2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Gaussian Mixture Likelihood-based Adaptive MPC for Interactive Mobile Manipulators

Dimitrios Rakovitis[1] and Dennis Mronga[1]

Abstract— **Mobile robots are nowadays frequently used for interaction tasks in the real world, e.g. for opening doors or for pick-and-place tasks. When used in real-world environments, adapting the robot controllers to uncertain contact dynamics is a significant challenge. Adaptive Model Predictive Control (AMPC) is an approach for controlling robot motions while adapting to uncertain or changing dynamics. However, most of the existing AMPC approaches used in mobile manipulation require either expert tuning or extensive training, making it very difficult to introduce novel or diverse tasks. In addition, the adjustment of several, independent environment parameters is usually not considered in the AMPC formulation. In this work, we introduce a hierarchical approach that uses Gaussian Mixture Models (GMMs) and Gaussian Mixture Regression (GMR) to predict the dynamic model parameters of MPC based on proprioceptive measurements and perform tasks with multiple unknown environmental parameters. The approach is evaluated in simulation and in real experiments on a mobile manipulator and compared to several baseline methods. It is shown that it outperforms standard MPC and an existing AMPC approach on several tasks such as carrying, pushing, and door opening.**

I. INTRODUCTION

In recent years, Model Predictive Control (MPC) has had great success in controlling complex robots to perform dynamic motions [1]. However, when the robot has to interact with an uncertain or unknown environment, the MPC will generate insufficient commands due to the lack of knowledge of the real dynamics. Examples are cases where the robot has to carry or manipulate payloads with unknown weight and inertial properties [2], or walk on surfaces with unknown friction or inclination [3]. To address these issues, Adaptive Model Predictive Control (AMPC) has shown great promise [2], [3], [4], [5], [6], [7], [8], [9]. In this work, we use AMPC for predicting the dynamic parameters of the contacted environment and adapting the MPC model at each time step before solving the optimization problem. Several approaches have been considered in previous works for similar purpose, using Adaptive Control (AC), System Identification (SI), Machine Learning (ML), and Robust MPC (RMPC).

In the work presented in [4], AC is combined with MPC to make a quadruped robot pull or carry an unknown weight while tracking a desired trajectory. A Control Lyapunov Function derived from Lyapunov analysis as used in AC for manipulators [10] is added as inequality constraint, to

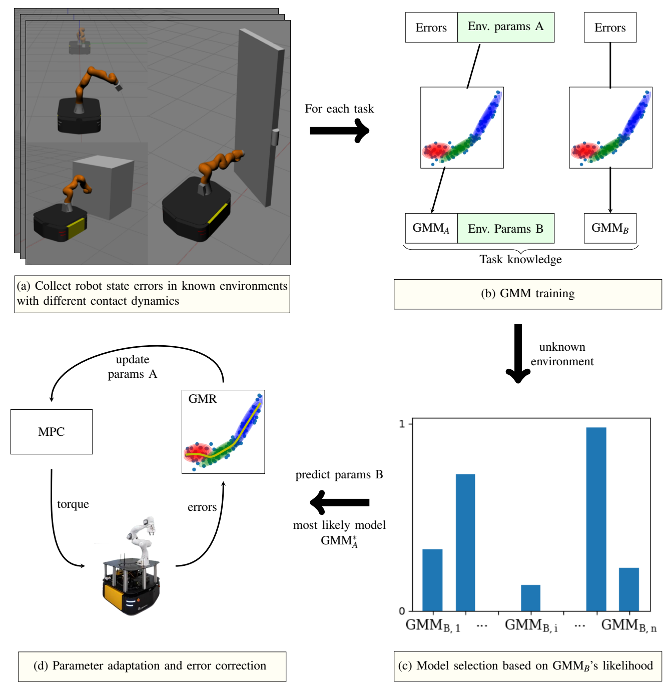


Fig. 1: Graphical abstract of the proposed AMPC approach.

show that stable behaviors can be achieved with smaller horizons and without the need of terminal cost. In [2], the environment is modeled as a spring-mass-damper (SMD) and its parameters are predicted through AC or SI. The approach is used to control a mobile manipulator with whole-body AMPC for object lifting and door opening tasks, in which standard MPC fails. The authors show that for those tasks, AC-MPC generalizes better in comparison with the SI approach, that uses a Kalman filter for prediction. Similarly, in [9] a contact force is predicted using AC, which is then regulated by MPC in a loco-manipulation formulation to achieve a pushing task with uncertainty on a quadruped robot.

On the ML side, Reinforcement Learning (RL) has been used to learn a model adaptation policy that predicts the change of the robot's mass, inertia and ground friction coefficient used in the MPC formulation [3]. Here, a four-legged robot is trained in simulation using proprioceptive sensor measurements to follow a desired speed while walking on low friction ground, various ramps or carrying different weights on its back. In a similar use case [5], the authors model the dynamics as a neural network (NN), instead of predicting specific parameters. They follow a model-based RL approach and incorporate meta-learning to adapt the

*This research was done in the HARTU project (grant number 101092100) funded by the European Union.
[1]All authors are with Robotics Innovation Center of German Research Center for Artificial Intelligence GmbH (DFKI), Bremen, Germany. Corresponding author's email: dimitrios.rakovitis@dfki.de

network's weights online. Using a small history of data, they select initial weights corresponding to the most similar training condition and then adapt them with gradient-based methods. In [11], the authors present a RMPC approach, which learns model uncertainties for quadrupedal locomotion through RL. For manipulation tasks, deep learning has been used in conjunction with MPC to learn how to cut a variety of objects with unknown dynamics [6]. Furthermore, the authors of [7] use supervised learning to make a mobile manipulator reach a certain pose or open doors, when either the robot or the environment model is inaccurate. They describe the modeling error using linearly combined trigonometric basis functions and then perform learning in two separate stages: An offline optimization stage that fits the hyperparameters of the error model to a data set obtained from user demonstrations on multiple tasks, and an online adaptation stage, where they use a Kalman filter to perform task-specific parameter adaptation using online measurements. In [8], modeling errors are addressed online by fitting them to a limited window of past data. The dynamics are expressed in control-affine form, and model mismatch is determined using least-squares after predicting the robot's acceleration through sensor fusion.

Although the above methods show promising results, they have the drawback that it is difficult to incorporate novel, previously unknown tasks into the respective MPC framework. In the case of AC, the Lyapunov function must be defined and a large number of parameters must be manually adjusted, e.g., the adaptation and control gains, to achieve a good fit to the set of observed tasks. When a completely different task is considered, the procedure must be repeated by an expert to find the optimal parameters for the desired dynamic behavior in all tasks. The same is true for SI and learning procedures that use Kalman filters, which require proper tuning of hyperparameters, like model and noise matrices [2], [7], [12], while also for RMPC approaches which use predefined uncertainty sets [11], [13]. Similarly, other SI methods [14], [15] require excitation trajectories or assumptions that are tied to the task at hand and are usually not applicable to diverse cases. In RL, integrating new tasks typically requires hours of training in a realistic simulation environment and manual tuning of the reward function. Model-based RL can solve some of these problems. However, as the complexity of the model increases, the computational cost can negatively impact the maximum control frequency [5].

Moreover, the simultaneous change of several unknown dynamic parameters is rarely studied in the existing approaches. Unknown factors may include mass inertia properties, ground friction, damping and stiffness, etc. Different combinations of mass and friction can result in the same static force at steady state. In such scenarios it is be very difficult to accurately predict the correct dynamics.

In this work, the above challenges are addressed using the following approach. Since many robotic tasks can be described by a combination of simple carrying, pushing, and pulling tasks, a SMD system is used to model the contact forces exerted by the robot on the environment. The parameters of the SMD system are adjusted online while an unknown task is executed. To achieve this, we model the joint probability distribution of the dynamic parameters and the state error measurements as multivariate Gaussian Mixture Models (GMMs), one for each executed task. For prediction, we use a hierarchical approach. First, the most likely GMMs for the current task are selected by comparing their log-likelihoods using state error measurements as input. Then, the computed probabilities are used to predict the environmental damping and stiffness by weighted averages. Second, the most likely model is used to predict the remaining dynamic model parameters, i.e., spring mass, inertia, and static force. GMMs have been used before to estimate robotic tasks and modify the robot control mode, for example in rehabilitation robotics [16] and balancing applications [17], or to classify outdoor environments for inspection robots [18], [19]. Other data-driven approaches, such as NNs, could also be used for such purposes [11]. However, GMMs are employed here because they require significantly less training time, without compromising performance.

The mobile manipulator shown in Fig. 1.a is used as experimental platform. It consists of an omnidirectional mobile base which carries either the KUKA iiwa14 (simulation) or the Franka Panda Emika arm (real hardware), both with 7 degrees of freedom. Three different experiments are performed for evaluation. First, the generality of the proposed method is compared to state-of-the-art methods by evaluating the performance on executing unseen, unrelated tasks with increasing complexity (see Fig. 1.a). Second, we focus on similar tasks where the environment parameters are very difficult to distinguish because of overlapping effects. Third, the proposed approach is implemented on a real system and compared to standard MPC for carrying and pulling payloads with unknown properties. In summary, the contributions of this work are the following:

- We introduce a hierarchical, GMM-based AMPC approach, which allows intuitive and fast model training, and which generalizes well to unknown tasks by using a likelihood-based model selection strategy.
- We address the problem of contact interaction tasks with more than one variable environment parameter.
- We validate the proposed method in simulation and in real-world experiments showcasing its advantage with respect to a state-of-the-art AMPC approach.

The paper proceeds as follows: Section II outlines the proposed control architecture methodology, Section III presents experimental results showcasing its effectiveness, and Section IV concludes with thoughts on future improvements.

II. METHODOLOGY

In this section, the control architecture of the proposed approach as shown in Fig. 2 is described. First, the MPC problem is formulated, then the GMM training and the model selection procedure to choose the most appropriate GMM for the current task is explained. Finally, the approach to predict the unknown parameters of the dynamic model is explained.

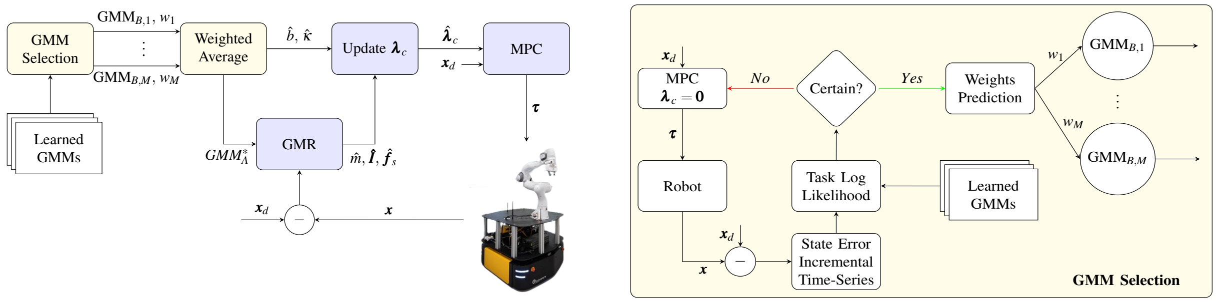


Fig. 2: Framework overview (left) and model selection procedure (right). For each task, GMMs are trained offline from proprioceptive measurements and known environmental parameters. The robot compares their probabilities using the state error to find the most similar models for each task. These are weighted to predict the environmental damping and stiffness, while the best model is passed to the GMR, which continuously predicts the remaining parameters.

A. AMPC

The following non-linear MPC problem is formulated in joint-space

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u}} \quad & ||\boldsymbol{x}_d(T)-\boldsymbol{x}(T)||_{\boldsymbol{Q}}^2 + \int_0^T (||\boldsymbol{x}_d-\boldsymbol{x}||_{\boldsymbol{Q}}^2 + ||\boldsymbol{u}_d-\boldsymbol{u}||_{\boldsymbol{R}}^2)\ dt \\
\text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
& \boldsymbol{x}_{min} \le \boldsymbol{x} \le \boldsymbol{x}_{max} \\
& \boldsymbol{u}_{min} \le \boldsymbol{u} \le \boldsymbol{u}_{max}
\end{aligned} \tag{1}$$

where $\boldsymbol{x} = (\boldsymbol{q},\dot{\boldsymbol{q}})$ is the robot state, $\boldsymbol{u} = \boldsymbol{\tau}$ the commanded torque, $\boldsymbol{x}_d,\boldsymbol{u}_d$ are the respective desired values, $\boldsymbol{q},\dot{\boldsymbol{q}}$ are the robot joint positions and velocities, $\boldsymbol{f}$ is a function describing the robot dynamics, $\boldsymbol{x}_0$ is the initial known state, T the optimization horizon, and $\boldsymbol{x}_{min},\boldsymbol{x}_{max},\boldsymbol{u}_{min},\boldsymbol{u}_{max}$ are the respective state and torque limits. The robot dynamics $\boldsymbol{f}$ are described as follows, assuming a rigid contact at the end-effector (EE):

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{J}_c^T \boldsymbol{\lambda}_c \tag{2}$$

where $\boldsymbol{H}$ is the joint space mass-inertia matrix, $\ddot{\boldsymbol{q}}$ is the joint's acceleration, $\boldsymbol{C}$ accounts for the Coriolis-centrifugal and gravity forces, $\boldsymbol{\tau}$ is the torque applied by the joint motors, and $\boldsymbol{J}_c$ is the contact Jacobian. The contact wrench $\boldsymbol{\lambda}_c \in \mathbb{R}^6$ applied at the EE, is modeled as a SMD as in [2], because of its simplicity along with the ability to approximate a wide range of real-world systems. Thus,

$$\boldsymbol{\lambda}_c = \boldsymbol{H}_{env}\ddot{\boldsymbol{x}}_{ee} + \boldsymbol{B}_{env}\dot{\boldsymbol{x}}_{ee} + \boldsymbol{K}_{env}\boldsymbol{e}_{\mathrm{x}_{ee}} + \boldsymbol{\lambda}_s \tag{3}$$

where $\dot{\boldsymbol{x}}_{ee},\ddot{\boldsymbol{x}}_{ee}$, are the EE twist and spatial acceleration, $\boldsymbol{H}_{env}$, $\boldsymbol{B}_{env}$, $\boldsymbol{K}_{env} \in \mathbb{R}^{6\times 6}$ are diagonal matrices representing the mass-inertia properties, damping, and stiffness of the contacted environment, respectively, $\boldsymbol{\lambda}_s \in \mathbb{R}^6$ is the static contact wrench, and $\boldsymbol{e}_{\mathrm{x}_{ee}}$ is the EE pose error. In cases where the EE acceleration can't be measured directly, the following PD controller can be used instead

$$\ddot{\boldsymbol{x}}_{ee} = \ddot{\boldsymbol{x}}_{d,ee} + \boldsymbol{K}_D\,\boldsymbol{e}_{\dot{\mathrm{x}}_{ee}} + \boldsymbol{K}_P\,\boldsymbol{e}_{\mathrm{x}_{\mathrm{ee}}} \tag{4}$$

where $\ddot{\boldsymbol{x}}_{d,ee}$ is the desired spatial acceleration, $\boldsymbol{K}_D, \boldsymbol{K}_P > 0$ are diagonal gain matrices, and $\boldsymbol{e}_{\dot{\mathrm{x}}_{ee}}$ is the EE twist error.

The objective is to estimate the dynamic parameters of the contacted environment $\boldsymbol{H}_{env}$, $\boldsymbol{B}_{env}$, $\boldsymbol{K}_{env}$, and $\boldsymbol{\lambda}_s$ using a hierarchical GMM-GMR approach as described in the following sections. Given these parameters, the contact wrench $\boldsymbol{\lambda}_c$ is estimated at each time step and given to the MPC before solving. Concerning the dynamic characteristics of the robot, it is assumed that they are known with good accuracy. Since we use an industrial robot for experimentation, this assumption is well founded.

B. GMM Training

We model the joint probability distribution of the dynamic environment parameters and the proprioceptive sensor measurements (task errors) as multi-variate GMM. As shown in [3], [20], the use of proprioceptive observations is a suitable method to predict dynamic model parameters. Each task to be performed by the robot is repeated with different known environment parameters in order to create a data set with samples of the form $\boldsymbol{S} = [\boldsymbol{e}_{\mathrm{x}}, \boldsymbol{a}^T]$, with $\boldsymbol{e}_{\mathrm{x}} = [\boldsymbol{e}_{\mathrm{x}_{ee}}^T, \boldsymbol{e}_{\dot{\mathrm{x}}_{ee}}^T, \boldsymbol{e}_{\mathrm{x}_{bl}}^T, \boldsymbol{e}_{\dot{\mathrm{x}}_{bl}}^T]$, where $\boldsymbol{e}_{\mathrm{x}_{bl}}, \boldsymbol{e}_{\dot{\mathrm{x}}_{bl}}$ are the pose and twist errors of the mobile base (base link), and $\boldsymbol{a}$ is a vector containing the dynamic parameters of the contacted environment. From these samples, we train one GMM per task using Expectation Maximization [21]. Each GMM is defined as

$$p(\boldsymbol{e}_{\mathrm{x}}, \boldsymbol{a}^T) = \sum_{k=1}^{K} \pi_k \mathcal{N}_k(\boldsymbol{e}_{\mathrm{x}}, \boldsymbol{a} | \boldsymbol{\mu}_{e_{\mathrm{x}}a}^k, \boldsymbol{\Sigma}_{e_{\mathrm{x}}a}^k) \tag{5}$$

where, $\mathcal{N}_k$ are Gaussian distributions with mean $\boldsymbol{\mu}_{e_{\mathrm{x}}a}^k$ and covariance $\boldsymbol{\Sigma}_{e_{\mathrm{x}}a}^k$, weighted by priors $\pi_k \in [0,1]$ which sum up to one, and K is the number of Gaussians. An appropriate selection of the hyperparameter K is done through grid search on the obtained dataset. Now, for a previously unseen environment, the dynamic parameters of the contacted environment could be predicted through GMR from the conditional distribution $p(\boldsymbol{a}^T \mid \boldsymbol{e}_{\mathrm{x}})$. However, combining all environment parameters in one model is not ideal, because different parameter combinations can result in similar error measurements. To avoid this, the parameters are separated and estimated hierarchically in two steps.

To achieve this, we train two sets of GMMs, where each set contains one GMM per task. The first set, denoted as GMM_A, models the joint distribution of the environmental parameters $\boldsymbol{a} = [m, \boldsymbol{I}^T, \boldsymbol{f}_s^T]^T$ together with the state error measurements as $p(\boldsymbol{e}_{\mathrm{x}}, \boldsymbol{a}^T)$. Note that the mass m and inertia

$\boldsymbol{I} = [I_{xx}, I_{yy}, I_{zz}]^T$ are the diagonal entries of the mass-inertia matrix $\boldsymbol{H}_{env}$, and $\boldsymbol{f}_s$ is the linear part of the static contact wrench $\boldsymbol{\lambda}_s$. The angular part of $\boldsymbol{\lambda}_s$ is computed using the forward kinematics of the robot. The second set, denoted as GMM_B, models the joint distribution only of the state error measurements $p(\boldsymbol{e}_\text{x})$. It is used to identify the executed task and predict the damping b and stiffness κ of the contacted environment. Note that b and κ are the diagonal entries of the damping and stiffness matrices $\boldsymbol{B}_{env}$ and $\boldsymbol{K}_{env}$, respectively.

C. GMM Selection

Here, we use GMM_B to identify the task at hand and select the appropriate model for predicting the parameters $\boldsymbol{a} = [m, \boldsymbol{I}^T, \boldsymbol{f}_s^T]^T$.

In our approach, the robot has no prior information about the environment. Model selection is performed online during task execution, where initially the contact wrench in (1) is set to $\boldsymbol{\lambda}_c = \mathbf{0}$. At each time step, the system state error is used to compute the log-exp-log likelihood score for each $\text{GMM}_{B,j}$ in GMM_B (for each known task), and identify the currently executed task as follows:

$$j = \underset{\text{GMM}_{B,j}}{\arg\max} \left(\log \left(\exp \left(\frac{1}{N} \sum_{k=1}^{N} \log(\boldsymbol{e}_{x,k} \mid \text{GMM}_{B,j}) \right)\right)\right) \quad (6)$$

where N is the number of state error samples and j is the identifier of the most probable task. This procedure is performed until convergence to a specific task is achieved.

D. Two-Step Parameter Prediction

Having identified the task at hand, we predict the dynamic parameters of the contacted environment in two steps. In the first step, we use the M most probable models from GMM_B computed in (6), which are the ones with log likelihood score $\neq -\infty$, to predict the damping and stiffness for the current task as follows:

$$\begin{aligned} \hat{b} &= \frac{w_1 b_1 + \cdots + w_M b_M}{\sum_{i=1}^{M} w_i} \\ \hat{\kappa} &= \frac{w_1 \kappa_1 + \cdots + w_M \kappa_M}{\sum_{i=1}^{M} w_i} \end{aligned} \quad (7)$$

where b_i, κ_i, with $i = 1, \cdots, M$ are the known damping and stiffness values from the i-th (known) task, and w_i are weighting factors, which are equal to the normalized log-exp-log likelihood obtained from the i-th model in GMM_B. Note that the accuracy of the predictions depends on the quantity and quality of the selected models. On the one hand, the more known models are available, the better the results. On the other hand, the more the current task differs from the training scenarios, the worse the results become.

In the second step, the remaining environmental parameters are predicted through GMR, which uses the model from the set GMM_A with the highest log-exp-log likelihood. This model that we refer to as GMM_A^* corresponds to the most probable task T_j. Hence, given its joint probability distribution $p_{T_j}(\boldsymbol{e}_\text{x}, \boldsymbol{a}^T)$, the update law of the parameters $\boldsymbol{a} = [m, \boldsymbol{I}^T, \boldsymbol{f}_s^T]^T$ is defined as:

$$\dot{\hat{\boldsymbol{a}}} = \mathbb{E}(\boldsymbol{a})\, dt \quad (8)$$

where $\mathbb{E}(\boldsymbol{a})$ is the expectation of the parameters $\boldsymbol{a}$, derived by GMR using the conditional distribution $p_{T_j}(\boldsymbol{a}^T \mid \boldsymbol{e}_\text{x})$, and $\hat{\boldsymbol{a}}$ represents the estimate of $\boldsymbol{a}$. This particular separation of parameters was undertaken under the assumption that the contacted environments maintain fixed damping and stiffness.

III. Experimental Evaluation

In this section, the proposed methodology is evaluated in simulation and real-world experiments. The MPC problem formulated in (1) is solved at $150Hz$ with the Feasibility-driven Differential Dynamic Programming method of the Crocoddyl library [22], using the pinocchio library [23] for computing the robot dynamics.

In the following, the results from three experiments are presented: In the first experiment, our approach is compared with several baseline methods on carrying, pushing, and door opening tasks, which represent diverse and unrelated environment characteristics. Secondly, we focus on the case of dealing with tasks that are difficult to identify, such as pushing objects over different floors and opening doors with friction. Thirdly, our method is applied on the real system to show its feasibility for carrying and pulling tasks. The first two experiments are performed in the Gazebo simulation environment using ODE as physics engine, where the entire robot is torque controlled, assuming an ideal floor, while the third experiment is performed on the real robot. Since the omnidirectional base of the real robot is velocity-controlled, the torque commands calculated for the base from (1) are transformed into twist commands using admittance control. The effect of the payload on the wheels is chosen empirically in each experiment.

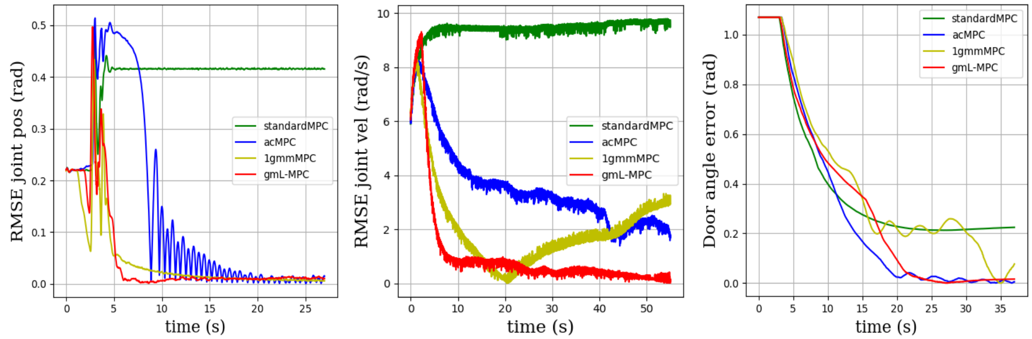
Four different MPC methods are used for comparison:

- *standardMPC*: Classic MPC without adaptation, $\boldsymbol{\lambda}_c = \mathbf{0}$.
- *adaptiveControlMPC (acMPC)*: AMPC method which models the contact force as SMD, whose parameters are estimated using AC as in [2]. The AC parameters are tuned manually.
- *1gmmMPC*: AMPC method, which learns the contact force through GMM-GMR from a single data set containing all tasks.
- *gmL-MPC*: The proposed approach as described in Section II.

The target evaluation tasks are defined as:

- **Carrying**: Follow a reference trajectory to go from point A to point B while carrying a cube ($0.1m \times 0.1m \times 0.1m$) (simulation) or a weight plate (real robot) with unknown mass.
- **Push/Pull**: Reach a desired state while pushing a cube ($1m \times 1m \times 1m$) (simulation) or pulling a handcart (real robot) with unknown mass over a surface with friction.
- **Door Opening**: Open a door ($0.1m \times 1.0m \times 2.2m$) with unknown mass, damping, and stiffness up to a desired angle. Also, friction is considered in case where the door is in touch with the ground.

To train the learning approaches the tasks are executed with several variations of the manipulated objects, given in



(a) Carrying task (b) Push task (c) Door Opening Task

Fig. 3: Results of experiment 1: (a) RMSE of joint positions of the arm for carrying a cube of $m = 12kg$, (b) RMSE of joint velocities of the base for pushing a cube of $m = 12kg$ on a floor with friction coefficient $\mu = 0.5$, (c) Door angle error for the door opening task of a floating door with $m = 5kg$, $b = 2.5kg/s$, $\kappa = 12kg/s^2$.

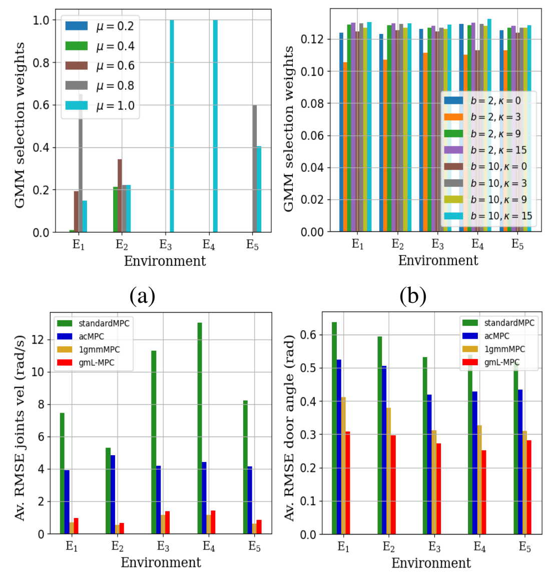
	b	κ	m	μ
Simulation				
Carrying	$\{0\}$	$\{0\}$	$\{0,1,3,6\}$	$\{0\}$
Push/Pull	$\{0\}$	$\{0\}$	$\{0,1,3,6\}$	$\{0.2,0.4,0.6,0.8,1.0\}$
Door Opening	$\{2,10\}$	$\{0,3,9,15\}$	$\{0,1,3,6\}$	$\{0\}$
Door Opening w/ friction	$\{2,10\}$	$\{0,3,9,15\}$	$\{0,10,30,50\}$	$\{1\}$
Real-world				
Carrying	$\{0\}$	$\{0\}$	$\{0,0.5,1,1.2\}$	$\{0\}$
Push/Pull	$\{0\}$	$\{0\}$	$\{0,5,10,16\}$	$\{0.5\}$

TABLE I: Variations of the environment parameters used for training *gmL-MPC* and *1gmmMPC*. Units: $m(kg)$, $b(kg/s)$, $\kappa(kg/s^2)$.

Table I. The zero mass case represents the reference behavior for each task.

A. Exp. 1: Task Variability

In this experiment, the objective is to see how well the compared methods generalize over different tasks with increasing complexity. Three tasks are performed, carrying a $12kg$ cube, pushing a $12kg$ cube on a floor with $\mu = 0.5$ and opening a frictionless door with mass $m = 5kg$, damping $b = 2.5kg/s$, and stiffness $\kappa = 12kg/s^2$. Note that with each of the three tasks an additional unknown parameter is added to the problem. The results are presented in Fig. 3, where Fig. 3a shows the Root Mean Square Error (RMSE) of the robot arm's joint positions over time for the Carrying task, Fig. 3b the RMSE of the joint velocities of the mobile base over time for the Push task, and Fig. 3c the door angle error over time for the Door Opening task. As it can be seen, *gmL-MPC* behaves similar to *standardMPC* in the beginning, until it selects the correct task model online. After that, it smoothly converges to minimal error, in contrast to the other approaches which mostly perform worse. *standardMPC* has a significant steady state error for all tasks, because of its inherently lack of knowledge of the real dynamic model. *acMPC* works well for lower weights, as in the Door Opening task, but continuously overshoots before convergence in the Carrying task, while in the Pushing



(a) (b) (c) (d)

Fig. 4: Results from pushing (left) and door opening (right) over different previously unseen environments given in Table II. The top images show the weights assigned to the trained models during GMM selection, while the bottom ones show the average RMSE of the base joint's velocity and of the door angle, respectively.

task the adaptation rate is very slow. This indicates that task-specific tuning of the control and adaptation gains is required for better results, something that can be avoided with the proposed *gmL-MPC* method. Finally, *1gmmMPC* smoothly converges in the Carrying task, but is unstable in the remaining tasks. The reason is that the prediction model cannot distinguish well between the different tasks. Similar inputs of different tasks might lead to different model predictions, which means that one large GMM is not suitable when dealing with large task variations.

B. Exp. 2: Increasing Task Complexity

The second experiment is designed to show the behavior of the different AMPC methods over tasks where the

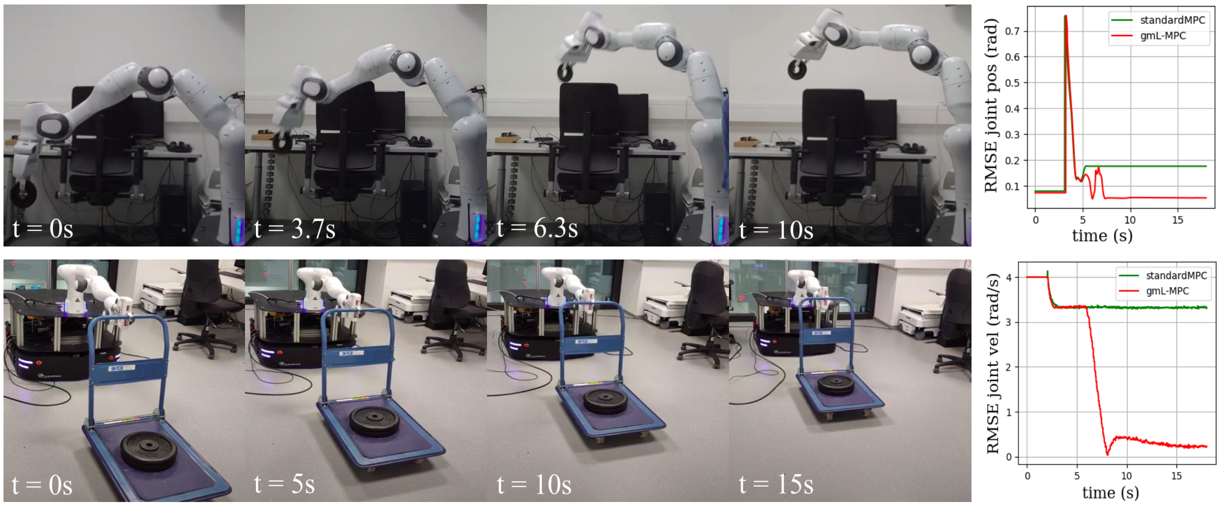


Fig. 5: Evaluation of *gmL-MPC* on the real robot and comparison with *standardMPC* using the Franka arm on top of the mobile base to perform unknown tasks: (top) Carrying $1.23kg$ weight plate, (bottom) Pull $15kg$ handcart.

measured errors have very similar responses with different combination of environment parameters, which makes them hard to observe. Two tasks are considered, pushing a cube with unknown weight and unknown friction, and opening of a door with known friction. For each case, the models are trained only for the corresponding task. Thus, *1gmmMPC* is trained only for pushing or for door opening, while *gmL-MPC* is similarly only given the respective models from the training phase. Each task is repeated for 5 different unknown environments with the parameters that are given in Table II.

The results are shown in Fig. 4. In the top images, one can see the output of the GMM selection step (task likelihood), while the bottom ones show the average RMSE of the corresponding task objective, which are the same as in experiment 1. As expected, *standardMPC* has always the highest error, while *acMPC* shows a better behavior in the pushing than the door scenario, proving again that different tuning is required for better outcome at each case. The rest of the approaches have much better behavior in both cases. However, it is clear from Fig. 4a and 4b that the model selection fails to accurately predict the correct task with the most similar parameters. Nevertheless, in both cases *gmL-MPC* keeps the RMSE low enough (see Fig. 4c and 4d), because the steady state prediction of the wrong model is very close to the real one. Notice that, even though *1gmmMPC* has the best performance in the pushing scenario, as the task complexity increases when adding door damping and stiffness, *gmL-MPC* outperforms the other approaches.

C. Exp. 3: Real-World Evaluation

In the last experiment, the proposed approach is evaluated on the real system and compared against *standardMPC*, for carrying a $1.23kg$ weight plate and pulling a $15kg$ handcart, both of which are unknown to the system. The performance metrics for each task are the same as in experiment 1. The results are given in Fig. 5. Notice that the behavior of *gmL-MPC* is not as smooth as in simulation. This is due to imperfect knowledge of the Franka arm model, the response

	E_1	E_2	E_3	E_4	E_5
push	$m=17$ $\mu=0.25$	$m=8$ $\mu=0.37$	$m=12$ $\mu=0.52$	$m=10$ $\mu=0.7$	$m=5$ $\mu=0.9$
door	$m=42$ $b=3$ $\kappa=2$ $\mu=1$	$m=33$ $b=6$ $\kappa=5$ $\mu=1$	$m=25$ $b=8$ $\kappa=7$ $\mu=1$	$m=20$ $b=9$ $\kappa=12$ $\mu=1$	$m=15$ $b=11$ $\kappa=14$ $\mu=1$

TABLE II: Unknown environment parameters for experiment 2. Units: $m(kg)$, $b(kg/s)$, $\kappa(kg/s^2)$.

of the mobile base, and hardware limitations related to safety. However, it can be seen that *gmL-MPC* behaves similarly to *standardMPC* in both cases until it selects the appropriate task model and corrects the errors accordingly. After that, it outperforms *standardMPC*.

IV. CONCLUSION AND OUTLOOK

In this paper, a novel AMPC framework for mobile manipulation, combining GMMs and MPC, is proposed. Training of the task models is extremely simple, as only measurement errors from proprioceptive sensors and known environmental parameters are required. By weighting the likelihood of those models, it is possible to predict multiple unknown environment parameters in a hierarchical manner using weighted average for some and GMR for others, separating them to simplify the prediction problem. It has been shown that the approach, which we called *gmL-MPC*, outperforms standard MPC and Adaptive Control-based AMPC, in completely unrelated tasks, like carrying, pushing, and opening doors. The approach was proven to provide smoother control signals and converge faster to a minimal error in contrast to other methods, which makes it more suitable to use in general purpose tasks.

In future, we are planning to research strategies to combine the available data more efficiently depending on the observed task and create more accurate models for each individual case. In addition, we plan to apply the method to assembly tasks in real industrial manufacturing environments.

REFERENCES

[1] S. Katayama, M. Murooka, and Y. Tazaki, "Model predictive control of legged and humanoid robots: models and algorithms," *Advanced Robotics*, vol. 37, no. 5, pp. 298–315, 2023. [Online]. Available: 10.1080/01691864.2023.2168134

[2] M. V. Minniti, R. Grandia, K. Fäh, F. Farshidian, and M. Hutter, "Model predictive robot-environment interaction control for mobile manipulation tasks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1651–1657. [Online]. Available: 10.1109/ICRA48506.2021.9562066

[3] S. Xu, L. Zhu, and C. P. Ho, "Learning efficient and robust multi-modal quadruped locomotion: A hierarchical approach," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4649–4655. [Online]. Available: 10.1109/ICRA46639.2022.9811640

[4] M. V. Minniti, R. Grandia, F. Farshidian, and M. Hutter, "Adaptive clf-mpc with application to quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 565–572, 2022. [Online]. Available: 10.1109/LRA.2021.3128697

[5] T. Anne, J. Wilkinson, and Z. Li, "Meta-learning for fast adaptive locomotion with uncertainties in environments and robot dynamics," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4568–4575. [Online]. Available: 10.1109/IROS51168.2021.9635840

[6] I. Lenz, R. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control," 07 2015. [Online]. Available: 10.15607/RSS.2015.XI.012

[7] E. Arcari, M. V. Minniti, A. Scampicchio, A. Carron, F. Farshidian, M. Hutter, and M. N. Zeilinger, "Bayesian multi-task learning mpc for robotic mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3222–3229, 2023. [Online]. Available: 10.1109/LRA.2023.3264758

[8] Y. Sun, W. L. Ubellacker, W.-L. Ma, X. Zhang, C. Wang, N. V. Csomay-Shanklin, M. Tomizuka, K. Sreenath, and A. D. Ames, "Online learning of unknown dynamics for model-based controllers in legged locomotion," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8442–8449, 2021. [Online]. Available: 10.1109/LRA.2021.3108510

[9] M. Sombolestan and Q. Nguyen, "Hierarchical adaptive loco-manipulation control for quadruped robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 156–12 162. [Online]. Available: 10.1109/ICRA48891.2023.10160523

[10] S. J-JE and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987. [Online]. Available: 10.1177/027836498700600303

[11] A. Pandala, R. T. Fawcett, U. Rosolia, A. D. Ames, and K. A. Hamed, "Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced- and full-order models," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6622–6629, 2022. [Online]. Available: 10.1109/LRA.2022.3176105

[12] Y. Wang, H. Kusano, and T. Sugihara, "Transporting a heavy object on a frictional floor by a mobile manipulator based on adaptive mpc framework," in *2021 IEEE/SICE International Symposium on System Integration (SII)*, 2021, pp. 807–812. [Online]. Available: 10.1109/IEEECONF49454.2021.9382761

[13] S. Xu, L. Zhu, H.-T. Zhang, and C. P. Ho, "Robust convex model predictive control for quadruped locomotion under uncertainties," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4837–4854, 2023. [Online]. Available: 10.1109/TRO.2023.3299527

[14] J. Xu, Y. Zheng, X. Jiang, S. Yang, L. Xiang, and Z. Zhang, "Real-time inertial parameter identification of floating-base robots through iterative primitive shape division," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5960–5966. [Online]. Available: 10.1109/ICRA46639.2022.9812214

[15] G. Tournois, M. Focchi, A. Del Prete, R. Orsolino, D. G. Caldwell, and C. Semini, "Online payload identification for quadruped robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4889–4896. [Online]. Available: 10.1109/IROS.2017.8206367

[16] M. Li, J. Zhang, G. Zuo, G. Feng, and X. Zhang, "Assist-as-needed control strategy of bilateral upper limb rehabilitation robot based on gmm," *Machines*, vol. 10, no. 2, 2022. [Online]. Available: 10.3390/machines10020076

[17] H. O and G. W., "Probabilistic balance monitoring for bipedal robots," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 245–256, 2009. [Online]. Available: 10.1177/0278364908095170

[18] D. M. Steinberg, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Towards autonomous habitat classification using gaussian mixture models," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4424–4431. [Online]. Available: 10.1109/IROS.2010.5652480

[19] A. Maligo and S. Lacroix, "Classification of outdoor 3d lidar data based on unsupervised gaussian mixture models," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 5–16, 2017. [Online]. Available: 10.1109/TASE.2016.2614923

[20] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020. [Online]. Available: 10.1126/scirobotics.abc5986

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977. [Online]. Available: http://www.jstor.org/stable/2984875

[22] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. [Online]. Available: 10.1109/ICRA40945.2020.9196673

[23] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019. [Online]. Available: 10.1109/SII.2019.8700380