Human-Robot Collaborative Handling of Highly Deformable Materials

Daniel Kruse, Richard J. Radke, and John T. Wen
{krused2, wenj}@rpi.edu, rjradke@ecse.rpi.edu

***Abstract*— Robotic manipulation of highly deformable materials like cloth is a challenging problem due to the high dimensionality and the need for multiple grasped points to fully manipulate the material. Previous methods focused on developing detailed models for the material representation and its interaction with the robot in order to generate a motion plan. In this paper, we expand on our previous work in co-robotic manipulation by considering the collaborative transport of a deformable material by human and a *mobile* dual-arm robot. Our approach is to first determine optimal position and velocity setpoints from human pose information; the mobile dual-arm robot then attempts to follow these setpoints while satisfying other constraints. The goal is to stably and efficiently transport the material to a desired state. As the human changes his/her pose, the robot determines the relative optimal pose of its end effectors and uses a feedback control law to determine desired end effector velocities. The control signal to the actuators that meets the desired end effector velocities is then calculated according a constrained quadratic program. We evaluate the controller in simulation using the Bullet physics engine to approximate the cloth behavior.**

I. Introduction

As "human friendly robots" become more pervasive and affordable, the opportunity increases for collaboration between such robots and humans in manufacturing or even households. We are motivated by small-throughput manufacturing jobs from the composites industry, which are highly variable in scope and include the relatively complex task of collaboratively transporting a carbon fiber sheet to a workstation. A robot manipulating cloth typically has to calculate the path in advance of the motion, which is time consuming to compute, as was previously demonstrated with robotic laundry folding [1]. Since flexible materials are extremely high-dimensional, even in discrete approximations, planning algorithms are currently infeasible for online computation. However, in a collaborative task we wish to take advantage of having the human in the loop to do the high level path planning and decision making, while the robot reacts to the human motion and cloth state. For effective collaboration, the robot must move intuitively with the human while maintaining the material integrity.

Cloth manipulation has recently gained increased attention in different applications over the past 5 years [1]–[7]. Miller et al. demonstrated autonomous laundry folding in [1]. Alonso-Mora et al. demonstrated pre-planned cloth transportation with a team of mobile robots under formation control in [2]. Lee et al. put together a machine learning solution for path planning the motion of flexible sheets using industrial robot arms in [3]. Sun et al. demonstrated visual detection and straightening of cloth wrinkles with stereo cameras and a robot arm in [4]. In our previous work [7], we demonstrated an algorithm to control the force and cloth deformation on a stationary robot based on visual and force feedback. Phillips-Grafflin and Berenson have presented similar work on human-robot collaboration for cloth manipulation in simulation in [5], [6].

We expand on prior work [7] by solving for an optimal human-robot relative pose and incorporating this solution towards a multi-level feedback control algorithm for a dual-arm mobile manipulator. An offline static optimization determines the desired pose of the robot task space based on minimal cloth internal strain. A proportional control law based on the error between current and desired pose gives a task-space velocity vector for the dual-arm system. We then form a quadratic program with a weighted sum of costs to solve for the full body joint command signal. The program is designed to resolve redundancy and minimize error from the desired task velocity without violating a set of equality and inequality constraints. The algorithm allows for a simplified representation of the system state and a feasible computation for online control operation. We evaluate the controller in simulation using the Bullet physics engine [8] to approximate the cloth behavior.

II. Problem Formulation

A. System Description

We discretize the cloth into N nodes as the set $\mathbb{C}$ as shown in Figure 1. Each node $i \in \mathbb{C}$ has position p_i and is connected to a surrounding neighborhood of nodes $\mathcal{N}_i$. The set of all nodes $\mathbb{C}$ is partitioned into a subset containing free nodes $\mathbb{F}$ and a subset of grasped nodes $\mathbb{G} = \{\mathbb{G}_{R_l}, \mathbb{G}_{R_r}, \mathbb{G}_{H_l}, \mathbb{G}_{H_r}\}$. We assume that the grasped nodes are at the corners of the cloth and attached to either the robot's left or right end effectors $\mathbb{G}_{R_l}$, $\mathbb{G}_{R_r}$ or the left or right human hands $\mathbb{G}_{H_l}$, $\mathbb{G}_{H_r}$.

The grasped nodes $\mathbb{G}$ are rigidly connected to the body manipulating them, and form boundary conditions for the free nodes $\mathbb{F}$, which move deterministically based on external forces, such as gravity or drag, and resist internal forces from the inter-node connections.

Instead of treating the two hands as independent bodies, we consider them as joined by a virtual rod that has one degree of freedom (dof), allowing translation along the line between the hands. A rigid connection between the two hands reduces the 12-dof system to a 6-dof system, constraining hand orientations to be the same and hand positions a fixed distance apart. Adding the translational

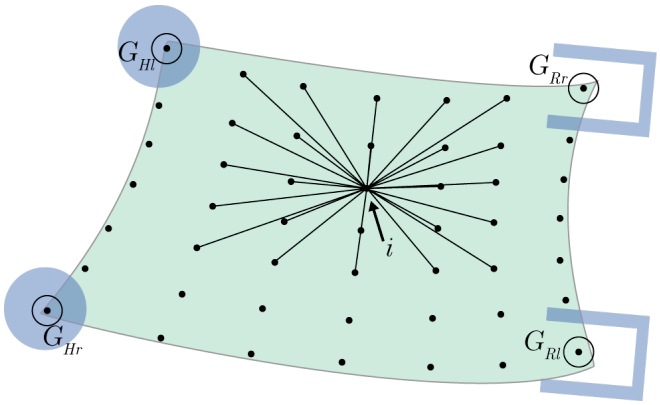


Fig. 1. The model representation for the discretized cloth being held at the left two corners by human hands, and the right two corners by robot grippers. Grasped nodes within $\mathbb{G}$ are circled while the remaining nodes are part of $\mathbb{F}$ and free-moving. Each node i is connected to its entire 2-neighborhood, $\mathcal{N}_i$.

degree of freedom between hands results in a 7-dof representation of the dual-arm task frame [7]. These additional constraints reduce the dimension of the optimization problem in Section II-C. Define the pose of the virtual-handed hands in terms of the position, orientation, and separation below (the subscript $X = H$ for human and $X = R$ for robot:

$$\begin{aligned} R_{OX_t} &= \left[\frac{p_{rl}\times\hat{z}}{\|p_{rl}\times\hat{z}\|} \quad \frac{p_{rl}}{\|p_{rl}\|} \quad \frac{(p_{rl}\times\hat{z})\times p_{rl}}{\|(p_{rl}\times\hat{z})\times p_{rl}\|}\right] \\ p_{OX_t} &= \left(p_{OX_l} + p_{OX_r}\right)/2 \\ w_X &= \|p_{OX_l} - p_{OX_r}\| \end{aligned} \quad . \tag{1}$$

where R_{OX_t} defines an orthonormal frame based on the line between the hands, $p_{rl} = p_{OX_l} - p_{OX_r}$, and the upward-pointing unit vector, $\hat{z}$, p_{OX_t} is the position of the frame (mid-point between the hands), and w_X is the distance between the hands. For computation, we use the unit quaternion q_{OX_t} representation for R_{OX_t}, : This task frame is expressed visually in Figure 2.

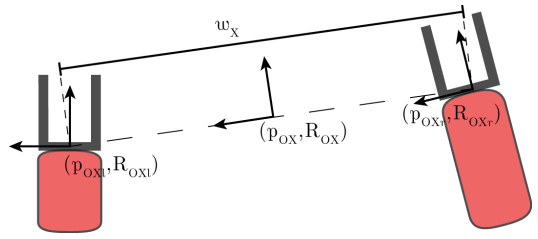


Fig. 2. The dual hand task representation $\psi_{OX} = [p_{OX}, \mathrm{q}_{OX}, w_X]^\top$ used for both the human and robot. A coordinate frame is placed directly between the two hands and oriented such that the $\hat{y}$ axis points from the right to left hand, and the $\hat{x}$ axis points outwards.

B. Robot Platform

The robot platform in use is the Baxter-On-Wheels (BOW) [9]. Shown in Figure 3, the robot consists of a Baxter robot mounted on a controllable wheelchair base. BOW has two 7-dof arms and a nonholonomic base which fits the common unicycle model:

$$\begin{bmatrix} \dot{x}_{OB} \\ \dot{y}_{OB} \\ \dot{\theta}_{OB} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{OB}) & 0 \\ \sin(\theta_{OB}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_B \\ \omega_B \end{bmatrix}. \tag{2}$$

Each arm has a gripper at the end with position and orientation $(\mathrm{q}_{OR_l}, p_{OR_l})$, $(p_{OR_r}, \mathrm{q}_{OR_r})$ for the left and right grippers respectively. There is also a Kinect RGBd sensor mounted on the 'head' for a downward facing view of the cloth and hand positions. We use this robot due to its relatively low cost as well as its inherently safe design that complies to external disturbance. This compliance makes it an excellent candidate for interaction with humans.

Fig. 3. The Baxter-On-Wheels (BOW). The robot has two 7-degree of freedom arms and a nonholonomic wheelchair base. An RGB-d sensor is mounted on the head for a downward-facing view of the cloth.

C. Task-Level Optimization

The human hands define the node positions for the nodes in $\mathbb{G}_{H_l}$, $\mathbb{G}_{H_r}$, while the robot can directly manipulate the cloth nodes within $\mathbb{G}_{R_l}$, $\mathbb{G}_{R_r}$ such that the free nodes settle into more favorable positions. A stretched out or deformed fabric could potentially damage the material, so we define a cost function proportional to the total spring energy penalizing deformations from its resting state:

$$V(p) = \sum_{i\in\mathbb{C}} \sum_{j\in\mathcal{N}_i} \left(\|p_i - p_j\| - \ell_{ij}\right)^2, \tag{3}$$

where p parameterizes the state of the fabric as a stacked vector containing the positions of all nodes in $\mathbb{C}$ and ℓ_{ij} is the resting edge length between nodes i and j. Given static human and robot hand/end-effector poses, (ψ_{OH}, ψ_{OR}), p may be determined numerically using a soft body simulator such as the Bullet Soft Body Physics Engine [8]. Examples of a rectangular cloth in different configurations artificially colored according to $V(p)$ are shown in Figure 4. Intuitively, large displacements between the grasping bodies generate larger inter-node displacements and accordingly have larger costs.

Since the grasped nodes define the positions of the free nodes, the position-level optimization problem therefore is to choose the robot task state ψ_{OR} defined in (1) that minimizes $V(p)$ based on the human task state ψ_{OH}:

$$\begin{aligned} \min_{\psi_{OR}} \quad & V(p(\psi_{OR}, \psi_{OH})) \\ \text{s.t.} \quad & h_I(\psi_{OR}, \psi_{OH}) < 0 \end{aligned} \tag{4}$$

where $h_I(\psi_{OR}, \psi_{OH})$ defines a set of inequality constraints to prevent collisions between the grasped nodes. This position level optimization provides a desired robot pose, ψ^d_{OR}, for a given hand pose, ψ_{OH}. Note that the optimal $V(p)$ may not be zero due to the presence of gravity and the pose of the human hands.

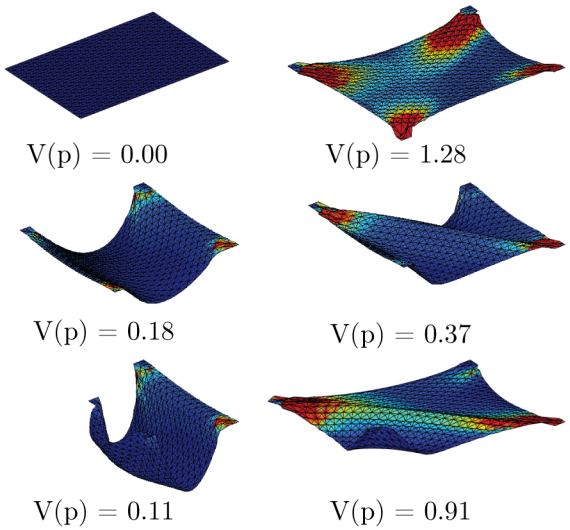


Fig. 4. Examples of cloth model in different configurations, colored by individual node costs. Note that V(p) = 0 is unattainable due to gravity.

III. Robot Control

From our initial task-level optimization and measurement of ψ_{OH}, we thus have a ψ^d_{OR} that we wish to achieve at every time step. We use a feedback law to map the error between the current ψ_{OR} and desired ψ^d_{OR} to a task-level velocity vector and then perform a joint-level optimization to meet this desired task space velocity while meeting a set of equality constraints to maintain relative pose and inequality constraints to avoid undesirable configurations.

A. Task Controller

At each time instant, we extract the human hand positions using the RGBd sensor mounted on the robot head. Given these positions, we estimate the human task space ψ_{OH} using (1). From the off-line optimization of (4) we have ψ^d_{OR}, which is the desired task state determined from the measured human task space. We use a proportional feedback law to get the velocity terms

$$\dot{\psi}^d_R = \begin{bmatrix} \omega^d_R \\ v^d_R \\ \dot{w}^d_R \end{bmatrix} = -K_t \begin{bmatrix} e_{\mathrm{q},0} e_{\mathrm{q},v} \\ p_{OR_t} - p^d_{OR_t} \\ w_R - w^d_R \end{bmatrix} \tag{5}$$

where K_t is a positive gain matrix and $e_{\mathrm{q},0}$, $e_{\mathrm{q},v}$ are the scalar and vector components of the error quaternion $e_{\mathrm{q}} = \mathrm{q}_{OR_t} \otimes \bar{\mathrm{q}}^d_{OR_t}$. Here $\otimes$ signifies the quaternion product and $\bar{\mathrm{q}}$ signifies the quaternion complement.

The robot hand positions are related to the task position through rigid translations:

$$p_{OR_l} = p_{OR_t} + \frac{w_R}{2} R_{OR_t} \hat{y}, \quad p_{OR_r} = p_{OR_t} - \frac{w_R}{2} R_{OR_t} \hat{y}.$$

Taking the time derivative of this relationship, we obtain the desired individual end effector velocities in terms of the desired task velocity:

$$\begin{bmatrix} v^d_{R_l} \\ v^d_{R_r} \end{bmatrix} = \begin{bmatrix} -\frac{w_R}{2} (R_{OR_t} \hat{y})^\times & I & \frac{1}{2} R_{OR_t} \hat{y} \\ \frac{w_R}{2} (R_{OR_t} \hat{y})^\times & I & -\frac{1}{2} R_{OR_t} \hat{y} \end{bmatrix} \dot{\psi}^d_R \tag{6}$$

where $(\cdot)^\times$ is the cross product matrix and $\hat{y} = [0, 1, 0]^\top$.

Additionally we have to maintain the equality constraints for the end effector orientations:

$$\mathrm{q}_{OR_l} = \mathrm{q}_{OR_t} \otimes \mathrm{q}_{R_t R_l}, \quad \mathrm{q}_{OR_r} = \mathrm{q}_{OR_t} \otimes \mathrm{q}_{R_t R_r},$$

where the quaternions $\mathrm{q}_{R_t R_l}$, $\mathrm{q}_{R_t R_r}$ are constants determined at initialization. The feedback control law used to maintain these constraints is

$$\begin{bmatrix} \omega^d_{R_l} \\ \omega^d_{R_r} \end{bmatrix} = -K_E \begin{bmatrix} e_{\mathrm{q}_l,0} e_{\mathrm{q}_l,v} \\ e_{\mathrm{q}_r,0} e_{\mathrm{q}_r,v} \end{bmatrix} \tag{7}$$

where K_E is a constant gain matrix and

$$e_{\mathrm{q}_l} = \mathrm{q}_{OR_l} \otimes \bar{\mathrm{q}}_{R_t R_l} \otimes \bar{\mathrm{q}}^d_{OR_t}, \quad e_{\mathrm{q}_r} = \mathrm{q}_{OR_r} \otimes \bar{\mathrm{q}}_{R_t R_r} \otimes \bar{\mathrm{q}}^d_{OR_t}$$

B. Full Body Controller

The BOW has two 7-dof arms with joint positions q_L, q_R and a 3-dof nonholonomic mobile base with state $q_B = [x_B, y_B, \theta_B]^\top$. The control signal $u = [u_L, u_R, u_B]^T$ maps to the state velocity as $\dot{q}_L = u_L$, $\dot{q}_R = u_R$ and $\dot{q}_B$ relates to u_B according to the unicycle model (2).

The task is to solve for u_L, u_R, u_B such that the end effector velocities from (6) are achieved subject to the constraints expressed in (7). The forward kinematic Jacobians relate the control signal to the end effector spatial velocities

$$\begin{bmatrix} \omega_{R_l} \\ v_{R_l} \\ \omega_{R_r} \\ v_{R_r} \end{bmatrix} = \begin{bmatrix} J_{\omega_{R_l}}(q) \\ J_{v_{R_l}}(q) \\ J_{\omega_{R_r}}(q) \\ J_{v_{R_r}}(q) \end{bmatrix} u. \tag{8}$$

In order to achieve $\dot{\psi}^d_R$ we formulated a variation of the full body velocity controller described in [10] for dual-arm mobile robots. The control minimization was originally presented for single arm mobile robots, and intuitively intended to represent a weighted combination of costs for

- deviation from the desired task velocity vector,
- base motion command $u_B = Cu$,
- internal motion (null space of J_T, J_E and C).

As expressed in [10], we first constrain the control signal u based on our equality constraints (7) as v_E

$$u = J_E^+ v_E + J_E^\perp \xi \tag{9}$$

where J_E is the stacked matrix of $J_{\omega_{R_l}}$ and $J_{\omega_{R_r}}$, and ξ parameterizes the kinematic redundancy of the robotic system. The redundancy may be resolved by solving the quadratic program below:

$$\begin{aligned} \min_{\xi} \quad & \|J_T(J_E^+ v_E + J_E^\perp \xi) - v_T^d\|^2 + \\ & \lambda_1 \left\|J_{TEC}^\perp (J_E^+ v_E + J_E^\perp \xi)\right\|^2 + \\ & \lambda_2 \|C(J_E^+ v_E + J_E^\perp \xi)\|^2, \end{aligned} \tag{10}$$

where J_T is the stacked matrix of $J_{v_{R_l}}$, $J_{v_{R_r}}$, and v_T^d are the vector terms from (6) and $J_{TEC}^\perp$ is a basis of the null space of the stacked matrix for J_T, J_E, and C. The quadratic problem is subject to the inequality constraints

$$\frac{\partial h_i(q)}{\partial u}(J_E^+ v_E + J_E^\perp \xi) \geq \sigma_i, \forall i = 1, \ldots, m \tag{11}$$

where each σ_i is a scalar lower bound and $h_i(q) > 0$ is an inequality constraint such as a joint boundaries $q > q_{\text{min}}$, $q_{\text{max}} > q$ or a self-collision prevention. The block diagram for this entire control algorithm is shown in Figure 5. Note that (10)–(11) is convex (since the weighting matrix in the quadratic term in (10) is positive definite), therefore will always yield a unique solution. However, this will only provide an incremental motion direction and may not be able to achieve the desired pose from (4), especially if the motion direction conflicts with the base nonholonomic constraint.

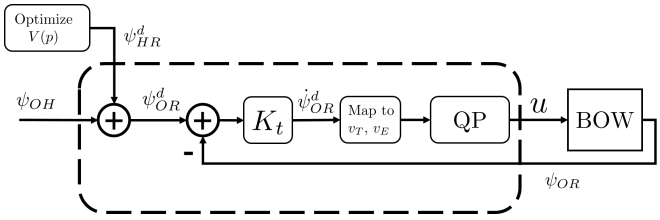


Fig. 5. Controller diagram for the cooperative cloth manipulation algorithm. On-line computations are shown in the dashed rectangle.

IV. Results

A. Simulation Environment

Simulation of the cloth was conducted using the open-source Bullet Soft Body Physics Engine [8]. Bullet uses position-based dynamics [11] to rapidly compute physical behaviors based on position-level constraints. For soft bodies, Bullet steps forward according to external forces (gravity, lift, drag, etc.) and then iteratively adjusts the node positions to satisfy the soft constraint that, for connected nodes i and j,

$$C(p_i, p_j) = \|p_i - p_j\|^2 - \ell_{ij}^2 > 0.$$

This acts as an approximation of the internal spring forces, with the physical stiffness closely tied to a scalar multiple of $C(p_i, p_j)$ and the number of iterations performed.

In simulation, we discretized a $0.6m \times 0.4m$ piece of rectangular cloth into 31×31 nodes that are each connected to their individual 2-neighborhood. The robot and human motion simulation is calculated in MATLAB with calls to Bullet to simulate the cloth dynamics. After numerically solving for (4) using `fmincon` in MATLAB, we found the optimal cloth state as shown in Figure 6.

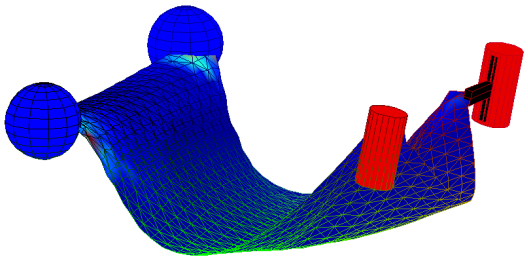


Fig. 6. Numerically determined static optimum of the 7-dof dual-arm pose (robot grippers shown as red cylinders) for a given hand grip (blue spheres). The optimal cost for $V(p) = 0.108$.

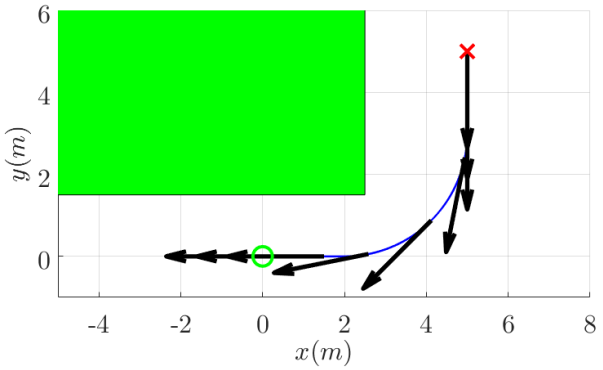
B. Simulation Results

We evaluated our control algorithm on several human paths with varying degrees of difficulty for different applications. Specifically, we considered a path with the human directing the robot to turn a corner and a 'forward park' path where the human and robot end on either side of an obstacle. These paths were chosen since they combine atomic human motions of straight line motion, moving with a nonzero radius, and rotating about a point. In the 'backwards turn' path shown in 7a, the human both backs up in a straight line motion and backs up with a nonzero radius of curvature. In the 'forward park' path shown in 7b the human both moves towards the robot in a straight line and rotates about the robot. Their dynamics were chosen based on observed motions in human-human collaborative fabric transport.

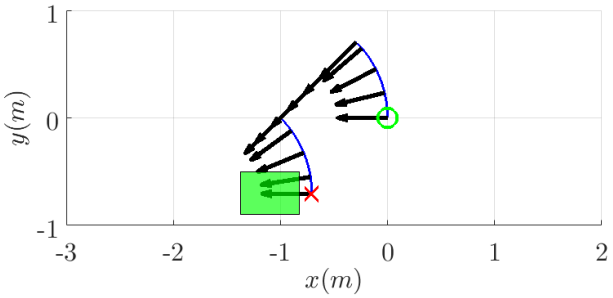
For the backwards turn path, the human first backs up at $.25m/s$ until $t = 8s$, turns around the corner with angular velocity $11.25°/s$ and linear velocity $0.6m/s$ until $t = 16s$, then resumes backing up at $.25m/s$ until $t = 24s$ where it finally pauses until $t = 26s$.

Selected times during the simulation are shown in Figure 8. At $t = 5s$, the robot is able to match the human speed mostly with base motion and the arms stay static during the straight path. During the turn, at $t = 12s$ the arms become more engaged to maintain the desired cloth state since the nonholonomic base cannot slide laterally. Due to a conflict with the joint boundary constraint, the left end effector is out of position. At $t = 18s$, the human is moving in a straight path again and the robot arms are settling back into the relaxed state. The evaluation of $V(p)$ (3) over time is shown in Figure 9. There is a transient spike up to $V(p) = 0.3$ from cloth dynamics during the initial step to the desired end effector pose that settles to equilibrium after 2 time steps. The higher velocity of the human during the turn between $t = 8s$ and $t = 16s$ naturally causes an increase in cloth deformation, but when the human slows back down again at $t = 16s$ the robot base catches up again to the robot, settling back into the lower energy state.

For the forward park path, the human first rotates at $4.5°/s$ with linear velocity $0.08m/s$ until $t = 10s$, then at $t = 12s$ moves forward at $0.07m/s$ until $t = 26s$, and finally at

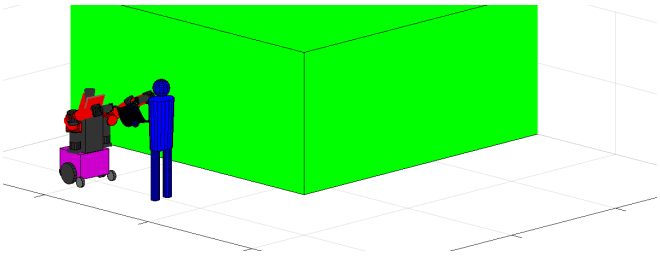


(a) Human "backwards corner turn" path. The human backs up and turns backwards around a corner and then continues to back up until it reaches the red X.

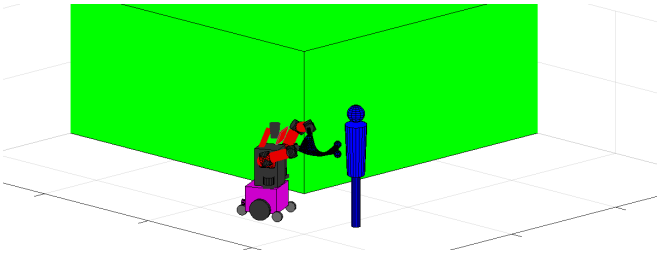


(b) Human "forward parking path". The human moves to the right while facing the robot, moves forward to push the robot behind the workpiece (in green), and then moves back to the left to the red X.

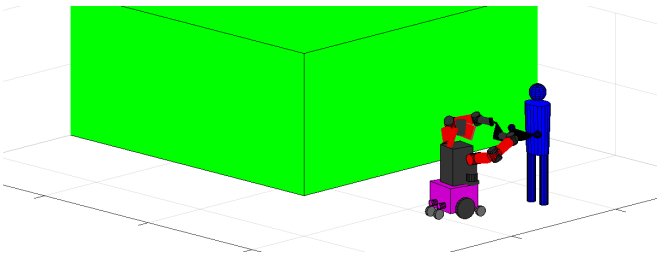
Fig. 7. We evaluate the algorithm using the above human paths. A green circle shows the starting point of the human, and a red X shows the end point. The arrows show the human's orientation at those particular times.



(a) $t = 5s$



(b) $t = 12s$



(c) $t = 18s$

Fig. 8. Selected times in the 'backwards turn' path.

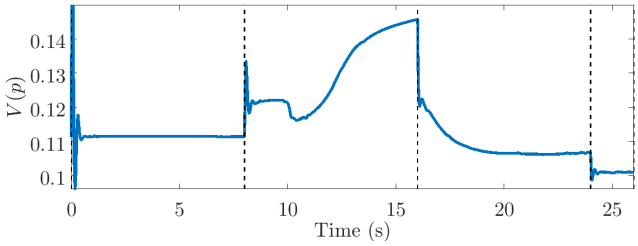
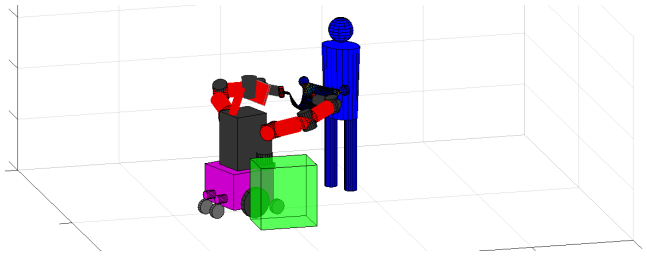


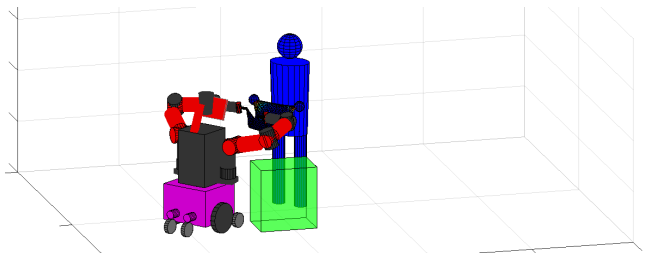
Fig. 9. Cost vs. time for 'backwards turn' human path. Note that the initial time step shows a spike that rises off the displayed axis to $V(p) = 0.3$.

$t = 28s$ rotates in the opposite direction at $4.5°/s$ until $t = 38s$. The goal of this path is to reposition the robot from the side of the workstation (shown in green in the figures) to the back of it. During the initial rotation, the robot uses the arms to rotate until reaching a joint boundary constraint and then engages the base with the arms essentially locked. The time instant in Figure 10a shows the relaxed arm pose during this period. While the robot is backing up, as shown in Figure 10b, it settles into a similar arm pose once the base motion dominates the control signal. The final rotation, shown in Figure 10c also attempts to solve the motion using arm motion until the human rotates past the joint boundary for the right shoulder. The evaluation of $V(p)$ (3) over time is plotted in Figure 11, showing that the control law adequately maintained a low cost over the entire path. Again, there is a transient spike up to $V(p) = 0.3$ as in the corner turning path from cloth dynamics during the initial step settles to equilibrium after 2 time steps.

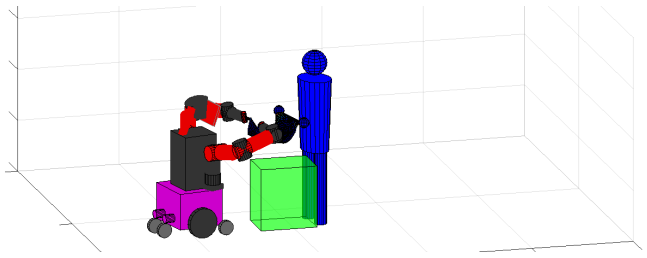
For both paths, the robot was able to follow the human with reasonable results; however, we found that turning or rotating motions can potentially make the system fail if the radius of curvature is too small. In this case, the nonholonomic base ends up in a contorted position and ultimately fails. In Figure 12 we show key time steps for when the human performs an antagonistic motion, i.e. rotating in place at $11.25°/s$. The robot initially tries to handle the motion using its arms and then by zig-zagging in the no-slip direction, but ultimately fails. Since the human is tasked with planning the motion and knows that the robot base is nonholonomic, this is obviously a poor decision. The failed performance suggests the use of a full motion planner or a predictive algorithm for difficult trajectories. An omni-directional base would also resolve the issue and simplifies the control problem. As the human motion strongly affects the robot motion, a human reaction model may be incorporated to study more realistic jamming scenarios.



(a) $t = 8s$



(b) $t = 22s$



(c) $t = 35s$

Fig. 10. Selected times in the 'forward park' path.

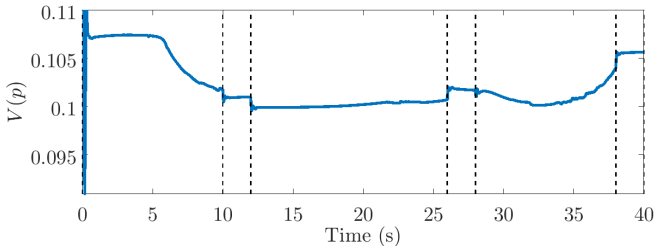
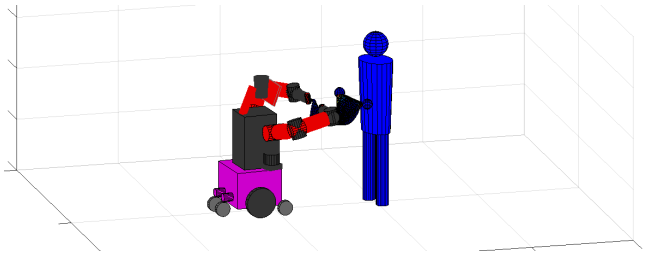


Fig. 11. Cost vs. time for 'forward park' human path. Note that the initial time step shows a spike that rises off the displayed axis to $V(p) = 0.3$.

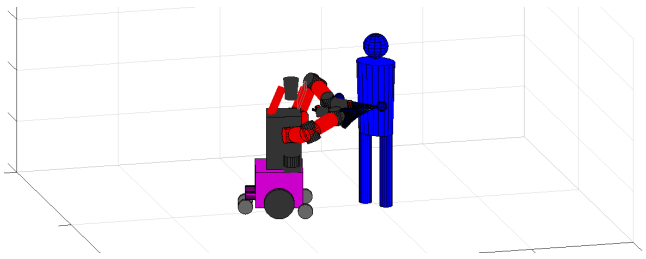
V. Conclusions and Discussion

We presented initial results of a mobile manipulator cooperatively transporting cloth with a human. The controller demonstrated satisfactory performance under simulated human motions within bounds of angular or linear velocity. In future work we intend to apply the control system on our physical robot, which we have already started to implement[1], and incorporate additional feedback information such as force and 3D cloth information, as well as feedforward for better human motion tracking and path space based predictive control to better address the nonholonomic base which could cause contorted arm poses.

[1] `http://www.youtube.com/watch?v=j7Q8KtIbGR0`



(a) $t = 1.5s$



(b) $t = 3.5s$

Fig. 12. Selected times in the antagonistic case of the human rotating in place.

Acknowledgments

This work is supported primarily by the Center for Automation Technologies and Systems (CATS) under a block grant from the New York State Empire State Development Division of Science, Technology and Innovation (NYSTAR) under contract C130145.

References

[1] S. Miller, J. van den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel. A geometric approach to robotic laundry folding. *International Journal of Robotics Research*, 31(2):249–267, December 2011.

[2] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus. Local motion planning for collaborative multi-robot manipulation of deformable objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

[3] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel. Learning force-based manipulation of deformable objects from multiple demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

[4] L. Sun, G. Aragon-Camarasa, S. Rogers, and J. P. Siebert. Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

[5] Calder Phillips-Grafflin and Dmitry Berenson. A representation of deformable objects for motion planning with no physical simulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014.

[6] Dmitry Berenson. Manipulation of deformable objects without modeling and simulating deformation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013.

[7] D. Kruse, R. J. Radke, and J. T. Wen. Collaborative human-robot manipulation of highly deformable materials. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

[8] E. Coumans. Bullet physics library. `bulletphysics.org`, 2016.

[9] Lu Lu and J.T. Wen. Baxter-On-Wheels (BOW): An assistive mobile manipulator for mobility impaired individuals. In Yue Wang and Fumin Zhang, editors, *Trends in Control and Decision-Making for HumanRobot Collaboration Systems*, chapter 3, pages 41–63. Springer-Verlag, London, U.K., 2017.

[10] Lu Lu and J. T. Wen. Human-directed coordinated control of assistive mobile manipulator. *International Journal of Intelligent Robotics and Applications*, 2016. `doi:10.1007/s41315-016-0005-3`.

[11] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, April 2007.