An Approximate Dynamic Programming Approach for Dual Stochastic Model Predictive Control [⋆]

Elena Arcari, Lukas Hewing, Melanie N. Zeilinger

Institute for Dynamic Systems and Control, ETH Zürich, Zürich, Switzerland,
(e-mail: earcari,lhewing,mzeilinger.ethz.ch).

Abstract: Dual control explicitly addresses the problem of trading off active exploration and exploitation in the optimal control of partially unknown systems. While the problem can be cast in the framework of stochastic dynamic programming, exact solutions are only tractable for discrete state and action spaces of very small dimension due to a series of nested minimization and expectation operations. We propose an approximate dual control method for systems with continuous state and input domain based on a rollout dynamic programming approach, splitting the control horizon into a dual and an exploitation part. The dual part is approximated using a scenario tree generated by sampling the process noise and the unknown system parameters, for which the underlying distribution is updated via Bayesian estimation along the horizon. In the exploitation part, we fix the resulting parameter estimate of each scenario branch and compute an open-loop control sequence for the remainder of the horizon. The key benefit of the proposed sampling-based approximation is that it enables the formulation as one optimization problem that computes a collection of control sequences over the scenario tree, leading to a dual model predictive control formulation.

Keywords: dual control, stochastic optimal control, nonlinear predictive control, learning

1. INTRODUCTION

Many advanced control methods require a model of the system to be controlled. The standard procedure is to perform offline system identification (Ljung (1986)) of the plant and develop a posteriori a controller based on the identified model. This approach, however, assumes that all system characteristics have been observed during offline identification, which is becoming increasingly challenging and time-consuming for complex systems and environments. Adaptive control techniques (Åström and Wittenmark (2008)) address this problem by simultaneously learning and controlling an uncertain dynamical system. This leads to the well-known exploration-exploitation trade-off, which is computationally difficult to address. For this reason, many techniques revert to passive learning, i.e. the control inputs are chosen only with respect to the performance objective and not in order to drive active exploration of the unknown dynamics.

In this paper, we develop an approximate dual control algorithm that aims at optimally balancing active exploration and exploitation in the optimal control of partially unknown systems with continuous state and input spaces, while offering a tractable optimization-based formulation. The proposed method casts the exploration-exploitation trade-off as a stochastic optimal control problem by extending the state with the current parameter estimate, for which the optimal solution can be determined by applying stochastic dynamic programming (DP) with imperfect state information (Bertsekas (2017)). The solution relies on the information state of the system, the dimensionality of which grows rapidly along the horizon. For discrete state, input and parameter spaces this results in partially observable Markov decision processes (POMDP) (Poupart et al. (2006)), which are hard to solve even for small and medium sized systems (Santamaría et al. (1997)).

Stochastic DP provides the optimal exploration-exploitation trade-off with respect to the specified cost, and can therefore be regarded as the exact solution to the dual control problem (Feldbaum (1961), Mesbah (2018)). However, exact DP solutions are computationally tractable only for discrete state and action spaces of very small dimension. Among the approximate solutions proposed in the literature, dual control methods based on approximations of DP are known as *implicit* dual control methods (Filatov and Unbehauen (2000)), since the dual effect is implicit in the approximate solution of the Bellman equation. Solutions can be found by using approximate dynamic programming (ADP) (Tse and Bar-Shalom (1973), Bayard and Eslami (1985)) or the wide-sense property (Klenske and Hennig (2016)). An alternative is given by *explicit* dual control methods, which provide the dual effect explicitly, e.g. by heuristically adding probing features to the control inputs or to the cost (Marafioti et al. (2014), Tanaskovic et al. (2014), Heirung et al. (2017)). We refer to (Mesbah (2018)) for more references and an overview of the literature.

The formulation proposed in this paper is an implicit dual control technique that makes use of an ADP strategy, generally referred to as a rollout approach (Bertsekas (2017)). The method is based on splitting the control horizon in a dual and an exploitation part. In the dual part, we approximate the DP solution by constructing a scenario tree from samples of the uncertain system parameters and process noise. Each subtree is associated with a control input and a parameter distribution conditioned on the previously seen realizations and obtained at each time step by Bayesian estimation. A crucial feature of this formulation is its ability to maintain the dual effect by

[⋆] This work was supported by the Swiss National Science Foundation under grant no. PP00P2 157601 / 1 and by the CTI-Project Nr. 25959.2 PFIW-IW.

continuously updating the parameter distribution in the prediction based on scenarios. In the exploitation part, the parameter distribution is fixed for the remaining time steps of the control horizon. Furthermore, the control sequence associated with each sampled trajectory is optimized in open-loop, as in model predictive control (MPC). Similar techniques developed in an MPC setting are presented in (Thangavel et al. (2018)), in which future observations are used to update disturbance confidence bounds in a robust formulation, and (Hanssen and Foss (2015)), in which the unknown system parameter samples are updated using an ensemble Kalman filter.

In contrast to these approaches, we systematically derive a dual control method with regard to the underlying stochastic optimal control problem and its exact DP solution. Sample-based computations approximate the expected value operations in the Bellman equation, resulting in the construction of a scenario tree. Within each scenario branch, we update the available information by Gaussian conditioning of the parameter distribution on observations. This explicit connection to DP and Bayesian estimation comes with several advantages. It allows the connection to a number of established ADP strategies, for instance motivating the use of a rollout approach, which provides computational tractability for longer control horizons. This interpretation also provides a statistically consistent analysis of concepts introduced in (Hanssen and Foss (2015)). Furthermore, a Bayesian update of the parameter distribution within each scenario branch provides a flexible framework. In this paper, we specifically address parameter affine systems under Gaussian noise, for which the Bayesian update can be analytically expressed. By providing a clear interpretation of different approximation steps in the DP framework, the proposed method can be easily extended to other system classes and distributions. Finally, we show that the overall optimization problem, involving both the dual and exploitation part, can be formulated with respect to a collection of control sequences along the branches of the scenario tree, resulting in a dual stochastic MPC approach which can be solved using gradient-based optimization techniques.

The paper is organized as follows. In Section 2 we formulate the problem and recount the stochastic dynamic programming solution. The proposed approximate dual control method is presented in Section 3. In Section 4 we discuss examples and in Section 5 final remarks.

1.1 Notation

With $\mathrm{P}[\cdot|\cdot]$ we refer to the conditional probability. A normally distributed variable x with mean μ and covariance Σ is denoted $x \sim \mathcal{N}(\mu, \Sigma)$. $\mathbb{E}_x[\cdot]$ represents the expected value with respect to the random variable x. We use upper script indices to refer to samples of a quantity, and lower script indices for the time step, i.e. x_k^j denotes a sample j of state variable x at stage k. The zero matrix is defined as $0_{n,m}$ and the identity matrix as $\mathbb{I}_{n,m}$, with n rows and m columns.

2. PRELIMINARIES

2.1 Problem Formulation

We consider the control of discrete-time dynamical systems with parametric uncertainty subject to additive process noise, which can be described by

$$x_{k+1} = \Phi(x_k, u_k)\theta + w_k, \tag{1}$$

where $\Phi(x_k, u_k) \in \mathbb{R}^{n_x \times n_b}$ is a matrix composed of nonlinear basis functions $\phi_{ij}(x_k, u_k)$ mapping the states $x_k \in \mathbb{R}^{n_x}$ and control inputs $u_k \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ to scalar values, i.e. $\phi_{ij} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$. The set $\mathbb{U}$ defines the input constraints, the disturbance $w_k \sim \mathcal{N}(0, \Sigma_w)$ is assumed to be i.i.d. Gaussian and $\theta \in \mathbb{R}^{n_b}$ is a vector of fixed but uncertain parameters. In this paper, we consider the case of Gaussian distributed continuous parameters $\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$. Note that system (1) includes, e.g., the case of LTI systems with parametric uncertainty.

The goal is to find an optimal policy sequence $\Pi = \{\pi_0, \ldots, \pi_{N-1}\}$ for system (1), where $\pi_k : \mathbb{R}^{n_x} \to \mathbb{U}$, minimizing the finite horizon cost

$$J_N(\Pi, x_0) := \mathbb{E}_{\theta, w_0, \ldots, w_{N-1}}\left[\sum_{k=0}^{N-1} l_k(x_k, \pi_k(x_k)) + l_N(x_N)\right], \tag{2}$$

where N is the length of the control horizon, $l_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is a potentially time-varying stage cost and $l_N : \mathbb{R}^{n_x} \to \mathbb{R}$ is the terminal cost.

We address this finite-horizon stochastic optimal control problem using stochastic dynamic programming (DP), outlined in the following, offering the key property of inherently leading to a dual control policy.

2.2 Stochastic DP with Parametric Uncertainty

In order to apply DP to problem (2), we describe system (1) using the augmented state $[x_k, \theta_k]^T$, i.e.

$$\begin{bmatrix} x_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} \Phi(x_k, u_k)\theta_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} w_k \\ 0_{n_b,1} \end{bmatrix}, \tag{3}$$

for which information regarding the constant and unknown parameter state $\theta_k = \theta$ is only accessible through measurements of the state x_k. This formulation allows for phrasing the problem in the framework of systems with imperfect state information (Bertsekas (2017)), for which the DP solution should take into account the information about the parameter distribution gained along the control horizon. The information available at time step k can be formalized using the information vector I_k, defined as

$$I_k := [x_k, \ldots, x_0, u_{k-1}, \ldots, u_0],$$

with $I_0 := [x_0]$. DP recursively computes the optimal policy by evaluating the Bellman equation (see e.g. Bellman (1966)) as a function of the information vector I_k, i.e.

$$J_k^*(I_k) := \min_{\pi_k} l_k(x_k, \pi_k) + \mathbb{E}_{\theta, w_k}\left[J_{k+1}^*(I_{k+1})\middle| I_k\right], \tag{4}$$

for $k = 0, \ldots, N-1$, which is initialized with $J_N^*(I_N) := l_N(x_N)$. The expected value of the optimal cost-to-go is evaluated with respect to the process noise and the unknown parameters, given the available information. The evolution of the probability distribution of the parameter vector, conditioned on the information state, can be recursively obtained via Bayesian estimation (Mesbah (2018))

$$\mathrm{P}[\theta|I_k] = \frac{\mathrm{P}[x_k|\theta, u_{k-1}, I_{k-1}]\mathrm{P}[\theta|I_{k-1}]}{\mathrm{P}[x_k|u_{k-1}, I_{k-1}]}, \tag{5}$$

with $\mathrm{P}[\theta|I_0] := \mathrm{P}[\theta]$, i.e. the prior distribution over the parameter vector θ.

The resulting DP policy therefore takes into account future observations, which affect the knowledge about the parameter distribution. As a consequence, the solution inherently explores the system as necessary to optimally solve the control problem given by (2), providing *dual control* with optimal exploration-exploitation trade-off. Unfortunately, the alternation of minimization and expectation steps induced by the Bellman equation is generally computationally intractable even under the given assumption that the unknown parameter is Gaussian distributed and the state is perfectly measured (Klenske and Hennig (2016)), in which case (5) is analytically tractable.

In the next section, we propose an approximate dual control method based on stochastic DP, which provides a tractable formulation by splitting the control horizon in a dual part and exploitation part and approximating the solution via sampling.

3. DUAL STOCHASTIC MPC

The approach introduced in the following relies on a suboptimal solution to the Bellman equation, often referred to as the rollout approach. Optimally solving problem (2) requires to carry out the DP algorithm for all steps of the control horizon of length N. The idea of a rollout approach is to use a truncated horizon of length $L < N$ and approximating the cost to go $\tilde{J}_L(I_L)$ for the remainder of the horizon. The approximate DP recursion is given by

$$\tilde{J}_k(I_k) := \min_{\pi_k} l_k(x_k, \pi_k) + \mathbb{E}_{\theta, w_k}\left[\tilde{J}_{k+1}(I_{k+1}) \middle| I_k\right], \qquad (6)$$

for $k = 0, \ldots, L-1$, which is initialized with $\tilde{J}_L(I_L)$ and results in a L-step lookahead problem (Bertsekas (2017)). The terminal cost $\tilde{J}_L(I_L)$ is evaluated with respect to a suboptimal base policy $\tilde{\pi}_L$, resulting in a suboptimal control policy with respect to the N-step horizon cost (2). It can, however, be shown that the policy obtained from (6) is always improving with respect to the base policy (Bertsekas (2017)). For this reason, rollout approaches are generally implemented in a receding horizon fashion.

We propose an approximate dual control algorithm for solving (2) based on the principles of the rollout approach. The control horizon is split into a dual part of length L, which is formulated as an L-step lookahead problem (6), and an exploitation part of length $N-L$, which is captured via the terminal cost-to-go $\tilde{J}_L(I_L)$. While the rollout approximation in (6) maintains the property that the solution is inherently dual, the stochastic DP algorithm is still computationally intractable. We address this problem by further approximating (6) for the dual part and evaluating the expectations as averages over samples of process noise and unknown parameters. These samples are used to build a scenario tree, as exemplified in Figure 1 for $L = 2$. The subtree for every time step is defined by a control input and an updated information state, which in turn affects the parameter distribution and subsequently samples of the parameters for $k = 0, \ldots, L-1$.

Using averages instead of evaluating the expected value allows for unnesting the minimizations arising in (6) and for simultaneously optimizing all control inputs associated to each subtree, without having to explicitly carry out the DP recursion. Nevertheless, this approximation still provides a solution that depends on future observations, while reacting to the sampled state associated with each subtree, hence ensuring dual control.

The terminal cost $\tilde{J}_L(I_L)$ is obtained from a base policy, which fixes the information state and optimizes over a single control sequence for each sampled trajectory for the remainder of the horizon of length $N-L$, corresponding to a (non-dual) stochastic MPC problem. Overall, this formulation of the dual and the exploitation part allows for merging the two optimization subproblems into one optimization problem that computes a collection of control sequences for the entire control horizon N, for all the branches of the scenario tree.

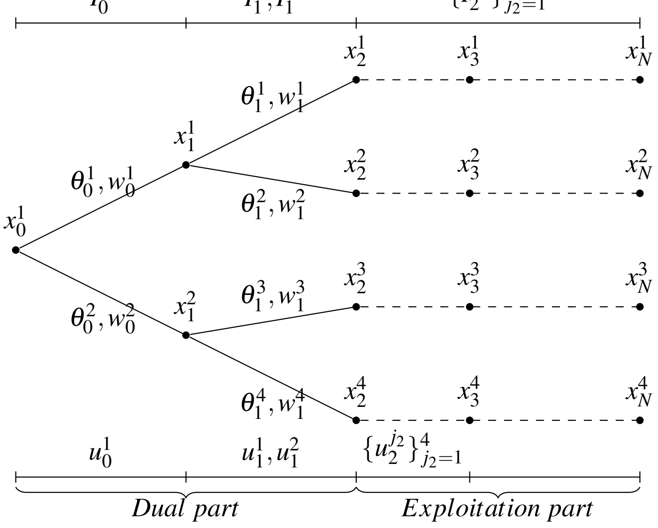

Fig. 1. Scenario tree for $L = 2$: The state propagation starts at a given initial condition and evolves, under a given input sequence, according to the $N_s = 2$ sampled parameters and disturbances. The exploitation part further predicts the trajectory using the parameter distribution from step $k = 2$.

In the following subsections, further details are given on the scenario tree for solving the dual part and on the parameter inference (5) performed under the assumption of a Gaussian distributed parameter. Furthermore, we discuss the optimal control problem for the exploitation part defining the terminal cost-to-go $\tilde{J}_L(I_L)$ and finally the overall optimization problem that merges the two parts and approximates the dual control problem in the form of a dual MPC approach.

3.1 Dual Part

Scenario Tree Generation. In order to approximate the DP recursion in the dual part, we consider a scenario tree generated by repeated sampling of the noise w_k and parameter vector θ, given the distribution defined by the available information vector I_k. The nodes of the tree are denoted by the state samples $x_k^{j_k}$, where step k indicates the depth level of the tree and $j_k = 1, \ldots, N_s^k$ indicates a sample at this level, with N_s^k being the k-th power of the number of scenarios N_s. At each node $x_k^{j_k}$ in the tree, we generate N_s scenarios of $w_k \sim \mathscr{N}(0, \Sigma_w)$ and $\theta \sim \mathrm{P}[\theta | I_k^{j_k}]$, that is the distribution of θ according to the information vector $I_k^{j_k}$ at that node. This generates the child nodes according to (1), i.e.

$$x_{k+1}^{j_{k+1}} = \Phi(x_k^{j_k}, u_k^{j_k})\theta_k^{j_{k+1}} + w_k^{j_{k+1}}, \quad j_{k+1} = 1, \ldots, N_s^{k+1} \qquad (7)$$

where node j_k at step k is the parent of node j_{k+1} at step $k+1$, i.e. $x_k^{j_k} = p(x_{k+1}^{j_{k+1}})$. At the first time step, for instance, we draw N_s sample pairs $(w_0^{j_1}, \theta_0^{j_1})$ according to $w_0 \sim \mathscr{N}(0, \Sigma_w)$ and the prior distribution $\theta \sim \mathrm{P}[\theta]$. Starting from x_0^1 and applying an input u_0^1, this gives rise to N_s child nodes $x_1^{j_1}$, $j_1 = 1, \ldots, N_s$, with information $I_1^{j_1} = [x_0^1, u_0^1, x_1^{j_1}]$. Applying this procedure over L steps, we obtain the scenario tree depicted in Figure 1 for $L = 2$ and $N_s = 2$.

The updated distributions $\mathrm{P}[\theta|I_k^{j_k}]$ are computed for each child node using Bayes rule (5), enabling recursive application of the procedure, which will be detailed in the following. The scenario tree depends on the selection of inputs $u_k^{j_k}$ which, as opposed to a classical DP recursion, will be optimized in one batch (see subsection 3.1.3 for further details).

Parameter Update. The vector of available information at each node of the scenario tree is given by

$$I_{k+1}^{j_{k+1}} = [x_{k+1}^{j_{k+1}}, u_k^{j_k}, I_k^{j_k}], \quad \text{with } I_1^0 = x_0^1$$

such that for the element $x_k^{j_k}$ of $I_k^{j_k}$, the relation $x_k^{j_k} = p(x_{k+1}^{j_{k+1}})$ holds. This vector collects the states visited up to time k and inputs applied until $k-1$ for each node j_k, and is used to update the probability distribution of the unknown parameters θ using Bayesian estimation (5). The case considered here, where in (1) the uncertain parameters affect the system linearly and the noise as well as the prior parameter distribution is Gaussian, allows inference to be carried out in closed form.

Assuming that, given the current information vector $I_k^{j_k}$ at stage k and node j_k, the parameter vector is Gaussian distributed, i.e.

$$\mathrm{P}[\theta|I_k^{j_k}] = \mathcal{N}(\mu_{\theta_k}^{j_k}, \Sigma_{\theta_k}^{j_k}),$$

we generate samples $\theta_k^{j_{k+1}}$ from $\mathrm{P}[\theta|I_k^{j_k}]$ and find the likelihood of the state at the next time step as

$$\mathrm{P}[x_{k+1}|\theta_k^{j_{k+1}}, u_k^{j_k}, I_k^{j_k}] = \mathcal{N}(\Phi(x_k^{j_k}, u_k^{j_k})\theta_k^{j_{k+1}}, \Sigma_w)\,.$$

By additionally drawing N_s samples from the noise distribution and choosing an input $u_k^{j_k}$, we generate realizations of the state $x_{k+1}^{j_{k+1}}$ at nodes j_{k+1}, which define the information state $I_{k+1}^{j_{k+1}}$. According to Bayes' rule (5), at these nodes we have a parameter distribution

$$\mathrm{P}[\theta|I_{k+1}^{j_{k+1}}] \propto \mathrm{P}[x_{k+1}|\theta_k^{j_{k+1}}, u_k^{j_k}, I_k^{j_k}]\ \mathrm{P}[\theta|I_k^{j_k}]. \tag{8}$$

Since the Gaussian distribution is self-conjugate, the posterior will also be Gaussian, with updated mean and covariance matrix defined as (Bishop (2006))

$$\begin{aligned} \left[\Sigma_{\theta_{k+1}}^{j_{k+1}}\right]^{-1} &= \left[\Sigma_{\theta_k}^{j_k}\right]^{-1} + \Phi(x_k^{j_k}, u_k^{j_k})^T \Sigma_w^{-1} \Phi(x_k^{j_k}, u_k^{j_k}), \\ \mu_{\theta_{k+1}}^{j_{k+1}} &= \Sigma_{\theta_{k+1}}^{j_{k+1}} \left(\left[\Sigma_{\theta_k}^{j_k}\right]^{-1} \mu_{\theta_k}^{j_k} + \Phi(x_k^{j_k}, u_k^{j_k})^T \Sigma_w^{-1} x_{k+1}^{j_{k+1}}\right). \end{aligned} \tag{9}$$

This provides a recursive update of the first and second moments of the parameter distribution, based on the information available at the last measured node $I_{k+1}^{j_{k+1}}$. Starting this procedure from a Gaussian prior $\mathrm{P}(\theta)$ therefore provides an analytic recursion, resulting in Gaussian parameter estimates at each node in the scenario tree.

This Bayesian framework provides great flexibility and can be readily adjusted to different use cases, for instance when the updated distributions cannot be analytically computed. Examples found in (Subramanian et al. (2015), Hanssen and Foss (2015)) make use of unscented and ensemble Kalman filters, which can be understood as an approximate Bayesian parameter update.

DP Approximation using Scenario Tree. We outline the procedure for solving (6) using the scenario tree and assuming that $\tilde{J}_L(I_L^{j_L})$ is given for each sampled path. The formulation of the terminal cost-to-go will then be discussed in subsection 3.2. The expectation in (6) is approximated at each step as an average sum over realizations of process noise and unknown parameters, drawn from the updated distribution (8), i.e.

$$\begin{aligned} \tilde{J}_0(I_0^1) := \min_{u_0^1}\ & l_0(x_0^1, u_0^1) + \frac{1}{N_s} \sum_{j_1=1}^{N_s} \Big[\min_{u_1^{j_1}}\ l_1(x_1^{j_1}, u_1^{j_1}) \\ & + \frac{1}{N_s^2} \sum_{j_2=1}^{N_s^2} \Big[\cdots + \frac{1}{N_s^{L-1}} \sum_{j_{L-1}=1}^{N_s^{L-1}} \Big[\min_{u_{L-1}^{j_{L-1}}}\ l_{L-1}(x_{L-1}^{j_{L-1}}, u_{L-1}^{j_{L-1}}) \\ & + \frac{1}{N_s^L} \sum_{j_L=1}^{N_s^L} \tilde{J}_L(I_L^{j_L})\Big]\ldots\Big]\Big]. \end{aligned} \tag{10}$$

Differently from problem (6), which requires the evaluation of nested minimizations and expectations for L steps, the sampled expected value allows to unnest the minimizations

$$\begin{aligned} \tilde{J}_0(I_0^1) := \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{L-1}}\ & l_0(x_0^1, u_0^1) + \frac{1}{N_s} \sum_{j_1=1}^{N_s} l_1(x_1^{j_1}, u_1^{j_1}) + \\ & + \frac{1}{N_s^2} \sum_{j_2=1}^{N_s^2} l_2(x_2^{j_2}, u_2^{j_2}) + \cdots + \frac{1}{N_s^L} \sum_{j_L=1}^{N_s^L} \tilde{J}_L(I_L^{j_L}), \end{aligned} \tag{11}$$

where $\mathbf{u}_k = \{u_k^j\}_{j=1}^{N_s^k}$ is the collection of control inputs at time k for all sampled trajectories N_s^k. Note that this reformulation is possible since cost functions in outer minimizations are independent of the inner minimizations. This can be easily seen by considering the simple example of minimizing functions $f_1(\cdot), f_2(\cdot)$ with respect to variables x_1, x_2:

$$\min_{x_1} f(x_1) + \min_{x_2} f(x_1, x_2) = \min_{x_1, x_2} f(x_1) + f(x_1, x_2).$$

The reformulation in (11) avoids the explicit DP recursion and enables simultaneous minimization of the control sequences associated with each sampled path.

3.2 Exploitation Part

In the exploitation part, we fix the distribution of the parameters in the prediction and solve a simplified problem minimizing the expected cost for the remainder of the horizon over an input sequence $u_{L:N-1}^{j_L} = \{u_L^{j_L}, \ldots, u_{N-1}^{j_L}\}$, giving rise to the terminal cost $\tilde{J}_L(I_L^{j_L})$ in problem (11):

$$\tilde{J}_L(I_L^{j_L}) = \min_{u_{L:N-1}^{j_L}} J_L(I_L^{j_L}, u_{L:N-1}^{j_L})\,, \tag{12}$$

$$J_L(I_L^{j_L}, u_{L:N-1}^{j_L}) = \mathbb{E}_{\theta, w_L, \ldots, w_{N-1}}\left[\sum_{k=L}^{N-1} l_k(x_k^{j_L}, u_k^{j_L}) + l_N(x_N^{j_L})\,\middle|\, I_L^{j_L}\right].$$

The cost-to-go is thereby implicitly defined and can be directly integrated with the optimization-based formulation of the approximate dual problem, as summarized in section 3.3.

In the general case, the expected value needs to be numerically approximated, e.g. again by sampling. For some cost functions, for instance the commonly used quadratic cost, analytical expressions of the expected value are available in terms of mean and variance information. Other possible choices with this property are, e.g., linear or saturating cost functions (Deisenroth and Rasmussen (2011)). We focus here on the quadratic case, for which the expected value can be computed as

$$\begin{aligned} \mathbb{E}_{x_k}[\,l(x_k, u_k)\,] &= \mathbb{E}_{x_k}[\,||x_k||_Q^2\,] + ||u_k||_R^2 \\ &= ||\mu_{x_k}||_Q^2 + tr(Q\Sigma_{x_k}) + ||u_k||_R^2. \end{aligned}$$

Even under the assumption of a Gaussian distributed parameter vector θ, the state will not remain normally distributed due to the product with the basis function matrix, such that mean and variance need to be approximated. We consider two common approaches: a certainty equivalent (CE) approach and a first order Taylor approximation of (3) around the mean value of the state. In the rest of this subsection we refer to one sample index and omit it for the sake of simplicity.

For the certainty equivalent approach, consider system (3) evaluated at the mean values, i.e.

$$\begin{bmatrix}\mu_{x_{k+1}}\\ \mu_{\theta_{k+1}}\end{bmatrix} = \begin{bmatrix}\Phi(\mu_{x_k},u_k)\mu_{\theta_L}\\ \mu_{\theta_L}\end{bmatrix}, \tag{13}$$

where μ_{θ_L} is the mean obtained in the last step of the dual part (8). For the first-order Taylor approximation, the state dynamics are given by

$$\begin{bmatrix}x_{k+1}\\ \theta_{k+1}\end{bmatrix} \approx \begin{bmatrix}\Phi(\mu_{x_k},u_k)\mu_{\theta_L}\\ \mu_{\theta_L}\end{bmatrix} + \begin{bmatrix}w_k\\ 0_{n_b,1}\end{bmatrix} + \begin{bmatrix}\nabla_x[\Phi(\mu_{x_k},u_k)\mu_{\theta_L}] & \Phi(\mu_{x_k},u_k)\\ 0_{n_b,n_x} & \mathbb{I}_{n_b,n_b}\end{bmatrix}\begin{bmatrix}x_k-\mu_{x_k}\\ \theta_k-\mu_{\theta_L}\end{bmatrix},$$

providing simple update equations for both mean and covariance based on properties of affine transformations of Gaussian distributed variables:

$$\begin{aligned}\begin{bmatrix}\mu_{x_{k+1}}\\ \mu_{\theta_{k+1}}\end{bmatrix} &= \begin{bmatrix}\Phi(\mu_{x_k},u_k)\mu_{\theta_L}\\ \mu_{\theta_L}\end{bmatrix}\\ \Sigma_{k+1} &= \bar{\Sigma}_w + \bar{A}\Sigma_k\bar{A}^T,\end{aligned} \tag{14}$$

where

$$\begin{aligned}\Sigma_k &= \begin{bmatrix}\Sigma_{x_k} & \Sigma_{x_k,\theta_k}\\ \Sigma_{x_k,\theta_k}^T & \Sigma_{\theta_k}\end{bmatrix}\\ \bar{A} &= \begin{bmatrix}\nabla_x[\Phi(\mu_{x_k},u_k)\mu_{\theta_L}] & \Phi(\mu_{x_k},u_k)\\ 0_{n_b,n_x} & \mathbb{I}_{n_b,n_b}\end{bmatrix}\\ \bar{\Sigma}_w &= \begin{bmatrix}\Sigma_w & 0_{n_x,n_b}\\ 0_{n_b,n_x} & 0_{n_b,n_b}\end{bmatrix}.\end{aligned}$$

Therefore, the prediction in (12) can be carried out using either (13) or (14) for steps $k = L,\dots,N-1$, with initialization at step L being

$$\begin{aligned}\begin{bmatrix}\mu_{x_L}\\ \mu_{\theta_L}\end{bmatrix} &= \begin{bmatrix}x_L\\ \mu_{\theta_L}\end{bmatrix}\\ \Sigma_L &= \begin{bmatrix}0_{n_x,n_x} & 0_{n_x,n_b}\\ 0_{n_b,n_x} & \Sigma_{\theta_L}\end{bmatrix},\end{aligned}$$

where Σ_{θ_L} is the covariance obtained in the last step of the dual part (8).

Based on the presented approximations of the expected value in the cost-to-go (12), we can formulate the overall optimization problem in the next subsection.

3.3 Final Approximate Dual MPC Problem

The formulation of the dual and exploitation part in the form of an optimization problem allows for merging the two subproblems into one optimization problem with respect to a collection of input sequences along the control horizon of length N:

$$\begin{aligned}\min_{\mathbf{u}_0,\dots,\mathbf{u}_{N-1}} \quad & \sum_{k=0}^{L-1}\frac{1}{N_s^k}\sum_{j_k=1}^{N_s^k} l_k(x_k^{j_k},u_k^{j_k}) + \frac{1}{N_s^L}\sum_{j_L=1}^{N_s^L} J_L(I_L^{j_L},u_{L:N-1}^{j_L})\\ \text{s.t.} \quad & (7) && k=0,\dots,L-1,\\ & \theta_k^{j_{k+1}} \text{ drawn from (8)} && k=0,\dots,L-1,\\ & (13) \text{ or } (14) && k=L,\dots,N-1,\\ & \mathbf{u}_k \in \mathbb{U}^k && k=0,\dots,N-1\end{aligned} \tag{15}$$

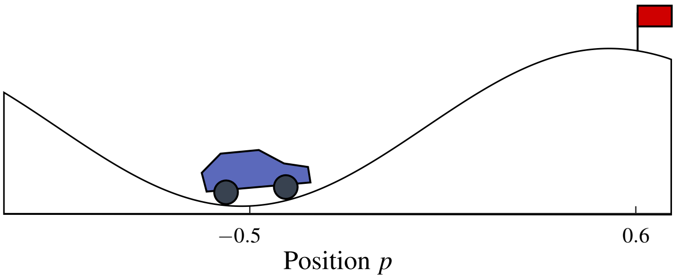

Fig. 2. The mountain car problem: A car is initially placed in a valley at position −0.5 and the goal is to pass a hill at position 0.6. Depending on the parameter realization, the car is unable to climb the hill directly and needs to 'swing up'.

Problem (15) is solved in a receding horizon fashion, updating at each time step the distribution of the parameters.

The parameters that mostly affect the outcome of this formulation are the length of the dual part L, and the number of initial samples N_s. As L tends toward the control horizon N, and as the number of of samples increases, (15) approaches the exact DP solution. In practice, however, the choice of these two parameters will mostly be a trade-off depending on the specific problem application and on the available computational resources.

Remark 1. Note that input constraints are directly incorporated in (15). State constraints can similarly be added, however no closed-loop feasibility guarantees are provided.

It is important to note that in problem (15), the parameter distribution (8) from which the samples are drawn at each stage, is a function of the optimization variables $\mathbf{u}_k$. By assuming the parameters to be Gaussian, we can however perform online sampling by generating at first N_s^{k+1} offline realizations from a standard normal distribution $\mathcal{N}(0,\mathbb{I}_{n_b,n_b})$, and then linearly transforming them online with the analytical expressions of mean and variance (9) for each parent node j_k, i.e.

$$\theta_k^{j_{k+1}} = \mu_k^{j_k} + \mathrm{chol}(\Sigma_k^{j_k})\bar{\theta}_k^{j_{k+1}},$$

where $\bar{\theta}_k^{j_{k+1}}$ is drawn from $\mathcal{N}(0,\mathbb{I}_{n_b,n_b})$, with $\mathrm{chol}(\cdot)$ standing for the Cholesky decomposition. As a consequence, we can avoid the explicit resampling of the unknown parameter in problem (15), yet generating realizations from an updated distribution computed exactly via Bayesian estimation.

4. SIMULATION EXAMPLES

We demonstrate the proposed algorithm with two different simulations. The first example is an illustrative scalar LTI system with unknown actuator gain and additive Gaussian process noise, for which we analyze the solution to problem (15) and provide closed-loop results. The second example is typically referred to as the mountain car problem (Sutton and Barto, 2018), for which we compare the performance of our approach with respect to a certainty equivalent and an adaptive MPC controller.

4.1 Scalar example

As a first illustrative example, we consider the control of a scalar system described by

$$x_{k+1} = x_k + \theta u_k + w_k,$$

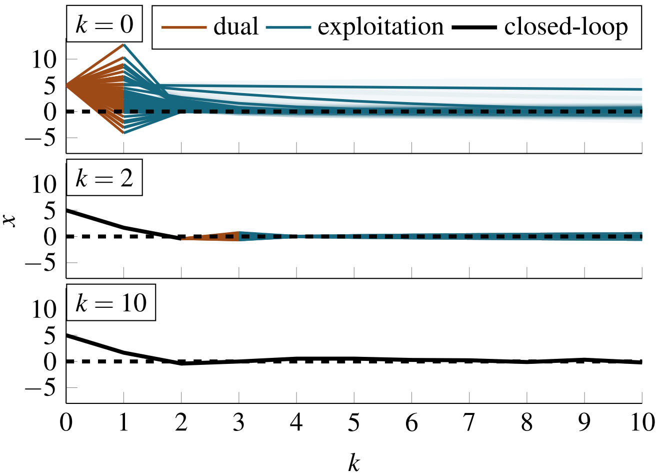

Fig. 3. Regulation of LTI scalar system. Depicted are the closed-loop trajectory, mean and 1σ bounds of the $N_s = 50$ sampled open-loop trajectories. From the top: Plot 1 shows at time k=0 the predicted open-loop sequences. Plot 2 shows the closed-loop state for two time steps and the open-loop prediction generated at k=2. Plot 3 shows the closed-loop trajectory for 10 time steps.

where θ is the unknown parameter, assumed to be distributed as $\theta \sim \mathcal{N}(1,(3.1)^2)$, $w_k \sim \mathcal{N}(0,(0.3)^2)$ is process noise, and the true parameter is $\theta_{\text{true}} = 3$. Linearity with respect to the unknown parameter allows inference to be carried out in closed-form (8). Furthermore, for this simple set-up, propagation of state mean and variance in the exploitation part can be evaluated exactly, as a result of linearly combined Gaussian distributed variables. The goal is to regulate the system to the origin from an initial condition $x_0 = 5$. We define a quadratic cost function $l_k(x_k,u_k) = ||x_k||^2_Q + ||u_k||^2_R$, where $Q = 10$ and $R = 0.01$, and solve (15) using $N_s = 50$ samples and $L = 1$ lookahead steps.

The purpose of this illustrative example is to investigate the ability of the dual controller to actively explore and gain information about the unknown system parameter. Nevertheless, even this simple task would be very difficult for a non-dual controller, e.g. certainty equivalent MPC, since the direction of the actuator gain is unknown. Figure 3 shows the predictions for both the dual and the exploitation part and the closed-loop solution at time steps $k = 0,2,10$. Notice that at time $k = 0$, the control action is selected to test several strategies, including to go in the wrong direction with respect to the origin, in order to gain information about θ. At time step $k = 2$, the parameter is almost identified, and its updated distribution is $\theta \sim \mathcal{N}(3.1,(0.25)^2)$. The exploration is now reduced, and for the remaining time steps the controller regulates the system to the origin.

4.2 Mountain Car

As a second example we consider the mountain car problem with parametric model uncertainties. The goal is to drive a car from a valley past the top of a hill, as illustrated in Figure 2. The dynamics of the system are given by (Sutton and Barto, 2018)

$$x_{k+1} = \begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} p_k + T_s v_k \\ v_k - T_s \cos(3p_k)\theta_1 + T_s u_k \theta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k\,,$$

where p is the position and v is the velocity of the vehicle, input u is given by the acceleration and disturbances $w_k \sim \mathcal{N}(0,0.001^2)$. We choose a sampling time of $T_s = 7$. The task in the mountain car problem is to climb a hill, characterized by the target position $p_{\text{target}} = 0.6$ starting from an initial position $p_0 = -0.5$ located in a valley. We express this goal by choosing a simple linear cost function $l_k(x_k,u_k) = -p_k$, such that the system is encouraged to maximize its position as quickly as possible, eventually exceeding the hill.

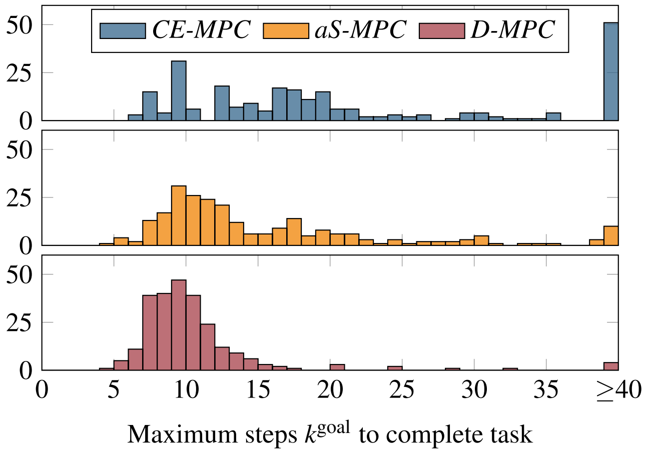

Fig. 4. Histogram of step number k^{goal} to complete task in for the mountain car problem for different controller formulations, for 250 noise realizations. The finishing time k^{goal} is the time at which the car position first exceeds $p_{\text{goal}} = 0.6$.

The parameters θ_1, θ_2 are typically chosen such that the vehicle cannot reach the goal directly, but needs to 'swing back' in order to gain speed and then climb the hill. We consider the case where the parameters are uncertain, specifically

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0.002 \\ 0.0025 \end{bmatrix}, \begin{bmatrix} 0.001^2 & 0 \\ 0 & 0.001^2 \end{bmatrix} \right),$$

such that for some realizations swinging back is necessary, for others not. In this setup, the optimal controller should first find the right strategy for the given parameter realization and subsequently execute it to minimize the expected cost.

To solve this task we generate an approximate dual controller, referred to as *D-MPC*, with lookahead $L = 3$ and $N_s = 5$ scenario realizations in each time step for an overall prediction horizon of $N = 15$. We compare the *D-MPC* to two variants, namely

CE-MPC Certainty equivalent receding horizon controller computing open-loop control sequences based on the prior maximum likelihood parameter value μ_θ.

aS-MPC Adaptive stochastic receding horizon controller. The controller uses N_s^L parameter and noise samples and computes the optimal input sequence by optimizing the sampled average of the cost, approximating the expected value. The controller passively learns and adapts the parameter knowledge in closed loop using (9).

As performance indicator we investigate k^{goal}, i.e. the number of time steps required until exceeding the target state $p_{\text{goal}} = 0.6$ for the different controllers using 250 simulation experiments with different noise realizations. A histogram of the resulting distributions of k^{goal} is given in Figure 4. The results clearly show that the certainty equivalent controller (*CE-MPC*) is unable to exceed the hill within 40 time steps and fails in a large portion of the trials. One can furthermore observe two modes in the finishing times, the first around 10 time steps,

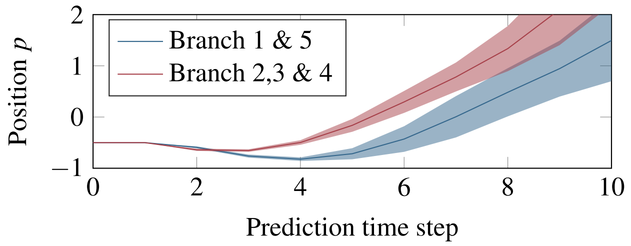

Fig. 5. Prediction of the dual controller in the mountain car problem at time step 0. Displayed are the mean and 2σ bounds of the $N_s^L = 5^3$ planned trajectory samples, separated in two groups depending on the 5 initial tree branches reflecting different strategies. The controller plans to swing back and depending on the collected information to start the hill climb at different points in time.

corresponding to a success on first trial, and a second starting around 15 time steps, corresponding to success after a first failure. The adaptive stochastic controller (*aS-MPC*) manages to complete most trials within 40 time steps, since it is able to learn the parameter distribution over time. The two modes, however, are still visible, meaning that a number of climbing attempts by the *aS-MPC* fail on first trial. The dual controller (*D-MPC*) in contrast is able to successfully climb the hill on the first trial in almost all cases, exceeding the other controllers in performance.

Figure 5 illustrates the prediction of the dual controller in the first time step, which generates $N_s^L = 5^3$ sampled trajectories. By observing the related scenario tree, the $N_s = 5$ branches can be grouped in two distinct solution strategies, based on a short and longer swing-back. The controller therefore plans to swing back, and depending on the information gained, start the hill climb earlier or later. This flexibility in the planning, due to the ability of the controller to select a different control input based on the information gained during execution, greatly helps to reduce conservatism.

5. CONCLUSIONS

We have presented an approximate dual control approach that is systematically derived from a stochastic DP framework and a rollout strategy. The formulation is based on separating the control horizon into a dual and an exploitation part. The dual part is formulated using a scenario tree with realizations of noise and unknown parameters, and the exploitation part optimizes over open-loop control sequences for the remainder of the horizon. By approximating the expected values using sampling, the two subproblems can be merged, simultaneously optimizing over a collection of input sequences along the control horizon. The problem of sampling online from a distribution that depends on the optimization variables is solved by generating samples from a standard normal distribution and affinely transforming them with the exact mean and covariance computed via Bayesian estimation over the prediction horizon. The proposed technique, thereby, offers a tractable procedure while maintaining the dual features of stochastic DP. The results were illustrated for a scalar LTI system and for the mountain car problem.

REFERENCES

Åström, K.J. and Wittenmark, B. (2008). *Adaptive Control*. Dover Publications, 2 edition.

Bayard, D.S. and Eslami, M. (1985). Implicit dual control for general stochastic systems. *Optimal Control Applications and Methods*, 6(3), 265–279.

Bellman, R. (1966). Dynamic programming. *Science*, 153(3731), 34–37.

Bertsekas, D.P. (2017). *Dynamic Programming and Optimal Control*. Athena Scientific, 4th edition.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Deisenroth, M.P. and Rasmussen, C.E. (2011). PILCO: A model-based and data-efficient approach to policy search. In *28th Int. Conf. Machine Learning*, 465–472.

Feldbaum, A.A. (1961). Dual control theory. Part I & II. *Automation and Remote Control*, 21(9), 874–880, 1033–1039.

Filatov, N. and Unbehauen, H. (2000). Survey of adaptive dual control methods. *Control Theory and Applications*, 147, 118 – 128.

Hanssen, K.G. and Foss, B. (2015). Scenario based implicit dual model predictive control. *Conf. Nonlinear Model Predictive Control*, 416 – 421.

Heirung, T.A.N., Ydstie, B.E., and Foss, B. (2017). Dual adaptive model predictive control. *Automatica*, 80, 340–348.

Klenske, E.D. and Hennig, P. (2016). Dual control for approximate Bayesian reinforcement learning. *J. Machine Learning Research*, 17(1), 4354–4383.

Ljung, L. (1986). *System Identification: Theory for the User*. Prentice-Hall.

Marafioti, G., Bitmead, R.R., and Hovd, M. (2014). Persistently exciting model predictive control. *Int. J. Adaptive Control and Signal Processing*, 28(6), 536–552.

Mesbah, A. (2018). Stochastic model predictive control with active uncertainty learning: A survey on dual control. *Annual Reviews in Control*, 45, 107 – 117.

Poupart, P., Vlassis, N., Hoey, J., and Regan, K. (2006). An analytic solution to discrete Bayesian reinforcement learning. In *23rd Int. Conf. Machine Learning*, 697–704.

Santamaría, J.C., Sutton, R.S., and Ram, A. (1997). Experiments with reinforcement learning in problems with continuous state and action spaces. *Adaptive Behavior*, 6(2), 163–217.

Subramanian, S., Lucia, S., and Engell, S. (2015). Economic multi-stage output feedback nmpc using the unscented kalman filter. *IFAC-PapersOnLine*, 48(8), 38 – 43. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2 edition.

Tanaskovic, M., Fagiano, L., Smith, R., and Morari, M. (2014). Adaptive receding horizon control for constrained MIMO systems. *Automatica*, 50(12), 3019–3029.

Thangavel, S., Lucia, S., Paulen, R., and Engell, S. (2018). Dual robust nonlinear model predictive control: A multi-stage approach. *J. Process Control*, 72, 39 – 51.

Tse, E. and Bar-Shalom, Y. (1973). An actively adaptive control for linear systems with random parameters via the dual control approach. *Trans. Automatic Control*, 18(2), 109–117.