The Neck of Pinobo, a Low-cost Compliant Robot

Arnaud Blanchard, Djamel Mebarki

The Neck of Pinobo, a Low-cost Bio-inspired Robot

Arnaud Blanchard and Djamel Mebarki

ETIS Lab., Cergy-Pontoise, 95000 France
`<firstname>.<lastname>@ensea.fr`,
`http://www-etis.ensea.fr`

Abstract. We introduce in this paper a robotic neck with 3 degrees of freedom. The Pinobos neck has three important features: open source software, solid, and easy to made. This early prototype is light, fast and compliant. After explaining how to build the Pinobos neck, we analyze the different properties of this system in order to obtain a robust controller. Pinobo is equipped with several sensors such as accelerometers, gyrometers and voltage of its motors. The robot is designed for students and researchers for scanning environment, stabilizing a head or interacting with humans.

Introduction

The Interest of a Minimalist Approach

A robotic neck has several advantages. First of all, it allows any sensors such as camera, microphone or even artificial nose to scan the environment. Usually, this is made through pan tilt supports (yaw and pitch movements). Secondly, it allows the stabilization of these sensors even when the support is moving — typically on the top of a rolling or walking robot. Thirdly, movement of the head improves human-robot interaction by helping humans to perceive what robots pay attention to. For example, a robot can express facial expressions toward an object to share emotional states [1]. A moving neck gives the robot a lifelike behavior.

Several contributions have been done in this area with high performances. However, they are often complex or fragile [2],[3],[4],[5] and some are still in simulation [6],[7],[8].

The consequences of this complexity are that researchers and students avoid to working directly on robots. It is not a problem of financial cost but of time, personal resources to maintain the platforms and risks for the platform and the user. In this context, the simulation allows the development of efficient algorithms (optimization) by testing a huge number of times the simulated robot in different environments. Afterward, the optimized algorithm can be successfully applied to a real robot.

However fitting simulations with real robots implies very accurate simulators and very precise realization of the mechanical parts of the robots. Otherwise,

even though the simulator is accurate it would be based on a slightly different robot than the reality. Moreover, to keep the simulation valid, the environment must be perfectly modeled and therefore it constraints the place where the robot can evolve. An important consequence is that the robots are used as illustrations or proofs of concept but rarely for an everyday tool of research or a final usable industrial product.

In this article, we present a robotic neck which can be used and modified in everyday research work by students and researchers. The robotic platform must be affordable, easy to use and maintain. A minimalist approach is used to provide useful features. The goal is to provide open source hardware and software in order to spread the approach (see also [9],[10],[3]).

Features for Scanning, Stabilizing and Interacting

In the introduction we claim that a robotic neck has three main advantages:

Scanning The neck must allow moving sensors in order to scan the environment. This is one of the first use of a robotic neck in robotics as it was used to orient a sonar, a laser, a camera, a microphone... This implies that the robot must be able to move fast in yaw and pitch directions it also has to be quiet if we want to use a microphone.

Stabilizing The neck can be used to compensate the movement of the body.

Interacting The neck must improve interaction with humans, the control must be smooth and reproduce a natural dynamic. The way the robot is perceived and its own dynamics should look lifelike [11]. We consider that several degrees of freedom and a structure close to humans can improve the interaction. Moreover, the compliance is an important property allowing to interact physically with the robot.

Moreover if possible (this is a loose constraint as we want to keep a minimalist approach) we would like to approach a model of the real neck. This would allow to improve human likeness and to model bio-mechanic phenomenon.

Bio-Inspiration

Even though bio-inspiration is not the main constraint, we study the bio-mechanics of the neck. We can notice that human neck has complex distributed joints spread over seven vertebrae but that the first two (C1 and C2) are very special. They are the main actors of the head's rotations. C1 implements a ball joint with the skull and C2 implements a pivot joint with C1 [12]. We neglect for now the other vertebrae (see Fig. 1).

There are eighteen groups of muscles in the neck with very complex configurations. However, we do not need to manage the jaw, the esophagus or the trachea. We consider only the 3 degrees of freedom of the first two vertebrae in our model. The simplest way (in term of mechanics) to move a device is to use a motor directly with a cable and a pulley. This avoids gears, rods, or other costly
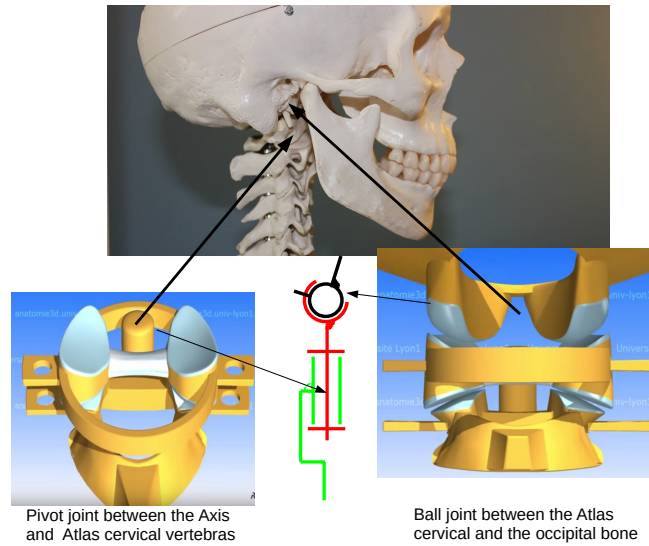
Fig. 1. Scheme of the two first vertebrae of the neck. 3D models extracted from the online video [13] with the authorization of the author.

material and this is also quiet allowing to place the motors away from the joints. The effect of a motor with a cable and a pulley is quite similar to a muscle and we use it. Some robots mimic biology in a more realistic way [14],[10] but they are too complex for this study.

We follow the kinematics of these vertebrae as reviewed in [12].

Hardware

Mechanics

Important issues for robustness are the quality of the joints. Low quality of joints leads to fast deterioration and mechanical backlash of any robot. We use simple but high-quality gimball and ball bearing. Serial structures are simpler to control but have several flaws such as the accumulation of backlash and the fact that the first joints have to carry all the weight including the motors [6],[8].

We propose a 3 degrees of freedom joint pulled by cables with pulleys and motors. In theory we only need 3 motors and a pulling spring because the movement of the cable cannot be reversible (see an example [15]). We do not use a pulling spring but 4 motors which over actuate the system. The redundancy can be used to set the stiffness. Another advantage is that for each motion, two motors are working at the same time which divides the required power. The fact that we do not use servo-motors but simple DC motors improves the reversibility

and helps us to get access to low level control which is important for learning algorithms. It also increases power for the same size and cost. Moreover it is more robust as we avoid problems with the reducer. The reduction is done by using small pulleys attaching cables far from the center of rotation. We optimize the phenomenon by maximizing the torque when the head reaches extreme positions (see Fig. 2).
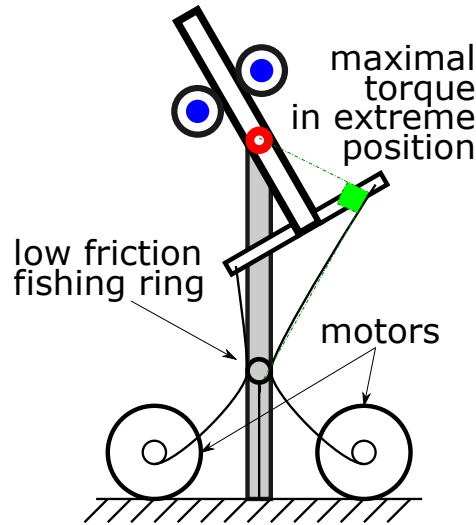

Fig. 2. Attachment of the cables

Motors' Driver

In order to drive the speed or force of these motors we need an electric chopper. We chose a Grove I2C motor driver which is simple and robust. However the chopping frequency is low (30Hz) thus we add a 4700F capacitor on each motor in order to smooth the signal. This value is a compromise between a too low value which does not filter enough and a high one which makes the driver heat. We also add a radiator on the circuit to cool it down. In order to access to the compliance, we need to detect the current passing through the motors. This can be indirectly done by accessing the voltage on the motor — the motor driver as an internal resistance, if the intensity increases, the voltage decreases. We use an analog to digital converter (MCP3800) to access to these values. We use four classical 4 euros DC motors (6V, 1.34A, 46.2gcm and 6 effective watts). Due to the limits of the driver it is not useful to use bigger motors. We started with smaller motors (1.5 euros), it was working but we were most of the time working outside the limits of these motors.

Sensors

The minimum requirement to control the head is to know its position all the time. To keep the system simple we avoid to use coder on each motor or joint, therefore we use the values of an accelerometer giving the pitch and roll positions. Trough this sensor we cannot get the yaw position as the gravity does not change when we are turning the head on the vertical axis. Thus we include a potentiometer—implanted inside the wood structure—on the axis of the vertical pivot joint and access the voltage through another analog to digital converter.

You can get a picture of the device at a very prototypical stage Fig. 3, the electric scheme appears Fig. 4. The typical power is 5 volts, but the drivers of the motor can be powered by a tension of 10 volts if you want to increase force and speed. The design is open source (Creative Commons Attribution-ShareAlike 4.0 International).

Fig. 3. Prototype of the neck.

For the software, we use the open source Basic Libraries And Applications for Robotics [16] and we publish the specific code in [17].
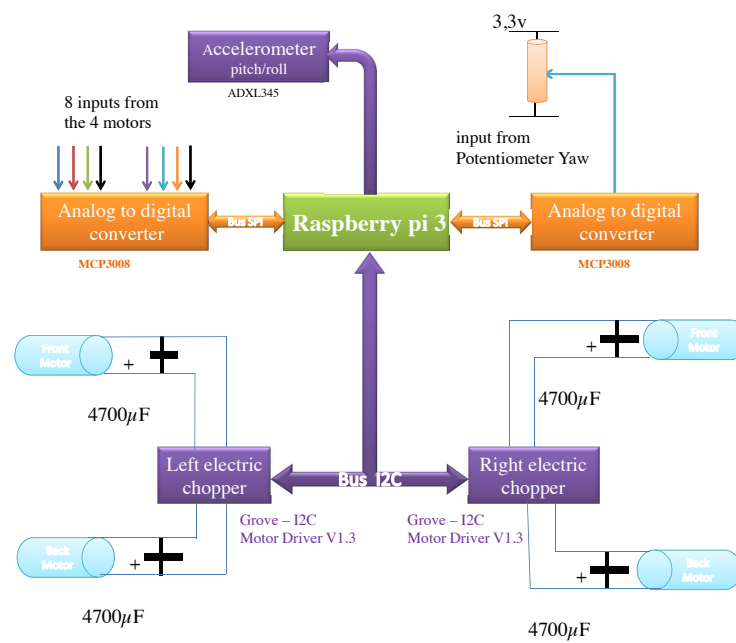
Fig. 4. Full electrical scheme of the robot. The input power is typically 5V.

The principle is simple: if you need a pitch movement toward the front, turn on the two front motors. For a roll movement on the left, turn on the two motors on the left and for a yaw movement, turn on the two diagonal motors. For each axis, turn on the two opposite motors to get an opposite movement. You can combine any movement by summing the activation on the motors. The result is not linear but we plan anyway to use learning systems to learn the non-linearity.

Results

Dynamics of the Motors and Drivers Now that we have a mechanical neck, we want to analyze its properties. This is fundamental as one of our main target are students and researchers. Even if the mechanics is not optimal we need to know the limits and potentials. First of all, we analyze the driver to check if we are able to have the quality we need (i.e. speed, force control and compliance or reversibility). We send the command to the motors at different force (25%, 50%, 75%, 100%). We check if the voltage varies properly and if we can detect the variation of force on the motor. We run two tests each time one with the motor running freely and one with the motor externally forced to stop. You can see the results Fig. 5.
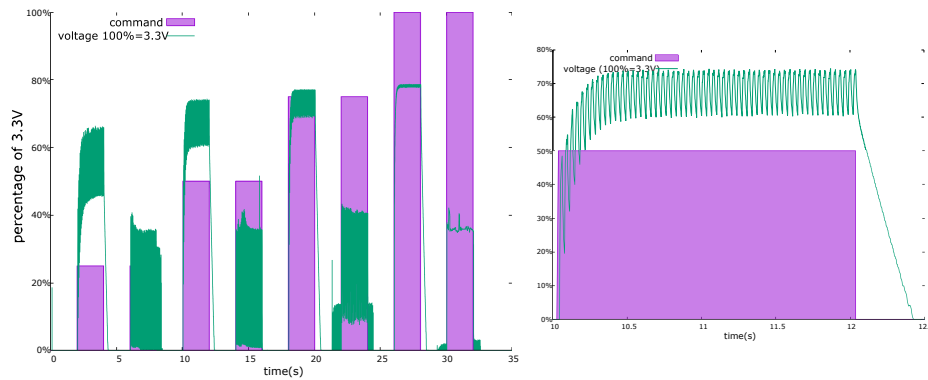

Fig. 5. Voltage applied on the motor depending on the command. In the second try of each set of twice the same command, the motor is externally blocked. This demonstrates the effect of the consumption of the motor on the voltage. The right side of the figure shows a zoom presenting the effect of the chopper and the capacitor on the voltage applied to the motor. We can notice that the delay between the command and the effect on the voltage is short (1ms).

We can definitely change the force and the speed of the motor. You can notice the defect of the driver chopping at 30Hz in the right graph.

Compliance

As we mention earlier, it is important for us to make the robot compliant. To do so we avoid gearboxes to prevent any passive frictions. This reduces complexity, cost and size but the main advantage is that an effect on the head induces a direct effect on the motors. We check how sensitive the detection of external forces on the motors can be. To apply a controlled and variant force on a cable, we use a second motor pulling the cable at different voltages with the following protocol. First, we wound the cable on the first motor (the one we want to test). Then a second motor winds the cable at different speeds. We repeat this test for 9 different speeds. You can see the results Fig. 6. We observe tensions on the passive motor due to the fact that the second motor makes it move. We can check that the speed increases at each try because the time to fully wind the cord decreases. The acquisition is achieved by the system itself through the circuit MCP3008.
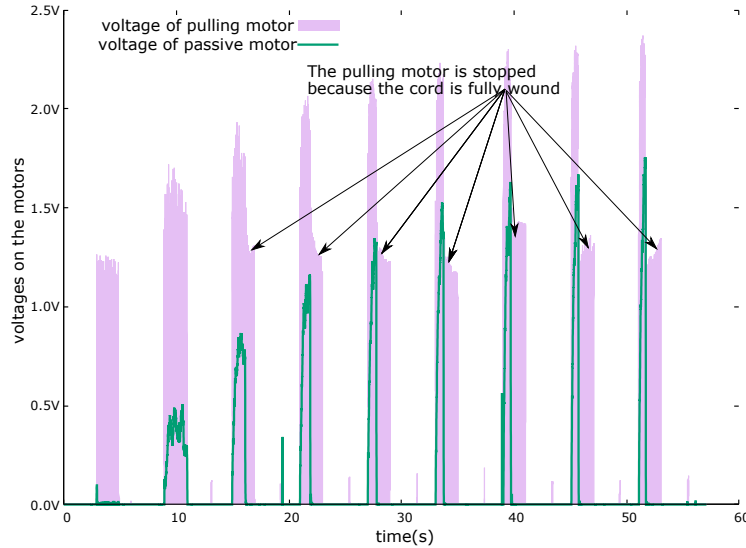

Fig. 6. A pulling motor wound the cable from the passive motor at different speeds. We can detect that the passive motor is moved because of the tension that it generates. Note that between each try, the passive motor actively rewinds the cable. It does not appear here because it is a negative voltage.

Actual Movements

We have shown the property of an individual motor, now we validate the overall behavior of the system. We first test a single axis. The pitch and roll movements

are blocked (we block the gimbals in fact) and we the yaw rotation is activated. To do so we activate either the front left motor with the back right one or the front right motor with the back left one for the opposite direction. We present Fig. 7 the resulting dynamics of a sequence of commands.
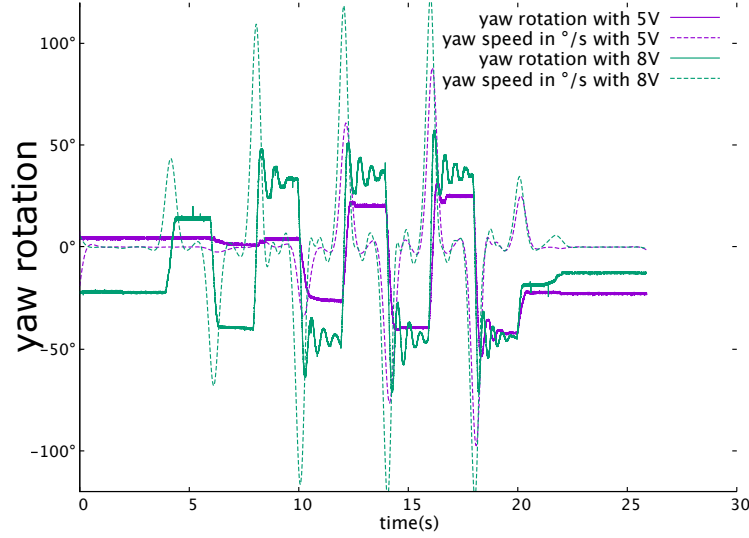

Fig. 7. Yaw rotation when the power is 5 or 8 volts. We can see that the range of movement is over +/- 40 degrees and the maximal speed is over 100 degrees/s. Oscillations are due to the fact we send a square signal. This reveals the mechanical properties as the controller is only proportional. The software part will be elaborated in the future.

Controller

This article is mainly focused on the hardware part of the system because we want to use it as a tool to test and implement control and learning algorithms. However, in order to show that this neck can be used and controlled, we implement simple movements. We apply a proportional controller on smoothed data from the normalized sensors. It is especially important to smooth the data from the accelerometers as they are sensitive to noise and vibrations. A low pass filter is applied with ϵ as the parameter of the filtering for each axis. See Eq. 1.

$$\begin{cases} \overline{yaw} = & \overline{yaw} + \epsilon_y.(\overline{yaw} - yaw\_sensor), \epsilon_y = 0.9 \\ \overline{pitch} = & \overline{pitch} + \epsilon_p.(\overline{pitch} - pitch\_accelerometer), \epsilon_p = 0.075 \\ \overline{roll} = & \overline{roll} + \epsilon_r.(\overline{roll} - roll\_accelerometer), \epsilon_r = 0.075 \end{cases} \qquad (1)$$

The differences between the commands and the smoothed sensors are amplified and used to activate the corresponding motors. However, as we explained, for each axis two motors are pulling which may lead to contradictory pulling. Therefore we subtract the activity of the second weakest motor to the activity of the others motors. As a result, only the two most activated motors pull the cables. On our raspberry pi 3, the algorithm was working at a refresh rate of 1kHz. The gains from yaw, pitch, and roll axes are respectively 10, 8 and 8.

We test this algorithm successfully for different tasks (driving with the keyboard, using a sequence, stabilizing a change in orientation ...). You can see the videos in [17].

While we send a sequence of commands, we record the result that you can see Fig. 8 you can download the data (100Hz resolution) from [17].

The results are not optimal because we need to use a better controller with at least an integration of the errors in order to reduce static errors. There are many other ways to improve the control [7], [18] or see [8] for a review. Here, however, we focus on the hardware.
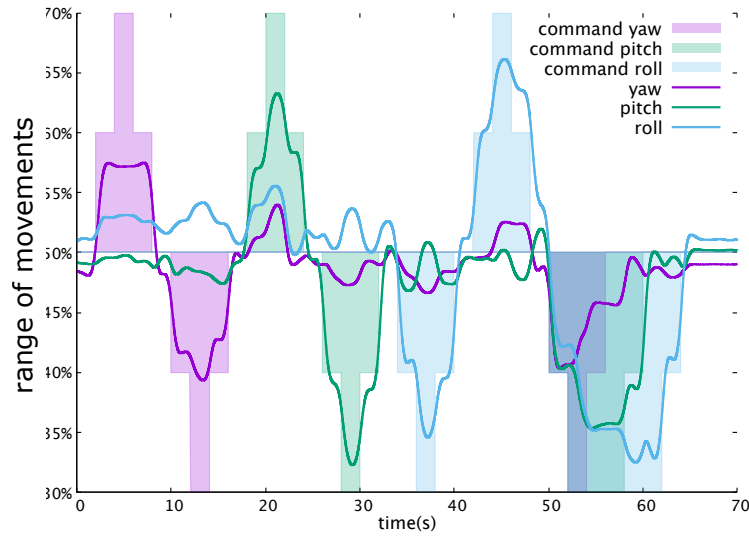

Fig. 8. Real normalized movements are compared to the commands. After 50 seconds, we combine the commands. For the readability of the graph, the values of the sensors have been smoothed. The extreme commands are never reached because we need a better controller to at least integrate the errors.

Another important feature we have tested (see in the videos [17]) is the compliance. Actually, the neck of Pinobo is very compliant and resilient. It does not resist much as the motors are small but it quickly comes back to its desired position after perturbation. It looks like it is on a spring but there is no spring at

all and the cable are very stiff. All this compliance comes from the reversibility of the motors.

Conclusion

We have introduced a prototypical robotic neck with modest properties but strong potential. It is easy to build (few hours), cheap (about 100 euros) and more importantly: it is robust. It is easy to add or try different elements. The motor drivers are not efficient (heating, chopping at only 30Hz) but simple to use. We were satisfied of their robustness and solidity (we will be still replaced by more efficient systems). The cables as well are reliable and we did not need to change one despite the fact we did some bad and difficult tests.

We have few intermediate elements to maximize the understanding and control of the overall system. No servo means no external controller. No spring and no gearbox, means a full control of the mechanics. The neck also includes many sensors like the voltage of the motors (and indirectly the intensity), accelerometers, gyrometers, and position of the yaw rotation.

This, combined with the fact that we use simple libraries [16] with the same philosophy (direct control, few abstractions), makes Pinobo an ideal platform for testing and prototyping new ideas.

In the future, the first thing we will do is to improve the motor drivers. We will apply learning algorithms to make the robot able to predict its dynamic and improve its control. Then we will add eyes with embedded cameras and ears with microphones. We may develop several versions improving either speed, life-likeness, force, or amplitude of movements depending on the need of each studied task. Knowing the environment and problems encountered in the field of robotics, which often exceed the skills of researchers (especially due to logistics, mechanical problems, constraints of physical or logical interfaces), the results we have obtained, the number of available inputs and, the simplicity of the interfaces (electronic and software), we hope to bring an asset to students and researchers in the field of humanoid robots to focus on their research.

Acknowledgment

References

1. Boucenna, S., Gaussier, P., Hafemeister, L.: Development of first social referencing skills: Emotional interaction as a way to regulate robot behavior. Autonomous Mental Development, IEEE Transactions on **6** (03 2014) 42–55
2. Alfayad, S., El Asswad, M., Abdellatif, A., Ouezdou, F.B., Blanchard, A., Beaussé, N., Gaussier, P.: Hydroïd humanoid robot head with perception and emotion capabilities: Modeling, design, and experimental results. Frontiers in Robotics and AI **3** (2016) 15

3. Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Rosander, K., Lopes, M., Santos-Victor, J., et al.: The icub humanoid robot: An open-systems platform for research in cognitive development. Neural Networks **23**(8-9) (Oct 2010) 1125–1134

4. Asfour, T., Regenstein, K., Azad, P., Schroder, J., Bierbaum, A., Vahrenkamp, N., Dillmann, R.: Armar-iii: An integrated humanoid platform for sensory-motor control. In: Humanoid Robots, 2006 6th IEEE-RAS International Conference on, IEEE (2006) 169–175

5. Zollo, L., Guglielmelli, E., Teti, G., Laschi, C., Eskiizmirliler, S., Carenzi, F., Bendahan, P., Gorce, P., Maier, M.A., Burnod, Y., et al.: A bio-inspired neuro-controller for an anthropomorphic head-arm robotic system. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE (2005) 12–17

6. Hoang, N.P., Pham, H.T.: Design of a compliant bio-inspired camera-positioning mechanism for autonomous mobile robots. In: 2017 International Conference on System Science and Engineering (ICSSE). (July 2017) 325–329

7. Gao, B., Zhu, Z., Zhao, J., Jiang, L.: Inverse kinematics and workspace analysis of a 3 dof flexible parallel humanoid neck robot. Journal of Intelligent and Robotic Systems **87**(2) (Mar 2017) 211–229

8. Lingampally, P.K., Selvakumar, A.A.: A humanoid neck using parallel manipulators. 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA) (Dec 2016)

9. Lapeyre, M.: Poppy: open-source, 3D printed and fully-modular robotic platform for science, art and education. Theses, Université de Bordeaux (November 2014)

10. Marques, H.G., Jäntsch, M., Wittmeier, S., Holland, O., Alessandro, C., Diamond, A., Lungarella, M., Knight, R.: Ecce1: The first of a series of anthropomimetic musculoskeletal upper torsos. In: 2010 10th IEEE-RAS International Conference on Humanoid Robots. (Dec 2010) 391–396

11. Złotowski, J., Proudfoot, D., Yogeeswaran, K., Bartneck, C.: Anthropomorphism: Opportunities and challenges in human–robot interaction. International Journal of Social Robotics **7**(3) (Nov 2014) 347–360

12. Bogduk, N., Mercer, S.: Biomechanics of the cervical spine. i: Normal kinematics. Clinical Biomechanics **15**(9) (Nov 2000) 633–648

13. Thiriet, P.: La colonne cervicale haute atlas et axis. https://www.youtube.com/watch?v=OP924h2bc-w (2015)

14. Asano, Y., Okada, K., Inaba, M.: Design principles of a human mimetic humanoid: Humanoid platform to study human intelligence and internal body system. Science Robotics **2**(13) (2017)

15. Nori, F., Jamone, L., Sandini, G., Metta, G.: Accurate control of a human-like tendon-driven neck. In: Humanoid Robots, 2007 7th IEEE-RAS International Conference on, IEEE (2007) 371–378

16. Blanchard, A.: BLAAR : Basic Libraries And Application for Robotics. http://blaar.org (2014–2018)

17. Blanchard, A.: Pinobo : A bio-inspired compliant robot. http://blaar.org/pinobo.html (2018)

18. Falotico, E., Cauli, N., Kryczka, P., Hashimoto, K., Berthoz, A., Takanishi, A., Dario, P., Laschi, C.: Head stabilization in a humanoid robot: models and implementations. Autonomous Robots **41**(2) (Jul 2016) 349–365