Continuous Trajectory Optimization for Autonomous Humanoid Door Opening

Matt Zucker

Department of Engineering
Swarthmore College
Swarthmore PA
Email: mzucker1@swarthmore.edu

Youngbum Jun, Brittany Killen,
Tae-Goo Kim, and Paul Oh

Drexel Autonomous Systems Laboratory
Drexel University
Philadelphia, PA
Email: {yj55,bak48,tk357,paul.yu.oh}@drexel.edu

***Abstract*—The upcoming DARPA Robotics Challenge (DRC) presents a demanding set of real-world tasks to be accomplished autonomously by robots. In this paper, we describe the development of a system to control an existing humanoid robot to open a door, one of the many tasks of the DRC. Special emphasis is placed upon generating smooth trajectories which minimize unnecessary motion of the robot. We describe methods for generating and optimizing trajectories for the robot, and present preliminary results demonstrated on the physical robotic platform. To the best of our knowledge, we demonstrate the first large scale application of the CHOMP trajectory optimization in a situation with closed kinematic chain constraints.**

I. Introduction

The DARPA Robotics Challenge (DRC) is a U.S. government-sponsored competition aimed at promoting innovation in robotic technology for disaster response operations. The primary goal is to develop robots capable of operating with human-like levels of competence in areas which are too hazardous for humans to go, for instance nuclear disaster sites. A secondary goal is to broaden the set of tools and technologies for those interested in developing hardware and software for ground robot systems. Central to the DRC are eight tasks including driving a utility vehicle, locomotion over rough terrain, clearing debris, climbing a ladder, and operating and repairing equipment, to name just a few. In this work, we focus on the task of autonomous door opening.

Drexel University was the recent recipient of an NSF Major Research Infrastructure (MRI) grant which provided funding to furnish several U.S. based institutions with the HUBO+ humanoid robot, designed and manufactured by KAIST. An earlier version of the robot is described in [1]. Hence, a team was formed consisting of Drexel and several partner institutions who were uniquely poised to use the HUBO+ platform to work on the DRC in a distributed fashion. Work on the door opening task is being conducted jointly between Swarthmore College and Drexel University, at the Drexel Autonomous Systems Laboratory.

The present phase of our development on the door opening task deals only with generating correct kinematic motion of the robot. Future works will address onboard perception and more complex dynamics. Quasi-static stability (i.e. maintaining the center of mass inside the support polygon of the robot) is the only dynamical consideration in this work. Hence, our chief goal is generating smooth motion of the robot to solve the door opening task, which we accomplish by using the Covariant Hamiltonian Optimization and Motion Planning (CHOMP) algorithm [2].

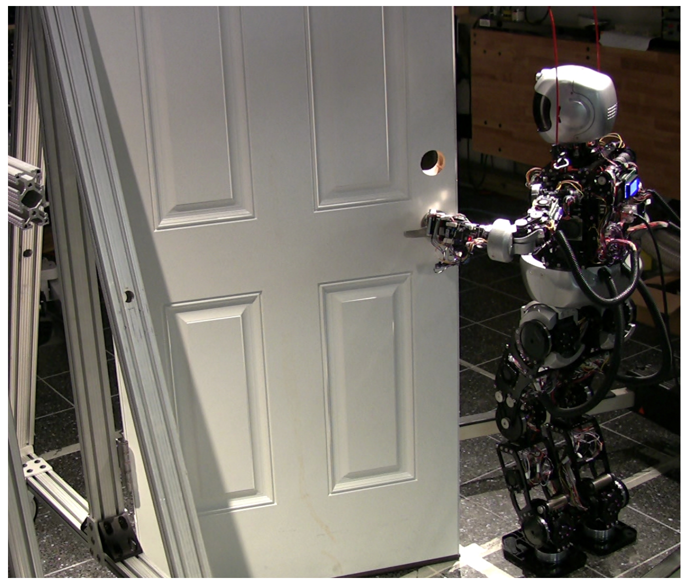

Fig. 1. HUBO+ humanoid robot opening a standard door with ADA-compliant handle.

In the remainder of the paper, we review related work, describe some of the theory behind the CHOMP algorithm, and describe implementation on the physical robotic platform.

A. Related Work

The DRC is not the first instance of autonomous door opening in robotics. In 2009, Arisumi et al. developed a system to allow the HRP-2 robot to open doors by applying impulsive forces to a swinging door [3]. The system also controls the robot to hold the door open as it walks through. In the same year, the Willow Garage PR-2 robot completed a milestone of

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) award #N65236-12-1-1005 for the DARPA Robotics Challenge.

autonomously completing over 25 miles of indoor navigation, including many instances of autonomous door opening to navigate between rooms [4]. We expect this work to prove useful especially as we begin to explore how to integrate onboard perception into our door opening experiments.

II. Theory

The CHOMP algorithm [2] was founded upon the observation that state-of-the-art sampling based motion planners such as PRM or RRT [5], [6] are simultaneously both over- and under-powered. They are overpowered in that they are designed to overcome so-called "narrow passages" in the free configuration space of the robot, which are not necessarily present in many real-world motion generation tasks such as manipulation or locomotion, and they are underpowered in that the paths they generate typically contains jerk or unnecessary motion of the robot which must be removed in a subsequent post-processing step. The goal of CHOMP is to directly produce smooth motion which satisfies the myriad constraints (collision free, kinematically and dynamically valid) on the robot. CHOMP has been used successfully to generate robot motion in a variety of scenarios, including legged locomotion over rough terrain [7].

Although the original version of CHOMP did not directly address complex kinematic constraints beyond joint limits and kinematic reachability, subsequent updates to the algorithm included them [8]. We use such constraints in this work to model the joint system of robot and door as a closed kinematic chain. In the following subsections, we derive a simplified version of the constrained CHOMP algorithm which neglects collisions (since the space around the door is assumed to be relatively uncluttered). Although we generally follow the derivations in [2] and [8], a more detailed derivation is provided in a paper currently submitted for publication [9].

A. The CHOMP smoothness objective

Let $q \in \mathcal{C} \subset \mathbb{R}^m$ be a robot configuration. Then a trajectory of the robot ξ can be represented as a set of n regularly spaced samples $q^{(1)}, \ldots, q^{(n)}$ with fixed endpoints $q^{(0)}$ and $q^{(n+1)}$, and separated by a timestep Δt. Hence, independent of the endpoints (which are treated separately), ξ can be viewed as a vector in $\mathbb{R}^{mn}$.

The smoothness of the trajectory can be expressed as some sum of squared derivatives, e.g. velocity. Hence, we write the *smoothness objective* as a sum of squared finite differences

$$f(\xi) = \frac{1}{2} \sum_{j=1}^{m} \sum_{t=1}^{n+1} \left(q_j^{(t)} - q_j^{(t-1)} \right)^2 \tag{1}$$

where $q_j^{(t)}$ denotes the value of the j^{th} degree of freedom at time t. We omit the constant factor of Δt^2 in the expression above since it does not affect the minimum attained by the optimization. Since the differentiation above is a linear operator, we can rewrite the expression in terms of an $mn \times mn$ finite differencing matrix

$$K = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ -1 & 1 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 1 & \ldots & 0 & 0 \\ & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & -1 & 1 \\ 0 & 0 & 0 & \ldots & 0 & -1 \end{bmatrix} \otimes I_{m \times m}$$

where $\otimes$ denotes the Kronecker (tensor) product. By defining an appropriate vector e which accounts for the fixed endpoints, we can then rewrite (1) as a simple quadratic form on ξ:

$$f(\xi) = \frac{1}{2} \xi^T A \xi + \xi^T b + c \tag{2}$$

where $A = K^T K$, $b = K^T e$, and $c = \frac{1}{2} e^T e$. We note that A is positive definite and very sparse (owing to the sparsity of K), and that A, b, and c are constant with respect to the interior points of the trajectory. Without any additional constraints, the minimum velocity trajectory in this case is a straight line in configuration space from $q^{(0)}$ to $q^{(n+1)}$, found by minimizing (2) for ξ:

$$\xi = -A^{-1} b \tag{3}$$

However, as we will show in the next section, we consider several types of constraints in addressing the door opening task, which prohibit such trivial solutions. Furthermore, in this work, we minimize the sum of squared accelerations, not velocities, which can be straightforwardly obtained by a suitable modification of K and e above.

Since A is sparse and symmetric positive definite, we obtain solutions to equations of the form of (3) in $O(mn)$ time using the skyline Cholesky decomposition, as opposed to the naïve runtime of $O(m^3 n^3)$ [10].

B. Constrained CHOMP

Now we wish to minimize (2) subject to the k equality constraints of the form $h(\xi) = 0$, with $h : \mathbb{R}^n \mapsto \mathbb{R}^k$. To accomplish this, we will use an iterative method to find a small perturbation δ which improves the current trajectory ξ. That is, given the current trajectory ξ, we wish to minimize $f(\xi + \delta)$ subject to $h(\xi + \delta) = 0$, while maintaining a small step size δ. The objective to be optimized is therefore the Lagrangian

$$L(\delta, \lambda) = f(\xi + \delta) + \frac{1}{2\alpha} \|\delta\|_A^2 + \lambda^T h(\xi + \delta) \tag{4}$$

Here α denotes the step size of the algorithm (small α penalizes large δ), and $\|\delta\|_A^2 = \delta^T A \delta$ denotes the norm of δ with respect to the matrix A described in the previous subsection, used here as a Riemannian metric.[1] Linearizing (4) about ξ, we see that

$$L(\delta, \lambda) = f(\xi) + \delta^T \nabla f(\xi) + \frac{1}{2\alpha} \delta^T A \delta + \lambda^T [h(\xi) + H\delta]$$

where $\nabla f(\xi) = A\xi + b$ denotes the gradient of f evaluated at ξ, and H denotes the $k \times n$ Jacobian of h evaluated at ξ.

[1] This is the covariant aspect of CHOMP, the description of which is omitted here for brevity. See [2] for the motivation of the use of A as a metric.

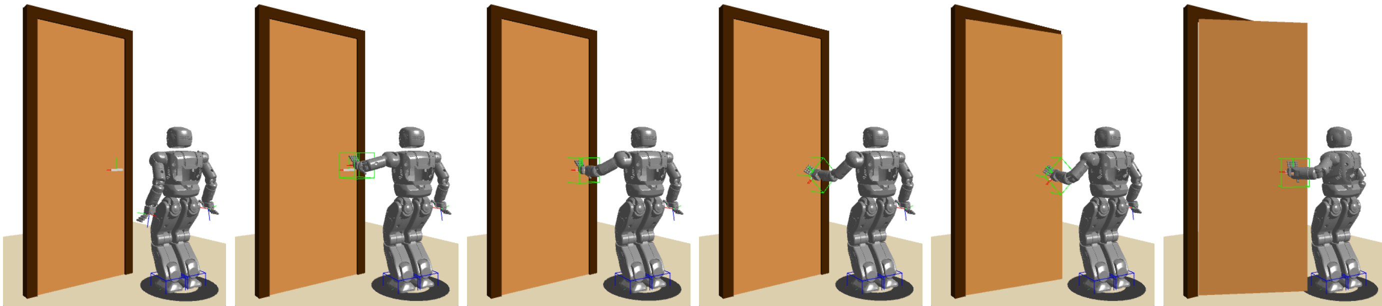

Fig. 2. Key poses for door opening task. Left to right: initial pose, pregrasp, grasp, handle turn, partially open, final.

The constrained optimization problem is solved by setting the gradient of the Lagrangian to zero:

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial \delta} \\ \frac{\partial L}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} \nabla f(\xi) + \frac{1}{\alpha} A\delta + H^T \lambda \\ h + H\delta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Rearranging terms, we can rewrite the above equation as the linear system

$$\begin{bmatrix} \frac{1}{\alpha} A & H^T \\ H & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(\xi) \\ -h(\xi) \end{bmatrix}$$

Using block-wise matrix inversion, we find that the δ which satisfies this is given by

$$\delta = \alpha A^{-1}(H^T Q H A^{-1} - I)\,\nabla f(\xi) - A^{-1} H^T Q\, h(\xi) \quad (5)$$

where $Q = (HA^{-1}H^T)^{-1}$. We need not solve for λ at all since we are not interested in the value of the Lagrange multiplier in this problem, only its effect on the solution δ. After solving (5) for δ, we update the solution ξ via the update rule $\xi \leftarrow \xi + \delta$ and repeat until the objective function value stops decreasing.

III. Implementation

The initial trajectory for the robot is formed by interpolating between a set of key poses using inverse kinematics (IK). The poses include (1) the initial pose of the robot, (2) a “pregrasp” position, which positions the hand near the door handle, (3) a “grasp” pose which places the hand grasping the handle, (4) a “handle turn” pose which turns the door handle 45° prior to opening, (5) a “partially open” pose which opens the door enough to begin releasing the handle, and (6) a “final” pose, which further opens the door and returns the handle to its original horizontal position. See Fig. 2 for an illustration. Our software lets us specify the parameters of the key poses via a configuration file. Future work will include a more advanced motion generation and validation scheme to choose the parameters automatically via some simple heuristics. Again, we hypothesize that since the environment is uncluttered and since doors have a regular and predictable structure, it is unnecessary to employ a complete motion planner for this task. Once the initial trajectory for the robot is generated, IK is no longer needed since the constrained CHOMP algorithm handles kinematic constraints via the manipulator Jacobians.

In our current implementation, we optimize over $m = 9$ degrees of freedom: six arm joints, the hip yaw joint, and

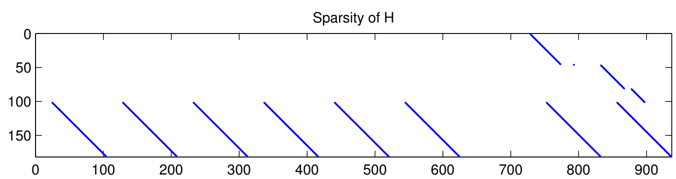

Fig. 3. Sparsity structure of the constraint Jacobian matrix H visualized with MATLAB’s `spy` function. Each row corresponds to a single equality constraint, and each column corresponds to a single degree of freedom at a single timestep within the trajectory. Rows 1-46 fix the hinge angle of the door, 47-101 fix the handle angle, and the remaining rows enforce the closed kinematic chain between the door handle and the end effector. The top 101 rows each have a single non-zero entry and correspond to a simple positional constraint. The bottom rows have multiple non-zero entries to reflect the coupling between degrees of freedom to handle the closed chain.

both the door hinge and door handle angles. Some of the constraints are simple positional constraints: for instance, the door hinge and handle must remain motionless before the robot has grasped the door, and the door handle must be fully turned during the phase of the trajectory when the door is initially being opened. Other constraints reflect the closed kinematic chain formed by the robot and the door during grasping. These constraints specify that the sum of squared differences between the end effector position and the target position on the door handle must be zero at each timestep when grasping is active.

We discretize the trajectory using a timestep of 40 ms. For the 4.2 seconds of door opening trajectory, this results in a trajectory of length $n = 104$. Over those 104 trajectory elements, there are $k = 181$ constraints active: 46 to fix the door hinge angle, 55 to fix the handle angle, and 80 to govern the closed chain kinematics between the end effector and the door. Although it it may seem odd that $k > n$, this is because multiple constraints are active at certain times during the trajectory. See Fig. 3 for an illustration of the sparsity structure of the constraint Jacobian H.

A. Numerical optimization

We implemented the constrained CHOMP algorithm using the linear algebra routines provided by the OpenCV [11] library. Since OpenCV has no native support for the skyline Cholesky decomposition of sparse banded matrices, a custom implementation was written to provide fast solution to equations of the form of (3).

Creating key poses and interpolating between them to create initial trajectory takes approximately 300 ms. Trajectory optimization takes approximately 40 s. Times were measured on a 2010 Mac Book Pro with a 2.5 GHz Intel Core 2 Duo processor. The bulk of the initial trajectory generation time is spent performing numerical IK using a Levenberg-Marquart solver. Our team is currently developing an analytical IK solution, which should speed IK computations greatly. The vast majority of optimization time is spent in computing the Q matrix, which requires solving a non-sparse linear system of size 181×181 (currently accomplished using OpenCV's Cholesky decomposition solver). Although the optimization performance is currently far from realtime, we expect it to improve in the future – see section IV for details.

B. Preliminary robotic experiments

Development on the HUBO+ platform is in its initial phases. At present, the robot control software plays back trajectories from the trajectory generation without feedback, and there is no perception at all. Instead, the position of the robot relative to the door is surveyed by hand before trajectory generation. Additionally, precise control of the robot's hands is not enabled in our current control software, leading us to design trajectories which could result in door opening using a fixed hand shape.

Despite the preliminary nature of the lab setup, we were able to successfully demonstrate door opening in the lab, as illustrated in Fig. 5. Because we are still determining the safety limits of the robot in terms of joint torques and velocities, the trajectory is run at 1/2 speed, resulting in approximately 8 seconds of motion.

IV. CONCLUSIONS AND FUTURE WORK

To the best of our knowledge, this work represents the first large-scale application of constrained CHOMP to closed kinematic chains. Earlier implementations of constrained CHOMP handled only adjustable trajectory endpoints [8] or end effector orientation constraints which entail many fewer degrees of freedom [9]. Although the runtime performance of our current system is about 10x real-time performance, we anticipate several ways to speed it up. Migrating to a different linear algebra library, especially one which explicitly uses parallelization (either on the CPU or GPU) could boost performance significantly. Additionally, we are investigating multi-grid approaches to CHOMP which exploit the fact that the solution should change minimally if it is downsampled in time. The multigrid approach to CHOMP would therefore involve recursively solving a coarser version of the problem, and upsampling in time before further refining the trajectory.

Our current software does not optimize for either self-collisions of the robot geometry, or collisions between the robot and the door. Although it is fairly easy to find collision free trajectories given the relatively uncluttered environments around doors, we would nonetheless like to include consideration of collisions into our approach, as this was one of the great strengths of the original CHOMP algorithm. To support this, the authors of the FCL library [12] have agreed to assist us by adding more support for fast computation of closest point pairs between arbitrary convex objects.

As mentioned in section III-B, we are currently surveying the relative transformation between the robot and the door by hand. We plan to eventually migrate to an onboard perception system to detect doors and handles, similar to the approach of [4]; however, as a stopgap measure in the mean time, we will use our lab's OptiTrack motion capture system to localize the robot and door, in order to test the system across a wider variety of initial poses.

The current door setup in the lab consists of a standard lightweight wooden door with an ADA-compliant handle. At present the door requires minimal force to open; however, in the future, we will likely add a spring and damper to the door, which will necessitate consideration of more complex dynamics during execution. Stilman, for example, extends the ZMP walking controller to handle external forces [13]. An alternative approach by Mistry et al. comes up with a principled approach for inverting the full rigid body dynamics of the system [14].

REFERENCES

[1] I. Park, J. Kim, J. Lee, and J. Oh, "Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot 3: HUBO)," in *Proc. IEEE-RAS Int'l Conf. on Humanoid Robots*. IEEE, 2005.

[2] N. Ratliff, M. Zucker, J. Bagnell, and S. Srinivasa, "CHOMP: Gradient Optimization Techniques for Efficient Motion Planning," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, May 2009.

[3] H. Arisumi, J. Chardonnet, and K. Yokoi, "Whole-body motion of a humanoid robot for passing through a door-opening a door by impulsive force," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*. IEEE, 2009.

[4] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. B. Rusu, B. Marthi, G. Bradski, K. Konolige, B. P. Gerkey, and E. Berger, "Autonomous door opening and plugging in with a personal robot," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2010.

[5] L. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration space," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[6] J. Kuffner and S. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, San Francisco, CA, Apr. 2000, pp. 995–1001.

[7] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. Bagnell, C. Atkeson, and J. Kuffner, "Optimization and learning for rough terrain legged locomotion," *International Journal of Robotics Research*, vol. 30, no. 2, February 2011.

[8] A. Dragan, N. Ratliff, and S. Srinivasa, "Manipulation planning with goal sets using constrained trajectory optimization," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, May 2011.

[9] M. Zucker, N. Ratliff, A. Dragan, M. Pivtoraiko, M. Klingensmith, C. Dellin, J. Bagnell, and S. Srinivasa, "CHOMP: Covariant hamiltonian optimization and motion planning," *International Journal of Robotics Research, submitted, under review*.

[10] A. George and J. Liu, *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Inc., 1981.

[11] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[12] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *Proc. IEEE Int'l Conf. on Robotics and Automation*. IEEE, 2012.

[13] M. Stilman, "Navigation among movable obstacles," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, October 2007.

[14] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *Proc. IEEE Int'l Conf. on Robotics and Automation*. IEEE, 2010.

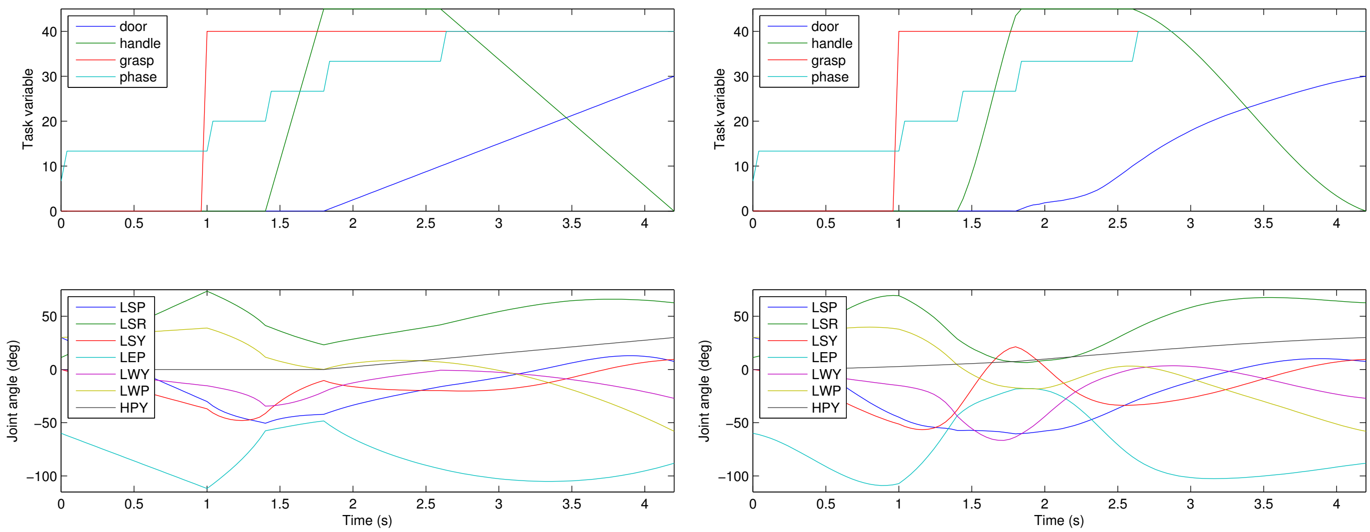

Fig. 4. Robot trajectories before (left) and after (right) optimization to minimize squared acceleration via CHOMP. Top plots indicate door hinge and handle angles, as well as indicator variables for the phase of the motion and whether the robot is grasping the door. Bottom plots show six arm joint angles, as well as the hip yaw. Although the trajectories are transformed significantly during optimization, kinematic constraints are maintained, resulting in good grasping while visibly smoothing trajectories.

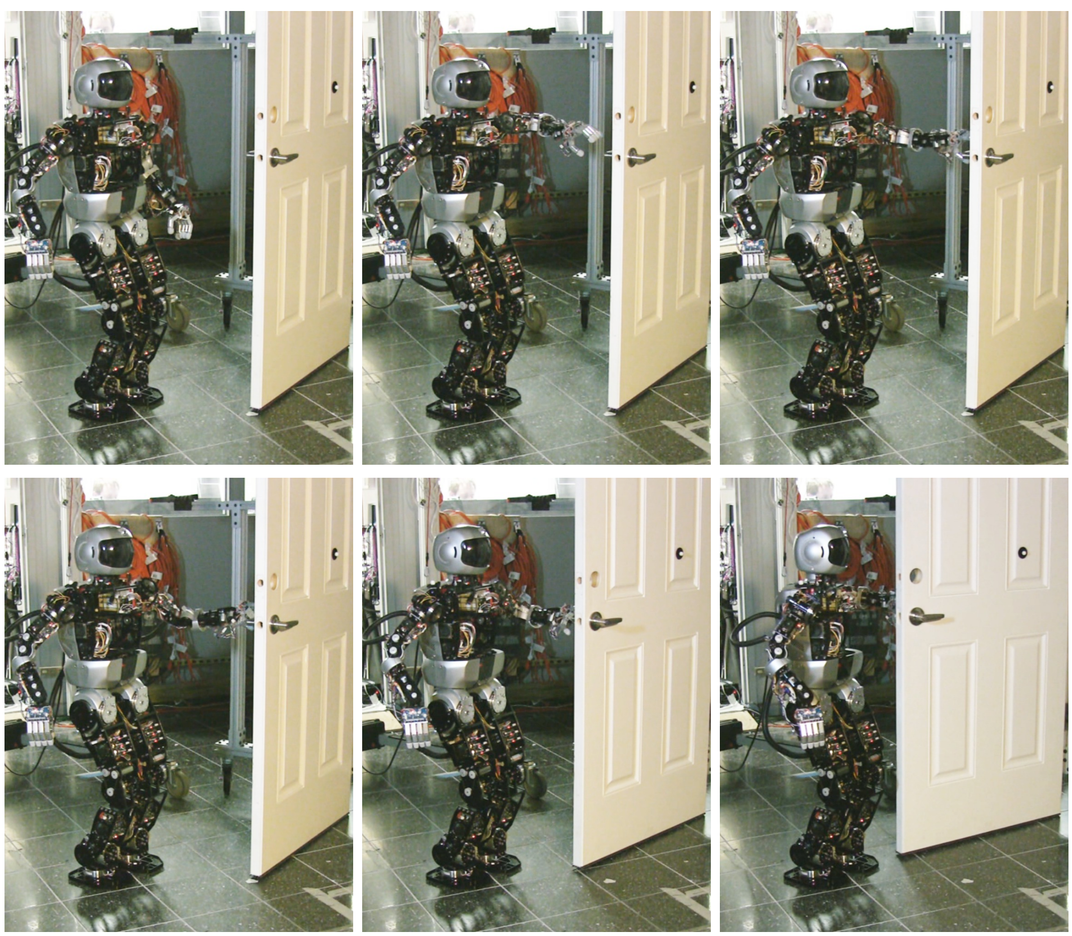

Fig. 5. Demonstration of successful door opening with the HUBO+ humanoid robot.