

Planning 3-D Collision-Free Dynamic Robotic Motion Through Iterative Reshaping

Eiichi Yoshida, *Member, IEEE*, Claudia Esteves, Igor Belousov, Jean-Paul Laumond, *Fellow, IEEE*, Takeshi Sakaguchi, and Kazuhito Yokoi, *Member, IEEE*

Abstract—We propose a general and practical planning framework for generating 3-D collision-free motions that take complex robot dynamics into account. The framework consists of two stages that are applied iteratively. In the first stage, a collision-free path is obtained through efficient geometric and kinematic sampling-based motion planning. In the second stage, the path is transformed into dynamically executable robot trajectories by dedicated dynamic motion generators. In the proposed iterative method, those dynamic trajectories are sent back again to the first stage to check for collisions. Depending on the application, temporal or spatial reshaping methods are used to treat detected collisions. Temporal reshaping adjusts the velocity, whereas spatial reshaping deforms the path itself. We demonstrate the effectiveness of the proposed method through examples of a space manipulator with highly nonlinear dynamics and a humanoid robot executing dynamic manipulation and locomotion at the same time.

Index Terms—Collision avoidance, dynamics, humanoid, motion planning, space manipulator.

I. INTRODUCTION

RECENT progress in motion planning research now makes it possible to compute complex motions that avoid collisions for complex robots in cluttered 3-D environments. This development can be attributed to efficient algorithms based on sampling-based planning methods [1]–[3]. In industry, such techniques are used to plan the assembly or disassembly sequence of products composed of many parts with complicated geometry [4].

Manuscript received October 19, 2007; revised March 28, 2008 and May 7, 2008. First published September 9, 2008; current version published October 31, 2008. This paper was recommended for publication by Associate Editor O. Brock and Editor K. Lynch upon evaluation of the reviewers' comments. The work of J.-P. Laumond was supported by the European Aeronautic Defense and Space (EADS) Foundation Project Zeuxis. This paper was presented in part at the IEEE Robotics and Automation Society International Conference on Humanoid Robots in Tsukuba, Japan, 2005, and the IEEE/RSJ International Conference on Intelligent Robot and Systems in Beijing, China, 2006.

E. Yoshida, T. Sakaguchi, and K. Yokoi are with the AIST/IS-CNRS/ST2I Joint French–Japanese Robotics Laboratory (JRL), National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba 305-8568, Japan (e-mail: e.yoshida@aist.go.jp).

C. Esteves is with the Facultad de Matemáticas, Universidad de Guanajuato, Guanajuato, 36000 Gto., Mexico.

I. Belousov is with HP Laboratories, 115054 Moscow, Russia.

J.-P. Laumond is with Joint French–Japanese Robotics Laboratory (JRL), Laboratoire d'Architecture et d'Analyse des Systèmes, Centre National de la Recherche Scientifique (LAAS-CNRS), University of Toulouse, 31077 Toulouse, France.

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org>, provided by the author. The size is 12 MB (11 MB after compression by ZIP).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2008.2002312

Those motion planning techniques are usually based on geometric reasoning in configuration space. We believe that the advances in this research field have now arrived at a stage where those techniques should be applied to dynamic motion planning for robots with many DOFs.

Typical applications that require dynamic motion planning include nonlinear space manipulators and humanoid robots that require dynamic balance during task execution and locomotion.

The difficulty of planning motions for those robots lies in the high complexity of geometric and kinematic planning of 3-D collision-free motion in complex environments while accounting for dynamic constraints. Due to this complexity, there have been few planning systems that can deal with such dynamic robotic systems.

In this paper, we propose a general and practical planning framework that integrates a geometric path planner and a dynamic motion generator. It is based on an iterative two-stage motion planning method, by exploiting the efficiency of sampling-based planning and advanced dynamic motion generation. Within the classification proposed in [5], our two-stage approach fits alongside state-space and sensor-based approaches. In the first stage, a “path” is generated by geometric and kinematic planning, which is transformed into a dynamically executable “trajectory” through appropriate dynamic motion generators in the second stage. Due to dynamic effects, the path supporting the new trajectory may differ slightly from the initial path. Then, the output trajectory is again verified with respect to the collision avoidance by the first stage and reshaped if necessary. This process is iterated until a valid dynamic trajectory is obtained.

Although the necessity of reshaping for dynamic collision-free motion has been recognized [6], it has not been systematically addressed. The proposed general framework recaptures the two-stage method with a global view by introducing two types of iterative reshaping methods: temporal and spatial. The former resolves collisions by reducing the motion velocity along the generated trajectory, whereas the latter deforms the trajectory itself.

In this paper, we emphasize the practical aspect of our approach through realistic simulations and experiments. To our knowledge, there are currently few motion planners that can generate 3-D collision-free dynamic motions including locomotion and manipulation by humanoids with as much geometric and dynamic complexity as we deal with here.

The paper is organized as follows. Section II provides related work and states the main contribution. The proposed iterative two-stage framework is outlined in Section III. Geometric and

kinematic path planning utilized in the first stage is briefly presented in Section IV. Then, the trajectory reshaping methods used in the second stage, the temporal and spatial approaches are presented in Sections V and VI, respectively, together with simulation and experimental results. After discussing the limitations and future developments of the proposed method in Section VII, conclusions are given in Section VIII.

II. RELATED WORK AND CONTRIBUTION

The two-stage approach has also been studied as “decoupled planning,” in which a collision-free path is computed first and a time-scaling function is next derived to define the execution velocity. In such an approach, the problem is stated as finding a time-optimal scaling function, subject to the actuator limits, which allows the robot to precisely follow the path provided from the first stage [7].

The state-space method has been addressed as “kinodynamic motion planning” that includes the derivatives in the system state spaces [8], [9]. In this method, the next state of the system is estimated by taking into account its dynamics to find also time-optimal solutions. Although the dynamic motion is directly planned, the size of search space increases exponentially as the number of the states increases. It is also known that finding exact kinodynamic solutions is intractable even for 2-D problems [10].

For far more complicated robots such as space manipulators or humanoids, it is now reasonable to employ efficient algorithms based on sampling-based planning methods such as Rapidly-exploring Random Trees (RRT) [1], [2], Probabilistic RoadMap (PRM) [3], and all their variants (see [11] and [12] for recent overviews). The sampling methods compute a graph called a roadmap whose nodes are collision-free configurations chosen at random and whose edges model the existence of collision-free local paths between two nodes. One extension for sampling-based methods adjusts the obstacle size for easier planning [13], [14]. An iterative method has been proposed based on the automatic tuning of obstacle penetration tolerance [14]. In this method, the thinned obstacles are used to make the planning easier. Then, they are grown gradually by increasing the tolerance to deform the path so that the robot stays away from them. We will utilize this method in our planning scheme.

Spatial reshaping has been studied in various contexts. In robotics, a general reshaping method has been introduced in [15]. The idea is to perturb the input functions of the control system along the path to be modified, in such a way as to decrease an optimization criterion. However, this method requires the analytical expression of the control model, which is not the case for the humanoid robot we deal with. The concept of elastic band [16]–[18] is a more practical approach. An energy function is applied along the path, and an optimization algorithm allows path reshaping in order to minimize its cost. The energy function is derived from a distance computation between the moving body and the obstacles. Distance computation is a costly operation that often requires simplified geometry for a good performance.

Our spatial reshaping method is inspired by a technique for key frame editing in the context of computer animation [19].

This approach places more emphasis on the gradual transition from the colliding trajectory by maintaining the motion timing in constrained environments. Here, the complexity is determined by the resolution induced by the key framing, and less influenced by the geometric complexity.

One typical application of the proposed framework is space manipulation systems [20], [21]. Space manipulators have long arms in order to transport high payload and are usually tele-operated. Although path-planning methods have been proposed for space manipulators [22], it is not practical to analytically plan the trajectory of such a highly nonlinear system in a cluttered environment. In this research, we take advantage of capacity of a sampling-based planner by combining it with a dynamic simulator in a two-stage planning scheme. So far, automatic planning of 3-D collision-free motions in cluttered environments for such a highly nonlinear system has not been achieved.

Another application is planning of dynamic humanoid motions, which is an even more complex problem where we must cope with more than 20 DOF as well as dynamics. Kuffner *et al.* made the initial progress in this area and have been actively working on this topic. They proposed a method to generate collision-free dynamically stable humanoid motion using RRT [6] through a two-stage method. The random planner makes a search to find a collision-free path using precomputed statically stable configurations. The path is then transformed into dynamic trajectory by verifying dynamic zero moment point (ZMP) constraints. In case of collision, the trajectory is slowed down, which corresponds to temporal reshaping in our research. In this research, support state (double or single support) does not change and locomotion is not included. Khatib and his colleagues have been working on dynamic whole-body motion generation for humanoid robots by using task specification in operational space [23]. Their research on local reactive motions is being extended to be combined with global planner of mobile manipulators [24].

Several other studies have been conducted on whole-body motion planning for humanoids for specific tasks [25]–[27] and on geometrical and kinematic planning digital actor animation [28]. However, 3-D collision-free motion planning including simultaneous locomotion and manipulation tasks by humanoids has not yet been accomplished. Although simpler versions of spatial reshaping for humanoids have been reported in our previous work [29], [30], it has not been discussed in a global framework. Moreover, this paper provides a more detailed reshaping algorithm, an experiment in a more complex environment with extended analysis of the results.

The main contribution of this paper is to provide a general framework for motion planning including dynamics for robots with many DOFs to generate 3-D collision-free motions in complex environments, based on an iterative two-stage planning scheme.

As an additional contribution, we show practical applications of the proposed framework for two different types of robots: space manipulators and humanoids. These two robots correspond to typical examples of temporal and spatial reshaping whose scope is introduced in the next section.

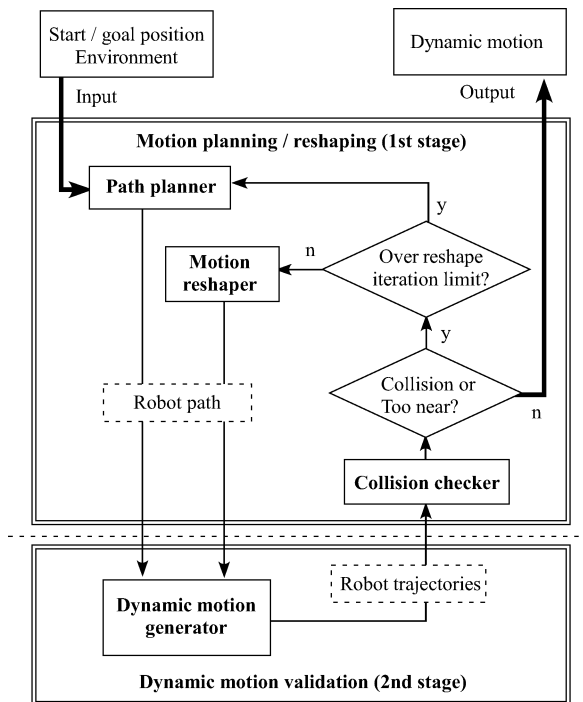


Fig. 1. Two-stage motion planning framework. In the first stage, the geometric and kinematic planner plan the collision-free path that is transformed into dynamic motion in the second stage. Then, the output dynamic trajectory is passed back to the first stage. This process is repeated until a collision-free dynamic trajectory is obtained.

III. GENERAL FRAMEWORK OF DYNAMIC COLLISION-FREE MOTION PLANNING IN 3-D

The proposed general framework of dynamic collision-free motion planning based on the two-stage planning method is illustrated in Fig. 1. It outputs dynamic collision-free trajectories from the inputs of initial and goal configurations together with the geometry information of the environment. The resulting dynamic trajectories should be ready to be given to low-level robot controllers.

The path planner finds a geometric and kinematic collision-free path in 3-D at the first stage (upper part of Fig. 1). Any available planning method can be used for this part. Then, in the second stage, the dynamic motion generator to transform the given path into dynamically executable robot trajectory (lower part in Fig. 1). A dedicated dynamic controller can be put depending on the application. The path planner will be detailed in Section IV.

The generated trajectory may deviate from the planned path due to robot dynamics, which may cause unpredicted collision with obstacles. The reshaper is placed in the first stage as a mechanism that interacts with the dynamic motion generator iteratively to remove those local collisions. This corresponds to the inner iteration loop in Fig. 1.

We introduce two temporal and spatial reshaping methods as summarized in Fig. 2. Iterative application and different types of reshaping have not been explicitly addressed in previous work of two-stage planning.

Temporal reshaping is described in Fig. 3. If collisions are detected in the trajectory, reshaping is applied to the colliding portion of trajectory by slowing down the control inputs. This procedure is repeated several times until a collision-free dynamic motion is obtained.

Spatial reshaping locally deforms the portions of the path whose corresponding dynamic motion is colliding (see Fig. 4). If the dynamically colliding local paths become blocked by the “grown” obstacles, the replanning process is activated, as shown in the bottom left of Fig. 2. In the replanning process, the path planner searches for another path that avoids the blocked passage. This corresponds to the outer loop in Fig. 1.

Temporal reshaping makes the planning framework converge once a statically stable motion is found. However, this condition does not guarantee completeness for planning of dynamic motions. There can exist dynamic motions that cannot be executed statically. Moreover, temporal reshaping may create unnecessarily slow motions. The practical interest of the spatial reshaping is to overcome this drawback, though it does not cover all possible dynamic motions.

Since our framework allows the integration of existing path planning and control techniques, we can choose among the variety of existing algorithms for those that apply best to the problem at hand. The proposed method is offline on the assumption that the environment is completely known.

Two applications of the proposed method, for space manipulators and humanoid robots, are shown in Sections V and VI to demonstrate the validity of the method.

IV. PLANNING GEOMETRIC AND KINEMATIC PATH

For the geometric and kinematic motion planner needed by the first stage, we have adopted an automatic parameter-tuning algorithm [14] that is appropriate for highly constrained spaces where the motion to be computed is close to the contact space.

The algorithm is characterized by iterative refinement of the path and automatic parameter tuning, as illustrated in Fig. 5. In this algorithm, the first path is computed allowing some penetration in the obstacles [see Fig. 5(a) and (c)]. Then, the current paths are iteratively replanned by decreasing the allowed penetration threshold. The cases of failure of the iterative process are automatically detected and solved. Concretely, when a portion of path cannot be locally modified into a collision-free one, then the search starts again to find a roadmap composed of the portions of the roadmaps that are collision-free [see Fig. 5(b) and (d)]. The method is parameter free in the sense that the penetration threshold is automatically tuned through the observation of the distances during the search. The steering directions are also automatically controlled by shooting random configurations in function of the already constructed roadmap. This algorithm has been verified to be efficient in constrained environments [14].

V. TEMPORAL RESHAPING: SPACE MANIPULATOR

The proposed method is applied to an on-board space manipulator equipped in Russian space shuttle “Buran.” Temporal

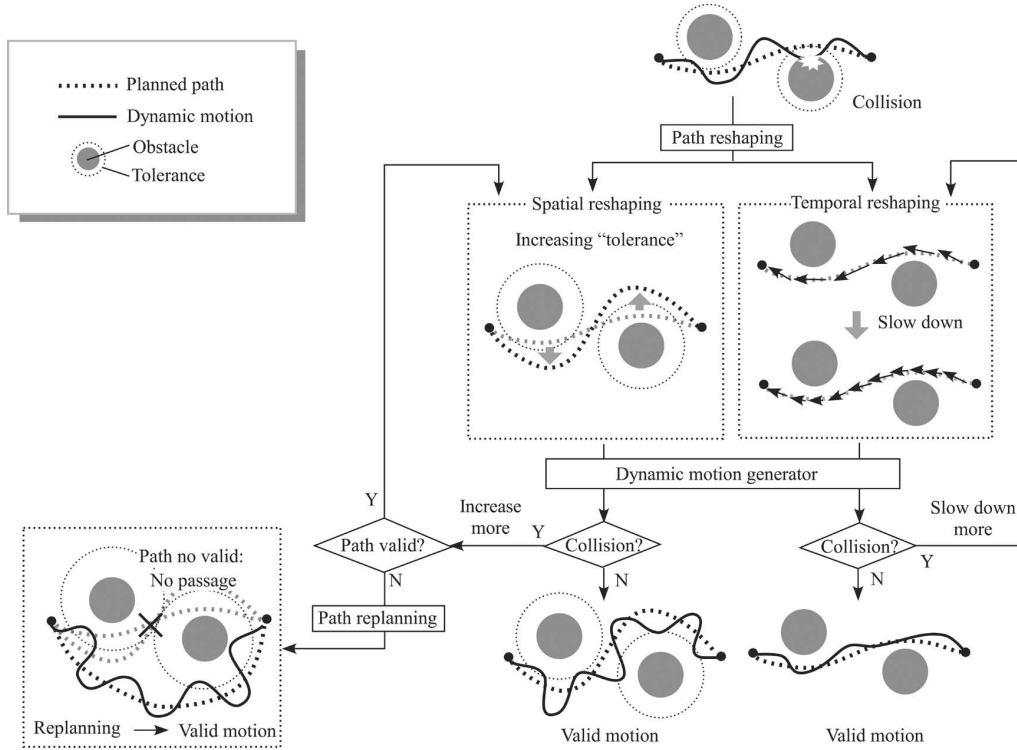


Fig. 2. Reshaping for valid dynamic motions. If collisions are detected, either spatial (left) or temporal (right) reshaping is applied according to the problem. Spatial reshaping may lead to replanning if the collision is not resolved by local reshaping (bottom left).

PLANWITHTEMPORALRESHAPING(q_{init} , q_{goal})

```

begin
  stop = False;
  path ← PLANCOLFREE( $q_{init}$ ,  $q_{goal}$ );
  vel ← CONTROLINPUTVEL (path, 0, T);
  while ¬stop or  $i < NumberTry$ 
    traj ← DYNMOTIONGENFROMVEL (vel, 0, T);
    if COLLISIONDETECT(traj) = True then
      velcol ← EXTRACTCOLLISIONPORTION(vel, traj);
      vel ← SLOWINPUTVELCOLLISION(velcol, vel);
      i = i + 1;
    else stop ← True;
  end
end

```

Fig. 3. Algorithm for temporal reshaping. Once a collision-free path is obtained, appropriate control inputs are searched by adjusting velocity of colliding portion.

reshaping is adopted to make full use of its property to converge to the planned collision-free path in the narrow passages. Spatial reshaping is not suitable because of the small clearance.

A. System Description

The manipulator is 15.5 m long and has 6 DOF. It has been developed as the virtual robotic test-bed (VRT), in accordance with project “Servicing” of Russian Space Agency. A dynamic simulator dedicated to space manipulator has been developed in this framework [31].

Fig. 6 shows a 3-D collision-free path planned in the first stage to dock the satellite by the Buran on-board manipulator inside the Buran cargo bay. The output of the path planner is a

PLANWITHSPATIALRESHAPING(q_{init} , q_{goal})

```

begin
  stop ← False;
  tol ← InitTolerance;
  invalid_paths ← ∅;
  while ¬stop or  $i < NumberTry$ 
    path ← PLANCOLFREETOL( $q_{init}$ ,  $q_{goal}$ , tol, invalid_paths);
    replan ← False;
    while ¬stop and ¬replan
      traj ← DYNMOTIONGENFROMPATH (path, 0, T);
      if COLLISIONDETECT(traj) = True then
        pathcol ← EXTRACTCOLLISIONPORTION(path, traj);
        tol ← INCREASETOLERANCE(tol);
        if ¬VALIDATE(pathcol, tol) then
          replan ← True;
          tol ← InitTolerance;
          invalid_paths.add(pathcol);
        else
          path ← RESHAPEPATH(pathcol, path, tol);
        else stop ← True;
      end
      i = i + 1;
    end
  end

```

Fig. 4. Algorithm for spatial reshaping. The portion of the path that collides is locally deformed by increasing the tolerance until no collisions are found. If the colliding local path is blocked due to increased tolerance, replanning procedure is called. Note that replanning is necessary for temporal reshaping.

finite sequence of waypoints, connected by linear interpolation in the configuration space.

The manipulator base can be regarded as fixed since the large difference in inertia renders the movement of the base

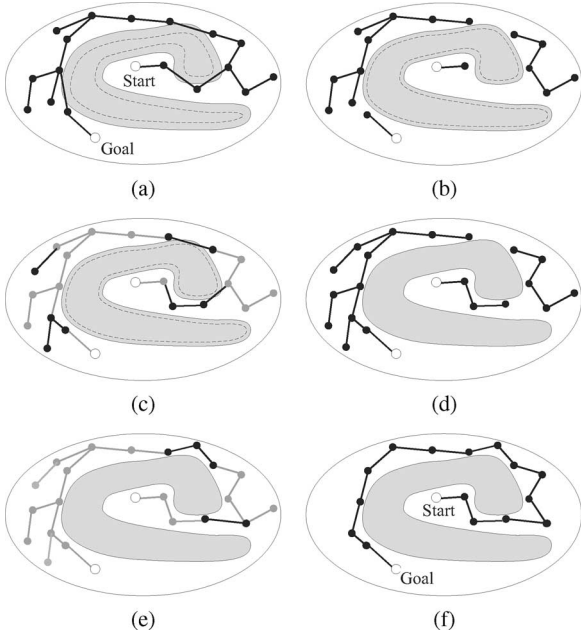


Fig. 5. Path planning algorithm based on RRT with automatic tuning of penetration [14]. (a) A roadmap is generated with a threshold: the path is allowed to penetrate in the obstacle. (b) Threshold is reduced and colliding edges are removed. (c) Roadmap is reconstructed by using collision-free portion of (b). (d) Threshold is set to zero and colliding edges are removed again. (e) Final roadmap is built starting from the partial roadmap (d). (f) Collision-free path is obtained.

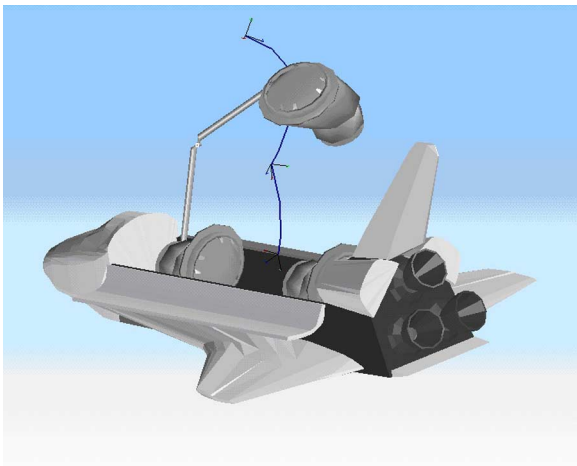


Fig. 6. Collision-free path before taking system dynamics into account.

negligible. However, in contrast to industrial robot arms, the dynamic behavior of the Buran on-board manipulator is extremely complex due to elasticity of its joints, large mass and inertia of its links, and the presence of nonlinear elements in mechanical gears. In particular, the following dynamic features have been revealed during investigation of the dynamic model of the Buran manipulator and experiments with real robot on the air-bearing test bed:

- 1) oscillation with large amplitude (up to 0.5 m) and small frequency (0.2–1 [Hz])
- 2) large accelerations and decelerations along path, resulting from the important inertia of the manipulator links.

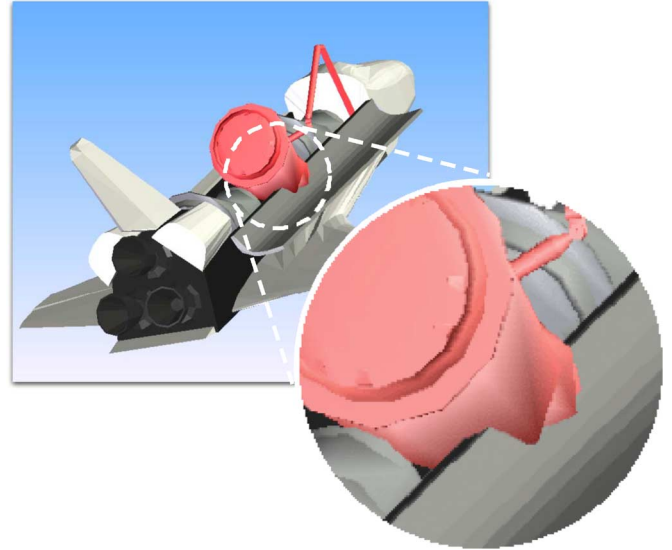


Fig. 7. Detection of the collision in the second stage where the dynamic trajectory is simulated.

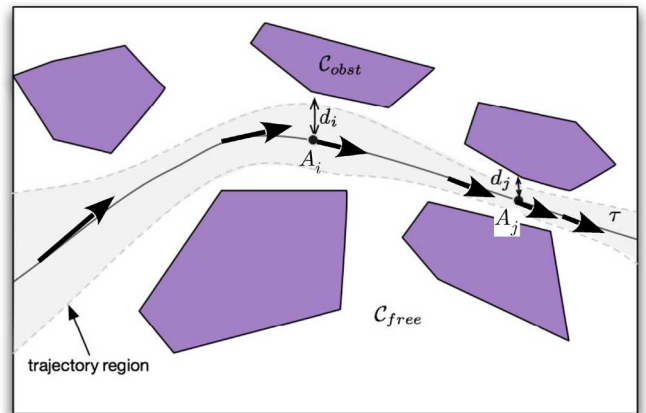


Fig. 8. Control points are attributed according to the distance to the closest obstacles where the velocities are controlled.

Fig. 7 illustrates the collision detected after applying the dynamic motion generator in the second stage. To prevent collisions, we introduce a temporal reshaping of the path.

B. Temporal Reshaping

Let us consider a geometrically and kinematically planned end-effecting gripper path. The path is discretized into “control points” A_i according to several criteria to determine the commands of velocity control at each point, as illustrated in Fig. 8. The velocity commands are validated using a simulator [31] that computes the dynamic motion of the manipulator.

The number of control points depends on the desired maximal linear and angular velocities for the end effector v_{max} and ω_{max} . Let the robot control period be t_{contr} (defines the frequency of the control inputs change at the high level of the robot control system). Then, the approximate number of necessary control

points is $N = T/t_{\text{contr}}$ as follows:

$$T = \max\left(\frac{L_p}{v_{\max}}, \frac{\phi_p}{\omega_{\max}}\right) \quad (1)$$

where L_p and ϕ_p represent the length of the planned path and difference of orientation angle.

Parameters v_{\max} and ω_{\max} define how fast the robot will follow the trajectory. Using these parameters, we can define the profile for linear \mathbf{v}_i and angular $\boldsymbol{\omega}_i$ velocities of the end effector along the trajectory:

$$\begin{aligned} \mathbf{v}_i &= \frac{d_i}{d_{i\max}} v_{\max} \frac{A_i A_{i+1}}{\|A_i A_{i+1}\|} \\ \boldsymbol{\omega}_i &= \frac{d_i}{d_{i\max}} \omega_{\max} \mathbf{n} \end{aligned} \quad (2)$$

where d_i and $d_{i\max}$ are the distance between the robot and the closest obstacle at point A_i and its maximum value $d_{i\max} = \max(d_i)$, \mathbf{n} is the vector of rotation between two end effector orientations in points A_i and A_{i+1} . Such a choice of \mathbf{v}_i and $\boldsymbol{\omega}_i$ allows for accelerated motion when the robot is far away enough from the obstacles (i.e., the coefficient $d_i/d_{i\max}$ is about 1), and decelerated motion when the clearance d_i is small (i.e., the coefficient $d_i/d_{i\max}$ is correspondingly less than 1).

Then, for each point on the trajectory, we calculate desired values for joint velocities, multiplying the vector $\dot{\mathbf{r}} = [\mathbf{v}_i^T, \boldsymbol{\omega}_i^T]^T$ by the inverse Jacobian matrix of the manipulator. If those velocities are derived, a proportional–derivative (PD) control law can be applied to determine the command for the manipulator.

Therefore, the approximation problem here is simply choosing admissible velocities to keep collision-free motion and to minimize overall time for trajectory following. We have used a dichotomy method to cope with that problem. This means that we first try maximum values of linear and angular velocities, which can be realized physically by the robot. If these values lead to collisions during dynamics simulation, we divide them by 2, try new values, and so on.

Note that in practice, we do not demand the end effector to arrive to the proximity of the current control point, such as in the aforementioned algorithm—regardless the end effector will achieve the control point or not, the control system switch it to the next one with fixed frequency $1/t_{\text{contr}}$.

C. Simulation Results

The resulting dynamic trajectory for the task of docking the satellite inside the Buran cargo bay is presented in Fig. 9. The payload for this task is 3000 kg.

The number of control points necessary to guarantee collision avoidance is 26. The algorithm provides the dynamic solution for the extremely tight space—the range of minimal distances between the robot with satellite and workspace varied from just 0.02 – 0.03 m to 0.18 m for a 15.5-m-long manipulator with a 4-m-long satellite. Trajectory length is 18.20 m and total time to perform the operation is 98.09 s. The average end effector linear speed is 0.186 m/s. This value exceeds maximum speed for Buran manipulator (0.1 m/s) with the maximum payload

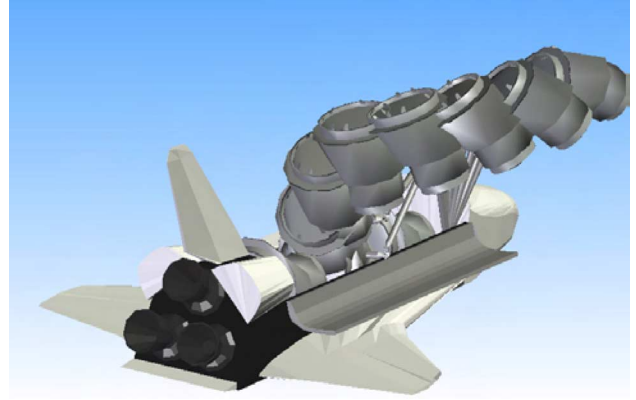


Fig. 9. Dynamic motion after applying the temporal reshaping. The collisions presented in Fig. 7 disappeared.

(50 000 kg) and just a little below its maximum possible speed without payload, equal to 0.3 m/s.

This result demonstrates the benefit of the proposed reshaping method since a very high velocity of the end effector becomes possible. Conventionally, the end-effector speed has been limited to 0.01 m/s because motions with higher speed without such planning may cause oscillations and collisions due to highly nonlinear dynamic behavior of the manipulator.

On a 1.2 GHz PowerPC G4, the computation time for kinematic path is 0.67 s and the dynamic motion generator takes 0.003 s to generate the dynamic trajectory.

VI. SPATIAL RESHAPING: DYNAMIC TASK BY HUMANOID

This section deals with an application of the proposed framework to an object transportation task by a humanoid robot. The humanoid robot must manipulate an object, avoid obstacles, and keep its balance while walking. We employ spatial reshaping to obtain high-quality motions through efficient dynamic motion generators for humanoid locomotion (e.g., [32]). Usage of temporal reshaping on quasi-static paths would make the robot motions unnecessarily slow.

A. Whole-Body Motion Generation

We utilize here a functional decomposition of the robot body that has already been applied to motion planning for virtual mannequins [28]. At this level, the robot is modeled as a geometric parallelepiped bounding box (see Fig. 10). Only that box and the object to be manipulated are considered with respect to collision avoidance. The robot motion is expressed by the planar position and orientation of its waist $\mathbf{r}(x, y, \theta)$ (3 DOF) and the object motion by its position and orientation $\mathbf{R}_o(x_o, \boldsymbol{\Theta}_o)$ (6 DOF) with respect to a global coordinate system Σ_0 . The configuration space to be searched is then nine-dimensional. To explore such a space, we use a sampling approach as in [28].

In our case, the robot path is planned as Dubins curves composed of line segments and arcs of a circle [33]. Given the configuration of the robot waist and object, the joint angles (\mathbf{q}_u) of the upper-body motion are derived by using inverse kinematics (IK) described later.

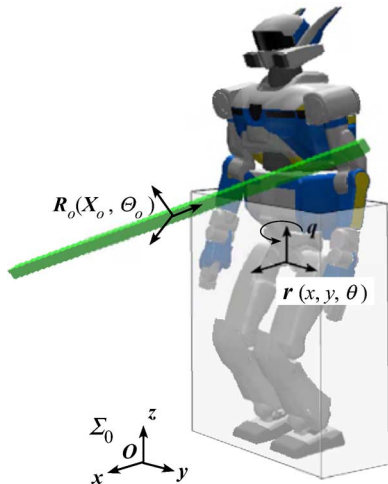


Fig. 10. Humanoid modeled by rectangle box with a bar. In the first stage, the geometric and kinematic path planner generates collision-free path for the 9-DOF system including robot waist (r , 3 DOF) and object (R_o , 6 DOF).

Then, at the second stage, the planned motions r and q_u are given to the dynamic pattern generator [32] of humanoid robots to transform the input planar path into a dynamically executable motion. The walking pattern generator is based on preview control of ZMP proposed by Kajita *et al.* [32]. The reference ZMP trajectory is derived from the foot placements obtained from the planned planar robot path. Based on preview control of ZMP position for an invert pendulum model, this method is able to generate dynamically stable biped walking motion that always maintains the ZMP inside the support polygon formed by the foot (feet).

Moreover, the pattern generator accepts upper-body motion q_u as auxiliary input to compute the mixed whole-body motion for desired upper-body movement and walking motion in the following manner. First, the pattern generator estimates the difference of the resulting ZMP from the reference when applying the desired upper-body trajectory for manipulation motion. This disturbance on the ZMP may perturb the robot stability. Then, the biped locomotion trajectory is slightly modified by shifting the waist position to compensate this difference to maintain the ZMP inside the support polygon. The dynamically stable motion is thus obtained that performs the manipulation and locomotion at the same time [32].

As a result, the whole-body humanoid motion is computed as the 6 DOF waist position and orientation $R_d(X_d, \Theta_d)$ and joint angles of whole body (q) at sampling time of 5 ms by taking dynamic balance based on ZMP into account.

B. Smooth Spatial Reshaping

Given the list of whole-body configurations at each sampling time as input, each configuration is tested for collision. For each colliding portion of the trajectory, the value of minimum tolerance from the obstacle is augmented and configurations are rechecked for collisions. If collisions are found within the lower part of the robot, the original path is modified with the new tolerance value to generate dynamic motion again.

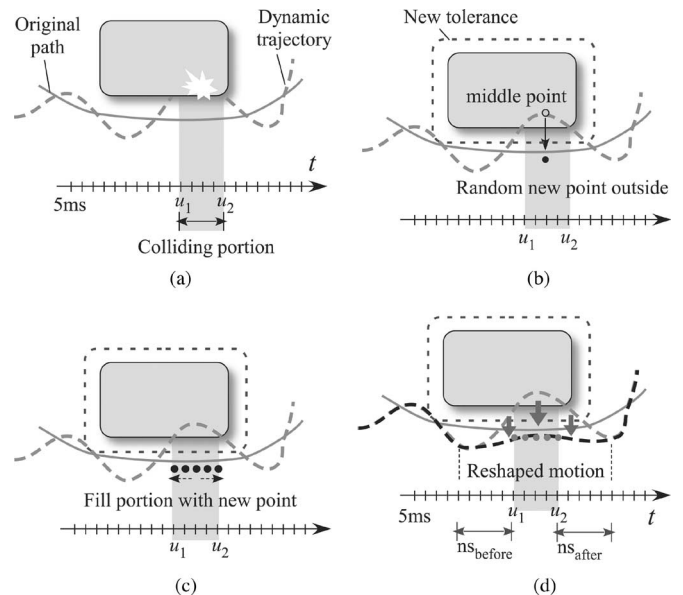


Fig. 11. (a) Trajectory is sampled at 5 ms intervals, and each sampled configuration is checked for collisions. (b) Collision-free configuration is found by augmenting the minimum tolerance value to the obstacle from the middle point of the portion. (c) Colliding configuration is replaced by the newly generated collision-free one. (d) New collision-free motion replaces the colliding one and is smoothly interpolated with the rest of the trajectory.

If collisions are found within the upper part of the body, the following reshaping procedure is applied, as illustrated in Fig. 11. This process includes two essential parts: smooth motion reshaping and IK calculation for the smoothed path.

1) *Smooth Motion Reshaping*: The output of a dynamic simulation of the planned trajectory is a sequence of robot's configurations uniformly sampled at 5 ms intervals, which is the control sampling rate of the robot. With this output, the colliding portions of his trajectory can be reshaped using motion-editing techniques that enforce spatial and temporal constraints usually employed in computer animation [19]. Our reshaping method is inspired on the motion-editing step described in [28].

After identifying the endpoints u_1 and u_2 of each colliding portion as in Fig. 11(a), a collision-free configuration is found with the augmented tolerance by randomly sampling in the task space (the object configuration) within a nearby range of the middle point of the portion, as shown in Fig. 11(b). The new collision-free configuration should also be within the reachable area of the arms. All the configurations within the portion are then replaced by this new collision-free configuration.¹

Even though collisions are eliminated at this stage, discontinuities in velocity and acceleration may occur at u_1 and u_2 . In order to smoothly anticipate the new configuration and regain the original trajectory, the reshaping portion is extended by adding the number of samples ns_{before} and ns_{after} . The anticipation and regaining times are computed to perform the motion

¹ Although practically one middle point suffices, if in rare cases the velocity of the object v exceeds a reference v_{ref} due to long colliding portion, middle points of half-portion is utilized to generate other collision-free configurations. This dichotomy method is applied until v becomes below v_{ref} .

between two configurations ($\mathbf{q}(u_1 - n_{\text{sbefore}})$, $\mathbf{q}(u_1)$) for anticipation, ($\mathbf{q}(u_1)$, $\mathbf{q}(u_1 + n_{\text{safter}})$) for regaining using the joint angular velocity $\dot{\mathbf{q}}_{\text{ref}}$ and v_{ref} of the object in workspace as reference [see Fig. 11(c)].

Finally, in order to obtain smooth motion for the end effector, configurations of the carried object are interpolated between $\mathbf{R}_o(u_1 - n_{\text{sbefore}})$ and $\mathbf{R}_o(u_2 + n_{\text{safter}})$. We use well-known interpolation techniques such as cubic spline. In order to respect time constraints, the time at which each of the specified configurations should be reached is specified along with position when fitting the curve [34]. Finally, the curve is resampled to get the new object's configuration at each 5 ms interval [see Fig. 11(d)].

2) *Solving Inverse Kinematics for Smooth Motion*: In our algorithm, an IK solver is used to satisfy the constraints of the hands at each sample of the reshaped trajectory that synchronizes the upper-body task with the lower-body motion. We work under the assumption that the displacement achieved in the 5 ms between samples is small enough to use (3) to relate the robot's posture variation $\dot{\mathbf{q}}_i$ to the change on the configuration \mathbf{r}_i of the end effector using Jacobian matrix \mathbf{J}_i corresponding to imposed task i

$$\dot{\mathbf{r}}_i = \mathbf{J}_i \dot{\mathbf{q}}_i. \quad (3)$$

As we are dealing with a mechanism that is redundant with respect to the number of imposed tasks, we use (4) to solve the IK problem for the task 1 (one hand grasping the bar):

$$\dot{\mathbf{q}}_1 = \mathbf{J}_1^\# \dot{\mathbf{r}}_1 + (\mathbf{I}_n - \mathbf{J}_1^\# \mathbf{J}_1) \mathbf{y}_1 \quad (4)$$

where $\mathbf{J}_1^\#$ is the pseudoinverse of the Jacobian matrix, \mathbf{I}_n is the n -dimensional identity matrix, and \mathbf{y}_1 is an arbitrary optimization vector.

Our IK solver considers two geometric task, which are the position and orientation constraints imposed on both hands on the carried object. In this case, the second task is achieved in the null space of the first task in such a way that the second task does not modify the first one by obtaining \mathbf{y}_1 that minimizes $\|\dot{\mathbf{r}}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1\|$ [35]:

$$\mathbf{y}_1 = \hat{\mathbf{J}}_2 (\dot{\mathbf{r}}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1) + (\mathbf{I}_n - \hat{\mathbf{J}}_2^\# \hat{\mathbf{J}}_2) \mathbf{y}_2$$

where $\hat{\mathbf{J}}_2 \equiv \mathbf{J}_2 (\mathbf{I}_n - \mathbf{J}_1^\# \mathbf{J}_1).$ (5)

Then, we obtain the following resulting motion:

$$\dot{\mathbf{q}}_2 = \dot{\mathbf{q}}_1 + \hat{\mathbf{J}}_2^\# (\dot{\mathbf{r}}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1) + (\mathbf{I}_n - \mathbf{J}_1^\# \mathbf{J}_1) (\mathbf{I}_n - \hat{\mathbf{J}}_2^\# \hat{\mathbf{J}}_2) \mathbf{y}_2 \quad (6)$$

where \mathbf{y}_2 is an arbitrary vector.

The reference velocity v_{ref} for the hand is taken into account as in [36]. Joint limits and priority levels can be treated as in the iterative algorithm proposed in [37].

C. Simulation and Experimental Results

We have conducted simulations and experiments of the proposed humanoid motion planner using simulator OpenHRP [38] and hardware humanoid platform HRP-2 [39]. HRP-2 has 30 DOF, is 1.54 m tall, and weighs 58 kg. This robot has two chest

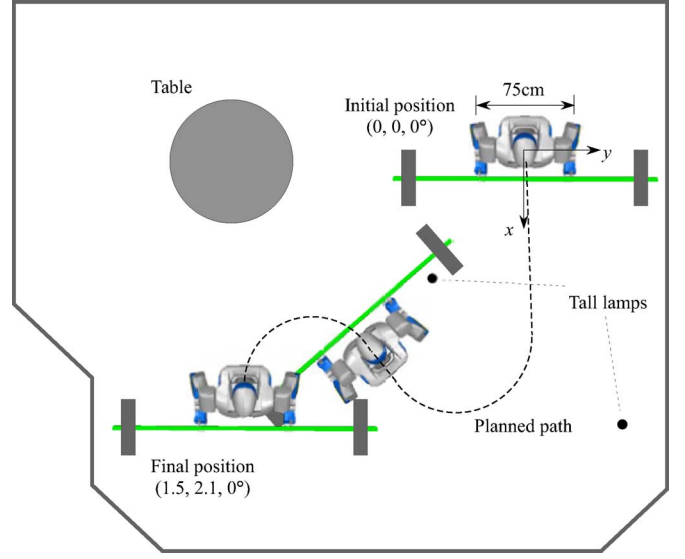


Fig. 12. Top view of the simulation and experiment environment with two poles and a table. The initial and final configurations of the robot are also shown.

joints for pitch and yaw rotation, which extends the motion capability to include lying down on the floor and standing up. It can carry load up to 2 kg at each hands.

For the following experiment, we took an example task of carrying a bar in an environment populated by obstacles. The length, diameter, and weight of the bar are 2.0 m, 24 mm, and 1.2 kg, respectively, with two disks whose diameter and thickness 0.4 m are 40 mm at 0.1 m from each extremity. The reference velocity v_{ref} in workspace is 0.3 m/s and 30°/s for translation and rotation, and reference joint angles $\dot{\mathbf{q}}_{\text{ref}}$ are defined from the hardware specification.

A 3-D collision-free whole-body motion of the humanoid robot is generated based on the proposed planning method in an environment of a flat plane with obstacles. Fig. 12 shows the top view of the environment with the initial and goal position and orientation (x, y, θ) on the plane, and planned walking path composed of line segments and arcs (dotted line) through the narrow passage. Two high poles are set with the distance of 1.75 m, which is shorter than the bar length 2 m. The robot holds the bar at the height of 0.95 m at the initial configuration. Since the robot cannot tilt the bar completely perpendicular to the ground, the planning and reshaping explore all the 6 DOF of the object in order to pass between the poles and underneath the table whose height is 1.1 m. The environment is therefore an example of cluttered one that imposes 3-D collision avoidance to verify the effectiveness of the proposed method.

Fig. 13 shows the planned results of 3-D collision-free dynamic motion. Fig. 13(a) and (b) shows the initial and goal configurations. Since the distance between the two lamps is shorter than the bar length, the bar should pass through with an angle. At the beginning of the motion, the computed trajectory for the bar makes the robot move to the left, then walk forward with a certain angle to path through the gap [see Fig. 13(c)–(e)]. The motion of the upper part of the robot is computed using

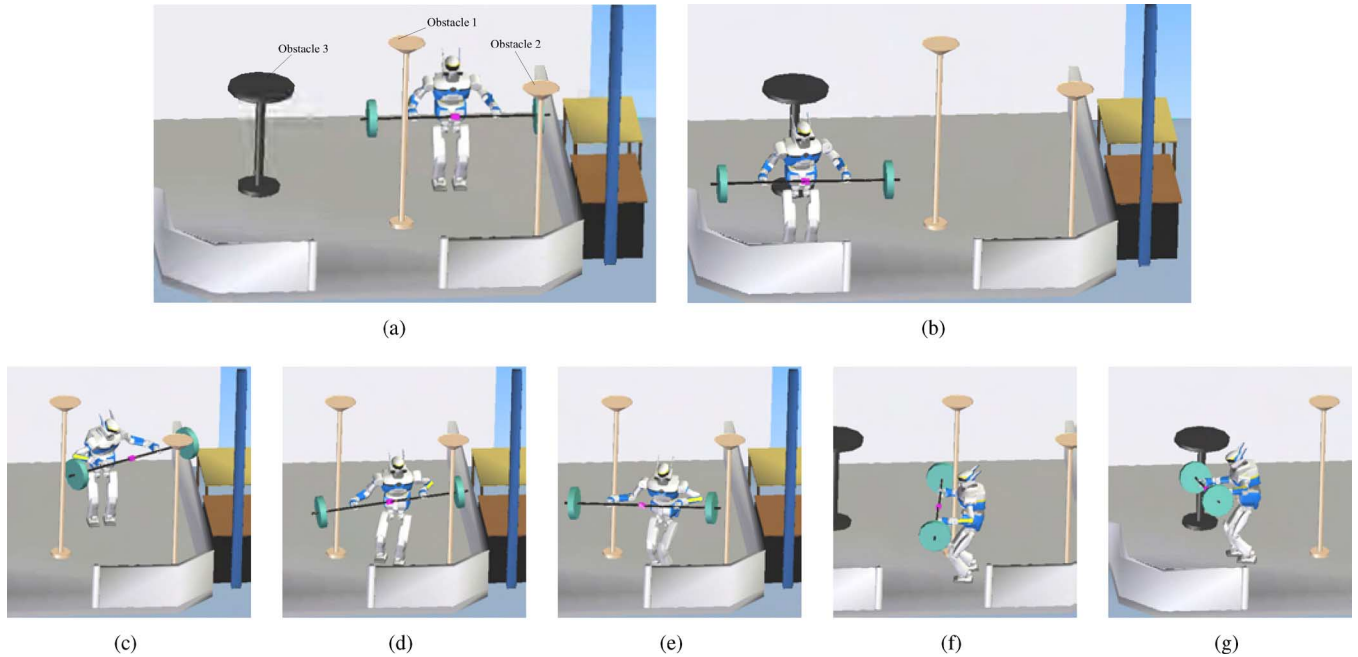


Fig. 13. Simulation of 3-D collision-free motion for bar-carrying task by humanoid robot HRP-2 from initial. (a) To final configuration (initial position). (b) Using whole-body motion (goal position). The robot rotates the bar horizontally to make the bar go through a gap between poles whose distance is shorter than that of the bar (c)–(e). By making use of the concave part of the carried object (f) for 3-D collision avoidance, it arrives at the goal configuration with another avoidance motion (g).

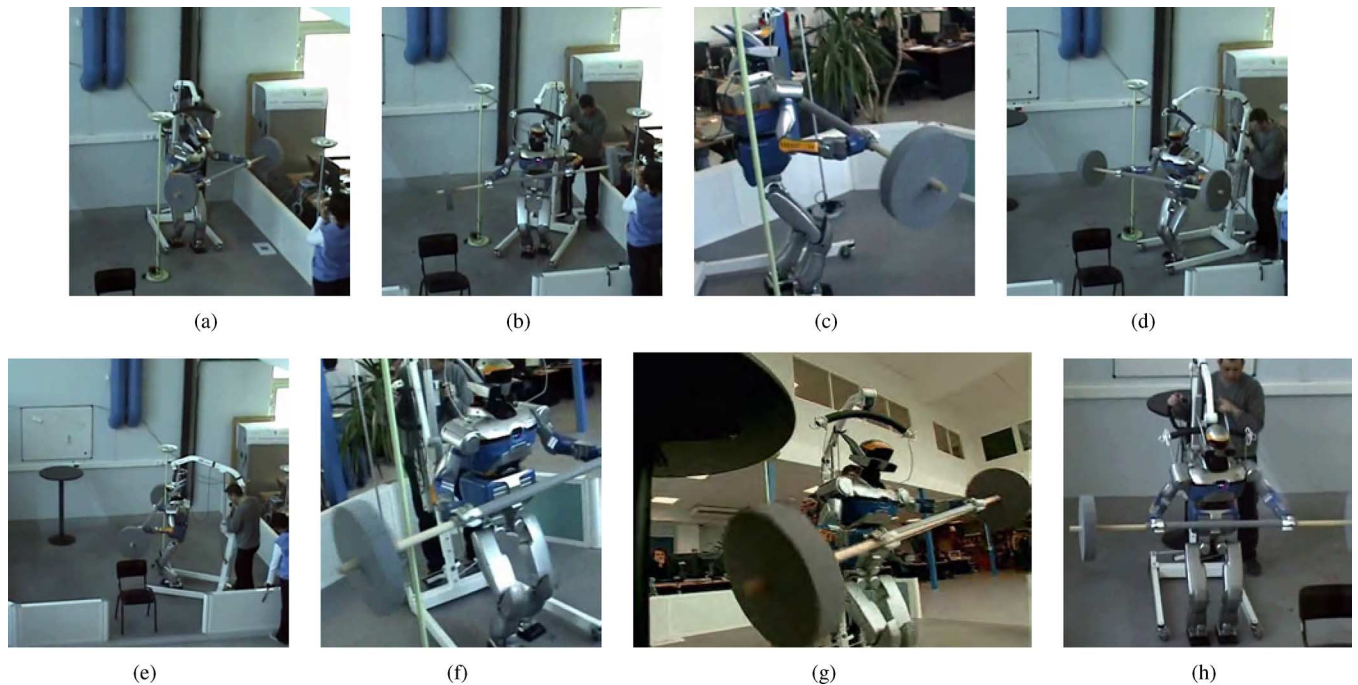


Fig. 14. Experiment of 3-D collision-free motion for bar-carrying task. The planned 3-D collision-free dynamic motion has been successfully performed by the humanoid robot.

a generalized IK algorithm based on (4) and (6). The position and orientation of each hand is dealt with as a single task. We can observe that the motions of two chest joints are involved in completing both tasks.

This example also shows that the complete geometry of the object is exploited in the collision-detection and path-

planning procedure and that no bounding box has been used [see Fig. 13(f)] where the concave part of the disk and the bar is close to the lamp. At the end of the motion [see Fig. 13(g)], the tall table is avoided. The complete trajectory execution time is around 28.8 s. The simulation has been done with KineoWorks and OpenHRP dynamic simulator. Fig. 14 shows

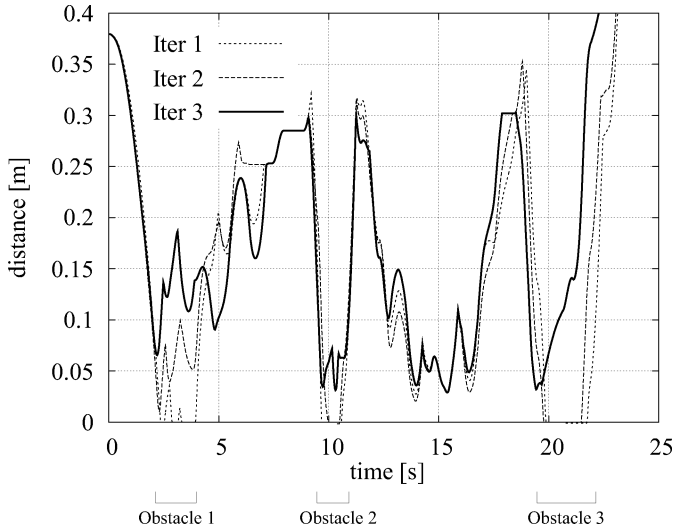


Fig. 15. Minimum distance between the robot with the carried object and the obstacles for three iteration with tolerances 2, 10, and 50 mm that correspond iterations 1, 2, and 3. Collisions (distance equaling 0) that occur with the first two iterations are eliminated at third iteration.

the real execution of the robotic platform HRP-2 #14 at the Laboratoire d'Architecture et d'Analyse des Systèmes, Centre National de la Recherche Scientifique (LAAS-CNRS).

The computation time of kinematic path in the first stage is 3.21 s. Then, dynamic motion generation for locomotion taking 28.8 s in the second stage takes 8 s. The time to detect and to identify collision portions to find a new configuration and to interpolate in workspace is 0.76 s. Finally, the computation of IK takes 145 s (5609 configurations) with precision of 1 mm.

The reshaping takes place three times to remove the collisions, with tolerances of 2, 10, 50 mm. With small tolerance values, collisions are detected since deviation from the planned trajectory occurs due to the dynamic motion. The tolerance was therefore increased twice to eliminate collisions.

Fig. 15 shows the minimum distances between the robot including carried object and the obstacles for those three iterations, where the distance 0 means that there are collisions. The transition of robot configurations through the three iterations can be well observed in Fig. 16. At the first iteration, there are collisions with all obstacles. The collision with obstacle 1 was resolved at the second iteration, but collisions with obstacle 2 still remains. Finally, all the collisions are removed at the third iteration.

Although the minimum distance sometimes becomes smaller than the desired tolerance 50 mm due to dynamic motion, a collision-free trajectory has been derived.

VII. DISCUSSIONS: LIMITATION AND FUTURE DEVELOPMENTS

In this section, several discussion points are raised that mainly concern humanoids.

A. Box Approximation in Path Planning

The bounding box in Fig. 10 is set so that it gives a good approximation of the swept volume during walking. Its depth is derived from the knee-bended walking posture and upper body,

and its width considers the lateral sway during walking without arms. The box is therefore a sufficiently accurate indicator to know if the robot can go through a given environment at the first stage. We can reduce the search space of the robot path into 3 DOF instead of whole robot joint space. Total DOFs for the path planner at the first stage depend on the constraints imposed on the carried object. For example, if a horizontal constraint is imposed on the object, its DOF becomes 4, which makes a total of 7 DOF with robot path.

Since the collisions at upper and lower body are separately resolved in the proposed spatial planning, there are limitations. The robot cannot make whole-body coordinated motions to avoid collisions, for example, swaying motion including lower body or stepping over obstacle in its way. This should be addressed through the close incorporation of dynamics into the global motion planner framework. Especially for humanoid applications, the next stage of the research is to develop a more efficient planner toward the unification of two stages of kinematic and dynamic motion planning. One of the possibilities is to include an efficient whole-body motion planner [40] based on a generalized IK framework with prioritized tasks including stepping. Another possibility is applying a reactive dynamic motion generation [23] based on task specification in operational space. In this case, a tool that integrates torque controller to global position-based motion planner need to be devised.

B. Low-Level Control and Perception Integration

So far, we have described an offline planning framework. Once the motion generator outputs a dynamic trajectory in the second stage, we assume that it can be executed accurately enough by a low-level controller of the robot. In the case of humanoid, this low-level controller is composed of a servo controller and a stabilizer mechanism. In general, the fidelity of the real motion with respect to the planned motion is quite high, up to 10 mm at the end effectors. We estimate this error as maximum 50 mm at the tip of the object, including oscillation of the arm. This value was used as the tolerance.

The main purpose of the experiment is to demonstrate the capacity of the planner to generate dynamic 3-D collision avoidance motion in a relatively small environment. However, the integration of sensor feedback in the planning becomes a critical issue to overcome the limitations of offline planner against the accumulated motion errors or unexpected obstacles during longer motions in real environments. One possible extension is the integration of online modeling of static or moving obstacles. The localization capacity should also be incorporated to be able to correct the accumulated positioning errors during the motion.

For real-time execution, the computation is fast enough in the case of the space manipulator. However, for the humanoid robot, reshaping, including IK and collision avoidance, takes a long computation time. By improving the implementation efficiency, we could increase the frequency of the loop between planning and perception. Efficient online walking pattern generators are now studied intensively for reactive modification of humanoid locomotion [41], [42]. They will help to extend the planner to

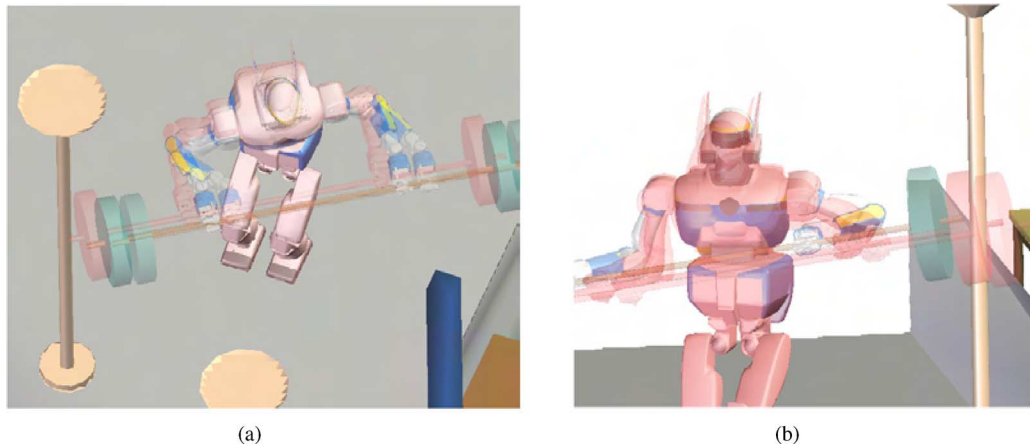


Fig. 16. Transition of robot configurations during the reshaping. The colliding part of the carried object goes away from the obstacle by increasing tolerance. (a) For obstacle 1, the collision is removed after the second iteration: only the leftmost configuration of the disk is colliding. (b) On the other hand, the collisions with obstacle 2 are resolved at the third iteration. The two configurations of the disk near the pole are in collision.

deal with challenging reactive planning problems with moving obstacles.

VIII. CONCLUSION

In this paper, we have presented a general planning framework for collision-free dynamic motion in 3-D using iterative reshaping. The proposed framework has two stages and benefits from advanced motion planning technology and powerful dynamic motion generators at each stage. The method allows us to plan motions for robots that have complex dynamics, such as space manipulators and humanoid robots, by applying appropriate reshaping methods. Temporal reshaping is utilized to make use of its convergence to the planned collision-free path in narrow passage. We employ spatial reshaping for the humanoid robot taking advantage of an efficient dynamic locomotion generator. As a result, the planned 3-D collision-free motion has successfully been verified by simulations and experiments in both cases.

Our approach still does not resolve the general motion-planning problem for dynamic systems. Nevertheless, it constitutes a general framework that can successfully apply in practice.

ACKNOWLEDGMENT

The authors would like to thank J. Himmelstein of LAAS-CNRS for his help.

REFERENCES

- [1] D. Hsu, J. C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *Int. J. Comput. Geometry Appl.*, vol. 4, pp. 495–512, 1999.
- [2] S. LaValle and J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, K. M. Lynch, D. Rus, and , Eds. Wellesley, MA: A K Peters, 2001, pp. 293–308.
- [3] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [4] J.-P. Laumond, "Kineo CAM: A success story of motion planning algorithms," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 90–93, Jun. 2006.
- [5] J. Kuffner, "Motion planning with dynamics," *Physique*, Mar. 1998.
- [6] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," *Auton. Robots*, vol. 1, no. 12, pp. 105–118, 2002.
- [7] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Control*, vol. AC-30, no. 6, pp. 531–541, Jun. 1985.
- [8] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, no. 5, pp. 1048–1066, 1993.
- [9] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [10] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, no. 5, pp. 1048–1066, 1993.
- [11] S. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [12] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA: MIT Press, 2006.
- [13] B. Baginski, "Efficient dynamic collision detection using expanded geometry models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1997, pp. 1714–1720.
- [14] E. Ferre and J.-P. Laumond, "An iterative diffusion algorithm for part disassembly," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 3149–3154.
- [15] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE Trans. Robot.*, vol. 20, no. 6, pp. 967–977, Dec. 2004.
- [16] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. 1993 IEEE Int. Conf. Robot. Autom.*, pp. 802–807.
- [17] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *Int. J. Robot. Res.*, vol. 21, no. 12, pp. 1031–1052, 2002.
- [18] M. Khatib, H. Jaouni, R. Chatila, and J.-P. Laumond, "Dynamic path modification for car-like nonholonomic robots," in *Proc. 1997 IEEE Int. Conf. Robot. Autom.*, pp. 2920–2925.
- [19] M. Gleicher, "Comparing constraint-based motion editing method," *Graph. Models*, vol. 63, pp. 107–134, 2001.
- [20] G. Hirzinger, B. Brunner, R. Lampariello, K. Landzettel, J. Schott, and B.-M. Steinmetz, "Advances in orbital robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 898–907.
- [21] M. Oda, "Experiences and lessons learned from the ETS-VII robot satellite," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 914–919.
- [22] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robot manipulators," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 3–17, 1985.
- [23] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2641–2648.
- [24] Y. Yang and O. Brock, "Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation in dynamic environments," in *Proc. Robot.: Sci. Syst.*, Philadelphia, PA, Aug. 2006.

- [25] T. Takubo, K. Inoue, K. Sakata, Y. Mae, and T. Arai, "Mobile manipulation of humanoid robots—Control method for CoM position with external force," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 1180–1185.
- [26] H. Harada, S. Kajita, H. Saito, M. Morisawa, F. Kanehiro, K. Fujiwara, K. Kaneko, and H. Hirukawa, "A humanoid robot carrying a heavy object," in *Proc. 2005 IEEE Int. Conf. Robot. Autom.*, 2005, pp. 1712–1717.
- [27] E. Yoshida, P. Blazevic, V. Hugel, K. Yokoi, and K. Harada, "Pivoting a large object: Whole-body manipulation by a humanoid robot," *J. Appl. Bionics Biomech.*, vol. 3, no. 3, pp. 227–235, 2006.
- [28] C. Esteves, G. Arechavaleta, J. Pettré, and J.-P. Laumond, "Animation planning for virtual characters cooperation," *ACM Trans. Graph.*, vol. 25, no. 2, pp. 319–339, 2006.
- [29] E. Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond, "Humanoid motion planning for dynamic tasks," in *Proc. 2005 IEEE-RAS Int. Conf. Humanoid Robots*, 2005, pp. 1–6.
- [30] E. Yoshida, C. Esteves, T. Sakaguchi, J.-P. Laumond, and K. Yokoi, "Smooth collision avoidance: Practical issues in dynamic humanoid," in *Proc. 2006 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 827–832.
- [31] I. Belousov, V. Kartashev, and D. Okhotsimsky, "Real time simulation of space robots on the virtual robotic test-bed," in *Proc. 7th Int. Conf. Adv. Robot.*, 1995, pp. 195–200.
- [32] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1620–1626.
- [33] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, pp. 497–516, 1957.
- [34] R. Parent, *Computer Animation: Algorithms and Techniques*. San Mateo, CA: Morgan Kaufmann, 2002.
- [35] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Boston, MA: Addison-Wesley Longman, 1991.
- [36] K. Yamane and Y. Nakamura, "Natural motion animation through constraining and deconstraining at will," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 3, pp. 352–360, Jul.–Sep. 2003.
- [37] P. Baerlocher and R. Boulic, "An inverse kinematics architecture enforcing and arbitrary number of strict priority levels," *Visual Comput.*, vol. 20, pp. 402–417, 2004.
- [38] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture humanoid robotics platform," *Int. J. Robot. Res.*, vol. 23, no. 2, pp. 155–165, 2004.
- [39] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "The humanoid robot HRP-2," in *Proc. 2004 IEEE Int. Conf. Robot. Autom.*, pp. 1083–1090.
- [40] E. Yoshida, O. Kanoun, J.-P. Laumond, and C. Esteves, "Task-driven support polygon reshaping for humanoids," in *Proc. 2006 IEEE-RAS Int. Conf. Humanoid Robots*, pp. 208–213.
- [41] K. Nishiwaki and S. Kagami, "High frequency walking pattern generation based on preview control of ZMP," in *Proc. IEEE Int. Conf. Robot. Autom.*, (ICRA 2003), 2006, pp. 2667–2672.
- [42] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *Proc. IEEE Int. Conf. Robot. Autom.*, (ICRA 2003), 2007, pp. 3989–3994.



Eiichi Yoshida (S'94–M'96) received the M.E. and Ph.D. degrees on precision machinery engineering from the Graduate School of Engineering, University of Tokyo, Tokyo, Japan, in 1993 and 1996, respectively.

In 1996, he joined former Mechanical Engineering Laboratory, Tsukuba, Japan. From 1990 to 1991, he was a Visiting Research Associate at the Swiss Federal Institute of Technology at Lausanne (EPFL). He is currently the Co-Director of AIST/IS–CNRS/ST2I Joint French–Japanese Robotics Laboratory (JRL),

Laboratoire d'Architecture et d'Analyse des Systèmes, Centre National de la Recherche Scientifique (LAAS–CNRS), Toulouse, France. He is also a Senior Research Scientist in the Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba. His current research interests include robot task and motion planning, modular robotic systems, and humanoid robots.



Claudia Esteves received the Ph.D. degree in computer science from the National Polytechnic Institute of Toulouse, Toulouse, France, in 2007.

Her Ph.D. research was conducted at the Laboratoire d'Architecture et d'Analyse des Systèmes, Centre National de la Recherche Scientifique (LAAS–CNRS) on motion planning for virtual characters and humanoid robots. She is currently an Associate Professor in the Faculty of Mathematics, University of Guanajuato, Guanajuato, Mexico. Her current research interests include motion planning for human-like mechanisms and motion planning with constraints.



Igor Belousov received the Graduate degree from the Department of Mechanics and Mathematics, the Ph.D. degree in theoretical mechanics in 1993, both from the Lomonosov Moscow State University (MSU), Moscow, Russia, and the Doctor of Phys.-Math. Sciences degree in theoretical mechanics from Keldysh Institute of Applied Mathematics, Russian Academy of Sciences (KIAM-RAS), Moscow, in 2003.

During 1991–2005, he was a Leading Researcher at the KIAM-RAS. During 2002–2005, he was a Leading Researcher (part-time) at the MSU. Since 2005, he has been with Hewlett-Packard Laboratories (HP Labs), Moscow, where he is currently the Regional Lead for Russia in Open Innovation Office. He has been an Invited Researcher at the Robotics and Computer Graphics Centers, Laboratoire d'Architecture et d'Analyse des Systèmes, Centre National de la Recherche Scientifique (LAAS–CNRS), France, De Montfort University, U.K., and Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), CNRS, France. He received acknowledged results in real-time dynamics simulation of robot manipulators, network robotics, and robot teleoperation. His current research interests include space robotics, man–machine interfaces, remote education, and Web technologies.

Dr. Belousov was awarded by the personal research grants from Russian Academy of Sciences and INTAS (EU). His research on network robotics has been chosen for Annual Report of the President of the Russian Academy of Sciences (2002) as best work on mechanics.



Jean-Paul Laumond (M'95–SM'04–F'07) received the M.S. degree in mathematics, the Ph.D. degree in robotics, and the Habilitation degree in robotics, in 1976, 1984, and 1989, respectively, all from the University Paul Sabatier, Toulouse, France.

In Fall 1990, he had been an Invited Senior Scientist at Stanford University, Stanford, CA. From 1991 to 1995, he was a member of the French Comité National de la Recherche Scientifique. He has been a Co-ordinator for two European Esprit projects, PROMotion during 1992–1995 and MOLOG during 1999–

2002, both dedicated to robot motion planning technology. During 2001–2002, he created and managed Kineo CAM, a spin-off company from the Laboratoire d'Architecture et d'Analyse des Systèmes, Centre National de la Recherche Scientifique (LAAS–CNRS), to develop and market motion-planning technology. He is currently a Directeur de Recherche at the LAAS–CNRS, Toulouse, France, where he is also associated with the AIST/IS–CNRS/ST2I Joint French–Japanese Robotics Laboratory (JRL) as its Co-Director. He has been teaching robotics at the Ecole Nationale Supérieure de Techniques Avancées (ENSTA), Paris, France, and Ecole Normale Supérieure, Paris. He is the author or coauthor of more than 100 papers published in international journals and conferences in computer science, automatic control, and robotics, and is the Editor of three books. His current research interests include human motion studies along three perspectives: artificial motion for humanoid robots, virtual motion for digital actors and mannequins, and natural motions of human beings.

Dr. Laumond is a member of the IEEE Robotics and Automation Society (RAS) AdCom. He is currently an IEEE Distinguished Lecturer.



Takeshi Sakaguchi received the B.Sc. degree in control engineering, the M.Sc. degree in design and control of the D.D. robot, and the Ph.D. degree in dynamic manipulation dealing with moving objects, in 1987, 1989, and 1993, respectively, all from Osaka University, Osaka, Japan.

He is currently a Senior Researcher in the AIST/IS-CNRS/ST2I Joint French-Japanese Robotics Laboratory (JRL), National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His current research interests include ubiquitous robotics, ambient intelligence, and interactive human-space design and intelligence.



Kazuhito Yokoi (M'92) received the B.E. degree in mechanical engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1984, and the M.E. and Ph.D. degrees in mechanical engineering science from Tokyo Institute of Technology, Tokyo, Japan, in 1986 and 1994, respectively.

In 1986, he joined the Mechanical Engineering Laboratory, Ministry of International Trade and Industry. From November 1994 to October 1995, he was a Visiting Scholar at Robotics Laboratory, Computer Science Department, Stanford University. He is currently the Scientific Group Leader of the Autonomous Behavior Control Research Group and the Co-Director of the AIST/IS-CNRS/ST2I Joint French-Japanese Robotics Laboratory (JRL), Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He is also an Adjunctive Professor of Cooperative Graduate School, University of Tsukuba, Tsukuba. His current research interests include humanoids, human-centered robotics, and intelligent robot systems.

Dr. Yokoi is a member of the IEEE Robotics and Automation Society.