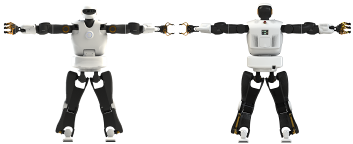


Humanoid robot motion generation @Gepetto

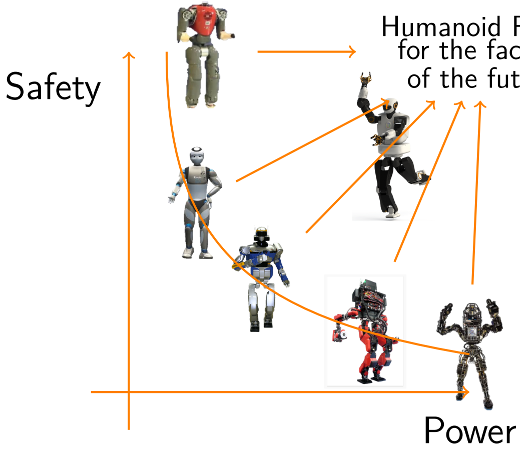
Workshop on Dynamic Legged Locomotion in Realistic Terrains, ICRA, 2018

25^{th} May 2018, Brisbane, Australia

O. Stasse, N. Mansard, A. Del Prete, S. Tonneau, J. Carpentier, M. Geisert, F. Lamiraux, J.-P. Laumond, GEPETTO

Acknowledgements

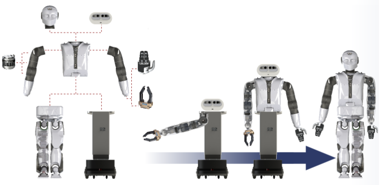
Testing our algorithms in real industrial use case



125 millions of holes drilled by year

75 % are done *manually*

Modular robots



Decreasing the robot cost by sharing parts

Autonomous Motion generation : the problem

$$\begin{cases} \min f(\mathbf{u}(t), \mathbf{v}(t)) \\ \mathbf{g}(\mathbf{u}(t), \mathbf{v}(t)) < 0 \\ \mathbf{h}(\mathbf{u}(t), \mathbf{v}(t)) = 0 \end{cases}$$

[Saidi,IROS 2007]

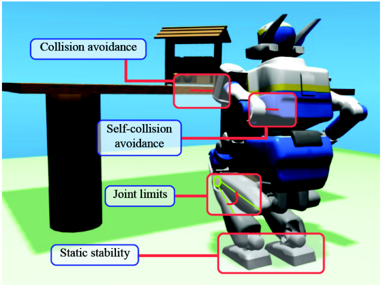
$\mathbf{f}(t)$: The cost function

$\mathbf{u}(t)$: The control vector

$\mathbf{g}(t)$: The inequality constraints

$\mathbf{h}(t)$: The equality constraints

$\mathbf{v}(t)$: The environment representation



Autonomous Motion generation : the problem

$$\begin{cases} \min_{\mathbf{q}(\cdot),\mathbf{u}(\cdot)} & \int_0^T L(t,\mathbf{q}(t),\mathbf{u}(t))dt \\ \dot{\mathbf{q}}(t) = f(\mathbf{q}(t),\mathbf{u}(t)), & t \in [0, T] \\ \mathbf{q}(0) = \mathbf{q}_0, \; T > 0 & \\ 0 \leq \mathbf{gh}(\mathbf{q}(t),\mathbf{u}(t)), & t \in [0, T] \end{cases}$$

[Saidi,IROS 2007]

f : The system dynamics

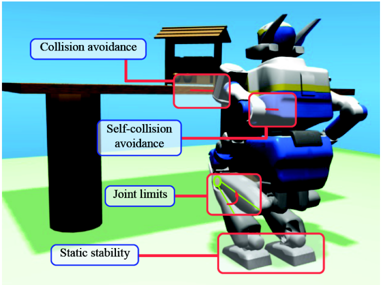
u : The control vector

q : The system state

gh : The (in)equality constraints

v : The environment representation

L : The cost

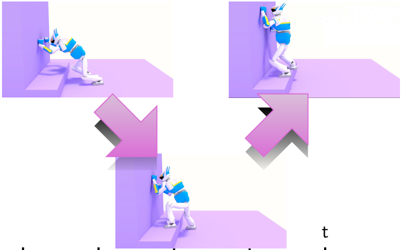


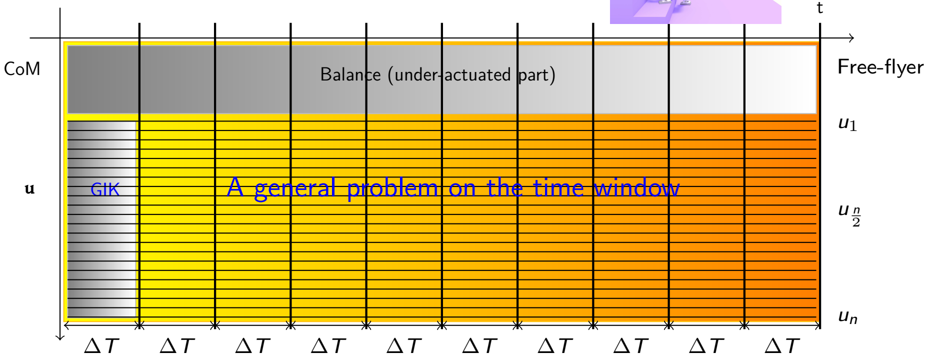
Motion generation : Model Predictive Control

- Size of the problem

$$1.6 \times 200 \times 30 = 9600 \text{ variables}$$

- Non linear constraints
- Discrete nature due to contacts





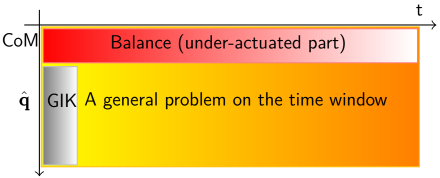
MuJoCo [Koenemann, IROS 2015]

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

Motion generation : the constraints

Pattern generator

- Focus on the underactuated part
- Model predictive control



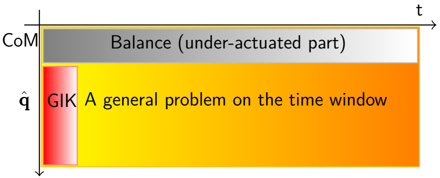
Simplifying the walking problem to control only the CoM reference

$$
\begin{cases}
\mathbf{M}_1(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_1(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_1(\mathbf{q}) = \mathbf{T}_1(\mathbf{q})\mathbf{u} + \mathbf{C}_1^\top(\mathbf{q})\lambda & \text{Actuated dynamics of the robot} \\
\mathbf{M}_2(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_2(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_2(\mathbf{q}) = \mathbf{C}_2^\top(\mathbf{q})\lambda & \text{Underactuated dynamics of the robot} \\
f(\lambda) \in \mathcal{F} & \text{General balance criteria} \\
\mathbf{u}_{min} < \mathbf{u} < \mathbf{u}_{max} & \text{Torques limits} \\
\hat{\mathbf{q}}_{min} < \hat{\mathbf{q}} < \hat{\mathbf{q}}_{max} & \text{Joints limits} \\
d(\mathcal{B}_i(\mathbf{q}), \mathcal{B}_j(\mathbf{q})) > \epsilon, \forall (i,j) \in \mathcal{P} & \text{(self-)collisions} \\
\ddot{\mathbf{e}}_i = \dot{\mathbf{J}}_i(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}_i(\mathbf{q})\ddot{\mathbf{q}} & \text{Tasks}
\end{cases}
$$

Motion generation : the constraints

Inverse dynamics

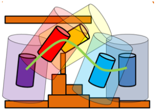
- Focus on the inertia matrix
- Forces
- Complete constraints

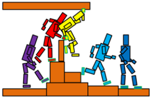


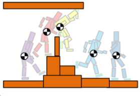
$$
\begin{cases}
\mathbf{M}_1(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_1(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_1(\mathbf{q}) = \mathbf{T}_1(\mathbf{q})\mathbf{u} + \mathbf{C}_1^\top(\mathbf{q})\lambda & \text{Actuated dynamics of the robot} \\
\mathbf{M}_2(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_2(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_2(\mathbf{q}) = \mathbf{C}_2^\top(\mathbf{q})\lambda & \text{Underactuated dynamics of the robot} \\
f(\lambda) \in \mathcal{F} & \text{General balance criteria} \\
\mathbf{u}_{min} < \mathbf{u} < \mathbf{u}_{max} & \text{Torques limits} \\
\hat{\mathbf{q}}_{min} < \hat{\mathbf{q}} < \hat{\mathbf{q}}_{max} & \text{Joints limits} \\
d(\mathcal{B}_i(\mathbf{q}), \mathcal{B}_j(\mathbf{q})) > \epsilon, \forall (i,j) \in \mathcal{P} & \text{(self-)collisions} \\
\ddot{\mathbf{e}}_i = \dot{\mathbf{J}}_i(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}_i(\mathbf{q})\ddot{\mathbf{q}} & \text{Tasks}
\end{cases}
$$

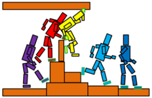
Loco3D pipeline

Reachability planner → Contacts Sequence → Centroidal dynamics → Whole body control



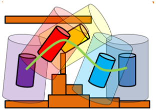


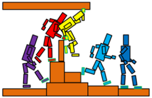


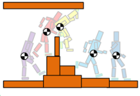


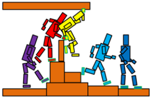
Loco3D pipeline

Reachability planner → Contacts Sequence → Centroidal dynamics → Whole body control





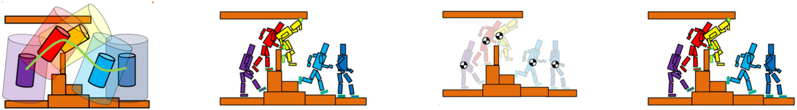




[Tonneau TRO 2018,Tonneau ISRR 2015]

Loco3D pipeline

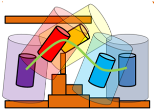
Reachability planner → Contacts Sequence → Centroidal dynamics → Whole body control

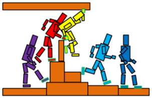


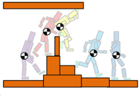
[Tonneau TRO 2018,Tonneau ISRR 2015]

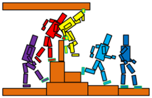
Loco3D pipeline

Reachability planner → Contacts Sequence → Centroidal dynamics → Whole body control









Newton

$$m(\ddot{\mathbf{c}}(\mathbf{u}) - \mathbf{g}) = \sum_{i=1}^{N_c} \mathbf{f}_i$$

Euler

$$m\mathbf{c}(\mathbf{u})_{\times}(\ddot{\mathbf{c}}(\mathbf{u}) - \mathbf{g}) + \mathbf{w}^{\top}(\mathbf{u})\mathbf{I}\mathbf{w}^{\top}(\mathbf{u}) = \sum_{i=1}^{N_c} \mathbf{p}_{i\times}\mathbf{f}_i + \mu_i$$

Previous work

[Luo, ICRA 2014]

[Vaillant, Humanoids 2014]

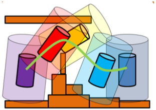
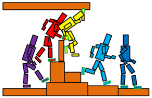
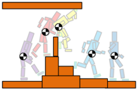
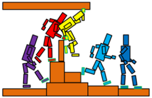
[Noda, ICRA 2014]

[Hirukawa, ICRA 2007]

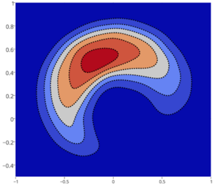
[Carpentier ICRA 2016]

Feedback from failure : from manual to automatic

Reachability planner ← Contacts Sequence ← Centroidal dynamics ↔ Whole body control

Considers kinematics constraints

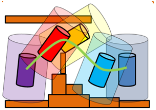
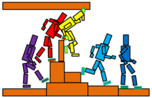
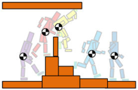
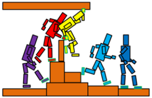


Probablity distribution of CoM positions

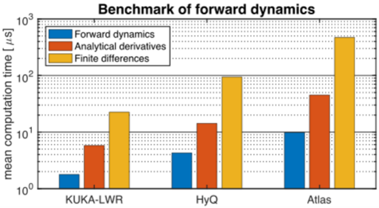
[Carpentier RSS 2017]

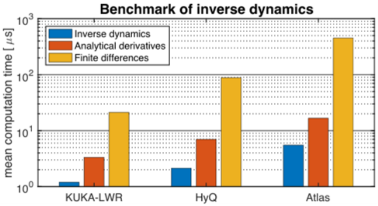
Feedback from failure : from manual to automatic

Reachability planner ← Contacts Sequence ← Centroidal dynamics ↔ DDP ↔ Whole body control

Analytical derivatives for Inverse Dynamics (*ID*) and Forwards Dynamics (*FD*)





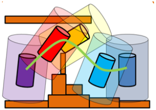
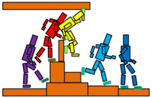
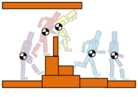
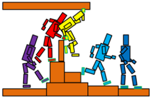
$\tau = FD(model, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}_{ext})$ with $\frac{\partial ID}{\partial \mathbf{q}}, \frac{\partial FD}{\partial \dot{\mathbf{q}}}, \frac{\partial FD}{\partial \ddot{\mathbf{q}}}, \frac{\partial FD}{\partial \mathbf{f}_{ext}}$

[Carpentier RSS 2018, to be presented]

$\tau = ID(model, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}_{ext})$ with $\frac{\partial ID}{\partial \mathbf{q}}, \frac{\partial ID}{\partial \dot{\mathbf{q}}}, \frac{\partial ID}{\partial \ddot{\mathbf{q}}}, \frac{\partial ID}{\partial \mathbf{f}_{ext}}$

Feedback from failure : from manual to automatic

Reachability planner ← Contacts Sequence ↔ Centroidal dynamics ↔ Whole body control

$$m(\ddot{\mathbf{c}}(\mathbf{u})-\mathbf{g}) = \sum_{i=1}^{N_c}\mathbf{f}_i$$

$$m\mathbf{c}(\mathbf{u})_\times(\ddot{\mathbf{c}}(\mathbf{u})-\mathbf{g}) + \mathbf{w}^\top(\mathbf{u})\mathbf{I}\mathbf{w}^\top(\mathbf{u}) = \sum_{i=1}^{N_c}\mathbf{p}_{i\times}\mathbf{f}_i + \mu_i$$

$$\mathbf{H}^{\{p\}}\underbrace{\begin{bmatrix} m(\ddot{\mathbf{c}}-\mathbf{g}) \\ m\mathbf{c}\times(\ddot{\mathbf{c}}-\mathbf{g})+\dot{\mathbf{L}}\end{bmatrix}}_{\mathbf{w}} \leq \mathbf{h}^{\{p\}}$$

$$\mathbf{c}(t) = \sum_{i=0}^{n} B_i^n(t/T)\mathbf{P}_i$$

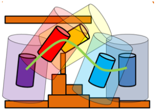
$$\mathbf{w}_i(\mathbf{y}) = \mathbf{w}_i^y\mathbf{y} + \mathbf{w}_i^s$$

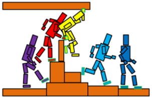
$$\mathbf{c}(t,\mathbf{y}) = \sum_{i\in\{0,1,2,4,5,6\}} B_i^6(t/T)\mathbf{P}_i + B_3^6(t/T)\mathbf{y}$$

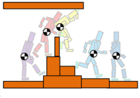
$$\boxed{\begin{array}{rl} \textbf{find} & \mathbf{y} \\ \text{subject to} & \mathbf{K}^{\{p\}}\mathbf{y} \leq \mathbf{k}^{\{p\}} \\ & (m\mathbf{H}^{\{p\}}\mathbf{w}_i^y)\mathbf{y} \leq \mathbf{h}^{\{p\}} + m\mathbf{H}^{\{p\}}\left(\begin{bmatrix}\mathbf{g}\\0\end{bmatrix} - \mathbf{w}_i^s\right) \quad ,\forall i \end{array}}$$

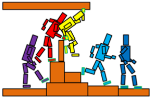
Feedback from failure : from manual to automatic

Reachability planner ← Contacts Sequence ↔ Centroidal dynamics ↔ Whole body control



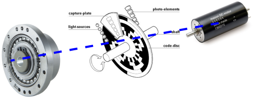
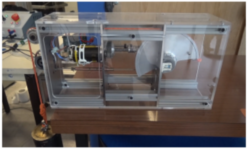






[Fernbach IROS 2018, Submitted]

Actuator dynamics

Classical actuator	SEA - Romeo	Mc Kibben Muscle - [IROS 2016]
		

Actuator dynamics : necessity to control

$$
\begin{cases}
\phi(\mathbf{u}) = \tau_j & \text{Actuators dynamics} \\
\mathbf{M}_1(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_1(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_1(\mathbf{q}) = \mathbf{T}_1(\mathbf{q})\tau_j + \mathbf{C}_1^\top(\mathbf{q})\mathbf{f} & \text{Actuated dynamics of the robot} \\
\mathbf{M}_2(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_2(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_2(\mathbf{q}) = \mathbf{C}_2^\top(\mathbf{q})\mathbf{f} & \text{Underactuated dynamics of the robot} \\
f(\mathbf{f}) \in \mathcal{F} & \text{General balance criteria} \\
\mathbf{u}_{min} < \mathbf{u} < \mathbf{u}_{max} & \text{Torques limits} \\
\hat{\mathbf{q}}_{min} < \hat{\mathbf{q}} < \hat{\mathbf{q}}_{max} & \text{Joints limits} \\
d(\mathcal{B}_i(\mathbf{q}), \mathcal{B}_j(\mathbf{q})) > \epsilon, \forall (i,j) \in \mathcal{P} & \text{(self-)collisions} \\
\ddot{\mathbf{e}}_i = \dot{\mathbf{J}}_i(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}_i(\mathbf{q})\ddot{\mathbf{q}} & \text{Tasks}
\end{cases}
$$

Torque control

T. Flayols and al. [ICHR, 2017]

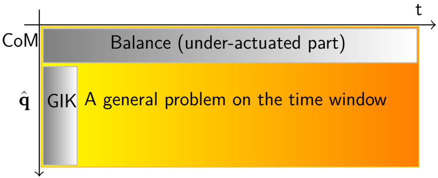
MPC : Solving the complete problem

Model Predictive Control Take Away Message

- Pros : Highly capable
- Widely used
- Cons : High computation power
- Maybe sensitive to models error
- Local method
- Sensitive to initial solution

Motion generation

$$
\begin{cases}
\min f(\mathbf{u}(t), \mathbf{v}(t)) \\
\mathbf{g}(\mathbf{u}(t), \mathbf{v}(t)) < 0 \\
\mathbf{h}(\mathbf{u}(t), \mathbf{v}(t)) = 0
\end{cases}
$$

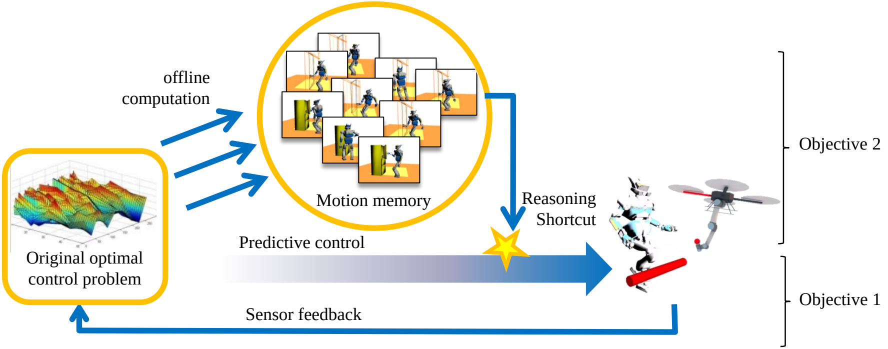


- Planning and control solve the same problem
  - Planning is looking for a global feasible solution
  - Control is looking for on online sensor grounded local solution
- Planning is too long when simulating the control
- Control can fails
  - Local minima leading to an incomplete behavior
  - Mismatch between the control and the hardware
- Accessibility set [Majumdar, ICRA Best Paper Award 2013]

Planning learned accessibility sets

[Perrin and al., TRO, 2009]

Memory of motion

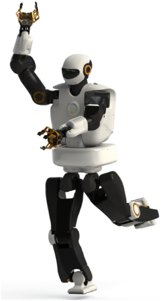


Memmo H2020 Project - Scientific coordinator : N. Mansard

Take away messages

- The *embodiment* (mechanical body, limits and controllers) defines the motion capabilities of the robot.
- We need to connect the accessibility set of our controllers to the planner.
- We need an efficient computation of the mechanical quantities
- We need to break down the problem complexity with small but representative problems
- We need to push higher the semantic level of our motion controllers

Control of complex dynamical systems



Model Predictive Control

- Pros : Highly capable
- Widely used
- Cons : High computation power
- Maybe sensitive to models error

Model free approach

- Pros : Computational time
- Adaptation
- Cons : Off line learning phase taking too long
- Exhaustive exploration of state space

Combining MPC and model free approaches

MPC to speed up the traning of model free approaches

- Local optimization trajectory to speed up desired control policy
- Speed up training phase
- Explore a larger part of the state space
- Needs a very accurate learning to avoid unstable policies

Learning models/techniques to improve model based techniques

- Learn a function off-line to use it in a MPC online
- Value Function, system dynamics, cost function
- Library of optimal trajectories
- Value function in a RRT

Combining MPC and model free approaches

MPC to speed up the traning of model free approaches

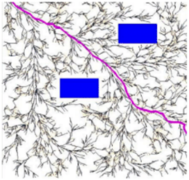
- Local optimization trajectory to speed up desired control policy
- Speed up training phase
- Explore a larger part of the state space
- Needs a very accurate learning to avoid unstable policies

Learning models/techniques to improve model based techniques

- Learn a function off-line to use it in a MPC online
- Value Function, system dynamics, cost function
- Library of optimal trajectories
- Value function in a RRT

Our method : combining both

Our approach



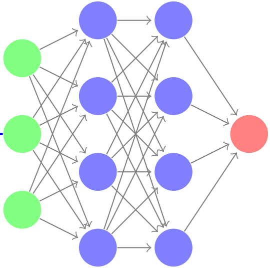
Kino Dynamic
Probabilistic Roadmap
30-50 states, densely connected

Sampling

Data set of subtrajectories

10-100k items

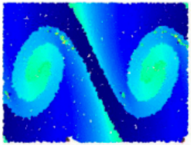
Regression ↓ (stoch. grad.)



Neural network, (2x512 hidden units)

Query

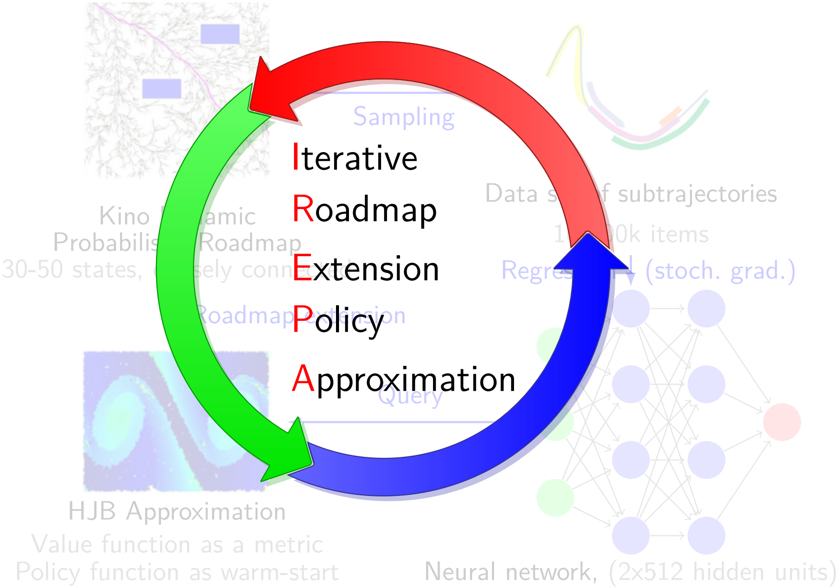
↑ Roadmap extension



HJB Approximation

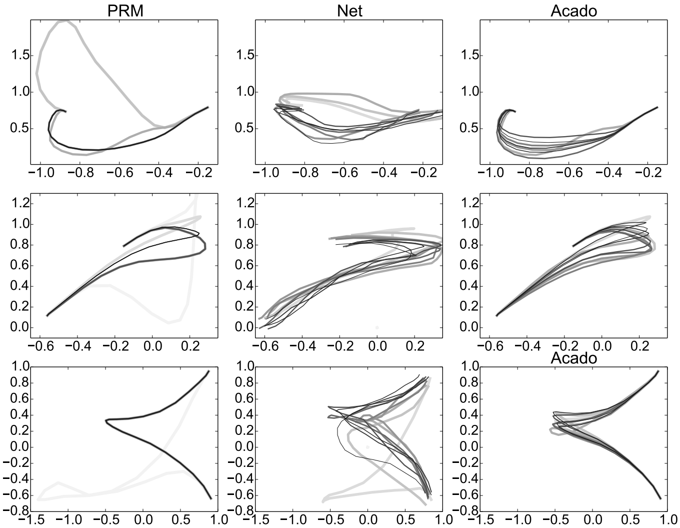
Value function as a metric
Policy function as warm-start

Our approach

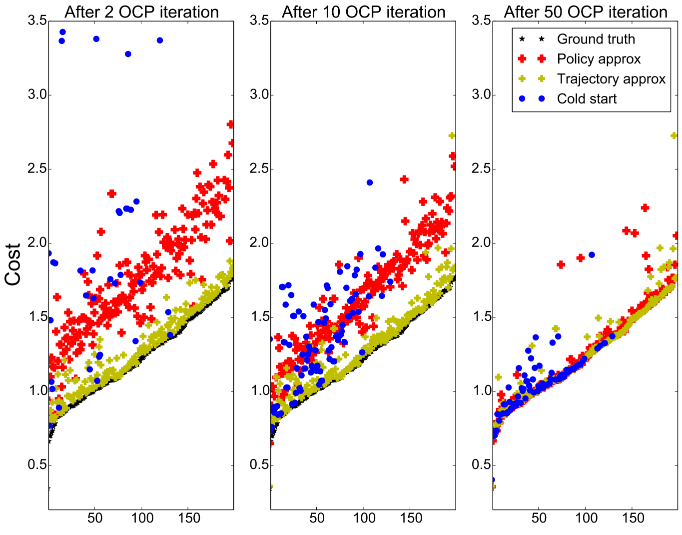


Experiments

Learned trajectories



Convergence predictive controller



Conclusions

Providing a good initial guess to warm start a predictive controller

- Sampled based kinodynamic planning
- Optimal control
- Function approximation
- Extension to higher-dimensional systems

Thanks for your attention

Questions ?