2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Machine-Learning Based Control of a Human-like Tendon-driven Neck

Lorenzo Jamone, Matteo Fumagalli, Giorgio Metta
University of Genoa and Italian Institute of Technology
Via Morego 30, Genova, ITALY
lorenzo.jamone, matteo.fumagalli, giorgio.metta}@iit.it

Lorenzo Natale, Francesco Nori, Giulio Sandini
Italian Institute of Technology
Via Morego 30, Genova, ITALY
lorenzo.natale, francesco.nori, giulio.sandini}@iit.it

Abstract— **This paper describes the control of a human-like robotic neck actuated with tendons. The controller regulates the length of the tendons to achieve a desired orientation of the neck and at the same time it maintains the tension of the tendons within certain limits. The solution we propose does not use any model of the system, but it relies on online learning of the different Jacobian mappings required by the controller. Learning, data acquisition and control are simultaneous; thus learning is completely autonomous, and purely online. We show that after enough iterations the controller produces straight trajectories in the task space and is able to maintain the tension of the tendons within safe limits.**

I. Introduction

We propose a strategy based on autonomous online learning to control the orientation of a tendon-driven parallel system. The controller regulates the length of the tendons to fulfill a primary task (controlling the neck orientation), and at the same time it tries to keep the tension of the tendons within safe limits. The robotic neck we used in this paper is a peculiar mechanical system that emulates the human neck. This setup is a particularly interesting test platform for the following reasons. Firstly it is a tendon driven system. Tendon driven systems are getting popularity in robotics because they allow sophisticated routing of the actuation thus to reduce weight and inertia of the mechanical structure. Secondly, this setup is (roughly speaking) a redundant system in which the number of actuators is higher than the mechanical degrees of freedom. Finally, the neck is equipped with a sensory system that provide a rich feedback about motor positions, neck orientation and tendon tensions.

Previous works [1], [2] tested different solutions to this problem, all exploiting more or less accurate analytical models of the system. In these approaches errors arise from discrepancies between the model and the real system: of course, the more complex the system the more probable the errors. In particular, non-linear and time-varying parameters (e.g. due to elasticity or deformable parts) are difficult to describe. On the other hand, the absence of a model reduces the possibility to control the system. In these cases learning offers an elegant and efficient solution to the problem, especially if we can exploit a considerable amount of sensory measurements.

Unfortunately learning in the literature is most often performed off-line (or batch). Learning and data acquisition are performed in separate phases. On the other hand we are interested in implementing techniques that produce a constant adaptation of the parameters of the controller.

The paper is organized as follows. In sections II we give a description of the hardware platform, focusing in particular on the neck structure. In sections III the structure of the controller is described, discussing how the initial exploration is linked to the subsequent behavior. In sections IV the learning strategy is explained in details and in section V we analyze the controller performances during and after training. Finally, in section VI we present our conclusions.

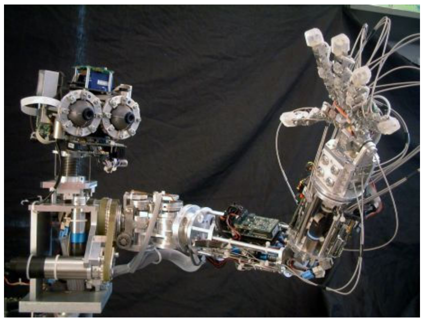

Fig. 1. The humanoid robot James. The control of its neck is the topic of this paper.

II. Robotic platform

The work described in this paper has been carried out on the head of the humanoid robot James [3]. James head has 7DOFs on the whole: two independent moving eyes (4 DOFs), one motor actuating the yaw movement (1 DOF, rotation around the axis perpendicular to the neck base), a particular structure constituting the neck (2 DOFs). The description of the peculiarities of such structure and the sensors used for its control are described in II-A and II-B respectively.

A. The neck design

The neck main body is a steel spring supporting the head. This spring can bend forward and laterally allowing pitch and roll rotations. These 2 DOFs are actuated by a system of three motors pulling three tendons that surround the spring 120° apart. The tendons are connected to the top of the neck (the top 'vertebra') on one side, and to capstans actuated by the motors on the other side. When the motors pull the tendons, forces are transmitted to the upper 'vertebra'. When these forces are unbalanced a bending moment is produced on the top of the spring, causing the movement of the head (see figure 2).

The structure of James neck recalls the design of a tendon-driven parallel manipulator (see [4], [5] for details). It has been shown in [6] that a tendon driven system with open-ended tendons requires more tendons than DOFs to be fully controllable. In particular, James neck is moved actuating three motors ($\mathbf{q} \in \mathbb{R}^3$) in order to perform the pitch and roll movements ($\mathbf{x} \in \mathbb{R}^2$). Mechanical constraints prevents the arbitrary choice of $\mathbf{q}$ for a desired $\mathbf{x}_d$. In particular, due to the elasticity of the structure, it exists a set of motor positions $\bar{\mathbf{q}} \subseteq \mathbb{R}^3$ for a given desired orientation $\mathbf{x}_d$. Among the possible $\mathbf{q} \in \bar{\mathbf{q}}$ there is an ideal motor configuration $\mathbf{q}^\star$ which generates an optimal value of stress of the three tendons, while achieving the main positioning task. Values of stress that are too small can send the tendons out of the capstans, while too large values can misalign the spring spirals or break the tendons.

B. Sensors and electronics

James head is equipped with 8 motors, 4 moving the two spherical eyes, one performing the yaw movement of the head and three actuating the neck. Vision is provided by two digital CCD cameras (PointGrey Dragonfly remote head), located in the eyeballs. A 3-axis orientation tracker (Intersense iCube2) has been mounted on top of the head, to emulate the vestibular system. This sensor is used as absolute position sensor, providing the actual pitch and roll position $\mathbf{x}$.

The neck is composed of three DC motors placed on the root of the robot. Each motor pulls one of the tendons actuating the structure, in order to perform the pitch and roll movements. Motors are controlled with a custom made control board which uses a programmable 16-bit DSP processor and is provided of a CAN-bus interface for communication. The actual position of the motors are provided by optical encoders mounted on the motor shaft.

A force sensor [2] measures the forces that the actuation system transmits to the neck through the tendons (see figure 3). This sensor is positioned on the upper 'vertebra' of the neck. For each tendon, force is measured indirectly by measuring the deformation of a cantilever beam structure; this quantity is proportional to the tendon tension, which is the measure we want to control ($F \in \mathbb{R}^3$). Semiconductor strain gages (SSGs) are employed in a Wheatstone bridge configuration. A mechanical stop limit is exploited to avoid the risk of damaging the sensing structure. A custom made board provides for the force sensor data acquisition. The board is based on a 16 bit DSP from Microchip (dsPIC30F4013) which samples up to a maximum of 6 analog channels for strain gauges sensors in a bridge configuration. In our specific case, the sampling rate is 1 kHz.

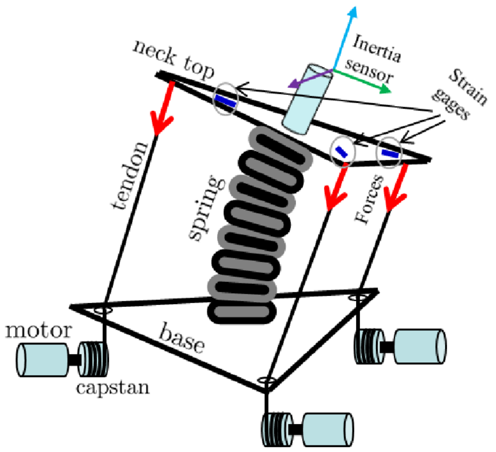

Fig. 2. Sketch of the neck actuation system and kinematics. Each motor pulls a tendon which passes trough a hole in the neck base. In this way the effective tendon length can be reduced to bend the spring in different directions. Three force sensors measure the tendon force using strain gages (SG).

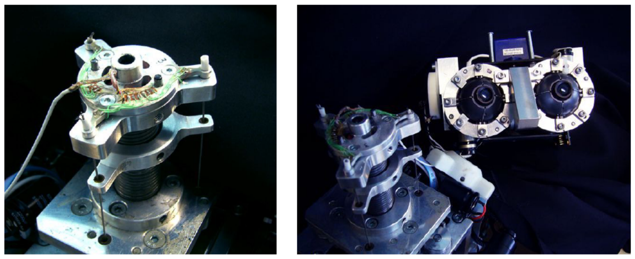

Fig. 3. Views of the neck force sensor from the top (left) and of the disassembled platform (right).

III. Control Law

Typically, learning is performed off-line. In this case training and 'exploitation' are separated phases. On the contrary, on-line learning approaches need these two phases to be somehow merged. Anyway, in both cases, an efficient strategy should be implemented to gather the necessary sensory data to train the system (*exploratory movements*, i.e. movements whose main aim is to explore the state space). In the on-line learning scenario we would like to integrate training and execution phases in a unique and continuous behavior. To achieve this, these initial exploratory movements should be driven by the same goal that will guide the 'normal' behavior after learning, employing the same rule to generate the movements. If we have a look at what happens in humans, converging evidences show that even the primitive neonatal behaviors are goal-directed actions rather than reflexes [7], [8].

In our implementation, we designed a control law which is used during the whole robot life, from the very beginning (*exploration*) to the end (*exploitation*). The performances of such a controller improve with time, as learning occurs.

We want to achieve the main task of canceling the difference between the head position $\mathbf{x}$ and the desired value $\mathbf{x_d}$. Furthermore, we want $\mathbf{x}$ trajectories to be linear in the operational space. As a secondary task we would like to regulate the tendon tension F by minimizing $F - F_d$ with $F_d = \frac{F_{max}+F_{min}}{2}$, calling F_{max} and F_{min} the maximum and minimum acceptable tendon force values. Practically, these two objectives can be obtained by solving the following optimization problem:

$$\min_{\mathbf{q}} \frac{1}{2} \|F(\mathbf{q}) - \mathbf{F_d}\|^2 \ \ s.t. \ \mathbf{x}(\mathbf{q}) - \mathbf{x_d} = \mathbf{0} \qquad \text{(III.1)}$$

Following the approach of [9], we can tackle (III.1) setting the desired motor velocities $\dot{\mathbf{q}} \in \mathbb{R}^3$ as follows:

$$\dot{\mathbf{q}} = C_1 + C_2 \qquad \text{(III.2)}$$

where

$$C_1 = J^{\dagger}(\mathbf{q})(\mathbf{x}_d - \mathbf{x}) \qquad \text{(III.3)}$$
$$C_2 = (I - J^{\dagger}(\mathbf{q})J(\mathbf{q}))KJ_F(\mathbf{q})^T(F_d - F) \qquad \text{(III.4)}$$

where $J(\mathbf{q}) \in \mathbb{R}^{2\times3}$ is the jacobian matrix which maps from motor velocities to task velocities, $J^{\dagger}(\mathbf{q}) \in \mathbb{R}^{3\times2}$ is its Moore-Penrose pseudo-inverse and $J_F(\mathbf{q}) \in \mathbb{R}^{3\times3}$ is the jacobian matrix which maps from motor velocities to tendons tension velocities (i.e. tendons tension variations). These matrices are the core of the learning controller; they are initialized with random values and then updated during the head movements, relying on sensory measurements coming from tendon tension sensor, absolute position sensor and motor encoders. The way in which these matrices are initialized and then updated is the subject of section IV. Then, $K \in \mathbb{R}^{3\times3}$ is a positive definite diagonal gain matrix, $I \in \mathbb{R}^{3\times3}$ is the identity matrix, $\mathbf{x_d} \in \mathbb{R}^{\mathbf{2}}$ is the vector of desired positions and $F_d \in \mathbb{R}^3$ is the vector of desired tendons tension.
The control law (III.2) is the so called resolved motion rate control technique (see [9] for details and proof of convergence) applied to our problem. $J^{\dagger}(\mathbf{q})$ allows to control neck orientation following straight trajectories ($\mathbf{x}$) in the task space. The operator $(I - J^{\dagger}(\mathbf{q})J(\mathbf{q}))$ projects the term which minimize the tension error in the null space of the primary task (i.e. the neck orientation).
If the measured tendons tensions are outside the admissible range $[F_{min}\ F_{max}]$, the (III.2) controller is switched off, and a 'safety controller' is activated:

$$\dot{\mathbf{q}} = -G \cdot F_{err} = C_{safe} \qquad \text{(III.5)}$$

$$F_{err} = \begin{cases} (F - F_{max}) \text{ if } F > F_{max} \\ (F - F_{min}) \text{ if } F < F_{min} \end{cases} \qquad \text{(III.6)}$$

where $G \in \mathbb{R}^{3\times3}$ is a positive definite diagonal gain matrix. As soon as the measured tendon tensions have come back within the limits, the main controller (III.2) is switched on again. This 'safety controller' has the unique task of regulating the tendons tensions to the desired values, but of course it interferes with the main neck orientation task. We will show in section V how the activation of this controller is frequent in the beginning of learning and more and more absent in the later stages.

IV. Learning strategy

As previously stated, movements are generated by the (III.2) controller, in combination with the safe controller (III.5) when measured tendons tensions exceed the limits. Target orientations $\mathbf{x}_d$ are provided to the robot every 20 seconds, choosen randomly within the (safe) physical limits $[-35°\ 35°]$, with uniform distribution. During the motion data are gathered from absolute position sensor, $\mathbf{x} \in \mathbb{R}^2$, force sensor (tendons tensions), $F \in \mathbb{R}^3$, and motor encoders, $\mathbf{q} \in \mathbb{R}^3$. What is needed for learning are little variations of these quantities (displacements): $\Delta\mathbf{x}$, ΔF and $\Delta\mathbf{q}$. The time window on which these variations are computed is 50 ms, while the controller rate is 5 ms (200 Hz). Here follows the description of how $J(\mathbf{q})$ is learned from these sensory measurements; the same strategy applies to the learning of $J_F(\mathbf{q})$, and could be generalized to any other non-linear matrix.
From a set of couples $(\Delta\mathbf{x}, \Delta\mathbf{q})$ we can estimate a local Jacobian matrix $\hat{J} : \Delta\mathbf{x} = \hat{J}\Delta\mathbf{q}$ with Least Squares (LS) Regression. This $\hat{J}$ is an approximation of $J(\mathbf{q}) : \dot{\mathbf{x}} = J(\mathbf{q})\dot{\mathbf{q}}$ in a local region of the $\mathbf{q}$ space, whose accuracy depends basically on the size and distribution of the aforementioned set. In our implementation this estimation is performed on-line, with an incremental LS algorithm adapted from [10]. Then, since we need to build a global Jacobian matrix $J(\mathbf{q})$ starting from local models $\hat{J}$, we employed a Receptive Field Neural Network (RFWR [11], an on-line machine-learning tool) to map from $\mathbf{q}$ to the corresponding local Jacobian matrix $\hat{J}$. Every time a new couple $(\Delta\mathbf{x}, \Delta\mathbf{q})$ is computed (every 200 ms) the RFWR net is queried with the current motor configuration $\mathbf{q}$ to obtain the correspondent local model $\hat{J}$. The obtained constant matrix is updated with the couple $(\Delta\mathbf{x}, \Delta\mathbf{q})$ using incremental LS. The RFWR net is then trained with the current motor configuration $\mathbf{q}$ as input and the updated constant Jacobian matrix $\hat{J}$ as output. The algorithm steps are summarized below:

1. **collect new sample** $(\Delta\mathbf{x}, \Delta\mathbf{q})_i$ at time i
2. **retrieve** $\hat{J}_{\mathbf{q}_i} = RFWR(\mathbf{q}_i)$
3. **update** $\hat{J}_{\mathbf{q}_i}$ with $(\Delta\mathbf{x}, \Delta\mathbf{q})_i \rightarrow \hat{J}^{up}_{\mathbf{q}_i}$
4. **train** $RFWR(\mathbf{q}_i, \hat{J}^{up}_{\mathbf{q}_i})$

Since the incremental LS algorithm requires a non-null matrix to update, the RFWR net is initialized with an arbitrary $\hat{J}$, which becomes the output for every configuration $\mathbf{q}$ received as input (in the beginning, when the net is still empty). The same holds for $\hat{J}_F$.

V. Results

In this section the performances of the learning controller are evaluated and discussed. The discriminants are the steady-state orientation error (pitch and roll position errors) and the slope of its convergence to zero, the linearity of the trajectories in the operational (pitch/roll) space[1], the amount of tension measured on the tendons. In all the tests reported

[1]Trajectories would be perfectly linear if the Jacobian $J(\mathbf{q})$ were exactly known.

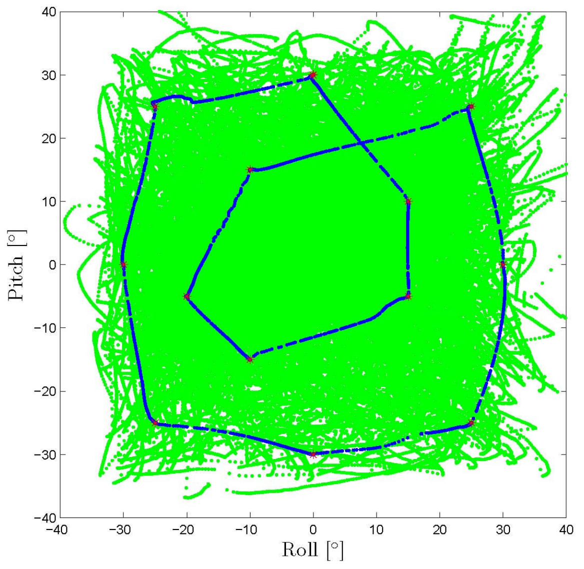

Fig. 4. Task space trajectories (blue line) connecting the desired via points (in red). The green dots are the points spanned during learning. This performance was achieved after 2 hours and 20 minutes of training. The linearity of the trajectories proves the convergence of the estimated Jacobian.

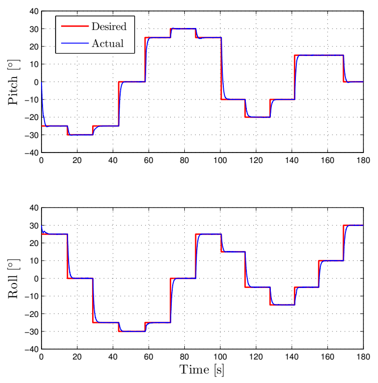

Fig. 5. Roll and pitch step response. Desired and actual positions relative to the trajectories of figure 4 are shown.

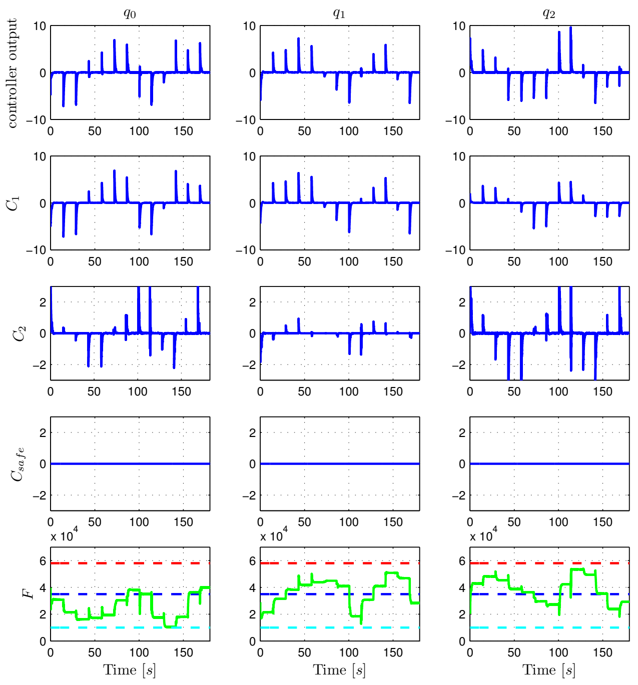

Fig. 6. Activations of the different controllers and tension measurements, for every q. On the first row the overall controller output, on the second row the C_1 component. The third row shows the contribution of the controller C_2. When $\mathbf{x} - \mathbf{x_d} = \mathbf{0}$, The contribution of C_2 converges to zero, which proves that the first order conditions of the secondary task are satisfied. On the fourth row the 'safety controller' activation is shown, while on the fifth row tension measurements are reported. The effect of the secondary task controller C_2, guarantees that tendon tension are within the limits $F_{min} < F < F_{max}$.

hereinafter, the same sequence of 13 target positions $\mathbf{x_d}$ is provided to the robot, one different position every 20 seconds: these positions have been chosen arbitrarily to cover homogeneously the task (roll/pitch) space. As previously explained, the movements during which training data are gathered and on-line learning is performed are generated providing random $\mathbf{x}_d$ to the robot.

The first set of graphs (figures 4, 5 and 6) shows the best performances we achieved on the system, after about 2 hours and 20 minutes of training (170000 samples of the form $(\Delta\mathbf{q}, \Delta\mathbf{x}, \Delta F)$ gathered).

The steady-state RMSE (Root Mean Squared Error) computed over the sequence of movements is $0.020°$ for the roll rotation and $0.018°$ for the pitch rotation. Figure 5 also shows the exponential convergence of the position error to zero. Furthermore, the tendons tensions are kept within the limits by the controller C_2, which acts in the null-space of the primary task.

To underline the importance of the controller C_2 (III.4), we tested the system controlling just the primary task, setting $C_2 = 0$. Figures 7, 8 and 9 describe the behavior of the system using such a controller. We can notice from figure 9 that without the contribution of C_2 the 'safety controller' C_{safe} is often active; the activation of this controller affects the performances of the main controller, sometimes preventing the system from cancelling the position error. The result is that with such a controller the system is not able to consistently cancel the position error, nor to regulate tendons tensions.

Figures 10, 11 and 12 show the improvements of the controller during learning. In the left figures the system has been trained with just 20000 samples (about 15 minutes), and it is clear that the performances are far from being acceptable. We will not quantify the errors in this case, since they are too big. In the middle figures, on the contrary, the system is already able to reach the desired orientations $\mathbf{x}_d$, but the error convergence is not always perfect and the task space trajectories are not so linear. The steady-state RMSE computed over the sequence of movement is $0.064°$ for the roll rotation and $0.020°$ for the pitch rotation. Furthermore, the tendons tensions are not regulated properly:

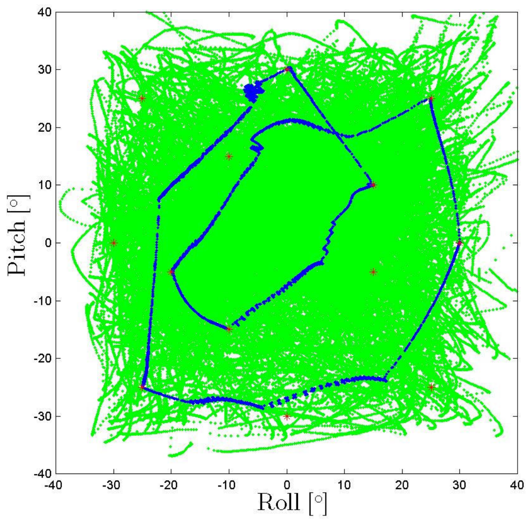

Fig. 7. Task space trajectories without tendon tensions control. With the only controller C_1 tendon tension arise over the force limits, thus activating the controller C_{safe}. In this way, the convergence to the target positions $\mathbf{x_d}$ is not guaranteed.

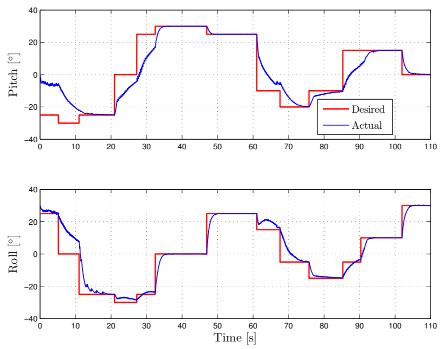

Fig. 8. Desired and actual roll and pitch positions without the activation of tendon tensions error projection in the null space of $J((\mathbf{q}))$ (the controller C_2 is not active). The convergence to the target positions is not guaranteed, because of the activation of the controller C_{safe}.

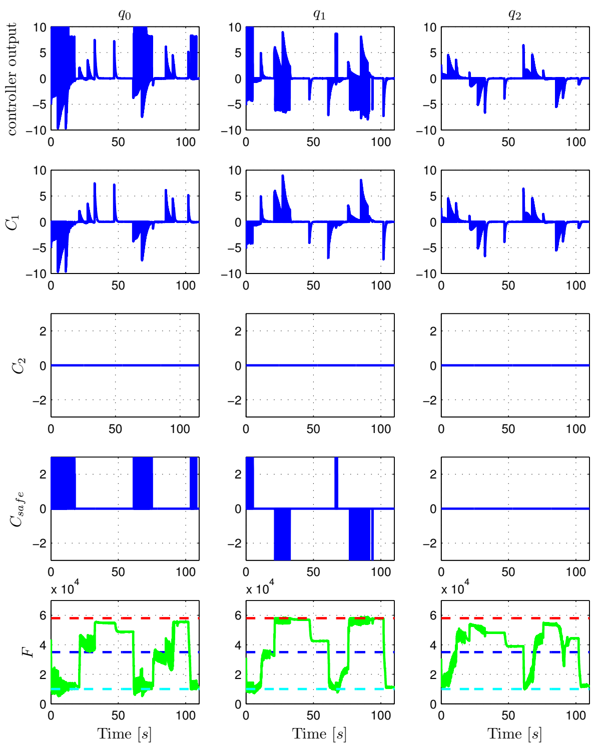

Fig. 9. Activations of the different controllers and tension measurements, for every q, without tendon tensions control. Controller C_{safe} is often active, because the controller of the secondary task is not present.

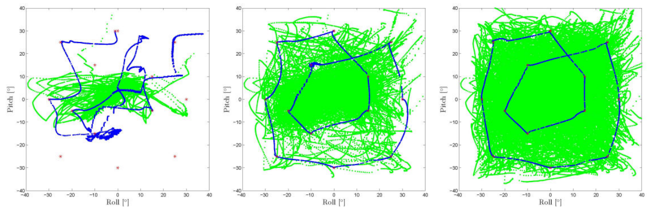

Fig. 10. Task space trajectories at three progressive learning stages (20000, 100000, 170000 training samples, from left to right). Left figure show that most of the points are not reached because the Jacobian estimation is not accurate enough. Central figure show that the estimation of the Jacobians is converging. Right figure show that the Jacobian evaluation has converged to a good estimation of the real Jacobian, which results in linear trajectories in the Cartesian space (see figure 4).

the 'safety controller' is activated twice, and there are some high-frequency oscillations. In the right figures the results obtained with the best controller (the one already discussed in the beginning of the section) are reported.

Finally, something can be added concerning the mere task of estimating $J(\mathbf{q})$ and $J_F(\mathbf{q})$.

Figure 13 shows the trend of the coefficients of $J(\mathbf{q})$, where $\mathbf{q} = 0$. Remarkably, these coefficients, initialized arbitrarily as stated in section IV, seem to converge to specific values ($J(0) = [0.2; -0.1; -0.1; 0.0; 0.2; -0.2]$). Furthermore, these values turn out to be reasonable if we analyze the structure of the neck and the considered motor configuration ($\mathbf{q} = 0$, which means also $\mathbf{x} = 0$). When the head is straight (i.e. $\mathbf{x} = 0$), to bend the neck forward the system should shorten the front tendon of a certain length (let's say L_1) and lengthen the two tendons in the back of half that length ($L_1 \cdot \sin(120°)$, being the three tendons separated 120° apart). This is consistent with the first row of $J(0)$, $[0.2; -0.1; -0.1]$. On the other hand, to bend the neck laterally the front tendon do not play any role, while the other two tendons must be displaced of the same quantity, but in opposite direction; this is consistent with the second row of $J(0)$, $[0.0; 0.2; -0.2]$. For a more precise description about this geometric model refer to [2].

Regarding $J_F(\mathbf{q})$, some additional experiments have been performed in order to test the quality of its estimation.

First the system has been driven to a desired position $\mathbf{x}_d$ using the controller C_1 (III.3) alone; the absence of the

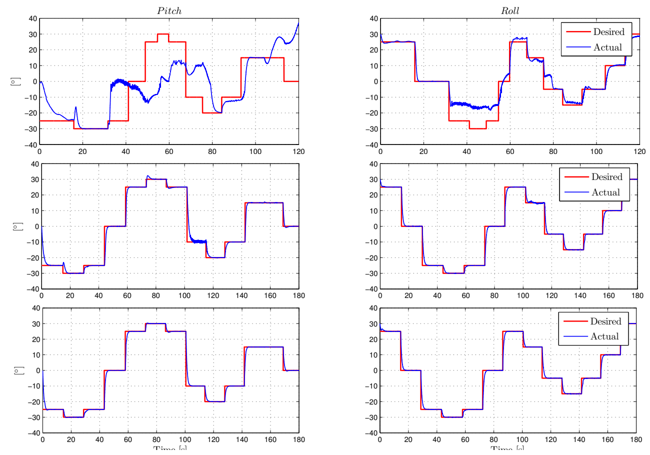

Fig. 11. Roll and pitch desired and actual positions at three progressive learning stages (20000, 100000, 170000 training samples, from top to bottom). Top figures show that most of the target positions are not reached because the Jacobian estimation is not accurate enough. Central figures show that the estimation of the Jacobians is converging. Steady state errors are limited. Bottom figures show that the Jacobian evaluation has converged to a good estimation of the real Jacobian. Here target positions $\mathbf{x_d}$ are reached.

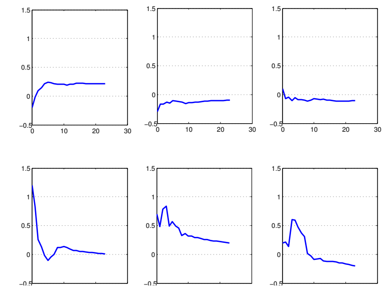

Fig. 13. Trend of the coefficients of $J(\mathbf{q})$ in $\mathbf{q} = 0$ during learning. They converge to particular values that turned out to be quite reasonable.

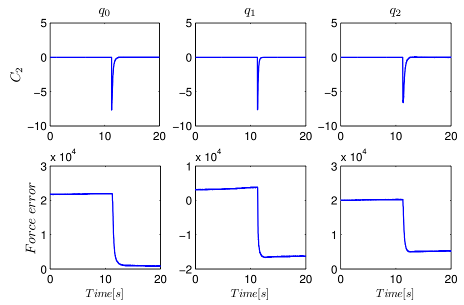

Fig. 14. Top figures: contribution of controller C_2 for every joint. Bottom figures: force errors $F - F_d$ on every tendon.

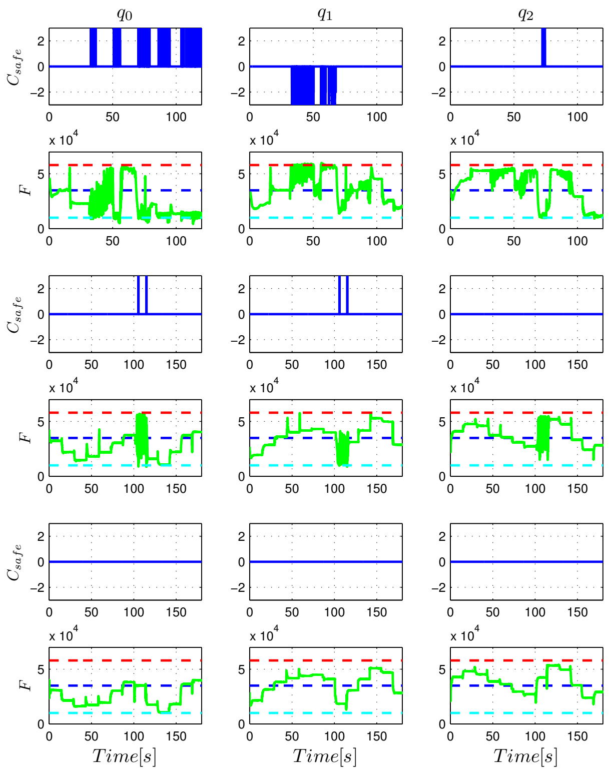

Fig. 12. Activations of the 'safety controller' and measured tendons tensions at three progressive learning stages (20000, 100000, 170000 training samples, from top to bottom). The first two rows show that when learned Jacobian is not accurate, forces arise over the limits, thus activating the 'safety controller' C_{safe}. The more learning improves the estimation, the more the 'safety controller' becomes unnecessary.

C_2 contribution causes the raising of the tendon tensions and consequently of the force error $F - F_d$. Then the controller C_2 was activated, with the effect of a sudden reduction of the norm of the force error, as shown in figure 15. Since $C_2 = 0$ at steady-state, the system is in a configuration which satisfies the first order condition of the primary problem (III.1). Figures 14 and 15 show that when C_2 is active the error $F(\mathbf{q}) - F_d$ decreases, while maintaining the task $\mathbf{x} - \mathbf{x}_d = 0$. This means that J_F^T is a good approximation of the real Jacobian $\frac{\partial F^T}{\partial \mathbf{q}}$, or, more precisely, that $J_F^T(F(\mathbf{q}) - F_d)$ is a descent direction for the secondary function $\|F(\mathbf{q}) - F_d\|$.

In an additional experiment, the system is driven toward an arbitrary $\mathbf{x}_d$ using the controller (III.2). When at steady-state, the desired tendon force F_d is changed periodically. The experiment is shown in figures 16 and 17. As expected, steps of the desired tendon force result in steps of the actual forces in the same direction. Of course, since the desired force is the same for all the tendons, and the primary task is the position control of the head (see figure 17), steady state errors are present in the response of the system. This experiment demonstrates that the controller C_2 is able to regulate tendon tensions following F_d, without interfering with the main orientation task.

VI. Conclusions

This paper proposes a machine-learning approach for the control of a human-like robotic neck, without any need of a-priori modeling. The neck mechanical structure is innovative and highly biologically inspired. The complexity of the system makes it difficult to build an accurate kinematic

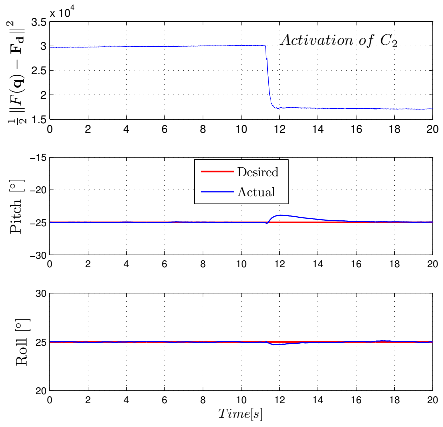

Fig. 15. Top figure: norm of the force errors. Bottom figures: pitch and roll desired and actual positions. The activation of the controller C_2 reduces the norm of the force errors, without affecting the main positioning task, at steady-state.

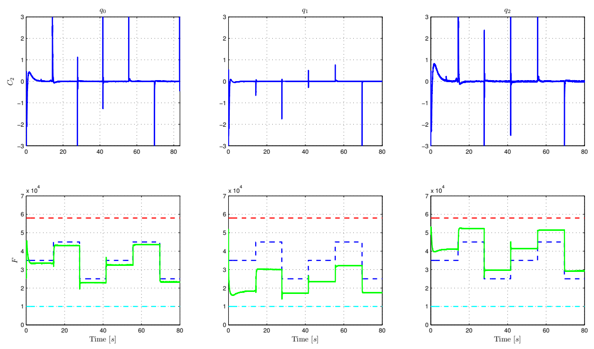

Fig. 16. Top figures: contribution of controller C_2 for every joint. Bottom figures: actual (green solid line) and desired (blue dashed line) forces on every tendon. Step variations of the desired tendon forces are given to evaluate the performances of the controller C_2.

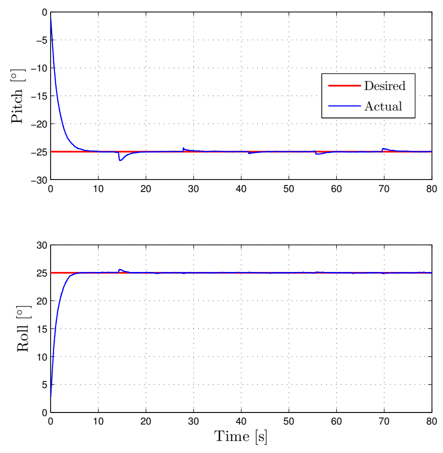

Fig. 17. Desired and actual pitch and roll positions. The step variations of desired forces does not influence the pitch and roll positions at steady-state.

model for its control. Conversely, the presence of a rich sensory system and the peculiarity of the structure has suggested the implementation of a controller based on on-line sensori-motor learning. Data gathered from absolute position sensor, tendon tension sensors and motor encoders have been used to learn a model of the system Jacobian, $J(\mathbf{q})$, useful to precisely control the neck in the operational space (pitch and roll rotations) generating linear trajectories; furthermore, a tensions-motors Jacobian, $J_F(\mathbf{q})$ has been learned in the same way, and employed to bound the tendons tensions by acting in the null-space of the system Jacobian. Learning occurs on-line, in a continuous and autonomous way, without any separation between the training phase and the execution phase. The controller performances have been analyzed on the basis of the steady-state orientation error, the profile of error convergence, the linearity of the operational space trajectories and the regulation of the tendons tensions. Results show that after some training (controlled movements toward randomly distributed targets $\mathbf{x}_d$) good estimations of $J(\mathbf{q})$ and $J_F(\mathbf{q})$ can be learned and efficiently used for control. The proposed controller allows to achieve a steady-state RMSE of $0.020°$ for the roll rotation and $0.018°$ for the pitch rotation, with exponential error convergence; trajectories in the task (roll/pitch) space are almost linear. Moreover, tendons tensions are kept bounded within the required limits.

VII. Acknowledgements

The authors were supported by the RobotCub project (IST-2004-004370) and by FP7 project CHRIS (FP7-IST-215805).

References

[1] F. Nori, L. Jamone, G. Metta, and G. Sandini, "Accurate control of a human-like tendon driven neck," in *International Conference on Humanoid Robots*, Pittsburgh, USA, 2007.

[2] M. Fumagalli, L. Jamone, G. Metta, L. Natale, F. Nori, A. Parmiggiani, M. Randazzo, and G. Sandini, "A force sensor for the control of a human-like tendon driven neck," in *International Conference on Humanoid Robots*, Paris, France, 2009.

[3] L. Jamone, F. Nori, G. Metta, and G. Sandini, "James: A humanoid robot acting over an unstructured world," in *International Conference on Humanoid Robots*, Genova, Italy, 2006.

[4] R. Verhoeven and M. Hiller, "Estimating the controllable workspace of tendon-based stewart platforms," in *7th International Symposium on Advances in Robot Kinematics (ARK)*, Portoroz, Slovenia, 2000, p. 277284.

[5] A. Hay and J. Snyman, "Optimization of a planar tendon-driven parallel manipulator for a maximal dextrous workspace," vol. 37(3), pp. 217 – 236, April 2005.

[6] L. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Wiley-Interscience, 1999.

[7] A. van der Meer, "Keeping the arm in the limelight: advanced visual control of arm movements in neonates," *European Journal of Paediatric Neurology*, no. 4, p. 103108, 1997.

[8] C. von Hofsten, "Eye-hand coordination in newborns," *Developmental Psychology*, no. 18, pp. 450–461, 1982.

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

[10] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1994.

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