

Steering a humanoid robot by its head

Manish N. Sreenivasa, Philippe Souères, Jean-Paul Laumond and Alain Berthoz

Abstract— We present a novel method of guiding a humanoid robot, including stepping, by allowing a user to move its head. The motivation behind this approach comes from research in the field of human neuroscience. In human locomotion it has been found that the head plays a very important role in guiding and planning motion. We use this idea to generate humanoid whole-body motion derived purely as a result of moving the head joint. The input to move the head joint is provided by a user via a 6D mouse. The algorithm presented in this study judges when further head movement leads to instability, and then generates stepping motions to stabilize the robot. By providing the software with autonomy to decide when and where to step, the user is allowed to simply steer the robot head (via visual feedback) without worrying about stability. We illustrate our results by presenting experiments conducted in simulation, as well as on our robot, HRP2.

I. PROBLEM STATEMENT AND CONTRIBUTION

The term "Humanoids" literally means human-like. The anthropomorphic structure that humanoid robots share with humans, provide them with several interesting properties. Probably the most powerful of these is the ability to walk. Legged locomotion in humanoids has opened up various potential avenues of application where the capability to step over obstacles is important. But due to the very fact that humanoids and humans are similar in structure, planning walking motion is a complicated task.

We argue that better understanding human movement may help in organizing humanoid robot whole body motion. This statement is based on reviewing and taking inspiration from literature in the field of human neuroscience. In this paper we show how a humanoid robot can be tele-operated, including stepping, only by considering the intentional motion of the head.

A. Robotics vs. Neuroscience perspective

From a robotics perspective, some early attempts at legged locomotion involved simplifying the humanoid model as an inverted pendulum and using the Zero Moment Point (ZMP) to plan stepping motion [1], [2]. The ZMP condenses the complicated dynamics of a humanoid, which is actually represented by the positions, velocities and accelerations of all

Manuscript received February 27, 2009. This work was partially supported by the pre-project ROMA financed by the interdisciplinary research program Neuroinformatique (CNRS) and the French ANR project LOCANTHROPE.

Manish N. Sreenivasa, Philippe Souères and Jean-Paul Laumond are with the Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de Recherche Scientifique, 7 Avenue du Colonel Roche, 31077 Toulouse, France (phone: +33 (0)5 61 33 79 02; fax: +33 (0)5 61 33 64 55; manu/soueres/jpl@laas.fr).

Alain Berthoz is with the Laboratoire de Physiologie de la Perception et de l'Action, Collège de France, Centre National de Recherche Scientifique, Paris, France alain.berthoz@college-de-france.fr.

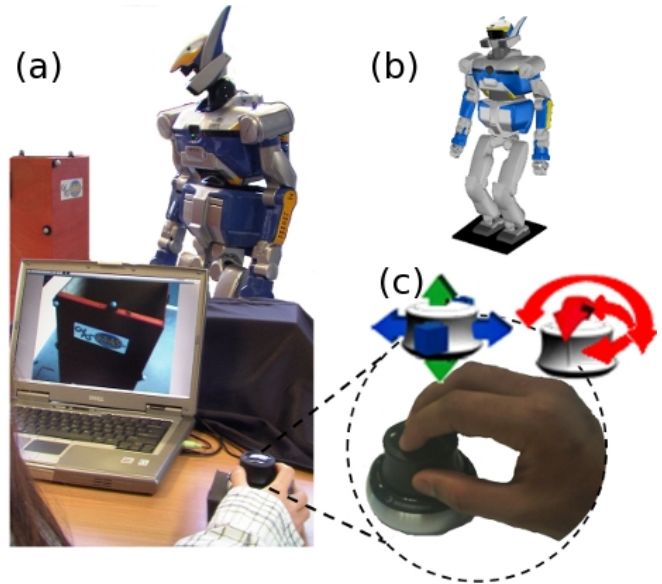


Fig. 1. (a) Snapshot of user maneuvering HRP2 while viewing the output from its cameras. (b) The humanoid robot HRP2 in default position (c) Magnified view of the 6D mouse with all available motion axes.

robot DoFs, to one single point [3]. In 2003, Kajita proposed a preview controller based approach that compensated for differences between the simplified humanoid model and the actual robot, to produce more robust walking motion [4]. Using this approach, we have seen humanoid robots accomplish several complicated tasks like, for example, stepping while lowering the Center of Mass (CoM) height [5], simultaneous reaching and stepping tasks [6], and manipulating objects while stepping [7]. Recently, other approaches have developed more general criteria for maintaining stability [8], [9].

In humans, it has been shown that the head and gaze play a very important role in locomotion, and in fact in any motor movement. For example, while grasping objects we direct our gaze towards it [10]. While reaching for objects out of immediate reach, it seems that humans create a gaze centered reference frame [11]. During dynamic equilibrium, as well as locomotion, the head is stabilized in rotation about the yaw axis [12]-[14]. This stabilization is probably useful to allow a more stable reference frame for egocentric visual motion perception and better visual-vestibular matching.

Another aspect of head behavior during locomotion is the anticipation of changes in trajectory. Research has shown that the head yaw angle anticipates body yaw (shoulder and trunk) and shift in locomotor trajectory [12]-[19]. Simply put, the head looks into a turn before the rest of the body

and before changing the walking course. This has even been found to occur in children as young as 3 to 5 years [20]. This anticipatory nature of head motion has been suggested to occur in order to gather advance visual information about the trajectory and potential obstacles [12], [14], [17]-[19]. The general evidence from these studies suggests that the control of the multiple degrees of freedom of the body during locomotion is organized from the head down, and not, as implemented in most humanoid robots, from the feet up [21].

B. Our contribution

This study implements the idea of tele-operating a humanoid robot, including stepping, by controlling its head. The idea of tele-operation of a humanoid robot is not new. One such study approached this issue by manually choosing and switching control between the various joints of the humanoid robot [22], [23]. While this does enable the user to control the robot, and accomplish a range of stepping and reaching motions, it is still not a very intuitive approach and, in principle, very different from how human motion is planned.

In our study we show that by only taking a 6D input from a user (3 translations + 3 rotations, Fig. 1(c)) and applying that to the head of a humanoid robot, we can generate deliberative whole body motion of the robot. The experience of steering the humanoid robot is accentuated by allowing the user to receive visual feedback about the environment from the robot's perspective (Fig. 1(a)). The algorithm developed in this study evaluates the intentional motion of the user from the mouse input. The architecture detailed in the following sections brings together this unique algorithm with state-of-the-art robotics approaches on step planning [4] and inverse kinematics [24].

The contribution of this paper is in showing that the decision of when and where to step can be deduced from the position and orientation of the head. This idea, motivated by neuroscience principles, is implemented in a working architecture provided in this paper. We present three scenarios to illustrate the flexibility of this approach in manipulating and maneuvering a humanoid robot through the environment.

II. GENERAL SOFTWARE ARCHITECTURE

The primary goals of the control software in this study were three fold:

- 1) Allow user to move the head of the robot in real time
- 2) Generate whole body motions in response to head motion
- 3) Check humanoid stability and generate stepping motions when required

Fig. 2 shows a simplified flowchart describing the various steps to implement these goals. Here we describe the various components of the software architecture in further detail.

III. TRANSFERRING INPUT FROM USER TO HUMANOID

We chose to use a 6 dimensional mouse (3DConnexion, Logitech) to record motion from a user and transfer it to

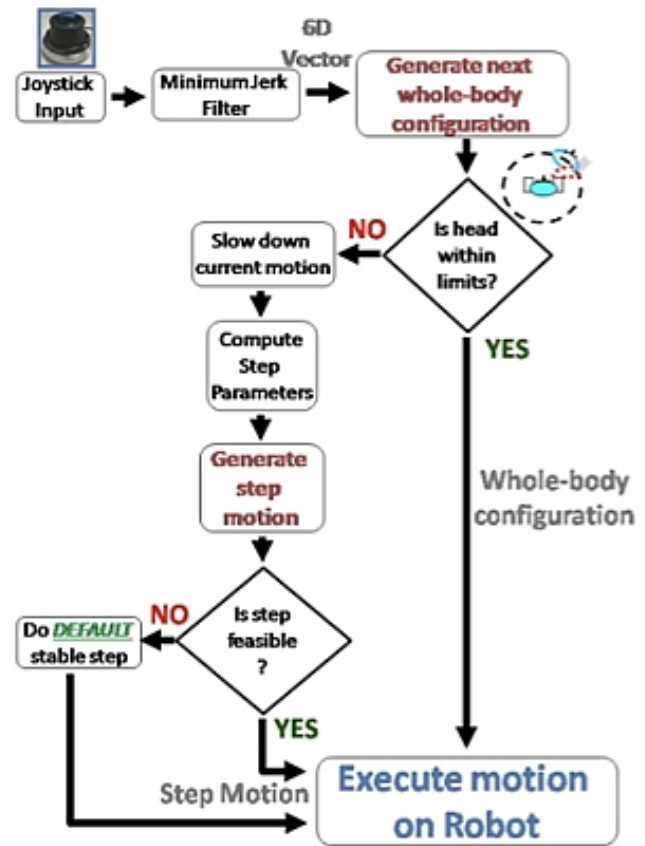


Fig. 2. Flowchart showing the sequence of steps from user input to motion execution on the humanoid.

the head joint of the robot. Input from the mouse was a 6D vector consisting of 3 translational (x, y, z) and 3 rotational motions (roll, pitch, yaw). In order to ensure a smooth and intuitive motion transfer we had to first process this vector using minimum jerk filtering. The minimum jerk model used to generate this motion was based on the work by Flash and Hogan [25]. Our implementation required that we discretize their time-continuous model as follows. For each of the 6 dimensional inputs we computed the next minimum jerk step q_{i+1} using:

$$q_{i+1} = [x_{i+1} \ v_{i+1} \ a_{i+1}] = A * q_i + B * x_f$$

Where,

$$v = \frac{dx}{dt}, \quad a = \frac{dv}{dt}$$

$$A = \begin{bmatrix} 1 & \Delta t & \Delta t^2 \\ 0 & 1 & \Delta t \\ \left(\frac{-60*\Delta t}{D^3}\right) & \left(\frac{-36*\Delta t}{D^2}\right) & \left(\frac{-9*\Delta t}{D} + 1\right) \end{bmatrix}$$

$$B = \left[0 \ 0 \ \frac{60*\Delta t}{D^3} \right]$$

x_f = target value

D = total time to end position

Δt = time step

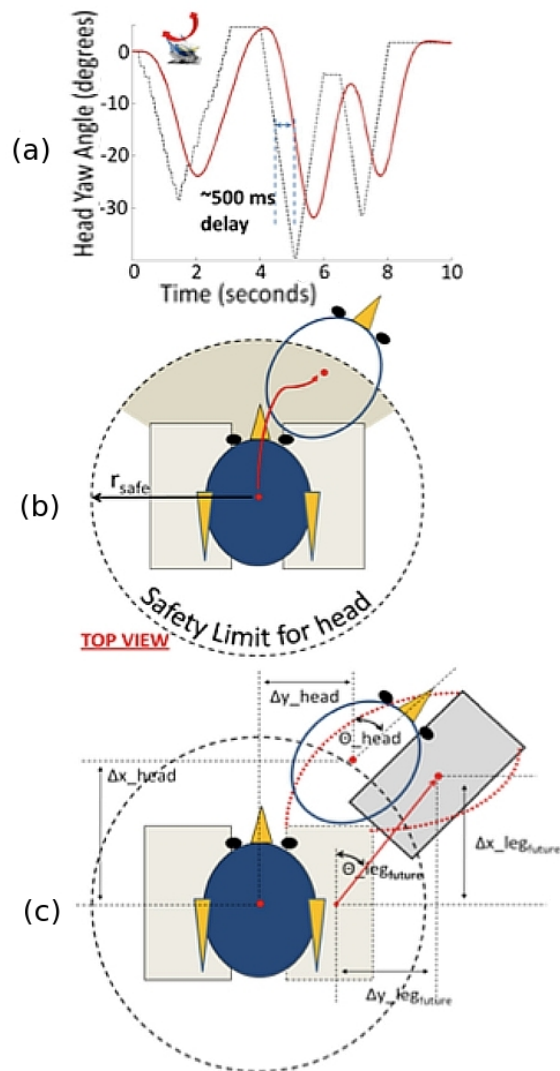


Fig. 3. (a) Input from yaw axis of mouse (dotted line) and output from minimum jerk model (solid line) (b) Dotted circle and shaded arc of circle, show the range of head position and yaw, respectively, without needing to step. (c) As head crosses the circle a stepping configuration is calculated based on current head position and orientation. The shaded rectangle represents the future position of the foot.

To illustrate the effectiveness of this method, Fig. 3(a) plots the raw output of the yaw axis of the mouse and the output after minimum jerk filtering. This output was then applied to the head joint of the humanoid. As an inevitable result of the filtering there was a delay of about 500 ms between user input and filter output. Rather than being a drawback, this delay gave an impression of being akin to motion inertia of the robot.

IV. GENERATION OF WHOLE BODY MOTION

In kinematic chains with multiple degrees of freedom, it may be possible to execute several tasks simultaneously. There are specialized algorithms that can solve this redundancy [26]. However, assigning multiple tasks could lead to conflicts and unsatisfactory configurations, especially if they are all treated with equal importance. This problem can

be solved by assigning priorities [27] to each of the tasks and then solving this stack [28]. For instance, in humanoid robots, an example of a stack of tasks could be keeping the feet flat on the ground (\uparrow high priority), maintaining a certain position for the CoM ($-$ mid priority) and then reaching for an object with the hand (\downarrow low priority).

In our architecture, we use the Generalized Inverse Kinematics (GIK) engine [24], developed by our own lab, to generate whole body motions. GIK implements the approach in [26] to solve redundancy, packaging it with helpful tools to plan robot whole body motion.

For our purpose we defined the following constraints: Maintain feet position and orientation on floor, position of CoM (at center of support polygon) and position and orientation of head joint. Whole-body motion was generated by updating the final constraint on the head joint, while keeping the other two unchanged. Basically this means that as the head moves, the feet remain planted on the ground and the CoM stays at its position while all other joints are free to move. We also experimented by allowing the CoM to move within the support polygon, but found that this resulted in unstable configurations under rapid user input. This stack of constraints was solved to give a whole-body configuration every 5 ms.

Updating the position and orientation of the head joint:

In order to transfer motion from the mouse to the head joint we first polled the current attitude of the head joint. The 6 dimensional vector input from the mouse was then transformed into the local coordinate system of the head joint (see Fig. 4(b) in Results section for illustration of coordinate systems on HRP2). This was done to create the impression of true tele-operation, i.e. the user feels like he/she is sitting inside the head of HRP2. Simply put, if the user pitches HRP2's head downwards and then pushes the mouse forwards, the head will move forward and down, and will take the rest of the body along with it.

V. STABILITY AND GENERATION OF STEPPING MOTION

If we were to simply move a humanoid's head, it will eventually reach the limits of its stability and fall. Since we constrain the projection of the CoM to remain at the center of the support polygon, in our case the humanoid will not be able to execute the task (further movement of the head in same direction) as no more DoFs are available.

The conventional method to evaluate dynamic stability is to ensure that the ZMP remains inside the support polygon of the robot. Theoretically, this means that the robot is stable in the entire region defined by the support polygon. But in practice, if the ZMP is allowed to reach the boundary of the support polygon, it becomes impossible to stop in time to avoid falling. Additionally, if the head travels too far away from the center of the support polygon it will drag the chest and waist along with it. In this posture, it is difficult to compute stable step motion since the method proposed by Kajita [4] assumes the waist to be close to vertical. To avoid these problems we devised a method that answers the two

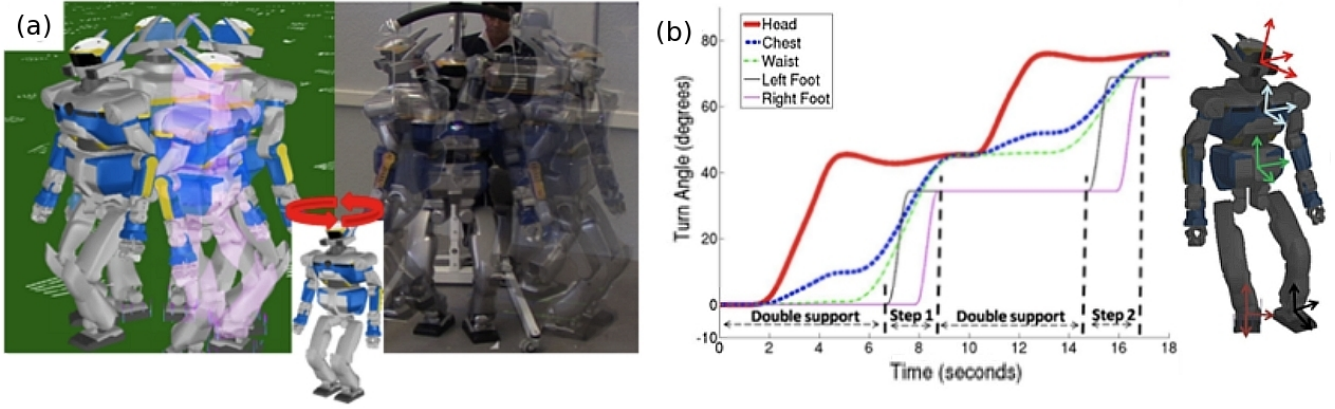


Fig. 4. (a) Pictures of HRP2 turning on the spot, in simulation (left) and on the real robot (right) (b) The progression of the yaw angles of the various joints of HRP2 during the first two steps of this scenario.

basic questions: When to step? & Where to step? The former question deals with distinguishing when the humanoid is at the limit of its movement range and the latter decides what future configuration will create a more stable posture for further movements.

A. When to step?

This was done by defining a safe circle around the center of the support polygon (dotted circle in Fig. 3(b)). As soon as the head projection on the floor reaches the boundary of this circle, it is stopped smoothly which in turn slows down the whole-body motion of the robot. This is done by setting a target velocity of zero for the head and letting the minimum jerk filter slow it down in a controlled manner. We found that even if the head was moving at maximum velocity at the point of crossing the circle boundary, it needed only 250 ms to slow down quickly enough to still make it possible to step. The radius of the circle, r_{safe} , was determined by exhaustively testing various body configurations.

Before stepping, we brought the chest and waist back to vertical position and planned stepping motions from this posture. This added another 1 second to the motion. The actual stepping motion was planned using the Kajita method detailed in [4]. From the time the head crossed the "safe circle", the user was disallowed from changing its position, since this would perturb the dynamic stability of the humanoid. The only exceptions to this were the head yaw and pitch angles. The user was allowed to modify these values while stepping since it did not affect dynamic stability much and simultaneously improved the tele-operation experience. Computing stepping motion was fast enough to avoid slowing down the control in any way. Depending on the step, it took approximately 2.5-3.5 seconds to shift from one double support phase to the other.

B. Where to step?

The question of where to step was solved by devising an algorithm that used the current head position and orientation to compute future foot configuration. We first decided which foot to use for stepping. This was done by picking the foot

which lay in the direction of head motion (example Fig. 3(c), translating head towards right chooses the right foot). However, there were exceptions which switched the choice of foot based on whether the motion was forward or backward, or if the chosen foot was already forward. Fig. 3(c) also shows the configuration of the right foot, before and after stepping. The future stepping position was calculated as

$$\Delta x_{leg_future} = \pm \alpha_x \pm \beta_x * \Delta x_{head}$$

$$\Delta y_{leg_future} = \pm \alpha_y \pm \beta_y * \Delta y_{head}$$

where, α_x and α_y decide the basic step size depending on the minimum safe distance between the two feet, and the maximum stepping distance achievable by the robot. β_x and β_y are used to tune the extent to which head displacement modifies feet placement. Δx_{head} and Δy_{head} are the current distances between head center and support polygon center in x and y directions. Δx_{leg_future} and Δy_{leg_future} are positive or negative depending on whether the step is forwards, backwards, left or right.

In addition to translating the foot we also turn the foot depending on the yaw angle of the head, i.e. $\Delta \theta_{leg_future} = \Delta \theta_{head}$. This enables the user to maneuver the robot in a way that makes it possible to walk in curved paths. The final choice of future foot position and orientation was verified to avoid collision with the non-stepping foot, as well as for collisions between the knees of the robot. Additionally, stepping motions were also activated when head joint yaw angle exceeded 40° relative to the waist (shaded area in Fig. 3(b)). This was done because twisting the humanoid head (and consequently the rest of the body) beyond this limit made it unrecoverable for further stepping. This type of stepping was achieved by first stepping with one foot, simultaneously rotating it by 40° , and then the other.

C. Recovering posture in critical situations

Due to the live nature of the control, it is difficult to predict and compensate for all unstable scenarios. In fact, after a period of time the humanoid will most likely arrive

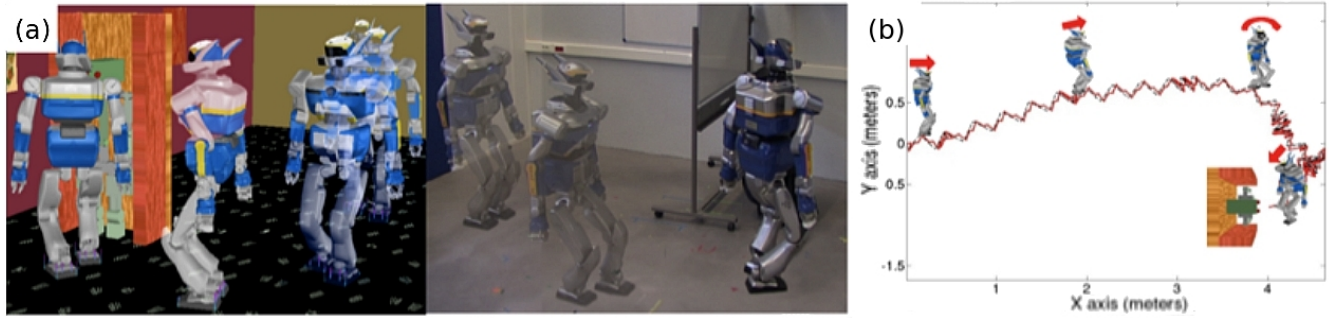


Fig. 5. (a) HRP2 being maneuvered through space to find distant, hidden objects (b) Plot of head position (solid line) and CoM position (dotted line) during motion. The zig-zag motion of the head and CoM position was due to the user alternatively looking left and right at every step. A constant forward (and then turning) motion of the mouse would also generate the same trajectory but without the zig-zags.

at a configuration where it cannot compute a stable future stepping position in the direction wanted by the user. For example, in Fig. 3(c), the robot has used the right leg to execute a step. If in this configuration, the user continues to turn and push the head in the same direction, the robot will have to move towards the right again. It is not possible to use the right leg for this because it is already forward and extended. So we need to swing the left leg forward while rotating it towards the right, thus freeing the right leg for further steps. This seems somehow intuitive from how humans would react in such a situation. But executing such a motion, although kinematically possible, would generate unstable dynamics and put the robot in an odd final position (knees pointing inwards).

In these cases, our architecture overrules the user input and returns the robot back to the default half-sitting configuration. This is done by moving the head projection on the floor, back to the center of the support polygon, and then stepping with both feet till they are 20 cm apart. During this motion, the chest, waist and feet orientations are made to face in the same direction as the head joint (Fig. 1(b) shows HRP2 in default half-sitting configuration). By thus rotating the robot we at least manage to satisfy the directional input from the user, if not the position.

VI. RESULTS

In this section we present the results from the simulations and real experiments conducted on our humanoid robot HRP2. In order to illustrate the flexibility of the control scheme we chose three scenarios which highlight its different aspects (video of the experiments also provided). For the scenarios presented, we used a "safe circle" of radius, $r_{safe} = 0.07$ m. The step size parameters feasible for HRP2 were $\alpha_x = 0.15$ m, $\alpha_y = 0.2$ m and $\beta_x = \beta_y = 0.05$.

A. Scenario 1: Turning on the spot

Fig. 4(a) shows snapshots of HRP2 turning 360° on the same spot (however, in the process of stepping the CoM does move a certain amount). It should be noted that the turning of the robot was a result of the head joint yaw angle increasing beyond a limit (discussed earlier in section IV) and thus necessitating a rotational step in order to preserve stability.

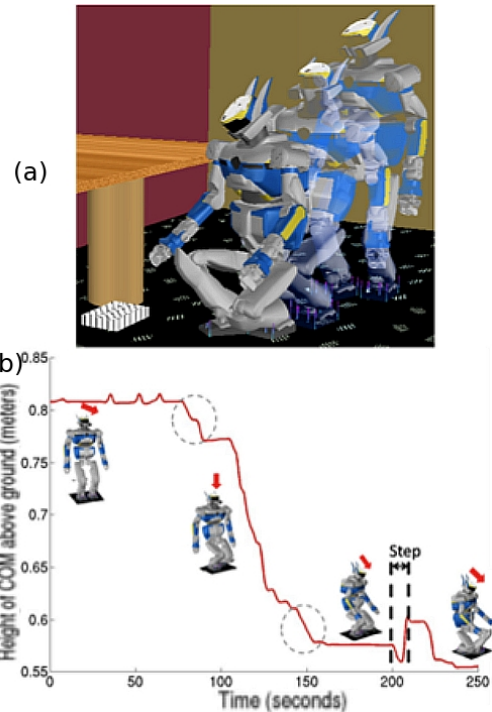


Fig. 6. (a) Lowering the humanoid robot in order to view objects at ground level. (b) Plot of the CoM height above ground. The dotted circles indicate the instances where the robot height was lowered by pitching the head down and then moving forward.

This type of movement can be imagined as being similar to that of a human trying to explore an unknown environment by taking in a 360° view. Based on the current limitations HRP2 required 11 steps to make a complete turn-around.

B. Scenario 2: Searching for hidden objects

The purpose of this scenario was to show how a user can maneuver HRP2 through space to discover hidden objects using visual feedback provided by the robot's camera. The user moves HRP2 forward and then turns the robot around to discover an object hidden behind a screen (Fig. 5(a) in simulation, left and on real robot, right). Fig. 5(b) shows the movement of the CoM and the center of the head during the motion.

C. Scenario 3: Looking under a table

This scenario was designed specifically to illustrate the possibility to lower the height of the robot (Fig. 6). In addition to this, we also show that the ZMP method used to generate stepping motions is still valid at such extremely low CoM heights.

The lowering of the CoM height occurs as the joystick user moves the head in a negative Z direction. It should also be pointed out that because of the way the control is implemented, the whole body also moves downwards when the head is pointed downwards and then moved forwards. These cases are shown as circled regions in Fig. 6(b). We only executed this scenario in simulation due to the excessive, and potentially damaging, leg currents that are generated in HRP2 while bending the knees very low.

VII. CONCLUSION

The core idea presented in this study shows how a humanoid robot can be steered and made to step, by driving the head. It is important to note here that stepping positions are decided automatically and in real-time without human input, and in this sense the tele-operation is autonomous. The goal of this study was not to develop an autonomous navigation strategy. However, the approach detailed in this study could easily be integrated with a higher level supervisor that would allow the execution of a sequence of tasks autonomously, using sensor based control loops such as visual-servoing.

Here, the human user closes the perception-action loop by viewing the environment from the robots perspective, and then reactively steering the head via a mouse. The reason for using the head to guide motion is due to the presence of important sensing systems (vision in humanoids and vision, vestibular and auditory systems in humans).

In this study we have taken inspiration from human behavior. Further studying human movement can give us additional hints towards organizing humanoid whole body motion. To this end, we are currently leading motion capture experiments to extract, from human behavior, kinematic or dynamic invariants that could be used to plan the next foot placement from the intentional motion of the head.

ACKNOWLEDGMENT

The authors would like to thank Oussama Kanoun, Anthony Mallet and Eiichi Yoshida for taking part in helpful discussions and assisting with the experiments on HRP2.

REFERENCES

- [1] J. Yamaguchi, A. Takahashi and I. Kato, "Development of a biped walking robot compensating for three-axis moment by trunk motion", in *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, 1993.
- [2] S. Kajita, T. Yamamura and A. Kobayashi, "Dynamic walking control of a biped robot along a potential energy conserving orbit", *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 431-438, 1992.
- [3] M. Vukobratovic and J. Stepanenko, "On the Stability of Anthropomorphic Systems", *Mathematical Biosciences*, vol.15, pp.1-37, 1972.
- [4] S. Kajita et al., "Biped walking pattern generator by using preview control of Zero-moment point", in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1620-1626, 2003.
- [5] O. Stasse et al., "Integrating walking and vision to increase humanoid autonomy", *Int. Journal of Humanoid Robotics*, vol. 5, pp. 287-310, 2008.
- [6] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, "Visually-guided grasping while walking on a humanoid robot", in *IEEE Int. Conf. on Robotics and Automation*, pp. 3041-3047, 2007.
- [7] E. Yoshida et al., "Whole-body motion planning for pivoting based manipulation by humanoids", in *IEEE Int. Conf. on Robotics and Automation*, pp. 3181-3186, 2008.
- [8] P. B. Wieber, "Viability and predictive control for safe locomotion", in *Proc. of IEEE/RSJ Conf. on Intelligent Robotics and Systems*, pp. 1103-1108, 2008.
- [9] H. Hirukawa et al., "A universal stability criterion of the foot contact of legged robots - adios ZMP", in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2006.
- [10] R. S. Johansson, G. Westling, A. Bäckström, J. R. Flanagan, "Eye-hand coordination in object manipulation", *Journal of Neuroscience*, vol. 21, pp. 6917-6932, 2001.
- [11] M. Flanders, L. Daghestani and A. Berthoz, "Reaching beyond reach", *Experimental Brain Research*, vol. 126, pp. 19-30, 1999.
- [12] H. Hicheur, S. Vieilledent and A. Berthoz, "Head motion in humans alternating between straight and curved walking path: combination of stabilizing and anticipatory orienting mechanisms", *Neuroscience Letters*, vol. 383, pp.87-92, 2005.
- [13] T. Imai, S. T. Moore, T. Raphan and B. Cohen, "Interaction of the body, head and eyes during walking and turning", *Experimental Brain Research*, vol. 136, pp. 1-18, 2001.
- [14] P. Prevost, Y. Ivanenko, R. Grasso and A. Berthoz, "Spatial invariance in anticipatory orienting behavior during human navigation", *Neuroscience Letters*, vol. 339, pp. 243-247, 2002.
- [15] G. Courtine and M. Schieppati, "Human walking along a curved path. I. Body trajectory, segment orientation and the effect of vision", *European Journal of Neuroscience*, vol. 18, pp. 177-190, 2003.
- [16] H. Hicheur, Q. C. Pham, G. Arechavaleta, J. P. Laumond and A. Berthoz, "The formation of trajectories during goal-oriented locomotion in humans I. A stereotyped behavior", *European Journal of Neuroscience*, vol. 26, pp. 2376-2390, 2007.
- [17] R. Grasso, P. Prevost, Y. P. Ivanenko and A. Berthoz, "Eye-head coordination for the steering of locomotion in humans: an anticipatory synergy", *Neuroscience Letters*, vol. 253, pp. 115-118, 1998.
- [18] M. A. Hollands, K. L. Sorensen and A. E. Patla, "Effects of head immobilization on the coordination and control of head and body reorientation and translation during steering", *Experimental Brain Research*, vol. 140, pp. 223-233, 2001.
- [19] M. N. Sreenivasa, I. Frissen, J. L. Souman and M. O. Ernst, "Walking along curved paths of different angles: the relationship between head and trunk turning", *Experimental Brain Research*, vol. 191, pp. 313-320, 2008.
- [20] R. Grasso, C. Assaiante, P. Prevost and A. Berthoz, "Development of anticipatory orienting strategies during locomotor tasks in children", *Neuroscience & Biobehavioral Reviews*, vol. 22, pp. 533-539, 1998.
- [21] A. Berthoz, "The brain's sense of movement", *Harvard University Press*, Cambridge, MA, 2000.
- [22] E. S. Neo, K. Yokoi, S. Kajita, F. Kanehiro, and K. Tanié, "A switching command-based whole-body operation method for humanoid robots", *IEEE/ASME Trans. on Mechatronics*, vol. 10, pp. 546-559, 2005.
- [23] E. S. Neo, K. Yokoi, S. Kajita, and K. Tanié, "Whole-body motion generation integrating operators intention and robots autonomy in controlling humanoid robots", *IEEE Trans. on Robotics*, vol. 23, pp. 763-775, 2007.
- [24] E. Yoshida, O. Kanoun, C. Esteves and J. P. Laumond, "Task-driven support polygon reshaping for humanoids", *IEEE-RAS Int. Conf. on Humanoid Robotics*, pp. 208-213, 2006.
- [25] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model", *Journal of Neuroscience*, vol. 5, pp. 1688-1703, 1985.
- [26] Y. Nakamura, "Advanced Robotics: Redundancy and Optimization", Addison-Wesley Longman Publishing, Boston, 1991.
- [27] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems", in *Proc. IEEE Int. Conf. Adv. Robot.*, pp. 1211-1216, 1991.
- [28] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control", *IEEE Trans. on Robotics*, vol. 23, pp. 60-72, 2007.