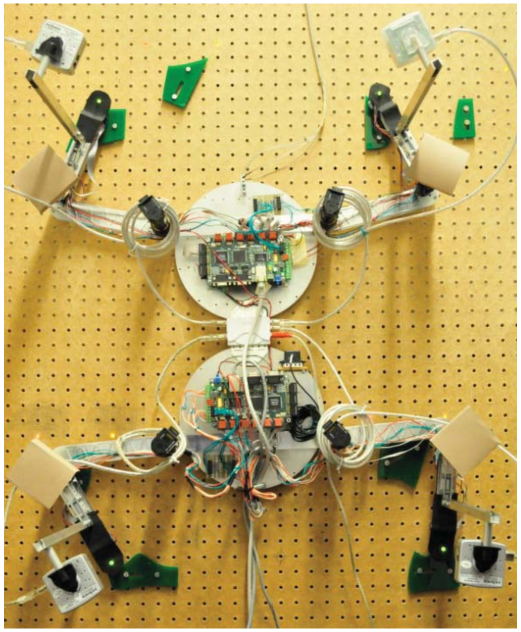
Design and Implementation of an Autonomous Climbing Robot[1]



Ruixiang Zhang

Computer Science Department
Stanford University
6/1/2010

Motivation

BRADLEY, Mars rover

(NASA)

SCARAB lunar rover

(NASA and CMU)

ASIMO

(Honda)

BIG DOG

(Boston Dynamics)

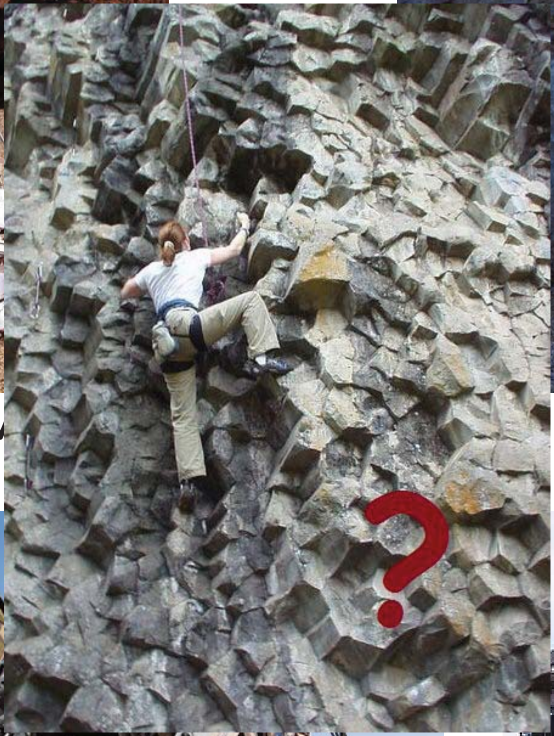
ATHLETE

(NASA, JPL)

Motivation

BRADLEY, Mars rov

(NASA)



SCARAB lunar rover

(NASA and CMU)

ASIMO

(Honda)

BIG DOG

(Boston Dynamics)

ATHLETE

(NASA, JPL)

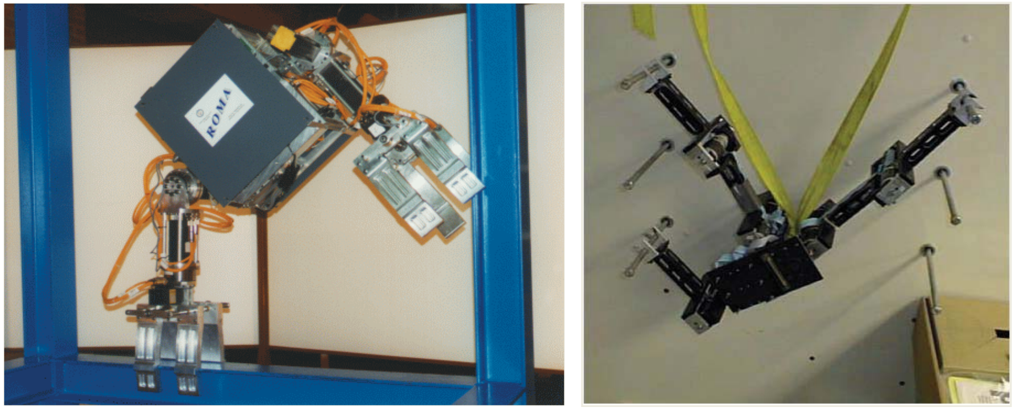
Climbing robots

Aid Climbing

- Use of special tools
- Engineered environment

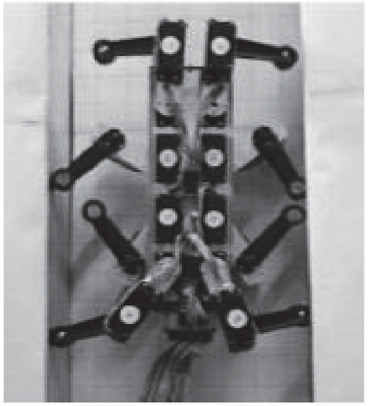
Adhesive

Hirose et al., 1991



Engineered

Balaguer et al., 2000; Bevly et al., 2000



Pipes and Ducts

Neubauer, 1994

Bio-inspired *material to generate adhesive* friction on flat surface.

Stickybot (flat surface)

Stanford

Micro-claws to climb on textured surfaces

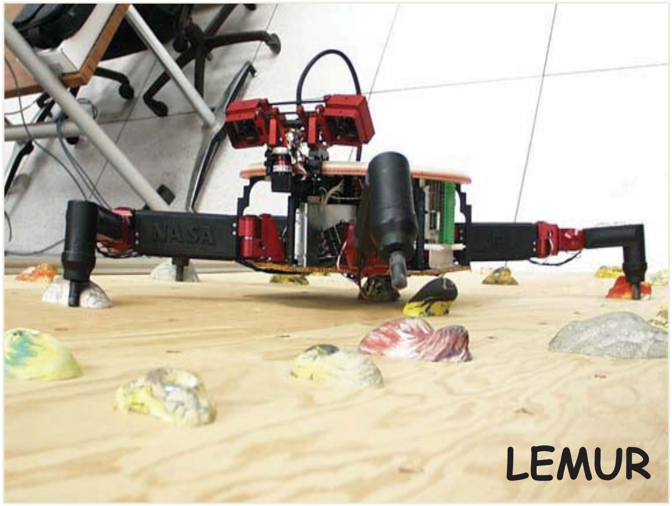
Spinybot (textured surface)

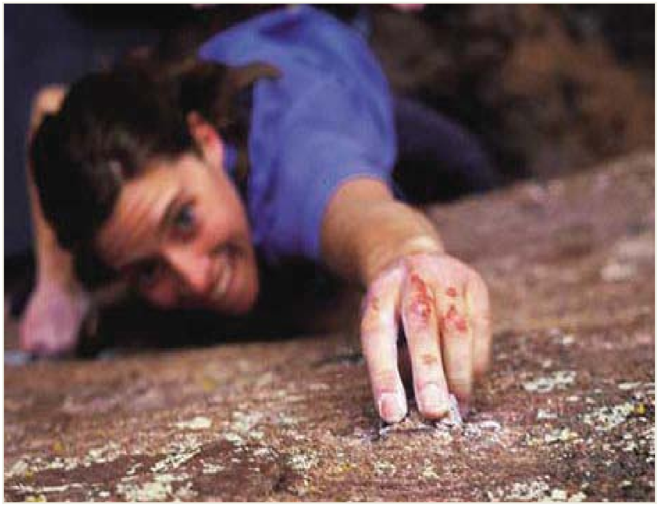
Boston Dynamics

Free Climbing

Relies only on the frictional contact between finger and terrain surface
No special terrain feature required (only protrusion or holes)
No special tools on robot finger

Free climbing requires deliberate planning and control

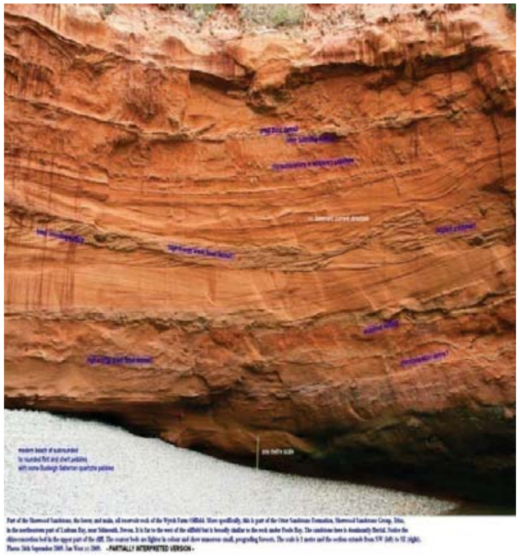


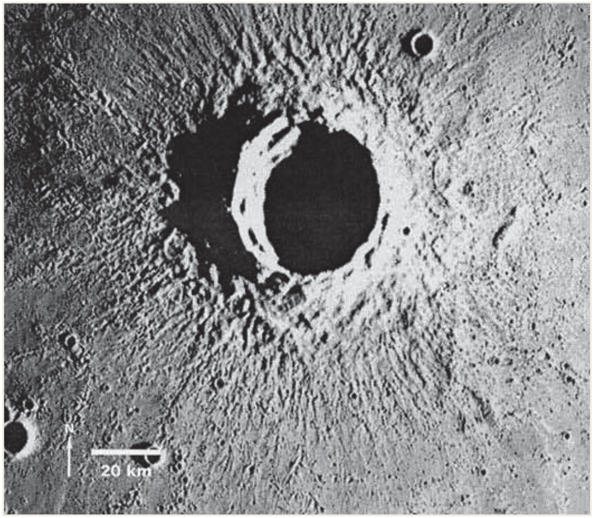


Dynamics? Good climber rarely do dynamics, dangerous, too much uncertainty…
Most motion can be achieved by quasi-static motion
Quasi-static motion and quasi-static equilibrium

Potential applications

- Cliffs of significant scientific and geological interest
- Planet exploration
- Search and rescue





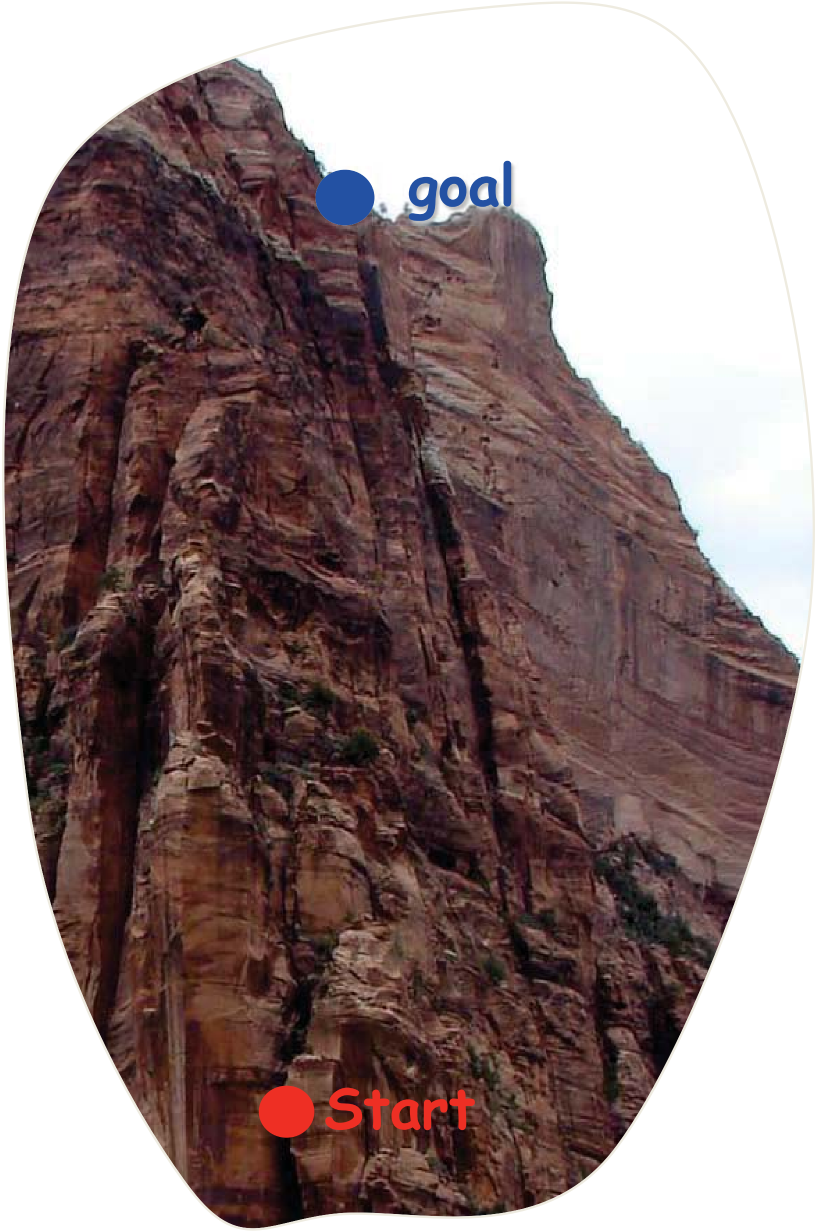
- Problem solving process, understanding locomotion in extreme uneven terrain

General Scenario

1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control

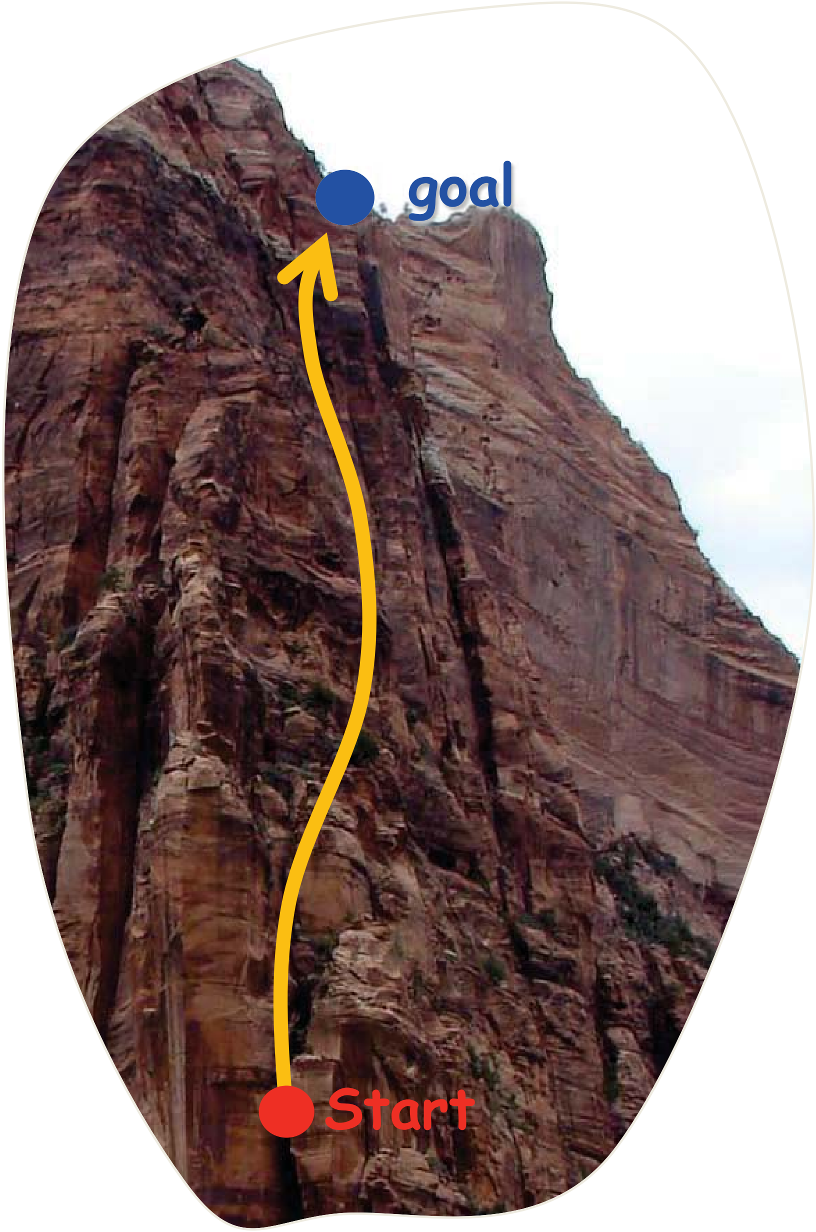
General Scenario

1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control



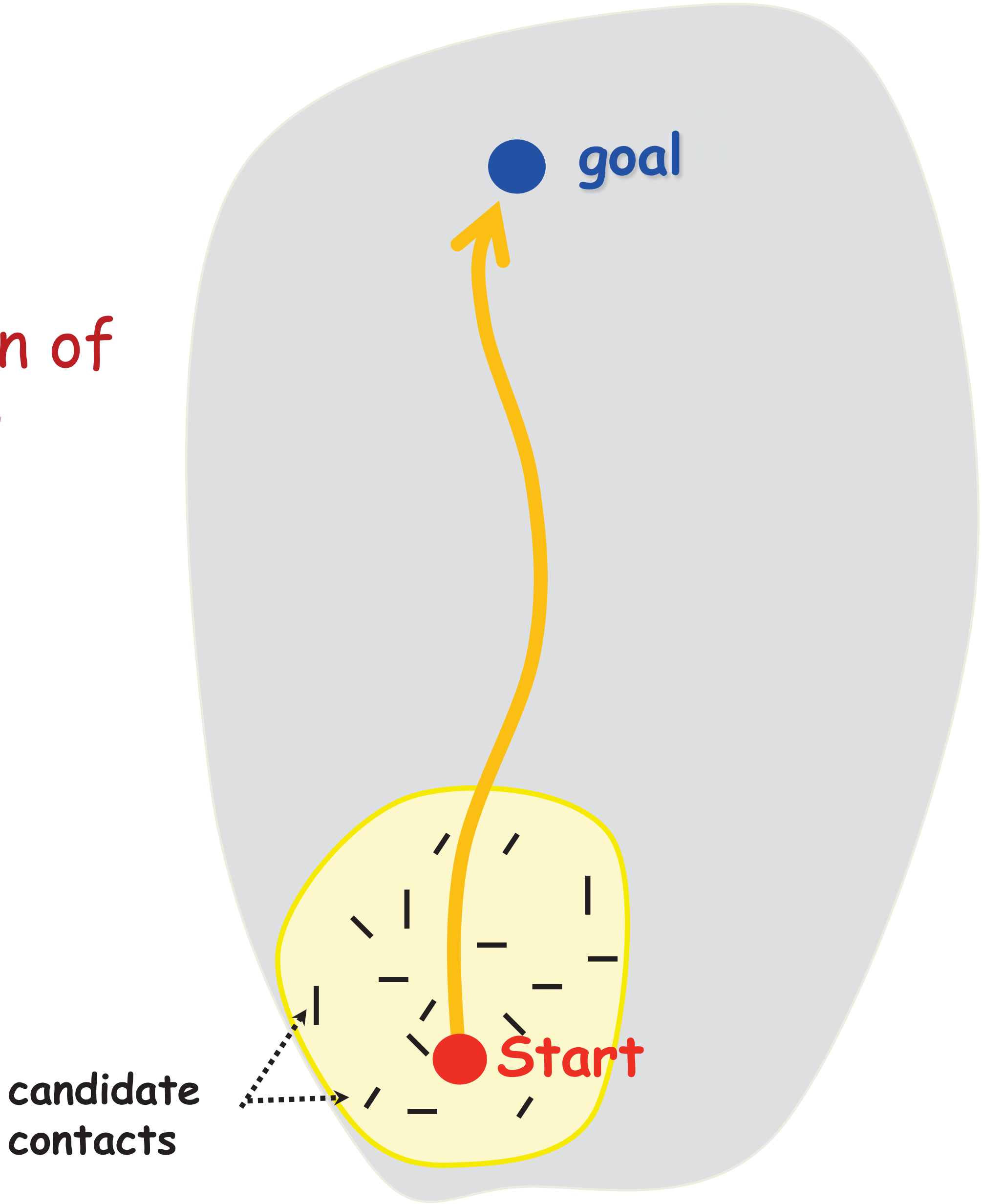
General Scenario

1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control



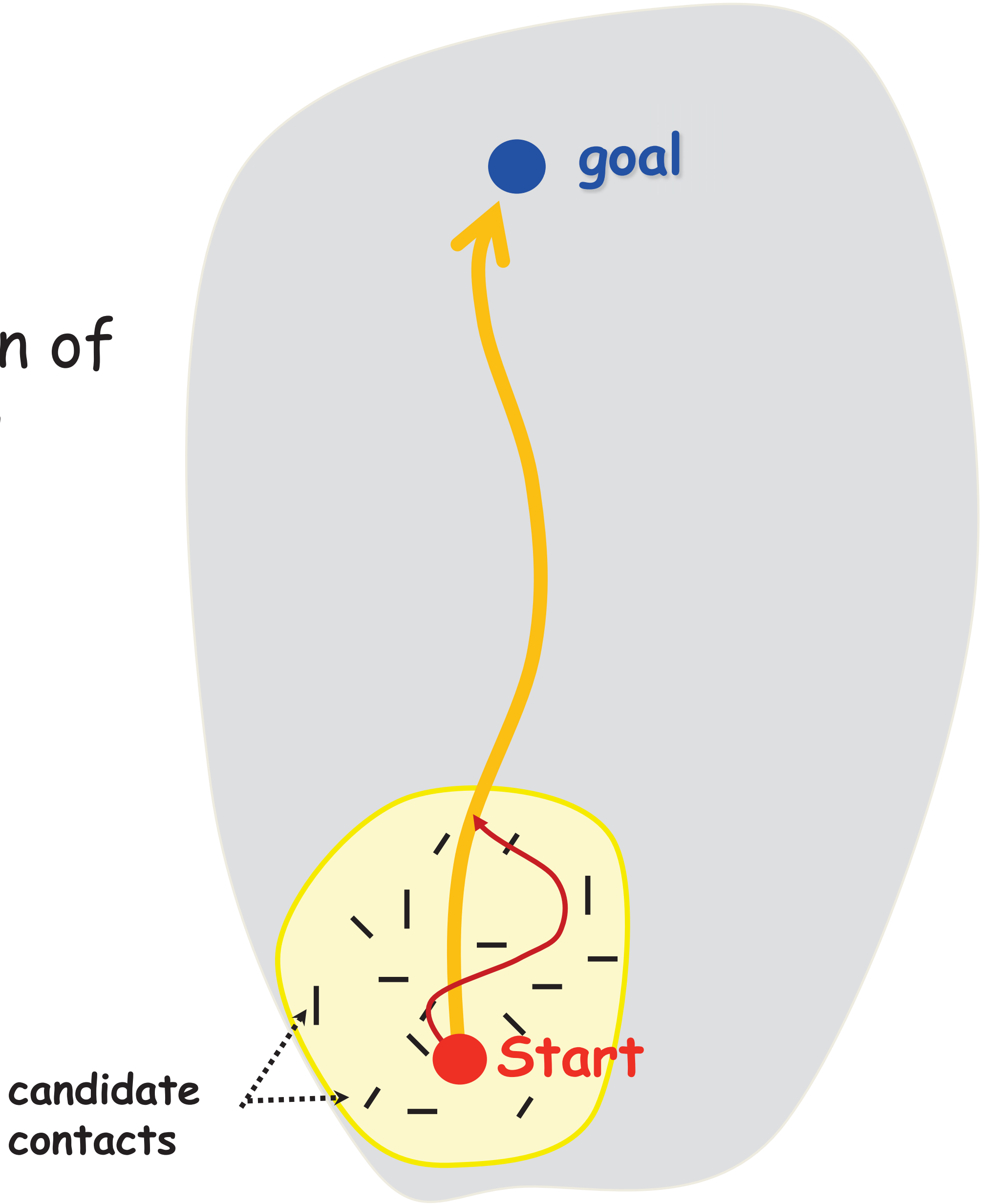
General Scenario

1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control



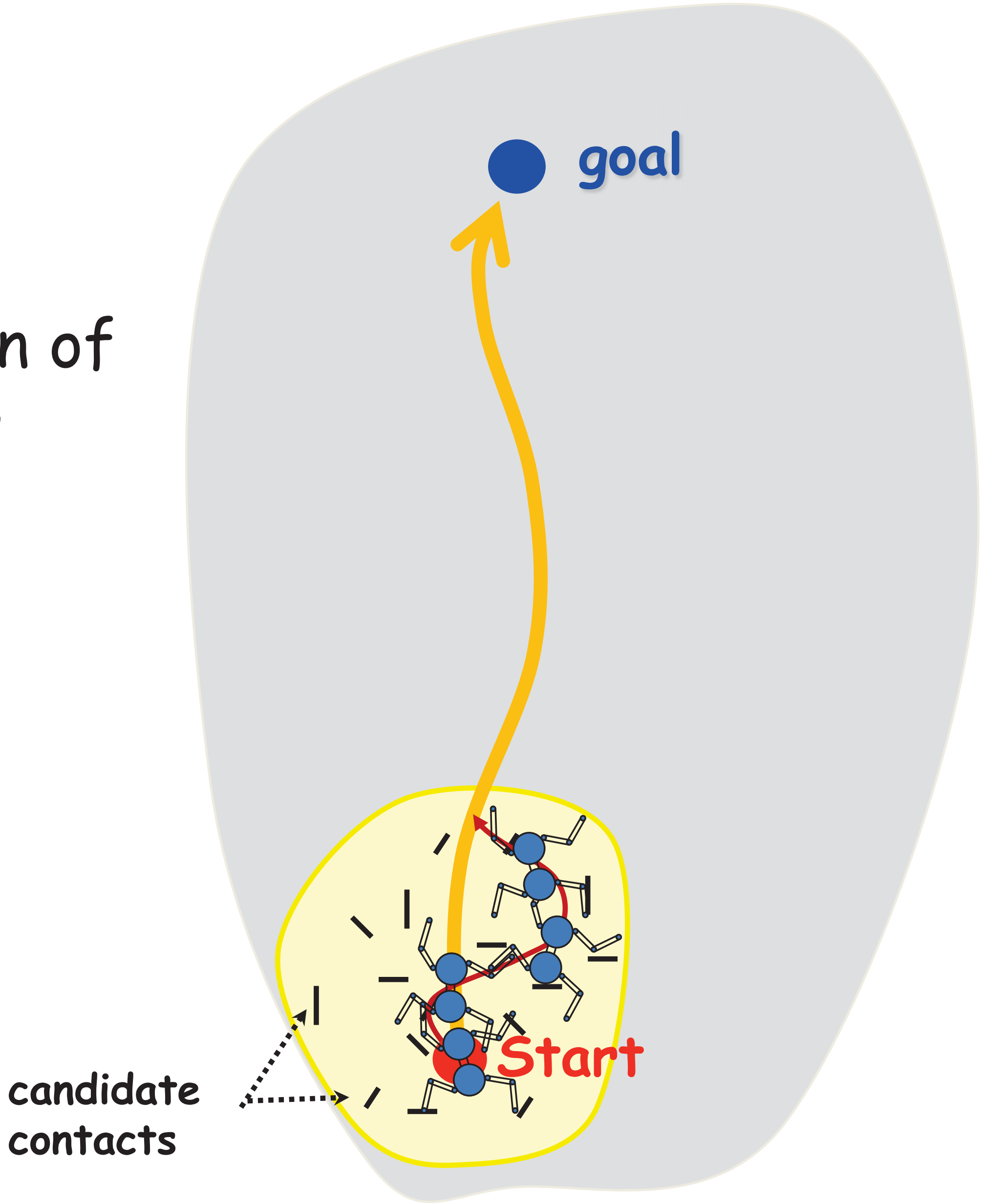
General Scenario

1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control



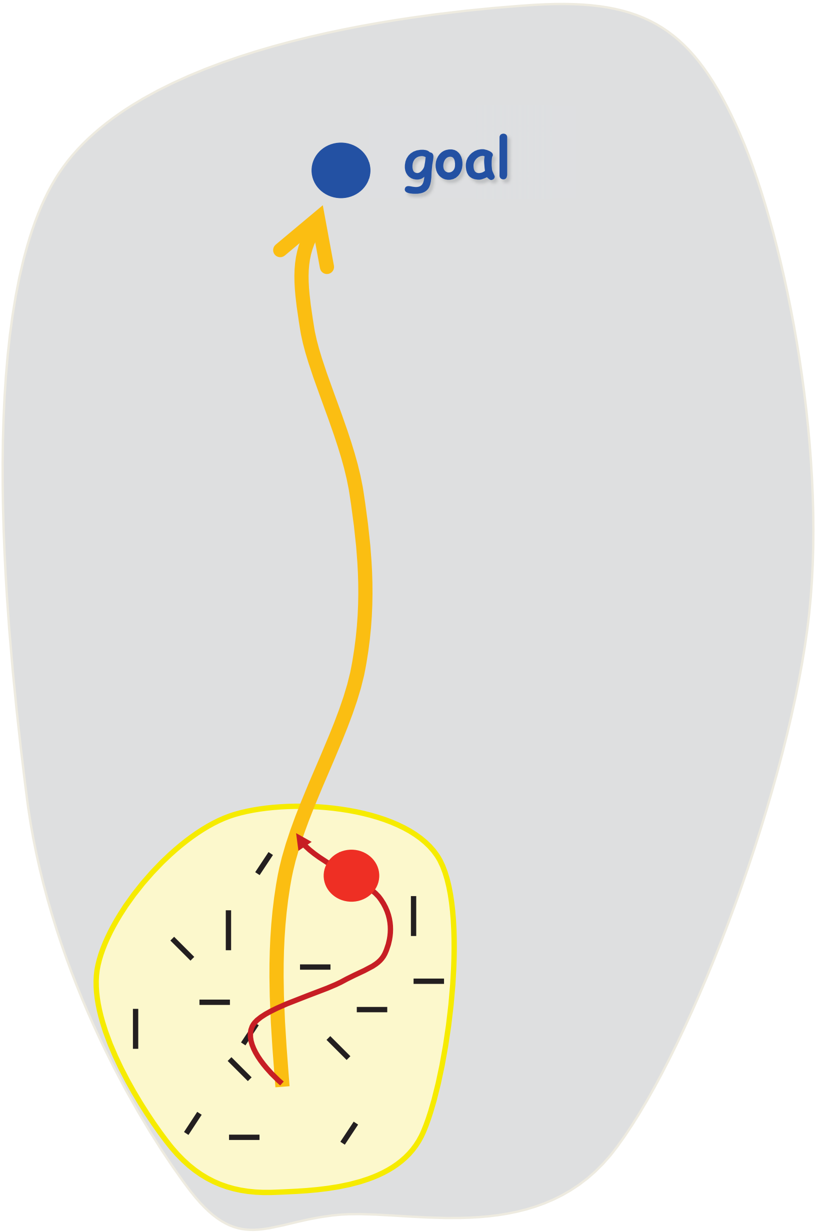
General Scenario

1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control

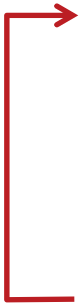


General Scenario

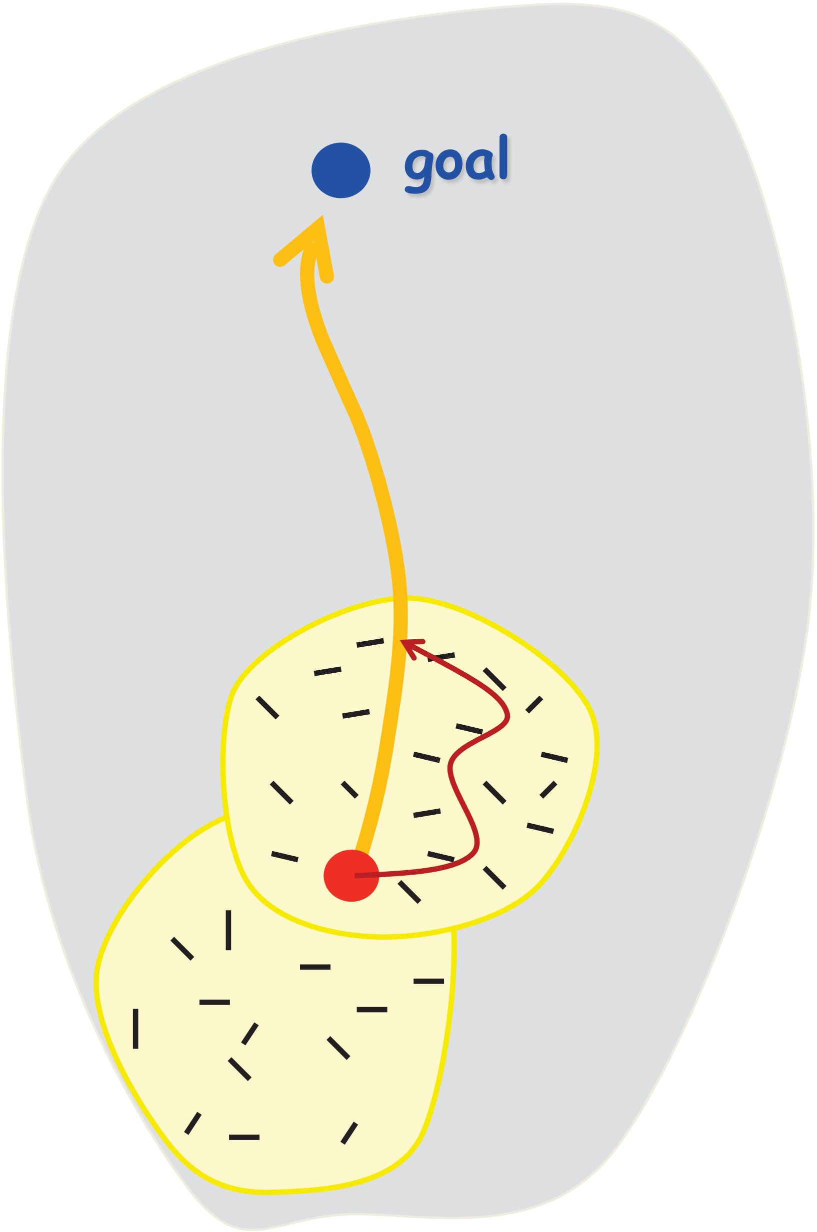
1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control



General Scenario



1. Global sensing of terrain
2. Planning of coarse route
3. Local sensing and detection of potential contact modeling
4. Detailed motion planning
5. Motion control



Previous work

Tim Bretl

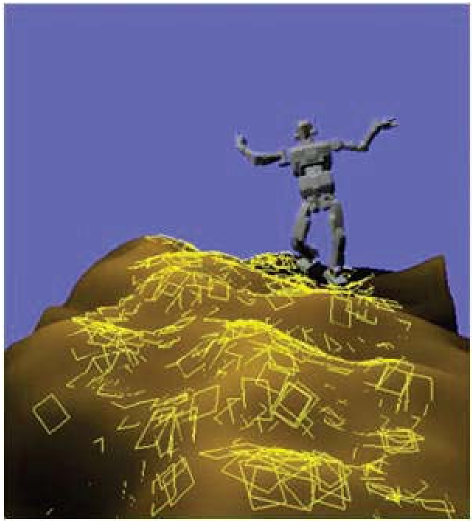
Designed the framework

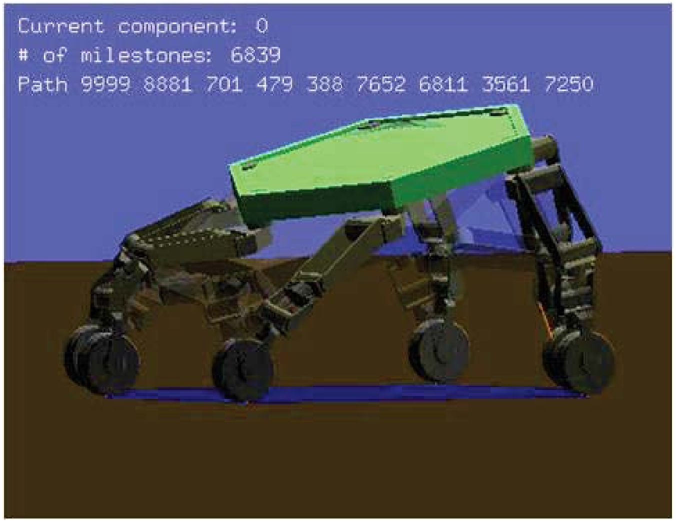
Implemented on LEMUR

Kris Hauser

Improved the generality and efficiency

Simulated HRP2 and ATHLETE





```
Current component: 0
# of milestones: 6839
Path 9999 8881 701 479 388 7652 6811 3561 7250
```

T. Bretl (2006) Motion Planning of Multi-Limbed Robots Subject to Equilibrium Constraints: The Free-Climbing Robot Problem. *International Journal of Robotics Research*

K. Hauser (2008) Motion Planning for Legged Robots on Varied Terrain. *International Journal of Robotics Research*

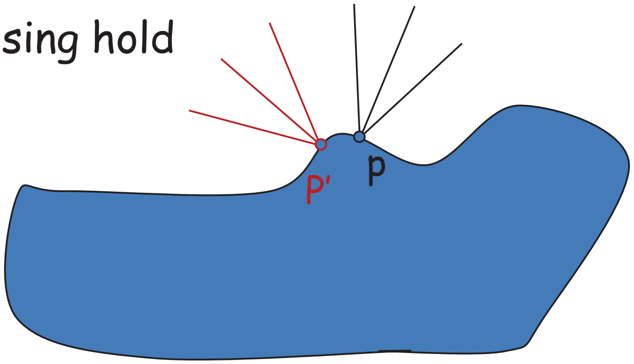
Lemur climbing

- LEMUR robot climbing open-loop up a climbing wall

LEMUR IIb
(Mechanical and Robotic Technologies Group, JPL)

Limitations of open-loop climbing

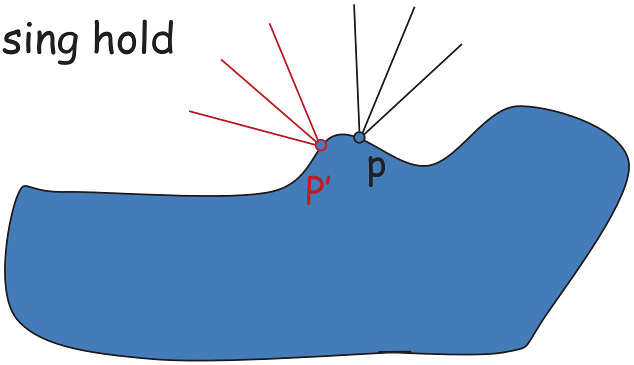
- Incorrect positions of contacts
  - Incorrect contact orientation → slipping
  - Missing hold



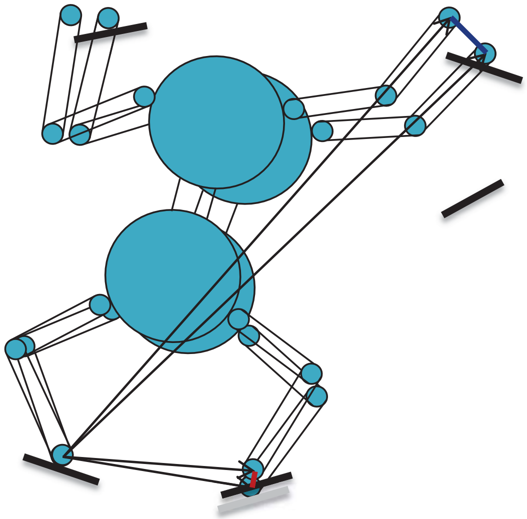
Uncertainties come from the sources:
Terrain sensing is not precise
Joint angle error in execution (a little backlash)
Finger slipping while climbing

Limitations of open-loop climbing

- Incorrect positions of contacts
  - Incorrect contact orientation → slipping
  - Missing hold

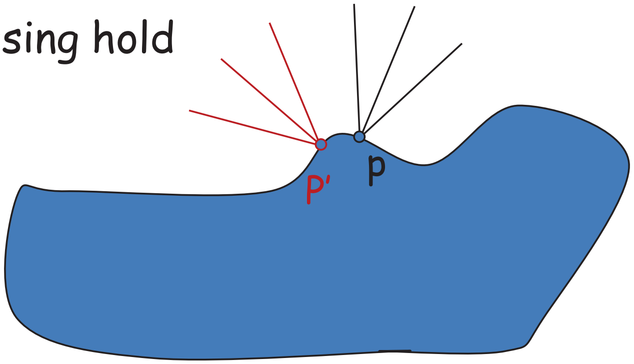


The position error of robot relative to terrain can be enlarged with climbing

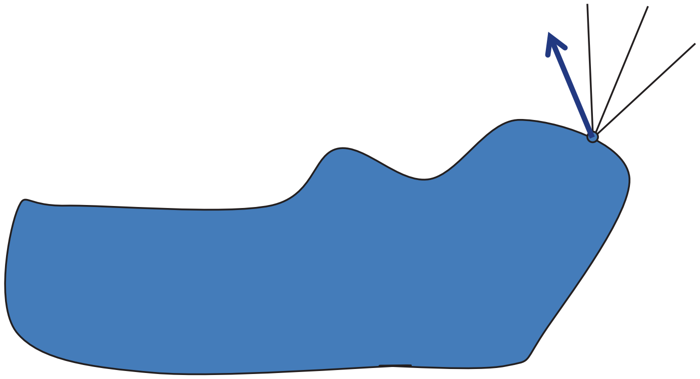


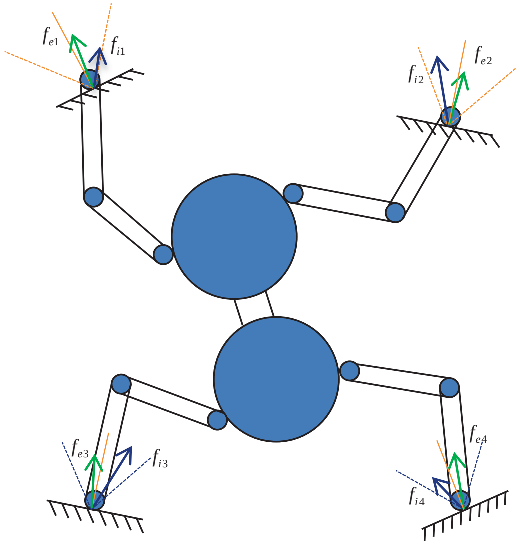
Limitations of open-loop climbing

- Incorrect positions of contacts
  - Incorrect contact orientation → slipping
  - Missing hold



- Incorrect contact forces
  - Out of friction cone, cause slipping
  - Not well balanced, cause torque exceeding limit





Even fingers are at desired contact positions, the contact forces will still be incorrect

Solutions

- Sensing
  - Force sensors
  - Vision sensors
- Feedback control
  - Force feedback control algorithm
  - Vision feedback control algorithm

My work

- Design of new robot: Capuchin
  - Design and build a multi-limb climbing robot
  - Equip the robot with various sensors, such as force and vision sensor, to sense the force and terrain
- Motion control algorithm
  - Design a control algorithm that takes advantage of various sensors feedback to make the climbing more precise and robust
- System integration
  - Integrate the planning, control and sensing system
- Experiments
  - Test the system on vertical artificial climbing wall to verify the robot design, sensing system and control algorithm

Outline of the rest of the talk

- Robot design
- Sensors
- Motion control algorithm
- Experiments

Kinematics design

4-limb design:

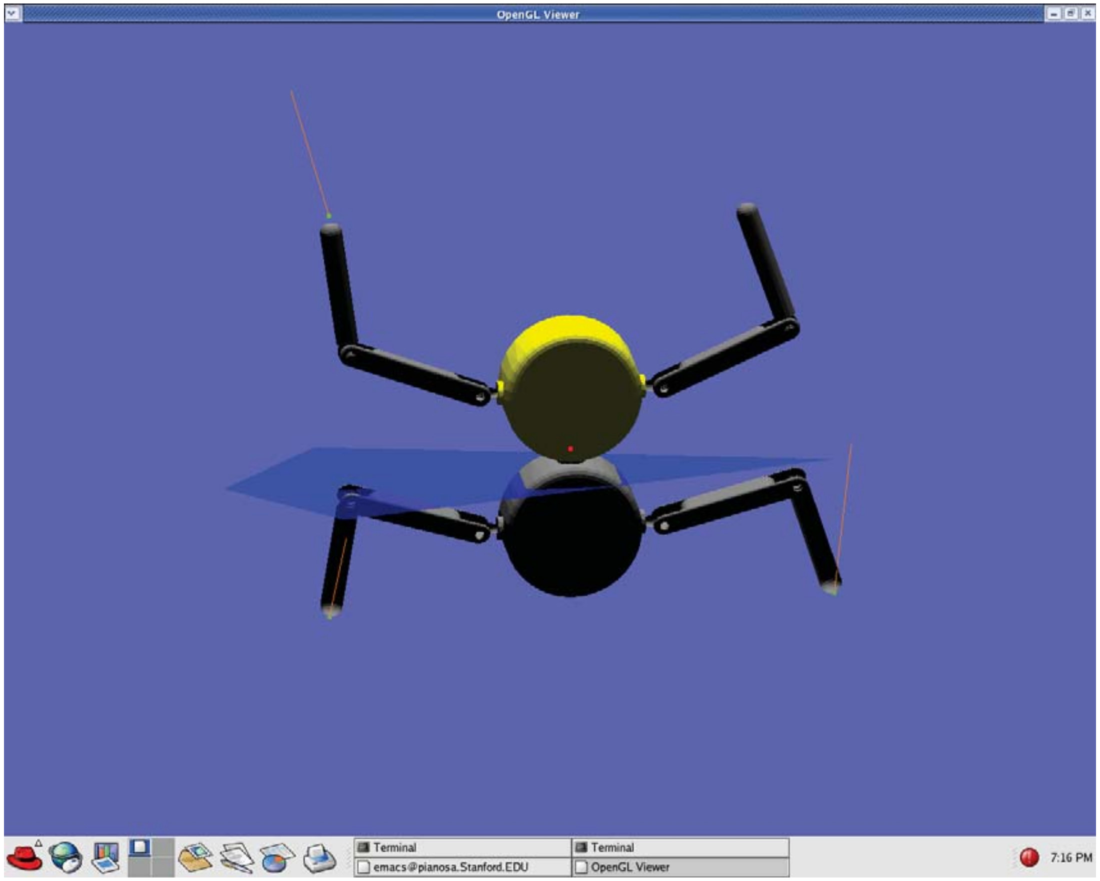
1) 3D terrain requires at least 4 limbs to climb.

2) A lot of 4-limb creatures are good at climbing. More limbs will make the system and control more complicated

Follow the structure of human and other 4-limb animals:

- Two links/joints on each limb

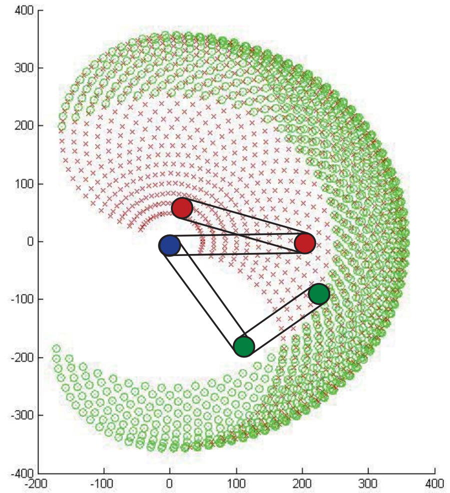
There were 5 DOFs (shoulder, hips, torso) for 3D dimensions. Due to time limitation, we did not implement these.



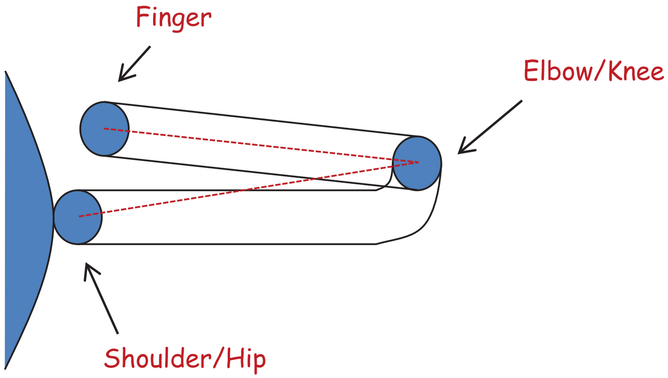
Robot joint configuration design

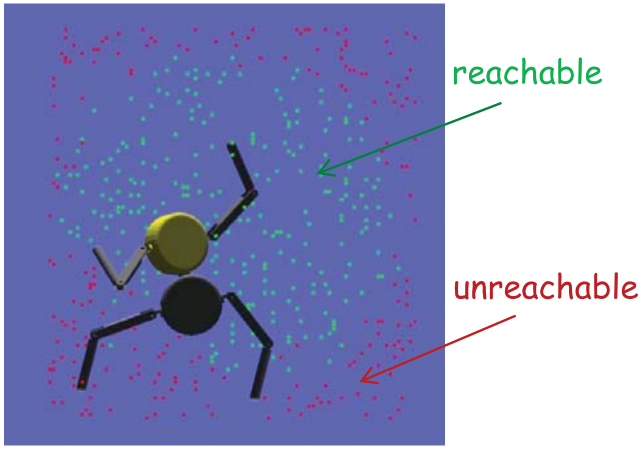
Limb design

- Hard limit on elbow/knee joint at 170
  - Avoid singularity
  - Avoid multi-solution for IK
- Elbow/knee joint range: 10-170
  - Maximize reachibility
- Upper limb:185mm Lower limb 172mm



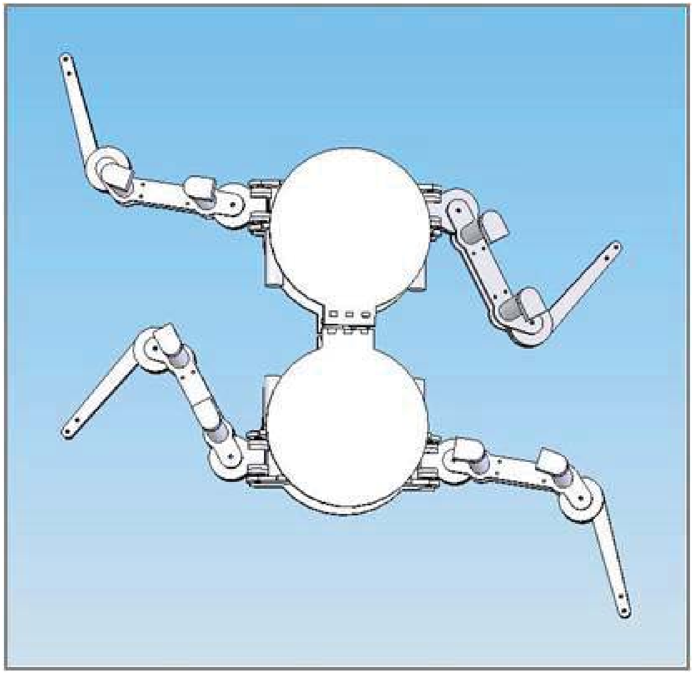
2D simulation of robot finger workspace for Capuchin and Lemur

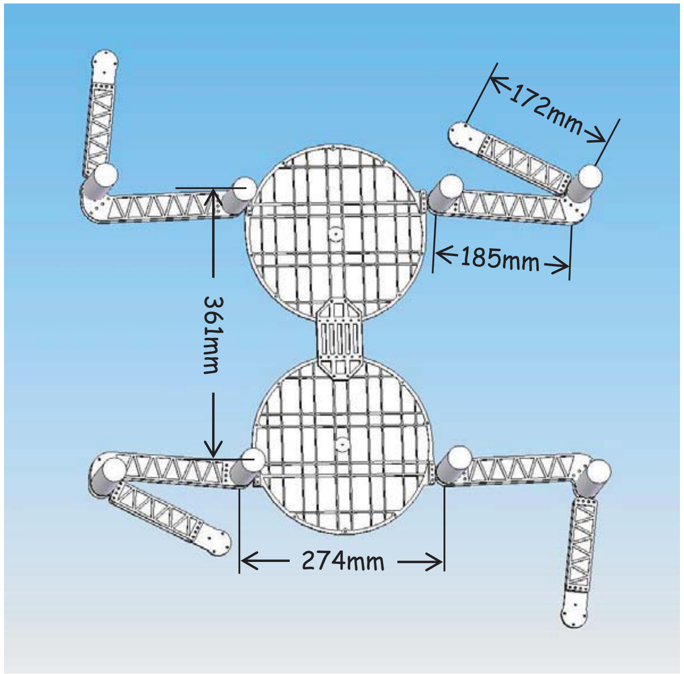




3D simulation of robot finger workspace

Actuator selection and mechanical design





Motor + Pulley drive
Low friction, backdrivable
Linear input to output
Reduction ratio 20:1
Max torque: 0.6Nm
Maxon Motor 118746

Motor + Gearhead
Large friction, less backdrivable
Not linear
Reduction ratio 190:1
Max torque: 5.5Nm
Maxon Motor 118746

Outline

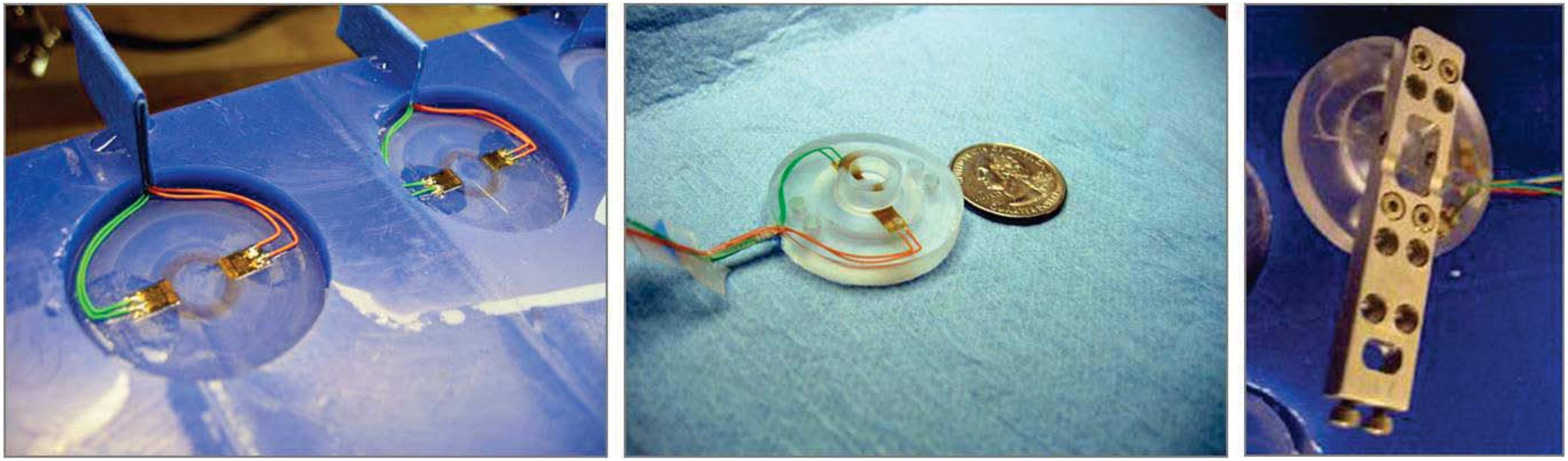
- Robot design
- Sensors
- Motion control algorithm
- Experiments

Force sensor

Small, light weight and durable force sensors to put on fingers

- First attempt: design our own force sensors

Motivation: low cost, easy to customize design



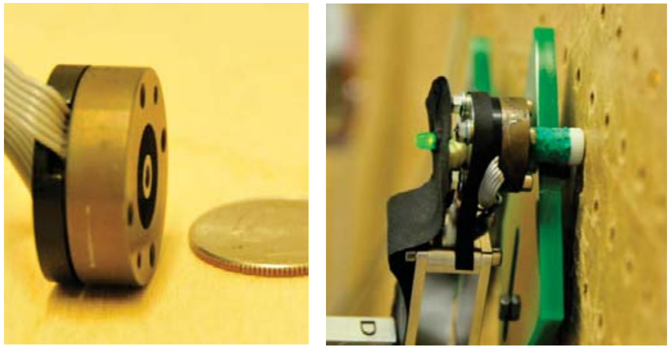
A joint work with Stanford Biomimetics and Dexterous Manipulation Lab, Prof. Mark R. Cutkosky

Problems: output is noisy and drifts with temperature

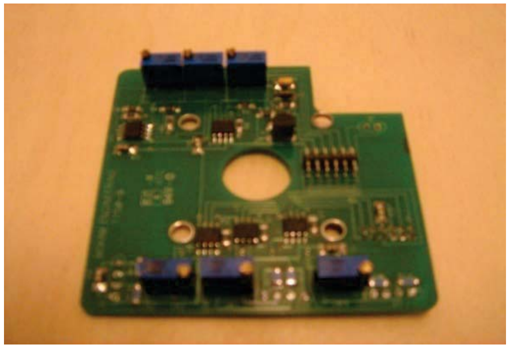
Force sensor

- Final choice: 3 axis strain gauge force sensor (Bokam Engineering Inc)

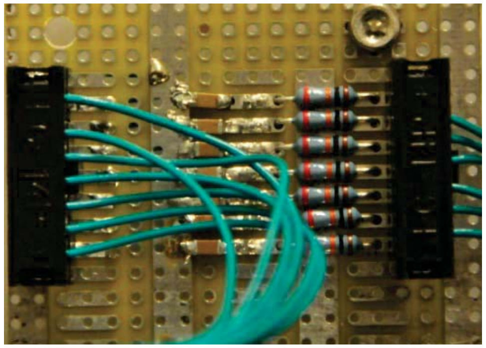
Small size, light weight, strong, linear output, almost no drifting



3 axis strain gauge force sensor
(Bokam Engineering Inc)



Analog amplifier



Analog low pass filter (RC)

Problem: some high frequency noises
Solution: digital filter → analog filter

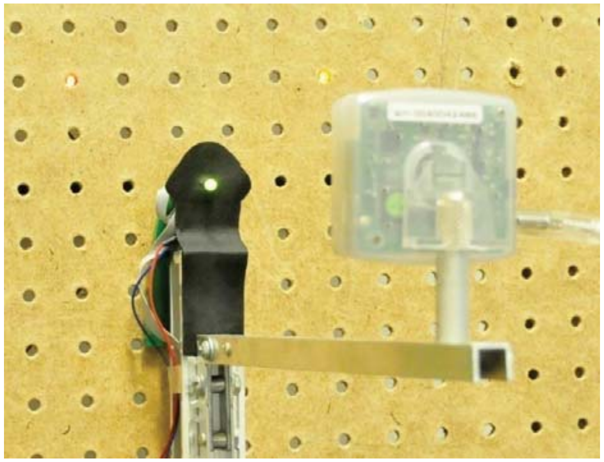
Parameters:

- Max force: 400 (3.50kg)
- Min force: 20 (0.175kg) (to surpass noise)
- Diameter: 28 mm
- Weight: 44 grams

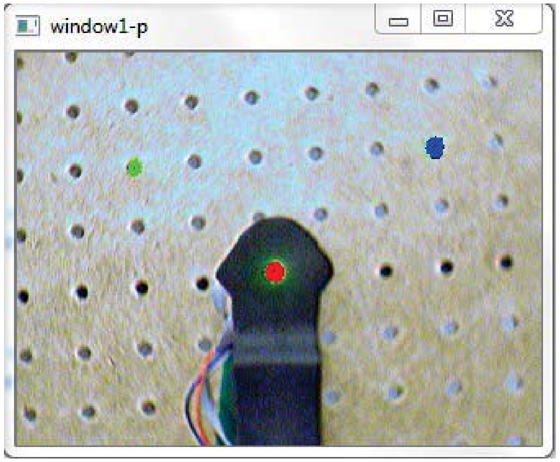
Vision sensor

Where to put the camera(s)? How many do we need?

Fire-i Firewire camera
(Unibrain)



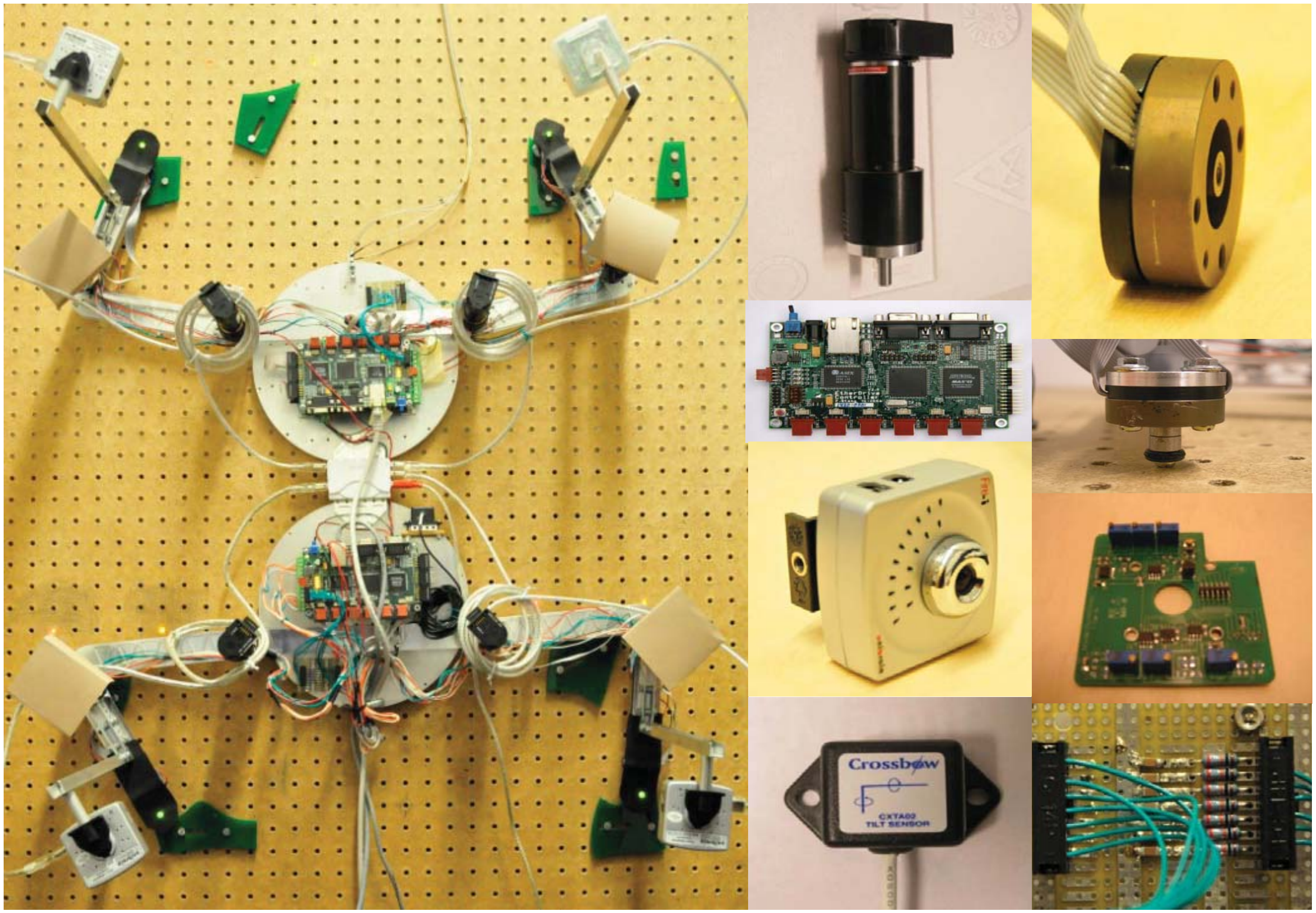
One camera above each limb (finger)
10 inches above finger to see enough arena



LEDs are used to localize holds and finger

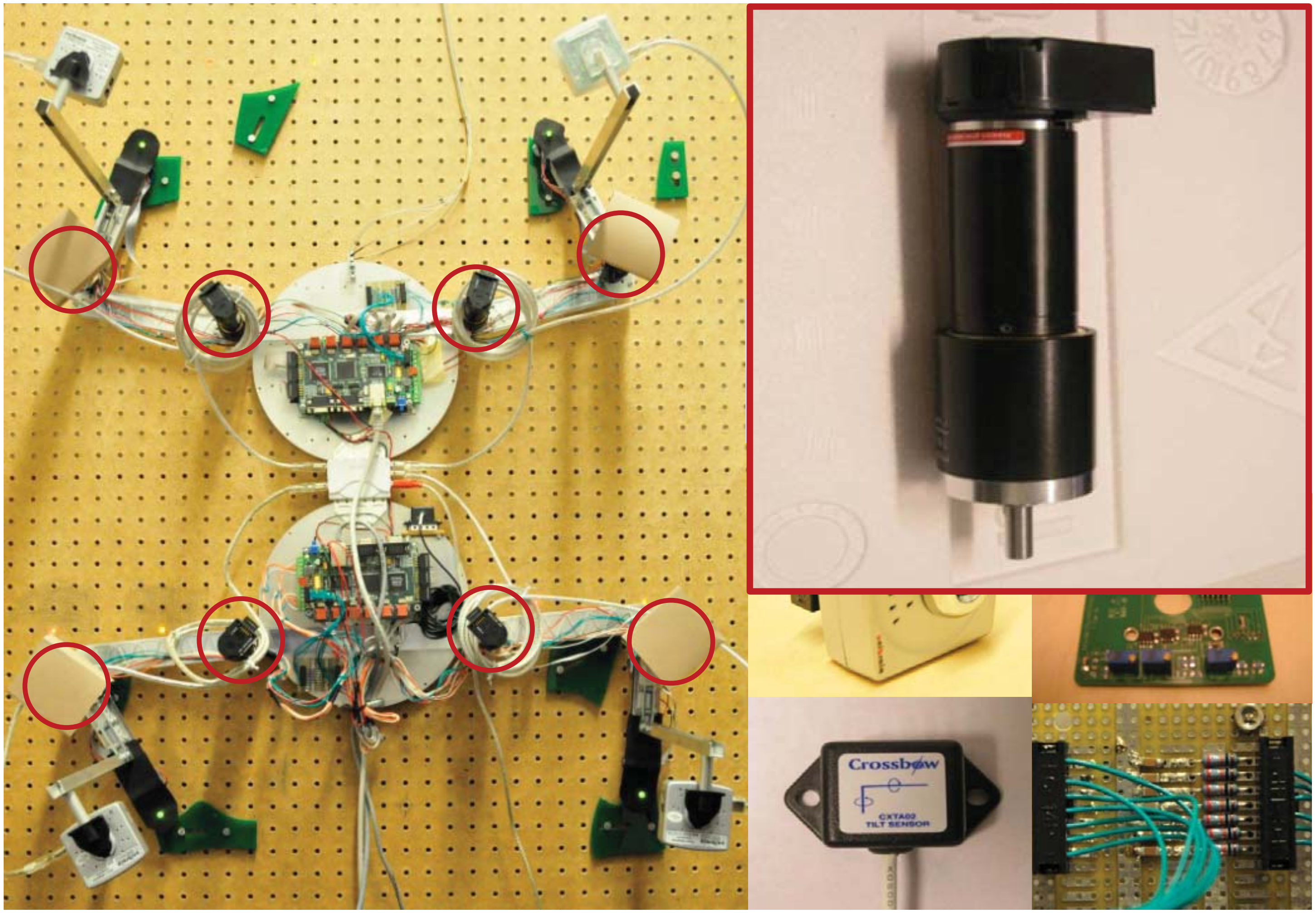
Vision processing program runs parallel to controller program for real time performance
Two programs communicate through shared memory (free limb, actual contact point position)

Robot assembly and testing environment

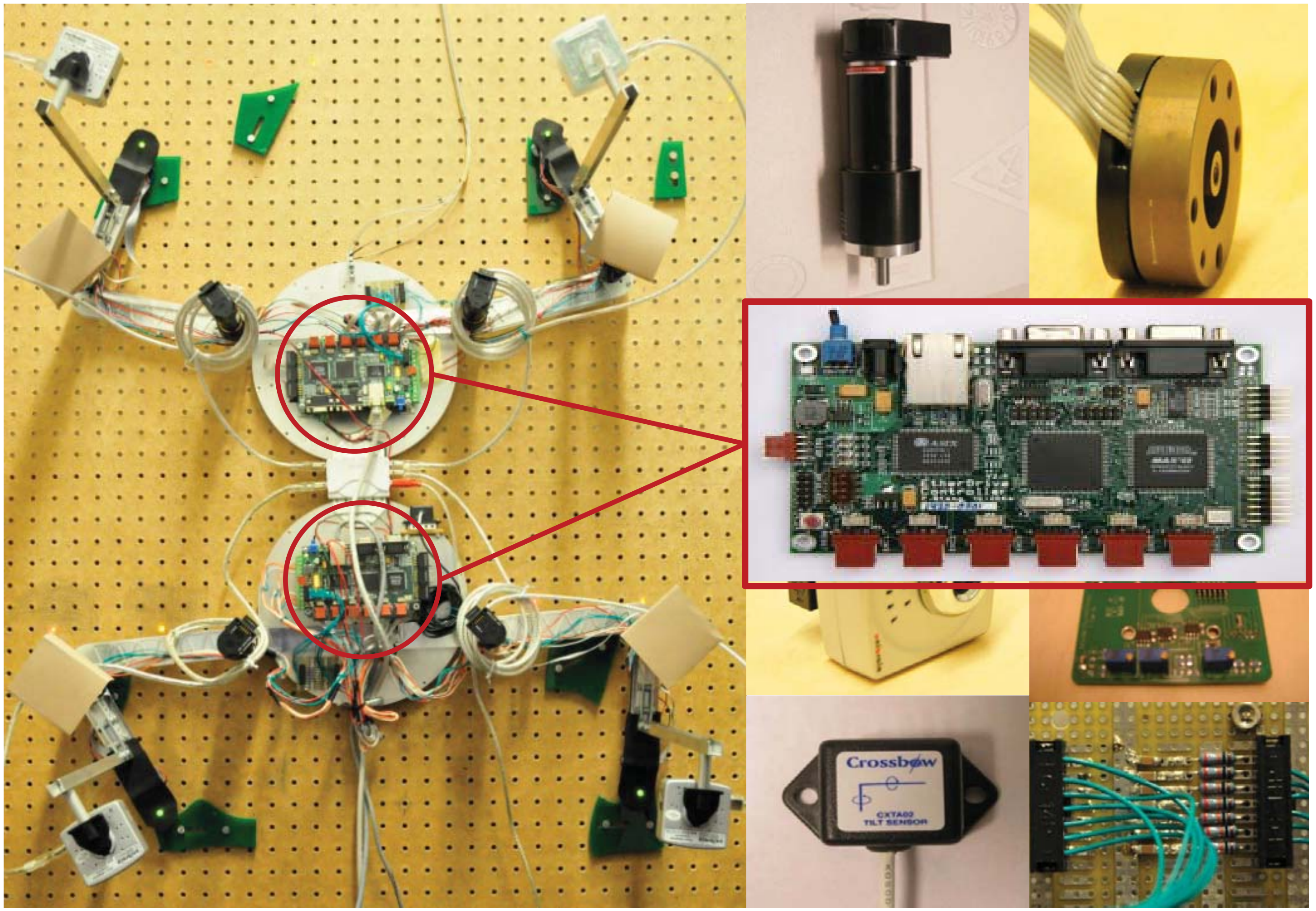


Weight 7kg

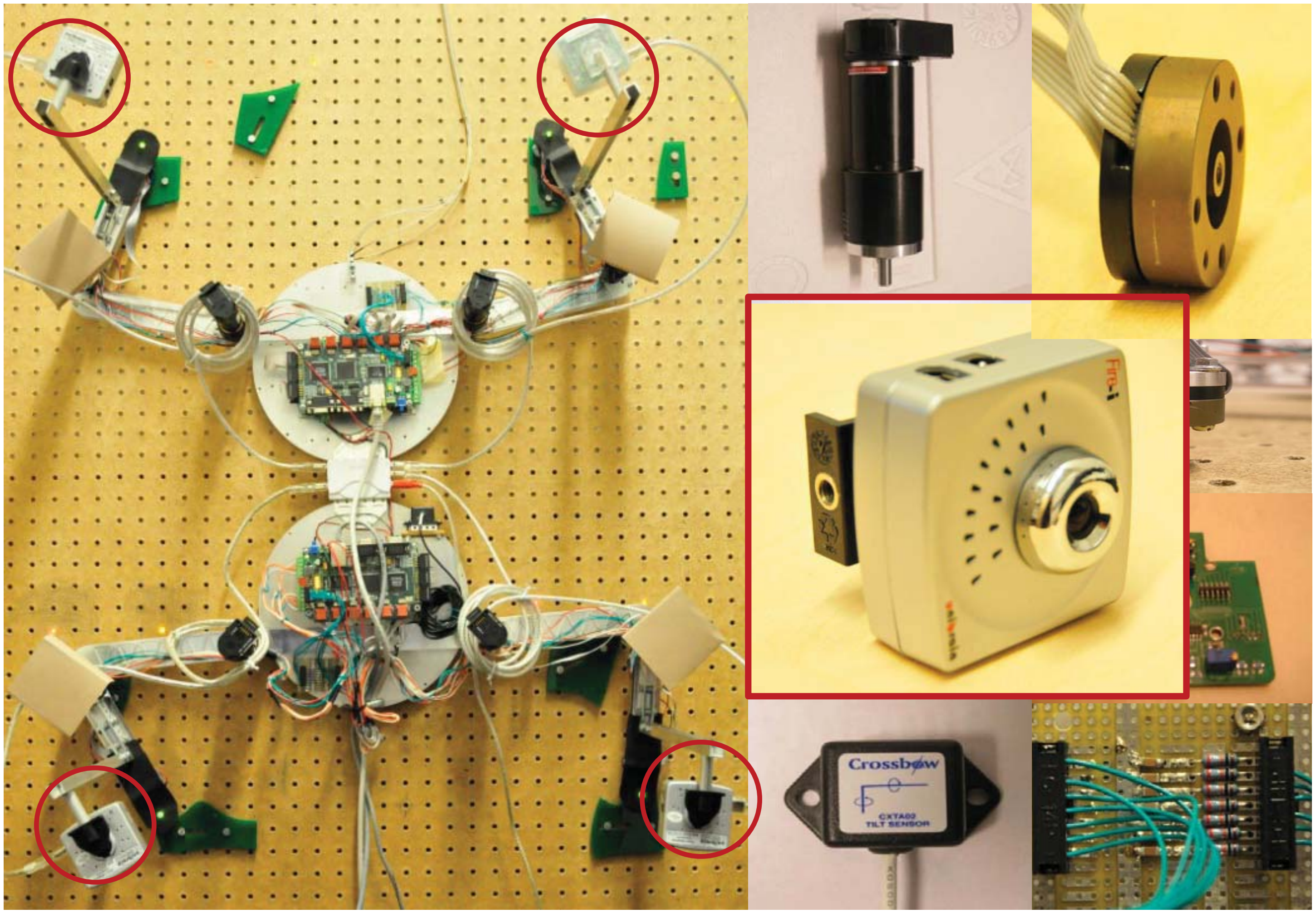
Robot assembly and testing environment



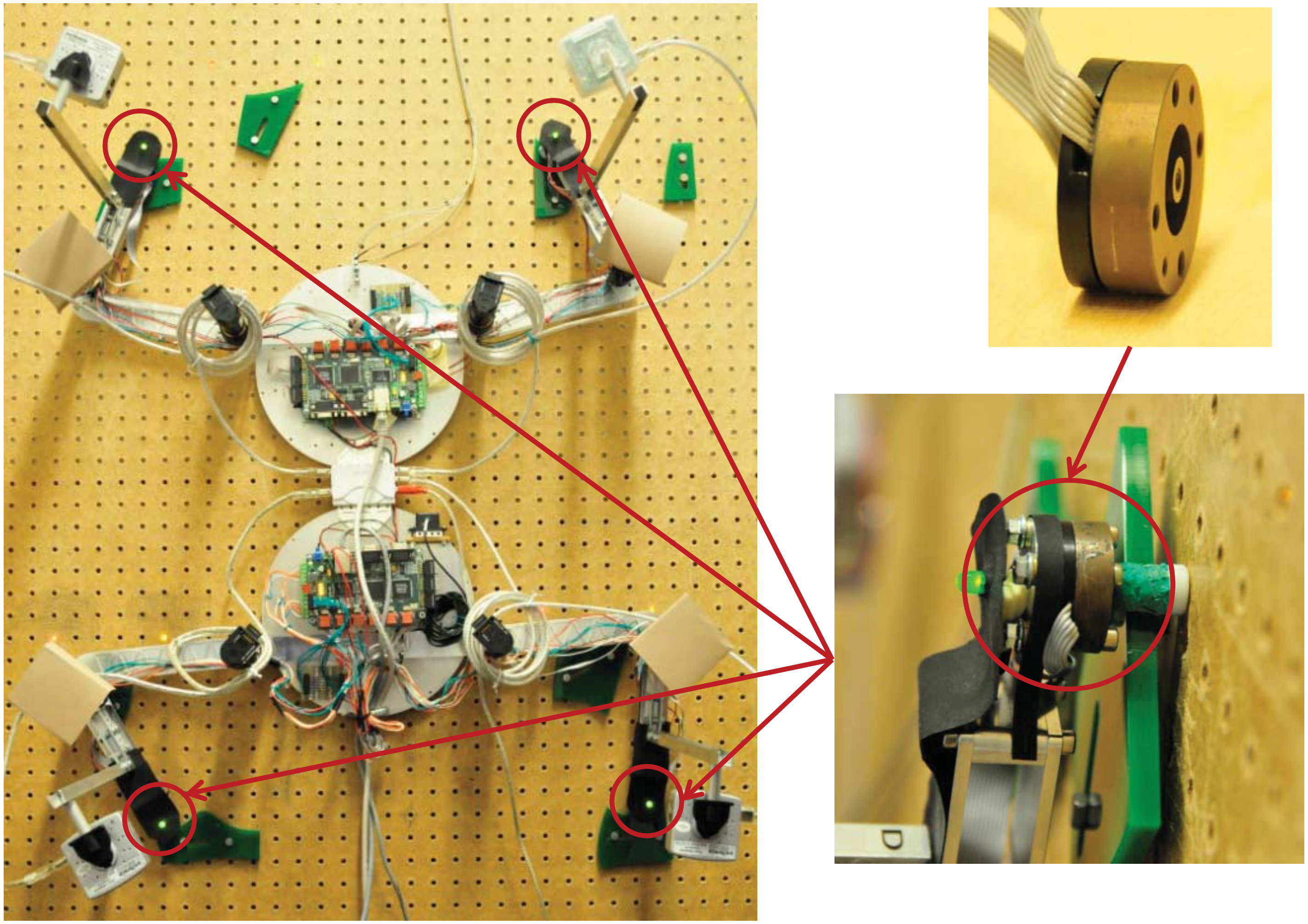
Robot assembly and testing environment



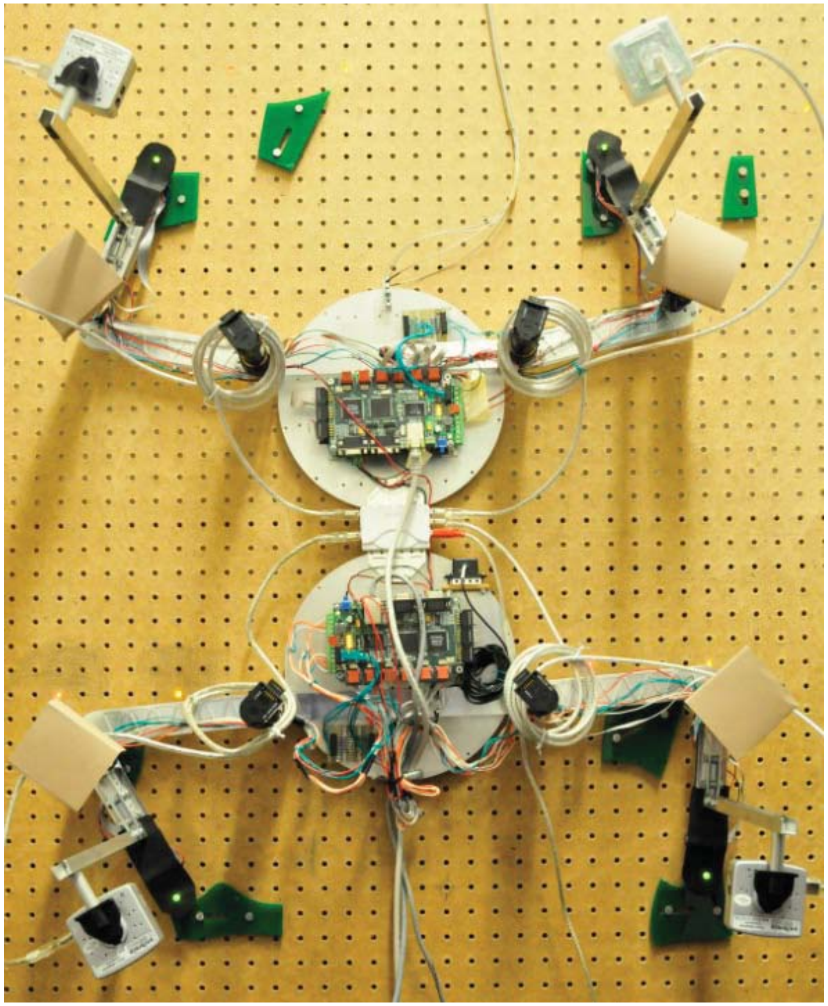
Robot assembly and testing environment

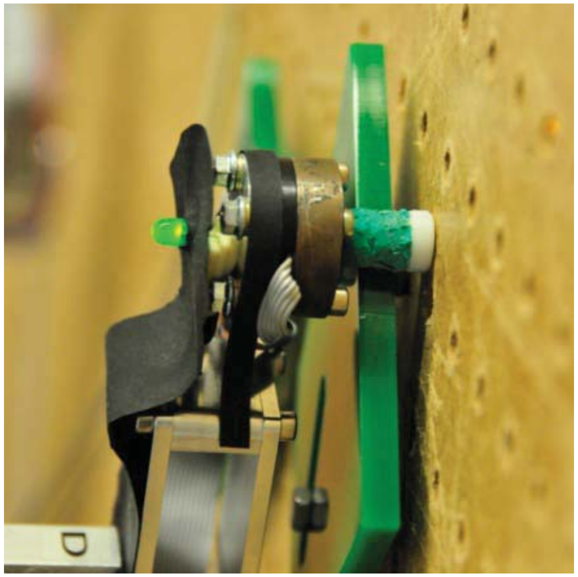


Robot assembly and testing environment



Robot assembly and testing environment

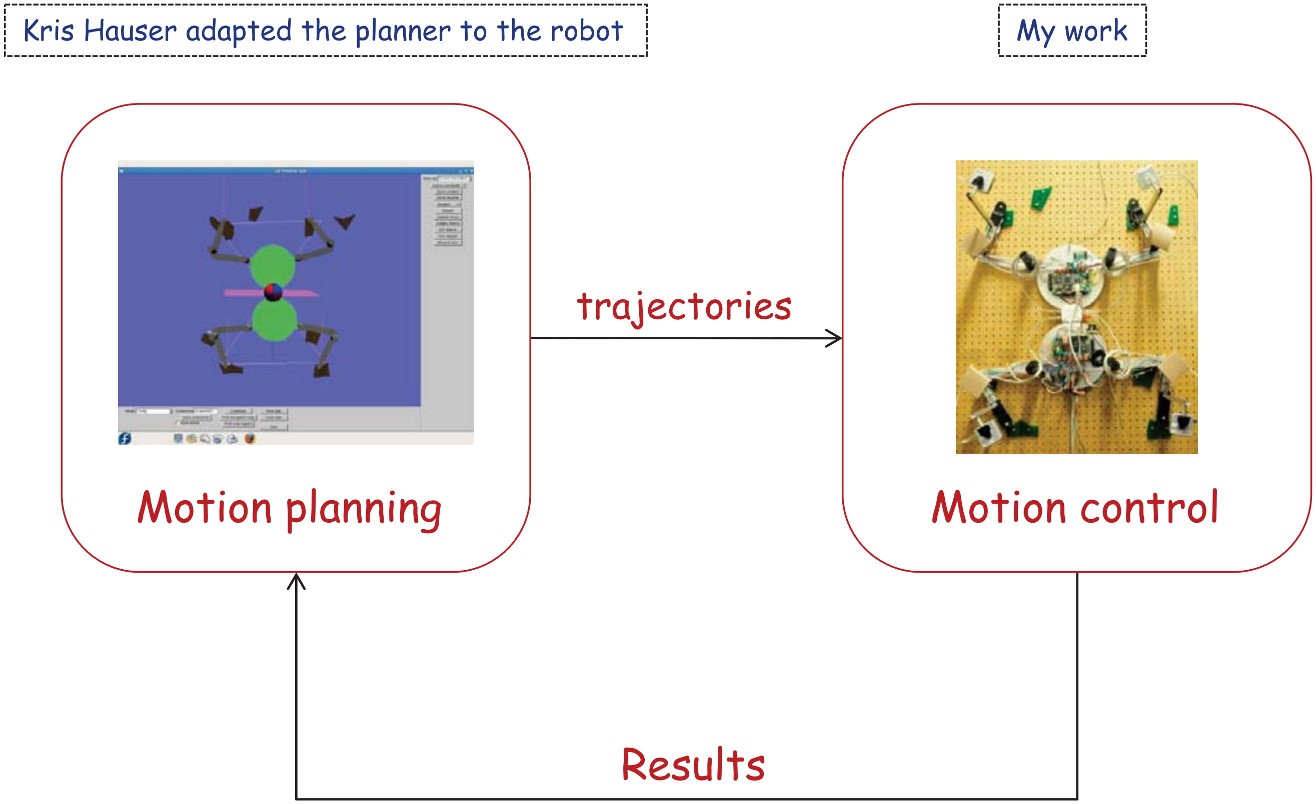




Outline

- Robot design
- Sensors
- Motion control algorithm
- Experiments

Planning and control



Kris Hauser adapted the planner to the robot

My work

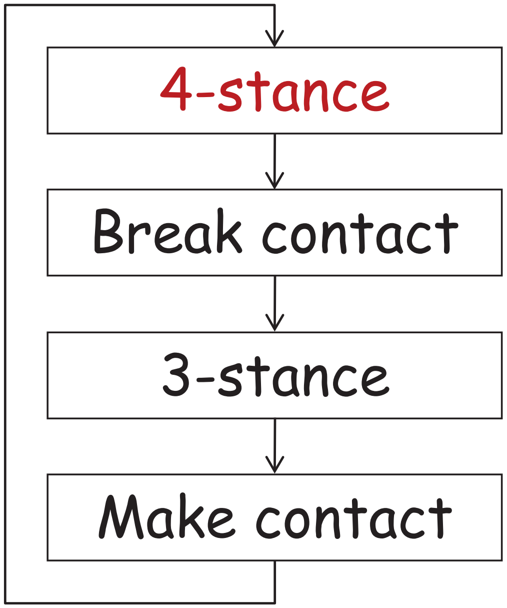
Motion planning

trajectories

Motion control

Results

Stance transition

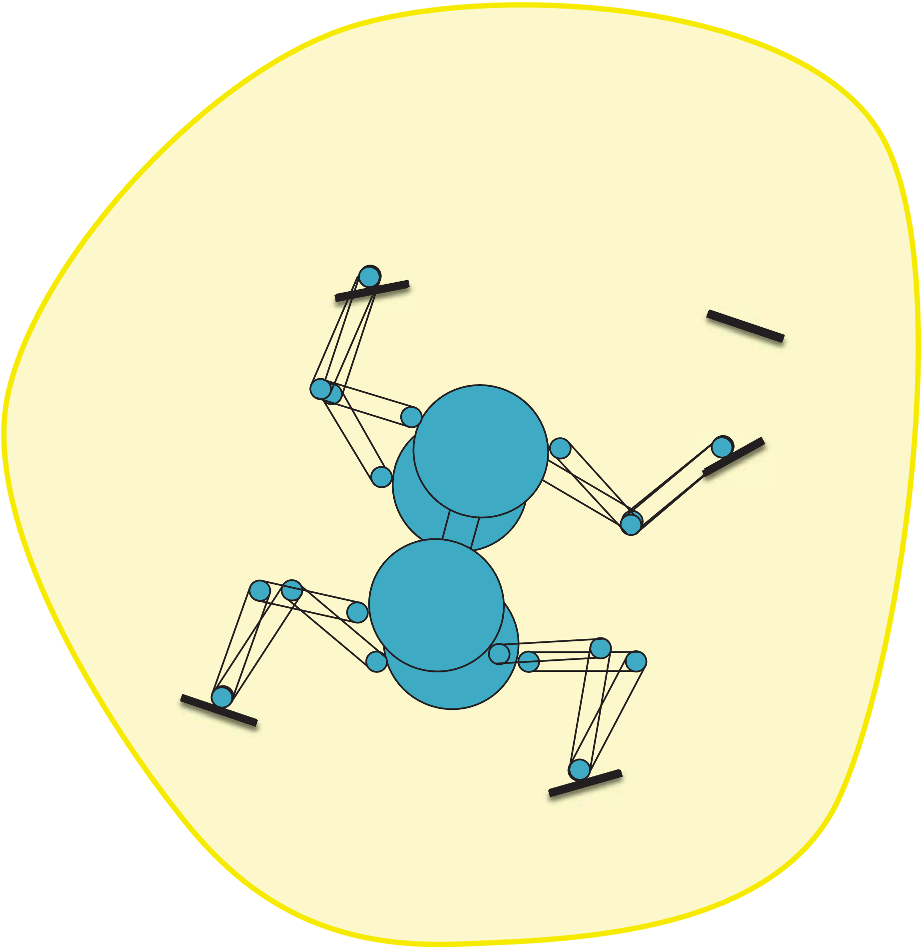


4-stance

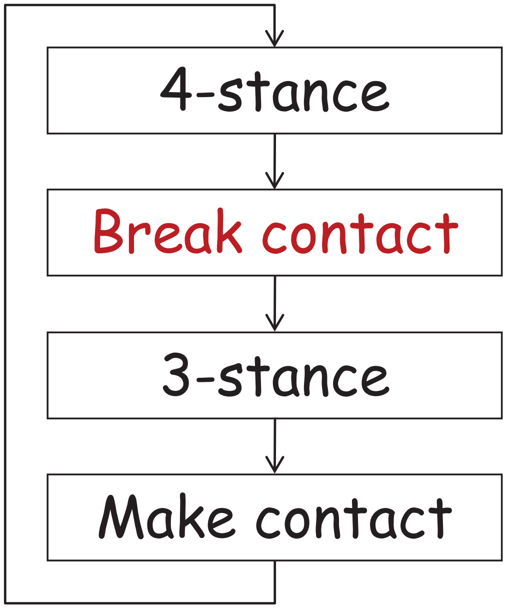
Break contact

3-stance

Make contact



Stance transition

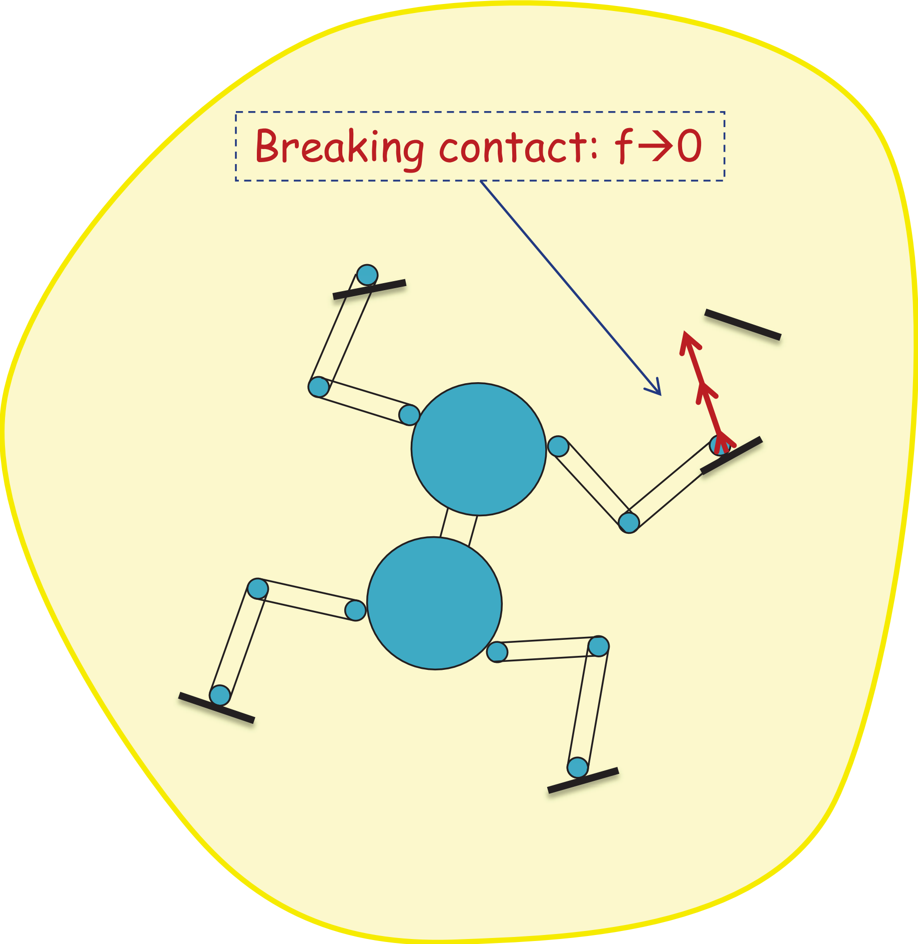


4-stance

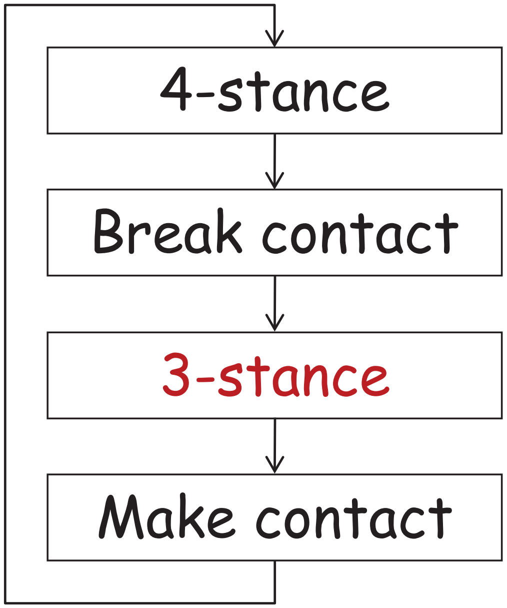
Break contact

3-stance

Make contact



Stance transition

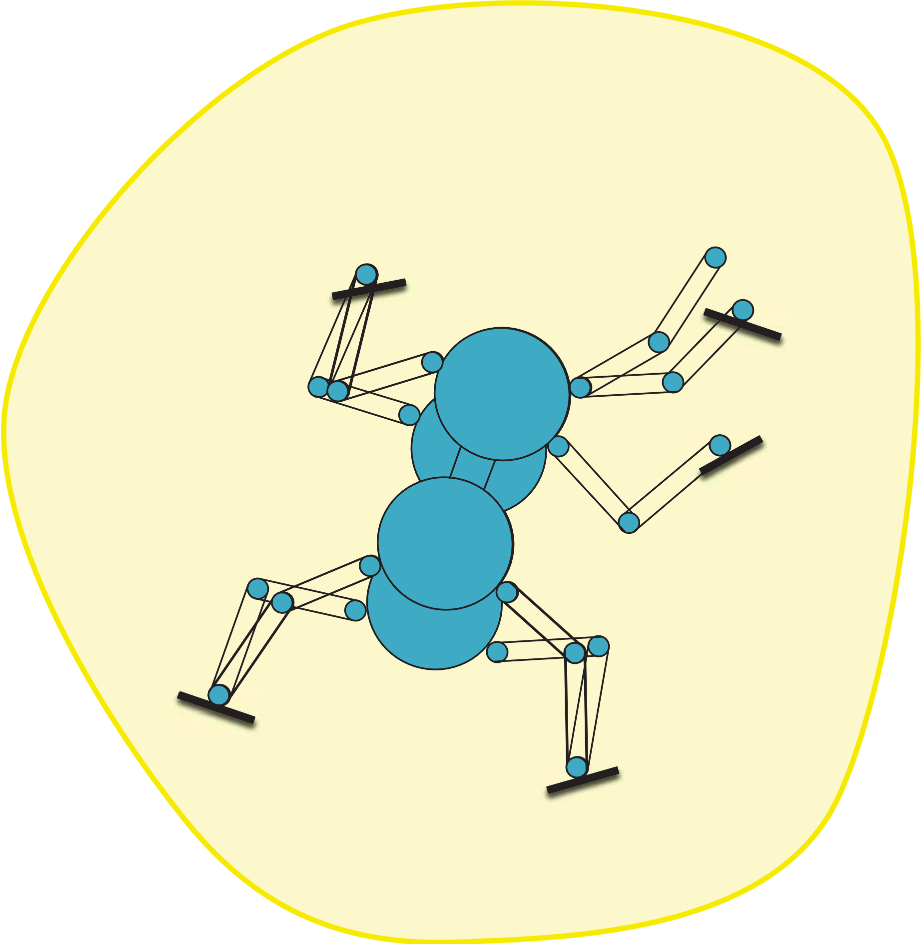


4-stance

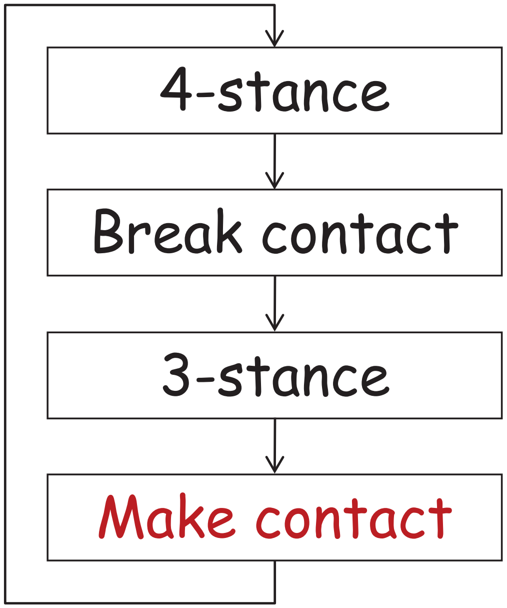
Break contact

3-stance

Make contact



Stance transition

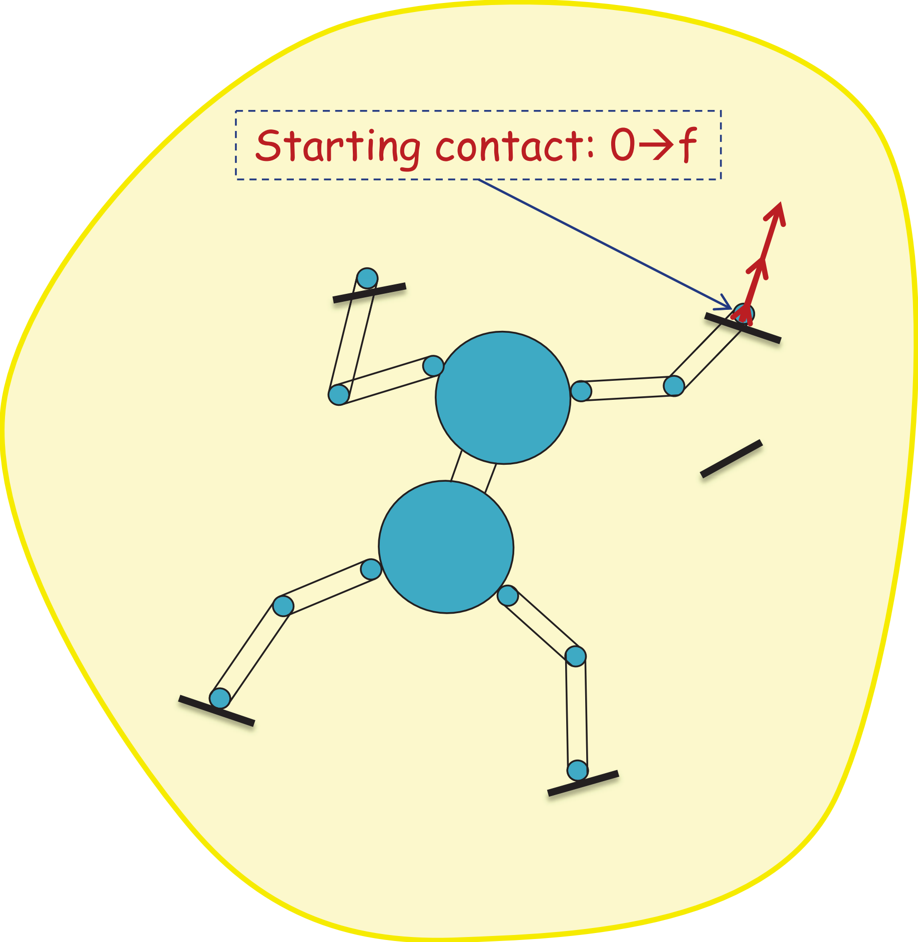


4-stance

Break contact

3-stance

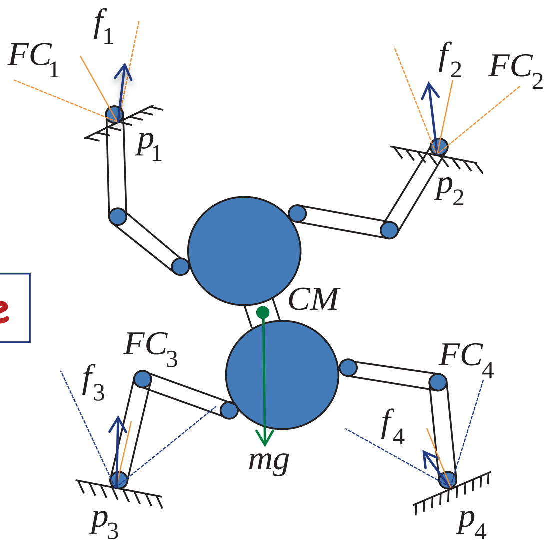
Make contact



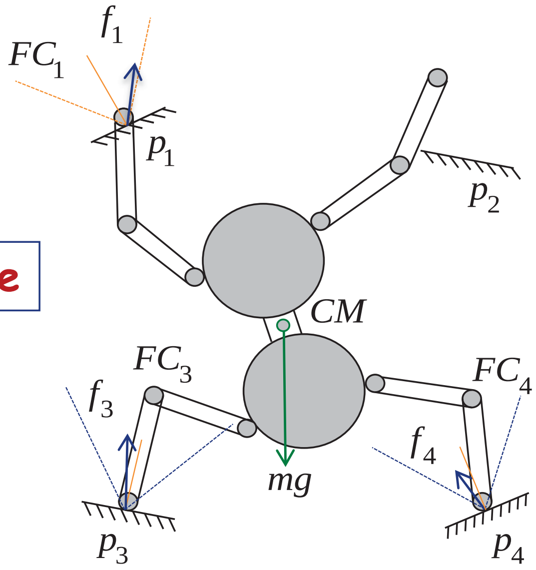
Starting contact: 0→f

Static equilibrium constraints

4-stance



3-stance



Force and torque balance

$$\sum_i f_i + mg = 0$$

$$\sum_i p_i \times f_i + \mathrm{CM}(q) \times mg = 0$$

$$f_i \in \mathrm{FC}_i \text{ for all } i$$

Friction cone

Related work on force control

Robot multiple contact control [Park et al. (2008)]

Force control is achieved with highest priority and motion control is executed using the rest of degree of freedom within the null-space of the force control. Dynamic control structure is used to control each contact force and motion independently.

Convex optimization force control [McGhee et al. (1976), Schelgl et al. (2001), Fujimoto et al. (1998)]

McGhee and Orin were among the first to note that it is possible to use mathematical programming to resolve redundant system. Schelgl et. al. used LPs to optimize forces on a real robot hand even while changing grasps. Fujimoto and Kawamura also used quadratic program to control endpoint forces of a simulated bipedal walking robot.

Control of A Climbing Robot Using Real-time Convex Optimization [Miller et al. (2007)]

This work extends the work of the above researchers, especially work of Fujimoto to climbing robot problem. PD control to generate desired force on body center and Convex Optimization is used to decide the torque on each limb.

Basic tasks of motion control

- Follow the planned trajectories

- Control contact forces
  - Keep quasi-static equilibrium
  - Avoid exceeding force(torque) limit

Objective:
Follow trajectory, NOT achieve specific forces

2-stage motion control

Posture transition control

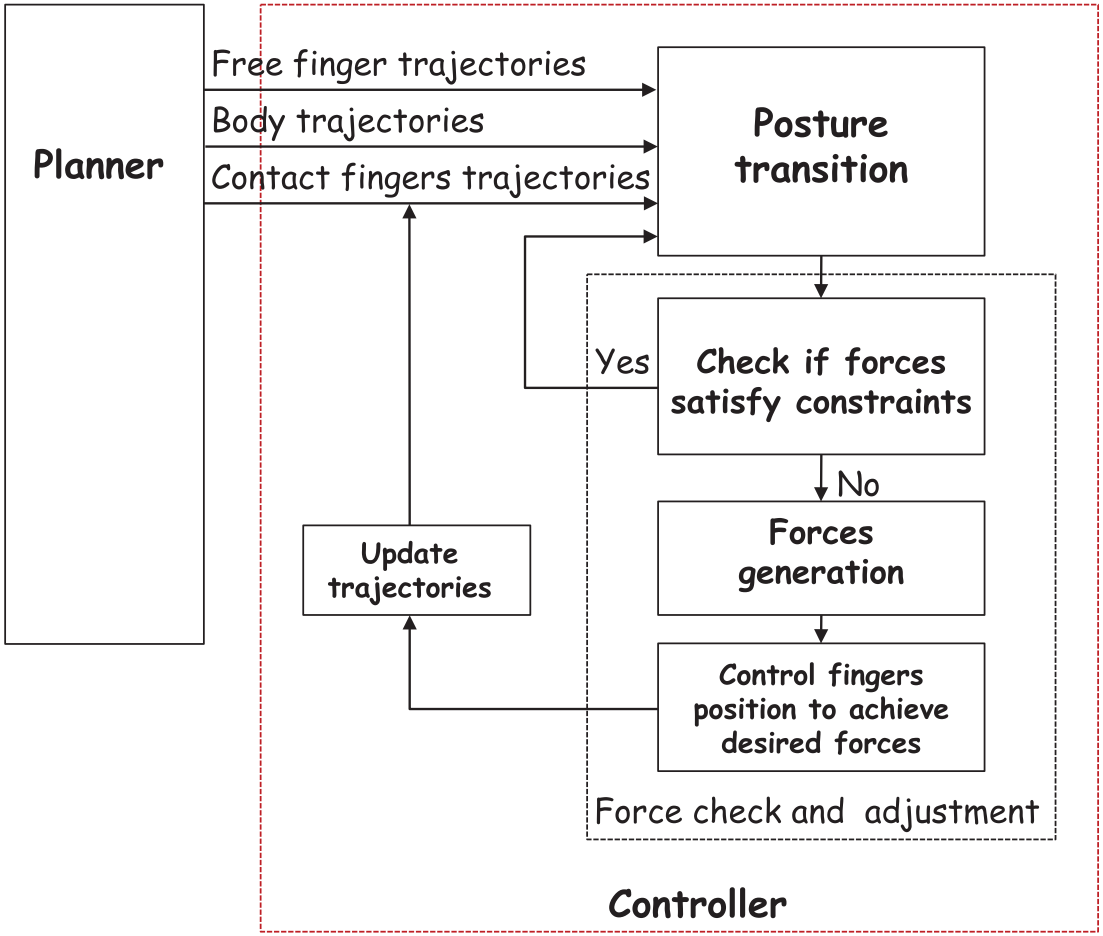
- Follow the planned trajectories

Force checking (and transition)

- Control contact forces
  - Keep quasi-static equilibrium
  - Avoid exceeding force(torque) limit

Use position control to follow trajectories and do occasional force adjustment only when needed

2-stage motion control block diagram



Planner

Free finger trajectories

Body trajectories

Contact fingers trajectories

Posture transition

Yes

Check if forces satisfy constraints

No

Forces generation

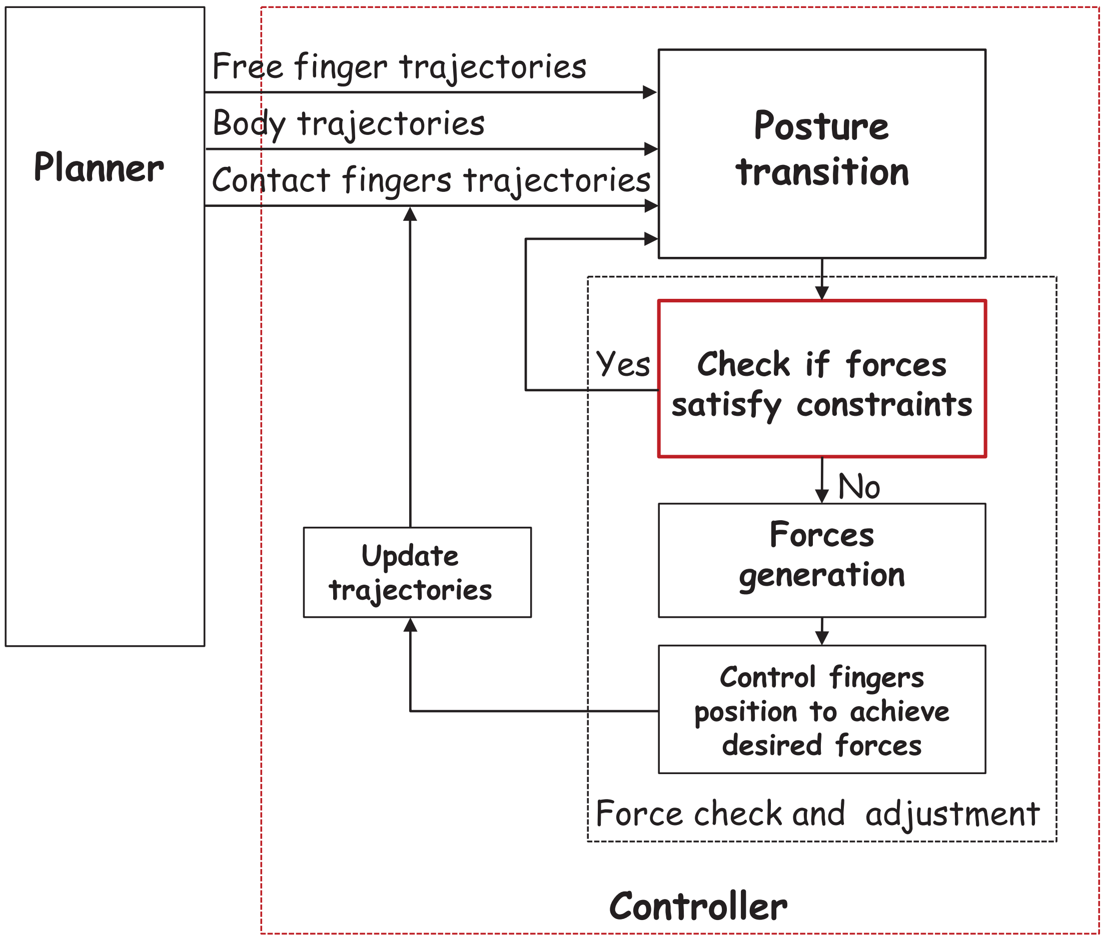
Control fingers position to achieve desired forces

Update trajectories

Force check and adjustment

Controller

2-stage motion control block diagram



Planner

Free finger trajectories

Body trajectories

Contact fingers trajectories

Posture transition

Check if forces satisfy constraints

Yes

No

Forces generation

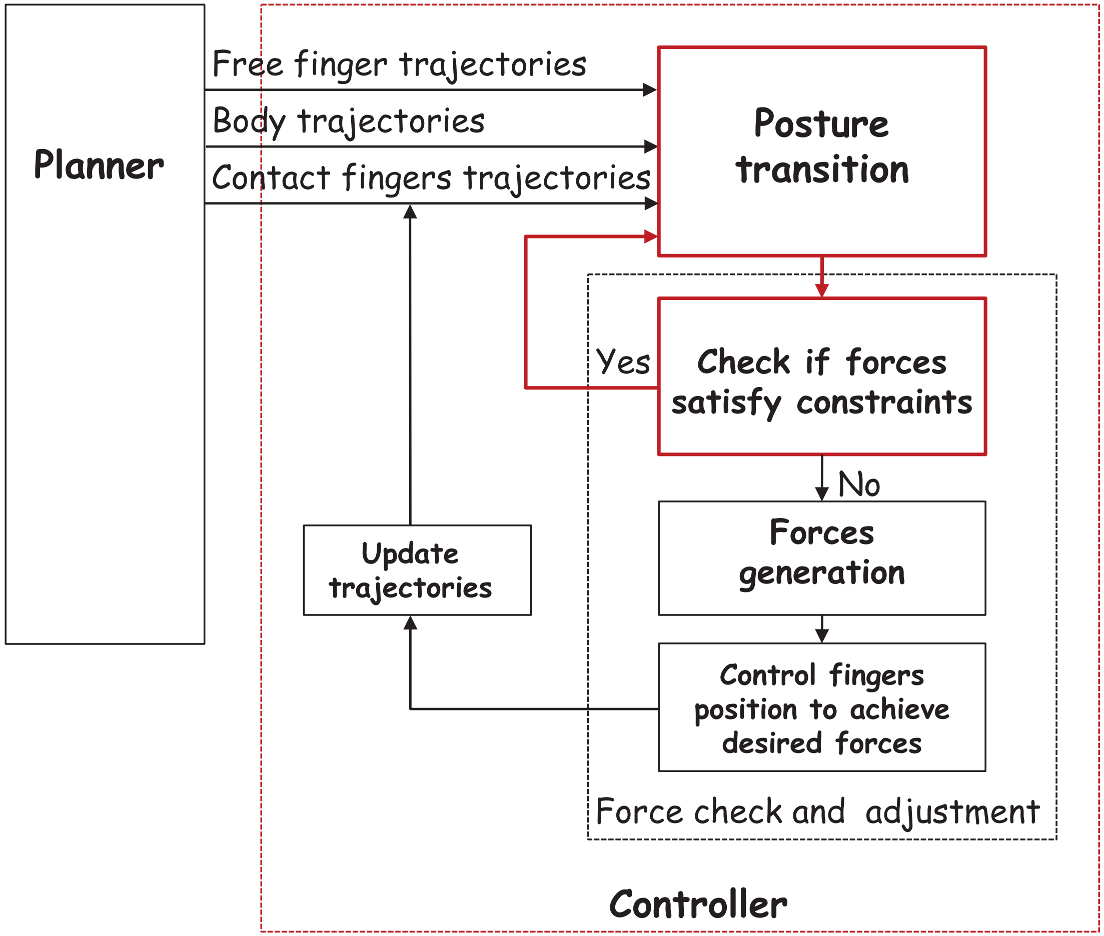
Control fingers position to achieve desired forces

Update trajectories

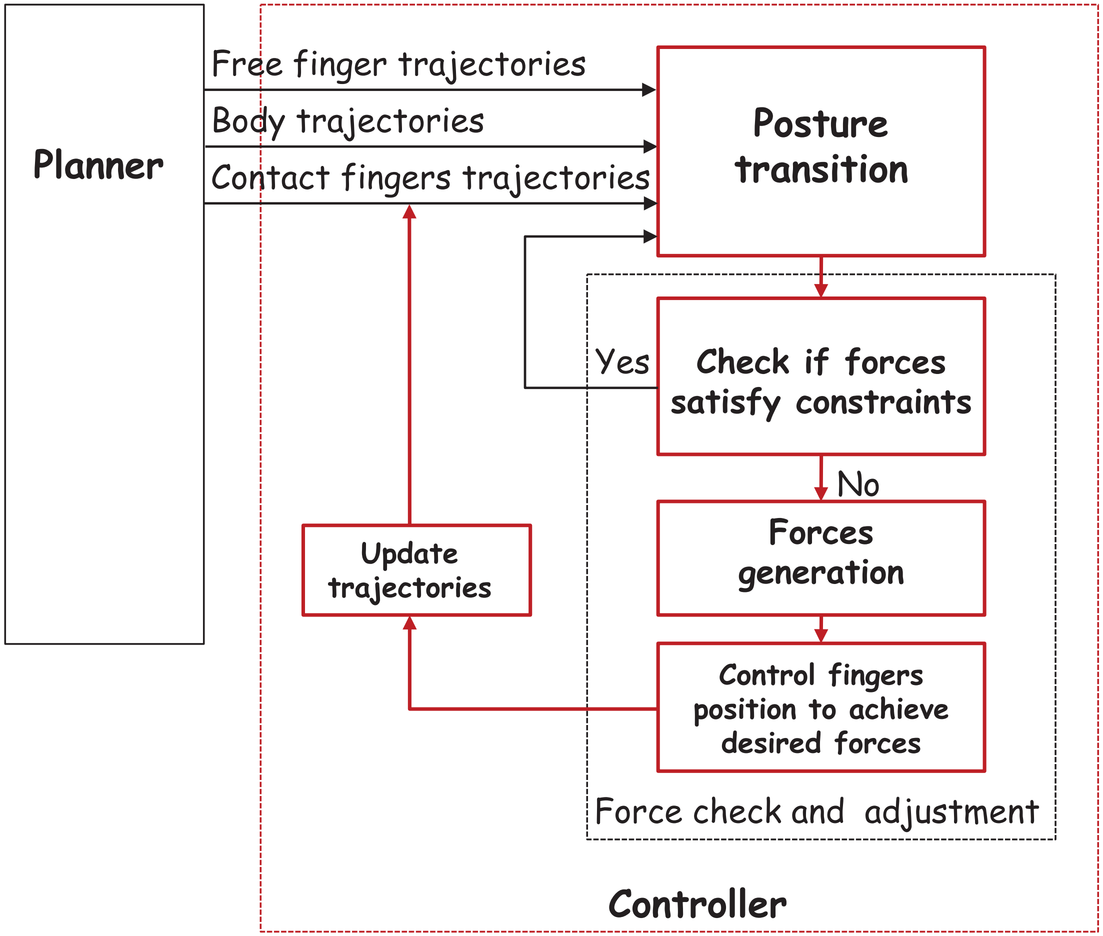
Force check and adjustment

Controller

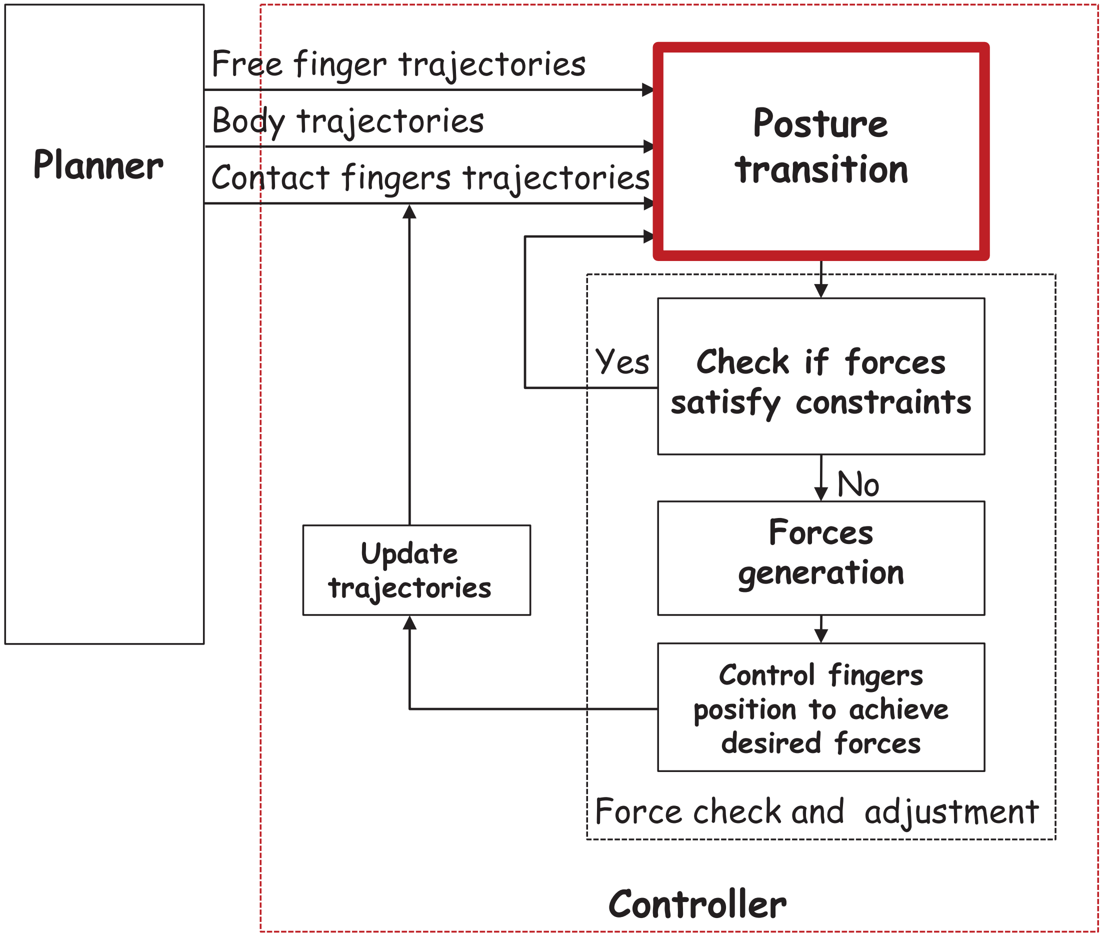
2-stage motion control block diagram



2-stage motion control block diagram



2-stage motion control block diagram



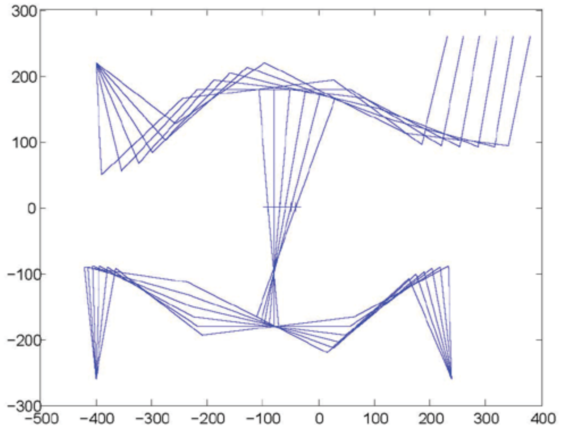
Posture transition

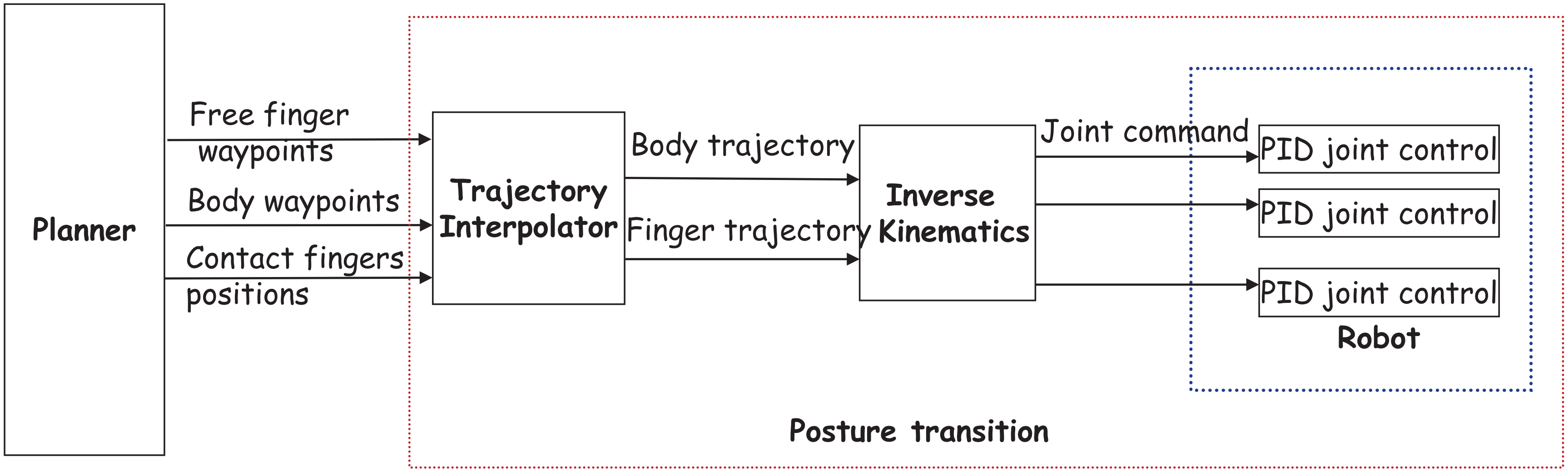
Feedback in Cartesian space

Input: Cartesian space trajectories generated by planner

- 4-stance
  - Body trajectory (position and orientation)
- 3-stance
  - Body trajectory
  - Free finger trajectory

Output: Joint trajectories





Planner

Free finger waypoints

Body waypoints

Contact fingers positions

Trajectory Interpolator

Body trajectory

Finger trajectory

Inverse Kinematics

Joint command

PID joint control

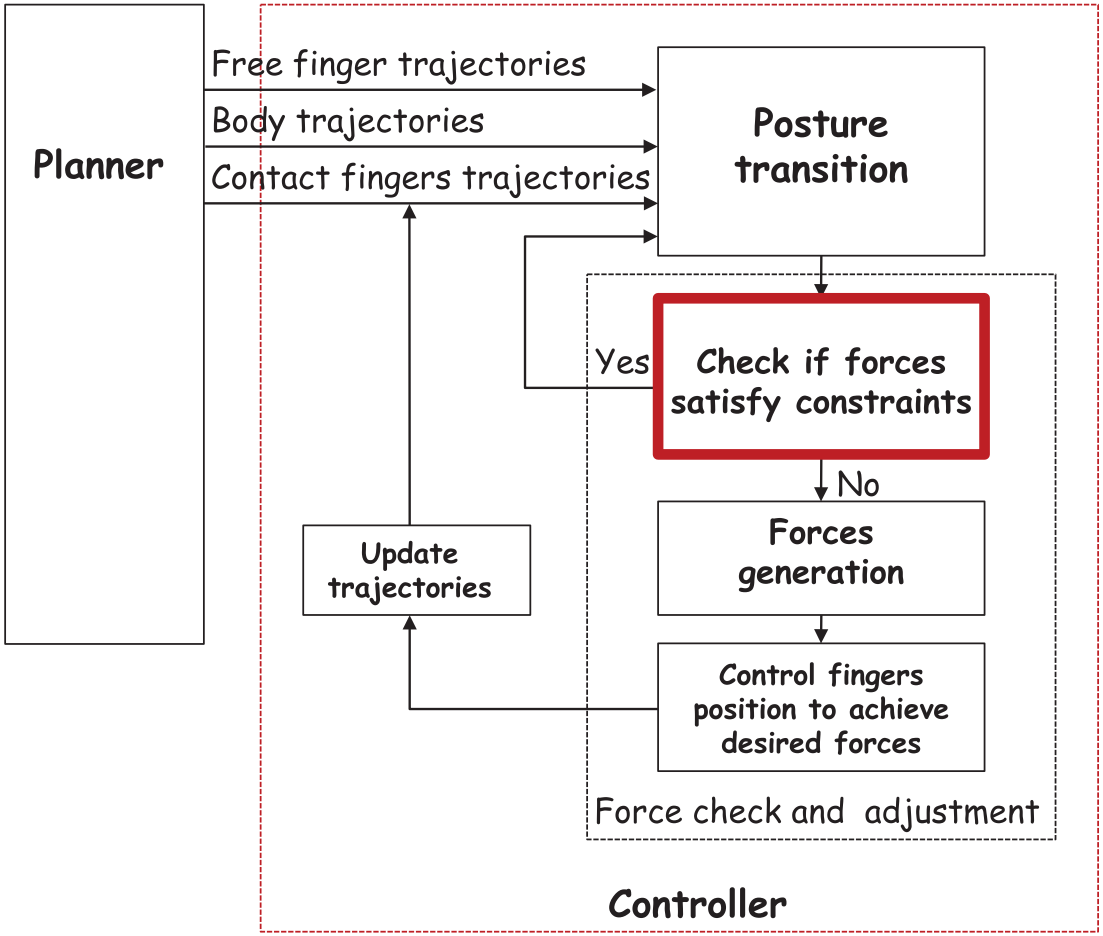
PID joint control

PID joint control

Robot

Posture transition

2-stage motion control block diagram



Force constraints

Quasi-static equilibrium constraints:

$$\sum_i f_i + mg = 0$$

$$\sum_i p_i \times f_i + \mathrm{CM}(q) \times mg = 0$$

$$f_i \in \mathrm{FC}_i \text{ for all } i$$

Force limits:

$$f_{lower} < f_i < f_{upper}$$

Safe force region

Quasi-static equilibrium constraints:

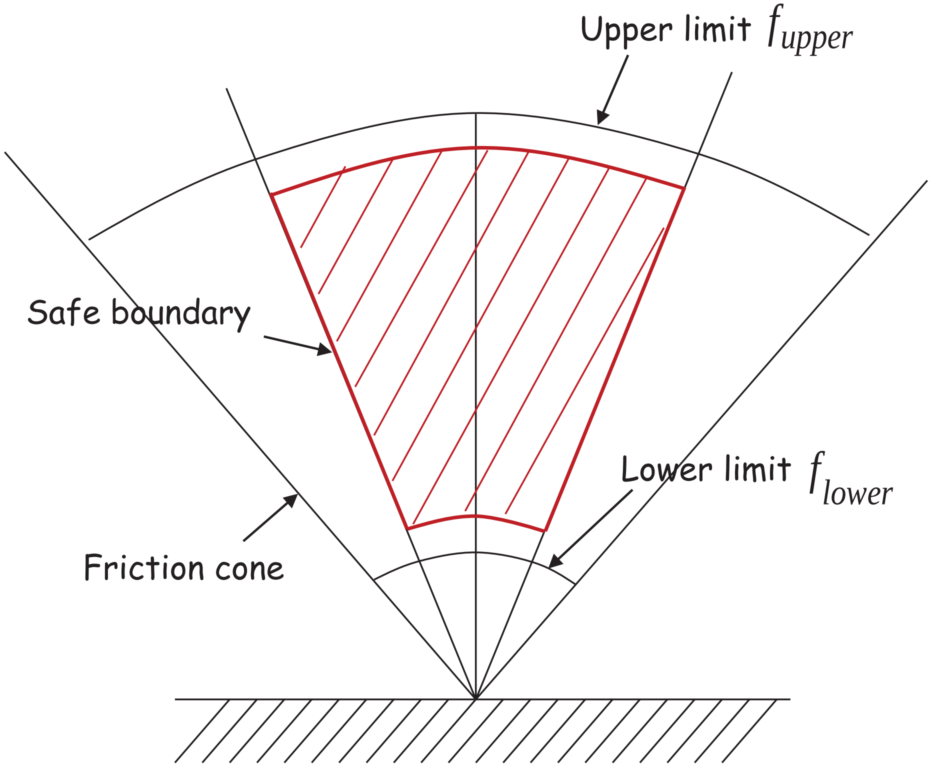
$$\sum_i f_i + mg = 0$$

$$\sum_i p_i \times f_i + \mathrm{CM}(q) \times mg = 0$$

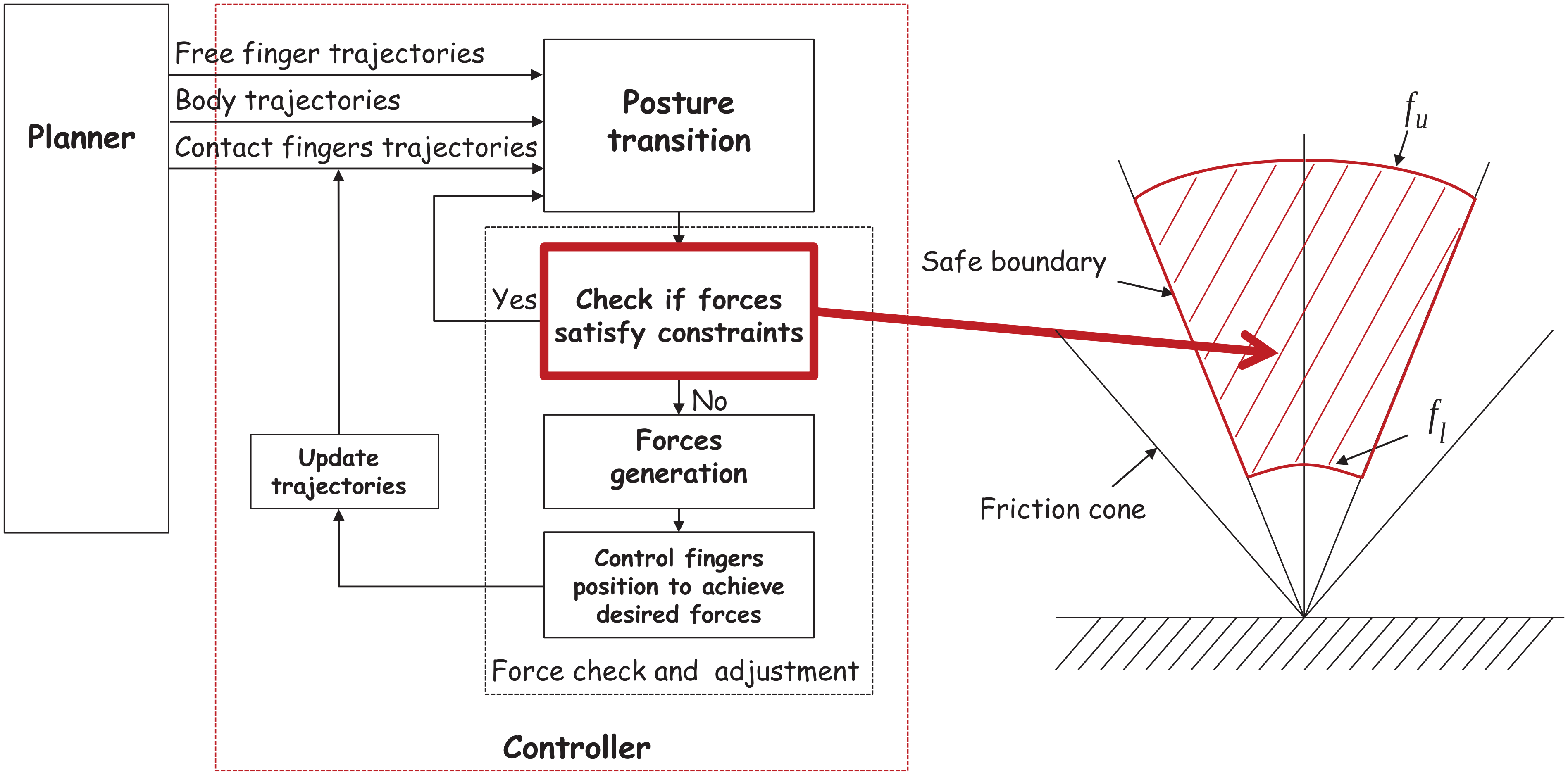
$$f_i \in \mathrm{FC}_i \text{ for all } i$$

Force limits:

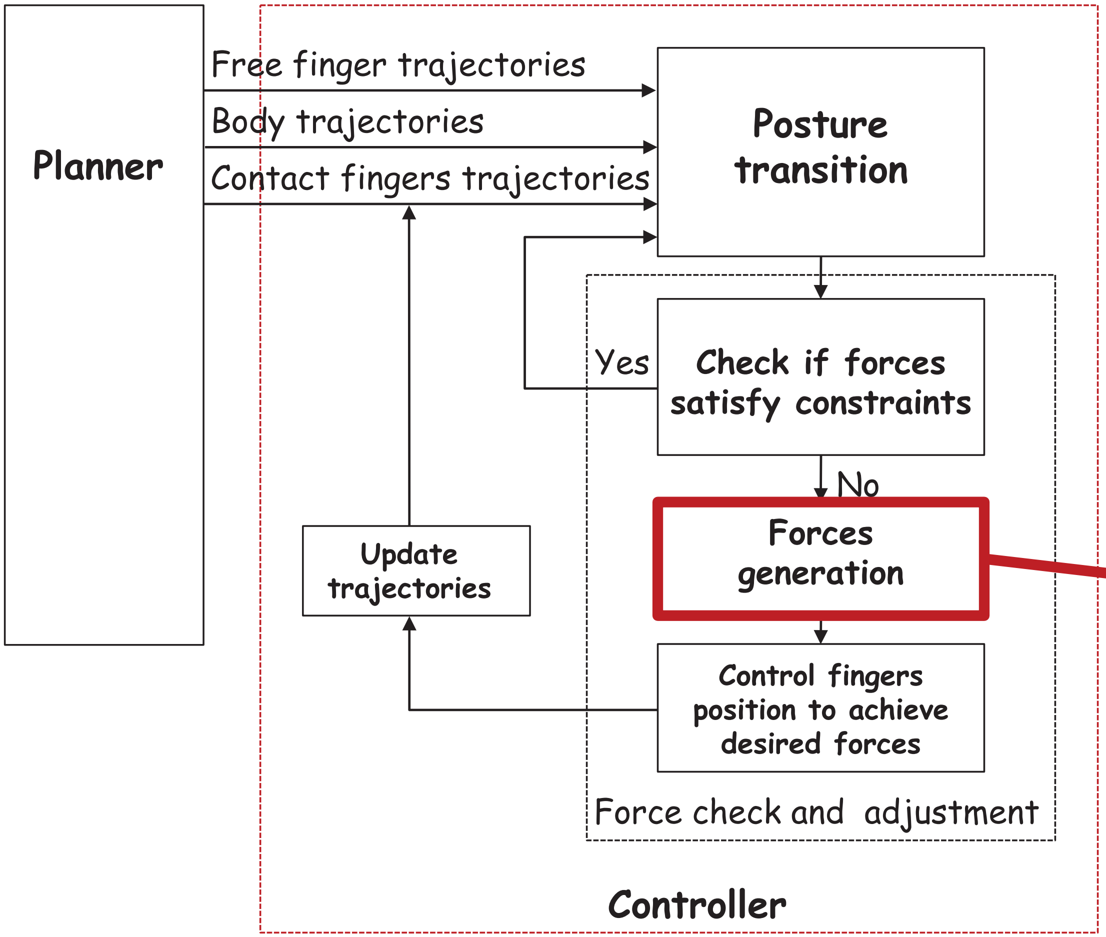
$$f_{lower} < f_i < f_{upper}$$



Forces checking



Forces generation



Planner

Free finger trajectories

Body trajectories

Contact fingers trajectories

Posture transition

Check if forces satisfy constraints

Yes

No

Forces generation

Control fingers position to achieve desired forces

Update trajectories

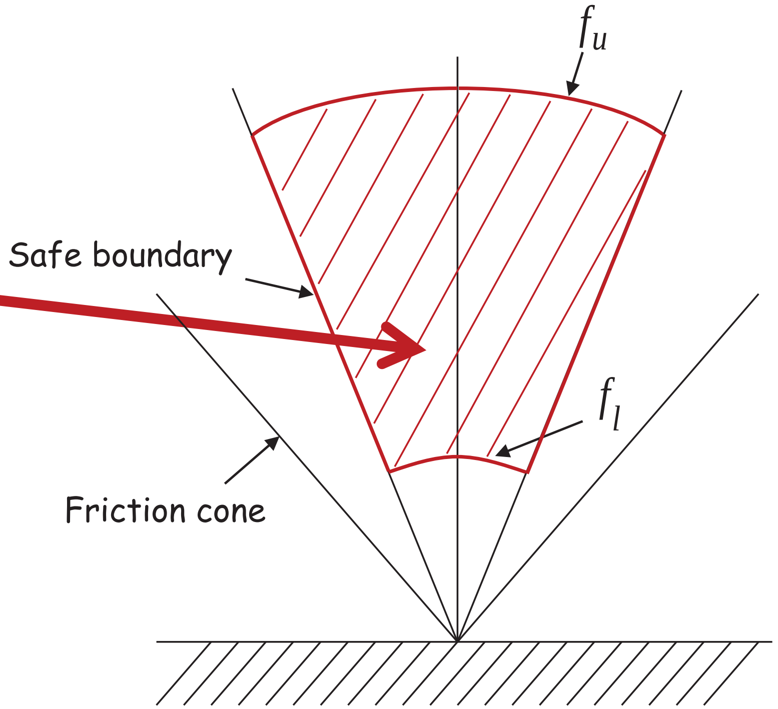
Force check and adjustment

Controller

Step 1: randomly generate free force variables in safe region

Step 2: solve equations of equilibrium constraints for all other force variables

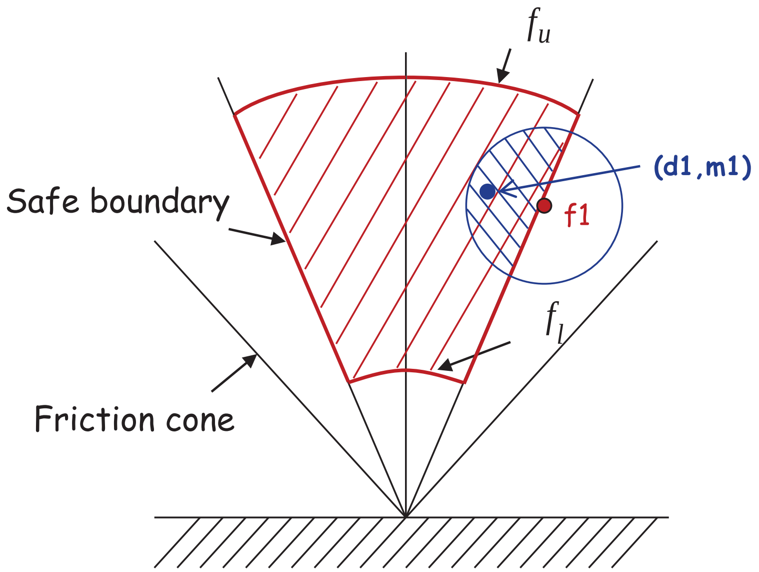
Step 3: check all forces

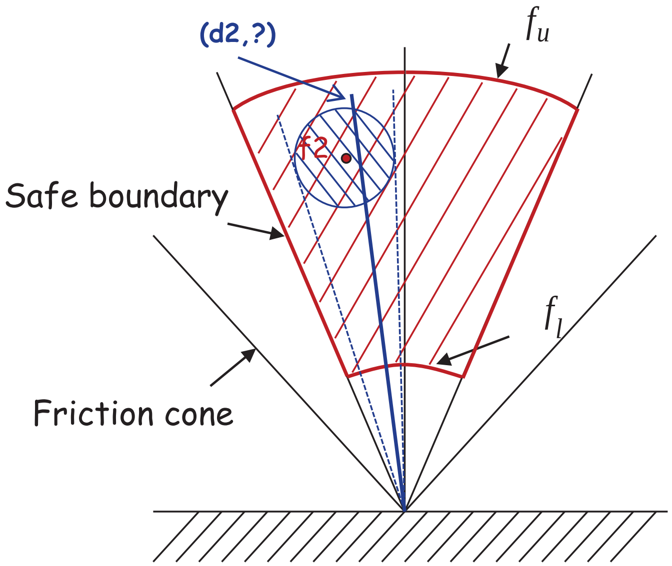


Forces generation

Example for 3-stance force generation:
1) Force is decided by direction and magnitude
2) 3-stance has 6 variables (3 directions, 3 magnitudes)
3) Only 3 constraints (horizontal force, vertical force and torque)

Pick 3 free variables: d1, m1 and d2
Generate these 3 and solve all the other variables





Force transition

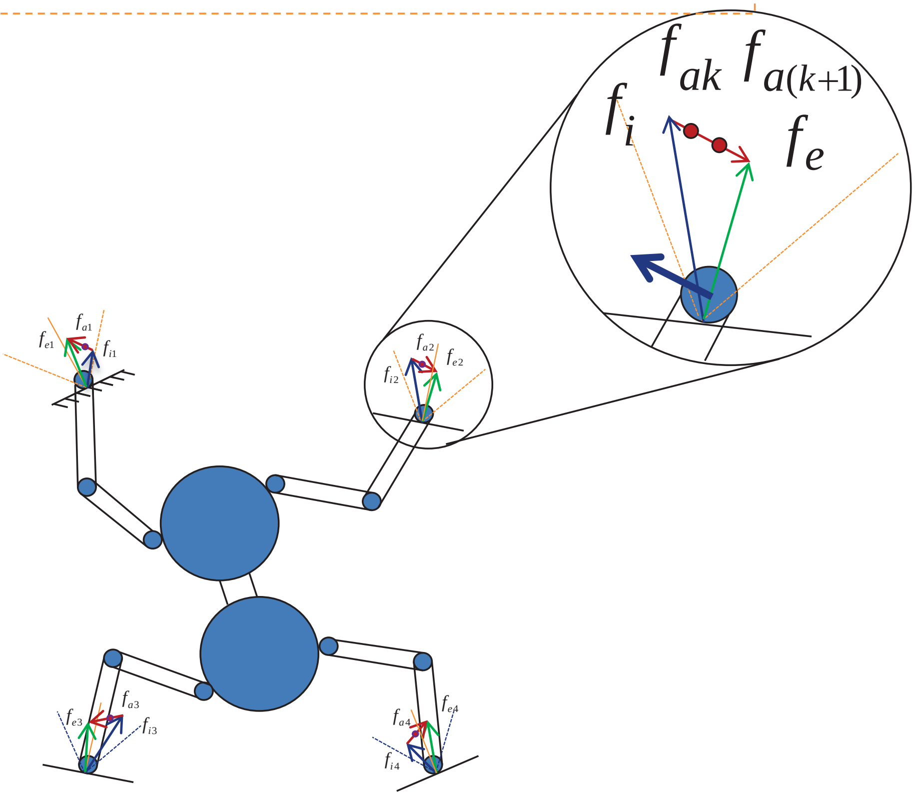
Highly geared system: torque → force is not precise (gearhead friction)
Achieve precise force by finger motion control with force feedback instead of direct joint torque control

Principle: in order to achieve a desired force change, finger should move opposite to the direction of the desired force change and speed is proportional to the force change magnitude

1) Interpolating n points $f_{a1} \ldots f_{an}$ between f_i and f_e
2) To achieve force $f_{a(k+1)}$ from f_{ak}

Finger motion P control:

$$\delta = -K(f' - f_{a(k+1)})$$



Force transition speed: usually less than 100 cycles at 300 Hz, about 0.3 seconds

The interpolation points (f_a) satisfy the quasi-static equilibrium constraints

f_{ak} is interpolation:

$$f_{ak} = f_{ik} + \delta(f_{ek} - f_{ik}), \delta \in [0,1]$$

f_{ik} satisfies constraints

$$\sum_{k=1}^{N} f_{ik} + G = 0$$

$$\sum_{k=1}^{N} f_{ik} \times C_k + G \times CM = 0$$

f_{ek} satisfies constraints

$$\sum_{k=1}^{N} f_{ek} + G = 0$$

$$\sum_{k=1}^{N} f_{ek} \times C_k + G \times CM = 0$$

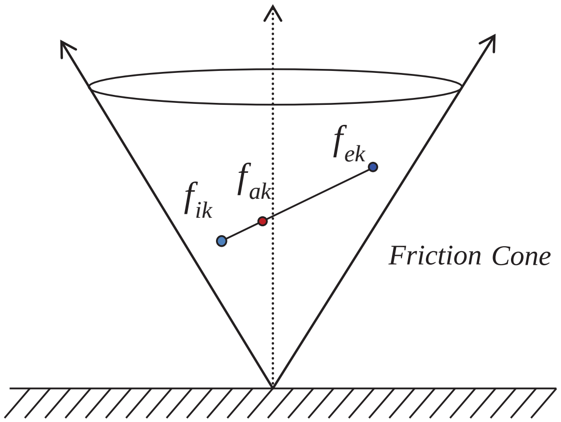
f_{ak} force balance:

$$\sum_{k=1}^{N} f_{ak} + G = \sum_{k=1}^{N} [f_{ik} + \delta(f_{ek} - f_{ik})] = \sum_{k=1}^{N} f_{ik} + \delta[\sum_{k=1}^{N} f_{ek} - \sum_{K=1}^{N} f_{ik}] + G = 0$$

f_{ak} torque balance:

$$\begin{aligned}\sum_{k=1}^{N} f_{ak} \times C_k + G \times CM &= \sum_{k=1}^{N} [f_{ik} + \delta(f_{ek} - f_{ik})] \times C_k + G \times CM \\ &= \sum_{k=1}^{N} f_{ik} \times C_k + \delta(\sum_{k=1}^{N} f_{ek} \times C_k - \sum_{K=1}^{N} f_{ik} \times C_k) + G \times CM = 0\end{aligned}$$

f_{ak} friction cone:



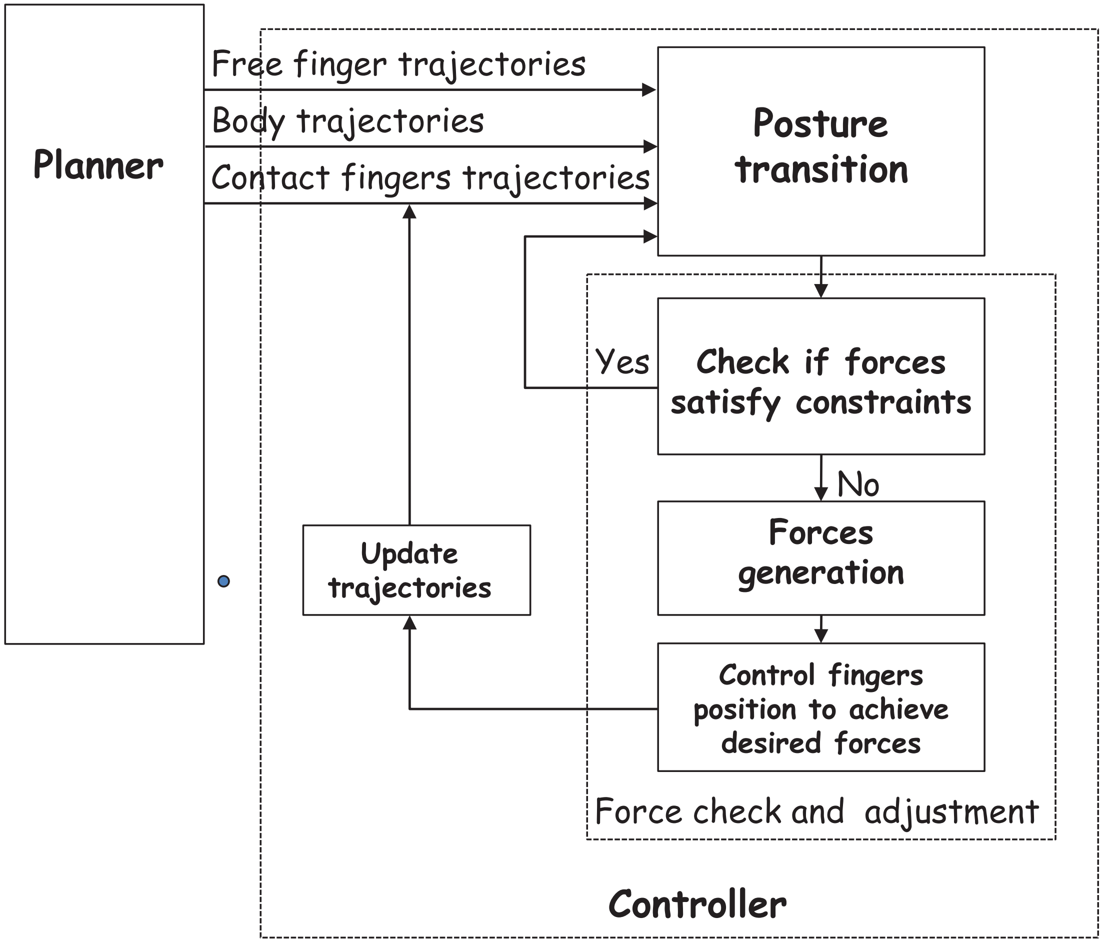
Motion control

- 2-stage control algorithm:
  - Follow the planned trajectories
  - Control contact forces

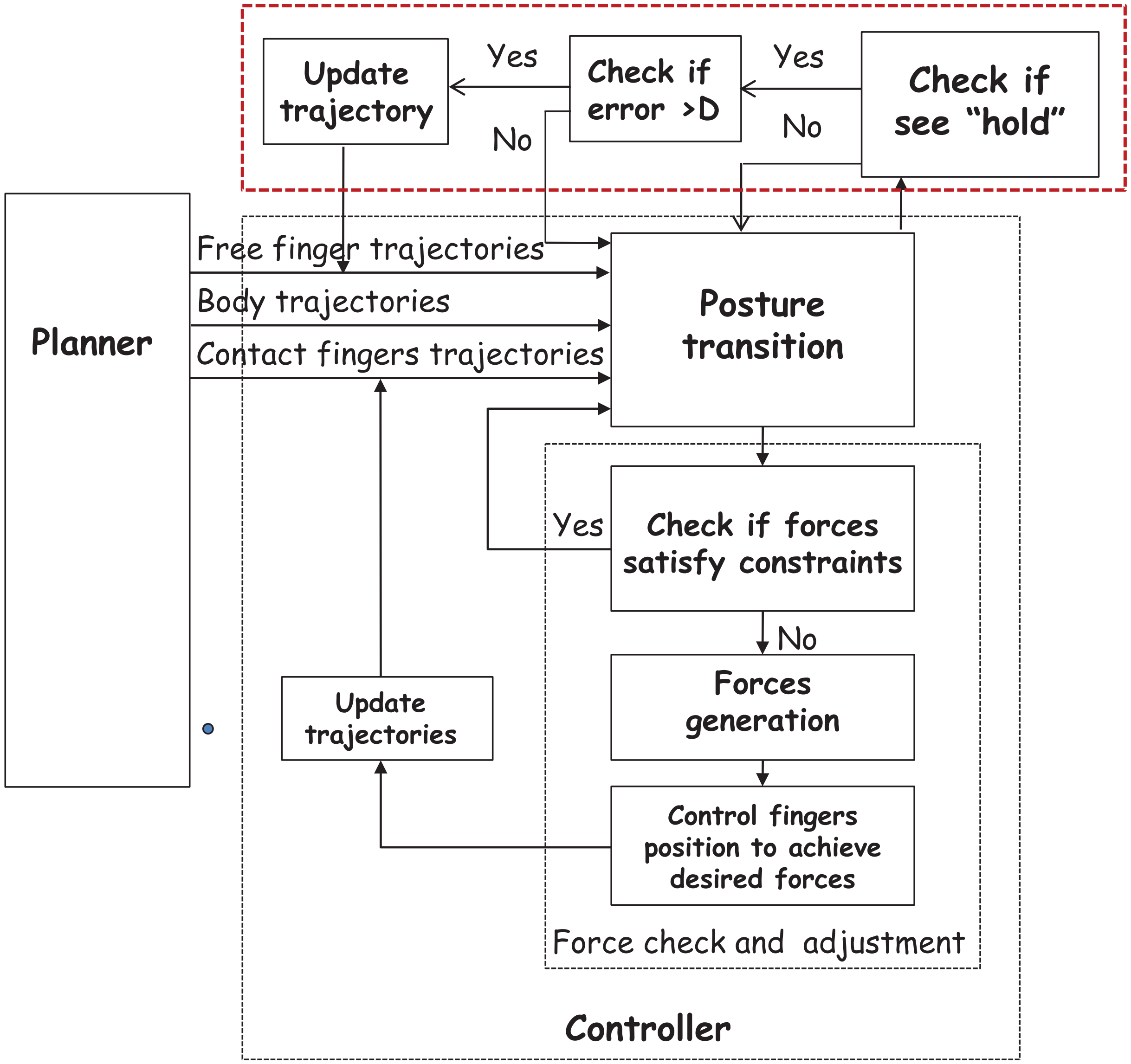
- Docking motion with vision feedback:
  - Navigate finger to desired contact position

Motion control with vision feedback

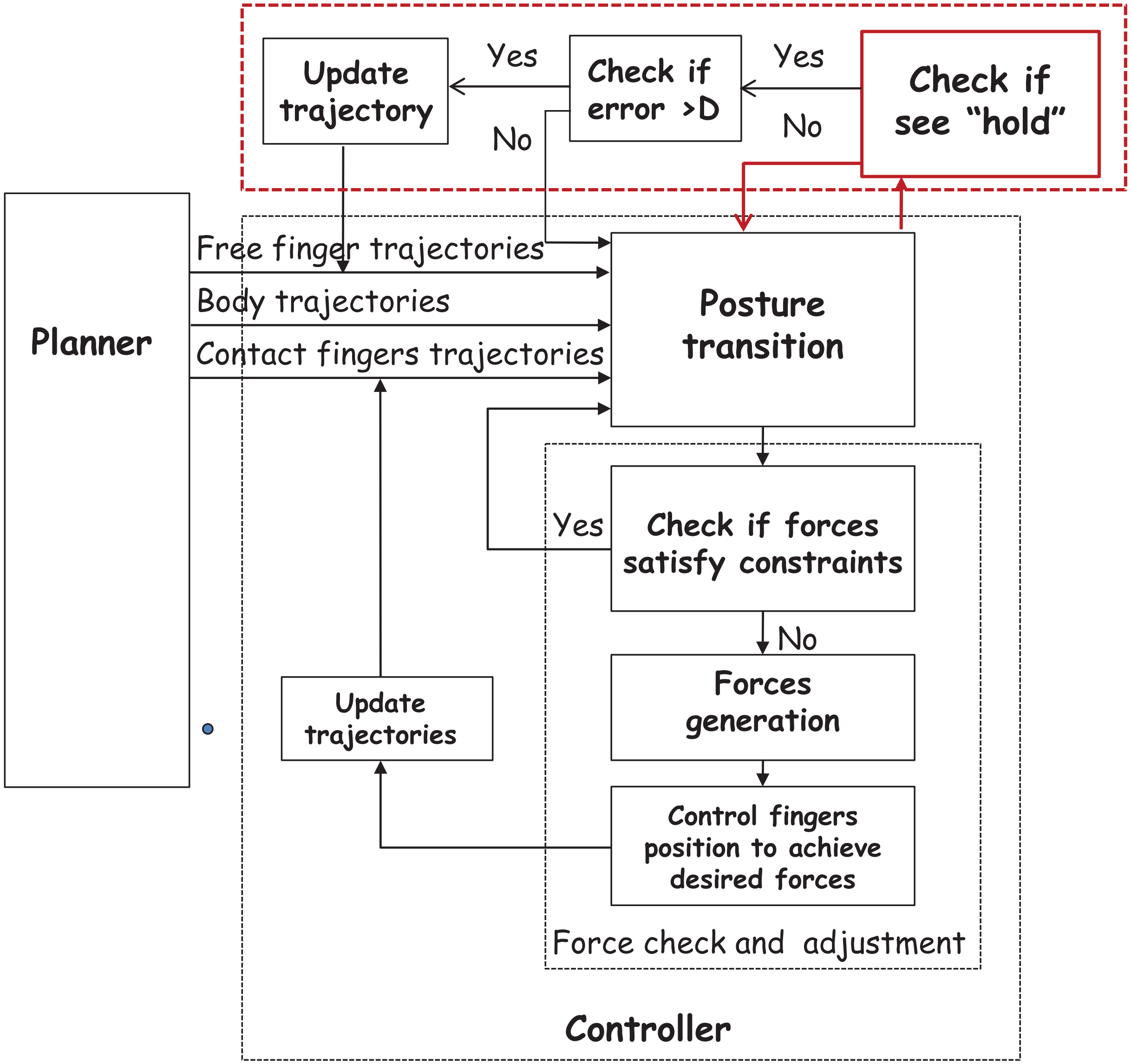
In docking motion, if the camera detects a position error, the vision feedback algorithm will correct the free finger motion trajectory based on the real contact point position.



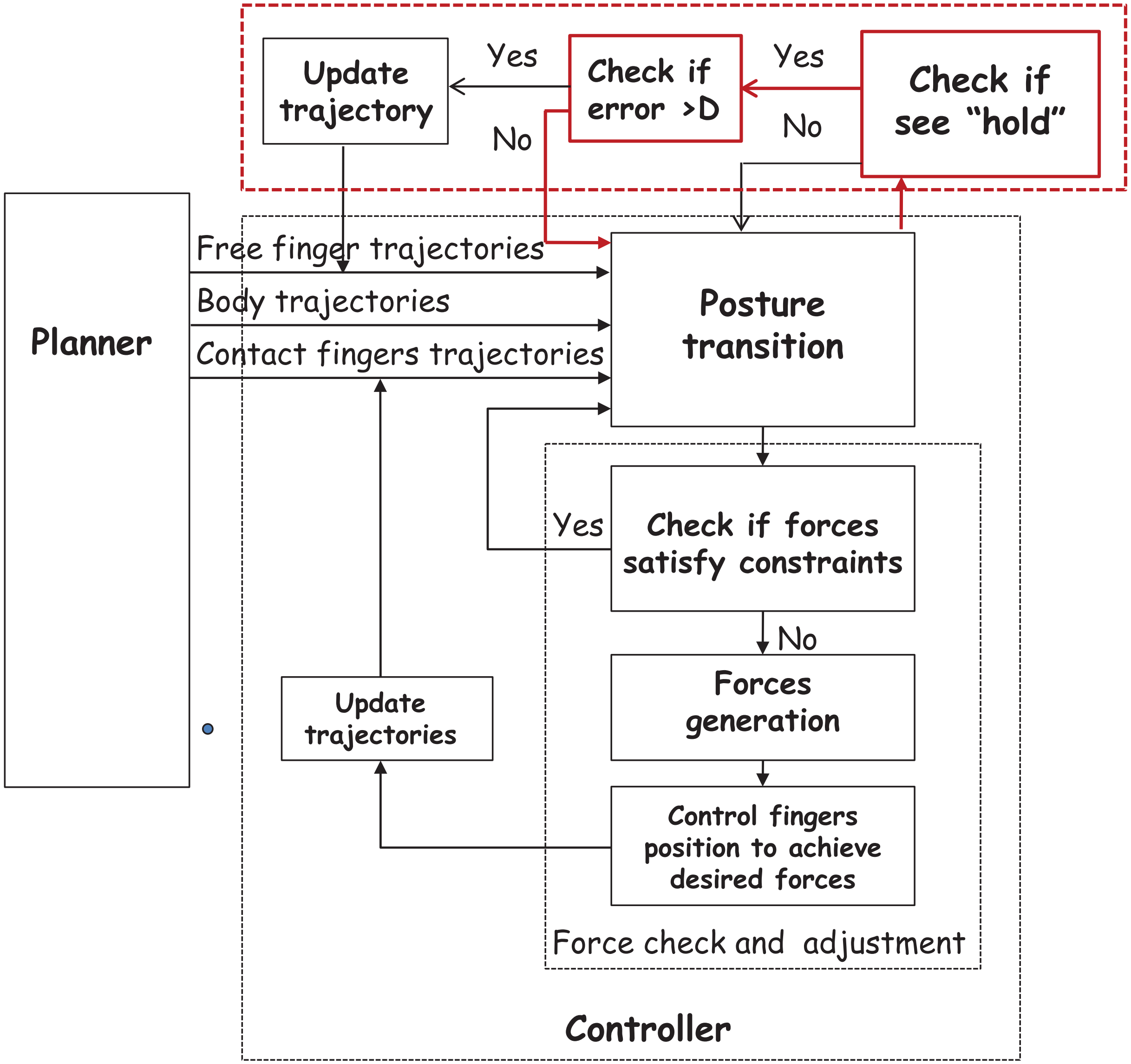
Motion control with vision feedback



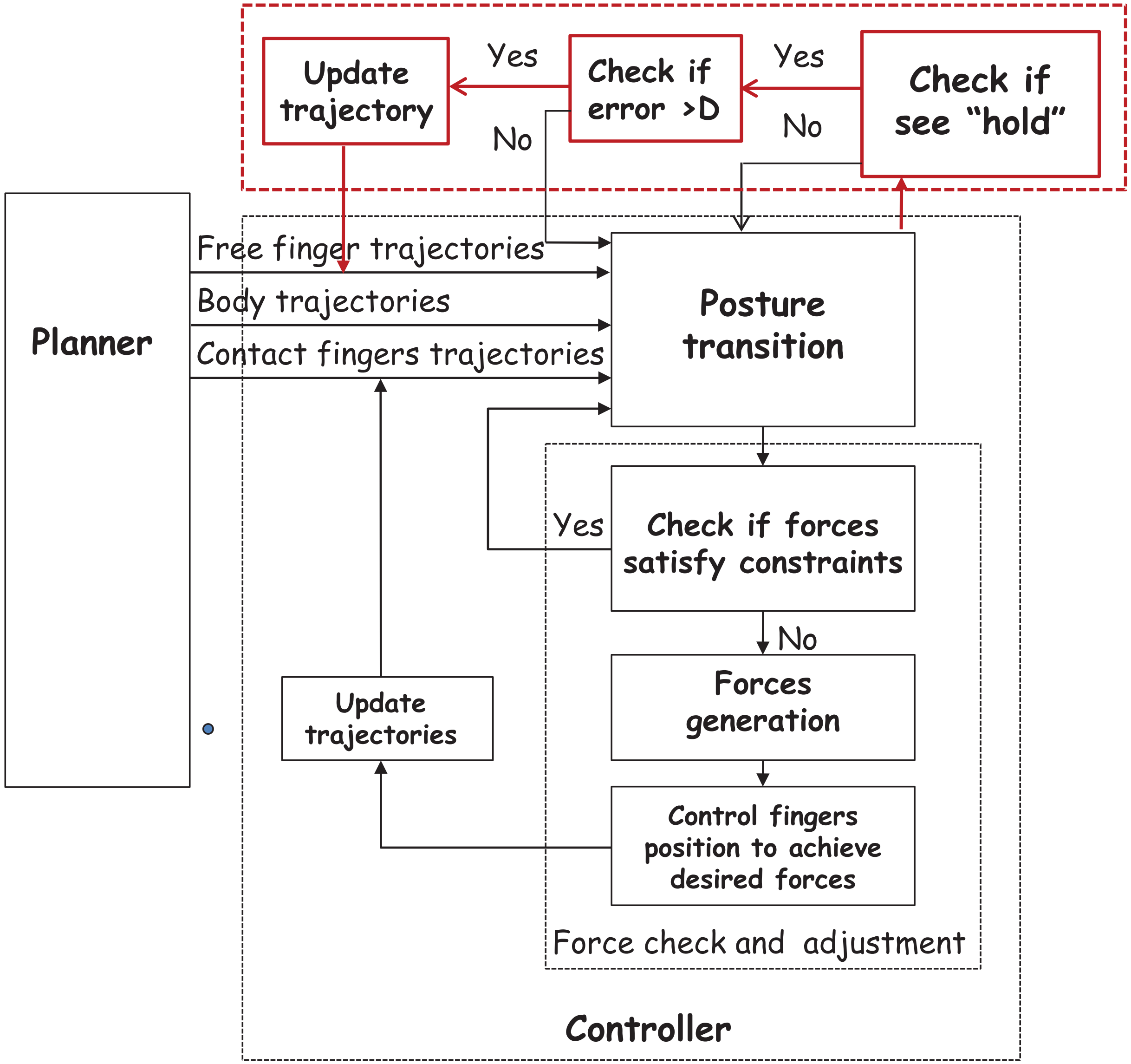
Motion control with vision feedback



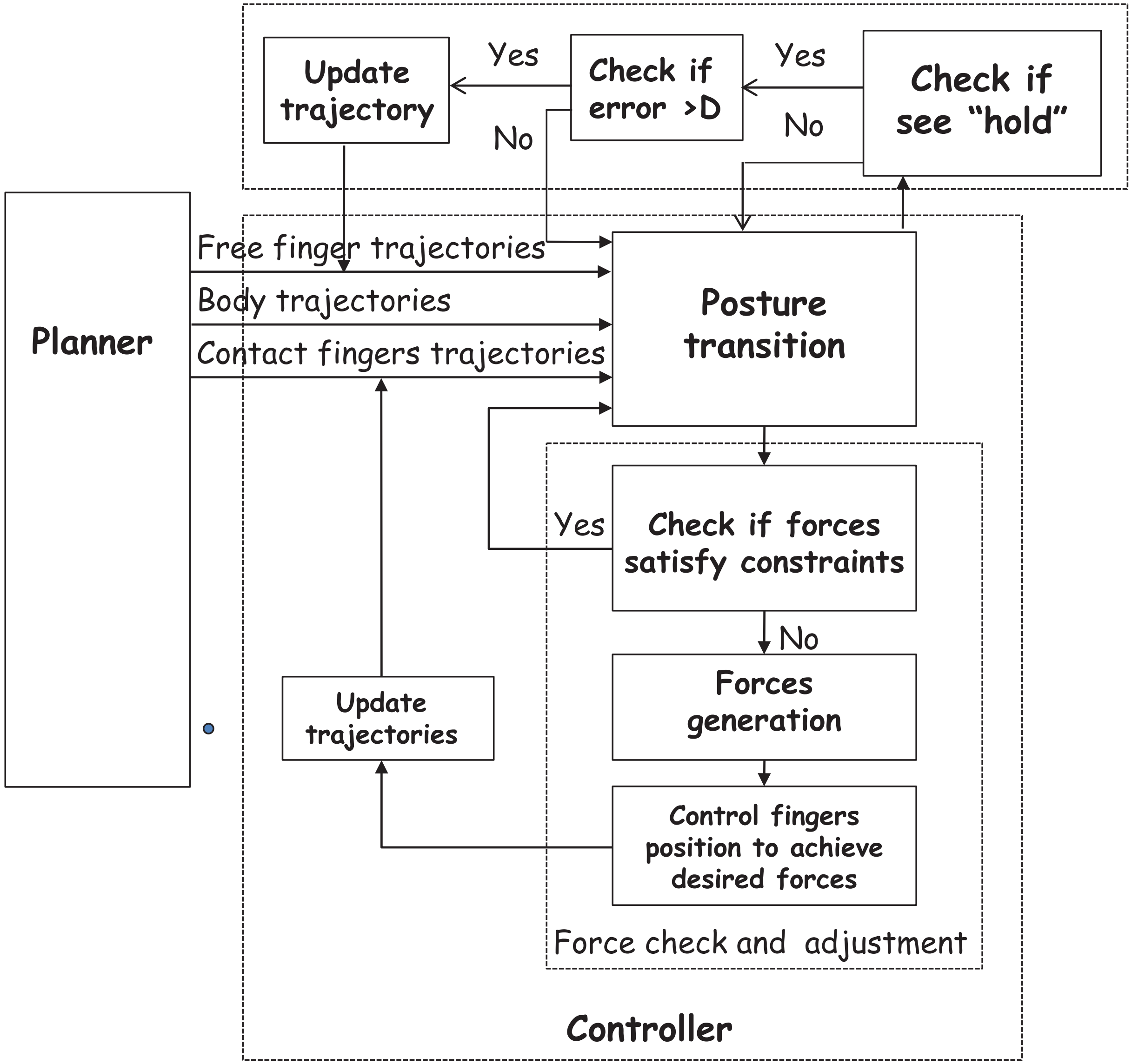
Motion control with vision feedback



Motion control with vision feedback



Vision feedback 2-stage motion control



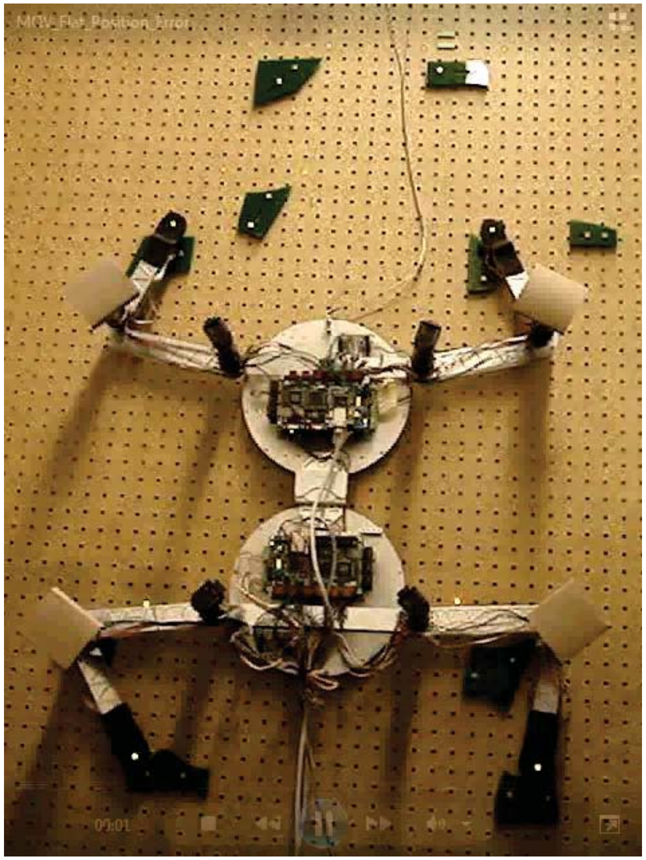
Outline

- Robot design
- Sensors
- Motion control algorithm
- Experiments

Experiments

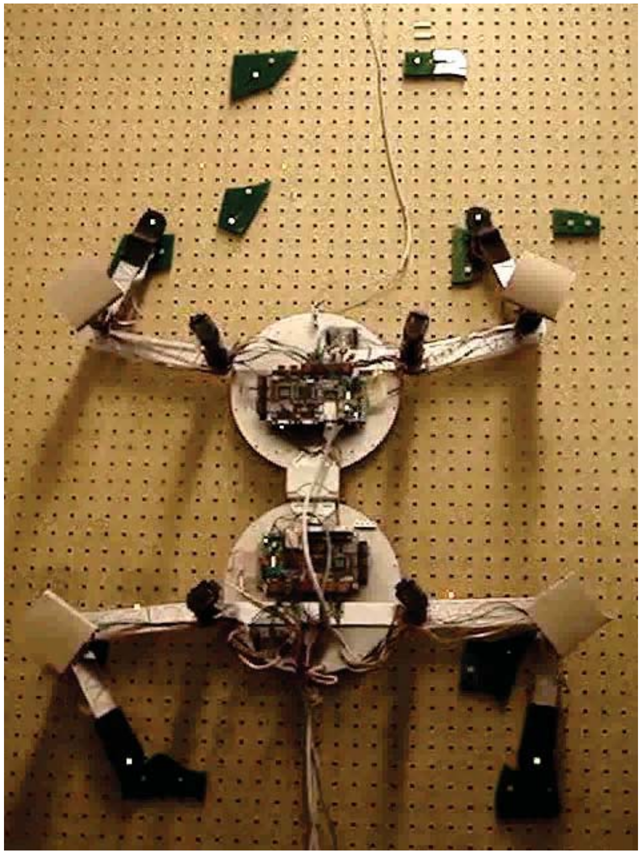
- Terrain I

an easy terrain with most holds horizontal



Open-loop position control

Climb successfully



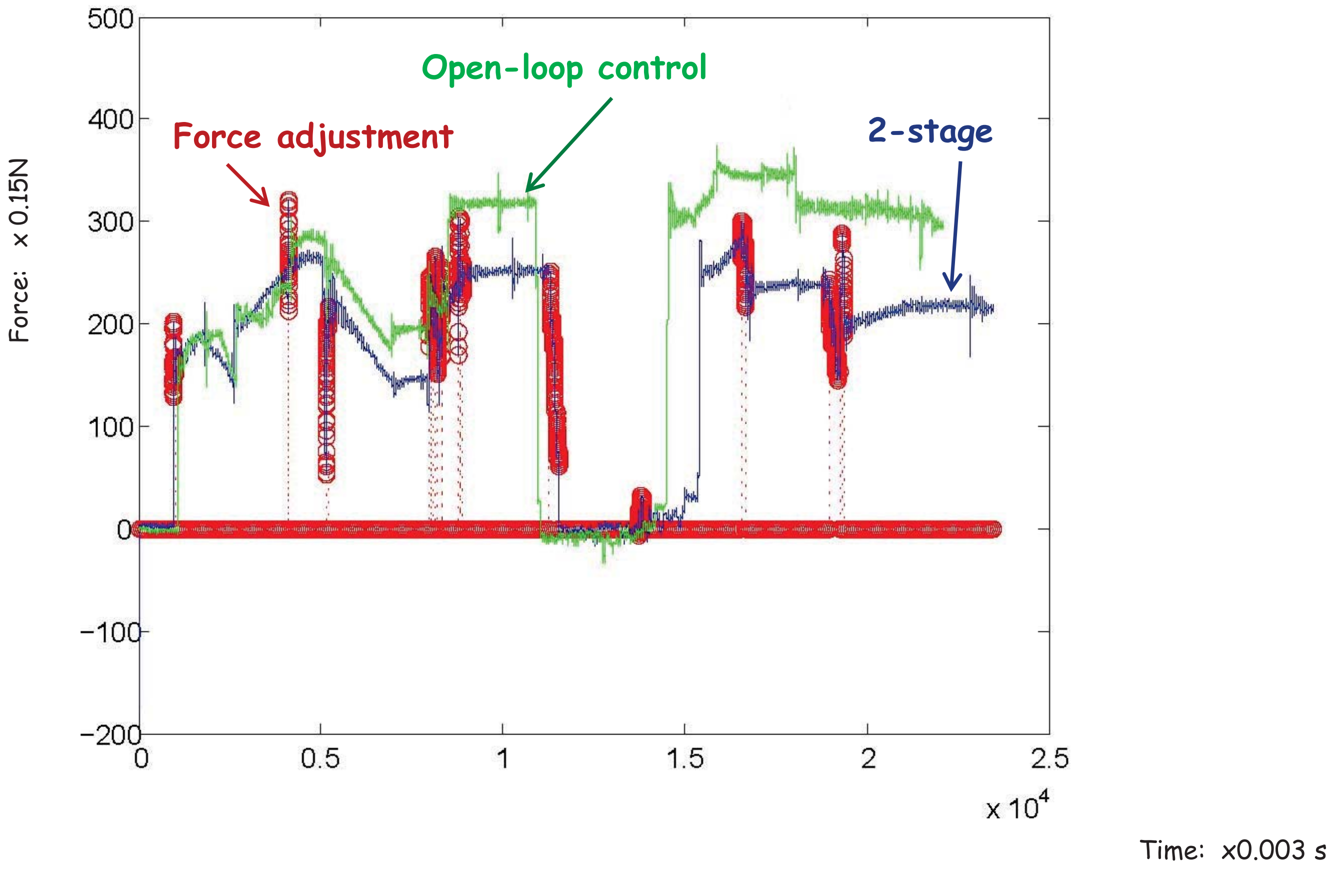
2-stage control

Climb successfully

Experiments

- Force analysis for climbing of terrain I

For this particular terrain, most of the time while climbing lower right finger has the largest contact forces

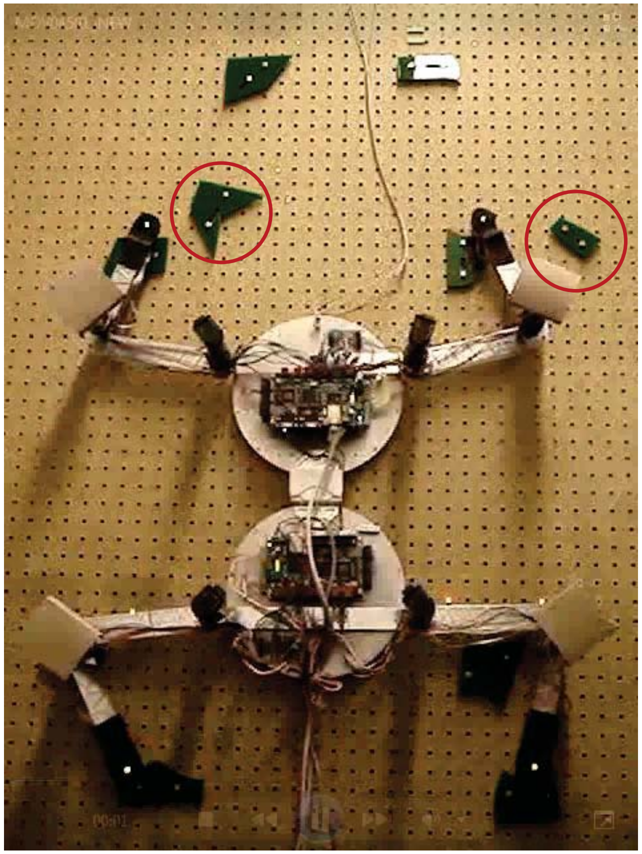


Contact forces of lower right finger

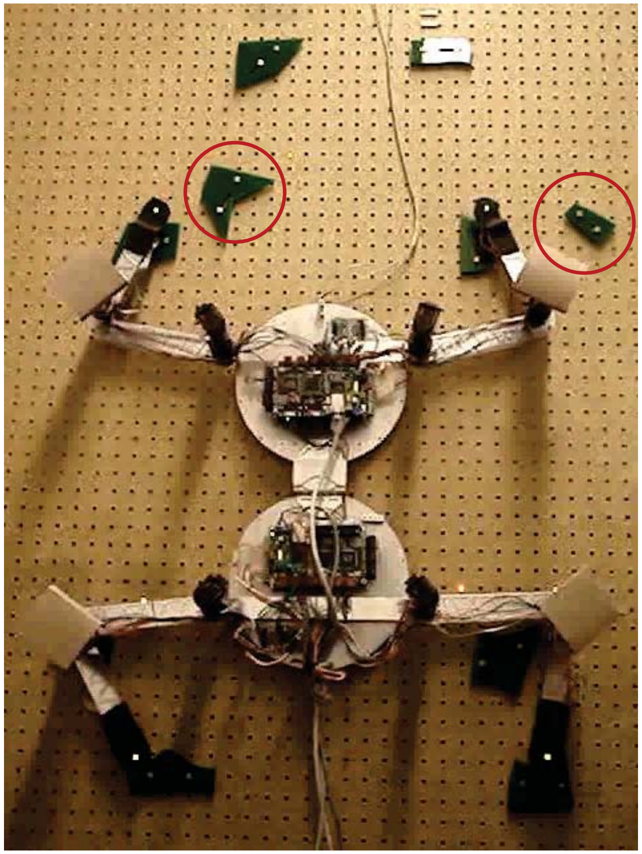
Experiments

- Terrain II

Two slanted holds facing the same direction



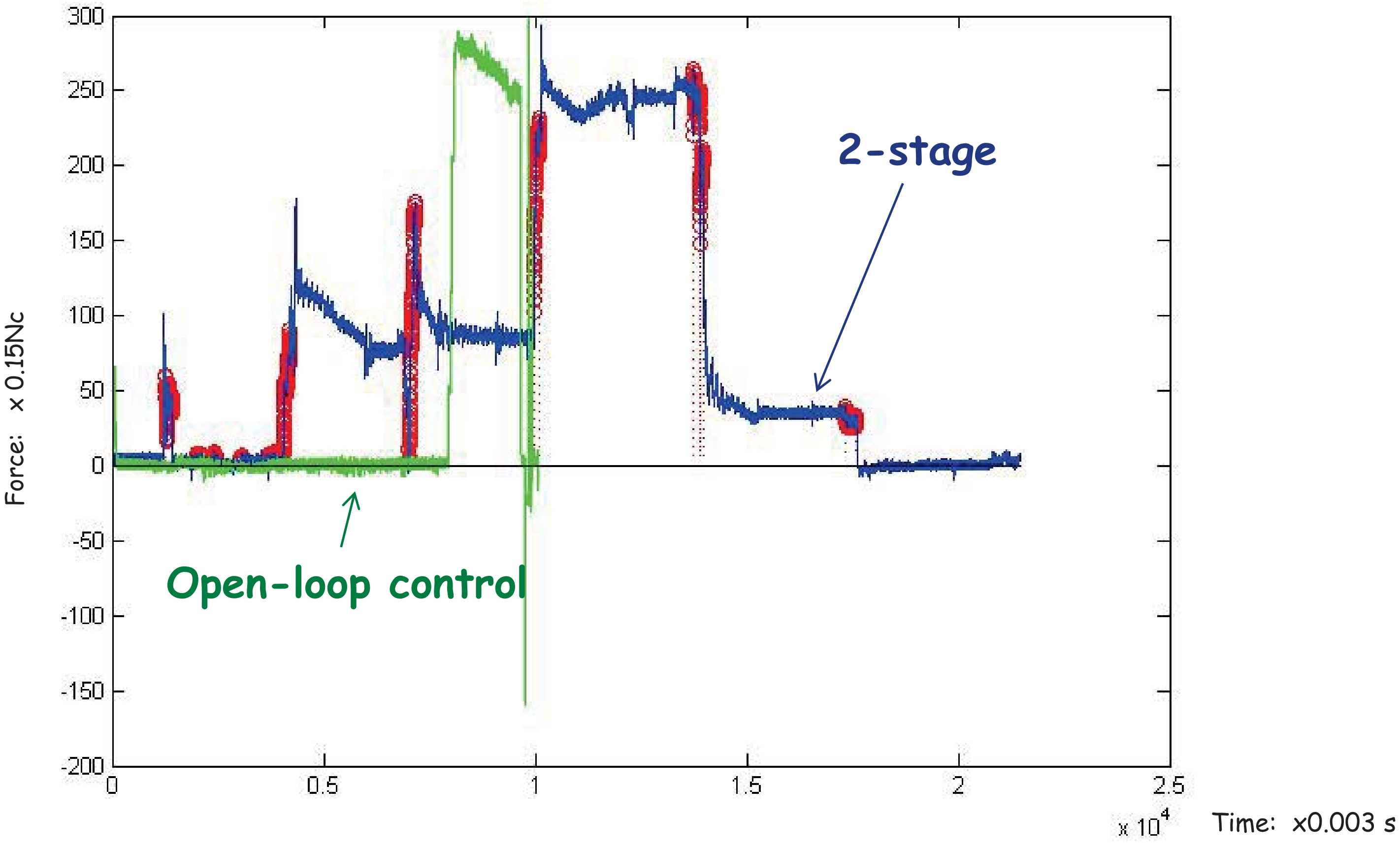
Open-loop position control
Slipping off hold



2-stage control
fingers on hold

Experiments

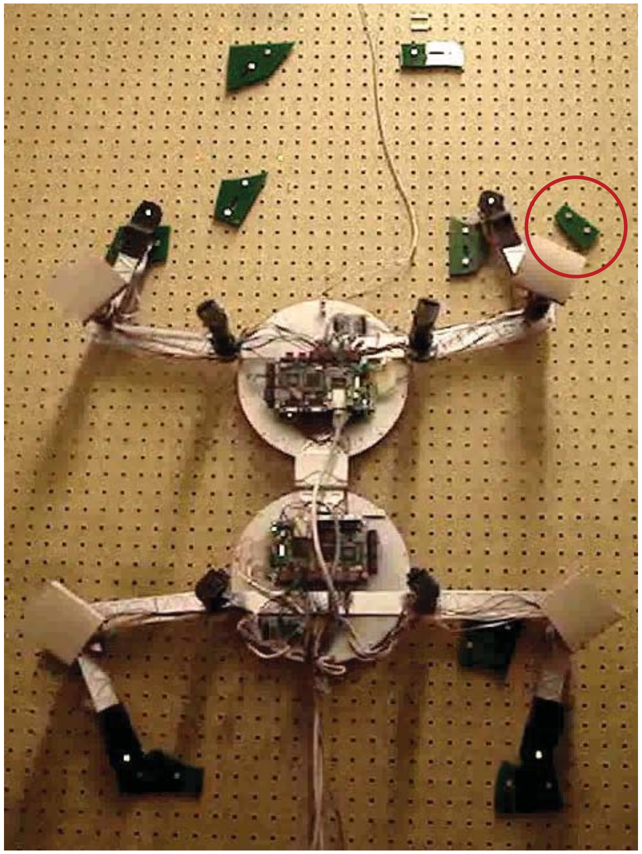
- Force analysis for Terrain II



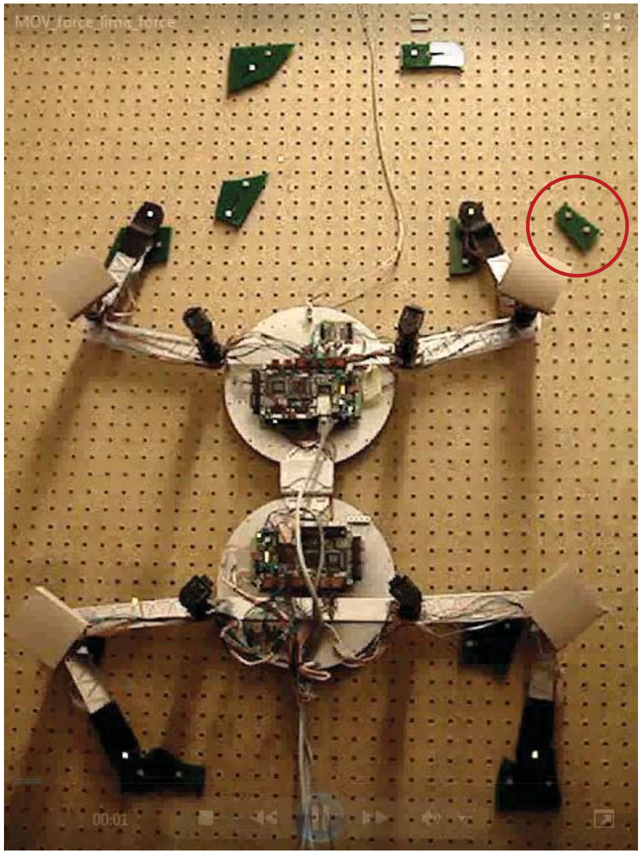
Forces for right upper finger

Experiments

- Terrain III



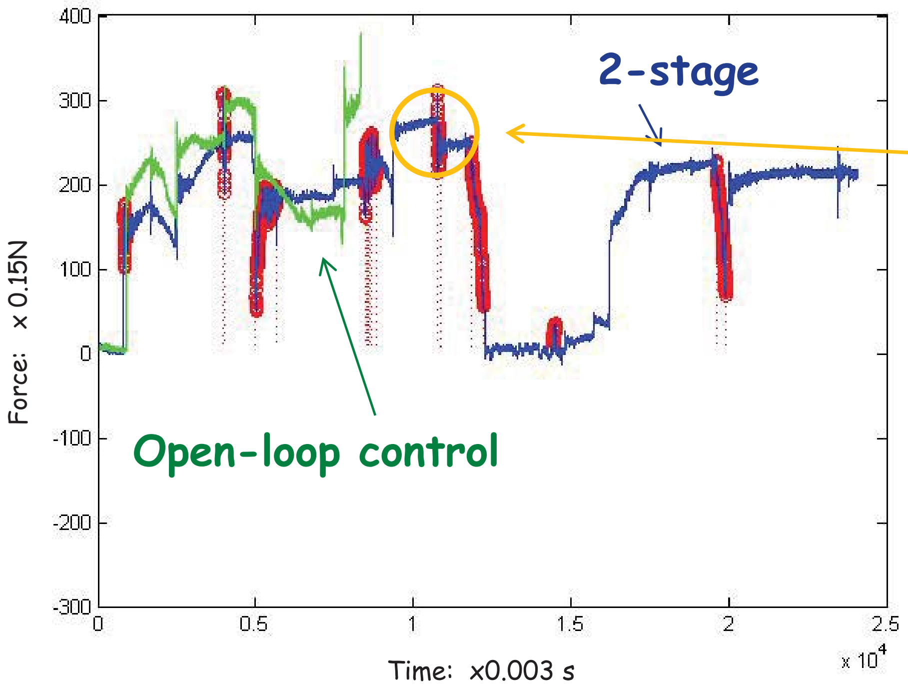
Open-loop position control
Stuck by force limit



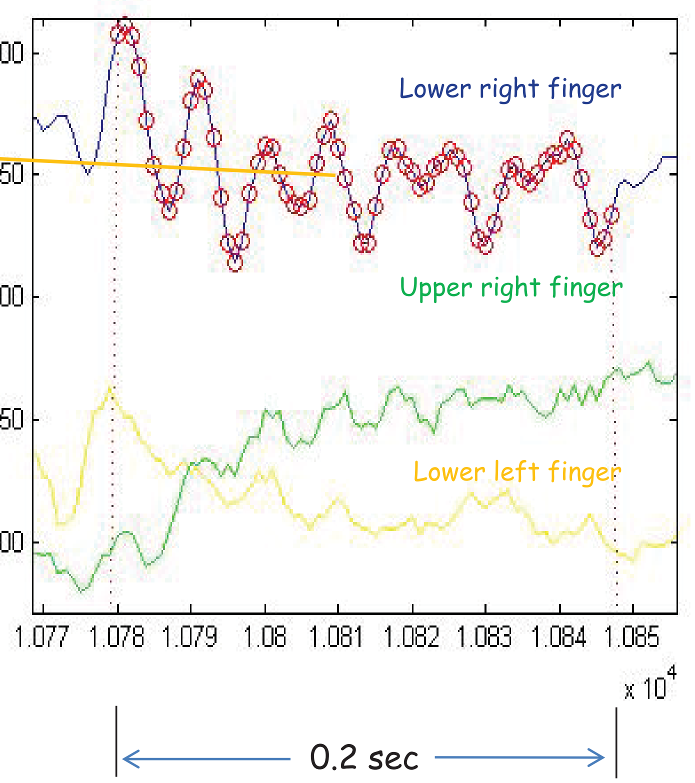
2-stage control
Force allocated properly

Experiments

- Force analysis for Terrain III



Force on lower right finger



Force on upper left finger

Comparing to force control algorithm

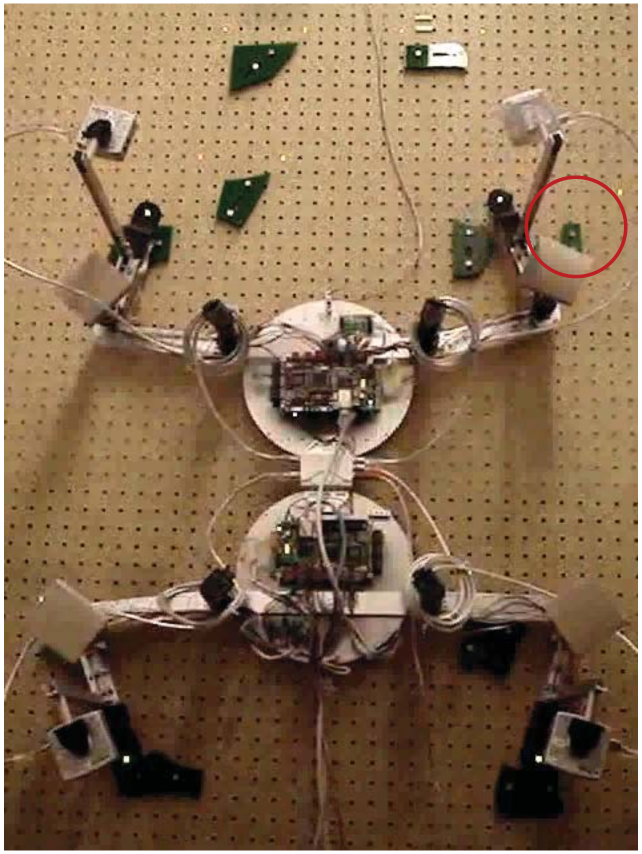
Teresa Miller has designed and tested her force control algorithm on Capuchin
Good performance has been achieve considering using torque control on a system with large joint friction

Our 2-stage algorithm has the following features:

1) motion smooth and stable
2) follow the planned trajectories closely

Experiments

- Docking and vision feedback



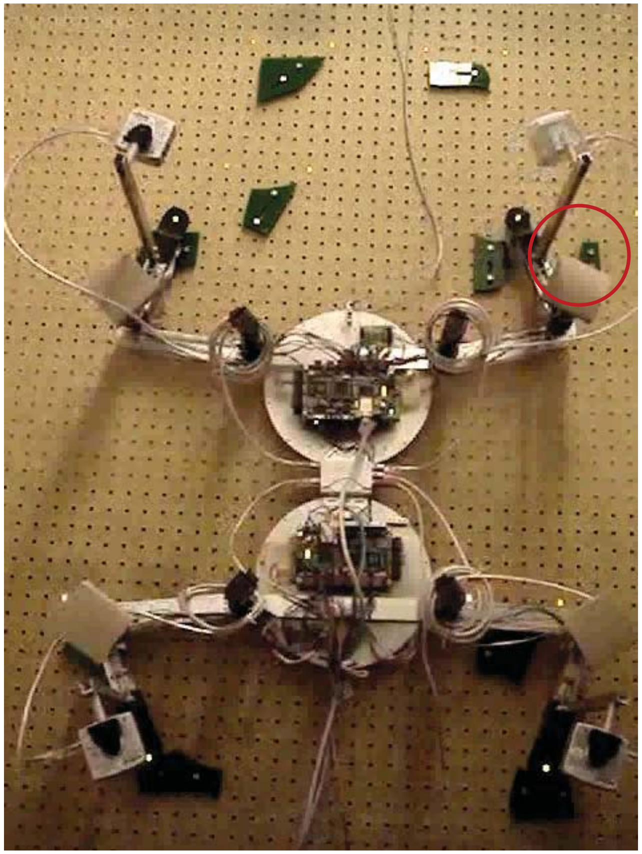
Error exists in the terrain input

No vision feedback used

2-stage control without vision feedback

Experiments

- Docking and vision feedback



Error exists in the terrain input
Vision feedback corrects docking motion

```
to here
hand: 0
width: 320 height: 240
PixIndex: 252
RandIndex: 172
mcRx: 157
mcRy: 133
PixIndex: 135
RandIndex: 45
PixIndex: 135
RandIndex: 89
BX: 315, 110
mcBx: 315
mcBy: 110
PixIndex: 69
RandIndex: 35
PixIndex: 69
RandIndex: 35
PixIndex: 69
RandIndex: 38
PixIndex: 69
RandIndex: 18
PixIndex: 69
RandIndex: 27
mcGx: 0
mcGy: 0
x: -1,y: -1
x: -1,y: -1
to here
hand: 0
```

```
frame = cvQueryFrame( capture3 );
```

Contributions

- Design
  - Designed and built a four-limb climbing robot, some features and design make the robot easy to climb.
- Sensing
  - Installed various sensors on the robot, such as force and vision sensor, to sense the forces and terrain
- Control algorithm
  - Designed a control algorithm that takes advantage of various sensors feedback and makes the climbing more precise and robust
  - It enables the robot to climb some difficult terrain where basic position control algorithm normally fails
- Implementation
  - Integrated the planner, the sensing system and control algorithm on the robot and made the robot climbed the vertical artificial climbing wall

System has been tested successfully

Main lesson from our work

For **quasi-static climbing**, it is not necessary to perform continuous force control.

It is sufficient to do continuous force monitoring and to perform occasional force adjustment.

This was not obvious at the beginning, but our implementation and tests have shown that a control approach based on force adjustment only when it is needed achieve reliability and reasonable performance.

Directions of future work

1) 3D terrain – 5 more DOFs +3D sensing, holds characterization
2) Incremental sensing and online planning
3) Taking advantage of dynamics

Acknowledgement

Advisor:
Professor Jean-Claude Latombe

Committee:
Professor Scott L. Delp,
Professor Oussama Khatib
Professor Stephen M. Rock,
Professor Kenneth Salisbury

Group members:
Ankur Dhanik, Peggy Yao, Liangjun Zhang, Kris Hauser, Tim Bretl,
Teresa Miller

Friends & family

Thank you!

References

[1] Ruixiang Zhang and Jean-Claude Latombe (2013). Capuchin: A Free-Climbing Robot, International Journal of Advanced Robotic Systems, Ellips Masehian (Ed.), ISBN: 1729-8806, InTech, DOI: 10.5772/56469. Available from: http://www.intechopen.com/journals/international_journal_of_advanced_robotic_systems/capuchin-a-free-climbing-robot

[2] Ruixiang Zhang (2008). Design of a climbing robot: Capuchin, Proc. 5th Intl. Conf. on Computational Intelligence, Robotics, and Autonomous Systems. June 2008

[3] Ruixiang Zhang, Prahlad Vadakkepat, CM Chew. Motion Planning for Biped Robot Climbing Stairs, Proceeding of FIRA Robot World Congress, Oct 2003, Vienna, Austia.