EFFICIENT ALGORITHMS FOR ROBOTS WITH HUMAN-LIKE STRUCTURES AND INTERACTIVE HAPTIC SIMULATION

O. Khatib, O. Brock, K.-S. Chang, D. Ruspini,
L. Sentis, F. Conti, and S. Viji
Robotics Laboratory, Department of Computer Science
Stanford University, Stanford, California 94305
{khatib, oli, kcchang, ruspini, lsentis, conti, sviji}@robotics.stanford.edu

Abstract A new field of robotics is emerging. Robots are today moving towards applications beyond the structured environment of a manufacturing plant. They are making their way into the everyday world that people inhabit. The paper focuses on models, strategies, and algorithms associated with the autonomous behaviors needed for robots to work, assist, and cooperate with humans. In addition to the new capabilities they bring to the physical robot, these models and algorithms and more generally the body of developments in robotics is having a significant impact on the virtual world. Haptic interaction with an accurate dynamic simulation provides unique insights into the real-world behaviors of physical systems. The potential applications of this emerging technology include virtual prototyping, animation, surgery, robotics, cooperative design, and education among many others.

Keywords: Operational Space Control, Dynamic Simulation, Mobile Manipulation, Real-Time Path Modification, Haptics

1. Introduction

The successful introduction of robotics into human environments will rely on the development of competent and practical systems that are dependable, safe, and easy to use. To work, cooperate, assist, and interact with humans, the new generation of robot must have mechanical structures that accommodate the interaction with the human and adequately fit in his unstructured and sizable environment. Human-compatible robotic structures must integrate mobility (legged or wheeled) and manipulation (preferably bi-manual), while providing the needed access to perception and monitoring (head camera) [Hirai et. al., 1998; Takanishi et. al., 1998; Khatib et. al., 1999; Asfour et. al., 1999; Nishiwaki et. al., 2000]. These requirements imply robots with branching structures - tree-like topology involving much larger numbers of degrees of freedom than those usually found in conventional industrial robots. The substantial increase in the dimensions of the corresponding configuration spaces of these robots renders the set of fundamental problems associated with their modeling, programming, planning, and control much more challenging. In this article we present algorithmic foundations developed in our laboratory to address these issues.

2. Whole-Robot Control: Task and Posture

Human-like structures share many of the characteristics of macro/mini structures [Khatib, 1995]: coarse and slow dynamic responses of the mobility system (the macro mechanism), and the relatively fast responses and higher accuracy of the arms (the mini device). Inspired by these properties of macro/mini structures, we have developed a framework for the coordination and control of robots with human-like structures.

2.1 Task Dynamic Behavior

The joint space dynamics of a manipulator are described by

$$A(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{\Gamma} \tag{1}$$

where $\mathbf{q}$ is the n joint coordinates, $A(\mathbf{q})$ is the $n \times n$ kinetic energy matrix, $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of centrifugal and Coriolis joint forces, $\mathbf{g}(\mathbf{q})$ is the vector of gravity, and $\mathbf{\Gamma}$ is the vector of generalized joint forces.

The *operational space formulation* [Khatib, 1987] provides an effective framework for dynamic modeling and control of branching mechanisms [Russakow et. al., 1995], with multiple operational points. The generalized torque/force relationship [Khatib, 1987; Khatib, 1995] provides the decomposition of the total torque, $\mathbf{\Gamma}$ (equation 1) into two dynamically decoupled command torque vectors: the torque corresponding to the task behavior command vector and the torque that only affects posture behavior in the null space:

$$\mathbf{\Gamma} = \mathbf{\Gamma}_{task} + \mathbf{\Gamma}_{posture} \tag{2}$$

For a robot with a branching structure of m effectors or operational points, the task is represented by the $6m \times 1$ vector, $\mathbf{x}$, and the $6m \times n$ Jacobian matrix is $J(\mathbf{q})$. This Jacobian matrix is formed by vertically concatenating the m $6 \times n$ Jacobian associated with the m effectors.

The task dynamic behavior is described by the operational space equations of motion [Khatib, 1995]

$$\Lambda(\mathbf{x})\ddot{\mathbf{x}} + \mu(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{p}(\mathbf{x}) = \mathbf{F} \tag{3}$$

where $\mathbf{x}$, is the vector of the $6m$ operational coordinates describing the position and orientation of the m effectors, $\Lambda(\mathbf{x})$ is the $6m \times 6m$ kinetic energy matrix associated with the operational space. $\mu(\mathbf{x}, \dot{\mathbf{x}})$, $\mathbf{p}(\mathbf{x})$, and $\mathbf{F}$ are respectively the centrifugal and Coriolis force vector, gravity force vector, and generalized force vector acting in operational space.

The joint torque corresponding to the task command vector $\mathbf{F}$, acting in the operational space is

$$\mathbf{\Gamma}_{\text{task}} = J^T(\mathbf{q})\mathbf{F} \tag{4}$$

The task dynamic decoupling and control is achieved using the control structure

$$\mathbf{F}_{\text{task}} = \widehat{\Lambda}(\mathbf{x})\mathbf{F}^{\star}_{\text{motion}} + \widehat{\mu}(\mathbf{x}, \dot{\mathbf{x}}) + \widehat{\mathbf{p}}(\mathbf{x}) \tag{5}$$

where, $\mathbf{F}^{\star}_{\text{task}}$ represents the inputs to the decoupled system, and $\widehat{.}$ represents estimates of the model parameters.

2.2 Posture Behavior

An important consideration in the development of posture behaviors is the interactions between the posture and the task. It is critical for the task to maintain its responsiveness and to be dynamically decoupled from the posture behavior. The posture can then be treated separately from the task, allowing intuitive task and posture specifications and effective whole-robot control. The overall control structure for task and posture is given by equation 2, where

$$\mathbf{\Gamma}_{\text{posture}} = N^T(\mathbf{q})\mathbf{\Gamma}_{\text{desired}-\text{posture}} \quad \text{with} \quad N^T(\mathbf{q}) = \left[I - \overline{J}(\mathbf{q})J(\mathbf{q})\right] \tag{6}$$

where $\overline{J}(\mathbf{q})$ is the *dynamically consistent generalized inverse* [Khatib, 1995], which minimizes the robot kinetic energy.

This relationship provides a decomposition of joint forces into two control vectors: joint forces corresponding to forces acting at the task, $J^T\mathbf{F}$, and joint forces that only affect the robot posture, $N^T\mathbf{\Gamma}_{\text{posture}}$ (see Figure 1). For a given task this control structure produces joint motions that minimize the robot's instantaneous kinetic energy.

Dynamic consistency is the essential property for the task behavior to maintain its responsiveness and to be dynamically decoupled from the posture behavior since it guarantees not to produce any coupling acceleration in the operational space given any τ_{null}. In addition, dynamic consistency enables task behavior and posture behavior to be specified independently of each other, providing an intuitive control of complex systems.

For instance, the robot posture can be controlled to maintain the robot total center-of-mass aligned along the $\mathbf{z}_0$ axis of the reference frame. This

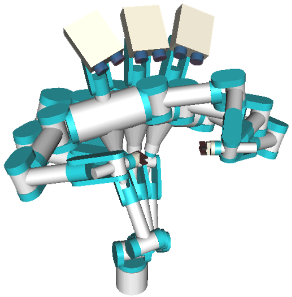

Figure 1. Dynamic Consistency: a sequence of snapshots from the dynamic simulation of a 24 degree-of-freedom humanoid system. The task consists of maintaining the position of the hands. The torso of the robot performs a motion in the posture space without affecting task execution.

posture can be simply implemented with a posture energy function

$$V_{\text{posture-energy}}(\mathbf{q}) = \frac{1}{2}k(x_{\text{CoM}}^2 + y_{\text{CoM}}^2) \tag{7}$$

where k is a constant gain, x_{CoM}, and y_{CoM} are the x and y coordinates of the center of mass. The gradient of this function provides the required attraction to the z axis of the robot center of mass. This is illustrated in the simulation shown in Figure 2.

Figure 2. Posture Behavior: a sequence of snapshots from the dynamic simulation of a 21-degree-of-freedom humanoid system. The hands are haptically teleoperated, while the remaining degrees of freedom are controlled automatically in the posture space. The posture controls the center of mass to balance the robot, while maintaining a natural arm configuration.

More complex posture behaviors can be obtained by combining various posture energies. We are currently exploring the generation of human-like natural motion from motion capture of human and the extraction of motion characteristics using human biomechanical models.

2.3 Efficient Operational Space Algorithms

Early work on efficient operational space dynamic algorithms has focused on open-chain robotic mechanisms. An efficient $O(n)$ recursive algorithm was developed using the spatial operator algebra [Kreutz-Delgado et. al., 1991] and the articulated-body inertias [Featherstone, 1987]. A different approach that avoided the extra computation of articulated inertias also resulted in an $O(n)$ recursive algorithm for the operational space dynamics [Lilly, 1992] Building on these early developments, our effort was aimed at algorithms for robotic mechanisms with branching structures that also address the issue of redundancy and dynamics in the null space.

The most computationally expensive element in the operational space whole-body control structure (equation 2) is the posture control, which involves the explicit inversion operation of the $n \times n$ joint space inertia matrix A,which requires $O(n^3)$. We have developed a computationally more efficient operational space control structure that eliminates the explicit computation of the joint space inertia matrix and its inverse. This elimination was achieved by combining the dynamically consistent null space control and the operational space control in a computationally more efficient dynamic control structure.

Using this control structure, we have developed a recursive algorithm for computing the operational space dynamics of an n-joint branching redundant articulated robotic mechanism with m operational points [Chang and Khatib, 2000]. The computational complexity of this algorithm is $O(nm + m^3)$, while existing symbolic methods require $O(n^3 + m^3)$. Since m can be considered as a small constant in practice, this algorithm attains a linear time $O(n)$ as the number of links increases. This work was extended for the dynamics of closed-chain branching mechanisms with an efficient $O(nm + m^3)$ algorithm [Holmberg and Khatib, 2000].

3. Interactive Haptic Simulation

Beyond their immediate application to physical robots, these efficient dynamic algorithms are making a significant impact on the simulation and interaction with the virtual world. The computational requirements associated with the haptic interaction with complex dynamic environments are quite challenging. In addition to the need for real-time free-motion simulation of multi-body systems, contact and impact resolution and constrained motion simulation are also needed.

Building on the operational space formulation, we developed a general framework [Ruspini and Khatib, 1999] for the resolution of multi-contact between articulated multi-body systems. A contact point is treated as an operational point and a contact space is defined. Similarly to the operational space inertia matrix, a contact space inertia matrix Λ is introduced to provide the effective masses seen at all the contact points and to characterize the dynamic relationships between them. Computing the contact space inertia matrices Λ for a number of m contact point on a branching mechanism is achieved with an efficient $O(nm + m^3)$ recursive algorithm.

The contact space representation allows the interaction between groups of dynamic systems to be described easily without having to examine the complex equations of motion of each individual system. As such, a collision model can be developed with the same ease as if one was considering interaction only between simple bodies. Impact and contact forces between interacting bodies can then be efficiently solved to prevent penetration between all the objects in the environment. Such dynamic interactions are shown in Figure 3.

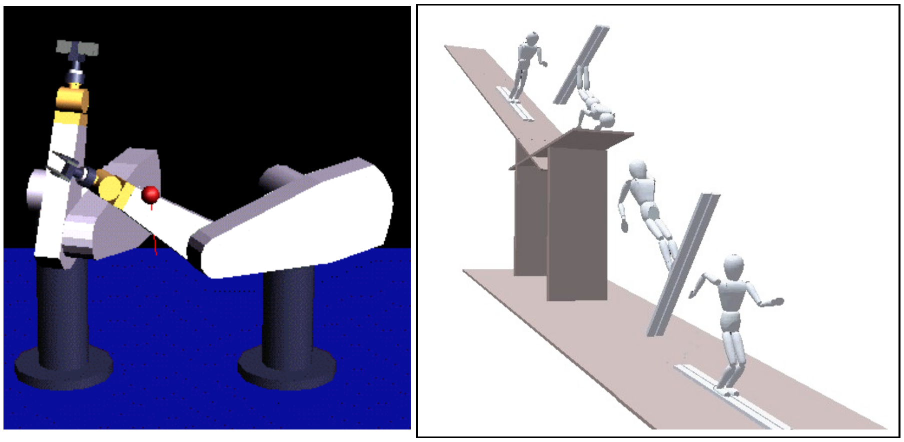

Figure 3. The capabilities of dynamic simulation are integrated with haptic force feedbeck to allow direct interaction between the user and the objects in the environment (left). The user can attach to any object in the scene and feel contact forces resulting from dynamic interactions. Dynamic simulation is capable of generating complicated, physically consistent behavior: The sliding motion and subsequent jump with the landing are simulated dynamically (right).

4. Task-Consistent Elastic Plans

The control methods presented in Section 2 allow the consistent control of task and posture for robots with complex mechanical structures. To perform or assist in the execution of complex actions, however, these control structures have to be linked with motion generated by a planner. Furthermore, since unstructured environments can be highly dynamic, such an integration has to accommodate unforeseen obstacle motion in real time, while conforming to constraints imposed by the task. We have developed algorithms that perform task-consistent, real-time path modification to address this issue.

Motion planners generally perform a global search in configuration space to determine a collision-free motion accomplishing a given task. Due to the high dimensionality of the configuration space of the class of robots we are concerned with in this paper, planning operations are too computationally complex to be performed in real time. As a consequence, motion in dynamic environments cannot adequately be generated by those planners. The elastic band framework [Quinlan and Khatib, 1993] was developed to allow real-time modification of a previously planned path, effectively avoiding a costly planning operation in reaction to changes in the environment. More recently, this framework was complemented by the elastic strip framework [Brock and Khatib, 1998], which allows reactive path modifcation for robots with many degrees of freedom.

In dynamic environments it is desirable to integrate path modification with task behavior. To accomplish this we extend the overall control structure for task and posture behavior (equation 2 by adding torques $\mathbf{\Gamma}_{obstacle}$ representing desired obstacle avoidance behavior:

$$\mathbf{\Gamma} = \mathbf{\Gamma}_{task} + \mathbf{\Gamma}_{posture} + \mathbf{\Gamma}_{obstacle}$$

Both, $\mathbf{\Gamma}_{posture}$ and $\mathbf{\Gamma}_{obstacle}$ have been mapped into the null space of the task, as shown in equation 6. Using this control structure, the task-consistent obstacle avoidance behavior shown in the second image of Figure 4 is achieved. Note how without task-consistency the end effector deviates significantly from the required straight-line trajectory. Using task-consistent obstacle avoidance, the end effector only deviates minimally from the task, as can be seen in the graphs shown in Figure 4.

This approach of integrating task and obstacle avoidance behavior can fail, however, when the torques resulting from mapping $\mathbf{\Gamma}_{obstacle}$ into the null space yield insufficient motion to ensure obstacle avoidance. In such a situation it would be desirable to suspend task execution and to realize obstacle avoidance with all degrees of freedom of the robot.

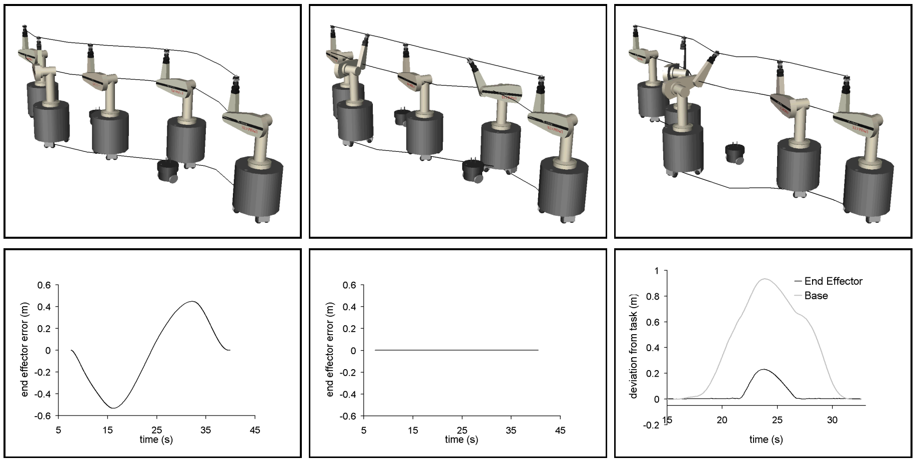

Figure 4. Top images show from left to right: real-time obstacle avoidance without task consistency, with task consistency, and transitioning between task-consistent and non task-consistent behavior. Lines indicate trajectories of the base, elbow, and end effector. The graphs show end effector error and base deviation from the task for the respective experiment.

The coefficient c corresponds to the ratio of the magnitude of the torque vector $\mathbf{\Gamma}_{obstacle}$ mapped into the null space of the task and its unmapped magnitude. This coefficient is an indication of how well the behavior represented by $\mathbf{\Gamma}_{obstacle}$ can be performed inside the null space of the task. We experimentally determine a value c_s at which it is desirable to suspend task execution in favor of the behavior previously mapped into the null space. Once the coefficient c assumes a value $c < c_s$, a transition is initiated. During this transition task behavior is gradually suspended and previous null space behavior is performed using all degrees of freedom of the manipulator. The motion of the manipulator is now generated using the equations

$$\begin{aligned}\mathbf{\Gamma}_{\text{task}-\text{consistent}} &= J^T(\mathbf{q})\,\mathbf{F} \;+\; N(\mathbf{q})^T\mathbf{\Gamma}_{\text{obstacle}} \\ \mathbf{\Gamma} &= \alpha\,\mathbf{\Gamma}_{\text{task}-\text{consistent}} \;+\; \overline{\alpha}\,\mathbf{\Gamma}_{\text{obstacle}}\end{aligned}$$

where α is a time-based transition variable, transitioning between 1 and 0 during task suspension and between 0 and 1 during resumption of the task, and $\overline{\alpha} = (1-\alpha)$ is defined as the complement of α.

The experimental results, performed on the Stanford Assistant Manipulator, for such transistioning behavior can be seen in Figure 4. The image on the top right shows how despite task-consistent obstacle avoidance the task has to be suspended to ensure obstacle avoidance. Below, the graph shows how the base deviates significantly from the straight line in response to the obstacle. The end effector, however, maintains the task until it has to be suspended. The graph also shows that the task is resumed in a smooth manner, after the base has passed the obstacle

5. Conclusion

Advances toward the challenge of robotics in human environments depend on the development of the basic capabilities needed for both autonomous operations and human/robot interaction. In this article, we have presented methodologies for whole-robot coordination and control, cooperation between multiple robots, interactive haptic simulation with contact, and the real-time modification of collision-free path to accommodate changes in the environment.

For the whole-robot coordination and control, we presented a framework which provides the user with two basic task-oriented control primitives: task control and posture control. The major characteristic of this control structure is the dynamic consistency it provides in implementing these two primitives: the robot posture behavior has no impact on the end-effector dynamic behavior.

Addressing the computational challenges of human-like robotic structures, we presented efficient $O(nm + m^3)$ recursive algorithms for the operational space dynamics of mechanisms involving branching structures and closed chains. Building on the operational space formulation, we also developed a framework for the resolution of multi-contact between articulated multi-body systems.

The elastic strip framework allows the seamless integration of reactive, real-time obstacle avoidance and the task-oriented control structure. It provides for real-time motion generation that combines obstacle avoidance and task execution. When kinematic or external constraints imposed by obstacles make it impossible to maintain the task, task-consistent obstacle avoidance is suspended and all degrees As the constraints are relaxed, the task is resumed in a smooth manner.

Acknowledgments

The financial support of Honda Motor Company and NSF (grants IRI-9320017) is gratefully acknowledged. Many thanks to Alan Bowling, Arancha Casal, Robert Holmberg, Jaeheung Park, Costas Stratelos, James Warren, and Kazuhito Yokoi for their valuable contributions to the work reported here.

References

Asfour, T., K. Berns, J. Schelling, and R. Dillmann (1999), Programming of Manipulation Tasks of the Humanoid Robot ARMAR, *Proc. of the Intl. Conf. on Advanced Robotics.*

Brock, O. and O. Khatib (1998), Elastic Strips: Real-Time Path Modification for Mobile Manipulation, *Robotics Research*, Springer Verlag.

Chang, K.-S. and O. Khatib (2000), Operational Space Dynamics: Efficient Algorithms for Modelling and Control of Branching Mechanisms, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pp. 850–856.

Featherstone, R. (1987), *Robot Dynamics Algorithms*, Kluwer Academic Publishers.

Holmberg, B. and O. Khatib (2000), Development of a Holonomic Mobile Robot for Mobile Manipulation Tasks, *Intl. J. of Robotics Research*, vol. 19, no. 11, pp. 1066-1074.

Hirai, K., M. Hirose, Y. Haikawa, and T. Takenaka (1998), The Development of Honda Humanoid Robot, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, vol. 2, pp. 1321–1326, Leuven, Belgium.

Khatib, O. (1987), A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation, *Intl. J. of Robotics and Automation*, vol. 3, no. 1, pp. 43–53.

Khatib, O. (1995), Inertial Properties in Robotics Manipulation: An Object-Level Framework, *Intl. J. of Robotics Research*, vol. 14, no. 1, pp. 19-36.

Khatib, O., K. Yokoi, K.-S. Chang, D. Ruspini, R. Holmberg, and A. Casal (1996), Coordination and Decentralized Cooperation of Multiple Mobile Manipulators, *Intl. J. of Robotic Systems*, vol. 13, no. 11, pp. 755–764.

Khatib, O., K. Yokoi, O. Brock, K.-S. Chang, and A. Casal (1999), Robots in Human Environments: Basic Autonomous Capabilities, *Intl. J. of Robotics Research*, vol. 18, no. 7, pp. 175–183.

Kreutz-Delgado, K., A. Jain, and G. Rodriguez (1991), Recursive Formulation of Operational Space Control, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pp. 1750–1753.

Lilly, K. W. (1992), *Efficient Dynamic Simulation of Robotic Mechanisms*, Kluwer Academic Publishers.

Nishiwaki, K., T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, and H. Inoue (2000), Design and Development of Research Platform for Perception-Action Integration in Humanoid Robot: H6, *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1559–1564.

Quinlan, S. and O. Khatib (1993), Twowards Real-Time Execution of Motion Tasks, *Experimental Robotics 2*, Springer Verlag.

Russakow, J., O. Khatib, and S. M. Rock, (1995), Extended Operational Space Formation for Serial-To-Parallel Chain (Branching) Manipulators, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, vol. 1, pp. 1056-1061.

Ruspini, D. C. and O. Khatib (1999), Collision/Contact Models for Dynamic Simulation and Haptic Interaction, *Proc. Intl. Symp. of Robotics Research*, pp. 185–195.

Takanishi, A., S. Hirano, and K. Sato (1998), Development of an anthropomorphic head-eye system for a humanoid robot-realization of human-like head-eye motion using eyelids adjusting to brightness, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, vol. 2, pp. 1308–1314, Leuven, Belgium.