YARP: YET ANOTHER RIDICULOUS PAPER

IIT ROBOTICS LAB

Abstract

In this paper we describe the actuation and control of a humanoid robot neck. Particular attention will be posed on the description of the neck actuation structure, the design of which results in a noticeable human similarity. Specifically, the final mechanical design was inspired by the human skeleton, with the neck bone movements constrained and actuated by the surrounding muscles. In our robotic platform, the neck bone was realized with a steel spring surrounded by steel tendons in place of muscles. The specific and innovative mechanical design have imposed the design of a non-standard actuation structure which, in turn, have lead to an innovative control scheme.

1. Introduction

Humans exhibit a wide and complex repertoire of movements far beyond the motor capabilities of modern robots. Clearly, the realization of an artificial system capable of more realistic movements passes trough a series of technological improvements, especially if we are interested in replicating both kinematic and dynamical aspects. Recently, there has been a growing interest in developing robots whose geometric and actuation structures resemble those of a human being. Probably, one of the most extreme steps in this direction is represented by the robot recently developed by O. Holland and colleagues [1]; another interesting example is the humanoid robot Kotaro [3].

2. Humanoid platform

The robotic platform on which the discussed controller has been implemented is the humanoid robot James [2].
James is a 22-DOF torso with moving eyes and neck, an arm and a highly anthropomorphic hand (see Figure 1). In the following subsections we briefly cover the robot design, its actuation, and sensorization. More details about the neck structure are given in section 3.

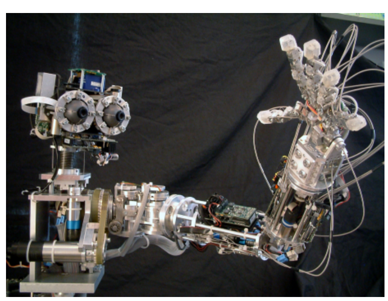

Figure 1. The humanoid robot James.

2.1. **Robot design.** The robot structure is similar to that of humans, both in size (approximatively that of a ten-year-old boy), number of DOFs and range of movements; the total weight (about 8 kg: 2 kg the head, 4 kg the torso and 2 kg arm and hand together) has been kept low by the employment of aluminum and Ergal.
As already pointed out, such a complex structure is, in our view, mandatory when trying to replicate human-like movements on a humanoid robot.
The head is equipped with two eyes, which can pan and tilt independently (4 DOFs), and is mounted on the 3-DOF neck, which allows the movement of the head as needed in the 3D rotational space.
The arm has 7 DOFs: three of them are located in the shoulder, one in the elbow and three in the wrist. The hand has five fingers, with complexively 17 degrees of freedom, underactuated by just 8 motors.

2.2. **Actuation system.** The 22 DOFs are actuated by a total of 23 motors, whose torque is transmitted to the joints by plastic toothed belts and stainless-steel tendons, provided with springs at critical locations.
This solution is appealing for at least two reasons. First, it allows to put the motors far from the joints, and so to distribute the weights within the robot in a smart way (i.e. put the heaviest motors on the static parts, like the torso). Second, the intrinsic elesticity of belts, tendons and springs gives a noteworthy compliance to the whole structure, as humans have, allowing the robot to move safely in a dynamic and unknown environment.

2.3. **Sensory system.** The robot is equipped with vision, proprioception, kinesthetic and tactile inputs.
Vision is provided by two digital CCD cameras (PointGrey Dragonfly remote head), located in the eyeballs.
The proprioceptive and kinesthetic senses are achieved through position sensors (magnetic incremental encoders connected to all motors and absolute-position sensors on the shoulder and in the fingers); furthermore, a 3-axis orientation tracker (Intersense iCube2) has been mounted on top of the head, to emulate the vestibular system.
Tactile information is extracted from several magnetic silicone-made pressure sensors which have been specifically designed and developed for James, placed in the fingers.

3. Neck structure

The neck is constituted by a steel spring, which holds the head giving it the possibility of bending forward (pitch) and laterally (roll). The actuation of these two degrees of freedom is obtained with a peculiar structure, recalling the design of a parallel manipulator. Specifically, the neck is surrounded by three steel tendons, whose length determine the position of the spring and therefore, the pitch and roll position of the neck. The length of the tendons is adjusted by means of three motors, positioned at the base of the neck (see Figure 2). On top of the spring, a fourth motor is mounted, directly actuating a third degree of freedom, the head yaw (i.e. rotation around an axis parallel to the pan axes of the two eyes).

3.1. **Redundancy of the actuation scheme.** Clearly, the above actuation structure is somehow redundant. Practically, the pitch and roll movements (two degrees of freedom) are actuated by three motors. In a mathematical sense, redundancy corresponds to the fact that the same configuration of the system can be achieved by different positions of the actuators. Classical techniques can be used to exploit the advantages of redundant systems [4]. However, in our case there are additional constraints that will rule out redundancy.

To understand the structure of the problem, let us reduce the structure in a two dimensional space. In this situation, we have two independent motors to actuate a single degree of freedom, nominally the slope of the surface on which the head is mounted. Consider first the system whose actuation scheme is given in Figure 3. Considering the surface slope

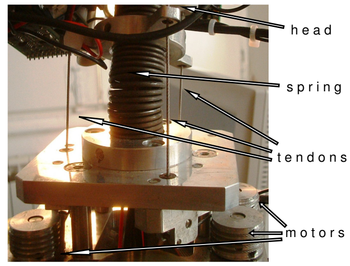

FIGURE 2. Neck actuation system. Each motor pulls a tendon in order to bring the spring (i.e. the neck) in the desired configuration.

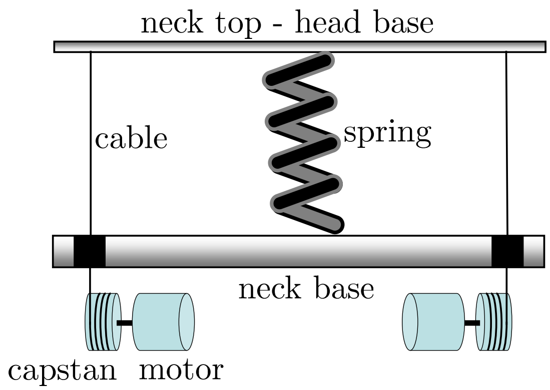

FIGURE 3. Two dimensional scheme of an actuation system similar to james's neck.

as the task of our control, this system is redundant. Roughly speaking, shortening both cables of the same length does not produce any slope movement but only a variation of the spring compression. Classically, this is what we define redundancy in the actuation. As previously said, in our case there are additional constraints that rule out this redundancy. Practically speaking, the spring at the base of the neck is entirely compressed by the weight of the carried electronics and mechanics[1] (motors, cameras, chassis, etc. etc.), as indicated in Figure 4. As a consequence of this fact, shortening/lengthening both cables simultaneously no longer produces a variation of the spring compression but only a varies the cable tensions. Remarkably, these tensions should be kept under control: high tensions damage the spring spirals and low tensions cause wrong alignment of the cables on the capstans. Ideally, the tension of the cables can be controlled with the use of suitable sensors capable of measuring the cable tensions themselves. In our system these sensors are not currently available. Therefore, in the present paper we show how to keep the cable tensions under control by means of a kinematic model of the system.

4. CONTROL OF THE NECK

The peculiar structure of the neck has required the design of an original control technique. The final design makes use of the 3-axis orientation tracker positioned on top of

[1]The description of the system in Figure 3 is merely for understanding the issue of redundancy. Practically speaking it would be very difficult to model the kinematics of this system. The actual system, i.e. the one schematically represented in Figure 4, will be much easier to model kinematically. This observation suggests why we did not choose a stiffer spring capable of sustaining the head weight.

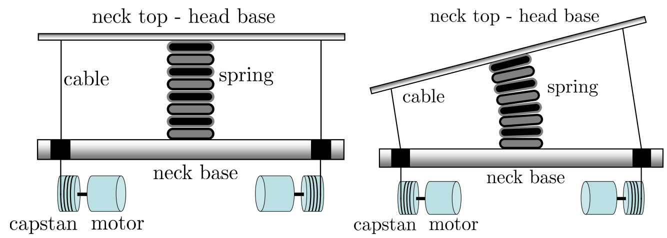

FIGURE 4. Equivalent two dimensional scheme of james's neck.

the robot head. Though the given sensor is capable of measuring its absolute rotation along three orthogonal axes, in the given application we used only part of the available information. In particular, we used the measurements corresponding to the head pitch and roll rotations, denoted θ_p and θ_r respectively. The yaw rotation θ_y is instead measured by the encoder mounted directly on the shaft of the associated motor.

4.1. **Neck control in details.** As already pointed out, the neck structure is characterized by three degrees of freedom: pitch θ_p, roll θ_r and yaw θ_y. The yaw movement, is directly actuated by a single dc motor; its control is based on a standard PID controller. The control strategy for the remaining two movements will be instead described in details in this section.

The design of the pitch and roll control loops has required the development of a MATLAB model of the neck structure[2]. As already pointed out the system is somehow redundant (3 actuators versus 2 degrees of freedom) but the redundancy needs to be ruled out in order to keep the tendons tension within certain limits and the spring spirals aligned. Practically, because of redundancy, the same neck orientation $\mathbf{x} = [\theta_p,\ \theta_r] \in \mathbb{R}^2$ (i.e. the orientation tracker measurements) can be achieved with different tendons (cables) configurations $\mathbf{q} = [d_1,\ d_2,\ d_3] \in \mathbb{R}^3$ (i.e. the position of the motors). Among all these configurations, there is an ideal one $\mathbf{q}^*$ which corresponds to straight tendons and constant curvature of the spring. This configuration can be easily computed (see appendix for details) thus leading to the following:

$$\mathbf{q}^* = f(\mathbf{x}). \tag{4.1}$$

The reader should notice that (4.1) is a sort of inverse kinematic model[3] expressing the configuration of the motors to achieve a desired neck orientation. Geometrically speaking, (4.1) defines a two dimensional manifold embedded in the three dimensional space of the cables configurations. If the model were perfectly corresponding to the real system, the problem of orienting the neck in a desired configuration $\mathbf{x}_d$ would be easily solved by computing the desired tendons length $\mathbf{q}_d = f(\mathbf{x}_d)$ and controlling the positions of the three motors[4] so as to regulate the tendons to the desired configuration. Practically speaking, every model has its own errors and therefore the proposed scheme will never orient precisely the neck.

Using this model we were able to compute the ideal tendons lengths given the pose of the neck, or equivalently the ideal tendons lengths $(l_1,\ l_2,\ l_3)$ given the inertia sensor

[2]The model is based on the assumption that the spring has a constant length. Practically, when the spring bends on a side, it maintains its length on that side (remember that the spring is compressed by the head weight) while stretching on the opposite side. This kinematic can be easily modeled with MATLAB. When the spring is bent, the assumption is that its curvature is constant along the entire spring length.

[3]Its forward counterpart (expressing the neck orientation $\mathbf{x} \in \mathbb{R}^2$ as a function of the tendons length $\mathbf{q} \in \mathbb{R}^3$) is much more complicated to be computed and its computation falls outside the scope of this paper.

[4]The three motors are equipped with encoders so that the motor position control has been easily achieved with a simple PID controller based on feedback from encoders.

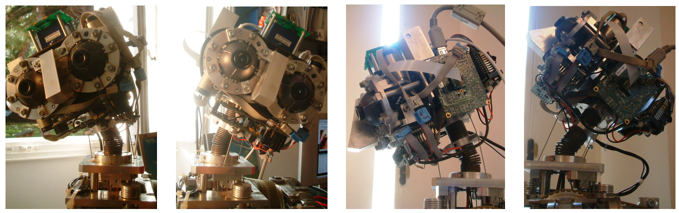

FIGURE 5. James head. Different configurations seen from different views. Its complex actuation system gives compliance and wide ranges to the motion.

measurement (θ_r, θ_p). Practically, the model of the system is a function $f : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ such that:

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = f(\theta_r, \theta_p). \tag{4.2}$$

The final control loop for positioning the neck in the desired configuration (θ_r^d, θ_p^d) is the following:

$$\begin{bmatrix} \frac{dl_1}{dt} \\ \frac{dl_2}{dt} \\ \frac{dl_3}{dt} \end{bmatrix} = -\begin{bmatrix} \frac{\partial f}{\partial \theta_r} & \frac{\partial f}{\partial \theta_p} \end{bmatrix} \begin{bmatrix} \theta_r - \theta_r^d \\ \theta_p - \theta_p^d \end{bmatrix}, \tag{4.3}$$

where $\begin{bmatrix} \frac{\partial f}{\partial \theta_r} & \frac{\partial f}{\partial \theta_p} \end{bmatrix}$ is the Jacobian of the function f with respect to θ_r, θ_p computed at the current configuration θ_r, θ_p.

The above model (4.1) is ideal and assumes that the three tendons are always subject to a minimum tension. Due to the imperfections in the model, the tendons may loose tension if the control strategy (4.3) is applied. Given a long enough time window the controller might drift. A corrective term is therefore required. The solution is:

$$\begin{bmatrix} \frac{dl_1}{dt} \\ \frac{dl_2}{dt} \\ \frac{dl_3}{dt} \end{bmatrix} = -(1-\gamma)\begin{bmatrix} \frac{\partial f}{\partial \theta_r} & \frac{\partial f}{\partial \theta_p} \end{bmatrix} \begin{bmatrix} \theta_r - \theta_r^d \\ \theta_p - \theta_p^d \end{bmatrix} - \gamma \left(\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} - f(\theta_r, \theta_p) \right), \tag{4.4}$$

where γ is an arbitrary constant in the range $[0, 1]$. The second term of the controller (the one multiplied by γ) guarantees that the the length of the cables remains similar to the length of the model. This strategy is sufficient to guarantee that the tendons maintain a tension which is more or less constant across different configurations of the structure (see Figure 5). In this final configuration the jacobian $\begin{bmatrix} \frac{\partial f}{\partial \theta_r} & \frac{\partial f}{\partial \theta_p} \end{bmatrix}$ can be substituted with a constant jacobian computed at the reference configuration θ_r=θ_p=0.

5. Experimental data

6. Future work

Preliminary works have been carried out concerning the development of a non-model-based controller for James neck.

The planned solution makes use of a Receptive Fields Neural Network to extimate the joint-space velocities needed to move the head with the desired task-space velocities, given the actual joint configuration. The learning algorithm is an implementation, with some modifications, of the Receptive Fields Weighted Regression proposed by Schaal and Atkenson [5].

After a training phase, in which the workspace is randomly explored by the robot and the network learns on-line from the gathered sensory data, the trained network is able to map the two spaces. This mapping is called, in mathematical terms, the global inverse Jacobian matrix.

Unfortunately, due to the redundant actuation system we deal with, our Jacobian matrix is not square, and so not invertible. This means that for a given set of actual joints positions and required task-space velocities there is not a unique solution in terms of joint-space velocities; nevertheless, within this family of solutions, just a very limited set is usable in practice, because a minimum amount of tension is needed along the tendons to avoid their possible outgo from the capstans.

To overcome this problem, the velocities applied to the motors during the training phase result from the summation of two weighted terms: the first is a random one, and has a bigger weight, and the second tries to keep a suitable tension along the tendons, acting in the null-space of the Jacobian.

A first stage of development for the discussed controller has been already reached, and an initial version has been tested on James. Results show that the system is able to learn a good approximation of the inverse Jacobian after a training stage of about a hour. The trained controller allows the head to reach the desired equilibrium position in the task-space, with logarithmic velocity profiles and a satisfying degree of accuracy.

7. Conclusions

References

[1] O. Holland and R. Knight. The anthropomimetic principle. In *Symposium on Biologically Inspired Robots, Bristol, England*. AISB, April 2006.

[2] L.Jamone, F. Nori, G.Metta, and G.Sandini. James: A humanoid robot acting over an unstructured world. In *Humanoids 2006. University of Genova, Italy*, 2006.

[3] I. Mizuuchi, T. Yoshikai, Y. Sodeyama, Y. Nakanish, A. Miyadera, T. Yamamoto, T. Niemel, M. Hayashi, J. Urata, Y. Namiki, T. Nishino, and M. Inaba. Development of musculoskeletal humanoid kotaro. In *International Conference on Robotics and Automation (ICRA)*. IEEE, May 2006.

[4] C. Samson, B. Espiau, and M. Le Borgne. *Robot Control: the Task Function Approach*. Oxford University Press, 1991.

[5] S. Schaal and C. G. Atkenson. Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084, 1998.