Dynamically-stable Motion Planning for Humanoid Robots

– 感覚行動統合全身型ヒューマノイド H6 での実現 –

○カフナー・ジェームズ　西脇 光一　加賀美 聡　稲葉 雅幸　井上 博允　東京大学情報工学専攻

○ James Kuffner, Koichi Nishiwaki, Satoshi Kagami, Masayuki Inaba, Hirochika Inoue
Dept. of Mechano-Informatics, The University of Tokyo

Abstract—We present an algorithm for computing stable collision-free motions for humanoid robots given full-body posture goals. Given a robot's internal model of the environment and a statically-stable desired posture, we use a randomized path planner to search the configuration space of the robot for a collision-free path. Balance constraints are imposed on incremental search motions in order to maintain the overall dynamic stability of the computed trajectories. An overview of the algorithm is presented along with preliminary results using an experimental implementation on a dynamic model of the H6 humanoid robot.

Key Words: Humanoid Robot, Dynamic Simulation, Path Planning, Randomized Algorithm

1. Introduction

Recent advances in hardware and software have enabled the implementation of sophisticated motion control strategies. In particular, dynamic simulation software has assisted in the realization of dynamic walking in several humanoid robots [1, 2, 3].

The algorithm described briefly in this paper and in detail in [4] represents our first attempt at automatically generating collision-free dynamically-stable motions from full-body posture goals. Our approach is to adapt techniques from an existing, successful path planner [5] by transforming incremental motions used during the search by a filter function that maintains dynamic balance constraints [6, 7]. Provided the initial and goal configurations correspond to statically-stable body postures, the path returned by the planner can be transformed into a collision-free and dynamically-stable trajectory for the entire body.

2. Robot Model and Assumptions

We have based our experiments on an approximate model of the H6 dynamic humanoid robot, including the kinematics and dynamic properties of the links. We assume that the robot is currently balanced in a statically-stable posture supported by either one or both feet. The location of the supporting foot or feet (in the case of dual-leg support) does not change during the planned motion. The robot is given a full-body goal posture that is statically-stable. The goal posture may be set explicitly by a human operator, or computed via inverse kinematics or other means (e.g. for reaching a limb towards a target location or object).

Any statically-stable trajectory can be transformed into a dynamically-stable trajectory by arbitrarily slowing down the motion. For these experiments, we utilize an online balance compensation scheme[7] as a method of generating a final dynamically-stable trajectory.

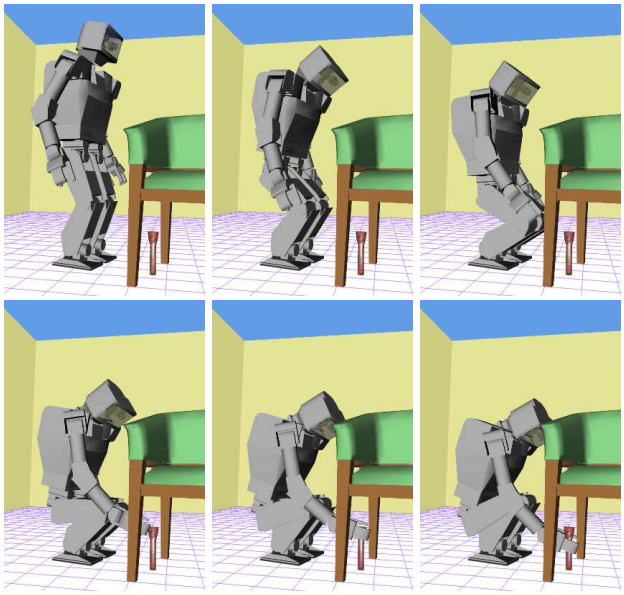

Fig.1 Dynamically-stable planned trajectory for grasping an object beneath a chair.

3. Dynamically-stable Path Planning

An approach to path planning was recently developed[8] that generates and connects two Rapidly-exploring Random Trees (RRTs) in a state space, which generalizes the configuration space ($\mathcal{C}$). A holonomic variant of this planner that adds a greedy heuristic to guide searches that can be conducted in $\mathcal{C}$ has also been devised[5]. The planner described here is similar to this method in that it performs its search in $\mathcal{C}$. However, it uses a balance compensation scheme to enforce dynamic constraints imposed upon the ZMP trajectory [7]. The compensation scheme is used as a filter function to incrementally transform purely kine-

```
EXTEND(T, q)
 1  q_near ← NEAREST_NEIGHBOR(q, T);
 2  if NEW_CONFIG(q, q_near, q_new) then
 3      T.add_vertex(q_new);
 4      T.add_edge(q_near, q_new);
 5      if q_new = q then Return Reached;
 7      else Return Advanced;
 9  Return Trapped;
```

```
RRT_CONNECT_STABLE(q_init, q_goal)
 1  T_a.init(q_init); T_b.init(q_goal);
 2  for k = 1 to K do
 3      q_rand ← RANDOM_STABLE_CONFIG();
 4      if not (EXTEND(T_a, q_rand) = Trapped) then
 5          if (EXTEND(T_b, q_new) = Reached) then
 6              Return PATH(T_a, T_b);
 7      SWAP(T_a, T_b);
 8  Return Failure
```

Fig.2 Algorithm for planning dynamically-stable motion.

matic paths into dynamically-stable trajectories.

Building a Tree: We modify the holonomic planner variant[5] that employs symmetric calls to the EXTEND function by altering the NEW_CONFIG function. The EXTEND function first generates a target configuration q_{target} by making an incremental step motion towards q from q_{near}. q_{new} is then generated by filtering the straight-line path connecting q_{near} and q_{target} through the dynamic balance compensator. This creates an incremental dynamically-stable trajectory from q_{near} towards q_{target}. The filter simulation terminates when the configuration converges or after a preset time limit is exceeded. If no collision occurs prior to termination, the most recent configuration output by the filter becomes q_{new} and is added to the tree $\mathcal{T}$. In this way, we are *guaranteed* that $q_{new} \in \mathcal{C}_{valid}$, the set of *collision-free*, *statically-stable* postures, and that there exists a *dynamically-stable trajectory* connecting them. In addition, rather than picking a purely random configuration $q_{rand} \in \mathcal{C}$ at every planning iteration, we pick a random configuration that also happens to correspond to a statically-stable posture of the robot. (i.e. $q_{rand} \in \mathcal{C}_{stable}$).

Path Search: Pseudocode for the complete modified algorithm is given in Fig 2. The main planning loop involves performing a simple iteration in which each step attempts to extend the RRT by adding a new vertex that is biased by a randomly-selected stable configuration. EXTEND selects the nearest vertex already in the RRT to the given sample configuration, q. The function NEW_CONFIG makes a dynamically-stable motion toward q as outlined previously using some fixed incremental distance ϵ to generate q_{target}. Three situations can occur: *Reached*, in which q is directly added to the RRT, *Advanced*, in which a new vertex $q_{new} \neq q$ is added to the RRT; *Trapped*, in which no new vertex is added due to the inability of the balance compensator to generate an incremental trajectory that lies in $\mathcal{C}_{valid}$.

4. Experiments

This section presents some preliminary experiments performed on a 270 MHz SGI O2 (R12000) workstation. We have implemented a prototype planner that runs within a graphical simulation environment. An operator can position individual joints or use inverse kinematics to specify body postures for the virtual robot. The filter function can be run interactively to ensure that the goal configuration is statically-stable. After specifying the goal, the planner is invoked to attempt to compute a dynamically-stable trajectory connecting the goal configuration to the robot's initial configuration (assumed to be a stable posture). Fig 1 shows a computed dynamically-stable motion for the H6 robot crouching to retrieve an object from beneath a chair obstacle (Fig 1). The total wall time elapsed in solving this query ranges from approximately 3.5 minutes to 12 minutes.

Task Description	Computation Time (seconds)			
	min	max	avg	stdev
Reach under chair	208	712	371	157

5. Discussion

This paper summarizes an algorithm for computing dynamically-stable collision-free motions for humanoid robots given full-body posture goals. Balance constraints are imposed upon incremental search motions computed by a randomized planner in order to maintain the overall dynamic stability of the computed trajectories. The current implementation of the planner is limited to body posture goals, and a fixed position for either one or both feet, but we are working to remove these limitations.

Acknowledgments: This research is funded by a Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowship for Foreign Scholars in Science and Engineering.

References

1) Kazuo Hirai. Current and future perspective of honda humanoid robot. In *Proc. of 1997 IEEE/RSJ Int. conf. on Intelligent Robots and Systems (IROS'97)*, pp. 500–508, 1997.
2) J. Yamaguchi, S. Inoue, D. Nishino, and A. Takanishi. Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk. In *Proc of 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98)*, pp. 96–101, 1998.
3) K. Nagasaka, M. Inaba, and H. Inoue. Walking pattern generation for a humanoid robot based on optimal gradient method. In *Proc. of 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1999.
4) J. Kuffner, S. Kagami, M. Inaba, and H. Inoue. Dynamically-stable motion planning for humanoid robots. In *Proc. of First IEEE-RAS Int. Conf. on Humanoid Robots*, September 2000. *To appear.*
5) J.J. Kuffner and S.M. LaValle. RRT-Connect: An efficient approach to single-query path planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA '2000)*, San Francisco, CA, April 2000.
6) Y. Tamiya, M. Inaba, and H. Inoue. Realtime balance compensation for dynamic motion of full-body humanoid standing on one leg. *Journal of the Robotics Society of Japan*, Vol. 17, No. 2, pp. 268–274, 1999.
7) Satoshi Kagami, Fumio Kanehiro, Yukiharu Tamiya, Masayuki Inaba, and Hirochika Inoue. AutoBalancer: An Online Dynamic Balance Compensation Scheme for Humanoid Robots. In *Proc. of the Fourth Int. Workshop on Algorithmic Foundations on Robotics (WAFR'00)*, 2000.
8) S.M. LaValle and J.J Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA '99)*, Detroit, MI, May 1999.