# A Study of Morphological Computation by using Probabilistic Inference for Motor Planning

Elmar Rückert*
Institute for Theoretical Computer Science
Graz University of Technology
Graz, 8010, Austria
rueckert@igi.tugraz.at

Gerhard Neumann
Intelligent Autonomous Systems
Technische Universität Darmstadt
64289 Darmstadt, Germany
neumann@igi.tugraz.at

**Abstract**

One key idea behind morphological computation is that many difficulties of a control problem can be absorbed by the morphology of a robot. The performance of the controlled system naturally depends on the control architecture and on the morphology of the robot. Because of this strong coupling most of the impressive applications in morphological computation typically apply minimalistic control architectures. Ideally, adapting the morphology of the plant and optimizing the control law interact such that finally, optimal physical properties of the system and optimal control laws emerge. As a first step towards this vision we apply optimal control methods for investigating the power of morphological computation. We use a probabilistic optimal control method to acquire control laws given the current morphology. We show that by changing the morphology of our robot, control problems can be simplified, resulting in optimal controllers with reduced complexity and higher performance. This concept is evaluated on a compliant 4-link model of a humanoid robot, which has to keep balance in the presence of external pushes.

**Keywords:** Probabilistic motor planning, stochastic optimal control, morphological computation, compliant robots.

*Corresponding author.

# 1 Introduction

The control of compliant robots is inherently difficult due to their non-linear stochastic dynamics. Morphological computation poses the vision that parts of the complexity of a control or learning task can be outsourced to the morphology of the robot [11, 12]. This has been illustrated in many impressive biologically inspired robotic applications. For example Tedrake et al. [14] showed that the task complexity of learning to walk is drastically reduced by exploiting the dynamics of a passive biped walker. The authors in Iida and Pfeifer [7] demonstrated that the stability of a simple four-legged robot exhibits a surprisingly robust behavior using a mixture of active and passive joints. For swimming [20] or flying [19] robots investigations of the morphologies yield in rich behavioral diversity using only a single actuator exploiting the dynamics of the environment.

These approaches typically used minimalistic control architectures, like open loop actuation with sinusoidal drives [7, 20, 19] or a simple linear combination of state dependend features [14]. The morphology is assumed to absorb much of the computation needed for controlling the robot, and, therefore less emphasis is placed on the controller. More sophisticated control architectures were omitted as these are typically more difficult to fine tune for a new morphology. Thus, finding a good control law defines a individual optimization problem for each morphology, which renders the joint optimization of the morphology and the controller challenging. However, an autonomous system has to encompass for both aspects — the morphology and the control architecture — for efficient learning of complex motor skills.

One remarkable exception to this typically used minimalistic control architectures is the theoretical study of the morphological contribution for compliant bodies in Hauser et al. [5]. The authors have demonstrated in simulations for simple but nonlinear generic models of physical bodies, based on mass-spring systems that by outsourcing the computation to the physical body, the difficult problem of learning to control a complex body, could be reduced to a simple and perspicuous linear learning task. Thus, it can not get stuck in local minima of an error function. However, learning the optimal morphology and the optimal control law remains an open problem for many motor control tasks using real robots.

As a first step towards this vision we propose to directly use optimal control methods for eliminating the dependency between the morphology of the robot and the control architecture. Thus, for evaluating the computational power of the morphology, we will always use the optimal control law connected to this morphology. This allows for a fair comparison of different morphologies and simplifies the search for an optimal morphology.

For determining our optimal control law for a given morphology, we can use one of the many tools provided by Stochastic Optimal Control (SOC) methods such as Todorov and Li [15], Kappen [8], Toussaint [16]. These methods have been shown to be powerful approaches for movement planning in high-dimensional robotic systems. We will use Approximate Inference Control (AICO) [16], as it is a state of the art planning method for stochastic optimal control tasks. AICO is based on probabilistic inference for motor control. The beauty of this approach is that there is no distinction between sensor and motor, perception and action. We can include a multitude of variables, some of which might represent some features of the state, some of which might represent goals, constraints or motivations in the future and some of which might represent future actions or motor signals.

As all other stochastic optimal control methods, AICO minimizes a cost function, which is in our case given by the quadratic distance to a target state and the used energy of the movement. In order to apply the AICO method to torque constraint dynamical models we will briefly explain how to extend the algorithm to systems with control limits in Section 2.

Quantifying how much computation is done by the morphology of the robot and how much is done by the controller is often a difficult problem. Despite of the advancements for theoretical models of morphological computation [5], where the computational power of the plant can be assessed in a principled manner, this remains an open problem for many motor control tasks using real robots. Yet, by the use of optimal control laws the computation done by the morphology can be quantified by investigating the complexity of the controller. Thus, if the controller needs to do a lot of computation less computation is provided by the morphology in order to fulfill a task.

AICO also provides us with a time-varying linear feedback controller as policy to generate movement trajectories. We will use the variance of the control gains as complexity measure of the controller. If the control gains are almost constant in time, the control law is close to linear and does not perform much computation. However, if the control gains are highly varying, the controller needs to do a lot of computation and therefore, less computation is provided by the morphology.

As different complexity measures are possible we additionally use the final costs and the total jerk of a movement trajectory for a comparison. We illustrate the power of morphological computation combined with optimal control on a dynamical model of a humanoid robot (70kg, 2m). The robot is modelled by a 4-link pendulum, which has to keep balance in the presence of external pushes. We will show in Section 3 that by changing the morphology of a robot like the joint friction, the link lengths or the spring constants, the resulting optimal controllers have reduced complexity. As we will demonstrate there are optimal values for the physical properties for a given control task, which can only be found by the use of optimal control laws. With naive control architectures, different, sub-optimal morphologies would be chosen.

## 1.1 Related Work

We propose to use optimal control methods to eliminate the dependency between the control architecture and the morphology of the robot. These stochastic optimal control (SOC) methods such as Todorov and Li [15], Kappen [8], Toussaint [16] have been shown to be powerful methods for controlling high-dimensional robotic systems. For example the incremental Linear Quadratic Gaussian (iLQG) [15] algorithm is one of the most commonly used SOC methods. It uses Taylor expansions of the system dynamics and cost function to convert the non-linear control problem in a Linear dynamics, Quadratic costs and Gaussian noise system (LQG). The algorithm is iterative - the Taylor expansions are recalculated at the newly estimated optimal trajectory for the LQG system.

A SOC problem can be reformulated as inference problem in a graphical

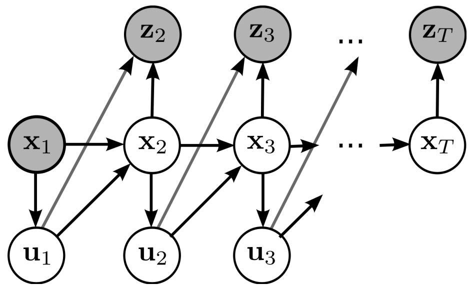



Figure 1: This figure illustrates the graphical model for probabilistic planning, where we consider finite horizon tasks with $T$ time steps. The state variable $\mathbf{x}_t$ denotes for example the joint angles and joint velocities of a robot. Controls are labelled by $\mathbf{u}_t$. The beauty of probabilistic inference for motor control is that we can include a multitude of variables, some of which might represent some features of the state, some of which might represent goals, constraints or motivations in the future and some of which might represent future actions or motor signals. These constraints are expressed in the model by the task variables $\mathbf{z}_t$.

model [16, 6], which has the nice property that there is no distinction between sensor and motor, perception and action. The unknown quantities, i.e. the states and actions can be efficiently inferred using for example the Approximate Inference Control (AICO) algorithm [16]. The graphical model is given by a simple dynamic Bayesian network with states $\mathbf{x}_t$, actions $\mathbf{u}_t$ and task variables $\mathbf{z}_t$ (representing the costs) as nodes, see Figure 1. In this graphical model observations are denoted by shaded nodes in contrast to the unknown random variables, which are indicated by circles. If beliefs in the graphical model are approximated as Gaussian the resulting algorithm is very similar to iLQG. Gaussian message passing iteratively re-approximates local costs and transitions as LQG around the current mode of the belief within a time slice. A difference to iLQG is that AICO uses forward messages instead of a forward roll-out to determine the point of local LQG approximation and can iterate belief re-approximation with in a time slice until convergence, which may lead to faster overall convergence [16].

The dynamic Bayesian network shown in Figure 1 is fully specified by the conditional distributions encoded by the cost function and by the state transition model. Like most SOC methods [15, 8, 16] we assume that the cost function and the state transition model are known. However, for model learning many types of function approximators can be applied [18, 9, 10]. For learning the cost function inverse reinforcement learning methods such as Abbeel and Ng [1], Boularias et al. [3] could be used. Since we demonstrate in this paper how morphological computation and optimal control can benefit from each other in general, an extension to the more complex learning problems remains for future research.

The original formulation of the AICO method [16] does not consider torque limits, which are important for many robotic experiments as well for the dynamic balancing experiments we consider in this paper. Therefore we extended the algorithm, which is discussed in the next section.

# 2 Probabilistic Inference for Motor Planning

We use the probabilistic planning method Approximate Inference Control (AICO) [16] as optimal control method. Most applications of AICO are in the kinematic planning domain. Here, we want to apply AICO to fully dynamic, torque controlled robot simulations. Therefore we had to extend the AICO framework with control or torque limits, which will be explained in the next subsections.

## 2.1 Approximate Inference Control

We will briefly clarify the notation for our discussion. Let $\mathbf{x}_t$ denote the state and $\mathbf{u}_t$ the control vector at time step $t$. A trajectory $\tau$ is defined as sequence of state control pairs, $\tau = \langle \mathbf{x}_{1:T}, \mathbf{u}_{1:T-1} \rangle$, where $T$ is the length of the trajectory. Each trajectory has associated costs

$$L(\tau) := \sum_{t=1}^{T} c_t(\mathbf{x}_t, \mathbf{u}_t), \tag{1}$$

where $c_t(\mathbf{x}_t, \mathbf{u}_t)$ represents the cost function for a single time step, which is in our case given by the quadratic distance to a target state and the used energy of the movement. Solely the cost function $c_{1:T}(\cdot)$ and the initial state $\mathbf{x}_1$ are known to the optimal control algorithm. The unknown trajectory $\tau$ is the result of the inference process.

AICO uses message passing in graphical models to infer the trajectory $\tau$. In order to transform the minimization of $L(\tau)$ into an inference problem, for each time step an individual binary random variable $z_t$ is introduced. This random variable indicates a reward event. Its probability is given by

$$P(z_t = 1 | \mathbf{x}_t, \mathbf{u}_t, t) \propto \exp(-c_t(\mathbf{x}_t, \mathbf{u}_t)).$$

AICO now assumes that a reward event $z_t = 1$ is observed at every time step, see Figure 1. Given that evidence, AICO calculates the posterior distribution $P(\mathbf{x}_{1:T}, \mathbf{u}_{1:T} | z_{1:T} = 1)$ over trajectories.

We will use the simplest version of AICO, where an extended Kalman smoothing approach is used to estimate the posterior $P(\mathbf{x}_{1:T}, \mathbf{u}_{1:T} | z_{1:T} = 1)$. The extended Kalman smoothing approach uses Taylor expansions to linearize the system and subsequently uses Gaussian messages for belief propagation in a graphical model. Gaussian message passing iteratively re-approximates local costs and transitions as a Linear dynamics, Quadratic costs and Gaussian noise system (LQG) around the current mode of the belief within a time slice. Concise derivations of the messages for AICO are given in the Appendix, as they are used to further extend the algorithm to implement control constraints.

AICO provides us with a linear feedback controller for each time slice of the form

$$\mathbf{u}_t = \mathbf{O}_t \mathbf{x}_t + \mathbf{o}_t,$$

where $\mathbf{O}_t$ is the inferred feedback control gain matrix and $\mathbf{o}_t$ denotes the linear feedback controller term. For our evaluations we also use a complexity measure which is proportional to the variance of this feedback control law:

$$\mathrm{Var}(\mathbf{O}_{1:T-1}, \mathbf{o}_{1:T-1}) = \sum_{i=1}^{D^x} \sum_{j=1}^{D^u} \mathrm{var}(\mathbf{O}_{i,j,1:T-1}) + \sum_{j=1}^{D^u} \mathrm{var}(\mathbf{o}_{j,1:T-1}), \tag{2}$$

where the dimension of the states is denoted by $D^x$ and the dimension of the controls is denoted by $D^u$.

AICO is only a local optimization method and we have to provide an initial solution which is used for the first linearization. We will simply use the first state as initialization $\mathbf{x}_{2:T} = \mathbf{x}_1$.

If we use AICO with a constant cost and dynamic model for each time step, the algorithm reduces to calculating a Linear Quadratic Regulator (LQR), which is often used in optimal control. A LQR is the optimal linear feedback controller for a given linear system. In contrast, AICO uses a time-varying linear feedback controller, which may be different for each time step. In our experiments we will compare both approaches on a dynamic non-linear balancing task. Thus we evaluate the benefit of using AICO (time-varying linear feedback control) and a LQR (constant linear feedback control).

### 2.2 Cost Function for Constrained Systems

In order to apply the AICO algorithm to torque controlled robots, we have to extend the framework to incorporate control limits as the available motor torque is typically limited. This is done by adding a control dependent punishment term $c_t^u(\mathbf{u}_t)$ to the cost function in Equation 1. Thus for a trajectory $\tau$ we specify the costs

$$L(\tau) := \sum_{t=1}^{T} c_t(\mathbf{x}_t, \mathbf{u}_t) + c_t^u(\mathbf{u}_t).$$

We use this additional term to punish controls $\mathbf{u}_t$ that exceed a given bound $\mathbf{u}_{B_t}$ at time $t$:

$$c_t^u(\mathbf{u}_t) = (\mathbf{u}_t - \mathbf{u}_{B_t})^T \mathbf{H}_{B_t} (\mathbf{u}_t - \mathbf{u}_{B_t}).$$

As a consequence, the resulting Gaussian distributions which are used to represent the costs $c_t$ change. This distributions have typically zero mean in the control space due to the typically used quadratic control costs. In the case of control limits the mean of the distribution is *non-zero*. Consequently, also the message passing update equations used for the AICO algorithm changes. The exact message passing equations of AICO with control limits are presented in the Appendix.

## 3 Experiments

We investigate morphological computation combined with optimal control on dynamic non-linear balancing tasks [2], where a robot gets pushed with a specific force and has to move such that it maintains balance. The optimal strategy is a non-linear control law which returns the robot to the upright position. For our evaluations different initial postures and multiple forces are used as sketched in Figure 2. We study the morphological computation by changing different physical properties of a 4-link model of a humanoid robot. In particular we investigate the computation done by different friction coefficients and the link lengths. Inspired by the work of Hauser et al. [5] we will also incorporate springs to our model for analyzing different spring constants.

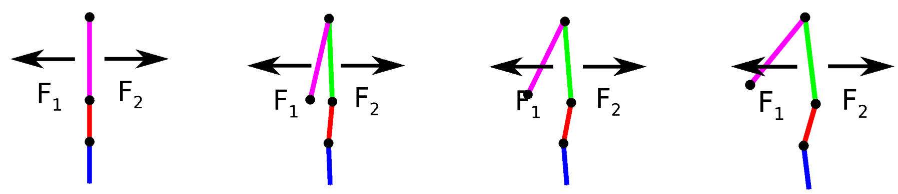



Figure 2: This figure illustrates different initial postures of a 4-link pendulum modelling a humanoid robot (70kg, 2m). The robot gets pushed with specific forces, i.e. $F_1$ and $F_2$ and has to move such that it maintains balance. At the end of the movement the robot should stabilize at the upright position.

### 3.1 Setting

We use a 4-link robot as a simplistic model of a humanoid (70kg, 2m) [2]. The 8-dimensional state $\mathbf{x}_t$ is composed of the ankle, the knee, the hip and the arm positions and their velocities. Table 1 in Appendix B shows the initial velocities (resulting from the force $F$ which always acts at the shoulder of the robot) and the valid joint angle range for the task. Additionally to the joint limits, the controls are limited to the intervals $[\pm 70, \pm 500, \pm 500, \pm 250]$Ns (ankle, knee, hip and arm). For more details we refer to Atkeson and Stephens [2].

We use a quadratic cost function given by

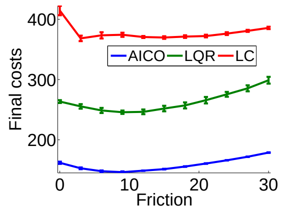
$$c_t(\mathbf{x}_t, \mathbf{u}_t) = (\mathbf{x}_t - \mathbf{x}^*)^T \hat{\mathbf{R}}_t (\mathbf{x}_t - \mathbf{x}^*) + \mathbf{u}_t^T \hat{\mathbf{H}}_t \mathbf{u}_t,$$

where the final state is denoted by $\mathbf{x}^* = \mathbf{0}$. The precision matrix $\hat{\mathbf{R}}_t$ determines how costly it is not to reach $\mathbf{x}^*$. The diagonal elements of $\hat{\mathbf{R}}_{1:T}$ are set to $2 \cdot 10^3$ for joint angles and to 0.2 for joint velocities. Controls are punished by $\hat{\mathbf{H}}_{1:T-1} = 0.05\mathbf{I}$.

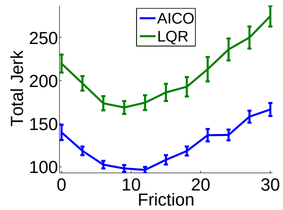
The movement trajectories are simulated with a time-step of 0.5ms. For the AICO algorithm we used a planning time step of $\Delta t = 5$ms and a horizon of $T = 500$, which results in a movement duration of 2.5s. We use a torque dependent noise model — the controls are multiplied by Gaussian noise $\epsilon$ with mean 1 and a standard deviation of 0.25, i.e. we use 25% torque dependent noise. The noise affects the system dynamics $\dot{\mathbf{x}} = f(\mathbf{x}_t, \mathbf{u}_t + \epsilon)$ while simulating a trajectory, where $\epsilon \sim \mathcal{N}(\epsilon | 0, 0.25^2 \cdot \mathrm{abs}(\mathbf{u}_t))$. The experiments are performed for multiple forces sampled from $F \sim \mathcal{U}[-15, +15]$Ns.

### 3.2 Complexity Measures

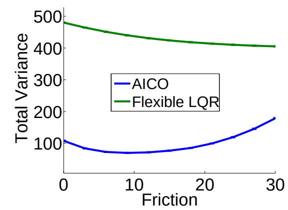
We evaluate the average values of the final costs $L(\tau)$ in Equation 1, the variance of the time-varying controller gains returned by the AICO approach in Equation 2 and the total jerk $J(\tau) = \Delta t \sum_{t=0}^{T} \dot{\mathbf{u}}_t^T \dot{\mathbf{u}}_t$ of the trajectory (proportional to the squared derivative of the torque). Different complexity measures would be possible. However, the chosen ones are plausible since the complexity of controlling the robot is reflected by how well the costs $L(\tau)$ are optimized. As the jerk of a movement tracks the derivative of the torques, this also seems to be a reliable complexity measure. Finally the variance of the time-varying controller gains quantifies the complexity of the control law in comparison to linear controllers (no variance).




(a) Friction vs. Final Costs




(b) Friction vs. Jerk




(c) Controller Variance

Figure 3: This figure shows the influence of the friction coefficient on controlling the 4-link model using the optimal control methods AICO and LQR. Illustrated are the mean and the standard error over 50 trajectories. As complexity measures of the morphology we used the final costs (a), the total jerk (b) and the controller variance (c). For the final costs we additionally compare to a simple linear feedback controller denoted by LC. For (c) we compare to a LQR controller with time varying costs in contrast to the standard LQR method which uses constant costs for all time steps.

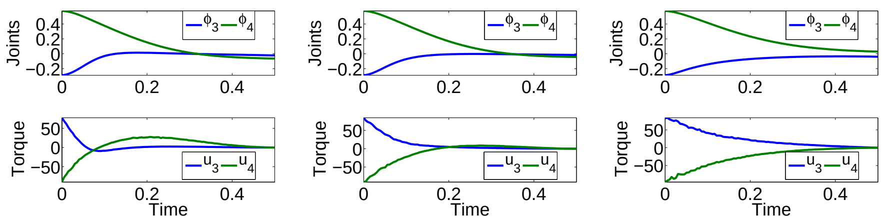
## 3.3 Friction induces Morphological Computation

In this experiment we evaluate the influence of the friction coefficient $\gamma$ on the quality of the optimal controller for the 4-link model. The friction coefficient directly modulates the acceleration of the joints, i.e. $\ddot{\phi}_\gamma = -\gamma\dot{\phi}$.

Figure 3(a) shows the resulting final costs for different friction coefficients. In order to illustrate the need for sophisticated control architectures, we compare the optimal control method AICO with an LQR controller and a simple, constant linear feedback controller. While AICO calculates the optimal control law for the non-linear system, the LQR controller linearizes the system at the upright position and subsequently calculates the closed form solution of the optimal controller for the resulting LQG system. The constant linear feedback controller does not adapt to the morphology of the robot. As we can see from Figure 3(a), we cannot determine the optimal morphology with the simple linear controller, while we can recognize a clear minimum for the AICO and the LQR controller.

AICO could find the most simple control law according to the jerk criteria in Figure 3(b) and was also able to find control laws with considerably decreased costs. In Figure 3 (c), we illustrated the variance of the time-varying controller gains. The complexity of the resulting controllers is reduced by changing the friction coefficient. The minimum lies in the same range as the minimum of the cost function. Thus, in this case a simpler controller resulted in better performance because the morphology (the friction) of the robot has simplified the control task. Still, we needed a more complex optimal control algorithm to discover this simpler control law if we compare to the LQR.

In Figure 4 we depict the joint and torque trajectories for the hip and the arm joint, where we applied the friction coefficients $\gamma = 0$, $\gamma = 12$ and $\gamma = 30$. With $\gamma = 0$ the trajectories overshoot as shown in Figure 4 (a), whereas in (c) more energy is needed for stabilizing the robot with $\gamma = 30$. The trajectories for the optimal friction coefficient $\gamma = 12$ (according to the jerk complexity measure) are shown in (b).




(a) $\gamma = 0$ (b) $\gamma = 12$ (c) $\gamma = 30$

Figure 4: The plots show the mean trajectories of the hip ($\phi_3$) and the arm ($\phi_4$) joints over 100 runs for the first 0.5s, where we used the AICO approach for the friction coefficients $\gamma = 0$, $\gamma = 12$ and $\gamma = 30$.

## 3.4 The Influence of the Link Lengths

To show the effects of morphological computation on a more complex variation of morphologies, we consider in this experiment the variation of the shank, the thigh, the trunk, and the arm length $\mathbf{l} = [l_1, l_2, l_3, l_4]$. Initially the link lengths are $\mathbf{l}_0 = [0.5, 0.5, 1, 1]$m and the weights of these links are specified by $\mathbf{m}_0 = [17.5, 17.5, 27.5, 7.5]$kg.

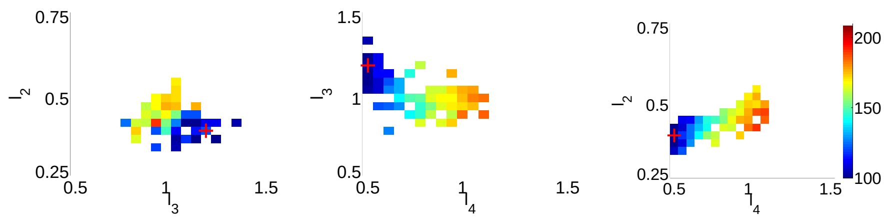
We consider variations of $\mathbf{l}$ of up to 50% from their initial configuration. Different link lengths of the 4-link robot result in different initial velocities [2], which influences the complexity of the motor control problem. For this reason, we consider constant inertias for a fair comparison. Thus, for each link-length $\mathbf{l}$ the masses of the links are given by $\mathbf{m} = \mathbf{m}_0 {\mathbf{l}_0}^2 / \mathbf{l}^2$. To match the physical properties of humans we assume equal lengths for the shank and the thigh link $l_1 = l_2$.

In addition to the link lengths of the thigh, the trunk and the arm we still optimize the friction coefficient, resulting into 4 parameters of our morphology. A naive search for optimal parameters like in previous experiment is in the multi-dimensional case intractable. Therefore we apply the non-linear stochastic optimizer Hansen et al. [4] as a local search method.

The stochastic optimizer locally explores the parameter space ($l_2$, $l_3$ and $l_4$) until convergence. For example, the resulting final cost values for different link lengths are illustrated in Figure 5 when using AICO as optimal control law. For this experiment we used a friction coefficient of $\gamma = 12$, where we averaged over multiple initial states and multiple forces. The optimizer converged to link lengths of $l_2 = 0.39$, $l_3 = 1.18$ and $l_4 = 0.5$ for the thigh, the torso and the arm.

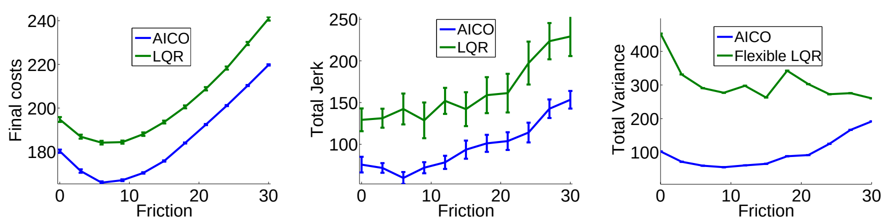
For multiple friction coefficients an evaluation of the presented complexity measures is shown in Figure 6. According to the final costs in Figure 6 (a) the best controller was found with a friction coefficient of $\gamma = 9$. For the jerk criteria in Figure 6 (b) and the variance as complexity measure shown in Figure 6 (c), the minimum of the complexity measures lies in the same range.

The morphologies found by the optimizer for the AICO and the LQR controller are sketched in Figure 7 for the friction coefficients $\gamma = 0$, $\gamma = 9$ and $\gamma = 30$. The initial configuration of the robot is shown in Figure 7 (a) and in Figure 7 (e). Interestingly with increasing values of the friction coefficients the link lengths change to compensate for the larger motor controls. This is shown in Figure 7 (b-d) for the AICO controller and in Figure 7 (f-h) for the LQR




(a) $l_2$ vs. $l_3$ (b) $l_3$ vs. $l_4$ (c) $l_2$ vs. $l_4$

Figure 5: This figure illustrates the explored link lengths for the thigh, the torso and the arm ($l_2$, $l_3$ and $l_4$) using AICO with the friction coefficient $\gamma = 12$. For this illustration we discretized the parameter space to visualize the final cost values, which are denoted by the color of the dots. Darker dots correspond to lower cost values as specified by the colorbar on the right. The optimal values of the link lengths are denoted by the large crosses ($l_2 = 0.39$, $l_3 = 1.18$ and $l_4 = 0.5$).




(a) Friction vs. Final Costs (b) Friction vs. Jerk (c) Controller Variance

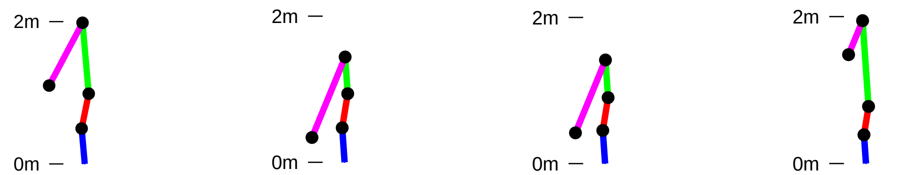
Figure 6: This figure shows the results for optimizing the link lengths and the friction coefficient of the 4-link model. Like before we used the final costs (a), the total jerk (b) and the variance of the time-varying feedback controller (c) as complexity measure.

controller. AICO could again find significantly different morphologies as the LQR controller. As is illustrated in the captions of Figure 7, the morphologies found by AICO could produce lower costs as the morphologies found by LQR.

## 3.5 Robot Model with Linear Springs

Inspired by the theoretical analysis of [5], we also evaluate the computational power of linear springs in our model. The springs are mounted on the 4-link model as illustrated in Figure 8. These springs support the control of the 4-link robot as they push the robot back into its upright position. Thus, the torques applied to the joints are given by the sum of the controls provided by the feedback controller and the spring forces $\mathbf{u}_t^* = \mathbf{u}_t + \mathrm{diag}(\mathbf{k})\mathbf{\Delta l_s}$. The vector $\mathbf{\Delta l_s}$ denotes the displacements of the four springs and $\mathbf{k}$ is a vector containing the 4 spring constants. Note that due to the calculation of the displacements, the springs act non-linearly on the joint torques.

The adjustable parameters of the morphology include the friction coefficient and the four spring constants $\mathbf{k} = [k_1, k_2, k_3, k_4]$ for the ankle, the knee, the hip, and the arm joint. Initially the spring constants are set to $\mathbf{k}_0 = [1, 10^3, 10^3, 1]$.

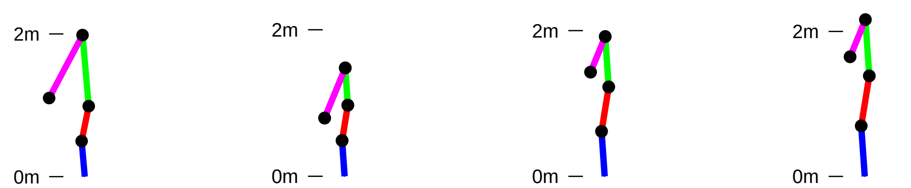



(a) initial configuration

(b) AICO with $\gamma = 0$, $L(\tau) = 180.2$

(c) AICO with $\gamma = 9$, $L(\tau) = 166.9$

(d) AICO with $\gamma = 30$, $L(\tau) = 219.7$




(e) initial configuration

(f) LQR with $\gamma = 0$, $L(\tau) = 194.7$

(g) LQR with $\gamma = 9$, $L(\tau) = 184.4$

(h) LQR with $\gamma = 30$, $L(\tau) = 240.9$

Figure 7: This figure shows the learned link lengths for multiple friction coefficients $\gamma = 0$, $\gamma = 9$ and $\gamma = 30$ for the AICO (b-d) and the LQR (f-h) controller. The link lengths are optimized with a stochastic optimizer and could vary up to 50% from the initial configuration shown in (a) and (e). With increasing friction coefficients the link lengths change to compensate for the larger control costs. The final costs $L(\tau)$ for these morphologies are given in the captions.

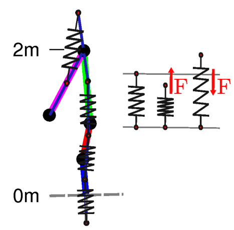
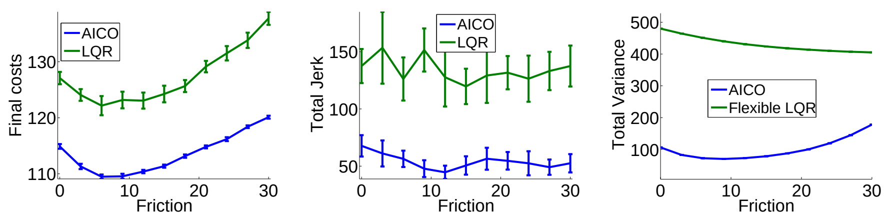



Figure 8: This figure illustrates the 4-link robot model endowed with linear springs which are mounted at the midpoints of the joints. These springs support the control of the 4-link robot as they push the robot back into its upright position. Due to the calculation of the spring length displacements, the springs act non-linearly on the joint torques.




(a) Friction vs. Final Costs (b) Friction vs. Jerk (c) Controller Variance

Figure 9: This figure shows the results for optimizing the spring constants and the friction coefficient of the 4-link model. We used the final costs (a), the total jerk (b) and the variance of the time-varying feedback controller (c) as complexity measure.

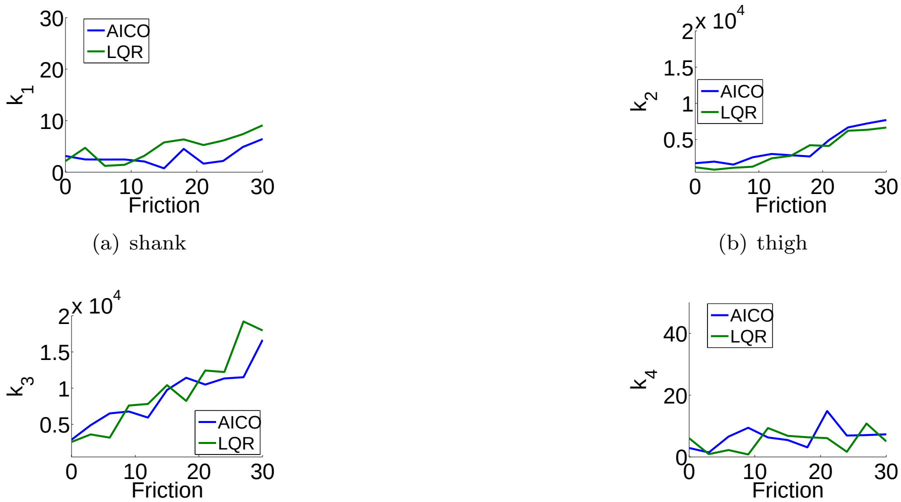
For the optimization a lower bound of $0.1\mathbf{k}_0$ and an upper bound of $100\mathbf{k}_0$ are used.

As in the previous experiment, for different friction coefficients $\gamma$, we optimize the spring constants using the stochastic search method Hansen et al. [4]. The results for the final costs, the total jerk and the variance of the feedback controller as complexity measures are shown in Figure 9. The benefit of the supporting springs is best illustrated by the final costs in Figure 9 (a), which are drastically reduced compared to the final costs using the 4-link model without springs shown in Figure 3 (a).

The optimal spring constants strongly depend on the used friction coefficient. This is illustrated in Figure 9, where we compare the learned spring constants using AICO and LQR as control method. With an increasing friction also the spring constants increase to compensate for larger control costs.

# 4 Conclusion

In this paper we have shown that optimal control and morphological computation are two complementary approaches which can benefit from each other. The search for an optimal morphology is simplified if we can calculate an optimal controller for a given morphology. This calculation can be done by new




(a) shank (b) thigh (c) trunk (d) arm

Figure 10: This figure shows the optimal spring constants for different friction coefficients using AICO and LQR.

approaches from probabilistic inference for motor control, i.e. the approximate inference control algorithm [16]. By the use of optimal control methods, we have shown that for dynamic non-linear balancing tasks, an appropriate setting of the friction, the link lengths or the spring constants of the incorporated springs of the compliant robot can simplify the control problem.

We have demonstrated that there are optimal values for the physical properties for a given control task, which can only be found by the use of optimal control laws. With naive control architectures such as the evaluated constant linear feedback controller which does not adapt to the morphology, different, sub-optimal morphologies would be chosen.

In the future, we plan to investigate more complex and more non-linear tasks. In this case the benefit of AICO in comparison to LQR controllers should be even more prominent. In the end we are planning to simultaneously evolve walking controllers based on AICO and the morphology of bipedal robots like a model of a planar walker.

# Acknowledgment

This paper was written under partial support by the European Union project FP7-248311 (AMARSI) and project FP7-506778 (PASCAL2). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

# References


[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the twenty-first International Conference on Machine learning*, (ICML 2004), pages 1–8, New York, NY, USA, 2004. ACM.

[2] C.G. Atkeson and B. Stephens. Multiple Balance Strategies from one Optimization Criterion. In *Proceedings of the 7th International Conference on Humanoid Robots*, pages 57–64, Pittsburgh, USA, 2007.

[3] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative Entropy Inverse Reinforcement Learning. *Journal of Machine Learning Research - Proceedings Track*, 15:182–189, 2011.

[4] N. Hansen, S.D. Muller, and P. Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[5] Helmut Hauser, Auke Jan Ijspeert, Rudolf M. Füchslin, Rolf Pfeifer, and Wolfgang Maass. Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics*, 105(5-6):355–370, 2011.

[6] Kappen H.J., Gmez V., and Opper M. Optimal control as a graphical model inference problem. *Machine Learning*, pages 1–24, 2012. ISSN 0885-6125.

[7] Fumiya Iida and Rolf Pfeifer. Sensing through body dynamics. *Robotics and Autonomous Systems*, 54(8):631–640, 2006.

[8] H. J. Kappen. An Introduction to Stochastic Control Theory, Path Integrals and Reinforcement Learning. In *Cooperative Behavior in Neural Systems*, volume 887 of *American Institute of Physics Conference Series*, pages 149–181, 2007.

[9] D. Nguyen-Tuong, J. Peters, M. Seeger, and B. Schölkopf. Learning Inverse Dynamics: A Comparison. In *16th European Symposium on Artificial Neural Networks*, (ESANN 2008), pages 13–18, Bruges, Belgium, 2008.

[10] D Nguyen-Tuong, M Seeger, and J Peters. Local Gaussian Process Regression for Real Time Online Model Learning and Control. In *Proceedings of 22nd Annual Conference on Neural Information Processing Systems*, (NIPS 2008), pages 1193–1200, Vancouver, BC, Canada, 2008.

[11] Rolf Pfeifer and Josh Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. The MIT Press, October 2006. ISBN 0262162393.

[12] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093, 2007. doi: 10.1126/science.1145803.

[13] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. An Approximate Inference Approach to Temporal Optimization in Optimal Control. In *Proceedings of 24nd Annual Conference on Neural Information Processing Systems*, (NIPS 2010), pages 2011–2019, Vancouver, BC, Canada, 2010.

[14] Russ Tedrake, Teresa Weirui Zhang, and H. Sebastian Seung. Learning to Walk in 20 Minutes. In *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems*, Yale University, New Haven, CT, 2005.

[15] E. Todorov and Weiwei Li. A generalized Iterative LQG Method for Locally-Optimal Feedback Control of Constrained Nonlinear Stochastic Systems. In *Proceedings of the 24th American Control Conference*, volume 1 of *(ACC 2005)*, pages 300 – 306, Portland, Oregon, USA, 2005.

[16] Marc Toussaint. Robot Trajectory Optimization using Approximate Inference. In *Proceedings of the 26th International Conference on Machine Learning*, (ICML 2009), pages 1049–1056, Montreal, Canada, 2009.

[17] Marc Toussaint. Lecture Notes: Gaussian Identities. Technical report, Freie Universität Berlin, 2011. URL `http://userpage.fu-berlin.de/mtoussai/ notes/gaussians.pdf`.

[18] S. Vijayakumar, A. D'Souza, and S. Schaal. Incremental Online Learning in High Dimensions. *Neural Computation*, 17(12):2602–2634, dec 2005.

[19] R.J. Wood. Design, fabrication, and analysis of a 3dof, 3cm flapping-wing MAV. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (IROS 2007), pages 1576–1581, Oct-Nov 2 2007.

[20] Marc Ziegler, Fumiya Iida, and Rolf Pfeifer. "Cheap" Underwater Locomotion: Roles of Morphological Properties and Behavioral Diversity. In *9th International Conference on Climbing and Walking Robots*, (CLAWAR 2006), Brussels, Belgium, September 2006.

# A Extension of the Approximate Inference Control Method

The original formulation of the Approximate Inference Control (AICO) method [16] does not consider a linear term for the control costs. However, this is needed to encode torque limits, which are important for our dynamic balancing experiments, and hence, we needed to extend AICO.

The introduction of a linear term for the control costs yields not only in a modified cost function but also results in different update equations for the messages and finally in different equations of the optimal feedback controller. For completeness we will first recap the main steps to derive the AICO method and will then discuss the modifications to implement control constraints.

## Approximate Inference Control without Torque Limits

For motor planning we consider the stochastic process:

$$P(\mathbf{x}_{1:T}, \mathbf{u}_{1:T-1}, \mathbf{z}_{1:T}) = P(\mathbf{x}_1) \prod_{t=1}^{T-1} P(\mathbf{u}_t|\mathbf{x}_t) \prod_{t=1}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \prod_{t=1}^{T-1} P(z_t|\mathbf{x}_t, \mathbf{u}_t).$$

where $P(\mathbf{u}_t|\mathbf{x}_t)$ denotes the state dependent prior for the controls, the distribution $P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ the state transition model and $P(\mathbf{x}_1)$ the initial state distribution. Here, we assume that the prior of the controls is independent of the states and thus we will simply use $P(\mathbf{u}_t|\mathbf{x}_t) = P(\mathbf{u}_t)$ for the rest of the appendix. The time horizon is fixed to $T$ time-steps. The binary task variable $z_t$ denotes a reward event, its probability is defined by $P(z_t = 1|\mathbf{x}_t, \mathbf{u}_t) \propto \exp(-c_t(\mathbf{x}_t, \mathbf{u}_t))$, where $c_t(\mathbf{x}_t, \mathbf{u}_t)$ is the intermediate cost function[1] for time-step $t$. It expresses a performance criteria (like avoiding a collision, or reaching a goal).

We want to compute the posterior $P(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}|z_{1:T} = 1)$ over trajectories, conditioned on observing a reward ($z_t = 1$) at each time-step $t$. This posterior can be computed by using message passing in the given graphical model of Figure 1. To simplify the computations we integrate out the controls:

$$P(\mathbf{x}_{t+1}|\mathbf{x}_t) = \int_{\mathbf{u}_t} P(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) P(\mathbf{u}_t|\mathbf{x}_t) d\mathbf{u}_t, \tag{3}$$

The marginal belief $b_t(\mathbf{x}_t)$ of a state at time $t$ is given by:

$$b_t(\mathbf{x}_t) = \alpha_t(\mathbf{x}_t)\beta_t(\mathbf{x}_t)\phi_t(\mathbf{x}_t), \tag{4}$$

where $\alpha_t(\mathbf{x}_t)$ is the forward message, $\beta_t(\mathbf{x}_t)$ is the backward message $\phi_t(\mathbf{x}_t)$ is the current task message. The messages are given by:

$$\begin{aligned}
\alpha_t(\mathbf{x}_t) &= \int_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1})\alpha_{t-1}(\mathbf{x}_{t-1})\phi_{t-1}(\mathbf{x}_{t-1})d\mathbf{x}_{t-1}, && (5)\\
\beta_t(\mathbf{x}_t) &= \int_{\mathbf{x}_{t+1}} P(\mathbf{x}_{t+1}|\mathbf{x}_t)\beta_{t+1}(\mathbf{x}_{t+1})\phi_{t+1}(\mathbf{x}_{t+1})d\mathbf{x}_{t+1}, && (6)\\
\phi_t(\mathbf{x}_t) &= P(z_t|\mathbf{x}_t). && (7)
\end{aligned}$$

We consider discrete-time, non-linear stochastic systems with zero mean Gaussian noise

$$P(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}|f_{\mathrm{Dyn}}(\mathbf{x}_t, \mathbf{u}_t), \mathbf{Q}_t).$$

The non-linear stochastic system $f_{\mathrm{Dyn}}$ is approximated by a Linear dynamics, Quadratic costs and Gaussian noise system (LQG) by Taylor expansion [16, 15]:

$$P(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}_t\mathbf{x}_t + \mathbf{a}_t + \mathbf{B}_t\mathbf{u}_t, \mathbf{Q}_t). \tag{8}$$

Thus, the system is linearized along a given trajectory $\langle \hat{\mathbf{x}}_{1:T}, \hat{\mathbf{u}}_{1:T-1} \rangle$ at every point in time. We will use $f_t$ as shorthand for $f_{\mathrm{Dyn}}(\mathbf{x}_t, \mathbf{u}_t)$. Then, the state transition matrix $\mathbf{A}_t$ is given by $\mathbf{A}_t = (\mathbf{I} + \frac{\delta f_t}{\delta \mathbf{x}_t}\Delta t)$, the control matrix $\mathbf{B}_t$ is $\mathbf{B}_t = \frac{\delta f_t}{\delta \mathbf{u}_t}\Delta t$ and the linear term reads $\mathbf{a}_t = (f_t - \frac{\delta f_t}{\delta \mathbf{x}_t}\mathbf{x}_t - \frac{\delta f_t}{\delta \mathbf{u}_t}\mathbf{u}_t)\Delta t$.

In the original formulation of the AICO algorithm the cost function $c_t$ is approximated as:

$$c_t(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^T\mathbf{R}_t\mathbf{x}_t - 2\mathbf{r}_t^T\mathbf{x}_t + \mathbf{u}_t^T\mathbf{H}_t\mathbf{u}_t.$$

Note that there is no linear term for the control costs as we only punish quadratic controls. We can now write $P(z_t = 1|\mathbf{x}_t, \mathbf{u}_t) = P(z_t = 1|\mathbf{x}_t)P(\mathbf{u}_t)$ as

$$\begin{aligned}
P(z_t = 1|\mathbf{x}_t) &\propto \mathcal{N}[\mathbf{x}_t|\mathbf{r}_t, \mathbf{R}_t], && (9)\\
P(\mathbf{u}_t) &= \mathcal{N}[\mathbf{u}_t|\mathbf{0}, \mathbf{H}_t], && (10)
\end{aligned}$$

[1] In this paper the immediate cost function is composed of the intrinsic costs and the constraint costs, i.e. $c_t(\mathbf{x}_t, \mathbf{u}_t) + c_p(\mathbf{x}_t, \mathbf{u}_t)$

where the distributions in Equation 9 and 10 are given in canonical form. The canonical form of a Gaussian is used because numerical operations such as products or integrals are easier to calculate in this notation. The canonical form is indicated by the *square* bracket notation and given by

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

A Gaussian in normal form can always be transformed into the canonical form by $\mathcal{N}(\mathbf{x}|\mathbf{a},\mathbf{A}) = \mathcal{N}[\mathbf{x}|\mathbf{A}^{-1}\mathbf{a},\mathbf{A}^{-1}]$. For more details we refer to the Gaussian Identities in Toussaint [17].

We can see in Equation 10 that our prior for applying the control $\mathbf{u}_t$ is given by the control costs, i.e. $\mathcal{N}[\mathbf{u}_t|\mathbf{0},\mathbf{H}_t]$. By integrating out the controls from our system dynamics we get the following state transition probabilities

$$\begin{aligned} P(\mathbf{x}_{t+1}|\mathbf{x}_t) &= \int_{\mathbf{u}_t}\mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}_t\mathbf{x}_t + \mathbf{a}_t + \mathbf{B}_t\mathbf{u}_t,\mathbf{Q}_t)\mathcal{N}[\mathbf{u}_t|\mathbf{0},\mathbf{H}_t]d\mathbf{u}_t, &(11)\\ &= \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}_t\mathbf{x}_t + \mathbf{a}_t, \mathbf{Q}_t + \mathbf{B}_t\mathbf{H}_t^{-1}\mathbf{B}_t), &(12)\end{aligned}$$

where the integral was solved using a reformulation of the *propagation* rule in Toussaint [17].

As we can see, all distributions in the approximated LQG system in Equation 12 are Gaussian, and thus, also all messages are Gaussians and can be calculated analytically. The resulting messages are given in Toussaint [16].

## Approximate Inference Control with Torque Limits

In order to implement torque and joint limits we introduce an additional cost function $c_p$ which punishes the violation of the given constraints. The function $c_p$ is just added to the current immediate costs. We use separate cost terms for control constraints $c_t^u$ and joint constraints $c_t^x$, i.e. $c_p(\mathbf{x}_t,\mathbf{u}_t) = c_t^x(\mathbf{x}_t) + c_t^u(\mathbf{u}_t)$. Here, we will only discuss how to implement the function $c_t^u(\mathbf{u}_t)$ for the torque constraints, joint constraints are implemented similarly.

The cost function $c_t^u$ is quadratic in $\mathbf{u}$ and punishes leaving the valid control limits of $\mathbf{u}$. In order to implement the upper bound $\mathbf{u}_{\max}$ for the torques, we use the following cost function

$$\begin{aligned} c_t^u(\mathbf{u}_t) &= \mathbf{u}_t^T\mathbf{H}_t\mathbf{u}_t + (\mathbf{u}_t - \mathbf{u}_{\max})^T\mathbf{H}_t^U(\mathbf{u}_t - \mathbf{u}_{\max}),\\ &= \mathbf{u}_t^T\mathbf{H}_t\mathbf{u}_t + \mathbf{u}_t^T\mathbf{H}_t^U\mathbf{u}_t - 2\mathbf{u}_{\max}^T\mathbf{H}_t^U\mathbf{u}_t + \mathbf{u}_{\max}^T\mathbf{H}_t^U\mathbf{u}_{\max},\\ &= \mathbf{u}_t^T\mathbf{H}_t\mathbf{u}_t + \mathbf{u}_t^T\mathbf{H}_t^U\mathbf{u}_t - 2\mathbf{u}_{\max}^T\mathbf{H}_t^U\mathbf{u}_t + \text{const},\end{aligned}$$

where matrix $\mathbf{H}_t$ denotes the quadratic control costs. The constrained costs are only imposed for the control variable $u_i$ if the torque value exceeds the upper bound $u_{\max,i}$. In order to do so $\mathbf{H}_t^U$ is a diagonal matrix where the $i$th diagonal entry is zero if $u_i \leq u_{\max,i}$ and non-zero otherwise. The lower bound $\mathbf{u}_{\min}$ is implemented likewise using an individual diagonal matrix $\mathbf{H}_t^L$.

We can again implement $c_t^u(\mathbf{u}_t)$ as prior distribution $P(\mathbf{u}_t)$ for the controls.

$$P(\mathbf{u}_t) \propto \mathcal{N}[\mathbf{u}_t|\mathbf{h}_t,\mathbf{H}_t], \tag{13}$$

where $\mathbf{h}_t = \mathbf{u}_{\max}^T\mathbf{H}_t^U + \mathbf{u}_{\min}^T\mathbf{H}_t^L$ and the precision $\hat{\mathbf{H}}_t = \mathbf{H}_t + \mathbf{H}_t^U + \mathbf{H}_t^L$. *As we can see, the linear term $\mathbf{h}_t$ of the prior distribution $P(\mathbf{u}_t)$ is now non-zero. This yields different message equations.*

Joint-limits can be imposed similarly by using additional terms costs for $c_t^x(\mathbf{x}_t)$. However, for joint limits the update equations stay the same because $P(z_t = 1|\mathbf{x}_t)$ has already a non-zero mean denoted by $\mathbf{r}_t$ in Equation 9.

To derive the messages we will first integrate out the controls to get the state transition probabilities:

$$\begin{aligned} P(\mathbf{x}_{t+1}|\mathbf{x}_t) &= \int_{\mathbf{u}_t} \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}_t\mathbf{x}_t + \mathbf{a}_t + \mathbf{B}_t\mathbf{u}_t, \mathbf{Q}_t)\mathcal{N}[\mathbf{u}_t|\mathbf{h}_t, \hat{\mathbf{H}}_t]d\mathbf{u}_t, \\ &= \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}_t\mathbf{x}_t + \mathbf{a}_t + \mathbf{B}_t\hat{\mathbf{H}}_t^{-1}\mathbf{h}_t, \mathbf{Q}_t + \mathbf{B}_t\hat{\mathbf{H}}_t^{-1}\mathbf{B}_t^T). \end{aligned} \tag{14}$$

Note that, since the cost function $c_t^u(\mathbf{u}_t)$ contains a non-zero linear term $\mathbf{h}_t$, we get a new linear term $\hat{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{B}_t\mathbf{H}_t^{-1}\mathbf{h}_t$ in the transition dynamics. The forward and the backward messages are the same like in Toussaint [16] except that $\mathbf{a}_t$ is replaced by $\hat{\mathbf{a}}_t$ and $\mathbf{H}_t$ by $\hat{\mathbf{H}}_t$.

Like in Toussaint [16] we use the canonical representations for the forward and the backward message:

$$\begin{aligned} \alpha_t(\mathbf{x}_t) &= \mathcal{N}[\mathbf{x}_t|\mathbf{s}_t, \mathbf{S}_t], \\ \beta_t(\mathbf{x}_t) &= \mathcal{N}[\mathbf{x}_t|\mathbf{v}_t, \mathbf{V}_t], \\ \phi_t(\mathbf{x}_t) &= P(z_t|\mathbf{x}_t) = \mathcal{N}[\mathbf{x}_t|\mathbf{r}_t, \mathbf{R}_t]. \end{aligned}$$

The messages are represented by Gaussians in canonical form, for which mathematical operations like products are simply performed by adding the linear terms and the precisions. The mean of the belief is given by $b_t(\mathbf{x}_t) = (\mathbf{S}_t + \mathbf{V}_t + \mathbf{R}_t)^{-1}(\mathbf{s}_t + \mathbf{v}_t + \mathbf{r}_t)$ (multiplying three canonical messages and a subsequent transformation to normal form). Furthermore we use the shorthand $\bar{\mathbf{Q}}_t = \mathbf{Q}_t + \mathbf{B}_t\hat{\mathbf{H}}_t^{-1}\mathbf{B}_t^T$ for the covariance in Equation 14.

The messages are computed by inserting the state transition probabilities given in Equation 14 in the message passing Equations 5 and 6. Subsequently the integrals are solved using the *propagation* rule in Toussaint [17]. The final equations in canonical form are:

$$\mathbf{S}_t = (\mathbf{A}_{t-1}^{-T} - \mathbf{K}_s)\mathbf{S}_{t-1}\mathbf{A}_{t-1}^{-1}, \tag{15}$$

$$\mathbf{s}_t = (\mathbf{A}_{t-1}^{-T} - \mathbf{K}_s)(\bar{\mathbf{s}}_{t-1} + \mathbf{S}_{t-1}\mathbf{A}_{t-1}^{-1}(\hat{\mathbf{a}}_{t-1} + \mathbf{B}_{t-1}\hat{\mathbf{H}}_{t-1}^{-1}\mathbf{h}_{t-1})), \tag{16}$$

$$\mathbf{K}_s = \mathbf{A}_{t-1}^{-T}\mathbf{S}_{t-1}(\mathbf{S}_{t-1} + \mathbf{A}_{t-1}^{-T}\bar{\mathbf{Q}}_{t-1}^{-1}\mathbf{A}_{t-1}^{-1})^{-1}. \tag{17}$$

And for the backward messages:

$$\mathbf{V}_t = (A_t^T - \mathbf{K}_v)\bar{\mathbf{V}}_{t+1}\mathbf{A}_t, \tag{18}$$

$$\mathbf{v}_t = (\mathbf{A}_t^T - \mathbf{K}_v)(\bar{\mathbf{v}}_{t+1} - \bar{\mathbf{V}}_{t+1}(\hat{\mathbf{a}}_t + \mathbf{B}_t\hat{\mathbf{H}}_t^{-1}\mathbf{h}_t)), \tag{19}$$

$$\mathbf{K}_v = \mathbf{A}_t^T\bar{\mathbf{V}}_{t+1}(\bar{\mathbf{V}}_{t+1} + \bar{\mathbf{Q}}_t^{-1})^{-1}. \tag{20}$$

To obtain the same mathematical form as in Toussaint [16] one needs to apply the Woodbury identity and reformulate the equations. In contrast to the update message in normal form [16], direct inversions of $\bar{\mathbf{S}}_{t-1}$ and $\bar{\mathbf{V}}_{t+1}$ are not necessary in the canonical form and therefore, the iterative updates are numerically more stable.

Finally, in order to compute the optimal feedback controller we calculate the joint state-control posterior

$$\begin{aligned}P(\mathbf{u}_t, \mathbf{x}_t) &= P(\mathbf{u}_t, \mathbf{x}_t | z_t = 1)\\ &= \int_{\mathbf{x}_{t+1}} \alpha_t(\mathbf{x}_t)\phi_t(\mathbf{x}_t)P(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)P(\mathbf{u}_t)\beta_{t+1}(\mathbf{x}_{t+1})\phi_{t+1}(\mathbf{x}_{t+1})d\mathbf{x_{t+1}},\\ &= P(\mathbf{x}_t)P(\mathbf{u}_t)\int_{\mathbf{x}_{t+1}} P(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)\mathcal{N}[\mathbf{x}_{t+1}|\bar{\mathbf{v}}_{t+1}, \bar{\mathbf{V}}_{t+1}]d\mathbf{x_{t+1}}.\end{aligned}$$

The conditional distribution is given by $P(\mathbf{u}_t|\mathbf{x}_t) = P(\mathbf{u}_t, \mathbf{x}_t)/P(\mathbf{x}_t)$, and the solution is

$$P(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{u}_t|\mathbf{M}_t^{-1}(\mathbf{B}_t^T\mathbf{V}_*(\bar{\mathbf{V}}_{t+1}^{-1}\bar{\mathbf{v}}_{t+1} - \mathbf{A}_t\mathbf{x}_t - \hat{\mathbf{a}}_t) + \mathbf{h}_t), \mathbf{M}_t^{-1}),$$

where $\mathbf{V}_* = (\mathbf{Q} + \bar{\mathbf{V}}_{t+1}^{-1})^{-1}$ and $\mathbf{M}_t = \mathbf{B}_t^T\mathbf{V}_*\mathbf{B}_t + \hat{\mathbf{H}}_t$. After a reformulation we can obtain an optimal feedback controller of the form $\mathbf{u}_t = \mathbf{o}_t + \mathbf{O}_t\mathbf{x}_t$ with

$$\mathbf{o}_t = \mathbf{M}_t^{-1}(\mathbf{B}_t^T\mathbf{V}_*\bar{\mathbf{V}}_{t+1}^{-1}\bar{\mathbf{v}}_{t+1} - \mathbf{B}_t^T\mathbf{V}_*\mathbf{a}_t + \mathbf{h}_t), \tag{21}$$

$$\mathbf{O}_t = -\mathbf{M}_t^{-1}\mathbf{B}_t^T\mathbf{V}_*\mathbf{A}_t. \tag{22}$$

Similar to Toussaint [16], we use an iterative message passing approach where we approximate the non-linear system by an Linear dynamics, Quadratic costs and Gaussian noise system (LQG) at the new mode of the trajectory. In Toussaint [16], this is done by using a learning rate on the current modes of the belief. However, in difference to Toussaint [16], we also need an estimate of the optimal action $\mathbf{u}_t$ in order to impose the control constraints. Using a learning rate on the control action $\mathbf{u}_t$ turned out to be very ineffective because feedback is extenuated. For this reason we will use a learning rate on the feedback controller.

The complete message passing algorithm considering state and control constraints is listed in Algorithm 1. This is a straightforward implementation of Gaussian message passing in linearized systems, similar to an extended Kalman smoother.

In Toussaint [16] or Rawlik et al. [13] more time efficient methods are presented, where for each time-step the belief is updated until convergence in contrast to updating all messages and iterating until the intrinsic costs $L(\tau)$ converge. The computational benefits of such an approach still needs to be evaluated for our messages.

# B 4-Link Robot Model Specifications

For the 4-link model the 8-dimensional state $\mathbf{x} = [\phi_1, \dot{\phi}_1, \phi_2, \dot{\phi}_2, \phi_3, \dot{\phi}_3, \phi_4, \dot{\phi}_4]$ is composed of the ankle, the knee, the hip and the arm positions and their velocities. In our experiments the velocities are instantaneously affected by the applied force $F$ [2]. These initial velocities and the valid joint angle range are shown in Table 1.

In addition we use multiple initial joint angles. The ankle angle is given by $\phi_1 = -\alpha$, the knee angle by $\phi_2 = -\alpha\mathrm{f}$, the hip angle by $\phi_3 = \alpha\mathrm{f}$ and the arm joint angle is $\phi_4 = 2\alpha\mathrm{f}$, where $\alpha$ is sampled from $\alpha \sim \mathcal{U}[0, 7]\pi/180$ and the linear factor is uniform distributed with $\mathrm{f} \sim \mathcal{U}[-3.3, -2.3]$. Some example initial states are illustrated in Figure 2.

**Algorithm 1:** Approximate Inference Control for Constrained Systems

**Data**: initial trajectory $\hat{\mathbf{x}}_{1:T}$, learning rate $\eta$
**Result**: $\mathbf{x}_{1:T}$ and $\mathbf{u}_{1:T-1}$
initialize $\mathbf{S}_1 = 1e10 \cdot \mathbf{I}$, $\mathbf{s}_1 = \mathbf{S}_1\mathbf{x}_1$, $k = 0$, $\hat{\mathbf{o}}_{1:T-1} = \mathbf{0}$, $\hat{\mathbf{O}}_{1:T-1} = 0 \cdot \mathbf{I}$ ;
**while** $L(\tau)$ *not converged* **do**
- **for** $t \leftarrow 1$ **to** $T$ **do**
  - Linearize Model: $\mathbf{A}_t, \mathbf{a}_t, \mathbf{B}_t$ using Equation 8
  - Compute Costs: $\hat{\mathbf{H}}_t, \mathbf{h}_t, \mathbf{R}_t, \mathbf{r}_t$ using Equation 9, 13
- **for** $t \leftarrow 1$ **to** $T$ **do**
  - Forward Messages: $\alpha_t(\mathbf{x}_t)$ using Equation 15 - 17
- **for** $t \leftarrow T-1$ **to** $1$ **do**
  - Backward Messages: $\beta_t(\mathbf{x}_t)$ using Equation 18 - 20
- **for** $t \leftarrow 1$ **to** $T-1$ **do**
  - Feedback Controller: $\mathbf{o}_t, \mathbf{O}_t$ using Equation 21, 22
  - **if** $k == 0$ **then**
    - $\mathbf{u}_t = \mathbf{o}_t + \mathbf{O}_t\mathbf{x}_t$
  - **else**
    - $\hat{\mathbf{o}}_t = (1-\eta)\hat{\mathbf{o}}_t + \eta\mathbf{o}_t$
    - $\hat{\mathbf{O}}_t = (1-\eta)\hat{\mathbf{O}}_t + \eta\mathbf{O}_t$
    - $\mathbf{u}_t = \hat{\mathbf{o}}_t + \hat{\mathbf{O}}_t\mathbf{x}_t$
  - $\mathbf{x}_{t+1} = \mathbf{A}_t\mathbf{x}_t + \mathbf{a}_t + \mathbf{B}_t\mathbf{u}_t$
- $k = k + 1$

| **Joint** | **Initial velocities** | **Lower Bound** | **Upper Bound** |
|---|---|---|---|
| ankle | $+1.2 \cdot 10^{-2}F$ | $-0.8$ | 0.8 |
| knee | $-7.4 \cdot 10^{-2}F$ | $-0.05$ | 2.5 |
| hip | $+5.1 \cdot 10^{-2}F$ | $-2.0$ | 0.1 |
| arm | $-4.2 \cdot 10^{-2}F$ | $-0.6$ | 3.0 |

Table 1: Joint angle configurations where a robot gets pushed by a force $F$.