Motion Synthesis on Learned Skill Manifolds

Ioannis Havoutis, Subramanian Ramamoorthy

Motivation

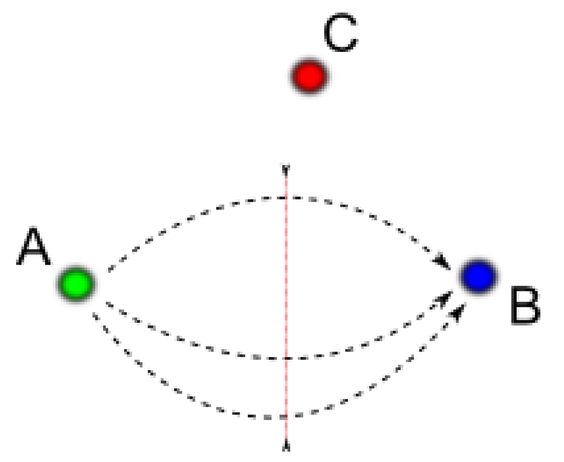
Humanoid robots are extremely flexible and complex platforms. We want them to be able to exhibit a variety of dynamical behaviours subject to:

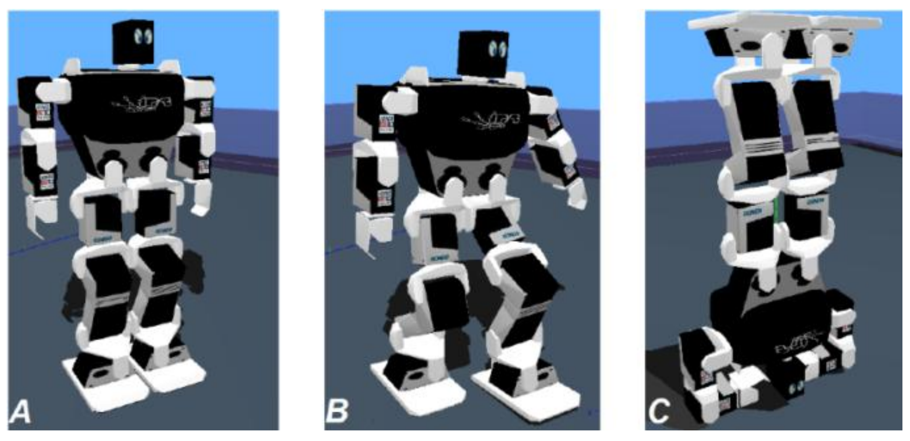
- Task constraints
  (Feasibility)
- Large disturbances
  (Reactive planning)

For this we need a flexible motion representation that would allow us to handle the complexity of the environment and the inherent complexity of the system.

How to represent *Skills* ?

- Families of paths in state space defined by system and task constraints.

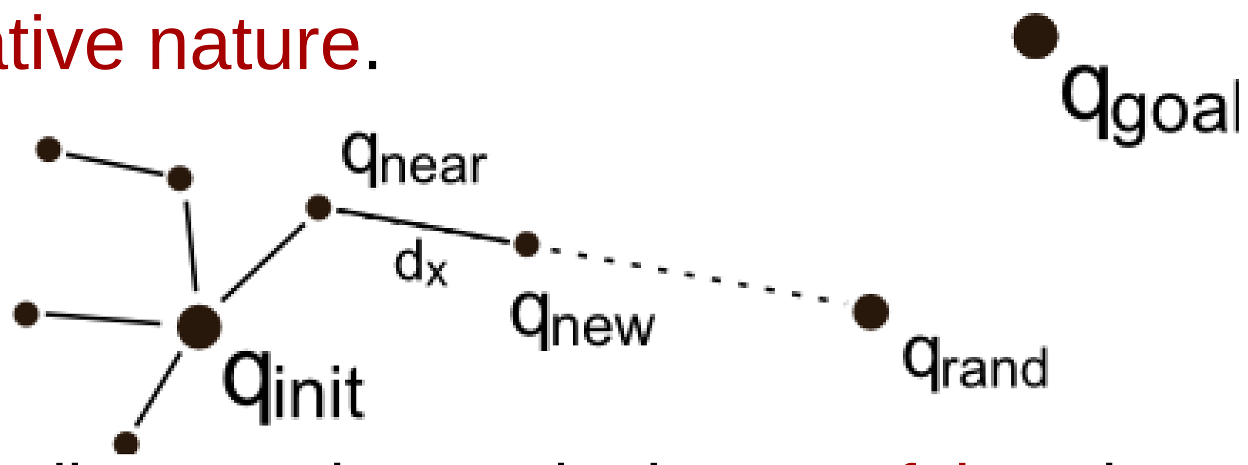




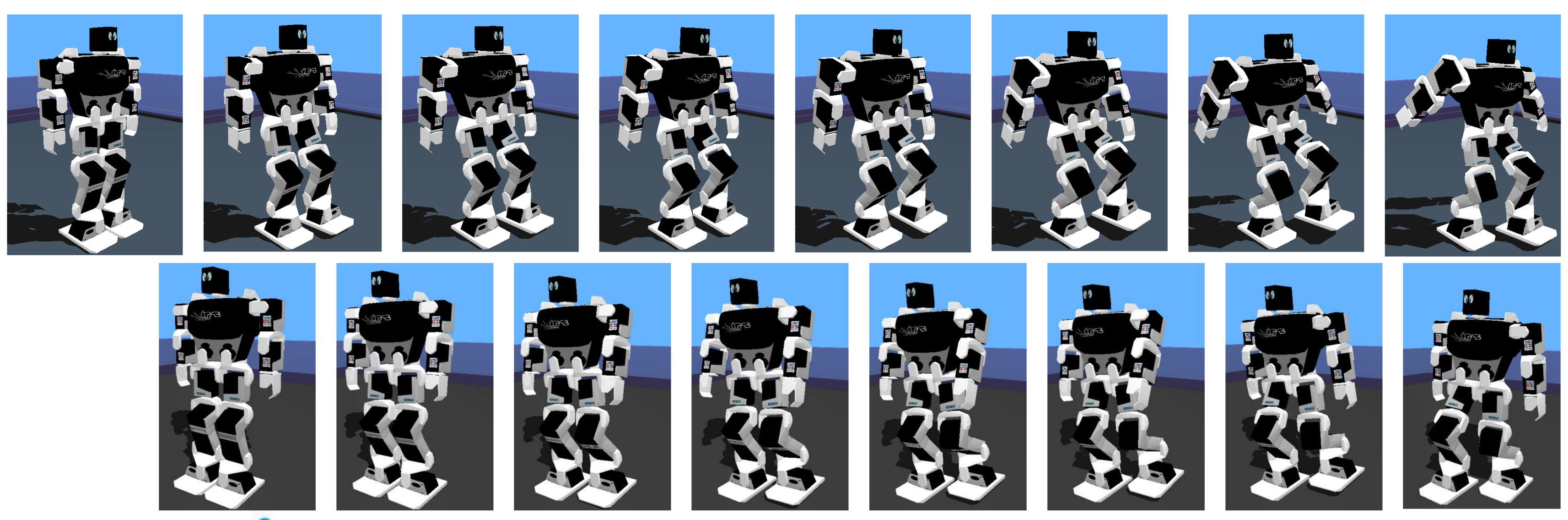
- Span an a priori unknown subspace of lower intrinsic dimensionality that we want to :
  - Capture from demonstration data
  - Leverage for motion planning

Sampling based motion planning

Rapidly-exploring random trees [1] is one of the most successful sampling based motion planning algorithms. Part of its success can be attributed to the computational simplicity and fast explorative nature.



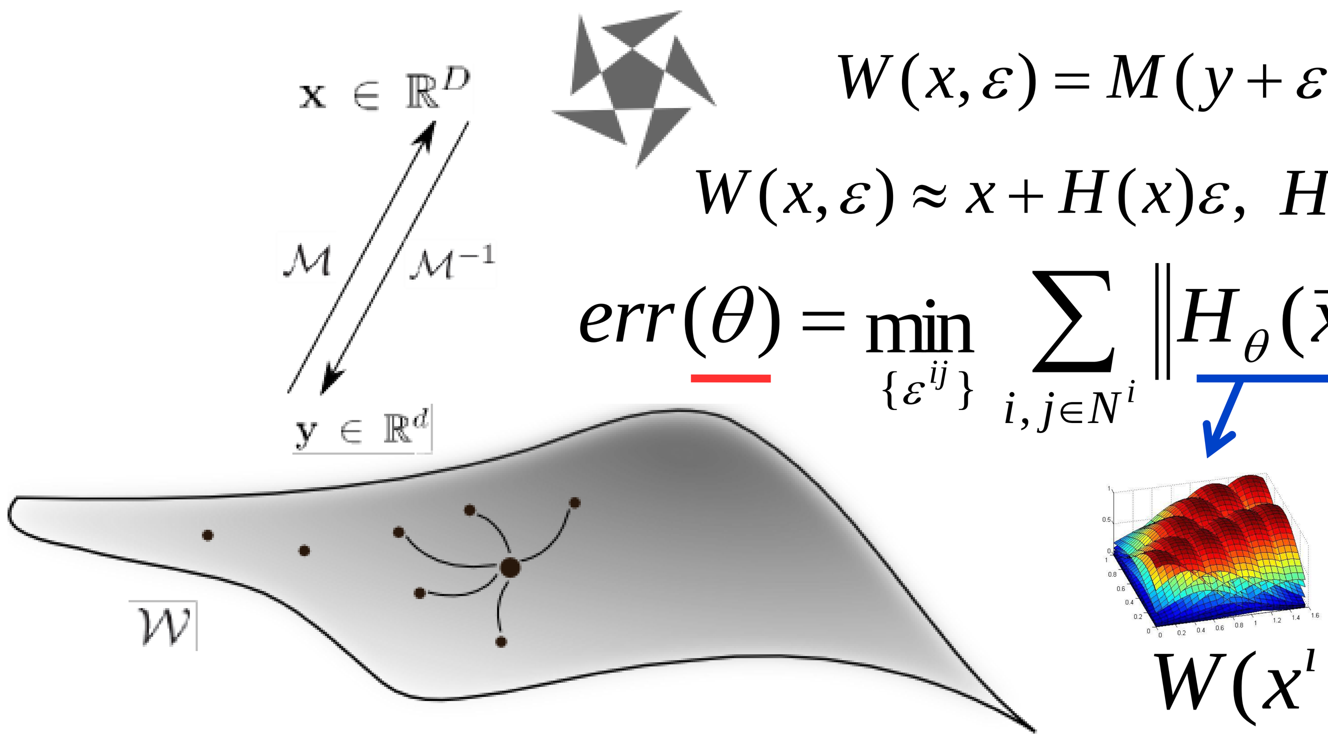
- Sampling gets increasingly wasteful as the dimensionality of the space increases, especially when high-dim paths lie on a structured subspace.
- Exploration does not take into account task constraints or demonstrated data.



From Data to Manifold

Demonstration data are drawn from an underlying skill manifold, learned using a manifold learning algorithm (Locally Smooth Manifold Learning [2]) . *LSML* allows for geometric operations as:

- Projection onto manifold
- Geodesic paths and distances



$$W(x,\varepsilon) = M(y+\varepsilon),\ y = M^{-1}(x)$$

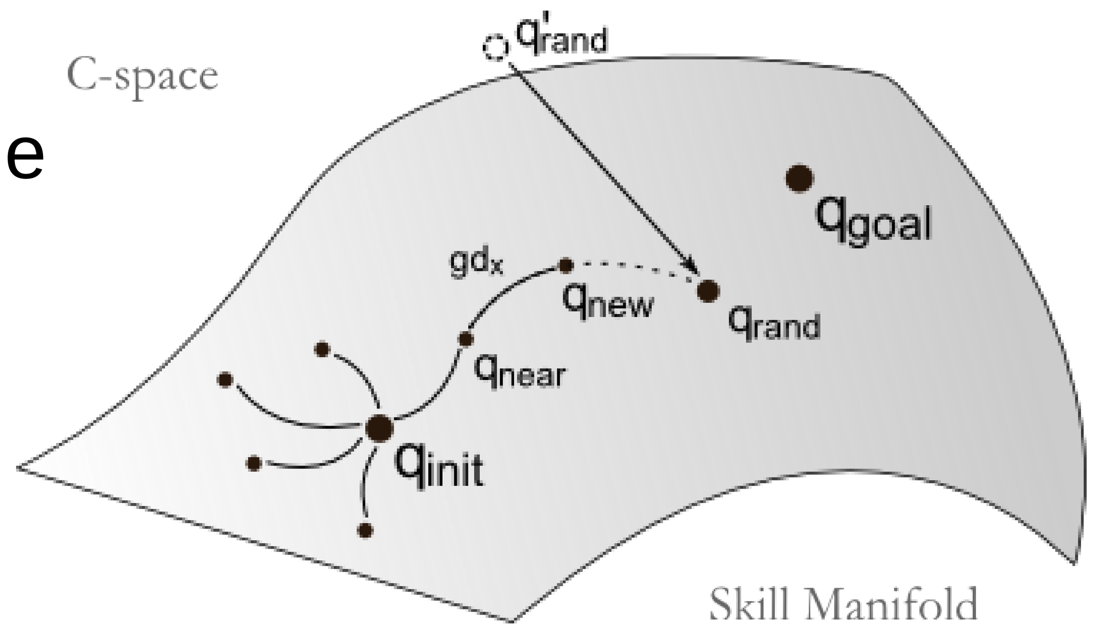
$$W(x,\varepsilon) \approx x + H(x)\varepsilon,\ H_\theta(\bar{x}^{ij})\varepsilon^{ij} \approx \Delta^i_{.j}$$

$$err(\theta) = \min_{\{\varepsilon^{ij}\}} \sum_{i,j \in N^i} \left\| H_\theta(\bar{x}^{ij})\varepsilon^{ij} - \Delta^i_{.j} \right\|_2^2$$

$$W(x^i)\varepsilon^{ij} = x^j$$

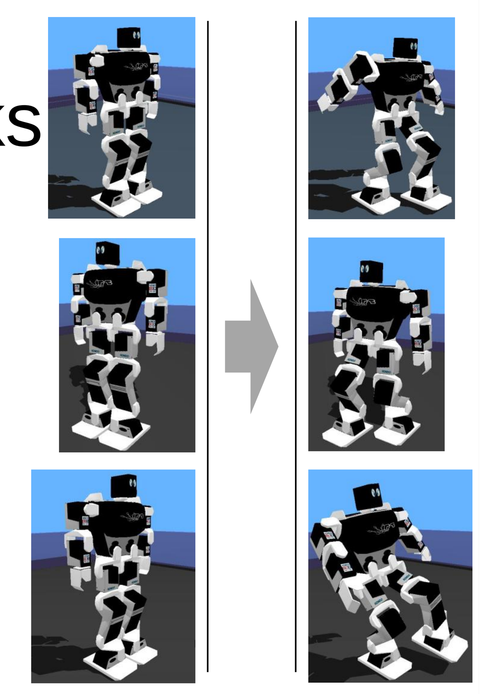
mRRT: Motion planning on manifolds

Leveraging *skill-relevant* knowledge in the form of a learned manifold into the planning process [3] we:



- Bias exploration towards *known good solutions*
- *Focus exploration* where it really matters

Results



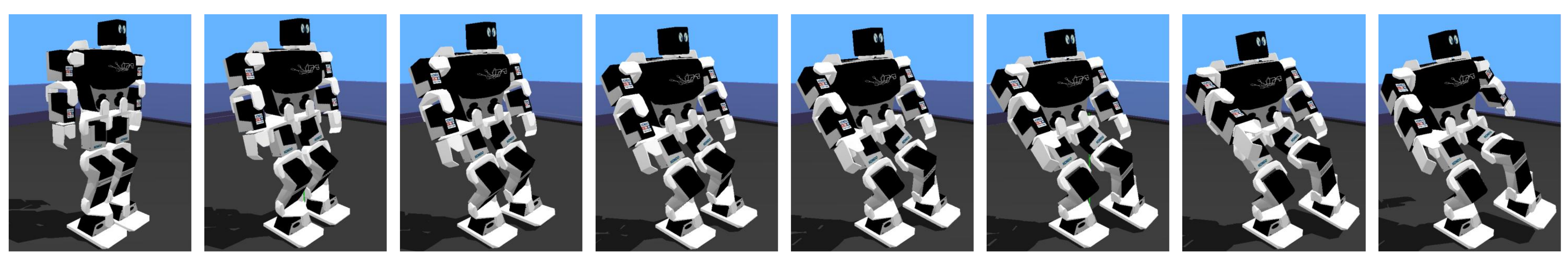
- Start from three demonstrated trajectories for each of the three tasks (step forward, kick, step backward)
- 10 trials for each task/algorithm
- No analytical model of the robot (evaluate samples in simulation)

Generalizing to unseen trajectories:

- 57.6% less invalid samples
- 25.2% decrease in overall planning steps

Why pay the computational cost of *mRRT*?

- Representation that captures *all feasible solutions*
- Enables *layered strategies*
- Basis for *Optimal Control* over manifold
- *Composition* of skill-manifolds



Task	*step*		*kick*		*backstep*	
	cRRT	***mRRT***	***cRRT***	***mRRT***	***cRRT***	***mRRT***
Average path length	40.9	38	52.5	49.4	47.2	37.5
Average number of samples	268.63	199.2	291	249.3	293.4	189.8
Average tree size	127.7	127	140.3	137.7	120.7	108.4
Average number of invalid samples	140.6	74.4	150.7	111.6	172.7	81.4
Smoothness {nRMSE}	0.0051	0.0049	0.0055	0.0041	0.0046	0.0043

References: [1] LaValle, S.M.: *Planning Algorithms*. Cambridge University Press, Cambridge, U.K. (2006)

[2] Dollar, P., Rabaud, V., Belongie, S.: *Learning to traverse image manifolds*. In: NIPS. (Dec. 2006)

[3] Havoutis, I., Ramamoorthy, S.: *Motion synthesis through randomized exploration on submanifolds in configuration space.* In Proc. RoboCup International Symposium, Lecture Notes in Artificial Intelligence, Springer Verlag, 2009

THE UNIVERSITY OF EDINBURGH