

A Bayesian view on motor control and planning

Marc Toussaint and Christian Goerick

Abstract The problem of motion control and planning can be formulated as an optimization problem. In this paper we discuss an alternative view that casts the problem as one of probabilistic inference. In simple cases where the optimization problem can be solved analytically the inference view leads to equivalent solutions. However, when approximate methods are necessary to tackle the problem, the tight relation between optimization and probabilistic inference has fruitfully lead to a transfer of methods between both fields. Here we show that such a transfer is also possible in the realm of robotics. The general idea is that motion can be generated by fusing motion objectives (task constraints, goals, motion priors) by using probabilistic inference techniques. In realistic scenarios exact inference is infeasible (as is the analytic solution of the corresponding optimization problem) and the use of efficient approximate inference methods is a promising alternative to classical motion optimization methods. In this paper we first derive Bayesian control methods that are directly analogous to classical redundant motion rate control and optimal dynamic control (including operational space control). Then, by extending the probabilistic models to be Markovian models of the whole trajectory, we show that approximate probabilistic inference methods (message passing) efficiently compute solutions to trajectory optimization problems. Using Gaussian belief approximations and local linearization the algorithm becomes related to Differential Dynamic Programming (DDP) (aka. iterative Linear Quadratic Gaussian (iLQG)).

Marc Toussaint

Technical University Berlin, Franklinstr. 28/29, 10587 Berlin, e-mail: mtoussai@cs.tu-berlin.de

Christian Goerick

Honda Research Institute Europe, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany, e-mail: christian.goerick@honda-ri.de

1 Introduction

Bayesian Networks and inference methods like message passing algorithms are a basic computational paradigm for information processing on coupled random variables. Inference methods compute the posterior distribution over random variables when all couplings are taken into account (we will be more formal later). In this view, it is not surprising that there are strong relations between the fields of optimization and probabilistic inference: In the context of optimization, the coupling of variables is typically described by additive decomposable functions; which is in analogy to the factorization of a joint distribution as described by a Bayesian network or factor graph (with the typical identification of cost with neg-log probability). Consequently methods that originated in optimization can be translated to solve inference problems and vice versa: Message passing algorithms can be used to address satisfiability problems (Culotta et al., 2007); graph cut algorithms are used to address inference problems (MAP estimation) in Markov-Random-Fields (Tappen & Freeman, 2003). The key of efficient methods is that local computations (e.g., local message passing equations) are used to achieve global coherence.¹

Motion control and optimization are fundamental and very interesting problems in robotics. The problem can be formalized as an optimization problem: devising an appropriate cost function we can derive classical solutions (e.g., motion rate control, stochastic optimal control) which provide the basis of modern robot control (Peters et al., 2005). Complementary to the optimization view on robot control, we can also address the problem from the point of view of probabilistic inference. The classical cost function is replaced by a joint distribution over coupled random variables (e.g., via a neg-log transform), and the classical solution methods are replaced by methods of probabilistic inference. In simple cases, in particular those where an exact solution can efficiently be computed in the optimization framework, the inference approach will only reproduce the same solution. As in the field of optimization, the transfer of methods becomes interesting when the optimization problem becomes hard and exact algorithms are computationally expensive. Inference methods like message passing algorithms are promising candidates to yield (approximate) solutions to the optimization problem.

The problem of motion control and planning typically involves “solving” a system of many coupled variables: transition or control costs couple the state variable in two consecutive time steps, task constraints couple a task variable with the state variable within a time slice. Classically, these couplings are implicit in the cost function. In the inference view, these couplings are explicitly formulated as conditional dependencies in a joint distribution. This view naturally extends to more complex and structured robotic systems where the state of the system is represented by a number of state variables instead of a single state variable. An example is hierarchical control, or decoupled (or weakly coupled) control problems, where we maintain separate state variables for the left and the right effector of a robot and their control

¹ More formally, “coherence” could denote the marginal consistency in the context of inference, or the consistency with constraints in the context of optimization.

and plans (posterior distributions) are only weakly coupled. In such cases we can use probabilistic inference methods that exploit structured (factored) representations of the problem. Generally, while Bayesian methods have become a standard tool for sensor processing and fusion, we investigate them to solve a fusion problem on the motor level, namely the problem of fusing motion objectives like task constraints, goals and motion priors into a posterior distribution over trajectories.

In this chapter we give an introduction to Bayesian motor control and planning, i.e., methods to compute motions *given* a (probabilistic) model. Although our focus is not directly on learning such models, the methods are interesting also from the point of view of “motor and interaction learning”: For instance, in (Vlassis & Toussaint, 2009) we show how inference methods for model-based planning can be translated to model-free Reinforcement Learning algorithms using stochastic sampling methods. (Bui et al., 2002) show how probabilistic motion inference is useful in the context of imitation learning. Generally, in a model-based learning approach (where the behavior learning problem is decomposed in a first stage of learning a model and a second stage of using the model to generate behavior) one should always expect the learned model to be uncertain (see other chapters of this book, such as (Salaun et al., 2009; Howard et al., 2009)). The Bayesian methods we propose here naturally address such uncertainty. The Bayesian framework also motivates new interesting learning problems in the context of motion, for instance, learning motion priors from data.

This chapter is organized as follows. In the next section we first address the kinematic and dynamic control problem, derive Bayesian control equations and highlight the close relation to classical control equations. Section 3 introduces analogous probabilistic models that represent the motion planning problem. Approximate inference methods in these models yield new algorithms. When we approximate the system locally these new equations are related to the Ricatti equation of the Linear Quadratic Gaussian (LQG) case. In the general non-LQG case we need approximate inference methods to solve the planning problem, for which we derive local message update equations. Section 4 presents experiments that illustrate the methods. Additionally, we discuss hierarchical planning (where one alternates between planning in the task space and planning in the q -space, see also (Li et al., 2004; Todorov & Li, 2004)). This paper is an extension of the work presented earlier in (Toussaint & Goerick, 2007); see also the more theoretical discussion (Toussaint, 2009) of the relation between iLQG and inference in general stochastic optimal control scenarios.

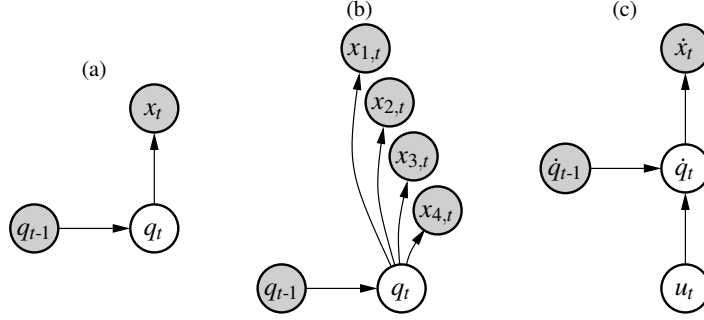


Fig. 1 Graphical models of Bayesian kinematic control, (a) for motion rate control, (b) under multiple task constraints, (c) for Bayesian dynamic control.

2 A Bayesian view on classical control

2.1 Kinematic case

We first address the case of kinematic control, i.e., the problem of deciding on the control signal given a desired constraint at the *next* time step. Throughout the derivation we will make use of identities for Gaussians which are summarized in the appendix. Let $q_t \in \mathbb{R}^n$ be a random variable referring to the robot's joint state at time t . And let x_t be a random variable referring to a task space of the robot (e.g. an endeffector state) at time t . Consider the joint probability distribution

$$P(x_t, q_t, q_{t-1}) = P(x_t | q_t) P(q_t | q_{t-1}) P(q_{t-1}) \quad (1)$$

as also illustrated by the graphical model in Figure 1(a). Here, we call $P(q_t | q_{t-1})$ the *motion prior* and assume

$$P(q_t | q_{t-1}) = \mathcal{N}(q_t | q_{t-1} + h, W^{-1}), \quad (2)$$

where $h \in \mathbb{R}^n$ is a vector that induces an *asymmetry in the motion prior* and W is the *motion metric* which enters this prior in terms of its covariance W^{-1} . Further, we call $P(x_t | q_t)$ the *task coupling* and assume

$$P(x_t | q_t) = \mathcal{N}(x_t | \phi(q_t), C). \quad (3)$$

Here, ϕ is a non-linear function (the task kinematics) with Jacobian $J = \frac{\partial \phi}{\partial q}$ and C denotes the covariance in this coupling (inversely, C^{-1} denotes the precision or tolerance of this coupling).

Given this model we can compute the posterior motion conditioned on a desired task constraint. That is, given x_t and q_{t-1} we compute $P(q_t | x_t, q_{t-1})$. We can derive the following result.

Theorem 1. *Given equations (1-3), the posterior motion is*

$$P(q_t | x_t, q_{t-1}) \approx \mathcal{N}(q_t | q_t^{\text{MAP}}, (J^T C^{-1} J + W)^{-1}) \quad (4)$$

with the MAP motion

$$q_t^{\text{MAP}} = q_{t-1} + (J^T C^{-1} J + W)^{-1} [J^T C^{-1} (x_t - \phi(q_{t-1})) + W h] \quad (5)$$

$$= q_{t-1} + J_{\text{WC}}^\# (x_t - \phi(q_{t-1})) + (\mathbf{I}_n - J_{\text{WC}}^\# J) h, \quad (6)$$

$$J_{\text{WC}}^\# := W^{-1} J^T (J W^{-1} J^T + C)^{-1}.$$

The approximation refers to the linearization of $\phi(q_t) \approx \phi(q_{t-1}) + J(q_t - q_{t-1})$.

Proof. We have

$$P(q_t | x_t, q_{t-1}) \propto P(x_t | q_t) P(q_t | q_{t-1}) = \mathcal{N}(x_t | \phi(q_t), C) \mathcal{N}(q_t | q_{t-1} + h, W^{-1})$$

Using the linearization $\phi(q_t) \approx \phi(q_{t-1}) + J(q_t - q_{t-1})$ we get

$$P(q_t | x_t, q_{t-1}) \approx \mathcal{N}(x_t | \phi(q_{t-1}) + J(q_t - q_{t-1}), C) \mathcal{N}(q_t | q_{t-1} + h, W^{-1})$$

Applying the Gaussian identities given in the appendix we have

$$\begin{aligned} P(q_t | x_t, q_{t-1}) &= \mathcal{N}(J q_t | J q_{t-1} + x_t - \phi(q_{t-1}), C) \mathcal{N}(q_t | q_{t-1} + h, W^{-1}) \\ &\propto \mathcal{N}[q_t | J^T C^{-1} J q_{t-1} + J^T C^{-1} (x_t - \phi(q_{t-1})), J^T C^{-1} J] \mathcal{N}[q_t | W q_{t-1} + W h, W] \\ &\propto \mathcal{N}[q_t | (J^T C^{-1} J + W) q_{t-1} + J^T C^{-1} (x_t - \phi(q_{t-1})) - W h, J^T C^{-1} J + W] \\ &= \mathcal{N}(q_t | q_t^{\text{MAP}}, A) \\ A &= (J^T C^{-1} J + W)^{-1} \\ q_t^{\text{MAP}} &= q_{t-1} + A [J^T C^{-1} (x_t - \phi(q_{t-1})) + W h] \end{aligned}$$

To rewrite q_t^{MAP} we can use the Woodbury identity

$$(J^T C^{-1} J + W)^{-1} J^T C^{-1} = W^{-1} J^T (J W^{-1} J^T + C)^{-1} \quad (7)$$

and get

$$q_t^{\text{MAP}} = q_{t-1} + J_{\text{WC}}^\# (x_t - \phi(q_{t-1})) + (J^T C^{-1} J + W)^{-1} W h.$$

Further, using the identity

$$\begin{aligned} \mathbf{I}_n &= (J^T C^{-1} J + W)^{-1} (J^T C^{-1} J + W) \\ &\Rightarrow (J^T C^{-1} J + W)^{-1} W = \mathbf{I}_n - (J^T C^{-1} J + W)^{-1} J^T C^{-1} J = \mathbf{I}_n - J_{\text{WC}}^\# J \end{aligned} \quad (8)$$

we get

classical view	Bayesian view
metric W of the pseudo-inverse $J_W^\# = W^{-1}J^T(JW^{-1}J^T)^{-1}$	covariance W^{-1} of the motion prior $\mathcal{N}(q_{t+1} q_t + h, W^{-1})$
nullspace motion $(\mathbf{I} - J_W^\# J) h$	asymmetry h of the motion prior $\mathcal{N}(q_{t+1} q_t + h, W^{-1})$
regularizer in the singularity-robust $J_W^\# = W^{-1}J^T(JW^{-1}J^T + k\mathbf{I}_d)^{-1}$	covariance C of the task coupling $\mathcal{N}(x_t \phi(q_t), C)$

Table 1 Correspondences between the classical and Bayesian view.

$$q_t^{\text{MAP}} = q_{t-1} + J_{WC}^\# (x_t - \phi(q_{t-1})) + (\mathbf{I}_n - J_{WC}^\# J) h .$$

■

The theorem gives two expressions (5) and (6) to compute q_t^{MAP} , related by the Woodbury identity. Note that the second expression (6) includes only d -dimensional matrix inversions rather than n -dimensional (neglecting W^{-1} , which can be precomputed). Thus, in practice the second expression will be more efficient to implement.

The second expression (6) also shows that the MAP motion q_t^{MAP} is very similar to classical kinematic motion rate control using the pseudo-inverse Jacobian (Nakamura & Hanafusa, 1986) – in the tight constraint limit $C \rightarrow 0$, $J_{WC}^\#$ coincides with the standard pseudo inverse $J_W^\# = W^{-1}J^T(JW^{-1}J^T)^{-1}$ and the two are equivalent. For non-zero covariance C (i.e., when loosening the task constraint) the MAP motion q_t^{MAP} corresponds to classical control with regularization in the computation of the pseudo-inverse Jacobian. In fact, a standard approach to deal with singularities (where JJ^T is not invertible) is to consider the so-called singularity-robust inverse (Nakamura & Hanafusa, 1986) $\tilde{J}_W^\# = W^{-1}J^T(JW^{-1}J^T + k\mathbf{I}_d)^{-1}$, which directly corresponds to the regularized pseudo-inverse $J_{WC}^\#$ as defined in (6). The regularizer can be interpreted as measuring the tolerance of the task constraint. The asymmetry h of the motion prior $\mathcal{N}(q_{t+1} | q_t + h, W^{-1})$ is the Bayesian counterpart of nullspace motion. Table 1 summarizes the relations between classical quantities and their Bayesian counterparts.

2.2 Multiple task variables

Theorem 1 directly extends to the case when we have multiple task variables $x_1, \dots, x_m$, with x_i d_i -dimensional, corresponding to different task mappings $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}^{d_i}$. The full joint (see Figure 1(b)) then reads

$$P(x_t, q_t, q_{t-1}) = \left[\prod_{i=1}^m P(x_{i,t} | q_t) \right] P(q_t | q_{t-1}) P(q_{t-1}) \quad (9)$$

where the motion prior is as before and for each task coupling

$$P(x_{i,t} | q_t) = \mathcal{N}(x_{i,t} | \phi_i(q_t), C_i) \quad (10)$$

we have a different task covariance C_i .

The extension can be subsumed in the previous derivation by introducing the joint random variable $x = (x_1, \dots, x_m)$ (d -dimensional with $d = \sum_i d_i$) and defining the covariance matrix $C = \text{diag}(C_1, \dots, C_m)$ to be the block matrix with sub-matrices C_i . Nevertheless, the explicit derivation allows us to establish interesting relations to classical prioritized inverse kinematics (Nakamura et al., 1987; Baerlocher & Boulic, 2004).

Corollary 1. *In the case of multiple task variables, as given by equations (9,10,2), the motion posterior is*

$$P(q_t | x_t, q_{t-1}) \approx \mathcal{N}(q_t | q_t^{\text{MAP}}, (\sum_{i=1}^m J_i^T C_i^{-1} J_i + W)^{-1}) \quad (11)$$

with the MAP motion

$$q_t^{\text{MAP}} = q_{t-1} + \left[\sum_{i=1}^m J_i^T C_i^{-1} J_i + W \right]^{-1} \left[\sum_{i=1}^m J_i^T C_i^{-1} (x_{i,t} - \phi_i(q_{t-1})) + W h \right] \quad (12)$$

The corollary follows directly from equation (5). The question of the classical limit is particularly interesting in the case of multiple variables. Let us investigate the case when we hierarchically require tightness in the task constraints. More specifically, one can iteratively take the limit $C_i \rightarrow 0$ starting with $i = 1$ up to $i = m$; in other terms this limit can be generated when defining $C_i = \varepsilon^{m-i} \mathbf{I}_{d_i}$ and simultaneously taking the limit $\varepsilon \rightarrow 0$. It turns out that this limit is exactly equivalent to prioritized inverse kinematics (Nakamura et al., 1987; Baerlocher & Boulic, 2004).

For $m = 2$ task variables one can prove the equivalence between prioritized inverse kinematics and the hierarchical classical limit of the MAP motion exactly (by directly applying the Woodbury identity). For $m > 2$ we could not find an elegant proof but we numerically confirmed this limit for up to $m = 4$.

Non-zero task variances can again be interpreted as regularizers. Note that without regularizers the standard prioritized inverse kinematics is numerically brittle. Handling many control signals (e.g., the over-determined case $\sum d_i > n$) is problematic since the nullspace-projected Jacobians will become singular (with rank $< d_i$). For non-zero C_i the computations in equation (12) are rather different to iterative nullspace projections and numerically robust.

2.3 Dynamic case

We address the case of dynamic motion control by moving to velocity space and considering the random variables $\dot{q}_t \in \mathbb{R}^n$ and $\dot{x}_t \in \mathbb{R}^d$, which refer to the joint velocities and task velocities, respectively. In addition to these variables, let $u_t \in \mathbb{R}^n$ be

a random variable that refers to a (torque) control signal we apply to the actuators. Consider the joint probability distribution

$$P(\dot{x}_t, \dot{q}_t, u_t, \dot{q}_{t-1}) = P(\dot{x}_t | \dot{q}_t) P(\dot{q}_t | u_t, \dot{q}_{t-1}) P(u_t) P(\dot{q}_{t-1}) \quad (13)$$

as also illustrated by the graphical model in Figure 1(c). Here, we call $P(u_t)$ the *control prior* and assume

$$P(u_t) = \mathcal{N}(u_t | h, H^{-1}), \quad (14)$$

where $h \in \mathbb{R}^n$ is a vector that induces an *asymmetry in the control prior* and H is a *control metric* which enters this prior in terms of its covariance H^{-1} . Further, $P(\dot{q}_t | u_t, \dot{q}_{t-1})$ is the *system dynamics* and we assume

$$P(\dot{q}_t | \dot{q}_{t-1}, u_t) = \mathcal{N}(\dot{q}_t | \dot{q}_{t-1} + M^{-1}(u_t + F), Q), \quad (15)$$

where M is the generalized mass matrix, $F \in \mathbb{R}^n$ the generalized force, and Q describes the control stochasticity. Finally, for the task coupling we assume

$$P(\dot{x}_t | \dot{q}_t) = \mathcal{N}(\dot{x}_t | J\dot{q}_t, C), \quad (16)$$

where the Jacobian $J = \frac{\partial \phi}{\partial q}$ relates the task and joint space velocities and, as before, C denotes the covariance in this coupling.

Bayesian dynamic control now computes the posterior control $P(u_t | \dot{x}_t, \dot{q}_{t-1})$ conditioned on the desired task velocity $\dot{x}_t$. We can derive the following result

Theorem 2. *Given equations (13-16), the posterior control is*

$$P(u_t | \dot{x}_t, \dot{q}_{t-1}) = \mathcal{N}(u_t | u_t^{MAP}, (T^T A^{-1} T + H)^{-1}) \quad (17)$$

$$T := JM^{-1}, \quad A := JQJ^T + C,$$

and the MAP control

$$u_t^{MAP} = (T^T A^{-1} T + H)^{-1} [T^T A^{-1} (\dot{x}_t - J\dot{q}_{t-1} - TF) + Hh] \quad (18)$$

$$= T_{HA}^\# (\dot{x}_t - J\dot{q}_{t-1} - TF) + (\mathbf{I}_n - T_{HA}^\# T) h, \quad (19)$$

$$T_{HA}^\# := H^{-1} T^T (TH^{-1} T^T + A)^{-1}. \quad (20)$$

Proof. We have

$$\begin{aligned} P(u_t | \dot{x}_t, \dot{q}_{t-1}) &\propto \int_{\dot{q}_t} d\dot{q}_t P(\dot{x}_t | \dot{q}_t) P(\dot{q}_t | \dot{q}_{t-1}, u_t) P(u_t) \\ &= \int_{\dot{q}_t} d\dot{q}_t \mathcal{N}(\dot{x}_t | J\dot{q}_t, C) \mathcal{N}(\dot{q}_t | \dot{q}_{t-1} + M^{-1}(u_t + F), Q) \mathcal{N}(u_t | h, H^{-1}) \\ &= \int_{\dot{q}_t} d\dot{q}_t \mathcal{N}[\dot{q}_t | J^T C^{-1} \dot{x}_t, J^T C^{-1} J] \mathcal{N}[\dot{q}_t | Q^{-1} \dot{q}_{t-1} + Q^{-1} M^{-1}(u_t + F), Q^{-1}] \mathcal{N}(u_t | h, H^{-1}). \end{aligned}$$

Applying the product rule (37) produces a Gaussian over $\dot{q}_t$ which integrates to 1. Using the short hand $A := JQJ^T + C$ we get

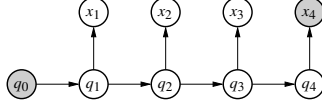


Fig. 2 The same graphical model as for redundant motion control (Figure 1(a)) but for multiple time slices.

$$\begin{aligned}
 P(u_t | \dot{x}_t, \dot{q}_{t-1}) &= \mathcal{N} \left[\dot{q}_{t-1} + M^{-1}(u_t + F) \mid J^T A^{-1} \dot{x}_t, J^T A^{-1} J \right] \mathcal{N}[u_t \mid Hh, H] \\
 &\propto \mathcal{N} \left[u_t \mid M^{-\top} J^T A^{-1} (\dot{x}_t - J \dot{q}_{t-1} - JM^{-1}F), M^{-\top} J^T A^{-1} JM^{-1} \right] \mathcal{N}[u_t \mid Hh, H] .
 \end{aligned}$$

Again applying the product rule (37) and using the short hand $T := JM^{-1}$ we get

$$\begin{aligned}
 P(u_t | \dot{x}_t, \dot{q}_{t-1}) &\propto \mathcal{N} \left[u_t \mid T^T A^{-1} (\dot{x}_t - J \dot{q}_{t-1} - TF) + Hh, T^T A^{-1} T + H \right] \\
 &= \mathcal{N}(u_t \mid u_t^{\text{MAP}}, B) \\
 B &= (T^T A^{-1} T + H)^{-1} \\
 u_t^{\text{MAP}} &= B[T^T A^{-1} (\dot{x}_t - J \dot{q}_{t-1} - TF) + Hh]
 \end{aligned}$$

Using the Woodbury identity as in (7) and (8) we can rewrite this expression as

$$u_t^{\text{MAP}} = T_{HA}^{\sharp} (\dot{x}_t - J \dot{q}_{t-1} - TF) + (\mathbf{I}_n - T_{HA}^{\sharp} T) h .$$

■

Again, the theorem provides two expressions (18) and (19) for u_t^{MAP} . In the classical limit $C \rightarrow 0$ and $Q \rightarrow 0$ (tight task constraint and zero control noise) the second expression directly retrieves the general optimal dynamic control law presented in (Peters et al., 2005),

$$u_t^{\text{MAP}} \stackrel{Q \rightarrow 0}{\underset{C \rightarrow 0}{=}} T_W^{\sharp} (\dot{x}_t - J \dot{q}_{t-1} - TF) + (\mathbf{I}_n - T_W^{\sharp} T) h . \quad (21)$$

Again, C can be understood as a regularizer for a singularity-robust matrix inversion. As discussed in (Peters et al., 2005), special choices of the control metric H in the dynamic control, e.g., $H = M^{-1}$, $H = M^2$, or $H = \mathbf{I}_n$ correspond to special classical control strategies. For instance, Khatib's operational space control follows from choosing $H = M^{-1}$.

3 A Bayesian view on motion planning

From the last two theorems we may conclude that the Bayesian approach applied to control in a single time-slice largely reproduces regularized classical solutions

in a different interpretation. In particular, in these ‘single time slice’ models one is typically only interested in the MAP solutions. The additional covariances we derived do not seem of much practical relevance.

The situation changes when we move from single time slice models of the immediate control to time extended models of the whole trajectory. The probabilistic inference approach naturally extends to inference over time and will become a planning method. For instance, we will consider inference in the model of Figure 2 as the direct temporal extension to Figure 1(a). The resulting posterior over $q_{1:4}$ will describe the set of likely trajectories starting in q_0 that are consistent with the constraint x_4 .

The inference techniques in such temporal models typically have the flavor of forward-backward algorithms, similar to inference in HMMs or Kalman smoothing in state-space models. A flexible description of such inference techniques is in terms of message passing algorithms. These algorithms also extend to more structured temporal models with many random variables in one time slice, as we will investigate them later. In most realistic cases exact inference is infeasible because the shape of the exact probability distributions would be very complex (not part of a simple parametric family of distributions). Again, message passing is a convenient framework to handle approximations systematically by maintaining approximate belief representations in the spirit of assumed density filtering or Expectation Propagation (Minka, 2001b).

A more detailed description of message passing in general factor graphs is given in (Toussaint, 2008), see also (Yedidia et al., 2001; Murphy, 2002; Minka, 2001a). Here we only give the message equations in pair-wise coupled networks: Given two random variables X_i and X_j coupled via a potential $f_{ij}(X_i, X_j)$, the message passing equations are

$$\mu_{j \rightarrow i}(X_i) = \sum_{X_j} f_C(X_i, X_j) \prod_{k: k \neq i} \mu_{k \rightarrow j}(X_j) , \quad (22)$$

where k indicates variables coupled to j other than i and, roughly speaking, the message $\mu_{j \rightarrow i}(X_i)$ is a distribution over the random variable X_i which encodes the information over X_i that results from its coupling to X_j . Given all incoming messages to a variable, the posterior marginal belief is given as the product of these,

$$b_i(X_i) := \prod_j \mu_{j \rightarrow i}(X_i) . \quad (23)$$

In the continuous case (as in the following) summations are replaced by integrals.

3.1 Kinematic case

We can now derive the messages for inference in the motion planning scenario. As before, let $q_t \in \mathbb{R}^n$ be a random variable referring to the robot’s joint state at time

t . And let x_t be a random variable referring to a task space of the robot (e.g. an endeffector state) at time t . We consider the joint probability distribution

$$P(x_{1:T}, q_{0:T}) = \left[\prod_{t=1}^T P(x_t | q_t) P(q_t | q_{t-1}) \right] P(q_0) \quad (24)$$

as also illustrated by the graphical model in Figure 2. We choose the motion prior and the task coupling exactly as before

$$P(q_t | q_{t-1}) = \mathcal{N}(q_t | q_{t-1} + h_t, W^{-1}) , \quad (25)$$

$$P(x_t | q_t) = \mathcal{N}(x_t | \phi(q_t), C_t) . \quad (26)$$

Please note that we have indexed the task covariance C_t with time – this means we can choose for each time t separately how tight we want to follow a given task constraint. In particular, in the typical planning case we might be interested only in the final endeffector position x_T ; in this formalism this corresponds to choosing $C_T \rightarrow 0$ very tight while choosing all other covariances $C_{1:T-1} \rightarrow \infty$ very large (they will drop out of the messages). The messages take the following form

Theorem 3. *Given equations (24 - 26) and using a local linearization of ϕ at $\hat{q}_t$ in time slice t , the message update equations read*

$$\begin{aligned} \mu_{q_{t-1} \rightarrow q_t}(q_t) &= \mathcal{N}(q_t | s_t, S_t) , \\ s_t &= h_t + (S_{t-1}^{-1} + R_{t-1})^{-1} (S_{t-1}^{-1} s_{t-1} + r_{t-1}) \\ S_t &= W^{-1} + (S_{t-1}^{-1} + R_{t-1})^{-1} \end{aligned} \quad (27)$$

$$\begin{aligned} \mu_{q_{t+1} \rightarrow q_t}(q_t) &= \mathcal{N}(q_t | v_t, V_t) , \\ v_t &= -h_t + (V_{t+1}^{-1} + R_{t+1})^{-1} (V_{t+1}^{-1} v_{t+1} + r_{t+1}) \\ V_t &= W^{-1} + (V_{t+1}^{-1} + R_{t+1})^{-1} \end{aligned} \quad (28)$$

$$\begin{aligned} \mu_{x_t \rightarrow q_t}(q_t) &\approx \mathcal{N}[q_t | r_t, R_t] , \\ r_t &= J^T C_t^{-1} (x_t - \phi(\hat{q}) + J\hat{q}) \\ R_t &= J^T C_t^{-1} J \end{aligned} \quad (29)$$

Further, given these messages, the belief $b_t(q_t)$ over the posture at time t reads

$$\begin{aligned} b_t(q_t) &= \mathcal{N}[q_t | b_t, B_t] , \\ b_t &= S_t^{-1} s_t + V_t^{-1} v_t + r_t , B_t = S_t^{-1} + V_t^{-1} + R_t \end{aligned} \quad (30)$$

Proof. Since all factors are pairwise we can use the expression (22) for the messages. We have

$$\begin{aligned} \mu_{q_{t-1} \rightarrow q_t}(q_t) &= \int_{q_{t-1}} dq_{t-1} P(q_t | q_{t-1}) \mu_{q_{t-2} \rightarrow q_{t-1}}(q_{t-1}) \mu_{x_{t-1} \rightarrow q_{t-1}}(q_{t-1}) \\ &= \int_{q_{t-1}} dq_{t-1} \mathcal{N}(q_t | q_{t-1} + h_t, W^{-1}) \mathcal{N}(q_{t-1} | s_{t-1}, S_{t-1}) \mathcal{N}[q_{t-1} | r_{t-1}, R_{t-1}] \end{aligned}$$

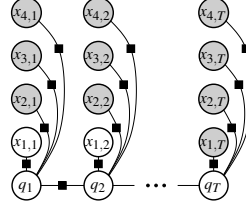


Fig. 3 Factor graph for Bayesian kinematic motion planning under multiple constraints.

Using the product rule (38) on the last two terms gives a Gaussian $\mathcal{N}(s_{t-1} | R_{t-1}^{-1} r_{t-1}, S_{t-1} + R_{t-1}^{-1})$ independent of q_t which we can subsume in the normalization. What remains is

$$\mu_{q_{t-1} \rightarrow q_t}(q_t) \propto \int_{q_{t-1}} dq_{t-1} \mathcal{N}[q_{t-1} | W(q_t - h_t), W] \mathcal{N}[q_{t-1} | S_{t-1}^{-1} s_{t-1} + r_{t-1}, S_{t-1}^{-1} + R_{t-1}^{-1}]$$

Using the product rule (37) and integrating over q_{t-1} we finally get

$$\mu_{q_{t-1} \rightarrow q_t}(q_t) = \mathcal{N}(q_t - h_t | [S_{t-1}^{-1} + R_{t-1}^{-1}]^{-1} [S_{t-1}^{-1} s_{t-1} + r_{t-1}], W^{-1} + [S_{t-1}^{-1} + R_{t-1}^{-1}]^{-1}).$$

The message $\mu_{q_{t+1} \rightarrow q_t}(q_t)$ can be derived in exactly the same way. Concerning $\mu_{x_t \rightarrow q_t}(q_t)$ equation (22) simplifies to

$$\mu_{x_t \rightarrow q_t}(q_t) = P(x_t | q_t) = \mathcal{N}(x_t | \phi(q_t), C_t)$$

Linearizing the task coupling at $\hat{q}_t$ we have

$$\begin{aligned} \mu_{x_t \rightarrow q_t}(q_t) &\approx \mathcal{N}(x_t | \phi(\hat{q}_t) + J(q_t - \hat{q}_t), C_t) \\ &= \frac{1}{|J|} \mathcal{N}[q_t | J^T C_t^{-1} (x_t - \phi(\hat{q}_t) + J\hat{q}_t), J^T C_t^{-1} J] \end{aligned}$$

Concerning the belief $b_t(q_t)$, by definition (23) we have

$$\begin{aligned} b_t(q_t) &= \mu_{q_{t-1} \rightarrow q_t} \mu_{q_{t+1} \rightarrow q_t} \mu_{x_t \rightarrow q_t} \\ &= \mathcal{N}[q_t | S_t^{-1} s_t, S_t^{-1}] \mathcal{N}[q_t | V_t^{-1} v_t, V_t^{-1}] \mathcal{N}[q_t | r_t, R_t] \end{aligned}$$

Applying the product rule (37) twice and neglecting normalization terms we have

$$b_t(q_t) \propto \mathcal{N}[q_t | S_t^{-1} s_t + V_t^{-1} v_t + r_t, S_t^{-1} + V_t^{-1} + R_t].$$

■

The theorem provides two recursive equations for the forward messages (27) and the backward messages (28). A standard algorithm would resolve the recursive equations by first iterating forward over time to compute the forward messages, and then iterate backward over time to compute the backward messages. However, an extra complication in our case is that in (29) we used a linearization of ϕ at $\hat{q}_t$ in each time slice. We will choose this point of linearization as follows: in the

Algorithm 1 Bayesian kinematic motion planning

```

1: Input: start posture  $q_0$ , metric  $W$ , motion prior  $h_t$ , task targets  $x_{0:T}$ , kinematic and Jacobian
   functions  $\phi, J$ 
2: Output: trajectory  $q_{0:T}$ 
3: initialize  $s_0 = q_0, S_0^{-1} = 10^{10}, v_{0:T} = 0, V_{0:T}^{-1} = 0, r_{0:T} = 0, R_{0:T} = 0, k = 0$ 
4: repeat
5:   for  $t = 1 : T$  do                                // forward sweep
6:     update  $s_t$  and  $S_t$  using (27)
7:     update  $v_t$  and  $V_t$  using (28)
8:     if  $k = 0$  then
9:        $\hat{q}_t \leftarrow s_t$ 
10:    else
11:       $\hat{q}_t \leftarrow (1 - \alpha)\hat{q}_t + \alpha b_t$ 
12:    end if
13:    update  $r_t$  and  $R_t$  using (29)
14:    update  $b_t$  and  $B_t$  using (30)
15:    if  $|\hat{q}_t - b_t|^2 > \theta$  then
16:       $t \leftarrow t - 1$                                 // repeat this time slice
17:    end if
18:  end for
19:  for  $t = T - 1 : 0$  do                                // backward sweep
20:    ..same updates as above...
21:  end for
22:   $k \leftarrow k + 1$ 
23: until convergence

```

first forward iteration we use the previous time step's MAP belief $\hat{q} = b_{t-1}(q_{t-1})$ while in subsequent backward and forward iterations we linearize at the old MAP belief $\hat{q}_t = b_t^{\text{old}}(q_t)$. Since the messages depend on the point of linearization we have to iterate the forward and backward sweeps until convergence. The algorithm is analogous to extended Kalman smoothing where observations are replaced by task constraints. The pseudo code is given in table 1.

As in the kinematic control case it is straightforward to extend these equations to the case of multiple task variables. The task coupling message then reads

$$r_t = R_t \hat{q} + \sum_i J_i^T C_{i,t}^{-1} (x_{i,t} - \phi_i(\hat{q})) , \quad R_t = \sum_i J_i^T C_{i,t}^{-1} J_i .$$

In this case we can choose different task variances $C_{i,t}$ for each task variables and in each time slot. This flexibility allows one to determine when and which task constraint has to be fulfilled by which precision. One can also follow the cascaded limit approach we mentioned in section 2.1 which in effect leads to a prioritization of the tasks, which might vary over time.

Finally, we note that the Bayesian motion planning scheme can be extended to the dynamic case in the same way as we extended the Bayesian control to the dynamic case. Due to the limited space we omit this derivation here; a general derivation can be found in (Toussaint, 2009).

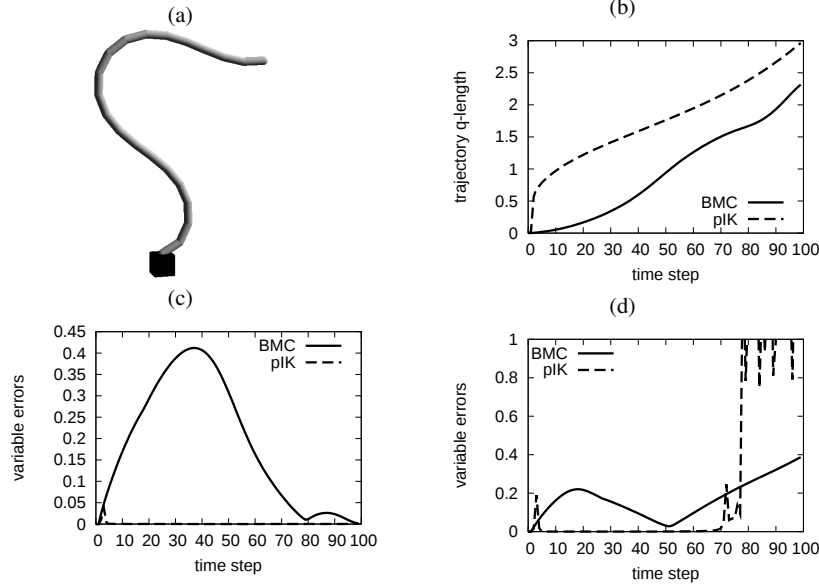


Fig. 4 Snake benchmark with four concurrent constraints. (b) Accumulative trajectory length $L = \sum_t |q_t - q_{t-1}|$ in q -space over time using prioritized inverse kinematic (pIK, dashed) and a soft Bayesian motion control (BMC, solid). (c) Total error $\sum_i e_i$ during the movement. (d) Errors during a movement towards an unreachable target.

4 Experiments

4.1 Kinematic control

Concerning motion control, we showed theoretically that Bayesian motion control is equivalent to regularized classical pseudo-inverse control. Our experiments confirm the close similarity to classical control. For illustration, we give examples on the simple snake benchmark proposed by (Baerlocher & Boulic, 2004) and specifically focus on “critical” situations where the regularization implicit in the Bayesian equations becomes apparent in comparison to non-regularized hierarchical control. The problem is to control a simulated planar snake configuration composed of 20 segments under four constraints: (1) the center of gravity should always remain vertically aligned with the foot, (2) the goal for the first segment (foot) is to be oriented with 30° from the ground, (3) the positional goal for the last segment (finger) is at $(.5, 0, 1)$, (4) the orientational goal for the last segment (finger) is to be oriented parallel to the ground. Figure 4(a) shows the snake in a valid target configuration. In (Baerlocher & Boulic, 2004) the problem is solved using prioritized inverse kinematics (pIK) with priority as listed above. We solve the problem by defining four variables $x_1, \dots, x_4$ according to the constraints above. For all task variables x_i we define that we want to follow a target that linearly interpolates from the start position

x_i^0 (straight upward posture) to the goal position x_i^* ,

$$x_{i,t} = \frac{t}{T}x_i^0 + \frac{T-t}{T}x_i^*.$$

In the first experiment we assumed rather tight and prioritized precisions $C_i^1 = v_i \mathbf{I}$ with $v_{1:4} = (10^6, 10^5, 10^4, 10^3)$. As a result, the joint trajectories generated with BMC and the classical prioritized inverse kinematics are virtually indistinguishable: the max norm $\|q_{1:T} - q'_{1:T}\|^\infty$ between the two trajectories is < 0.01 .

In the second experiment we assume that we require high task precision only at the final time step T . An efficient way to realize this with BMC is to start with rather soft precisions $v_{2:4} = 1$ at $t = 0$ and then slowly increasing them to the desired precision $v_{2:4} = (10^5, 10^4, 10^3)$ at $t = T$. As an exception we always keep the precision of the balance constraint high, $v_1 = 10^6$. The trajectory generated by BMC is now quite different from the previous one: it is much smoother. We can measure the quality of the trajectory in terms of the integrated length of the joint trajectory $L = \sum_t |q_t - q_{t-1}|$. Figure 4(b) shows pIK and BMC behavior very differently w.r.t. this measure. Nevertheless, all target variables meet the final goal constraint with high precision. This can be seen in Figure 4(b) which shows the errors $e_i = |x_i - x_i^*|$ in the control variables and the total error $E = \sum_i e_i$ during the movement for both approaches. In effect, the BMC tolerates larger errors during the movement (where we have only required loose task coupling) in favor of a shorter trajectory – but the final task constraints at time $t = T$ are met precisely.

As a final illustration we address conflicting and infeasible constraints. Assume we want to generate trajectories where the snake curvature is minimal, as measured by a fifth variable $x_5 = \sum_{i=1}^n q_i^2$ and a target $x_5^* = 0$. This curvature constraint is in conflict with most other constraints. As a result, the prioritized IK numerically breaks down when adding this fifth variable without further regularization. In contrast, BMC (with constant $v_{1:5} = (10^6, 10^1, 10^1, 10^1, 10^0)$) yields a smooth movement which eventually fulfills the targets for $x_{1:4}$ but additionally realizes a final curvature $e_5 = 1.57494$ much lower than without this additional constraint ($e_5 = 3.11297$). In another case, assume we set a target $(1, 0, .5)$ for the last segment (finger) which is actually out of reach. BMC (with constant $v_{1:4} = (10^6, 10^1, 10^1, 10^1)$) yields smooth and balanced postures where necessarily the error e_2 increases. As can be seen in Figure 4(d), classical pIK drastically diverges as soon as the hard constraint in finger position becomes infeasible (at $t = 75$).

4.2 Kinematic motion planning

We first investigate the kinematic motion planning algorithm on a reaching problem with a simulated humanoid figure (39 DoFs) as illustrated in Figure 5. We consider a trajectory of length $T = 200$. Starting from an upright posture (right image) the goal is to reach a target point (black dot) with the right finger while avoiding collisions

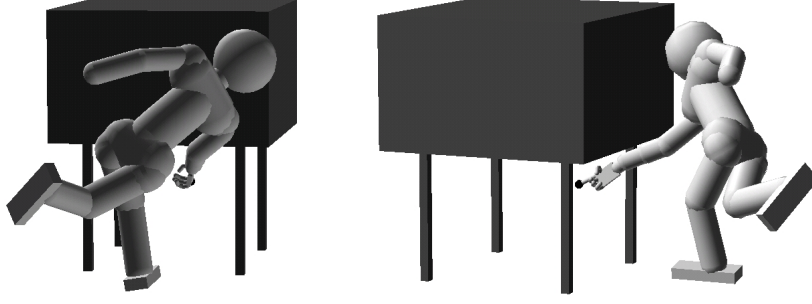


Fig. 5 Reaching test scenarios. The two images display the goal posture, the start posture is upright.

fwd-bwd iterations k	$\int \dot{q} dt$	$E = \int \sum_i e_{i,t} dt$
$\frac{1}{2}$ (reactive controller)	13.0124	0.0873
1	7.73616	0.1366
$1\frac{1}{2}$	7.70018	0.0071
2	7.68302	0.0027
5	7.65795	0.0013
10	7.62888	0.0012

Table 2 Trajectory length and control errors for Bayesian motion planning.

and keeping balance on the one foot rigidly anchored to the ground. We introduce three control variables for the finger tip (endeffector) position, the center of gravity, and a global collision cost. The desired motion is defined by trajectories $x_{i,1:T}$ for each control variable x_i . We defined these to be linear interpolations from the start state to the target with $T = 100$, while keeping the precisions $v_{1:3} = (10^3, 10^3, 10^6)$ constant over time. Table 2 displays the trajectory length and control errors after different numbers of forward-backward iterations of belief propagation for Bayesian motion planning. $k = \frac{1}{2}$ means a single forward pass and corresponds to the reactive application of the single time-slice Bayesian motion control. $k = 1$ additionally includes a single backward smoothing. For instance, if we fix the total computational cost to 3 times that of the Bayesian forward controller ($k = 1\frac{1}{2}$ iterations) we find an improvement of 40.8% w.r.t. the trajectory length and 91.9% w.r.t. control errors as compared to the forward controller. These improvements are due to the fact that the forward controller chooses a non-efficient path which first moves straight towards the obstacle and then needs a longer path to circumvent the obstacle. In contrast, the probabilistic smoothing of extended BMC leads to early nullspace movements (leaning to the right) which make the later circumvention of the obstacle more efficient.

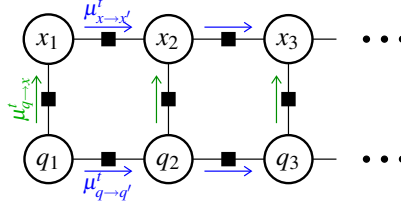


Fig. 6 The factor graph for the decomposed planning scenario.

4.3 Planning in more structured models

Consider the model in Figure 6 with factors

$$\begin{aligned}
 f_1(q_{t+1}, q_t) &= \mathcal{N}(q_{t+1} | q_t, W^{-1}) \\
 f_2(q_t) &= \mathcal{N}(q_t | 0, 1.0) \\
 f_3(x_{t+1}, x_t) &= \mathcal{N}(x_{t+1} | x_t, 0.1) \\
 f_4(x_t, q_t) &= \mathcal{N}(x_t | \phi(q_t), 0.001) \\
 f_5(x_T) &= \mathcal{N}(x_T | x_T^*, 0.001)
 \end{aligned} \tag{31}$$

The first factor is the usual motion prior in the joint state. The second factor places a prior $P(q_t)$ which limits the joint range – for simplicity we use again a Gaussian potential ($q = 0$ indicates the joint centers). The third factor expresses a prior about the endeffector movements – since we do not have a strong prior about endeffector movements we assume a weak potential with standard deviation of endeffector movements of 0.1. The fourth factor is the usual coupling between joint state and endeffector. Generally, in factor graphs conditioning a variable can be expressed as including a Kronecker factor. Hence, the fifth factor represents the goal constraint, conditioning the target of the endeffector to be close to x_T^* .

This model is different to the one of the previous section in two respects: We condition only on the final task state x_T , and we included a motion prior also within the task space. We investigate this graph because it allows for a qualitative new approach to decompose planning. In (Li et al., 2004; Todorov & Li, 2004) an algorithm was proposed for hierarchical planning. In the first stage it first computes an optimal trajectory only in the task space. In the second stage, constrained on this task space trajectory, it computes an optimal trajectory in joint space. Clearly, this approach is limited since the task space trajectory was computed without any concern whether this task space trajectory might lead to high costs when realized in joint space.

In the factor graph 6 we can follow a very similar approach to hierarchically decompose planning – but fully account for the mutual coupling of task and joint variables. Given the explicit model we can derive all necessary messages for belief propagation from equation (23). The algorithm we propose is given by the following message passing scheme:

1. Initialize all beliefs uniformly, except for q_0 , x_0 and x_T .

2. Update the task space beliefs

$$b(x_t) = \mu_{x_{t-1} \rightarrow x_t} \mu_{x_{t+1} \rightarrow x_t}^t \mu_{q_t \rightarrow x_t}^t ,$$

first iterating forward for $t = 1, \dots, T$, then iterating backward for $t = T-1, \dots, 1$.

This will yield a *preliminary belief* over possible trajectories *in task space*.

3. Update the q -space beliefs

$$b(q_t) = \mu^{q_{t-1} \rightarrow q_t} \mu_{q_{t+1} \rightarrow q_t} \mu_{x_t \rightarrow q_t}^t ,$$

first iterating forward for $t = 1, \dots, T$, then iterating backward for $t = T-1, \dots, 1$.

This procedure is exactly as described in the previous section, using local linearizations of the kinematics at $\hat{q}_t = \langle b(q_t) \rangle$. This generates a belief over possible trajectories in q -space.

4. Iterate steps (ii) and (iii).

Conceptually, the most interesting aspect is that in step (ii) we do not compute a *single* task space trajectory, but rather represent the *whole variety of possible task space trajectories* by the beliefs. The coupling to the q -space then narrows down this variety according to the prior in q -space. Iterating steps (ii) and (iii) means to propagate up ($\mu_{q_t \rightarrow x_t}^t$) and down ($\mu_{x_t \rightarrow q_t}^t$) messages between the x -level and the q -level until coherence between both levels is achieved.

4.3.1 Illustration on a planar arm

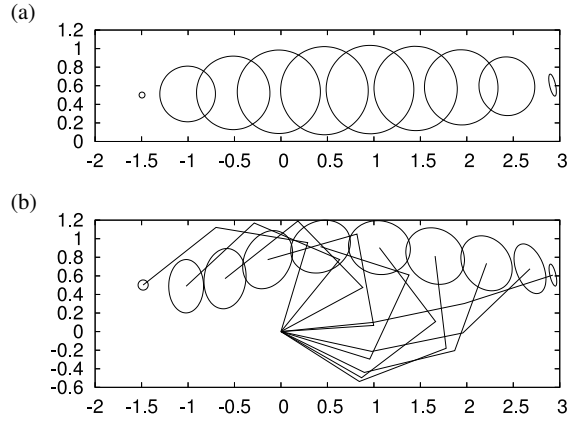


Fig. 7 (a) The belief over the endeffector trajectory after the first stage of inference – neglecting the coupling to the joint state. (b) The belief over the endeffector trajectory after coupling to the joint state; also the MAP joint configurations for each time step are displayed.

We would first like to illustrate the approach on a simple 3-link planar arm described by the joint state $q \in \mathbb{R}^3$ and the endeffector position $x \in \mathbb{R}^2$. We are given the initial configuration $q_0 = (.5, .5, .5)$, the endeffector target $x_T^* = (-1.5, .5)$ and $T = 20$.

Figure 7(a) displays the preliminary belief over endeffector states after the first stage of inference (step (ii)). We find that at this stage, the belief over the endeffector trajectory is simply a straight line with quite a large standard deviation associated with each via-point. This “Gaussian worm” can be interpreted as the range of possible endeffector trajectories neglecting the coupling to any other variables or constraints. All subsequent inference steps will further refine and narrow down this initial belief by fusing it with messages from the q -space. Figure 7(b) displays the belief over endeffector trajectories after a cycle of inference steps (ii), (iii), (ii), i.e., the coupling to the joint state has been accounted for. Also the MAP joint configuration is displayed at each time step. As expected from the discussion above, the MAP endeffector trajectory is not anymore a straight line. The reason is that the constraints we induced via the prior joint transition probability (31) favors small steps in joint space.

4.3.2 Illustration with a humanoid upper body

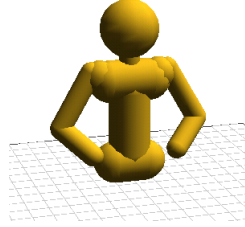


Fig. 8 A humanoid upper body with $n = 13$ hinge joints. The hip is fixed, the right hand serves as endeffector.

As another illustration, consider the $n = 13$ joint humanoid upper body displayed in Figure 8. We take the right hand as the endeffector and plan a target reaching trajectory (of length $T = 50$) to a goal in the upper left working domain of the robot.

Figures 9(a&b) display the result after 2 iterations of the inference steps (1-4), which provided sufficient convergence. The figures display the maximum a posteriori joint configuration (MAP, the maximum of the posterior joint state distribution) and the standard deviations of the endeffector distribution at different time steps. The MAP endeffector trajectory is not a straight line. To give a quantitative measure of the quality of the trajectory we compare the MAP joint trajectory computed via probabilistic inference with the joint trajectory that results from a standard redundant control approach. More precisely, the redundant control approach first pre-

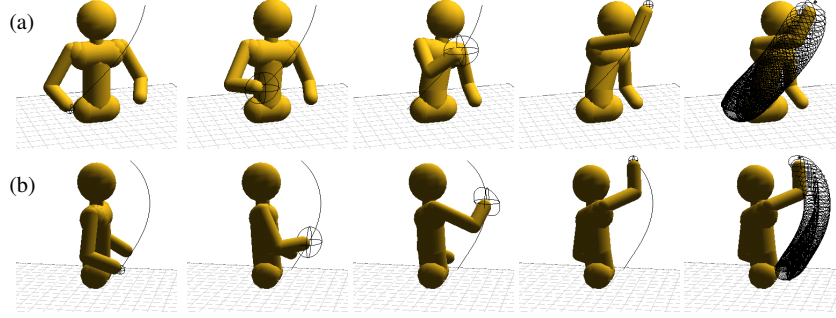


Fig. 9 Results of probabilistic inference planning with an humanoid upper body. Reaching to a target without obstacles, displayed from two different perspectives. We see the MAP joint configuration and the Gaussian endeffector distribution (indicated as ellipsoid) at different intermediate time steps. The optimal trajectory in joint space leads to a curve trajectory in endeffector space.

sumes a straight endeffector line equally divided in $T = 50$ segments and then uses standard motion rate control.

We define a global trajectory cost using the q -space metric W ,

$$C(q_{1..T}) = \sum_{t=1}^{T-1} \|q_{t+1} - q_t\|_W.$$

Table 3 displays the trajectory costs for the trajectories computed via the forward controller and the MAP trajectory computed via probabilistic inference. The MAP trajectory is clearly more efficient in terms of this cost. This stems from the fact that equation (31) imposes a prior transition likelihood $f_1(q_{t+1}, q_t) \propto \mathcal{N}(q, W^{-1})$ which reflects exactly this metric in joint space. The curve of the endeffector trajectory is a result of this efficiency.

	trajectory cost $C(q_{1..T})$
forward controller	11.19
MAP trajectory	8.14

Table 3 Global cost of joint space trajectories.

4.4 Coupling with collision constraints

In (Li et al., 2004; Todorov & Li, 2004) the idea of hierarchical decomposition of planning was considered, where a planning problem is first solved on a reduced task. This results is then coupled into a second-stage planning problem on the q -space

level. This strict two-stage procedure neglects that constraints (or priors) on the q -space level should eventually also influence the optimal task space plan. Intuitively one might come up with an algorithm that reiterates planning on both levels until their mutual interaction leads to a coherent plan on both levels. This is exactly what belief propagation does automatically in our framework.

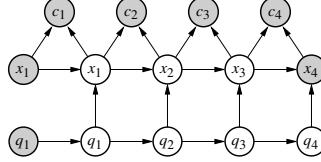


Fig. 10 Model for decomposed planning under collision constraints.

Let q_t and x_t be as before. In addition, let c_t be a binary random variable that indicates collision of the endeffector with an obstacle (a table in the following experiment). Consider the model

$$P(c_{1:T}, x_{0:T}, q_{0:T}) = P(x_0) P(q_0) \left[\prod_{t=1}^T P(c_t | x_t, x_{t-1}) P(x_t | q_t, x_{t-1}) P(q_t | q_{t-1}) \right] \quad (32)$$

as illustrated in Figure 10. We assume the motion prior $P(q_t | q_{t-1})$ as before and

$$P(c_t | x_t, x_{t-1}) = \mathcal{N}(c_t | \phi^c(x_t, x_{t-1}), D) , \quad (33)$$

$$P(x_t | q_t, x_{t-1}) \propto \mathcal{N}(x_{t+1} | x_t, C^{xx}) \mathcal{N}(x_t | \phi(q_t), C^{xq}) . \quad (34)$$

Here, ϕ^c is a function that determines the maximal penetration depth with the obstacle when the endeffector moves from x_{t-1} to x_t . We compute gradients for this as in (Toussaint et al., 2007). Further, note that we have now included a *task motion prior* $\mathcal{N}(x_{t+1} | x_t, C^{xx})$ as an additional factor in the model, even though this might be a rather weak prior (we will choose $C^{xx} = .1$ in the experiments).

In the specific experiment we condition on the final endeffector position $x_T = x^*$ and we condition on each collision variable $c_t = 0$ being zero. We perform inference in this model by a message passing scheme that effectively alternates between forward-backward inference on the task level and on the q -space level until a coherent posterior over both is found. We apply the following message passing scheme:

1. propagate forward & backward on x : first compute the messages $\mu_{x_{t-1} \rightarrow x_t}$ for $t = 1, \dots, T$, then the messages $\mu_{x_{t+1} \rightarrow x_t}$ for $t = T-1, \dots, 1$
2. compute all the messages $\mu_{c \rightarrow (x_{t-1}, x_t)}$ using local linearizations of ϕ^c at the current MAP task belief $b_t(x_t)$
3. propagate down: compute all the messages $\mu_{x_t \rightarrow q_t}$
4. propagate forward & backward on q : first compute the messages $\mu_{q_{t-1} \rightarrow q_t}$ for $t = 1, \dots, T$, then the messages $\mu_{q_{t+1} \rightarrow q_t}$ for $t = T-1, \dots, 1$

5. propagate up: compute all the messages $\mu_{q_t \rightarrow x_t}$
6. iterate steps (i)-(v)

In the first iteration, step (i) will compute a preliminary task space belief neglecting the collision constraint. In step (ii) the collision constraint is then coupled into the task space belief which is then in step (iii) propagated to the q -space belief, et cetera.

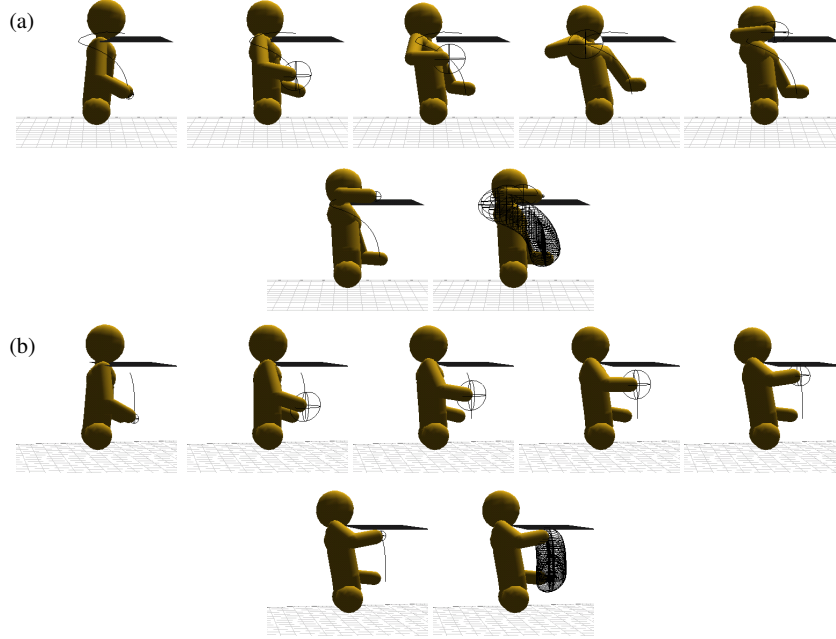


Fig. 11 Reaching to a target above (a) the table and below (b) the table whilst avoiding collision.

Figures 11(a&b) display the result after two iterations of this message passing scheme for $T = 30$. In the first case (Figure (a)) the target endeffector position is slightly above the table and the generated movement avoids the obstacle. In the second case, the target is only slightly displaced but now below the table. Here, the result is a rather straight trajectory. A standard forward controller that follows a gradient of a superimposed target potential and obstacle avoidance potential would typically get trapped under the table in the first case. Also, the local target potential of a reactive controller would hardly distinguish between the first and the second case.

The experiments ran on a laptop with a 1.1GHz Centrino Mobile processor. The first experiment ($T = 50$, without constraints, $k = 2$ inference sweeps) takes 3.56 seconds, the second experiment ($T = 50$, with constraints, $k = 2$ sweeps) takes 3.97 seconds.

5 Discussion

Let us discuss here specific aspects in relation to the derived algorithms and the Bayesian approach in general.

Local vs. global planning

An important aspect often discussed in the context of robotic planning is locality. Many classical approaches like RRTs (Kuffner & LaValle, 2000; Kuffner et al., 2003) try to tackle global planning problems, for instance where one first has to move away from a goal before moving towards the goal becomes possible. Theoretically, planning based on *exact* inference would also generate globally correct posterior distributions about *all* possible trajectories and thereby perform global planning. For discrete MDPs this is feasible and has been demonstrated (Toussaint et al., 2006). However, in almost any realistic robotics scenario exact inference is infeasible since this would require to represent very complex distributions (beliefs) which are not part of a feasible parametric family of distributions. In the algorithms we derived we assumed Gaussian belief representations and used local linearizations to stay in this family of belief representations. If we tried exact inference in domains with collision constraints, the exact beliefs had very complex, discontinuous forms. The Gaussian belief approximations effectively introduce a kind of “locality” since the set of likely trajectories is assumed close to a mean trajectory. Other kinds of belief representations would give a more global character to planning, e.g. sample based representations (particle filters) or mixture of Gaussians. In conclusion, it is very much a matter of which approximations and belief representations are chosen which determine how global the inference approach is.

Computational complexity

The complexity of inference is linear in the number of message computations needed; each message computation requires operations on *symmetric* matrices that approximately with n^2 . A single forward-backward pass along one variable is linear in T . If the number of neighbors to each variable is bounded, then the total number of edges in the factor graph is $O(TK)$, i.e., linear in the number K of variables. For complex (e.g., hierarchically deep) DBNs it is however an open question how many iterations of inference sweeps one needs until convergence.

Handling delayed feedback in control

In realistic cases the sensor feedback on the precise robot configuration (q_t and $\dot{q}_t$) is delayed. Thus, for instance in the case of torque control, the direct Bayesian dynamic control law (19) is not applicable. However, it is straight-forward to explicitly

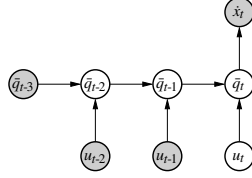


Fig. 12 Bayesian dynamic control in the case of delayed feedback. Here, $\bar{q}_t = (q_t, \dot{q}_t)$ subsumes positions and velocities.

include the delay in the probabilistic model and then apply Bayesian inference as we discussed it in the planning case. Figure 12 is an example for representing delayed feedback in the model – here the feedback is delayed for 2 time steps. The Bayesian control law is then given by computing $P(u_t | \bar{q}_{t3}, u_{t1}, u_{t2}, \hat{x}_t)$. In a sense, this approach naturally combines a probabilistic state estimation of $\bar{q}_{t1}$ with the ordinary Bayesian dynamic control.

6 Conclusion

Bayesian motion control and planning is based on the idea of *fusing motion objectives* (task constraints, goals, motion priors, etc) in a way similar to Bayesian sensor fusing. We formulate probabilistic models which represent these objectives, condition on constraints and goals, and use probabilistic inference to compute a posterior distribution over the possible trajectories and control signals. This distribution is a representation of the variety of likely trajectories and the corresponding control signals given that constraints and goals must be met.

The main contribution of this paper are derivations of explicit Bayesian control and planning algorithms. In the case of control we have addressed the problems of kinematic and dynamic task control by deriving a posterior distribution over the posture q_t and the control u_t , respectively. A straight-forward way to apply these results is to choose the MAP posture q_t^{MAP} as the next kinematic control point, or the MAP control u_t^{MAP} as the current control signal. We have shown that these MAP control laws are closely related to the classical control laws. More specifically, q_t^{MAP} corresponds to classical motion rate control including nullspace movement and a regularized pseudo-inverse Jacobian – where the regularizer is now interpreted as the tightness of the task constraint and the nullspace motion as the asymmetry of the motion prior. And u_t^{MAP} correspond to the classical dynamic control (as given, e.g., in (Peters et al., 2005)) generalized to include also a non-zero task variance C and a stochastic control variance Q .

To solve planning problems we extended the approach to multiple time slices, i.e. we proposed to use Bayesian inference to compute posterior trajectories conditioned on (future) constraints and goals. We derived explicit message passing algorithms for the basic cases of kinematic and dynamic planning. However, the general ap-

proach of inference and message passing algorithms can also be applied to more structured representations. Structure means that either the state q_t is factored into multiple variables (e.g., when we need to include variables describing objects or other properties of the environment) or that the task x_t is factored in multiple variables (the multi task variable scenario we mentioned is an example). In particular when the dependencies between such variables are sparse inference and message passing algorithms can exploit this structure to increase computational efficiency. This leads us away from the classical bottleneck of planning: the need to represent state in one big state space. Rather than deriving more explicit message passing algorithms for specific cases of structured models, the theorems showed how message equations can systematically be derived for specific graphical models. Our last experimental scenario addressed such an alternative model structure which is related to hierarchical planning where one alternates between planning in the task space and planning in the q -space. In section 5 we have also discussed in what sense *exact* inference would correspond to optimal global planning, whereas the more realistic case of approximate inference corresponds to more local planning. Future research on Bayesian motion planning should focus on exactly these two points: (1) exploiting the benefits of message passing algorithms in more structured models and (2) exploring different belief representation techniques (such as particle representations) for better approximations during inference.

Acknowledgments

This work was supported by the German Research Society (DFG), Emmy Noether fellowship TO 409/1-3.

Used Gaussian identities

We define a Gaussian over x with mean a and covariance matrix A as the function

$$\mathcal{N}(x|a, A) = \frac{1}{|2\pi A|^{1/2}} \exp\left\{-\frac{1}{2}(x-a)^T A^{-1} (x-a)\right\} \quad (35)$$

with property $N(x|a, A) = N(a|x, A)$. We also define the canonical representation

$$\mathcal{N}[x|a, A] = \frac{\exp\{-\frac{1}{2}a^T A^{-1} a\}}{|2\pi A^{-1}|^{1/2}} \exp\left\{-\frac{1}{2}x^T A x + x^T a\right\} \quad (36)$$

with properties

$$\mathcal{N}[x|a, A] = \mathcal{N}(x|A^{-1}a, A^{-1}) , \quad \mathcal{N}(x|a, A) = \mathcal{N}[x|A^{-1}a, A^{-1}] .$$

The product of two Gaussians can be expressed as

$$\mathcal{N}[x|a, A] \mathcal{N}[x|b, B] = \mathcal{N}[x|a+b, A+B] \mathcal{N}(A^{-1}a|B^{-1}b, A^{-1}+B^{-1}) , \quad (37)$$

$$\mathcal{N}(x|a, A) \mathcal{N}(x|b, B) = \mathcal{N}[x|A^{-1}a + B^{-1}b, A^{-1} + B^{-1}] \mathcal{N}(a|b, A + B), \quad (38)$$

$$\mathcal{N}(x|a, A) \mathcal{N}[x|b, B] = \mathcal{N}[x|A^{-1}a + b, A^{-1} + B] \mathcal{N}(a|B^{-1}b, A + B^{-1}). \quad (39)$$

Linear transformations in x imply the following identities,

$$\mathcal{N}(Fx + f|a, A) = \frac{1}{|F|} \mathcal{N}(x|F^{-1}(a - f), F^{-1}AF^{-\top}), \quad (40)$$

$$= \frac{1}{|F|} \mathcal{N}[x|F^T A^{-1}(a - f), F^T A^{-1}F], \quad (41)$$

$$\mathcal{N}[Fx + f|a, A] = \frac{1}{|F|} \mathcal{N}[x|F^T(a - Af), F^T AF]. \quad (42)$$

The joint Gaussian of two linearly dependent Gaussian variables reads

$$\mathcal{N}(x|a, A) \mathcal{N}(y|b + Fx, B) = \mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix} \middle| \begin{pmatrix} a \\ b + Fa \end{pmatrix}, \begin{pmatrix} A & A^T F^T \\ FA & B + FA^T F^T \end{pmatrix}\right) \quad (43)$$

Bibliography

- Baerlocher, P., & Boulic, R. (2004). An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*.
- Bui, H., Venkatesh, S., & West, G. (2002). Policy recognition in the abstract hidden markov models. *Journal of Artificial Intelligence Research*, 17, 451–499.
- Culotta, A., McCallum, A., Selman, B., & Sabharwal, A. (2007). Sparse message passing algorithms for weighted maximum satisfiability. *New England Student Colloquium on Artificial Intelligence (NESCAI)*.
- Howard, M., Klanke, S., Gienger, M., Goerick, C., & Vijayakumar, S. (2009). Methods for learning control policies from variable-constraint demonstrations. In *From motor to interaction learning in robots*. Springer.
- Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., & Inoue, H. (2003). Motion planning for humanoid robots. *Proc. 20th Int. Symp. Robotics Research (ISRR'03)*.
- Kuffner, J. J., & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. *Proc. of IEEE Int'l Conf. on Robotics and Automation*.
- Li, W., Todorov, E., & Pan, X. (2004). Hierarchical optimal control of redundant biomechanical systems. *26th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*.
- Minka, T. (2001a). A family of algorithms for approximate bayesian inference. PhD thesis, MIT.
- Minka, T. P. (2001b). Expectation propagation for approximate Bayesian inference. *Proc. of the 17th Annual Conf. on Uncertainty in AI (UAI 2001)* (pp. 362–369).
- Murphy, K. (2002). Dynamic bayesian networks: Representation, inference and learning. PhD Thesis, UC Berkeley, Computer Science Division.
- Nakamura, Y., & Hanafusa, H. (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement and Control*, 108.
- Nakamura, Y., Hanafusa, H., & Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *Int. Journal of Robotics Research*, 6.

- Peters, J., Mistry, M., Udwadia, F. E., Cory, R., Nakanishi, J., & Schaal, S. (2005). A unifying framework for the control of robotics systems. *IEEE Int. Conf. on Intelligent Robots and Systems (IROS 2005)* (pp. 1824–1831).
- Salaun, C., Padois, V., & Sigaud, O. (2009). Learning forward models for the operational space control of redundant robots. In *From motor to interaction learning in robots*. Springer.
- Tappen, M. F., & Freeman, W. T. (2003). Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. *IEEE Intl. Conference on Computer Vision (ICCV)*.
- Todorov, E., & Li, W. (2004). Hierarchical optimal feedback control of redundant systems. *Advances in Computational Motor Control IV*, Extended Abstract.
- Toussaint, M. (2008). Lecture notes: Factor graphs and belief propagation. <http://ml.cs.tu-berlin.de/~mtoussai/notes/>.
- Toussaint, M. (2009). Robot trajectory optimization using approximate inference. *Proc. of the 26rd Int. Conf. on Machine Learning (ICML 2009)*.
- Toussaint, M., Gienger, M., & Goerick, C. (2007). Optimization of sequential attractor-based movement for compact behaviour generation. *7th IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids 2007)*.
- Toussaint, M., & Goerick, C. (2007). Probabilistic inference for structured planning in robotics. *Int Conf on Intelligent Robots and Systems (IROS 2007)* (pp. 3068–3073).
- Toussaint, M., Harmeling, S., & Storkey, A. (2006). *Probabilistic inference for solving (PO)MDPs* (Technical Report EDI-INF-RR-0934). University of Edinburgh, School of Informatics.
- Vlassis, N., & Toussaint, M. (2009). Model-free reinforcement learning as mixture learning. *Proc. of the 26rd Int. Conf. on Machine Learning (ICML 2009)*.
- Yedidia, J., Freeman, W., & Weiss, Y. (2001). Understanding belief propagation and its generalizations.