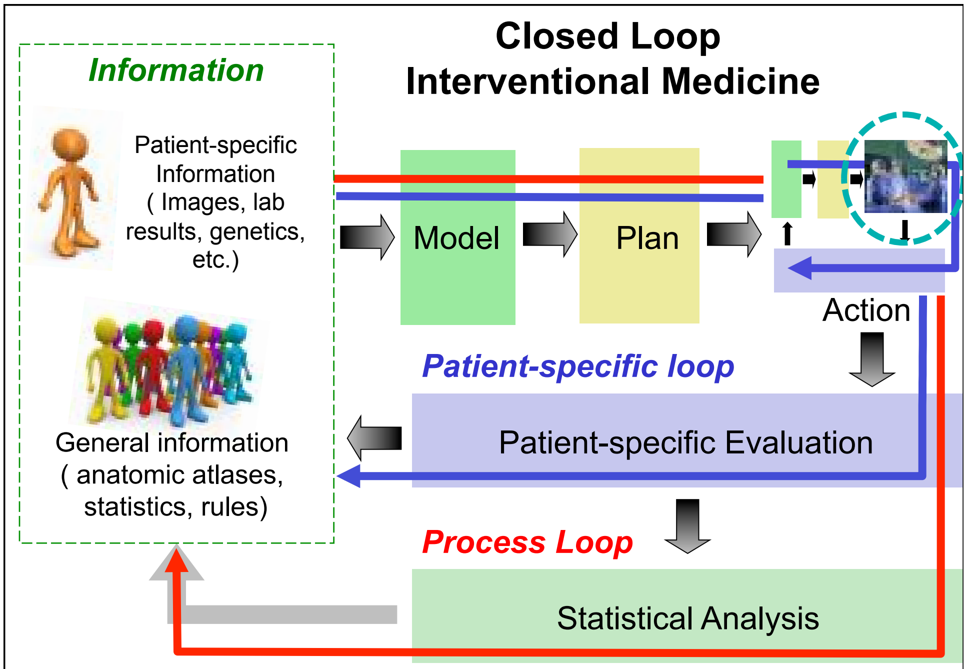
Constrained Robot Motion Control and "Virtual Fixtures"

Russell H. Taylor

600.445

Closed Loop Interventional Medicine



Information

Patient-specific Information (Images, lab results, genetics, etc.)

General information (anatomic atlases, statistics, rules)

Model

Plan

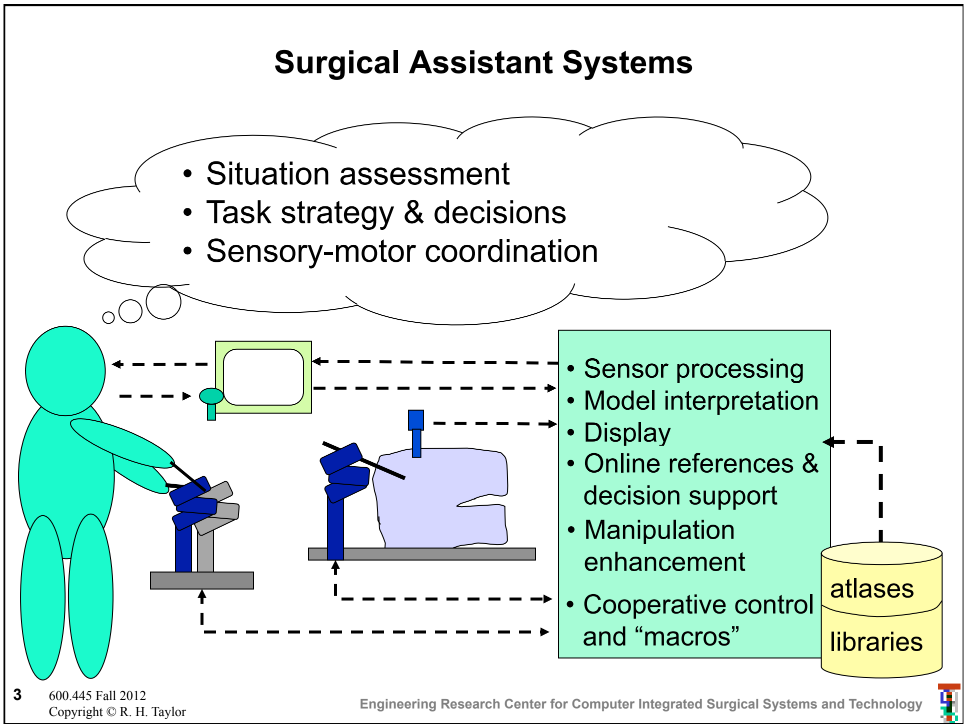
Action

Patient-specific loop

Patient-specific Evaluation

Process Loop

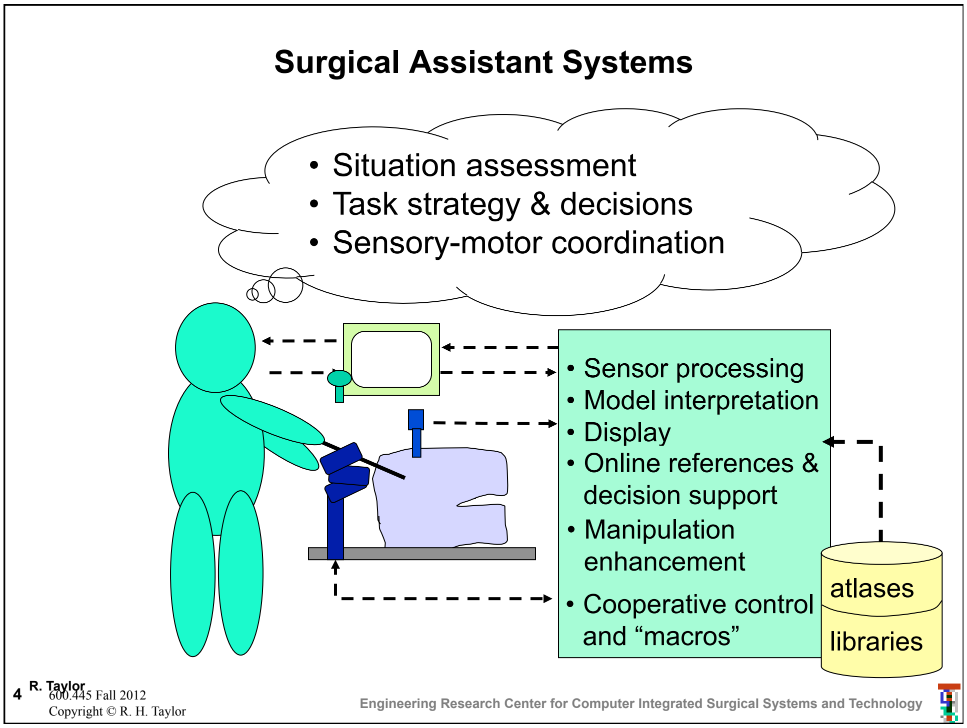
Statistical Analysis



Surgical Assistant Systems

- Situation assessment
- Task strategy & decisions
- Sensory-motor coordination

- Sensor processing
- Model interpretation
- Display
- Online references & decision support
- Manipulation enhancement
- Cooperative control and “macros”

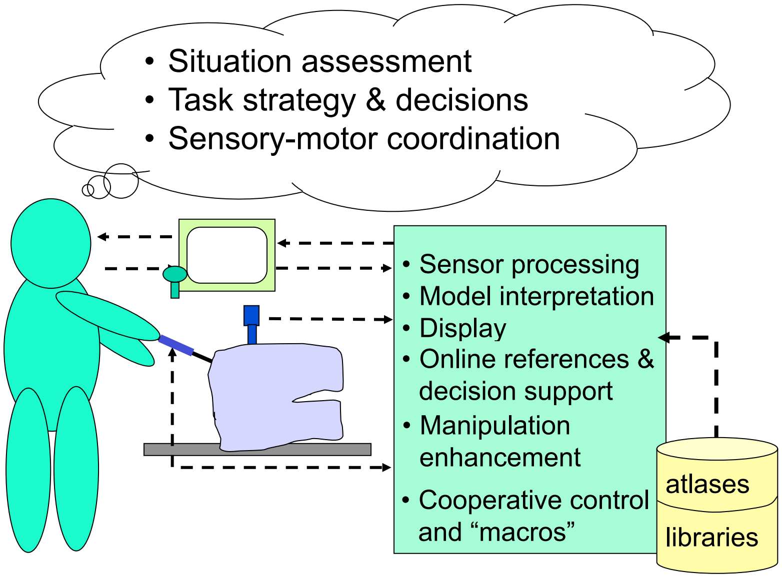


Surgical Assistant Systems

- Situation assessment
- Task strategy & decisions
- Sensory-motor coordination

- Sensor processing
- Model interpretation
- Display
- Online references & decision support
- Manipulation enhancement
- Cooperative control and “macros”

Surgical Assistant Systems



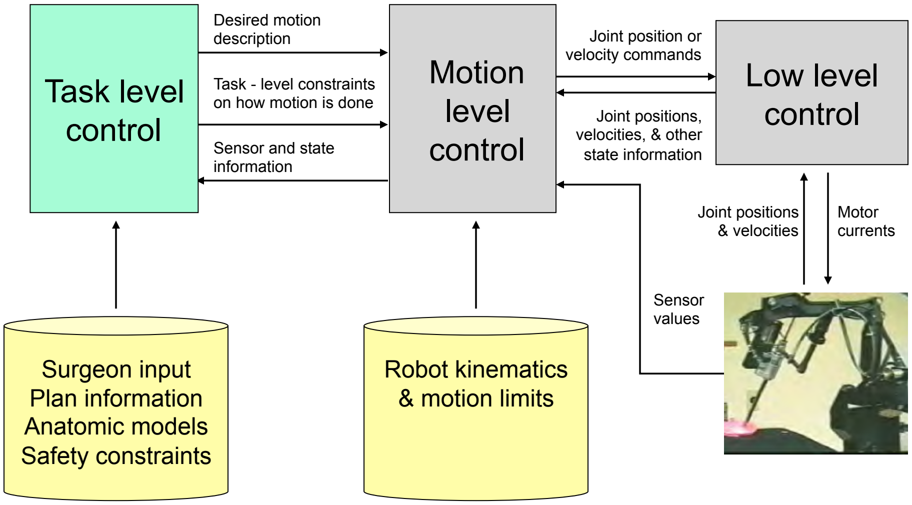
- Situation assessment
- Task strategy & decisions
- Sensory-motor coordination

- Sensor processing
- Model interpretation
- Display
- Online references & decision support
- Manipulation enhancement
- Cooperative control and “macros”

atlases

libraries

Problem: specifying motion for a [medical] robot



Task level control

Desired motion description

Task - level constraints on how motion is done

Sensor and state information

Motion level control

Joint position or velocity commands

Joint positions, velocities, & other state information

Low level control

Joint positions & velocities

Motor currents

Sensor values

Surgeon input
Plan information
Anatomic models
Safety constraints

Robot kinematics
& motion limits

Background: Jacobean Robot Motion Control

Let $\mathbf{F}=[\mathbf{R},\vec{\mathbf{p}}]$ be the current pose of a robot end effector and $\vec{\mathbf{q}}=[q_1,\cdots,q_N]$ be the current joint position values corresponding to **F**. I.e., $\mathbf{F}=Kins(\vec{\mathbf{q}})$, where $Kins(\cdots)$ is a function computing the "forward kinematics" of the robot.

$$\text{Pose } \mathbf{F}(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}}) = kins(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}})$$

$$\Delta\mathbf{F}\bullet\mathbf{F} = kins(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}})$$

$$\Delta\mathbf{F} = kins(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}})\,kins(\vec{\mathbf{q}})^{-1}$$

Background: Jacobean Robot Motion Control

Let $\mathbf{F}=[\mathbf{R},\vec{\mathbf{p}}]$ be the current pose of a robot end effector and $\vec{\mathbf{q}}=[q_1,\cdots,q_N]$ be the current joint position values corresponding to **F**. I.e., $\mathbf{F}=Kins(\vec{\mathbf{q}})$, where $Kins(\cdots)$ is a function computing the "forward kinematics" of the robot. Let $\Delta\mathbf{F}\bullet\mathbf{F}=Kins(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}})$

For small $\Delta\vec{\mathbf{q}}$, we can write the following expression for $\Delta\mathbf{F}=[Rot(\vec{\alpha}),\vec{\varepsilon}]$

$$\Delta\mathbf{F} = Kins(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}})Kins(\vec{\mathbf{q}})^{-1}$$

which we typically linearize as

$$\Delta\vec{\mathbf{x}} = \begin{bmatrix} \vec{\alpha} \\ \vec{\varepsilon} \end{bmatrix} \approx \mathbf{J}_{Kins}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$

Note that here we are computing $\Delta\mathbf{F}$ in the base frame of the robot. If we want to compute $\Delta\mathbf{F}$ in the end effector frame, so that $\mathbf{F}\bullet\Delta\mathbf{F}=Kins(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}})$, then we will get a slightly different expression for $\mathbf{J}_{Kins}(\vec{\mathbf{q}})$, though the flavor will be the same

Background: Jacobean Robot Motion Control

Joint positions $\vec{\mathbf{q}}$

Pose $\mathbf{F} = kins\left(\vec{\mathbf{q}}\right)$

$$\text{Pose } \mathbf{F}\left(\vec{\mathbf{q}} + \Delta\vec{\mathbf{q}}\right) = kins\left(\vec{\mathbf{q}} + \Delta\vec{\mathbf{q}}\right)$$

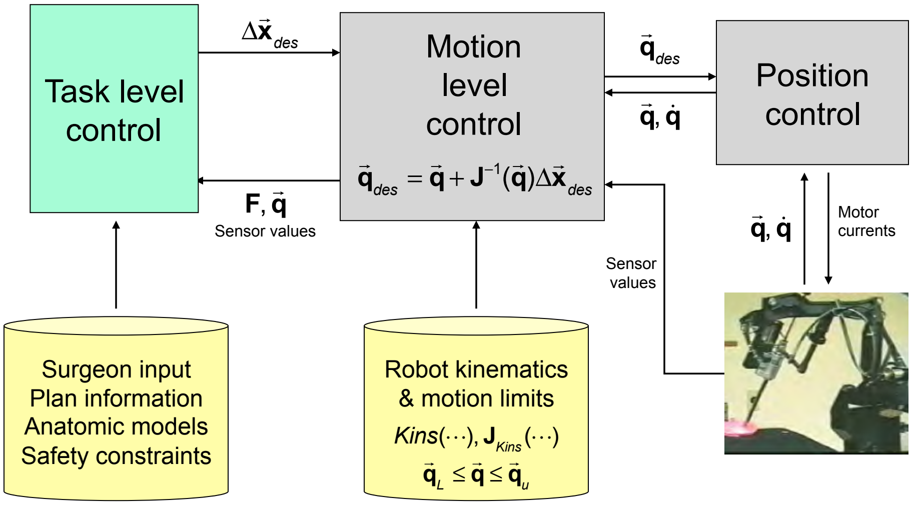
$$\Delta\mathbf{F} \bullet \mathbf{F} = kins\left(\vec{\mathbf{q}} + \Delta\vec{\mathbf{q}}\right)$$

$$\Delta\mathbf{F} = kins\left(\vec{\mathbf{q}} + \Delta\vec{\mathbf{q}}\right) kins\left(\vec{\mathbf{q}}\right)^{-1}$$

$$\begin{bmatrix} \vec{\alpha} \\ \varepsilon \end{bmatrix} \approx \mathbf{J}\left(\vec{\mathbf{q}}\right)\Delta\vec{\mathbf{q}}$$

$$\Delta\vec{\mathbf{q}} \approx \mathbf{J}\left(\vec{\mathbf{q}}\right)^{-1} \begin{bmatrix} \vec{\alpha} \\ \varepsilon \end{bmatrix}$$

One implementation



Task level control

$\Delta\vec{\mathbf{x}}_{des}$

Motion level control

$\vec{\mathbf{q}}_{des} = \vec{\mathbf{q}} + \mathbf{J}^{-1}(\vec{\mathbf{q}})\Delta\vec{\mathbf{x}}_{des}$

$\vec{\mathbf{q}}_{des}$

$\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}$

Position control

$\mathbf{F}, \vec{\mathbf{q}}$

Sensor values

$\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}$

Motor currents

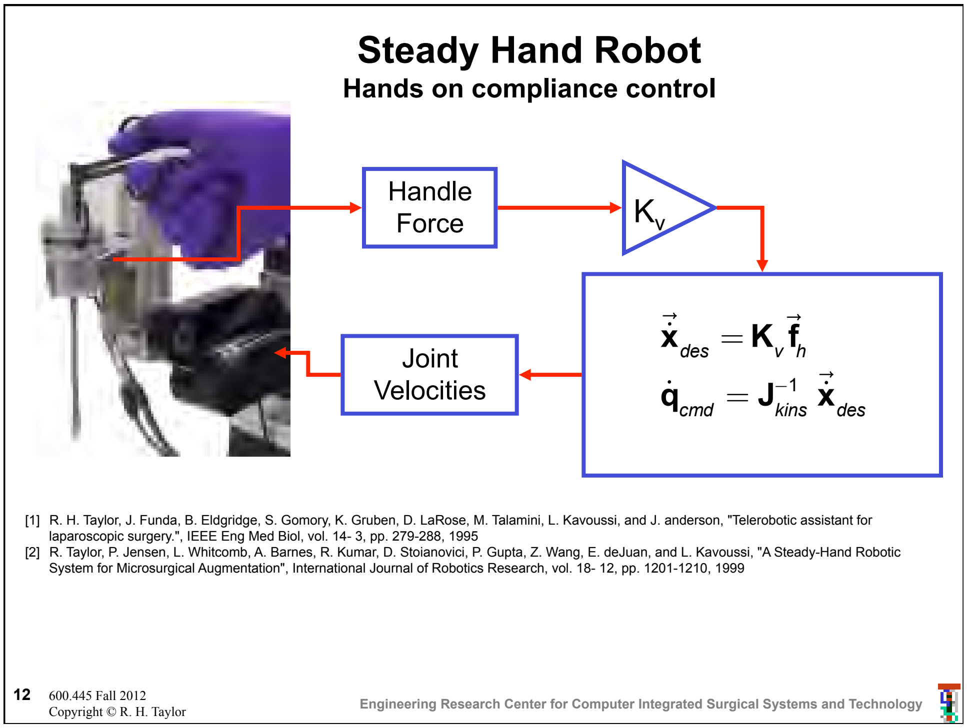
Sensor values

Surgeon input
Plan information
Anatomic models
Safety constraints

Robot kinematics & motion limits

$Kins(\cdots), \mathbf{J}_{Kins}(\cdots)$

$\vec{\mathbf{q}}_L \le \vec{\mathbf{q}} \le \vec{\mathbf{q}}_u$



Steady Hand Robot

Hands on compliance control

Handle Force

K_v

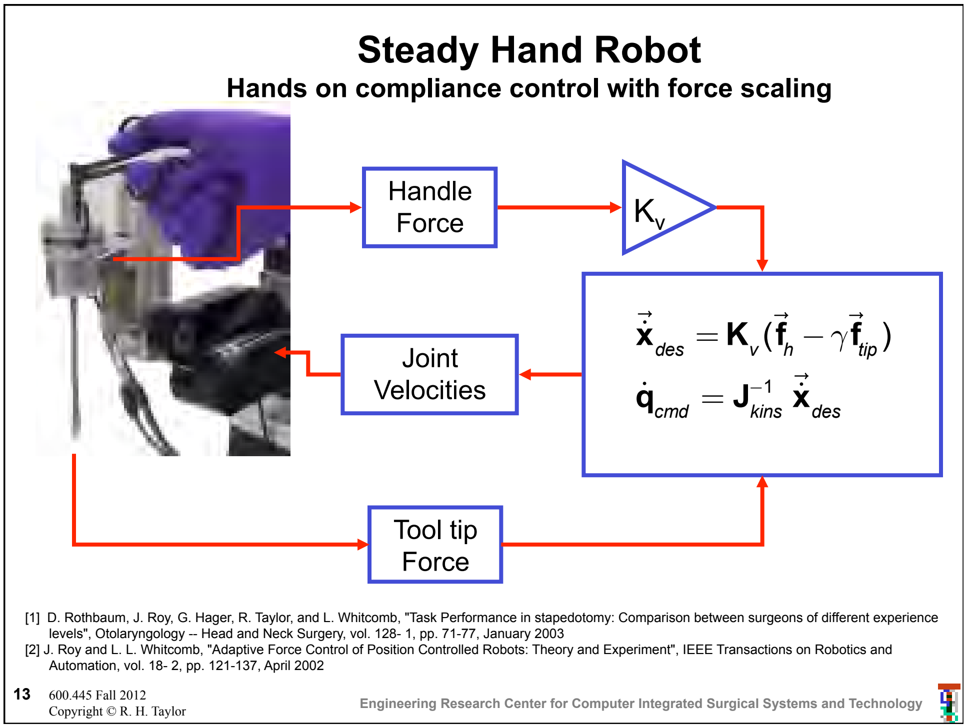
$$\dot{\vec{\mathbf{x}}}_{des} = \mathbf{K}_v \vec{\mathbf{f}}_h$$

$$\dot{\mathbf{q}}_{cmd} = \mathbf{J}^{-1}_{kins} \dot{\vec{\mathbf{x}}}_{des}$$

Joint Velocities

[1] R. H. Taylor, J. Funda, B. Eldgridge, S. Gomory, K. Gruben, D. LaRose, M. Talamini, L. Kavoussi, and J. anderson, "Telerobotic assistant for laparoscopic surgery.", IEEE Eng Med Biol, vol. 14- 3, pp. 279-288, 1995

[2] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. deJuan, and L. Kavoussi, "A Steady-Hand Robotic System for Microsurgical Augmentation", International Journal of Robotics Research, vol. 18- 12, pp. 1201-1210, 1999



Steady Hand Robot

Hands on compliance control with force scaling

Handle Force

K_v

$$\dot{\vec{\mathbf{x}}}_{des} = \mathbf{K}_v (\vec{\mathbf{f}}_h - \gamma \vec{\mathbf{f}}_{tip})$$

$$\dot{\mathbf{q}}_{cmd} = \mathbf{J}^{-1}_{kins} \dot{\vec{\mathbf{x}}}_{des}$$

Joint Velocities

Tool tip Force

[1] D. Rothbaum, J. Roy, G. Hager, R. Taylor, and L. Whitcomb, "Task Performance in stapedotomy: Comparison between surgeons of different experience levels", Otolaryngology -- Head and Neck Surgery, vol. 128- 1, pp. 71-77, January 2003

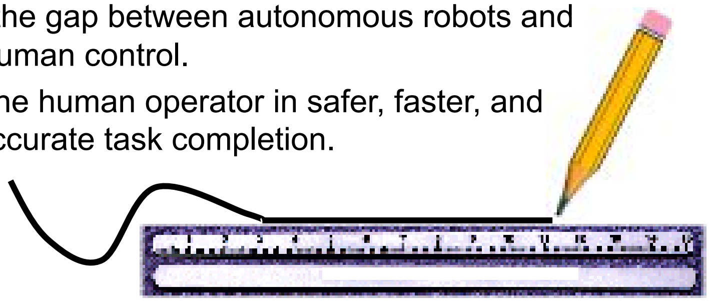
[2] J. Roy and L. L. Whitcomb, "Adaptive Force Control of Position Controlled Robots: Theory and Experiment", IEEE Transactions on Robotics and Automation, vol. 18- 2, pp. 121-137, April 2002

Example: Fenestratration of Stapes Footplate

Example: Fenestratration of Stapes Footplate

Virtual Fixtures

- Bridge the gap between autonomous robots and direct human control.
- Assist the human operator in safer, faster, and more accurate task completion.



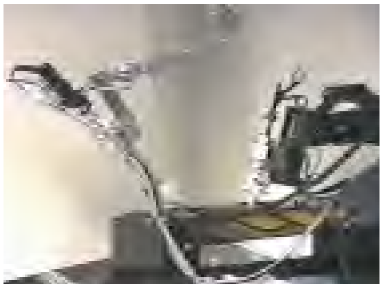
- Broadly Categorized
  - Guidance VF
  - Forbidden Region VF
- Different implementation
  - Tele-manipulation
  - Cooperative Control

Background: Virtual Fixtures

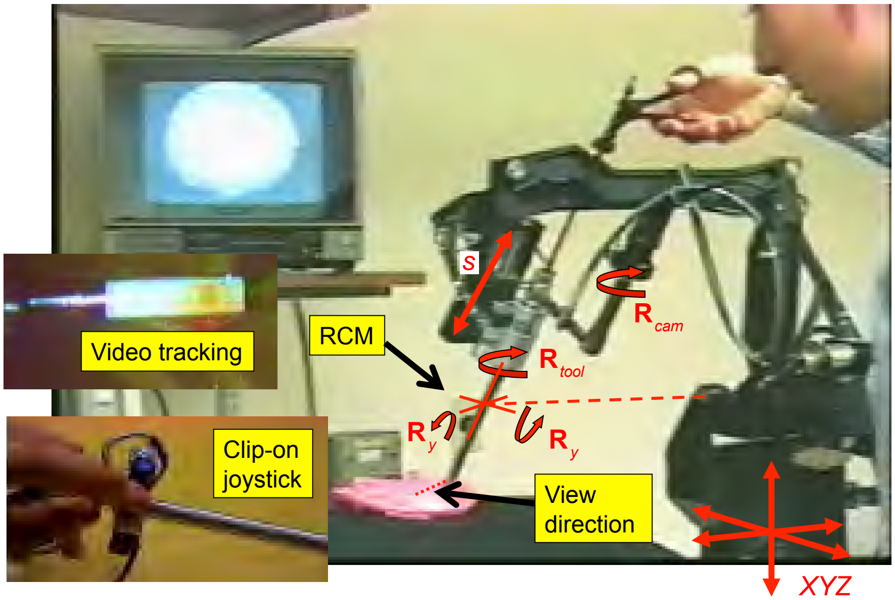
- First proposed for complex telerobotic tasks, but draw upon rich prior research in robot assembly and other manufacturing automation applications
- Many authors, e.g.,
  - L. B. Rosenberg, "Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation," *Proc. IEEE Virtual Reality International Symposium,* 1993.
  - B. Davies, S. Harris, M. Jakopec, K. Fan, and J. cobb, "Intraoperative application of a robotic knee surgery system", *MICCAI* 1999.
  - S. Park, R. D. Howe, and D. F. Torchiana, "Virtual Fixtures for Robotic Cardiac Surgery", *MICCAI* 2001.
  - S. Payandeh and Z. Stanisic, "On Application of Virtual Fixtures as an Aid for Telemanipulation and Training," *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2002.
- Discussion that follows draws upon work at IBM Research and within the CISST ERC at JHU. E.g.,
  - Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben, "Constrained Cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation,* vol. 12, pp. 453-466, 1996.
  - R. Kumar, An Augmented Steady Hand System for Precise Micromanipulation, Ph.D thesis in Computer Science, The Johns Hopkins University, Baltimore, 2001.
  - M. Li, M. Ishii, and R. H. Taylor, "Spatial Motion Constraints in Medical Robot Using Virtual Fixtures Generated by Anatomy," *IEEE Transactions on Robotics*, vol. 2, pp. 1270-1275, 2006.
  - A. Kapoor, M. Li, and R. H. Taylor "Constrained Control for Surgical Assistant Robots," in *IEEE Int. Conference on Robotics and Automation*, Orlando, 2006, pp. 231-236.
  - A. Kapoor and R. Taylor, "A Constrained Optimization Approach to Virtual Fixtures for Multi-Handed Tasks," in IEEE International Conference on Robotics and Automation (ICRA), Pasadena, 2008, pp. 3401-3406.
  - M. Li, *Intelligent Robotic Surgical Assistance for Sinus Surgery*, PhD Thesis in Computer Science Baltimore, Maryland: The Johns Hopkins University, 2005.
  - Ankur Kapoor, *Motion Constrained Control of Robots for Dexterous Surgical Tasks*, Ph.D. Thesis in Computer Science, The Johns Hopkins University, Baltimore, September 2007

Original Motivation for IBM Work

- Kinematic control of robots for MIS
- E.g., LARS and HISAR robots
- LARS and other IBM robots were kinematically redundant
  - Typically 7-9 actuated joints
- But tasks often imposed kinematic constraints
  - E.g., no lateral motion at trocar
- Some robots (e.g., IBM/JHU HISAR and CMI's AESOP) had passive joints
- General goals
  - Exploit redundancy in best way possible
  - Come as close as possible to providing desired motion subject to robot and task limits
- **Our approach:** view this as a constrained optimization problem



LARS degrees of freedom



LARS Video

Copyright © R. H. Taylor

LARS Video

Copyright © R. H. Taylor

Motion Specification Problem

- **Requirements**
  - The tool shaft must pass within a specified distance of the entry port into the patient's body
  - The individual joint limits may not be exceeded
- **Goals**
  - Aim the camera as close as possible at a target
    - ***or*** move view in direction indicated by clip-on pointing device
    - ***or*** move to track a video target on an instrument
    - ***or*** aim the working channel of the endoscope at a target
    - ***or*** something else (maybe a combination of goals)
  - Keep the view as "upright" as possible
  - Tool should pass as close as possible to entry port center
  - Keep joints far away from their limits, to preserve options for future motion
  - Minimize motion of XYZ joints
  - *Etc.*

Our approach: view as an optimization problem

- Currently formulate problem as constrained least squares problem
- Express goals in the objective function
- If multiple goals, objective function is a weighted sum of individual elements
- Add constraints for requirements
- Express constraints and objective function terms in whatever coordinate system is convenient
- Use Jacobean formulation to transform to joint space
- Solve for joint motion

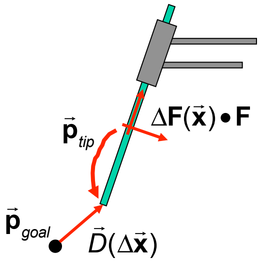
Example: keep tool tip near a point

$$\vec{D}(\vec{\mathbf{x}}) = \Delta\mathbf{F}(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\bullet\mathbf{F}\bullet\mathbf{p}_{tip} - \vec{\mathbf{p}}_{goal}$$

$$= \vec{\alpha}\times\vec{\mathbf{t}}+\vec{\varepsilon}+\vec{\mathbf{t}}-\vec{\mathbf{p}}_{goal} \quad \text{where} \quad \vec{\mathbf{t}}=\mathbf{F}\bullet\mathbf{p}_{tip}$$

$$\vec{\alpha}=\mathbf{J}_{\vec{\alpha}}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$

$$\vec{\varepsilon}=\mathbf{J}_{\vec{\varepsilon}}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$



Suppose we want to stay as close as possible while never going beyond 3mm from goal and also obeying joint limits

$$\Delta\mathbf{q}_{des}=\underset{\Delta\vec{\mathbf{q}}}{\arg\min}\ \left\|\vec{\mathrm{D}}(\Delta\vec{\mathbf{x}})\right\|^2=\left\|\vec{\alpha}\times\vec{\mathbf{t}}+\vec{\varepsilon}+\vec{\mathbf{t}}-\vec{\mathbf{p}}_{goal}\right\|^2$$

Subject to

$$\vec{\alpha}=\mathbf{J}_{\vec{\alpha}}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$

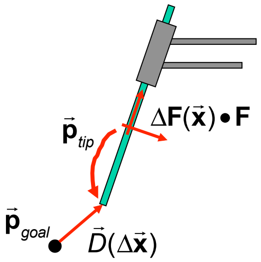
$$\vec{\varepsilon}=\mathbf{J}_{\vec{\varepsilon}}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$

$$\left\|\vec{\alpha}\times\vec{\mathbf{t}}+\vec{\varepsilon}+\vec{\mathbf{t}}-\vec{\mathbf{p}}_{goal}\right\|\leq 3$$

$$\vec{\mathbf{q}}_L-\vec{\mathbf{q}}\leq\Delta\vec{\mathbf{q}}\leq\vec{\mathbf{q}}_U-\vec{\mathbf{q}}$$

Example: keep tool tip near a point

Suppose we want to stay as close as possible while never going beyond 3mm from goal and also obeying joint limits, but we also want to minimize the change in direction of the tool shaft



$$\Delta\mathbf{q}_{des}=\underset{\Delta\vec{\mathbf{q}}}{\arg\min}\ \zeta\left\|\vec{\mathrm{D}}(\vec{\mathbf{x}}+\Delta\vec{\mathbf{x}})\right\|^2+\eta\ \left\|\alpha\times\mathbf{R}\bullet\vec{\mathbf{z}}\right\|^2$$

Subject to

$$\vec{\mathbf{x}}=\mathbf{F}\bullet\vec{\mathbf{p}}_{tip}$$

$$\vec{\mathrm{D}}(\vec{\mathbf{x}}+\Delta\vec{\mathbf{x}})=\vec{\alpha}\times\vec{\mathbf{t}}+\vec{\varepsilon}+\vec{\mathbf{x}}-\vec{\mathbf{p}}_{goal}$$

$$\vec{\alpha}=\mathbf{J}_{\vec{\alpha}}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}\ ;\ \vec{\varepsilon}=\mathbf{J}_{\vec{\varepsilon}}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$

$$\left\|\vec{\mathrm{D}}(\vec{\mathbf{x}}+\Delta\vec{\mathbf{x}})\right\|\leq 3$$

$$\vec{\mathbf{q}}_L-\vec{\mathbf{q}}\leq\Delta\vec{\mathbf{q}}\leq\vec{\mathbf{q}}_U-\vec{\mathbf{q}}$$

$$\vec{\mathbf{x}}=\mathbf{F}\bullet\vec{\mathbf{p}}_{tip}$$

Solving the optimization problem

- **Constrained linear least squares**
  - Combine constraints and goals from task and robot control
  - Linearize and constrained least squares problem

$$\Delta\vec{\mathbf{q}}_{des} = \arg\min_{\Delta\vec{\mathbf{q}}} \left\| \mathbf{E}_{task}\Delta\vec{\mathbf{x}} - \vec{\mathbf{f}}_{task} \right\|^2 + \left\| \mathbf{E}_{\vec{\mathbf{q}}}\Delta\vec{\mathbf{x}} - \vec{\mathbf{f}}_{\vec{\mathbf{q}}} \right\|^2$$

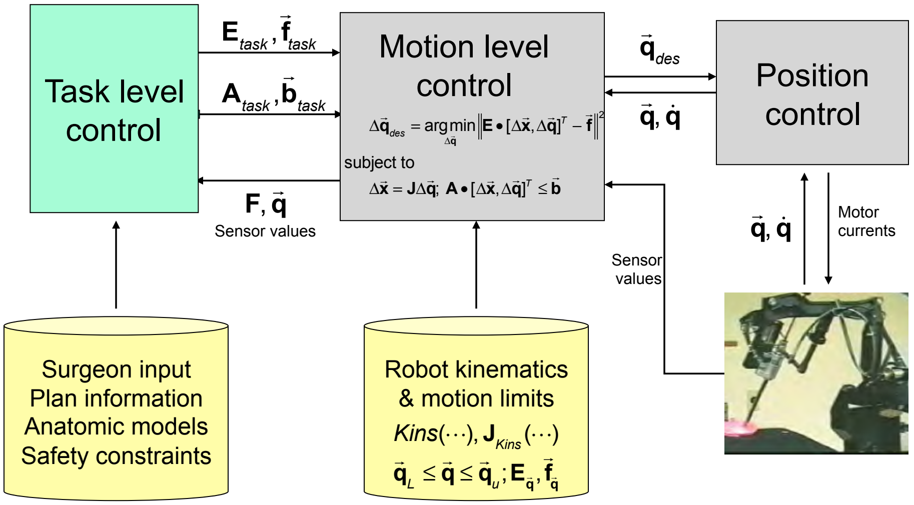
subject to

$$\Delta\vec{\mathbf{x}} = \mathbf{J}\Delta\vec{\mathbf{q}};\ \ \mathbf{A}_{task}\Delta\vec{\mathbf{x}} \le \vec{\mathbf{b}}_{task}\ \ ;\ \mathbf{A}_{\vec{\mathbf{q}}}\Delta\vec{\mathbf{q}} \le \vec{\mathbf{b}}_{\vec{\mathbf{q}}}$$

  - E.g., using "non-negative least squares" methods developed by Lawson and Hanson
  - Approach used in our IBM work and in Kumar, Li, Kapoor theses

- **Constrained nonlinear least squares**
  - Approach explored by Kapoor (discuss later)

Linear least squares implementation



Task level control

$\mathbf{E}_{task}, \vec{\mathbf{f}}_{task}$

$\mathbf{A}_{task}, \vec{\mathbf{b}}_{task}$

$\mathbf{F}, \vec{\mathbf{q}}$

Sensor values

Motion level control

$\Delta\vec{\mathbf{q}}_{des} = \arg\min_{\Delta\vec{\mathbf{q}}} \left\| \mathbf{E} \bullet [\Delta\vec{\mathbf{x}}, \Delta\vec{\mathbf{q}}]^T - \vec{\mathbf{f}} \right\|^2$

subject to

$\Delta\vec{\mathbf{x}} = \mathbf{J}\Delta\vec{\mathbf{q}};\ \mathbf{A} \bullet [\Delta\vec{\mathbf{x}}, \Delta\vec{\mathbf{q}}]^T \le \vec{\mathbf{b}}$

$\vec{\mathbf{q}}_{des}$

$\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}$

Position control

$\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}$

Motor currents

Sensor values

Surgeon input
Plan information
Anatomic models
Safety constraints

Robot kinematics
& motion limits
$Kins(\cdots), \mathbf{J}_{Kins}(\cdots)$
$\vec{\mathbf{q}}_L \le \vec{\mathbf{q}} \le \vec{\mathbf{q}}_u; \mathbf{E}_{\vec{\mathbf{q}}}, \vec{\mathbf{f}}_{\vec{\mathbf{q}}}$

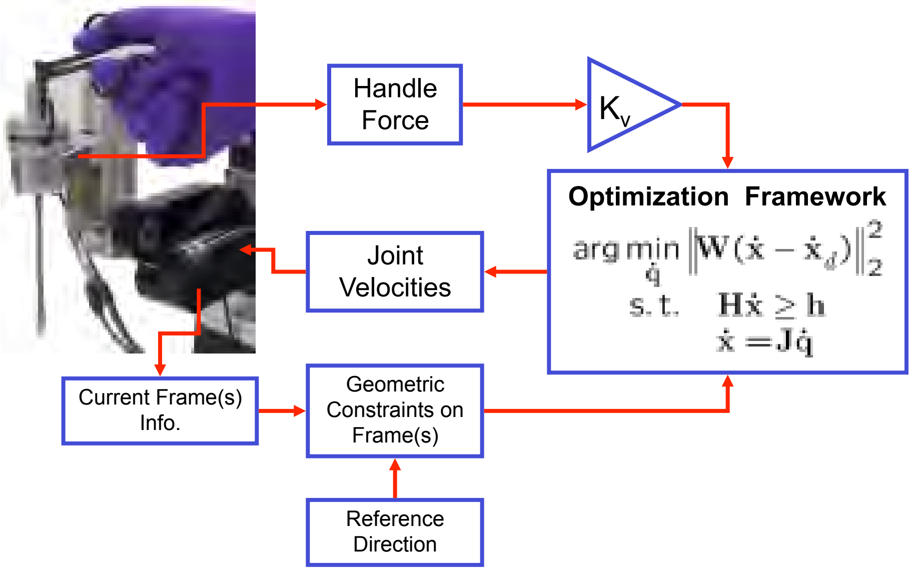
Some IBM Movies

Early Constrained Motion System (LapSYS)

Vision-guided targeting

Steady Hand Robot

High Level Constrained Control



Handle Force

K_v

Optimization Framework

$$\arg\min_{\dot{q}} \left\| W(\dot{x} - \dot{x}_d) \right\|_2^2$$
$$\text{s.t.} \quad H\dot{x} \geq h$$
$$\dot{x} = J\dot{q}$$

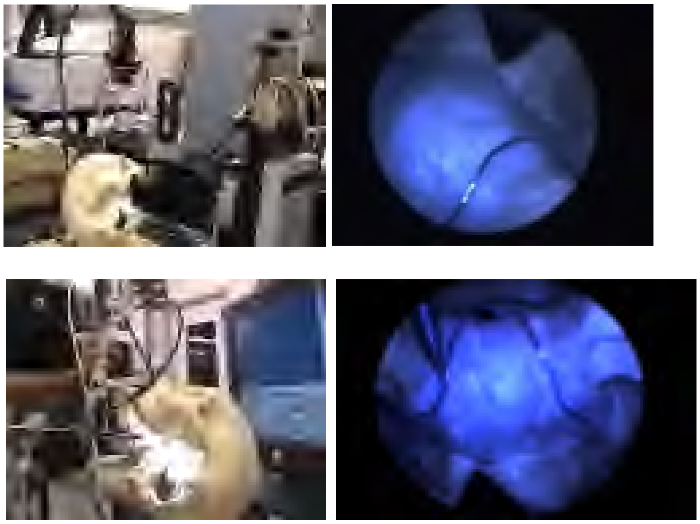
Joint Velocities

Current Frame(s) Info.

Geometric Constraints on Frame(s)

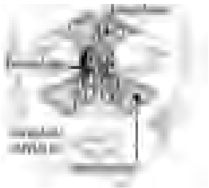
Reference Direction

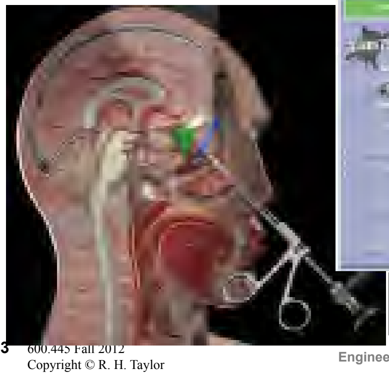
Sample task: steady hand path tracing



M. Li *et al.*

Background: endoscopic sinus surgery

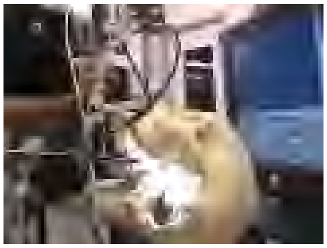


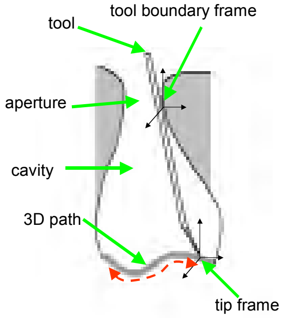


Kennedy, D.W., W.E. Bolger , S. J. Zinreich , J. Zinreich, *Diseases of the Sinuses: Diagnosis and Management.* 2001.

Goal: robotically-assisted sinus surgery

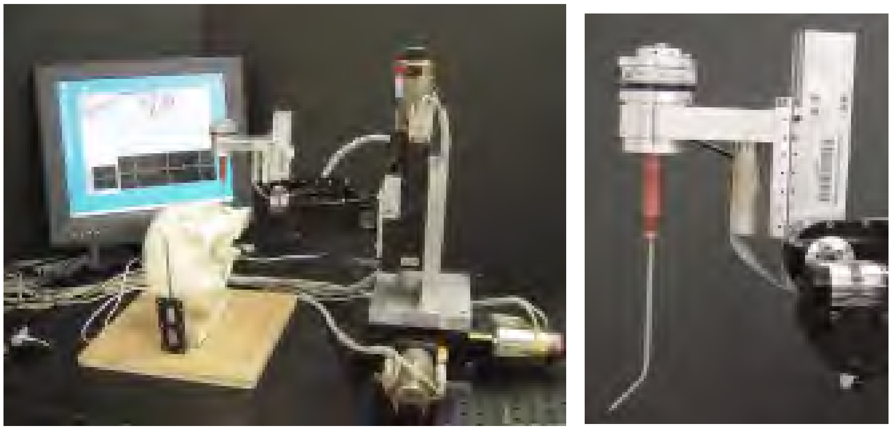
- **Difficulties with conventional approach**
  - Complicated geometry
  - Safety-critical structures
  - Limited work space
  - Awkward tools
- **Our approach**
  - Cooperatively controlled “Steady hand” robot
  - Registered to CT models
  - “Virtual fixtures” automatically derived from models





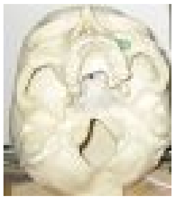
M. Li *et al.*

Experiment Setup

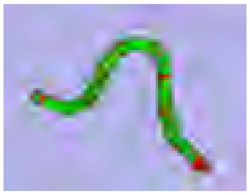


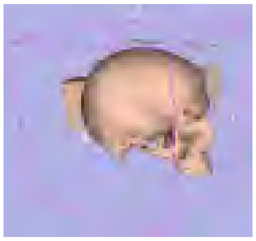
M. Li *et al.*

Experimental setup



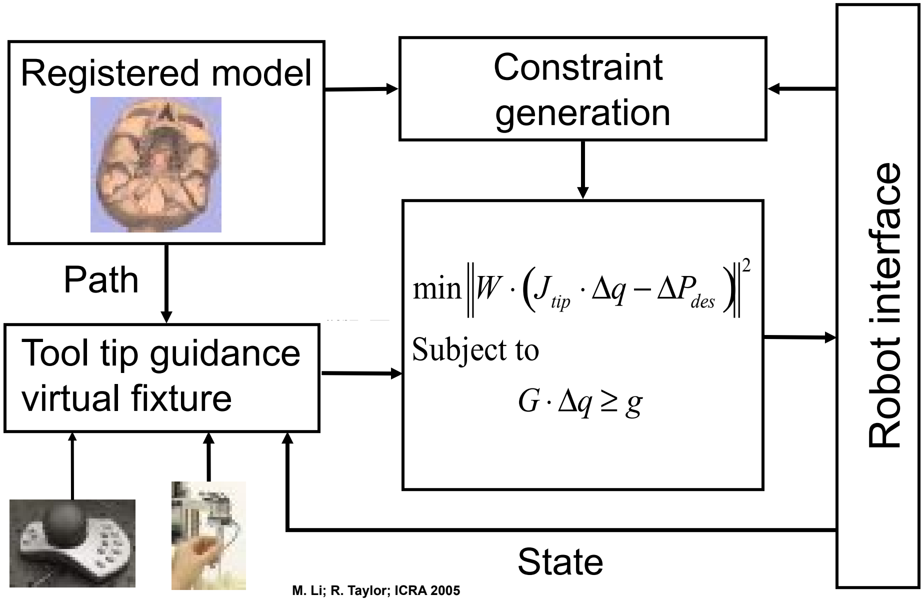
- Plastic Skull Phantom
  - Target path defined by embedded wire
  - Radioopaque fiducials implanted on skull for registration
- Computer model
  - Extracted from CT scan using standard software (Slicer)
- 3D tracking of tools, etc. using Northern Digital Optotrak®
- Co-register model, robot, and optical tracker using standard techniques





M. Li *et al.*

Virtual Fixture Online Implementation



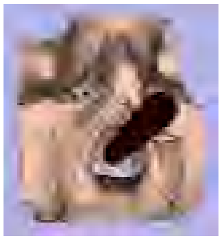
$$\min \left\| W \cdot \left(J_{tip} \cdot \Delta q - \Delta P_{des} \right) \right\|^2$$

Subject to

$$G \cdot \Delta q \ge g$$

M. Li; R. Taylor; ICRA 2005

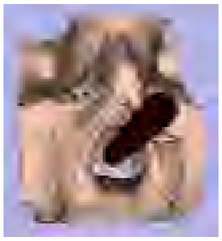
Boundary Constraints Generation



- Anatomy – triangulated surface models
  - Patient-specific model of nose & sinus derived from CT
  - High complexity: 182,000 triangles & 99,000 vertices
- Tool shaft -- cylinder
- The boundary constraint generation requires us to find close-point pairs between boundary surface model & tool shaft

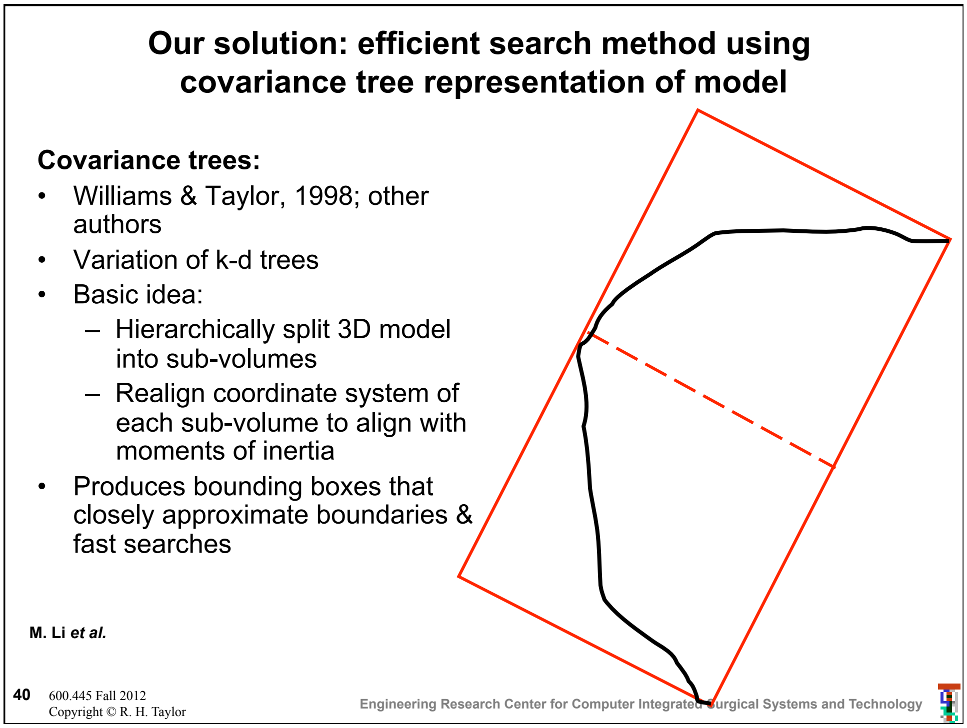
M. Li *et al.*

Boundary Constraints Generation



- Anatomy – triangulated surface models
  - Patient-specific model of nose & sinus derived from CT
  - High complexity: 182,000 triangles & 99,000 vertices
- Tool shaft -- cylinder
- The boundary constraint generation requires us to find close-point pairs between boundary surface model & tool shaft
- **Problem: How can we generate the right constraints in real time???**

M. Li *et al.*



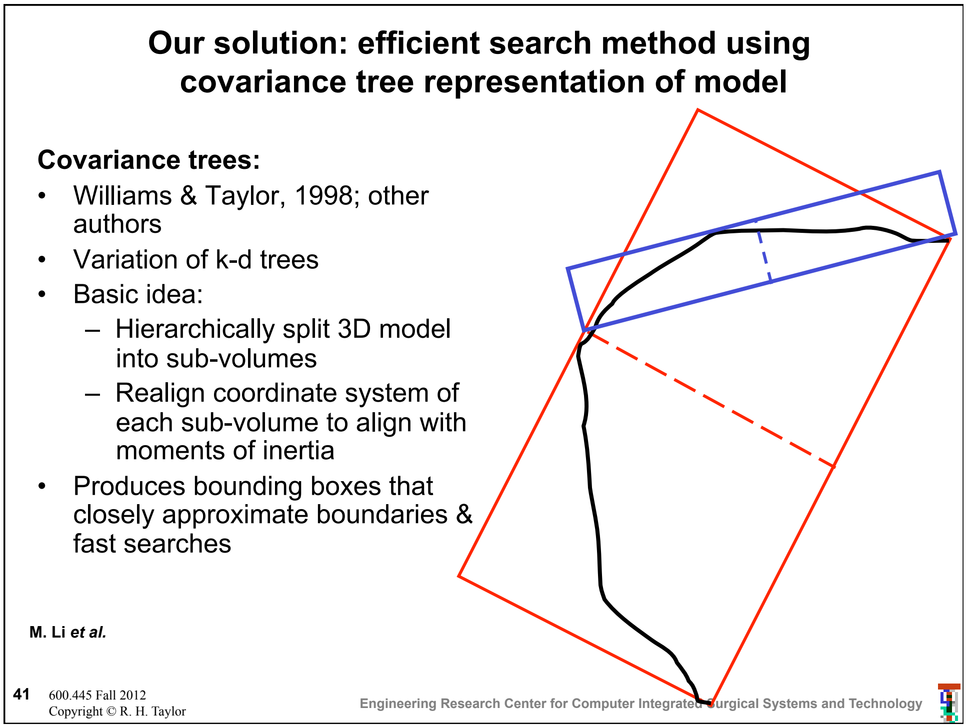
Our solution: efficient search method using covariance tree representation of model

Covariance trees:

- Williams & Taylor, 1998; other authors
- Variation of k-d trees
- Basic idea:
  - Hierarchically split 3D model into sub-volumes
  - Realign coordinate system of each sub-volume to align with moments of inertia
- Produces bounding boxes that closely approximate boundaries & fast searches

M. Li *et al.*

600.445 Fall 2012
Copyright © R. H. Taylor



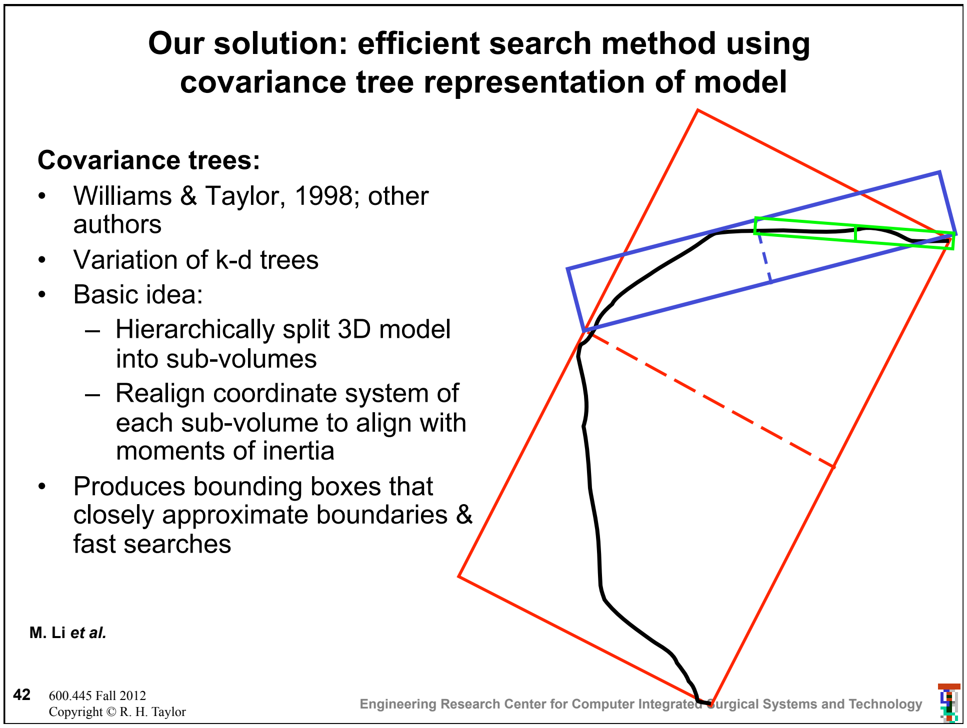
Our solution: efficient search method using covariance tree representation of model

Covariance trees:

- Williams & Taylor, 1998; other authors
- Variation of k-d trees
- Basic idea:
  - Hierarchically split 3D model into sub-volumes
  - Realign coordinate system of each sub-volume to align with moments of inertia
- Produces bounding boxes that closely approximate boundaries & fast searches

M. Li *et al.*

600.445 Fall 2012
Copyright © R. H. Taylor

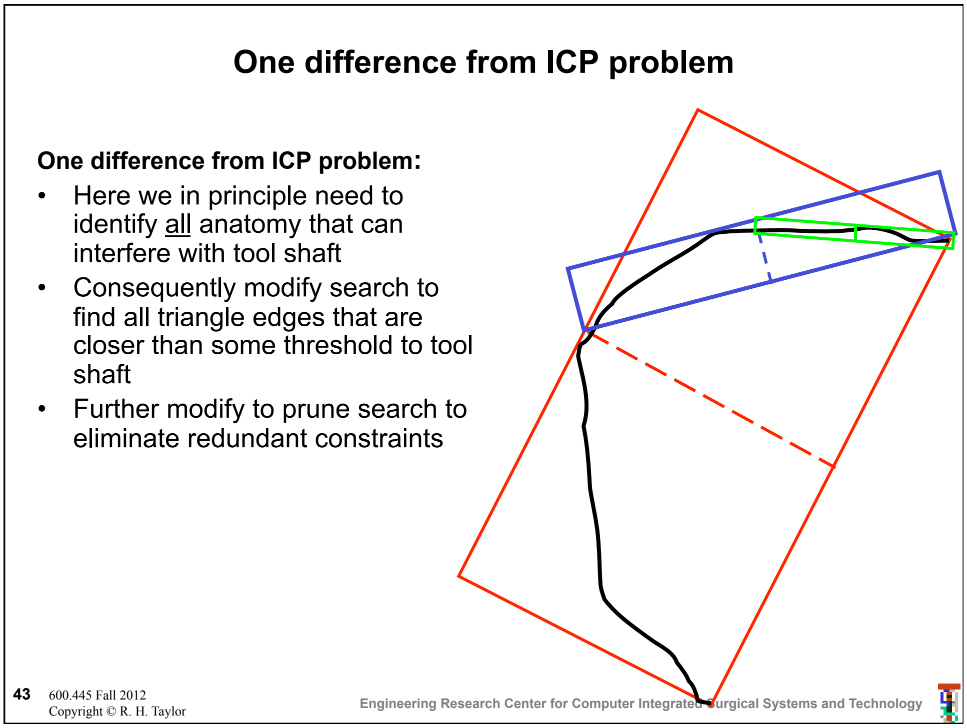


Our solution: efficient search method using covariance tree representation of model

Covariance trees:

- Williams & Taylor, 1998; other authors
- Variation of k-d trees
- Basic idea:
  - Hierarchically split 3D model into sub-volumes
  - Realign coordinate system of each sub-volume to align with moments of inertia
- Produces bounding boxes that closely approximate boundaries & fast searches

M. Li *et al.*



One difference from ICP problem

One difference from ICP problem:

- Here we in principle need to identify all anatomy that can interfere with tool shaft
- Consequently modify search to find all triangle edges that are closer than some threshold to tool shaft
- Further modify to prune search to eliminate redundant constraints

Control Implementation

- Formulate constrained least squares problem
- Constraints & objective function include terms for desired tip motion, joint limits, boundary constraints

$$\zeta = \min_{\Delta q} \left\| \begin{bmatrix} W_{tip} & & \\ & W_k & \\ & & W_{joints} \end{bmatrix} \cdot \left(\begin{bmatrix} J_{tip}(q) \\ J_k(q) \\ I \end{bmatrix} \Delta q - \begin{bmatrix} \Delta P_{tip-des} \\ 0 \\ 0 \end{bmatrix} \right) \right\|$$

$$\text{subject to} \quad \begin{bmatrix} H_{tip} & & \\ & H_k & \\ & & H_{joints} \end{bmatrix} \cdot \begin{bmatrix} J_{tip}(q) \\ J_k(q) \\ I \end{bmatrix} (\Delta q) \geq \begin{bmatrix} h_{tip} \\ h_k \\ h_{joints} \end{bmatrix}$$

M. Li *et al.*

Control Implementation

- Tip frame $\Delta P_{tip} = J_{tip}(q) \cdot \Delta q$

$\left\| \Delta P_{tip} - \Delta P_{tip-des} \right\|$

$\Delta P_{tip_d}{}^{T} \cdot \Delta P_{tip} \geq THD$

min $\zeta_{tip} = \left\| W_{tip} \cdot \left(J_{tip}(q) \Delta q - \Delta P_{tip-des} \right) \right\|$

subject to $H_{tip-des} J_{tip}(q) \Delta q \geq h_{tip}$

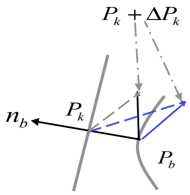
- Boundary constraint $\Delta P_k = J_k(q) \cdot \Delta q$

$\left\| W_k \cdot \Delta P_k \right\|$

$n_b{}^{T} \cdot (P_k + \Delta P_k - P_b) \geq d$

min $\zeta_k = \left\| W_k J_k(q) \Delta q \right\|$

subject to $H_k J_k(q) \Delta q \geq h_k$



- Joints limitation

$\left\| W_{joint} \cdot \Delta q \right\|$

$q_{\min} - q \leq \Delta q \leq q_{\max} - q$

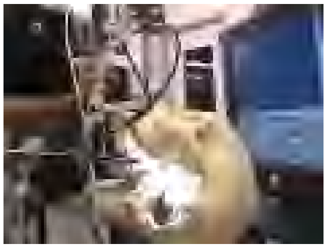
min $\zeta_{joints} = \left\| W_{joints} \Delta q \right\|$

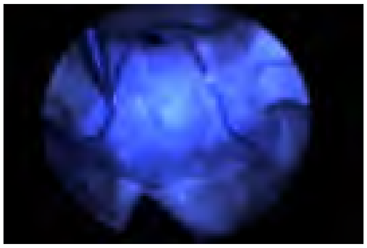
subject to $H_{joints} \Delta q \geq h_{joints}$

M. Li *et al.*

Control implementation

- Solve problem numerically with standard methods (Lawson & Hanson, 1974)
- Performance:
  - 6 ms/iteration on 2GHz Pentium 4 PC
  - Typically 20 to 39 constraints

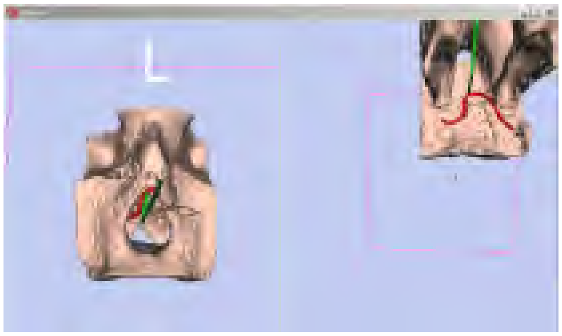




M. Li *et al.*

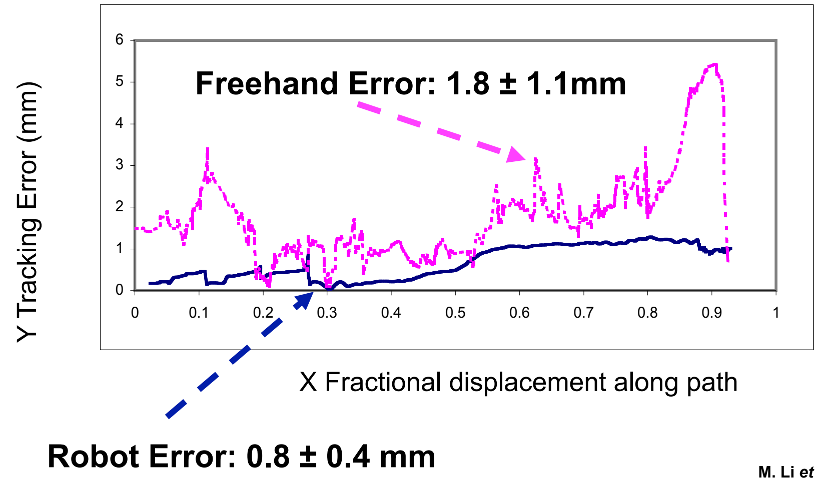
Results

The average time in each control loop for the boundary searching is ~6ms



M. Li *et al.*

Results: Robot *vs* Freehand



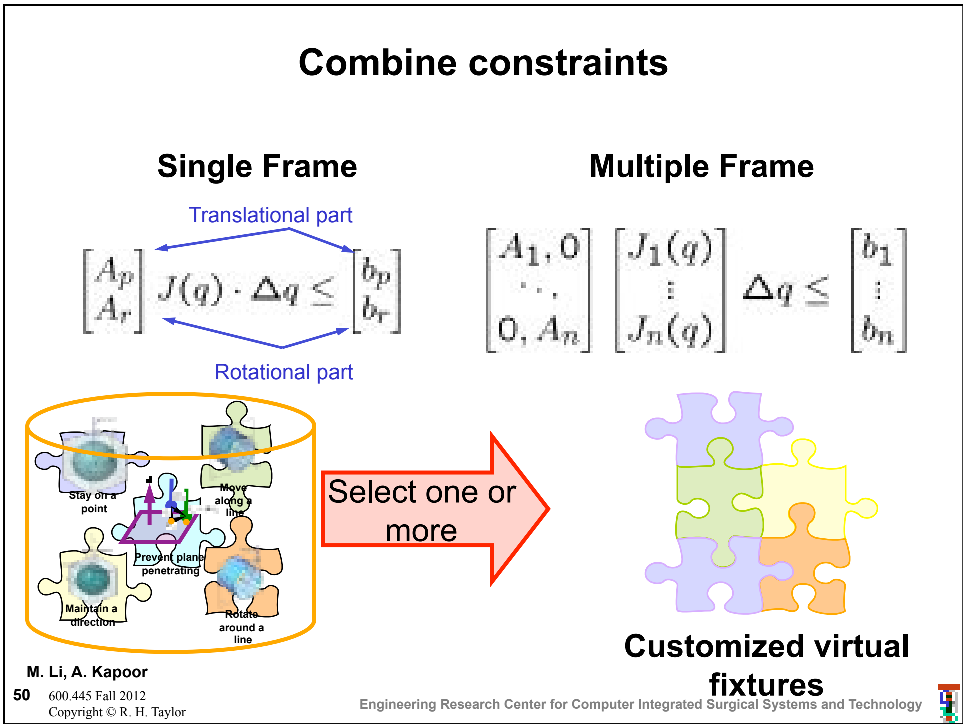
M. Li *et al.*

Results: Robot *vs* Freehand

Trial#	Free hand		Robot Guidance	
	Average Error (mm)	Average Time (s)	Average Error (mm)	Average Time (s)
1	1.705	26.354	0.736	10.972
2	1.632	29.358	0.757	15.275
3	1.796	27.372	0.765	16.29
4	2.061	25.436	0.779	19.439
5	2.119	24.533	0.777	16.209
avg	1.819	26.611	0.763	17.237
std	1.126	1.863	0.395	1.848

Approx 1.5:1 improvement in time!

M. Li *et al.*



Combine constraints

Single Frame

Translational part

$$\begin{bmatrix}A_p\\ A_r\end{bmatrix} J(q)\cdot\Delta q \le \begin{bmatrix}b_p\\ b_r\end{bmatrix}$$

Rotational part

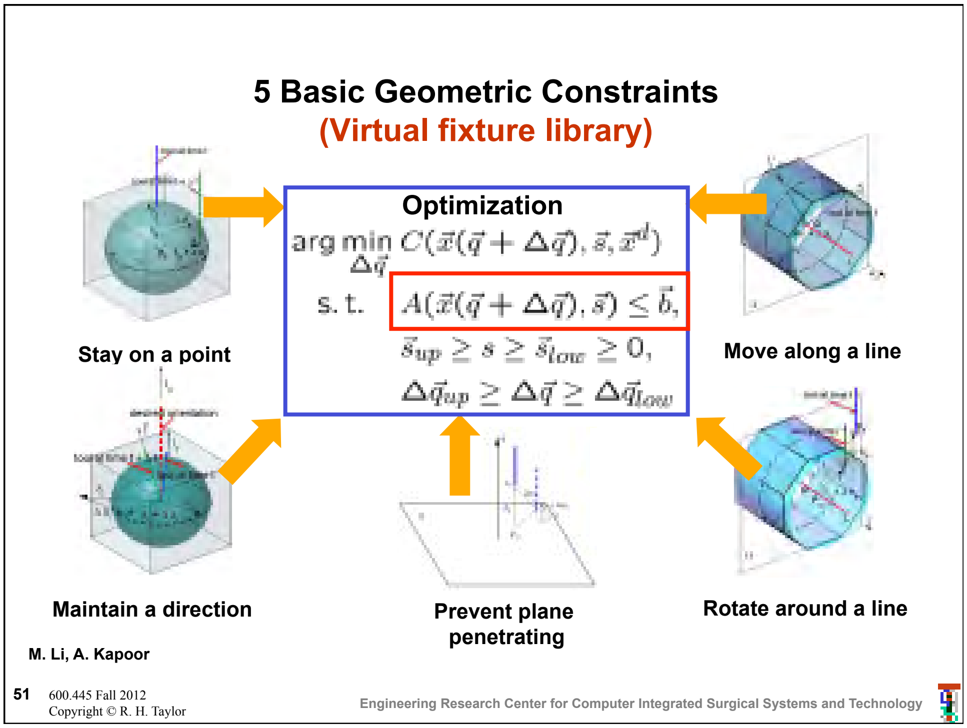
Multiple Frame

$$\begin{bmatrix}A_1, 0\\ \ddots\\ 0, A_n\end{bmatrix}\begin{bmatrix}J_1(q)\\ \vdots\\ J_n(q)\end{bmatrix}\Delta q \le \begin{bmatrix}b_1\\ \vdots\\ b_n\end{bmatrix}$$

Select one or more

Customized virtual fixtures

M. Li, A. Kapoor



5 Basic Geometric Constraints
(Virtual fixture library)

Optimization

$$\arg\min_{\Delta\vec{q}} C(\vec{x}(\vec{q}+\Delta\vec{q}), \vec{s}, \vec{x}^d)$$

$$\text{s.t.}\quad \boxed{A(\vec{x}(\vec{q}+\Delta\vec{q}), \vec{s}) \le \vec{b},}$$

$$\vec{s}_{up} \ge \vec{s} \ge \vec{s}_{low} \ge 0,$$

$$\Delta\vec{q}_{up} \ge \Delta\vec{q} \ge \Delta\vec{q}_{low}$$

Stay on a point

Move along a line

Maintain a direction

Prevent plane penetrating

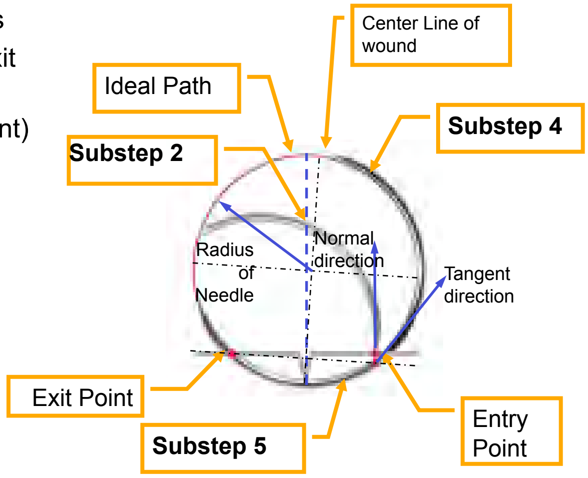
Rotate around a line

M. Li, A. Kapoor

Example: Suturing

The suturing task involves

- Select entry and exit points
- Align (Move & Orient) Needle
- Bite: Pass Needle
- Loop
- Knot

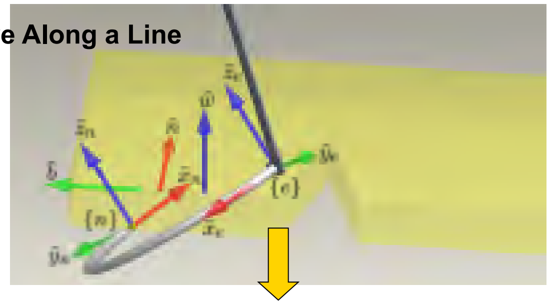


M. Li, A. Kapoor

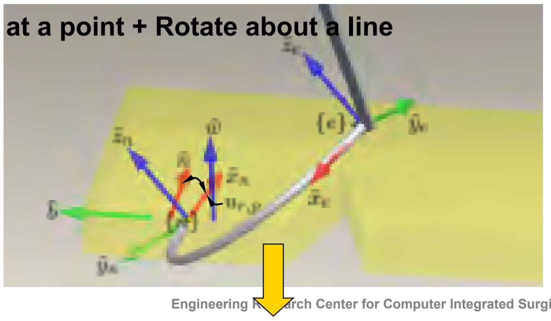
Copyright © R. H. Taylor

Suturing: Align Step

0. Move Along a Line



1. Stay at a point + Rotate about a line

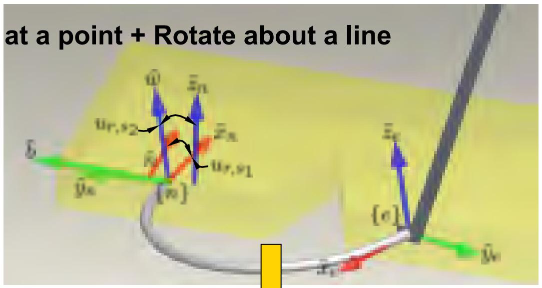


M. Li, A. Kapoor

Copyright © R. H. Taylor

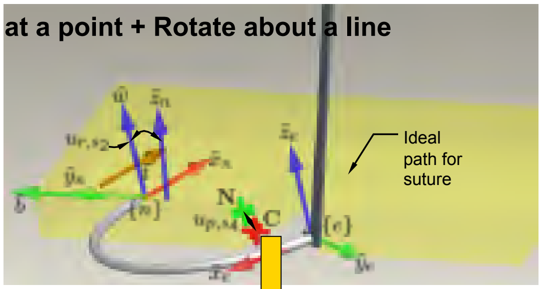
Suturing: Align Step

2. Stay at a point + Rotate about a line



3. Puncture

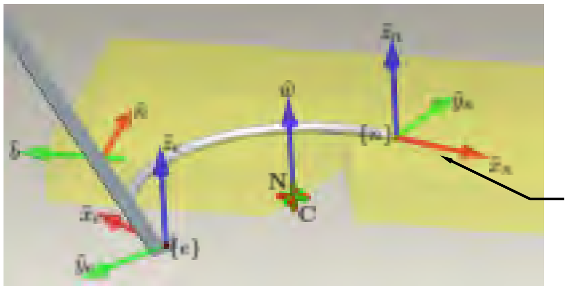
4. Stay at a point + Rotate about a line



Ideal path for suture

M. Li, A. Kapoor

Suturing: Bite Step



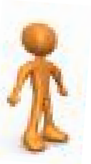
Ideal path for suture

- Ideal trajectory is a circle with radius equal to needle radius.
- Needle plane is parallel to entry and exit points and surface normal.

M. Li, A. Kapoor

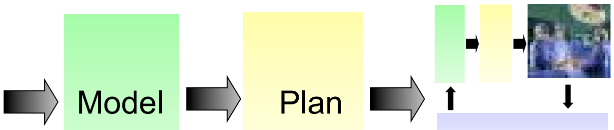
Example: “Virtual fixtures” for suturing assistance

Information



Patient-specific Information (Images, lab results, genetics, etc.)

General information (anatomic atlases, statistics, rules)



M. Li, A. Kapoor, *et al*

Suturing: Results

The average error (mm) in ideal and actual points as measured by OptoTrak®
Preliminary data collected from 4 users 5 trials each.

Error	**Entry (mm)**	**Exit (mm)**
Robot	0.6375; σ = 0.12	0.7742; σ = 0.37
Manual	--	2.1; σ = 1.2

- Suturing task using VF showed significant improvement in performance over freehand.
  - Can be performed at awkward angles
  - Avoids multiple trials and large undesirable movements inside tissue.

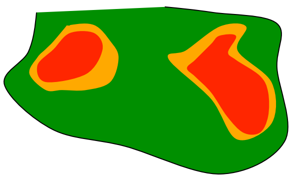
M. Li, A. Kapoor

Hard and soft constraints

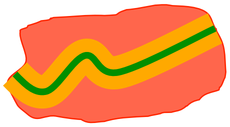
🟩 Preferred region

🟧 Safety region

🟥 Forbidden region



Avoidance



Line following

- Constraints on the task can be "hard" or "soft"
- The relative sizes depend on the procedure, ranging from micros to tenths of millimeter.
- Soft constraints allow the controller to accommodate uncertainties inherent in surgical procedures.

Thanks: **A. Kapoor**

Copyright © R. H. Taylor

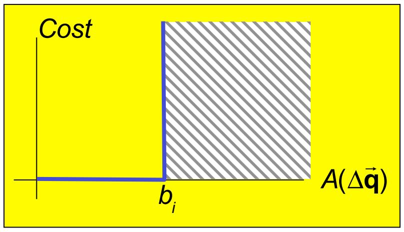
"Soft" constraint implementation

Suppose that we have a problem of the form

$$\Delta\vec{\mathbf{q}}_{des} = \arg\min \left\| E(\Delta\vec{\mathbf{q}}) \right\|^2$$

subject to a constraint of the form

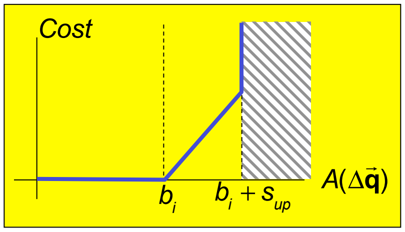
$$A_i(\Delta\vec{\mathbf{q}}) \le b_i$$



Copyright © R. H. Taylor

"Soft" constraint implementation

But suppose we want to make the barrier "soft". I.e., allow the robot to go beyond the barrier at increasing cost until it hits a harder barrier later



Add an explicit slack s_i and add a penalty term to the objective function

$$\Delta\vec{\mathbf{q}}_{des} = \arg\min \left\| E(\Delta\vec{\mathbf{q}}) \right\|^2 + \eta_i s_i^2$$

subject to a constraint of the form

$$A_i(\Delta\vec{\mathbf{q}}) - s_i \le b_i$$

$$0 \le s_i \le s_{up,i}$$

This process can be repeated several times to produce progressively steeper costs

---

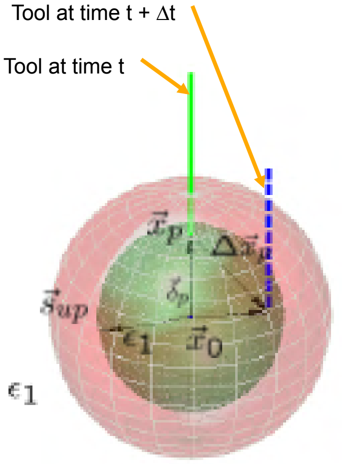
Example: Stay near a point

Target Position: $\vec{x}_0$

After incremental motion

$\vec{x}_p + \Delta\vec{x}_p$ close to $\vec{x}_0$

We want…



$$A(\vec{x}, s) = \|\vec{\delta}_p + \Delta\vec{x}_p\|^2 - s \le \epsilon_1$$

where $\vec{\delta}_p = \vec{x}_p - \vec{x}_0$

A. Kapoor, ***et al.***

Using Linear Constrained Quadratic Optimization

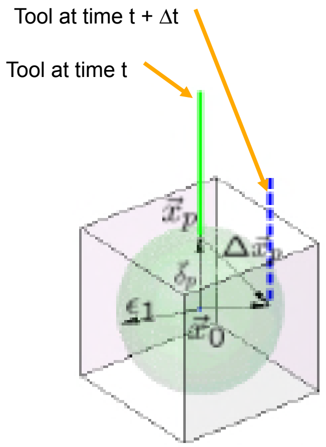
Matrix representation

$$A \cdot \Delta\vec{x} - s \leq b$$

Use Constrained Least Squares to solve

$$\underset{\Delta\vec{q}}{\arg\min} \quad \|\Delta\vec{x} - \Delta\vec{x}^d\|^2$$

$$s.t \quad A \cdot \Delta\vec{x} - s \leq b$$

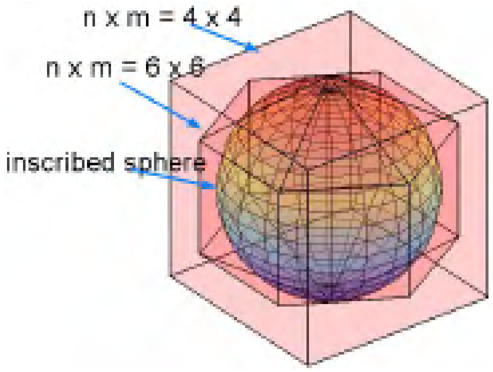

Tool at time t + Δt

Tool at time t

A. Kapoor, *et al.*

Linear approximation for constraints

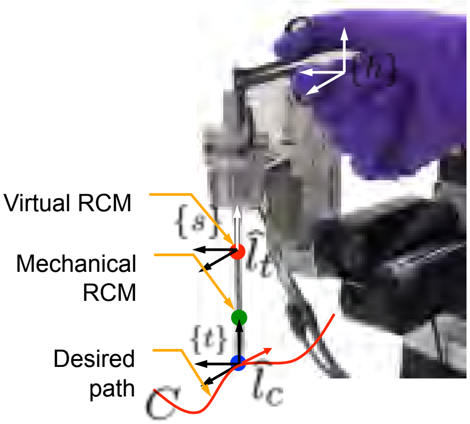
- n x m increase
  - Polyhedron approaches the inscribed sphere
  - Linearized conditions are a better approximation
  - More constraints require more time to solve the optimization problem
- Symmetrical polyhedron
  - nxm = 4x4
- Bounded polyhedron
  - nxm = 3x3



A. Kapoor, *et al.*

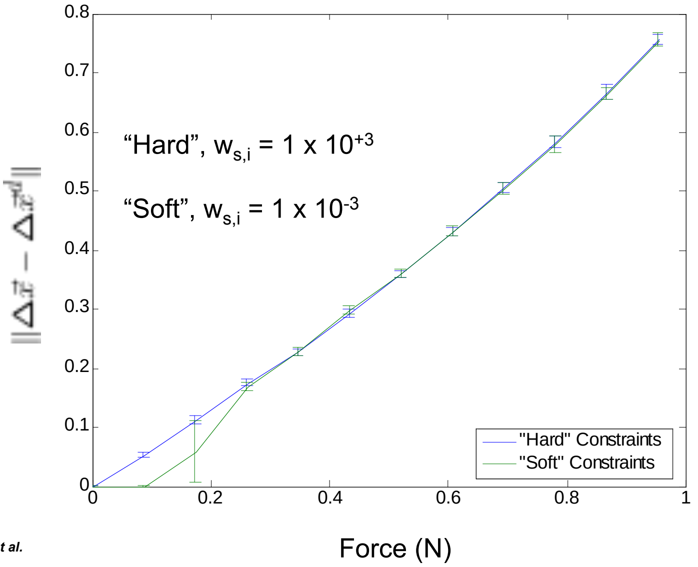
Example Task

- Constraint 1: Tip to move along curve C
- Constraint 2: Origin of {s} to move along
- Objective: Handle to follow user input



A. Kapoor, ***et al.***

Results for Example Task



A. Kapoor, ***et al.***

Nonlinear Optimization

- One problem with linearized least squares is the proliferation of constraints to approximate the real constraints
- Consequently, it is worth considering alternatives that can handle more general formulas "directly"

$$\Delta\vec{\mathbf{q}}_{des} = \arg\min_{\Delta\vec{\mathbf{q}}} C(\Delta\vec{\mathbf{x}}, \Delta\vec{\mathbf{q}}, \vec{\mathbf{s}})$$

subject to

$$\Delta\vec{\mathbf{x}} = \mathbf{J}\Delta\vec{\mathbf{q}}$$

$$\mathbf{A}(\Delta\vec{\mathbf{x}}, \Delta\vec{\mathbf{q}}, \vec{\mathbf{s}}) \le \vec{\mathbf{b}}$$

Using Non-Linear Constrained Optimization

- Use Sequential Quadratic Program* method
- SQP solves the following problem iteratively

$$\begin{array}{ll} \arg\min\limits_{d^k} \nabla C(\vec{x}(\vec{q} + \Delta\vec{q}^k), \vec{s}^k, \vec{x}^d)^t \vec{d} + \frac{1}{2}\vec{d}^t B_k \vec{d} & \\ s.t. \qquad \nabla A_{A_k}(\vec{x}_i(\vec{q} + \Delta\vec{q}^k), \vec{s}^k)^t \vec{d} \le \vec{b}_{A_k}, & \end{array}$$

- Start with a solution $[\Delta q^k, s^k]^t$
- Descent direction along with step size determine next solution $[\Delta q^{k+1}, s^{k+1}]^t$

*P. Spellucci, *Math. Prog.*, '98

A. Kapoor, *et al.*

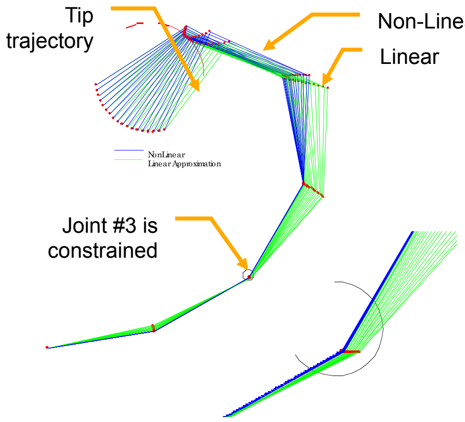
Remarks: Non-Linear Constraints

- Current incremental motion can be used as starting guess for next motion
- Worst case number of constraints n times m, n = # variables, m = # nonlinear constraints
- Analytical gradient increases speed

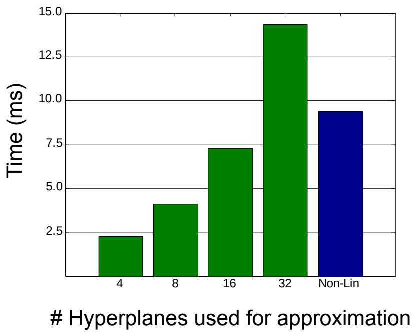
A. Kapoor, ***et al.***

Linear v. Non-Linear Constraints

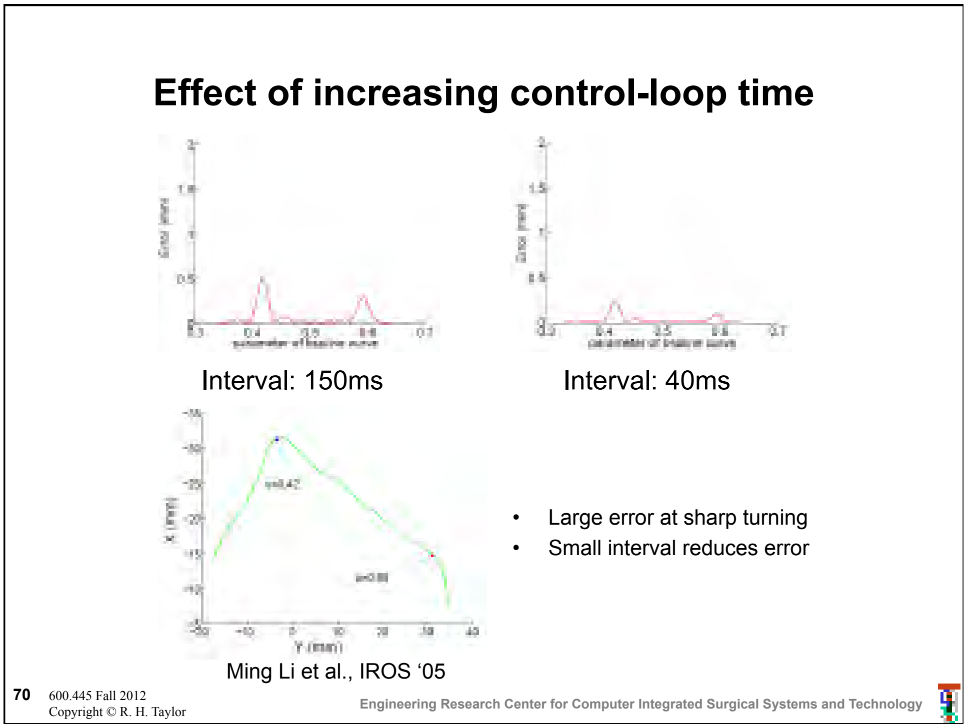
Accuracy



Time



A. Kapoor, ***et al.***



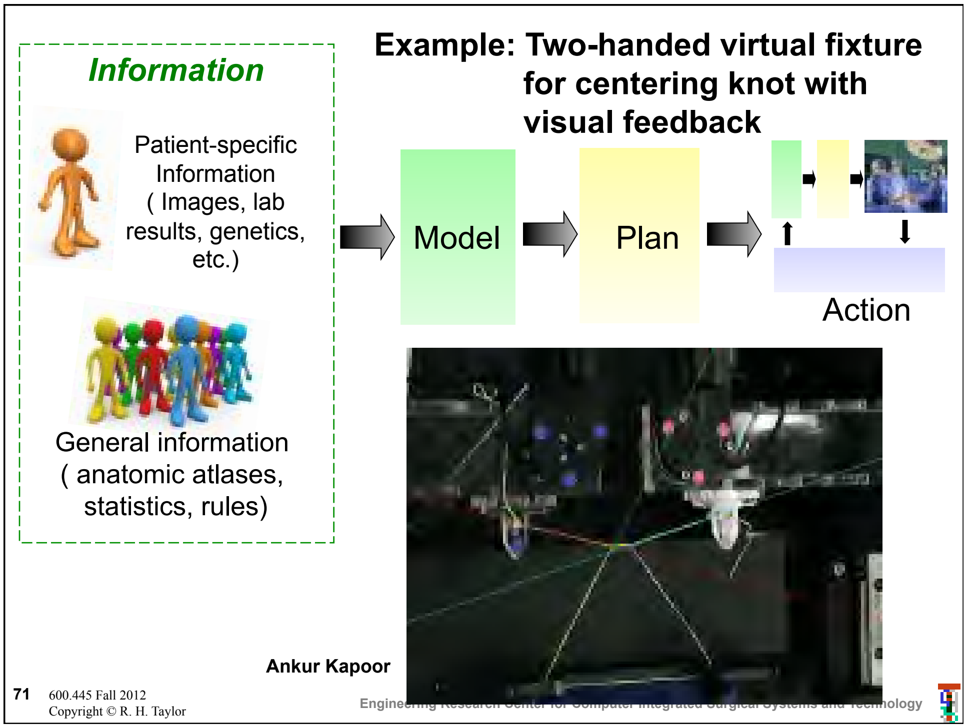
Effect of increasing control-loop time

Interval: 150ms

Interval: 40ms

- Large error at sharp turning
- Small interval reduces error

Ming Li et al., IROS '05



Example: Two-handed virtual fixture for centering knot with visual feedback

Information

Patient-specific Information (Images, lab results, genetics, etc.)

General information (anatomic atlases, statistics, rules)

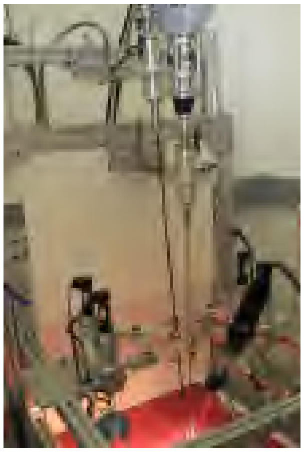
Model

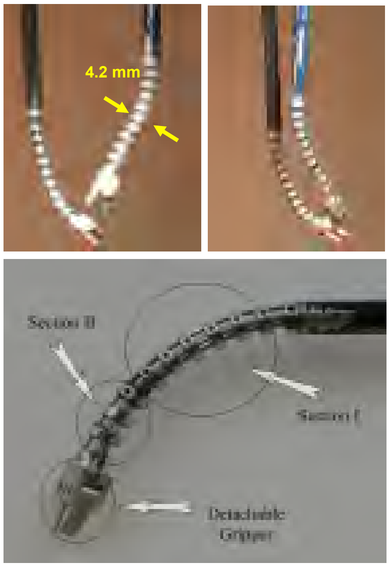
Plan

Action

Ankur Kapoor

Scalable Robot for Dexterous Surgery in Small Spaces (aka Snake Like Robot)

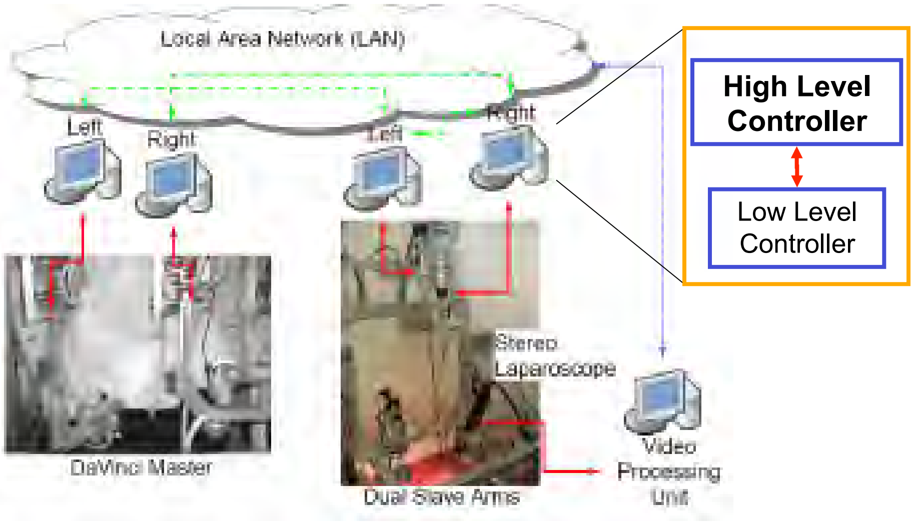




Team: A. Kapoor, Kai Xu, Wei Wei, N. Simaan, P. Kazanzides, R. H. Taylor
Collaborator: P. Flint, MD

Copyright © R. H. Taylor

Snake Like Robot
System Architecture



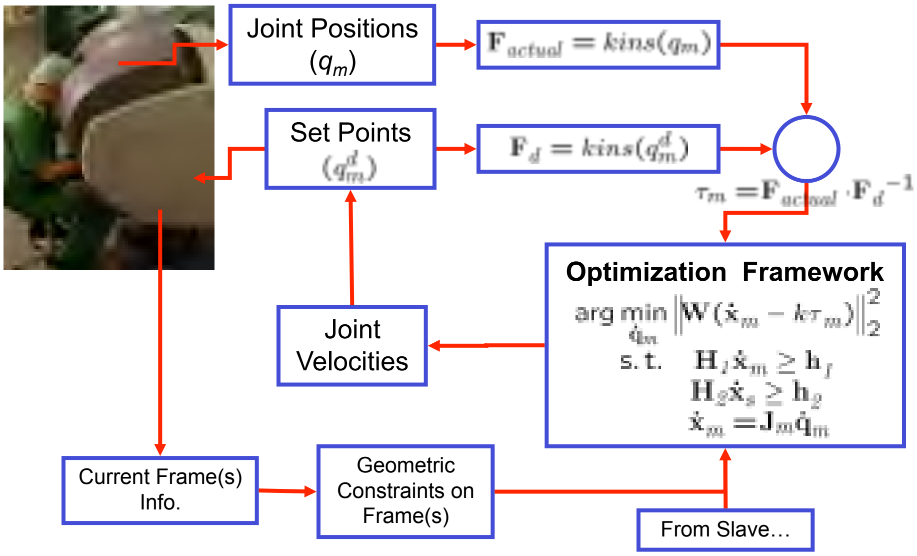
High Level Controller

Low Level Controller

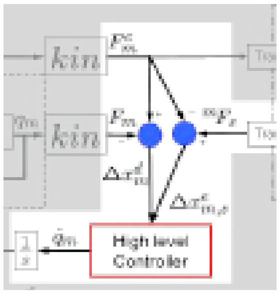
Copyright © R. H. Taylor

DaVinci Master Robot

High Level Constrained Control



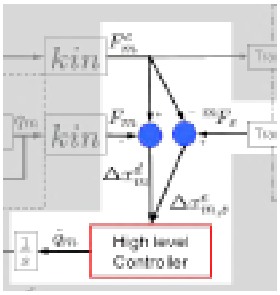
Master Side High-Level Controller



$$\min_{\Delta \vec{q}_m} \left\| W_{m,t}\left(\Delta x_m - K_t\left[p^e_m ; \theta_m \omega_m\right]\right)\right\|$$
$$+ \left\| W_{m,s}\left(\Delta x_m - K_f\left[p^e_{m,s} ; \theta_{m,s} \omega_{m,s}\right]\right)\right\|$$
$$+ \left\| W_{m,j} \Delta q_m \right\|$$

- **Objectives**:
  - Minimize error between desired motion and actual motion
  - Oppose motion that increases master-slave tracking error
  - Minimize the extraneous motion of the joints, and
  - Avoid large incremental joint motions that could occur near singularities

Master Side High-Level Controller



$$H_m \Delta q_m \ge h_m$$

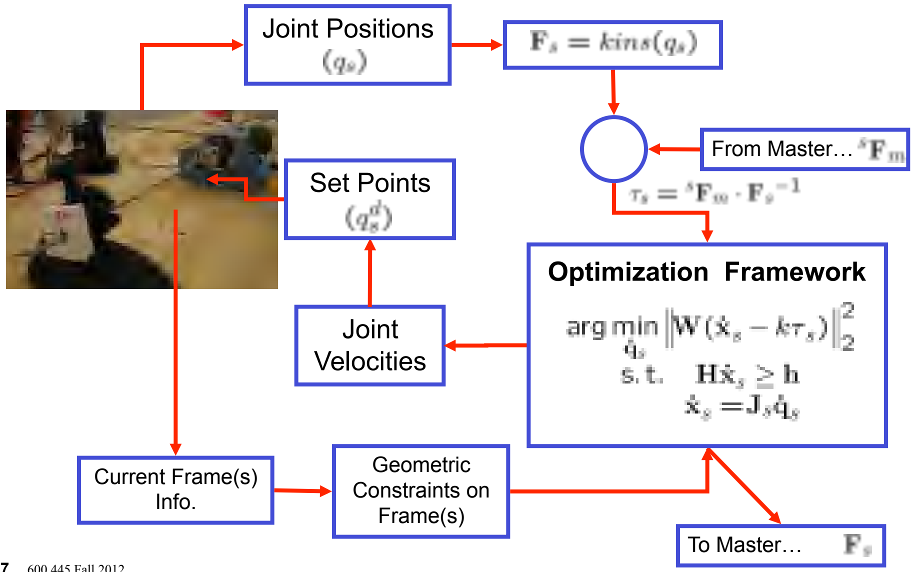
that is $\begin{bmatrix} I \\ -I \\ I \\ -I \end{bmatrix} \Delta q_m \ge \begin{bmatrix} q_{m,L} - q_m \\ q_m - q_{m,U} \\ \dot q_{m,U} \cdot \Delta t \\ \dot q_{m,U} \cdot \Delta t \end{bmatrix}$

- **Constraints**:
  - General form: $H_{m,j} + \Delta q_m \ge h_{m,j}$
  - Not allow motion outside joint range
  - Not allow motion that exceeds joint velocity limits
  - Additional constraints can be added from the VF Library

Copyright © R. H. Taylor

DaVinci Slave Robot

High Level Constrained Control



Joint Positions (q_s)

$\mathbf{F}_s = kins(q_s)$

From Master… ${}^s\mathbf{F}_m$

$\tau_s = {}^s\mathbf{F}_m \cdot \mathbf{F}_s{}^{-1}$

Set Points (q_s^d)

Optimization Framework

$$\arg\min_{\dot{\mathbf{q}}_s} \left\| \mathbf{W}(\dot{\mathbf{x}}_s - k\tau_s) \right\|_2^2$$
$$\text{s.t.} \quad \mathbf{H}\dot{\mathbf{x}}_s \ge \mathbf{h}$$
$$\dot{\mathbf{x}}_s = \mathbf{J}_s \dot{\mathbf{q}}_s$$

Joint Velocities

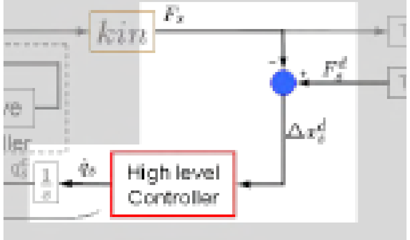
Current Frame(s) Info.

Geometric Constraints on Frame(s)

To Master… $\mathbf{F}_s$

Copyright © R. H. Taylor

Slave Side High-Level Controller

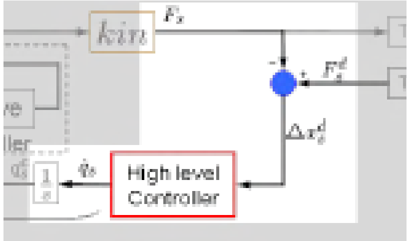


$$\min_{\Delta q_s} \left\| W_{s,t} \left(\Delta x_s - K_\alpha \left[p_s^d ; \theta_s \omega_s \right] \right) \right\| + \left\| W_{s,j}(q) \Delta q_s \right\| + \left\| W_{s,s} s \right\|$$

- **Objectives**:
  - Minimize error between desired motion and actual motion
  - Minimize the extraneous motion of the joints, and
  - Avoid large incremental joint motions that could occur near singularities

---

Slave Side High-Level Controller



such that $\begin{bmatrix} I \\ -I \\ I \\ -I \end{bmatrix} \Delta q_s \geq \begin{bmatrix} q_{s,L} - q_s \\ q_s - q_{s,U} \\ \dot{q}_{s,U} \cdot \Delta t \\ \dot{q}_{s,U} \cdot \Delta t \end{bmatrix}$

and $\|\vec{d}\| + \Delta x_b \cdot \hat{d} + \vec{v} \cdot \hat{d} + s \geq d_{safe}$

$0 \leq s \leq s_{lim}$

- **Constraints**:
  - Not allow motion outside joint range
  - Not allow motion that exceeds joint velocity limits
  - **Collision avoidance between slaves**
  - More constraints can be added from the VF Library

µForce Scaling Cooperative Control

Cooperative Control

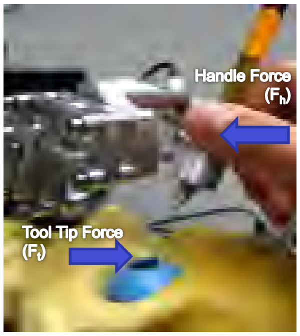
Velocity at the tool (V) is proportional to (α gain) the user's input force at the handle (F_h)

$$\dot{x} = \alpha F_h$$

µForce Scaling

Amplifies (ϒ gain) the human-imperceptible forces sensed at the tool tip (F_t) to handle interaction forces (F_h) by modulating robot velocity.

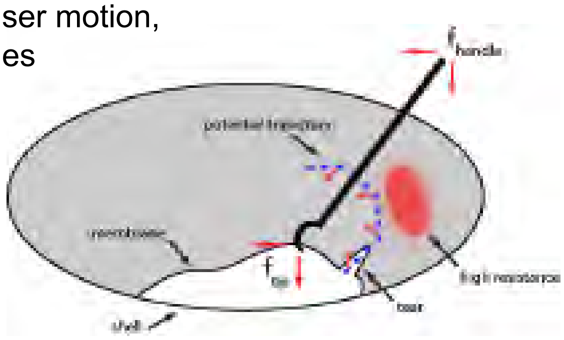
$$\dot{x} = \alpha\left(F_h - \gamma F_t\right), \quad \text{e.g., } \gamma = 500$$



Kumar et al (ICRA' 00) ; Balicki et al. (MICCAI'10); Uneri et al., BioRob 2010

µForce Guided Cooperative Control

- ¬ User fights against ever increasing resistance
- ✓ Ensure safety tip force limits
- ¬ User interaction is limited at high-resistance regions
- ✓ Try to avoid those regions for later peeling
- ¬ User gets "stuck", gives up, tries re-approach
- ✓ Ensure continuous user motion, even at the boundaries



Uneri *et al.*, BioRob 2010

µForce Guided Cooperative Control

- Global Limiting
  - Task-specific tip force limit
  - User controlled limit distribution
- Continuous motion at the constraint boundaries
- Virtual spring construct to ensure stability

- $$f_{lim} = f_{max}\frac{|f_h|}{\|f_h\|}$$

$$\dot{x}_{lim} = \dot{x}\left(\frac{f_{lim} - |f_t|}{f_{spring}}\right)$$

Local Force Minimization

•Guiding user towards direction of minimum resistance

•Sensitivity variable allows user override

•Haptically intuitive response

•Avoids / postpones reaching limits

$$\dot{x}_{min} = k_p\left(1 - s\frac{|f_t|}{\|f_t\|}\right)f_h$$

Uneri *et al.*, BioRob 2010

Experimental Platform

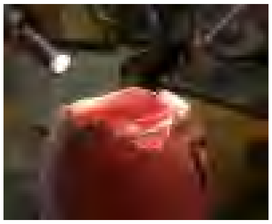
Focusing on:

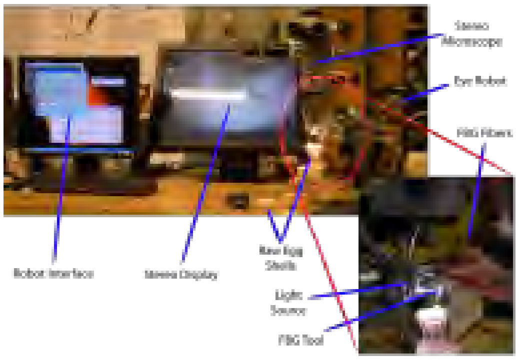
•Properties of the tissue we interact with

•The method of interaction, i.e. performance of our algorithms

Performed on:

- **Inner shell membrane of raw eggs**
- **Surrogate tissue for epiretinal membrane peeling**

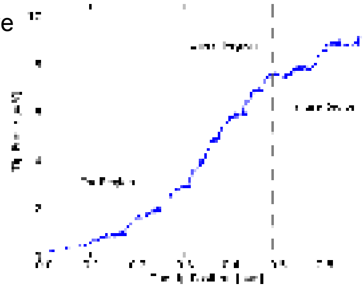




Uneri *et al.*, BioRob 2010

Experiment:
Tissue Force Characterization

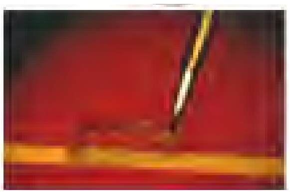
- A corrected position allows us to observe tissue strain
- Controlled constant force application
  - Incremented by 1mN, with 10s delay, over a range of 1-10mN
- Characteristic curve obtained reveals a similar pattern to those seen in fibrous tissue tearing
  - Toe region: Safe
  - Linear region: Predictive
  - Failure region: Peeling

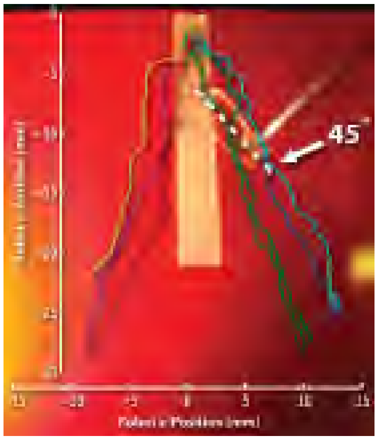


Uneri *et al.*, BioRob 2010

Experiment:
µForce Guided Cooperative Control

- Task: delaminate PVC strip with acrylic adhesive from a wax surface.
- Strip is peeled at an average of 45°
- User was guided away from the centerline in the direction of lowest resistance

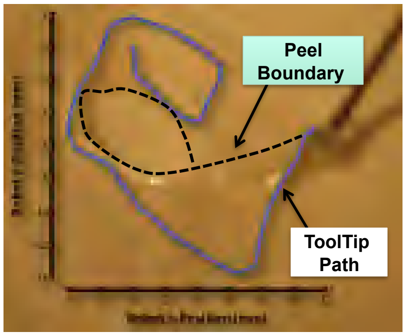




Uneri *et al.*, BioRob 2010

Experiment:
μForce Guided Cooperative Control

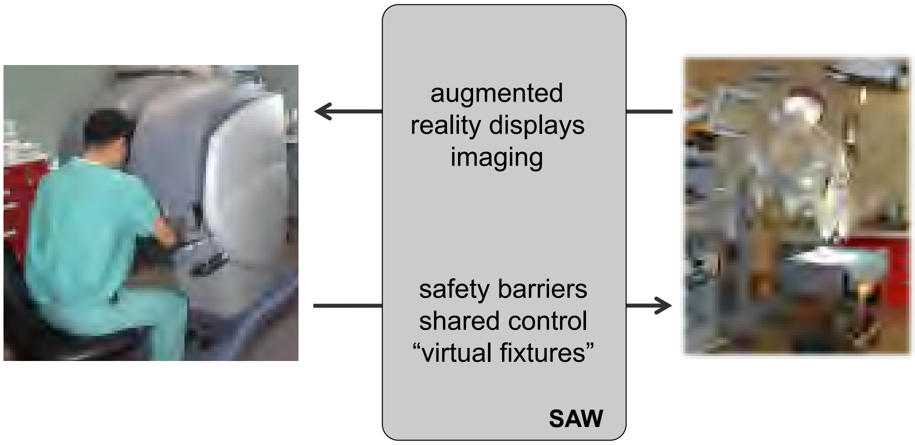
- Goal: Remove a section of egg inner shell membrane
- Circular trajectory consistent with the results from the strip peeling experiment
- Magnify the perception of tip forces lateral to direction of desired motion
- Results in a peel pattern seen Capsulorhexis maneuver



Peeling Inner Egg Shell Membrane

Uneri *et al.*, BioRob 2010

Information-enhanced robotic surgery

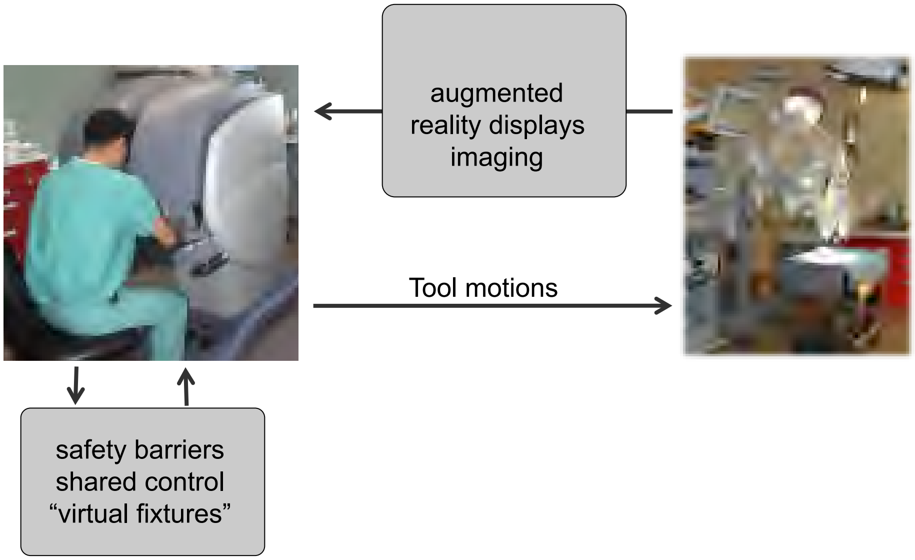


augmented reality displays imaging

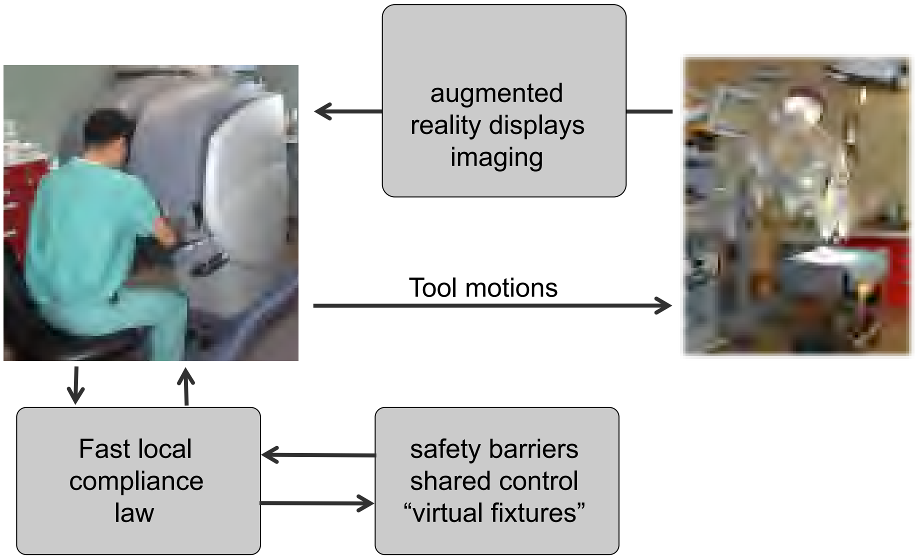
safety barriers shared control “virtual fixtures”

SAW

Information-enhanced robotic surgery



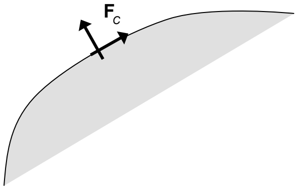
Information-enhanced robotic surgery



Virtual Fixture “Hook” in DaVinci API

- **Experimental interface not in any clinical or commercial product.**
- Specification developed jointly by JHU and Intuitive to support research
- Prototyped at JHU by Tian Xia and Russ Taylor
- Current version implemented in DaVinci “S” model by Lawton Verner at ISI, with “hooks” in a proprietary ISI Application Program Interface
- Accessed through cisst/SAW libraries

Compliance virtual fixtures



$\mathbf{F} = \left[\mathbf{R}, \vec{\mathbf{p}}\right] =$ current pose; $\dot{\mathbf{p}} =$ current velocity

$\mathbf{F}_c = [\mathbf{R}_c, \vec{\mathbf{p}}_c] =$ position compliance frame

$\vec{\mathbf{k}}^{(+)}, \vec{\mathbf{k}}^{(-)} =$ position stiffness factors

$\vec{\mathbf{b}}^{(+)}, \vec{\mathbf{b}}^{(-)} =$ damping factors

$\vec{\mathbf{g}}^{(+)}, \vec{\mathbf{g}}^{(-)} =$ force bias terms

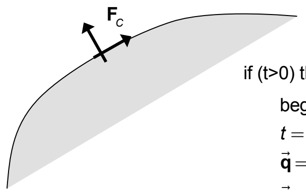
$\mathbf{R}_o =$ orientation compliance frame

$\vec{\mathbf{k}}_o^{(+)}, \vec{\mathbf{k}}_o^{(-)} =$ orientation stiffness factors

$\vec{\tau}^{(+)}, \vec{\tau}^{(-)} =$ torque bias terms

$t =$ time remaining on timeout counter

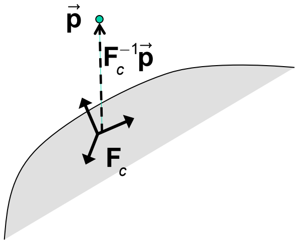
Compliance virtual fixtures



```
if (t>0) then
    begin
    $t = t - 1$
    $\vec{\mathbf{q}} = \mathbf{F}_C^{-1}\vec{\mathbf{p}} = \mathbf{R}_C^{-1}(\vec{\mathbf{p}} - \vec{\mathbf{p}}_c)$
    $\vec{\mathbf{v}} = \mathbf{R}_C^{-1}\dot{\vec{\mathbf{p}}}$
    $\vec{\mathbf{h}} = \vec{\mathbf{0}}; \vec{\psi} = \vec{\mathbf{0}}$
    for $i \in \{x, y, z\}$ do
        $\{$if $\vec{\mathbf{q}}_i \le 0$ then $\vec{\mathbf{h}}_i = \vec{\mathbf{g}}_i^{(-)} + \vec{\mathbf{k}}_i^{(-)}\vec{\mathbf{q}}_i + \vec{\mathbf{b}}_i^{(-)}\vec{\mathbf{v}}_i$ else $\vec{\mathbf{h}}_i = \vec{\mathbf{g}}_i^{(+)} + \vec{\mathbf{k}}_i^{(+)}\vec{\mathbf{q}}_i + \vec{\mathbf{b}}_i^{(+)}\vec{\mathbf{v}}_i\}$;
    $\vec{\mathbf{f}} = \mathbf{R}_C\vec{\mathbf{h}}$; add $\vec{\mathbf{f}}$ to the forces exerted on the master

    $\vec{\theta} =$ Rodrigues vector corresponding to $\Delta\mathbf{R} = \mathbf{R}_o^{-1}\mathbf{R}$
    for $i \in \{x, y, z\}$ do
        $\{$if $\vec{\theta}_i \le 0$ then $\vec{\psi}_i = \vec{\tau}_i^{(-)} + \vec{\mathbf{k}}_i^{(-)}\vec{\theta}_i$ else $\vec{\psi}_i = \vec{\tau}_i^{(+)} + \vec{\mathbf{k}}_i^{(+)}\vec{\theta}_i\}$;
    add $\mathbf{R}_o\vec{\psi}$ to the torques exerted on the master
    end
```

Surface following virtual fixture



Goal: Stay on a surface; bias force drawing toward the surface; spring force resisting penetration

$\vec{\mathbf{p}}_c =$ closest point on surface

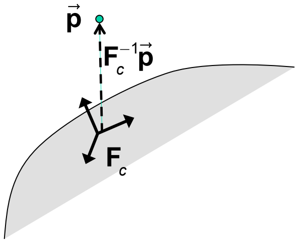
$\mathbf{R}_c\vec{\mathbf{z}} =$ surface normal at $\vec{\mathbf{p}}_c$

$\vec{\mathbf{k}}^{(-)} = [0, 0, -\mathit{stiffness}]$

$\vec{\mathbf{g}}^{(+)} = [0, 0, -\mathit{bias}]$

$\mathit{Others} = 0$

Curve following virtual fixture



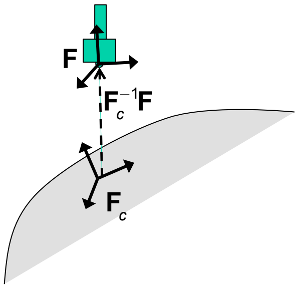
Goal: Stay on a surface; bias force drawing toward the surface; spring force resisting penetration; follow curve on surface

$\vec{\mathbf{p}}_c =$ closest point on line on surface
$\mathbf{R}_c\vec{\mathbf{z}} =$ surface normal at $\vec{\mathbf{p}}_c$
$\mathbf{R}_c\vec{\mathbf{x}} =$ line tangent at $\vec{\mathbf{p}}_c$

$\vec{\mathbf{k}}^{(-)} = [0, -\textit{follow stiffness}, -\textit{penetration stiffness}]$
$\vec{\mathbf{k}}^{(+)} = [0, -\textit{follow stiffness}, -\textit{follow stiffness}]$
$\vec{\mathbf{g}}^{(+)} = [0, 0, -\textit{bias}]$ (**Note**: may just set to 0)
$\textit{Others} = 0$

Copyright © R. H. Taylor

Surface following virtual fixture



Goal: Stay on a surface; bias force drawing toward the surface; spring force resisting penetration; torque to align to surface normal

$\vec{\mathbf{p}}_c =$ closest point on surface
$\mathbf{R}_c\vec{\mathbf{z}} =$ surface normal at $\vec{\mathbf{p}}_c$

$\vec{\mathbf{k}}^{(-)} = [0, 0, -\textit{stiffness}]$
$\vec{\mathbf{g}}^{(+)} = [0, 0, -\textit{bias}]$
$\vec{\mathbf{k}}_o^{(+)} = \vec{\mathbf{k}}_o^{(-)} = [-\textit{orient stiffness}, -\textit{orient stiffness}, 0]$
$\textit{Others} = 0$

Copyright © R. H. Taylor

Limitation and Extensions

- The specific abstraction just presented has some limitations. In particular, it separates the position and orientation compliance in a way that makes coupling of orientations and translations non-trivial.
- This can be gotten around to some extent by continually updating the virtual fixture compliance parameters.
- There are several obvious extensions that may be tried. For example, one can provide fuller matrices for virtual fixture force/torque generation. E.g.:

Compute $\vec{\mathbf{q}},\vec{\mathbf{v}},\vec{\theta},\vec{\phi}$ from $\mathbf{F}_c$ and $\mathbf{R}_o$, *where* $(\vec{\phi} = d\vec{\theta} / dt)$

Compute a region *i* of local configuration space from $\vec{\mathbf{q}}$ and $\vec{\theta}$

$$\begin{bmatrix} \vec{\mathbf{h}} \\ \vec{\phi} \end{bmatrix} = \mathbf{K}_i \bullet \begin{bmatrix} \vec{\mathbf{q}} \\ \vec{\theta} \end{bmatrix} + \mathbf{B}_i \bullet \begin{bmatrix} \vec{\mathbf{v}} \\ \vec{\phi} \end{bmatrix} + \begin{bmatrix} \vec{\mathbf{g}}_i \\ \vec{\tau}_i \end{bmatrix}$$