Autonomization of a Mobile Hexapedal Robot Using a Global Positioning System Module

Phillip Dupree

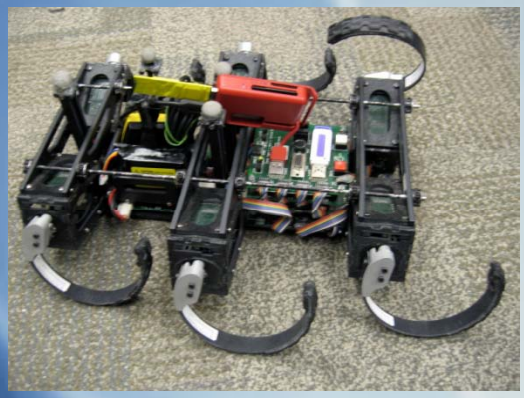
Advisors: Daniel E. Koditschek;

Galen Clark Haynes

Abstract:

- Background
- Linear Control System
- Simulation
- Utilizing GPS data
- Conclusions/Future Work

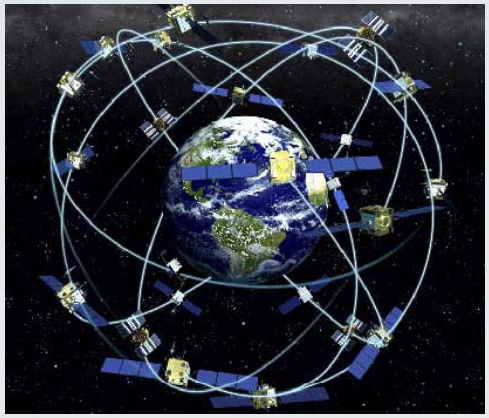
Background



Rhex:

- A highly mobile Hexapedal Robot.
- Utilizes C-legs and an Alternating Tripod Gait to walk.

Background



Navstar Global Positioning System:
Constellation of 24 satellites orbiting the earth every 12 hours at a height of about 10,900 nautical miles .

- Robot can “know” its current position.
- Follow a “breadcrumb path” of waypoints.
- *Only IF* it possesses an adequate control system.

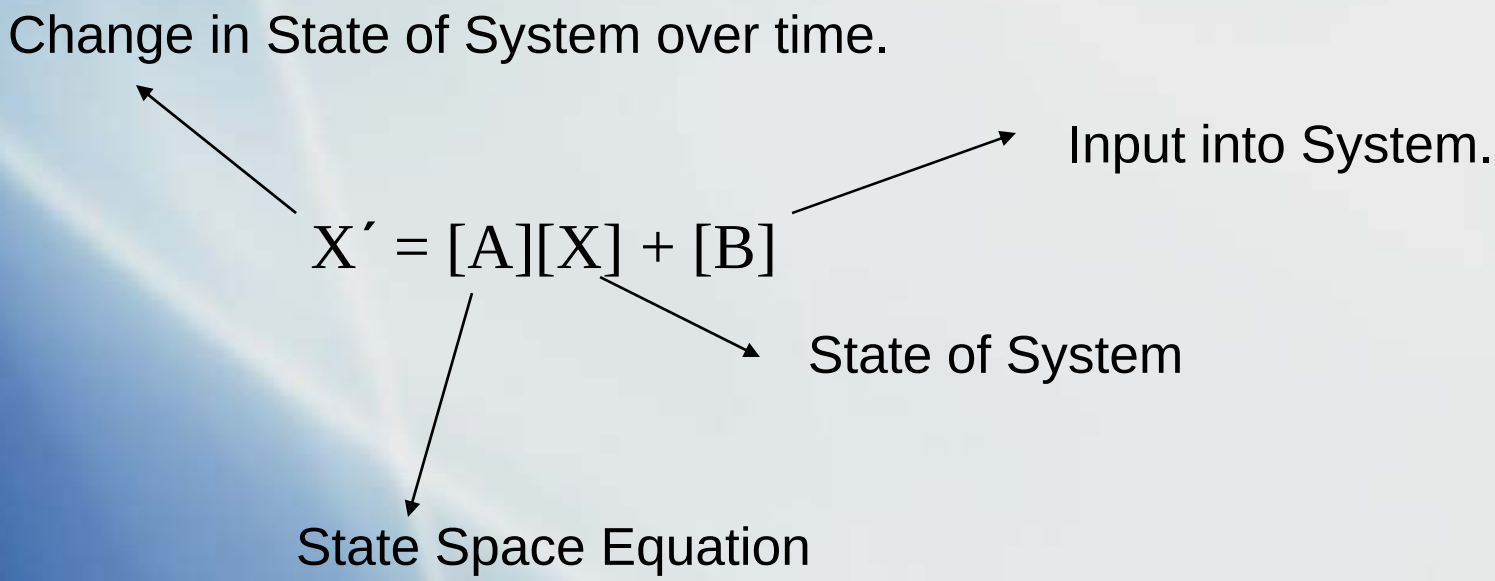
The Linear Control System:

A set of functions used to mathematically compute appropriate commands to move a robot to a desired position.

$$X' = [A][X] + [B]$$

The Linear Control System:

A set of functions used to mathematically compute appropriate commands to move a robot to a desired position.



Unicycle Control System:

- Speed is Constant
- Only *Orientation (angle of movement) is controlled.*

$$X' = [A][X] + [B]$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & \sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ s \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu \end{bmatrix}$$

Unicycle Control System:

- Speed is Constant
- Only *Orientation (angle of movement) is controlled.*

$$X' = [A][X] + [B]$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & \sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ s \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu \end{bmatrix}$$

Input into the system = $\theta' = k_p \cdot [\theta^* - \theta]$

Unicycle Control System:

- Speed is Constant
- Only *Orientation (angle of movement) is controlled.*

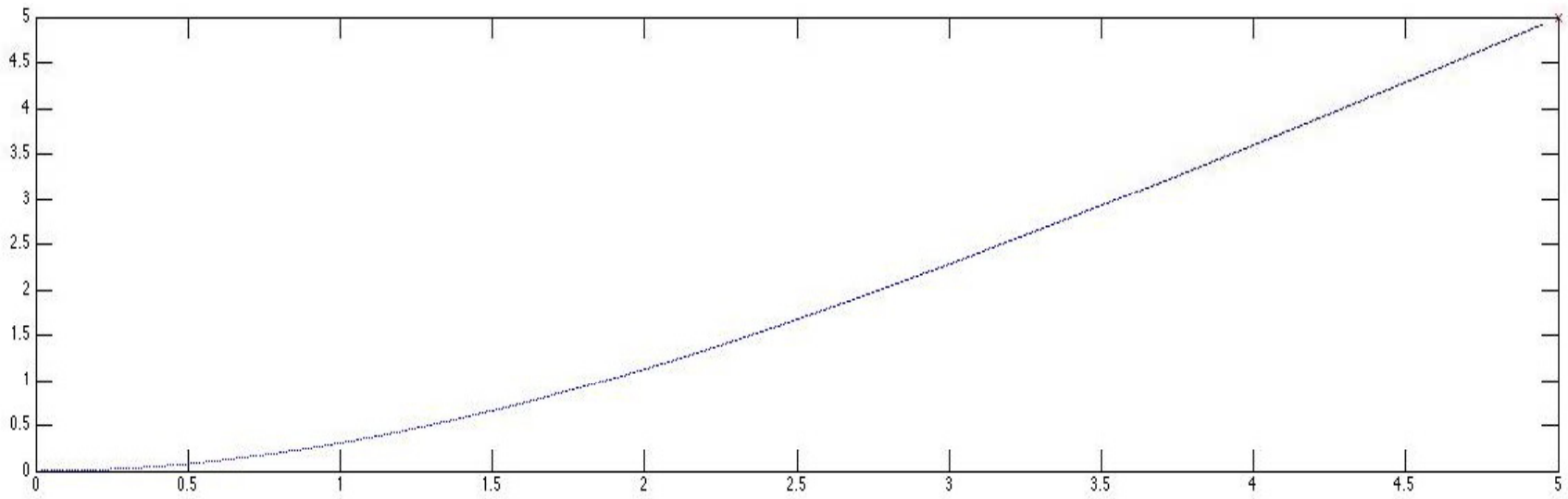
$$X' = [A][X] + [B]$$

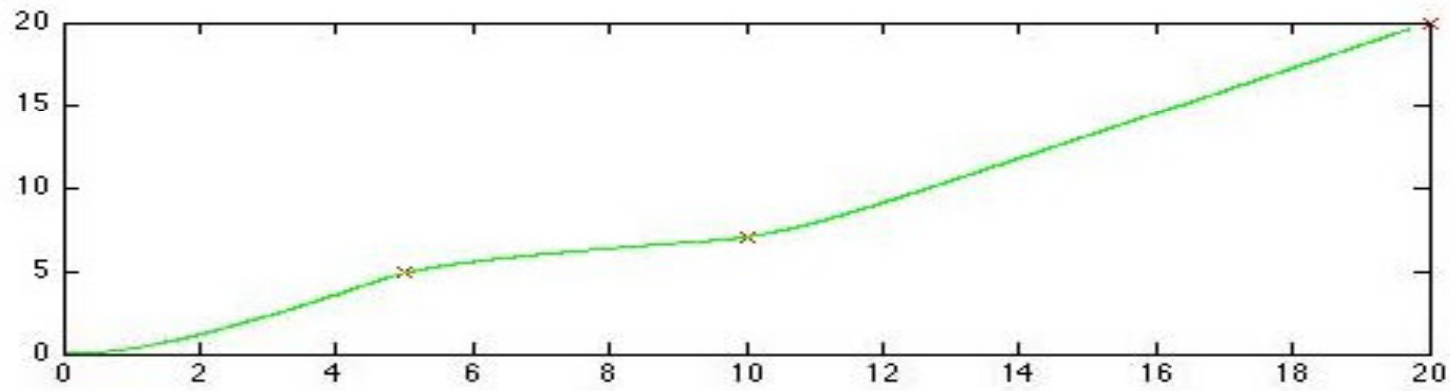
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & \sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ s \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu \end{bmatrix}$$

$$x' = s \cdot \cos(\theta)$$
$$y' = s \cdot \sin(\theta)$$
$$s' = 0$$
$$\theta' = kp \cdot [\theta^* - \theta]$$

Input into the system = $\theta' = k_p \cdot [\theta^* - \theta]$

Basic Controller:





Enhanced Controller:

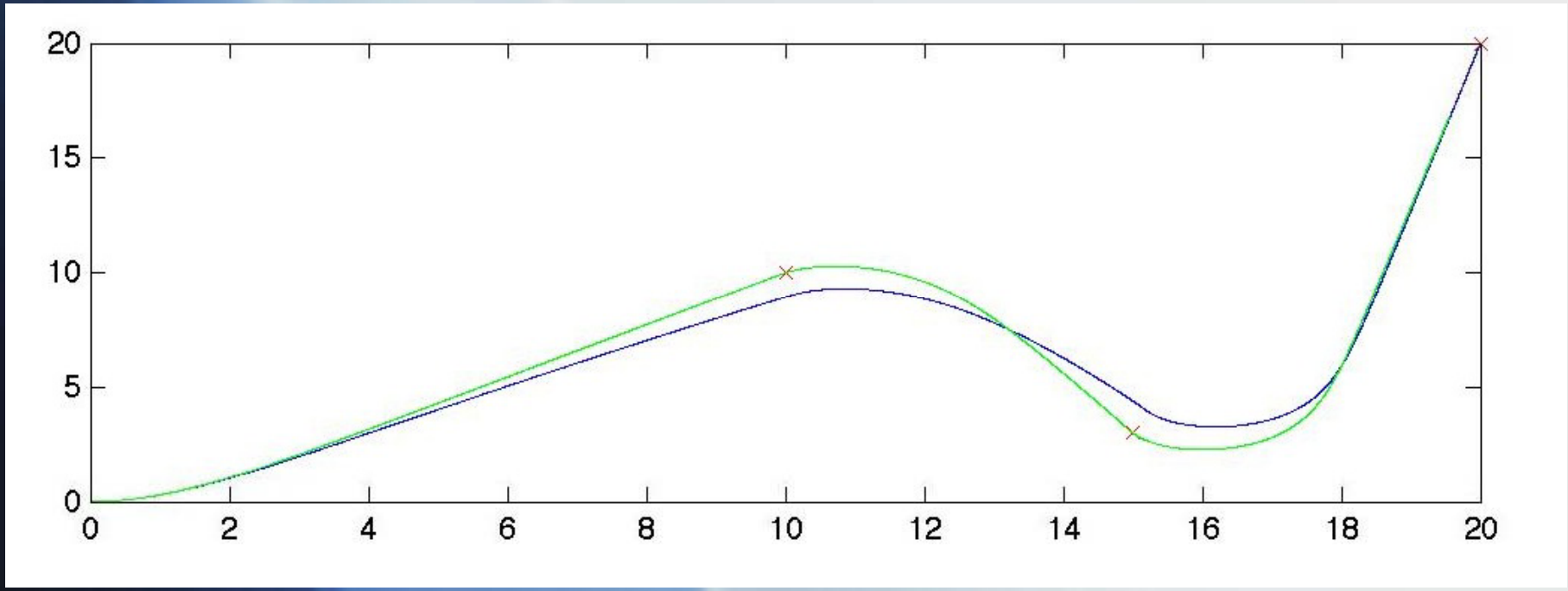
- "Rounds Corners" by taking input from both the nth destination point, and the (n + 1)th point.

$$\theta^r = k_p \cdot [\theta^* - \theta] + \frac{k_{p2}}{\| \mathbf{X}_1 - \mathbf{X} \|} \cdot [\theta^{*2} - \theta]$$

Enhanced Controller:

- "Rounds Corners" by taking input from both the n^{th} destination point, and the $(n + 1)^{th}$ point.

$$\theta^r = k_p \cdot [\theta^* - \theta] + \frac{k_{p2}}{\| X_1 - X \|} \cdot [\theta^{*2} - \theta]$$



Enhanced Controller:

- "Rounds Corners" by taking input from both the n^{th} destination point, and the $(n + 1)^{th}$ point.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & \sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ s \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ k_p \cdot [\theta^* - \theta] + \dfrac{k_{p2}}{\| X_1 - X \|} \cdot [\theta^{*2} - \theta] \end{bmatrix}$$

Utilizing the GPS Data:

- GPS gives data in spherical longitude and latitude coordinates.
- However, controller utilizes Cartesian coordinates.
- Must convert latitude/longitude to flat-earth approximate Cartesian coordinates.

Utilizing the GPS Data:

- GPS gives data in spherical longitude and latitude coordinates.
- However, controller utilizes Cartesian coordinates.
- Must convert latitude/longitude to flat-earth approximate Cartesian coordinates.

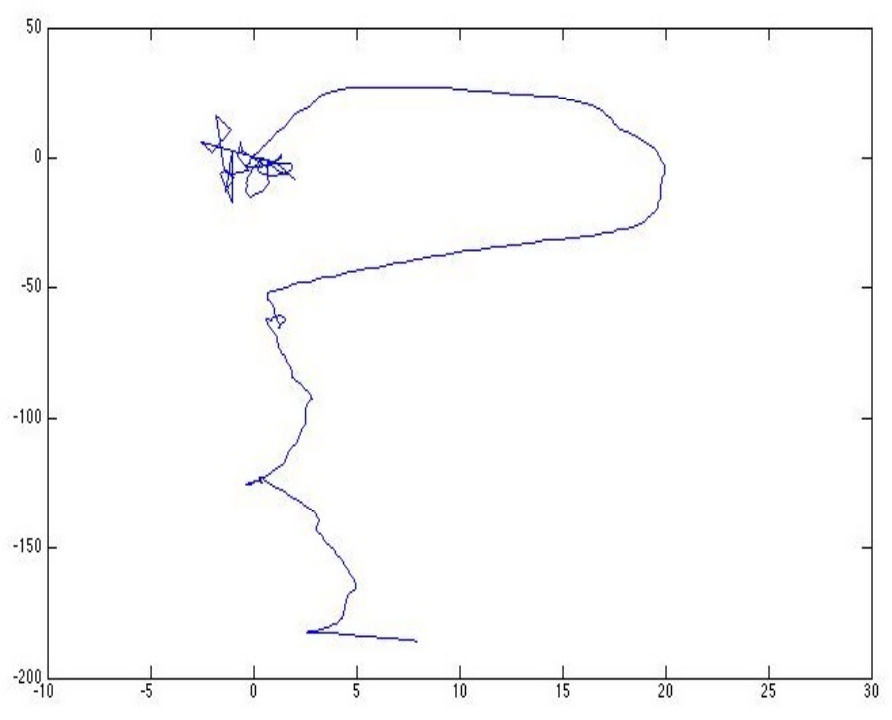
$$x = \cos(\phi) \cdot \frac{1}{\sqrt{\left(\frac{\sin(\phi)}{a}\right)^2 + \left(\frac{\cos(\phi)}{c}\right)^2}} \cdot \left[(longitude2 - longitude1) * \frac{\pi}{180}\right]$$

$$y = \frac{1}{\sqrt{\left(\frac{\sin(\phi)}{a}\right)^2 + \left(\frac{\cos(\phi)}{c}\right)^2}} \cdot \left[(latitude2 - latitude1) \cdot \frac{\pi}{180}\right]$$

$$\phi = \frac{\pi}{2} - \frac{(latitude1 + latitude2)}{2} \cdot \frac{\pi}{180}$$

Path shown approximately
On Google Earth.

Translated into
Cartesian coordinates.



Conclusions and Continuations

Code works, but not as well as it could.

- Current robot control works, but is quite basic.
- Upgrade current robot controllers to the sophistication level of the simulation controls.

Thanks for watching!

Special thanks to:

Professor Koditschek,

Galen Clark Haynes,

Aaron “ReallyTallGuy” Johnson,
Goren “Professor” Lynch,

Deniz “the Turk” Ilhan,

Mackenzie “ScaryNavyDude” White.