Auton Robot (2016) 40:561–580
DOI 10.1007/s10514-016-9546-4

# Multi-contact vertical ladder climbing with an HRP-2 humanoid

Joris Vaillant[1,2] · Abderrahmane Kheddar[1,2] · Hervé Audren[1,2] · François Keith[1,2] · Stanislas Brossette[1,2] · Adrien Escande[1] · Karim Bouyarmane[1,2] · Kenji Kaneko[1] · Mitsuharu Morisawa[1] · Pierre Gergondet[1] · Eiichi Yoshida[1] · Suuji Kajita[1] · Fumio Kanehiro[1]

Received: 16 August 2014 / Accepted: 9 January 2016 / Published online: 3 February 2016

**Abstract** We describe the research and the integration methods we developed to make the HRP-2 humanoid robot climb vertical industrial-norm ladders. We use our multi-contact planner and multi-objective closed-loop control formulated as a QP (quadratic program). First, a set of contacts to climb the ladder is planned off-line (automatically or by the user). These contacts are provided as an input for a finite state machine. The latter builds supplementary tasks that account for geometric uncertainties and specific grasps procedures to be added to the QP controller. The latter provides instant desired states in terms of joint accelerations and contact forces to be tracked by the embedded low-level motor controllers. Our trials revealed that hardware changes are necessary, and parts of software must be made more robust. Yet, we confirmed that HRP-2 has the kinematic and power capabilities to climb real industrial ladders, such as those found in nuclear power plants and large scale manufacturing factories (e.g. aircraft, shipyard) and construction sites.

**Keywords** Humanoid robots · Multi-contact motion planning and control · Field humanoid robots · Disaster humanoid robots

**Electronic supplementary material** The online version of this article (doi:10.1007/s10514-016-9546-4) contains supplementary material, which is available to authorized users.

✉ Abderrahmane Kheddar
kheddar@gmail.com

[1] CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/RL, Tsukuba, Japan

[2] CNRS-UM2 LIRMM Interactive Digital Human Group, UMR5506, Montpellier, France

## 1 Introduction

Humanoid robots reached a noticeable level in technological maturity for walking on flat grounds. The Honda's Asimo is a good illustration of such an achievement. Despite tremendous research efforts, such technology maturity is not observed in walking on uneven or deforming terrains, or in non-gaited motion requiring whole-body multi-contact motion such as climbing ladders or irregular stairs of any kind.

The DARPA Robotics Challenge (DRC)[1] trials included industrial ladder climbing as one of the eight challenging tasks to be performed autonomously by a robot. Indeed, ladders of different heights and angles of inclination can be found in nuclear power plants, construction sites, and large scale manufacturing (e.g. shipyard and aircraft). Climbing ladders for maintenance, repair or building operations is one of the recurrent tasks humans achieve easily. For intervention in disaster sites, fires, or nuclear power-plants dismantling, ladders can be brought to areas for which the usual access ways are damaged, not practicable, or do not exist anymore. Even in houses, we use ladders to perform various makeshift tasks. Humans climb ladders up and down with ease, whereas the same task is very complex for robotic systems, even for climbing customized ones. The DRC prepared two ladders: one inclined by 70° and one by 60°. Each of these ladders have 10 cm-wide rungs, and handrails that can optionally be removed. During the 2013 December DRC qualifiers, several teams tried different strategies for ladder climbing. But all teams chose the less inclined 60° ladder. This is more of a stairs than a ladder. As a mater of fact, the winning team, SHAFT, climbed it with feet only. However, their robot used two-feet-one-rung intermediate transitions before climbing

[1] http://www.theroboticschallenge.org/.

each next rung, and climbed up backward to avoid collisions between knees and rungs when bending the knees. The HUBO+ humanoid robot, based on multi-contact planning technology (Zhang et al. 2013), could also climb almost all of it, using a similar backward strategy (Luo et al. 2014) and a two-feet on one-rung transition phases, together with arm grasps on stringers. It failed at the last rung. None of the remaining participant teams succeeded in climbing the ladder.

Prior to the DRC, a number of customized ladder climbing robots were made. For example, in Iida et al. (1989), a Japanese team from Toshiba company designed a four limbs robot for nuclear power plants. This robot had four prismatic arms with grippers. The climbing sequence consisted in transiting one arm at a time for the preparation of the lifting that is made with the four arms in contact: two per different rung. In Bevly et al. (2000), a planar three legged climbing robot was demonstrated climbing pegs disposed as a vertical ladder. This study revealed that by identifying key motion-primitives and using physics simulation, the planning is tractable and can be optimized. This idea is interesting and could be investigated further. Ladder climbing was also demonstrated with a deformable-on-demand legs robot in Nakai et al. (2002). The latter work is more a concept demonstration than a plausible solution. In Fujii et al. (2008), a six legged spider-like robot is programmed to climb successfully a vertical ladder. Interestingly, this study showed that having enough limbs would allow climbing without firm grasps, since contact formations are all of hook-like type. Yet, contact forces should be monitored since despite having many legs, geometric discrepancies may cause high internal efforts and a bad distribution of the load among the legs.

In Yoneda et al. (2008) a gorilla-type robot was shown to climb a vertical ladder. The authors achieved three different climbing gaits: transverse, pace with constant velocity and trot with acceleration. This study reveled the importance in considering dynamic effects and suggested to pay particular attention to the axis of yawing. Lastly, Noda et al. (2014) demonstrated capabilities of the HRP-2 in climbing inclined ladders (two steps and reaching) and took a strategy which consists in distributing contact forces and moments together with joint torques. Although the authors used different names, the general approach is similar to our multi-contact strategies described in Bouyarmane and Kheddar (2012) and Escande et al. (2013).

We address the climbing of vertical ladders by the HRP-2 robot, see Fig. 1, and extend our work in Vaillant et al. (2014) with more experiment details and results, a description of the overview architecture with a new posture generator, a description of the FSM, etc. Two main challenges drove our research:

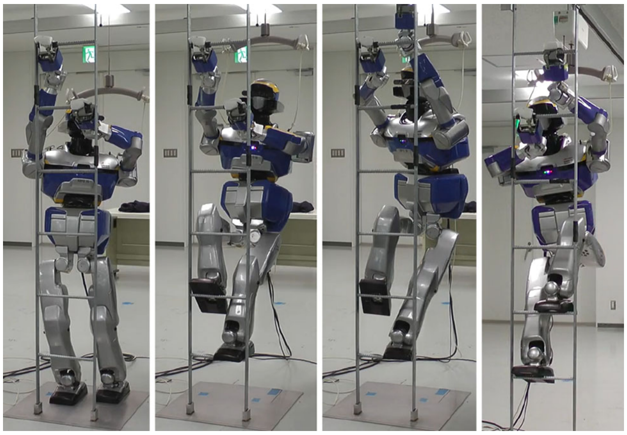


**Fig. 1** HRP-2 climbing a vertical ladder. Notice that: (i) it is not possible to put two feet on one rung (ii) closed grippers do not grab firmly the rungs (iii) each foot can be freely positioned on any rung: the tilting of the *right* foot increases the attainability of a rung by the *left* gripper

(1) The first one was to address directly vertical industrial-norm ladders, prohibiting any change or adjustment as this would not be possible in practice.
(2) The second is to use HRP-2 as it is and exploit its capabilities to their limits. The idea is to work on the software as much as possible prior to any hardware modification. As will be seen in the section dedicated to experiments, this was not a 'reasonable' option since we had a very hard time with the current design of the grippers. Relatively to Yoneda et al. (2008), Noda et al. (2014) and Luo et al. (2014) and others, we cannot use a two-feet on one-rung transitions.

The main objectives of our work are as follows:

- Evaluate our multi-contact planner and controller in the context of ladder climbing;
- Check the capability of HRP-2 to climb a vertical industrial norm ladder;
- Draw lessons for software and hardware modifications.

## 2 Ladder multi-contact planning and control

Figure 2 illustrates the main components of our software architecture; they are explained in more details along the paper.

Our multi-contact planner is model-based. It needs the models of the HRP-2, the ladder, and some parts of the environment that are used to generate, off-line, the sequence of postures in contact to climb the ladder. These postures are passed to a multi-contact finite state machine (FSM) that will split them into subtasks in order to achieve safe contacts and transitions. The FSM elaborates additional steps with their associated tasks, and changes on-line their objectives to deal with different phases of the climbing (unilateral

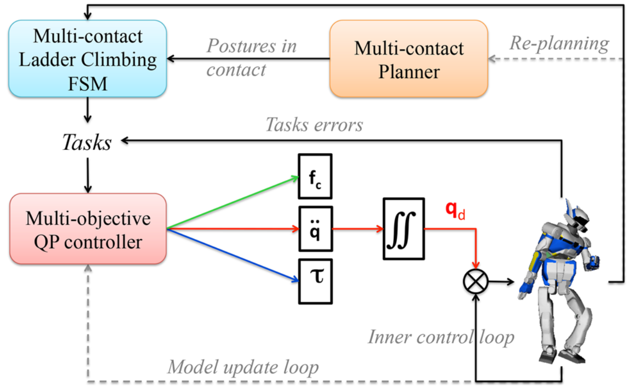


**Fig. 2** Main components of the overall architecture

contact adding and removal, grasps and their release, center of mass (CoM) transfers, etc). These tasks are added to our multi-objective QP controller, which generates implicit trajectories using task-space closed-loop control. The dashed lines in Fig. 2 are modules that are not yet developed.

## 3 Multi-contact planner (MCP)

Contact planning using sampling methods [see early work in Bretl (2006)] was applied to the ladder climbing in simulation (Hauser et al. 2008) and experimented in Zhang et al. (2013). Another contact search method was proposed in Mordatch et al. (2012), which, in practice, can only be used as a preliminary guess of the plan, see Bouyarmane et al. (2009) for another approach. Our contact planning is summarized in Escande et al. (2013) (a ladder climbing example is provided). None of the previous work considered extensions to gather manipulation and locomotion in a single framework, what we did in Bouyarmane and Kheddar (2012). Our approach is distinguishable in that (i) we do not sample the contacts *a priori*, (we consider contact to occur on any part of the robot body and the environment), (ii) it is applied to robotics, hence contrary to the computer animation field, torque limits, collision avoidance, physical plausibility, equilibrium (Wieber 2002; Bretl and Lall 2008)… cannot be ignored, (iii) we can handle other tasks (as far as they are written as constraints) all along the contact planning process (Escande and Kheddar 2009).

Our MPC requires the models of the robot (kinematics including limits, inertia and geometry), the ladder and the environment, as well as a description of the possible surfaces (of both the robot and the environment) which can be used to create contacts. The parametrization of the ladders is similar to that proposed in Luo et al. (2014). The robot is modeled using triangular meshes and each limb is covered by a strict convex hull for a continuous-gradient distance computation (Escande et al. 2014). We specify the areas where contacts are permitted to occur on the ladder (all of it), the robot (on the grippers and on the feet's soles) and the environment. We plan contacts for climbing the ladder by two different ways:

1. using our planner (Escande et al. 2013; Bouyarmane and Kheddar 2012) for which the previously described models are the input. We provide a median ladder straight-line as a potential field along which contact search toward climbing is guided. Then we let the planner find the contact stances and postures automatically;
2. one can also provide the contact pairs interactively, using simulation.

Our planner has a greedy search behavior and seeks for all possible contacts. It is time consuming and can result in non-optimal and sometimes strange climbing gaits. It is necessary to guide the search process by favoring a climbing hands/feet sequence behavior. Since our planner builds the tree of contact stances by either removing a contact or creating a new one at a time, we provided more weight to common transverse climbing sequences, e.g. left (right) hand, right (left) feet, right (left) hand, and left (right) feet or any other such items combination. This is somehow similar to the strategy adopted in Zhang et al. (2013), in the sense that we do not provide the contact stances, but we rather suggest pairs of surfaces of both the robot and ladder that can be in contact.

In all our versions of the MCP, a posture generator (PG)[2] is paired with the search space module (Escande et al. 2013; Bouyarmane and Kheddar 2012). The PG is a non-linear optimization formulation of a generalized inverse kinematics problem. It seeks for viable statically stable postures that can remove or can create contacts as suggested by the space explorer part of the (interactive) planner. The posture must fulfill constraints of joint and torque limits, reaction forces within friction cones, equilibrium, and be free from auto-collision and non-desired collisions. Add other task constraints such as gaze or field-of-view is possible (Escande and Kheddar 2009). If a viable posture is found, the resulting contact and posture is returned and added to the contact tree builder with a cost. Otherwise (i.e. no viable posture is found), failure means a request for an alternative suggestion in terms of robot-ladder contact pairing (creation or removal), or eventually another area from the ladder to try with.

Relatively to our previous PG in Bouyarmane and Kheddar (2012), we brought two novelties in this paper: (i) a richer contact model, and (ii) the possibility for a multi-posture generation (MPG), which generates optimal postures that minimize a cost over the entire path. This proved to be useful for minimizing gripper torques at each contact transition.

[2] The PG is available at https://github.com/jorisv/PG.

### 3.1 Contact modeling in posture generation

In our previous work, to enforce stable contact formation, we favored constraints of the type plane/plane (Escande et al. 2013). In order to compute forces on contacting areas in Escande et al. (2013) or Bouyarmane and Kheddar (2012), we predefined lists of contacting areas on the robot and its surroundings, and we limited the possible contact to cases where a given contact surface of body A was fully included in contact surface from body B or *vice-versa*. To overcome this limitation we often define smaller surfaces (Bouyarmane and Kheddar 2010). The leftmost image in Fig. 3 illustrate inclusive surface constraints, e.g. the foot is entirely contained in the ground surface. This approach allows keeping a constant contact surface during the PG optimization process.

Predefining contact surfaces obviously restricts the possibilities of the planner. For the ladder climbing case, this was very limiting. We proposed in Brossette et al. (2014) another contact model that generates non-inclusive contacts with any position and orientation: the model enforces that a big enough ellipse exists in the intersection of the pair of surfaces in contact. We illustrate this method by the middle image of Fig. 3, where solution of the inscribed ellipse for the left foot on a rung is illustrated (in blue). Finally, we implemented a contact of the type plane/cylinder to have more realistic simulation of the sole/rung contact specifically for ladders with cylinder rungs, which is illustrated by the right most image of Fig. 3.

### 3.2 Posture generation with gripper torque optimization

Our proposed MPG assumes that we have $N$ stances (i.e. contact transitions) for the climbing, we consider that we have $N$ similar robots, each one with its own associated variables and dedicated to a given stance $i \in [1 \dots N]$. We use the following notation in the rest of the paper:

- $X_i$ is the link $i$ transformation matrix w.r.t the overall reference frame;
- $r_i$ is the link $i$ translation vector (component of $X_i$);
- $E_i = [T_i, B_i, N_i]$ the orientation matrix (from $X_i$) and its vector components (the nomenclature of the latter means Tangent and Bi-tangent (tangent space components), and Normal component that are useful to tag contact frames).

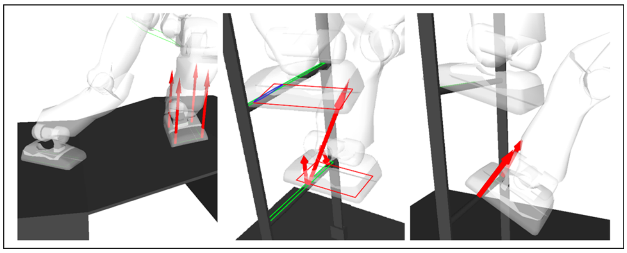


Fig. 3 Three contact models used in the posture generator

For the '$N$ robots', $x = [\mathbf{q}_1^T, \cdots, \mathbf{q}_N^T, \mathbf{f}_1^T, \cdots, \mathbf{f}_N^T]^T$ is the optimization vector, where $\mathbf{q}_i$ is the robot $i$ configuration vector and $\mathbf{f}_i$ the robot $i$ contact forces vector. We use super- or sub-script $i$ to refer to the $i$-th robot. Each robot must satisfy the following constraints:

- Static equilibrium:

$$\underline{\tau} \leq J^i(\mathbf{q}_i)^T \mathbf{f}_i - g^i(\mathbf{q}_i) \leq \overline{\tau} \tag{1}$$

$J$ is the Jacobian matrix of all contact points, $\underline{\tau}$ and $\overline{\tau}$ are the minimum and maximum steady state (static) torque bounds respectively, and $g$ is the gravity term.

- Joint limits:

$$\underline{\mathbf{q}}_i \leq \mathbf{q}_i \leq \overline{\mathbf{q}}_i \tag{2}$$

$\underline{\mathbf{q}}_i$ and $\overline{\mathbf{q}}_i$ are the upper and lower bounds for the robot $i$. Of course, the range of the joint limits for any $i$ are the same for a given joint, but $\mathbf{q}$ is ordered differently for each robot $i$ because of the change of reference base.

- Self-collisions:

$$\delta(X^i_j(\mathbf{q}_i), X^i_k(\mathbf{q}_i)) > \varepsilon_{jk} \quad \forall (j,k) \in \mathscr{I}^i_{\text{self-collisions}} \tag{3}$$

$\delta$ is the distance function, $X^i_l(\mathbf{q}_i)$ is the volume occupied by the $l$-th body of robot $i$ in configuration $\mathbf{q}_i$, $\varepsilon_{jk}$ is the user-defined minimum distance for pair $(j, k)$, and $\mathscr{I}^i_{\text{self-collisions}}$ the set of self-collision pairs to avoid for robot $i$.

- Other collisions:

$$\delta(X^i_j(\mathbf{q}_i), X_k) > \varepsilon_{jk} \quad \forall (j,k) \in \mathscr{I}^i_{\text{robot-environment}} \tag{4}$$

$\mathscr{I}^i_{\text{robot-environment}}$ the set of robot-environment collisions to avoid.

- Non-sliding contacts:

$$\mu_j N^i(\mathbf{f}_i, j) > \|TB^i(\mathbf{f}_i, j)\|, \ \forall j \in \mathscr{I}^i_{\text{contact}} \tag{5}$$

$\mathscr{I}^i_{\text{contact}}$ the set of contact points at $i$, $\mu_j$ is the friction coefficient at the contact point $j$, $N^i(\mathbf{f}_i, j)$ the $j$-th normal force component, $TB^i(\mathbf{f}_i, j)$ the tangent force vector components.

- Fixed contacts:

$$\begin{aligned}
& r^i_j(\mathbf{q}_i) - r_k = \mathbf{0} \\
& N^i_j(\mathbf{q}_i) \cdot T_k = 0 \\
& N^i_j(\mathbf{q}_i) \cdot B_k = 0 \\
& B^i_j(\mathbf{q}_i) \cdot T_k = 0 \\
& N^i_j(\mathbf{q}_i) \cdot N_k \geq 0 \\
& B^i_j(\mathbf{q}_i) \cdot B_k \geq 0
\end{aligned} \tag{6}$$

where the subscript $k$ stands for environment surface and $j$ the robot surface.

- Planar contacts:

$$\begin{aligned} &(r^i_j(\mathbf{q}_i) - r_k)\cdot N_k = 0\\ &N^i_j(\mathbf{q}_i)\cdot T_k = 0\\ &N^i_j(\mathbf{q}_i)\cdot B_k = 0\\ &N^i_j(\mathbf{q}_i)\cdot N_k \geq 0\\ &\mathrm{conv}(\mathscr{P}_j) \subseteq \mathrm{conv}(\mathscr{P}_k) \end{aligned} \tag{7}$$

where $\mathscr{P}_j$ and $\mathscr{P}_k$ are the surface of points $j$ and $k$, conv is the convex hull.

- Cylindrical contacts:

$$\begin{aligned} w_{\min} \leq (r^i_j(\mathbf{q}_i) - r_k)\cdot T_k &\leq w_{\max}\\ (r^i_j(\mathbf{q}_i) - r_k)\cdot B_k &= 0\\ (r^i_j(\mathbf{q}_i) - r_k)\cdot N_k &= 0\\ T^i_j(\mathbf{q}_i)\cdot B_k &= 0\\ T^i_j(\mathbf{q}_i)\cdot N_k &= 0\\ T^i_j(\mathbf{q}_i)\cdot T_k &\geq 0 \end{aligned} \tag{8}$$

where $w_{\min}$ and $w_{\max}$ are the width of the surface.

- Link all the common contacts of the robot:

$$\begin{aligned} r^i_k(\mathbf{q}_i) - r^j_k(\mathbf{q}_j) &= \mathbf{0}\\ \mathrm{Err}(E^i_k(\mathbf{q}_i), E^j_k(\mathbf{q}_j)) &= \mathbf{0} \end{aligned} \quad \forall (i,j,k) \in \mathscr{I}_{\text{common}} \tag{9}$$

this equation establishes the connections between contact transitions. See Appendix 1 for the computation of Err.

We use the cost function $\mathscr{C} = \sum_{i=1}^{N} \mathscr{C}_i$, where for each robot $i$:

$$\begin{aligned} \mathscr{C}_i(\mathbf{q}_i, \mathbf{f}_i) = w_q \|\mathbf{q}_i - \mathbf{q}^d_i\|^2 &+ \sum_{j \in \mathscr{I}_{\text{contF}}} w_j \|F(\mathbf{f}_i, j)\|^2\\ &+ \sum_{j \in \mathscr{I}_{\text{contT}}} w_j \sum_{p \in \mathscr{I}_{\text{points}_j}} \|M_j \cdot (r_p \times F(\mathbf{f}_i, p))\|^2\\ &+ \sum_{j \in \mathscr{I}_{\text{posT}}} w_j \|r^i_j(\mathbf{q}_i) - r^d_j\|^2\\ &+ \sum_{j \in \mathscr{I}_{\text{rotT}}} w_j \|\mathrm{Err}(E^i_j(\mathbf{q}_i), E^d_j)\|^2 \end{aligned}$$

All the $\mathscr{I}_x$ represent sets of $x$, all the $w_x$ are the cost weights for the cost part $x$, $\mathbf{q}^d_i$ is the target configuration vector, $F(\mathbf{f}_i, j)$ is the $j$-th force vector of $\mathbf{f}_i$, $M_j$ is the motor rotation axis vector, $r_p$ is the motor to point $p$ translation, $\mathscr{I}_{\text{points}_j}$ the set of contact points in contact $j$, $r^d_j$ and $E^d_j$ denote target positions and orientations respectively.

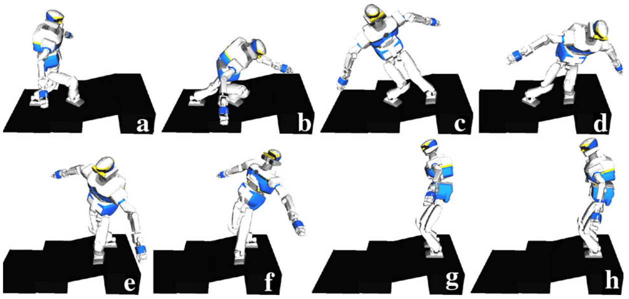


**Fig. 4** Example solution of MPG

Figure 4 illustrates a contact planning solution computed with the MPG for $N = 8$. We use an objective function to make the robot stand in the opposite direction for the last contact transition (h). As for the solver, we used IpOpt (Wächter and Biegleri 2006) with the RobOptim framework.[3] It computed this MPG problem in $\simeq$1 s (136 iterations). We do not use the free-flyer coordinates to model the kinematic tree. We use instead a fixed, planar or cylindrical contact as the base for each robot. By doing so, we remove one contact constraint at each contact transition (stance). For the problem illustrated in Fig. 4, the left foot constraint in stances (a) and (b) are fixed, and planar in all the remaining stances, whereas the right foot constraint is planar in all stances.

Notice that the Eq. (9) can be redundant with Eqs. (6), (7), and (8). For e.g., if the constraint Eq. (9) links the right foot between (b) and (c) it is then possible to remove the constraint Eq. (7) from one of the two previous stances.

In order to track these redundancies, we use Dijkstra's graph search algorithm to find the contact representation that minimizes the number of constraints. We model the constrains in an oriented graph whose vertices, with a unique identifier $j$, are a kinematic and contact constraint for the stance $i$. Each vertex is eventually connected to all its possible $i + 1$ children. To be valid, a path $P = \{v^1_j, \cdots, v^N_j\}$ must have at least one occurrence of each contact. The graph is colored by the number of constraints. For the example in Fig. 4, this tracking algorithm leads to write an equivalent problem of 64 contact constraints instead of 86.

## 4 Multi-contact finite state machine (FSM)

The output of the MCP is a sequence of static contacts and postures to climb the ladders, see Fig. 2. There are many reasons why this output cannot be given directly as tasks.

First, when using a task-space controller (see Sect. 5), assigning a target position (eventually with an orientation) for a frame attached to a given robot's link, would result in a 'straight line' motion. Collision avoidance might bias the

[3] http://www.roboptim.net/.

motion but could also lead to local minima. We solve this problem by adding way-points (one per motion is enough for ladder climbing). These waypoints are intermediate goals that do not need to be reached. Alternatively, one can also use real-time planners such as CHOMP (Zucker et al. 2013), which path is added as a task with a low priority objective.

Second, we assume that geometric discrepancies are unavoidable, therefore guarded motions on forces, velocities or positions near expected locations of contacts are needed.

Third, some contacts are created by grasps. Intermediate phases to reach or release grasps are needed.

Finally, postures that embed a desired CoM position and CoM transfer between stances need to be elaborated.

For all the previously cited reasons and possibly others (such as dealing with unforeseen problems), a multi-contact FSM is devised. To each contact transition of the MCP, we associate an action call $A^i$ that can be one of the followings:

- AC: Add a (unilateral) contact
- RC: Remove a (unilateral) contact
- MC: Move the CoM
- AG: Add a grasp contact
- RG: Remove a grasp contact

We also designate $B^i$ as the body to which the task frame is attached at contact transition (stance) $i$. $F_{B^i}$ is the most pertinent force acting on $B^i$ (measured or estimated). We use $\sigma^i_{\text{action}}$ to denote the threshold related to the measure in achieving *action* for stance $i$. For robustness and flexibility purposes we define any threshold to be specific to each stance $i$. In what follows we explain the FSM in details.

Figure 5 illustrates the implementation of the FSM for the ladder climbing. The *remove unilateral contact* RC is the leftmost branch which is achieved with successful transitions t1 and t2 as follows:

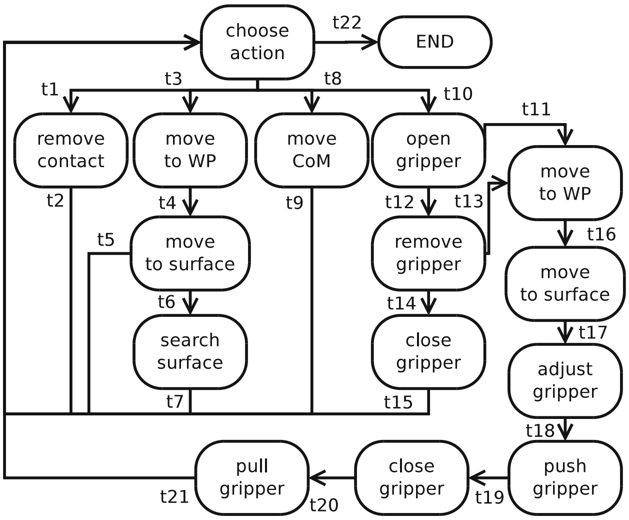


**Fig. 5** The current implementation of the FSM for ladder climbing

t1: $A^i \leftarrow$ RC

sets $A^i$, which results in building a set of tasks to be achieved by the controller (Sect. 5) in order for $B^i$ to remove its current contact state progressively, with a force guarded motion. We assume the contact is removed when:

t2: $F_{B^i} < \sigma^i_{\text{RC force}}$

i.e. the contact force $F_{B^i}$ might not be exactly 0.

The *add unilateral contact* action AC starts by:

t3: $A^i \leftarrow$ AC

Then a move to way-point task is built and passed to the controller. This task is assumed to be achieved when:

t4: $\varepsilon(X_{B^i}, X^i_{\text{way-point}}) < \sigma^i_{\text{way-point}}$

The way-point $X^i_{\text{way-point}}$ for $B^i$. $\varepsilon$ is the measure of the error to target (here, the way-point): this condition means that $B^i$'s position and/or orientation reached the way-point with a threshold $\sigma_{\text{way-point}}$.

Once t4 is done, a force, position and velocity guarded go-to-contact task is built and added to the controller. Two situations may then occur:

(1) the contact is encountered before reaching the expected location (on the ladder); this would correspond to transition:

t5: $F_{B^i} > \sigma^i_{\text{AC force}}$

(2) the expected position of the contact is reached with a small speed (and obviously with $F_{B^i} \simeq 0$). This situation is a *go near to contact* task and the additional speed check is due to warrant robust implementation, that is:

t6: $\varepsilon(X_{B^i}, X^i_{\text{AC}}) < \sigma^i_{\text{AC}}$ and $\|v_{B^i}\|_2 < \sigma^i_{\text{AC velocity}}$

This transition leads to search for contact surface tasks (even strategies) to be passed to the controller until:

t7: $F_{B^i} > \sigma^i_{\text{AC force}}$

which means that the contact is found and that the AC task at stance $i$ is done.

The next branch in the FSM tree represents the *move CoM* or *CoM transfer task* MC, which is triggered by:

t8: $A^i \leftarrow$ MC

Now the CoM transfer task is built and passed to the controller; it is assumed done when:

t9: $\varepsilon(r^i_{\text{CoM}}, r^i_{\text{CoM}_d}) < \sigma^i_{\text{MC}}$ and $v^i_{\text{CoM}} < \sigma^i_{\text{MC}}$

where $r^i_{\text{CoM}_d}$ is the CoM target computed from the stance $i$.

The remaining two branches concerns the tasks *remove a grasp contact* RG or *make a grasp contact* AG respectively. Both tasks need to be set by the FSM first, that is:

t10: $A^i \leftarrow$ AG or $A^i \leftarrow$ RG

In both cases, the FSM would trigger an open gripper task. Following this, we have either:

t11: Gripper opened and not in contact and $A^i =$ AG

which leads to the branch of grasp rung or stringer or handrail. or,

t12: Gripper opened and gripper in contact

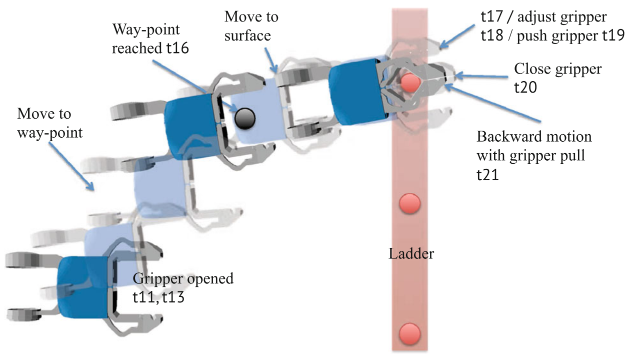


**Fig. 6** Add gripper contact (grasp) task: illustration of the different phases of the associated FSM AG action. Target orientation frames are not represented for the clarity of the figure

which leads to the branch of releasing grasp. In this case, two transitions are possible for the gripper in question:
(1) either the next action involves a grasp with this gripper, which then means:
t13: Gripper removed and ($A^{i+1} = \mathtt{AG}$ and $B^i = B^{i+1}$), or
(2) the next action does not involve a grasp with this gripper, which means:
t14: Gripper removed and ($A^{i+1} \neq \mathtt{AG}$ or $B^i \neq B^{i+1}$)
in this case, the gripper is closed and leads to transition: t15: Gripper closed.

The last branch (that is AG at transition t11 or t13) is dedicated to grasping (the steps are illustrated in Fig. 6). We build a move to way-point task, add it to the controller, and assume that we reach the way-point when:
t16: $\varepsilon(X_{B^i}, X^i_{\text{way point}}) < \sigma^i_{\text{way-point}}$ and $\|v_{B^i}\|_2 < \sigma^i_{\text{way-point}}$
When the way-point is reached, we launch a go-to-contact task, which is assumed fulfilled when:
t17: $\varepsilon(X_{B^i}, X^i_{\text{AG}}) < \sigma^i_{\text{AG}}$
This is followed by adjusting the gripper, which is fulfilled when:
t18: $\varepsilon(\tilde{X}_{B^i}, X^i_{\text{AG}}) < \sigma^i_{\text{AG}}$ and $\|v_{B^i}\|_2 < \sigma^i_{\text{AG velocity}}$
where $\tilde{X}_{B^i}$ is the moving body transformation estimated with the compliance (Sect. 5.3). This task is achieved under guarded force motion for the transition condition:
t19: $F_{B^i} > \sigma^i_{\text{AG push}}$
This means that the contact with an open-gripper is made with a rung or a stringer or a handrail. We can then close the gripper and check the next transition:
t20: Gripper closed
Once the gripper is closed, we sustain the contact by a guarded force backward motion which ends when:
t21: $F_{B^i} < \sigma^i_{\text{AG pull}}$
All previous AG action steps are illustrated in Fig. 6.

When the climbing ends, the FSM state switches to:
t22: $i \geq N$.
Otherwise, the next stacked action is chosen.

## 5 Multi-objective quadratic program controller (QP)

Tasks from Sect. 4 need to be transformed into joint motions under various constraints, which turn to be additional tasks. Our controller is formulated as a model-based QP.

Current trends in task-space control prioritize tasks (i) in a weighted least-squares form, or (ii) in a strict hierarchy with equality or inequality constraints, or (iii) a mix of both.

A weighted priority formulation expresses as hard constraints the tasks that are already fulfilled, and as part of the cost function, those that are not yet achieved. Hierarchy among the tasks in the cost function is made through their weighting (task gains). Such an approach was proposed in computer animation (Abe et al. 2007), where standing and balancing with legs was demonstrated with unilateral contacts and under various kind of perturbations. In Collette et al. (2007), they used a two-level cascade of QP controllers (the first computes static postures, part of which is use by the second for a dynamic balance computation); this work includes also grasps and other more complex contact tasks. In Liu et al. (2012) a passivity guaranteed formulation is suggested together with task force predictions. These works were developed for virtual characters and did not implemented collision avoidance in the controller.

Application to humanoids in simulation is proposed for momentum based balance control (Lee and Goswami 2012). There is also the remarkable work in Salini et al. (2010, 2011) considering torque and state limits. In Bouyarmane and Kheddar (2011) collision avoidance is introduced and the QP controller is used to control multi-contact whole-body non-gaited motion (Bouyarmane et al. 2012). Recently, in Kuindersma et al. (2014) a QP formulation of the ZMP-based walking is proposed with an efficient fast resolution of the problem. None of the previous works experienced such controllers on a real humanoid robot. This is what we achieve here. Hyon et al. (2007) and Ott et al. (2011) used a force control formalism for balancing in multi-contact configuration with experimentations conducted on torque based controlled humanoid robots, yet without contact transitions.

Strict-prioritized controllers based on null-space projectors are the heritage of early works by Liégeois (1977), Nakamura et al. (1987) and Siciliano and Slotine (1991). Application to humanoid robots in simulation using the operational space formulation of the dynamics was illustrated in Sentis et al. (2010) and experimented on a wheeled humanoid torso in Sentis et al. (2013). Prioritized mixed equality and inequality constraints was integrated successfully in Mansard et al. (2009) using the null-space operator projection, whereas Kanoun et al. (2011) used a cascade of QP. A full control framework using the hierarchical QP (Escande et al. 2014) was experimented on the HRP-2 humanoid in open-loop (Saab et al. 2012). On the contrary, our work is the first to experience a task-based formalism on HRP-2 using dynamics in a closed-loop control. In Righetti and Schaal (2012) the control was explicitly derived to optimize contact forces. Experiments on a quadruped robot is reported in Righetti et al. (2013), and on a torque-based legged robot in Herzog et al. (2014). In Wensing and Orin (2013) a conic programming formulation of prioritized task space control using dynamics but without discretizing friction cones is proposed and shown to be twice as fast as a QP. In Lasa et al. (2010) a mix of projection and QP formulations is proposed. It was applied mainly for the control of walking and jumping of different virtual characters and used efficiently in Mordatch et al. (2010).

The previous controllers do not produce any anticipatory behaviors considering up-coming tasks. Task-based Model Predictive Control was firstly considered for stylized human locomotion (Silva et al. 2008) using motion capture. Recently Ibanez et al. (2014) tackled this problem in a more general weighted prioritized formulation and did an excellent review of the task-based approaches and how they relate to basic optimization schools. In Audren et al. (2014) a model-preview controller for general multi-contact motion is proposed using a reduced model (CoM) preview written also as a QP. Multi-contact whole-body non-linear formulation, see e.g. Lengagne et al. (2013) and Posa et al. (2014), does not yet meet time computation requirements. In practice, our investigations revealed that a preview does not bring any substantial added value (in terms of performance) given the time taken in guarded motions for contact formation and removal and the relatively slow transfer motion of each limb for safety reasons.

In climbing, all critical tasks that constitute the QP's constraints such as non-sliding contacts, equilibrium, state variables limitations, and non-desired collisions, are critical and—in fact, have the same priority. Other tasks can be weighted in the cost function to be achieved at best. Therefore, strict-hierarchy priority may end up with a two-priority problem and is not substantially superior w.r.t a weighted priority QP. In particular, our pilot experiments show that we often go to joints or reachability limits where strict prioritized formalism is known to not behave well.

### 5.1 Model-based QP multi-contact controller

We redesigned the weighted-task QP framework developed in Bouyarmane et al. (2012) to achieve real-time performance and be efficiently implemented as a low-level controller. The data we need for the QP are similar to those used in the planning. The tasks are formulated as linear constraints or quadratic costs and the QP is solved at each $dt$. The optimization variables are composed of $\mathbf{x} = [\ddot{\mathbf{q}}^T, \boldsymbol{\lambda}^T]^T$ where $\ddot{\mathbf{q}}$ is the joint acceleration vector and $\boldsymbol{\lambda}$ is the vector of linearized friction cones' base weights. The vector of contact forces $\mathbf{f}$ is equal to $K\boldsymbol{\lambda}$ where $K$ is the discretized friction cone matrix. We do not make any distinction between the robot joint and the free-flyer non actuated coordinate. The desired acceleration $\ddot{\mathbf{q}}$ is integrated twice to feed the low level built-in PD control of HRP-2. We do not make use of the force $\mathbf{f}$ and the torques that can be obtained from the QP solution. The QP controller writes:

$$\underset{x}{\text{minimize}} \sum_{i=1}^{N} w_i \| E_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \|^2 + w_\lambda \|\boldsymbol{\lambda}\|^2$$

subject to

$$(1)\ \underline{\tau} \leq M(\mathbf{q})\ddot{\mathbf{q}} + N(\mathbf{q}, \dot{\mathbf{q}}) - J^T\mathbf{f} \leq \overline{\tau}$$

$$(2)\ S(J_i\ddot{\mathbf{q}} + \dot{J}_i\dot{\mathbf{q}}) = -S\frac{v_i}{dt}\ \ \forall i \in \mathscr{I}_{\text{contact}}$$

$$(3)\ \max\left(\underline{\dot{\mathbf{q}}}, \xi\frac{(\mathbf{q} - \underline{\mathbf{q}}) - q_s}{q_i - q_s}\right) - \dot{\mathbf{q}} \leq \ddot{\mathbf{q}}dt$$

$$(4)\ \ddot{\mathbf{q}}dt \leq \min\left(\overline{\dot{\mathbf{q}}}, \xi\frac{(\overline{\mathbf{q}} - \mathbf{q}) - q_s}{q_i - q_s}\right) - \dot{\mathbf{q}}$$

$$(5)\ \dot{\delta} + \ddot{\delta}dt > \xi\frac{\delta - \delta_s}{\delta_i - \delta_s}$$

Constraint (1) accounts for the torque bounds $\underline{\tau}$ and $\overline{\tau}$, using the dynamic equation in which $M(\mathbf{q})$ is the whole-body inertia matrix, $N(\mathbf{q}, \dot{\mathbf{q}})$ is the non-linear Coriolis and Gravity vector and $J$ is the contact points Jacobian matrix.

Constraint (2) enforces zero acceleration for the bodies that are in contact (no-sliding). In all the previously cited works, this constraint writes rather as $J_i\ddot{\mathbf{q}} + \dot{J}_i\dot{\mathbf{q}} = 0$ . $J_i$ is the translation and rotation Jacobian of the body $i \in \mathscr{I}_{\text{contact}}$. In practice, we noticed that countering the contact body velocity $v_i$ leads to a better numerical behavior. Indeed, the controller is computed on the basis of a simulated robot model. We also added $S \in \mathbb{R}^{n,6}$, a selection matrix that allows one to free directions to be eventually controlled in impedance.

Constraints (3) and (4) enforce joint speed and range limits and use a velocity damper $\xi$ (Kanehiro et al. 2010), $q_s$ as a security range, and $q_i$ as the interactive (triggering) threshold.

Constraint (5) deals with collision avoidance (that we integrate in the controller instead of checking *a priori* or *a posteriori*). Relatively to Kanehiro et al. (2010) we 'track' one witness point per link or body when paired for collision checking. $\delta$ is the distance between a pair of bodies computed with the SCH library (Escande et al. 2014).[4] $\dot{\delta} = N^T J\dot{\mathbf{q}}$ and $\ddot{\delta} = \dot{N}^T J\dot{\mathbf{q}} + N^T(\dot{J}\dot{\mathbf{q}} + J\ddot{\mathbf{q}})$. $N$ is the normal (distance) vector (that is straightforwardly determined from the witness points if $\delta > \sigma_\delta$ ($\sigma_\delta$ is a user defined threshold), or from the surface's normal of one of the two witness points. $\dot{N}$ is computed by finite difference. Our QP controller

[4] Available at https://github.com/jrl-umi3218/sch-core.

computes collision avoidance constraints in real-time. However, if many collision pairs are active at any same time, we noticed a bad computational behavior of the QP. As an ad-hoc solution the interaction (i.e. triggering) distance $\delta_i$ can be adapted on-line to be different for each pair of bodies so that not many distance constraints are active at the same time.

The QP objective function is made of two terms: (i) the task errors appear in a sum of weighted least-squares term, noted $E_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$, and (ii) a damping term with weight $w_\lambda$ which ensures that the Hessian matrix is positive definite. For ladder climbing we only use the Set Point objective task (Abe et al. 2007; Lasa et al. 2010; Bouyarmane and Kheddar 2011) written as:

$$J_{\mathscr{T}_i}\ddot{\mathbf{q}} + \dot{J}_{\mathscr{T}_i}\dot{\mathbf{q}} + 2\sqrt{k_i}\dot{\mathscr{T}_i} + k_i\mathscr{T}_i \qquad (10)$$

with $\mathscr{T}_i \in \mathbb{R}^n$ an $n$-dimensional task error, and $J_{\mathscr{T}_i}$ its associated Jacobian. We use the following tasks:

- Posture task: $\mathscr{T}_{\text{posture}} = \mathbf{q}_d - \mathbf{q}$
- Body $i$ position task: $\mathscr{T}_{\text{position}} = r_i^d - r_i(\mathbf{q})$
- Body $i$ orientation task: $\mathscr{T}_{\text{orientation}} = \text{Err}(E_i^d, E_i(\mathbf{q}))$
- Body $i$ linear velocity task: $\mathscr{T}_{\text{linear-velocity}} = v_i^d - v_i(\mathbf{q}, \dot{\mathbf{q}})$
- CoM task: $\mathscr{T}_{\text{CoM}} = \text{CoM}_d - \text{CoM}(\mathbf{q})$

For $\mathscr{T}_{\text{orientation}}$, see Appendix 1. In constraints (3), (4) and (5) velocity dampers $\xi$ can cause large deceleration and may lead to problems if not well tuned. We use the following expression to compute $\xi$ once (i.e when the constraint is firstly triggered and the velocity damper is activated):

$$\xi = \frac{d_i - d_s}{d - d_s}\dot{d} + \xi_{\text{offset}} \qquad (11)$$

where $d$ is the distance between any constraint and its nearest bound, $d_i$ is the interactive (triggering) distance, $d_s$ is the security distance, $\xi_{\text{offset}}$ is a fixed offset enabling the velocity damper to accelerate a bit and avoid over-constraining the problem. This allows us having a damping coefficient that is adapted to the current velocity.

### 5.2 QP solver

The QP controller is built at each $dt$, we can either use an off-the-shelf QP solver or develop our own. We have favored the first option for robustness and fast development time reasons. From a quick review of the literature, trials of common (free) solvers, and discussions we initiated with several community researchers we decided to benchmark two QP solvers: LSSOL (Gill et al. 1986) (cold and warm start) and QLD (Schittkowski 1986).

We benchmarked the two QP solvers with two scenarios that are representative of the complexity of the climbing: (1) a CoM transfer with four contacts, and (2) a leg transfer with three contacts. The tasks specifications in terms of optimization variables and other parameters are described in Table 1.

We used an i7 2.6GHz laptop (see later Fig. 13). As one can see in Fig. 7, the LSSOL warm start is substantially superior to the remaining two (LSSOL cold start and QLD) for the leg transfer task; but performances, although best, are less pronounced for the CoM task. Therefore we adopted LSSOL to be our QP solver. During our experiments (Sect. 6.2), we noticed that the median computation time of the whole problem (cost function, constraint matrix, distance query and QP solving) for the ladder climbing (3 or 4 contacts) is ≃1 ms.

**Table 1** Tasks specifications for QP solvers benchmarking

| | CoM transfer | Leg transfer |
|---|---|---|
| Number of contacts | 4 | 3 |
| Optimization variables | 100 | 84 |
| Degrees of freedom used | 36 | 36 |
| Size of λ (force) | 64 | 48 |
| Collisions constraints | 54 | 54 |
| Min number of constraints | 95 | 91 |
| Max number of constrains | 98 | 100 |

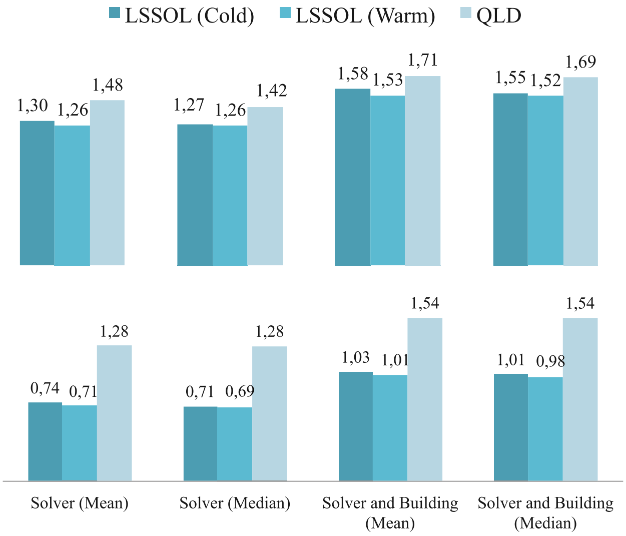


**Fig. 7** Benchmarking the QP solvers LSSOL (cold and warm start) and QLD for a four contact transfer task (*first line*) and the leg transfer task (*second line*), see also Sect. 6.2. The computation time, in milliseconds, is given on the *top* of each *bar*. We represent mean and median times for the solver alone (i.e. with the problem already set) and the solver plus the QP building, which gathers other computations such as the dynamics, Jacobians, distance, etc. Notice the superiority of the LSSOL warm start w.r.t cold start and QLD (Color figure online)

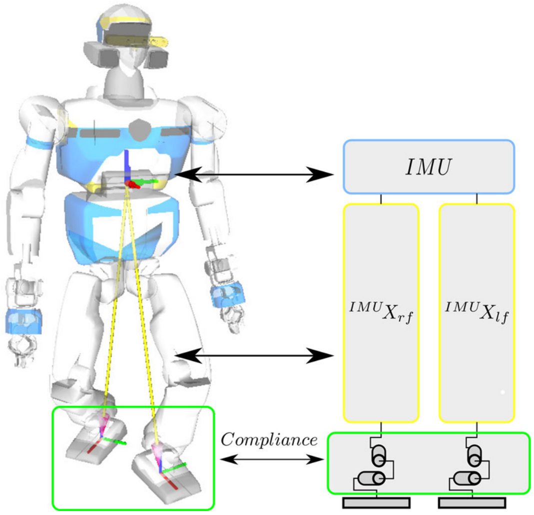


**Fig. 8** Compliance kinematic model

### 5.3 Dealing with ankle shock absorbing compliance

At each feet of many humanoid robots, there is a shock absorbing compliant mechanism. It prevents the force sensor from malfunctioning and from breaking should high impacts occur. Moreover, compliance is important to absorb light discrepancies at contact formation/removal or during multi-contact motions; hence, it has also a *stabilizing effect*. Unfortunately, this compliance, since passive, makes the attitude of the robot hard to control. For example, the HRP-2 has a built-in stabilizer to counter the compliance effects. Alas, it is tuned only for walking on flat terrains and assumes coplanar contacts. Therefore, it has to be omitted in climbing or any non-coplanar multi-contact motion.

To compensate for the ankles' compliance, we estimate its effect using the robot embedded inertia measurement unit (IMU) and inverse kinematics. We substitute each compliance with two revolute passive joints, see Fig. 8. Each leg (contact) is modeled as a fixed base with 2 dof and an end effector passing through the IMU. Since in our experiments, at least two contacts always hold, we are always having at least one closed-kinematic chain between contacts. We exploit this fact to estimate the 4 virtual joints' values. In the example illustrated in Fig. 8, we consider all the open kinematic chains that go from the IMU frame, and write the conditions to close the kinematic chain in position, and secondary (at best) in orientation considering the least possible motion. Yet, we assume that the joint encoders and the IMU measures are reliable, and that we know each contact type (e.g. planar, cylindrical).

Let $\mathbf{q}_{\text{model}}$ be the robot configuration used by the controller and $\mathbf{q}_{\text{estimate}}$ be the robot configuration that include the compliance virtual joints. We made first trials using the compliance estimation as a task, $\mathscr{T}_{\text{position}}$ and $\mathscr{T}_{\text{orientation}}$, where the position and the orientation error and velocity are computed from the $\mathbf{q}_{\text{estimate}}$. As a result, the robot tried fixing its position without counterbalancing the compliance's dynamics, and falls or oscillate. Instead, we reduced the dynamics of the motion with the following definition of the task error (that would apply to any position and orientation task $\mathscr{T}$):

$$\mathscr{T} = K_p \mathscr{T}(\mathbf{q}_{\text{model}}) + K_i \int_{T_i}^{T_e} \mathscr{T}(\mathbf{q}_{\text{estimate}}) dt \qquad (12)$$

where $K_p \gg K_i$, $T_i$ and $T_e$ are the task insertion and removal times, which allows converging to a zero error with slower dynamics. See the illustration in Fig. 19, Sect. 6.3.

### 5.4 Gripper/rung contacts

The ankle's compliance can cause the robot to lose the grip of the rung. Since we can release the null velocity constraint on any chosen axis (thanks to the selection matrix $S$ in Sect. 5.1), we added a simple force scheme: when the force goes below $f_{c_\sigma}$ we relax the null velocity constraint on the insertion axis of the gripper's (the $z$-axis), and then add a target position task with a high weight as follows:

$$z_{\text{target}} = z_{\text{init}} + \min(\kappa(f_{c_\sigma} - f_c), z_{\max}) \qquad (13)$$

$z_{\text{init}}$ is the initial position of the contact, $z_{\max}$ is the maximum displacement of the contact, $\kappa$ is a unit converting gain.

## 6 Experiments and results

### 6.1 Simulated scenarios

Figure 9 illustrates the climbing of the ladder used in our real experiments. In this set-up, we put an obstacle (a long tube) traversing the ladder. This tube would induce a change in motion of the left arm when grasping the fifth rung, and that of the left leg transfer from the first to the third rung. These motions are different from those produced in the absence of the obstacle. Also, in the absence of the seventh rung, the MCP manages to find a combination of rung and stringer grasps for the last phase of the climbing.

Figure 10 shows the computation time of the QP controller (building blocks plus LSSOL warm start solver) for the simulation illustrated in Fig. 9. We observed similar timing in our experiments.

The accompanying multimedia for this simulation is annotated with the contact forces, actuator torques at the grippers, the COM and the grasp or contact objectives, the distance (only the two most pertinent ones to avoid overloading the video) computed between the robot limbs and the tube, and

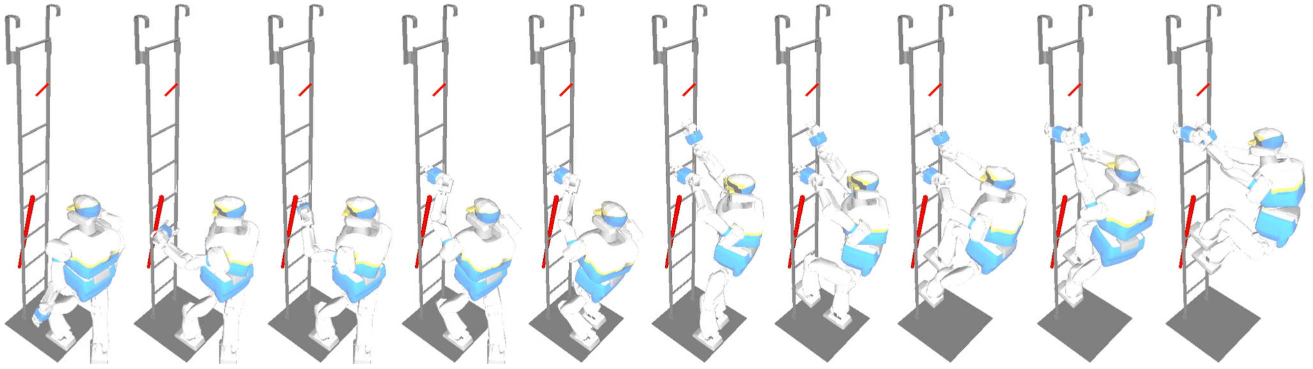


**Fig. 9** Snapshots from simulated climbing of the *vertical* ladder used in real experimentations with a *red* tube added as obstacle (Color figure online)

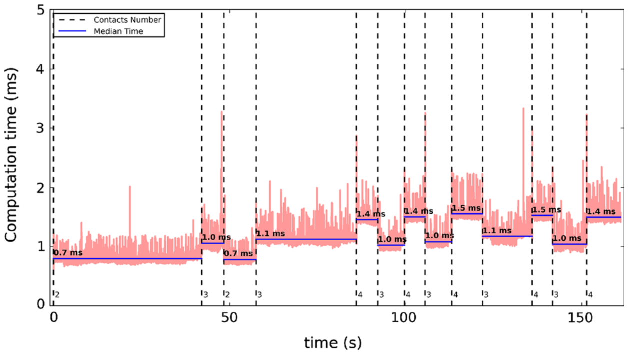


**Fig. 10** The QP controller computation time, which is below the HRP-2's 5 ms control-loop. The *dots vertical lines* show contact transitions; the number of contacts is mentioned nearby each *line*

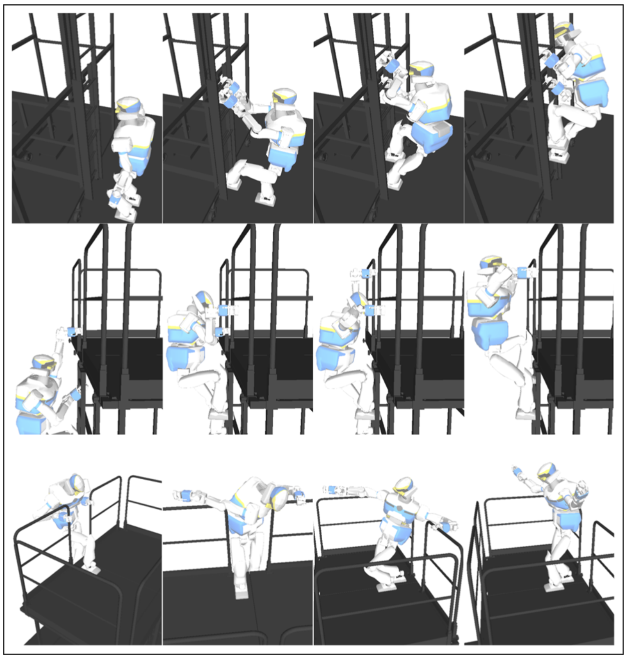


**Fig. 11** The *first line* shows postures from the ladder climbing. The *second line* shows the ladder-to-scaffolding transition. In the *third line* the HRP-2 reached the scaffolding

also part of the FSM tasks during the climbing process. The simulation uses physical-based animation and we could emulate uncertainties in the position of the robots and the objects to assess the FSM and the guarded motion tasks prior to real experiments on the robot.

Figure 11 illustrates the MCP plan obtained from a simulated scenario of a real set-up that is available at our experimental room. The ladder and scaffolding settings are modeled with a precision of 1mm. This simulation assumes firm grasps on the rungs and the stringers. Note that the MCP found transit strategies from the ladder to the scaffolding via the narrow passage (kept with similar dimension as those found in industry). The ladder climbing, ladder-to-scaffolding transition and scaffolding reaching phases are made at once. Yet, from the many simulations we made, not all generated contact plans where successfully reproduced by the QP. Also, it took more time for the planner to find the ladder-to-scaffolding transition plan.

### 6.2 Experiments with HRP-2

For the experiments, we used a ladder whose parameters are represented in Fig. 12. The ladder consists of eight rungs. The last one cannot be used because it is too close to the gantry crane and the roof. The ladder is hooked to the gantry crane and fixed to the floor. The HRP-2 is set to a precomputed initial posture near the ladder. All our experiments are performed without triggering the HEP-2 native stabilizer. Instead, we use the ankle compliance compensation described in Sect. 5.3. Since the ankles's compliance is compensated in the QP closed-loop control, the robot can reach the first rung with the gripper; without it, the robot falls sideways or backwards as soon as the arm starts moving.

Our control architecture is split on two computers: (i) the one that runs the QP Controller and the compliance estima-

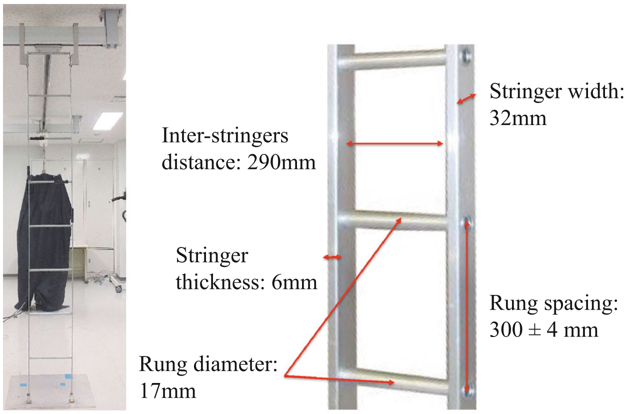


**Fig. 12** The *vertical* ladder (*left*) and its parameters (*right*) used in HRP-2 climbing experiments

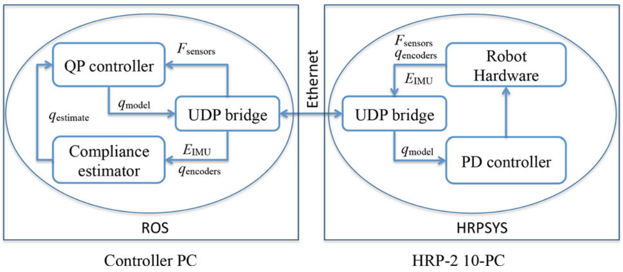


**Fig. 13** The hardware and software architecture. The QP controller includes the FSM. The robot hardware is a process that reads and filters the robot sensors data. The UDP bridge is a process allowing the two computers to communicate through an Ethernet link

tor; and (ii) the HRP-2 on-board computer that samples the measurements of the sensors and that runs the PD controller.

The two computers communicate with a direct Ethernet link and exchange data with a UDP network bridge, see Fig. 13. The QP controller is developed using the ROS middleware.[5] The controller part is written in C++ and the FSM part is in Python.[6] The FSM and the controller part run on the same process, it is convenient for fast prototyping and debugging.

We choose to use two computers because the current HRP2-10 on-board computer is not powerful enough to run the controller with the ROS framework. Thus, by running the control software on an external portable computer we are able to monitor more easily the robot. This architecture also allows us to run the same controller on our HOAP-3 and HRP-4 humanoids.

We report the main results obtained from different experiments with the HRP-2 climbing the vertical ladder in Fig. 12. The first problem we faced was to secure the robot during our trials. The strings attaching the robot to the gantry crane (XY−Z roof trail) were not easy to operate in these conditions, but we managed to find proper adjustments that minimize damage in accidental or malfunction situations. We also developed debugging tools and intermediate sequential steps validations. Before achieving a complete autonomous climbing, we went through different assessment phases. In all cases, and prior to any experiment, we cancelled the recovery parts of the FSM, assuming the contacts to occur as expected, and played the entire climbing motion with the robot floating. This step was useful to confirm that the motion was indeed doable without self-collision.

As the width of the ladder does not enable having both feet on one rung, climbing was made in two main phases:

1. Arm transfer (removal and creation of a grasp), which is always made while maintaining three contacts (the two feet plus one grasp).
2. Leg transfer (removal and creation of a leg contact), which is coupled with whole-body lifting and always made with two grasp contacts.

This strategy is somehow similar to the transverse mode in Yoneda et al. (2008). It yet differs from those chosen for the customized climbing robots in Iida et al. (1989) and Fujii et al. (2008).

– *Preliminary trials: grasps on stringers*
First, we examined a climbing policy that uses grips on the stringers, and we forced the grasping areas to be nearby (up) the rungs, see Fig. 14. This choice prevents the gripper from sliding downwards. We generated a multi-contact plan, which contact stances are passed to the QP. The latter generated joint trajectories in simulation. These trajectories are then executed in open-loop by the robot. To keep a perfectly calibrated environment, it was the duty of the user to 'close the loop' by adjusting the robot when needed by direct touch, see Fig. 14. Our aim was to check (i) the capability of the grippers to hold contacts on the stringers, and (ii) the capability of the robot to lift its body by the strength of legs and arms. As a result, we confirmed the following, see Vaillant et al. (2014):

1. one gripper was not capable to hold a grasp on a stringer when the other one releases its grasp: this is due to the lack of firm grabs and grip power;
2. in a four contact configuration, the robot was able to lift its body without any noticeable problem.

These first experimental trials confirmed that with the help of the human operator (adjusting the contact of the grippers with the stringers during limb transfer and recovering discrepancies), the HRP-2 humanoid robot is capable of climbing the ladder, see detailed comments in Fig. 14.

[5] http://wiki.ros.org/.

[6] The code for dynamic computation can be found at https://github.com/jorisv/RBDyn. The code of the controller will be made available soon.

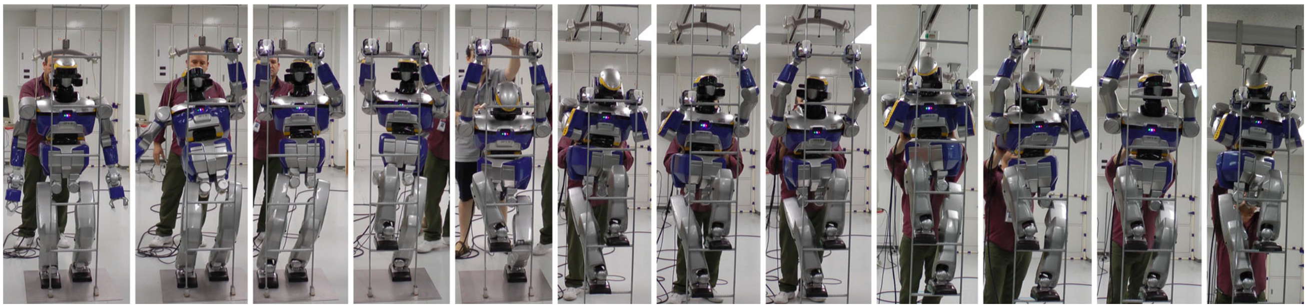


**Fig. 14** The ladder climbing with an off-line generated trajectory. Adjustments of the grippers on the stringers is made when needed by the operator (behind the robot) using light pushes toward the ladder. In this experiment, the user compensates for the lack of firm grasps of the stringers, but does not intervene when the two grippers are in grasp (i.e. the four contact lifts of the HRP-2 body and the feet transfer or positioning cases). In this trial, the HRP-2 could climb four rungs (the maximum possible, as the head would reach the roof and the protection ropes cannot be tightened)

– *Trials with grasps on rungs*

The second climbing policy uses rungs only. The rung diameter is greater than the stringer width. We also increased the gains of the PD controller for each gripper's actuator. In this trials we challenged a fully autonomous climbing in close-loop control with the use of the FSM.

First, the HRP-2 climbs up until both feet left the ground to be on the ladder and then climb down –by reversing the plan. This was achieved successfully and repeatedly without any intervention from the user. The accompanying multimedia shows this case, which is illustrated by the two first snapshots of Fig. 1. Notice that in this case, the robot grasps the fifth rung with left arm, then the sixth rung with right arm, put left leg on the first run, then lifts whole body while positioning the right leg on the second rung.

After we assessed this experimental step, we attempted to go further by repositioning the left arm then the left leg. This is shown by the third and fourth image in Fig. 1.

But we faced several problems that we circumvented by ad-hoc solutions since their common cause was the limitations due to the grippers design (see Sect. 6.3).

The first problem is that the release of the left gripper induces a light rotation of the robot around (approximately) the median vertical axis of the ladder. This is due to the fact that the contacts are (not only coplanar but also) nearly collinear, and, as for the stringer, the rung is still not firmly grabbed. We could compute a posture that minimizes the moment around that axis. In fact, having a light rotation wouldn't be a problem, if not for the next problem.

The second problem is due to kinematic reachability limits. Since we use only rungs and only one foot can be put on a rung, the HRP-2 can barely reach the last rung, but not enough for the FSM to confirm the contact and to close the gripper (condition `t19` cannot be achieved). The problem, as can be seen from the third image in Fig. 1, is that the left leg and arm are completely stretched, where as the right leg is fully bended with the knee touching the third rung. Therefore, no more motion is possible toward the rung by the left arm. We circumvented this intrinsic hardware limiting problem by allowing –for this step only, the user to close the gripper by a keyboard instruction.

Finally, a third problem is that in this configuration, the gripper cannot hold the closing during the last left leg transfer, which also comes with another robot lift. We circumvent this limitation by asking another person to maintain (using his hands) the gripper closed during the left leg transfer. By two punctual adjustments, we could achieve the complete cycle of climbing as illustrated in Fig. 15.

Figure 16 shows, as ground truth, the normal forces computed by the QP controller, and those measured from rough force sensing in the wrists and feet, i.e. without off-set calibration nor filtering, since we do not use force data in the control loop. Moreover, the changes on the HRP-2's inertia w.r.t the factory model are not considered here. Yet, one cans see that the QP controller predicts a plausible choice of force distribution. These results are extremely encouraging for future work. Indeed, the reliability of predicted contact forces would allow exploiting them for posture adjustment, for internal forces reduction and balance, and for on-line fault or problem detection from force discrepancies monitoring.

Because of the lack of heat and torque monitoring of the actuators, we used an infra-red camera for monitoring. Figure 17 displays snapshots of this monitoring and shows the most solicited actuators during climbing (arm, wrists, grippers, hip, knees and ankle). As can be seen from the color gradient, the PC and actuator locations are highlighted. In particular, wrist and ankle actuators are the most solicited.

## 6.3 Discussion

We demonstrate for the first time, a humanoid robot climbing a vertical industrial norm ladder. Our trials can certainly be improved in many ways, but they already show that the HRP-

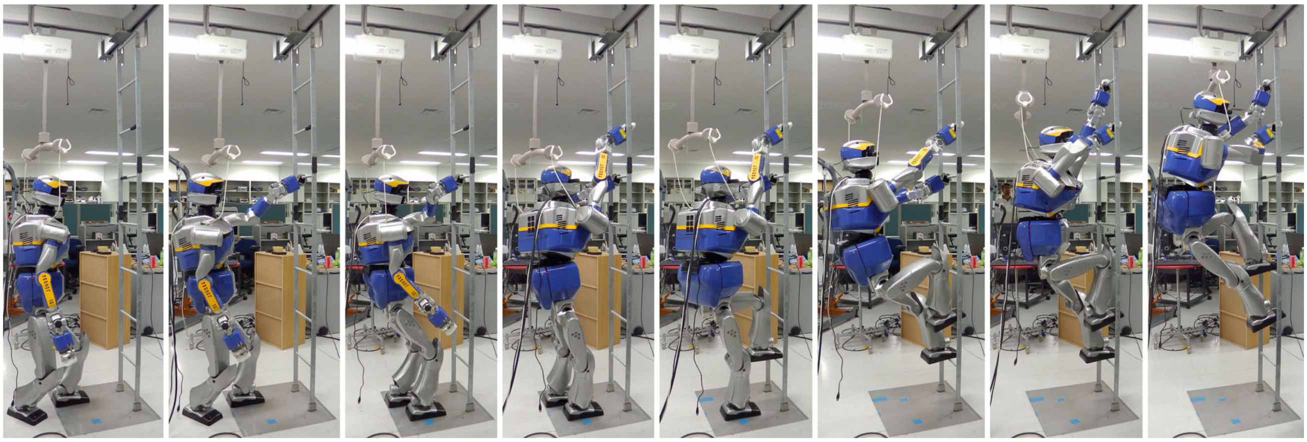


**Fig. 15** From *left to right* initial posture of the HRP-2, *left* arm grasps the fifth rung, *right* leg brought near the ladder, *right* arm grasps the sixth rung, *left* leg on first rung, *right* leg on second rung with a robot lift from the ground, *left* arm grasps the seventh rung (the *left* leg and *left* arm are completely stretched and the *right* leg knee touches the third rung), *left* leg transfer to the third rung with a robot lift

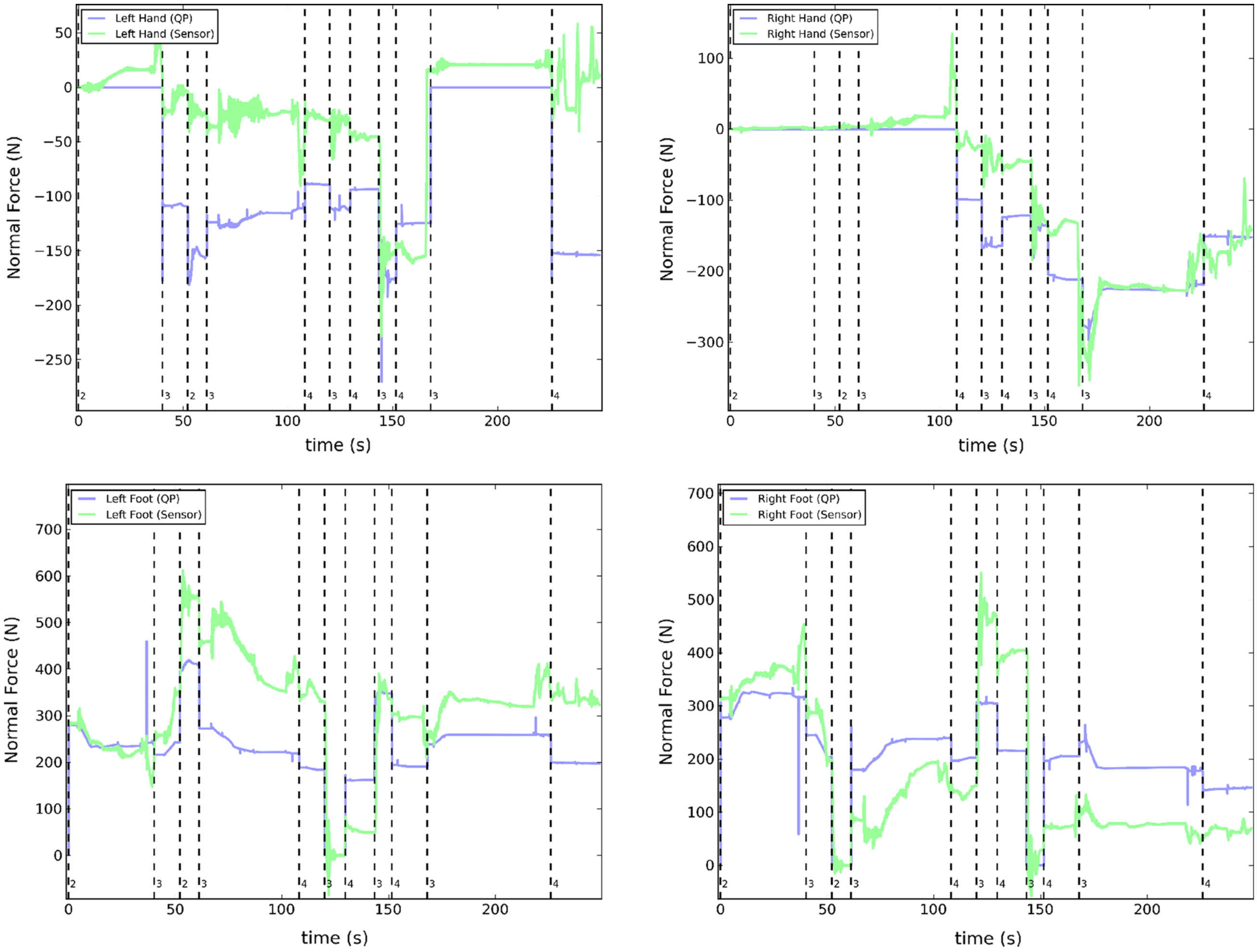


**Fig. 16** Recorded force data from the experiment illustrated in Fig. 15. QP output normal forces versus real force sensing data from *left* and *right* hands (*first line*) and *left* and *right* foot (*second line*)

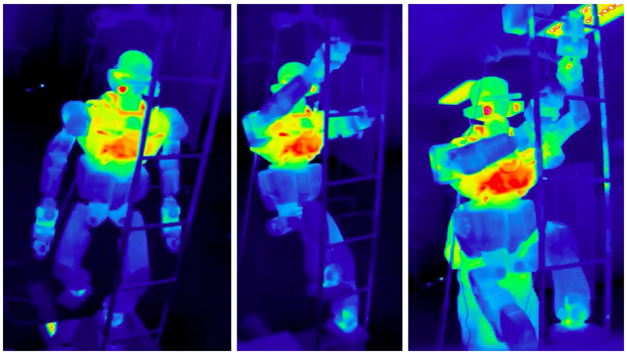


**Fig. 17** Infra-red camera monitoring of HRP-2's actuators during third phase climbing experiments. From *left to right* initial posture, first lift, second lift. Thermal intensity range from cold (*dark blue*) to hot (*red*) (Color figure online)

2 humanoid robot has the capability to climb vertical ladders, what no humanoid platform proved up-to-now. We capitalized valuable factual knowledge (lessons) that will allow us to undertake several improvements prior to experimenting more varieties of ladders with transitions to other modalities. We report here the most relevant open issues.

– *Grippers and grasps*

The most critical problem we faced in our experiments is the HRP-2's grippers design, which restricted the possible climbing plans. In order to explain the problem in technical terms, we illustrate the two possible ladder grasps in Fig. 18, where the gripper is closed completely around the stringer and the rung. It is easy to understand from Fig. 18 that the grasping configuration is not blocking (i.e. not a firm grab). Notice, in both grasps, the large gap that remains inside the closure, and within which the gripper is free to move or rotate w.r.t. the rung or the stringer. We mentioned in Sect. 6.2 that yawing could occur in some cases when one grasp is released. This may generate sliding resulting in a substantial posture discrepancy. When both grippers hold the ladder, the closed kinematic chain linking both arms and the leg in contact would prohibit such yawing to occur durant the motion.

Moreover, each gripper has a limiting grasping power. As a consequence, it is often difficult to sustain the grasp with the stringer or the rung. Figure 18 illustrates this limitation with a detailed technical explanation in the caption. In brief, the pulling forces apply at contacts that are situated in the weakest parts of the gripper. To circumvent partly this problem, we increased the gains of the gripper's servo motor. This temporary solution allowed maintaining the grippers closed during rung grasps in most situations. But, it was not sufficient in the case of stringer grasps. Simulations and open-loop experiments showed that if the ladder stringers can be grabbed firmly, the ladder climbing is easier, the robot has more space to be near the ladder and this may offer the planner more solutions. For example, alternate and combine stringer/rung grasps as in Sect. 6.1.

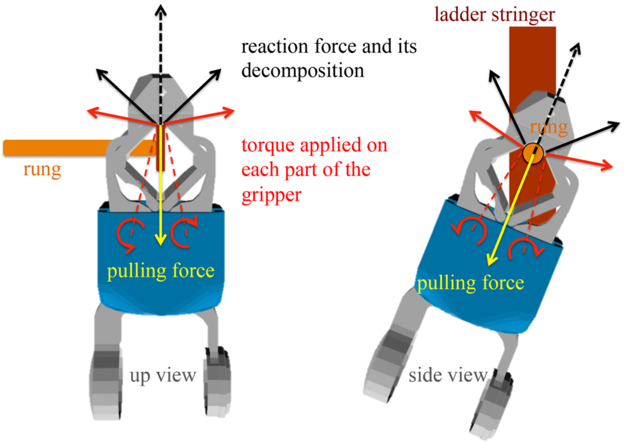

ladder stringer
reaction force and its decomposition
rung
torque applied on each part of the gripper
rung
pulling force
pulling force
up view
side view

**Fig. 18** Disposition of the rung and the stringer within the HRP-2's gripper once closed. The reaction to the pulling forces can be decomposed at the contact locations into forces between the stringer/rung and the gripper's fingers. These contact forces can also be projected onto the line (*dashed red*) linking the contact point to the finger's rotation axis. This decomposition leads to two force vector components: the one along the *dashed red line*, and the other one is orthogonal to the latter, illustrated by the *red* vector. These forces produce torques (represented by *red semi-arrowed-circles*) around the finger's axes of rotation that must be compensated by motor servo PD. If not, the gripper opens. Note that the thinness of the stringer would require only small opening to slip out of the gripper. The rung is thicker, hence it requires higher pulling forces to get out of the gripper (Color figure online)

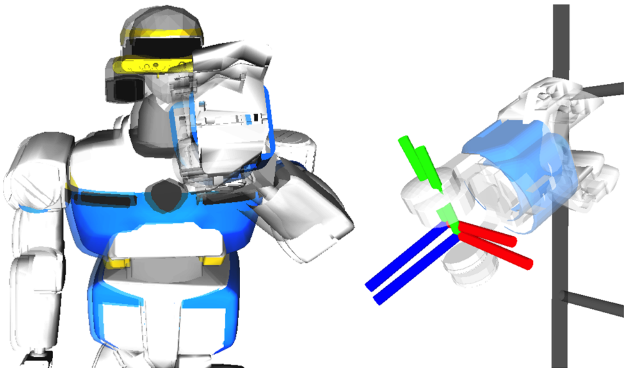


**Fig. 19** Illustration of the effect of ankles compliance compensation. The transparent (clear) robot is the posture obtained from the (QP) model whereas the darker robot is the posture computed after compliance compensation and servoing using Eq. 12

– *Ankle compliance*

Figure 19 illustrates the recovery of the posture discrepancy due to the ankle compliance. Discrepancies may cause high internal forces, as also reported in Fujii et al. (2008). They may also engender moments, which make the robot yaw, when a contact is released. Indeed, not only the contact points are almost coplanar, but they are nearly aligned, which result in yawing if the moment cannot be controlled. All these behaviors were observed in practice.

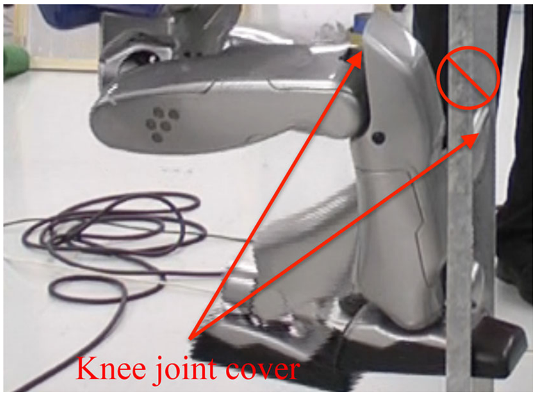


**Fig. 20** *Left* leg knee-cover stuck on a rung during *left*-leg transfer

Compliant shock absorbing mechanisms in the ankles and in the wrists, absorb not only the shocks during contact formation and removal, but also light perturbations shall they occur during transfer motions. Torque controlled humanoid would nicely comply to more important perturbations. In the contrary, a stiff position-controlled humanoid would behaves like a 'rock', and any substantial perturbation, yawing or posture mis-adjustment result in contact sliding or breaking. For the time being the light perturbations we emulate by touch during trials do not seem critical for the climbing tasks. However, we are planning to servo the robot with low PD gains and a feedforward term $u = K_p\varepsilon + K_s\dot{\varepsilon} + \mathscr{D}(q, \dot{q})$: $K_x$ being the gains, $\varepsilon$ the servo position error and $\mathscr{D}$ the feedforward term. This idea is also discussed in Luo et al. (2014), where the $K_p$ gain was adjusted in the gripper at the cost of losing precision, whereas $\mathscr{D}$ was left for future work.

– *Miscellaneous*

Although the HRP-2 seems to be already well-designed in finding good compact postures, free from auto-collisions, we noticed reachability problems that need to be considered. This suggests to elongate some links of the arms and legs, what would be welcome if only rungs can be used.

We thought about the possibility to consider more dynamical gaits similar to Yoneda et al. (2008). For instance, by computing the CoM trajectory with a preview of up-coming contact, like what we already did in Audren et al. (2014). This is certainly not necessary since the vertical ladder climbing requires slow motion strategies at contact formation (including grasps) and removal and we do not use hook-like designed grippers.

In one experiment, the knee cover in Fig. 20 was stuck on a rung during the left leg transfer from the ground to the ladder, which resulted in an excess of torques that switched-off the HRP-2 servo. Extra-care shall be taken in designing the cover of the robot so that these situations are avoided. This also suggest that the FSM, see Sect. 4, should monitor all the motion in a guarded way. For example, monitoring task errors profiles in any situation to prevent excess of torque and take less radical recovery procedures.

## 7 Conclusion

We successfully conducted the climbing of a vertical ladders having industrial norms with the HRP-2 humanoid robot. In Sect. 1, we stated three objectives behind this work.

As for the second objective: our experiments revealed that the HRP-2 humanoid robot has the capability and strength to climb vertical ladders. Its design proved to be efficient in finding good compact postures.

Concerning the third objective: we found that the current grippers design is very limiting. Firm grasps is critical (as for humans) to climb up and down ladders efficiently and prevent from yawing when a contact is released. Subsequently, hardware modifications are performed to change the gripper's clamps into new ones. The arms and legs' links are slightly elongated to increase reachability encountered in some key configurations when only rungs can be used.

As for the first objective, we need to enhance the robustness of the controller and the planner. Also, the visual perception tasks are to be integrated in our multi-objective controller to achieve visual servoing using model/cloud matching. We already started trials for planning on point cloud (Brossette et al. 2013), contact areas can even be extracted and understood directly from the point cloud (Eilering et al. 2014).

As near-future work concerning vertical ladders climbing, we want to challenge multi-modal transitions illustrated in Fig. 11. We also want to tackle ladders with protection cages, which may offer more contact possibilities, e.g. between the robot's back and the cage.

**Acknowledgments** This work is supported partly by internal Grants from IS-AIST, the JSPS Kakenhi B No. 25280096, and the EU FP7 KoroiBot Project www.koroibot.eu. Part of this work was published in Vaillant et al. (2014).

## Appendix: Computing orientation task Err

To compute the task orientation error Err between two rotation matrices $E_1$ and $E_2$, we use the logarithmic matrix formula described in Murray et al. (1994).

Let $E = E_1^T E_2$, $\hat{\omega} \in \mathbb{R}^{3\times 3}$ the skew matrix representing angular velocities, and the rotational speed matrix $\dot{E} = \hat{\omega}E$,

$$E = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix}$$

We have $\log(\mathrm{E}) = \log(\exp(\hat{\omega})) \equiv \hat{\omega}$. To compute the rotational speed vector $\omega \in \mathbb{R}^3$ we use the following formula:

$$\omega = \frac{1}{2\mathrm{sinc}(\theta)} \begin{bmatrix} E_{32} - E_{23} \\ E_{13} - E_{31} \\ E_{21} - E_{12} \end{bmatrix}$$

where $\theta = \cos^{-1}\left(\frac{E_{11}+E_{22}+E_{33}-1}{2}\right)$. Then we simply set:

$$\mathrm{Err}(E_1, E_2) = \omega$$

## References

Abe, Y., da Silva, M., & Popović, J. (2007). Multiobjective control with frictional contacts. In *Eurographics/ACM SIGGRAPH symposium on computer animation* (pp. 249–258). San Diego, CA.

Audren, H., Vaillant, J., Kheddar, A., Escande, A., Kaneko, K., & Yoshida, E. (2014). Model preview control in multi-contact motion– application to a humanoid robot. In *IEEE/RSJ international conference on intelligent robots and systems*. Chicago, IL.

Bevly, D.M., Farritor, S., & Dubowsky, S. (2000). Action module planning and its application to an experimental climbing robot. In *IEEE international conference on robotics and automation* (pp. 4010–4015). San Francisco, CA.

Bouyarmane, K., Escande, A., Lamiraux, F., & Kheddar, A. (2009). Potential field guide for humanoid multicontacts acyclic motion planning. In *IEEE international conference on robotics and automation* (pp. 1165–1170). Kobe.

Bouyarmane, K., & Kheddar, A. (2010). Static multi-contact inverse problem for multiple humanoid robots and manipulated objects. In *IEEE-RAS international conference on humanoid robots* (pp. 8–13). Nashville, TN. doi:10.1109/ICHR.2010.5686317.

Bouyarmane, K., & Kheddar, A. (2011). Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations. In *IEEE/RSJ international conference on intelligent robots and systems*. San Fransico, CA.

Bouyarmane, K., & Kheddar, A. (2012). Humanoid robot locomotion and manipulation step planning. *Advanced Robotics*, *26*(10), 1099–1126.

Bouyarmane, K., Vaillant, J., Keith, F., & Kheddar, A. (2012). Exploring humanoid robots locomotion capabilities in virtual disaster response scenarios. In *IEEE-RAS international conference on humanoid robots*. Osaka: Businness Inoovation Center.

Bretl, T. (2006). Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. *International Journal of Robotics Research*, *25*(4), 317–342.

Bretl, T., & Lall, S. (2008). Testing static equilibrium for legged robots. *IEEE Transactions on Robotics*, *24*(4), 794–807.

Brossette, S., Escande, A., Vaillant, J., Keith, F., Moulard, T., & Kheddar, A. (2014). Integration of non-inclusive contacts in posture generation. In *IEEE/RSJ international conference on intelligent robots and systems*. Chicago, IL.

Brossette, S., Vaillant, J., Keith, F., Escande, A., & Kheddar, A. (2013). Point-cloud multi-contact planning for humanoids: Preliminary results. In *IEEE conference on robotics* (pp. 19–24). Manila: Automation and Mechatronics Philippines.

Collette, C., Micaelli, A., Andriot, C., & Lemerle, P. (2007). Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts. In *IEEE/RAS international conference on humanoid robots* (pp. 81–88). Pittsburgh, PA.

da Silva, M., Abe, Y., & Popović, J. (2008). Simulation of human motion data using short-horizon model-predictive control. *Computer Graphics Forum*, *27*(2), 371–380.

de Lasa, M., Mordatch, I., & Hertzmann, A. (2010). Feature-based locomotion controllers. *ACM Transactions on Graphics (SIGGRAPH)*, *29*(4), 1.

Eilering, A., Yap, V., Johnson, J., & Hauser, K. (2014). Identifying support surfaces of climbable structures from 3d point clouds. In *IEEE international conference on robotics and automation*. Hong Kong.

Escande, A., & Kheddar, A. (2009). Contact planning for acyclic motion with tasks constraints. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 435–440). St. Louis, MO.

Escande, A., Kheddar, A., & Miossec, S. (2013). Planning contact points for humanoid robots. *Robotics and Autonomous Systems*, *61*(5), 428–442.

Escande, A., Mansard, N., & Wieber, P. B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, *33*(7), 1006–1028.

Escande, A., Miossec, S., Benallegue, M., & Kheddar, A. (2014). A strictly convex hull for computing proximity distances with continuous gradient. *IEEE Transactions on Robotics*, *30*(3), 666–678.

Fujii, S., Inoue, K., Takubo, T., Mae, Y., & Arai, T. (2008). Ladder climbing control for limb mechanism robot ASTERISK. In *IEEE international conference on robotics and automation* (pp. 3052–3057). Pasadena, CA.

Gill, P.E.E., Hammarling, S.J., Murray, W., Saunders, M.A., & Wright, M.H. (1986). User's guide for LSSOL (version 1.0): A Fortran package for constrained linear least-squares and convex quadratic programming. Technical Report 86-1. Standord, CA: Standford University.

Hauser, K., Bretl, T., Latombe, J. C., Harada, K., & Wilcox, B. (2008). Motion planning for legged robots on varied terrain. *International Journal of Robotics Research*, *27*(11–12), 1325–1349.

Herzog, A., Righetti, L., Grimminger, F., Pastor, P., & Schaal, S. (2014). Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *IEEE/RSJ international conference on intelligent robots and systems*. Chicago, IL.

Hyon, S. H., Hale, J. G., & Cheng, G. (2007). Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Transactions on Robotics*, *23*(5), 884–898.

Ibanez, A., Bidaud, P., & Padois, V. (2014). A distributed model predictive control approach for robust postural stability of a humanoid robot. In *IEEE international conference on robotics and automation* (pp. 202–209). Hong Kong.

Iida, H., Hozumi, H., & Nakayama, R. (1989). Development of ladder climbing robot LCR-1. *Journal of Robotics and Machatronics*, *1*, 311–316.

Kanehiro, F., Morisawa, M., Suleiman, W., Kaneko, K., & Yoshida, E. (2010). Integrating geometric constraints into reactive leg motion generation. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 4069–4076).

Kanoun, O., Lamiraux, F., & Wieber, P. B. (2011). Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, *27*(4), 785–792.

Kuindersma, S., Permenter, F., & Tedrake, R. (2014). An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *IEEE international conference on robotics and automation*. Hong Kong.

Lee, S. H., & Goswami, A. (2012). A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots*, *33*(4), 399–414.

Lengagne, S., Vaillant, J., Yoshida, E., & Kheddar, A. (2013). Generation of whole-body optimal dynamic multi-contact motions.

*The International Journal of Robotics Research*, *32*(9–10), 1104–1119.

Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, *7*(12), 868–871.

Liu, M., Micaelli, A., Evrard, P., Escande, A., & Andriot, C. (2012). Interactive virtual humans: A two-level prioritized control framework with wrench bounds. *IEEE Transactions on Robotics*, *28*(6), 1309–1322.

Luo, J., Zhang, Y., Hauser, K., Park, H.A., Paldhe, M., Lee, C.G., Grey, M., Stilman, M., Oh, J.H., Lee, J., Kim, I., & Oh, P. (2014). Robust ladder-climbing with a humanoid robot with application to the DARPA robotics challenge. In *IEEE international conference on robotics and automation* (pp. 2792–2798). Hong Kong.

Mansard, N., Khatib, O., & Kheddar, A. (2009). A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, *25*(3), 670–685.

Mordatch, I., de Lasa, M., & Hertzmann, A. (2010). Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics*, *29*(3).

Mordatch, I., Todorov, E., & Popovi'c, Z. (2012). Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (SIGGRAPH)*, *31*(4), 43.

Murray, R. M., Sastry, S. S., & Zexiang, L. (1994). *A mathematical introduction to robotic manipulation*. Boca Raton, FL: CRC Press Inc.

Nakai, H., Kuniyoshi, Y., Inaba, M., & Inoue, H. (2002). Metamorphic robot made of low melting point alloy. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 2025–2030). Lausanne.

Nakamura, Y., Hanafusa, H., & Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *International Journal of Robotics Research*, *6*(2), 3–15.

Noda, S., Murooka, M., Nozawa, S., Kakiuchi, Y., Okada, K., & Inaba, M. (2014). Generating whole-body motion keep away from joint torque, contact force, contact moment limitations enabling steep climbing with real humanoid robot. In *IEEE international conference on robotics and automation* (pp. 1775–1781). Hong Kong.

Ott, C., Roa, M.A., & Hirzinger, G. (2011). Posture and balance control for biped robots based on contact force optimization. In *IEEE-RAS international conference on humanoid robots* (pp. 26–33). Bled.

Posa, M., Cantu, C., & Tedrake, R. (2014). A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, *33*(1), 69–81.

Righetti, L., Buchli, J., Mistry, M., Kalakrishnan, M., & Schaal, S. (2013). Optimal distribution of contact forces with inverse-dynamics control. *The International Journal of Robotics Research*, *32*(3), 280–298.

Righetti, L., & Schaal, S. (2012). Quadratic programming for inverse dynamics with optimal distribution of contact forces. In *IEEE-RAS international conference on humanoid robots*. Osaka: Businness Inoovation Center.

Saab, L., Ramos, O. E., Keith, F., Mansard, N., Souères, P., & Fourquet, J. Y. (2012). Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, *29*(2), 346–362.

Salini, J., Barthélemy, S., & Bidaud, P. (2010). *LQP-based controller design for humanoid whole-body motion* (pp. 177–184). Berlin: Springer.

Salini, J., Padois, V., & Bidaud, P. (2011). Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions. In *IEEE international conference on robotics and automation* (pp. 1283–1290). Shanghai.

Schittkowski, K. (1986). Qld: A Fortran code for quadratic programming, user's guide. Technical report. Mathematisches Institut, Universität Bayreuth.

Sentis, L., Park, J., & Khatib, O. (2010). Compliant control of multi-contact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on Robotics*, *26*, 483–501.

Sentis, L., Petersen, J., & Philippsen, R. (2013). Implementation and stability analysis of prioritized whole-body compliant controllers on a wheeled humanoid robot in uneven terrains. *Autonomous Robots*, *35*(4), 301–319.

Siciliano, B., & Slotine, J.J.E. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *International conference on advanced robotics* (Vol. 2, pp. 1211–1216). Pisa.

Vaillant, J., Kheddar, A., Audren, H., Keith, F., Brossette, S., Kaneko, K., Morisawa, M., Yoshida, E., & Kanehiro, F. (2014). Vertical ladder climbing by HRP-2 humanoid robot. In *IEEE-RAS international conference on humanoid robots* (pp. 671–676). Madrid.

Wächter, A., & Biegleri, L. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*(1), 25–57.

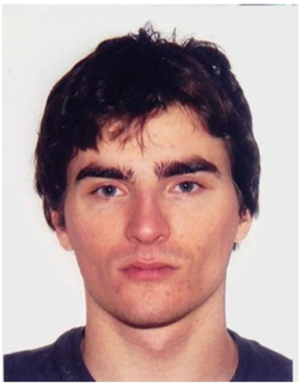
Wensing, P.M., & Orin, D.E. (2013). Generation of dynamic humanoid behaviors through task-space control with conic optimization. In *IEEE international conference on robotics and automation* (pp. 3088–3094). Karlsruhe.

Wieber, P.B. (2002). On the stability of walking systems. In *IARP international workshop on humanoid and human friendly robotics*. Tsukuba.

Yoneda, H., Sekiyama, K., Hesegawa, Y., & Fukuda, T. (2008). Vertical ladder climbing motion with posture control for multi-locomotion robot. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 3579–3584). Nice.

Zhang, Y., Luo, J., Hauser, K., Ellenberg, R., Oh, P., Park, H.A., & Paldhe, M. (2013). Motion planning of ladder climbing for humanoid robots. In *IEEE international conference on technologies for practical robot applications* (pp. 1–6).

Zucker, M., Ratliff, N., Dragan, A., Pivtoraiko, M., Klingensmith, M., Dellin, C., et al. (2013). CHOMP: Covariant hamiltonian optimization for motion planning. *International Journal of Robotics Research*, *32*(9–10), 1164–1193.



**Joris Vaillant** received the Master degree in interactive system at Paul Sabatier University, Toulouse, France, in 2010. He is currently a Ph.D. candidate in robotics with the Interactive Digital Human group at the CNRS-UM2 Laboratory of Informatics, Robotics, and Microelectronics, Montpellier, France. His research interest is contact planning and whole body motion for humanoids and virtual avatars.

**Abderrahmane Kheddar** received the BSCS degree from the Institut Nation d'Informatique ESI, Algiers, the M.Sc. and Ph.D. degrees in robotics, both from the University of Pierre and Marie Curie, Paris 6. He is presently Directeur de Recherche at CNRS www.cnrs.fr. He is the Director of the CNRS-AIST Joint Robotic Laboratory (JRL), UMI3218/RL, Tsukuba, Japan; and leader of the Interactive Digital Humans (IDH) team at CNRS-UM2 LIRMM at Montpellier, France. His research interests include humanoid robotics, haptics, and recently thought-based control using brain machine interfaces. He is a founding member of the IEEE/RAS chapter on haptics (acting also as a senior advisor), the co-chair and co-founding member of the IEEE/RAS Technical committee on model-based optimization. He is presently Editor of the IEEE Transactions on Robotics, the Journal of Intelligent and Robotic Systems, and Frontiers in Bionics; he is a founding member of the IEEE Transactions on Haptics and served in its editorial board during three years (2007–2010), he also served as associate editor in the MIT Press PRESENCE journal. He coordinated or acted as a PI for several EU projects. He is titular (full) member of the National Academy of Technologies of France (NATF) and a Senior Member of the IEEE Society.

**Hervé Audren** received a Master degree in applied mathematics from Ecole des Mines de Paris, France, in 2013. He is currently working toward the Ph.D. Degree in Robotics with the Laboratory of Informatics, Robotics, and Microelectronics, Montpellier, France (LIRMM) and the CNRS-AIST Joint Robotics Laboratory, Tsukuba, Japan (JRL). His research interests encompass optimal control, human-robot interaction and motion planning.

**François Keith** received the M. Eng. degree from the École Nationale Supérieure d' Informatique d' Industrie et d' Entreprise, Évry, France, in 2007, and the Ph.D. degree in computer science from the University of Montpellier II, Montpellier, France, in 2010. Since 2007, he has been with the Centre National de la Recherche Scientifique – National Institute of Advanced Industrial Science and Technologies Joint Robotics Laboratory, UMI3218/CRT, Tsukuba, Japan, and the Montpellier Laboratory of Informatics, Robotics, and Microelectronics (LIRMM), where he is currently a Research Engineer. He spent one year with the Institut National de Recherche en Informatique et en Automatique Grenoble with P-B. Wieber. His research interests include humanoid robots and task sequencing for planning and control.

**Stanislas Brossette** received a Master degree in Computational Mechanics at the École Normale Supérieure de Cachan, France in 2011. He is currently working toward the Ph.D. Degree in Robotics with the Laboratory of Informatics, Robotics, and Microelectronics, Montpellier, France (LIRMM) and the CNRS-AIST Joint Robotics Laboratory, Tsukuba, Japan. His research interest includes multi-contact posture generation and environment perception for humanoid robots.

**Adrien Escande** received the MS degree in 2005 from École des Mines de Paris, France and the Ph.D. degree in 2008 in robotics from Université d' Évry Val-d' Essonne, France after spending three years in the CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan. He then worked as a research scientist in CEA-LIST at Fontenay-aux-Roses, France, until the end of 2012 and is now back at JRL. His current research interests include whole-body planning and control for humanoid robots and mathematical optimization for robotics.

**Karim Bouyarmane** received the Ingénieur diploma from École Polytechnique in Palaiseau in 2007 and from École Nationale Supérieure des Mines de Paris in 2008, and the Ph.D. degree from the University of Montpellier in 2011. He completed the Ph.D. program in the Joint Robotics Laboratory (JRL) at the National Institute of Advanced Industrial Science and Technology (AIST) in Tsukuba, Japan. He then held a Japan Society for the Promotion of Science (JSPS) post-doctoral fellowship at the Advanced Telecommunications Research Institute International (ATR) in Kyoto, in Mitsuo Kawato's Computational Neuroscience Laboratories working on brain-robot interfaces. He is currently a CNRS research associate.

**Kenji Kaneko** received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University in 1988, 1990, and 1997, respectively. In 1990, he joined the Mechanical Engineering Laboratory, Agency of Industrial Science and Technology, Ministry of Industrial Science and Technology (AIST-MITI). From September 1995, he was a visiting researcher at the Carnegie Mellon University, U.S.A. for half a year. From September 1999, he was a visiting researcher at the "Laboratoire de Robotique de Paris (LRP)", CNRS, France for one year. He was in charge of the development of the HRP series humanoid robots. He received the IEEE-RAS International Conference on Humanoid Robots, Best Paper Award (Oral Category) in 2009. He is currently the chief senior researcher of the Intelligent Systems research Institute, AIST. His research interests include motion control and humanoid robot.

**Mitsuharu Morisawa** received the Ph.D. degree in engineering from Keio University, Japan, in 2004. Researcher of the Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technolog (AIST), Japan, from 2004. He was a visiting researcher at the LAAS-CNRS, France for one year from April 2009, and a inviting researcher at the INRIA-Rhone-Alpes, France for three month from September 2011. He is currently the senior researcher at AIST. His research interests include biped locomotion, stabilization and whole body motion control of the humanoid robot.

**Pierre Gergondet** graduated from Ecole Centrale de Paris with a master in embedded systems. He obtained a Ph.D. degree from the University of Montpellier and conducted his thesis in CNRS-AIST JRL, UMI3218/CRT, Tsukuba, Japan where is is now a research associate working on embedded software for humanoids. His research interest includes humanoid robotics and the usage of brain-computer interfaces to control a humanoid robot.

**Eiichi Yoshida** received Ph.D. degrees on Precision Machinery Engineering from Graduate School of Engineering, the University of Tokyo in 1996. He then joined former Mechanical Engineering Laboratory, which was reorganized into National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. From 1990 to 1991, he was visiting research associate at Swiss Federal Institute of Technology at Lausanne (EPFL). He served as Co-Director of AIST/IS-CNRS/ST2I Joint French-Japanese Robotics Laboratory (JRL) at LAAS-CNRS, Toulouse, France, from 2004 to 2008. He is currently Co-Director of CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT, Intelligent Systems Research Institute, AIST, Japan, since 2009. His research interests include robot task and motion planning, modular robotic systems, and humanoid robots.

**Suuji Kajita** Received M.E. (1985) and Dr.E. (1996) degrees in control engineering from Tokyo Institute of Technology, Japan. In 1985, he joined the Mechanical Engineering Laboratory, Ministry of International Trade and Industry. Meanwhile he was a Visiting Researcher at California Institute of Technology, 1996-1997. Currently he is a senior researcher at the National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan, which was reorganized from AIST-MITI in April 2001. His research interests include robotics and control theory. He is a member of Society of Instrument and Control Engineers, Robotics Society of Japan and IEEE(Robotics and Automation Society).

**Fumio Kanehiro** received the B.E., M.E., and Ph.D. in engineering from The University of Tokyo, Japan, in 1994, 1996, and 1999, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science in 1998–1999. In 2000, he joined the Electrotechnical Laboratory, Agency of Industrial Science and Technology, Ministry of Industrial Science and Technology (AIST-MITI). From April 2007, he was a visiting researcher at the LAAS-CNRS for one year and three months. He is currently a senior researcher of the Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST). His research interests include the software platform development and whole body motion control of the humanoid robot.