2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Conservative to Confident: Treating Uncertainty Robustly Within Learning-Based Control

Chris J. Ostafew, Angela P. Schoellig, and Timothy D. Barfoot

***Abstract*— Robust control maintains stability and performance for a fixed amount of model uncertainty but can be conservative since the model is not updated online. Learning-based control, on the other hand, uses data to improve the model over time but is not typically guaranteed to be robust throughout the process. This paper proposes a novel combination of both ideas: a robust Min-Max Learning-Based Nonlinear Model Predictive Control (MM-LB-NMPC) algorithm. Based on an existing LB-NMPC algorithm, we present an efficient and robust extension, altering the NMPC performance objective to optimize for the worst-case scenario. The algorithm uses a simple *a priori* vehicle model and a learned disturbance model. Disturbances are modelled as a Gaussian Process (GP) based on experience collected during previous trials as a function of system state, input, and other relevant variables. Nominal state sequences are predicted using an Unscented Transform and worst-case scenarios are defined as sequences bounding the 3σ confidence region. Localization for the controller is provided by an on-board, vision-based mapping and navigation system enabling operation in large-scale, GPS-denied environments. The paper presents experimental results from testing on a 50 kg skid-steered robot executing a path-tracking task. The results show reductions in maximum lateral and heading path-tracking errors by up to 30% and a clear transition from robust control when the model uncertainty is high to optimal control when model uncertainty is reduced.**

Fig. 1. A Clearpath Husky robot autonomously negotiating challenging terrain with side slopes, inclines, and variable wheel traction. In practice, simple vehicle models rarely represent reality and limit the performance and stability of model-based, path-tracking controllers. In this work, we present a robust learning-based controller that automatically transitions from robust to optimal control throughout the process of learning to track a path.

I. Introduction

High-performance, path-tracking controllers for outdoor mobile robots require techniques to mitigate the effects of unknown surface materials, terrain topography, and complex robot dynamics. However, finding rich, accurate models *a priori* is difficult because (i) the terrain is often not known ahead of time, (ii) robot-terrain interaction models often do not exist, and (iii) even if such models did exist, finding corresponding model parameters is cumbersome.

Learning controllers alleviate the need for significant engineering work identifying and modelling all disturbances prior to operation by enabling the robot to acquire and apply experience *in situ* [1, 2]. In previous work, we presented a non-parametric, Learning-Based Nonlinear Model Predictive Control (LB-NMPC) algorithm [3] to reduce path-tracking errors within the context of an on-board, real-time, Visual Teach and Repeat (VT&R) mapping and navigation system [4]. In this work, we extend the algorithm by investigating a robust Min-Max LB-NMPC (MM-LB-NMPC) algorithm. The combined robust, learning controller merges the best of both worlds: robust, conservative control during initial trials when model uncertainty is high, converging to optimal control during later trials when model uncertainty is reduced.

The learning algorithm is based on a process model composed of two components: (i) a unicycle model representing the kinematics of the robot, and (ii) a learned disturbance model representing both unmodelled robot dynamics and systematic environmental disturbances. We model disturbances as a Gaussian Process (GP) [5] based on observations gathered during previous path traversals as a function of system state, input, and other relevant system variables. By modelling the disturbances as a GP, the algorithm is able to predict both the mean and uncertainty of disturbances affecting the *a priori* process model. We use an Unscented Transform [6] to efficiently compute the mean and variance of the nominal state sequence given the two-component, learned, stochastic model. The MM-LB-NMPC cost function is optimized for the worst-case sequence bounding the nominal 3σ confidence region. We demonstrate the robust, learning control algorithm on a 50 kg Clearpath Husky robot and show reductions of worst-case path-tracking errors by up to 30% and a clear transition from robust towards optimal control with only a 5% increase in computation time.

The key characteristics of this work are: (i) a path-tracking, robust MM-LB-NMPC algorithm based on a fixed, *a priori* known kinematic process model and a learned GP disturbance model, (ii) efficient prediction of nominal

The authors are with the University of Toronto Institute for Aerospace Studies, Toronto, Ontario, Canada. Email: chris.ostafew@mail.utoronto.ca, schoellig@utias.utoronto.ca, and tim.barfoot@utoronto.ca.

and 3σ bounding sequences using an Unscented Transform, (iii) navigation based on vision only, and (iv) experimental results on a 50 kg robot. To our knowledge, this paper is the first to demonstrate a robust MM-LB-NMPC algorithm that automatically transitions from robust to optimal control as model uncertainty varies with experience.

II. Related Work

MPC is a framework in which the current control action is obtained by solving, at each sampling instant, a finite-horizon optimal control problem using the current state of the plant as the initial state [7, 8]. Among a growing list of examples, MPC has been demonstrated in several real-world applications on ground robots [9–14]. However, in each of these examples, the system model is fixed and assumed to represent the system accurately. As a result, these controllers achieve stability and good performance in operating regimes limited in part by the *a priori* model and tuning parameters. In contrast, our algorithm simultaneously includes the ability to learn from experience and robustness to model uncertainty, thus enabling reliable operation on robots of vastly different masses and in a variety of terrains.

Learning-based control aims to improve performance over time by correcting the system model using experience (i.e., past measurements) [15–17]. Kocijan et al. [15] presents a LB-MPC algorithm for a simulated pH neutralization process. In addition to tracking errors and control input, the cost function penalized model uncertainty resulting in a controller that avoided uncertain states. In contrast, our Min-Max approach uses the model uncertainty for robust control, maintaining performance and stability despite model uncertainty. Lehnert and Wyeth [16], and Park et al. [17] present LB-MPC algorithms for an elastic joint manipulator and an omni-directional mobile robot, respectively. In each of these cases, the controllers considered only the mean predicted disturbance. Our approach considers both the learned mean and variance, enabling automatic shifts between robust and optimal control as model uncertainty varies. Finally, Tanaskovic et al. [18] present robust adaptive MPC for constrained systems. While adaptive control is only capable of *reacting* to modelling errors, our approach is based on a learned disturbance model and can therefore act in anticipation of repeatable disturbances.

Min-Max MPC maintains controller stability and performance despite model uncertainty by optimizing the performance objective for a worst-case scenario [19–24]. Scokaert and Mayne [21] present a Min-Max algorithm for robust performance of systems with bounded disturbances. In contrast, we assume normally-distributed disturbances and use an Unscented Transform [6] to predict a nominal sequence, 3σ confidence region, and the associated boundary scenarios for the Min-Max algorithm. Bemporad et al. [22] and Kerrigan and Maciejowski [23] present algorithms that reduce the computation time of Min-Max MPC. In our work, we derive worst-case scenarios from the 3σ confidence region surrounding the predicted nominal sequence, representing a small increase in computation relative to our original learning algorithm [3]. Raimondo et al. [24] present a nonlinear Min-Max algorithm that separates state-dependent and state-independent disturbances to reduce conservativeness. In contrast, we reduce conservativeness over time by learning an improved nominal process model. Effectively, our controller naturally transitions to an optimal controller as model uncertainty decreases. To our knowledge, our work is the first to propose a MM-LB-NMPC algorithm.

Scenario MPC is a technique similar to Min-Max MPC [25–27]. However, instead of identifying a relatively small number of worst-case disturbance sequences, Scenario MPC relies on a (typically) large number of randomly sampled state sequences over the prediction horizon given the model uncertainty. Unlike Scenario MPC, our algorithm relies on a small number of worst-case scenarios bounding the nominal 3σ confidence region. This enables online operation and integration into our existing LB-NMPC algorithm.

Otherwise, Berkenkamp and Schoellig [28] combine robust control with machine learning techniques to adapt the model uncertainty over time. While they present a learning, robust controller to stabilize an operating point, we derive a controller for path-tracking. Aswani et al. [29] present a robust, linear LB-MPC algorithm that guarantees performance and stability by placing tube-shaped constraints on predicted sequences. In this work, we use Min-Max MPC, a less conservative approach to robust MPC, optimizing for worst-case scenarios.

III. Visual Teach & Repeat

Localization for the controller is provided by an on-board Visual Teach & Repeat (VT&R) mapping and navigation algorithm developed by Furgale and Barfoot [4] where the sole sensor is an on-board stereo camera. In the first operational phase, the teach phase, the robot is piloted along the desired path. Localization in this initial phase is obtained relative to the robot's starting position by visual odometry (VO). In addition to the VO pipeline, path vertices are defined at short and regular intervals along the path while simultaneously storing key frames composed of local feature descriptors and their 3D positions. During the repeat phase, the VT&R algorithm estimates the pose of the robot relative to the nearest path vertex by re-localizing against the stored key frames. Re-localization is achieved by matching feature descriptors to generate feature tracks between the current robot view and the teach-pass robot view. As long as sufficient correct feature matches are made, the system generates consistent localization over trials and is able to support a learning control algorithm.

IV. Mathematical Formulation

A. MM-LB-NMPC Overview

NMPC finds a sequence of control inputs that optimizes the plant behavior over a prediction horizon based on the current state. The first control input in the optimal sequence is then applied to the system, resulting in a new system state. The entire process is then repeated at the next sample time for the new system state.

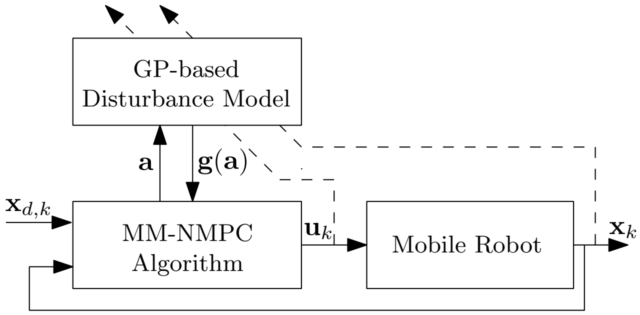

Fig. 2. The controller is composed of two components: 1) the robust, path-tracking, Min-Max NMPC algorithm, and 2) the GP-based Disturbance Model, providing experience-based disturbance estimates.

In traditional NMPC implementations, the process model is specified *a priori* and remains unchanged during operation. In our previous work on LB-NMPC, we augmented a simple process model with the mean of an experience-based disturbance model. Effectively, the controller used experience to reduce path-tracking errors, compensating for effects not captured by the simple process model. In this work, we incorporate both the disturbance mean and uncertainty into the NMPC algorithm, resulting in an efficient robust extension that reduces the worst-case errors (Fig. 2). We consider a stochastic, learned process model,

$$\mathbf{x}_{k+1} = \overbrace{\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}^{\textit{a priori} \text{ model}} + \overbrace{\mathbf{g}(\mathbf{a}_k)}^{\text{learned model}} , \quad (1)$$

with Gaussian system state, $\mathbf{x}_k \sim \mathcal{N}(\bar{\mathbf{x}}_k, \mathbf{\Sigma}_k) \in \mathbb{R}^n$, disturbance dependency, $\mathbf{a}_k \in \mathbb{R}^p$, and control input, $\mathbf{u}_k \in \mathbb{R}^m$, all at time k. The models $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are nonlinear models: $\mathbf{f}(\cdot)$ is a simple, *a priori* vehicle model and $\mathbf{g}(\cdot)$ is an (initially unknown) disturbance model representing discrepancies between the nominal model and the actual system behavior. Disturbances are modelled as a Gaussian Process (Sec. IV-C), thus $\mathbf{g}(\cdot)$ is normally distributed, $\mathbf{g}(\cdot) \sim \mathcal{N}(\boldsymbol{\mu}(\cdot), \mathbf{\Sigma}_{\text{gp}}(\cdot))$. In our previous work, we showed that $\mathbf{g}(\cdot)$ could be used to learn higher-order dynamics by including historic states in the disturbance dependency [3]. However, for simplicity, we assume for now that $\mathbf{a}_k = (\bar{\mathbf{x}}_k, \mathbf{u}_k)$.

As previously mentioned, the goal of NMPC is to find a set of controls that optimizes the plant behavior over a given prediction horizon. To this end, we define the cost function to be minimized over the next K time-steps as

$$J(\check{\mathbf{x}}, \mathbf{u}) := (\mathbf{x}_d - \check{\mathbf{x}})^T \mathbf{Q} \, (\mathbf{x}_d - \check{\mathbf{x}}) + \mathbf{u}^T \mathbf{R} \, \mathbf{u}, \quad (2)$$

where $\mathbf{Q}$ is positive semi-definite, $\mathbf{R}$ is positive definite, $\mathbf{u}$ is a sequence of inputs, $\mathbf{u} = (\mathbf{u}_k, \ldots, \mathbf{u}_{k+K})$, $\mathbf{x}_d$ is a sequence of desired states, $\mathbf{x}_d = (\mathbf{x}_{d,k+1}, \ldots, \mathbf{x}_{d,k+K+1})$, and $\check{\mathbf{x}}$ is a sequence of predicted states, $\check{\mathbf{x}} = (\check{\mathbf{x}}_{k+1}, \ldots, \check{\mathbf{x}}_{k+K+1})$. Previously, the objective was optimized for the mean of the nominal sequence, $\check{\mathbf{x}} = \bar{\mathbf{x}}$, where $\mathbf{x}_{\text{nom}} = \{\{\bar{\mathbf{x}}_{i+1}, \mathbf{\Sigma}_{i+1}\} | i = k, \ldots, k+K\}$. In this work, the objective is optimized for the worst-case sequence given the uncertainty in the learned model. Specifically, the nominal sequence is predicted using an Unscented Transform and 2^n worst-case scenarios, $\check{\mathbf{x}}^{(l)}$, $l \in \{1, \ldots, 2^n\}$, are defined in Sec. IV-B as sequences bounding the nominal 3σ confidence region. Finally, the optimal control sequence is given by

$$\mathbf{u}_{\text{opt}} = \arg\min_{\mathbf{u}} \max_{l} J(\check{\mathbf{x}}^{(l)}, \mathbf{u}). \quad (3)$$

Algorithm 1: MM-LB-NMPC

Data: $\mathbf{x}_d$, $\{\bar{\mathbf{x}}_k, \mathbf{\Sigma}_k\}$, and $\mathbf{u}_{\text{init}}$
Result: $\mathbf{u}_{\text{opt}}$
1 initialization: $\tilde{\mathbf{u}} = \mathbf{u}_{\text{init}}$;
2 **while** $\|\delta\mathbf{u}\| > \alpha$ **do**
3 Compute $\mathbf{x}_{\text{nom}}$ given $\tilde{\mathbf{u}}$ and (1);
4 Compute boundary sequences **(NEW, cf. [3])**;
5 Find worst-case boundary sequence **(NEW, cf. [3])**;
6 Linearize (2) around worst-case sequence and solve for $\delta\mathbf{u}$;
7 Update control, $\tilde{\mathbf{u}} \leftarrow \tilde{\mathbf{u}} + \delta\mathbf{u}$;

Since both our process model and disturbance model are nonlinear, the optimal control sequence, $\mathbf{u}_{\text{opt}}$, is found iteratively (Alg. 1) using a nonlinear optimization technique. In this paper, we use unconstrained Gauss-Newton minimization [30]. However, there are other nonlinear optimization algorithms, such as the constrained Gauss-Newton algorithm [31], that could be used to incorporate constraints on states and control inputs.

At each time-step, we begin with the current system state, $\{\bar{\mathbf{x}}_k, \mathbf{\Sigma}_k\} = \{\hat{\mathbf{x}}_k, \hat{\mathbf{\Sigma}}_k\}$, provided by the vision-based localization system, and an initial guess for the optimal control input sequence, $\tilde{\mathbf{u}}$, such as the sequence of optimal inputs computed in the previous time-step (Alg. 1, Step 1). We compute the worst-case boundary sequence (Sec. IV-B, Alg. 1, Steps 3-5) based on $\tilde{\mathbf{u}}$, (1), and

$$l^* = \arg\max_{l \in \mathcal{L}} J(\check{\mathbf{x}}^{(l)}, \tilde{\mathbf{u}}). \quad (4)$$

We linearize (2) around the worst-case sequence with $\mathbf{u} = \tilde{\mathbf{u}} + \delta\mathbf{u}$, and $\check{\mathbf{x}}^{(l^*)} = \tilde{\mathbf{x}}^{(l^*)} + \delta\mathbf{x}^{(l^*)}$ (Alg. 1, Step 6). We write a linearized equation for the state,

$$\delta\mathbf{x}^{(l^*)} = \mathbf{H} \, \delta\mathbf{u}, \quad (5)$$

where $\mathbf{H}$ is the block-Jacobian of (1) with respect to $\mathbf{u}$. Substituting (5), $\check{\mathbf{x}}^{(l^*)} = \tilde{\mathbf{x}}^{(l^*)} + \delta\mathbf{x}^{(l^*)}$, and $\mathbf{u} = \tilde{\mathbf{u}} + \delta\mathbf{u}$ into (2) results in $J(\cdot)$ being quadratic in $\delta\mathbf{u}$. We can easily find the value of $\delta\mathbf{u}$ that minimizes $J(\check{\mathbf{x}}^{(l^*)}, \mathbf{u})$, update our control input (Alg. 1, Step 7),

$$\tilde{\mathbf{u}} \leftarrow \tilde{\mathbf{u}} + \delta\mathbf{u}, \quad (6)$$

and iterate to convergence, $\|\delta\mathbf{u}\| < \alpha$, with tuned value, α. In accordance with NMPC, we apply the resulting control input for one time-step and start all over at the next time-step.

The addition of the '$\max$' in (3) is the extension from our previous work [3]: the algorithm now takes into account the worst-case boundary sequence, limiting the worst-case errors and guaranteeing stability for large model uncertainties. Moreover, there is an automatic transition as uncertainty decreases, from robust control, with an uncertain model, to optimal control, with a rich and accurate model.

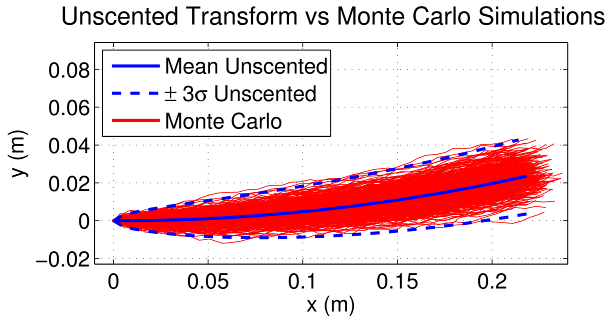

Fig. 3. Here we show the lateral 3σ boundaries of the mean sequence. Since our model uncertainty is normally distributed, we use an Unscented Transform to compute the mean and uncertainty of a predicted state sequence. However, unlike Scenario MPC, where many scenarios are sampled in order to find boundary sequences (red), the Unscented Transform efficiently estimates the 3σ confidence region (dashed blue).

B. Computing Worst-Case Sequences

In our previous work, the cost function is optimized based on the mean of the nominal state sequence. In this work, the cost function is optimized for the worst-case sequence bounding the nominal 3σ confidence region. Worst-case sequences are computed in two steps. First, the nominal state sequence is computed (Alg. 1, Step 3). Second, the worst-case sequences are extracted from the nominal sequence (Alg. 1, Step 4).

Since $\mathbf{x}_k$ is normally-distributed and (1) is nonlinear, we use an Unscented Transform [6] to iteratively predict the nominal state sequence, $\mathbf{x}_{\text{nom}}$, given $\mathbf{u}$, $\{\bar{\mathbf{x}}_k, \boldsymbol{\Sigma}_k\}$, and (1). We define an initial state, $\mathbf{z}_k := (\bar{\mathbf{x}}_k, \boldsymbol{\mu}(\mathbf{a}_k)) \in \mathbb{R}^{2n}$, with uncertainty, $\mathbf{P}_k := \text{diag}(\boldsymbol{\Sigma}_k, \boldsymbol{\Sigma}_{\text{gp}}(\mathbf{a}_k))$. We compute $4n+1$ sigma points, $\mathcal{Z}_{k,i} := (\mathcal{X}_{k,i}, \mathcal{M}_{k,i})$, where $\mathcal{X}_{k,i}$ and $\mathcal{M}_{k,i}$ are the sigma points of $\mathbf{x}_k$ and $\boldsymbol{\mu}(\mathbf{a}_k)$,

$$\mathcal{Z}_{k,0} := \mathbf{z}_k \tag{7}$$

$$\mathcal{Z}_{k,i} := \mathbf{z}_k + \sqrt{2n+\gamma}\,\text{col}_i\,\mathbf{S}_k, \quad i = 1\ldots 2n \tag{8}$$

$$\mathcal{Z}_{k,i+2n} := \mathbf{z}_k - \sqrt{2n+\gamma}\,\text{col}_i\,\mathbf{S}_k, \quad i = 1\ldots 2n \tag{9}$$

where $\mathbf{S}_k\mathbf{S}_k^T = \mathbf{P}_k$ with $\mathbf{S}_k$ derived from the Cholesky decomposition of $\mathbf{P}_k$, $\text{col}_i\,\mathbf{S}_k$ is the ith column of $\mathbf{S}_k$, and γ is a tuning parameter. The sigma points are then passed through the nonlinear model,

$$\mathcal{X}_{k+1,i} := \mathbf{f}(\mathcal{X}_{k,i}, \mathbf{u}_k) + \mathcal{M}_{k,i}, \quad i \in \{0, \ldots, 4n\}, \tag{10}$$

where $\mathbf{f}(\cdot)$ is our *a priori* vehicle model. We combine the sigma points into the predicted mean and uncertainty,

$$\bar{\mathbf{x}}_{k+1} := \frac{1}{2n+\gamma}\left(\gamma\mathcal{X}_{k+1,0} + \frac{1}{2}\sum_{i=1}^{4n}\mathcal{X}_{k+1,i}\right) \tag{11}$$

$$\boldsymbol{\Sigma}_{k+1} := \frac{1}{2n+\gamma}\left(\gamma(\mathcal{X}_{k+1,0} - \bar{\mathbf{x}}_{k+1})(\mathcal{X}_{k+1,0} - \bar{\mathbf{x}}_{k+1})^T + \frac{1}{2}\sum_{i=1}^{4n}(\mathcal{X}_{k+1,i} - \bar{\mathbf{x}}_{k+1})(\mathcal{X}_{k+1,i} - \bar{\mathbf{x}}_{k+1})^T\right). \tag{12}$$

This process is repeated K times, until the complete nominal sequence, $\mathbf{x}_{\text{nom}}$, is generated. Finally, we compute 2^n boundary sequences (Alg. 1, Step 4) based on $\mathbf{x}_{\text{nom}}$ while assuming 3σ noise (Fig. 3). Assuming for now $n=3$, and defining $\boldsymbol{\sigma}_k := (\sqrt{\boldsymbol{\Sigma}_k(1,1)}, \ldots, \sqrt{\boldsymbol{\Sigma}_k(3,3)})$, and

$$\begin{aligned}
\boldsymbol{\Gamma}^{(1)}_{\text{sign}} &= \text{diag}(1,1,1), & \boldsymbol{\Gamma}^{(5)}_{\text{sign}} &= \text{diag}(-1,1,1),\\
\boldsymbol{\Gamma}^{(2)}_{\text{sign}} &= \text{diag}(1,1,-1), & \boldsymbol{\Gamma}^{(6)}_{\text{sign}} &= \text{diag}(-1,1,-1),\\
\boldsymbol{\Gamma}^{(3)}_{\text{sign}} &= \text{diag}(1,-1,1), & \boldsymbol{\Gamma}^{(7)}_{\text{sign}} &= \text{diag}(-1,-1,1),\\
\boldsymbol{\Gamma}^{(4)}_{\text{sign}} &= \text{diag}(1,-1,-1), & \boldsymbol{\Gamma}^{(8)}_{\text{sign}} &= \text{diag}(-1,-1,-1),
\end{aligned} \tag{13}$$

then $\check{\mathbf{x}}^{(l)}_{i+1} = \bar{\mathbf{x}}_{i+1} + \boldsymbol{\Gamma}^{(l)}_{\text{sign}}\,3\boldsymbol{\sigma}_{i+1}$, $i \in \{k, \ldots, k+K\}$.

C. Gaussian Process Disturbance Model

We model the disturbance, $\mathbf{g}(\cdot)$, as a GP based on past observations. Since we provided a detailed explanation of the model in previous work [3], here we provide only a high-level sketch. The learned model depends on observations of disturbances collected during previous trials. At time k, we use the estimated poses, $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{k-1}$, from the VT&R system (Sec. III), the disturbance dependency, $\mathbf{a}_{k-1}$, and the control input, $\mathbf{u}_{k-1}$, to solve (1) for $\hat{\mathbf{g}}(\mathbf{a}_{k-1})$,

$$\hat{\mathbf{g}}(\mathbf{a}_{k-1}) = \hat{\mathbf{x}}_k - \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}). \tag{14}$$

The resulting data pair, $\{\mathbf{a}_{k-1}, \hat{\mathbf{g}}(\mathbf{a}_{k-1})\}$, represents an individual experience. We collect all experiences into one large dataset, $\mathcal{D}$, with generally N observations, and drop the time-step index on each data pair in $\mathcal{D}$, so that when referring to $\mathbf{a}_{\mathcal{D},i}$ or $\hat{\mathbf{g}}_{\mathcal{D},i}$, we mean the ith pair of data in the superset $\mathcal{D}$.

In our work, we train a separate GP for each dimension in $\mathbf{g}(\cdot) \in \mathbb{R}^n$ to model disturbances as the robot travels along a path. For simplicity of discussion, we will assume for now that $n = 1$ and denote $\hat{\mathbf{g}}_{\mathcal{D},i}$ by $\hat{g}_{\mathcal{D},i}$. The GP model assumes a measured disturbance originates from a process model,

$$\hat{g}(\mathbf{a}_{\mathcal{D},i}) \sim \mathcal{GP}\left(0, k(\mathbf{a}_{\mathcal{D},i}, \mathbf{a}_{\mathcal{D},i})\right), \tag{15}$$

with zero mean and kernel function, $k(\mathbf{a}_{\mathcal{D},i}, \mathbf{a}_{\mathcal{D},i})$, to be defined. We assume that each disturbance measurement is corrupted by zero-mean additive noise with variance, σ_n^2, so that $\hat{g}_{\mathcal{D},i} = g_{\mathcal{D},i} + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Then a modelled disturbance, $g(\mathbf{a}_k)$, and the N observed disturbances, $\hat{\mathbf{g}} = (\hat{g}_{\mathcal{D},1}, \ldots, \hat{g}_{\mathcal{D},N})$, are jointly Gaussian,

$$\begin{bmatrix} \hat{\mathbf{g}} \\ g(\mathbf{a}_k) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k}(\mathbf{a}_k)^T \\ \mathbf{k}(\mathbf{a}_k) & k(\mathbf{a}_k, \mathbf{a}_k) \end{bmatrix}\right), \tag{16}$$

where

$$(\mathbf{K})_{i,j} = k(\mathbf{a}_{\mathcal{D},i}, \mathbf{a}_{\mathcal{D},j}), \quad \mathbf{K} \in \mathbb{R}^{N\times N}, \tag{17}$$

such that $(\mathbf{K})_{i,j}$ is the (i,j)th element of $\mathbf{K}$, and

$$\mathbf{k}(\mathbf{a}_k) = \begin{bmatrix} k(\mathbf{a}_k, \mathbf{a}_{\mathcal{D},1}) & \ldots & k(\mathbf{a}_k, \mathbf{a}_{\mathcal{D},N}) \end{bmatrix}.$$

In our case, we use the squared-exponential kernel function [5],

$$k(\mathbf{a}_i, \mathbf{a}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{a}_i - \mathbf{a}_j)^T\mathbf{M}^{-2}(\mathbf{a}_i - \mathbf{a}_j)\right) + \sigma_n^2\,\delta_{ij},$$

where δ_{ij} is the Kronecker delta, that is 1 iff $i = j$ and 0 otherwise, and the constants $\mathbf{M}$, σ_f, and σ_n are hyperparameters. In our implementation with $\mathbf{a}_k \in \mathbb{R}^p$, the

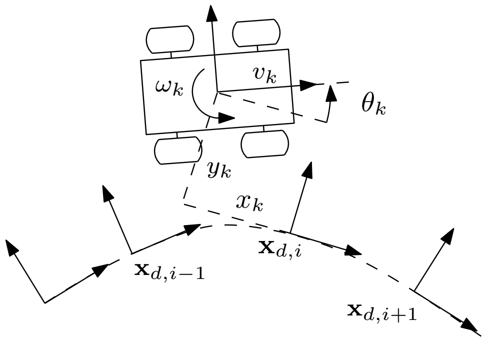

Fig. 4. Definition of the robot velocities, v_k and ω_k, and the three pose variables, x_k, y_k, and θ_k, calculated relative to the nearest path vertex by Euclidean distance.

constant $\mathbf{M}$ is a diagonal matrix, $\mathbf{M} = \mathrm{diag}(\mathbf{m})$, $\mathbf{m} \in \mathbb{R}^p$, representating the relevance of each component in $\mathbf{a}_k$, while the constants σ_f^2 and σ_n^2, represent the process variation and measurement noise, respectively. Finally, we have that the prediction, $g(\mathbf{a}_k)$, of the disturbance at an arbitrary state, $\mathbf{a}_k$, is also normally distributed,

$$g(\mathbf{a}_k)|\mathbf{g} \sim \mathcal{N}\bigg(\mathbf{k}(\mathbf{a}_k)\mathbf{K}^{-1}\hat{\mathbf{g}},\ k(\mathbf{a}_k,\mathbf{a}_k) - \mathbf{k}(\mathbf{a}_k)\mathbf{K}^{-1}\mathbf{k}(\mathbf{a}_k)^T\bigg).$$

Unlike our previous work, which used only the predicted mean of a disturbance in the model predictive controller, here we make use of both the predicted mean and variance. As detailed in Sec. IV-B, the variance of the disturbance is used to produce worst-case sequences that the controller should actively mitigate. During initial trials, when the model uncertainty is high, the predicted variance is large and the resulting control conservative. However, as the algorithm collects more experience, the predicted variance decreases, and the algorithm naturally transitions to an optimal controller.

V. Implementation

In our work, robots are modelled as unicycle-type vehicles with position, $\mathbf{x}_k = (x_k,\, y_k,\, \theta_k)$, calculated relative to the nearest path vertex by Euclidean distance, and velocity, $\mathbf{v}_k = (v_{\mathrm{act},k}, \omega_{\mathrm{act},k})$ (Fig. 4). The robots have two control inputs, their linear and angular velocities, $\mathbf{u}_k = (v_{\mathrm{cmd},k}, \omega_{\mathrm{cmd},k})$. The commanded linear velocity is set to a desired, scheduled speed at the nearest path vertex, leaving only the angular velocity, $\omega_{\mathrm{cmd},k}$, for the NMPC algorithm to choose (i.e., we do not optimize the commanded linear velocity, leaving it to be scheduled in advance [32]).

When the time between control signal updates is defined as Δt, the resulting *a priori* model used by the algorithm is

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k + \begin{bmatrix} \Delta t\, \cos\theta_k & 0 \\ \Delta t\, \sin\theta_k & 0 \\ 0 & \Delta t \end{bmatrix} \mathbf{u}_k, \qquad (18)$$

which represents a simple kinematic model for our robot; it does not account for dynamics or environmental disturbances. As described in our previous work [3], we use an extended disturbance dependency in practice, $\mathbf{a}_k = (\bar{\mathbf{x}}_k, \bar{\mathbf{v}}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k-1})$, enabling the algorithm to learn higher-order disturbances in addition to kinematics. Since our robot is not equipped with velocity sensors, we approximate $\mathbf{v}_k$ according to

$$v_{\mathrm{act},k-1} = \frac{\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}{\Delta t},$$

and

$$\omega_{\mathrm{act},k-1} = \frac{(\theta_k - \theta_{k-1})}{\Delta t}.$$

This is preferable to using wheel encoders because we want the true speeds with respect to the ground and wheel encoders are unable to measure slip. Because $\mathbf{x}_k$ comes from our vision-based localization system, we are able to measure wheel slip in this way.

VI. Experimental Results

A. Overview

We tested the MM-LB-NMPC algorithm on a 50 kg Clearpath Husky robot traveling at 0.5 m/s. The controller described in Sec. IV was implemented and run in addition to the VT&R software on a Lenovo W530 laptop with an Intel 2.6 Ghz Core i7 processor with 16 GB of RAM. The camera used for localization was a Point Grey Bumblebee XB3 stereo camera. The resulting real-time localization and path-tracking control signals were generated at approximately 10 Hz. Since GPS was not available, the improvement due to the MM-LB-NMPC algorithm was quantified by the localization of the VT&R algorithm.

B. Tuning Parameters

The performance of the system was primarily adjusted using the NMPC weighting matrices, $\mathbf{Q}$ and $\mathbf{R}$. We selected a 3:3:1 ratio balancing heading errors, position errors, and control inputs. Otherwise, the prediction horizon, $K = 10$, and the convergence criterion, $\alpha = 0.001\,K$, were selected to enable online operation. The Sigma-Point scaling paramter, $\gamma = 2$, was selected to enable accurate prediction of mean and uncertainty. Hyperparameters were set automatically by maximizing the log-likelihood of the measured disturbances [3]. Finally, path vertices were defined after each 0.2 m of travel or 3.5° of rotation during the VT&R teach phase to reduce discontinuities in estimated and desired state.

C. Results

Over five trials, the algorithm successfully reduced the maximum lateral and heading errors by up to 30%. Fig. 5 highlights the procession from robust control, when model uncertainty was high, to optimal control, when the system had acquired experience and model uncertainty was reduced. In practice, the learned model uncertainty never goes to zero due to measurement noise. As a result, the MM-LB-NMPC algorithm shifted towards optimal control but ultimately struck a balance between robust and optimal control over time (Fig. 5).

In general, the Min-Max algorithm incurred an increase in computation time of only 5-10%. This confirms our selection of an Unscented Transform (Sec. IV-B) as an

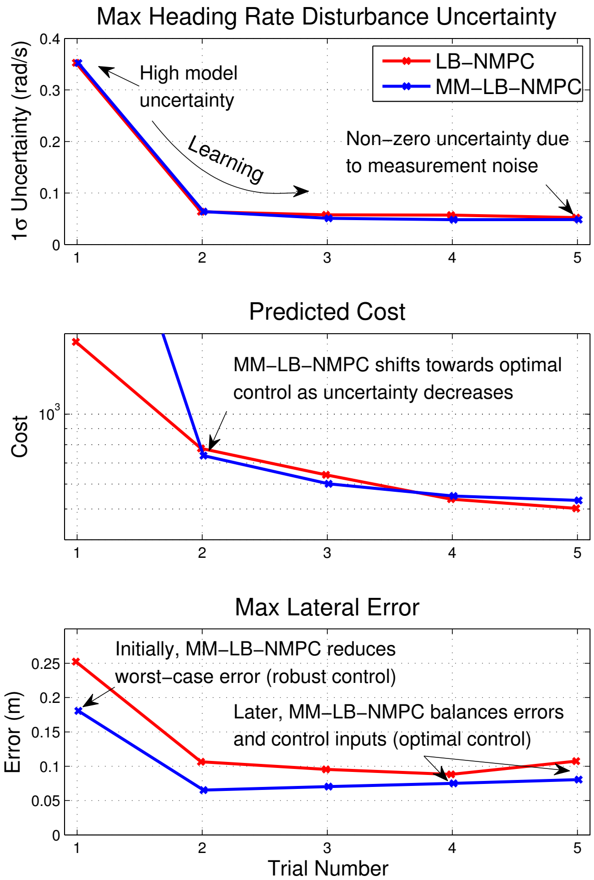

Fig. 5. Here we show the procession of model uncertainty (e.g, the maximum heading rate disturbance uncertainty), predicted costs, and maximum lateral error over several trials. The plots show the automatic transition between robust control, when uncertainty is high and maximum errors are reduced significantly, to optimal control, when model uncertainty is low and the controller finds a balance between errors and control inputs. Further, the algorithm reduces errors due to non-repetitive noise, such as measurement noise, that the learning algorithm is incapable of predicting.

efficient method of predicting the 3σ confidence region without resorting to generating a large number of scenarios.

As shown in Fig. 6, we tested on a short path highlighting the effect of the Min-Max algorithm. On this path, the measured disturbance affecting the heading rate of the robot peaked at approximately 0.5 rad/s, representing nearly 50% of the commanded input 7 m along the path (Fig. 7). The general trend observed is that path-tracking errors are reduced over the entire path (Fig. 8). However, the errors are not cancelled completely due to the optimality of the NMPC algorithm, which finds a balance between path-tracking errors and control inputs.

VII. Conclusion

In summary, this paper presents a novel, robust Min-Max Learning-Based Nonlinear Model Predictive Control (MM-LB-NMPC) algorithm. We derive an efficient and robust extension to an existing LB-NMPC algorithm, altering the performance objective to optimize for the worst-case scenario. The algorithm uses a simple *a priori* vehicle model and a learned disturbance model. Disturbances are modelled as a Gaussian Process (GP) based on experience collected during previous traversals as a function of system state, input and other relevant variables. Nominal state sequences are predicted using an Unscented Transform and worst-

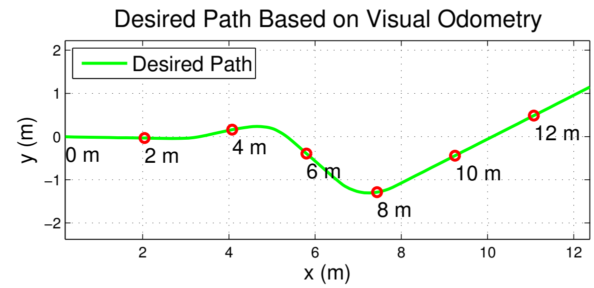

Fig. 6. The test path for the MM-LB-NMPC algorithm. A short, demonstrative path was selected to highlight the improvements due to the Min-Max algorithm.

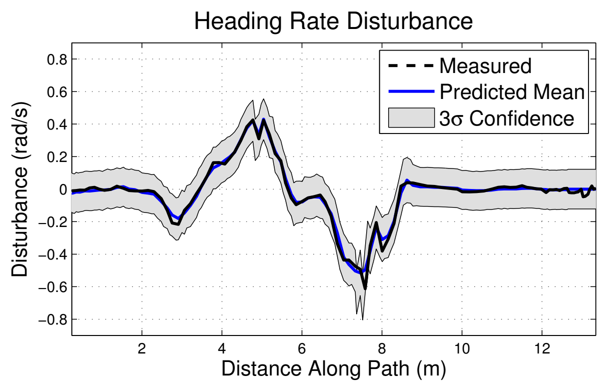

Fig. 7. Trial 5 heading rate disturbance vs distance. The 3σ bounds represent the model uncertainty used to compute the 3σ confidence region and associated boundary sequences (Sec. IV-B). The goal of the Min-Max algorithm is to reduce worst-case errors given the model uncertainty.

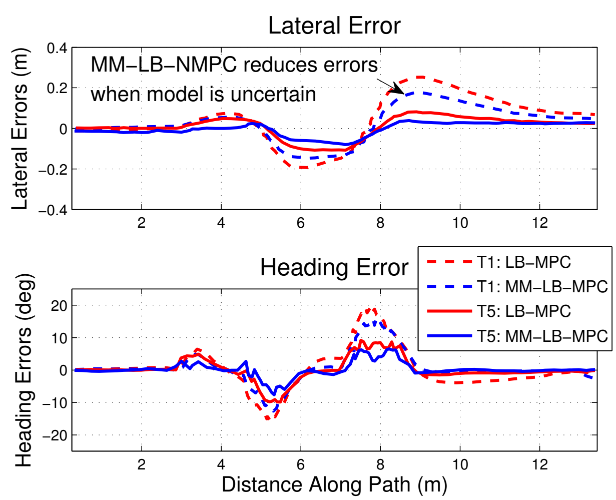

Fig. 8. Path-tracking errors vs distance for trials 1 (dashed) and 5 (solid). Reducing errors when model uncertainty is high (i.e., trial 1) is important for controller stability and perspective-dependent, vision-based localization algorithms. As model uncertainty decreases, the MM-LB-NMPC algorithm naturally transitions towards an optimal control, balancing tracking errors and control inputs.

case scenarios are defined as sequences bounding the 3σ confidence region.

Experimental results are provided from tests with a 50 kg Clearpath Husky robot on a demonstrative path. The results show reductions in maximum lateral and heading path-tracking errors by up to 30% and a clear transition from robust control reducing worst-case errors, when the model uncertainty is high, to optimal control balancing tracking errors and control inputs, when model uncertainty is reduced. Furthermore, the algorithm requires only a 5-10% increase in computation time relative to the learning algorithm.

In retrospect, the MM-LB-NMPC algorithm offers an effective and efficient method of simultaneously exploiting and decreasing model uncertainty to improve controller performance and guarantee stability. Leveraging this work, our future work focuses on using the 3σ confidence region to provide safety guarantees and real-time speed scheduling.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank the Ontario Ministry of Research and Innovation's Early Research Award Program for funding our research and Clearpath Robotics for funding the Husky A200 robot. This work was also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN).

REFERENCES

[1] S. Schaal and C. G. Atkeson. Learning Control in Robotics. *IEEE Robotics & Automation Magazine*, 17(2):20–29, 2010.

[2] D. Nguyen-Tuong and J. Peters. Model Learning for Robot Control: A Survey. *Cognitive Processing*, 12(4):319–340, 2011.

[3] C. Ostafew, A. P. Schoellig, and T. D. Barfoot. Learning-Based Nonlinear Model Predictive Control to Improve Vision-Based Mobile Robot Path-Tracking in Challenging Outdoor Environments. In *Proceedings of the International Conference on Robotics and Automation*, pages 4029–4036, 2014.

[4] P. Furgale and T. D. Barfoot. Visual Teach and Repeat for Long-Range Rover Autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010.

[5] C. E. Rasmussen. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[6] S. J. Julier and J. K. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[7] M. Morari and J. H. Lee. Model Predictive Control: Past, Present and Future. *Computers and Chemical Engineering*, 23(4):667–682, 1999.

[8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. *Automatica*, 36(6):789–814, 2000.

[9] F. Kühne, W. F. Lages, and J. M. G. Silva. Mobile Robot Trajectory Tracking using Model Predictive Control. In *Proceedings of the Latin-American Robotics Symposium*, pages 1–7, 2005.

[10] G. Klančar and I. Škrjanc. Tracking-Error Model-Based Predictive Control for Mobile Robots in Real Time. *Robotics and Autonomous Systems*, 55(6):460–469, 2007.

[11] F. Xie and R. Fierro. First-State Contractive Model Predictive Control of Nonholonomic Mobile Robots. In *Proceedings of the American Control Conference*, pages 3494–3499, 2008.

[12] S. Peters and K. Iagnemma. Mobile Robot Path Tracking of Aggressive Maneuvers on Sloped Terrain. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 242–247, 2008.

[13] M. Burke. Path-Following Control of a Velocity Constrained Tracked Vehicle Incorporating Adaptive Slip Estimation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 97–102, 2012.

[14] T. M. Howard, C. J. Green, and A. Kelly. Receding Horizon Model-Predictive Control for Mobile Robot Navigation of Intricate Paths. In *Proceedings of the 7th Annual Conference on Field and Service Robotics*, pages 69–78, 2009.

[15] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard. Gaussian Process Model Based Predictive Control. In *Proceedings of the American Control Conference*, volume 3, pages 2214–2219, 2004.

[16] C. Lehnert and G. Wyeth. Locally Weighted Learning Model Predictive Control for Nonlinear and Time Varying Dynamics. In *Proceedings of the International Conference on Robotics and Automation*, pages 2604–2610, 2013.

[17] S. Park, S. Mustafa, and K. Shimada. Learning-Based Robot Control with Localized Sparse Online Gaussian Process. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1202–1207, 2013.

[18] M. Tanaskovic, L. Fagiano, R. Smith, and M. Morari. Adaptive Receding Horizon Control for Constrained MIMO Systems. *Automatica*, 50(12):3019–3029, 2014.

[19] H. S. Witsenhausen. A Minimax Control Problem for Sampled Linear Systems. *IEEE Transactions on Automatic Control*, 13(1):5–21, 1968.

[20] P. J. Campo and M. Morari. Robust Model Predictive Control. In *Proceedings of the American Control Conference*, pages 1021–1026, 1987.

[21] P. O. M. Scokaert and D. Q. Mayne. Min-Max Feedback Model Predictive Control for Constrained Linear Systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

[22] A. Bemporad, F. Borrelli, and M. Morari. Min-Max Control of Constrained Uncertain Discrete-Time Linear Systems. *IEEE Transactions on Automatic Control*, 48(9):1600–1606, 2003.

[23] E. C. Kerrigan and J. M. Maciejowski. Feedback Min-Max Model Predictive Control using a Single Linear Program: Robust Stability and the Explicit Solution. *International Journal of Robust and Nonlinear Control*, 14(4):395–413, 2004.

[24] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. Camacho. Min-Max Model Predictive Control of Nonlinear Systems: A Unifying Overview on Stability. *European Journal of Control*, 15(1):5 – 21, 2009.

[25] L. Blackmore. A Probabilistic Particle Control Approach to Optimal, Robust Predictive Control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[26] J. Matsuko and F. Borrelli. Scenario-Based Approach to Stochastic Linear Predictive Control. In *Proceedings of the Conference on Decision and Control*, pages 5194–5199, 2012.

[27] G. C. Calafiore and L. Fagiano. Robust Model Predictive Control via Scenario Optimization. *IEEE Transactions on Automatic Control*, 58 (1):219–224, 2013.

[28] F. Berkenkamp and A. P. Schoellig. Learning-Based Robust Control: Guaranteeing Stability while Improving Performance. In *Proceedings of the Workshop on Machine Learning in Planning and Control of Robot Motion, International Conference on Intelligent Robots and Systems*, 2014.

[29] A. Aswani, H. Gonzalez, S. Shankar Sastry, and C. Tomlin. Provably Safe and Robust Learning-Based Model Predictive Control. *Automatica*, 49:1216–1226, 2013.

[30] J. Nocedal and S. Wright. *Numerical Optimization*, volume 2. Springer New York, 1999.

[31] M. Diehl, H. J. Ferreau, and N. Haverbeke. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation. In *Lecture Notes in Control and Information Sciences, Volume 384, Nonlinear Model Predictive Control*, pages 391–417. Springer, 2009.

[32] C. Ostafew, J. Collier, A. P. Schoellig, and T. D. Barfoot. Speed Daemon: Experience-Based Mobile Robot Speed Scheduler. In *Proceedings of the International Conference on Computer and Robot Vision*, pages 56–62, 2014.