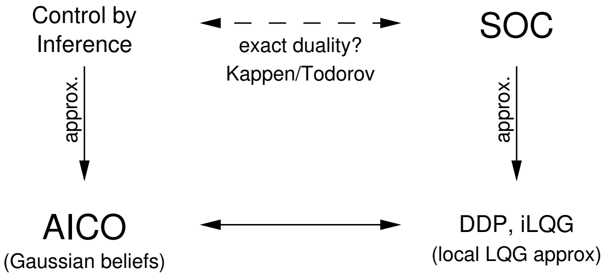
Approximate Inference Control

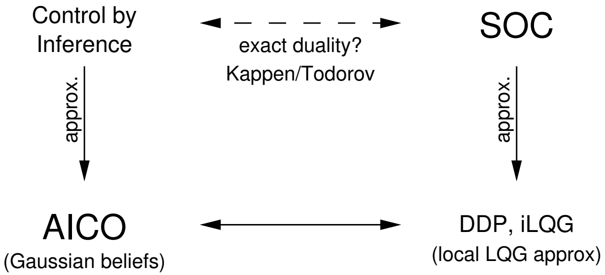
Marc Toussaint

Machine Learning & Robotics group – TU Berlin
mtoussai@cs.tu-berlin.de

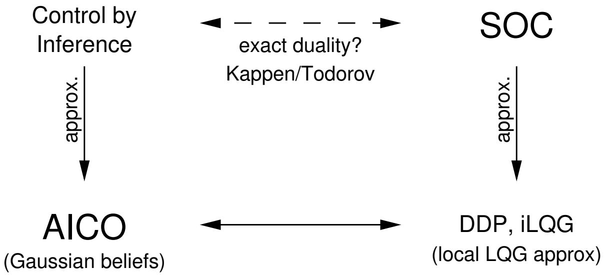
NIPS workshop "Probabilistic Approaches for Control and Robotics"

December 11, 2009


Control by
Inference
exact duality?
Kappen/Todorov
SOC
approx.
approx.
AICO
(Gaussian beliefs)
DDP, iLQG
(local LQG approx)



- good approximations are actually the real challenge
  $\rightarrow$ draw on probabilistic inference work for new approximations



- good approximations are actually the real challenge
  $\rightarrow$ draw on probabilistic inference work for new approximations

- this talk: Approximate Inference Control (AICO)
  (perhaps simplest version of "control by inference")

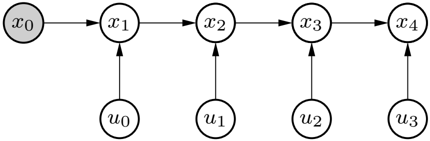
Outline

- Graphical model for control by inference
- Approximate inference
- Examples in robotics

Stochastic Optimal Control

- discrete time stochastic controlled process:

$$x_{t+1} = f_t(x_t, u_t) + \xi \;, \qquad \xi \sim \mathcal{N}(0, Q_t)$$



x_t	state at time t
u_t	control signal at time t
f	system dynamics
ξ	Gaussian noise

Stochastic Optimal Control

- classical notion of costs:

$$C(x_{0:T}, u_{0:T}) = \sum_{t=0}^{T} c_t(x_t, u_t)$$

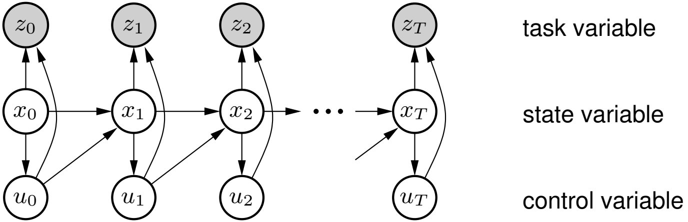
- problem: find a control policy $\pi_t^* : x_t \mapsto u_t$ that minimizes expected cost

$$\mathrm{E}_{x_{0:T}, u_{0:T}; \pi}\{C(x_{0:T}, u_{0:T})\}$$

- Bellman view:
  – optimal value function $J_t(x) = \min_{u_{t:T}} \mathrm{E}_{x_{t:T} \mid u_{t:T}, x_t = x}\{\sum_{k=t}^{T} c_k(x_k, u_k)\}$
  – Bellman equation $J_t(x) = \min_u \Big[c_t(x, u) + \int_{x'} P(x' \mid u, x)\ J_{t+1}(x')\Big]$

Inference control model

- introduce a binary "task" variable z_t to represent costs



$$
\begin{aligned}
&P(x_0)\\
&P(x_{t+1} \mid u_t, x_t) \;=\; \mathcal{N}(x_{t+1} \mid f_t(x_t, u_t), Q_t)\\
&P(z_t = 1 \mid u_t, x_t) \;=\; \exp\{-c_t(x_t, u_t)\}\\
&P(u_t \mid x_t; \theta)
\end{aligned}
$$

- idea of "costs/utilities/rewards $\to$ binary RV" is old
  (Cooper, 1988; Shachter, 1988)
  but here in neg-log space...

[Toussaint, ICML 2009; Toussaint & Goerick, IROS 2007]

- given the model, we can:

  a) compute the *trajectory posterior*

  $$q(x_{0:T}) = P(x_{0:T} \mid z_{0:T}\!=\!1)$$

- given the model, we can:

  a) compute the *trajectory posterior*

  $$q(x_{0:T}) = P(x_{0:T} \mid z_{0:T}=1)$$

  b) compute the *MAP control* conditioned on the state,

  $$u_t^{\mathsf{MAP}}(x_t) = \operatorname{argmax}_{u_t} P(u_t \mid x_t, z_{0:T}=1)$$

- given the model, we can:

  a) compute the *trajectory posterior*

  $$q(x_{0:T}) = P(x_{0:T} \mid z_{0:T}=1)$$

  b) compute the *MAP control* conditioned on the state,

  $$u_t^{\mathsf{MAP}}(x_t) = \mathrm{argmax}_{u_t} P(u_t \mid x_t, z_{0:T}=1)$$

  c) compute the *maximum likelihood* parameter

  $$\theta^{\mathsf{ML}} = \mathrm{argmax}_\theta P(z_{0:T}=1; \theta)$$

  b) and c) require a)

Relation to SOC

- trajectory log-likelihood $=$ negative cost:

$$\log P(z_{0:T}\!=\!1 \,|\, \xi) = -C(\xi)\;, \qquad\qquad \xi \equiv (x_{0:T}, u_{0:T})$$

$\Rightarrow$ *ML trajectory = "optimal" trajectory*

Relation to SOC

- trajectory log-likelihood $=$ negative cost:

$$\log P(z_{0:T}\!=\!1 \,|\, \xi) = -C(\xi)\;, \qquad\qquad \xi \equiv (x_{0:T}, u_{0:T})$$

$\Rightarrow$ *ML trajectory = "optimal" trajectory*

- log-likelihood $\neq$ expected cost:

$$\begin{aligned}\log P(z_{0:T}\!=\!1) &= \log \mathrm{E}_\xi\{P(z_{0:T}\!=\!1 \,|\, \xi)\} \\ &\geq \mathrm{E}_\xi\{\log P(z_{0:T}\!=\!1 \,|\, \xi)\} = -\mathrm{E}_\xi\{C(\xi)\}\;,\end{aligned}$$

Relation to SOC

$\Rightarrow$ SOC and the inference model are, in general, different

Relation to SOC

$\Rightarrow$ SOC and the inference model are, in general, different

- they imply different notions of optimality, for example:

  SOC: minimize expected costs associated with collisions
  inference: maximize likelihood of non-collision

Relation to SOC

$\Rightarrow$ SOC and the inference model are, in general, different

- they imply different notions of optimality, for example:

  SOC: minimize expected costs associated with collisions
  inference: maximize likelihood of non-collision

  ...who decides which notion of optimality is "better"?

Relation to SOC

$\Rightarrow$ SOC and the inference model are, in general, different

- they imply different notions of optimality, for example:

  SOC: minimize expected costs associated with collisions
  inference: maximize likelihood of non-collision

  ...who decides which notion of optimality is "better"?

- however, in the LQG case they coincide...

Inference

- LQG case:

$$\text{LQG} \leftrightarrow \text{Gaussian graphical model}$$

$\rightarrow$ inference is trivial (Kalman smoothing)
$\rightarrow$ bwd messages equivalent to Ricatti equation
$\rightarrow$ MAP controller is also the optimal SOC controller
("Kalman duality")

Inference

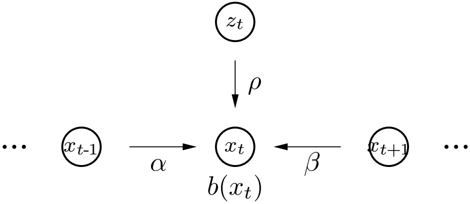
- non-LQG case
  - particles? (high dim...)
  - extended Kalman? (crude but fast)
  - UCT? (many evaluations in high dim.)
  - EP? (see poster)

- in our applications:
  - evaluating costs (collisions) is expensive
  - linearization, message computations, etc are cheap

  $\rightarrow$ extended Kalman messages and EP

inference

- Gaussian message updates à la extended Kalman smoothing
- linearize at mode of current belief



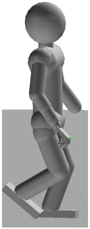
loop back-and-forth over t $\{$

update messages α, β, ϱ until $b(x_t)$ converges

$\}$

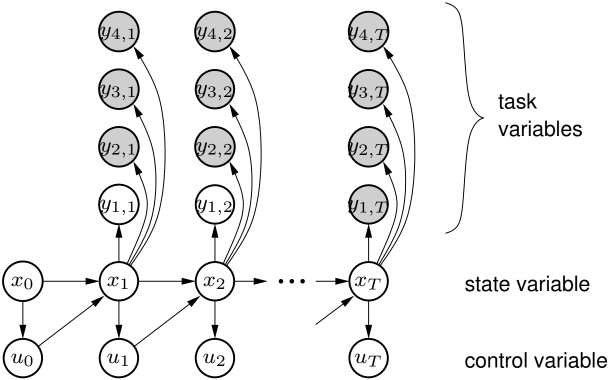
- focus computational efforts, avoid recomputing same things

Example 1



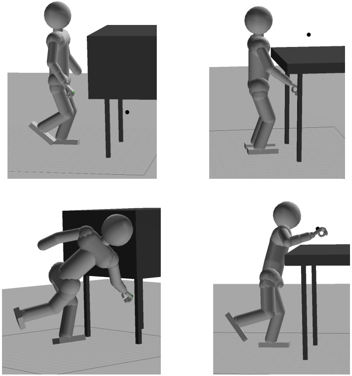
- in typical robotics scenarios we have multiple task variables
- e.g., we have a humanoid
  - x_t = posture of humanoid at time t
  - 1. task variable: $y_1 \in \mathbb{R}^3$ is the robot's finger tip position
  - 2. task variable: $y_2 \in \mathbb{R}^2$ is the robot's balance (horizontal offset)
  - 3. task variable: $y_3 \in \mathbb{R}$ measures collision/proximity
- for each task variable, we have
  - the kinematic function $\phi_i : x \mapsto y_i$, its Jacobian $J_i(x)$
  - desired values $y^*_{i,t}$ and variances $C_{i,t}$ for each time step t (corresponds to quadratic cost terms – could be more general..)

- this corresponds to the model



$$\forall_i : \; P(y_{i,t} \,|\, x_t) = \mathcal{N}(y_{i,t} \,|\, \phi_i(x_t), C_{i,t})$$

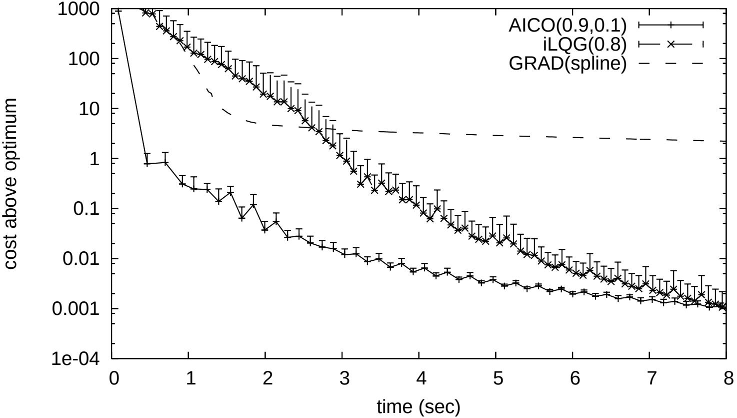
Example 1



- $\sim$30 DoF robot, task variables:
  - $y_1 \in \mathbb{R}^3$ is the robot's finger tip position
  - $y_2 \in \mathbb{R}^2$ is the robot's balance (horizontal offset to support)
  - $y_3 \in \mathbb{R}$ measures collision/proximity

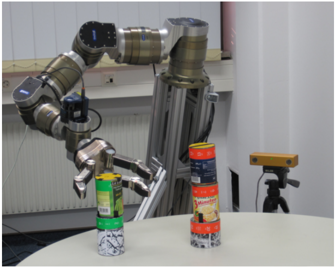
- cost parameters:
  (a) $C_{1,T} = 10^{-5}$, $C_{1,0:T\text{-}1} = 10^4$, $C_{2,0:T} = C_{3,0:T} = 10^{-5}$
  (b) $C_{1,T} = 10^{-2}$

1(a)



AICO(0.9,0.1)
iLQG(0.8)
GRAD(spline)
1000
100
10
1
0.1
0.01
0.001
1e-04
cost above optimum
0
1
2
3
4
5
6
7
8
time (sec)

Example 2



- hardware:
  - Schunk arm LWA (7DoF)
  - Schunk hand SDH (7DoF)
  - tactile sensor arrays
  - vision (Bumblebee stereo camera)
- 14 joints, dynamic $\rightarrow x_t \in \mathbb{R}^{28}$

- PRADA to plan on the stochastic relational level (Tobias Lang)
- AICO to generate fluent reach-and-pre-grasp trajectories

  we condition on:

  – no collisions, no limits along the whole trajectory

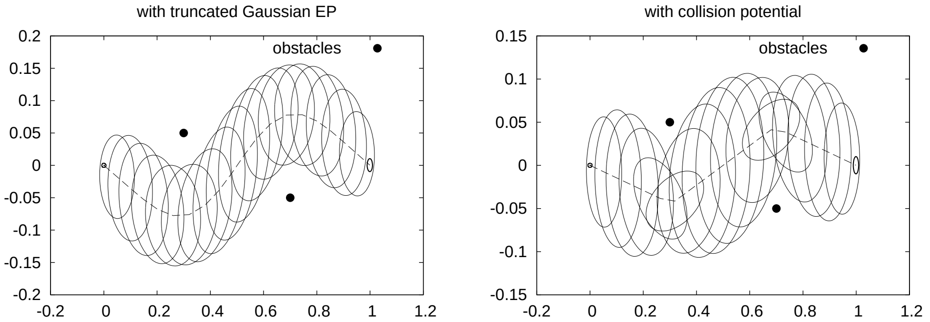
  – final endeffector (center of palm) position = object position

  – final finger-surface distance = 3cm

  – final finger normals are opposing
- vision to localize cans

Example 3

- Expectation Propagation when hard constraints truncate Gaussian beliefs
  – local collision hyperplanes
  – joint limits



Summary

- bottom line: *control as inference*

- my primary focus: *fast* approximate inference (AICO)

- goals for the near future:
  – additional computational tricks
  $\rightarrow$ speedup at least another factor of 10
  $\rightarrow$ fully online planning

(code at my webpage)

thanks!