

Mid-Level Planning and Control for Articulated Locomoting Systems

Howie Choset, CMU

Magnus Edgerstedt, GATech

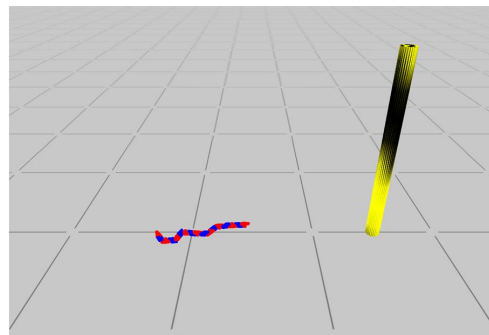
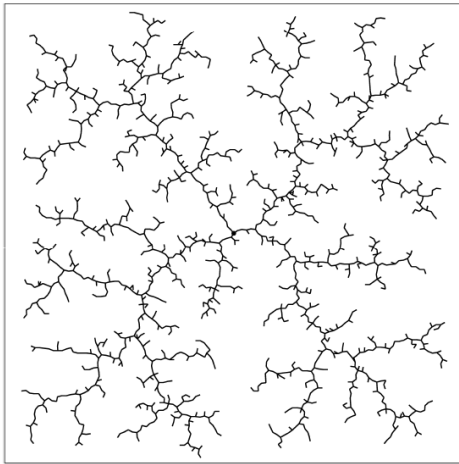
Innovative Claims

Innovative Claim	Novelty and Uniqueness
A computationally efficient mid-level planning framework that allows for high level reasoning while respecting low level considerations	Many robot architectures exist that claim to bridge the divide between high and low-level controls. Building on our prior work of controlling systems with open-loop gaits, we will separate external position variables from the many internal ones and exploit the mapping between the two to create the ideal separation for planning and control purposes.
Adaptive hybrid control of undulatory locomoting systems	Our work will allow for adaptive creation of new gaits as well as reparametrization of existing ones in response to changes in the environment. This will allow the planner to discover new gaits for undulatory systems as well as to operate in completely unknown spaces.
Multi-terrain snake robot locomotion	Virtually all controllers for snake robots are limited to executing one gait at a time, and therefore are limited to one or perhaps two terrains in a single run. Our hybrid controller will allow for seamless and robust composition of gaits thereby allowing snake robots to operate in multiterrain environments.

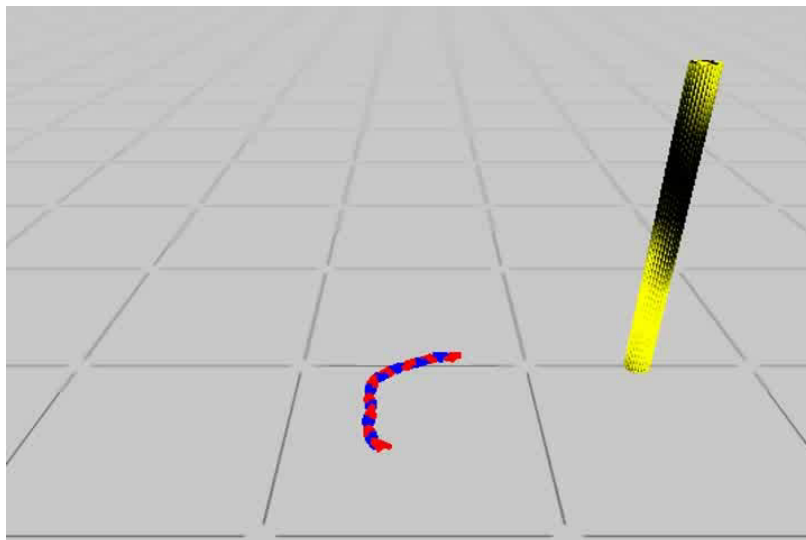
Innovative Claims

Innovative Claim	Specific Contributions
A computationally efficient mid-level planning framework that allows for high level reasoning while respecting low level considerations	Planning in the space of gaits (and bounding boxes) Virtual Chassis
Adaptive hybrid control of undulatory locomoting systems	Whole body gait transitions Progressive partial body gait transitions
Multi-terrain snake robot locomotion	Identified key challenge problems

Curse the Dimensionality



RRT with 15 joints – heading for the pole



Planning directly in joint-space is infeasible:

1. Produces “bad” paths because the set of “good” paths forms a low-dimensional subspace in solutions
2. Takes a VERY long time to find a path
3. Takes an inordinate amount of memory

Implementation Issue

1. Locomoting systems are underactuated, ie.,
2. Lack of metric
3. Lack of good local planner
4. Friction makes the trials non-repeatable

Gaits

$$\alpha(n, t) = \begin{cases} \beta_{\text{odd}} + A_{\text{odd}} \sin(\theta_{\text{odd}}) & \text{odd} \\ \beta_{\text{even}} + A_{\text{even}} \sin(\theta_{\text{even}} + \delta) & \text{even} \end{cases}$$

$$\theta_{\text{odd,even}} = (\Omega_{\text{odd,even}} n + \omega_{\text{odd,even}} t),$$

Benefit:

Reduce Dimensionality

Approach

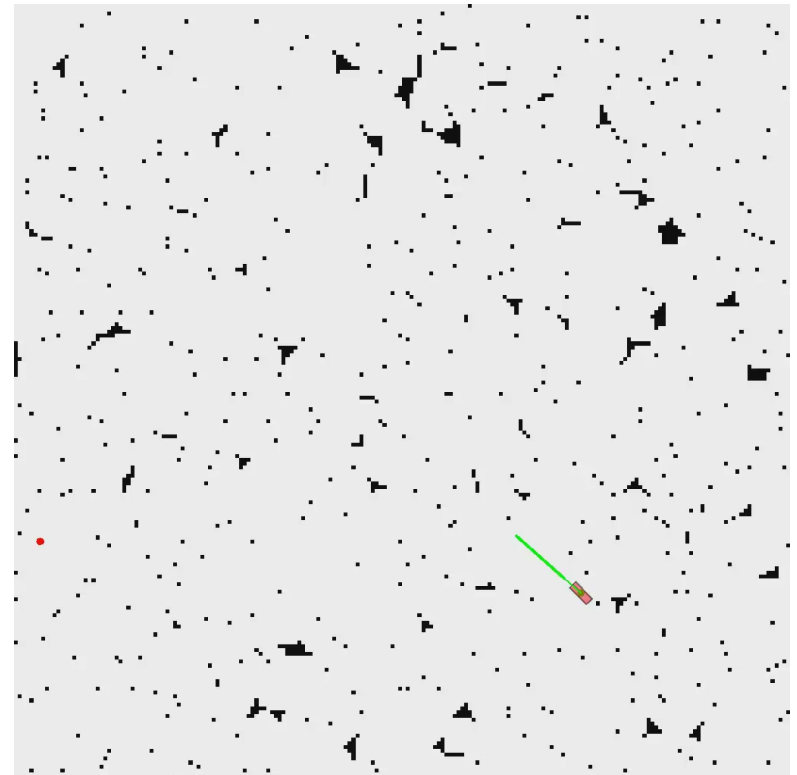
Search in the Space of Gaits



Toward Planning in Gait Space Snake Abstractions

- Separate position and internal shape variables
- Plan for position and almost defer shape variables until execution
- Chose position variables based on “best gait”
- Discrete Planner: sample position (rectangle), extend straight line, maximize efficiency and satisfy constraints

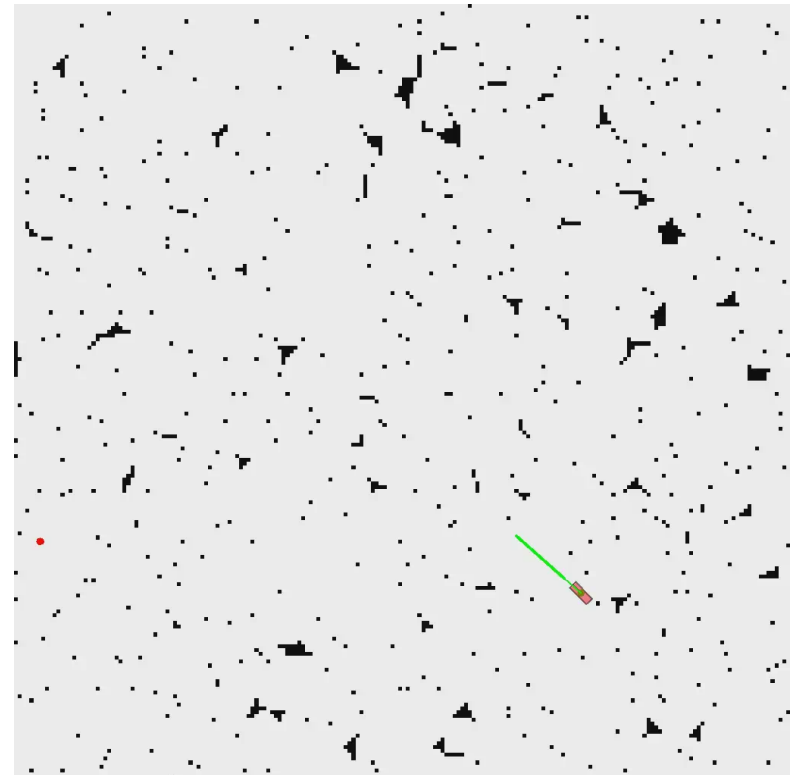
<u>Gait</u>	<u>Velocity</u>	<u>Power</u>	<u>Efficiency</u>
Slither (longitudinal)	2.8 cm/sec	13.1 W	130.2 mWh/m
Rolling (transverse)	26.4 cm/sec	28.7 W	30.2 mWh/m
Rotate	22.5 deg/sec	29.0 W	0.481 mWh/ deg
Transition between gaits (lasts 1 sec)	0 cm/sec	28.5 W	0 mWh/m (costs 3.65 mWh)



Toward Planning in Gait Space Snake Abstractions

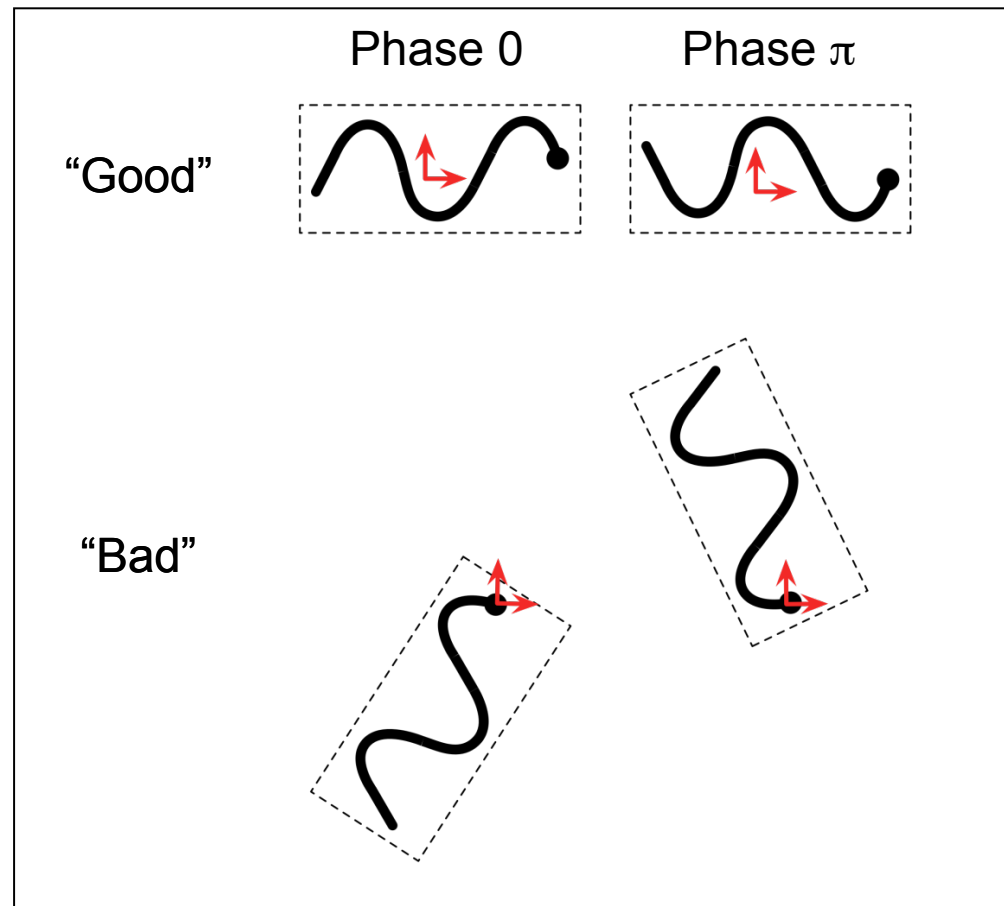
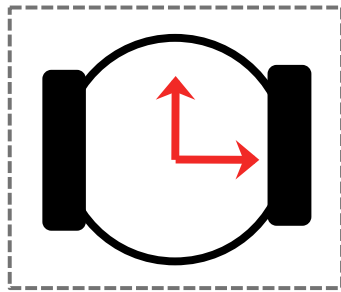
- Separate position and internal shape variables
- Plan for position and almost defer shape variables until execution
- Chose position variables based on “best gait”
- Discrete Planner: **sample position** (rectangle), **extend straight line**, maximize efficiency and satisfy constraints

<u>Gait</u>	<u>Velocity</u>	<u>Power</u>	<u>Efficiency</u>
Slither (longitudinal)	2.8 cm/sec	13.1 W	130.2 mWh/m
Rolling (transverse)	26.4 cm/sec	28.7 W	30.2 mWh/m
Rotate	22.5 deg/sec	29.0 W	0.481 mWh/deg
Transition between gaits (lasts 1 sec)	0 cm/sec	28.5 W	0 mWh/m (costs 3.65 mWh)



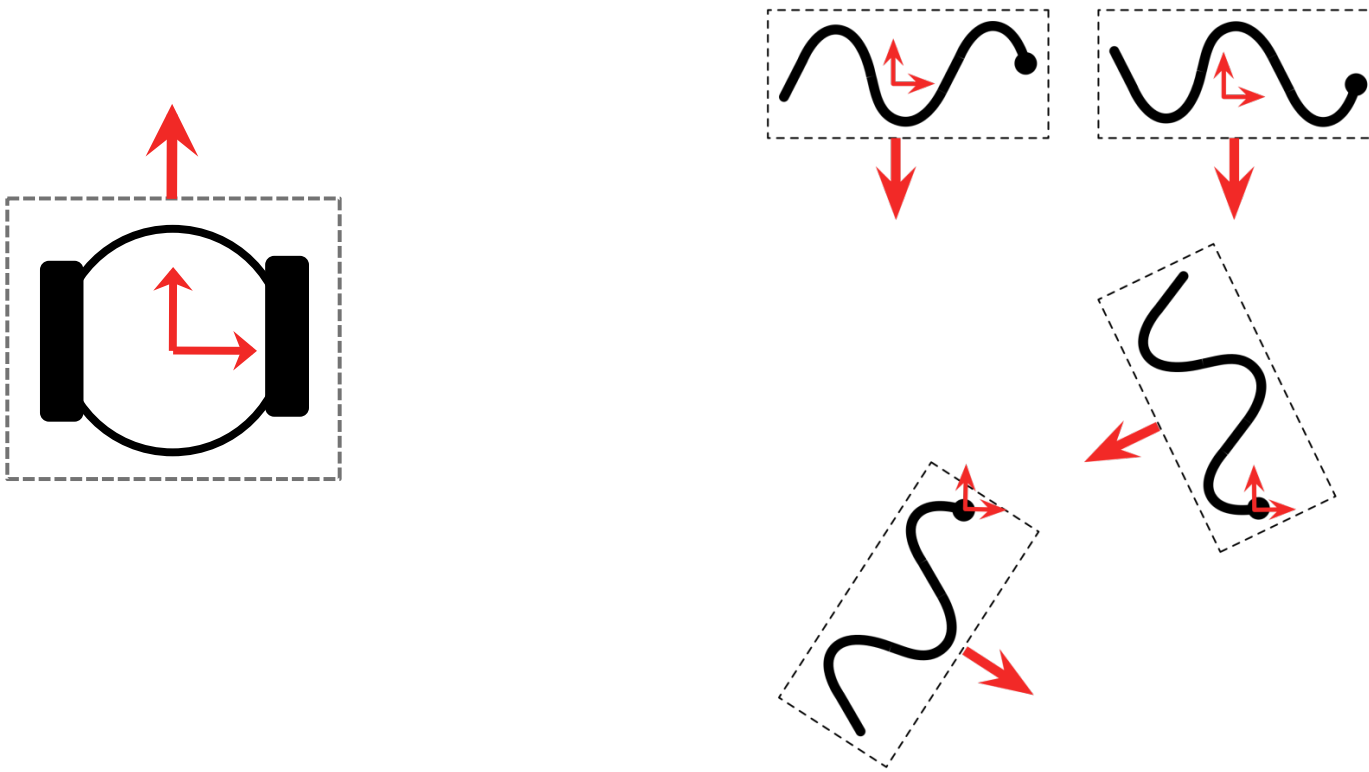
Representing Pose

Relationship between the bounding box and body frame must be independent of shape variables



Representing Motion

Direction of motion in the body frame to be independent of the phase within a gait.



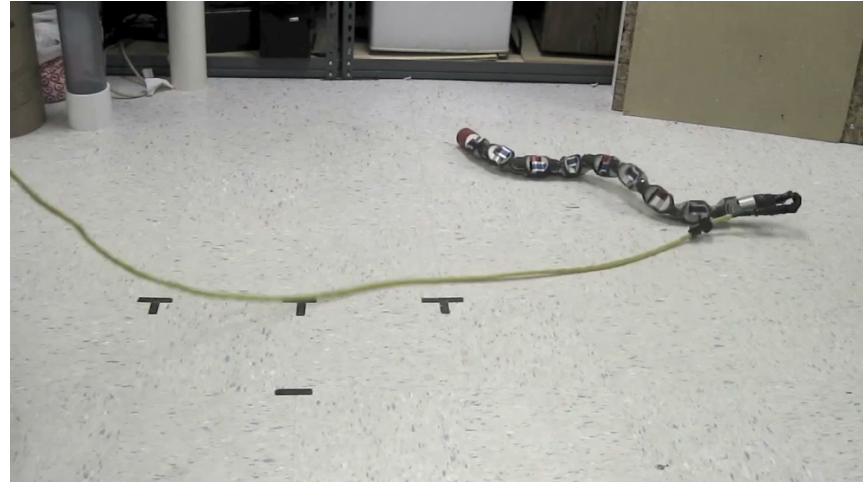
Towards Simple Motion Models

- With wheeled robots we have a basic no-slip / no-slide model that gives an intuitive relationship between rotation of the wheels and motion of the chassis.
- We would like work with snake robots in much the same way as we already do with wheeled robots.
 - Pick directions and drive those directions.
 - Do not need to move an integer number of cycles.

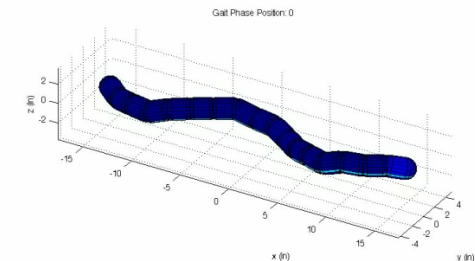
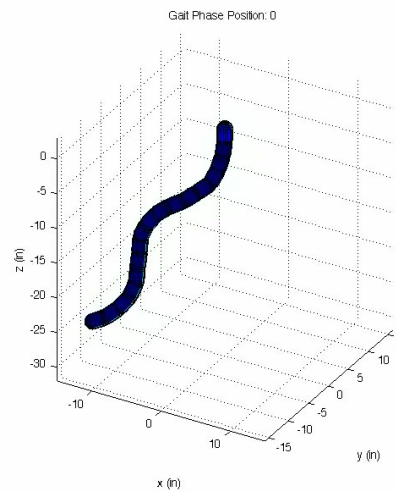


Virtual Chassis

Choice of body frame impacts representation of snake's motion in position



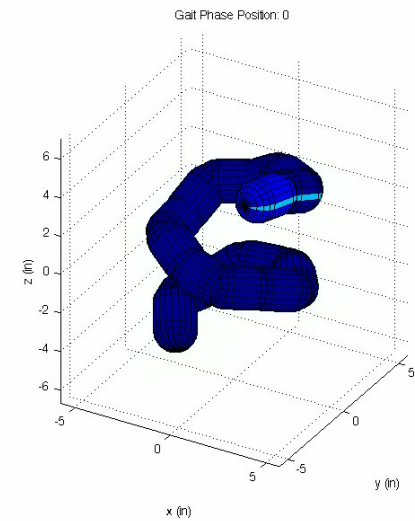
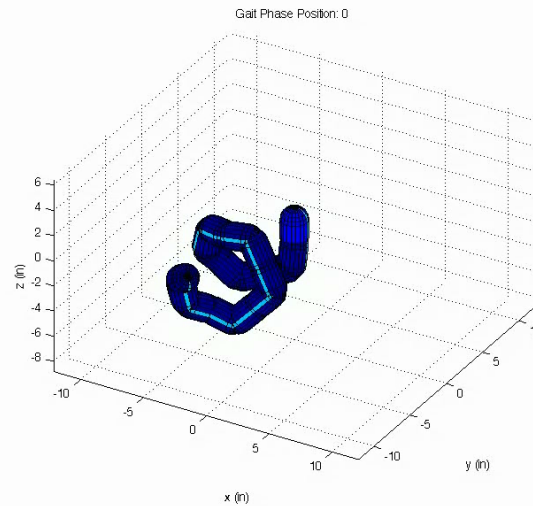
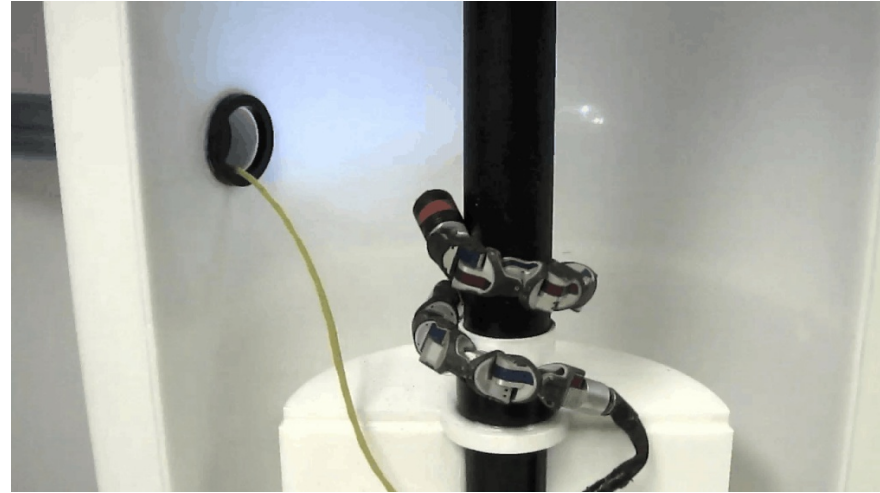
- Fixed to a link is bad (but easy to define)
- Based on shape is good (but more calculations)



Virtual Chassis

Define a body frame that depends on the overall shape of the robot.

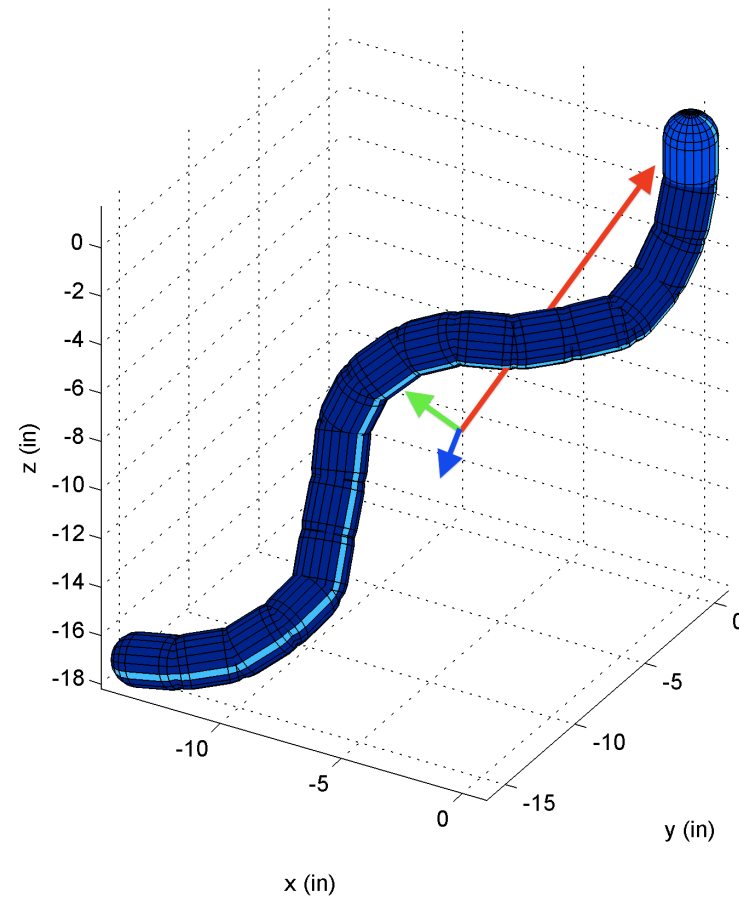
- *General:* Align with principal shape components via SVD.
- *Pipes:* Exploit symmetry to further align the frame with the centerline of the robot's shape.



Virtual Chassis

Benefits for Planning

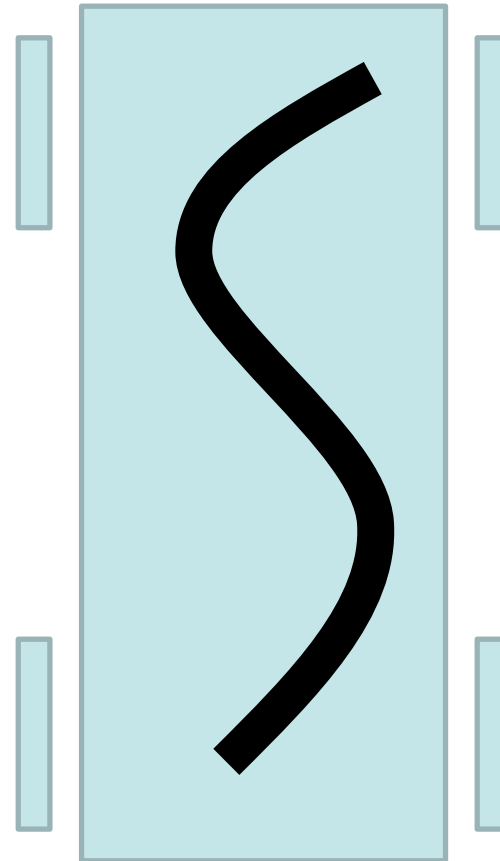
- “Straighter” paths in the world while executing gaits.
- Direction of displacement is independent of starting phase in the gait.



Why a snake?

How is this better than a wheeled robot?

- More options for moving than a car has
- The bounding box can change
- Plan for snakes on other terrains, such as climbing poles and crevices



Rich Snake Planning

What we need to do to employ benefits?

- More options for moving than a car has

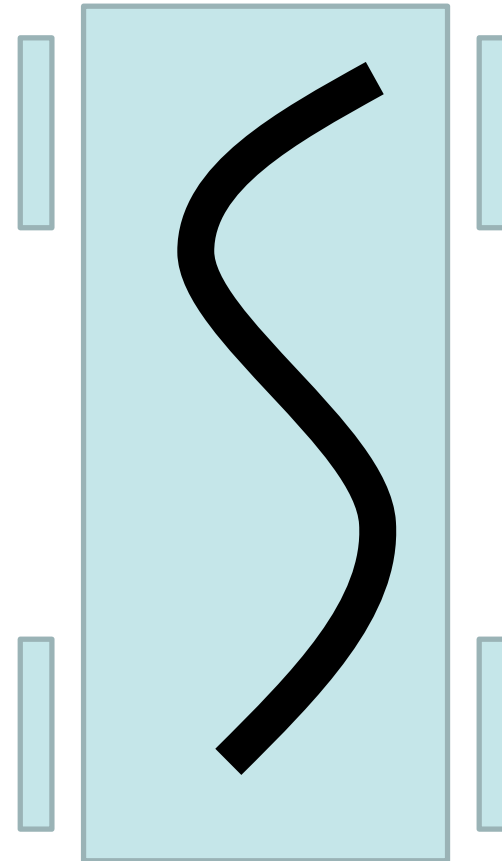
Gait libraries

- The bounding box can change

Shape space transitions: moving from one gait to another

- Plan for snakes on other terrains, such as climbing poles and crevices

Position space transitions: change in topology or constraints in position space



Transition



A trajectory in shape space that connects two gaits

Challenge: Obstacle Constraints,



Transition



A trajectory in shape space that connects two gaits

Challenge: Obstacle Constraints, fixturing, overtipping, situational awareness, sensors



Determine a need for a transition



Reach around pipe and maintain fix base



Maintain two fixed bases



Commit to letting go and beginning next gait

Transition



A trajectory in shape space that connects two gaits

Challenge: Obstacle Constraints, fixturing, overtipping, situational awareness, sensors

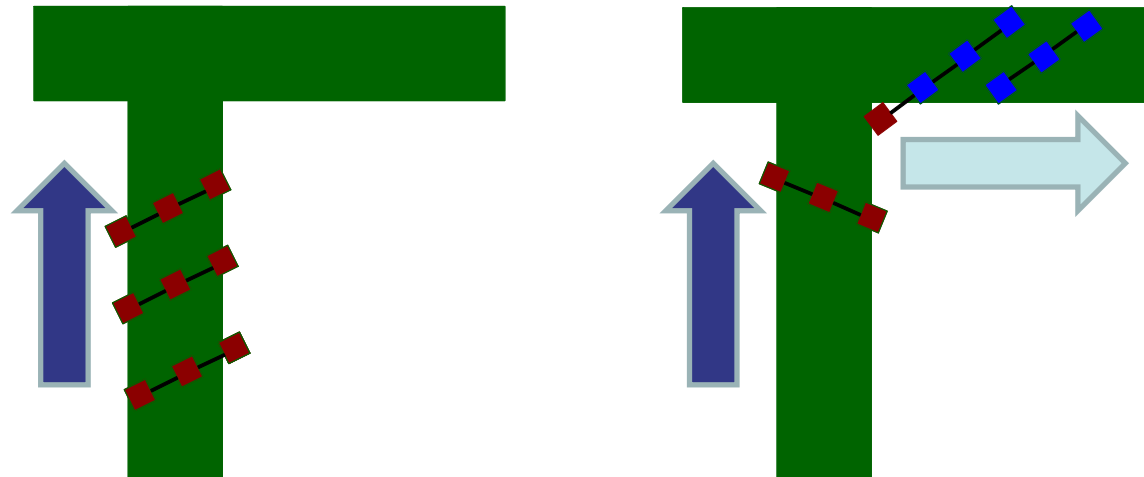
Intra vs. inter gait transitions



Commit to
lettering go and
begin next gait

Acyclic Gaits

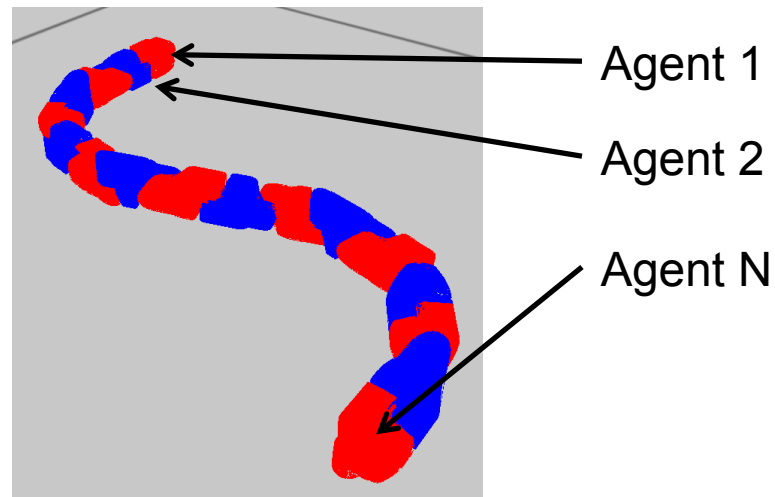
- Observation: Not all segments should be executing the same gait at the same time.



- Solution: Propagate the gaits through the snake
- Decentralized? Gait selection should depend only on adjacent segments
- Optimized? The best gaits should be selected given the current environment

Decentralized Gaits Strategies

- Each segment on the snake can be viewed as an agent in a multi-agent system



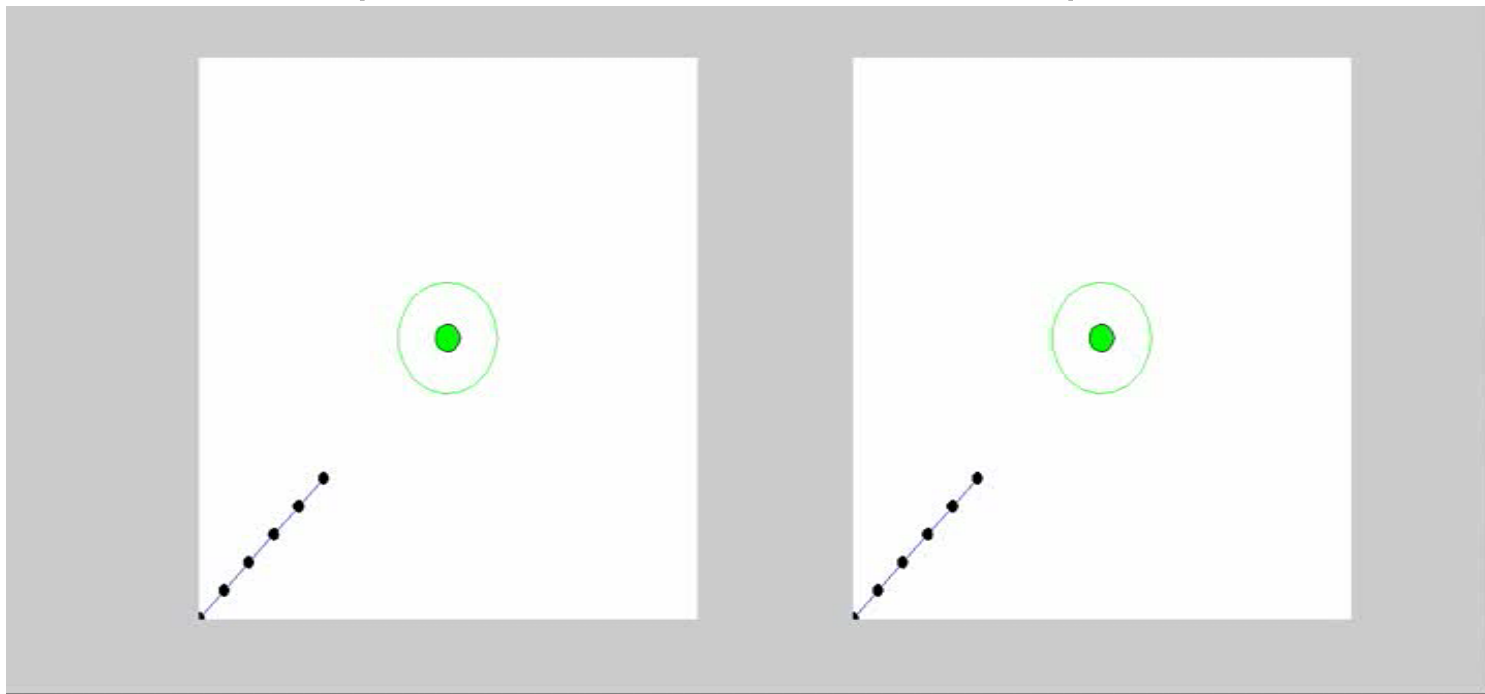
- How well a gait is performing is encoded through a cost function defined on each segment
- Dual Decomposition: each agent operates on local versions of the variables that it requires

Propagating Gaits

- Example: “Snake” orbiting an obstacle
- Only local information is used by each segment for determining when to transition from one “gait” to another.

Optimized

Unoptimized

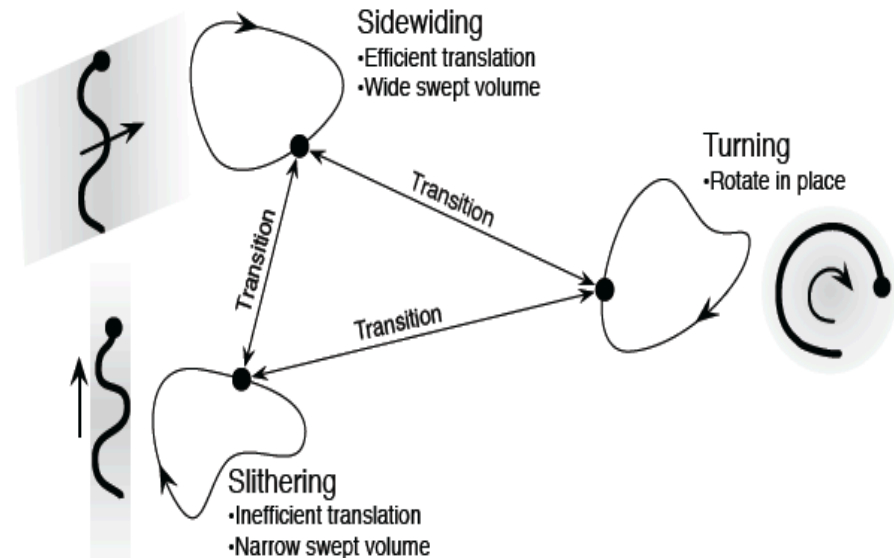


Planning in the Space of Gaits

A computationally efficient mid-level planning framework that allows for high level reasoning while respecting low level considerations

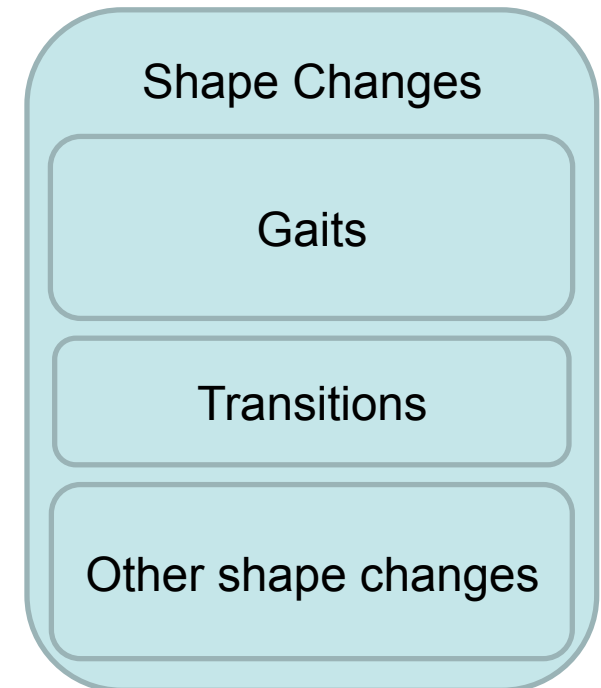
Gait Libraries

- Why is a gait library good?
- How should the gait library be formally structured? What information should be stored in gait tags and how is it used?
- How should a planner select and sequence gaits?



Shape Space Trajectories

- Gaits and transitions are specific cases of *shape changes*
 - *Gaits* are cyclic
 - *Transitions* move from a point on one gait to another
- Shape changes include special-purpose planned motions to:
 - Overcome obstacles
 - Move the camera to get a better view
 - Any motion of the snakes joints, for any purpose



Next Steps

- Implement gait library with (naïve) planner
- Transitions, Transitions, Transitions
- Contact sensor?
- Virtual chassis generality, biology?
- Characterize planners for gaits with contact in-mind
- Planning with both gaits and wiggles, characterize complexity