human sensorimotor system, error correction schemes, such as PID controllers, are a very effective way to control robotic manipulators [Franklin, 1993]. This aspect often allows robotic control system designers to avoid the integration of a predictive component, which as described in the previous section, is essential for sensorimotor control in humans. A large number of well established methods to compute feed-forward compensation commands (such as the computed torque method) can be found in the robotic literature (see [Spong et al., 2006] and references therein). These methods generally rely on an precise analytical formulation of the robots' dynamics, which is however rather difficult to derive because of model errors, actuator dynamics and unmodeled nonlinearities. Moreover the overdamped dynamic characteristic caused by high speed reductions contributes to cancel forces arising dynamically. These factors altogether contribute to making high gain PID trajectory tracking the standard solution for the control of robots.

Biological systems have been forced to evolve open-loop predictive controllers, because of large latencies and slow dynamics. On the other hand, robotic systems are fast and responsive enough to render this problem less critical. However feedback control inherently constrain the performance of the system and this will probably turn out to be a limiting factor in the near future. As the research in the field of physical robot interaction advances, robotic platforms tend to increasingly back-drivable and compliant designs. In such platforms the role of intrinsic dynamic forces (i.e. forces not actively generated by the robot) will become progressively more important, eventually to the extent it will not be possible to neglect them any more. For this reason the study of dynamic tasks and of suitable controllers to perform them, eventually inspired to the way primates control their movements, is considered as a interesting and crucial research topic which is thus worth pursuing.

7

Planning dynamic tasks

Let us define dynamic tasks as tasks to be performed in short lapses of time and in which the effect of forces not actively generated by the robot (such as inertial or gravitational forces), considerably affect the achievement of the task.

In this context the two link pendulum swing-up has become in the last fifteen years a standard benchmark problem. In this task a two link pendulum, as the one represented in Fig.7.1(a), with limited torque at the joints, has to move from its stable downward pointing equilibrium position to its upright position shown in Fig.7.1(b).

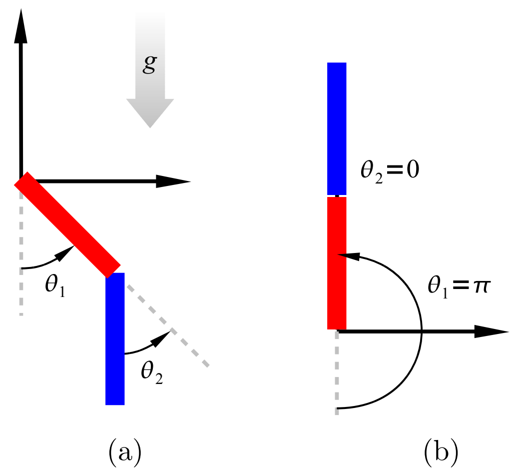


Figure 7.1: Two link pendulum swing up problem. The figure represents the benchmark problem of the two link pendulum swing up problem. The angle conventions are defined in (a) while the goal state is represented in (b).

The equation that generically describes the rigid body dynamics of a manipulator is:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + \boldsymbol{B}(\dot{\boldsymbol{\theta}}) + \boldsymbol{G}(\boldsymbol{\theta}) \tag{7.1}$$

where $\boldsymbol{\tau}$ denotes the vector of joint torques, $\ddot{\boldsymbol{\theta}}$, $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ the vectors of joint angular accelerations, velocities and positions, $\boldsymbol{M}(\boldsymbol{\theta})$ the configuration dependent inertia matrix, $\boldsymbol{C}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta})$ the Coriolis term, $\boldsymbol{B}(\dot{\boldsymbol{\theta}})$ the friction term and $\boldsymbol{G}(\boldsymbol{\theta})$ the gravity torques. For the system represented in Fig.7.1, Eq.7.1 can be explicited as the following system of equations:

$$\begin{cases} \tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 - C\dot{\theta}_2^2 - 2C\dot{\theta}_1\dot{\theta}_2 + G_1 + B_{11}\dot{\theta}_1 + B_{12}\dot{\theta}_2 \\ \tau_2 = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 - C\dot{\theta}_1^2 + G_2 + B_{21}\dot{\theta}_1 + B_{2,2}\dot{\theta}_2 \end{cases} \tag{7.2}$$

where M_{ij} denotes the ith row and jth column element of the inertia matrix:

$$\begin{aligned} M_{11} =& m_1 s_1^2 + J1 + m_2 s_2^2 + J_2 + m_2(l_1^2 + 2l_1 s_2 \cos\theta_2) \\ M_{12} =& M_{21} = m_2 s_2^2 + J_2 + m_2 l_1 s_2 \cos\theta_2 \\ M_{22} =& m_2 s_2^2 + J_2 \end{aligned} \tag{7.3}$$

and the other terms are:

$$\begin{aligned} C =& m_2 l_1 s_2 + \sin\theta_2 \\ G_1 =& g(m_1 s_1 \sin\theta_1 + m_2(s_2 \sin(\theta_1 + \theta_2) + l_2 \cos\theta_1)) \\ G_2 =& g m_2 s_2 \sin(\theta_1 + \theta_2) \end{aligned} \tag{7.4}$$

where m_1 and m_2 denote the links masses, l_1 and l_2 their lengths, s_1 and s_2 the distances of the centres of masses of the two links from the preceding joint, J_1 and J_2 the inertias of the two links and g the gravity acceleration. From Eq.7.2 angular accelerations can be computed as:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{M}(\boldsymbol{\theta})^{-1}(\boldsymbol{\tau} - \boldsymbol{C}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) - \boldsymbol{B}(\dot{\boldsymbol{\theta}}) - \boldsymbol{G}(\boldsymbol{\theta})) \tag{7.5}$$

Eq.7.5 is the forward dynamic equation of the system that, if integrated, allows to determine the future states of the system, given a current state $\boldsymbol{x}$ and control action $\boldsymbol{u}$. Let the state be $\boldsymbol{x} = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^t$ and the control inputs $\boldsymbol{u} = [\tau_1, \tau_2]^t$ we can write Eq.7.5 as:

$$\ddot{\boldsymbol{\theta}} = h(\boldsymbol{x}, \boldsymbol{u}). \tag{7.6}$$

Being joint torques limited the system cannot reach directly any point of the state space. It is however possible to transition to the upright position by exploiting system dynamics, that is by gathering momentum by swinging back and forth. Once enough energy has been acquired, the final swing bringing the system in the upright position can be performed.

Among the first investigators to consider the problem was Spong, which proposed a swing-up strategy based on partial feedback linearization and a linear quadratic regulator (LQR) [Spong, 1994].

Other pioneering studies on exploiting dynamics for the manipulation of objects were conducted Lynch and Mason from 1996. In [Lynch and Mason, 1996] the authors increased the manipulation capabilities of a 1 DoF robot with open loop control, by exploiting the intrinsic dynamics of the manipulated object. For motor planning a sequential quadratic programming (SQP) gradient descent non-linear optimization method was used.

In the context of exploiting dynamics Williamson used oscillators to control the humanoid robot COG (see section 2.4) to successfully perform a number of dynamic tasks such as juggling, turning cranks, playing with a Slinky toy, sawing wood, throwing balls, hammering nails and drumming [Williamson, 1999].

It is worth to note that problems such as dynamics exploitation and feedforward control of fast movements have been thoroughly considered in the robotics literature on robotic juggling (see [Aboaf et al., 1989; Buehler and Koditschek, 1990; Riley and Atkeson, 2002; Rizzi and Koditschek, 1992; Schaal and Atkeson, 1993] to cite a few).

Grzeszczuk and Terzopulos proposed an algorithm which was based on the acquisition of a model of the plant to be controlled [Grzeszczuk and Terzopoulos, 1998]. The model was learnt with a feedforward neural network trained with backpropagation. The learnt model was then used to plan the task by optimizing a cost function by gradient descent. This approach has several similarities in common with the distal learning method introduced in [Jordan and Rumelhart, 1992].

Another interesting approach is the one introduced by Rosenstein in [Rosenstein, 2003] and further developed in [Rosenstein et al., 2006]. In these works it has been shown how a 3DoF weightlifting robot can optimize motor synergies, to exploit dynamics to perform a task with better performances or under severe mechanical constraints. Their control architecture is based on equilibrium point motor control whose parameters are gradually tuned with a simple random search (SRS) reinforcement learning algorithm.

A problem similar to the two link pendulum swing up was also used as an example by Todorov and Li in [Todorov and Li, 2005], where the authors introduce an approximate iterative method (iLQG) to solve efficiently non-linear optimal control problems.

One of the major limitations of the iLQG method is that it relies on an accurate analytic description of the system dynamics which is in practice hard to derive. In [Mitrovic et al., 2010] the authors propose an alternative solution based on learning the forward dynamics of the system.

More recently Kober and Peters have developed an algorithm applicable to complex motor learning tasks [Kober and Peters, 2008] such as the ball-in-a-cup game. In this work they put a strong focus on episodic case reinforcement learning by using a parametrized policy which is then optimized with the expectation-maximization algorithm. In [Kober and Peters, 2009] the same authors show how a good initialization of the algorithm provided by demonstration allows to speed up considerably the convergence of the algorithm.

As the aim of this study was to develop a controller to demonstrate dynamic capabilities on a real robot all the methods and analyses focused on discrete time formulations. The state transition equation Eq.7.7 can be discretized by Euler integration with time-step Δt. In this case for any time t, given the state transition equation h, the current state $\boldsymbol{x}_t$ and control action $\boldsymbol{u}_t$:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + h(\boldsymbol{x}_t, \boldsymbol{u}_t)\Delta t = f(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{7.7}$$

This approach variant nicely adapts to the discrete way of handling data of the digital signal processors (DSP) used to control robots.

7.1 Finite time decision problem

In general all the methods cited so far propose an approximate solution of the finite time decision problem. This problem is solved by minimizing the so called value function V which represents how good it is to perform a given action in a given state. The strategy followed by the agent when trying to minimize the cost is called "policy", and will be denoted with π. For a deterministic system, the policy defines the action that will be taken at time t in state x_t: $u = \pi(t, x_t)$. The finite time decision problem takes the following general form:

$$V(x_t) = \min_{\pi} \left(\phi(x_T) + \sum_{t=s}^{T-1} C(x_t, \pi(t, x_t)) \right) \tag{7.8}$$

$$= \min_{u_{t \to T-1}} \left(\phi(x_T) + \sum_{t=s}^{T-1} C(x_t, u_t) \right) \tag{7.9}$$

where x_t denotes the state at time t, u_t the control action at time t, $\phi(x_T)$ the at cost at the terminal time T, and $C(x_t, u_t)$ the cost for taking action u_t from state x_t. The system dynamics are described by the forward dynamics state transition function f:

$$x_{t+\Delta t} = f(x_t, u_t) \tag{7.10}$$

Among all the possible policies there is always one that has the the property of being better or equal to all other policies. This policy is called the "optimal policy" and is denoted with π^* The minimization of Eq.7.9 yields a sequence of actions u^* that constitute the optimal policy π^*. All the optimal policies share the same optimal value function V^*:

$$V^*(x_t) = \min_{\pi} V_\pi(x_t) \tag{7.11}$$

A property of the optimal policy is that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. This allowed Bellman to formulate a recursive solution the finite time decision problem. Function $V^*(x_t)$ can indeed be computed recursively by backstepping from the final time:

$$V^*(x_T) = \phi(x_T) \tag{7.12}$$

$$\begin{aligned}
V^*(x_t) &= \min_{u_{t \to T-1}} \left(\phi(x_T) + \sum_{s=t}^{T-1} C(x_s, u_s) \right) \\
&= \min_{u_t} \left(C(x_t, u_t) + \min_{u_{t+1 \to T-1}} \left(\phi(x_T) + \sum_{s=t+1}^{T-1} C(x_s, u_s) \right) \right) \\
&= \min_{u_t} \left(C(x_t, u_t) + V(x_{t+\Delta t}) \right) \\
&= \min_{u_t} \left(C(x_t, u_t) + V(x_t + f(x_t, u_t)) \right)
\end{aligned} \tag{7.13}$$

The dimension of this problem scales linearly in time and exponentially in the number of dimensions (because of the well known "curse of dimensionality" problem). This implies that solving Eq.7.13 becomes numerically intractable for systems with more than three or four states.

The various approaches cited in the previous section propose algorithms to overcome this problem. The reinforcement learning equilibrium point approach [Rosenstein et al., 2006], and the iLQG framework [Todorov and Li, 2005] seemed particularly promising and were therefore taken as test cases. Finally a preliminary implementation of LD-iLQG was developed.

7.2 Reinforcement learning

Reinforcement learning (RL) is a computational approach to learning in which an agent tries to minimize the total amount of costs (or to maximize the total amount of rewards), it receives while interacting with a given environment [Sutton and Barto, 1998]. As the rewards and costs are not given immediately, and the environment (i.e. function f) is generally unknown the solution of this learning problem is not straight-forward. Nonetheless this framework has the advantage of being extremely general and can thus be applied to a wide range of problems. Broadly RL algorithms work as follows. Being V_π the value function for policy π is defined as:

$$V_\pi(x_t) = E_\pi(C(x)) \tag{7.14}$$

where E_π denotes the expected value for an agent following policy π. The environment is initially unknown, therefore as the agent explores and interacts with it, the value function is constructed progressively. The objective of RL is to compute the optimal value function V^* that allows to compute the optimal policy π^*:

$$V^*(x_t) = \min_\pi V_\pi(x_t) \tag{7.15}$$

To do this practically all existing RL methods adopt the general policy iteration (GPI) approach. This method consist in iteratively computing the value function for a policy, then improving the policy. As the process is iterated the RL algorithm eventually converges to the optimal value function V^* and the optimal policy π^*. It is important to note that to perform these operations it is not necessary to have a model of the environment as the knowledge about it is gathered with experience.

Let us now consider the method reported in [Rosenstein et al., 2006] for the swing up task. In this work the authors define the policy as a sequence of i proportional derivative (PD) controllers in the form

$$\boldsymbol{\tau}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{W}_i(K_p(\boldsymbol{\theta}_i^* - \boldsymbol{\theta}) - K_d)\dot{\boldsymbol{\theta}} \tag{7.16}$$

where $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ are the joint positions and velocities, $\boldsymbol{\tau}$ is the vector of joint torques, $\boldsymbol{W}_i$ is the i^{th} gain matrix, $\boldsymbol{\theta}_i^*$ represents the i^{th} equilibrium point, and K_p and K_d the proportional and derivative gains respectively. The policy switches from a PD controller to the following one at the instants t_i. The final PD controller is set in the goal position to heuristically help the convergence of the algorithm. This is not very strong problem simplification as the knowledge of the final desired state might be assumed a priori. The policy search iteration adapts the $\boldsymbol{\theta}_i^*$, $\boldsymbol{W}_i$ and the vector of switching instants $\boldsymbol{t}$ parameters with the simple random search (SRS) algorithm. The SRS algorithm is a derivative-free optimization algorithm similar to simulated annealing. In this method however a "memory" of the current minimum is maintained: if while exploring, the solution ends up in local minima it is attracted back to the current best minimum.

7.3 iLQG

As mentioned in section 7.1 an alternative to global optimization is constituted by local methods, which efficiently improve in an iterative way a suboptimal solution. The iLQG method falls in this category [Todorov and Li, 2005]. The approach is based on a iterative linearization of the non-linear plant dynamics around the current trajectory. For quadratic cost functions this procedure allows to derive locally valid Riccati-like equation that are then used to improve the trajectory by quadratic optimization. Non-linear constraints can be introduced by modifying the linear feedback gain matrix. Optimizing a quadratic function is only possible when the Hessian is positive definite. Non-linear dynamics, and control constraints can cause the Hessian to have negative eigenvalues. To avoid problems the Hessian is "fixed": this procedure yields a new quadratic approximation of the cost function which is successively accounted for when minimizing the cost-to-go function. For mathematical details the reader should refer to [Li, 2006; Todorov and Li, 2005] where a more comprehensive description of this method is presented. It is rather important to remember that this method relies on the knowledge of the explicit model of the system to derive the approximate discrete equations.

The two-link pendulum swing-up problem can be solved with the iLQG method by discretizing Eq.(7.5) as in Eq.(7.7). Let the goal state be $\boldsymbol{\theta}^* = [\pi, 0]^t$, the quadratic cost-to-go function can be written as:

$$V = w_p \sum_{i=1}^{2} (\theta_i(T) - \theta_i^*)^2 + w_v \sum_{i=1}^{2} \dot{\theta}_i(T)^2 w_c \sum_{t=0}^{T} u_t^2 \tag{7.17}$$

where w_p, w_p and w_r denote the weighting factors for the distance to the target at final time, velocity at final time and accumulated squared torques terms respectively.

Although the iLQG algorithm was developed for stochastic systems the study presented in this thesis focused on its deterministic version, which has the advantage of having an simpler mathematical formulation.

7.4 iLQG-LD

Among the most important limitations of the iLQG algorithm is that it relies on the analytic formulation of the system dynamics. In robotics deriving or etimating accurate analytic dynamic models is a rather complex problem. In the iLQG-LD framework [Mitrovic et al., 2010] this it is avoided with the help of non-linear function regression. This method is based on learning the function $f'(\boldsymbol{x}_i, \boldsymbol{u}_i)$ which best approximates the forward dynamics of the system $f(\boldsymbol{x}_i, \boldsymbol{u}_i)$. The approximation of function $f(\boldsymbol{x}_i, \boldsymbol{u}_i)$ inevitably introduces errors ε:

$$f(\boldsymbol{x}_i, \boldsymbol{u}_i) = f'(\boldsymbol{x}_i, \boldsymbol{u}_i) + \varepsilon \tag{7.18}$$

which will in turn be included in the planning of the movement. A consequence of this is that, in general, the control command sequence $\boldsymbol{u}'$ planned with $f'(\boldsymbol{x}_i, \boldsymbol{u}_i)$ will differ from the optimal one, namely $\boldsymbol{u}$ planned with $f(\boldsymbol{x}_i, \boldsymbol{u}_i)$. When the action sequence $\boldsymbol{u}'$ is executed the "real" system dynamics will cause the system to drift from the predicted trajectory $\boldsymbol{x}'$. This deviation eventually prevents the system from reaching the goal state. To cope with this problem a feedback control action can be added to the feedforward control commands. The appropriate feedback signal can be calculated

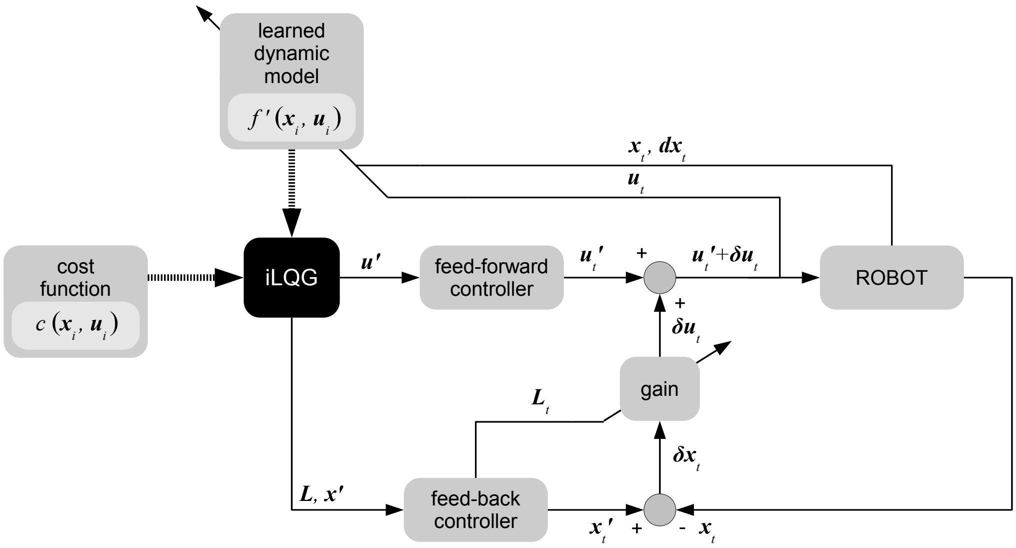


Figure 7.2: iLQG-LD learning and control scheme. The figure shows a block diagram of the iLQG-LD learning and control scheme.

given the sequence optimal feedback gains $\boldsymbol{L}$ and the predicted trajectory $\boldsymbol{x}'$ calculated by the iLQG algorithm and the current state of the system $\boldsymbol{x}_t$ as:

$$\delta \boldsymbol{u}_t = \boldsymbol{L}_t'(\boldsymbol{x}_t' - \boldsymbol{x}_t) \tag{7.19}$$

The iLQG-LD learning and control scheme is represented in Fig.7.2.

In [Mitrovic et al., 2010] the authors used locally weighted projected regression (LWPR) [Vijayakumar et al., 2005] to approximate the forward dynamics of the robotic manipulator.

Notably this approach closely resembles the method previously proposed in [Grzeszczuk and Terzopoulos, 1998], which however did not comprise the feedback controller.

8

Learning experiments

8.1 Reinforcement learning

The policy was composed as the succession of three PD controllers Similarly to [Rosenstein et al., 2006] the SRS algorithm was allowed to tune the gain matrices $\boldsymbol{W}_i$ the equilibrium points $\boldsymbol{\theta}_i^*$ and the vector defining in which moment to activate the different force fields $\boldsymbol{t}$. The initial value of the parameters were initialized at random. The joint torques were limited to the [-5,5][Nm] range.

An extensive search to optimize the parameters of the SRS algorithm was conducted. Every set of parameters was tested in a so-called parameter run. Each run was constituted by 5000 trials. Each trial consisted in a four second time lapse in which the task had to be accomplished. Each simulation was run with a constant 1[ms] time step.

A complete experiment employed, roughly, 50000 seconds, that is around 16 hours, on a normal desktop PC (Intel Core 2, 1.8[GHz]). A low percentage of the trials (≈10%) did not converge to a good solution. An example of a good trial is shown in Fig.8.1 and in Fig.8.2.

Among the good aspects of this approach was that it did not depend strongly on the set of parameters . One of the major drawbacks of this method was its initialization. Several tests confirmed that the quality of the solution found with the SRS algorithm and the rate of convergence to the optimal solution heavily depended on the random choice of the initial policy. This, in turn, did not allow to discern if the learning procedure benefited mostly by the SRS optimization or by the random policy initialization.

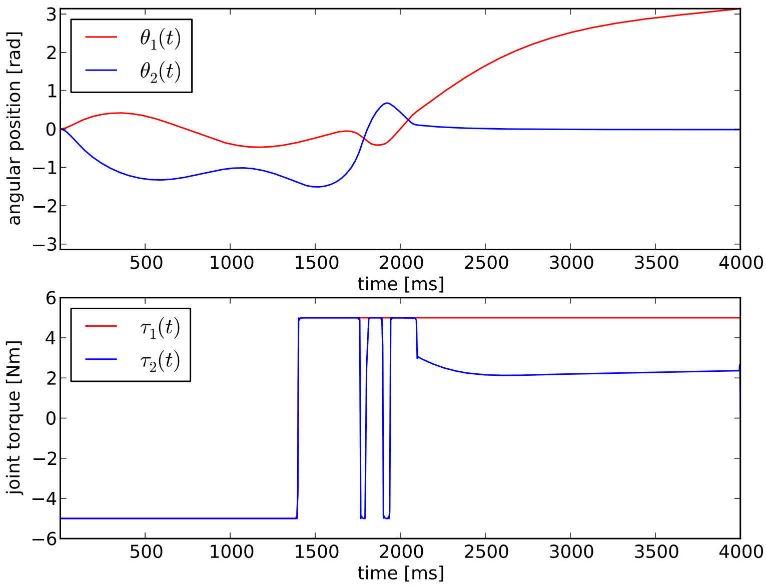


Figure 8.1: Successful swing-up trial. The figure shows the evolution of the system's angular position in a successful trial (top), and the torque profiles generated by the attractor based controller (bottom). As can be seen in (bottom) joint torques saturate for long periods.

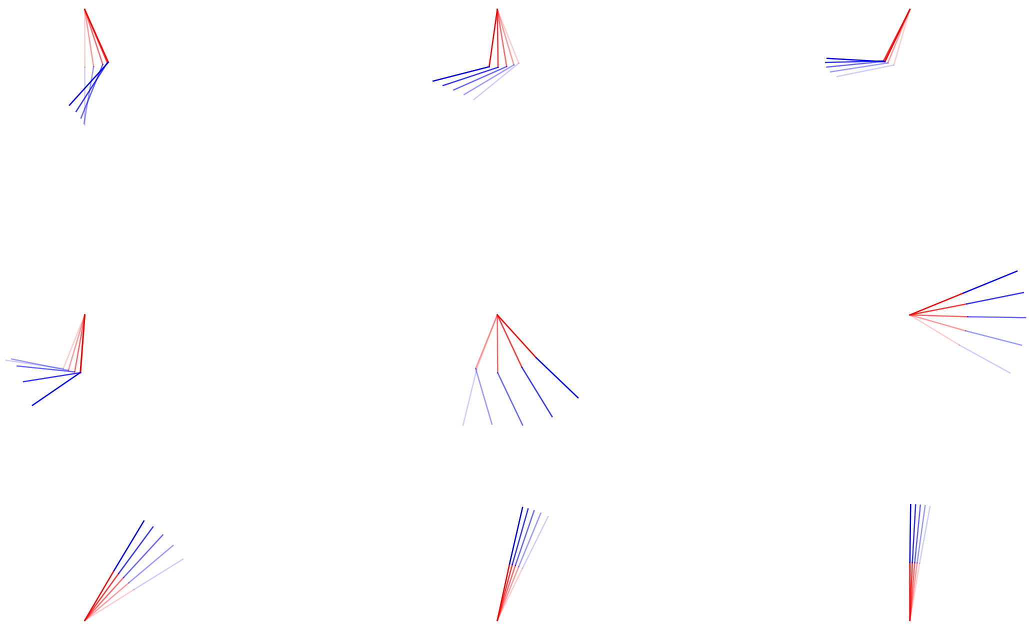


Figure 8.2: Reinforcement learning trial. The figure shows the outcome of a successful learning run with the stick diagram of the swing up.

Another limiting factor of this approach was that the policy parametrization was somehow "stiff". To allow the learning of more complex movements it would be necessary to increase the number of PD controllers or to make it variable. But this would further complicate the long solution search process. Because the big number of required trials would make it impractical to run the algorithm on a real robot, and because of the aforementioned limitations, it was decided to try a more structured model-based approach.

8.2 iLQG

The algorithm described in section 7.3 was run with a 10[ms] timestep. The parameters w_p, w_p and w_r were set to $1.e^4$, $1e^3$ and 1. respectively. The time in which the task had to be accomplished was set to 1.5[s] As in the previous experiment the joint torques were limited to the [-5,5][Nm] range. After parameter tuning, the algorithm converged in 6[s] on a normal laptop (Intel Core 2, 1.8[GHz]). The result of the converged algorithm is shown in Fig.8.3 and Fig.8.4.6

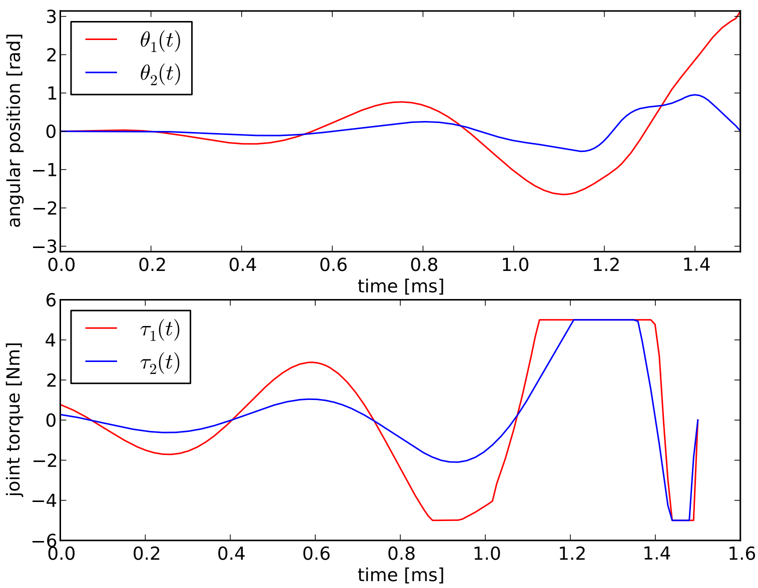


Figure 8.3: Converged swing-up trial. The figure shows the evolution of the system's angular positions in a converged trial (top), and the torque profiles generated by the iLQG controller (bottom).

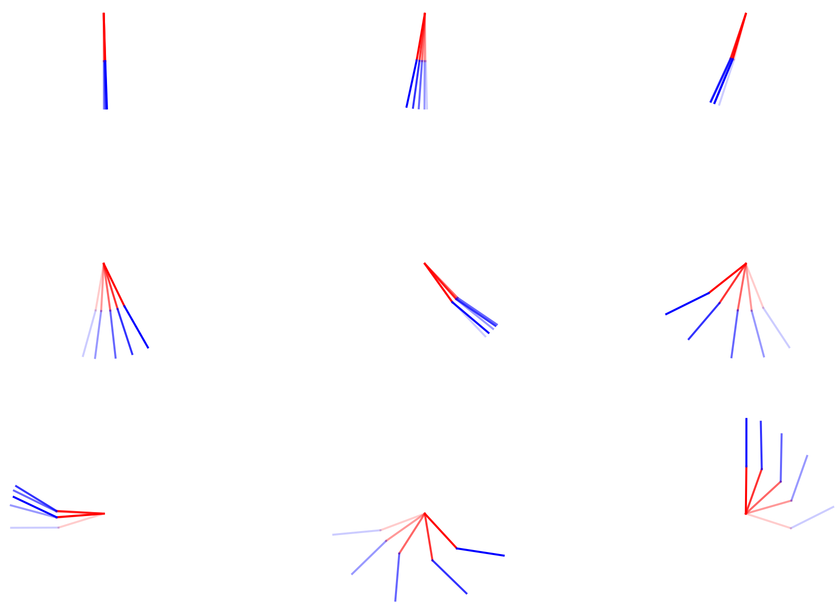


Figure 8.4: iLQG learning trial. The figure shows the outcome of a learning run with the stick diagram of the swing up.

The iLQG algorithm internally relies on the Levenberg-Marquardt gradient descent algorithm (see [Nocedal and Wright, 2006] for details) to optimize the sequence of control commands $\boldsymbol{u}$. The Levenberg-Marquardt algorithm in turn, is based on the iterative adaptation of a damping factor λ with a scaling factor μ. The initial values λ_0 and μ are generally chosen in an empirical way.

To verify the robustness of the iLQG algorithm to variations of these parameters several simulations were performed on the test problem. Fig.8.5 shows the plots of the cost versus computation time for different combinations of λ_0 and μ. As can be seen from the trend of the curves, the cost decrease rate does not depend strongly on the choice of the initialization parameters. Excluding tests with high values of μ (i.e. $\mu > 10$), the time required to reach convergence is below 6 seconds. Low values of the initial damping factor seem to allow faster convergence. In general $\lambda_0 = 0.1$ and $\mu = 8$ seem a reasonable choice for the LMA parameter initialization.

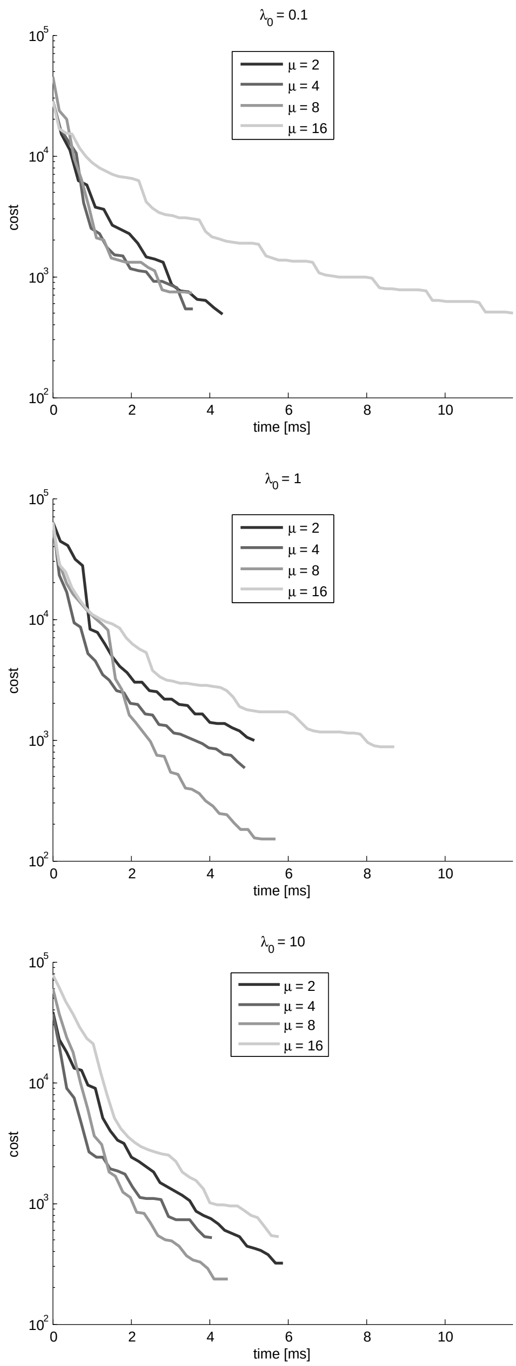


Figure 8.5: Algorithm convergence. The figure shows the effect of the LMA parameters on the decrease of the total cost.

Notably there is a dramatic difference in the time taken to solve the same task by iLQG and the RL method described in section 8.1. RL methods are a viable alternative for solving optimal control problems when no information about the environment is available. For many problems however, RL alone is impractical and the learning problem has to be structured to take advantage of domain knowledge. The iLQG method on the other hand, strongly depends on the knowledge of the system's dynamics for fast convergence.

8.3 iLQG-LD

Solving the swing up problem with the iLQG-LD method required learning the forward dynamics of the system. Two different methods were used to approximate Eq.7.5:

- a feedforward neural network,
- a LWPR neural network.

The input vectors $\boldsymbol{i}$ were constructed by stacking the state and control vectors $\boldsymbol{i} = [\boldsymbol{x}, \boldsymbol{u}]^t$. The output vectors $\boldsymbol{o}$ contained the angular accelerations: $\boldsymbol{o} = [\ddot{\boldsymbol{\theta}}_1, \ddot{\boldsymbol{\theta}}_2]^t$. The dataset comprised $1e^6$ samples generated with the analytic forward dynamic model. Input and output data were normalized to be bounded in the $[-1, 1]$ interval.

The feed forward neural network had two layers: a sigmoidal input layer, and a hidden layer of 30 linear neurons. The neural network was trained with the backpropagation algorithm for 277 epochs and achieved a nMSE in the order of $6.8e^{-6}$. The system dynamics equation learnt with the feed forward neural network trained with the back-propagation algorithm will be denoted as $f_{ffbp}(\boldsymbol{x}_t, \boldsymbol{u}_t)$.

The LWPR neural network was initialized by setting the initial distance metric to 1. Gaussian kernels were chosen as basis functions. To approximate the input-output relationship the LWPR algorithm allocated a total of 91 receptive fields. After training the neural network exhibited a rather high nMSE, in the order of $2.3e^{-2}$. Although several network initializations were tried no extensive parameter search was performed. The system dynamics equation learnt with the LWPR technique will be denoted as $f_{lwpr}(\boldsymbol{x}_t, \boldsymbol{u}_t)$.

The models were then used to solve the task described in section 8.2.

Fig.8.6 compares the cost decrease curves of iLQG-LD with the one obtained with the analytic model. As can be seen, when the iLQG algorithm has to rely on inexact system models the overall performance of the method decreases.

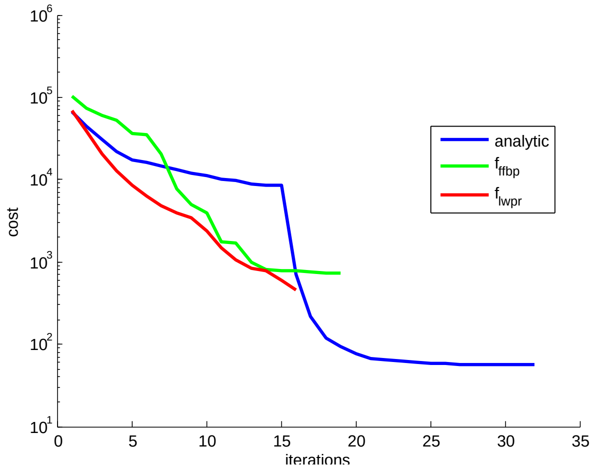


Figure 8.6: Convergence of the iLQG-LD algorithm. The figure shows the cost decrease curves for the iLQG algorithm run with the analytic, the feed-forward neural network, and the LWPR neural network forward dynamic models.

The predicted trajectories are plotted in Fig.8.7 The action sequences computed with the iLQG-LD algorithm however do not take into account model errors. If the feed-forward controls are applied to the "real" the system is driven as shown in Fig.8.8 This prevents the systems whose control actions were planned with learnt models to reach the desired final state. More in detail the control sequence planned with $f_{ffbp}(\boldsymbol{x}_t, \boldsymbol{u}_t)$ fails in bringing θ_2 to θ_2^*, while the one planned with $f_{lwpr}(\boldsymbol{x}_t, \boldsymbol{u}_t)$ fails in bringing θ_1 to θ_1^*. The error in the former case is lower than the error in the latter case: this is probably caused by the large nMSE difference between $f_{ffbp}(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and $f_{lwpr}(\boldsymbol{x}_t, \boldsymbol{u}_t)$.

The iLQG-LD method presents itself an interesting alternative to avoid the requirement of the explicit forward dynamics equations typical of the iLQG algorithm. Although the preliminary results shown so far seem promising, additional tests are required to determine to which extent prediction errors can be tolerated.

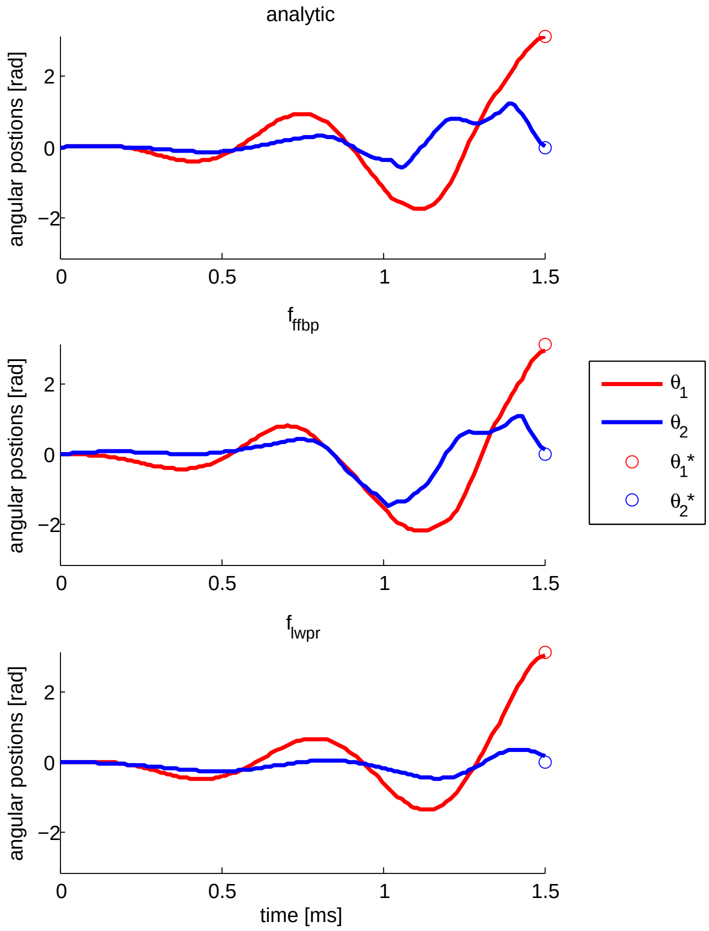


Figure 8.7: Predicted trajectories. The figure shows the predicted joint trajectories for the action sequences planned with the analytic, the feed-forward neural network, and the LWPR neural network forward dynamic models.

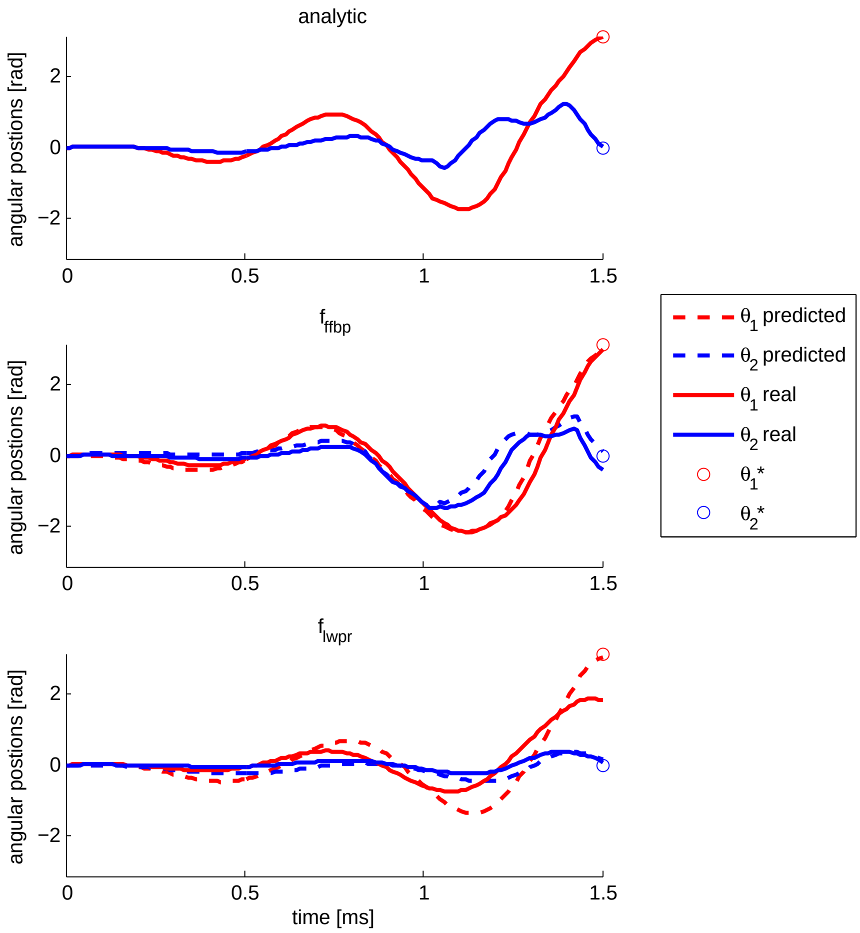


Figure 8.8: Real trajectories. The figure shows the real joint trajectories for the action sequences planned with the analytic, the feed-forward neural network, and the LWPR neural network forward dynamic models. The real trajectories are plotted as solid lines. The predicted trajectories are plotted as dotted lines.

Conclusions

Despite the advances of 60 years of research in the field of robotics the dexterous manipulation capabilities of human are difficult to replicate in artificial system. Indeed this might be due to the inherent "hardware" differences between robots and humans. Robotic actuators are rather fast and powerful respect to muscles. Nevertheless they are characterized by a very low back-drivability: this aspect considerably complicates physical interactions.

In the first part of this thesis the problem back-drivability was introduced and discussed. It was furthermore shown that to increase the mechanical admittance of a robot joint torque control can be used. Joint torque sensors were designed, tested and finally integrated in a iCub arm prototype.

A backdrivable robot can then be used to demonstrate dynamic motor skills. The problem of planning dynamic tasks was addressed in the second part of this thesis. To this end, numerical, machine learning based, methods were implemented and tested in simulation to identify a suitable way to synthetize controllers capable of performing dynamic tasks. The iLQG method turned out to be rather efficient for the present purpose. Preliminary tests integrating a learnt forward dynamics model were also carried out. Further tests in this direction are needed to check how the method scales with the dimensions of the problem, and how tolerant it is to model errors.

Unfortunately because of time limitations it was not feasible to test them directly on the robot: this part remains to be done as future work.

The current trend in research and development of humanoid robots tends to increasingly backdrivable designs, more similar to their human counterparts. The current robot control paradigm, based on trajectory tracking with high gain feedback loops, will prove to be incapable of guaranteeing high performances, as tomorrow's robots structures and actuators evolve. In this context the study of dynamic tasks based on a combination of feedforward and feedback control will become increasingly relevant.

The work presented in this thesis goes in this direction and shall be considered a step in the direction of bridging the gap between the manipulation capabilities of humans and robots.

Acknowledgements

I would like to thank my supervisors Lorenzo Natale and Giorgio Metta without whose help and guidance this work would not have been possible. These last years have been an extremely important experience that would not have been possible it not at the RBCS department: I therefore wish to express my gratitude to Prof. Giulio Sandini. I would like to thank all the people that helped me to accomplish this work (as it would be an endless list and I was afraid of forgetting someone I preferred not to explicitly write all the names). Finally I would like to thank my family and Monica for the constant moral support.

Alberto Parmiggiani

References

— (1960). IRE standards on circuits: definitions of terms for linear signal flow graphs, 1960. *Proceedings of the IRE*, 48(9):1611–1612. 10

Aboaf, E., Drucker, S., and Atkeson, C. (1989). Task-level robot learning: Juggling a tennis ball more accurately. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 14–19. 91

Aghili, F., Buehler, M., and Hollerbach, J. (1997). A joint torque sensor for robots. In *ASME International Mechanical Engineering Congress & Exposition*. 57, 64

Aghili, F., Buehler, M., and Hollerbach, J. M. (2001). Design of a hollow hexaform torque sensor for robot joints. *Int. Jour. of Robotics Research*, 20(12):967–976. 57

Aghili, F., Buehler, M., and Hollerbach, J. M. (2002). Development of a high-performance direct-drive joint. *Advanced Robotics*, 16:233–250. 25

Aghili, F., Hollerbach, J. M., and Buehler, M. (2007). A modular and high-precision motion control system with an integrated motor. *IEEE/ASME Transactions on Mechatronics*, 12:317–329. 25

Albu-Schaffer, A., Ott, C., Frese, U., and Hirzinger, G. (2003). Cartesian impedance control of redundant robots: recent results with the DLR light- weight-arms. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, volume 3, page 37043709. 29

An, C. H. (1986). *Trajectory and force control of a direct-drive arm.* PhD thesis, Massachusetts Institute of Technology. 25

Asada, H. and Kanade, T. (1981). Design of direct-drive mechanical arms. Technical Report CMU-RI-TR-81-04, Robotics Institute, Pittsburgh, PA. 25

Asada, H., Kanade, T., and Takeyama, I. (1982). Control of a direct-drive arm. Technical Report CMU-RI-TR-82-04, Robotics Institute, Pittsburgh, PA. 24

Asada, H. and Youcef-Toumi, K. (1987). *Direct Drive Robots.* MIT Press. 24

Atkeson, C. G., Hale, J. G., Pollick, F., Riley, M., Kotosaka, S., Schaal, S., Shibata, T., Tevatia, G., Ude, A., Vijayakumar, S., and Kawato, M. (2000). Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, 15(4):46–56. 35

Bentivegna, D. C., Atkeson, C. G., and Kim, J.-Y. (2008). Compliant control of a hydraulic humanoid joint. In *IEEE/RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*, volume 2. 35

Berret, B., Darlot, C., Jean, F., Pozzo, T., Papaxanthis, C., and Gauthier, J. (2008). The inactivation principle: mathematical solutions minimizing the absolute work and biological implications for the planning of arm movements. *PLoS Computational Biology*, 4(10):. 87

Brooks, R. A., Breazeal, C., Marjanovic, M., Scassellati, B., and Williamson, M. (1999). *The Cog project: Building a humanoid robot*, page 5287. Springer, New York. Lecture Notes in Artificial Intelligence 1562. 32

Buehler, M. and Koditschek, D. (1990). From stable to chaotic juggling: Theory, simulation, and experiments. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 1976–1981. 91

Chandler, R. F., Clauser, C. E., McConville, J. T., Reynolds, H. M., and Young, J. W. (1975). Investigation of inertial properties of the human body. Technical report, Air Force Aerospce Medical Research Lab. Wright-Patterson, AFB, OH, USA. Final rept. Apr 1972-Dec 1974. 13

Chen, W.-K. (1997). *Graph theory and its engineering applications.* World Scientific. 10

Cheng, G., Sang-Ho, H., Ude, A., Morimoto, J., Hale, J., Hart, J., Nakanishi, J., Bentivegna, D., Hodgins, J., Atkeson, C., Mistry, M., Schaal, S., and Kawato, M. (2007). CB: Exploring neuroscience with a humanoid research platform. *Advanced robotics*, 21(10):1097–1114. 35

Colgate, J. E. (1988). *The control of dynamically interacting systems.* PhD thesis, Massachusetts Institue of Technology. 23

Constantinos Mavroidis, C. P. and Mosley, M. (1999). *Automation, Miniature Robotics and Sensors for Non-Destructive Testing and Evaluation*, chapter Conventional actuators, shape memory alloys, and electrorheological fluids. Y. Bar-Cohen Editor. 33

Desmurget, M. and Grafton, S. (2000). Forward modeling allows feedback control for fast reaching movements. *Trends in Cognitive Sciences*, 4(11):423–431. 85

Edsinger-Gonzales, A. and Weber, J. (2004). Domo: a force sensing humanoid robot for manipulation research. *International Journal of Humanoid Robotics*, 1:279–291. 33

Eppinger, S. and Seering, W. (1987). Understanding bandwidth limitations in robot force control. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 4, pages 904–909. 24

Franklin, G. (1993). *Feedback control of dynamic systems.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA. 87

Grzeszczuk, R. and Terzopoulos, D. (1998). Automated learning of muscle-actuated locomotion through control abstraction. In *Proc. SIGGRAPH*, pages 9–20. 91, 97

HarmonicDrive (2010). Product website. `http://www.harmonicdrive.net/media/support/catalogs/pdf/csd-shd-catalog.pdf`. 17, 18, 41

Hashimoto, M. (1989). Robot motion control based on joint torque sensing. In *Proc. IEEE Int. Conf on Robotics and Automation*, pages 256–261. 58

Hashimoto, M., Hattori, T., Horiuchi, M., and Kamata, T. (2002). Development of a torque sensing robot arm for interactive communication. In *Proc. IEEE Int. Workshop on Robot and Human Interactive Communication*, pages 344–349. 58

Hashimoto, M., Kiyosawa, Y., and Paul, R. (1993). A torque sensing technique for robots with harmonic drives. *IEEE Transactions on Robotics and Automation*, 9(1):108–116. 58

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. (1998). The development of Honda humanoid robot. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1321–1326. 15

Hirzinger, G., Albu-Schaffer, A., Hahnle, M., Schaefer, I., and Sporer, N. (2001). On a new generation or torque controlled light-weight robots. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3356–3363. 27

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. 57

Hirzinger, G., Butterfass, J., M. Fischer M. Grebenstein, M. H., Liu, H., Schaefer, I., and Sporer, N. (2000). A mechatronics approach to the design of light-weight arms and muitifingered hands. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 46–54. 27, 58

Hirzinger, G., Sporer, N., Albu-Schaffer, A., Hahnle, M., Krenn, R., Pascucci, A., and Schedl, M. (2002). DLR's torque-controlled light weight robot III-are we reaching the technological limits now? In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1710–1716. 27

Hogan, N. and Buerger, S. P. (2004). *Robotics and automation handbook*, chapter 19. Impedance and interaction control. CRC Press. 23

Hollerbach, J. M., Hunter, I. W., and Ballantyne, J. (1992). *A comparative analysis of actuator technologies for robotics*, pages 299–342. MIT Press, Cambridge, MA, USA. 25, 33

Hollerbach, J. M. and Jacobsen, S. C. (1996). Anthropomorphic robots and human interactions. In *1st Int. Symp on Humanoid Robots, Waseda University*, pages 83–91. 34

Holmberg, R., Dickert, S., and Khatib, O. (1992). A new actuation system for high-performance torque-controlled manipulators. In *in Proc. of the Ninth CISM-IFToMM Symp. on the Theory and Practice of Robots and Manipulators*, pages 285–292. 27, 57

Hong, W. and Slotine, J. J. (1995). Experiments in hand-eye coordination using active vision. In *Proc. of the Int. Symp. on Experimental Robotics*. 27

Ishida, T., Kuroki, Y., and Yamaguchi, J. (2003). Mechanical system of a small biped entertainment robot. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1129–1134. 15

Ishida, T. and Takanishi, A. (2006). A robot actuator development with high backdrivability. In *IEEE Conf. on Robotics, Automation and Mechatronics*. 6

Iwata, H. and Sugano, S. (2009). Design of human symbioic robot TWENDY-ONE. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 580–586. 33

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:207–354. 91

Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., and Akachi, K. (2008). Humanoid robot hrp-3. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2471–2478. 15

Karnopp, D. C., Margolis, D., and Rosemberg, R. C. (2000). *System dynamics - modeling and simulation of mechatronic systems*. Wiley Interscience. 9

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727. 84

Kawato, M., Furukawa, K., and Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57(3):169–185. 84

Kazerooni, H. (1991). Instrumented harmonic drives for robotic compliant maneuvers. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, volume 3, pages 2274–2279 vol. 58

Kazerooni, H. (1995). Dynamics and control of instrumented harmonic drives. *Journal of dynamic systems, measurement, and control*, 117:15. 58

Kelso, J. S. (1982). *Human motor behavior: an introduction.* Lawrence Erlbaum Associates, Inc. 1

Kober, J. and Peters, J. (2008). Policy search for motor primitives in robotics. *Advances in Neural Information Processing Systems (NIPS)*, :. 92

Kober, J. and Peters, J. (2009). Learning motor primitives for robotics. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2509–2515. 92

Kollmorgen-DanaherMotion (2010). Product website. http://www.kollmorgen-seidel.de/website/com/eng/download/document/200512291032290.RBE_Series_Motors_Brochure.pdf. 17, 41

Li, W. (2006). *Optimal control for biological movement systems.* PhD thesis, Engineering Sciences, University of California, San Diego. 95

Lynch, K. M. and Mason, M. T. (1996). Dynamic underactuated nonprehensile manipulation. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* 91

Matweb (2010a). 39NiCrMo3 data-sheet. http://www.matweb.com/search/datasheet_print.aspx?matguid=697130f21da64542a68bf61911f2f495. 43

Matweb (2010b). Al 6082 data-sheet. http://www.matweb.com/search/datasheet_print.aspx?matguid=fad29be6e64d4e95a241690f1f6e1eb7. 43

Matweb (2010c). Al 7075 data-sheet. http://www.matweb.com/search/datasheet_print.aspx?matguid=9852e9cdc3d4466ea9f111f3f0025c7d. 43

McMahon, T. A. (1984). *Muscles, reflexes, and locomotion.* Princeton University Press. 12

Metta, G., Sandini, G., Vernon, D., Caldwell, D., Tsagarakis, N., Beira, R., Victor, J. S., Ijspeert, A., Righetti, L., Cappiello, G., Stellin, G., and Becchi, F. (2005). The robotcub project: an open framework for research in embodied cognition. In *Proc. IEEE/RAS Int. Conf. on Humanoid Robots (HUMANOIDS).* 1, 15, 39

Metta, G., Vernon, D., and Sandini, G. (2004). Deliverable 8.1. Initial specification of the iCub open system. `http://www.robotcub.org/index.php/robotcub/content/download/614/2215/file/D8.1.pdf`. 40

Mitrovic, D., Klanke, S., and Vijayakumar, S. (2010). *From Motor Learning to Interaction Learning in Robots*, chapter Adaptive optimal feedback control with learned internal dynamics models, pages 65–84. Springer. 92, 96, 97

Mizuuchi, I. (2006). A musculoskeletal flexible-spine humanoid kotaro aiming at the future in 15 years time. *Mobile Robots-Towards New Applications*, :45–56. 37

Mizuuchi, I., Tajima, R., Yoshikai, T., Sato, D., Nagashima, K., Inaba, M., Kuniyoshi, Y., and Inoue, H. (2002). The design and control of the flexible spine of a fully tendon-driven humanoid "kenta". In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2527–2532. 37

Mizuuchi, I., Waita, H., Nakanishi, Y., Yoshikai, T., Inaba, M., and Inoue, H. (2004). Design and implementation of reinforceable muscle humanoid. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1, pages 828–833. 37

Morrell, J. and Salisbury, J. (1995). Parallel coupled actuators for high performance force control: a micro-macro concept. In *Proc. ot the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1. 36

Nagakubo, A., Kuniyoshi, Y., and Cheng, G. (2000). Development of a high-performance upper-body humanoid system. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 3, pages 1577–1583. 27

Nagakubo, A., Kuniyoshi, Y., and Cheng, G. (2003). The ETL-humanoid system - a high-performance full-body humanoid system for versatile real-world interaction. *Advanced robotics*, 17(2):149–164. 15, 27

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer. 103

Oatis, C. A. (1993). The use of a mechanical model to describe the stiffness and damping characteristics of the knee joint in healthy adults. *Phis. Ther.*, 73(11):740–749. 12

Paynter, H. (1961). *Analysis and design of engineering systems*. MIT-press. 9

Pratt, G. and Williamson, M. (1995). Series elastic actuators. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 399–406. 29

Prochazka, A., Bennett, D. J., Stephens, M. J., Patrick, S. K., Sears-Duru, R., Roberts, T., and Jhamandas, J. H. (1997). Measurement of rigidity in Parkinson's disease. *Movement Disorders*, 12(1):24–32. 12

Riley, M. and Atkeson, C. G. (2002). Robot catching: Towards engaging human-humanoid interaction. *Auton. Robots*, 12(1):119–128. 91

Rizzi, A. and Koditschek, D. (1992). Progress in spatial robot juggling. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 775–780. 91

Rosenstein, M. T. (2003). *Learning to Exploit Dynamics for Robot Motor Coordination.* PhD thesis, University of Massachusetts Amherst. 91

Rosenstein, M. T., Barto, A. G., and Emmerik, R. E. A. V. (2006). Learning at the level of synergies for a robot weightlifter. *Robotics and Autonomous Systems*, 54(8):706–717. 91, 94, 95, 99

Salisbury, J. K., Townsend, W. T., DiPietro, D. M., and Eberman, B. S. (1991). Compact cable transmission with cable differential. Patent. US 4 903 536. 26, 43

Schaal, S. and Atkeson, C. (1993). Open loop stable control strategies for robot juggling. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 913–913. 91

Schaal, S. and Schweighofer, N. (2005). Computational motor control in humans and robots. *Current Opinion in Neurobiology*, 15(6):675–682. 83

Sharon, A. (1989). *The macro/micro manipulator: an improved architecture for robot control.* PhD thesis, Massachusetts Institute of Technology. Dept. of Mechanical Engineering. 36

Shin, D., Sardellitti, I., Park, Y., Khatib, O., and Cutkosky, M. (2008). Design and control of a bio-inspired human-friendly robot. In *Proc. of the 11th Int. Symp. on Experimental Robotics*, volume 2008. Springer. 36

Spong, M. (1994). Swing up control of the acrobot. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, volume 3, pages 2356–2361. 91

Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Dynamics and Control*. John Wiley and Sons. 88

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: an introduction*. MIT Press. 94

Taghirad, H. and Belanger, P. (1999). Intelligent built-in torque sensor for harmonic drive systems. *IEEE Transactions on Instrumentation and Measurement*, 48(6):1201–1207. 58

Tellez, R., Ferro, F., Garcia, S., Gomez, E., Jorge, E., Mora, D., Pinyol, D., Oliver, J., Torres, O., Velasquez, J., and Faconti, D. (2008). Reem-B: an autonomous lightweight human-size humanoid robot. In *Proc. IEEE/RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*, pages 462–468. 15

Timoshenko, S. P. and Goodier, J. (1970). *Theory of Elasticity*. McGraw-Hill. 61, 64

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9):907–915. 84, 86

Todorov, E. and Li, W. (2005). A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proc. of the American Control Conference*, volume 1, page 300. 2, 92, 94, 95

Townsend, W. T. (1988). *The effect of transmission design on force-controlled manipulator performance*. PhD thesis, Massachusetts Institue of Technology. 6, 43

Townsend, W. T. and Salisury, J. K. (1993). Mechanical design for whole-arm manipulation. In Dario, P., Sandini, G., and Aebischer, P., editors, *Robots and biological systems: towards a new bionics?*, pages 153–164. Springer. 6, 26, 43

Tsagarakis, N.G., Metta, G., Sandini, G., Vernon, D., Beira, R., Becchi, F., Righetti, L., Santos-Victor, J., Ijspeert, A.J., Carrozza, M.C., Caldwell, and D.G. (2007). icub: the design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics*, 21(10):1151–1175. 52

Tsai, L.-W. (1999). *Robot analysis*. Wiley interscience. 44, 46, 50

Van Ham, R., Sugar, T., Vanderborght, B., Hollander, K., and Lefeber, D. (2009). Compliant actuator designs. *Robotics & Automation Magazine, IEEE*, 16(3):81–94. 21

Venture, G., Yamane, K., Nakamura, Y., and Hirashima, M. (2007). Estimating viscoelastic properties of human limb joints based on motion capture and robotic identification technologies. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 624–629. 12, 13, 18

Vijayakumar, S., D'souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634. 97

Vischer, D. and Khatib, O. (1995). Design and development of high-performance torque-controlled joints. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 537–544. 27, 57

Williamson, M. M. (1995). Series elastic actuators. Master's thesis, Massachussets Institute of Technology. 29

Williamson, M. M. (1999). *Robot arm control exploiting natural dynamics.* PhD thesis, Massachusetts Institute of Technology. 91

Wolpert, D. and Ghahramani, Z. (2000). Computational principles of movement neuroscience. *nature neuroscience*, 3:1212–1217. 84, 85

Wolpert, D. and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8):1317–1329. 86

Wolpert, D., Miall, R., and Kawato, M. (1998). Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347. 84, 85

Wu, C. and Paul, R. P. (1980). Manipulator compliance based on joint torque control. In *IEEE Conference on Decision and Control*, volume 19, pages 88–94. 21

Wyrobek, K., Berger, E., der Loos, H. V., and Salisbury, K. (2008). Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. 36

Youcef-Toumi, K. (1985). *Analysis, design and control of direct-drive manipulators.* PhD thesis, Massachusetts Institute of Technology. 25

Zanasi, R. (1991). Power-oriented modeling of dynamical systems for simulation. In *IMACS Symp. on Modeling and Control of Technological Systems (MCTS)*, volume 2, pages 31–35. 9

Zanasi, R. (1993). Power-oriented graphs for modeling robotic systems. In *Proc. IMACS/IFAC Second Int. Symp. (MIM-S2)*, pages 2–16. 9

Zanasi, R. (1994). Dynamics of a N-links manipulator by using power-oriented graphs. In *Proc. Int. IFAC Symp. on Robot Control (SYROCO)*, volume 2. 9

Zanasi, R. and Salisbury, K. J. (1992). Dynamic modeling, simulation and parameter identification for the WAM arm. Technical Report A.I. Memo No. 1387, MIT, Cambridge, MA, USA. 9

Zinn, M., Khatib, O., Roth, B., and Salisbury, J. K. (2004). Playing it safe [human-friendly robots]. *IEEE Robotics & Automation Magazine*, 11(2):12–21. 36