Algorithmic Gait Synthesis for a Snake Robot

[1]Columbia University, New York USA

Gagan Khandate[1], Emily Hannigan[1], Maximilian Haas-Heger[1], Bing Song[1], Ji Yin[1] and Matei Ciocarlie[1]

IEEE Robotics & Automation Society · IEEE · Columbia Engineering, The Fu Foundation School of Engineering and Applied Science

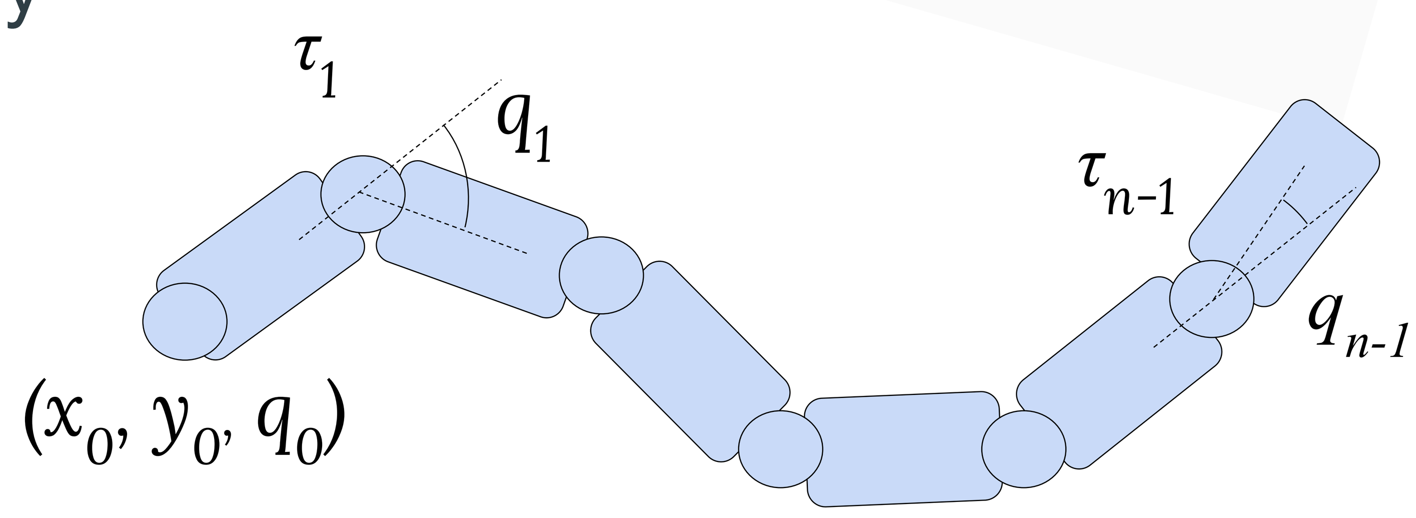
2019 IEEE International Conference on Robotics and Automation

May 20-24, 2019 Montreal, Canada

Abstract

- Snake robots are articulated and slender which allows them to reach confined spaces and makes them great candidates for amphibious locomotion
- Mimicking gaits found in nature has been effective for simple environments but little work has been on generating gait pattern algorithmically

Snake and Environment Force Models



- We consider snake robots both on land and in water with both low and moderate Reynolds number flow regimes

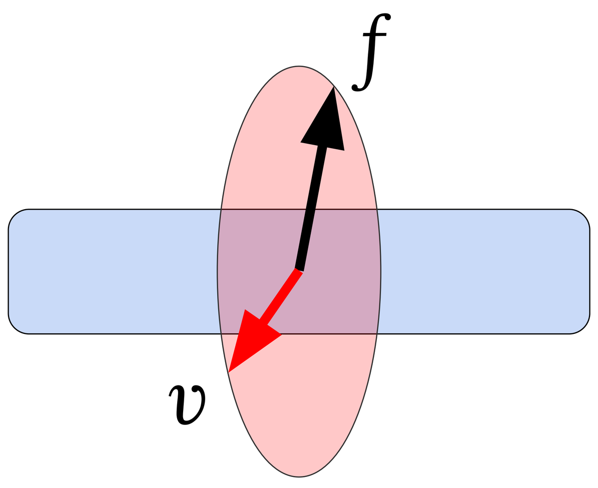


Figure 1 - Dry and Viscous friction on link center of mass

Dry Friction

$$\begin{pmatrix} f_n \\ f_t \end{pmatrix} = \begin{pmatrix} -\mu_n \sin(\arctan(\frac{\mu_n v_n}{\mu_t v_t})) \\ -\mu_t \cos(\arctan(\frac{\mu_n v_n}{\mu_t v_t})) \end{pmatrix}$$

Viscous Friction

$$\begin{pmatrix} f_n \\ f_t \end{pmatrix} = \begin{pmatrix} -\mu_n v_n \\ -\mu_t v_t \end{pmatrix} \quad \text{Low } Re$$

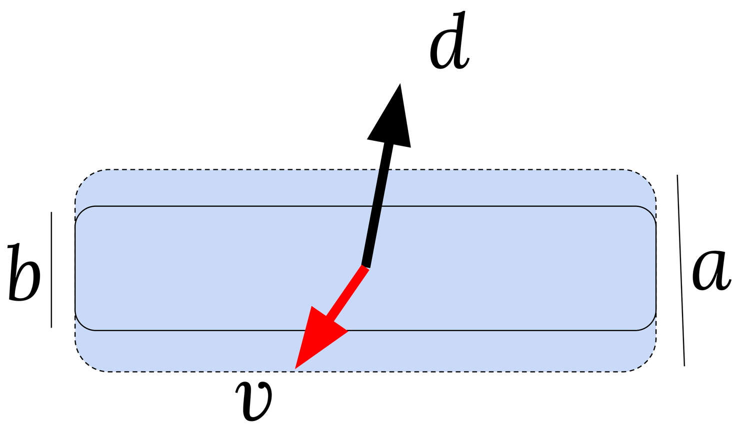


Figure 2 - Fluid dynamic drag on link center of mass

Fluid Dynamic Drag

$$\begin{pmatrix} d_n \\ d_t \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}\rho C_d a l |v_n| v_n \\ -\frac{1}{2}\rho \pi C_f \frac{a+b}{4} l |v_t| v_t \end{pmatrix}$$

$$m_a = \rho \pi C_a \frac{a^2}{4} l \quad \text{added mass}$$

Algorithmic Gait Synthesis

- The problem of gait synthesis can be viewed as a planning problem where we compute a sequence of robot actions which result in desired robot state
- We employ different classes of planning algorithms both offline and online - RRT (sampling based planning), iLQG (model predictive control) and Proximal Policy Optimization (model-free RL)

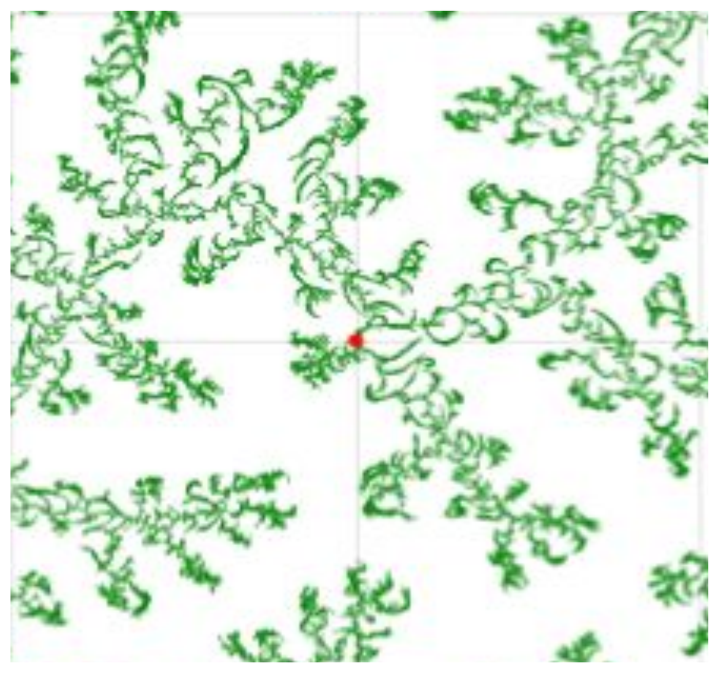


Figure 3 - Part of RRT tree for dry friction

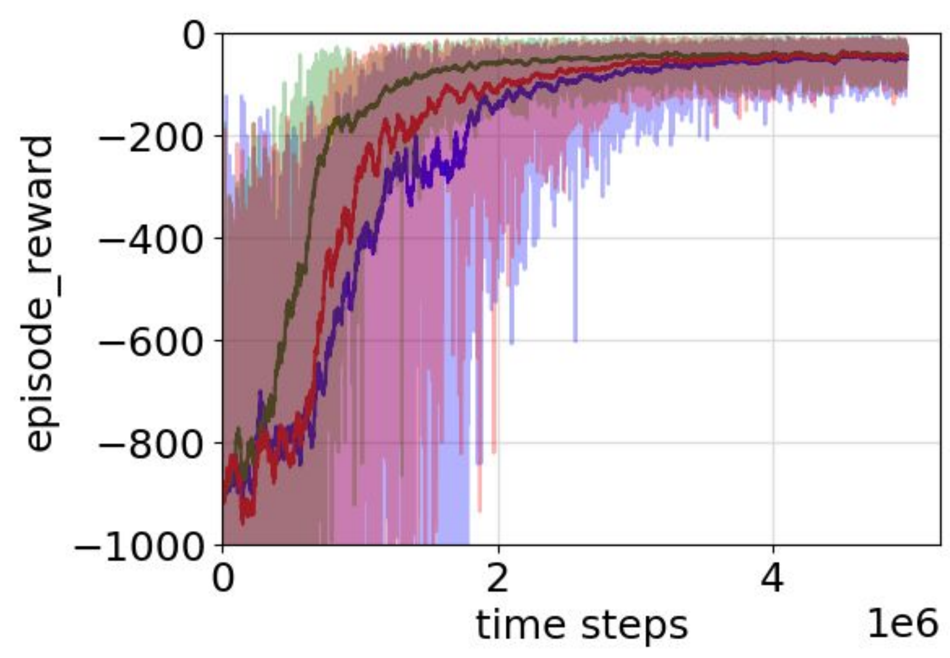


Figure 4 - Training neural network policy with PPO for dry friction

Results

- Kinodynamic RRT and PPO require offline computation (Fig. 3 & Fig. 4)
- While the trajectories obtained from RRT (Fig. 5 are inefficient, trajectories from iLQG and PPO with same cost/reward result in gaits which closely resemble those found in nature (Fig. 6)
- The performance of the three planning algorithms in terms of time and energy is compared (Fig.7)

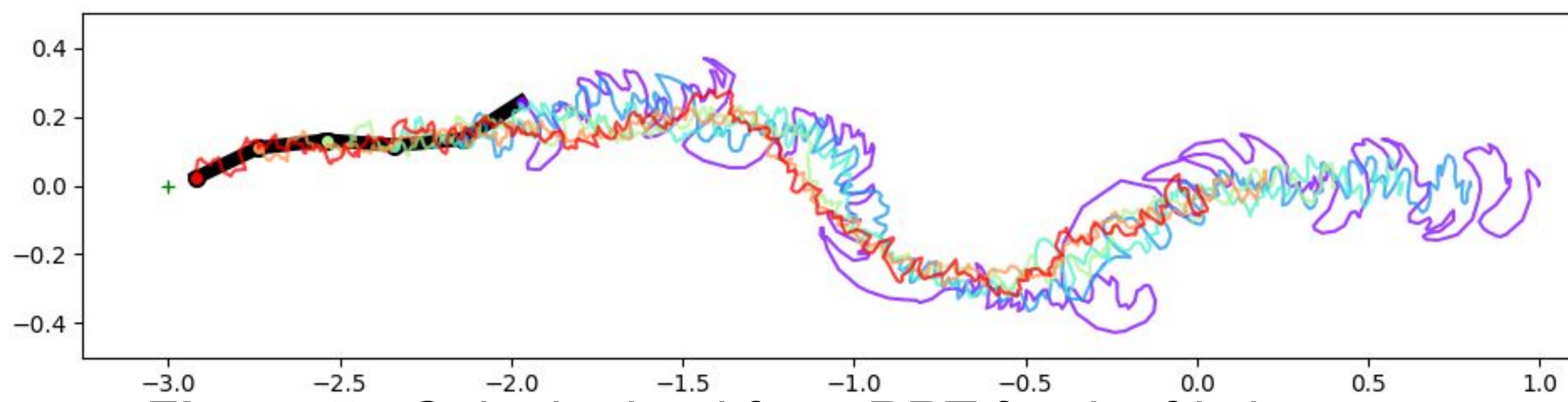


Figure 5 - Gait obtained from RRT for dry friction

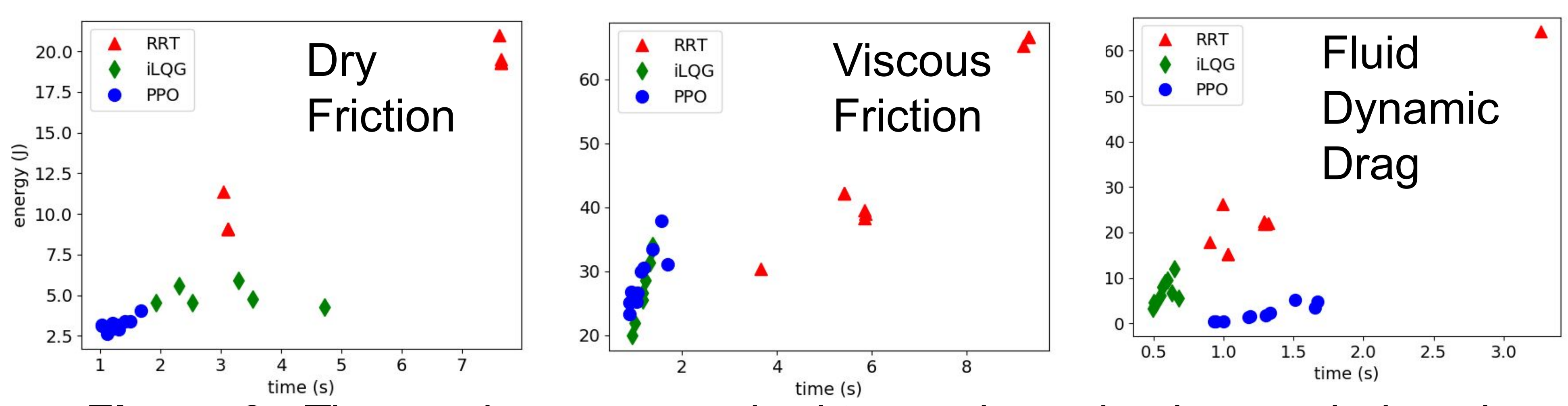
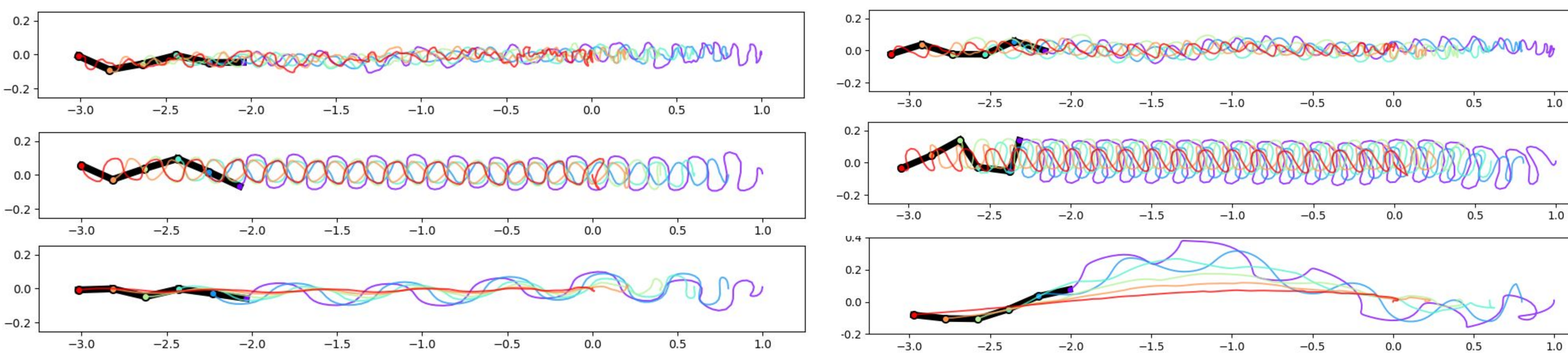


Figure 6 - Time and energy required to reach randomly sampled goals



(a) iLQG (b) PPO

Figure 7 - Gaits obtained from (a) iLQG (b) PPO for dry friction, viscous friction, fluid dynamic drag respectively