Technische Universität Berlin

Forschungsberichte
der Fakultät IV – Elektrotechnik und Informatik

Bayesian inference for motion control and planning

Marc Toussaint

Bericht-Nr. 2007 - 22

ISSN 1436 - 9915

Bayesian inference for motion control and planning

Marc Toussaint

Machine Learning group, TU Berlin

Franklinstr. 28/29, FR 6-9, 10587 Berlin, Germany

December 21, 2007

Abstract. Bayesian motion control and planning is based on the idea of *fusing motion objectives* (constraints, goals, priors, etc) using probabilistic inference techniques in a way similar to Bayesian sensor fusing. This approach seems promising for tackling two fundamental problems in robotic control and planning: (1) Bayesian inference methods are an ideal candidate for fusing many sources of information or constraints – usually employed in the sensor processing context. Complex motion is characterised by such a multitude of concurrent constraints and tasks and the Bayesian approach provides a solution of which classical solutions (e.g., prioritised inverse kinematics) are a special case. (2) In the future we will require planning methods that are not based on representing the system state as one high-dimensional state variable but rather cope with structured state representations (distributed, hierarchical, hybrid discrete-continuous) that more directly reflect and exploit the natural structure of the environment. Probabilistic inference offers methods that can in principle handle such representations. Our approach will, for the first time, allow to transfer these methods to the realm of motion control and planning.

The first part of this technical report will review standard optimal (motion rate or dynamic) control from an optimisation perspective and then derive Bayesian versions of the classical solutions. The new control laws show that motion control can be framed as an inference problem in an appropriately formulated probabilistic model. In the second part, by extending the probabilistic models to be Markovian models of the whole trajectory, we show that probabilistic inference methods (belief propagation) yield solutions to motion planning problems. This approach computes a posterior distribution over trajectories and control signals conditioned on goals and constraints.

1 Introduction

Planning in high-dimensional robotic systems is a fundamental and critical problem when the goal are more autonomous, problem-solving robots in natural environments. Many behaviours that we would like robots to exhibit, such as autonomously reasoning about tools to manipulate objects, considering different paths to a goal, and generally coping with unexperienced situations, cannot be based on purely reactive behaviour and require some kind of reasoning. Real-world robotic environments are highly structured. The scalability of planning and reasoning methods to cope with complex problems in such environments crucially depends on exploiting this structure.

We propose a new approach to motion control and planning in robotics based on proba-

bilistic inference. The method uses structured probabilistic models to represent the scenario and efficient inference techniques (belief propagation) to solve planning problems. The idea is that Bayesian methods are ideal candidates to find solutions to problems combining multiple criteria or sources of information and hence are a standard tool for sensor processing and fusion. Similarly, we investigate Bayesian methods to solve a fusion problem on the motor level. Further, probabilistic inference techniques can be applied to structured representations and hence are a promising approach for coping with models that reflect the structuredness of natural environments.

In contrast most approaches to planning in robotics try to directly cope with the high-dimensional state space, using smart heuristics to find paths in this space, e.g., using probabilistic road maps, or Rapidly Exploring Random Trees (Kavraki, Svestka, Latombe, & Overmars 1996; Kuffner & LaValle 2000). Although these approaches provide practical solutions to many real-world applications (such as foot step planning), the fundamental question of scalability remains unsolved when the full high-dimensional state space has to be considered. In particular, structural knowledge about the system is not exploited. What we aim for is a new approach to planning based on a factored representation of state, i.e., based on models of the environment and the robot that use multiple variables to describe the current state. These variables might represent different "parts" of the state (such as different body parts, different objects, or different constraints), but they may also provide different levels of abstractions of the current state ("situations") as in hierarchical models.

Both fields, Reinforcement Learning (RL) and Probabilistic inference, developed techniques to address structured models. For instance, in the field of RL, they include work on Factored Markov Decision Processes (Boutilier et al. 1995; Koller & Parr 1999; Guestrin et al. 2003; Theocharous et al. 2004; Kveton & Hauskrecht 2005), abstractions (Hauskrecht et al. 1998), and relational models of the environment (Zettlemoyer et al. 2005). On the other hand, recent advances in inference techniques show how structure can be exploited both for exact inference as well as making efficient approximations. Examples are message passing algorithms (loopy BP, Expectation Propagation), variational approaches, approximate belief representations (particles, Assumed Density Filtering, Boyen-Koller) and arithmetic compilation (see, e.g., Minka 2001; Murphy 2002; Chavira et al. 2006).

Applied to the realm of planning, the latter inference techniques would provide a fundamentally new perspective on decomposing planning processes. The idea of using probabilistic inference for planning in Markov Decision Processes is not new. Ng, Parr, & Koller (1999) use inference techniques to efficiently compute the policy gradient for gradient-based policy search. From a complexity point of view, the equivalence between inference and planning is well-known (see, e.g., Littman et al. 2001). Bui, Venkatesh, & West (2002) have used inference on Abstract Hidden Markov Models for policy recognition, i.e., for reasoning about executed behaviours, but do not address the problem of computing optimal policies from such inference. Attias (2003) proposed a framework which suggests a straight-forward way to translate the problem of planning to a problem of inference: A Markovian state-action model is assumed, which is conditioned on a start state x_0 and a goal state x_T. However, this approach does not compute an optimal solution in the sense of maximising an expected future reward and the total time T has to be fixed. Raiko & Tornio (2005) introduced the same idea independently as optimistic inference control. Verma & Rao (2006) used inference to compute plans (considering the maximal probable explanation (MPE) instead of the MAP action sequence) but again the total time has to be fixed and the plan is not optimal in the expected return sense. Recently, (Toussaint & Storkey 2006; Toussaint, Harmeling, & Storkey 2006) developed a framework which exactly translates the problem of computing an optimal policy in a Markov Decision Process to a likelihood maximisation problem. The latter is then solved using inference in the E-step and a standard policy update in the M-step. Also the work by Wiegerinck, van den Broek, & Kappen (2006) is related to translating the problem of planning to a free energy minimisation problem.

Section 2 will briefly review classical control laws as a precursor to the reformulation in Bayesian terms presented in section 3. Once formulated this way it is straight-forward to extend the approach to probabilistically reason about whole trajectories. We will do this in section 4 and thereby derive algorithms for Bayesian motion planning. We conclude this report with a discussion in section 6.

2 Preliminaries: the loss function of classical control laws

In this section we will first review classical control laws as minimisation of a simple loss function. Since this loss function has a Bayesian interpretation it will be straight-forward to develop also a Bayesian view on these control laws. The Bayesian inference approach can then be generalised to what we actually aim for: motion planning in temporal probabilistic models, as discussed in section 4.

Let $x \in \mathbb{R}^d$ and $q \in \mathbb{R}^n$. Given x, consider the problem of finding q that minimises

$$L = \|x - Jq\|^2_{C^{-1}} + \|q\|^2_W - 2h^T W q \tag{1}$$

where $\|q\|^2_W = q^T W q$ denotes a norm and C and W are symmetric positive definite matrices. This loss function can be interpreted as follows: The first term "measures" how well a constraint $x = Jq$ is fulfilled relative to a covariance matrix C, the second term "measures" $q = 0$ with metric W, the third term "measures" the scalar product between q and h w.r.t. W.

The solution can easily be found by taking the derivative

$$\begin{aligned}
\frac{\partial L}{\partial q} &= -2(x - Jq)^T C^{-1} J + 2q^T W - 2h^T W \\
\frac{\partial L}{\partial q} &= 0 \quad \Rightarrow \quad q = (J^T C^{-1} J + W)^{-1} (J^T C^{-1} x + Wh)
\end{aligned}$$

Let us collect some identities which will allow us to rewrite this solution in several forms. The *Woodbury identity*

$$(J^T C^{-1} J + W)^{-1} J^T C^{-1} = W^{-1} J^T (J W^{-1} J^T + C)^{-1} ,$$

holds for any positive definite C and W. Further we have the identity

$$\mathbf{I}_n - (J^T C^{-1} J + W)^{-1} J^T C^{-1} J = (J^T C^{-1} J + W)^{-1} W . \tag{2}$$

We define the pseudo-inverse of J w.r.t. W as

$$J^\sharp_W = W^{-1} J^T (J W^{-1} J^T)^{-1}$$

and similar quantity as

$$J^\natural_C = (J C^{-1} J^T)^{-1} J^T C^{-1} .$$

Using these identities and definitions we can rewrite the solution in several forms:

$$\begin{aligned}
q &= (J^T C^{-1} J + W)^{-1} (J^T C^{-1} x + Wh) \\
&= (J^T C^{-1} J + W)^{-1} \; J^T C^{-1} \; x \; + \; [\mathbf{I}_n - (J^T C^{-1} J + W)^{-1} \; J^T C^{-1} J] \; h \\
&= (J^T C^{-1} J + W)^{-1} \; J^T C^{-1} \; (x - Jh) \; + \; h
\end{aligned} \tag{3}$$

$$= W^{-1}J^T\,(JW^{-1}J^T + C)^{-1}\,x \;+\; [\mathbf{I}_n - W^{-1}J^T\,(JW^{-1}J^T + C)^{-1}J]\,h \tag{4}$$
$$= W^{-1}J^T\,(JW^{-1}J^T + C)^{-1}\,(x - Jh) \;+\; h\;.$$

This also allows us to properly derive the following limits:

$$C \to 0: \qquad q = J_W^\sharp x + (\mathbf{I}_n - J_W^\sharp J)\,h \;\; = J_W^\sharp (x - Jh) + h \tag{5}$$
$$W \to 0: \qquad q = J_C^\natural x + (\mathbf{I}_n - J_C^\natural J)\,h \;\; = J_C^\natural (x - Jh) + h$$
$$W = \lambda \mathbf{I}_n: \qquad q = J^T(JJ^T + \lambda C)^{-1}x \;+ [\mathbf{I}_n - J^T(JJ^T + \lambda C)^{-1}J]\,h \tag{6}$$
$$C = \sigma \mathbf{I}_d: \qquad q = (J^TJ + \sigma W)^{-1}J^Tx + [\mathbf{I}_n - (J^TJ + \sigma W)^{-1}J^TJ]\,h$$

These limits can be interpreted as follows. $C \to 0$: we need to fulfil the constraint $x = Jq$ exactly. $C = \sigma\mathbf{I}_d$: we use a standard squared error measure for $x \approx Jq$. $W \to 0$: we do not care about the norm $\|q\|_W^2$ (i.e., no regularisation); but interestingly, the cost term h^TWq has a nullspace effect also in this limit. $W = \lambda\mathbf{I}_n$: we use a standard ridge as regulariser.

The simple loss function (1) has many applications, as we discuss in the following.

2.1 Ridge regression

In ridge regression, when we have d samples of n-dimensional inputs and 1D outputs, we have a minimisation problem

$$L = \|y - X\beta\|^2 + \lambda\|\beta\|^2$$

with a input data matrix $X \in \mathbb{R}^{d\times n}$, an output data vector $y \in \mathbb{R}^d$ and a regressor $\beta \in \mathbb{R}^n$. The first term measures the standard squared error (with uniform output covariance $C = \mathbf{I}_d$), the second is a regulariser (or stabiliser) as introduced by Tikhonov. The special form $\lambda\|\beta\|^2$ of the regulariser is called ridge. Replacing the previous notation with $q \rightsquigarrow \beta$, $x \rightsquigarrow y$, $J \rightsquigarrow X$, $C \rightsquigarrow \mathbf{I}_d$, $W \rightsquigarrow \lambda\mathbf{I}_n$, $h \rightsquigarrow 0$, equation (6) tells us the solution

$$\beta = X^T(XX^T + \lambda\mathbf{I}_d)^{-1}y\;.$$

In the Bayesian interpretation of ridge regression, the ridge $\lambda\|\beta\|^2$ defines a prior $\propto \exp\{-\frac{1}{2}\lambda\|\beta\|^2\}$ over the regressor β. The above equation gives the MAP β. Since ridge regression has a Bayesian interpretation, also standard motion rate control, as discussed shortly, will have a Bayesian interpretation.

2.2 Redundant motion rate control

Consider a robot with n DoFs and a d-dimensional endeffector ($d < n$). The current joint state is q. In a given state we can compute the end-effector Jacobian J and we are given a joint space potential $H(q)$. We would like to compute joint velocities $\dot q$ which fulfil the task constraint $\dot x = J\dot q$ while minimising the absolute joint velocity $\|\dot q\|_W^2$ and following the negative gradient $h = -W^{-1}\nabla H(q)$. The solution is readily given by equation (5) when replacing the previous notation $q \rightsquigarrow \dot q$, $x \rightsquigarrow \dot x$:

$$\dot q = J_W^\sharp \dot x - (\mathbf{I}_n - J_W^\sharp J)\,W^{-1}\nabla H(q)\;. \tag{7}$$

2.3 Redundant dynamic control

Consider a robot with dynamics $\ddot q = M^{-1}(u+F)$, where M is some generalised mass matrix, F subsumes external (also Coriolis and gravitational) forces, and u is the n-dimensional

torque control signal. We want to compute a control signal u which generates an acceleration such that a general task constraint $\ddot{x} = J\ddot{q} + \dot{J}\dot{q}$ remains fulfilled while also minimising the absolute norm $\|u\|_W^2$ of the control. Reformulating the constraint as $\ddot{x} - \dot{J}\dot{q} - JM^{-1}(u + F) = 0$, replacing the previous notation as $q \rightsquigarrow u$, $x \rightsquigarrow \ddot{x} - \dot{J}\dot{q} - JM^{-1}F$, $J \rightsquigarrow JM^{-1}$, and taking the strict constraint limit $C \rightsquigarrow 0$ we can consult equation (5) and get the solution

$$u = T_W^\sharp(\ddot{x} - \dot{J}\dot{q} - TF) + (\mathbf{I}_n - T_W^\sharp T)\, h\ , \quad \text{with } T = JM^{-1}\ , \tag{8}$$

which, for $h = 0$, is identical to Theorem 1 in (Peters et al. 2005). Peters et al. discuss in detail important special cases of this control scheme.

2.4 Multiple variables and prioritisation

The equations trivially extend to the case when we have multiple task variables $x_1, .., x_m$ with x_i d_i-dimensional, Jacobians $J_1, .., J_m$ for each task variable and, we want to minimise

$$L = \sum_i \|x_i - J_i q\|_{C_i^{-1}}^2 + \|q\|_W^2 + 2h^T W q\ .$$

This is the same as combining all task variables to a single variable $x = (x_1, .., x_m)$ (d-dimensional with $d = \sum_i d_i$) and defining the covariance matrix $C = \text{diag}(C_1, .., C_m)$ to be the block matrix with sub-matrices C_i. For instance, equation (3) decomposes into

$$\dot{q} = (\sum_i J_i^T C_i^{-1} J_i + W)^{-1}\ [\sum_k J_k^T C_k^{-1}\ \dot{x}_i + Wh]\ .$$

Classical prioritised inverse kinematics (Nakamura, Hanafusa, & Yoshikawa 1987; Baerlocher & Boulic 2004) addresses the multiple variable case from a different perspective. It assumes that the task variables $x_1, .., x_m$ are given strict priority (w.l.o.g. following our indexing). The control law is based on standard motion rate control but iteratively projects each desired task rate $\dot{x}_i$ in the remaining nullspace of all higher level control signals. Initialising the nullspace projection with $N_0 = \mathbf{I}$ and $\delta q_0 = 0$, the control law is defined by iterating for $i = 1, .., m$

$$\bar{J}_i = J_i N_{i\text{-}1} \tag{9}$$

$$\dot{q}_i = \dot{q}_{i\text{-}1} + \bar{J}_i^\sharp(\dot{x}_i - J_i\ \dot{q}_{i\text{-}1}) \tag{10}$$

$$N_i = N_{i\text{-}1} - \bar{J}_i^\sharp \bar{J}_i\ . \tag{11}$$

We call $\bar{J}_i$ a nullspace Jacobian which has the property that $\bar{J}_i^\sharp$ projects to changes in q that do not change control variables $x_{1,..,i\text{-}1}$ with higher priority. Also an additional nullspace movement h in the remaining nullspace of all control signals can be included when defining the final control law as

$$\widehat{\dot{q}} = \dot{q}_m + N_m h\ . \tag{12}$$

In effect, the first task rate $\dot{x}_1$ is guaranteed to be fulfilled exactly (up to 1st order). The second $\dot{x}_2$ is guaranteed to be fulfilled "as best as possible" given that $\dot{x}_1$ must be fulfilled, et cetera. Handling many control signals (e.g., the over-determined case $\sum d_i > n$) is problematic with this control law since the nullspace Jacobians $\bar{J}_i$ will become singular (with rank $< d_i$) and the computation of the pseudo-inverse needs extra regularisation.

This prioritised inverse kinematics can be retrieved from the general equation (3) when iteratively taking the limit $C_i \to 0$ starting with $i = 1$ up to $i = m$. This cascaded limit can for instance be generated when defining $C_i = \epsilon^{m-i}\mathbf{I}_{d_i}$ and simultaneously taking the limit $\epsilon \to 0$. We could not find an elegant proof for the equality between this limit and the above explained prioritised IK, but we confirmed this equivalence numerically for random matrices.

Figure 1: Graphical models of Bayesian control, (a) for motion rate control, (b) for dynamic control.

3 Bayesian motion control

The previous section derived basic classical control laws from a single loss function. We will now show that these control laws can also be understood in a Bayesian view: optimal control means to "fuse different objectives" in a Bayesian way – similar to how it is often done for sensor fusion. However, here the objectives are given by the possible transition in joint space and the desired endeffector position after a step. Indeed, the MAP solutions in this Bayesian approach will exactly reproduce the classical control laws as a special case. However, they generalise them in some respects (e.g., by the kind of control noise that can be taken into account) and they will lead the way to temporal reasoning about control: Bayesian motion planning.

In the following we will make use of some algebraic identities for Gaussians which are summarised in the footnote.[1]

3.1 Redundant kinematic control

Consider random variables $q_{t\text{-}1}$, q_t, x_t and their joint distribution according to Figure 1(a) and

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

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

$$\mathcal{N}(x \mid a, A) = \frac{1}{|2\pi A|^{1/2}} \exp\{-\frac{1}{2}(x-a)^T A^{\text{-}1} (x-a)\}$$

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

$$\mathcal{N}[x \mid a, A] = \frac{\exp\{-\frac{1}{2}a^T A^{\text{-}1} a\}}{|2\pi A^{\text{-}1}|^{1/2}} \exp\{-\frac{1}{2}x^T A\, x + x^T a\}$$

with properties

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

Here $\phi(q)$ is the task kinematics with Jacobian $J = \frac{\partial \phi}{\partial q}$. We want to compute the posterior over q_t given the previous state $q_{t\text{-}1}$ and the constraint x_t,

$$\begin{aligned}
P(q_t \,|\, x_t, q_{t\text{-}1}) &\propto P(x_t \,|\, q_t)\; P(q_t \,|\, q_{t\text{-}1}) \\
&= \mathcal{N}(x_t \,|\, \phi(q_t), C)\; \mathcal{N}(q_t \,|\, q_{t\text{-}1} + h, W^{\text{-}1}) \\
&\approx \mathcal{N}(x_t \,|\, \phi(q_{t\text{-}1}) + J(q_t - q_{t\text{-}1}), C)\; \mathcal{N}(q_t \,|\, q_{t\text{-}1} + h, W^{\text{-}1}) \\
&= \mathcal{N}(Jq_t \,|\, Jq_{t\text{-}1} + x_t - \phi(q_{t\text{-}1}), C)\; \mathcal{N}(q_t \,|\, q_{t\text{-}1} + h, W^{\text{-}1}) \\
&\propto \mathcal{N}[q_t \,|\, J^T C^{\text{-}1} J q_{t\text{-}1} + J^T C^{\text{-}1}(x_t - \phi(q_{t\text{-}1})),\, J^T C^{\text{-}1} J]\; \mathcal{N}[q_t \,|\, W q_{t\text{-}1} + Wh, W] \\
&\propto \mathcal{N}[q_t \,|\, (J^T C^{\text{-}1} J + W) q_{t\text{-}1} + J^T C^{\text{-}1}(x_t - \phi(q_{t\text{-}1})) - Wh,\, J^T C^{\text{-}1} J + W] \\
&= \mathcal{N}(q_t \,|\, a, A) \qquad (13) \\
&\qquad A = (J^T C^{\text{-}1} J + W)^{\text{-}1}\,, \quad a = q_{t\text{-}1} + A[J^T C^{\text{-}1}(x_t - \phi(q_{t\text{-}1})) + Wh] \\
q_t^{\text{MAP}} &= q_{t\text{-}1} + (J^T C^{\text{-}1} J + W)^{\text{-}1}[J^T C^{\text{-}1}(x_t - \phi(q_{t\text{-}1})) + Wh] \\
&\overset{C \to 0}{=} q_{t\text{-}1} + J_W^\sharp (x_t - \phi(q_{t\text{-}1})) + (\mathbf{I}_n - J_W^\sharp J)\, h
\end{aligned}$$

In the third line we use the linear approximation to the kinematics (as in the extended Kalman filter). The fifth line switches to canonical form of the Gaussians and the sixth line uses the rule for products of Gaussians (see the footnote). The MAP solution is the same as equation (3) and hence the limit $C \to 0$ also equals (5).

3.2 Redundant dynamic control

Consider the model in Figure 1(b) with

$$\begin{aligned}
&P(u_t) = \mathcal{N}(u_t \,|\, h, W^{\text{-}1}) \\
&P(\dot{q}_t \,|\, \dot{q}_{t\text{-}1}, u_t) = \mathcal{N}(\dot{q}_t \,|\, \dot{q}_{t\text{-}1} + M^{\text{-}1}(u_t + F), Q) \\
&P(\dot{x}_t \,|\, \dot{q}_t) = \mathcal{N}(\dot{x}_t \,|\, J\dot{q}_t, C)
\end{aligned}$$

We compute the posterior on u_t conditioned on a desired motion $\dot{x}_t$ and the previous motion $\dot{q}_{t\text{-}1}$,

$$\begin{aligned}
P(u_t \,|\, \dot{x}_t, \dot{q}_{t\text{-}1}) &\propto \sum_{\dot{q}_t} P(\dot{x}_t \,|\, \dot{q}_t)\; P(\dot{q}_t \,|\, \dot{q}_{t\text{-}1}, u_t)\; P(u_t) \\
&= \sum_{\dot{q}_t} \mathcal{N}(\dot{x}_t \,|\, J\dot{q}_t, C)\; \mathcal{N}(\dot{q}_t \,|\, \dot{q}_{t\text{-}1} + M^{\text{-}1}(u_t + F), Q)\; \mathcal{N}(u_t \,|\, h, W^{\text{-}1}) \\
&= \sum_{\dot{q}_t} \mathcal{N}[\dot{q}_t \,|\, J^T C^{\text{-}1} \dot{x}_t, J^T C^{\text{-}1} J]\; \mathcal{N}[\dot{q}_t \,|\, Q^{\text{-}1}\dot{q}_{t\text{-}1} + Q^{\text{-}1} M^{\text{-}1}(u_t + F), Q^{\text{-}1}]\; \mathcal{N}(u_t \,|\, h, W^{\text{-}1}) \\
&= \mathcal{N}\Big[\dot{q}_{t\text{-}1} + M^{\text{-}1}(u_t + F) \,|\; J^T A \dot{x}_t,\; J^T A J\Big]\; \mathcal{N}[u_t \,|\, Wh, W]\,, \quad A = (JQJ^T + C)^{\text{-}1} \\
&\propto \mathcal{N}\Big[u_t \,|\; M^{\text{-}T} J^T A(\dot{x}_t - J\dot{q}_{t\text{-}1} - JM^{\text{-}1}F),\; M^{\text{-}T} J^T A J M^{\text{-}1}\Big]\; \mathcal{N}[u_t \,|\, Wh, W] \\
&\propto \mathcal{N}\Big[u_t \,|\; T^T A(\dot{x}_t - J\dot{q}_{t\text{-}1} - TF) + Wh,\; T^T A T + W\Big]\,, \quad T = JM^{\text{-}1} \\
&= \mathcal{N}(u_t \,|\, b, B) \qquad (14) \\
&\qquad B = (T^T A T + W)^{\text{-}1}\,, \quad b = B[T^T A(\dot{x}_t - J\dot{q}_{t\text{-}1} - TF) + Wh] \\
u_t^{\text{MAP}} &= (T^T A T + W)^{\text{-}1}[T^T A(\dot{x}_t - J\dot{q}_{t\text{-}1} - TF) + Wh] \qquad (15) \\
&\overset{Q \to 0, C \to 0}{=} T_W^\sharp (\dot{x}_t - J\dot{q}_{t\text{-}1} - TF) + (\mathbf{I}_n - T_W^\sharp T)\, h
\end{aligned}$$

The last line shows that the classical dynamical law (8) is a special case of this Bayesian

setup for $Q \to 0$ and $C \to 0$. The solution (15) generalises this to account for control noise $Q \neq 0$ and potentially soft task constraints $C \neq 0$.

3.3 Summary

We derived two new control laws in the previous sections, *Bayesian motion rate control (BMC)*

$$\begin{aligned} q_t^{\text{MAP}} &= q_{t\text{-}1} + (J^T C^{\text{-}1} J + W)^{\text{-}1} [J^T C^{\text{-}1} (x_t - \phi(q_{t\text{-}1})) + Wh] \\ &= q_{t\text{-}1} + W^{\text{-}1} J^T (J W^{\text{-}1} J^T + C)^{\text{-}1} (x_t - \phi(q_{t\text{-}1})) + [\mathbf{I}_n - W^{\text{-}1} J^T (J W^{\text{-}1} J^T + C)^{\text{-}1} J]\, h \,, \end{aligned} \tag{16}$$

and *Bayesian dynamic control (BDC)*

$$\begin{aligned} u_t^{\text{MAP}} &= (T^T A T + W)^{\text{-}1} [T^T A (\dot{x}_t - J \dot{q}_{t\text{-}1} - TF) + Wh] \\ &= W^{\text{-}1} T^T (T W^{\text{-}1} T^T + A^{\text{-}1})^{\text{-}1} (\dot{x}_t - J \dot{q}_{t\text{-}1} - TF) + [\mathbf{I}_n - W^{\text{-}1} T^T (T W^{\text{-}1} T^T + A^{\text{-}1})^{\text{-}1} T]\, h \,, \end{aligned} \tag{17}$$

with $A = (JQJ^T + C)^{\text{-}1}$ and $T = JM^{\text{-}1}$. In the seconds lines we used the matrix identities to rewrite both equations to yield only inversions of d-dimensional matrices rather than n-dimensional ones. These two control laws have the classical motion rate control (7) and dynamic control (8) as limits,

$$\begin{aligned} \lim_{C \to 0} \; q_t^{\text{MAP}} &= q_{t\text{-}1} + J_W^\sharp (x_t - \phi(q_{t\text{-}1})) + (\mathbf{I}_n - J_W^\sharp J)\, h \\ \lim_{\substack{Q \to 0 \\ C \to 0}} \; u_t^{\text{MAP}} &= T_W^\sharp (\dot{x}_t - J \dot{q}_{t\text{-}1} - TF) + (\mathbf{I}_n - T_W^\sharp T)\, h \,. \end{aligned}$$

We can now summarize some correspondences between the classical and the Bayesian views in the following table

classical view	Bayesian view
metric W of the pseudo-inverse $J_W^\sharp = W^{\text{-}1} J^T (J W^{\text{-}1} J^T)^{\text{-}1}$	covariance W^{-1} of the transition prior $\mathcal{N}(q_{t+1} \,|\, q_t + h, W^{-1})$
nullspace motion $(\mathbf{I} - J_W^\sharp J)\, h$	asymmetry h of the transition prior $\mathcal{N}(q_{t+1} \,|\, q_t + h, W^{-1})$
regulariser in the singularity-robust $\tilde{J_W^\sharp} = W^{\text{-}1} J^T (J W^{\text{-}1} J^T + k \mathbf{I}_d)^{\text{-}1}$	covariance C of the task coupling $\mathcal{N}(x_t \,|\, \phi(q_t), C)$
computation of an optimal motion minimizing the loss (1)	computation of the motion posterior

As discussed in (Peters et al. 2005), special choices of regularisation W in the dynamic control, e.g., $W = M^{\text{-}1}$, $W = M^{\text{-}2}$, or $W = \mathbf{I}_n$ correspond to special classical control strategies, e.g., Khatib's operational space control for $W = M^{\text{-}1}$. Below we will also briefly discuss the close relation of Bayesian motion rate control to singularity-robust inverse kinematics. However, the Bayesian control laws not only computes the MAP control signals but also variance for the control signal in equations (13) and (14). Ordinarily these might not be of much practical relevance. However, they can be exploited when including other criteria for action selection. For instance, when the variance (as computed by (14)) along a certain dimension of the control signal u is very large this means that we do not require large precision in the control along this dimension.

3.3.1 Comparison to singularity-robust inverse kinematics

The Bayesian control laws are closely related to classical approaches to deal with singularities (where JJ^T is not invertible). One typically considers a singularity-robust inverse

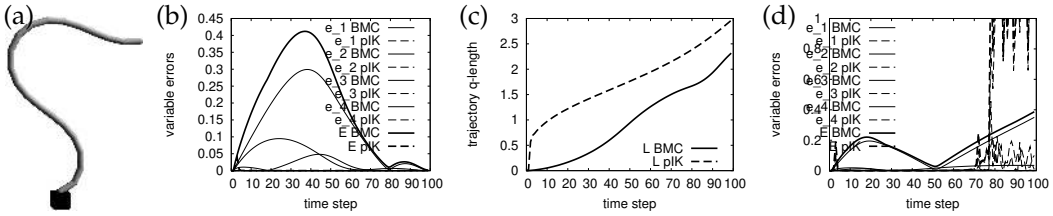

Figure 2: Snake benchmark with four concurrent constraints. (b) Errors $e_i = |x_i - x_i^*|$ in each control variable (thin lines) and the total error $\sum_i e_i$ (think lines) during the movement using prioritised inverse kinematic (pIK, dashed) and a soft BMC (solid). (c) Accumulative trajectory length in q-space over the movement. (d) Errors during a movement towards an unreachable target.

(Nakamura & Hanafusa 1986)

$$\tilde{J^\sharp_W} = W^{-1} J^T (J W^{-1} J^T + k\mathbf{I}_d)^{-1} ,$$

which is the standard pseudo-inverse with an additional regularisation. The scale factor k of the regularisation can heuristically be tuned to be non-zero in the neighbourhood of a singularity and zero when sufficiently far away from singular configurations. For instance, a common heuristic is

$$k = \begin{cases} k_0(1 - w/w_0)^2 & w < w_0 \\ 0 & w \geq w_0 \end{cases}$$

where w is some manipulability measure (e.g., $\sqrt{\det(JJ^T)}$ or the minimum singular value of J), and w_0 is a fixed threshold.

Clearly, this approach is equivalent to the Bayesian MAP solution when we reinterpret C as a regularisation of the pseudo-inverse. Note that the covariance C was initially be interpreted as determining the standard deviation from the task constraint we are willing to accept. We may conclude that the Bayesian approach applied to control in a single time-slice largely reproduces classical solution techniques in a different interpretation. However, the Bayesian approach generalises also to reasoning over many time slices, which we discuss in the next section and will lead to a Bayesian planning technique.

3.4 Basic illustration of Bayesian motion control

As these limits show, in standard situations Bayesian control will yield results very similar to the classical control laws. One can also think of the Bayesian laws as the classical laws with certain additional regularisers – as long as we are far from singularities the two behave very similar. In practise we will see a difference only in extreme cases. We illustrate this here on a simple snake benchmark proposed by (Baerlocher & Boulic 2004). The problem is to control a simulated planar snake configuration composed of 20 segments under four constraints: (1) the centre 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 2(a) shows the snake in a valid target configuration. (Baerlocher & Boulic 2004) solve the problem using prioritised IK with priority as listed above. We solve the problem by defining four variables $x_1, .., x_4$ 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^*,

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

We investigate the following cases:

Reproducing hierarchical prioritisation We first assumed rather tight and prioritised precisions $C_i^{-1} = \nu_i \mathbf{I}$ with $\nu_{1:4} = (10^6, 10^5, 10^4, 10^3)$. As a result, the joint trajectories generated with BMC and the classical prioritised inverse kinematics (9-12) are virtually indistinguishable: the max norm $\|q_{1:T} - q'_{1:T}\|^\infty$ between the two trajectories is < 0.01.

Smoother trajectories with BMC Assume that we require high task precision only at the final time step T. An efficient way to realise this with BMC is to start with rather soft precisions $\nu_{2:4} = 1$ at $t = 0$ and then slowly increasing them to the desired precision $\nu_{2:4} = (10^5, 10^4, 10^3)$ at $t = T$. As an exception we always keep the precision of the balance constraint high, $\nu_1 = 10^6$. The trajectory generated by BMC is now quite different from the previous one: it is much smoother. For instance, the integrated length of the joint trajectory $L = \sum_t |q_t - q_{t\text{-}1}|$ is $L = 2.31$ for BMC and $L = 2.96$ for pIK. Nevertheless, all target variables meet the final goal constraint with high precision. Figure 2(b) 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, and Figure 2(c) also plots the accumulative length L.

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 prioritised IK numerically breaks down when adding this fifth variable without further regularisation. In contrast, BMC (with constant $\nu_{1:5} = (10^6, 10^1, 10^1, 10^1, 10^0)$) yields a smooth movement which eventually fulfils the targets for $x_{1:4}$ but additionally realises 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 $\nu_{1:4} = (10^6, 10^1, 10^1, 10^1)$) yields smooth and but balanced postures where necessarily the error e_2 increases. As can be seen in Figure 2(d), classical pIK drastically diverges as soon as the hard constraint in finger position becomes infeasible (at $t = 75$).

Illustration on a Humanoid reaching task Figure 6 shows the posture BMC creates for a more challenging problem. A simulated humanoid figure (39 DoFs) is to reach a target point with the right finger under a big obstacle standing on one foot without loosing balance. In particular collisions between head or arm and the obstacle become critical. We introduced three control variables for the finger tip position, the centre of gravity, and a global collision cost. The naive application of prioritised IK fails since the Jacobian of the collision cost is highly non-stationary and most of the time singular (zero, in fact). Only additional heuristics to switch on and off the collision constraint at appropriate times can repair the classical approach. BMC (with $\nu_{1:3} = (10^3, 10^3, 10^6)$) robustly generates a movement from an upright initialisation to the posture seen in Figure 6.

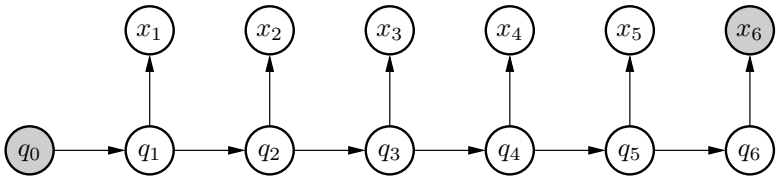

Figure 3: The same graphical model as for redundant motion control (Figure 1(a)) but for multiple time slices.

4 Bayesian motion planning

In the previous section we developed a Bayesian inference view on classical control laws. One may consider them as mere reformulations of well-known classical laws including some regularisations. In particular, in these 'single time slice' models one is typically only interested in the MAP solutions and the additional covariances we derived do not seem of much practical relevance.

This changes when we move on 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 eventually become a planning method. For instance, we discussed inference in Figure 1(a). What if we now do inference in Figure 3? The resulting posterior over $q_{1:6}$ will in some sense describe the set of likely trajectories starting in q_0 that are consistent with the constraint x_6.

The inference techniques in such temporal models typically have the flavour of forward-backward algorithms, similar to the well-known Baum-Welch in HMMs. A modern and very flexible description of such inference techniques is in terms of belief propagation or message passing algorithms. These also extend to more complex temporal models with many random variables in one time slice. In most realistic cases exact inference is infeasible because the shape of the probability distributions would be very complex (not part of a simple parametric family of distributions). Again, belief propagation is a convenient framework to handle approximations systematically by maintaining approximate belief representations.

In this section we introduce Bayesian motion planning. Technically, this mainly amounts to deriving the necessary messages for belief propagation in a factor graph describing the problem. Therefore we start with a brief introduction to factor graphs.

4.1 Probabilistic models to represent planning problems

Factor graphs (Kschischang, Frey, & Loeliger 2001) are structured probability distributions in the form

$$P(X_{1:n}) = \prod_C f_C(X_C) \ ,$$

where $C \subseteq \{X_1, .., X_n\}$ indexes different subsets of variables. Pictorially, such a structural property of the probability distribution is illustrated with a (bi-partite) graph (e.g., see Figure 4) where the random variables X_i are one type of nodes (circles), and the factors f_i are of another type (black boxes).

Clearly, Bayesian Networks (which are distributions structured in the form $P(X_{1:n}) = \prod_i P(X_i \,|\, X_{\text{parents}(i)})$) and also Dynamic Bayesian Networks as well as undirected graphical models (e.g., Markov Random fields) fall into this category. Belief propagation on

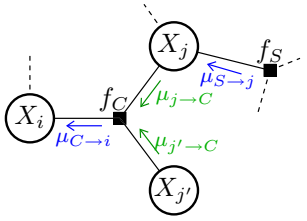

Figure 4: Part of a factor graph. Circle nodes represent random variables X_i, black boxes represent factor functionals $f_i = f_i(X_{C_i})$ where $C_i \subseteq \{X_1, .., X_n\}$ are subsets of variables, the graph as a whole represents the structure of the joint probability distribution $P(X_{1:n}) = \prod_i f_i(X_{C_i})$. Some messages illustrate the defining equations (18) and (19) of belief propagation.

factor graphs is defined by the following recursive equations for messages from factors to random variables and messages from random variables to factors:

$$\mu_{C\to i}(X_i) = \sum_{X_C \setminus X_i} f_C(X_C) \prod_{j\in C\setminus i} \mu_{j\to C}(X_j) \qquad = \frac{1}{\mu_{i\to C}(X_i)} \sum_{X_C\setminus X_i} b_C(X_C) \tag{18}$$

$$\mu_{j\to C}(X_j) = \prod_{\{S:\, j\in S\}\setminus C} \mu_{S\to j}(X_j) \qquad = \frac{1}{\mu_{C\to j}(X_j)}\, b_j(X_j) \tag{19}$$

with

$$b_j(X_j) = \prod_{S:\, j\in S} \mu_{S\to j}(X_j)\,, \quad b_C(X_C) = f_C(X_C) \prod_{j\in C} \mu_{j\to C}(X_j)\,. \tag{20}$$

To clarify this notation, assume $C = \{X_1, X_2, X_3\}$, then "$\sum_{C\setminus X_2} \equiv \sum_{X_1}\sum_{X_3}$" and "$\prod_{j\in C\setminus 2} \equiv \prod_{j\in\{1,3\}}$". When a factor depends only on two variables, $C = \{X_i, X_j\}$, we can simplify the equations by directly passing a message from j to i,

$$\mu_{j\to i}(X_i) = \sum_{X_j} f_C(X_i, X_j) \prod_{\{S:\, j\in S\}\setminus C} \mu_{S\to j}(X_j) \quad = \sum_{X_j} f_C(X_i, X_j) \frac{b_j(X_j)}{\mu_{i\to j}(X_j)}\,. \tag{21}$$

For chains and trees it is straight-forward to show that belief propagation eventually computes a belief b_i equal to the correct posterior marginal $\sum_{X_j: j\neq i} P(X_{1:n})$ (by comparison with standard variable elimination). For loopy graphs there is no order to resolve these recursive equations but BP can still be applied iteratively – and in practise often is – but convergence is not guaranteed and if it converges then only to a certain (Bethe) approximation of the true marginals (Yedidia, Freeman, & Weiss 2001). Further discussion of BP is beyond the scope of this paper, please refer to (Yedidia, Freeman, & Weiss 2001; Murphy 2002; Minka 2001) for more details. The given background should be sufficient to derive BP algorithms for the motion planning scenarios we discuss in the following.

4.2 Motion planning under multiple constraints

Consider the model in Figure 5 with

$$P(q_t \,|\, q_{t\text{-}1}) = \mathcal{N}(q_t \,|\, q_{t\text{-}1}, W^{\text{-}1})$$
$$P(x_{i,t} \,|\, q_t) = \mathcal{N}(x_{i,t} \,|\, \phi_i(q_t), C_i)\,.$$

The choice of these factors should be apparent from the analogy with the single time slice models and their relation to classical redundant motion rate control (section 2.2). All dependencies are pair-wise and the DBN amounts to a simple factor graph with only pair-wise

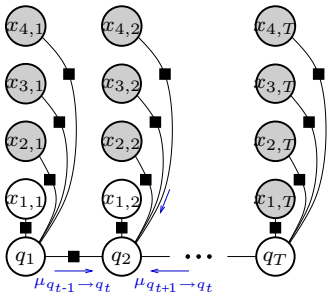

Figure 5: Factor graph for motion planning under multiple constraints. All coupling are pair-wise so we omit indicating the factor as black boxes.

factors. We represent the current belief over q_t as a Gaussian $b(q_t) = \mathcal{N}(q_t \,|\, \hat{b}_t, B_t)$. Using equations (20) and (21) we get the following recursive system of equations,

$$\begin{aligned}
\mu_{q_{t-1} \to q_t}(q_t) &= \textstyle\int_{q_{t-1}} P(q_t \,|\, q_{t-1})\; \mu_{x_{1:m,t-1} \to q_{t-1}}(q_{t-1})\; \mu_{q_{t-2} \to q_{t-1}}(q_{t-1}) = \mathcal{N}(q_t \,|\, a_t, A_t)\;, \\
A_t &= W^{-1} + (A_{t-1}^{-1} + R_{t-1})^{-1}\;, \quad a_t = (A_{t-1}^{-1} + R_{t-1})^{-1}(A_{t-1}^{-1} a_{t-1} + r_{t-1}) \\
\mu_{q_{t+1} \to q_t}(q_t) &= \textstyle\int_{q_{t+1}} P(q_{t+1} \,|\, q_t)\; \mu_{x_{1:m,t+1} \to q_{t+1}}(q_{t+1})\; \mu_{q_{t+2} \to q_{t+1}}(q_{t+1}) = \mathcal{N}(q_t \,|\, z_t, Z_t)\;, \\
Z_t &= W^{-1} + (Z_{t+1}^{-1} + R_{t+1})^{-1}\;, \quad z_t = (Z_{t+1}^{-1} + R_{t+1})^{-1}(Z_{t+1}^{-1} z_{t+1} + r_{t+1}) \\
\mu_{x_{1:m,t} \to q_t}(q_t) &= \prod_{i=1}^{m} P(x_{i,t} \,|\, q_t) \approx \mathcal{N}(q_t \,|\, R_t^{-1} r_t, R_t^{-1}) \qquad (22) \\
r_t &= R_t \hat{q} + \sum_i J_i^T C_i^{-1}(x_{i,t} - \phi_i(\hat{q}))\;, \quad R_t = \sum_i J_i^T C_i^{-1} J_i \\
b(q_t) &= \mu_{q_{t-1} \to q_t}\; \mu_{q_{t+1} \to q_t}\; \mu_{x_{1:m,t} \to q_t} = \mathcal{N}(q_t \,|\, \hat{b}_t, B_t) \qquad (23) \\
B_t^{-1} &= A_t^{-1} + Z_t^{-1} + R_t\;, \quad \hat{b}_t = B_t[A_t^{-1} a_t + Z_t^{-1} z_t + r_t]
\end{aligned}$$

Here, in equation (22), we used a linearisation of each ϕ_i at $\hat{q}$. As with all the algorithms we will derive, these equations tell us how to compute messages but they are recursive and we need to specify the order in which we update beliefs with these equations. Here we apply the following scheme:

- **(i)** Initialise all $b(q_t)$ to be the uniform density (we use precision matrices), and $b(q_0) = \mathcal{N}(q_0, 10^{-10})$.
- **(ii)** Update all $b(q_t)$ iterating forward $t = 1, .., T$ using equation (23). Linearise at $\hat{q} = \langle b^{\text{old}}(q_t) \rangle$, except for the first iteration, where we linearise at $\hat{q} = \mu_{t_{t-1} \to q_t}$.
- **(iii)** Update all $b(q_t)$ iterating backward for $t = T-1, .., 1$ using equation (23). Linearise at $\hat{q} = \langle b^{\text{old}}(q_t) \rangle$.
- **(iv)** Iterate (ii) and (iii). (The points of linearisation will refine.)

Before exemplifying the approach, let us summarise some properties of this Bayesian kind of motion planning.

(1) Clearly, BMC (equation (16)) is a special case of this motion planning for $T = 1$ and if not iterating the belief propagation. Iterating BP makes a difference since the point $\hat{q}$ of linearisation of kinematics moves towards the final mean $\hat{b}_1$ and thus the result of BP converges to an exact step rather than the usual 1st order approximation.

(2) This form of Bayesian motion planning allows us to continuously adjust the tightnesses C_i^{-1} of the control variables depending on time. For instance we can impose that a desired

Figure 6: Solution to a complex reaching movement under balance and collision constraints. The target for the right finger is illustrated as a black dot. Views from 3 different perspectives.

fwd-bwd iterations k	$\int \lvert\dot{q}\rvert \, dt$	$E = \int \sum_i e_{i,t} \, dt$
½ (reactive controller)	13.0124	0.0873
1	7.73616	0.1366
1½	7.70018	0.0071
2	7.68302	0.0027
5	7.65795	0.0013
10	7.62888	0.0012

Table 1: Trajectory length and control errors after different number of forward-backward iterations of extended BMC. $k = ½$ means a single forward pass and corresponds to the reactive forward controller using the single time slice BMC. $k = 1$ additionally includes a single backward smoothing.

control trajectory $x_i(t)$ is to be followed tightly in the early stage of the movement while only loosely in a later stage.

(3) This form of Bayesian motion planning solves a generalisation of redundant movement planning: Redundant planning means that a future goal is only determined by an endeffector constraint $x_T = x_T^*$ rather than a state space goal $q_T = q_T^*$. By adjusting the tightness (precision C^{-1}) of a control variable to be 0 for $t = 1, .., T$-1 but tight in the last time step $t = T$ we can reproduce the redundant planning problem. Our approach generalises this in that we can have an arbitrary number of control variables, and can adjust the control tightness e.g. to introduce intermediate goals or intermediate feasible ranges.

(4) Even when we require tightness in the controls over the full trajectory, the extended BMC solves a planning problem in the remaining nullspace.

4.2.1 Illustration on a Humanoid reaching task

We again consider the problem illustrated in Figure 6 of reaching under an obstacle while keeping balance. This time the desired motion is not defined by reactive dynamical systems but rather 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 $\nu_{1:3} = (10^3, 10^3, 10^6)$ constant over time. Table 1 displays the trajectory length and control errors after some forward-backward iterations. Note that the first line $k = ½$ (only one forward pass) corresponds to the reactive application of the single time-slice BMC we discussed in the previous section. For instance, if we fix the total computational cost to 3 times that of a simple forward controller ($k = 1½$ iterations) we find an improvement of 40.8% w.r.t. the trajectory length and 91.9% w.r.t. control errors of extended BMC versus the forward controller. The reason for these improvements are that the forward controller

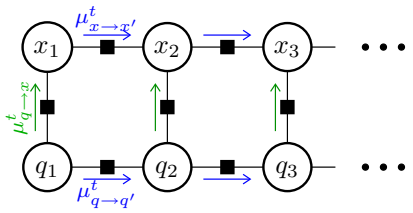

Figure 7: The factor graph for the decomposed planning scenario.

chooses a non-efficient path which first moves straight towards the obstacle and later need longer motions 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.

4.3 Decomposed planning

Consider the model in Figure 7 with factors

$$\begin{aligned} f_1(q_{t+1}, q_t) &= \mathcal{N}(q_{t+1} \,|\, q_t, W^{-1}) \\ f_2(q_t) &= \mathcal{N}(q_t \,|\, 0, 1.0) \\ f_3(x_{t+1}, x_t) &= \mathcal{N}(x_{t+1} \,|\, x_t, 0.1) \\ f_4(x_t, q_t) &= \mathcal{N}(x_t \,|\, \phi(q_t), 0.001) \\ f_5(x_T) &= \mathcal{N}(x_T \,|\, x_T^*, 0.001) \end{aligned} \tag{24}$$

The 1st factor is the usual transition prior in the joint state. The 2nd factor implies a prior $P(q_t)$ which limits the joint range – for simplicity we use again a Gaussian potential ($q = 0$ indicates the joint centres). The 3rd factor expresses a prior about the endeffector movements – since we do not have a strong prior about this, we assume a weak potential with standard deviation of endeffector movements of 0.1. The 4th 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 5th factor represents the goal constraint, conditioning the target of the endeffector to be close to x_T^*.

This model is different to the previously discussed one in two respects: We condition only on the final task state x_T, and we included a transition prior also within the task space. The reason we investigate this graph is because it allows for a qualitative new approach to decompose planning. Li, Todorov, & Pan (2004, Todorov & Li (2004) have proposed an algorithm for hierarchical planning which first computes and optimal trajectory only in the task space and then in a second stage, constraint on this task space trajectory, 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 realised in joint space.

In the factor graph 7 we can follow a very similar approach to hierarchically decompose planning – but guarantee a more global optimality. It is straight forward to derive all necessary messages for belief propagation from equation (20). The algorithm we propose is given by the following message passing scheme:

(i) Initialise all beliefs uniformly, except for q_0, x_0 and x_T.

(ii) Update the task space beliefs

$$b(x_t) = \mu^{x_{t-1}\to x_t}\ \mu^t_{x_{t+1}\to x_t}\ \mu^t_{q_t\to x_t}\ ,$$

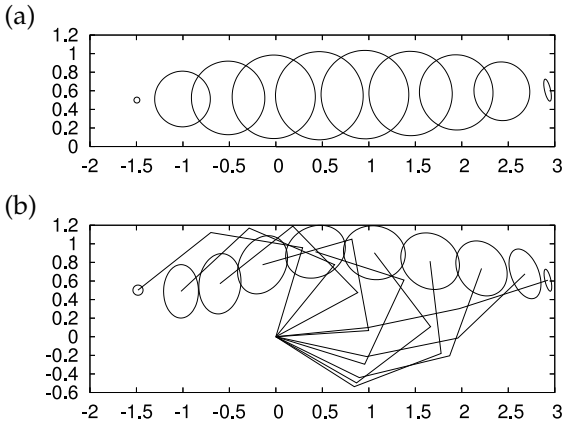

Figure 8: (a) The belief over the endeffector trajectory after the first stage of inference – neglecting the coupling to the joint state. (b) The belief over the endeffector trajectory after coupling to the joint state; also the MAP joint configurations for each time step are displayed.

first iterating forward for $t = 1, .., T$, then iterating backward for $t = T$-$1, .., 1$. This will yield a *preliminary belief* over possible trajectories *in task space*.

(iii) Update the q-space beliefs

$$b(q_t) = \mu^{q_{t\text{-}1} \to q_t} \; \mu_{q_{t+1} \to q_t} \; \mu^t_{x_t \to q_t} \; ,$$

first iterating forward for $t = 1, .., T$, then iterating backward for $t = T$-$1, .., 1$. This exactly as described in the previous section, using local linearisations of the kinematics at $\hat{q}_t = \langle b(q_t) \rangle$. This generates a belief over possible trajectories in q-space.

(iv) 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* as given 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 ($\mu^t_{q_t \to x_t}$) and down ($\mu^t_{x_t \to q_t}$) messages between the x-level and the q-level until coherence between both levels is achieved.

4.3.1 Illustration on a planar arm

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 8(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 Gaussian variance 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 8(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.

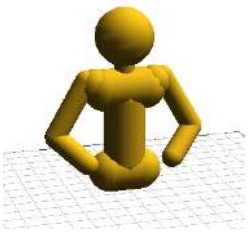

Figure 9: A humanoid upper body with $n = 13$ hinge joints. The hip is fixed, the right hand is used as endeffector.

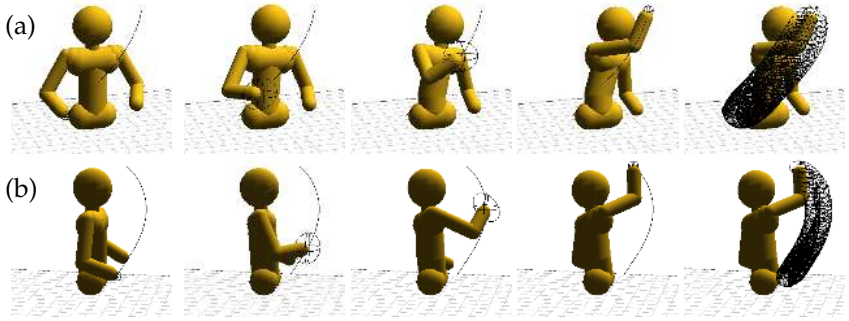

Figure 10: Results of probabilistic inference planning with an humanoid upper body. Reaching to a target without obstacles, displayed from two different perspectives. We see the MAP joint configuration and the Gaussian endeffector distribution (indicated as ellipsoid) at different intermediate time steps. The optimal trajectory in joint space leads to a curve trajectory in endeffector space.

Also the MAP joint configuration is displayed at each time step. We find that the MAP endeffector trajectory is not anymore a straight line. The reason is that the constraints we induced via the prior joint transition probability favours small steps in joint space.

4.3.2 Illustration with a humanoid upper body

As another illustration, consider the $n = 13$ joint humanoid upper body displayed in Figure 9. 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 10(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 variance of the endeffector distribution at different time steps. As we can see, the MAP endeffector trajectory is not a straight line. We can give a more 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 presumes a straight endeffector line equally divided in $T = 50$ segments and then used equation (7) for control.

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

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

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

Table 2: Global cost of joint space trajectories.

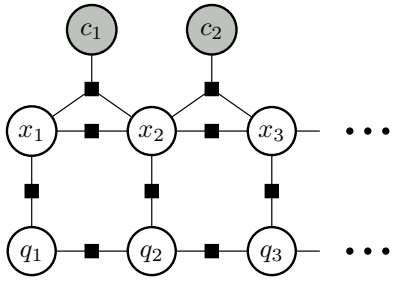

Figure 11: The factor graph including collision constraints.

Table 2 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 (24) 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.

4.3.3 Coupling with collision constraints

Coupling extra constraints into the system is straight-forward. To demonstrate this we augment the model with a collision risk variable c_t. In the experiment, we will consider collisions of the endeffector with a table. Figure 11 displays the factor graph augmented with this variable. Note that we decided to couple this variable *not* to a single endeffector position x_t but rather to an endeffector *transition* given by a tuple (x_t, x_{t+1}). This turned out to be more robust and accurate, particularly in view of the time discretisation used.

We define the risk variable with a non-linear sigmoidal function as follows. Let $z_t, y_t \in \mathbb{R}$ be z- and y-coordinate of the endeffector position $x_t \in \mathbb{R}^3$, let z^* be the table height. Then

$$c(x_{t+1}, x_t) = \begin{cases} 0 & \text{if } z_t, z_{t+1} > z^* + \delta \text{ or } z_t, z_{t+1} < z^* - \delta \\ \psi(y_t + y_{t+1}) & \text{otherwise} \end{cases}$$

with $\delta = .02$. Thus, the risk is zero if x_t and x_{t+1} are both either above or below the table, and a sigmoid depending on the y-distance to the table corner otherwise. We used $\psi(y) = 1 - \frac{1}{1+\exp(-3y)}$. Just as for the kinematics, this risk function defines a coupling factor

$$f_6(c_t, x_{t+1}, x_t) \propto \mathcal{N}(c(x_{t+1}, x_t), .001) \ .$$

Importantly, we also impose a prior on the risk variable, effectively constraining it to be low, by including the factor

$$f_7(c_t) \propto \mathcal{N}(0, .1) \ .$$

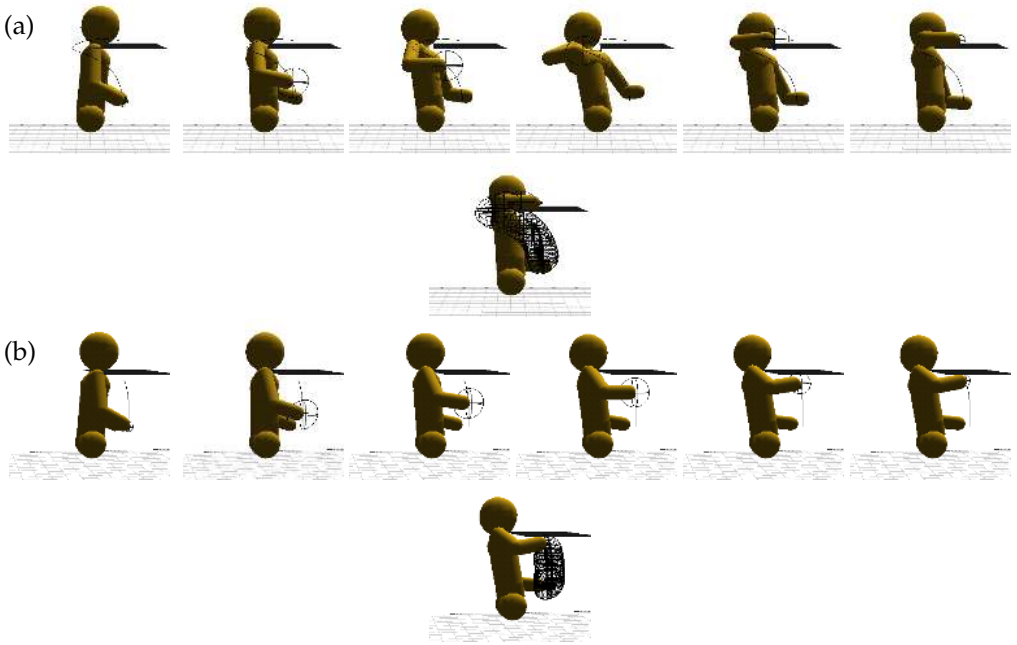

Figure 12: Reaching to a target above (a) the table and below (b) the table whilst avoiding collision.

The other factors $f_1, .., f_5$ remain the same as in the previous experiment, and together with f_6 and f_7 define the factor graph in Figure 11. Since f_6 is not pair-wise but depends on three variables, we maintain beliefs also for this factor. We apply the following message passing scheme: 11, we choose

(i) Initialise all beliefs uniform, except for q_0, x_0, x_T, and the collision variables c_t.

(ii) Update all task space beliefs $b(x_t)$ iterating forward ($t = 1, .., T$) and backward ($t = T\text{-}1, .., 1$).

(iii) Update all beliefs $b(c_t, x_t, x_{t+1})$ for the f_6 factors for $t = 1, .., T\text{-}1$.

(iv) Update all task space beliefs $b(x_t)$ iterating forward ($t = 1, .., T$) and backward ($t = T\text{-}1, .., 1$).

(v) Update all q-space beliefs $b(q_t)$ iterating forward ($t = 1, .., T$) and backward ($t = T\text{-}1, .., 1$).

(vi) Iterate steps (ii)-(v)

In the first iteration, step (ii) will compute a preliminary task space belief neglecting the collision constraint, since the f_6-beliefs are still uniform. In step (iv) the collision constraint is then coupled into the task space belief.

Figures 12(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 –

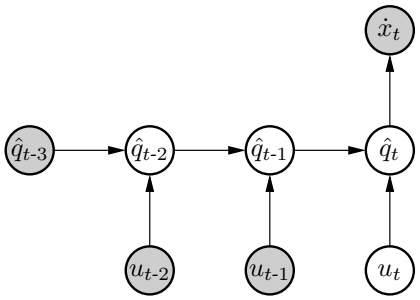

Figure 13: Dynamic control in the case of delayed feedback. Here, $\hat{q}_t = (q_t, \dot{q}_t)$ subsumes positions and velocities.

so we cannot present a quantitative comparison here. Also, the local target potential of a reactive controller would hardly distinguish between the first and the second case.

The experiments ran on a simple 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 Online application issues

5.1 Handling delayed feedback

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 (17) is not applicable. However, it is straight-forward to explicitly include the delay in the probabilistic model and then apply Bayesian inference as before. Figure 13 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 $P(u_t \mid \hat{q}_{t\text{-}3}, u_{t\text{-}1}, u_{t\text{-}2}, \dot{x}_t)$. In some sense, this approach naturally combines a probabilistic state estimation of $\hat{q}_{t\text{-}1}$ with the ordinary Bayesian dynamic control.

5.2 Online plan update

Assume we have applied Bayesian motion planning to compute a posterior over $q_{1:T}$ conditioned on q_0 and x_T. Further assume that we want to use motion rate control to follow this trajectory but the control is imprecise and might lead to perturbations form the MAP sequence $q^{\text{MAP}}_{1:T}$.

In the beginning, for $t = 0$, we may directly sent q^{MAP}_1 as the motion rate control. However, for $t = 1$ we have additional sensor information on the actual state q_1. To guarantee optimality we have to recompute the motion posterior conditioned on the new observation q_1.

Let us (unrealistically) assume that we have done *exact* inference without using approximations or linearisations. In this case, to compute the updated posterior motion over $q_{2:T}$ we only have to recompute the forward messages! The backward messages remain unchanged. This is a basic fact of inference over Markov chains. Further, for motion rate control at time $t = 1$ we are actually not interested in the exact posterior over the whole motion but only at the correct posterior over q_2. Hence, we only need to recompute a *single*

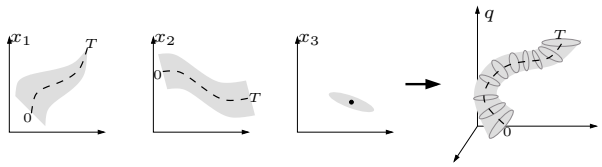

Figure 14: Illustration of BMP in analogy to Bayesian sensor fusion: On $m = 3$ low-dimensional control variables x_i we have desired trajectories and associated tolerances. Bayesian motion planning takes these as "input" and "fuses" them to a posterior distribution over possible joint space trajectories.

message $\mu_{q_1 \to q_2}$. To conclude, in the (unrealistic) case of *exact* inference *there is no need for replanning* because all the initially computed backward messages remain valid. This is fundamentally different from classical planning which only computes a single trajectory – the core of this difference is that inference computes full distributions which remain valid also when perturbed from the MAP path (at least the backward messages remain valid).

In the realistic case of approximate inference using approximate belief representations (e.g. Gaussians) and linearising the kinematics locally, the previous statement only holds approximately. We still compute full distributions instead of single trajectories which naturally lends to handling also perturbations from the MAP path. However, the backward messages do depend on the forward messages (and thereby on the start state conditioning) because of the iterated belief updates (e.g., the point of linearisation of the kinematics depends also on the forward messages). Hence, ideally we would have to recompute the full posterior when we get the new observation of q_1 by performing a full forward-backward propagation of messages. However, we may expect that many of the approximations (e.g., points of linearisations) hardly change during this update. In practise, it is reasonable to only update the first forward message $\mu_{q_1 \to q_2}$ as in the case of exact inference; and in certain intervals of time update all forward and backward messages to account for the perturbations. In conclusion, we might need some form of replanning occasionally when we expect that the perturbations accumulated so large that the approximations (linearisations) we have made to compute messages do not hold anymore.

6 Discussion

Bayesian motion control and planning is based on the idea of *fusing motion objectives* (constraints, goals, priors, etc) in a way similar to Bayesian sensor fusing, which is illustrated again in Figure 14. 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.

We have shown in section 3 that classical redundant motion rate control and dynamic control can be derived as special cases of Bayesian motion control when analogous probabilistic models are formulated. These models are one-time-slice models. By extending the approach to multiple time slices, i.e. inference over whole motion trajectories, we derived new motion planning algorithms. Since inference techniques exist also for factored (distributed) state representations we gained some freedom in formulating the probabilistic models. As examples we derived algorithms in the cases of multiple constraints and for the case of decomposed planning in task and joint space.

An important aspect often discussed in the context of robotic planning is locality. Many

classical approaches like RRTs (Kuffner & LaValle 2000; Kuffner, Nishiwaki, Kagami, Inaba, & Inoue 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, Harmeling, & Storkey 2006). However, in almost any realistic robotics case 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. For instance, in the algorithms we derived for BMP we assumed Gaussian belief representations and had to use local linearisations to stay in this simple family of belief representations. If we had tried exact inference in domains with collision constraints, the exact beliefs would have had very complex, discontinuous forms. Hence, in particular the Gaussian belief approximations effectively introduce a kind of "locality" since the set of likely trajectories is assumed close to some mean trajectory. If we would use other kinds of belief representations, e.g. sample based representations (particle filters) or mixture of Gaussians, the planning would have a "more global character". In conclusion, the inference approach is certainly not miraculously solving the problem of global planning. It is very much a matter of which approximations and belief representations are chosen which determine how global this approach is.

Acknowledgements

MT is grateful to Honda RI Europe for their hospitality as a guest scientist. MT also acknowledges support by the German Research Society (DFG), Emmy Noether fellowship TO 409/1-3.

References

Attias, H. (2003). Planning by probabilistic inference. In C. M. Bishop & B. J. Frey (Eds.), *Proc. of the 9th Int. Workshop on Artificial Intelligence and Statistics*.

Baerlocher, P. & R. Boulic (2004). An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*.

Boutilier, C., R. Dearden, & M. Goldszmidt (1995). Exploiting structure in policy construction. In *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI 1995)*, pp. 1104–1111.

Bui, H., S. Venkatesh, & G. West (2002). Policy recognition in the abstract hidden markov models. *Journal of Artificial Intelligence Research* **17**, 451–499.

Chavira, M., A. Darwiche, & M. Jaeger (2006). Compiling relational bayesian networks for exact inference. *International Journal of Approximate Reasoning* **42**, 4–20.

Guestrin, C., D. Koller, R. Parr, & S. Venkataraman (2003). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research (JAIR)* **19**, 399–468.

Hauskrecht, M., N. Meuleau, L. P. Kaelbling, T. Dean, & C. Boutilier (1998). Hierarchical solution of Markov decision processes using macro-actions. In *Proc. of Uncertainty in Artificial Intelligence (UAI 1998)*, pp. 220–229.

Kavraki, L., P. Svestka, J. Latombe, & M. Overmars (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* **12**, 566–580.

Koller, D. & R. Parr (1999). Computing factored value functions for policies in structured MDPs. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI 1999)*, pp. 1332–1339.

Kschischang, Frey, & Loeliger (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* **47**.

Kuffner, J., K. Nishiwaki, S. Kagami, M. Inaba, & H. Inoue (2003). Motion planning for humanoid robots. In *Proc. 20th Int. Symp. Robotics Research (ISRR'03)*.

Kuffner, J. J. & S. M. LaValle (2000). RRT-connect: An efficient approach to single-query path planning. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*.

Kveton, B. & M. Hauskrecht (2005). An MCMC approach to solving hybrid factored MDPs. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*.

Li, W., E. Todorov, & X. Pan (2004). Hierarchical optimal control of redundant biomechanical systems. In *26th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*.

Littman, M. L., S. M. Majercik, & T. Pitassi (2001). Stochastic boolean satisfiability. *Journal of Automated Reasoning 27*(3), 251–296.

Minka, T. (2001). A family of algorithms for approximate bayesian inference. PhD thesis, MIT.

Murphy, K. (2002). Dynamic bayesian networks: Representation, inference and learning. PhD Thesis, UC Berkeley, Computer Science Division.

Nakamura, Y. & H. Hanafusa (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement and Control* **108**.

Nakamura, Y., H. Hanafusa, & T. Yoshikawa (1987). Task-priority based redundancy control of robot manipulators. *International Journal of Robotics Research* **6**.

Ng, A. Y., R. Parr, & D. Koller (1999). Policy search via density estimation. In *Advances in Neural Information Processing Systems 12*, pp. 1022–1028.

Peters, J., M. Mistry, F. E. Udwadia, R. Cory, J. Nakanishi, & S. Schaal (2005). A unifying framework for the control of robotics systems. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS 2005)*, pp. 1824–1831.

Raiko, T. & M. Tornio (2005). Learning nonlinear state-space models for control. In *Proc. of Int. Joint Conf. on Neural Networks (IJCNN 2005)*.

Theocharous, G., K. Murphy, & L. Kaelbling (2004). Representing hierarchical POMDPs as DBNs for multi-scale robot localization. In *Intl. Conf. on Robotics and Automation (ICRA 2004)*.

Todorov, E. & W. Li (2004). Hierarchical optimal feedback control of redundant systems. Advances in Computational Motor Control IV, Extended Abstract.

Toussaint, M. & C. Goerick (2007). Probabilistic inference for structured planning in robotics. In *Int Conf on Intelligent Robots and Systems (IROS 2007)*.

Toussaint, M., S. Harmeling, & A. Storkey (2006). Probabilistic inference for solving (PO)MDPs. Research Report EDI-INF-RR-0934, University of Edinburgh, School of Informatics.

Toussaint, M. & A. Storkey (2006). Probabilistic inference for solving discrete and continuous state Markov Decision Processes. In *23nd Int. Conf. on Machine Learning (ICML 2006)*.

Verma, D. & R. P. N. Rao (2006). Goal-based imitation as probabilistic inference over graphical models. In *Advances in Neural Information Processing Systems 18 (NIPS 2005)*.

Wiegerinck, W., B. van den Broek, & H. Kappen (2006). Stochastic optimal control in continuous space-time multi-agent systems. In *Proc. of Conf. on Uncertainty in Artificial Intelligence (UAI 2006)*.

Yedidia, J., W. Freeman, & Y. Weiss (2001). Understanding belief propagation and its generalizations.

Zettlemoyer, L., H. Pasula, & L. P. Kaelbling (2005). Learning planning rules in noisy stochastic worlds. In *Proc. of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*.