Modelling Dynamic Brachiation

Glen Berseth   Michiel van de Panne   Department of Computer Science, University of British Columbia

Problem

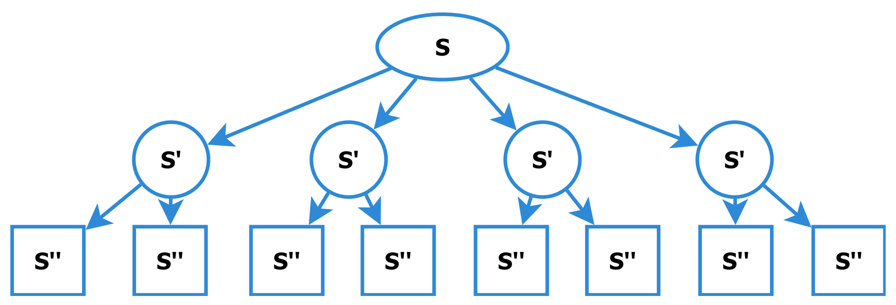
Learning dynamic navigation skills for Complex and Sparse Domains

- Brachiation is very dynamic and difficult
- Agent needs to learn navigation strategy in sparse domain
- Next target selection

Open Question

Is it better to do explicit or implicit motion planning?

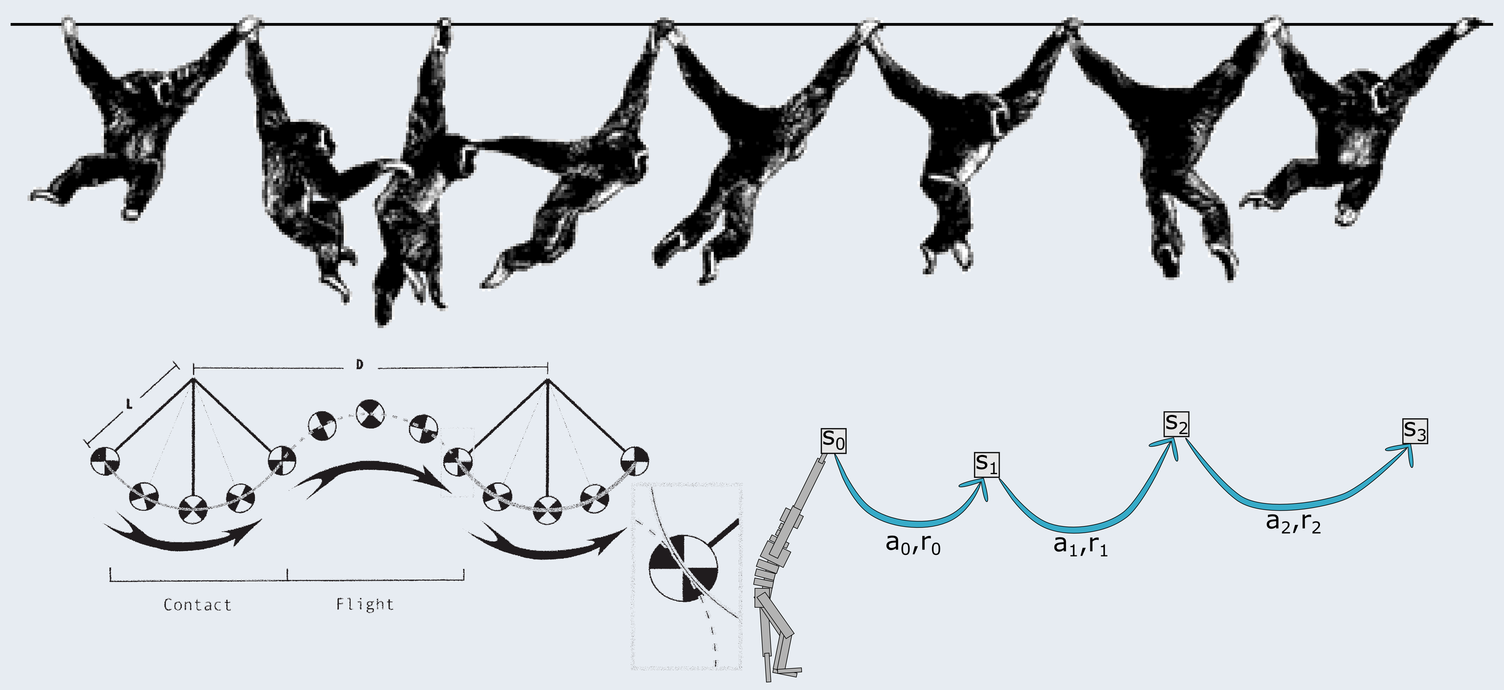
- Implicit Planning: Direct prediction based on current state
- Explicit Planning: Plan n steps ahead, using explicit successive state prediction
- Hybrid: Parts of both
- Use value function to compare states



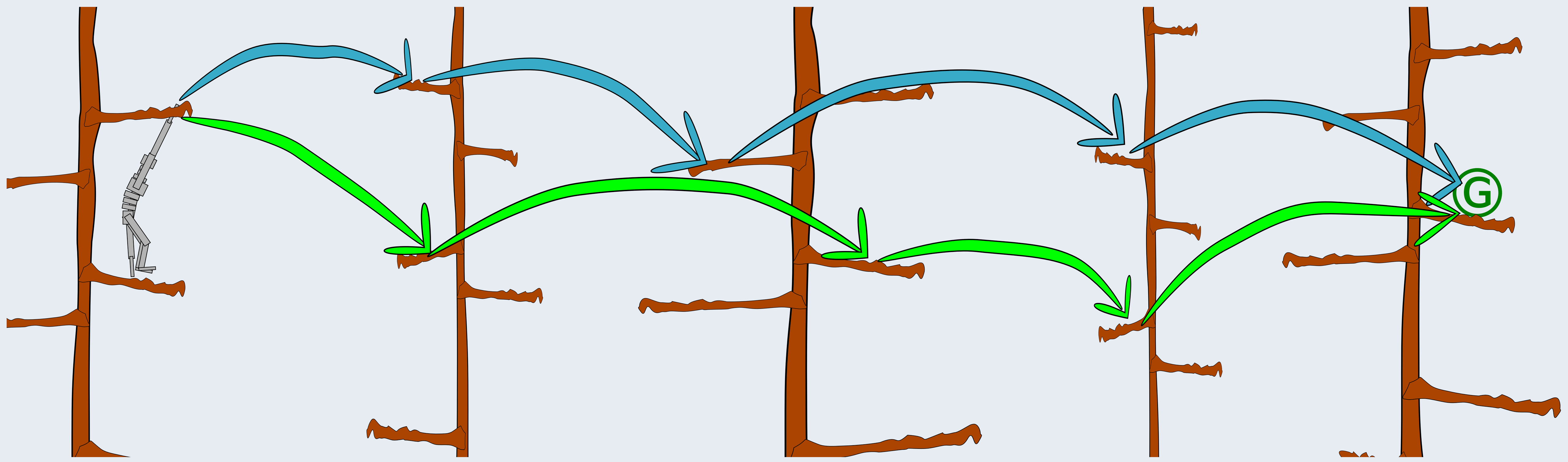
Brachiation

- Gibbons can leap great distances
- Pendulum-like dynamics
- Unlike terrestrial locomotion solution space is sparse
- Tight coupling between capabilities and affordances
- Action selection highly dependant on previous state

Motion Dynamics



Motion Planning



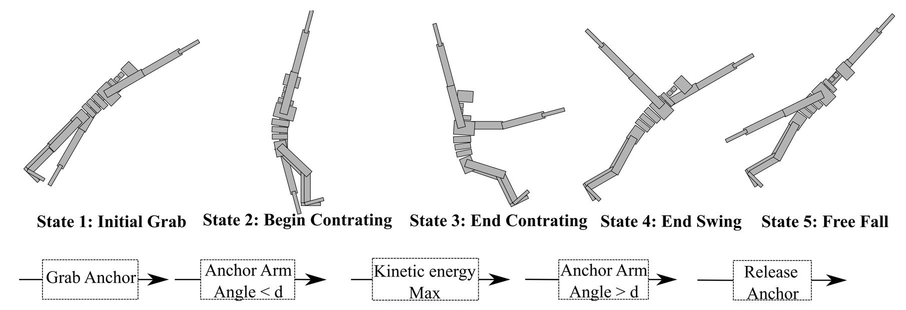
Implicit/Explicit/Hybrid

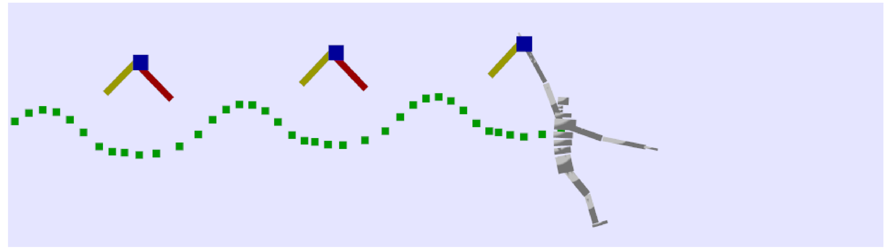
- Implicit:
  - Policy: $\mathbf{a} = \pi(\mathbf{s})$ and Value Function: $v(\mathbf{s})$
  - Forward dynamics are encoded in $\pi(\mathbf{s})$
- Explicit:
  - Forward Dynamics: $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$ and Heurisitic: $h(\mathbf{s})$
  - Explicitly plan from $\mathbf{s}_0$ to goal
- Hybrid:
  - Use Value Function $v(\mathbf{s})$ and Forward Dynamics $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$
  - Search action space to find high *value* actions

Planning Method

Input: Suggested action, $\mathbf{a}$
Input: Current state $\mathbf{s}$
Initilize: $v_{max} \leftarrow -\inf$
for $i \in (1, n)$ **do**
  $\delta \leftarrow \mathcal{N}(0, \sigma)$
  if $v(f(\mathbf{s}, \mathbf{a} + \delta)) > v_{max}$ **then**
    $\delta^* \leftarrow \delta, v_{max} \leftarrow v(f(\mathbf{s}, \mathbf{a} + \delta))$
return $\mathbf{a} + \delta^*$

Physics-Based Gibbon Simulation





Future Work

- Learn accurate forward dynamics function
- Determine varience/confidence in policy
- Improve Physics-based gibbon model
- Evaluate different sampling methods for action selection

References

[1] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[2] Xue Bin Peng, Glen Berseth, and Michiel van de Panne. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics (Proc. SIGGRAPH 2016)*, 35(5), 2016. to appear.

Glen Berseth   Michiel van de Panne
gberseth|van@cs.ubc.ca
Department of Computer Science, University of British Columbia