In: Proc. International Symposium of Robotics Research (ISRR), Siestre-Levante, Italy (2015).

An Approximate Inference Approach to Temporal Optimization for Robotics

Konrad Rawlik, Dmitry Zarubin, Marc Toussaint, and Sethu Vijayakumar

Abstract Algorithms based on iterative local approximations present a practical approach to optimal control in robotic systems. However, they generally require the temporal parameters (for e.g. the movement duration or the time point of reaching an intermediate goal) to be specified a priori. Here, we present a methodology that is capable of jointly optimizing the temporal parameters in addition to the control command profiles. The presented approach is based on a Bayesian formulation of the optimal control problem, which includes the time course of the movement as a random variable. An approximate EM algorithm is derived that efficiently optimizes both the time course of the movement and the control commands offering, for the first time, a practical approach to tackling generic via point problems in a systematic way under the optimal control framework. The proposed approach, which is applicable to plants with non-linear dynamics as well as arbitrary state dependent and quadratic control costs, is evaluated on realistic simulations of a redundant robotic plant and on a simulated KUKA robotic arm.

Konrad Rawlik
University of Edinburgh, UK
e-mail: konrad.rawlik@roslin.ed.ac.uk

Dmitry Zarubin
University of Stuttgart, Germany
e-mail: dmitry.zarubin@ipvs.uni-stuttgart.de

Marc Toussaint
University of Stuttgart, Germany
e-mail: marc.toussaint@informatik.uni-stuttgart.de

Sethu Vijayakumar
University of Edinburgh, UK
e-mail: sethu.vijayakumar@ed.ac.uk

1 Introduction

Control of sensorimotor systems, artificial or biological, is inherently both a spatial and temporal process. Not only do we have to specify *where* the plant has to move to but also *when* it reaches that position. In some control schemes, the temporal component is implicit. For example, with an infinite horizon, discounted cost based controller, movement duration results from the application of the feedback loop. In other cases it is explicit, like for example in finite horizon objective based formulations, where the time horizon is set explicitly as a parameter of the problem (Stengel, 1986).

Although control based on an optimality criterion is certainly attractive, practical approaches for stochastic systems are currently limited to the finite horizon objective or the first exit time objective. The former does not optimize temporal aspects of the movement, i.e., duration or the time when costs for specific sub-goals of the problem are incurred, assuming them as given *a priori*. However, how should one choose these temporal parameters? This question is non-trivial and important, even when considering a simple reaching problem. The solution generally employed in practice is to use an *a priori* fixed duration, chosen experimentally. This can result in not reaching the goal, having to use an unrealistic range of control commands or excessive (wasteful) durations for short distance tasks. The alternative first exit time formulation, on the other hand, either assumes specific exit states in the cost function, and computes the shortest duration trajectory which fulfils the task, or assumes a time stationary task cost function and computes the control which minimizes the joint cost of movement duration and task cost (Toussaint and Storkey, 2006; Barber and Furmston, 2009; Kulchenko and Todorov, 2011). This formalism is thus directly applicable only to tasks which do not require sequential achievement of multiple goals. Although this limitation could be overcome by chaining together individual time optimal single goal controllers, such a sequential approach has several drawbacks. First, if we are interested in placing a cost on overall movement duration, we are restricted to linear costs if we wish to remain time optimal. A second more important flaw is that future goals should influence our control even before we have achieved the previous goal.

In this paper, we extend standard finite horizon Stochastic Optimal Control (SOC) problem formulation with additional cost terms on temporal aspects of a control policy.

2 Problem formulation

2.1 Finite Horizon Stochastic Optimal Control Problem

Let us consider a general controlled process, with state $x \in \mathbb{R}^{D_x}$ and controls $u \in \mathbb{R}^{D_u}$, given by the stochastic differential equation of the form

$$\mathrm{d}x = f(x,u)\ \mathrm{d}t + \mathrm{d}\xi\ , \quad \left\langle \mathrm{d}\xi\ \mathrm{d}\xi^\top \right\rangle = Q\ . \tag{1}$$

with non-linear dynamics f and Brownian motion ξ. Fixing a finite time horizon t_f we denote by $x(\cdot)$ and $u(\cdot)$ the state and control trajectories over the interval $t \in [0,t_f]$. For a given state-control trajectory we define the cost function as

$$C(x(\cdot),u(\cdot)) = \int_0^{t_f} c(x(t),u(t),t)\ \mathrm{d}t + c_f(x(t_f))\ , \tag{2}$$

where $c(x,u,t)$ is a cost rate for being in state x and applying controls u at time t, and c_f denotes a final state cost term. The finite horizon stochastic optimal control problem is to find the (non-stationary) control policy $\pi^* : (x,t) \to u$ that minimizes the expected total cost given a start state $x(0)$ and t_f,

$$\pi^* = \underset{\pi}{\operatorname{argmin}} \left\langle C(x(\cdot),u(\cdot)) \right\rangle_{x(\cdot),u(\cdot)|\pi,x(0)}\ . \tag{3}$$

Here we take the expectation w.r.t. the distribution $P(x(\cdot),u(\cdot) \mid \pi,x(0))$ over state-control trajectories conditional on the given start state and control policy.

2.2 Temporal Optimisation problem

In practical robotics applications cost can generally be divided into subgoals, these being costs dependent only on state and incurred at intermediate time instances, and stationary costs incurred throughout the movement. We express this by considering a cost of the following form,

$$C(x(\cdot),u(\cdot),\mathscr{T}) = \int_0^{t_f} c(x(t),u(t))\ \mathrm{d}t + \sum_{i=1}^{f} c_i(x(t_i)) + C_{\mathscr{T}}(\mathscr{T}) \tag{4}$$

where $\mathscr{T} = \{t_1,..,t_f\}$ is a set of time instances—the *time course*—at which specific subgoals, captured by the corresponding c_i's, are to be fulfilled. For instance, in a reaching movement, a cost that is a function of the distance to the target is incurred only at the final time t_f, while intermediate costs may represent subgoals like the alignment of an orientation some time *before* the reaching of a target. In our temporal optimisation framework, our objective shall be the optimisation of the time course $\mathscr{T}$ itself, including an explicit cost term $C_{\mathscr{T}}(\mathscr{T})$ that arbitrarily penalizes these time intervals. Note that this objective is broader than the duration optimisation, i.e., choice of only t_f, but of course includes it as the special case $\mathscr{T} = \{t_f\}$.

The problem now is to find the joint optimum for the control policy and the time course $\mathscr{T}$,

$$(\pi^*,\mathscr{T}^*) = \underset{\pi,\mathscr{T}}{\operatorname{argmin}} \left\langle C(x(\cdot),u(\cdot),\mathscr{T}) \right\rangle_{x(\cdot),u(\cdot)|\pi,x(0)}\ . \tag{5}$$

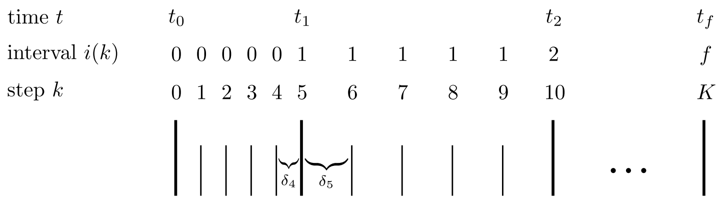


Fig. 1: Illustration of the notation used (in the case $K/f = 5$).

2.3 Time discretization

While our approach can equally be described fully in a continuous time framework, the presentation will be simplified when assuming a time discretization. Below we briefly discuss a continuous time formulation.

We discretize the time interval $[0,t_f]$ in K time steps, where each interval $[t_i,t_{i+1}]$ is discretized in K/f steps of uniform length $\delta_k = (t_{i(k)+1} - t_{i(k)})/K/f$, where $i(k) = \lfloor fk/K \rfloor$ denotes the interval that the k^{th} time step belongs to (see Figure 1 for illustration). Conversely, by $k(i) = iK/f$ we denote the step index that corresponds to the i^{th} intermediate cost c_i. Choosing different numbers of time steps per interval $[t_i,t_{i+1}]$ of non-uniform step lengths is a straight-forward extension of all the following.

In the discrete time case the problem takes the general form

$$x_{k+1} = f(x_k,u_k,\delta_k) + \varepsilon \ , \quad \varepsilon \sim \mathscr{N}(0,Q(\delta_k)) \tag{6}$$

$$C(x_{1:K},u_{1:K},\mathscr{T}) = \sum_{k=0}^{K} c(x_k,u_k)\delta_k + \sum_{i=1}^{f} c_i(x_{k(i)}) + C_{\mathscr{T}}(\mathscr{T}) \tag{7}$$

although the ideas presented can be easily adapted to alternative forms. For notational convenience, we will absorb the task costs $c_i(x_{k(i)})$ in the running costs by defining

$$\tilde{c}_k(x_k,u_k,\delta_k) = c(x_k,u_k)\delta_k + [k\%K = 0]\ c_{i(k)}(x_k) \tag{8}$$

$$C(x_{1:K},u_{1:K},\mathscr{T}) = \sum_{k=0}^{K} \tilde{c}_k(x_k,u_k,\delta_k) + C_{\mathscr{T}}(\mathscr{T}) \ , \tag{9}$$

where $k\%K$ denotes the modulo operator.

If instead we would like to stay in a time continuous framework we would define $d(t)$ as a function of time, thus augmenting the state space by a dimension. The quantity $d(t)$ can be regarded as a general resource variable and the general problem formulation (4) reformulated as a first exit time problem - details can be found in Rawlik (2013). Several algorithms applicable to problems with general non-linear dynamics have been developed, e.g. DDP (Theodorou et al., 2010), ILQG (Todorov and Li, 2005) to name a couple, all of which can be directly applied to this reformulation of the temporal optimisation problem. However, our experience has shown that naive application of such algorithms, in particular those listed, to the problem of temporal optimisation fails. This is generally due to the nature of these approximate algorithms as local optimisers. With a poor initialisation, setting $d(\cdot) = 0, \pi(\cdot,\cdot) = 0$, i.e., not moving for no time or close approximations thereof, often proves to be a dominant local minimum. We are therefore compelled to seek alternative optimisation schemes, which avoid the collapse of the solution to such undesirable outcomes. In the following we describe an approach based on alternate optimisation of the policy and $\mathscr{T}$. This is formulated in the AICO framework, which frames the problem as an inference problem, although a similar approach can be followed within classical stochastic optimal control formulations leading to similar results.

3 Approximate Inference approach

In previous work (Rawlik et al., 2012) it has been shown that a general SOC problem can be reformulated in the context of approximate inference, or more precisely, as a problem of minimizing a Kullback-Leibler divergence. This alternative problem formulation is useful in particular for derivation of approximation methods which would be non-obvious to derive in the classical formulation. In the following we will adopt the approximate inference perspective to propose a specific approximation method to solve the temporal optimization problem.

3.1 AICOT *formulation*

In the inference control formulation, given a stochastic control policy $\pi_k(u_k|x_k)$ we define the process

$$P(x_{1:K}, u_{1:K}|\pi, \mathscr{T}) = \pi(u_0|x_0) \prod_{k=1}^{K} \pi(u_k|x_k) P(x_{k+1}|x_k, u_k, \delta_k) \ , \tag{10}$$

where $P(x_{k+1}|x_k, u_k, \delta_k)$ is given by the discrete time dynamics (6). We further introduce an auxiliary (binary) random variable r_k with the likelihood

$$P(r_k = 1|x_k, u_k, \mathscr{T}) = \exp\{-\eta \tilde{c}_k(x_k, u_k, \delta_k)\} \ , \tag{11}$$

which can be interpreted as indicating (probabilistically) whether a task is fulfilled (or rather whether costs are low). It is straight-forward to verify that

$$C(x_{1:K}, u_{1:K}, \mathscr{T}) - C_{\mathscr{T}}(\mathscr{T}) = -\log P(r_{1:K} = 1|x_{1:K}, u_{1:K}) \ , \tag{12}$$

that is, we translated task and control costs into neg-log-likelihoods. In Rawlik et al. (2012) it has been show how *for fixed* $\mathscr{T}$ computing the posterior process $P(x_{1:K}, u_{1:K}|r_{1:K} = 1, \mathscr{T})$, that is, the distribution over state-control trajectories conditioned on *always* observing "task fulfillment" is related to solving the stochastic optimal control problem. In particular, this posterior also includes the posterior policy $P(u_k|x_k, r_{1:K} = 1, \mathscr{T})$, i.e. the posterior probability of choosing a control u_k in state x_k conditioned on constant "task fulfillment", which can be used in an interactive procedure to find the optimal control policy.

In the context of temporal optimisation we are interested in the computation of the posterior

$$P(\mathscr{T}, x_{1:K}, u_{1:K}|r_{1:K} = 1) \propto P(x_0) \prod_{k=0}^{K} P(x_{k+1}|x_k, u_k, \mathscr{T}) \exp\{-C(x_{1:K}, u_{1:K}, \mathscr{T})\} \ .$$

From this the MAP policy, and in this case MAP $\mathscr{T}$, are extracted. As this problem will in general be intractable, we proceed in two steps

$$\mathscr{T}^{\mathrm{MAP}} = \operatorname*{argmax}_{\mathscr{T}} P(\mathscr{T} \mid r_{1:K} = 1) \tag{13}$$

$$\pi^{\mathrm{MAP}} = \operatorname*{argmax}_{\pi} P(\pi \mid \mathscr{T}^{\mathrm{MAP}}, r_{1:K} = 1) \tag{14}$$

Note that the second step reduces exactly to standard AICO and may be solved with any of the methods proposed by Rawlik et al. (2012); Toussaint (2009). The main focus in the following is therefore on solving (13). The proposed approach is based on an iterative procedure alternating between approximation of the distribution $P(x_{1:K}, u_{1:K}|\mathscr{T}^{\mathrm{old}}, r_{1:K} = 1)$ and utilisation of this distribution to obtain an improved $\mathscr{T}^{\mathrm{new}}$. We call this general method AICOT. Two alternative forms of the improvement step are proposed, one gradient and one EM based. The relative merits of these two methods are then discussed in 3.4

3.2 Gradient Descent

We first consider direct optimisation of (13) by gradient descent. Let

$$\mathscr{L}(\mathscr{T}) = \log P(\mathscr{T} \mid r_{1:K} = 1) \tag{15}$$

and note that

$$\nabla\mathscr{L}(\mathscr{T}) \propto \frac{1}{\mathscr{L}(\mathscr{T})} \cdot \nabla P(r_{1:K} = 1|\mathscr{T}) - \nabla C_{\mathscr{T}}(\mathscr{T})$$

In the general case $P(r_{1:K} = 1|\mathscr{T})$ will not be tractable. We therefore propose taking, similar to the standard AICO algorithms, a Gaussian approximation. For brevity, let $z_{1:K} = (x_{1:K}, u_{1:K})$ denote the state-control trajectory. We define

$$\tilde{p}(z_{1:K}|\mathscr{T}) \approx P(r_{1:K} = 1, z_{1:K}|\mathscr{T})$$

as the unnormalized Gaussian approximation to $P(z_{1:K}|r_{1:K}=1,\mathscr{T})$. Using this approximation

$$\nabla_{\mathscr{T}}\mathscr{L}(\mathscr{T}) \approx \nabla_{\mathscr{T}} \int_{z_{1:K}} \tilde{p}(z_{1:K}|\mathscr{T}) \; .$$

We derive the approximate gradient, assuming a state-control LQ approximation, that is, we consider (6) and (9) are locally in the form

$$f(z_k,\delta_k) \approx a_k(\delta_k) + A_k(\delta_k)\, z_k + B_k(\delta_k) u_k \; , \quad Q_k = Q\delta_k \tag{16}$$

$$\tilde{c}_k(z_k,k) \approx [k\%K=0] \frac{1}{2} x_k^\top C_k(\mathscr{T}) x_k - c_k(\mathscr{T})^\top x_k + \frac{1}{2} u_k^\top H u_k \; , \tag{17}$$

where all terms may depend non-linearly on $\mathscr{T}$, or δ_k. In the interest of an uncluttered notation we will not further note this dependence explicitly. Eq. (17) assumes that the running costs are quadratic in u; as in Toussaint (2009) the squared control costs can equivalently translated to a Gaussian prior over u that combines with the process noise Q_k to a an uncontrolled process with noise $Q_k + B_k H^{\text{-}1} B_k$.

We can now write the unnormalized posterior $\tilde{p}$ as the product of an uncontrolled process and a Gaussian likelihood,

$$\tilde{p}(z_{1:K}|\mathscr{T}) = \underbrace{\mathscr{N}(z_{1:K}|\mu,\Sigma)}_{\text{dynamics prior}} \cdot \underbrace{\mathscr{N}[z_{1:K}|c,C]}_{\text{cost likelihood}}$$

where $\mathscr{N}[x|a,A] \propto \exp\{-\frac{1}{2}x^\top A x + x^\top a\}$ is a Gaussian in canonical form, with precision matrix A and mean $A^{\text{-}1}a$, $c=(c_1,...,c_K)^\top$ is as in (17) neglecting the $u_k^\top H u_k$ terms, and $C=\operatorname{diag}(C_1,...,C_K)$. The elements of μ are given by

$$\mu_i = (A_0 \cdots A_{i-1})\, z_0 + \sum_{k=1}^{i-1} (A_{k+1} \cdots A_{i-1})\, a_k$$

and Σ is the symmetric matrix with

$$\Sigma_{ij} = \Sigma_{ji}^\top = (A_{j-1} \cdots A_i) \sum_{k=0}^{i-1} (A_{i-1} \cdots A_k)(Q_k + B_k H^{\text{-}1} B_k)(A_{i-1}^\top \cdots A_k)^\top$$

for $i \leq j$. In practise, given a local linearization the unnormalized posterior $\tilde{p}(z_{1:K}|\mathscr{T})$ can be computed with same computational complexity as a Riccati or Kalman filter iterating over k (Toussaint, 2009).

Now let us define $\hat{\mathbf{z}}$ to be the subset of $z_{1:K}$ which have an associated intermediate cost, i.e., $\hat{\mathbf{z}} = \{z_k : \; [k\%C=0] = \{z_k : \; c_k \neq 0, C_k \neq 0\}\}$. (Note that, if we subsumed the control costs $u_k^\top H u_k$ in the uncontrolled process, only at $[k\%K=0]$ we have cost terms.) As we can marginalize the uncontrolled process for all $z_k \notin \hat{\mathbf{z}}$, we can retrieve $\tilde{p}(\hat{\mathbf{z}}|\mathscr{T})$ as

$$\tilde{p}(\hat{\mathbf{z}}|\mathscr{T}) = \mathscr{N}(\hat{\mu}|\hat{\mathbf{C}}^{-1}\hat{c}, \hat{\Sigma} + \hat{\mathbf{C}}^{-1})$$

where $\hat{\mu}$ and $\hat{\Sigma}$ denote the appropriate sub-vector and -matrix of μ and Σ respectively. Hence, with $m := \hat{\mathbf{C}}^{-1}\hat{c}$ and $\mathbf{M} := \hat{\Sigma} + \hat{\mathbf{C}}^{-1}$, the approximate derivatives take the general form

$$\nabla \int_{z_{1:K}} \tilde{p}(z_{1:K}|\mathscr{T}) = \mathscr{N}(\hat{\mu}|m,\mathbf{M}) \left[g^{\top}[\nabla(m-\hat{\mu})] \quad -\frac{1}{2}\operatorname{Tr}\left(\mathbf{M}^{-1}\nabla\mathbf{M}\right) + \frac{1}{2} g^{\top}[\nabla\mathbf{M}]g \right]$$

where $g = \mathbf{M}^{-1}(\hat{\mu} - m)$.

Combining the results, the overall approximation to the derivatives is obtained as

$$\begin{aligned}\nabla_{\delta_k}\mathscr{L}(\mathscr{T}) \approx -\nabla_{\delta_k} C_{\mathscr{T}}(\mathscr{T}) + \Big[& g^{\top}[\nabla_{\delta_k}(m-\hat{\mu})] \\ & -\frac{1}{2}\operatorname{Tr}\left(\mathbf{M}^{-1}\nabla_{\delta_k}\mathbf{M}\right) + \frac{1}{2} g^{\top}[\nabla_{\delta_k}\mathbf{M}]g \Big] \ .\end{aligned} \tag{18}$$

The gradient $\nabla_{\delta_k}\mathbf{M}$ and $\nabla_{\delta_k}(m-\hat{\mu})$ are straight-forward by their definition. With this we can use any gradient based scheme to obtain a new $\mathscr{T}^{\text{new}}$, which in turn gives rise to a new approximation.

3.3 Expectation Maximisation

The solution to (13) can alternatively be obtained using an Expectation Maximisation approach. Specifically, we form the bound

$$\mathscr{L}(\mathscr{T}) > \int_{z_{1:K}} \underbrace{P(z_{1:K}|r_{1:K}=1,\mathscr{T})}_{p(z_{1:K})} \log P(r_{1:K}=1, z_{1:K}|\mathscr{T})$$

which is alternately maximised with respect to p and $\mathscr{T}$, in an E- and M-step.

E-Step

In the E-Step we aim to calculate the posterior over the unobserved variables, i.e. the trajectories, given the current parameter values δ_k,

$$p(z_{1:K}) = P(z_{1:K}|r_{1:K}=1,\mathscr{T}) \ .$$

We approximate this with $\tilde{p}$ using AICO as before.

M-Step

In the M-Step, we solve

$$\mathscr{T}^{\text{new}} = \underset{\mathscr{T}}{\operatorname{argmin}} \underbrace{\big\langle \log P(r_{1:K} = 1, z_{1:K} | \mathscr{T}) \big\rangle_{\tilde{p}}}_{:=\tilde{\mathscr{L}}(\mathscr{T})} ,$$

where $\tilde{p}$ is the approximation calculated in the E-Step based on $\mathscr{T}^{\text{old}}$. We may expand the objective as

$$\tilde{\mathscr{L}}(\mathscr{T}) = \sum_{k=0}^{K-1} \big(\big\langle \log P(z_{k+t}|z_k, d_k)\big\rangle - \big\langle \tilde{c}_k(z_k, d_k)\big\rangle\big) + \mathscr{C},$$

where $\langle \cdot \rangle$ denotes the expectation with respect to $\tilde{q}$ and $\mathscr{C}$ is a constant. The required expectations, $\big\langle \tilde{c}_k(z_k, d_k)\big\rangle$ and

$$\big\langle \log P(z_{k+t}|z_k, d_k)\big\rangle = -\frac{D_z}{2} \log |Q_k| - \frac{1}{2} \big\langle (z_{k+t} - f(z_k))^\top Q_k^{-1} (z_{k+1} - f(z_k))\big\rangle ,$$

are in general not tractable. As previously, we therefore resort to a LQ approximation. This leads in the general case to an expression which can not be maximised analytically w.r.t. $\mathscr{T}$. However, if the approximation and discretization are chosen such that the system is also linear in δ, i.e.,

$$f(z_k) \approx (a_k + A_k z_k)\delta_k \ , \quad Q_k = Q\delta_k \ , \quad c_k(z_k, \delta_k) \approx (\frac{1}{2} z_k^\top C_k z_k - c_k^\top z_k)\delta_k$$

it can be shown that,

$$\frac{\partial}{\partial d_k} \tilde{\mathscr{L}}(\mathscr{T}) = \delta_k^{-2} g_2 + \delta_k^{-1} g_1 + (g_0 + 2 \ \frac{\mathrm{d}}{\mathrm{d}\delta} C_{\mathscr{T}} \Big|_{d_k}) \ , \tag{19}$$

with

$$\begin{aligned} g_2 =& \frac{1}{2} \operatorname{Tr} \Big(Q_k^{-1} (\big\langle z_{k+1} z_{k+1}^\top \big\rangle - 2\big\langle z_{k+1} z_k^\top \big\rangle + \big\langle z_k z_k^\top \big\rangle)\Big) \\ g_1 =& - \frac{D_z^2}{2} \\ g_0 =& - \frac{1}{2} \Big[\operatorname{Tr}(A_k Q_k^{-1} A_k^\top \big\langle z_k z_k^\top \big\rangle) + a_k^\top Q_k^{-1} a_k \\ & + 2a_k^\top Q_k^{-1} A_k \big\langle x_k \big\rangle + \operatorname{Tr}(C_k \big\langle z_k z_k^\top \big\rangle) - 2c_k^\top \big\langle z_k \big\rangle \Big] \ . \end{aligned}$$

In the general case we may use an efficient gradient ascent to compute the M-step (for fixed $\tilde{p}$) and improve on δ_k's. However, in the specific case where $C_{\mathscr{T}}$ is a linear function of δ_k's, (19) is quadratic in δ_k^{-1} and the unique extremum under the constraint $\delta_k > 0$ can be found analytically.

3.4 Discussion

The two proposed methods have different merits. From the point of view of computational complexity the EM based updates are preferable as they only require computation of the pair marginals (z_k, z_{k+1}) and operate entirely on matrices which are the size of z_k's dimension. The gradient method instead requires computation of the covariance of all cost conditioned states and controls. Due to the inversion of this matrix, gradient updates are usually more expensive to compute.

While computationally attractive, EM updates suffer from numerical instability in many problems. In general, the deficiency of EM algorithms in near deterministic regimes is a well known problem, e.g., Barber and Furmston (2009). In our case it leads to instability when $Q \approx 0$ or if the posterior trajectories are severally constrained by the cost terms. The problem arises in the M-Step, which may be written as

$$\operatorname*{argmax}_{\mathscr{T}} - \mathrm{KL}\left(p(z_{1:K}|\mathscr{T}^{\mathrm{old}}) \| P(z_{1:K}|r_{1:K}=1,\mathscr{T})\right) + \log \int_{z_{1:K}} P(r_{1:K}=1, z_{1:K}|\mathscr{T})$$

It is now apparent that for deterministic dynamics no change in δ_k is possible, lest the KL divergence becomes infinite.

4 Experiments

4.1 Evaluation on basic via-point tasks

We first evaluate the proposed method in simulation on a simple plant. As a basic plant, we used a simulation of a 2 degrees of freedom planar arm, consisting of two links of equal length. The state of the plant is given by $x = (q, \dot{q})$, with $q \in \mathbb{R}^2$ the joint angles and $\dot{q} \in \mathbb{R}^2$ associated angular velocities. The controls $u \in \mathbb{R}^2$ are the joint space accelerations. We also added some noise with diagonal covariance.

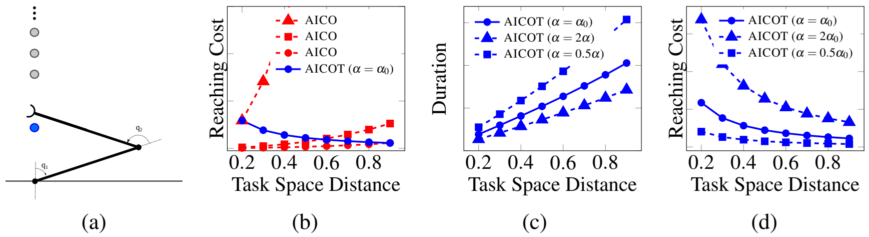
For all experiments, we used a trajectory cost of the form

$$C(x_{1:K}, u_{1:K}, \mathscr{T}) = c(x_{1:K}) + \sum_{k=0}^{K} \delta_k \, u_k^\top C^u u_k + \alpha \delta_k(\mathscr{T}) \tag{20}$$

where $C^u = 10^4 \cdot \mathbf{I}$. Note that $\alpha \sum_{k=0}^{K} \delta_k$, where δ_k depends on $\mathscr{T}$, penalizes the total movement duration linearly. The state dependent cost was

$$c(x_{1:K}) = \sum_{i=1}^{f} (\phi_n(x_{\hat{k}_i}) - y_i^*)^\top \Lambda_i (\phi_n(x_{\hat{k}_i}) - y_i^*) \ , \tag{21}$$

where the tuplets $(\hat{k}_i, \phi_i, \Lambda_i, y_i^*)$, consisting of a time step, a task space mapping, a diagonal weight matrix and the desired state in task space, define goals. For example, for point targets, the task space mapping is $\phi(x) = (x, y, \dot{x}, \dot{y})^\top$, i.e., the map from x



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

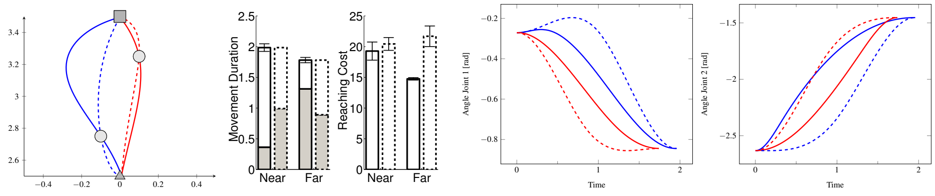
Fig. 2: Temporal scaling behaviour using AICOT. **(a)** Schematic of plant together with mean start position ○ and list of targets ○ **(b)** Comparison of reaching costs (control + error cost) for AICOT and a fixed duration approach, i.e. AICO. **(c)&(d)** Effect of changing time-cost weight α, (effectively the ratio between reaching cost and duration cost) on duration and reaching cost (control + state cost).

to the vector of end point positions and velocities in task space coordinates, and y^* is the target coordinate.

Variable Distance Reaching Task

In order to evaluate the behaviour of AICOT we applied it to a reaching task with varying start-target distance. Specifically, for a fixed start point we considered a series of targets lying equally spaced along a line in task space. It should be noted that although the targets are equally spaced in task space and results are shown with respect to movement distance in task space, the distances in joint space scale non-linearly. The state cost (21) contained a single term incurred at the final discrete step with $\Lambda_f = 10^6 \cdot \mathbf{I}$. 2(c)&(d) shows the movement duration ($= \sum_{k=0}^{K} \delta_k$) and standard reaching cost[1] for different temporal-cost parameters α (we used $\alpha_0 = 2 \cdot 10^7$), demonstrating that AICOT successfully trades-off the movement duration and standard reaching cost for varying movement distances. In 2(b), we compare the reaching costs of AICOT with those obtained with a fixed duration approach, in this case AICO. Note that although with a fixed, long duration (e.g., AICO with duration T=0.41) the control and error costs are reduced for short movements, these movements necessarily have up to 4× longer durations than those obtained with AICOT. For example for a movement distance of 0.2 application of AICOT results in a optimised movement duration of 0.07 (cf. 2(c)), making the fixed time approach impractical when temporal costs are considered. Choosing a short duration on the other hand (AICO (T=0.07)) leads to significantly worse costs for long movements. We further emphasis that the fixed durations used in this comparison were chosen post hoc by exploiting the durations suggested by AICOT; in absence of this, there would have been no practical way of choosing them apart from experimentation. Furthermore, we would like to highlight that, although the results suggests a simple

[1] n.b. the *standard reaching cost* is the sum of control costs and cost on the endpoint error, without taking duration into account.



(a) (b) (c)

Fig. 3: Comparision of AICOT (—) to AICO with the common modelling approach (- -) with fixed times on a via point task. **(a)** End point task space trajectories for two different via points ○ obtained for a fixed start point △. **(c)** The corresponding joint space trajectories. **(b)** Movement durations and reaching costs (control + error costs) from 10 random start points. The proportion of the movement duration spend before the via point is shown in light gray (mean in the AICOT case).

scaling of duration with movement distance, in cluttered environments and plants with more complex forward kinematics, an efficient decision on the movement duration cannot be based only on task space distance.

Via Point Reaching Task

We also evaluated the proposed algorithm in a more complex via point task. The task requires the end-effector to reach to a target, having passed at some point through a given second target, the via point. This task is of interest as it can be seen as an abstraction of a diverse range of complex sequential tasks that requires one to achieve a series of sub-tasks in order to reach a final goal. This task has also seen some interest in the literature on modelling of human movement using the optimal control framework (Todorov and Jordan, 2002). Here the common approach is to choose the time point at which one passes the via point such as to divide the movement duration in the same ratio as the distances between the start point, via point and end target. This requires on the one hand prior knowledge of these movement distances and on the other, makes the implicit assumption that the two movements are in some sense independent.

Here, we demonstrate the ability of our approach to solve such sequential problems, adjusting movement durations between sub-goals in a principled manner, and show that it improves upon the standard modelling approach. Specifically, we apply AICOT to the two via point problems illustrated in 3(a) with randomised start states[2]. For comparison, we follow the standard modelling approach and apply AICO to compute the controller. We again choose the movement duration for the standard case post hoc to coincide with the mean movement duration obtained with

[2] For the sake of clarity, 3(a)&(c) show mean trajectories of controllers computed for the mean start state.

AICOT for each of the individual via point tasks. Each task is expressed using a cost function consisting of two point target cost terms. Specifically, (21) takes the form

$$c(x_{1:K}) = (\phi(x_{\frac{K}{2}}) - y_v^*)^\top \Lambda_v (\phi(x_{\frac{K}{2}}) - y_v^*) + (\phi(x_K) - y_e^*)^\top \Lambda_e (\phi(x_K) - y_e^*) \ ,$$

with diagonal matrices

$$\begin{aligned} \Lambda_v &= \mathrm{diag}(\lambda_{pos}, \lambda_{pos}, 0, 0) \\ \Lambda_e &= \mathrm{diag}(\lambda_{pos}, \lambda_{pos}, \lambda_{vel}, \lambda_{vel}) \ , \end{aligned}$$

where $\lambda_{pos} = 10^5$ & $\lambda_{vel} = 10^7$ and vectors $y_v^* = (\cdot, \cdot, 0, 0)^\top$, $y_e^* = (\cdot, \cdot, 0, 0)^\top$ desired states for individual via point and target, respectively. Note that the cost function does not penalise velocity at the via point but encourages the stopping at the target. While admittedly the choice of incurring the via point cost at the middle of the movement ($\frac{K}{2}$) is likely to be a sub-optimal choice for the standard approach, one has to consider that in more complex task spaces, the relative ratio of movement distances may not be easily accessible and one may have to resort to the most intuitive choice for the uninformed case as we have done here. Note that although for AICOT this cost is incurred at the same discrete step, we allow δ_k before and after the via point to differ, but constrain them to be constant throughout each part of the movement, hence, allowing the cost to be incurred at an arbitrary point in real time. We sampled the initial position of each joint independently from a Gaussian distribution with a variance of 3°. In 3(a)&(c), we show mean trajectories in task space and joint space for controllers computed for the mean initial state. Interestingly, although the end point trajectory for the *near* via point produced by AICOT may look sub-optimal than that produced by the standard AICO algorithm, closer examination of the joint space trajectories reveal that our approach results in more efficient actuation trajectories. In 3(b), we illustrate the resulting average movement durations and costs of the mean trajectories. As can be seen, AICOT results in the expected passing times for the two via points, i.e., early vs. late in the movement for the near and far via point, respectively. This directly leads to a lower incurred cost compared to un-optimised movement durations.

Sequential and Joint Planning

In order to highlight the shortcomings of sequential time optimal control, we compare planning a complete movement, referred to as joint optimisation, to planning a sequence of individually optimised movements. We again use the via-point task of the previous section and performed (i) planning using AICOT on the entire task (ii) using AICOT to plan for to reaching tasks – start point to via-point and via-point to final target – by splitting the cost function. In the latter the end state of the first reaching movement, rather then the via-point, was used as initial state for the second sub-task. 4 summarises the results. As can be seen in 4(a) the two approaches lead

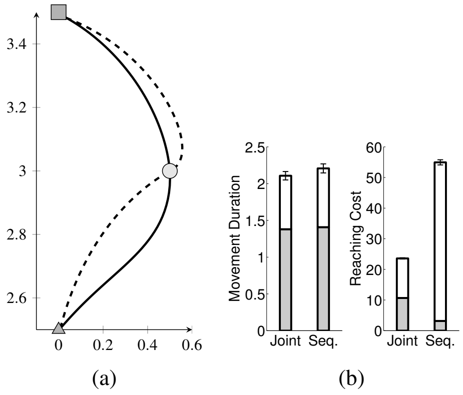


Fig. 4: Joint (—) vs. sequential (- -) optimisation using our approach on a via-point task as descibed in the main text. **(a)** Task space trajectories for the fixed start point ▲. Via-point and target are indicated by ○ and ■, respectively. **(b)** The movement durations and reaching costs for 10 random start points. The mean proportion of the movement duration spend before the via point is shown in light grey.

to solutions with substantially different end-effector trajectories in task space. The joint optimisation, accounting for the need to continue to the eventual target after the via-point, yields a different approach angle. The profound effect this has on the incurred cost can be seen in 4(b). While the joint planning incurs higher cost before the via-point the overall cost is more than halved. Importantly, as the plot of the movement durations illustrates, this reduction in cost is not achieved by an increase in movement duration, with both approaches leading to not significantly different durations. However, one should note that this effect would be less pronounced if the cost required stopping at the via-point, as it is the velocity away from the end target which is the main problem for the sequential planner.

4.2 7-DOF robotic manipulation tasks

We now turn to evaluating the method for planning with the 7-DOF Kuka lightweight robot. Our aim is two fold, on the one hand to demonstrate scalability to practical applications, and on the other hand, to demonstrate that in practical tasks temporal optimisation can significantly improve the results compared to naive selection of the movement durations.

The state of the plant is given by $x = (q, \dot{q})$, with $q \in \mathbb{R}^7$ the joint angles and $\dot{q} \in \mathbb{R}^7$ the associated angular velocities. The controls $u \in \mathbb{R}^7$ are the joint space accelerations. We also added some i.i.d. noise with diagonal covariance. The trajectory cost takes the general form

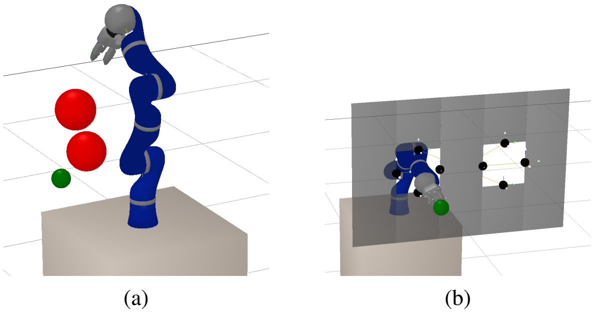

(a) (b)

Fig. 5: Example configurations for the tasks used with KUKA 7-DOF robotic system in simulation. **(a)** The simple obstacle task. The manipulator has to reach with it's end-effector to the target 🟢 whilst avoiding the obstacles 🔴. The task is randomised by sampling both the target and the obstacle positions. **(b)** The complex obstacle task. The manipulator starts in one hole and has to reach for the target 🟢 in the other, whilst avoiding collisions with the wall. The position of the wall is randomised.

Method	Simple Obstacles	Complex Obstacle
AICO	1	1
AICOT (end cost)	0.585 (± 0.337)	0.635 (± 0.085)
AICOT (full)	0.549 (± 0.311)	0.123 (± 0.047)

Table 1: Results for application of AICOT to the robotic manipulation with obstacles in the reaching tasks illustrated in 5. Shown are the mean ratio of expected cost relative to AICO and it's standard deviation.

$$C(x_{1:K},u_{1:K},\mathscr{T}) = \sum_{k}^{K}\Big(\sum_{m=1}^{M}\|\phi_m(x_k)-y_m^*\|^2_{\Lambda_{m,k}} + u_k^\top \delta_k C^u u_k\Big) \tag{22}$$

where the tuplets $(\phi_m,\Lambda_{m,k},y_m^*)$ define the task variables, consisting of a task space mapping, a time varying diagonal weight matrix and the desired state in task space.

In each task we compare three methods:

- **AICOT(full)** is the complete algorithm as described in 3.2.
- **AICOT(end cost)** is the algorithm as described in 3.2. However, the gradient is calculated taking only the reaching cost into account, i.e., ignoring joint limit and obstacle costs. The intention is to illustrate that selection of duration needs to take into account the entire problem and can not be simply based on a target-distance law as could be derived from, e.g., 2.
- **AICO** is the algorithm with fixed duration. This is to provide a comparison to the naive approach prevalent in the literature. Note however that, we set the duration the mean duration obtained by AICOT*(end cost)*. Hence it was in some sense adapted to the task distribution. Without AICOT, selection would have, at best, relied on manual selection based on an individual task instance or, at worst, a random guess. Both approaches lead to substantially worse results.

Simple Obstacle Reaching Task

We first consider a standard reaching task with obstacles. The task is defined via the following set of task variables

- **Reaching:** with $\phi_1(x) \in \mathbb{R}^6$ the arm's end effector position and velocity. The cost is incurred in the final time step only, i.e., $\Lambda_{1,k \neq K} = 0$, and y^* indicates the desired state end-effector positions with zero velocities.
- **Joint Limits:** with $\phi_2(x) \in \mathbb{R}$ a scalar indicating danger of violating joint limits. Specifically,

$$\phi_2(x) = \sum_j \mathscr{H}(d_j - \varepsilon)^2 \ , \tag{23}$$

  with d_j the distance to the joint limit between of joint j, $\mathscr{H}$ the heavy-side function and margin $\varepsilon = 0.1$rad. This task variable is considered throughout the trajectory, i.e. $\Lambda_{2,1} = \Lambda_{2,1} = \cdots = \Lambda_{2,K}$.
- **Collisions:** with $\phi_2(x) \in \mathbb{R}$ a scalar indicating proximity of obstacles. Specifically ϕ_2 takes the general form (23) with d_j the shortest distance between a pair j of collidable objects, i.e. the set of links of the arm and obstacles, and margin $\varepsilon = 0.02$m. Like the joint limits, this task variable is also considered throughout the trajectory.

Although the resulting finite cost functions can not guarantee that collisions with obstacles or joint limits will not occur, such approximations are typical in the literature (e.g., (Toussaint, 2009; Ivan et al., 2013)) and lead, under appropriate weighting between the reaching and collision components, to good results with low collision probability.

We consider a randomised task with two spherical obstacles, an example configuration being illustrated in 5(a). Specifically, both the target and obstacle positions are randomly sampled, the latter so that they lie near the direct path to the target so as to influence the solution. The results are summarised in 1. As different task instances can give rise to very different expected costs, we compare expected costs relative to AICO, i.e., the improvement of the methods over the baseline without temporal optimisation. The expected costs are estimated from sampled trajectories and we consider 50 task instances. As can be seen, temporal optimisation improves upon the naive application of AICO. In particular note that, instance specific durations as given by AICOT(end cost) improve significantly on selecting an informed constant duration (the mean duration over task instances). Furthermore, taking the entire problem into account leads to increasing gains as the problem complexity increases.

In general we note that a possible straightforward extension of the gradient based algorithm whereby we solve the problem incrementally, by using the solution of a reduced problem with intermediate cost terms removed, i.e., the AICO-T (end cost) approach, as an initialization of AICO-T (full) can significantly improve the computational complexity of the gradient based method for problems with many intermediate costs terms.

Complex Obstacle

We now consider a generic instance of a task involving manipulation in constrained spaces. It comprises the same basic task variables as used with the simple obstacle above. However instead of using spherical obstacles we use a wall with two holes as illustrated in 5(b). The end-effector starts reaching through one of the holes and the reaching target lies in the other hole. Due to their local nature direct application of AICO fails in this task, as do alternative local solvers like, e.g., ILQG. However, in the context of AICO (Zarubin et al., 2012) suggested using parallel inference in the normal state space and a abstract topological representations to overcome limitations of local planning in such tasks. With a suitable topological representation the task becomes nearly linear in the alternative representation, which then serves to regularise further inference in the plant's state space. Here we use the interaction mesh representation suggested by Zarubin et al. (2012), a scale and position invariant representation of relative positions of the plant and markers in the environment. This representation has been used for this task by Ivan et al. (2013) who also used AICO. For this experiment we again sampled the position of the wall relative to the manipulator and compared the relative expected costs averaged over 50 task instances. The results are shown in the second column of Table 1.

5 Conclusion

The contribution of this paper is a novel method for jointly optimizing a trajectory and its time evolution (temporal scale and duration) in the stochastic optimal control framework. In particular, two extension of the AICO method of Toussaint (2009) with complementary strength and weaknesses are presented. The gradient based approach, on the one hand, is widely applicable but can become computationally demanding. Meanwhile, the EM method provides an algorithm with lower computational cost, is however only applicable for certain classes of problems.

The experiments have concentrated on demonstrating the benefit of temporal optimisation in manipulation tasks. However, arguably it is dynamic movements which can benefit most from temporal adjustment. An example of this was seen in the brachiation task of (Nakanishi and Vijayakumar, 2012), where our framework was applied to brachiation with variable stiffness actuation, showing that an coordinated interplay of stiffness and temporal adjustment gives rise to gains in performance. We anticipate that, with the general rise of interest in variable impedance, e.g., in throwing (Braun et al., 2012), locomotion (Enoch et al., 2012) or climbing robots (Long et al., 2011), temporal optimisation will become a necessity if the capabilities of the dynamical system are to be fully exploited. Our framework provides a principled step in this direction.

Acknowledgements This work was supported by the EU FP7 project TOMSY (IST-FP7-270436).

References

D. Barber and T. Furmston. Solving deterministic policy (PO)MDPs using EM and antifreeze. In *Proc. of the 1st Int. Workshop on Learning and data Mining for Robots*, 2009.

D. Braun, M. Howard, and S. Vijayakumar. Optimal variable stiffness control: formulation and application to explosive movement tasks. *Autonomous Robots*, 33 (3):237–253, 2012.

A. Enoch, A. Sutas, S. Nakaoka, and S. Vijayakumar. BLUE: A bipedal robot with variable stiffness and damping. In *Proc. of. IEEE/RAS International Conference on Humanoid Robots*, 2012.

V. Ivan, D. Zarubin, M. Toussaint, and S. Vijayakumar. Topology-based representations for motion planning and generalisation in dynamic environments with interactions. *International Journal of Robotics Research*, (in press), 2013.

P. Kulchenko and E. Todorov. First-exit model predictive control of fast discontinuous dynamics: Application to ball bouncing. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011.

A. Long, T. D. Murphey, and K. Lynch. Optimal motion planning for a class of hybrid dynamical systems with impacts. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, page 4220–4226, 2011.

J. Nakanishi and S. Vijayakumar. Exploiting passive dynamics with variable stiffness actuation in robot brachiation. In *Robotics: Science and Systems VIII*, 2012.

K. Rawlik. *Approximate Inference Approaches to Stochastic Optimal Control*. PhD thesis, University of Edinburgh, 2013.

K. Rawlik, M. Toussaint, and S. Vijayakumar. On Stochastic Optimal Control and Reinforcement Learning by approximate inference. In *Proc. Robotics: Science and Systems VIII*, 2012.

R. F. Stengel. *Optimal Control and Estimation (Dover Books on Advanced Mathematics)*. Dover Publications, 1986.

E. Theodorou, Y. Tassa, and E. Todorov. Stochastic differential dynamic programming. In *Proc. of the American Control Conference*, pages 1125–1132. IEEE, 2010.

E. Todorov and M. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.

E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proc. of the American Control Conference*, pages 300–306, 2005.

M. Toussaint. Robot trajectory optimization using approximate inference. In *Proc. of the 26th Int. Conf. on Machine Learning*, pages 1049–1056. ACM, 2009. ISBN 978-1-60558-516-1.

M. Toussaint and A. Storkey. Probabilistic inference for solving discrete and continuous state Markov Decision Processes. In *Proc. of the 23rd Int. Conf. on Machine Learning*, pages 945–952, 2006.

D. Zarubin, V. Ivan, M. Toussaint, T. Komura, and S. Vijayakumar. Hierarchical motion planning in topological representations. In *Proc. of Robotics: Science and Systems VIII*, 2012.