Inverse KKT - Learning Cost functions of Manipulation from Demonstration

Englert, P., Vien, N. A., & Toussaint, M.

IJRR 2017

Presenter: Yu-Siang Wang

Outline

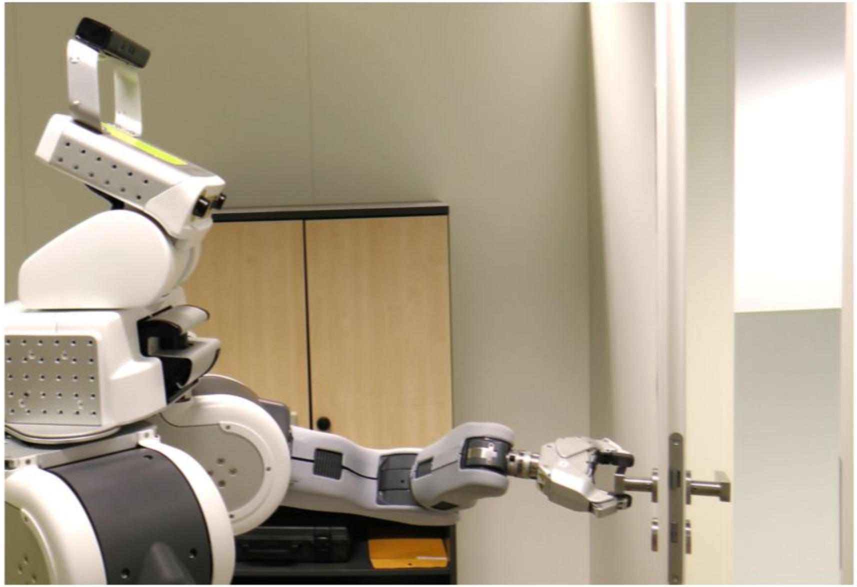
- Problem Statement
- Contribution
- Background
- Methods
- Experiments & Results
- Takeaway

Problem Statement

- Problem Statement
- Contribution
- Background
- Methods
- Experiments & Results
- Takeaway

Problem Statement

Learn the cost(reward) function from Demonstration → Inverse Optimal Control



Contribution

- Problem Statement
- Contribution
- Background
- Methods
- Experiments & Results
- Takeaway

Contribution

- Learn the cost function (Inverse Optimal Control) with the KKT condition for the constrained motion optimization
- A formulation of square hand-crafted features as cost function and a formulation of kernel method
- These two methods can be reduced as a constrained quadratic optimization problem and easily solved with the existing quadratic solver

Contribution

- Problem Statement
- Contribution
- Background
- Methods
- Experiments & Results
- Takeaway

Background - Optimization

Objective function $$x^* = arg\min_x f(x, y, w)$$

Background - Optimization

Objective function $$x^* = arg\min_x f(x, y, w)$$

Constraint s.t. $$h(x, y) = 0$$

Background - Optimization - Lagrangian Multiplier

Objective function $$x^* = arg\,\min_x f(x, y, w)$$

Constraint s.t. $$h(x, y) = 0$$

Lagrangian function $$L(x, \lambda) = f(x) + \lambda h(x)$$

Background - Optimization - Lagrangian Multiplier

Objective function $$x^* = arg\min_x f(x, y, w)$$

Constraint s.t. $$h(x, y) = 0$$

Lagrangian function $$L(x, \lambda) = f(x) + \lambda h(x)$$

$$\nabla L(x, \lambda) = \nabla f(x) + \lambda \nabla h(x) = 0$$

Background - Optimization

Objective function $$x^* = arg\min_x f(x, y, w)$$

Constraint s.t. $$h(x, y) = 0$$

$$g(x, y) \leq 0$$

Karush-Kuhn-Tucker conditions

Given general problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \ldots m \\ & \ell_j(x) = 0, \quad j = 1, \ldots r \end{aligned}$$

The **Karush-Kuhn-Tucker conditions** or **KKT conditions** are:

- $0 \in \partial f(x) + \sum_{i=1}^{m} u_i \partial h_i(x) + \sum_{j=1}^{r} v_j \partial \ell_j(x)$ (stationarity)
- $u_i \cdot h_i(x) = 0$ for all i (complementary slackness)
- $h_i(x) \leq 0, \ \ell_j(x) = 0$ for all i, j (primal feasibility)
- $u_i \geq 0$ for all i (dual feasibility)

Ref: Geoff Gordon & Ryan Tibshirani Optimization 10-725 / 36-725

Karush-Kuhn-Tucker conditions

Given general problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \ldots m \\ & \ell_j(x) = 0, \quad j = 1, \ldots r \end{aligned}$$

The **Karush-Kuhn-Tucker conditions** or **KKT conditions** are:

- $0 \in \partial f(x) + \sum_{i=1}^{m} u_i \partial h_i(x) + \sum_{j=1}^{r} v_j \partial \ell_j(x)$ (stationarity)
- $u_i \cdot h_i(x) = 0$ for all i (complementary slackness)
- $h_i(x) \leq 0, \; \ell_j(x) = 0$ for all i, j (primal feasibility)
- $u_i \geq 0$ for all i (dual feasibility)

Ref: Geoff Gordon & Ryan Tibshirani Optimization 10-725 / 36-725

Background - Optimization - KKT

Objective function $$x^* = arg\min_x f(x, y, w)$$

Constraint s.t. $h(x, y) = 0 \quad g(x, y) \leq 0$

Lagrangian function $$L(x, \lambda, \mu) = f(x) + \lambda h(x) + \mu g(x)$$

First KKT condition $$\nabla L(x, \lambda) = \nabla f(x) + \lambda \nabla h(x) + \mu \nabla g(x) = 0$$

Background --Task Settings - Features

Cost function: $f(x, y, w) = \sum_t^T w_t^\top \phi_t^2(x_t, y)$

ϕ: features. Differences between the forward kinematics mapping and object position (given by y)

- **Transition Features**: Smoothness of the motion (sum of squared acceleration or torques)
- **Position Features**: Represent a body position relative to another body
- **Orientation Features**: Represent orientation of a body relative to other body

Background -- Task Settings - weighting vector w

Cost function: $f(x, y, w) = \sum_t^T w_t^\top \phi_t^2(x_t, y)$

w_t: Weighting vector at time t. Given in optimal control. Required to solve in the inverse optimal control scenario

Background -- Task Settings - constraints

Cost function: $f(x, y, w) = \sum_t^T w_t^\top \phi_t^2(x_t, y)$

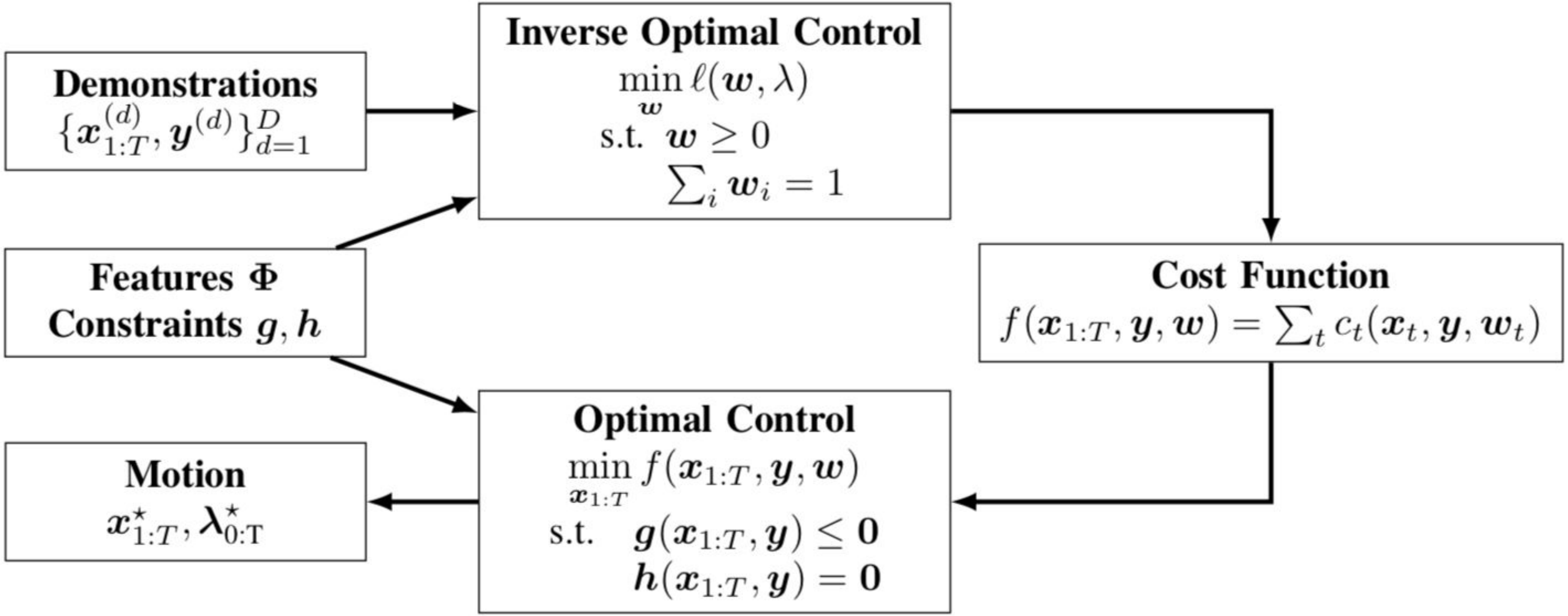
Constraint:

$g_t(x, y) \leq 0$: The smallest distance difference between the forward kinematics mapping and object position has to be larger than a threshold. [Body orientation or relative positions between robot and an object]

$h_t(x, y) = 0$: The distance between hand and object that should be exact zero

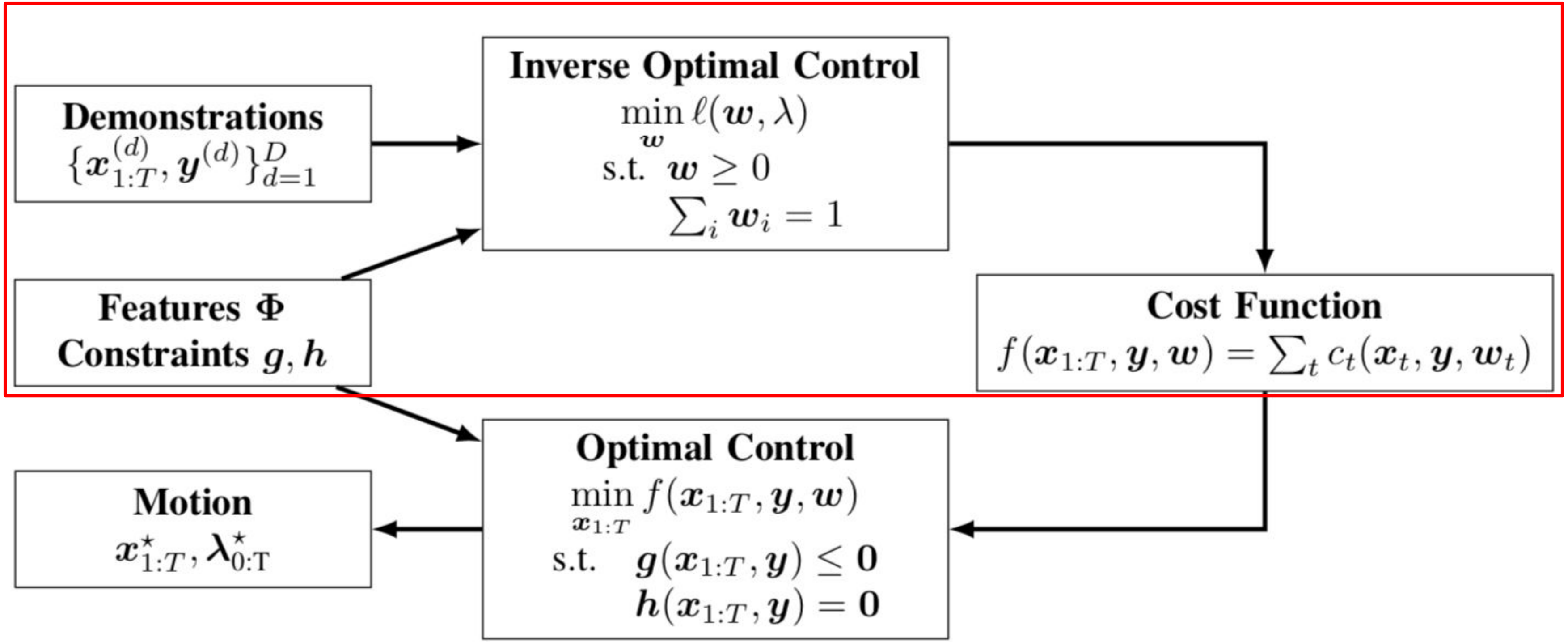
Optimal Control and Inverse Optimal Control

$$f(x, y, w) = w^\top \phi^2$$



Inverse KKT overview

$$f(x, y, w) = w^\top \phi^2$$



Methods

- Problem Statement
- Contribution
- Background
- Methods
- Experiments & Results
- Takeaway

Inverse Optimal Control -- features method

Cost function $$f(x, y, w) = w^\top \phi^2$$

Constraint s.t. $$h(x, y) = 0 \quad g(x, y) \leq 0$$

Goal: Given demonstration x* and y $\longrightarrow$ Find the optimal w

Inverse Optimal Control -- features method

Cost function $$f(x, y, w) = w^\top \phi^2$$

Constraint s.t. $$h(x, y) = 0 \quad g(x, y) \leq 0$$

Lagrangian function $$L(x, \lambda, w) = f(x) + \lambda^\top [h(x), g(x)]$$

First KKT condition $$\nabla L(x, \lambda, w) = \nabla f(x) + \lambda^\top \nabla [h(x), g(x)]$$
$$= 0$$

Inverse Optimal Control -- features method

If we assume the demonstration x* is the optimal demonstration

$$\nabla L(x^*, \lambda, w) = \nabla f(x^*) + \lambda^\top \nabla [h(x^*), g(x^*)] = 0$$

Inverse Optimal Control -- features method

If we assume the demonstration x* is the optimal demonstration

$$\nabla L(x^*, \lambda, w) = \nabla f(x^*) + \lambda^\top \nabla[h(x^*), g(x^*)] = 0$$

Just find the w and λ make the equation hold!

Inverse Optimal Control -- features method

If we assume the demonstration x* is the optimal demonstration

$$\nabla L(x^*, \lambda, w) = \nabla f(x^*) + \lambda^\top \nabla [h(x^*), g(x^*)] = 0$$

Just find the w and λ make the equation hold! **Very hard to do it!**

Inverse Optimal Control -- features method

Treat it as a loss function and find the optimal w through the optimization method

$$\nabla L(x^*, \lambda, w) = \nabla f(x^*) + \lambda^\top \nabla [h(x^*), g(x^*)]$$

Loss function: l, D: number of demonstration

$$l = \sum_{d=1}^{D} l^d$$

$$l^d = ||\nabla L(x^*, \lambda, w)||^2$$

Inverse Optimal Control -- features method

Goal: Find the optimal w.

Problem to solve w?

$$l = \sum_{d=1}^{D} l^d$$

$$l^d = ||\nabla L(x^*, \lambda, w)||^2$$

Inverse Optimal Control -- features method

Goal: Find the optimal w.
Problem to solve w?

$$l = \sum_{d=1}^{D} l^d$$

$$l^d = ||\nabla L(x^*, \lambda, w)||^2$$

Two unknown variables here! We don’t know λ!

Inverse Optimal Control -- features method

Goal: Find the optimal w.
Problem to solve w?

$$l = \sum_{d=1}^{D} l^d$$

$$l^d = ||\nabla L(x^*, \lambda, w)||^2$$

Two unknown variables here! We don’t know λ!
Represent λ with w to be a single variable optimization

$$\nabla_{\boldsymbol{\lambda}^{(d)}} \ell^{(d)}(\boldsymbol{w}, \boldsymbol{\lambda}^{(d)}) = \mathbf{0}$$
$$\Rightarrow \quad \boldsymbol{\lambda}^{(d)}(\boldsymbol{w}) = -2(\tilde{\boldsymbol{J}}_{\boldsymbol{c}} \tilde{\boldsymbol{J}}_{\boldsymbol{c}}^{\top})^{-1} \tilde{\boldsymbol{J}}_{\boldsymbol{c}} \boldsymbol{J}^{\top} \operatorname{diag}(\boldsymbol{\Phi}) \boldsymbol{w}$$

Inverse Optimal Control -- features method

Goal: Find the optimal w.

$$l = \sum_{d=1}^{D} l^d$$

$$l^d = ||\nabla L(x^*, \lambda, w)||^2$$

$$\ell^{(d)}(\boldsymbol{w}) = 4\boldsymbol{w}^\top \underbrace{\mathrm{diag}(\boldsymbol{\Phi})\boldsymbol{J}\Big(\boldsymbol{I} - \tilde{\boldsymbol{J}}_{\boldsymbol{c}}^\top (\tilde{\boldsymbol{J}}_{\boldsymbol{c}} \tilde{\boldsymbol{J}}_{\boldsymbol{c}}^\top)^{-1} \tilde{\boldsymbol{J}}_{\boldsymbol{c}}\Big)\boldsymbol{J}^\top \mathrm{diag}(\boldsymbol{\Phi})}_{\boldsymbol{\Lambda}^{(d)}} \boldsymbol{w}$$

$l^d(w)$: is a function of w and all the other terms are given

Inverse Optimal Control -- features method

Goal: Find the optimal w.

$$l = \sum_{d=1}^{D} l^d$$

$$l^d = ||\nabla L(x^*, \lambda, w)||^2$$

$$\ell^{(d)}(\boldsymbol{w}) = 4\boldsymbol{w}^\top \underbrace{\mathrm{diag}(\boldsymbol{\Phi})\boldsymbol{J}\left(\boldsymbol{I} - \tilde{\boldsymbol{J}}_{\boldsymbol{c}}^\top (\tilde{\boldsymbol{J}}_{\boldsymbol{c}} \tilde{\boldsymbol{J}}_{\boldsymbol{c}}^\top)^{-1} \tilde{\boldsymbol{J}}_{\boldsymbol{c}}\right)\boldsymbol{J}^\top \mathrm{diag}(\boldsymbol{\Phi})}_{\boldsymbol{\Lambda}^{(d)}}\boldsymbol{w}$$

$l^d(w)$: is a function of w and all the other terms are given

$$\min_w w^\top \Lambda^d w \quad \text{s.t.} \quad w \geq 0$$ (Quadratic optimization)

Inverse Optimal Control -- features method

Goal: Find the optimal w.

$$\min_w w^\top \Lambda^d w \quad \text{s.t.} \quad w \geq 0$$

Inverse Optimal Control -- features method

Goal: Find the optimal w.

$$\min_w w^\top \Lambda^d w \quad \text{s.t.} \quad w \geq 0$$

Problem?

Inverse Optimal Control -- features method

Goal: Find the optimal w.

$$\min_w w^\top \Lambda^d w \quad \text{s.t.} \quad w \geq 0$$

Problem?

w can be all zeros!

Inverse Optimal Control -- features method

Goal: Find the optimal w.

Add constraint for w! $\min_w w^\top \Lambda^d w$ s.t. $w \geq 0$ $\sum_i w_i = 1$

Inverse Optimal Control -- features method

Goal: Find the optimal w.

Add constraint for w! $\min_w w^\top \Lambda^d w$ s.t. $w \geq 0$ $\sum_i w_i = 1$

Linear Solution

$$w = A\rho$$

where A is given (one parameter to multiple task)

Inverse Optimal Control -- features method

Goal: Find the optimal w.

Add constraint for w! $\min_w w^\top \Lambda^d w$ s.t. $w \geq 0$ $\sum_i w_i = 1$

Nonlinear Solution

$$w = A(\rho)$$

w is a gaussian distribution function of t. Mean and variance in Gaussian is described by ρ

Inverse Optimal Control -- features method

Goal: Find the optimal w.

$$f(x, y, w) = w^\top \phi^2$$

$$l = \sum_{d=1}^{D} l^d$$

$$l^d = ||\nabla L(x^*, \lambda, w)||^2$$

$l^d(w)$: is a function of w and all the other terms are given

$$\min_w w^\top \Lambda^d w \quad \text{s.t.} \quad w \geq 0 \quad \sum_i w_i = 1$$

Method - Kernel Method

Kernel Method: Instead of using hand crafted features, using the features in the kernel space

Cost function f:

$$f(x, y, w) = w^\top \phi^2$$

Method - Kernel Method

Kernel Method: Instead of using hand crafted features, using the features in the kernel space

Cost function f:
α: weighting vector

$$\cancel{f(x, y, w) = w^\top \phi^2}$$

$$f(x, y, \alpha) = \alpha^\top k(x_1, x_2)$$

k: RBF kernel function Σ^{-1}: hyperparameters

$$k(x1, x2) = exp(-(x1 - x2)^\top \Sigma^{-1} (x_1 - x_2))$$

Method - Kernel Method

Goal: Solve α

$$f(x, y, \alpha) = \alpha^\top k(x_1, x_2)$$

Loss function will be optimized $l = \sum_{d=1}^{D} l^d$

$$l^d = ||\nabla L(x^*, \lambda, \alpha)||^2$$

Method - Kernel Method

Goal: Solve α

$$f(x, y, \alpha) = \alpha^\top k(x_1, x_2)$$

Loss function will be optimized $l = \sum_{d=1}^{D} l^d$

$$l^d = ||\nabla L(x^*, \lambda, \alpha)||^2$$

Represent loss function with α

$$\begin{aligned} \ell^{(d)}(\boldsymbol{\alpha}) &= \nabla \boldsymbol{f}_{\boldsymbol{x}_{1:T}}^\top \left(\boldsymbol{I} - \boldsymbol{J}_c^\top (\boldsymbol{J}_c \boldsymbol{J}_c^\top)^{-1} \boldsymbol{J}_c \right) \nabla \boldsymbol{f}_{\boldsymbol{x}_{1:T}} \\ &= \boldsymbol{\alpha}^\top \boldsymbol{\Omega}^{(d)} \boldsymbol{\alpha} \end{aligned}$$

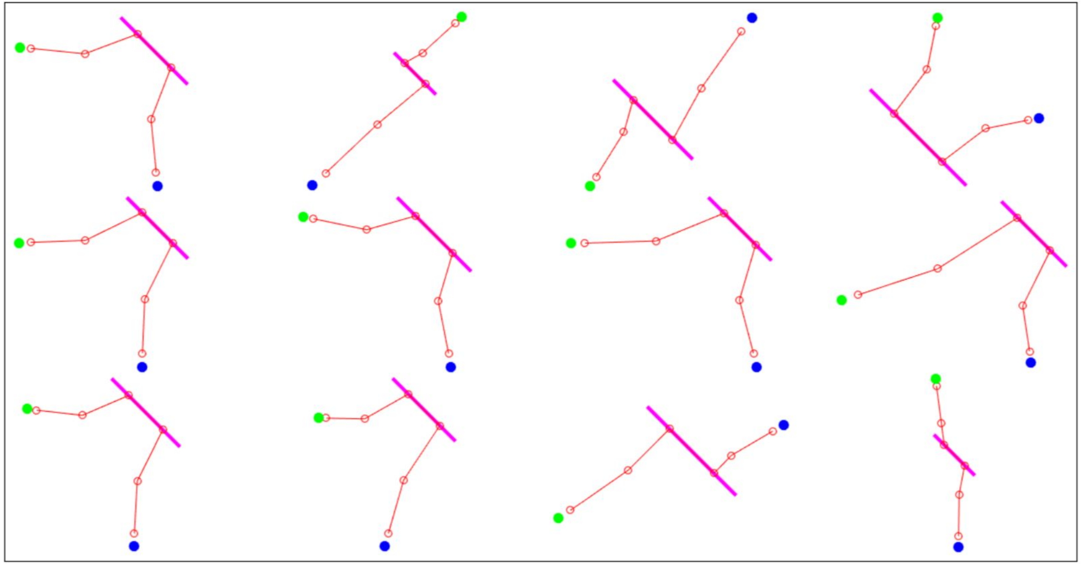
Solve α with quadratic solver

$$\min_\alpha \alpha^\top \Omega \alpha \quad \text{s.t.} \quad \alpha \geq 0 \quad \sum_i \alpha_i = 1$$

Experiments & Results

- Problem Statement
- Contribution
- Background
- Methods
- Experiments & Results
- Takeaway

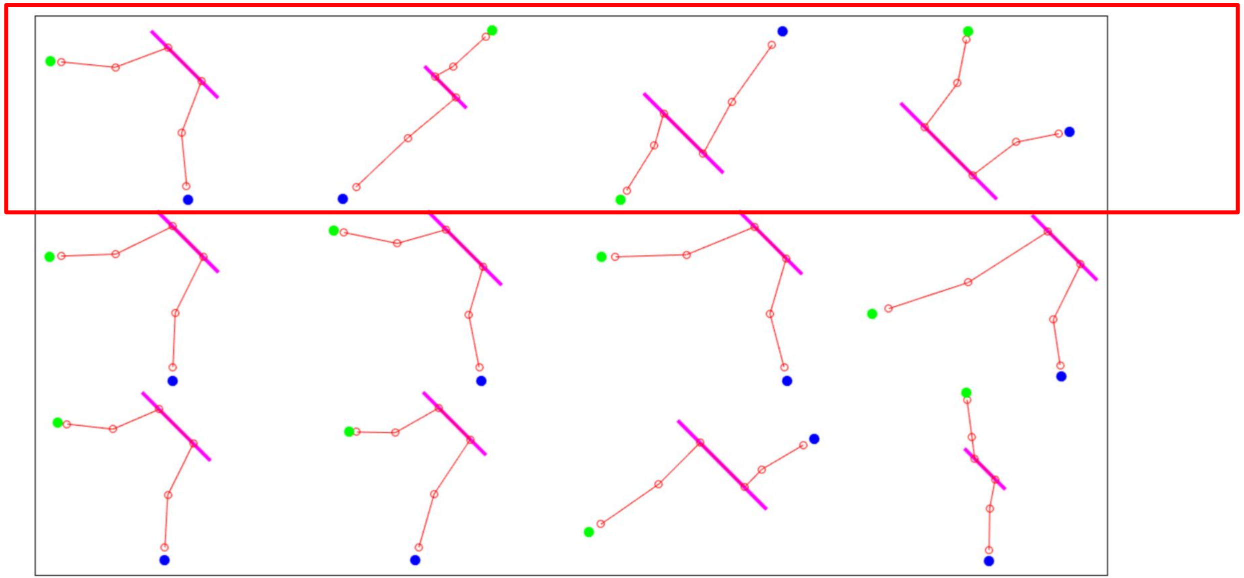
Experiments -- toy 2d example



Task: Start from green point and and end at blue point. 6 time steps in total and time step 3 and 4 should be in contact with the stick.

Experiments -- toy 2d example

Training Set

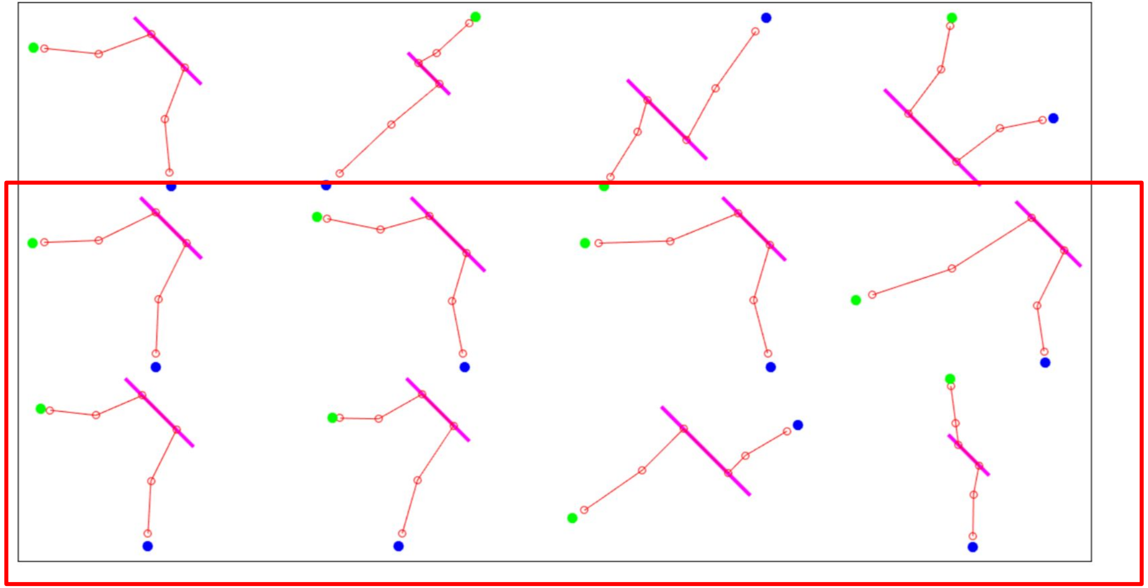


Task: Start from green point and and end at blue point. 6 time steps in total and time step 3 and 4 should be in contact with the stick.

Experiments -- toy 2d example

Training
Set

Testing
Set



Task: Start from green point and and end at blue point. 6 time steps in total and time step 3 and 4 should be in contact with the stick.

Results -- toy 2d example

method	error (train set)	error (test set)	constraint violation (train set)	constraint violation (test set)
IKKT (feature)	0.027475	0.46944	1.1102e-15	1.6653e-15
IKKT (kernel)	0.94625	66.065	4.4409e-16	8.2469e-16
CIOC	0.014732	0.64592	0.00058039	0.001128

Error: sum of absolute difference between the resulting motion with the learned weights w and the reference motion.

Constraint violation: Distance to the stick.

Ref: Levine and Koltun, Continuous Inverse Optimal Control with Locally Optimal Examples, ICML 2011

Results -- toy 2d example

method	error (train set)	error (test set)	constraint violation (train set)	constraint violation (test set)
IKKT (feature)	0.027475	0.46944	1.1102e-15	1.6653e-15
IKKT (kernel)	0.94625	66.065	4.4409e-16	8.2469e-16
CIOC	0.014732	0.64592	0.00058039	0.001128

Error: sum of absolute difference between the resulting motion with the learned weights w and the reference motion.

Error: Hand-crafted features << Kernel Method

Ref: Levine and Koltun, Continuous Inverse Optimal Control with Locally Optimal Examples, ICML 2011

Results -- toy 2d example

method	error (train set)	error (test set)	constraint violation (train set)	constraint violation (test set)
IKKT (feature)	0.027475	0.46944	1.1102e-15	1.6653e-15
IKKT (kernel)	0.94625	66.065	4.4409e-16	8.2469e-16
CIOC	0.014732	0.64592	0.00058039	0.001128

Constraint violation: Distance to the stick.

Constraint Violation Error: IKKT << CIOC

Ref: Levine and Koltun, Continuous Inverse Optimal Control with Locally Optimal Examples, ICML 2011

Experiments -- synthetic dataset

Synthetic dataset: longer time steps (50 time steps)

Groundtruth weighting vector w is known (But still requires to learn it)

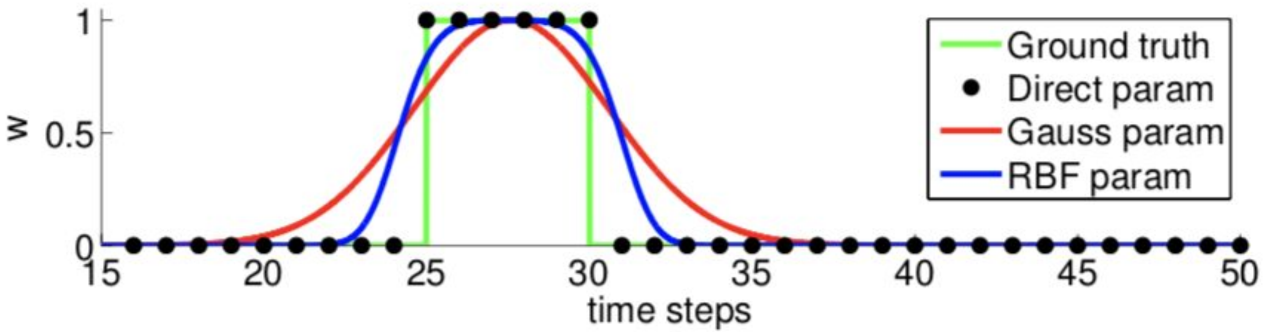
Experiments

Synthetic dataset: longer time steps (50 time steps)

Three methods

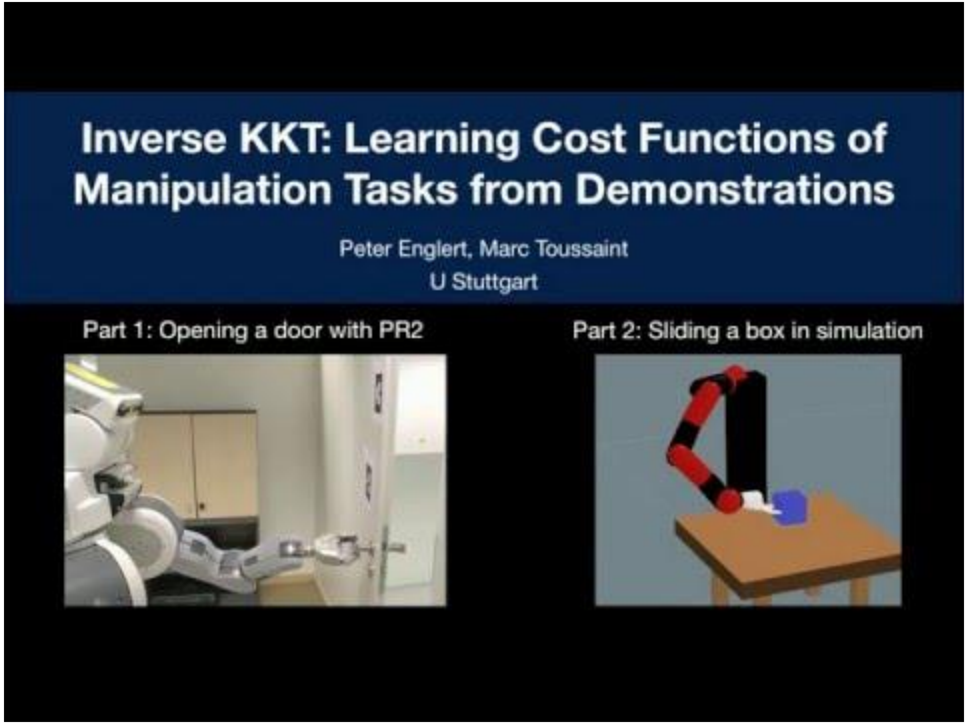
- Direct param: Each time step learn a parameter
- RBF param: 30 Gaussian with standard deviation 0.8 and uniformly distributed in 50 time steps.
- Nonlinear Gaussian: A single gaussian. The mean and the standard deviation are parametrized.

Results



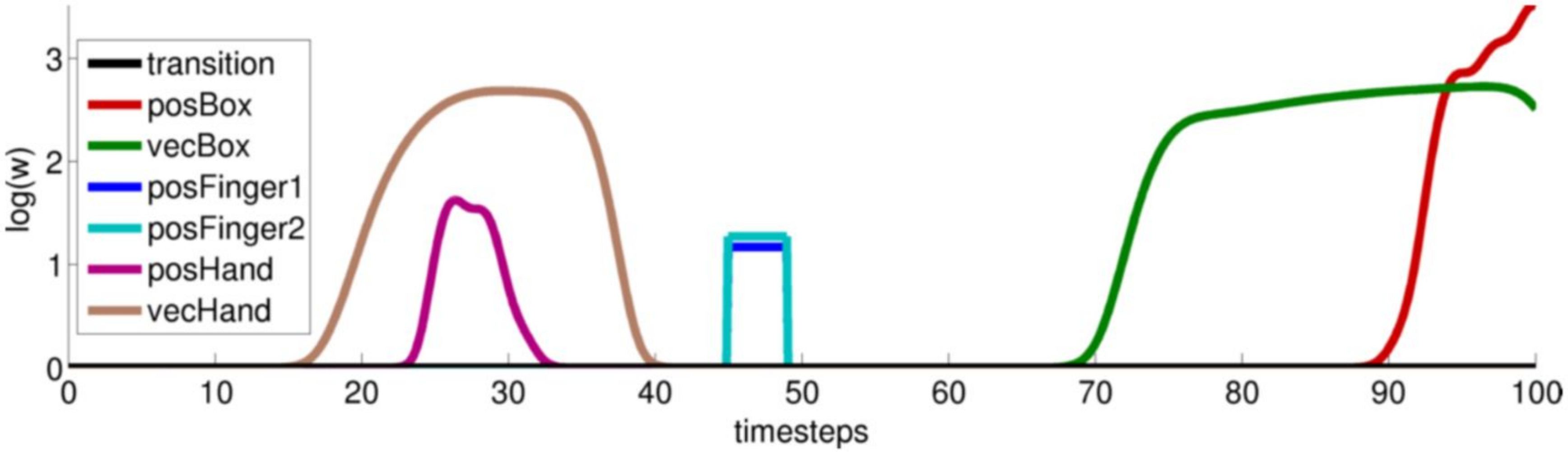
Direct param outperform the other methods

Experiments



https://www.youtube.com/watch?v=pO6XNiyJqNw

Results - Sliding Box on a table



Takeaway

- Problem Statement
- Contribution
- Background
- Methods
- Experiments & Results
- Takeaway

Takeaway

- Learn the cost function with the inverse KKT method for constrained motion optimization
- The author proposed two methods -- hand crafted features based method and kernel based method
- Both of the methods can be solved by existing quadratic solver

Discussion

- Handcrafted features works well. What if the task is too difficult and the handcrafted features are not good enough?
- Is $f(\boldsymbol{x}_{1:T}, \boldsymbol{y}, \boldsymbol{w}) = \boldsymbol{w}^\top \Phi^2(\boldsymbol{x}_{1:T}, \boldsymbol{y})$ a good enough cost function?

Questions

- The relation between optimal control and inverse optimal control
- The relation between loss function in inverse optimal control and the cost function in optimal control
- What two main methods do they use
- What’s the KKT first condition