

# Motion Planning of Human-Like Robots using Constrained Coordination

Liangjun Zhang   Jia Pan   Dinesh Manocha

{zlj,panj,dm}@cs.unc.edu

*Dept. of Computer Science, University of North Carolina at Chapel Hill*

*videos at <http://gamma.cs.unc.edu/DHM>*

**Abstract** We present a whole-body motion planning algorithm for human-like robots. The planning problem is decomposed into a sequence of low-dimensional sub-problems. Our formulation is based on the fact that a human-like model is a tightly coupled system and uses a constrained coordination scheme to solve the sub-problems in an incremental manner. We also present a local path refinement algorithm to compute collision-free paths in tight spaces and satisfy the statically stable constraint on CoM. We demonstrate the performance of our algorithm on an articulated human-like model and generate efficient motion strategies for walking, sitting and grabbing objects in complex CAD models.

## 1 Introduction

The problem of modeling and simulating human-like motion arises in different applications, including humanoid robotics, biomechanics, digital human modeling for virtual prototyping, and character animations. One of the main goals in this area is to develop efficient motion strategies for whole-body planning for various tasks including navigation, sitting, walking, running, object manipulation, etc. The entire human body consists of over 600 muscles and over 200 bones, and half of those are found in the hand and feet. Even the simplest human-like models represent the skeleton as an articulated model with 34 – 40 joints to model the different motions. The underlying complexity makes it hard for a planner to efficiently compute the motion due to the dimension of the configuration motion. In addition to collision-free constraints, the resulting planner also needs to satisfy the posture and dynamics constraints to generate realistic motions.

Recent research in robotics has focused on motion planning of humanoids due to the commercial availability of humanoid robot hardware [11, 20]. Many of these approaches use a simple bounding volume (e.g. a cylinder) approximation of the entire human model [19] or the lower body [1, 24] to compute the collision-free motions, and design appropriate gaits or locomotion controllers to follow those trajectories [14, 16]. Other approaches compute the motion for the whole body [13]. Most of these approaches are efficient for open environments, but their performances may degrade for cluttered environments. Besides humanoids, another driving application of human-like robots is digital modeling of humans or mannequins for design, assembly and maintenance in CAD/CAM and virtual prototyping. The digital human models can be inserted into a simulation or virtual environment to facilitate the prediction of performance, safety and ergonomic analysis of the CAD models [6, 21].

For example, human-line models are used in validating vehicle or aircraft designs and ensure that there will be sufficient clearance in the CAD model for a human operator to remove a complex part. In order to perform these tasks, we need to develop capabilities for complex motion strategies (e.g. sitting, bending), handling narrow passages, and planning in cluttered environments.

One strategy to solve high DOF planning problems is to decompose a problem into a set of lower dimensional sub-problems [1, 2, 15]. A human-like robot can be decomposed, for instance, into the lower body and the upper body. In order to deal with CAD/CAM applications, we need to handle cluttered environments and model many other motions, which cannot be efficiently generated by simple decompositions. In addition to collision-free constraints, the motion of human-like robots subjects to statically or dynamically stable constraints. There is a general perception that actual human motion results from simultaneously performing multiple objectives in a hierarchical manner, and researchers have developed similar models for dynamics control [27]. It would also be useful to develop approaches that use hierarchical decompositions for planning human-like motions.

**Main Results:** We present a whole-body motion planning approach for human-like robots by coordinating the motions of different body parts. Our approach uses a hierarchical decomposition and takes into account that a human body is a tightly coupled system. We describe a new constrained coordination scheme that incrementally computes the motion for different parts and satisfies both collision-free and statically stable constraints. In order to deal with cluttered or tight scenarios, we present a local path refinement algorithm which can take into account the workspace distance information to control the amount of modification on the path. The whole-body planning algorithm is decomposed into a sequence of low-DOF planning sub-problems, and we use constrained sampling and local interpolation techniques to compute the paths for each sub-problem. If any sample generated by the constrained coordination algorithm is not statically stable, we further modify the sample by using inverse kinematics (IK) so that the CoM at the new sample lies inside the approximate foot support polygon. We demonstrate the performance of our algorithm on an articulated human-like model. We generate various motion strategies corresponding to bending, standing-up, walking, sitting, and grabbing objects in different complex scenarios. In practice, our planner is able to compute a collision-free and statically stable motion in tens of seconds. Within the two-stage framework for planning dynamic motions, our approach can improve the efficiency of the stage of computing collision-free and statically stable motions.

The rest of the paper is organized in the following manner. We give a brief survey of related work in Section 2. We describe our hierarchical representation and give an overview of our approach in Section 3. Section 4 presents the constrained coordination algorithm as well as local path refinement. We describe our implementation in Section 5 and highlight its performance.

## 2 Previous Work

There is an extensive literature on motion planning, motion coordination and dynamic control of human-like robots. In this section, we give a brief overview of

prior work on motion planning for human-like robots, dimensionality reduction and path replanning.

### ***2.1 Motion Planning for Human-Like Robots***

Sampling-based approaches have been successfully applied to human-like robots to plan various tasks. These include efficient planning algorithms for reaching and manipulation that combine motion planning and inverse kinematics [7, 8] or computing the whole body motion [13]. The motion strategies for human-like robots such as walking can also be computed by walking pattern generators [14, 18, 16]. To plan collision-free and dynamically stable motions, most previous approaches use a decoupled two-stage framework [12, 20, 33]. Task-based controllers have also been presented to plan and control the whole-body motion [11, 27]. In the domain of computer animation, motion capture data are often used to synthesize natural human motion [30].

### ***2.2 Dimensionality Reduction***

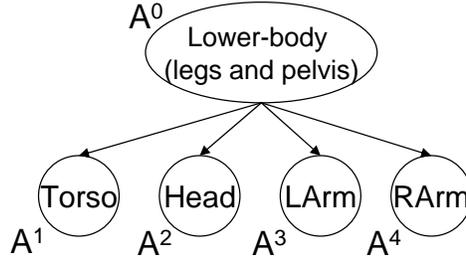
Decomposition techniques can reduce the overall dimensionality of motion planning problems and have been applied to articulated robots or multi-robot systems [2, 15]. Different coordination schemes for combining the solutions of lower dimensional sub-problems are presented in [9, 23, 26]. Simple decomposition schemes based on lower-body and upper-body can be used for planing the motion of human-like robots [1]. Another effective scheme for dimensionality reduction is to use the reduced kinematic models, such as using a bounding cylinder to approximate the lower body [19, 24]. A multi-level method to adjust the activated DOF according to the environment is presented in [32]. Finally, PCA-based analysis or various task constraints can also be used to guide the sampling towards the lower dimensional space [5, 28].

### ***2.3 Path Modification and Replanning***

The step of path modification is often required by many motion planning approaches. Retraction-based sampling approaches can effectively deal with narrow passages and cluttered environments [4, 34]. By performing randomly perturbation or penetration depth computation, a path with colliding configurations can be repaired. For motion planning among dynamic obstacles, local path modification algorithms modify the portion of the path to avoid the moving obstacle or to accommodate changes in the connectivity [25, 31]. Since global modification needs to replan for the entire connectivity data structure, they are usually much more expensive [10, 17].

## **3 Overview**

In this section, we introduce our notation and give an overview of our planning algorithm. Planning a path for a human-like robot by taking into account all the DOF is often difficult due to the underlying high dimensional search space. Our approach represents a human-like robot by using a set of body parts, i.e.  $\{A^0, A^1, \dots, A^n\}$ . We decompose the problem into multiple sub-problems of lower dimensions, and compute the motion for the body parts in a sequential order. A key feature of our algorithm is that planning the path of the  $k^{th}$  body part is coordinated with the paths



**Fig. 1** Our decomposition scheme for a human-like model. In this decomposition, we show a 2-level tree hierarchy. Our approach also extends to multiple level hierarchies. We compute the motion for the body parts sequentially by starting from the root of the hierarchy.

of the first  $k - 1$  body parts computed earlier. Furthermore, all these paths can be refined using a local refinement scheme. In this manner, the paths for the first  $k - 1$  body parts can possibly be updated during the planning of the  $k^{\text{th}}$  body part. This form of sequential planning along with path refinement helps us treat the whole-body as a tightly-coupled system.

### 3.1 Decomposition of A Human-like Model

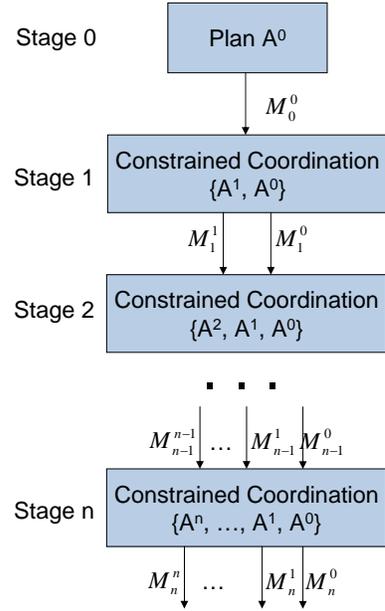
The simplest decomposition of a human-like robot decomposes the whole-body into different body parts  $\{A^0, A^1, \dots, A^n\}$ . In this case, it is assumed that each  $A^i$  has few DOF. Fig. 1 shows a decomposition scheme, where a human model is decomposed into parts: a lower-body (including legs and pelvis), torso, head, left arm and right arm. In this decomposition, the lower body is treated as the root of the tree. It is possible to compute another decomposition where the root node ( $A^0$ ) corresponds to the torso. Furthermore, we build a hierarchical representation based on the inter-connection between the parts. For example, each arm can be further decomposed into upper arm, lower arm, hand, etc.

We use the symbol  $\mathbf{q}$  to denote the configuration of a human-like robot.  $\mathbf{q}$  is composed of configurations  $\mathbf{q}^i$  for each body part, i.e.  $\mathbf{q} = \{\mathbf{q}^0, \mathbf{q}^1, \dots, \mathbf{q}^n\}$ , where  $\mathbf{q}^i$  corresponds to the configuration of  $A^i$ . Since we are dealing with articulated models, the configuration  $\mathbf{q}^i$  for  $A^i$  is determined by all of its actuated joints, including the joint through which  $A^i$  is connected to its parent body part in the decomposition tree. For the lower body part  $A_0$ , 6 additional unactuated DOF can be added to the system to specify the position and orientation of the coordinate frame associated with the pelvis. For instance, The basic motion planning problem for a human-like robot is to find a collision-free path between the starting configuration  $\mathbf{q}_s = (\mathbf{q}_s^0, \mathbf{q}_s^1, \dots, \mathbf{q}_s^n)$  to the goal configuration  $\mathbf{q}_g = (\mathbf{q}_g^0, \mathbf{q}_g^1, \dots, \mathbf{q}_g^n)$ . In practice, the resulting motion should also satisfy with statically or dynamically stable constraints.

### 3.2 Whole-Body Motion Planning using Constrained Coordination

A human-like model is a tight-coupled system and the inter-connection between the body parts needs to be maintained during planning. One possibility is to decompose this high dimensional robot into a multi-robot planning problem by treating each part as a separate robot. There is rich literature on multi-robot motion plan-

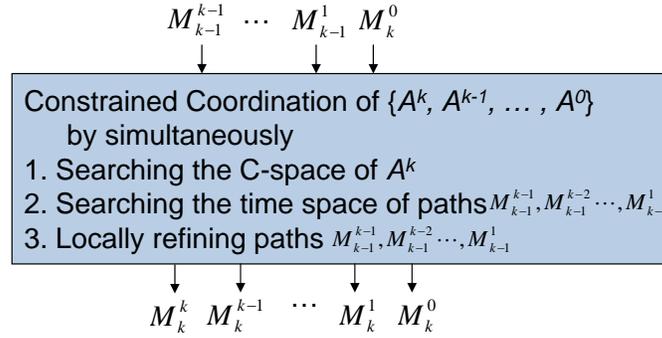
**Fig. 2** Whole-Body Motion Planning using Constrained Coordination. We first compute the path for  $A^0$  while ignoring all the other parts. When planning the motion for the system  $\{A^0, A^1\}$ , the motion of  $A^0$  is constrained on the path  $M_0^0(t)$  but this path can be locally refined. We refer this step as constrained coordination (Fig. 3).



ning and at a broad level prior approaches for multi-robot planning can be classified into centralized or decentralized methods. The centralized methods compose all the different robots into one large coupled system. The DOF of the coupled system corresponds to the sum of DOF of all the robots. Such an approach could be extremely inefficient for a human-like robot due to the high DOF configuration space. The decentralized planners compromise on the completeness by using a decoupled approach. The decentralized planner typically proceeds in two phases. In the first phase, a collision-free path is computed for each robot with respect to the obstacles and the collisions between the robots are handled in the second phase by adjusting their velocities. Since a human-like robot is a tightly coupled system, it would be hard using purely decoupled methods to maintain the inter-connection constraints between adjacent links of the robot.

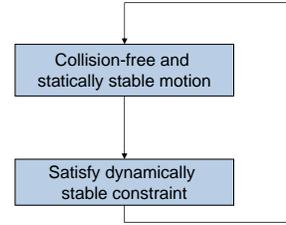
We propose a hybrid coordination scheme that is based on prior work on prioritized or incremental coordination approaches [9, 26]. Our algorithm proceeds hierarchically using the decomposition of the human model and computes the path of different nodes in the tree in a breadth first manner. The path computed for a part corresponding to a node, also takes into the account the path of its parent node and other paths computed so far.

We describe the main idea behind constrained coordination by taking into account two objects,  $A$  and  $B$ . Lets say  $A$  has  $m$  DOF and  $B$  has  $n$  DOF. By considering the two objects as a composite system,  $\{A, B\}$ , a centralized planner needs to search over a  $m + n$  dimensional space. On the other hand, decentralized approaches plan each object independently by searching the  $m$  and  $n$  dimensional spaces corresponding to each robot. We improve the decentralized planning by using an incremental coordination strategy. A collision-free path  $M^A(t)$  for  $A$  is computed by ignoring  $B$ .



**Fig. 3** Constrained coordination algorithm: It includes a path computation stage for the new part and refinement of the paths of the other parts.

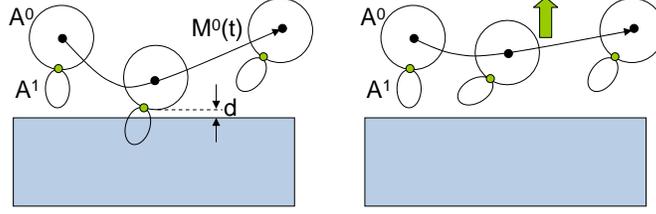
**Fig. 4** Within the decoupled two-stage framework for planning a motion satisfying with both collision-free and dynamically stable constraints, our approach can improve the efficiency of the first stage of collision-free and statically stable motion computation.



Next, a collision-free path for the system  $\{A, B\}$  is computed by coordinating  $A$  and  $B$ . During the coordination, a *path constraint* for  $A$  is imposed so that the configuration of  $A$  should lie on the path  $M^A(t)$ . The coordination of the system  $\{A, B\}$  is the  $n + 1$  dimensional search space, since  $A$  is constrained on a one dimensional path with the parameter  $t$  and  $A^1$  has  $n$  DOF. Intuitively, this approach computes a path for  $B$  (i.e.  $M^B(t)$ ), based on the original trajectory ( $M^A(t)$ ) computed for  $A$ . However, it is possible that the original path computed for  $A$ , may not result in a feasible path for  $B$  such that  $\{A, B\}$  may satisfy all the collision and dynamics constraints as shown in Fig. 5. In case of human-like motion, such a hard constraint can result in either an inefficient planner or a failure to compute a solution that satisfies all the constraints. In order to address this issue, we use a local refinement scheme that modifies the computed trajectory  $M^A(t)$ , as it computes a collision-free path for  $B$ . In Section IV, we present an implicit local path refinement algorithm based on constrained sampling and interpolations.

### 3.3 Planning Stable Motions

In addition to collision-free and joint limit constraints, the motion of human-like robots subjects to statically or dynamically stable constraints. The computed postures should either be statically stable, i.e. the projection of the center of mass of the robot (CoM) lies inside the foot support polygon, or dynamically stable, i.e. the zero moment point (ZMP) lies inside the support polygon [29]. However, due to the computational complexity to plan the collision-free and dynamic motion together, most previous approaches tend to use a decoupled two-stage framework [12, 20, 33]. For



**Fig. 5** Given an articulated robot with two links  $A^0$  and  $A^1$ , a path  $M^0(t)$  for  $A^0$  is first computed. However, when  $A^0$  moves along  $M^0(t)$ , it comes very close to the obstacle (shown in blue), as the separation distance  $d$  is very small. This leads to no feasible placement for  $A^1$ , as it collides. To resolve such cases, our constrained coordination scheme locally refines the path  $M^0(t)$  by moving it upwards (shown with green arrow), while planning the motion for  $A^1$ . In practice, such a local refinement approach is more efficient as compared to global replanning.

instance, a collision-free path can be first computed. The path then is transformed into a dynamically stable trajectory. Each of these stages is iterated until both types of constraints are satisfied (Fig. 4). Our approach can be extended to compute a statically stable motion. If any sample generated by the constrained coordination algorithm is not statically stable, we further modify the sample by using inverse kinematics (IK) so that the CoM at the new sample lies inside the approximate foot support polygon. Within the two-stage framework, our approach can improve the efficiency of the first stage on computing a collision-free and statically stable path. Such path is further processed by the second stage.

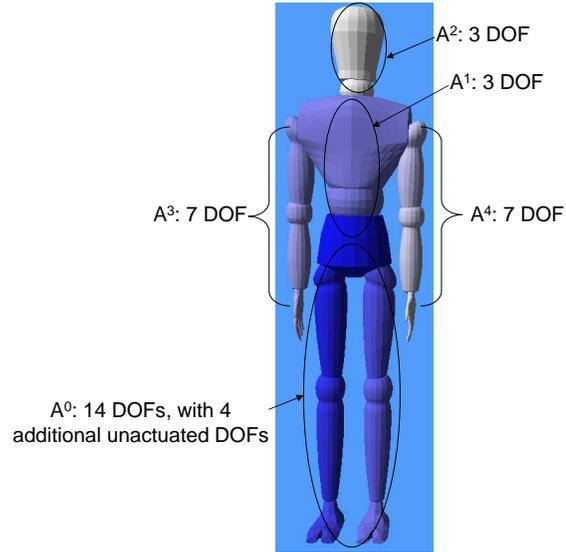
## 4 Constrained Coordination

In this section, we present our constrained coordination approach. It is primarily designed for human-like or tightly coupled robots that have high DOF. Our approach consists of two parts: a modified incremental coordination algorithm and local path refinement.

### 4.1 Path Computation

Our algorithm proceeds in multiple stages, as shown in Fig. 2. We use the symbol  $M_j^i(t)$  to denote the path of part  $A^i$  computed after stage  $j$ . After stage  $j$ , the algorithm has computed the following paths:  $M_j^i(t)$  for  $A^i$ , for  $i = 0, 1, \dots, j$ . As shown in Fig. 3, during this stage, the algorithm computes paths for  $A^0, A^1, \dots, A^j$  by simultaneously searching the C-space of  $A^j$  and the 1 dimensional time space of the set of paths  $(M_{j-1}^0(t), M_{j-1}^1(t), \dots, M_{j-1}^{j-1}(t))$ , and locally refining each of the paths  $M_{j-1}^0(t), M_{j-1}^1(t), \dots, M_{j-1}^{j-1}(t)$ . Later, we show the local path refinement can be performed implicitly within a sample-based planner. The algorithm traverses the entire hierarchy of body parts,  $\{A^0, \dots, A^n\}$ , sequentially in the breadth-first order of the tree. After stage  $n$ , the algorithm has computed a path for all the parts that satisfy the constraints.

**Fig. 6** A human-like robot with 40 DOF used in our simulations. We also highlight the DOF of each part according to the decomposition shown in Fig. 1.



## 4.2 Implicit Local Path Refinement

A key aspect of constrained coordination algorithm is refining the path that was computed at the previous stage. In this section, we present a local replanning algorithm that takes into account the decomposition of human-like robot and the path computation algorithm highlighted above. We observe that within an incremental coordination scheme for two objects  $\{A^0, A^1\}$ , the motion of  $A^0$  is strictly constrained on the path computed earlier. This can lead to the difficulty of planning a motion for the overall robot, or it fails in terms of finding a solution. Fig. 5 shows such an example for an articulated robot with two links  $A^0$  and  $A^1$ . When  $A^0$  moves along the path  $M_0^0(t)$ , its distance  $d$  to the obstacle becomes too small, which results in no feasible placement for  $A^1$ . This issue can arise when we are attempting to compute a collision free path in a cluttered environment or in a narrow passage. Since the robot is decomposed into many body parts, each body part is constrained by predecessors, as given by the breath first order of the tree. In this case, we refine the path for  $A^0$ , given as  $M_0^0(t)$ , and compute a new path  $M_1^0(t)$ .

Our algorithm uses a sample-based planner to compute a path during each stage and we design an implicit local refinement scheme that can be integrated with any sample-based approach. The two main steps of sample-based planning is generating samples in the free space and computing an interpolating motion between those samples using local planning. Instead of explicitly modifying the path computed in the previous stage, our algorithm performs *constrained sampling* and *constrained interpolation* so that the generated samples or local motions are allowed to move away from the constraining path up to a threshold. In this way, we achieve the path refinement implicitly. In the following, we present the algorithm for a composite system with two robots, which can be generalized to a system with  $n$  robots.

### 4.2.1 Constrained Sampling

Our algorithm (Alg. 1) generates a configuration for the system  $\{A^0, A^1\}$  subject to the path constraint. A free configuration for  $A^1$  is computed by randomly sampling its configuration space and performing collision checking with the obstacles. A configuration of  $A^0$  for the system is computed by randomly generating a value  $t_{rand}$  on the path  $M^0(t)$ , which lies in the coordination space  $[0, 1]$ . As part of the refinement step, we perturb these free-space configurations such as  $M^0(t_{rand})$ .

We determine the closest points between  $A^0$  at the configuration  $M^0(t_{rand})$  and the obstacles. Let  $\mathbf{p}$  denote the closest point on  $A^0$  using  $\mathbf{p}$  and let  $\mathbf{r}$  be the vector from the closest point of the obstacles to  $\mathbf{p}$ . The basic idea for perturbing the configuration  $M^0(t_{rand})$  for  $A^0$  is to increase the distance between  $A^0$  and the obstacles so that we can avoid the situations that are shown in Fig. 7. In order to perform such a perturbation, we randomly choose a scale factor  $\lambda$  between an interval of 1 and the maximum scale factor  $\varepsilon/||\mathbf{r}||$ , where  $\varepsilon$  is a user-control input. Furthermore, a Gaussian distribution function can be used when randomly choosing  $\lambda$  within the interval computed earlier so that the probability of choosing a value near to 1 is higher. Finally, we compute the amount of perturbation  $\delta\mathbf{q}^0$  for  $A^0$  by solving the following equation by using an inverse kinematic solver:

$$\lambda \mathbf{r} = \mathbf{J}_p^0 \delta\mathbf{q}^0, \quad (1)$$

where  $J_p^0$  is the Jacobian for the point  $p$  on  $A^0$ .

### 4.2.2 Constrained Interpolation

We address the issue of motion interpolation during our refinement algorithm. Given two configurations of the system  $\{A^0, A^1\}$ , our goal is to interpolate a motion between them that satisfies the path constraint. The interpolation between the two configurations of the body part  $A^1$  can be computed by linear or other interpolation algorithm. Differently, when interpolating the two configurations of  $A^0$ , the resulting motion of  $A^0$  should be constrained on the path  $M^0(t)$  computed earlier. Let us denote  $t_0$  and  $t_1$  as the parameters of the two configurations of  $A^0$  on the path  $M^0(t)$ . In order to perform the constrained interpolation for  $A_0$ , we first determine all the nodes (i.e. samples) along the path  $M^0(t)$  between  $t_0$  and  $t_1$  as shown in Fig. 7. Next, we locally perturb these nodes by using the perturbation describe above. This results in a new interpolating motion for  $A^0$ . Together with the interpolating motion for  $A^1$ , we finally obtain a constrained interpolating motion for the entire system, which can be used by a sampling-based planner.

### 4.3 Statically Stable Motion

Our constrained coordination algorithm can be extended to generate a statically stable motion for the robot. In the coordination algorithm, we modify the last stage for coordinating between  $A_n$  and  $\{A_0, A_1, \dots, A_{n-1}\}$ . At this stage, we additionally check whether the configuration  $\mathbf{q}$  generated from constrained sampling is statically stable, i.e. the projection of the center mass (CoM) point of the robot at  $\mathbf{q}$  lies inside the support polygon defined by the robot's support feet (foot for single foot support

**Algorithm 1:** Constrained Sampling

---

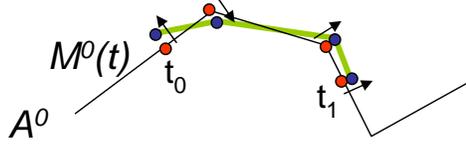
```

Input: Body parts  $A^0$  and  $A^1$ ;
        A collision-free path  $M^0(t), t \in [0, 1]$  for  $A^0$ 
Output: A random configuration  $\{\mathbf{q}^0, \mathbf{q}^1\}$  for  $\{A^0, A^1\}$  where  $\mathbf{q}^0$  subjects to the path
        constraint  $M^0$ 

begin
   $\mathbf{q}^1 =$  Random configuration of  $A^1$ 
  // Sampling the path  $M^0$ 
   $t_{rand} = Rand(0, 1)$ 
   $\mathbf{q}^0 = M^1(t_{rand})$ 
  // Perturbation
   $\mathbf{r} =$  Shortest vector between points from any obstacle to  $A^0$ 
   $\lambda =$  A random scale factor (See Section IV.B)
   $\Delta \mathbf{r} = \lambda \mathbf{r}$ 
   $\Delta \mathbf{q}^0 =$  InverseKinematics( $A^0, \mathbf{q}^0, \Delta \mathbf{r}$ ) // Eq. 1
   $\tilde{\mathbf{q}}^0 = \mathbf{q}^0 + \Delta \mathbf{q}^0$ 
  return  $\{\tilde{\mathbf{q}}^0, \mathbf{q}^1\}$ 
end

```

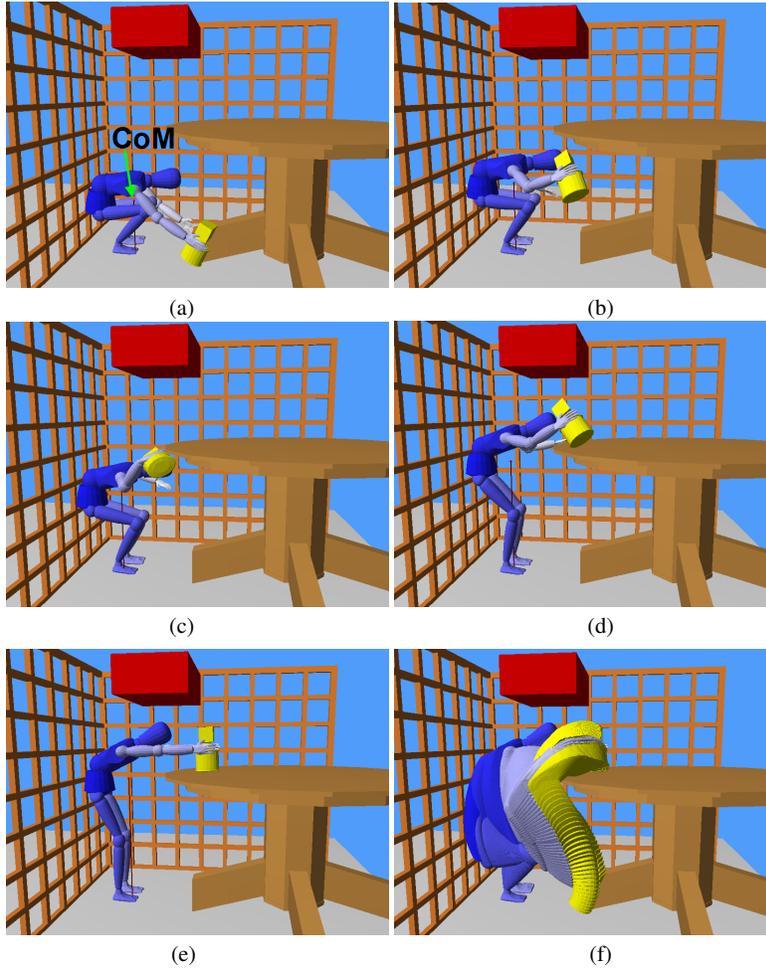
---



**Fig. 7** Our constrained coordination approach does not strictly constrain the motion of  $A^0$  on the path  $M^0(t)$ . Rather,  $A^0$  is allowed to move away from the path locally based on refinement. The extent of perturbation is determined by a Gaussian distribution function. Within a sample-based planner, the local path refinement is implicitly performed by using our constrained sampling and constrained interpolation schemes.

case). If the configuration  $\mathbf{q}$  is not statically stable, we perturb it to generate a statically stable configuration  $\mathbf{q}'$  meanwhile ensuring the foot placement is not changed. The process of perturbation can be reduced as an inverse kinematic problem. The projection of CoM point is treated as one end-effector in the IK problem. The Jacobian of this end-effector can be easily derived according to the kinematics of the robot and the mass of each body part. In the IK problem, this end-effector needs to be moved towards the center of the support polygon until it becomes inside the polygon. In order to maintain the foot contacting constraint, we choose three contacting points from each contacting foot as additional end-effectors. In the IK problem, the positions of these end-effectors are not changed. To solve the IK problem, a damped least squares method can be employed [3].

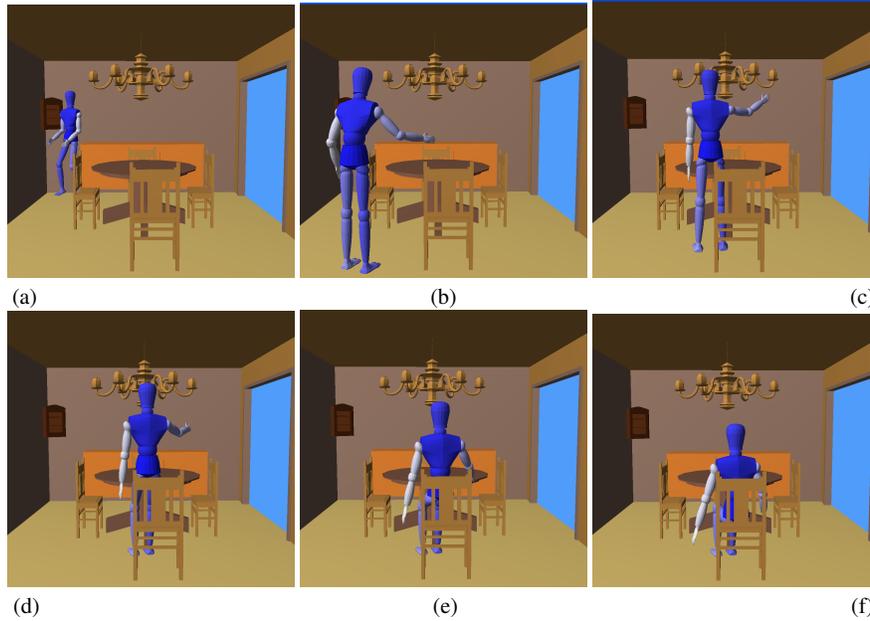
The modified constrained sampling allows us to generate statically stable samples for sampling-based planners. We also need to check whether the interpolating motion between samples are statically stable motion. One simple way is to discretely sample along the interpolating motion and check each sample individually. If any sample is not statically stable, we can perturb it by using our IK-based CoM perturbation algorithm.



**Fig. 8** The crouching robot picks the object from the ground and puts it on the table (a-e). When performing this task, the robot needs to avoid the collision with the environment and maintain its balance. Our algorithm can efficiently compute a valid motion for the robot within 10.117s. The entire motion is shown in (f). We highlight the center of mass (CoM) of the robot at each configuration. The projection of CoM onto the ground shows that the robot maintains the statically stable along the motion we have computed.

## 5 Implementation and Results

In this section, we describe our implementation and performance of our algorithm in many complex scenarios. We use a human-like robot with 40 DOF as shown in Fig. 6. The robot model is mobile and able to bend the torso or head, and sit. Six of the 40 DOF are unactuated and used to specify the position and the orientation of the virtual base. The robot is modeled by 22K triangles and it is decomposed into five body parts  $\{A^0, A^1, \dots, A^4\}$  in our benchmarks. The number of DOF for each body part are specified in Fig. 6.



**Fig. 9** The robot walks towards one chair and sits down. To avoid the collision with the overhead light, the robot needs to bend itself. This scenario has narrow passages and tight spaces, and therefore, the planner takes more time.

The underlying planner uses a sample-based path computation algorithm - bidirectional RRT [22]. We also augment the sampling and motion interpolation components to perform local path refinement. When planning the motion for first  $k$  parts of the robots, we ignore the rest of the body parts by temporarily deactivating those parts from motion. Moreover, we use PQP library for collision detection and closest distance queries with the obstacles and also among various parts of the robot. We use a damped least squares method for computing IK [3]. Our current implementation is not optimized and it is possible to improve the running time.

Figs. 8,9,10,11 show four complex scenarios that are used to analyze the performance of our algorithm. The resulting algorithm computes motion strategies corresponding to walking, sitting, bending and grabbing objects in complex scenarios. In Fig. 8, the robot is crouching. In order to pick the object from the ground and put it on the table, the robot needs to first stand up and then bend its torso. Our algorithm can efficiently compute a collision-free motion to achieve this task within 10.1s on a Pentium IV PC. The second benchmark scenario shows the motion of the human-like robot in a dining room (see Fig. 9). In this case, the robot walks from its initial position towards the dining table and eventually sits on the table. In Fig. 10, a whole-body motion for the robot is computed by our planner. The robot is able to pass through a tight space or a narrow passage between the two bookshelves and eventually sit down on the chair. When the robot passes through the narrow passage, it needs to coordinate its arm motion as well as the lower body motion to avoid col-

Stage	Bookshelves		Dinning Room		Car		Crouching	
	Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes
$A^0$	4.719	269	24.328	559	0.407	51	0.100	102
$A^0, A^1$	2.180	115	5.719	134	0.563	100	1.063	104
$A^0, A^1, A^2$	2.165	126	6.000	132	1.719	237	1.234	105
$A^0, A^1, A^2, A^3$	5.201	89	6.796	74	16.891	1,436	4.266	183
$A^0, A^1, A^2, A^3, A^4$	3.504	80	14.641	168	5.453	257	3.375	109
Overall Planning	18.328(s)		56.809(s)		25.078(s)		10.117(s)	

**Table 1** Performance of our approach on various benchmarks. We show the timing and the nodes in the resulting RRT at each stage of our constrained coordination. We also highlight the total timing for each benchmark. Our approach computes a collision-free path for the human-like robot with up to tens of seconds on various scenarios.

	Bookshelves	Dinning Room	Car	Crouching
Decomposition as Fig. 1 (s)	18.3	56.8	25.0	10.1
Decomposition of lower and upper bodies (s)	84.3	63.8	69.6	19.7
Centralized approach (s)	191.6	73.0	113.3	73.4

**Table 2** Comparison of the performance between our approaches based on different decomposition schemes and the centralized approach.

lisions with the obstacles. The total computational time to compute a collision-free path for this benchmark is 18.3s.

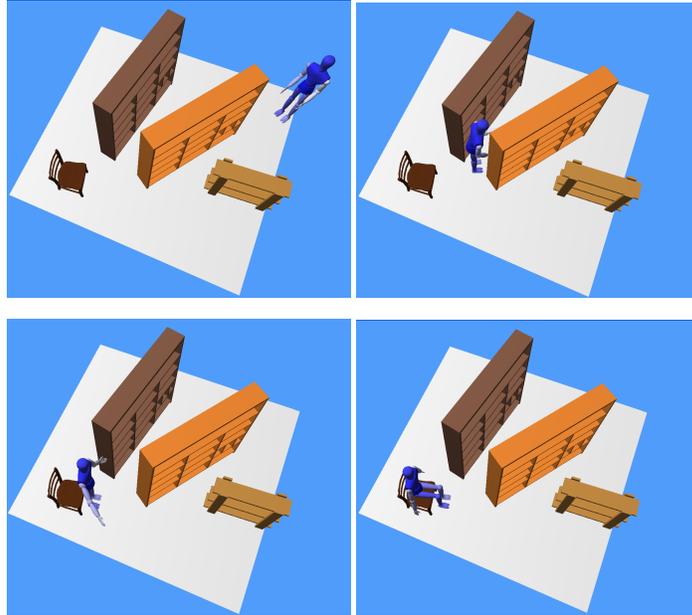
In Fig. 11, we show a scenario arising in CAD application. The human-like robot’s right hand is grabbing a tool. The human-like robot needs to move his body inside the to fix some parts using the tool. The CAD model of the car has 244k triangles and the algorithm needs to check for collisions with the car seat, roof and other parts. Our algorithm can efficiently compute a collision-free motion for this benchmark in 25.1s.

In table 1, we show the timing and nodes corresponding to each stage of the constrained coordination algorithm. In these examples, the locomotion such as walking, sitting, standing-up currently are generated using kinematic pattern generators (e.g. a walking cycle generator).

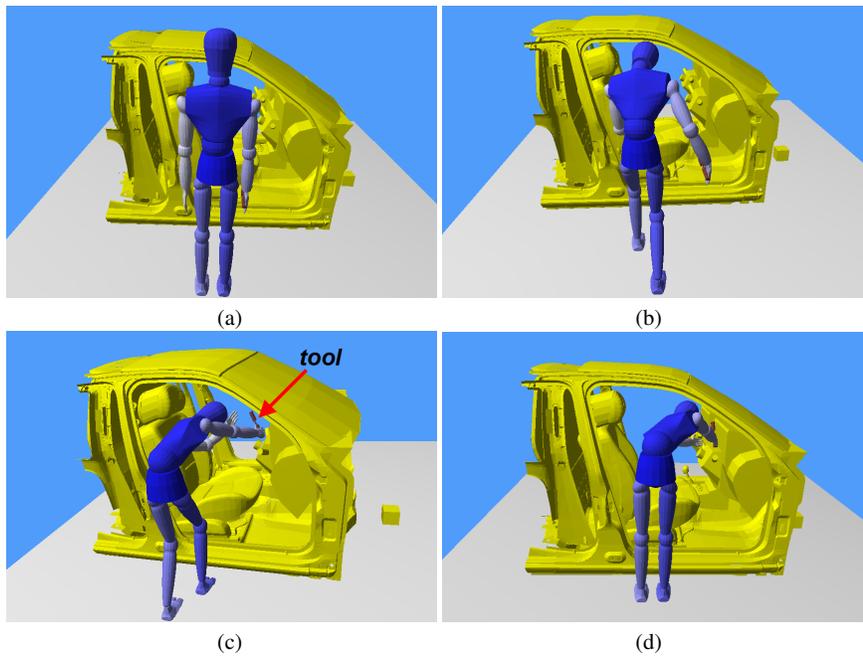
We compared the performance of our approach based on the decomposition as Fig. 1, our approach based on the lower-body and upper body decomposition, and the centralized planner applied to the entire robot. The table 2 shows up to 10 times performance speedups obtained by our approaches over the centralized approach. Our approaches often achieve more speedups in more cluttered environments.

## 5.1 Limitations

Our approach has many limitations. The underlying planner is not complete and its performance can vary with the scenario and the start or goal configurations. The performance depends considerably on the specific path computed for the root  $A^0$  of the tree in the sub-space of the configuration. In the subsequent stages we only use local refinement techniques to perform local modifications to the path. A poor path computed for  $A^0$  can affect the performance of the entire planner. Secondly,



**Fig. 10** The robot is able to pass through a narrow passage between two bookshelves and sit down.



**Fig. 11** The robot's right hand is grabbing a tool. The robot needs to move its upper-body inside a car to fix some parts with the tool. Our algorithm can efficiently compute a collision-free motion for the robot in 25.1s.

the motion computed by our planner can at times result in unnatural motion for the robot. This can happen if the initial and goal configurations are far apart and our constraints don't guarantee that the computed motion will look realistic.

## 6 Conclusions and Future Work

In this paper, we have presented an algorithm to compute whole-body motion for human-like robots. Our approach can handle high-DOF robots and uses decomposition strategies to reduce the problem to a sequence of low-dimensional problems. We use constrained coordination approach that solves each sub-problem incrementally, and performs local refinement to satisfy collision-free and statically stable constraints on CoM. We have demonstrated the performance on a 40-DOF robot in complex scenarios and generate collision-free motion paths corresponding to walking, sitting, bending in complex scenes with tight spaces and narrow passages.

There are many avenues for future work. We would like to compute dynamically stable motions by incrementally enforcing dynamics constraints within our coordination approach. In addition, we would like to apply our approach to more complex scenarios that arise in virtual prototyping including ergonomic analysis. We would like to demonstrate on complex models with more DOF and difficult narrow passages such as part removal from an engine.

## References

1. G. Arechavaleta, C. Esteves, and J.-P. Laumond. Planning fine motions for a digital factotum. pages 822–827, 2004.
2. O. Brock and L. E. Kavraki. Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces. In *Proceedings of International Conference on Robotics and Automation*, pages 1469–1474, 2001.
3. S. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods, 2004.
4. H.-L. Cheng, D. Hsu, J.-C. Latombe, and G. Sánchez-Ante. Multi-level free-space dilation for sampling narrow passages in PRM planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1255–1260, 2006.
5. S. Dalibard and J.-P. Laumond. Control of probabilistic diffusion in motion planning. In *Proc. of Eighth Workshop on Algorithmic Foundations of Robotics*, 2008.
6. H. Demirel and V. Duffy. Applications of digital human modeling in industry. In *Digital Human Modeling, Lecture Notes in Computer Science*, volume 4561, pages 824–832. Springer Berlin / Heidelberg, 2007.
7. R. Diankov, N. Ratliff, D. Ferguson, S. Srinivasa, and J. Kuffner. Bispaces planning: Concurrent multi-space exploration. In *Robotics: Science and Systems*, June 2008.
8. E. Drumwright and V. Ng-Thow-Hing. Toward interactive reaching in static environments for humanoid robots. In *IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS)*, pages 846–851, Beijing, China, Oct 2006.
9. M. Erdmann and T. Lozano-Perez. On multiple moving objects. *Proc. of IEEE Conf. on Robot. & Autom.*, pages 1419–1424, 1986.
10. D. Ferguson, N. Kalra, and A. Stentz. Replanning with RRTs. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2006.
11. M. Gienger, C. Goerick, and E. Körner. Whole body motion planning - elements for intelligent systems design. *Zeitschrift Künstliche Intelligenz*, 4:10–15, 2008.
12. K. Harada, S. Hattori, H. Hirukawa, M. Morisawa, S. Kajita, and E. Yoshida. Motion planning for walking pattern generation of humanoid. In *Proceedings of International Conference on Robotics and Automation*, pages 4227–4233, 2007.

13. K. Hauser, T. Bretl, K. H. J.C. Latombe, and B. Wilcox. Motion planning for legged robots on varied terrain. *The International Journal of Robotics Research*, 27(11-12):1325–1349, Nov-Dec 2008.
14. Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie. Planning walking patterns for a biped robot. *IEEE Transactions on Robotics and Automation*, 17:280–289, 2001.
15. P. Isto and M. Saha. A slicing connection strategy for constructing prms in high-dimensional cspaces. *Proceedings of International Conference on Robotics and Automation*, pages 1249–1254, 2006.
16. 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. *Proceedings of International Conference on Robotics and Automation*, pages 1620–1626, 2003.
17. M. Kallmann and M. Mataric. Motion planning using dynamic roadmaps. *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, April 2004.
18. F. Kanehiro, H. Hirukawa, K. Kaneko, S. Kajita, K. Fujiwara, K. Harada, and K. Yokoi. Locomotion planning of humanoid robots to pass through narrow spaces. In *Proceedings of International Conference on Robotics and Automation*, pages 604–609, 2004.
19. J. Kuffner. Goal-directed navigation for animated characters using real-time path planning and control. In *In Proceedings of Captech'98*, pages 171–186. Springer-Verlag, 1998.
20. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue. Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1):105–118, Jan. 2002.
21. J.-P. Laumond, E. Ferre, G. Arechavaleta, and C. Esteves. Mechanical part assembly planning with virtual mannequins. In *The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing*, pages 132–137, Tokyo, 1989.
22. S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. *Robotics: The Algorithmic Perspective (Proc. of the 4th Int'l Workshop on the Algorithmic Foundations of Robotics)*, 2000.
23. P. O'Donnell and T. Lozano-Perez. Deadlock-free and collision-free coordination of two robot manipulators. *Proc. IEEE International Conference on Robotics and Automation*, pages 484–489, 1989.
24. J. Pettre, J.-P. Laumond, and T. Simeon. 3d collision avoidance for digital actors locomotion. pages 400–405, 2003.
25. S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. *Proc. of IEEE Conf. on Robotics and Automation*, 1993.
26. M. Saha and P. Isto. Multi-robot motion planning by incremental coordination. In *Proc. of IROS*, pages 5960–5963, 2006.
27. L. Sentis and O. Khatib. Whole-body control framework for humanoids operating in human environments. *Proceedings of International Conference on Robotics and Automation*, 2006.
28. M. Stilman. Task constrained motion planning in robot joint space. *IEEE/RAS International Conference on Robots and Systems IROS'07*, 2007.
29. M. Vukobratovic and B. Borovac. Zero-moment point thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.
30. K. Yamane, J. J. Kuffner, and J. K. Hodgins. Synthesizing animations of human manipulation tasks. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3), Aug. 2004.
31. Y. Yang and O. Brock. Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation. *Proceedings of Robotics: Science and Systems*, August 2006.
32. E. Yoshida. Humanoid motion planning using multi-level dof exploitation based on randomized methods. pages 3378–3383, 2005.
33. E. Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond. Humanoid motion planning for dynamic tasks. *International Conference on Humanoid Robots*, pages 1–6, 2005.
34. L. Zhang and D. Manocha. A retraction-based RRT planner. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3743–3750, 2008.