Dynamic Model Learning and Manipulation Planning for Objects in Hospitals using a Patient Assistant Mobile (PAM) Robot

Roya Sabbagh Novin[1] Amir Yazdani[1] Tucker Hermans[2] Andrew Merryweather[1]

Abstract— **One of the most concerning and costly problems in hospitals is patients falls. We address this problem by introducing PAM, a patient assistant mobile robot, that maneuvers mobility aids to assist with fall prevention. Common objects found inside hospitals include objects with legs (i.e. walkers, tables, chairs, equipment stands). For a mobile robot operating in such environments, safely maneuvering these objects without collision is essential. Since providing the robot with dynamic models of all possible legged objects that may exist in such environments is not feasible, autonomous learning of an approximate dynamic model for these objects would significantly improve manipulation planning. We describe a probabilistic method to do this by fitting pre-categorized object models learned from minimal force and motion interactions with an object. In addition, we account for multiple manipulation strategies, which requires a hybrid control system comprised of discrete grasps on legs and continuous applied forces. To do this, we use a simple one-wheel point-mass model. A hybrid MPC-based manipulation planning algorithm was developed to compensate for modeling errors. While the proposed algorithm applies to a broad range of legged objects, we only show results for the case of a 2-wheel, 4-legged walker in this paper. Simulation and experimental tests show that the obtained dynamic model is sufficiently accurate for safe and collision-free manipulation. When combined with the proposed manipulation planning algorithm, the robot can successfully move the object to a desired position without collision.**

I. INTRODUCTION

Each year about 35% of individuals over age 65 experience falls providing the third leading cause of chronic disability worldwide [1]. In hospitals, patient falls occur most frequently at the bedside, and reports show that the rate of bedside falls comprise as high as 50% of all falls [2], [3]. A few fall preventative and protective strategies such as patient sitters, bed alarms and floor mats are currently being used, but none of these has solved an underlying problem; patients have little to no support while ambulating to and from the bedside to the bathroom [4], [5]. In hospitals, even when an alarm activates, it may take minutes for a nurse to respond and falls often happen during this response time period.

Assistive robots offer the potential to provide continuous care and monitoring in healthcare environments. However, most current robots like Giraff [6] and HOBBIT [7], only observe and communicate with patients without any physical engagement to prevent falls. We believe an autonomous patient assistant mobile (PAM) robot with object manipulation capabilities will help prevent patient falls by intervening with a mobility aid at the bedside prior to exit. Using monitoring data, the PAM robot can provide assistance to a patient, or clear the patient's path by moving obstacles away (Fig. 1).

Fig. 1: PAM robot manipulating a walker in a common room space. PAM is capable of both pushing and pulling objects which is helpful in small spaces and corners.

The problem of human-aware autonomous mobile robot navigation in cluttered and dynamic environments has been widely investigated [8], [9], [10]. However, many challenges still exist in manipulating objects while navigating through cluttered and unstructured environments such as hospitals or personal dwellings [11]. This includes estimating the dynamics of unknown objects, creating safe and collision-free maneuvering trajectories and dealing with discrete and continuous, i.e. hybrid actions. Medical environments are usually cluttered by various objects, including mobility aids, carts, chairs and tables. Since the dynamics of these objects are not necessarily known to the robot, the robot must be able to autonomously determine the object's dynamic model for a successful manipulation. Additionally, when manipulating legged objects, the robot must select not only a direction and magnitude of the pushing or pulling force, but must also make a discrete choice about which leg to push or pull. Thus, in this paper, we investigate the problem of manipulating an unknown legged object (2-wheeled walker), to a desired final position using our PAM robot [12] (Fig. 2).

Previous research concerning dynamic models mostly describe either mapping between actions and consequences for a specific task [13], [14], or rely on pure kinematics [15]. A more advanced approach mimics human sensorimotor

[1]Roya Sabbagh Novin, Amir Yazdani and Andrew Merryweather are with the Department of Mechanical Engineering, University of Utah, Utah, US roya.sabbaghnovin@utah.edu, amir.yazdani@utah.edu, a.merryweather@utah.edu
[2]Tucker Hermans is with the School of Computing and Robotics Center, University of Utah, Utah, USA thermans@cs.utah.edu

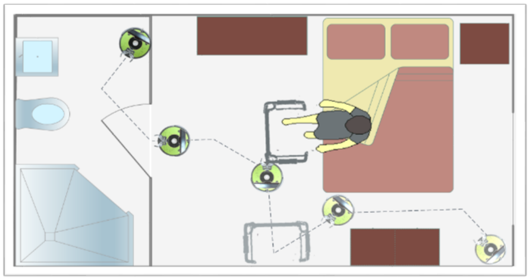

Fig. 2: Whenever needed, PAM can retrieve the walker and bring it to the patient to help with stability to prevent a fall.

learning behavior, in which a coarse dynamic model of the new object is learned based upon prior beliefs and experiences. Eventually, the coarse model is improved as more data are collected during the manipulation [16], [17]. In this research we chose a Bayesian regression model in order to incorporate knowledge about common legged furniture as priors to inform the dynamics learning algorithm [17].

For autonomous manipulation we use a combination of model predictive control (MPC) and mixed-integer convex optimization to overcome the imperfections of the dynamic model and avoid getting stuck in local minima. To make it convex, we linearize the dynamic model over a nominal trajectory for the object. We perform a hybrid control by penalizing actions that require changing legs based on the path between legs and costs associated with regrasping.

Below, we summarize the main contributions of this paper:
1. Patient Assistant Mobile (PAM) Robot: Development of a mobile robot with a 3-Finger gripper that can manipulate legged objects by pushing or pulling.
2. Unknown Object Model Learning: Use of a BRM-based method adopted from [18] to learn dynamic parameters of a walker using experimental data.
3. Hybrid Manipulation Planner: Development of a manipulation planner based on receding horizon concept and convex optimization with discrete actions of changing legs as well as continuous motion in manipulation.

II. RELATED WORK

Robust and safe autonomous object manipulation remains a challenging research topic [19]. Most of the existing approaches either require a large training dataset [14], or use kinematics-based methods for a specific task [15].

Early studies on an object's dynamic behavior suggest learning a map between actions and the resulting effects to inform future goal-directed behavior [13], [14]. Ogata proposed an active-sensing method based on recurrent neural networks to find mapping for all sequences of motion [13]. These methods do not obtain physics-based dynamics and cannot be used for other types of manipulation other than what is performed in the training process. Stilman et al. used pseudo-inverse of dynamics equations to obtain the dynamic parameters of large objects in [20]. However, they could not find a consistent relationship between acceleration and force and only used a viscous friction model and ignored mass and inertia effects.

Scholz et al. propose using physics-based reinforcement learning as an adaptive method to obtain non-linear dynamic model estimates for non-holonomic objects[16]. Later, they use this method to estimate the physical parameters of an office table and a utility cart with fixed front wheels [18].

There is also an ongoing effort to find planning frameworks that can effectively handle the uncertainty and hybridness associated with planning for both pushing and pulling actions. [21] first formulated the mechanics of planar pushing manipulation tasks. [22] created a forward empirical model of an unknown object for pushing using visual feedback. [23] focused on finding appropriate pushing actions and developing a push planner which can track a predefined trajectory using these actions based on a set of assumptions and a simplified model of two-agent point-contact push.

[24] used a model predictive path integral controller to plan push manipulations based on a learned model including uncertainties, obtained by Gaussian process regression and an ensemble of mixture density networks. Hermans et al. presented a data-driven approach for learning good contact locations for pushing unknown objects [25].

[26] addresses the problem of motion planning for non-holonomic cooperating mobile robots manipulating and transporting objects while holding them in a stable grasp. They use the calculus of variations (with high computational cost) to obtain optimal trajectories and actuator forces and torques for object manipulation in the presence of obstacles.

A few model-based hybrid manipulation controllers have been introduced [27], [28], [29]. The control strategies presented in the aforementioned papers are applied to systems with a priori knowledge of the contact mode sequencing or offline determination of optimal mode sequences. In [28] MPC is used to find an optimal sequence of robot motions to achieve a desired object motion. [30] solves the problem of finding an optimal sequence of hybrid controls under uncertainty using differential dynamic programming.

III. APPROACH

In this section, we first describe the mobile robot developed for our application of grasping legged furniture and mobility aids. We, then, discuss the learning algorithm used to find a coarse object dynamic model. Finally, we present our manipulation planner.

A. Robot Design

For the experimental studies, a low-cost 3-finger gripper was designed, developed and mounted on an iRobot Create2 (Fig. 3). We chose iRobot Create2 because of its size and cost. It is also a familiar mobile robot to many people, which may improve its acceptance.

The gripper design is adopted from [31] and extended to a 3-finger hand for stable grasping of legged objects with different sized legs. It has been designed for only enveloping grasp which is less complex, tolerates more external disturbances and fits the required application. The main dimensions of the gripper are described in Table I.

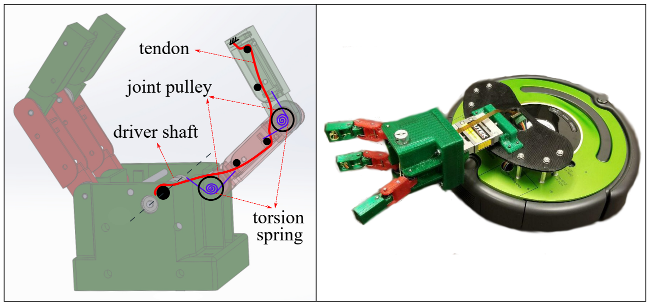

Fig. 3: PAM robot is equipped with a 3-finger gripper for stable grasping of legged objects with different leg diameters. A flexor tendon is wrapped around the driver shaft and passes through routing points to close the fingers. Torsion springs on joints passively extend the fingers.

These dimensions are used for grasping cylindrical objects with diameters from 30 mm to 50 mm.

Each finger of the gripper has two segments, distal and proximal, which are connected with revolute joints. A combination of MAXON EC-20 flat DC brushless motors and a 5.4:1 planetary gearhead provides actuation for flexing all three fingers using a tendon-based mechanism. A 0.4mm diameter Kevlar string is used as a flexor tendon. The tendons pass over pins and pulleys, defining the location of the force applied on the finger links. During flexion, if the proximal link contacts the object, the distal link will continue to flex around the object to complete an enveloping grasp. The fingers can extend passively by torsion springs on each joint.

We attach a one directional force sensor with 220N capacity to the gripper for planar force measurements. The robot is powered by the iRobot Create2 battery and controlled by a Raspberry Pi 3 connected wirelessly to a base station.

B. Modeling Method

We use Bayesian Regression Model to fit a point mass on a wheel model to define object dynamics (Fig. 4) adopted from [16]. The final model will be a probabilistic estimate of the dynamic parameters of the object. We only consider planar parameters since all the objects will be manipulated while sliding or rolling over the floor.

Dynamic parameters for a planar model include inertia and friction; (1) inertia requires four parameters for planar manipulation: one for mass, two for the position of the center of mass, and one for inertia in the XY plane. (2) Two friction coefficients in the X and Y directions define the anisotropic friction. The model parameter vector is:

$$\boldsymbol{\Pi} := < \boldsymbol{m}, \boldsymbol{I}, \boldsymbol{x_c}, \boldsymbol{y_c}, \boldsymbol{\mu_x}, \boldsymbol{\mu_y} > \tag{1}$$

In the framework of the Bayesian approach, unknown parameters of the model are regarded as random variables from a prior probability distribution. Then, given the observation, the conditional probability of possible values of the unknown parameters (posterior distribution) is obtained. Since the posterior distribution cannot be reasonably obtained by direct computation, we use a Markov Chain Monte Carlo (MCMC) method to sample from the distribution [32].

Dimension	Palm	Proximal Link	Distal Link	Shaft	Pulleys
Width	90	24	20		
Length	95	80	67		
Radius				4	13

TABLE I: Main dimensions of the gripper fingers in mm.

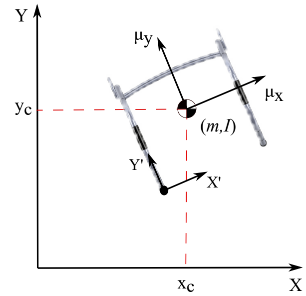

Fig. 4: Walker is modeled as a "point mass on a wheel" and its parameters are learned using Bayesian Regression Model.

Given the 2D state vector of the object $\mathbf{x}$ containing position, orientation, linear and angular velocities, and applied force vector $\mathbf{u}$, the dynamics equation can be written as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{2}$$

where $\mathbf{x} = [x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}]$ and $\mathbf{u} = [f_x, f_y, \tau]$. Reforming the second order differential in Eq. 2 as a discrete time integration over time steps δt_i results in:

$$\mathbf{x}_{i+1} = g(\mathbf{x}_i, \mathbf{u}_i, \delta\mathbf{t}_i, \boldsymbol{\Pi}) + \epsilon_i \tag{3}$$

which defines our Bayesian regression model. Here, ϵ is the Gaussian noise with zero mean and variance σ^2. Both input parameters and output noise include uncertainty and the goal is to find the probability of dynamic parameters and output noise using input dataset $\mathcal{D} = \{\mathbf{x}, \mathbf{u}\}$ and Bayes theorem:

$$P(\Pi, \sigma|\mathcal{D}) \propto P(\mathcal{D}|\Pi, \sigma)P(\Pi)P(\sigma) \tag{4}$$

We use physics-based priors for dynamic parameters and define them as distributions based on common knowledge about the objects. For instance, we know that the mass of the walker is less than 10kg, friction coefficients are in range (0, 1) and the center of mass is limited by the length and width of the walker. We use a truncated normal distribution since all the parameters have upper and lower bounds (a, b):

$$\pi \sim N_t(\mu, \sigma, a, b)$$

C. Hybrid Manipulation Planning Algorithm

An MPC-based manipulation planner is developed to obtain an optimal path to manipulate the object from one point to another performing a hybrid control of continuous force direction and magnitude and a discrete choice of which leg to push or pull. We formulate the manipulation planning problem as a mixed-integer convex optimization problem [33]. A convex optimization framework based on MPC for motion planning of robots has been developed in [34]. In this work, we extend the framework to manipulation planning.

Algorithm 1 shows the pseudo code of our hybrid manipulation planner. We initialize it with the walker's start and

(1) **Result:** Next action
(2) initialize with $\xi_s, \xi_g, \xi_r, \mathcal{O}, \Pi$;
(3) **while** $\delta_{\mathbf{g}} > \epsilon$ **do**
(4) $\quad$ *cost* $\leftarrow \infty$;
(5) $\quad$ $\mathcal{T}_n \leftarrow$ *NominalTraj*$(\xi_s, \xi_g, \mathcal{O})$;
(6) $\quad$ **for** leg in $\mathbf{ObjectLegs}$ **do**
(7) $\quad\quad$ c_T, *action* $\leftarrow$ *OptTraj*$(\Pi, \xi_s, \xi_g, \mathcal{O}, leg, \mathcal{T}_n)$;
(8) $\quad\quad$ $c_A \leftarrow$ *CostOfAction*$(\xi_r$,*action*,$\mathcal{O})$;
(9) $\quad\quad$ $TotalCost \leftarrow c_T + c_A$;
(10) $\quad\quad$ **if** $TotalCost < cost$ **then**
(11) $\quad\quad\quad$ $cost \leftarrow TotalCost$;
(12) $\quad\quad\quad$ $NextAction \leftarrow action$;
$\quad\quad$ **end**
$\quad$ **end**
(13) $\quad$ **if** *cost*$= \infty$ **then**
(14) $\quad\quad$ return $StayPut$;
$\quad$ **else**
(15) $\quad\quad$ return $NextAction$;
$\quad$ **end**
end

Algorithm 1: Hybrid manipulation controller

goal states, robot's state and walker's as (ξ_s, ξ_g, ξ_r) from feedback. State vectors include object's planar position and velocity ($\xi : \{x, y, \phi, \dot{x}, \dot{y}, \dot{\phi}\}$). Based on the walker's current and desired states, a nominal trajectory is obtained (Line (5)) for linearization of dynamic constraints. The optimization problem solved for nominal trajectory is a simple path planning problem with obstacle avoidance without dynamics constraints for a horizon length same as the main problem.

Subsequently, in Lines (7)-(12), for each leg of the walker, an optimal trajectory is obtained returning the optimal action and cost for a limited horizon. The *OptTraj* function considers all dynamics and obstacle constraints informed by dynamic parameters Π and obstacles $\mathcal{O}$. We will discuss the details of the main optimization problem later. In addition to the cost from *OptTraj* function, in Line (8), we find the cost of the robot to get ready to execute that optimal action using the *CostOfAction* function, which is the sum of two costs; the cost of changing legs and the cost of re-grasping.

Finally, we choose the leg with the minimum total cost to execute in the next time step. If the minimum cost is infinity, it means a feasible trajectory was not found, so the robot stays put. Otherwise, it executes the first action in the action sequence and repeat the same procedure in the next time step until the object reaches the goal zone.

1) Cost function: Although any convex objective function can be used as the cost function, for the purpose of manipulating the mobility aid on a smooth trajectory around obstacles and furniture, we define a minimum path cost function. For a mobile robot and the control horizon length h, the cost function for the shortest path is:

$$J = \omega_1 \delta_{\mathbf{g}} + \omega_2 \sum_{t=1}^{h} \delta_{\mathbf{t}} \tag{5}$$

where $\delta_{\mathbf{t}} = ||\xi_w(t) - \xi_w(t-1)||_2^2$ is the change in walker state between time $(t-1)$ and (t) and $\delta_{\mathbf{g}} = ||\xi_w(h) - \xi_g||_2^2$ shows the difference between the final state in the horizon and the goal state. (ω_1, ω_2) are weights to adjust based on the importance of each term in the cost function.

2) Obstacle avoidance: In order to keep the convex form of the optimization problem, all of the obstacles are written as equivalent surrounding convex forms. Therefore, each obstacle is estimated by a polygonal shape. This is a more conservative approach and provides safer results, but looses some possible paths. Polygon shapes are defined as the intersection of a series of half spaces:

$$\mathcal{O} : \{\xi | \mathbf{A}\xi < \mathbf{b}\} \tag{6}$$

The point ξ is outside of shape O with m number of sides if at least one of the $\mathbf{A}\xi < \mathbf{b}$ inequalities is satisfied:

$$\mathbf{A}\xi < \mathbf{b} + (\mathbf{v} - 1)M, \qquad \sum_{i=1}^{m} v_i \geq 1 \tag{7}$$

where $\mathbf{v}$ is a vector of binary variables and M is a large constant value used in the Big-M method [35]. Equation 7 ensures that at least one element of the vector v equals to 1, so point ξ would be out of the polygonal obstacle.

3) Kinematics constraints: This include starting states and limits on velocity and acceleration which are convex:

$$\xi_w(0) = \xi_s \tag{8}$$

$$-\dot{\xi}_{max} \leq \frac{\xi_w(t+1) - \xi_w(t)}{dT} \leq \dot{\xi}_{max}, \; t = 1, \ldots, h \tag{9}$$

4) constraints: Dynamic constraints are the main part of the optimization problem that connects the robot's movement to the object. Dynamic equations of the manipulation define the relationship between the applied force by the robot and the resulting trajectory of the object. These constraints are linearized over the nominal trajectory. Also, force is limited by the maximum force that robot can apply to the object.

$$\frac{\xi_w(t+1) - \xi_w(t)}{dT} = f(\xi_w(t), \mathbf{u}(t)), \quad t = 1, \ldots, h \tag{10}$$

$$-\mathbf{u}_{max} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}, \quad t = 1, \ldots, h \tag{11}$$

Considering all the above constraints and objective function, the final optimization problem in the *OptTraj* function is:

$$\begin{aligned}
\min_{\mathbf{u}} \quad & J = \omega_1 ||\xi_w(h) - \xi_g||_2^2 + \omega_2 \sum_{t=1}^{h} ||\xi_w(t) - \xi_w(t-1)||_2^2 \\
\text{s.t.} \quad & \xi_w(0) = \xi_s \\
& -\dot{\xi}_{max} \leq \frac{\xi_w(t) - \xi_w(t-1)}{dT} \leq \dot{\xi}_{max}, \quad \forall t \in \{1, .., h\} \\
& \frac{\xi_w(t) - \xi_w(t-1)}{dT} = f(\xi_w(t), \mathbf{u}(t)), \quad \forall t \in \{1, .., h\} \\
& -\mathbf{u}_{max} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}, \quad \forall t \in \{1, .., h\} \\
& \mathbf{A}_{\mathcal{O}} \xi_{w,p}(t) > \mathbf{b}_{\mathcal{O}} + (\mathbf{v}(t) - 1)M, \quad \forall t \in \{0, .., h\} \\
& \qquad \forall \mathcal{O} \in \mathcal{S}_{\mathcal{O}} \\
& \sum_{i=1}^{m} v_i(t) \geq 1, \quad \forall t \in \{0, .., h\}
\end{aligned}$$

In the above, $\xi_{w,p}$ is used to show that only the position part of walker's state is used for obstacle avoidance.

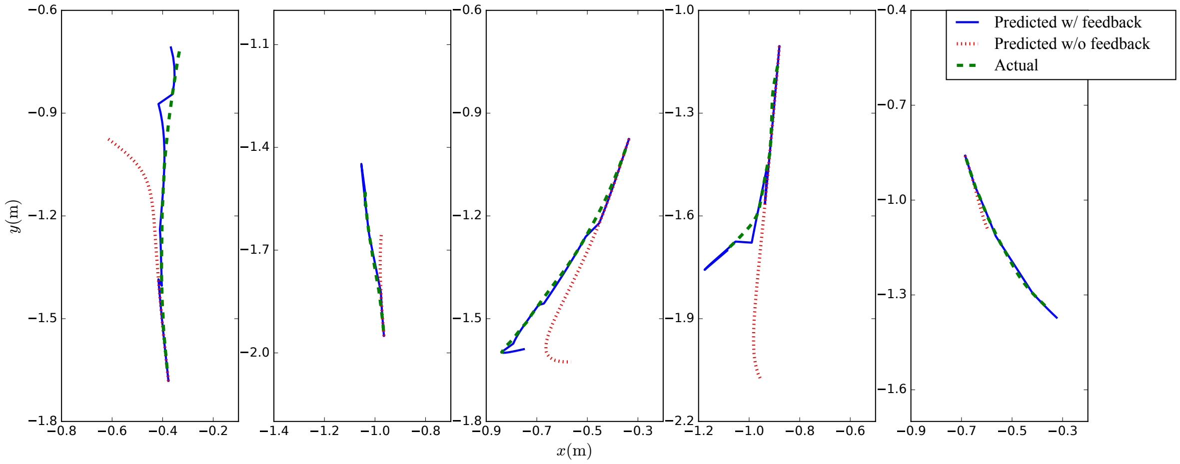

Fig. 5: Comparison between actual and predicted trajectories of walker, with and without feedback for five different trajectory episodes. The actual trajectory is from collected data using motion capture.

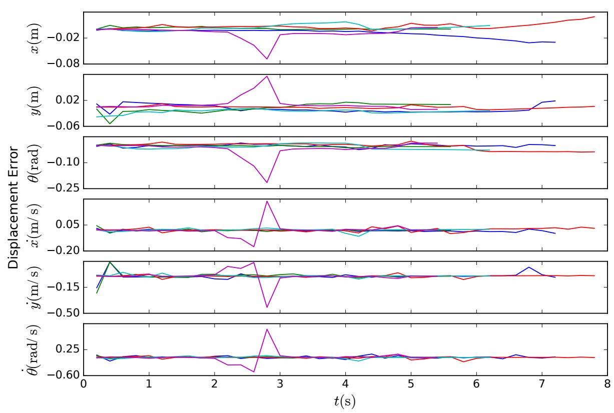

Fig. 6: Displacement errors in predicted trajectories of walker without feedback.

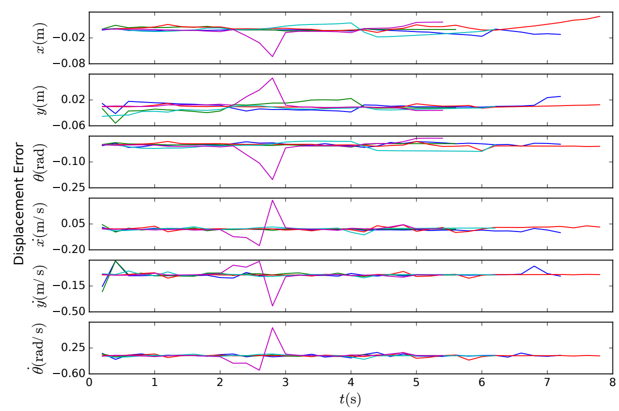

Fig. 7: Displacement errors in the predicted trajectories of a walker with feedback.

IV. IMPLEMENTATION

This section describes the implemented procedure for different parts of the proposed approach.

A. Data Collection and Parameter Estimation

We collected motion data using a 14 Flex13 cameras motion capture system (Optitrack, Naturalpoint, Inc.). In each trial, PAM pushes/pulls one of the walker legs starting from one of the 8 equally-distributed directions. Synchronized force data are collected by the robot's one directional force sensor (Futek Industries). The reason to avoid collecting torque data is to develop a simple and general model with minimal dataset. However, the algorithm could be implemented with a torque sensor as well. The motion capture system records the state of both walker and PAM using 7 tracking markers, 4 on the walker and 3 on the PAM's gripper. In total, we collected 39 trials of short trajectories (about 280,000 data points) and divided them into 34 training and 5 test sets. Each trial is about 5-7s and is collected at 1.2kHz sampling rate. For the Bayesian regression model, we have used PYMC package in python with 20000 samples.

B. Manipulation Planning

The manipulation problem is coded as a Mixed-Integer Quadratic Program and is solved efficiently using the Gurobi numerical optimization package [36]. The nominal trajectory is obtained by solving a simplified path planning problem.

Weights in the cost function are adjusted and normalized in regards to the problem scale and experiments. The goal region is defined as the area within 0.3 distance from the goal state. The horizon length is set to a value in range [7, 15] in most cases, but we also compare the effect of different values of horizon length on different models in the next section.

V. EVALUATION

A. Dynamic Model Learning Performance

As previously stated, the object's dynamic model is learned using MCMC sampling. On a Core i7 2.4GHz system, it takes about an hour to get 20000 samples.

After obtaining an object model based on the training dataset, we tested it on our test dataset containing 5 trajectory episodes. Each trajectory prediction begins from the actual starting point and then we only use the actual dataset as feedback input every 2 seconds. Figure 5 provides a comparison between the predicted trajectory without feedback, predicted trajectory with feedback and the actual trajectory.

For better evaluation of the model, a plot of displacement errors, which is the difference between predicted and actual displacement at each step is presented in Fig. 6. This figure

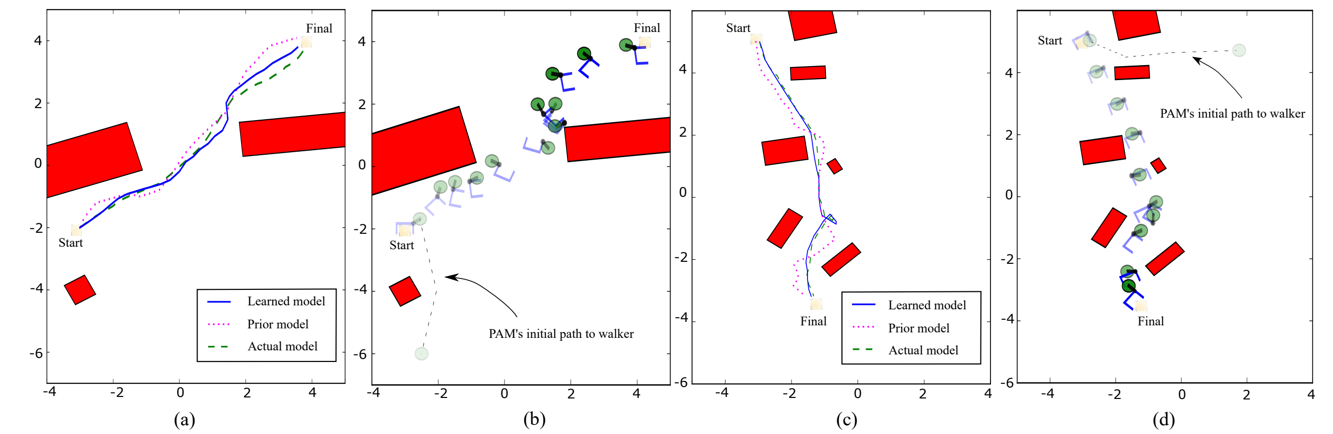

Fig. 8: Simulated results of walker manipulation by PAM. (a) Comparison between planning based on actual, prior and learned models in a simple environment. We can see that the one with the prior model is less smooth and can easily collide with the obstacles. (b) Simulated walker trajectory based on the learned model in a simple environment. PAM will move to the walker first before starting manipulation. (c) Comparison between planning based on actual, prior and learned models in a cluttered environment. (d) Simulated walker trajectory based on the learned model in a cluttered environment.

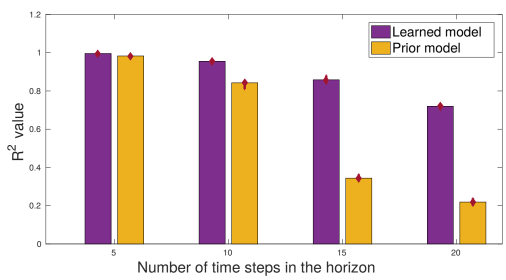

Fig. 9: Effect of horizon length on the smoothness of the planned trajectory.

shows the actual error at each step instead of accumulated error as in the trajectory plot. The error in the Y direction is more than the other axis because the trajectory is mostly in that direction. So, the inaccuracy in the force input is larger in magnitude which leads to larger errors. Using feedback helps to stay on the trajectory and eliminate the accumulated error every two seconds, but it does not change the displacement error much since this is based on the obtained model and noise in the system.

Additionally, we can see that in the far left case, errors get higher as we move towards the end of the trajectory. We believe this is due to forces applied by the gripper fingers which are not measured in this study. This happens in cases when the force direction is such that gripper fingers apply more force and actually play a role in influencing dynamics. To avoid this kind of error, in the manipulation planner, we assume that the robot repositions at each step to execute only straight force and not in any other direction. However, this statement may not hold in the real world. A better force and torque measurement approach is needed to get more accurate results for experimental studies.

B. Manipulation Planning Performance

We ran the manipulation planner for two different simulation setups: a simple environment with 3 obstacles and a cluttered environment. We define the actual model close to the obtained model from the dynamic learner. We also add noise to the system for simulating the resulting trajectory. For each setup we tested and compared 3 models: (1) model only based on priors (2) model obtained from the dynamic model learner (3) actual simulation model. Figure 8 Shows resulting trajectories for both of these simulation setups.

As expected, although the length of the obtained trajectory in both environments is almost the same, the simple environment needs fewer steps (about 40 steps with timestep=1s) to finish the task and the highly cluttered one takes nearly twice as long to move the walker to the desired region (about 80 steps with timestep=1s). This is because it has to move slower to be able to avoid the obstacles. Also manipulating in narrow passages is more difficult and takes multiple steps to navigate. We can improve this by using a more accurate obstacle avoidance method and avoid being overly conservative. Figure 9 presents the effect of the horizon length in MPC on the smoothness of the manipulated object's trajectory. We use R^2 value with respect to the actual model trajectory to show smoothness of the obtained trajectory. These values indicate that a longer horizon has a greater impact on a less accurate model compared to a better model.

Running on a Core i7 2.4GH platform, the computational time for each step is less than a millisecond which is considered real-time. However, this can vary based on the feasibility and complexity of the problem.

VI. CONCLUSION

Patient falls are a major healthcare challenge that often result when a patient egresses bed without assistance. A problem with current bed alarms and patient monitoring systems is that hospital staff are often outside the room and cannot respond in time to prevent the fall. Mobile robots offer the potential to bridge this gap by providing continuous care and monitoring with a physical presence in the room. We demonstrated that a Patient Assistant Mobile (PAM) robot can safety manipulate and deliver a walker to a bedside. To accomplish this we had to overcome two major challenges, i) object dynamic modeling and ii) Manipulation Planning. A Bayesian regression algorithm was used to estimate the object dynamic model and a novel manipulation algorithm based on MPC and hybrid optimization was developed. Our results show that using minimal data, the PAM robot can successfully manipulate a walker to a desired location within acceptable error limits. Future work will focus on reducing trajectory errors and learning dynamic models from other unknown objects that may need to be moved. Moreover, we would like to test the manipulation algorithm in real experimental setups.

REFERENCES

[1] A. F. Ambrose, G. Paul, and J. M. Hausdorff, "Risk factors for falls among older adults: a review of the literature," *Maturitas*, vol. 75, no. 1, pp. 51–61, 2013.

[2] J. V. Agostini and M. E. Tinetti, "Drugs and falls: rethinking the approach to medication risk in older adults," *Journal of the American Geriatrics Society*, vol. 50, no. 10, pp. 1744–1745, 2002.

[3] E. B. Hitcho, M. J. Krauss, S. Birge, W. Claiborne Dunagan, I. Fischer, S. Johnson, P. A. Nast, E. Costantinou, and V. J. Fraser, "Characteristics and circumstances of falls in a hospital setting," *Journal of general internal medicine*, vol. 19, no. 7, pp. 732–739, 2004.

[4] A. Kristoffersson, S. Coradeschi, and A. Loutfi, "A review of mobile robotic telepresence," *Advances in Human-Computer Interaction*, vol. 2013, p. 3, 2013.

[5] D. Oliver, M. Britton, P. Seed, F. Martin, and A. Hopper, "Development and evaluation of evidence based risk assessment tool (stratify) to predict which elderly inpatients will fall: case-control and cohort studies," *Bmj*, vol. 315, no. 7115, pp. 1049–1053, 1997.

[6] J. Pripfl, T. Körtner, D. Batko-Klein, D. Hebesberger, M. Weninger, C. Gisinger, S. Frennert, H. Eftring, M. Antona, I. Adami *et al.*, "Results of a real world trial with a mobile social service robot for older adults," in *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. IEEE Press, 2016, pp. 497–498.

[7] N. Casiddu, A. Cesta, G. Cortellessa, A. Orlandini, C. Porfirione, A. Divano, E. Micheli, and M. Zallio, "Robot interface design: The giraff telepresence robot for social interaction," in *Ambient Assisted Living*. Springer, 2015, pp. 499–509.

[8] D. Nakhaeinia, P. Payeur, T. S. Hong, and B. Karasfi, "A hybrid control architecture for autonomous mobile robot navigation in unknown dynamic environment," in *IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2015, pp. 1274–1281.

[9] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[10] R. S. Pol and M. Murugan, "A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods," in *International Conference On Industrial Instrumentation and Control (ICIC)*. IEEE, 2015, pp. 1339–1344.

[11] M. Mast, M. Burmester, B. Graf, F. Weisshardt, G. Arbeiter, M. Španěl, Z. Materna, P. Smrž, and G. Kronreif, "Design of the human-robot interaction for a semi-autonomous service robot to assist elderly people," in *Ambient Assisted Living*. Springer, 2015.

[12] A. Merryweather, R. Sabbagh Novin, and M. Yazdani, "Optimal motion and mobility aid manipulation planning to enable personal activity monitoring and facilitate safer sit-to-walk transitions," in *The Fifth International Conference on Ambulatory Monitoring of Physical Activity and Movement (ICAMPAM), (Poster presentation)*, 2017.

[13] T. Ogata, H. Ohba, J. Tani, K. Komatani, and H. G. Okuno, "Extracting multi-modal dynamics of objects using rnnpb," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2005, pp. 966–971.

[14] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action-initial steps towards artificial cognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3. IEEE, 2003, pp. 3140–3145.

[15] A. R. Vithani and K. C. Gupta, "Estimation of object kinematics from point data," in *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2002, pp. 395–403.

[16] J. Scholz, M. Levihn, C. L. Isbell, H. Christensen, and M. Stilman, "Learning non-holonomic object models for mobile manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5531–5536.

[17] K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, no. 6971, p. 244, 2004.

[18] J. Scholz, N. Jindal, M. Levihn, C. L. Isbell, and H. I. Christensen, "Navigation among movable obstacles with learned dynamic constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3706–3713.

[19] O. Khatib, "Mobile manipulation: The robotic assistant," *Robotics and Autonomous Systems*, vol. 26, no. 2-3, pp. 175–183, 1999.

[20] M. Stilman, K. Nishiwaki, and S. Kagami, "Learning object models for whole body manipulation," in *7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2007, pp. 174–179.

[21] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986.

[22] M. Salganicoff, G. Metta, A. Oddera, and G. Sandini, *A vision-based learning method for pushing manipulation*. University of Pennsylvania, 1993, vol. 54.

[23] Q. Li and S. Payandeh, "Manipulation of convex objects via two-agent point-contact push," *The international journal of robotics research*, vol. 26, no. 4, pp. 377–403, 2007.

[24] E. Arruda, M. J. Mathew, M. Kopicki, M. Mistry, M. Azad, and J. L. Wyatt, "Uncertainty averse pushing with model predictive path integral control," *arXiv preprint arXiv:1710.04005*, 2017.

[25] T. Hermans, F. Li, J. M. Rehg, and A. F. Bobick, "Learning contact locations for pushing and orienting unknown objects," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 435–442.

[26] J. P. Desai and V. Kumar, "Nonholonomic motion planning for multiple mobile manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4. IEEE, 1997, pp. 3409–3414.

[27] J. Z. Woodruff and K. M. Lynch, "Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4066–4073.

[28] F. R. Hogan, E. R. Grau, and A. Rodriguez, "Reactive planar manipulation with convex hybrid mpc," *arXiv preprint arXiv:1710.05724*, 2017.

[29] F. R. Hogan and A. Rodriguez, "Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics," *arXiv preprint arXiv:1611.08268*, 2016.

[30] J. Pajarinen, V. Kyrki, M. Koval, S. Srinivasa, J. Peters, and G. Neumann, "Hybrid control trajectory optimization under uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[31] M. Ciocarlie, F. M. Hicks, R. Holmberg, J. Hawke, M. Schlicht, J. Gee, S. Stanford, and R. Bahadur, "The velo gripper: A versatile single-actuator design for enveloping, parallel and fingertip grasps," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 753–767, 2014.

[32] J. M. Bernardo and A. F. Smith, "Bayesian theory," 2001.

[33] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[34] R. Sabbagh Novin, M. Tale Masouleh, and M. Yazdani, "Optimal motion planning of redundant planar serial robots using a synergy-based approach of convex optimization, disjunctive programming and receding horizon," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 230, no. 3, pp. 211–221, 2016.

[35] A. Richards and J. How, "Mixed-integer programming for control," in *Proceedings of the American Control Conference*. IEEE, 2005, pp. 2676–2683.

[36] Gurobi Optimization Inc., "Gurobi optimizer reference manual," 2018. [Online]. Available: http://www.gurobi.com