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

Joris Vaillant, Abderrahmane Kheddar, Hervé Audren, François Keith,
Stanislas Brossette, Adrien Escande, Karim Bouyarmane, Kenji Kaneko,
Mitsuharu Morisawa, Pierre Gergondet, et al.

Multi-Contact Vertical Ladder Climbing with a HRP-2 Humanoid

Joris Vaillant · **Abderrahmane Kheddar** · **Hervé Audren** · **François Keith** · **Stanislas Brossette** · **Adrien Escande** · **Kenji Kaneko** · **Mitsuharu Morisawa** · **Pierre Gergondet** · **Eiichi Yoshida** · **Suuji Kajita** · **Fumio Kanehiro**

Abstract We describe the research and integration methods we developed to give the HRP-2 humanoid robot the capability to climb vertical industrial-norm ladders. Our approach makes use of 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 additional intermediary tasks accounting for geometric uncertainties and specific grasps procedures to be realized by our multi-objective model-based QP controller. This controller 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 to be made on the HRP-2, and parts of software are to be made more robust. Yet, we confirmed that HRP-2 has the kinematic and power capabilities to climb real industrial ladders, which can be found in nuclear power plants and large scale manufacturing such as shipyards, aircraft factories and construction sites.

Keywords Humanoid robots · multi-contact motion planning and control · field humanoid robots · disaster humanoid robots

All authors
CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/RL, Tsukuba, Japan

J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette
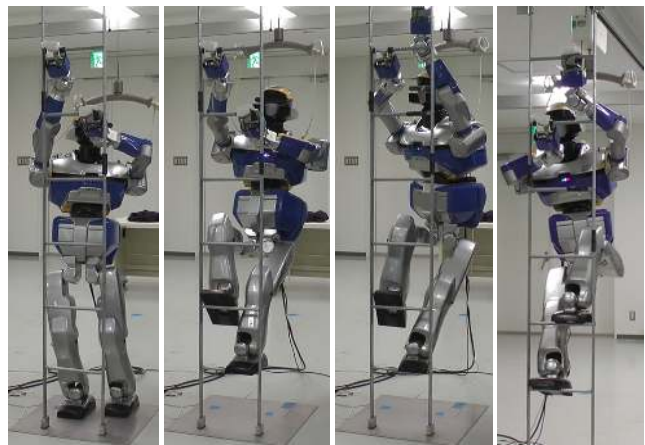CNRS-UM2 LIRMM Interactive Digital Human group, UMR5506, Montpellier, France

Fig. 1 HRP-2 climbing a vertical ladder. Notice that : (i) it is not possible to put two feet on a same rung (ii) closed grippers do not grab firmly the rungs (iii) each foot can be freely positioned on each rung: the right foot is rotated to increase the reaching range of the left arm toward the higher rung.

1 Introduction

Humanoid robots reached a noticeable level in technological maturity for walking on flat grounds. The Honda's Asimo humanoid robot 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 seen in numbers inside nuclear power plants, construction

---

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

sites, and large scale industries such as shipyards and aircrafts factories. Climbing various 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 70deg and one by 60deg. Each of these ladders have 10cm-wide rungs, and handrails that can optionally be removed. These dimensions accommodate robotic constraints for the first trials. During the 2013 December DRC qualifiers, several teams tried different strategies for ladder climbing. But all teams chose the less inclined 60deg ladder. This ladder is closer to stairs than to what can be found commonly in real sites. As a mater of fact, the winning team, SHAFT, climbed it with feet only. However, their robot used two-feet-one-rung intermediary transitions before climbing 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 [56], could also climb almost all of it, using a similar backward strategy [33] 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 [25], a Japanese team from Toshiba company designed a four limbs robot for nuclear power plants. This robot has four prismatic arms with grippers and ascends/descends ladders with a cyclic sequence of rung grasps. The climbing sequence consists in alternating one arm transition phases, while the three others maintained their contacts. Lifting the body is made with the four arms in contact, two per different rung. In [3], a planar three legged climbing robot was demonstrated climbing pegs disposed as a vertical ladder or an H bi-ladder form. This study revealed that by identifying key motion primitives coupled with physics simulation, the planning is tractable and can be optimized according to desired criteria. The idea is interesting and could be investigated further. Ladder climbing was also demonstrated with a deformable-on-demand legs robot in [38]. The latter work is more a concept demonstration than a plausible solution. In [20], 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. Moreover, although the climbing gait is simple (three legs are used on each rung prior to pulling the robot's body and at least two legs must hold each rung during leg transfer) it is clearly stable. But this approach strongly limits the movement possibilities. This study revealed however that contact forces should be monitored properly since despite having many legs and a safe climbing, geometric discrepancies may cause a non-proper distribution of forces which result in having internal efforts and a bad repartition of the load among the legs.

In [55] a gorilla-type robot was shown to climb a vertical ladder having in mind transitions toward multi-modal locomotion capabilities. 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 (which we also experienced in our trials). Lastly, [40] demonstrated capabilities of the HRP-2 in climbing inclined ladders (2 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 [7][17].

We address here the climbing of vertical ladders by the HRP-2 robot, see Fig. 1. There are various specificities and differences with the previously mentioned works that we explicit later. However, two main distinguishing challenges drive our research:

1) The first is to address directly vertical industrial-norm ladders. We prohibit any change or adjustment on the ladders as this would not be possible in practice. Although inclined ladder may have specific difficulties, at a certain inclination angle, the robot can adjust its center of mass (CoM) projection to be inside the contact support polygon. A (near to) vertical ladder forces the CoM to be always outside the contact polygon.

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 experimental section, this was not a 'reasonable' option since we had a very hard time with the current design of the grippers. Relatively to [55][40][33] 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

Fig. 2 illustrates the main components used to plan and experiment multi-contact ladder climbing with HRP-2; they are explained in more details along the paper.
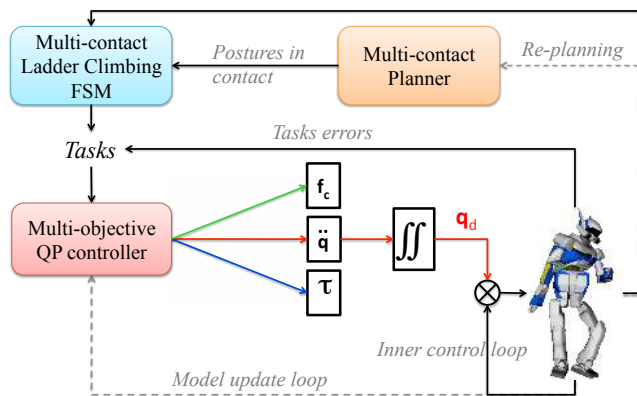
Fig. 2 Main components of the overall architecture.

Our multi-contact planner is model-based. It needs the models of the HRP-2, the ladder and the pertinent parts of the environment, to generate off-line a 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 to generate safe contacts and tansfers. The FSM elaborates additional steps with their associated tasks, and changes on-line their objectives to deal with different phases of the climbing (unilateral contact adding and removal, grasps and their release, center of mass (CoM) transfers, etc). These tasks are passed to our multi-objective QP controller, which also has few built-in tasks (constraints) that account for commonly known limits, and 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)

In order to climb a vertical ladder, a humanoid robot would alternate contacts, as humans do, using their feet on the rungs to support body transfer and grasping the hungs and stringers with gripper to help (which is unavoidable when the ladder is vertical). If the humanoid robot has a human scale and enough degrees-of-freedom, these type of contacts should be enough. Multi-modal contact planning using sampling methods (see early work in [9]) was demonstrated with a simulated ladder climbing in [21] and experimented recently in [56]. A continuous contact exploring method was proposed in computer graphics by [36]. In robotics, the latter work can be exploited as a preliminary guess of the plan, as suggested using another approach by [4], which guides a more refined planning in a second stage. Our work, which falls in the category of contact-before-motion approach is summarized in [17] (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 [7]. Our approach is distinguishable in that (i) we do not sample the contacts *a priori*, (we consider con-

tact to occur on any part of the robot body and the environment), (ii) it is applied to robotics, hence contrary to computer graphics animations, torque limits, collision avoidance, physical plausibility, equilibrium [54][10]... cannot be ignored, (iii) we can handle other tasks (as far as they write as constraints) all along the contact planning process [16].

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. We use the common parametrization for the ladders described in previous papers; e.g. in [33]. The robot is modeled using triangular meshes and each limb is covered by a stict convex hull for a continuous-gradient distance computation [19]. We specify the areas where contacts are permitted to occur for the ladder (all of it), the robot (on the gripper and on the feet's soles) and the environment. We generate contact transitions for climbing the ladder by two different ways:

1. using our planner [17][7] 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, to be used in teleoperation mode or in situations where the automatic planner fails.

The planning approach 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 provide 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 [56], in the sense that we do not provide the contact stances, but we rather suggest pairs from surfaces of both the robot and ladder that can be in contact.

In all our versions of MCP, a posture generator (PG)[2] is paired with the space exploration module [17][7]. 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, auto-collision and non-desired collision avoidance. We may also add other secondary task constraints such as gaze or field-of-view [16]. If a viable posture is found, the resulting contact

---

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

and joint state is returned and added to the tree builder with a given 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 detailed in [7], we brought two changes: (i) a richer contact model, and (ii) the possibility for a multi-posture generation (MPG), in order to generate optimal postures that minimize a cost over the entire path of contact transitions. This proved to be useful for minimizing gripper torques at each contact transition.

3.1 Contact modeling in posture generation

Contacts are defined as constraints in the PG. In our previous work, to enforce stable contact formation, we favored search for plan/plan type of contacts [17]. In order to compute forces on contacting areas in [17] or [7], we predefined lists of contacting areas on the robot and its surroundings, and we limited the possible contact to cases were 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 within one of the list [5]. 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.
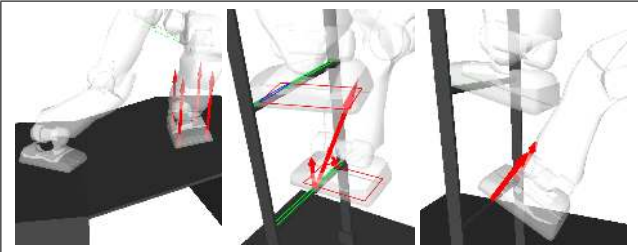

Fig. 3 Three contact models used in the posture generator.

Predefining contact surfaces in advance restricts the possible contact choices during planning. For the ladder climbing case, this is very limiting. We proposed in [11] another contact model that generates contacts with any position and orientation of the foot on the rungs without the need of having a full inclusion: 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 the Fig. 3, where solution of the inscribed ellipse for the left foot on 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 most right image of the Fig. 3.

3.2 Posture generation with gripper torque optimization

Our proposed MPG assumes we have N stances (i.e. contact transitions) for the climbing, we consider 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 contacts frames).

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-force 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:

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

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

• Other collisions:

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

$\mathscr{I}^i_{\text{robot-environment}}$ 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, μ_j is the friction 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 space force vector components.

• Fixed contacts:

$$r_j^i(\mathbf{q}_i) - r_k = \mathbf{0}$$
$$N_j^i(\mathbf{q}_i) \cdot T_k = 0$$
$$N_j^i(\mathbf{q}_i) \cdot B_k = 0$$
$$B_j^i(\mathbf{q}_i) \cdot T_k = 0 \qquad (6)$$
$$N_j^i(\mathbf{q}_i) \cdot N_k \geq 0$$
$$B_j^i(\mathbf{q}_i) \cdot B_k \geq 0$$

where k subscript is for environment surface and j the robot surface.

• Planar contacts:

$$(r_j^i(\mathbf{q}_i) - r_k) \cdot N_k = 0$$
$$N_j^i(\mathbf{q}_i) \cdot T_k = 0$$
$$N_j^i(\mathbf{q}_i) \cdot B_k = 0 \qquad (7)$$
$$N_j^i(\mathbf{q}_i) \cdot N_k \geq 0$$
$$\text{conv}(\mathscr{P}_j) \subseteq \text{conv}(\mathscr{P}_k)$$

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

• Cylindrical contacts:

$$w_{\min} \leq (r_j^i(\mathbf{q}_i) - r_k) \cdot T_k \leq w_{\max}$$
$$(r_j^i(\mathbf{q}_i) - r_k) \cdot B_k = 0$$
$$(r_j^i(\mathbf{q}_i) - r_k) \cdot N_k = 0$$
$$T_j^i(\mathbf{q}_i) \cdot B_k = 0 \qquad (8)$$
$$T_j^i(\mathbf{q}_i) \cdot N_k = 0$$
$$T_j^i(\mathbf{q}_i) \cdot T_k \geq 0$$

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

• Link all robots common contacts:

$$\begin{aligned} r_k^i(\mathbf{q}_i) - r_k^j(\mathbf{q}_j) &= \mathbf{0} \\ \text{Err}(E_k^i(\mathbf{q}_i), E_k^j(\mathbf{q}_j)) &= \mathbf{0} \end{aligned} \qquad \forall (i,j,k) \in \mathscr{I}_{\text{common}} \qquad (9)$$

this equation establishes the connections between contact transitions. See appendix A for Err computation.

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

$$\mathscr{C}_i(\mathbf{q}_i, \mathbf{f}_i) = w_q \|\mathbf{q}_i - \mathbf{q}_i^d\|^2 + \sum_{j \in \mathscr{I}_{\text{contF}}} w_j \|F(\mathbf{f}_i, j)\|^2 +$$
$$\sum_{j \in \mathscr{I}_{\text{contT}}} w_j \sum_{p \in \mathscr{I}_{\text{points}_j}} \|M_j \cdot (r_p \times F(\mathbf{f}_i, p))\|^2 +$$
$$\sum_{j \in \mathscr{I}_{\text{posT}}} w_j \|r_j^i(\mathbf{q}_i) - r_j^d\|^2 + \sum_{j \in \mathscr{I}_{\text{rotT}}} w_j \|\text{Err}(E_j^i(\mathbf{q}_i), E_j^d)\|^2$$

see appendix A for Err. All $\mathscr{I}_x$ represents sets of x, all w_x are cost weights for cost part x, $\mathbf{q}_i^d$ is the targeted configuration

vector, $F(\mathbf{f}_i, j)$ is the $\mathbf{f}_i$ j-th force vector, 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_j^d and E_j^d mean target positions and orientations respectively.
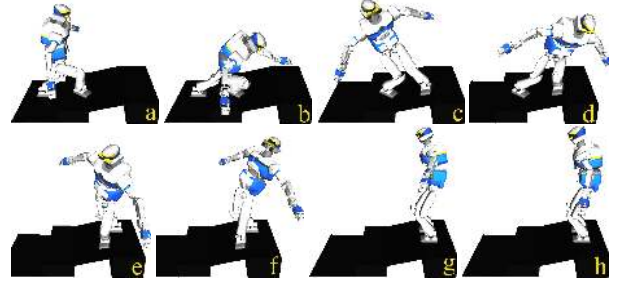

Fig. 4 Example of MPG solving.

Fig. 4 illustrates a contact planning computed with MPG in this case, $N = 8$. We use an objective function to have the robot standing in opposite direction for the last contact transition (h). We use the IpOpt non-linear solver [52] with the RobOptim framework[3] to solve this non-linear problem in $\simeq$ 1s with 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 this choice, we can remove one contact constraint in each contact transition (stance). For the problem illustrated in the 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 constraints eq. (9) can be redundant with eqs. (6), (7), and (8). For example, if the constraint eq. (9) links the right foot between the stance (b) and (c) it is then possible to remove the constraint eq. (7) from one of the two previous stances. Moreover, if we use a kinematic contact instead of a constraint contact we can minimize the number of constraints.

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 $i+1$ sons. To be valid, a path $P = \{v_j^1, \cdots, v_j^N\}$ 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.

---

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

4 Multi-contact finite state machine (FSM)

As illustrated in Fig. 2, the output of the MCP is a sequence of static contact stances/postures to climb the ladders. There are many reasons why such contacts and postures cannot be given directly as task function objectives.

First, when using a task-space approach (see Sec. 5), assigning a target position (including or not orientation) for a frame attached to a given robot's link, would result in a 'straight line' task trajectory. Even if collision avoidance is embedded in our closed-loop control, it is not sufficient to reach safely the desired contacts because of possible local minima. We solve this problem by using way-points (one per motion appear to be enough for ladder climbing). These waypoints are intermediary goals that do not need to be reached fully. Another option is to use real-time planners such as CHOMP [57] and provide the path/trajectory as secondary or (relatively) low priority task objective. We could also extract from the latter path/trajectory result an arbitrary number of way-points.

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 and phases to reach or release grasps are needed as intermediary steps.

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

For all 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

Also we 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 obtained from force sensing or estimation. We use σ^i_{action} to mean 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.

The Fig. 5 illustrates the ladder implementation of the FSM. The *remove unilateral contact* RC is the leftmost branch which is achieved with successful transitions t1 and t2 as follows:
t1: $A^i \leftarrow$ RC
sets A^i, which results in building a set of tasks to be achieved by the controller (section 5) in order for B^i to remove its current contact state progressively, with a force guarded motion. We assume the contact to be removed when:
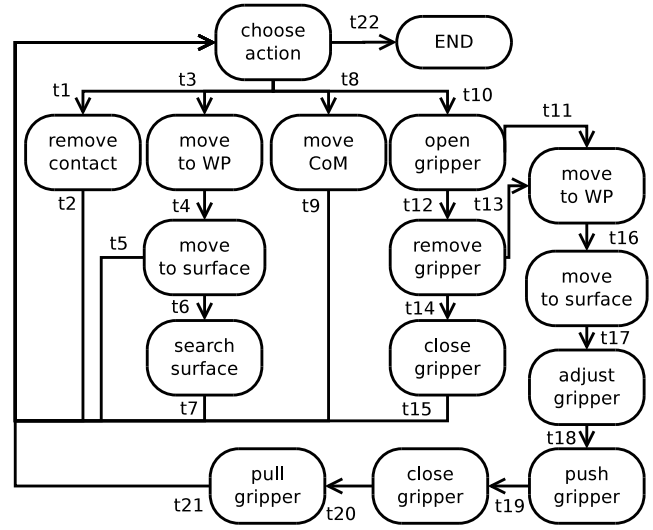

Fig. 5 The current implementation of the FSM for ladder climbing.

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 is either computed from heuristics or given by the user. ε is the measure for 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 sent 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 practical 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 build the 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 the contact is found and the AC task at stance i is considered done.

The next branch in the FSM tree represents the *move CoM* or *CoM transfer task* MC that 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

which leads to the branch of removing the gripper from contact. In this last 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 ($A^{i+1} =$ 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 ($A^{i+1} \neq$ AG or $B^i \neq B^{i+1}$)

in this case, the gripper is closed and leads to transition:
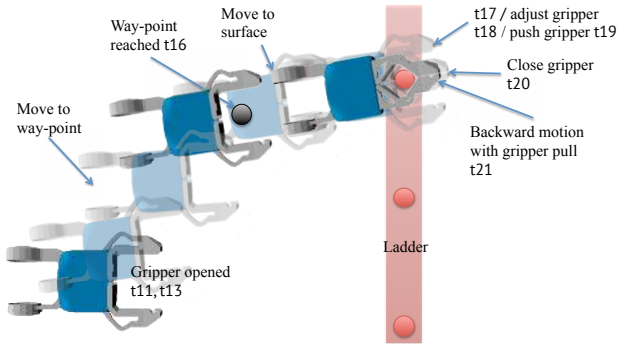
t15: Gripper closed.

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.

Now we assume that we are in the last branch (that is AG at transition t11 or t13), which means that we are going to create a contact by a grasp (steps can be tracked in Fig. 6). As for legs or any other limbs, we build a move to way-point task for the gripper, pass 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 build a go-to-contact task, which is assumed fulfilled when:

t17: $\varepsilon(X_{B^i}, X^i_{\text{AG}}) < \sigma^i_{\text{AG}}$

This step is followed by adjusting the gripper, that 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 (section 5.3). This task is achieved under the following 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 rung or stringer or handrail and now we can close the gripper and check the next transition:

t20: Gripper closed

Now that the gripper is closed (around rung or stringer or handrail), we can sustain the contact by building the guarded force backward motion which ends by:

t21: $F_{B^i} < \sigma^i_{\text{AG pull}}$

All previous AG action steps are illustrated on the Fig. 6.

When the entire climbing is terminated, the FSM state would switch to:

t22: $i \geq N$.

Otherwise, the next stacked action (from the MCP, section 2) is chosen.

5 Multi-objective quadratic program controller (QP)

Sub-tasks from previous section 4 need to be transformed into joint motions under various other constraints, which turn to be additional tasks. Our controller is formulated as a model-based QP optimization problem.

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

Priority weighted tasks formulation considers common hard equality and inequality tasks to be written as optimization constraints whereas the remaining tasks are to be met at best and put in the cost function. Hierarchy among cost function tasks can be made through assigning more weight (task gains) to the most prioritized ones. Examples of such an approach can be found in early work for controlling computer-animated avatars. Formulation of the controller as a quadratic program (QP) is proposed in [1], where standing and balancing with legs is demonstrated with unilateral contacts and under various kind of perturbations. Indeed, at a given instant, the robot dynamic and contact constraints can be expressed as linear functions of the joint torques and accelerations and the contact forces. Other constraints can be approximated by a linear forms. In [13] a richer task formulation is solved using a two-level cascade of local 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 [32] a passivity guaranteed formulation is proposed together with task force predictions, which is an interesting issue to be considered as extension in our multi-contact planning. All of these works apply for virtual characters and did not account for collision avoidance in the control.

Application to humanoid robots in simulation is proposed in [29] for momentum based balance control. There is also the remarkable work in [46], [47] considering torque and state limits for more general purpose tasks. In [6] collision avoidance is introduced and the QP controller is used to control multi-contact whole-body non-gaited motion [8]. The latter constitutes the starting of our controller. Recently, in [28] a QP formulation of the ZMP-based walking is proposed with an efficient fast resolution of the problem. Despite being very appealing, none of the previous works experienced such controllers with complex tasks on a real humanoid robot. This is what we achieve here. Works of [23] and [41] used a force control formalism for balance in multi-contact configuration with experimentations conducted on torque based controlled humanoid robots.

Strict prioritized approaches are the heritage of early work by [31], [39], and [50]. These are based on null-space projectors in the operational space. Application to humanoid robots in simulation using operational space formulation of the dynamics and null-space projectors was illustrated in [48] and experimented on a wheeled humanoid torso in [49]. Prioritized mixed equality and inequality constraints was integrated successfully in [34] using null-space operator projection in kinematics and dynamics, and using cascade of QP with equalities and inequalities in kinematics by [27]. Recently a full control framework using the hierarchical QP [18] was implemented in simulation in [45]. Experiments on HRP-2 humanoid were made based on the trajectory generated off-line. 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 [44] the dynamic problem was more efficiently formulated and the control explicitly derived to optimize contact forces. This work is experimented on a quadruped robot in [43] and recently on a torque-based robot in [22]. In [53] 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. However, our trials in [2] did not confirm such superior performances.

Finally [14] proposed a mix of projection and QP approach. It was applied mainly for walking and jumping motion control of different virtual characters. This work served as the low-level control for the planning in [35].

All of the previous controllers are local and do not anticipate on the robot behavior considering future tasks. Task-based Model Predictive Control was firstly considered for stylized human locomotion [51]; the input of the preview is originated from motion capture. Recently [24] tackled this problem in a more general weighted prioritized formulation and also proposed an excellent review of the task-based approaches and how they relate to basic optimization schools. In [2] a model-preview controller for general multi-contact motion is also proposed. This work is based on a reduced model (CoM) preview and whole-body second stage motion generation, both writes as QP. Multi-contact whole-body non-linear formulation is proposed in [30] and [42] but does not yet meet time computation requirements. In practice, climbing ladders does not imply dynamics that requires having a preview of next steps to be integrated in the current control computation. Our investigations revealed that a preview controller 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 security purpose.

In climbing situations, all critical tasks that constitute constraints such as non-sliding, equilibrium, state variables limitations, non-desired collisions are critical and in fact of similar priority. Other tasks can be weighted in the cost function to be achieved at best and monitored. 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 in which strict prioritized formalisms do not behave well.

5.1 Model-based QP multi-contact controller

We redesigned the weighted-task framework QP developed in [8] to fit real-time constraints and be efficiently implemented as a low-level controller. The data we need are the same as for 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 $\mathbf{x} = [\ddot{\mathbf{q}}^T, \lambda^T]^T$ where $\ddot{\mathbf{q}}$ is the joint acceleration vector and λ is the vector of linearized friction cone base weights. The vector of contact forces $\mathbf{f}$ is equal to $K\lambda$ 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 computed from the QP solution and the dynamic model. The QP controller writes:

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

subject to

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

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

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

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

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

The constraint 1) accounts for torques 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 the contact points Jacobian matrix.

The constraint 2) fulfills zero acceleration for bodies in contact (no-sliding). In all the works we cited, 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 $i \in \mathscr{I}_{\text{contact}}$. However, we noticed, in practice, that countering the contact body velocity v_i leads to a better numerical behavior. Notice that the controller is computed on a model basis and is therefore a simulated model of the robot. We also added $S \in \mathbb{R}^{n,6}$, a selection matrix that allows to free directions to be eventually controlled in impedance or admittance.

The constraints 3) and 4) enforce joint speed and range limits and use a velocity damper ξ, see [26], and q_s as a security range. The constraint is activated only when $q \le q_i$.

The constraint 5) deals with collision avoidance (that we integrate in the controller instead of checking *a priori* or *a posteriori*). Relatively to [26] we 'track' one witness point per link or body when paired for collision checking. δ is the distance between a pair of bodies computed with the SCH library [19][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 $\delta > \sigma_\delta$ (σ_δ is a predetermined distance threshold) or obtained from one witness point's surface's normal. $\dot{N}$ is computed by finite difference. Our QP controller 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 δ_i can be adapted on-line to be different for each pair of bodies so as not many distance constraints are active at the same time. But this is rather an indication of a potentially dangerous posture to avoid.

The QP objective function is made of two terms: a sum of weighted least-squares term involving tasks errors, noted $E_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$, and a damping term with weight w_λ which ensures that the Hessian matrix is positive definite. For ladder climbing we only use the Set Point objective task [1][14][6] written as:

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

with $\mathscr{T}_i \in \mathbb{R}^n$ a 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 A. In constraints 3) 4) and 5) velocity dampers ξ can cause large deceleration and may

---

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

lead to problems if not well tuned. We use the following expression to compute ξ once (i.e when the constraint is firstly triggered and the velocity damper is activated):

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

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, ξ_{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

Once 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. From a quick review of literature, trials of common (free) solvers, and discussions we initiated with several community researchers we decided to benchmark two QP solvers: LSSOL [?] (cold and warm start) and QLD [?].

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

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

We used an i7 2.6GHz laptop (see later Fig. 13). As we can see from the 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 controller solver. During our experiments (section 6.2), we noticed that the median computation time of the whole problem (cost function, constraint matrix, distance query and QP solving) during ladder climbing (3 or 4 contacts) is $\simeq 1$ms.
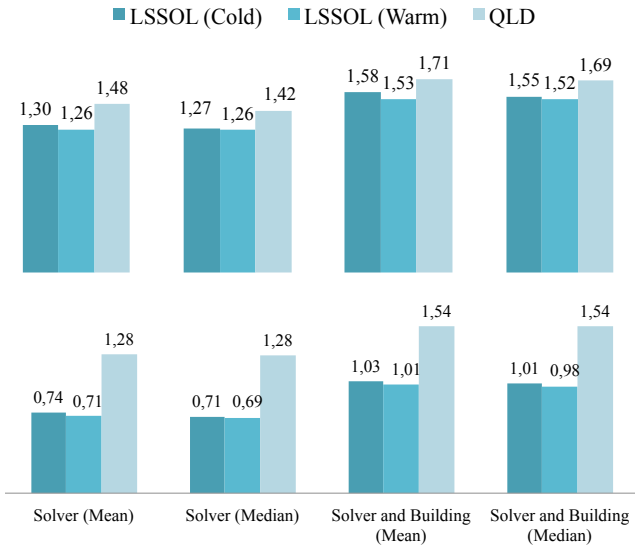
Fig. 7 Benchmarking the QP solver LSSOL (Cold and Warm start) and QLD for a four contact transfer task (first line) and the leg transfer task (second line), see also section 6.2). The computation time is given in milliseconds (ms) on the top of each bar representing the trials. We represent mean and median times for the solver alone (i.e. with the problem already set) and the solver plus the QP building, which integrates other computations such as the dynamics, Jacobians, distance computation with SCH [19], etc. Both benchmarks highlight the superiority of LSSOL Warm start w.r.t QLD and LSSOL Cold start.

5.3 Dealing with ankle shock absorbing compliance

At each ankle of HRP-2, there is a shock absorbing compliant mechanism linking the feet to the legs. 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 makes the attitude of the robot hard to control and this is the reason why a dedicated HRP-2 embedded stabilizer is necessary to achieve stable walking. The HRP-2 built-in stabilizer is well designed for walking on flat terrains and assumes coplanar contacts. Therefore, it has to be shut off in climbing or any non-coplanar multi-contact motion. This effect is similar to having passive joints at each ankle.

In order to compensate for the ankles' compliance, we estimate its resulting effect using the robot embedded IMU and inverse kinematics. We model each compliance as two revolute joints put in the robot's foot, see Fig. 8. Each leg (contact) is modeled as a fixed base with 2 dof and an end effector going through the IMU. Since in our experiments, two contacts at least are necessary, we are always having at least one closed-kinematic chain between contacts. We exploit this fact to estimate the 4 joints angular values that model the compliance effect. In the example illustrated in the Fig. 8, we will consider all the open kinematic chains

that go through the IMU and write, at the IMU frame, the conditions to close the kinematic chain in position, and secondary (at best) in orientation considering the least possible motion. The problem can be solved by optimization or prioritized task frameworks. We assume that the joint encoders and the IMU are reliable, and that we know the contact types (e.g. planar, cylindrical).
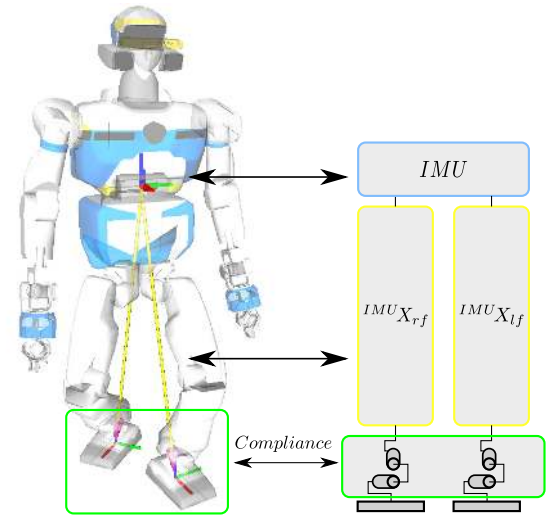

Fig. 8 Compliance kinematic model.

Let $\mathbf{q}_{\text{model}}$ be the robot configuration used by the controller and $\mathbf{q}_{\text{estimate}}$ be the robot configuration that accounts for the compliance. First, we tried 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, resulting in falls or oscillations. Instead, we reduce the dynamics of the motion with the following definition of the task error (that would apply to any position and orientation tasks $\mathscr{T}$):

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

where $K_p \gg K_i$, T_i and T_e are the task insertion and removal times. This allows converging to a zero error with slow dynamics. See illustration of such a control law result later in Fig. 19, section 6.3.

5.4 Gripper/rung contacts

The ankle's compliance can cause gripper/rung contact loss. To fix this issue we implemented a simple force control. This is made possible since we can release the null velocity constraint on a chosen axis (see selection matrix S in section 5.1). When the force goes below f_{c_σ} we remove the null
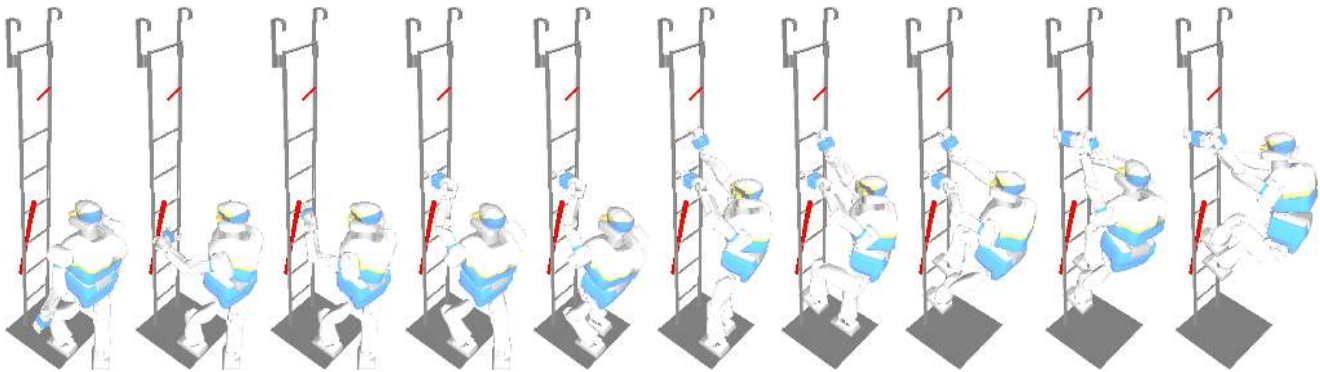
Fig. 9 Snapshots from simulated climbing of the vertical ladder used in real experimentations. We added an obstacle (red tube) and assumed the seventh rung broken. The second and third image (from the left) show the arm going around the tube obstacle.

velocity constraint on the insertion axis of the gripper's (z-axis) and add a position task with high weight. The targeted position is as follows:

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

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

6 Experiments and results

6.1 Simulated scenarios

The Fig. 9 illustrates climbing the ladder used in our real experiments (see next section). 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 while in the phase of grasping the fifth rung, and that of the left leg during the transfer from the first to the third run, on which the tube is posed. These motions are different from those generated in the absence of the obstacle. Also, in the absence of the seventh rung, the MCP still manages to find a combination of rung and stringer grasps for the last phase of the climbing.

The multimedia accompanying the paper for this simulation is annotated with the computed contact forces, actuator torques at the grippers, the COM and 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 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 confirm the correct behavior of the FSM geometric discrepancies recovery and the guarded motion tasks prior to real experiments on the robot. The Fig. 11 illustrates the MCP result obtained from a simulated scenario of a real set-up that is already available at our AIST experimental room. The ladder and scaffolding settings are modeled with a precision of 1mm.
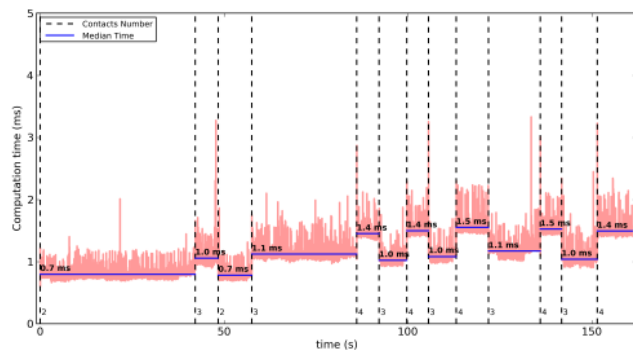

Fig. 10 QP controller computation time. Notice that it is below the critical 5ms, which means that the QP runs in real-time for an embedded similar processor. The dots vertical lines show contact transitions; the number of contacts is mentioned nearby each line.

Fig. 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.

This simulation assumes firm grasps on the rungs and the stringers. Note that our 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 without specific or dedicated phase planning. Yet, from the many simulations we made, not all generated contact stances plans where successfully reproduced by the QP. Also, it took more time for the planner to find the ladder-to-scaffolding transition.

6.2 Experiments with HRP-2

For our 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 precom-
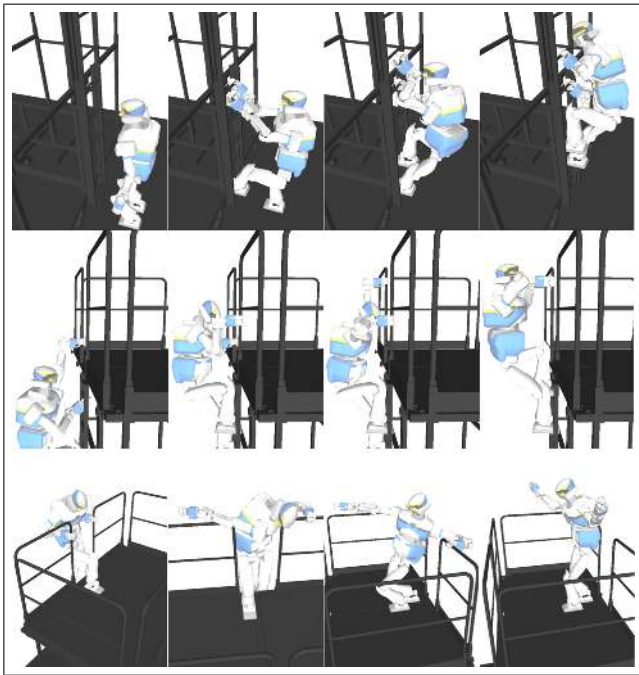
Fig. 11 HRP-2 climbing the ladder to reach a scaffolding. The first line illustrates phases from the ladder climbing by HRP-2. The second line shows some posture to negotiate the ladder-to-scaffolding transition. Finally, in the third line HRP-2 could reach the scaffolding.

puted initial posture near the ladder. This posture is computed so that the robot can reach the first rung. All our experiments are performed with the stabilizer switched off. Instead, we use the ankle compliance compensation described in section 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.
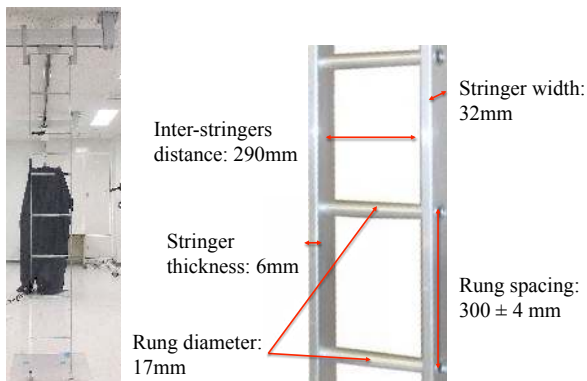

Fig. 12 The vertical ladder (left) and its parameters (right) used in HRP-2 climbing experiments. It comprises eight rungs in total and is fixed between the robot security gantry crane holder and the ground.

Our control architecture is split on two computers:

– the control computer that runs the QP Controller and the Compliance estimator;
– the HRP-2 on-board computer that reads sensors input and runs the PD controller

The two computers communicate with a direct Ethernet link and send information with a UDP network bridge (see Fig. 13).
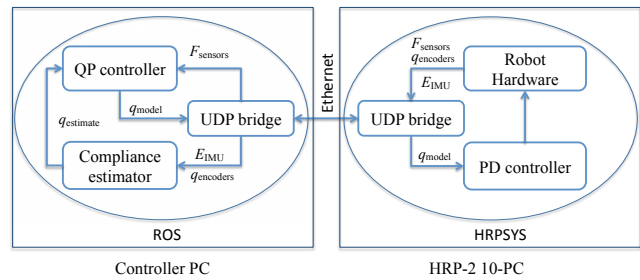

Fig. 13 A simplified view of the hardware and software architecture. QP Controller is the controller with the FSM. Robot Hardware is a process that reads and filters the robot sensors data. The UDP bridge is a process allowing the two computers to communicate through an Ethernet link.

The QP controller is developed using the `ROS` middleware[5]. The controller part is written in `C++` while 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 what happens while controlling the robot. This architecture is also less dependent from the robot on-board computer system. It 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 of 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 minimized damage in accidental or malfunction situations. We also developed debugging tools, logs, interactive commands and intermediary sequential steps validations. Before achieving a complete autonomous climbing, we went through different assessment phases. In all cases, and prior to any experiment, we switched off the recovery parts of the FSM, assuming the contacts to occur as expected, and played the entire climb-

---

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

[6] The code for dynamic computation can be found at `https://github.com/jorisv/RBDyn` . The code of the whole controller will be made available soon.
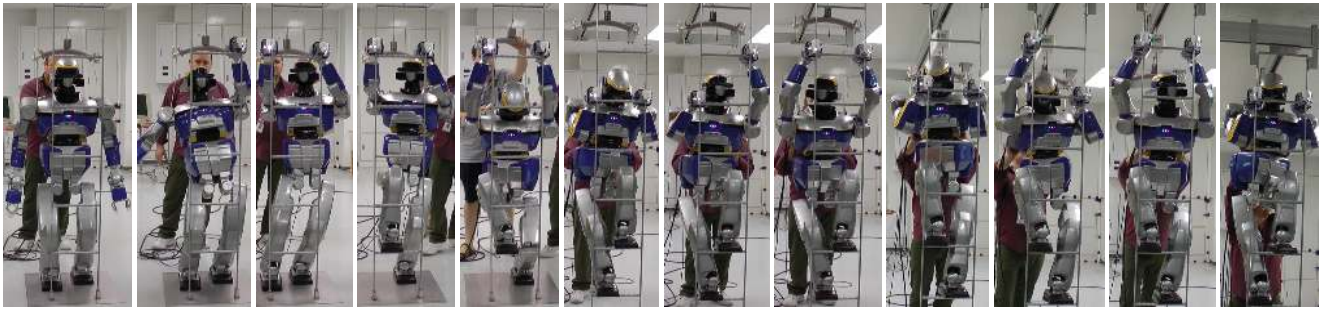
Fig. 14 Ladder climbing with off-line generated trajectories. When needed, in order to prevent change in contact posture or a gripper to lose contact during arm transfer, adjustments of the grippers positioning on the stringers is made by the operator (behind the robot) through light pushes toward the ladder. In this experiment, the operator compensates for the lack of firm grasps of the ladder's stringers, but does not intervene during phases where the two hands are grasping each stringer (four contact lifts of the HRP-2 body, feet transfer or positioning). In this first battery of tests, the HRP-2 could climb four rungs (the maximum possible, as the head would reach the roof and the protection ropes cannot be tightened).

ing motion with the robot in the air. This step was useful to confirm that the motion was indeed without self-collision and doable.

Because the width of the ladder's rungs did not allow having both feet on a same rung, climbing was made in two main phases:

1. Arm transfer (creation or removal of a gripper contact), which is always made while maintaining three contacts (two feet and one gripper).
2. Leg transfer (creation of removal of a leg contact) which is coupled with whole-body lifting (because it is not possible to position two legs on a one same rung at a time) and always made with two grasp contacts.

This strategy is somehow similar to the transverse mode in [55] but differs from those chosen for the customized climbing robots in [25][20].

– Preliminary trials: grasps on stringers
We first considered a MCP where climbing is made by alternating grasps on the stringers. Also, we forced the contact areas of the stringers to be nearby (up) the rungs, see Fig. 14. The reason for this choice is to prevent the gripper from a vertical sliding during limbs or body transfers. We generated a multi-contact plan, then stored in a file the trajectory that was generated by the QP controller in simulation. The climbing stored trajectory is then played with the robot in the air, and then directly on the ladder. 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. By doing so, our aim was to assess (i) the capability of the grippers to maintain contact on the stringer despite their design drawback (see Sec. 6.3), and (ii) the power capability of the robot to lift its body by the strength of legs and arms.

As a result, we confirmed the following facts, see [**?**]:

1. the grippers are not capable to hold a gripper/stringer contact during limb transfer: indeed, when a grasp is re-

leased to regrasp another part of the stringer, the robot would change the contact posture (rotate around the three other contacts) which result in substantial discrepancies or loose the other gripper/stringer contact, which results in the robot to fall from the ladder;
2. in a four contact configuration, the robot could lift its body autonomously 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 on the Fig. 14.

– Trials with grasps on rungs
Since it is not possible for HRP-2 to remain in contact when one of the grippers is released from stringers to be repositioned, we exploit contacts between grippers and rungs exclusively. Because the rung diameter is greater than the stringer width. We also increased the gains of the PD controller of each gripper actuator. Therefore, the second battery of experimental tests consists in using rungs and in trying a fully autonomous climbing in close-loop control with the use of the FSM.

Our first trial make HRP-2 climb up until both feet are 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 trial, which is illustrated by the two first snapshots of the Fig. 1. Notice that in this case, the robot grasp 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 confirmed this experimental phase, we attempted to go further by repositioning the left arm then the left leg. This is shown by the third and fourth image on the 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 Sec. 6.3).
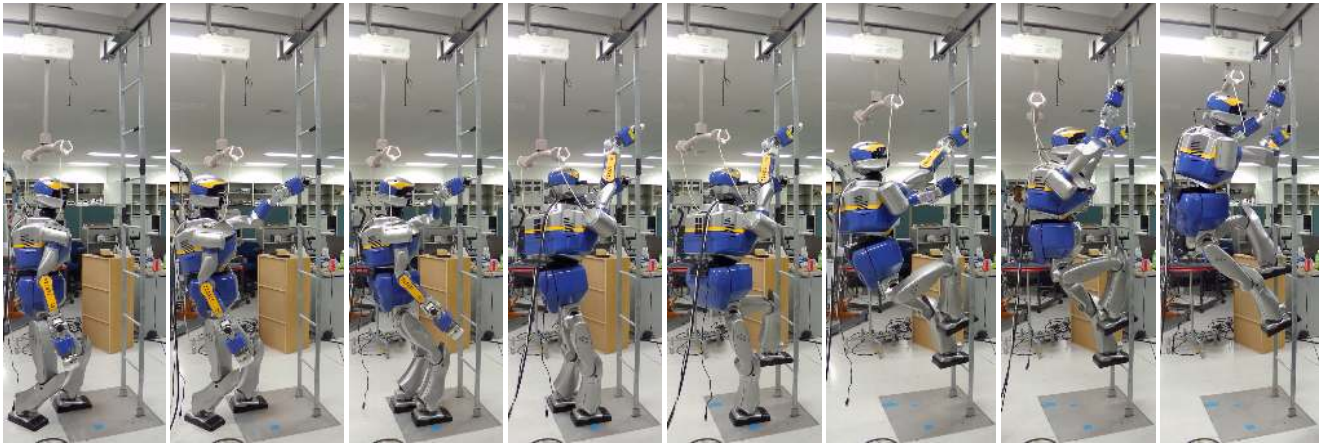
Fig. 15 From left to right: initial posture of HRP-2 in front of the ladder, left arm grasps the fifth rung, right leg brought near the ladder, right arm grasps the sixth rung, left leg on first rung, right leg and second rung accompanied by an entire body lift from the ground, left arm repositioned to grasp the seventh rung (notice 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 accompanied by whole-body lift. The eighth rung cannot be grasped and the security strings cannot be stretched to secure the robot.

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 grasped. This light rotation wouldn't occur with a firm grasp of the rung. We could compute a posture that minimizes the moment around that axis or even by prevent by hand such a rotation to occur. In fact, having a light rotation wouldn't be a problem, if not for the next problem.

The second problem is due to reachability limits. Since we use only rungs and only one foot can be put on a rung, 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 a teleoperation mode where the user adjusts the gripper, if needed, and to close it.

Finally, a third problem is that in this configuration, the gripper cannot keep closed 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 punctual adjustments, we could achieve the complete cycle of climbing as illustrated in the Fig. 15. The user interventions are minimal and compensate only hardware minor limitations. We can claim that an entire full autonomous climbing of an industrial norm ladder by HRP-2 is possible, providing hardware limitations can be overcome.

Fig. 16 shows, as ground truth the normal forces computed by the QP controller, and those measured from rough
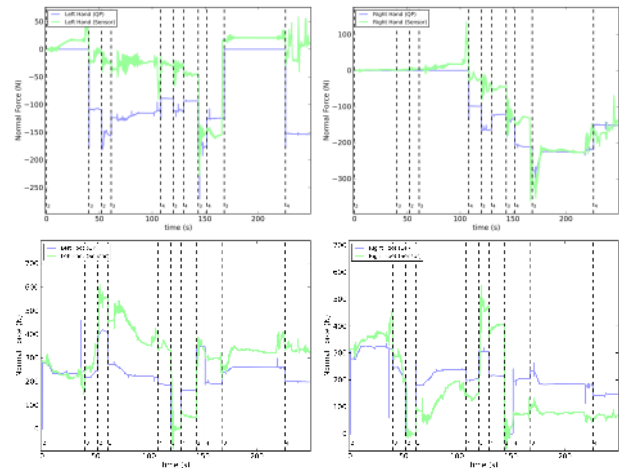

Fig. 16 Recorded force data from the experiment illustrated in Fig. 15. QP output normal forces versus real force sensing data from left and right hands (first line) and left and right foot (second line). Notice, Fig. 2, that we do not use the force output from QP controller.

force sensing in the wrists and feet, i.e. without off-set calibration, or filtering. Since we do not use force data in the control loop, we did not perform any calibration or filtering. Moreover, the changes on HRP-2's inertia w.r.t the factory model are not accounted for. Despite our lack of force sensing consideration, it can be seen that the QP controller predicts a plausible choice of force distribution. These results are extremely encouraging for future work. Indeed, reliability of predicted force computation would allow exploiting them for posture adjustment, internal forces reduction and balance. Reliable force prediction would also be used for on-line fault or problem detection from force discrepancies monitoring.

Because of the lack of heat and torque monitoring of the robot's actuators, we use an infra-red camera to monitor
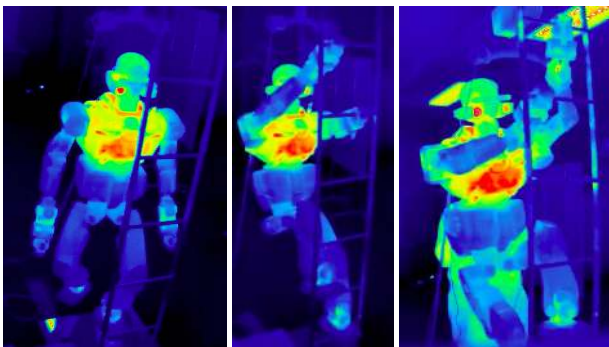
Fig. 17 Capture of infra-red camera monitoring of HRP-2's actuators during third phase climbing experiments. From left to right: initial posture, first lift, second lift. Thermal intensity range from cold (dark blue) to hot (red). In the last picture, the operator is behind the robot.

the robot during climbing. Fig. 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 location spots are distinctly highlighted from other parts. In particular, wrist and ankle actuators are the most solicited.

6.3 Discussions

To our best knowledge, 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 answered our second investigation objective, by showing that HRP-2 has the capability to climb vertical ladders, what no humanoid platform proved up-to-now. As for the third objective of our investigation, we capitalize 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 revealed by our experiments is the HRP-2's grippers design, which restricted the possible climbing variants and strategies. In order to explain the problem in technical terms, we illustrate the two possible ladder grasps: gripper/rung or gripper/stringer. Fig. 18 shows the two previous grasps with the gripper closed completely around the stringer and the rung respectively. It is easy to understand from Fig. 18 that the grasping contact-constraint configuration is not blocking. Notice, in both grasp cases, the large gap that remains inside the closure and within which the rung or the stringer is potentially free to move (e.g. forward) or rotate (e.g. around the orthogonal axis of rung, hence yawing). We mentioned in section 6.2 that when a grasp is released, yawing could occur in some postural configurations. This could result in unilateral contact to slide,

and as a consequence, substantial posture discrepancies to occur. When both grippers grasp the ladder, the closed kinematic chain linking both arms and the leg contact would not allow yawing during motion.
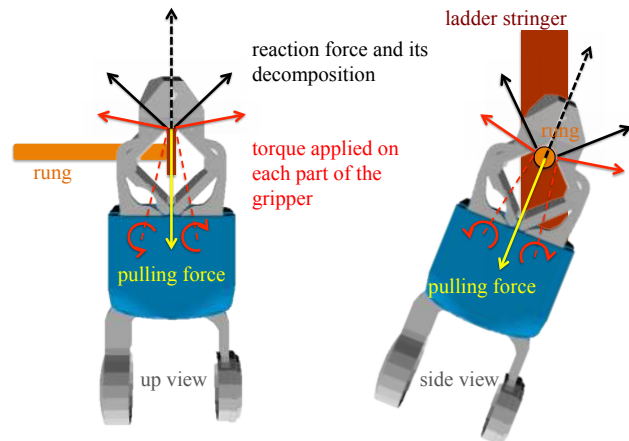

Fig. 18 Disposition of the rungs and stringers within the HRP-2's gripper once closed. The reaction to the pulling forces when they apply to the ladder's stringer or rung can be decomposed at contact spots into forces between the stringer/rung and the gripper's fingers. These contact forces can also be projected onto the line (dashed red) linking the contact point to the finger's rotation axis. This leads to two force vector components, the one along the dashed red line, and the one orthogonal to the latter one, illustrated by 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 be brought out of the gripper.

Moreover, each gripper has a limited grasping power. As a consequence, the grasp with the stringer or the rung often cannot be maintained. Fig. 18 illustrates this limitation with a detailed technical explanation in the caption. In brief, the pulling forces apply at contacts situated in the weakest part of the gripper. To circumvent partly this problem, we increased temporarily the gains of the gripper's servo motor. This temporary solution allowed maintaining the grippers closed during rung grasp in more situation, but was not sufficient in the case of stringer grasps. Open-loop experiments (with the human operator adjusting contact posture when needed) showed that if the ladder stringers can be grabbed firmly, the ladder climbing is less constrained, the robot has more space to be near the ladder and this may offer the planner better solutions to alternate and combine stringer/rung grasps as in section 6.1.

– Ankle compliance
Fig. 19 illustrates the recovery due to the posture discrepancy originated from ankle compliance. Yet discrepancies are not fully corrected. Discrepancies may create high inter-

nal forces, as also experienced in [20], and moments may easily make the robot yaw when releasing a contact. All these were observed in practice, because not only the contact points are almost coplanar, but they are nearly aligned, which result in yawing if the moment cannot be controlled.
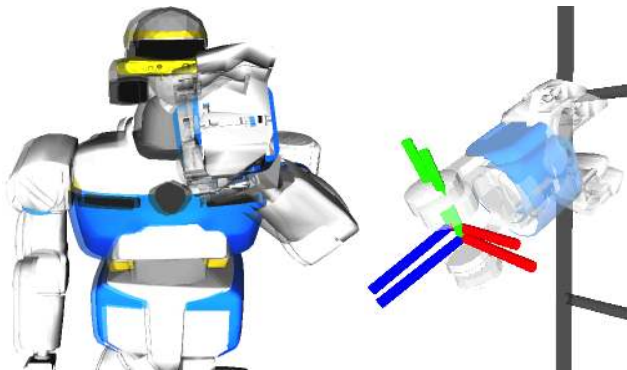

Fig. 19 Illustration of the effect of ankles compliance compensation. The transparent (clear) robot is the posture obtained from the (QP) model whereas the darker robot is the posture computed after compliance compensation and servoing using eq. 12.

Compliant shock absorbing mechanisms in the ankles (others are also present in both wrists) absorb contact shocks during contact formation and removal, but also light perturbations during transfer motions. Yet, they won't be able to absorb substantial perturbations during motion. Torque controlled humanoid would nicely comply to such perturbations, but a stiff position-controlled humanoid behaves like a 'rock', and any substantial perturbation, yawing or posture mis-adjustment means generally contact sliding up to losing unilateral contacts. 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, ε the servo position error and $\mathscr{D}$ the feedforward term. This idea is also discussed in [33], 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 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 in real practice environments. 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 [55]. For instance, by computing the CoM trajectory with a preview of up-coming contact, such as in [2]. This is certainly not necessary since vertical ladder climbing requires slow motion strategies at contact

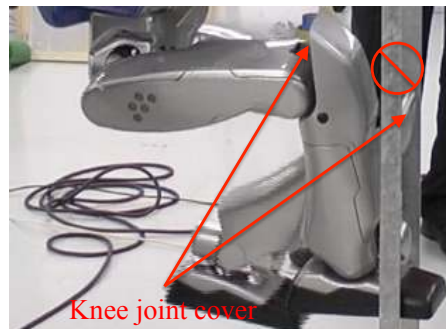formation (including grasps) and removal and we do not use hook-like designed grippers.

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

In one experiment, the knee cover in Fig. 20 was stuck on a rung during left leg transfer from 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 section 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 climbing vertical ladders having industrial norms with the HRP-2 humanoid robot. In section 1 we stated three objectives behind this work.

As for our second objective: experiments on different grasping strategies 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 in order to perform such tasks.

Concerning the third objective: we found that the grippers design is very limiting whereas firm grasps capability is key (as for humans) to climb up and down ladders efficiently and prevent from yawing during contact release. Following these experiments, hardware modifications are performed to change the gripper's clamps into new ones that adapt to the grasped ladder's bars, handrails, stringers, etc. The arm and leg's links are slightly elongated to avoid cases of reachability encountered in some key configurations when only rungs can be used.

As for the first objective, we believe that there remains work to be done to enhance the robustness of the controller and the planner. Also, vision 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 clouds [12], other recent work

suggests that contact areas can even be extracted and understood directly from point clouds [15].

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

A 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 [37].

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

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

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

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

References

1. Abe, Y., da Silva, M., Popović, J.: Multiobjective control with frictional contacts. In: Eurographics/ACM SIGGRAPH Symposium on Computer Animation, pp. 249–258. San Diego, California (2007)

2. Audren, H., Vaillant, J., Kheddar, A., Escande, A., Kaneko, K., Yoshida, E.: Model preview control in multi-contact motion– application to a humanoid robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Chicago, Illinois (2014)

3. Bevly, D.M., Farritor, S., Dubowsky, S.: 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 (2000)

4. Bouyarmane, K., Escande, A., Lamiraux, F., Kheddar, A.: Potential field guide for humanoid multicontacts acyclic motion planning. In: IEEE International Conference on Robotics and Automation, pp. 1165–1170. Kobe, Japan (2009)

5. Bouyarmane, K., Kheddar, A.: 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 (2010). DOI 10.1109/ICHR.2010.5686317

6. Bouyarmane, K., Kheddar, A.: 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 (2011)

7. Bouyarmane, K., Kheddar, A.: Humanoid Robot Locomotion and Manipulation Step Planning. Advanced Robotics 26(10), 1099–1126 (2012)

8. Bouyarmane, K., Vaillant, J., Keith, F., Kheddar, A.: Exploring humanoid robots locomotion capabilities in virtual disaster response scenarios. In: IEEE-RAS International Conference on Humanoid Robots. Businness Inoovation Center, Osaka, Japan (2012)

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

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

11. Brossette, S., Escande, A., Vaillant, J., Keith, F., Moulard, T., Kheddar, A.: Integration of non-inclusive contacts in posture generation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Chicago, Illinois (2014)

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

13. Collette, C., Micaelli, A., Andriot, C., Lemerle, P.: 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 (2007)

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

15. Eilering, A., Yap, V., Johnson, J., Hauser, K.: Identifying support surfaces of climbable structures from 3d point clouds. In: IEEE International Conference on Robotics and Automation. Hong Kong, China (2014)

16. Escande, A., Kheddar, A.: Contact planning for acyclic motion with tasks constraints. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 435–440. St. Louis, MO (2009)

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

18. Escande, A., Mansard, N., Wieber, P.B.: Hierarchical quadratic programming: Fast online humanoid-robot motion generation. The International Journal of Robotics Research (2014)

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

20. Fujii, S., Inoue, K., Takubo, T., Mae, Y., Arai, T.: Ladder climbing control for limb mechanism robot ASTERISK. In: IEEE International Conference on Robotics and Automation, pp. 3052–3057. Pasadena, CA, USA (2008)

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

22. Herzog, A., Righetti, L., Grimminger, F., Pastor, P., Schaal, S.: Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Chicago, Illinois (2014)

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

24. Ibanez, A., Bidaud, P., Padois, V.: 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, China (2014)

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

26. Kanehiro, F., Morisawa, M., Suleiman, W., Kaneko, K., Yoshida, E.: Integrating geometric constraints into reactive leg motion generation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4069–4076 (2010)

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

28. Kuindersma, S., Permenter, F., Tedrake, R.: An efficiently solvable quadratic program for stabilizing dynamic locomotion. In: IEEE International Conference on Robotics and Automation. Hong Kong, China (2014)

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

30. Lengagne, S., Vaillant, J., Yoshida, E., Kheddar, A.: Generation of whole-body optimal dynamic multi-contact motions. The International Journal of Robotics Research **32**(9-10), 1104–1119 (2013)

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

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

33. 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.: 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, China (2014)

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

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

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

37. Murray, R.M., Sastry, S.S., Zexiang, L.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Inc., Boca Raton, FL, USA (1994)

38. Nakai, H., Kuniyoshi, Y., Inaba, M., Inoue, H.: Metamorphic robot made of low melting point alloy. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2025–2030. Lausanne, Switzerland (2002)

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

40. Noda, S., Murooka, M., Nozawa, S., Kakiuchi, Y., Okada, K., Inaba, M.: 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, China (2014)

41. Ott, C., Roa, M.A., Hirzinger, G.: Posture and balance control for biped robots based on contact force optimization. In: IEEE-RAS International Conference on Humanoid Robots, pp. 26–33. Bled, Slovenia (2011)

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

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

44. Righetti, L., Schaal, S.: Quadratic programming for inverse dynamics with optimal distribution of contact forces. In: IEEE-RAS International Conference on Humanoid Robots. Businness Inoovation Center, Osaka, Japan (2012)

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

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

47. Salini, J., Padois, V., Bidaud, P.: 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, China (2011)

48. Sentis, L., Park, J., Khatib, O.: Compliant control of multicontact and center-of-mass behaviors in humanoid robots. IEEE Transactions on Robotics (2010)

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

50. Siciliano, B., Slotine, J.J.E.: A general framework for managing multiple tasks in highly redundant robotic systems. In: International Conference on Advanced Robotics, vol. 2, pp. 1211–1216. Pisa, Italy (1991)

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

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

53. Wensing, P.M., Orin, D.E.: Generation of dynamic humanoid behaviors through task-space control with conic optimization. In: IEEE International Conference on Robotics and Automation, pp. 3088–3094. Karlsruhe, Germany (2013)

54. Wieber, P.B.: On the stability of walking systems. In: IARP International Workshop on Humanoid and Human Friendly Robotics. Tsukuba, Japon (2002)

55. Yoneda, H., Sekiyama, K., Hesegawa, Y., Fukuda, T.: 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, France (2008)

56. Zhang, Y., Luo, J., Hauser, K., Ellenberg, R., Oh, P., Park, H.A., Paldhe, M.: Motion planning of ladder climbing for humanoid robots. In: IEEE International Conference on Technologies for Practical Robot Applications, pp. 1–6 (2013)

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