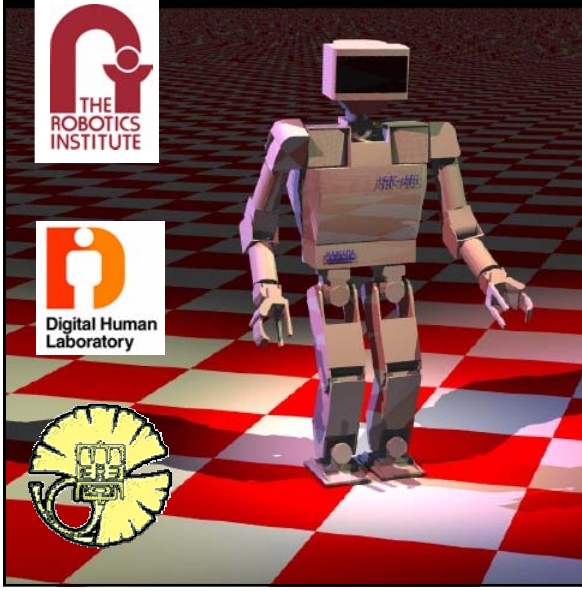
Self-Collision Detection and Prevention for Humanoid Robots



James Kuffner, Jr.
Carnegie Mellon University

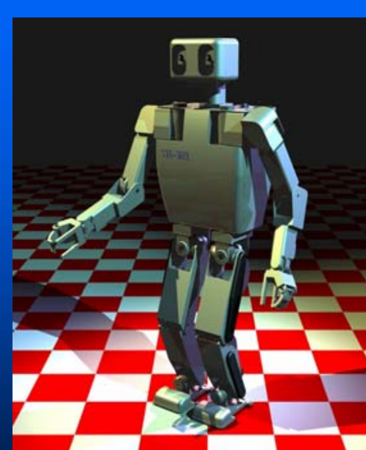
Koichi Nishiwaki
The University of Tokyo

Satoshi Kagami
Digital Human Lab (AIST)

Masayuki Inaba
Hirochika Inoue
The University of Tokyo

Talk Overview

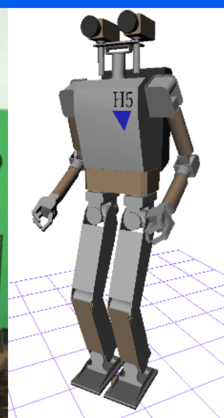
- Self-Collision Detection
- Feature-based Minimum Distance Computation:
  - Approximate Geometric Models
  - Minimum Distances for 3D Convex Polyhedral Models
  - Coherency and Efficiency
- Safe Biped Walking
- Experimental Results
- Conclusion & Future Work



JSK Humanoid Research Platforms

1998 - 1999

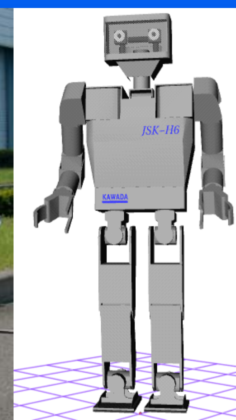
H5 (30 DOF ; 33kg ; 1.27m)



[Nagasaka, Konno, Nishiwaki, Kitagawa, Sugihara, Kagami, Inaba, Inoue]

2000 - 2001

H6 (35 DOF ; 55kg ; 1.37m)



[Nishiwaki, Sugihara, Kagami, Kanehiro, Inaba, Inoue]

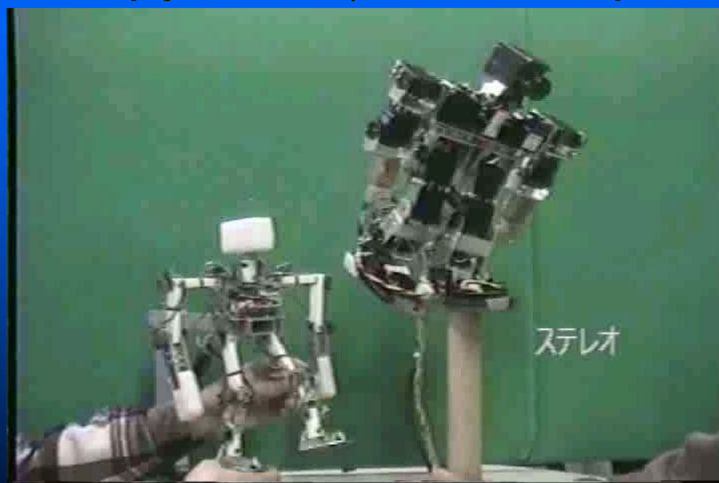
Humanoid "H7"

2001 - 2003

Balance Compensation as an Optimization Problem

AutoBalancer: online dynamics filter

[Kagami, Kanehiro, Tamiya, Inaba, Inoue : WAFR 2000]



$$\delta\boldsymbol{\theta} = \delta\boldsymbol{\theta}_{cmd} + \delta\boldsymbol{\theta}_{ret} + \delta\boldsymbol{\theta}_{comp}$$

$$E(\delta\boldsymbol{\theta}_{comp}) = \sum_{i=1}^{p} w_i \, (\delta\boldsymbol{\theta}_{comp,i})^2$$

Dynamic Stability Criteria

- *Static stability:*

$$\delta C_x(\delta\boldsymbol{\theta}) = 0$$

$$\delta C_y(\delta\boldsymbol{\theta}) = 0$$

$$\mathbf{C} = \mathbf{X}(\boldsymbol{\theta})$$

$$\dot{\mathbf{C}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

$$\delta\mathbf{C} = \mathbf{J}(\boldsymbol{\theta})\delta\boldsymbol{\theta}(t)$$

- *Dynamic stability:*

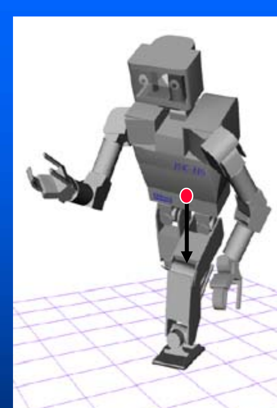
  Moment of Inertia Contstraints

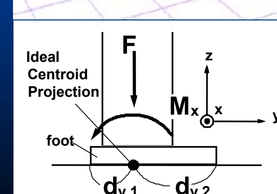
  Zero Moment Point (ZMP)

$$-d_{y,1}F < M_x(\delta\boldsymbol{\theta}) < d_{y,2}F$$

$$-d_{x,1}F < M_y(\delta\boldsymbol{\theta}) < d_{x,2}F$$

$$-\mu F < M_z(\delta\boldsymbol{\theta}) < \mu F$$





"Virtual Puppet" Interface

Online Walking Trajectory Generation

- Even without obstacles, balance criteria alone cannot guarantee safe motion!
- Need methods for detecting and preventing leg interference and body self-collision
  - Efficient (low computation time)
  - Reliable (computation time has low variance)
  - Numerically robust
- Should be conservative (error-bounds)

Self-Collision

- Occurs when one of more of the links of a robot collide
- Dangers of self-collision:
  – Can cause robot to damage itself
  – Through a loss of balance or control, can cause damage to the surrounding environment (including nearby people)
- *Detecting* and *preventing* potential self-collisions is fundamental to developing safe robots

Safe Biped Walking

IDEAL:

– A colliding posture is *never output* by the controller, thus all trajectories executed by the robot are *guaranteed* to be free of self-collision

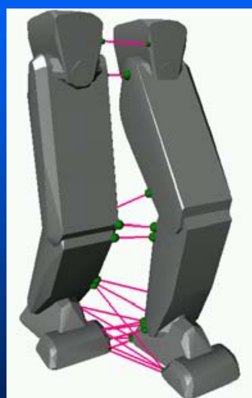
CURRENT:

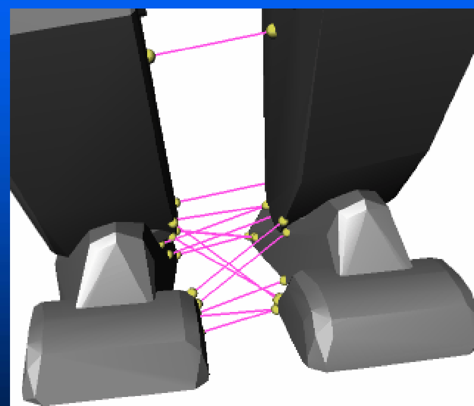
– All computed *desired* body trajectories are free of self-collision (prior to online balance compensation and PD control)

– Depends on the accuracy of both the kinematic and dynamic model of the robot and environment

Our Approach

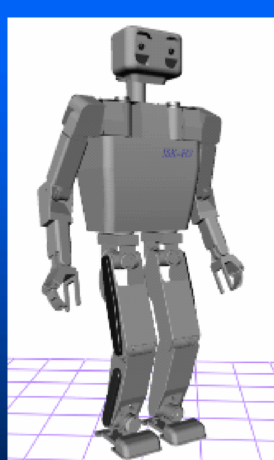
Maintain closest points between pairs of links



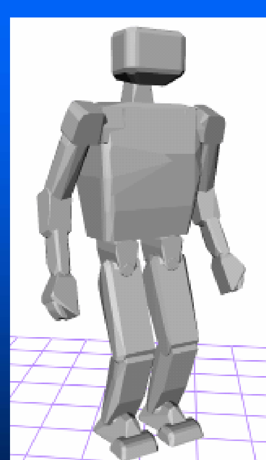


All pairs can be considered, or a subset of possibly-colliding pairs can be pre-computed **[Kanehiro, Hirukawa '01]**

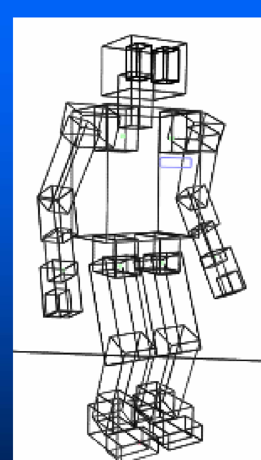
Approximating Link Geometry



Full CAD Model
(314,588 triangles)

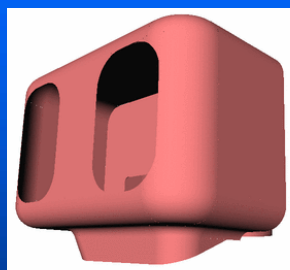


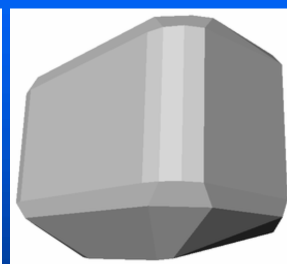
Convex Hulls
(2,702 triangles)

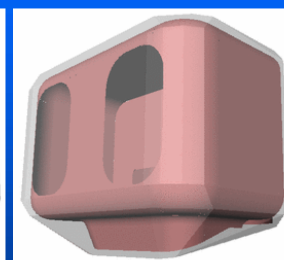


Bounding Boxes
(432 triangles)

Protective Convex Hulls







CAD model

Protective Hull

Combined View

Self-Collision Detection

- **GIVEN:**
  - Tree-like kinematic structure with N links:
- **ASSUME:**
  - joint limits specified to prevent collisions between adjacent links.
- Remaining pairs to check:

$$\frac{N(N-1)}{2}-(N-1)=\frac{N^2-3N+2}{2}$$

30 links = 454 pairs

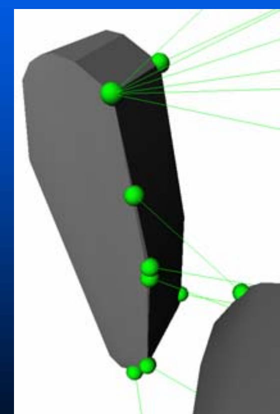
Minimum Distance Computation

CONVEX POLYHEDRA:

- GJK [Gilbert, Johnson, Keerthi '88]
- Lin-Canny [Lin, Canny '91]
- Enhanced-GJK [Cameron '97]
- V-Clip [Mirtich '98]
- H-walk [Guibas, Hsu, Zhang '99]

NON-CONVEX POLYHEDRA:

- Sphere Trees [Quinlan '94]
- CFL [Kawachi, Suzuki '00]
- SWIFT++ : [Ehmann, Lin '01]



Feature-based Minimum Distance Computation

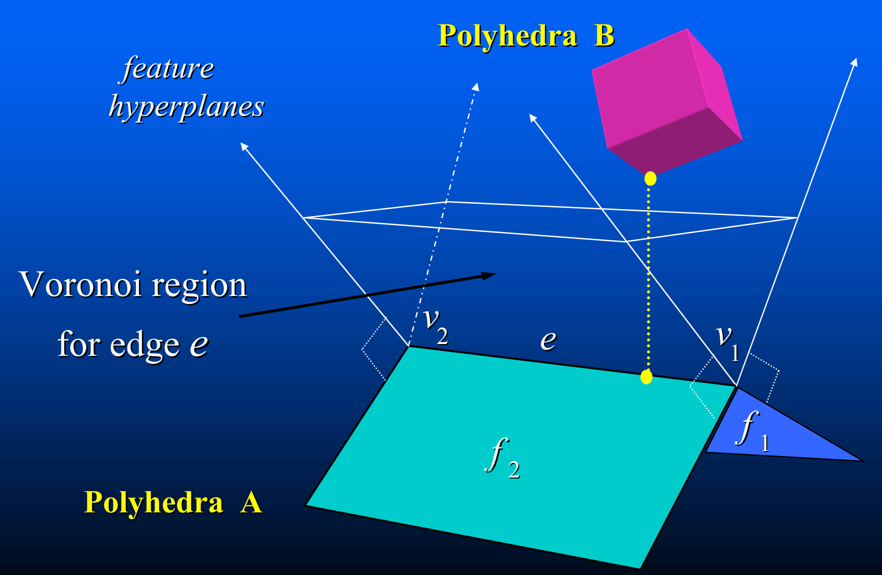
ADVANTAGES:

- Exploits spatial and temporal coherency
- Minimum distance value and pairs of closest feature points can be updated in "almost constant" time
- No performance loss for near-colliding cases

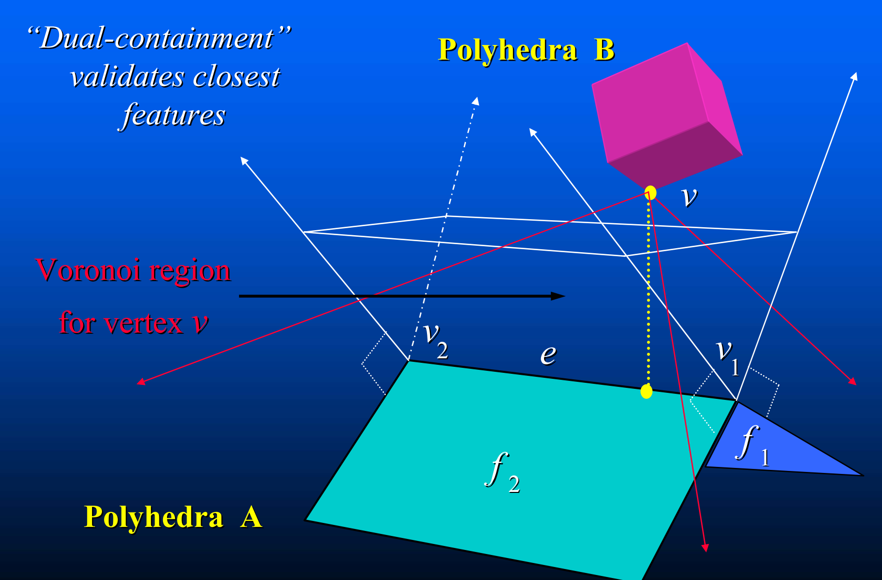
DISADVANTAGES:

- Typically limited only to closed, bounded, convex polyhedra (or hierarchical collections of them)

Voronoi-based Minimum Distances



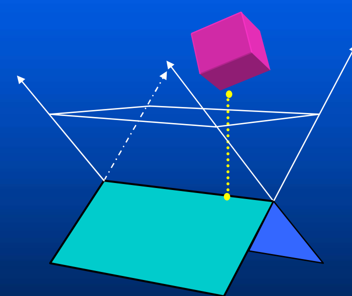
Voronoi-based Minimum Distances



Voronoi-based Minimum Distances

POSSIBLE STATES:

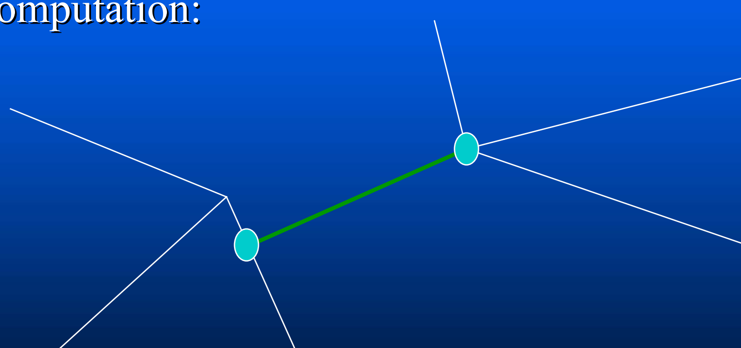
- Vertex – Vertex (V-V)
- Vertex – Edge (V-E)
- Vertex – Face (V-F)
- Edge – Edge (E-E)
- Edge – Face (E-F)



Algorithm keeps track of possible transitions between states

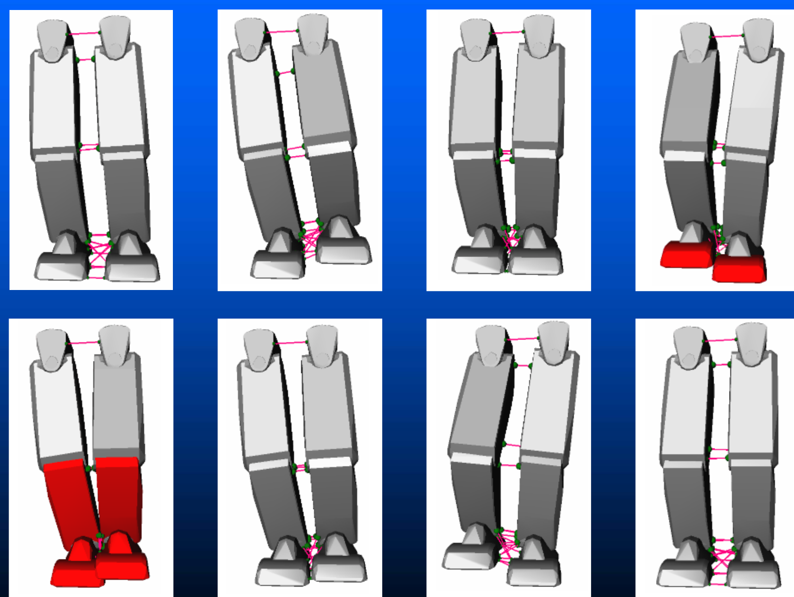
Coherency and Efficiency

- Closest feature pairs cached from previous computation:



Subsequent checks verify Voronoi containment constraints

Example 1 : In-Place Stepping



In-Place & Forward-Turning Step

14 links w/ 14 DOF
Select = 19 active pairs
Full = 73 active pairs

866 MHz dual Pentium III
Time per config = 0.128 ms
Time per config = 0.429 ms

Whole-Body Self-Collision Detection

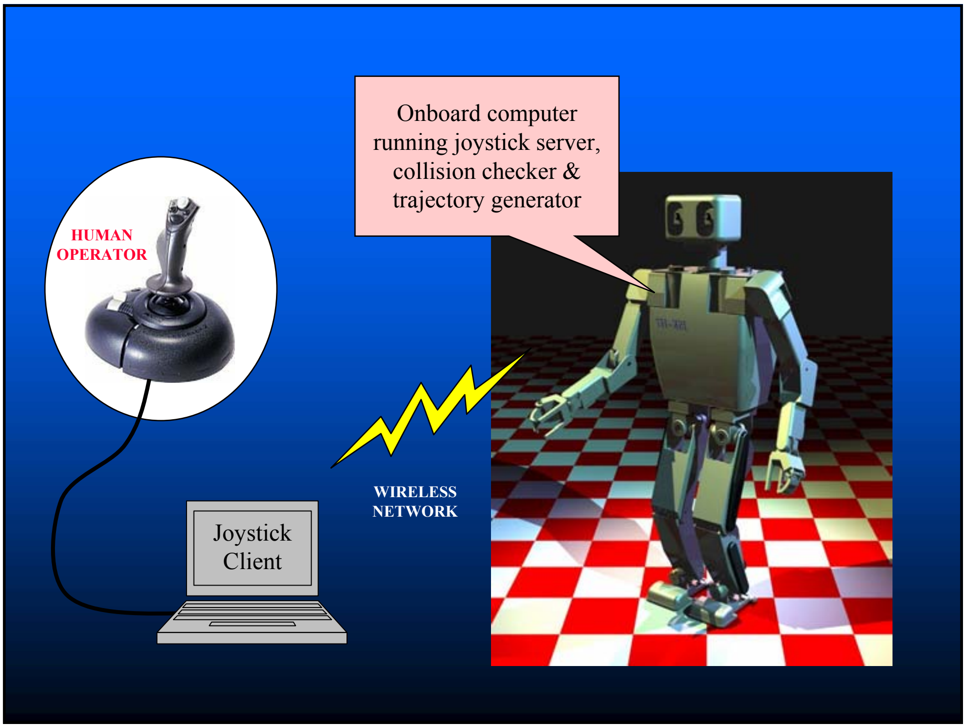
31 links w/ 30 DOF
76 active pairs

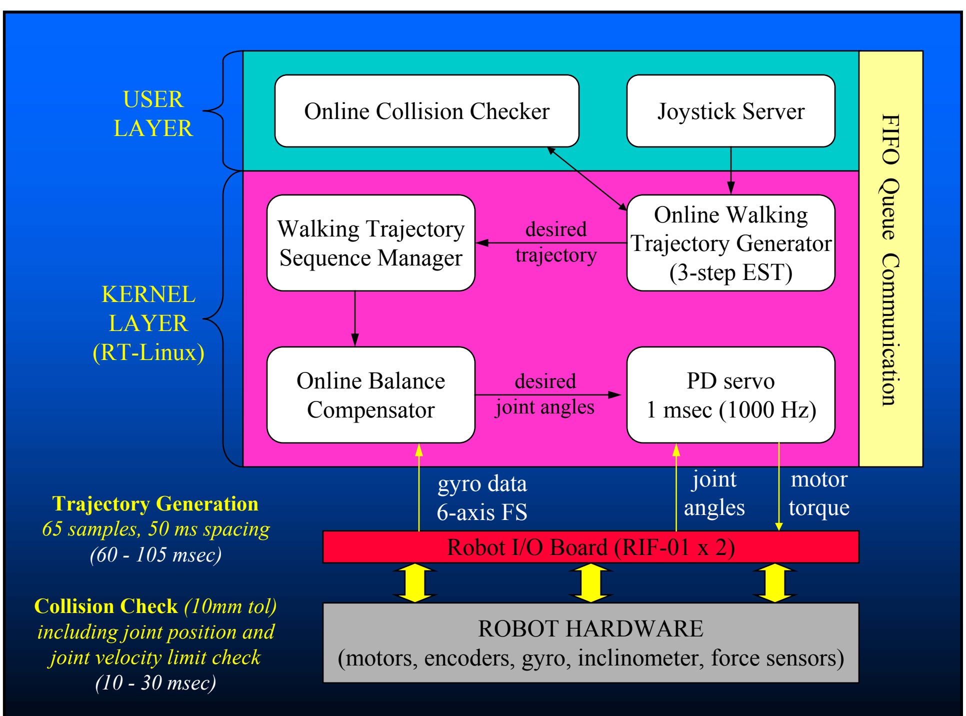
866 MHz dual Pentium III
Time per config = 0.44 ms

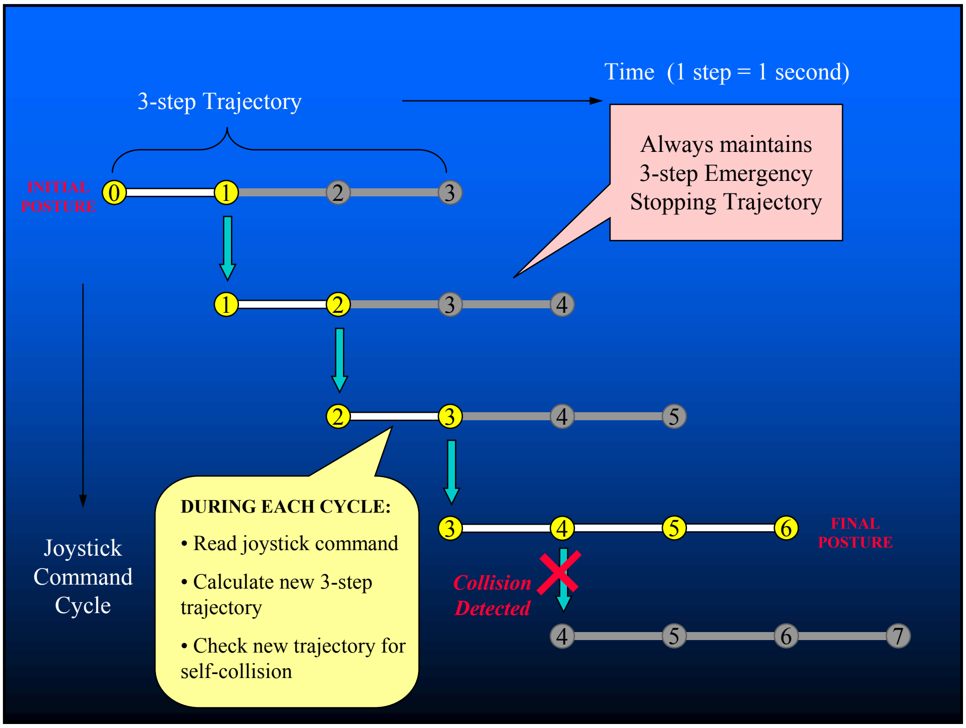
Whole-Body Self-Collision Detection

31 links w/ 30 DOF
Select = 76 active pairs
Full = 425 active pairs

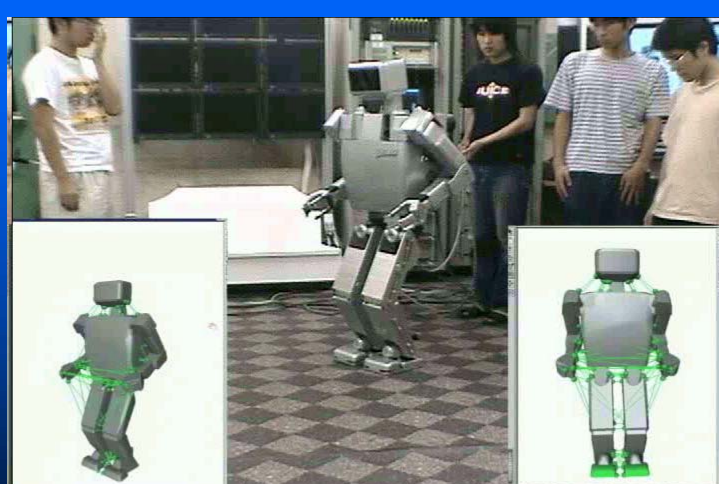
866 MHz dual Pentium III
Time per config = 0.44 ms
Time per config = 2.44 ms


Onboard computer running joystick server, collision checker & trajectory generator
HUMAN OPERATOR
WIRELESS NETWORK
Joystick Client


USER LAYER
Online Collision Checker
Joystick Server
FIFO Queue Communication
KERNEL LAYER (RT-Linux)
Walking Trajectory Sequence Manager
desired trajectory
Online Walking Trajectory Generator (3-step EST)
Online Balance Compensator
desired joint angles
PD servo 1 msec (1000 Hz)
gyro data 6-axis FS
joint angles
motor torque
Robot I/O Board (RIF-01 x 2)
ROBOT HARDWARE (motors, encoders, gyro, inclinometer, force sensors)
Trajectory Generation
65 samples, 50 ms spacing
(60 - 105 msec)
Collision Check (10mm tol) including joint position and joint velocity limit check
(10 - 30 msec)



Safe Online Joystick Control



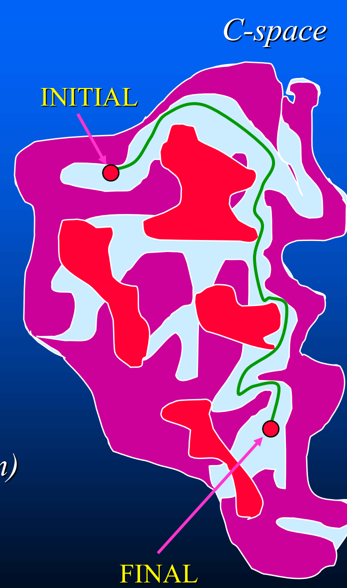
Self-collision check
3 steps in the future
(60-65 configurations)

Minimum Distance
Configuration

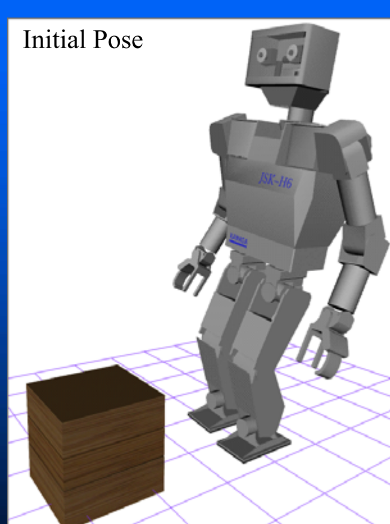
Other Applications : Motion Planning

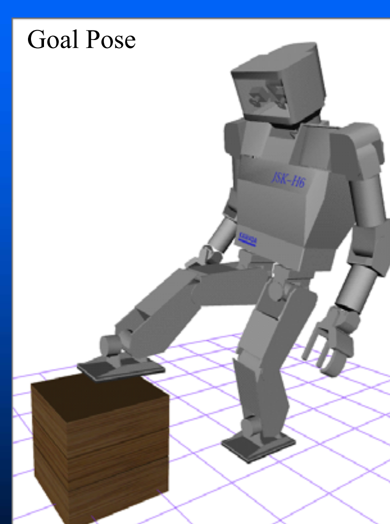
- High-dim. C-space (30 DOF)
- Balance and obstacle constraints restrict space of feasible configurations

GOAL: Compute a trajectory that:
- Connects the initial and final posture
- Collision-free *(including self-collision)*
- Dynamically-stable

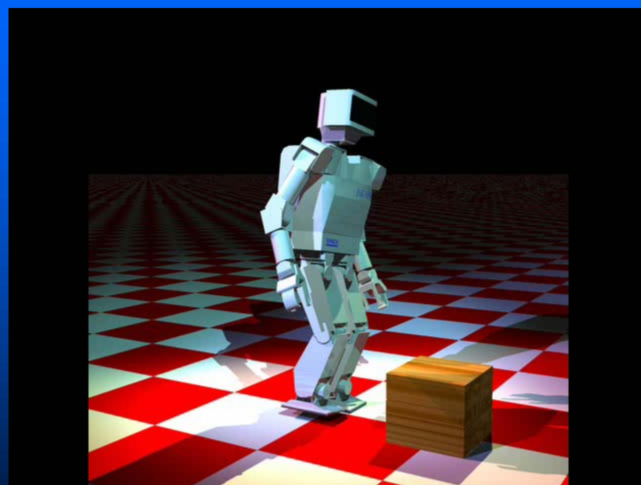


Single-leg Example



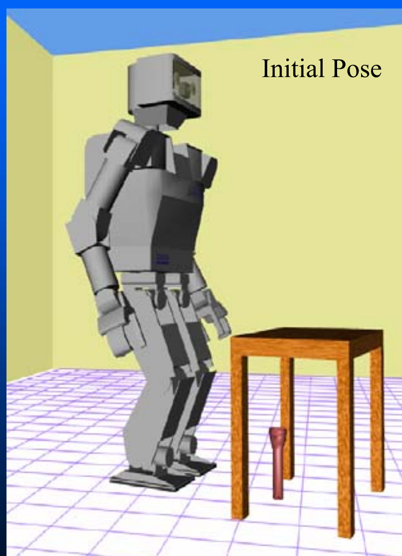


Single-leg Example

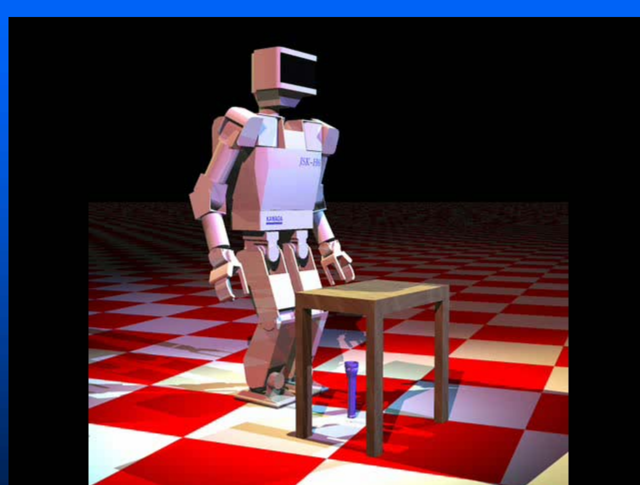


Single-leg Example

Dual-leg Example



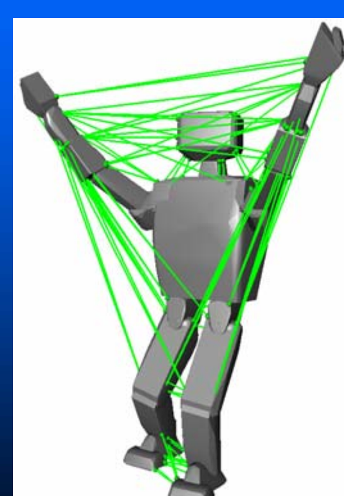
Dual-leg Example



Dual-leg Example

Summary

- Efficient approach to self-collision detection using:
  - Polyhedral convex protective hulls
  - Fast feature-based minimum distance determination algorithms
- Implemented & tested concurrently with online joystick control
- Reliable and reusable *software* is fundamental to improving robot functionality and autonomy
- Simulation systems for humanoid robots help reduce costs and safety-risks during testing



Future Work

- Bring Self-collision testing closer to the servo control layer
- Extend method to handle non-convex polyhedral models
- Optimize the number of distance calculations using maximum joint velocity bounds
- Can be used to prevent collisions between environment obstacles of known geometry
- Applications in multi-robot coordination and computer animation of articulated figures

Robodex 2003 ： *April 7, 2003*

The birthday of

鉄腕アトム “AstroBoy”

Robodex 2003

Honda's new "Silver ASIMO"

Robodex 2003

Sony Dream Robot (SDR-4X)

Robodex 2003

AIST Japan’s HRP2

Robodex 2003

AIST Japan’s HRP2