NYU POLYTECHNIC SCHOOL OF ENGINEERING

Design and Control of the BlueFoot Platform:

A Multi-terrain Quadruped Robot

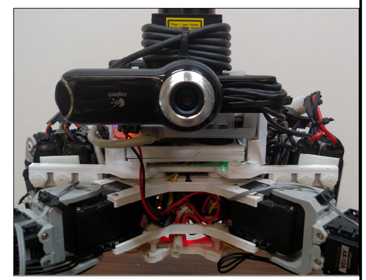
Brian Cairl

Advisor: Dr. Farshad Khorrami
5/13/2015

NYU POLYTECHNIC SCHOOL OF ENGINEERING — Overview

Overview of Presentation

Å Introduction
Å System Design and Software
Å Model of the BlueFoot Robot
Å Gait and Stability Control
Å Navigation Control
Å Concluding Remarks



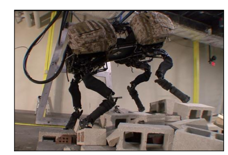
NYU POLYTECHNIC SCHOOL OF ENGINEERING

Introduction

NYU POLYTECHNIC SCHOOL OF ENGINEERING — Quadruped Robotics

Inspired by notable quadruped robotic systems from the past decade:

- LittleDog (Boston Dynamics)
- BigDog (Boston Dynamics)
- HyQ (Istituto Italiano di Tecnologia)
- Tekken (Kyoto Institute of Technology)
- Kolt (Stanford University)



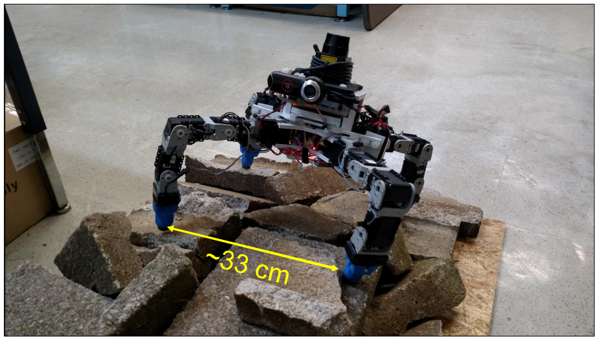
- **Relevant Research Areas:**
  - Gaiting design
  - Rough-terrain navigation/planning
- **Example Applications:**
  - Disaster recovery
  - Search and rescue
  - Environmental mapping

BlueFoot quadruped robot is designed to be:

- Small scale and compact
- Dexterous (16 actuated degrees of freedom)
  - Stabilize and reposition on variable terrain
  - Large range of trunk articulation
  - Ability to overcome raised/uneven terrain
- Capable from a computational and sensory standpoint

Particular elements achieved in design are as follows:

- 2.0+ kg of payload capacity
- 30+ battery life
- Multi-processor architecture
- Large range of sensory capabilities
  - IMU, Camera, LIDAR, GPS
  - Joint position, velocity, loading sensing (using smart servos)
  - Foot contact sensing



Overview of motion control strategies:

- Central Pattern Generator (CPG)-based gait control
- Zero-Moment Point (ZMP) body posture control
- Virtual-Force foothold controller
- NARX-Neural Network trunk-leveling controller

Overview of navigation control strategies:

- Potential-Fields/Visual-Servoing
- Surface Reconstruction for rough terrain navigation
  - Composing 3D point clouds from 2D LIDAR scans
  - Height-map Surface Representation
  - Surface (Normal) Estimation from 3D point clouds

System Design and Software

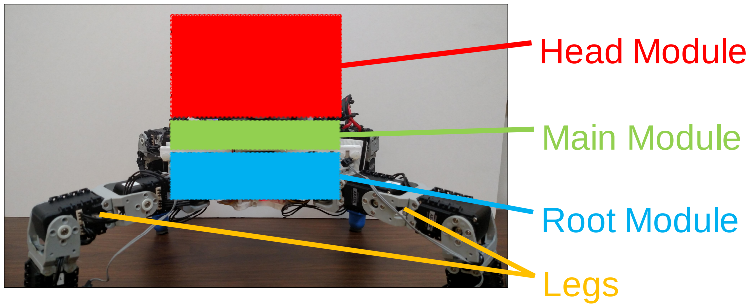
Modularity between body segments

- Robot designed with 3 main sections

3D printed structural components

- Lightweight
- Rapid design iterations
- Facilitates modular design

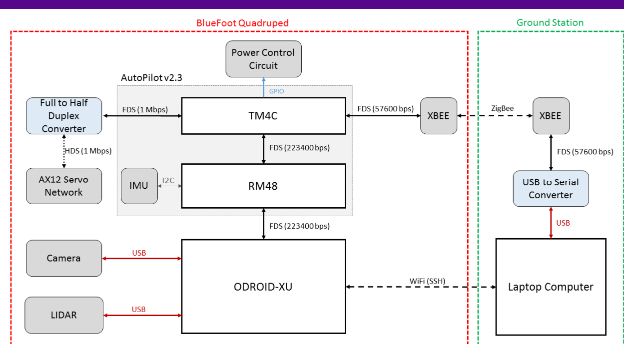
Core Devices



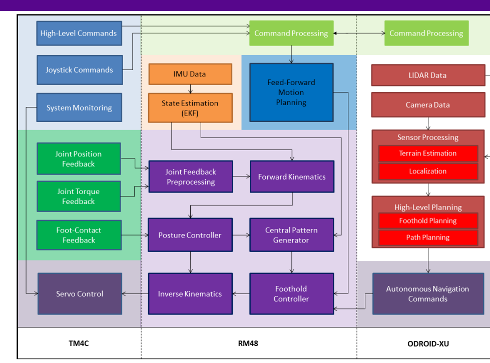
Core Devices

- AutoPilot (IMU, pressure sensor, temperature sensor)
- ODROID-XU computer
- Logitech9000 Web Camera
- Hokuyo-URG LIDAR
- Dynamixel AX12 Smart Servos (x16)
- Binary foot-contact sensors
- XBEE Wireless Radio
- GPS

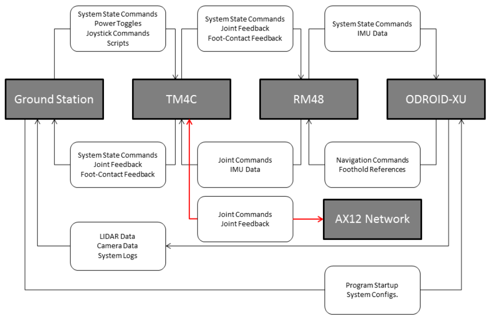
Core Devices



Core Devices



Core Devices



BlueFoot: Kinematic and Dynamical Modeling

State Representation

Considering the robot as a free-floating system with 4 legs (each 4-DOF):

$$\eta = [p_b^T, \theta_b^T, q^T]^T$$

Trunk Position $p_b \in \mathcal{R}^3$

Trunk Orientation $\theta_b \in \mathcal{R}^3$

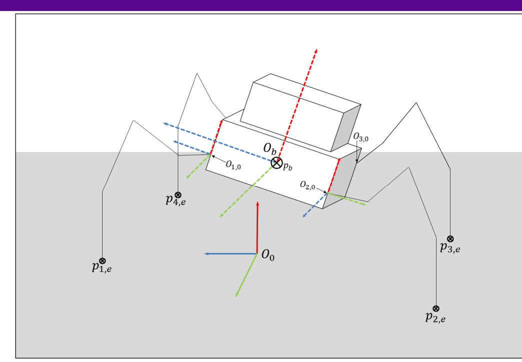
Joint angles $q \in \mathcal{R}^{16}$

Gives rise to the following state vector:

$$z = [\eta^T, \dot{\eta}^T]^T \in \mathcal{R}^{44}$$

$$z_1 = \eta \qquad z_2 = \dot{\eta}$$

Coordinate Frame Setup



Forward Kinematics

First leg frame with respect to world frame O_0:

$$H_0^{i,0} = \left[\begin{array}{c|c} R_{zyx}(\theta_b)R_z(\sigma_i) & R_{zyx}(\theta_b)\vec{\beta}_i + p_b \\ \hline 0 & 1 \end{array}\right] \qquad \begin{aligned} \sigma_i &= \tfrac{\pi}{2}(i-1) + \tfrac{\pi}{4} \\ \vec{\beta}_i &= R_z(\sigma_i)\vec{o}_\nu \\ \vec{o}_\nu &= [\nu, 0, 0]^T \end{aligned}$$

General transformation to each j^{th} joint from the 1st frame of attached to each i^{th} leg:

$$H_{i,j}^{i,0} = \left[\begin{array}{c|c} R_{i,j}^{i,0} & p_{i,j}^{i,0} \\ \hline 0 & 1 \end{array}\right]$$

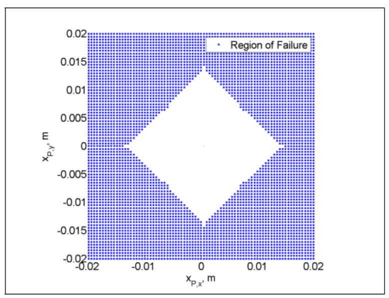
Inverse Kinematics

Joint configuration solution for the i^{th} leg, given $p_{i,e}$ and γ_i:

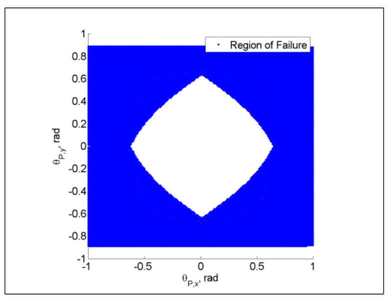
$$\begin{aligned} q_{i,1} &= \cos(i\pi)\left(\frac{\pi}{4} - \psi_i\right) \\ q_{i,2} &= \tan^{-1}\left(\frac{\zeta_{i,z}}{\sqrt{\zeta_{i,x}^2 + \zeta_{i,y}^2}}\right) \mp \cos^{-1}\left(\frac{a_3^2 - a_2^2 - \|\zeta_i\|^2}{2a_2\|\zeta_i\|}\right) \pm \pi \\ q_{i,3} &= \mp\cos^{-1}\frac{\|\zeta\|^2 - a_2^2 - a_3^2}{2a_2a_3} \\ q_{i,4} &= \boxed{\gamma_i} - q_{i,2} - q_{i,3} \end{aligned}$$

Ankle configuration

Kinematic Limits on Trunk Pose



Trunk position (x,y)



Orientation (roll and pitch)

Dynamics

Dynamics of the system takes the following form:

$$M(\eta)\ddot{\eta} + C(\eta,\dot{\eta})\dot{\eta} + G(\eta) + \boxed{\Delta H} = \tau + J^T(\eta)f_{ext}$$

Lump disturbance term

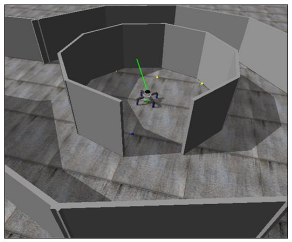
$$\tau = [0_{1x6}, \tau_q^T]^T$$

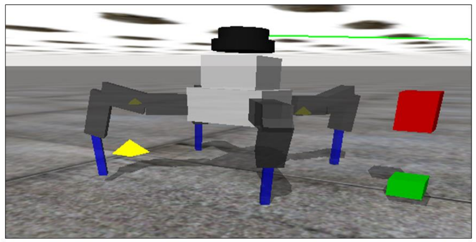
In state-space form:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= M^{-1}(z_1)(\tau + \Phi(z_1, z_2, f_{ext})) \\ \Phi(z_1, z_2, f_{ext}) &= J^T(z_1)f_{ext} - C(z_1,z_2)z_2 - G(z_1) - \Delta H \end{aligned}$$

NYU POLYTECHNIC SCHOOL OF ENGINEERING

Dynamics Simulator





NYU POLYTECHNIC SCHOOL OF ENGINEERING

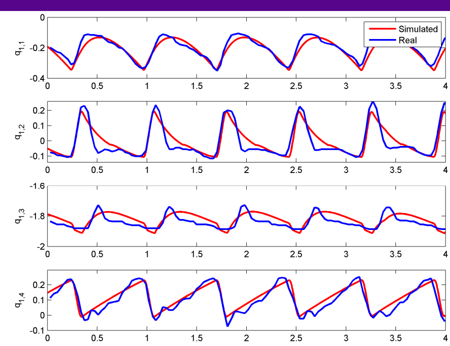
Dynamics Simulator

BlueFoot Simulator implemented using the *Open Dynamics Engine* C++ library

- Generates dynamical updates for all bodies via constrained quadratic minimization
- Rigid body and contact dynamics formulated as a Linear Complimentary problem (LCP)
- First-order integration with small time-step ($\Delta t \ll 1$)

NYU POLYTECHNIC SCHOOL OF ENGINEERING

Dynamics Simulator



NYU POLYTECHNIC SCHOOL OF ENGINEERING

Gaiting and Stability Control

Central pattern generators

- Network of limit-cycle oscillators (unit-oscillators)
  - Hopf oscillators are used in this implementation
- Tunable coupling
  - Particular tuning of coupling generates phase-locked loops, which are applied to driving robot DOFs

Each i^{th} unit oscillator has a state-pair

$$\{y_{1,i}, y_{2,i}\}$$

The full CPG network dynamics are

$$\begin{aligned}\dot{y}_1 &= A_1\left(\Psi_M M\left(y_1, y_2\right) - \Gamma\right) y_1 + \Psi_\omega W y_2 \\ \dot{y}_2 &= A_2\left(\Psi_M M\left(y_1, y_2\right) - \Gamma\right) y_2 - \Psi_\omega W y_1 + K y_2\end{aligned}$$

with

$$M\left(y_1, y_2\right) = \begin{bmatrix} y_{1,1}^2 + y_{2,1}^2 & 0 & 0 & 0 \\ 0 & y_{1,2}^2 + y_{2,2}^2 & 0 & 0 \\ 0 & 0 & y_{1,3}^2 + y_{2,3}^2 & 0 \\ 0 & 0 & 0 & y_{1,4}^2 + y_{2,4}^2 \end{bmatrix}$$

Reflexes are designed using modulation parameters:

$$\begin{aligned}\Psi_\omega &= I + A_\omega \mathrm{diag}\left(\psi_i\right) \\ \Psi_M &= I - A_\mu \mathrm{diag}\left(\psi_i\right) \\ \psi_i &= \mathrm{sig}\left(w_i v_i - w_i c_i\right) \mu_i\end{aligned}$$

Coupling matrices used for gaiting the BlueFoot Platform:

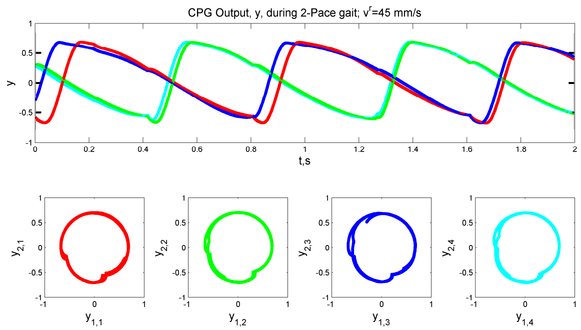
$$K \equiv \begin{bmatrix} 0 & -1 & 1 & -0.5 \\ -1 & 0 & -0.5 & 1 \\ -1 & -0.5 & 0 & -1 \\ -0.5 & 1 & -1 & 0 \end{bmatrix}$$

Produces 4-Pace Walking Gait

$$K \equiv \begin{bmatrix} 0 & -1 & 1 & -0.5 \\ -1 & 0 & -0.5 & 1 \\ -1 & 0.5 & 0 & -1 \\ 0.5 & -1 & -1 & 0 \end{bmatrix}$$

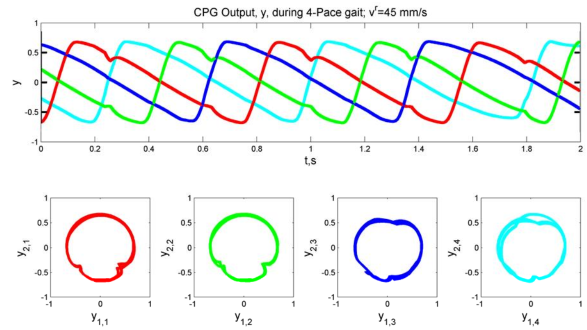
Produces 2-Pace Trotting Gait

Central Pattern Generators (CPG)



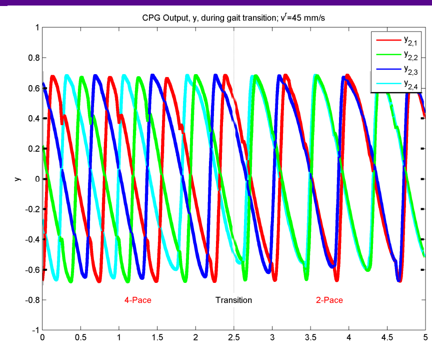
NYU POLYTECHNIC SCHOOL OF ENGINEERING

Central Pattern Generators (CPG)



NYU POLYTECHNIC SCHOOL OF ENGINEERING

Central Pattern Generators (CPG)



NYU POLYTECHNIC SCHOOL OF ENGINEERING

Foothold Control

Reference velocity of each foot is controlled by:

$$\dot{p}^r_{i,e} = \dot{\tilde{p}}^v_{i,e} + \dot{\tilde{p}}^y_{i,e}$$

with

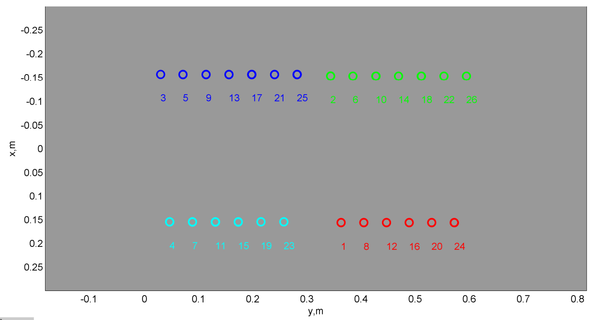
$$\dot{\tilde{p}}^v_{i,e} = \frac{1}{m_i}\int^t P_{\vec{h}_i}\left(F_{s,i} - b_c \dot{p}_{i,e} + F_{\epsilon,i}\right) d\tau$$

$$F_{s,i} = k_c\left(\tilde{p}_{i,e} - p_{i,e}\right) U(y_{2,i})$$

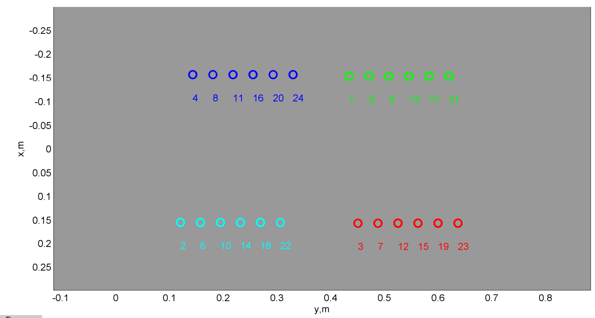
$$F_{\epsilon,i} = a_\epsilon \dot{\epsilon}_\theta U(y_{2,i})$$

$$\dot{\tilde{p}}^y_{i,e} = \left(\alpha_v v^r + \alpha_\omega \|\omega^r\|\right) \dot{y}_{2,i} \vec{h}_i$$

Foothold Control



Foothold Control



ZMP Body Control

Zero Moment Point (ZMP) Control

- Zero Moment Point is a point beneath the robot at which the net moment on robot body is zero
- Control robots center of gravity to reduce angular velocity of the trunk during gaiting (using trunk)

ZMP Body Control

ZMP computed by:

$$p^{b'}_{ZMP} = R_{z_P}\left(\frac{\pi}{2}\right)\left(\frac{\|g\|}{m_b}\right)\tau_{legs} + \hat{p}^{b'}_{COG}$$

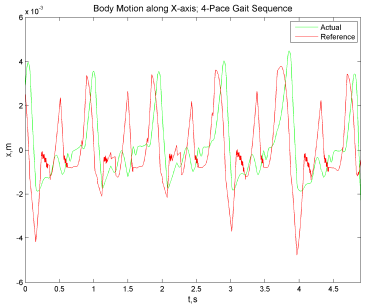
Body reference position controlled by:

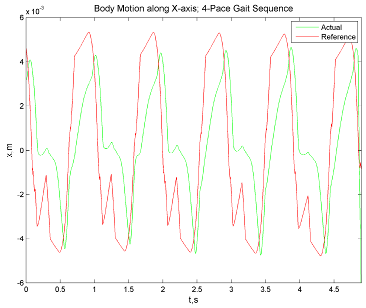
$$\dot{p}^{b',r}_{b} = P_{\tilde{h}_i}\left(K_Z(p^{b'}_{ZMP} - p^{b'}_{b}) + K_F\frac{F_r}{m_b}\right)$$

$$\begin{aligned} F_r &= \sum_{i=1}^{4} e^{(k_l r_i^-)} + e^{(k_l r_i^+)} \\ r_i^+ &= \left\|p^{b'}_{i,e} - p^{b'}_{b}\right\| - r_{max} \\ r_i^- &= r_{min} - \left\|p^{b'}_{i,e} - p^{b'}_{b}\right\| \end{aligned}$$

NYU POLYTECHNIC SCHOOL OF ENGINEERING

ZMP Body Control





NYU POLYTECHNIC SCHOOL OF ENGINEERING

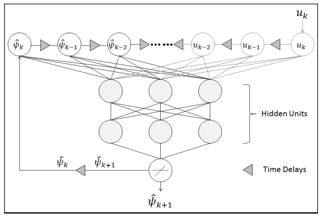
NARX-NN Trunk Leveling Controller

NARX-Neural Network (NARX-NN)

Used to capture periodic disturbance dynamics during gaiting

$$\begin{aligned} \hat{\psi}_{k+1} &= \mathcal{N}(\hat{\Psi}_k^N, U_k^N) \\ \hat{\Psi}_k^N &= [\hat{\psi}_k, \hat{\psi}_{k-1}, \ldots, \hat{\psi}_{k-N+1}] \\ U_k^N &= [u_k, u_{k-1}, \ldots, u_{k-N+1}] \end{aligned}$$

$$u_k = (z_{1,k}, z_{2,k}, f_{ext,k})$$



NYU POLYTECHNIC SCHOOL OF ENGINEERING

NARX-NN Trunk Leveling Controller

Isolate trunk dynamical subsystem:

$$\ddot{\theta}_b = \Gamma_1 M^{-1}(z_1)(\Gamma_2 \tau_q + \Phi) \qquad \ddot{\theta}_b = -K_b\theta_b - K_d\dot{\theta}_b$$

Estimate corrected torque using NARX-NN prediction:

$$\hat{\tau}_{q,k} = -\left[\Gamma_1 \hat{M}_{1,k}^{-1}\Gamma_2\right]^{\dagger}\left[\Gamma_1 \hat{M}_{1,k}^{-1}\hat{\psi}_{k+1} + K_b\theta_{b,k} + K_d\dot{\theta}_{b,k}\right]$$

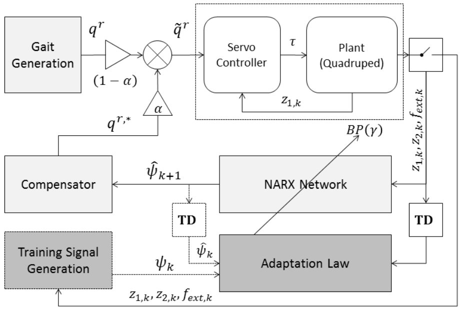
Generate corrected joint reference from inverted actuator dynamics:

$$\boxed{\tau_q = k_s(q^r - q)} \longrightarrow q_{1,k}^{r,*} = k_s^{-1}(\hat{\tau}_{q,k}) + q_{1,k}$$

$$\tilde{q}_k^r \leftarrow (1-\alpha)q_k^r + \alpha(q_k^{r,*})$$

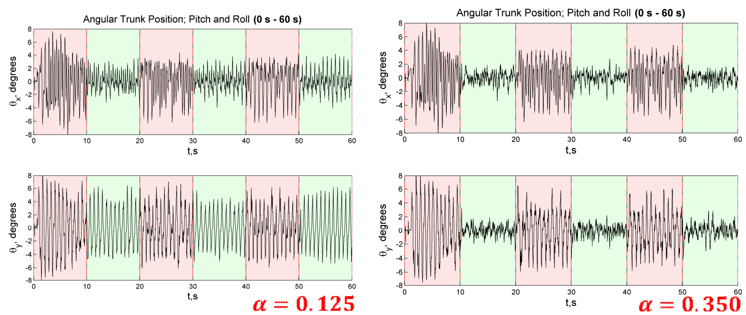
NYU POLYTECHNIC SCHOOL OF ENGINEERING

NARX-NN Body Leveling Controller



NYU POLYTECHNIC SCHOOL OF ENGINEERING

NARX-NN Trunk Leveling Controller



NYU POLYTECHNIC SCHOOL OF ENGINEERING

NARX-NN Trunk Leveling Controller

NYU POLYTECHNIC SCHOOL OF ENGINEERING

Navigation Control

NYU POLYTECHNIC SCHOOL OF ENGINEERING

Potential Fields/Visual-Servoing

LIDAR-Based Potential Fields

- Virtual forces generated from points of interest (obstacles) sensed using distance sensor (LIDAR).
- Forces guide the robot around obstacles

Image-Based Visual Servoing

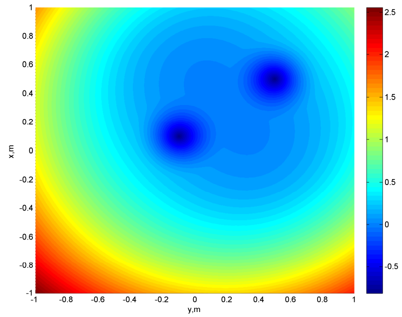
- Features (targets) extracted from images
- Robot is tracks features by gaiting and articulating its trunk
- Used to make the robot track a ñleaderò

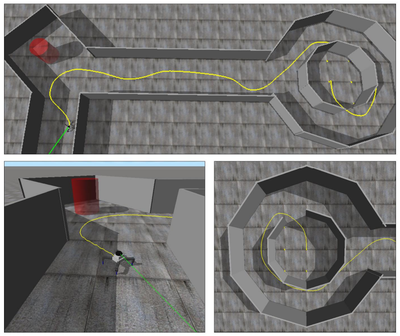
Potential Fields Formulation

$$
\begin{aligned}
\vec{u}^{r}_{L} &= \sum_{x_i^L \in S^L} g_{\psi}(x_i^L) f(x_i^L) \frac{x_i^L}{\|x_i^L\|} \\
P_L &= \alpha_p \sum_{x_i^L \in S^L} g_{\psi}(x_i^L) f(x_i^L) U\left(f(x_i^L)\right)
\end{aligned}
$$

Potential function:

$$
\begin{aligned}
\Delta d &= \|x\| - d_{min} \\
f(x) &= \begin{cases} -\lambda_{c,1}\left(\Delta d\right)^2 & \text{if } \Delta d < 0 \\ (\Delta d)\left(1 - e^{-\lambda_{c,2}(\Delta d)^2}\right) & \text{otherwise} \end{cases}
\end{aligned}
$$





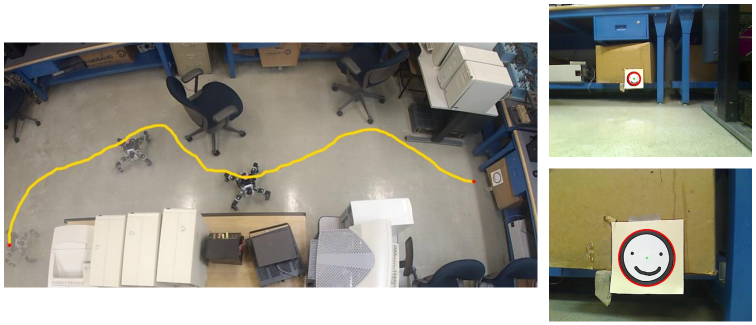
Feature extractor (blobs, circle, etc.) used to find feature locations:

$$
p_{Im} = [u, v]^T
$$

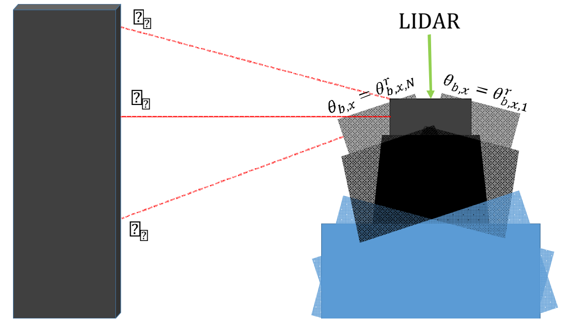
Visual servoing controller using feature location:

$$
\begin{aligned}
v^r_C &= v^{r,max}_C \left(1 - e^{-c_r(r - r_{min})^2}\right) \\
\omega^r_C &= \omega^{r,max}_C \left(\frac{w_{Im} - 2u}{w_{Im}}\right) \\
\theta^r_{b,x} &= \theta^{r,max}_{b,x} \left(\frac{2v - h_{Im}}{h_{Im}}\right)
\end{aligned}
$$

Potential Fields/Visual-Servoing



3D Point Cloud Composition



3D Point Cloud Composition

Trunk pitched according to:

$$\theta^*_{b,x,k} = (\theta_{s,max} - \theta_{s,min})\left(\frac{k}{K}\right) + \theta_{s,min}$$

$$\dot{\theta}^r_{b,x} = \alpha_\theta(\theta^*_{b,x,k} - \hat{\theta}_{b,x})$$

Points are transformed and collected to form 3D cloud:

$$\begin{bmatrix} \bar{x}_{i,k} \\ 1 \end{bmatrix} = H_b^L H_0^b \begin{bmatrix} x_i^L \\ 0 \\ 1 \end{bmatrix} \forall x_i^L \in S^L \longrightarrow \bar{S} = \bigcup_{k=1}^{N_s} \bar{S}_k \quad \bar{x}_{i,k} \in \bar{S}_k$$

Surface Representation from 3D Point Clouds

Height-Map Representation

Å Region of interest is discretized

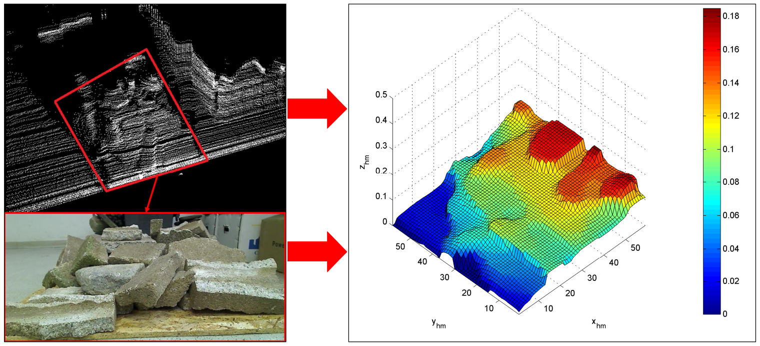
Å Heights of each sub-region are generated by averaging relative points heights (z-components in world frame)

Representation using Estimated Surface-Normals

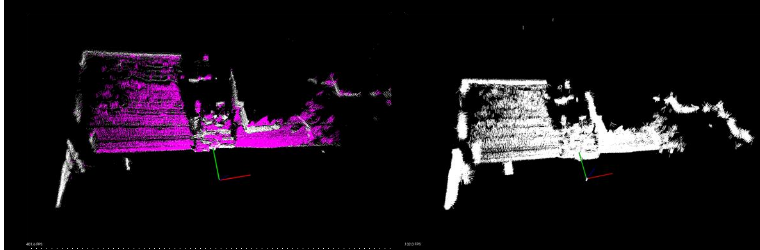
Å Plane fitting on moving subsets of points using principle component analysis technique

Å Normal emanating from each point generated to represent surface

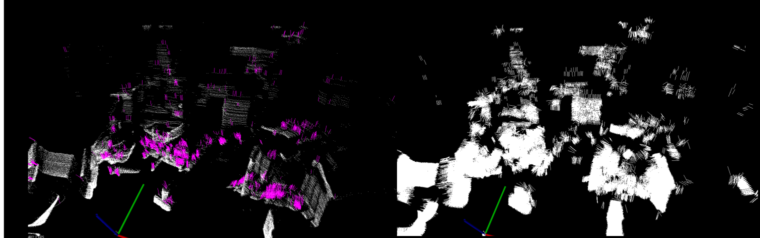
Height-Map Representation



Surface-Normal Representation



Surface-Normal Representation



Concluding Remarks

Future Directions

- Improvements to robotő actuator outfitting
- Surface maps to be used in rough-terrain planning
  - Foothold placement planning and optimization
- Prediction of terrain fitness based on few features
- Whole body motion planning for surface mapping with LIDAR/Camera during gaiting