Medical Robots,
Constrained Robot Motion Control,
and “Virtual Fixtures”

Russell H. Taylor
601.455/655

Disclosures & Acknowledgments

• This is the work of many people

• Some of the work reported in this presentation was supported by fellowship grants from Intuitive Surgical and Philips Research North America to Johns Hopkins graduate students and by equipment loans from Intuitive Surgical, Think Surgical, Philips, Kuka, and Carl Zeiss Meditec.

• Some of the work reported in this talk incorporates intellectual property that is owned by Johns Hopkins University and that has been or may be licensed to outside entities, including Intuitive Surgical, Varian Medical Systems, Philips Nuclear Medicine, Galen Robotics and other corporate entities. Prof. Taylor has received or may receive some portion of the license fees. Also, Dr. Taylor is a paid consultant to and owns equity in Galen Robotics, Inc. These arrangements have been reviewed and approved by JHU in accordance with its conflict of interest policy.

• Much of this work has been funded by Government research grants, including NSF grants EEC9731478 and IIS0999770 and NIH grants R01-EB016703, R01-EB007969, R01-CA127144, R42-RR019159, and R21-EB0045457; by Industry Research Contracts, including from Think Surgical; by gifts to Johns Hopkins University from John C. Malone, Richard Swinow and Paul Maritz; and by Johns Hopkins University internal funds.
Goal: Human-machine partnership to fundamentally improve interventional medicine

Physicians

Technology

Information
Complementary Capabilities

Humans

• Excellent judgment & reasoning
• Excellent optical vision
• Cannot see through tissue
• Do not tolerate ionizing radiation
• Limited precision, hand tremor
• No stereotactic accuracy
• Moderately strong
• High dexterity ("human" scale)
• Big hands and bodies
• Reasonable force sensitivity
• Must rely on memory of preoperative plans and data

Robots

• No judgment
• Limited vision processing
• Can use x-rays, other sensors
• Do not mind radiation
• High precision
• High stereotactic accuracy
• Variable strength
• Dexterity at different scales
• Variable sizes
• Can sense very small forces
• Can be programmed to use preoperative plans and data

Common classes of medical robots

• Surgical “CAD/CAM” systems
  – Goal is accurate execution of surgical plans
  – Typically based on medical images
  – Planning may be “online” or “offline”
  – Execution is often at least semi-autonomous but may still involve interaction with humans
  – Examples: Orthopaedic robots, needle placement robots, radiation therapy robots

• Surgical “assistant” systems
  – Emphasis is on interactive control by human
  – Human input may be through hand controllers (e.g., da Vinci), hand-over-hand (e.g., Mako, JHU "steady hand" robots)
  – Typically augmenting or supplementing human ability
  – Common applications include MIS, microsurgery

• Note that the distinction is really somewhat arbitrary
  – Most real systems have aspects of both.
**Surgical CAD/CAM: Orthopaedic Robots**

Robodoc

Blue Belt Technologies

D. Glozman & M. Shoham

ACROBOT surgical robot

Mako Robotics Rio
http://www.makosurgical.com/

**Image-guided needle placement**

Masamune, Fichtinger, Iordachita, ...

Okamura, Webster, ...

Krieger, Fichtinger, Whitcomb, ...

Fichtinger, Kazanzides, Burdette, Song ...

Iordachita, Fischer, Hata ...

Taylor, Masamune, Susil, Patriciu, Stoianovici,...
Image Guided Radiotherapy

- Radiation source mounted on robotic arm
- Automatic segmentation of targets
- Automated planning radiation beam path
- Image guide patient motion compensation for more accurate radiation targeting

Common classes of medical robots

- Surgical “CAD/CAM” systems
  - Goal is accurate execution of surgical plans
  - Typically based on medical images
  - Planning may be “online” or “offline”
  - Execution is often at least semi-autonomous but may still involve interaction with humans
  - Examples: Orthopaedic robots, needle placement robots, radiation therapy robots

- Surgical “assistant” systems
  - Emphasis is on interactive control by human
  - Human input may be through hand controllers (e.g., da Vinci), hand-over-hand (e.g., Mako, JHU "steady hand" robots), mouse, or other
  - Typically augmenting or supplementing human ability
  - Common applications include MIS, microsurgery

- Note that the distinction is really somewhat arbitrary
  - Most real systems have aspects of both
MIS Landscape

Opportunity For Robotic Impact

Superficial
- Laparoscopy
- Arthroscopy
- Pericardioscopy
- Thoracoscopy

Intraluminal Pathways
- Bronchoscopy
- Proctoscopy
- Gastroscopy
- ERCP

Intracavity Space
- Colposcopy
- Cystoscopy
- Laryngoscopy
- Tracheoscopy

Deep in Anatomy

Slide credit: Howie Choset + RHT

Precision Augmentation
Common classes of medical robots

- Surgical “CAD/CAM” systems
  - Goal is accurate execution of surgical plans
  - Typically based on medical images
  - Planning may be “online” or “offline”
  - Execution is often at least semi-autonomous but may still involve interaction with humans
  - Examples: Orthopaedic robots, needle placement robots, radiation therapy robots

- Surgical “assistant” systems
  - Emphasis is on interactive control by human
  - Human input may be through hand controllers (e.g., da Vinci), hand-over-hand (e.g., Mako, JHU “steady hand” robots)
  - Typically augmenting or supplementing human ability
  - Common applications include MIS, microsurgery

- Note that the distinction is really somewhat arbitrary
  - Most real systems have aspects of both.

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
- Surgeon input
- Plan information
- Anatomic models
- Safety constraints

Motion level control
- Desired motion description
- Task-level constraints on how motion is done
- Sensor and state information

Low level control
- Joint position or velocity commands
- Joint positions, velocities, & other state information

Robot kinematics & motion limits
- Sensor values

Motion level control
\[ \dot{q}_{\text{des}} = K_{\text{ins}}^{-1}(\Delta F_{\text{des}} F_{\text{cur}}) \]

Position control
- Motor currents

One implementation

Sensor values

Robot kinematics & motion limits
- \( K_{\text{ins}}(\cdots), J_{\text{ins}}(\cdots) \)
- \( \dot{q}_L \leq \dot{q} \leq \dot{q}_U \)
Background: Jacobean Robot Motion Control

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

$$\text{Joint positions } \dot{q} \rightarrow \text{Pose } F = \text{Kins}(\dot{q})$$

$$\text{Pose } F(\dot{q} + \Delta \dot{q}) = \text{Kins}(\dot{q} + \Delta \dot{q})$$

$$\Delta F \cdot F = \text{Kins}(\dot{q} + \Delta \dot{q})$$

$$\Delta F = \text{Kins}(\dot{q} + \Delta \dot{q}) \text{Kins}(\dot{q})^{-1}$$

Background: Jacobean Robot Motion Control

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

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

$$\Delta F = \text{Kins}(\dot{q} + \Delta \dot{q}) \text{Kins}(\dot{q})^{-1}$$

which we typically linearize as

$$\Delta \dot{x} = \begin{bmatrix} \vec{\alpha} \\ \vec{\varepsilon} \end{bmatrix} = J_{\text{Kins}}(\dot{q}) \Delta \dot{q}$$

Note that here we are computing $\Delta F$ in the base frame of the robot. If we want to compute $\Delta F$ in the end effector frame, so that $F \cdot \Delta F = \text{Kins}(\dot{q} + \Delta \dot{q})$, then we will get a slightly different expression for $J_{\text{Kins}}(\dot{q})$, though the flavor will be the same.
Background: Jacobean Robot Motion Control

\[ \text{Pose } F = \text{kins}(q) \]

\[ \text{Pose } F(q + \Delta q) = \text{kins}(q + \Delta q) \]

\[ \Delta F = \text{kins}(q + \Delta q) \]

\[ \Delta F = \text{kins}(q + \Delta q) \text{kins}(q)^{-1} \]

\[ \begin{bmatrix} \alpha \\ \varepsilon \end{bmatrix} \approx J(q) \Delta q \]

\[ \Delta q \approx J(q)^{-1} \begin{bmatrix} \alpha \\ \varepsilon \end{bmatrix} \]

---

Jacobian motion control implementation

Task level control

\[ \Delta x_{\text{des}} \]

\[ F, \dot{q} \]

Sensor values

Motion level control

\[ \ddot{q}_{\text{des}} = \dot{q} + J^{-1}(q) \Delta x_{\text{des}} \]

Position control

\[ \ddot{q}, \dot{q} \]

Sensor values

Robot kinematics & motion limits

\[ K\text{ins}(\cdot), J\text{kins}(\cdot) \]

\[ \dot{q} \leq \ddot{q} \leq \dddot{q} \]

Surgeon input

Plan information

Anatomic models

Safety constraints
Steady Hand Robot
Hands on compliance control

\[ \dot{x}_{des} = K_v \vec{f}_h \]
\[ q_{cmd} = J^{-1}_{kns} \dot{x}_{des} \]


Steady Hand Robot
Hands on compliance control with force scaling

\[ \dot{x}_{des} = K_v (\vec{f}_h - \gamma \vec{f}_{tip}) \]
\[ q_{cmd} = J^{-1}_{kns} \dot{x}_{des} \]


Steady Hand Robot (Alternative Formulation)
Hands on compliance control with force scaling

\[ \dot{x}_{\text{des}} = K_v (\hat{f}_h - \gamma \hat{f}_t) \]
\[ \dot{q}_{\text{cmd}} = \arg\min_{q_{\text{cmd}}} \| \dot{x}_{\text{des}} - J_{\text{cmd}} \dot{q}_{\text{cmd}} \| \]

Example: Fenestration of Stapes Footplate

Example: Fenestration 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.,

- Discussion that follows draws upon work at IBM Research and within the CISST ERC at JHU. E.g.,
  - 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

Video tracking

Clip-on joystick

RCM

View direction

LARS Video
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 Jacobian formulation to transform to joint space
- Solve for joint motion

Example: keep tool tip near a point

\[
\bar{D}(\Delta \dot{x}) = \Delta F(\Delta \dot{q}) \bullet F \bullet \Delta p_{sp} - \Delta \dot{p}_{goal} \\
= \bar{\alpha} \times \bar{\epsilon} + \bar{\epsilon} + \Delta \dot{p}_{goal} \quad \text{where} \quad \bar{\epsilon} = F \bullet \Delta p_{sp} \\
\bar{\alpha} = J_{\alpha}(\Delta \dot{q}) \\
\bar{\epsilon} = J_{\epsilon}(\Delta \dot{q})
\]

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

\[
\Delta q_{des} = \arg\min_{\Delta q} \|\bar{D}(\Delta \dot{x})\|^2 = \|\bar{\alpha} \times \bar{\epsilon} + \bar{\epsilon} + \Delta \dot{p}_{goal}\|^2
\]

Subject to

\[
\bar{\alpha} = J_{\alpha}(\Delta \dot{q}) \\
\bar{\epsilon} = J_{\epsilon}(\Delta \dot{q}) \\
\|\bar{\alpha} \times \bar{\epsilon} + \bar{\epsilon} + \Delta \dot{p}_{goal}\| \leq 3 \\
\bar{q}_L - \Delta \dot{q} \leq \Delta \dot{q} \leq \bar{q}_U - \Delta \dot{q}
\]
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

\[ \Delta q_{\text{des}} = \arg\min_{\Delta q} \sum \|D(\Delta \bar{x})\|^2 + \eta \|\alpha \times \bar{r}\|^2 \]

Subject to

\[ \bar{x} = F \cdot \bar{p}_{\text{tip}} \]
\[ D(\Delta \bar{x}) = \alpha \times \bar{r} + \bar{\epsilon} + \bar{x} - \bar{p}_{\text{goal}} \]
\[ \bar{\alpha} = J_q(\bar{q}) \Delta \bar{q} ; \quad \bar{\epsilon} = J_{\bar{q}}(\bar{q}) \Delta \bar{q} \]
\[ \|D(\Delta \bar{x})\| \leq 3 \]
\[ q_L - \bar{q} \leq \Delta \bar{q} \leq q_U - \bar{q} \]

Note weighting factors

---

Solving the optimization problem

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

\[ \Delta q_{\text{des}} = \arg\min_{\Delta q} \|E_{\text{task}} \Delta \bar{x} - \bar{f}_{\text{task}}\|^2 + \|E_q \Delta \bar{x} - \bar{f}_q\|^2 \]

subject to

\[ \Delta \bar{x} = J \Delta \bar{q}; \quad A_{\text{task}} \Delta \bar{x} \leq \bar{b}_{\text{task}} ; \quad A_q \Delta \bar{q} \leq \bar{b}_q \]

- 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)

- **Can also minimize other objective functions**
  - E.g., minimize an L1 norm (linear programming problem)
Linear least squares implementation

Task level control

Motion level control

Position control

Sensor values

Surgeon input
Plan information
Anatomic models
Safety constraints

Robot kinematics
& motion limits
\( Kins(\cdot), J_{\text{kin}}(\cdot) \)
\( \dot{q}_i \leq \ddot{q} \leq \ddot{q}_i, E_{\dddot{q}} f_{\dddot{q}} \)

Motion level control

\( \Delta q_{\text{task}} = \arg \min_{\Delta q} \| E_{\dddot{q}} f_{\dddot{q}} - \| \Delta x, \Delta \dot{q} \| \|

subject to

\( \Delta \dot{x} = J_{\text{kin}}(\cdot) \Delta \dot{q} \leq b \)

Some IBM Movies

Early Constrained Motion System (LapSYS)

Vision-guided targeting
**Steady Hand Robot**

**High Level Constrained Control**

- **Handle Force**
- **Joint Velocities**
- **Current Frame(s) Info.**
- **Geometric Constraints on Frame(s)**
- **Reference Direction**

**Optimization Framework**

\[
\begin{aligned}
\text{arg min}_{\dot{q}} \| W (\ddot{x} - \dot{x}_d) \|_2^2 \\
s. t. \quad H \ddot{x} \geq h \\
\dot{x} = J \dot{q}
\end{aligned}
\]

**Example: Hands-on Guiding with Forbidden Half Space**

\[
\Delta \ddot{q} = \arg \min_{\Delta \ddot{q}} \| K_v \ddot{f} - J_{rhs}(\ddot{q}) \Delta \ddot{q} \|
\]

Such that

\[
d \leq \hat{n} \cdot (F(\ddot{q}) \Delta F(\ddot{q}, \Delta \ddot{q}) \cdot \hat{p}_{sp})
\]

Note here we are using the right hand side Jacobean, since the force sensor is associated with the tool attachment point, and it is more natural for the motions to comply to pushes on the tool handle.
Example: Hands-on Guiding with Forbidden Half Space

\[ \Delta \mathbf{F}(\mathbf{q}, \Delta \mathbf{q}) \cdot \mathbf{F}(\mathbf{q}) \]

\[ \Delta \mathbf{q} = \arg \min_{\Delta \mathbf{q}} \| K \mathbf{f} - J_{ms}(\mathbf{q}) \Delta \mathbf{q} \| \]

Such that

\[ \begin{bmatrix} \alpha \\ \varepsilon \end{bmatrix} = J_{ms}(\mathbf{q}) \Delta \mathbf{q} \]

\[ d \leq \mathbf{n} \cdot (\mathbf{F}(\mathbf{q}) \cdot (\alpha \times \mathbf{p}_{sp} + \varepsilon + \mathbf{p}_{sp} + \mathbf{p}_{kin})) \]

i.e.,

\[ \begin{bmatrix} \alpha \\ \varepsilon \end{bmatrix} = J_{ms}(\mathbf{q}) \Delta \mathbf{q} \]

\[ d \leq \mathbf{n} \cdot (\mathbf{R}(\mathbf{q}) \cdot (\alpha \times \mathbf{p}_{sp} + \varepsilon + \mathbf{p}_{sp} + \mathbf{p}_{kin})) \]
Example: Hands-on Guiding with Forbidden Half Space

\[ \Delta \mathbf{q} = \arg \min_{\Delta \mathbf{q}} \| \mathbf{K} \mathbf{v} - J_{\text{kines}}(\mathbf{q}) \Delta \mathbf{q} \| \]

Such that

\[ d \leq \mathbf{n} \cdot (\Delta F(\mathbf{q}, \Delta \mathbf{q}) \cdot F(\mathbf{q}) \cdot \mathbf{p}_{sp}) \]

If we use the LHS Jacobean, we get something similar. Note however that in this case the gain matrix will likely be pose dependent, since the it is more natural for the surgeon’s hand to follow the tool. So it is useful to be able to make the conversion ...

LHS versus RHS Jacobians

\[ \Delta F(J_{\text{kines}}(\mathbf{q}) \Delta \mathbf{q}) \cdot F(\mathbf{q}) = F(\mathbf{q}) \cdot \Delta F(J_{\text{ma}}(\mathbf{q}) \Delta \mathbf{q}) \]

\[ \Delta R(J_{\text{kines}}(\mathbf{q}) \Delta \mathbf{q}) \cdot R(\mathbf{q}) = R(\mathbf{q}) \cdot \Delta R(J_{\text{ma}}(\mathbf{q}) \Delta \mathbf{q}) \]

Define

\[ J_{\text{mA}} = \begin{bmatrix} J_{\text{mA}}^{\Delta \mathbf{q}} \\ J_{\text{mA}1} \end{bmatrix} \quad \bar{c}_{\text{mA}} = J_{\text{mA}}^{\Delta \mathbf{q}} \Delta \mathbf{q} \quad J_{\text{ma}} = \begin{bmatrix} J_{\text{ma}}^{\Delta \mathbf{q}} \\ J_{\text{ma}1} \end{bmatrix} \quad \bar{c}_{\text{ma}} = J_{\text{ma}}^{\Delta \mathbf{q}} \Delta \mathbf{q} \]

SO

\[ \Delta R(J_{\text{ma}}(\mathbf{q}) \Delta \mathbf{q}) = R(\mathbf{q})^{-1} \Delta R(J_{\text{mA}}(\Delta \mathbf{q}) \cdot R(\mathbf{q}) \]

\[ I + sk(\alpha_{\text{mA}}) = I + R^{-1} sk(\alpha_{\text{mA}}) R \]

\[ sk(\bar{c}_{\text{ma}}) = sk(R^{-1} \bar{c}_{\text{mA}}) \]

\[ J_{\text{ma}} = R^{-1} J_{\text{mA}} \]

and one can do something similar for the \( \Delta \mathbf{p} \) parts (exercise).
Typical application domain: endoscopic sinus surgery


Sample task: steady hand path tracing

M. Li et al.
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

---

**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

Registered model → Constraint generation → State

Path → Tool tip guidance virtual fixture

Robot interface

\[ \min \| W \cdot (J_{tip} \cdot \Delta q - \Delta P_{dev}) \|^2 \]
Subject to
\[ G \cdot \Delta q \geq g \]
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.
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.
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
- NOTE: Generally, you only need to consider surfaces that are close enough so that the tool may move there in one time step.
Control Implementation

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

\[ \zeta = \min_{\Delta q} \left\| W_{\text{tip}} W_k W_{\text{joint.s}} \left( \begin{bmatrix} J_{\text{tip}}(q) \\ J_k(q) \end{bmatrix} \Delta q - \begin{bmatrix} \Delta P_{\text{tip-dec}} \\ 0 \\ 0 \end{bmatrix} \right) \right\| \]

subject to

\[ \begin{bmatrix} H_{\text{tip}} \\ H_k \\ H_{\text{joint.s}} \end{bmatrix} \cdot \begin{bmatrix} J_{\text{tip}}(q) \\ J_k(q) \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta q \end{bmatrix} \begin{bmatrix} h_{\text{tip}} \\ h_k \\ h_{\text{joint.s}} \end{bmatrix} \geq \begin{bmatrix} \Delta P_{\text{tip-dec}} \\ 0 \\ 0 \end{bmatrix} \]

Control Implementation

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

\[ \begin{bmatrix} \Delta P_{\text{tip}} - \Delta P_{\text{tip-dec}} \end{bmatrix} \quad \begin{bmatrix} \Delta P_{\text{tip-dec}} \end{bmatrix} \geq \begin{bmatrix} THD \end{bmatrix} \]

\[ \begin{bmatrix} W_{\text{tip}} \end{bmatrix} \begin{bmatrix} J_{\text{tip}}(q) \end{bmatrix} \begin{bmatrix} \Delta q \end{bmatrix} \begin{bmatrix} \Delta P_{\text{tip-dec}} \end{bmatrix} \]

\[ \min \zeta_{\text{tip}} = \begin{bmatrix} W_{\text{tip}} \end{bmatrix} \begin{bmatrix} J_{\text{tip}}(q) \end{bmatrix} \begin{bmatrix} \Delta q \end{bmatrix} \begin{bmatrix} \Delta P_{\text{tip-dec}} \end{bmatrix} \]

subject to \( H_{\text{tip-dec}} J_{\text{tip}}(q) \Delta q \geq h_{\text{tip}} \)

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

\[ \begin{bmatrix} W_k \cdot \Delta P_k \end{bmatrix} \quad \begin{bmatrix} n_k \cdot (P_k + \Delta P_k - P_k) \end{bmatrix} \geq d \]

\[ \min \zeta_k = \begin{bmatrix} W_k \end{bmatrix} \begin{bmatrix} J_k(q) \end{bmatrix} \begin{bmatrix} \Delta q \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} \]

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

- Joints limitation \( \Delta q \)

\[ \begin{bmatrix} W_{\text{joint.s}} \end{bmatrix} \begin{bmatrix} \Delta q \end{bmatrix} \begin{bmatrix} q_{\text{min}} - q \leq \Delta q \leq q_{\text{max}} - q \end{bmatrix} \]

\[ \min \zeta_{\text{joint.s}} = \begin{bmatrix} W_{\text{joint.s}} \end{bmatrix} \begin{bmatrix} \Delta q \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} \]

subject to \( H_{\text{joint.s}} \Delta q \geq h_{\text{joint.s}} \)
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

Results

The average time in each control loop for the boundary searching is ~6ms
Results: Robot vs Freehand

Freehand Error: 1.8 ± 1.1mm

Robot Error: 0.8 ± 0.4 mm

Approx 1.5:1 improvement in time!

<table>
<thead>
<tr>
<th>Trial#</th>
<th>Free hand</th>
<th>Robot Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>(mm)</td>
<td>(s)</td>
</tr>
<tr>
<td>1</td>
<td>1.785</td>
<td>26.354</td>
</tr>
<tr>
<td>2</td>
<td>1.632</td>
<td>29.358</td>
</tr>
<tr>
<td>3</td>
<td>1.796</td>
<td>27.372</td>
</tr>
<tr>
<td>4</td>
<td>2.061</td>
<td>25.436</td>
</tr>
<tr>
<td>5</td>
<td>2.119</td>
<td>24.533</td>
</tr>
<tr>
<td>avg</td>
<td>1.819</td>
<td>26.611</td>
</tr>
<tr>
<td>std</td>
<td>1.126</td>
<td>1.863</td>
</tr>
</tbody>
</table>

M. Li et al.
Combine constraints

**Single Frame**

\[
\begin{bmatrix}
A_p \\
A_r
\end{bmatrix}
J(q) \cdot \Delta q \leq 
\begin{bmatrix}
b_p \\
b_r
\end{bmatrix}
\]

**Multiple Frame**

\[
\begin{bmatrix}
A_1, 0 \\
\vdots \\
0, A_n
\end{bmatrix}
\begin{bmatrix}
J_1(q) \\
\vdots \\
J_n(q)
\end{bmatrix}
\Delta q \leq 
\begin{bmatrix}
b_1 \\
\vdots \\
b_n
\end{bmatrix}
\]

Select one or more

Customized virtual fixtures

5 Basic Geometric Constraints

(Virtual fixture library)

Optimization

\[\arg\min_{\Delta \tilde{q}} C(\tilde{x}(q + \Delta \tilde{q}), \tilde{s}, \tilde{x}^d)\]

s. t.

\[A(\tilde{x}(\tilde{q} + \Delta \tilde{q}), \tilde{s}) \leq \tilde{b},\]

\[\tilde{s}_{up} \geq \tilde{s} \geq \tilde{s}_{low} \geq 0,\]

\[\Delta \tilde{q}_{up} \geq \Delta \tilde{q} \geq \Delta \tilde{q}_{low} \]
Example: Suturing

The suturing task involves

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

Substep 4

Substep 2

Substep 5

Ideal Path

Center Line of wound

Exit Point

Entry Point

M. Li, A. Kapoor

Suturing: Align Step

0. Move Along a Line

1. Stay at a point + Rotate about a line

M. Li, A. Kapoor
Suturing: Align Step

2. Stay at a point + Rotate about a line

3. Puncture

4. Stay at a point + Rotate about a line

Suturing: Bite Step

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

<table>
<thead>
<tr>
<th>Error</th>
<th>Entry (mm)</th>
<th>Exit (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot</td>
<td>0.6375; σ = 0.12</td>
<td>0.7742; σ = 0.37</td>
</tr>
<tr>
<td>Manual</td>
<td>--</td>
<td>2.1; σ = 1.2</td>
</tr>
</tbody>
</table>

• 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

- 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.

“Soft” constraint implementation

Suppose that we have a problem of the form

\[
\Delta \hat{\mathbf{q}}_{\text{des}} = \arg \min \left\| E(\Delta \mathbf{q}) \right\|^2
\]

subject to a constraint of the form

\[
A_i(\Delta \mathbf{q}) \leq b_i
\]
"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

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

subject to a constraint of the form

$$A_i(\Delta \mathbf{q}) - s_i \leq b_i$$
$$0 \leq s_i \leq s_{\text{up},i}$$

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

Example: Stay near a point

Target Position: $\mathbf{x}_0$

After incremental motion

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

We want...

$$A(\mathbf{x}, s) = \| \delta_p + \Delta \mathbf{x}_p \|^2 - s \leq \epsilon_1$$

where $\delta_p = \mathbf{x}_p - \mathbf{x}_0$
Using Linear Constrained Quadratic Optimization

Matrix representation

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

Use Constrained Least Squares to solve

\[ \arg \min_{\Delta \vec{x}} \| \Delta \vec{x} - \Delta \vec{x}^d \|^2 \]

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

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
Example Task

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

Results for Example Task

- “Hard”, \(w_{b,i} = 1 \times 10^{-3}\)
- “Soft”, \(w_{b,i} = 1 \times 10^{-3}\)
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”

\[
\Delta \tilde{q}_{obs} = \arg \min_{\Delta \tilde{q}} C(\Delta \tilde{x}, \Delta \tilde{q}, \tilde{s})
\]

subject to
\[
\Delta \tilde{x} = J \Delta \tilde{q}
\]
\[
A(\Delta \tilde{x}, \Delta \tilde{q}, \tilde{s}) \leq \bar{b}
\]

Using Non-Linear Constrained Optimization

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

\[
\arg \min_{d^k} \nabla C(\bar{x}(\bar{q} + \Delta q^k), s^k, \bar{x}^d)\begin{pmatrix} d \end{pmatrix} + \frac{1}{2} d^t B_k d
\]

s.t. \[
\nabla A_k(\bar{x}(\bar{q} + \Delta q^k), s^k)^t \begin{pmatrix} d \end{pmatrix} \leq \bar{b} A_k
\]

• 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\]

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

Linear v. Non-Linear Constraints

<table>
<thead>
<tr>
<th># Hyperplanes used for approximation</th>
<th>4</th>
<th>6</th>
<th>10</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (ms)</td>
<td>2.5</td>
<td>3.5</td>
<td>5.0</td>
<td>7.5</td>
<td>12.5</td>
</tr>
</tbody>
</table>

Joint #3 is constrained

A. Kapoor, et al.
Effect of increasing control-loop time

- Large error at sharp turning
- Small interval reduces error

Ming Li et al., IROS '05

Information

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

General information
(anatomic atlases, statistics, rules)

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

Model → Plan → Action

Ankur Kapoor
High Dexterity in a Small Package

Taylor, Shoham, et al. 2002

Webster et al., 2006

Simaan, Taylor et al. 2004, 2007

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

4.2 mm
Snake Like Robot
System Architecture

Local Area Network (LAN)

Left
Right

High Level Controller

Low Level Controller

Stereo Laparoscope

Video Processing Unit

DaVinci Master

Dual Slave Arms

Snake Like Robot System Architecture

DaVinci Master Robot
High Level Constrained Control

Joint Positions ($q_m$)

Set Points ($q_m^d$)

Optimization Framework

$\arg\min_{\dot{q}_m} \| W(\ddot{x}_m - k\tau_m) \|^2$

s.t.

$H_1 \dot{x}_m \geq h_1$

$H_2 \dot{x}_s \geq h_2$

$\dot{x}_m = J_m \dot{q}_m$
Master Side High-Level Controller

- 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

\[
\min_{\Delta q_m} \left\| W_{m,t} \left( \Delta x_m - K_a \left[ p_m^c; \theta_m \omega_m \right] \right) \right\| \\
+ \left\| W_{m,s} \left( \Delta x_m - K_f \left[ p_m^s; \theta_m \omega_m \right] \right) \right\| \\
+ \left\| W_{m,j} \Delta q_m \right\|
\]

- Constraints:
  - General form: \( H_m \Delta q_m \geq h_m \)
  - Not allow motion outside joint range
  - Not allow motion that exceeds joint velocity limits
  - Additional constraints can be added from the VF Library

\[
\begin{bmatrix}
  I \\
  -I \\
  I \\
  -I
\end{bmatrix}
\begin{bmatrix}
  \Delta q_m \\
  \Delta q_m \omega_m \\
  \Delta q_m \omega_m \Delta t
\end{bmatrix}
\geq
\begin{bmatrix}
  q_m - q_m^L \\
  q_m^U - q_m \\
  q_m^U - q_m^L \\
  \Delta t
\end{bmatrix}
\]
DaVinci Slave Robot
High Level Constrained Control

Joint Positions ($q_s$) → $F_s = k_{ns}(q_s)$

Set Points ($q_s^d$)

Joint Velocities

Current Frame(s) Info. → Geometric Constraints on Frame(s)

From Master... $\tau_s = ^sF_m \cdot F_s^{-1}$

Optimization Framework

$$\arg\min_{q_s} \| W(\dot{x}_s - k\tau_s) \|^2_2$$

s. t. $H\dot{x}_s \geq h$

$\dot{x}_s = J_s q_s$

To Master... $F_s$

Slave-Side Snakes

- Actual snake section bends are a fairly complicated function of the linear displacements of the individual tubes and wires in the bending parts. But these displacements can be computed from the desired bending angles.
- Therefore, create pseudo-“joints” $q_{sec1}$ and $q_{sec2}$ corresponding to the bending angles in the two bend sections.
- Solve the optimization problem for $q_{sec1}$ and $q_{sec2}$ and the other joint angles of the slave robot. Then compute linear displacements from $q_{sec1}$ and $q_{sec2}$. This also involves some calculations for redundancy resolution that can be done with a similar optimization method or can be done analytically.
Slave Side High-Level Controller

- **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

- **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
Single Port Access Surgery

New technology finally allows true evaluation of the potential of single port access surgery. Systems raise new questions about control and telemanipulation infrastructure/cooperative control.

Single Port Access Robotic Surgery

Titan Medical Sport
https://www.youtube.com/watch?v=jllyvckA8xQ

Intuitive Surgical Sp
https://www.youtube.com/watch?v=jm5JdTrp4
Robot-Assisted Skull Base Surgery

Integration of a Snake-like Dexterous Manipulator for Head and Neck Surgery with the da Vinci Research Kit

Large Lumen, Dexterous Snake for MIS

- Joint project with JHU APL
- Innovative fabrication process completely isolates drive cables
- Current prototypes
  - 2 DoF (C-bend) and 4DoF (S-bend)
  - Nitinol structure with high stiffness
  - 6 mm OD; Large 4 mm lumen allows insertion of surgical instruments
- Initial application: Minimally-invasive curettage of osteolytic lesions

Treatment of Osteolytic Lesions

- Indication: Osteolysis behind a well-fixed acetabular component
  - Leads to component loosening and failure of THA
- Surgical Goals
  - Minimally invasive removal of the osteolytic lesion
  - Treatment of the lysis without full replacement of the acetabular component
- Surgical Procedure
  - Access the lesion through the screw holes of the acetabular component (minimally invasive)
  - Remove and grafting the lesion
  - Replace the polyethylene liner
Novel Shape Sensor Array (SSA) for Large Curvature Detecting

Using FBG sensors along with nitinol wires as the supporting substrates, we can prevent local stress concentration, therefore maximizing curvature detection range.
Curved Drilling of the Femoral Head

Alambeigi, et al.

- Osteonecrosis of the femoral head
  - More than 20,000 patients per year
  - To reduce the pressure in the femoral head, core decompression was developed more than three decades ago.

- Steerable “snake” with flexible drill provides better access to femoral head volume than does conventional straight drill
Curved Drilling of the Femoral Head
Alambeigi, et al.

- Sample Holder Mechanism: feeding motion and 6DOF force sensor
- Thermal Camera: "Real-time" tracking of the cutter


Curved Drilling of the Femoral Head
Alambeigi, et al.

S-Shape and multiple branch curved-drilling

Curved-Drilling Experiments on human cadaver specimens

Robot-Assisted Skull Base Surgery

Integration of a Snake-like Dexterous Manipulator for Head and Neck Surgery with the da Vinci Research Kit

S. Coemert, F. Alambeigi, A. Deguet, J. P. Carey, M. Armand, T. C. Lueth, R. H. Taylor

Handheld actuation concept: a) C-shaped b) S-shaped

Overview of the DVRK system [5] Video: Actuation of the SDM attached to DVRK

Challenges in Precise Minimally Invasive Head- and Neck Surgery

• Long (25cm) instruments
  – amplify hand tremor
  – reduce precision
• Tight spaces near sensitive anatomy
• Limited working area

The Robotic ENT Microsurgery System (REMS)

User interface:
• Hands-on control, surgeon “in the game”
• Foot pedal-controlled gain

Technical specs:
• Up to 0.025 mm precision on-demand
• 6 degrees of freedom
• 125x125x125mm work volume
• Calibrated accuracy ~50-150μm

Control modes:
• Free hand
• Remote center of motion
• Virtual fixture avoidance
• Teleoperation

K. Olds, Robotic Assistant Systems for Otolaryngology-Head and Neck Surgery, PhD thesis in Biomedical Engineering, Johns Hopkins University, Baltimore, March 2015.
Playing the “Operation Game” with Long Instruments

Microlaryngeal Phonosurgery "Operation" Game Demo

K. Olds, Robotic Assistant Systems for Otolaryngology-Head and Neck Surgery, PhD thesis in Biomedical Engineering, Johns Hopkins University, Baltimore, March 2015.

REMS Typical Applications

Laryngeal / Vocal Cord

Open Microsurgery

Other applications include:

- Otology
  - Stapes surgery
  - Mastoidectomy
  - Cochlear implant
- Craniotomy
- Spine
- Hand
- ...

Image-guided sinus surgery with virtual fixtures
Vitreoretinal Microsurgery

Microsurgery Assistant Workstation

- 3D Display with Overlays
- Stereo video Microscope
- OCT Display
- Audio Output
- EyeRobot2
- Force and OCT sensing tools
- FBG Interrogator
Retina Mosaicking, Annotation, and Registration

R. Richa, B. Vagvolgyi, R. Taylor, G. Hager, MICCAI 2012,

JHU Steady Hand “Eye Robot”

- Highly precise robot
- Hands-on cooperative control or teleoperation
- Several generations in lab
- Precise, stable platform for developing “smart” surgical instruments and sensors
- Virtual fixtures and advanced control
Retinal Microsurgery – *in vivo* experiments

- Overall System Performance
- System Ergonomics
- Collect Data
  - Robot / Force / OCT
  - Video / Audio

**Force-Sensing Micro-Forceps Tools for Vitreoretinal Surgery**

**2-DOF Force-Sensing Micro-Forceps (Handheld)**

**3-DOF Force-Sensing Micro-Forceps (Steady-Hand Robot Compatible) - Concept Design**

**Motorized 2-DOF Force-Sensing Micro-Forceps (Compatible with handheld manipulators)**
In-vivo experiments

- Test the force sensing micro-forceps in-vivo using rabbit in the operating room
- Force measurements, stereo microscopic video, and surgeon’s voice annotation were recorded with timestamps for synchronization and analysis

Dual Force Sensor

Xingchi He, Marcin Balicki, Jin U. Kang, Peter Gehlbach, James Handa, Russell Taylor, Iulian Iordachita

"Force sensing micro-forceps with integrated fiber Bragg grating for vitreoretinal surgery", SPIE 202

Dual Force Sensor

Follow the Motion of the Eye


µForce Scaling Cooperative Control

Cooperative Control
Velocity at the tool \((V)\) is proportional to \((\alpha\) gain\) the user’s input force at the handle \((F_h)\)

\[
\dot{x} = \alpha F_h
\]

µForce Scaling
Amplifies \((\gamma\) gain\) the human-imperceptible forces sensed at the tool tip \((F_t)\) to handle interaction forces \((F_p)\) by modulating robot velocity.

\[
x = \alpha\left(F_h - \gamma F_t\right), \quad \text{e.g., } \gamma = 500
\]
\textbf{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 \textit{et al.}, BioRob 2010

\[ f_{\text{tim}} = f_{\text{max}} \frac{|f_h|}{||f_h||} \]
\[ \dot{x}_{\text{tim}} = \dot{x} \left( \frac{f_{\text{tim}} - |f_t|}{f_{\text{spring}}} \right) \]

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

Local Force Minimization
- Guiding user towards direction of minimum resistance
- Sensitivity variable allows user override
- Haptically intuitive response
- Avoids / postpones reaching limits

Uneri \textit{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
Enhanced Cooperative Control Peeling Algorithm

The algorithm biases operator-robot interaction towards the direction of least tissue resistance while limiting forces.

Peeling angles converge to $45^\circ$

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 $45^\circ$
- User was guided away from the centerline in the direction of lowest resistance

Uneri et al., BioRob 2010
Experiment:
\( \mu \text{Force Guided Cooperative Control} \)

- 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

Uneri et al., BioRob 2010

Imaging (OCT) Built Into 0.5mm Surgical Tool
M. Balicki, J. Han, X. Liu, I. Iordachita, P. Gehlbach, J. Handa, R. Taylor, J. Kang.

- Fourier Domain Common Path OCT (FD CPOCT)
- Combined Superluminescent Diodes
- Functional and structural images
Autonomous Surface Following

M. Balicki, J.-H. Han, I. Iordachita, P. Gehlbach, J. Handa, R. H. Taylor, and J. Kang, MICCAI 2010

\* 500 µm/s Velocity Limit

Noise Rem. /Thresholded/Canny

OCT-feedback in fast hand-held robot

Safety Barrier

Smart Micro-Forceps
OCT-servoed Injections
J. U. Kang, G.W. Cheon, and P. Gehlbach

Free Hand

Active Approach

Lock-In

Target Position

Elapsed time [sec]

Position [cm]

Elapsed time [sec]


Robotically-Assisted Insertion of Cochlear Implants

- Setup
  - Phantom cochlea
  - Stiffer stylet
  - Surgeon and novice inserted implants into phantom using three methods:
    - Manual insertion
    - Robot-assisted insertion
    - Robot-assisted insertion with virtual fixtures enacted

Robotically-Assisted Insertion of Cochlear Implants

Novice’s deployment points

- Each point represents the deployment point reached using one of the insertion methods
- Manual spread very far, not very accurate
- Robot-assisted also spread far, closer to center
- Robot-assisted with virtual fixtures closely clustered and highly accurate

Information-enhanced robotic surgery

- Augmented reality displays imaging
- Safety barriers shared control “virtual fixtures”

SAW

Information-enhanced robotic surgery

- Augmented reality displays imaging
- Tool motions
- Safety barriers shared control “virtual fixtures”
Information-enhanced robotic surgery

- Augmented reality displays imaging
- Tool motions
- Positions
- Forces
- Fast local compliance law
- Safety barriers shared control "virtual fixtures"

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

\[ \mathbf{F} = [\mathbf{R}, \mathbf{p}] = \text{current pose}; \quad \dot{\mathbf{p}} = \text{current velocity} \]

\[ \mathbf{F}_c = [\mathbf{R}_c, \dot{\mathbf{p}}_c] = \text{position compliance frame} \]

\[ \mathbf{k}^+(\cdot), \mathbf{k}^-(\cdot) = \text{position stiffness factors} \]

\[ \mathbf{b}^+(\cdot), \mathbf{b}^-(\cdot) = \text{damping factors} \]

\[ \mathbf{g}^+(\cdot), \mathbf{g}^-(\cdot) = \text{force bias terms} \]

\[ \mathbf{R}_o = \text{orientation compliance frame} \]

\[ \mathbf{k}_o^+(\cdot), \mathbf{k}_o^-(\cdot) = \text{orientation stiffness factors} \]

\[ \mathbf{n}^+(\cdot), \mathbf{n}^-(\cdot) = \text{torque bias terms} \]

\[ t = \text{time remaining on timeout counter} \]

---

Compliance virtual fixtures

if \((t>0)\) then

begin

\[ t = t - 1 \]

\[ \mathbf{q} = \mathbf{F}_c \mathbf{p} = \mathbf{R}_c (\mathbf{p} - \mathbf{p}_c) \]

\[ \mathbf{v} = \mathbf{R}_c \dot{\mathbf{p}} \]

\[ \mathbf{h} = \mathbf{0}, \mathbf{\psi} = \mathbf{0} \]

for \(i \in \{x, y, z\}\) do

\( \begin{cases} 
\text{if } \dot{q}_i \leq 0 & \text{then } \dot{h}_i = \dot{g}_i + \dot{k}_i \dot{q}_i + \dot{b}_i \dot{v}_i \quad \text{else } \dot{h}_i = \dot{g}_i + \dot{k}_i \dot{q}_i + \dot{b}_i \dot{v}_i \end{cases} \)

end

\[ \mathbf{\tilde{q}} = \text{Rodrigues vector corresponding to } \Delta \mathbf{R} = \mathbf{R}_x \Delta \mathbf{R} \]

for \(i \in \{x, y, z\}\) do

\( \begin{cases} 
\text{if } \dot{\delta}_i \leq 0 & \text{then } \dot{\psi}_i = \dot{\alpha}_i + \dot{k}_\alpha \dot{\psi} + \dot{\beta}_i \dot{v}_i \quad \text{else } \dot{\psi}_i = \dot{\alpha}_i + \dot{k}_\alpha \dot{\psi} + \dot{\beta}_i \dot{v}_i \end{cases} \)

add \(\mathbf{R}_o \dot{\psi}_i\) 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

\[
\begin{align*}
\mathbf{p}_c &= \text{closest point on surface} \\
\mathbf{R}_c \hat{z} &= \text{surface normal at } \mathbf{p}_c
\end{align*}
\]

\[
\begin{align*}
\mathbf{k}^{(\cdot)} &= [0, 0, -\text{stiffness}] \\
\mathbf{g}^{(\cdot)} &= [0, 0, -\text{bias}] \\
\text{Others} &= 0
\end{align*}
\]

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. For example:
  
  Compute \( \mathbf{q}, \mathbf{v}, \mathbf{\dot{q}}, \mathbf{\ddot{q}} \) from \( \mathbf{F}_c \) and \( \mathbf{R}_o \), where \( \mathbf{\ddot{q}} = \frac{d\mathbf{\dot{q}}}{dt} \)
  
  Compute a region \( i \) of local configuration space from \( \mathbf{q} \) and \( \mathbf{\dot{q}} \)

  \[
  \begin{bmatrix}
  \mathbf{h}^{\cdot} \\
  \mathbf{\phi}^{\cdot}
  \end{bmatrix} = \mathbf{K} \cdot \begin{bmatrix}
  \mathbf{q} \\
  \mathbf{\dot{q}}
  \end{bmatrix} + \mathbf{B} \cdot \begin{bmatrix}
  \mathbf{v} \\
  \mathbf{\dot{q}}
  \end{bmatrix} + \begin{bmatrix}
  \mathbf{g}_i \\
  \mathbf{\tau}_i
  \end{bmatrix}
  \]

  Add \( \mathbf{R}_e \mathbf{h} \) to master forces and \( \mathbf{R}_e \mathbf{\phi} \) to master torques