Medical Robots, Constrained Robot Motion Control, and "Virtual Fixtures"

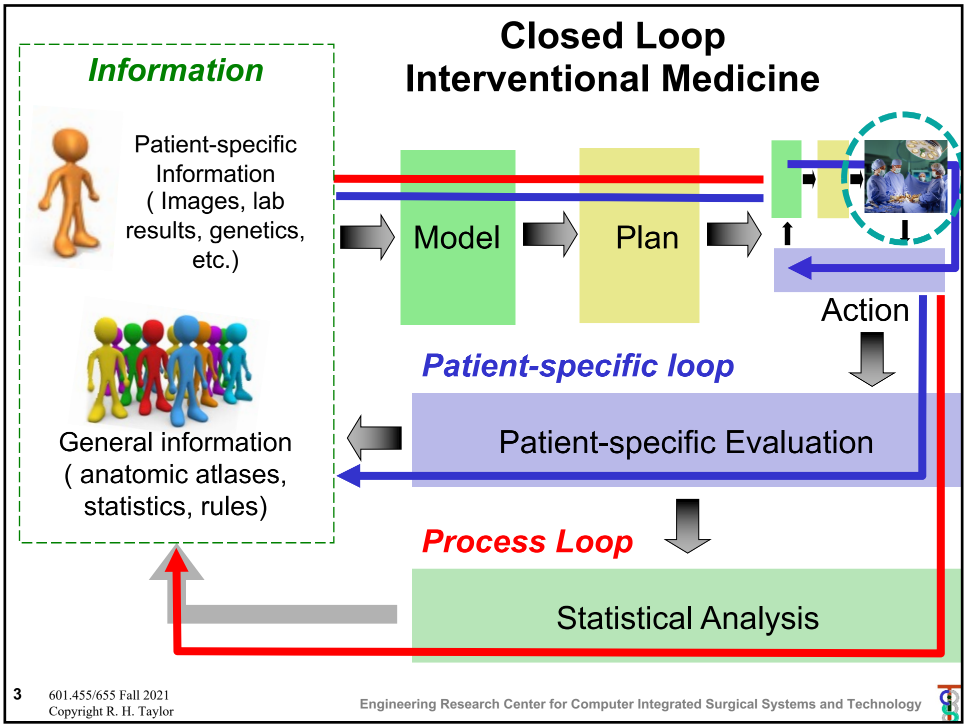
(Part 1)

Russell H. Taylor

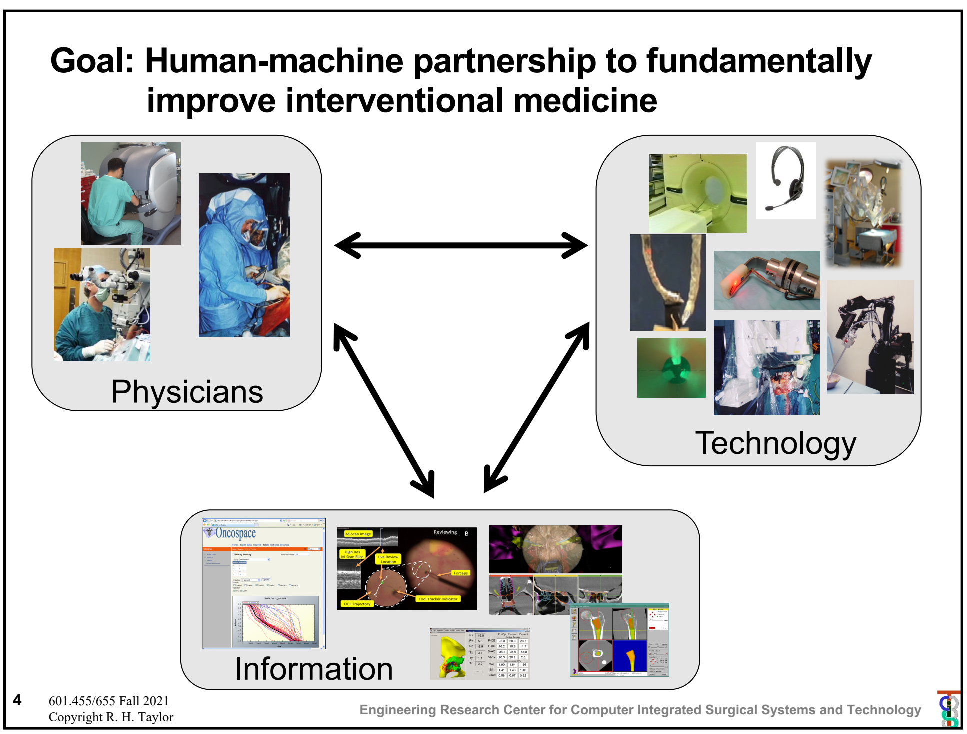
601.455/655

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 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 IIS0099770 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 Swirnow and Paul Maritz; and by Johns Hopkins University internal funds.


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
3
601.455/655 Fall 2021
Copyright R. H. Taylor
Engineering Research Center for Computer Integrated Surgical Systems and Technology

3


Goal: Human-machine partnership to fundamentally improve interventional medicine
Physicians
Technology
Information
4
601.455/655 Fall 2021
Copyright R. H. Taylor
Engineering Research Center for Computer Integrated Surgical Systems and Technology

4

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.

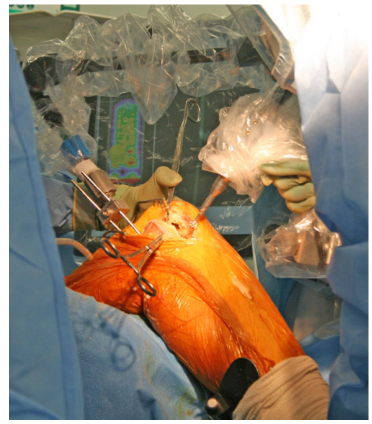
Surgical CAD/CAM: Orthopaedic Robots

Robodoc

Blue Belt Technologies

D. Glozman
& M. Shoham

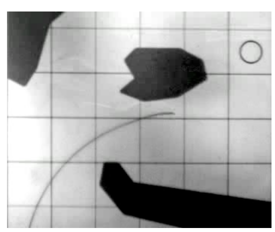
Mako Robotics Rio
http://www.makosurgical.com/



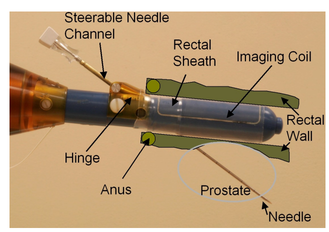
ACROBOT surgical robot

Image-guided needle placement

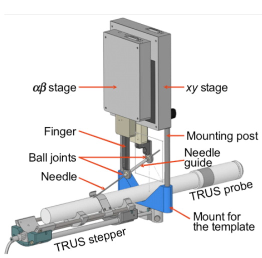
Masamune, Fichtinger, Iordachita, …



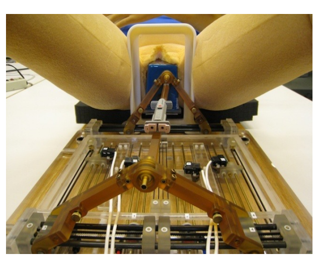
Okamura, Webster, …



Krieger, Fichtinger, Whitcomb, …



Fichtinger, Kazanzides,Burdette, Song …



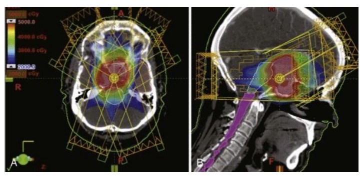
Iordachita, Fischer, Hata…

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

Cyberknife



Varian Trilogy System

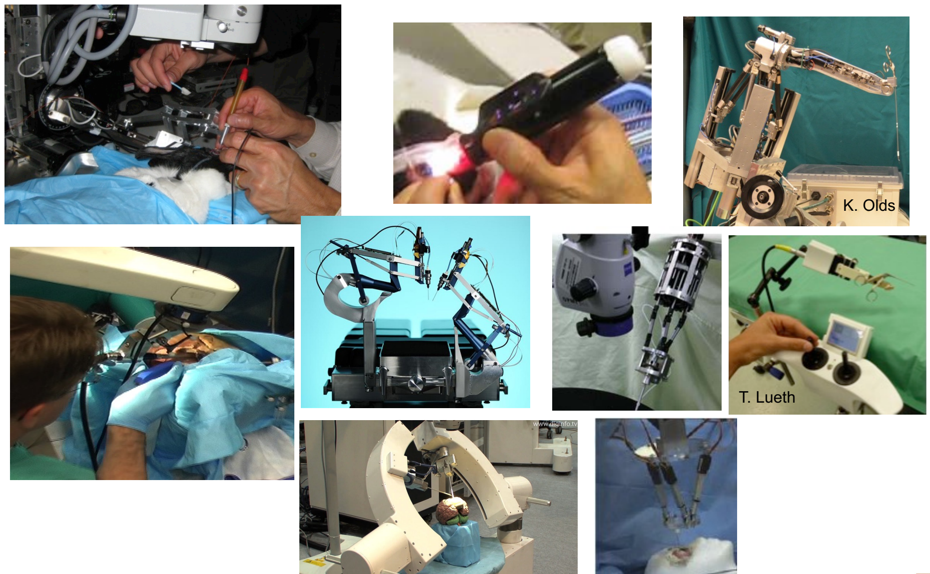
Slide credit: Howie Choset + RHT

http://www.varian.com/us/oncology/radiation_oncology/trilogy/

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

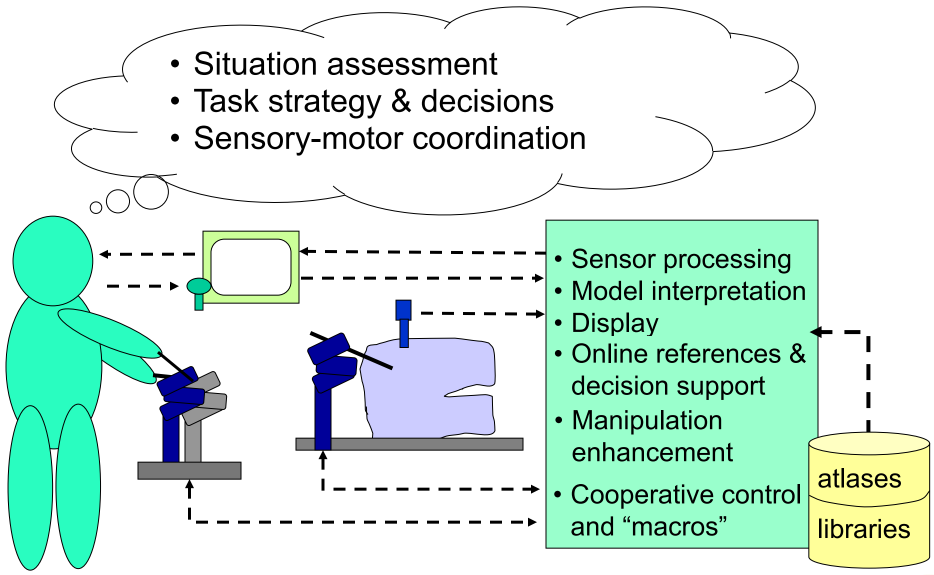
Precision Augmentation



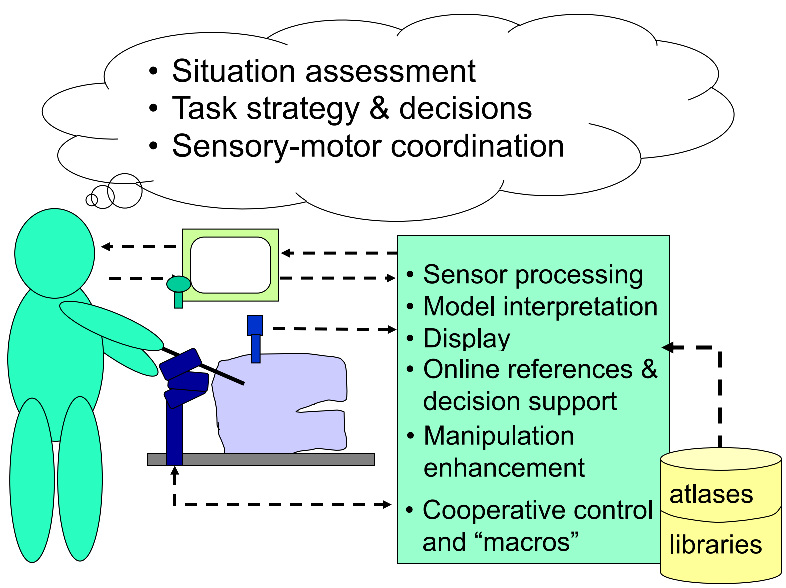
Common classes of medical robots

- **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.

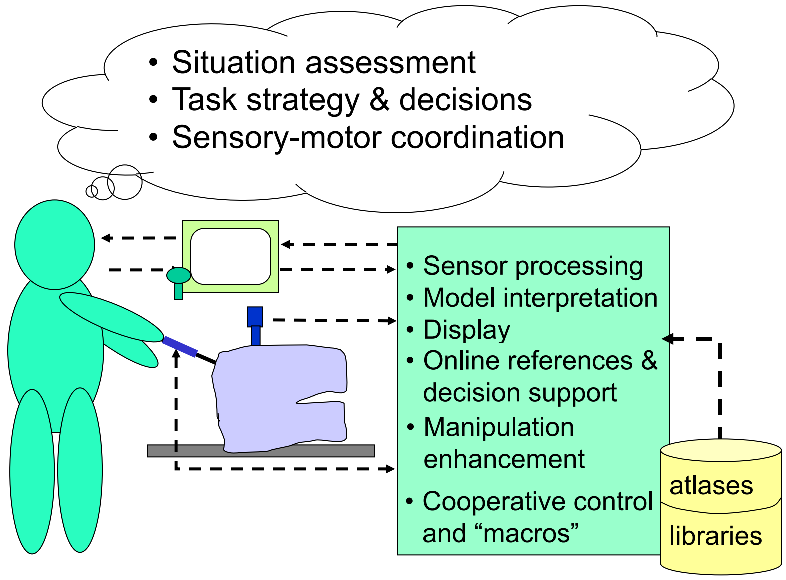
Surgical Assistant Systems



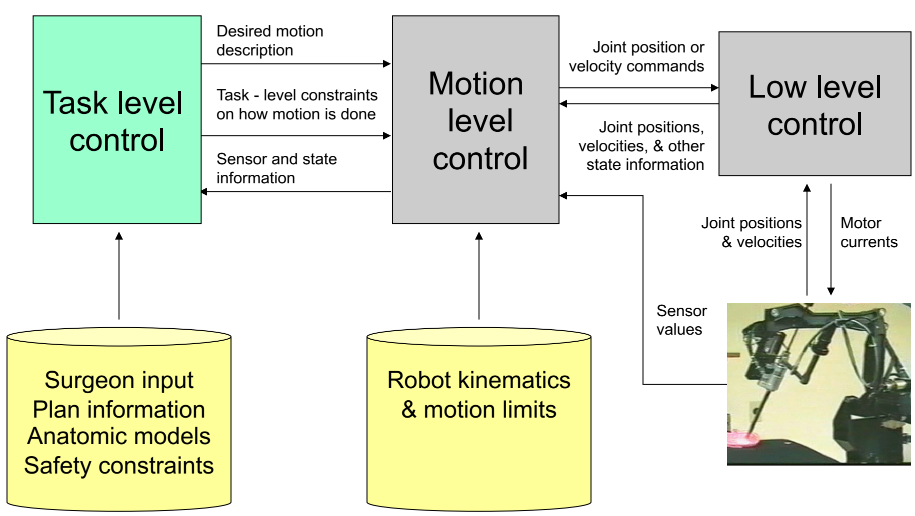
Surgical Assistant Systems



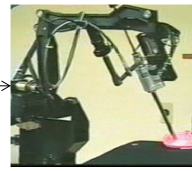
Surgical Assistant Systems



Problem: specifying motion for a [medical] robot



Background: Robot Kinematics

Joint positions $\vec{\mathbf{q}}$ →  → Pose $\mathbf{F} = kins(\vec{\mathbf{q}})$

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

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

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

One implementation

Task level control

$\Delta\mathbf{F}_{des}$

Motion level control

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

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

Position control

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

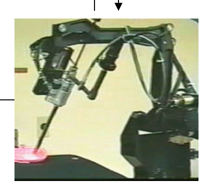
$\mathbf{F}, \vec{\mathbf{q}}$

Sensor values

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

Motor currents

Sensor values



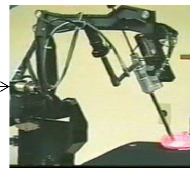
Surgeon input
Plan information
Anatomic models
Safety constraints

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

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 $\mathbf{F}$. I.e., $\mathbf{F}=Kins(\vec{\mathbf{q}})$, where $Kins(\cdots)$ is a function computing the "forward kinematics" of the robot.

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



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

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

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 $\mathbf{F}$. I.e., $\mathbf{F}=Kins(\vec{\mathbf{q}})$, where $Kins(\cdots)$ is a function computing the "forward kinematics" of the robot. Let $\Delta\mathbf{F}\bullet\mathbf{F}=Kins(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}})$

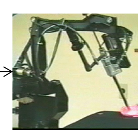
For small $\Delta\vec{\mathbf{q}}$, we can write the following expression for $\Delta\mathbf{F}=[Rot(\vec{\alpha}),\vec{\varepsilon}]$

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

which we typically linearize as

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

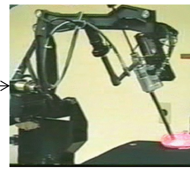
$\Delta\vec{\mathbf{q}}$



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

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

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

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

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

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

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

 601.455/655 Fall 2021

Engineering Research Center for Computer Integrated Surgical Systems and Technology

22

Background: Jacobean Robot Motion Control

Alternative way of solving:

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

$$\Delta\vec{\mathbf{q}} \approx \underset{\Delta\vec{\mathbf{q}}}{\arg\min}\left\| \mathbf{J}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}} - \begin{bmatrix} \vec{\alpha} \\ \varepsilon \end{bmatrix} \right\|^2$$

Advantages:

- Produces solution even if kinematically redundant or kinematically deficient
- Can add auxiliary constraints or objective function terms

 601.455/655 Fall 2021

Engineering Research Center for Computer Integrated Surgical Systems and Technology

23

Jacobean motion control 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

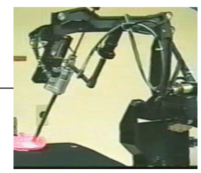
$\mathbf{F}, \vec{\mathbf{q}}$

Sensor values

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

Motor currents or voltages

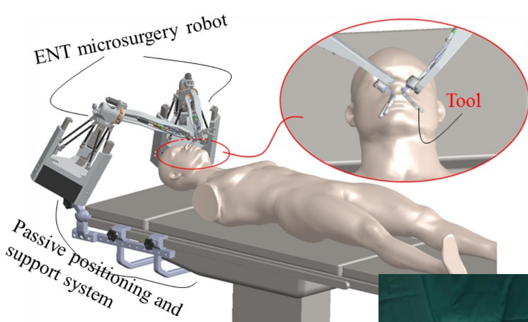
Sensor values

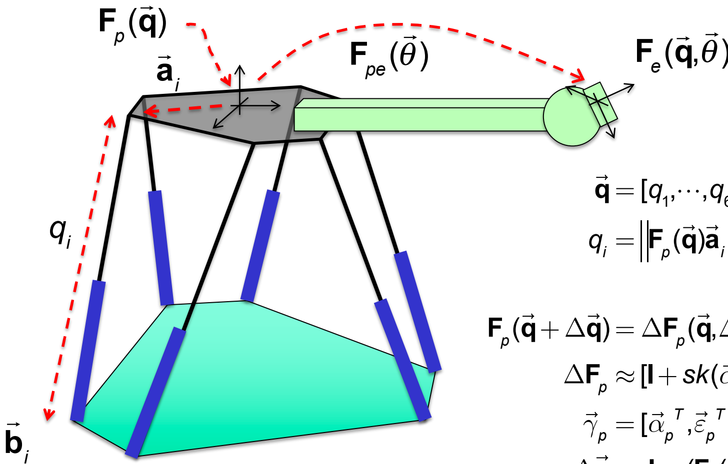


Surgeon input
Plan information
Anatomic models
Safety constraints

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

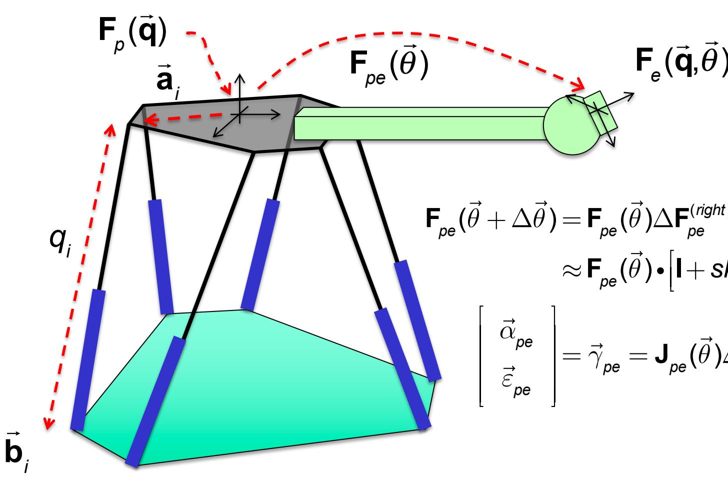
What about parallel-link robots?



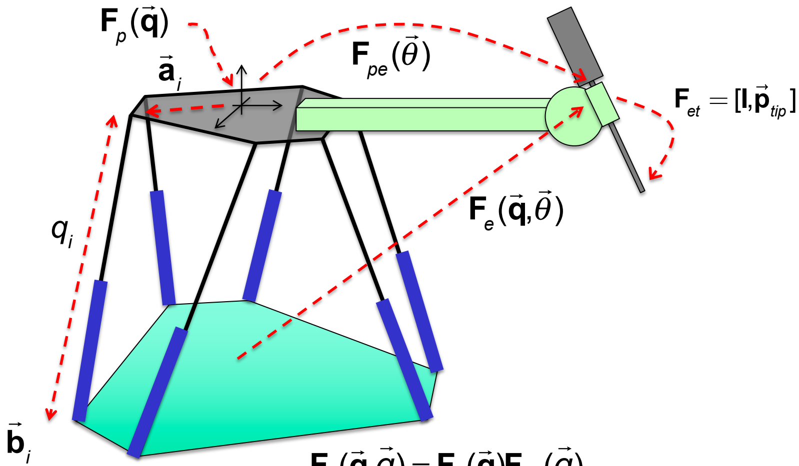


$$\vec{\mathbf{q}} = [q_1,\cdots,q_6]^T = invk(\mathbf{F}_p)$$
$$q_i = \left\| \mathbf{F}_p(\vec{\mathbf{q}})\vec{\mathbf{a}}_i - \mathbf{b}_i \right\|$$

$$\mathbf{F}_p(\vec{\mathbf{q}} + \Delta\vec{\mathbf{q}}) = \Delta\mathbf{F}_p(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})$$
$$\Delta\mathbf{F}_p \approx [\mathbf{I} + sk(\vec{\alpha}_p),\vec{\varepsilon}_p]$$
$$\vec{\gamma}_p = [\vec{\alpha}_p{}^T,\vec{\varepsilon}_p{}^T]^T$$
$$\Delta\vec{\mathbf{q}} \approx \mathbf{J}_{invk}(\mathbf{F}_p(\vec{\mathbf{q}}))\vec{\gamma}_p$$
$$\vec{\gamma}_p \approx \mathbf{J}_{invk}(\mathbf{F}_p(\vec{\mathbf{q}}))^{-1}\Delta\vec{\mathbf{q}}$$



$$\mathbf{F}_{pe}(\vec{\theta} + \Delta\vec{\theta}) = \mathbf{F}_{pe}(\vec{\theta})\Delta\mathbf{F}_{pe}^{(right)}(\vec{\theta},\Delta\vec{\theta})$$
$$\approx \mathbf{F}_{pe}(\vec{\theta}) \bullet \left[\mathbf{I} + sk(\vec{\alpha}_{pe}),\vec{\varepsilon}_{pe}\right]$$
$$\begin{bmatrix} \vec{\alpha}_{pe} \\ \vec{\varepsilon}_{pe} \end{bmatrix} = \vec{\gamma}_{pe} = \mathbf{J}_{pe}(\vec{\theta})\Delta\vec{\theta} = \begin{bmatrix} \mathbf{J}_{pe}^{\mathbf{R}}(\vec{\theta}) \\ \mathbf{J}_{pe}^{\vec{\mathbf{p}}}(\vec{\theta}) \end{bmatrix}\Delta\vec{\theta}$$



$$\mathbf{F}_e(\vec{\mathbf{q}}, \vec{q}) = \mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{q})$$

$$\mathbf{F}_e(\vec{\mathbf{q}} + \Delta\vec{\mathbf{q}}, \vec{q} + \Delta\vec{q}) = \Delta\mathbf{F}_p(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{q} + \Delta\vec{q})$$

$$\mathbf{F}_e(\vec{\mathbf{q}}, \vec{\theta}) = \mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta})$$

$$\mathbf{F}_e(\vec{\mathbf{q}} + \Delta\vec{\mathbf{q}}, \vec{\theta} + \Delta\vec{\theta}) = \Delta\mathbf{F}_p(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta} + \Delta\vec{\theta})$$

$$\Delta\mathbf{F}_e(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}}, \vec{\theta}, \Delta\vec{\theta})\mathbf{F}_e(\vec{\mathbf{q}}, \vec{\theta}) = \Delta\mathbf{F}_p(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta})\Delta\mathbf{F}_{pe}(\vec{\theta}, \Delta\vec{\theta})$$

$$\begin{aligned}\Delta\mathbf{F}_e(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}}, \vec{\theta}, \Delta\vec{\theta}) &= \Delta\mathbf{F}_p(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta})\Delta\mathbf{F}_{pe}(\vec{\theta}, \Delta\vec{\theta})\mathbf{F}_e(\vec{\mathbf{q}}, \vec{\theta})^{-1} \\ &= \Delta\mathbf{F}_p(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta})\Delta\mathbf{F}_{pe}(\vec{\theta}, \Delta\vec{\theta})\mathbf{F}_e(\vec{\mathbf{q}}, \vec{\theta})^{-1}\end{aligned}$$

$$\Delta\mathbf{R}_e(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}}, \vec{\theta}, \Delta\vec{\theta}) = \Delta\mathbf{R}_p(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}})\mathbf{R}_p(\vec{\mathbf{q}})\mathbf{R}_{pe}(\vec{\theta})\Delta\mathbf{R}_{pe}(\vec{\theta}, \Delta\vec{\theta})\Delta\mathbf{R}_{pe}(\vec{\mathbf{q}}, \vec{\theta})^{-1}\mathbf{R}_{pe}(\vec{\mathbf{q}}, \vec{\theta})^{-1}$$

$$\mathbf{I} + sk(\vec{\alpha}_e) \approx \left(\mathbf{I} + sk(\vec{\alpha}_p)\right)\mathbf{R}_e(\vec{\mathbf{q}}, \vec{\theta})\left(\mathbf{I} - sk(\vec{\alpha}_{pe})\right)\mathbf{R}_e(\vec{\mathbf{q}}, \vec{\theta})^{-1}$$

$$\mathbf{I} + sk(\vec{\alpha}_e) \approx \left(\mathbf{I} + sk(\vec{\alpha}_p)\right)\left(\mathbf{I} - \mathbf{R}_e(\vec{\mathbf{q}}, \vec{\theta})sk(\vec{\alpha}_{pe})\mathbf{R}_e(\vec{\mathbf{q}}, \vec{\theta})^{-1}\right)$$

$$\mathbf{I} + sk(\vec{\alpha}_e) \approx \left(\mathbf{I} + sk(\vec{\alpha}_p)\right)\left(\mathbf{I} - sk(\mathbf{R}_e(\vec{\mathbf{q}}, \vec{\theta})\vec{\alpha}_{pe})\right)$$

$$\vec{\alpha}_e \approx \vec{\alpha}_p - \mathbf{R}_e(\vec{\mathbf{q}}, \vec{\theta})\vec{\alpha}_{pe}$$

$$\vec{\alpha}_e \approx \mathbf{J}^{\vec{\alpha}}_{invk}(\mathbf{F}_p(\vec{\mathbf{q}}))^{-1}\Delta\vec{\mathbf{q}} - \mathbf{R}_e(\vec{\mathbf{q}}, \vec{\theta})\vec{\alpha}_{pe}$$

$$\mathbf{F}_e(\vec{\mathbf{q}},\vec{\theta}) = \mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta})$$

$$\mathbf{F}_e(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}},\vec{\theta}+\Delta\vec{\theta}) = \Delta\mathbf{F}_p(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta}+\Delta\vec{\theta})$$

$$\Delta\mathbf{F}_e(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}},\vec{\theta},\Delta\vec{\theta})\mathbf{F}_e(\vec{\mathbf{q}},\vec{\theta}) = \Delta\mathbf{F}_p(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\mathbf{F}_p(\vec{\mathbf{q}})\mathbf{F}_{pe}(\vec{\theta})\Delta\mathbf{F}_{pe}(\vec{\theta},\Delta\vec{\theta})$$

$$\Delta\mathbf{F}_e(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}},\vec{\theta},\Delta\vec{\theta})\vec{\mathbf{p}}_e = \Delta\mathbf{F}_p(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\mathbf{F}_e\Delta\vec{\mathbf{p}}_{pe}$$

$$\Delta\mathbf{R}_e(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}},\vec{\theta},\Delta\vec{\theta})\vec{\mathbf{p}}_e + \Delta\vec{\mathbf{p}}_e = \Delta\mathbf{R}_p(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\left(\mathbf{R}_e\Delta\vec{\mathbf{p}}_{pe} + \vec{\mathbf{p}}_e\right) + \Delta\vec{\mathbf{p}}_p$$

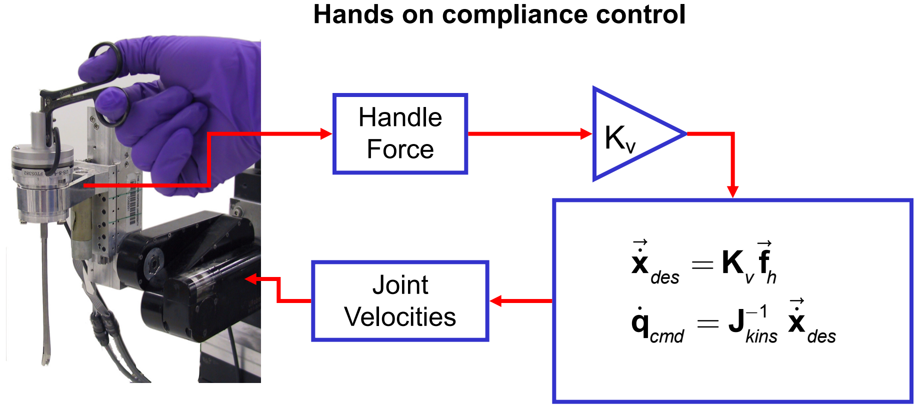
$$\left(\mathbf{I} + sk(\vec{\alpha}_e)\right)\vec{\mathbf{p}}_e + \vec{\varepsilon}_e \approx \left(\mathbf{I} + sk(\vec{\alpha}_p)\right)\left(\mathbf{R}_e\vec{\varepsilon}_{pe} + \vec{\mathbf{p}}_e\right) + \vec{\varepsilon}_p$$

$$\vec{\mathbf{p}}_e + sk(\vec{\alpha}_e)\vec{\mathbf{p}}_e + \vec{\varepsilon}_e \approx \vec{\mathbf{p}}_e + sk(\vec{\alpha}_p)\vec{\mathbf{p}}_e + \mathbf{R}_e\vec{\varepsilon}_{pe} + \vec{\varepsilon}_p$$

$$\vec{\varepsilon}_e \approx -sk(\vec{\mathbf{p}}_e)\vec{\alpha}_p + \mathbf{R}_e\vec{\varepsilon}_{pe} + \vec{\varepsilon}_p$$

Steady Hand Robot

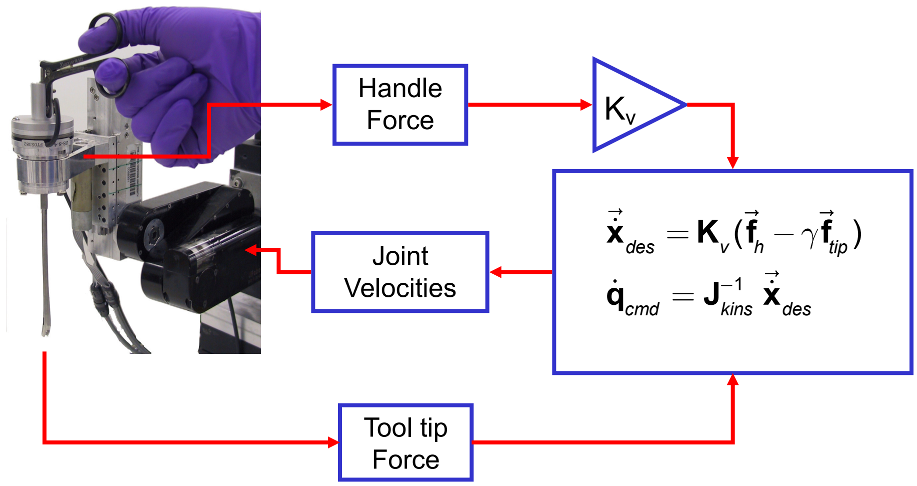
Hands on compliance control



[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



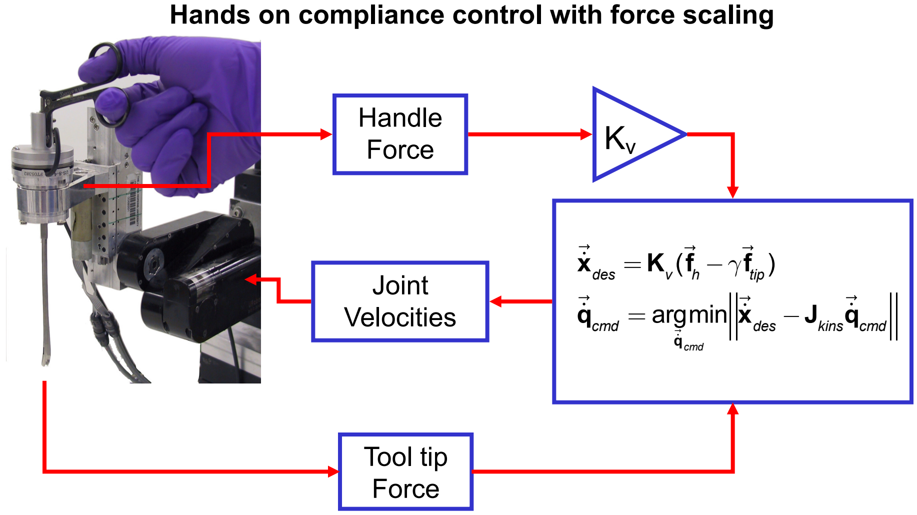
$$\dot{\vec{\mathbf{x}}}_{des} = \mathbf{K}_v(\vec{\mathbf{f}}_h - \gamma \vec{\mathbf{f}}_{tip})$$

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

[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

Steady Hand Robot (Alternative Formulation)

Hands on compliance control with force scaling

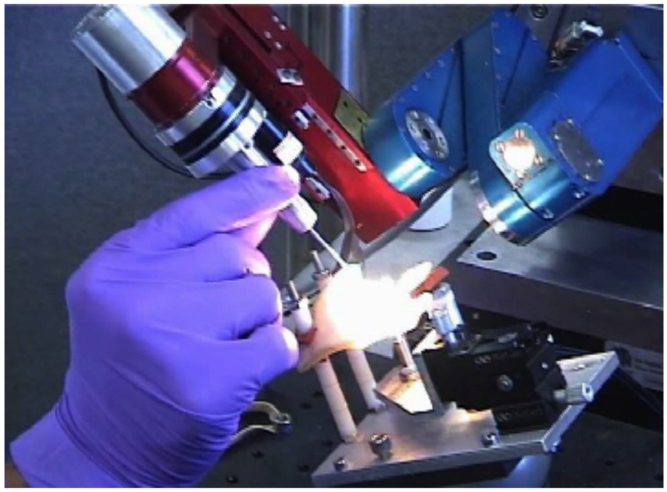


$$\dot{\vec{\mathbf{x}}}_{des} = \mathbf{K}_v(\vec{\mathbf{f}}_h - \gamma \vec{\mathbf{f}}_{tip})$$

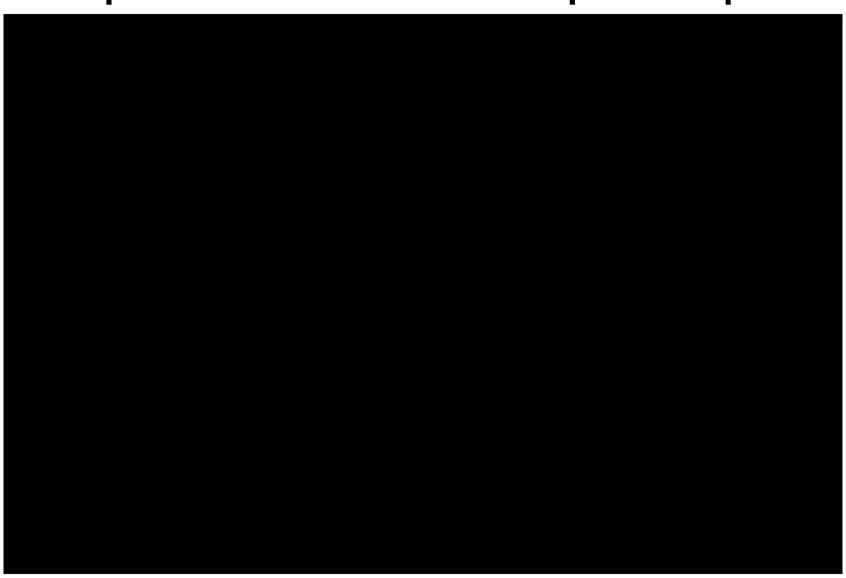
$$\dot{\vec{\mathbf{q}}}_{cmd} = \underset{\dot{\vec{\mathbf{q}}}_{cmd}}{\arg\min} \left\| \dot{\vec{\mathbf{x}}}_{des} - \mathbf{J}_{kins}\dot{\vec{\mathbf{q}}}_{cmd} \right\|$$

[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

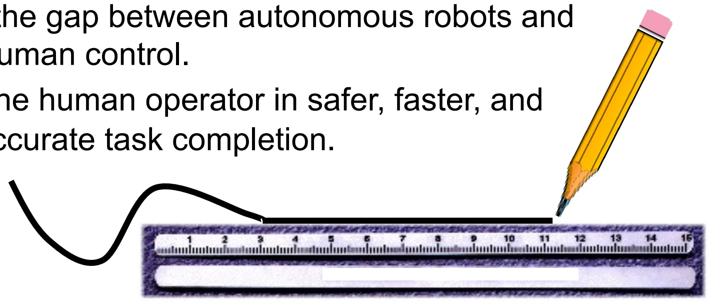


Example: Fenestratration of Stapes Footplate



Virtual Fixtures

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



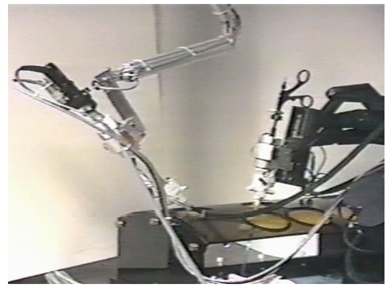
- Broadly Categorized
  - Guidance VF
  - Forbidden Region VF
- Different implementation
  - Tele-manipulation
  - Cooperative Control

Background: Virtual Fixtures

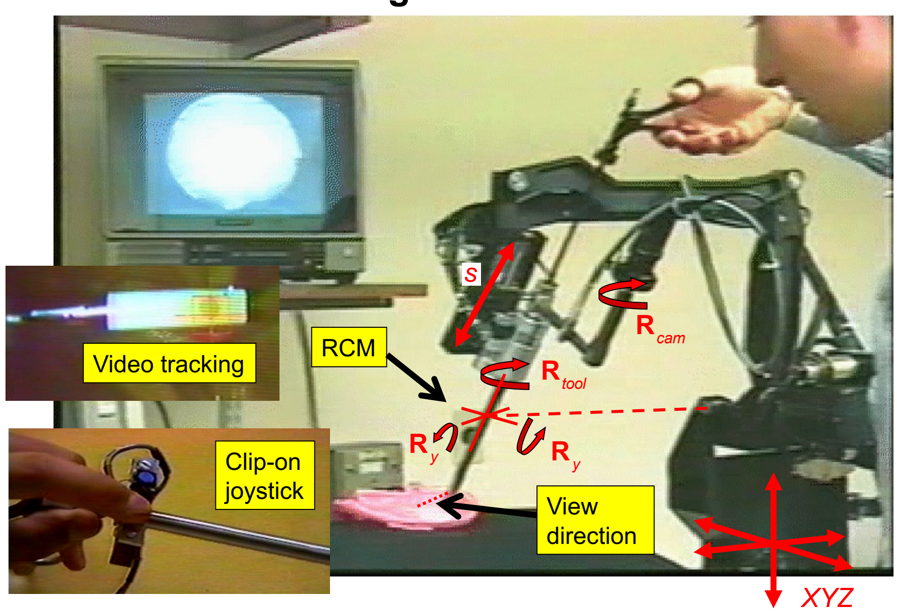
- First proposed for complex telerobotic tasks, but draw upon rich prior research in robot assembly and other manufacturing automation applications
- Many authors, e.g.,
  - L. B. Rosenberg, "Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation," *Proc. IEEE Virtual Reality International Symposium,* 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

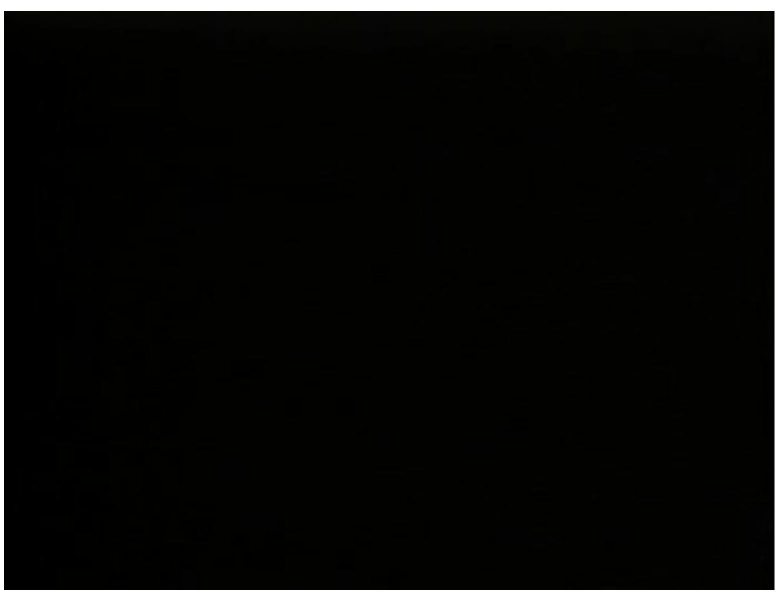


LARS degrees of freedom



LARS Video

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 Jacobean formulation to transform to joint space
- Solve for joint motion

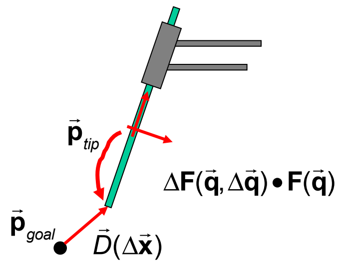
Example: keep tool tip near a point

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

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

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

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



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

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

Subject to

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

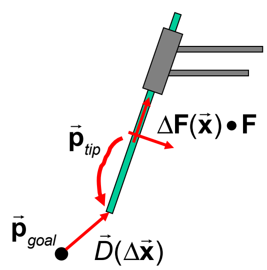
$$\vec{\varepsilon} = \mathbf{J}_{\vec{\varepsilon}}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$

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

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

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



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

Subject to

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

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

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

$$\left\|\vec{D}(\Delta\vec{\mathbf{x}})\right\| \le 3$$

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

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

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

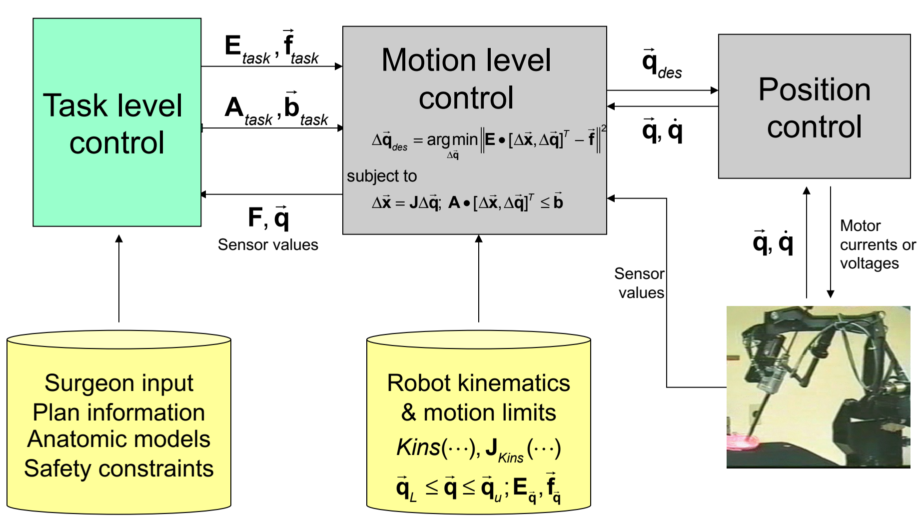
$$\Delta\vec{\mathbf{q}}_{des} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min} \left\| \mathbf{E}_{task}\Delta\vec{\mathbf{x}} - \vec{\mathbf{f}}_{task} \right\|^2 + \left\| \mathbf{E}_{\vec{\mathbf{q}}}\Delta\vec{\mathbf{x}} - \vec{\mathbf{f}}_{\vec{\mathbf{q}}} \right\|^2$$

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

Linear least squares implementation



Typical Mid-Level Control Loop

Step 1. Read robot state ($\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}, etc.$) and sensor values.

Step 2. Perform safety checks and stop motion if needed.

Step 3. Output $\Delta\vec{\mathbf{q}} / \Delta T$ values computed on previous clock tick (done now so commands to low level controller come at predictable times).

Step 4. Interpret any commands from task level controller.

Step 5. Compute $\mathbf{F}_{kins}(\vec{\mathbf{q}})$ and $\mathbf{J}_{kins}(\vec{\mathbf{q}})$.

Step 6. Based on the current commanded behavior, robot state & kinematics, sensor values, and registered anatomic model, compute the desired incremental joint motion $\Delta\vec{\mathbf{q}}$ for the next clock tick. Typically, this will involve formulating and solving an optimization problem, e.g.,

$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\operatorname{argmin}} \left\| \mathbf{A}_{\xi}\vec{\xi} - \vec{\mathbf{b}}_{\xi} \right\|^2 + \left\| \mathbf{A}_{\Delta\vec{\mathbf{q}}}\Delta\vec{\mathbf{q}} - \vec{\mathbf{b}}_{\Delta\vec{\mathbf{q}}} \right\|^2$$

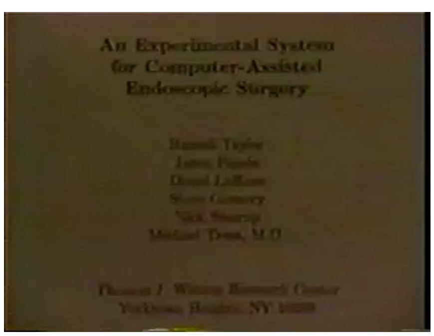
$$\vec{\xi} = \mathbf{J}_{kins}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$$

$$\mathbf{C}_{\xi}\vec{\xi} \le \vec{\mathbf{d}}_{\xi}$$

$$\mathbf{C}_{\Delta\vec{\mathbf{q}}}\Delta\vec{\mathbf{q}} \le \vec{\mathbf{d}}_{\Delta\vec{\mathbf{q}}}$$

Step 7. Go to sleep until the next clock tick.

Some IBM Movies

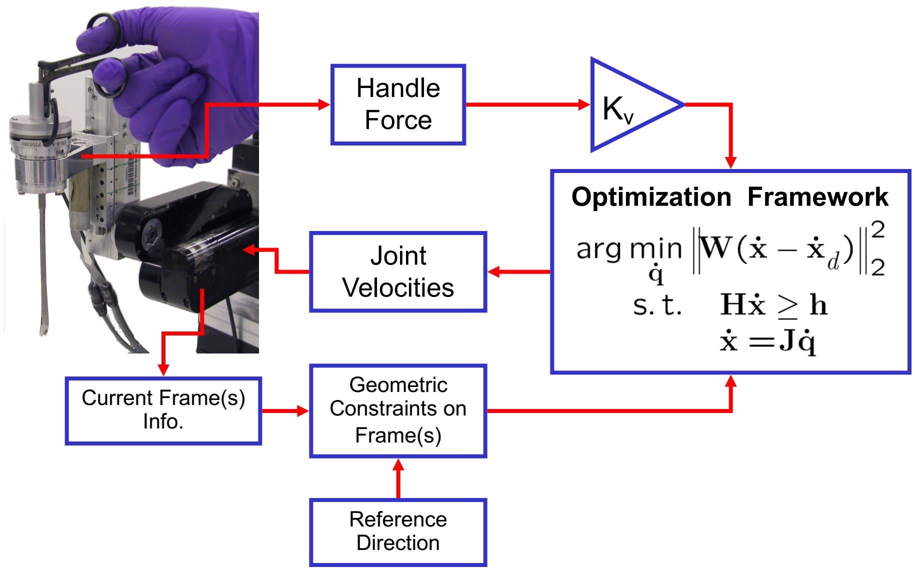


Early Constrained Motion System (LapSYS)

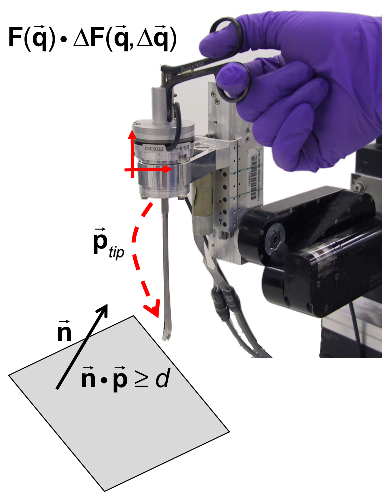
Vision-guided targeting

Steady Hand Robot

High Level Constrained Control



Example: Hands-on Guiding with Forbidden Half Space



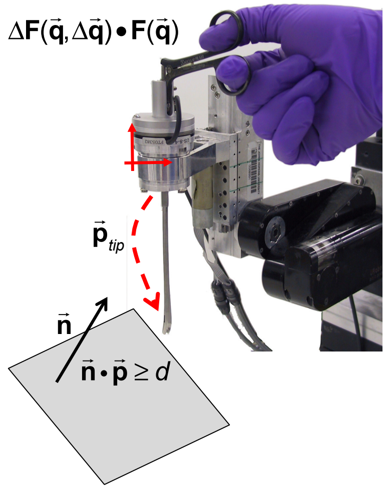
$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min} \left\| \mathbf{K}_v \vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}} \right\|$$

Such that

$$d \leq \vec{\mathbf{n}} \bullet (\mathbf{F}(\vec{\mathbf{q}})\Delta\mathbf{F}_{rhs}(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}}) \bullet \vec{\mathbf{p}}_{tip})$$

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



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min}\left\|\mathbf{K}_v\vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}\right\|$$

Such that

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

$$d \leq \vec{\mathbf{n}}\bullet\left(\mathbf{F}(\vec{\mathbf{q}})\bullet(\vec{\alpha}\times\vec{\mathbf{p}}_{tip}+\vec{\varepsilon}+\vec{\mathbf{p}}_{tip})\right)$$

I.e.,

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

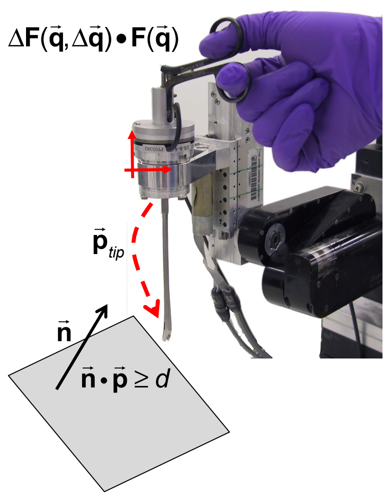
$$d \leq \vec{\mathbf{n}}\bullet\left(\mathbf{R}(\vec{\mathbf{q}})\bullet(\vec{\alpha}\times\vec{\mathbf{p}}_{tip}+\vec{\varepsilon}+\vec{\mathbf{p}}_{tip})+\vec{\mathbf{p}}_{kins}\right)$$

52 601.455/655 Fall 2021

Engineering Research Center for Computer Integrated Surgical Systems and Technology

52

Example: Hands-on Guiding with Forbidden Half Space



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min}\left\|\mathbf{K}_v\vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}\right\|^2$$

Such that

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

$$d - \vec{\mathbf{n}}\bullet\vec{\mathbf{x}} \leq \vec{\mathbf{n}}\bullet\left(\mathbf{R}(\vec{\mathbf{q}})\bullet(\vec{\alpha}\times\vec{\mathbf{p}}_{tip}+\vec{\varepsilon})\right)$$

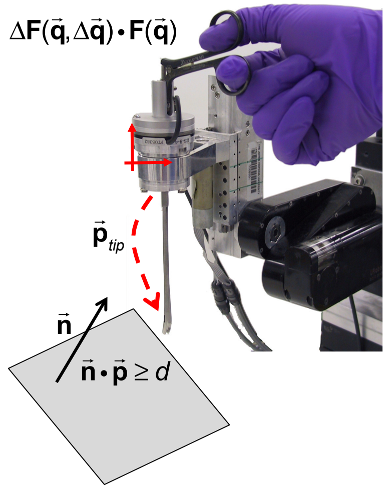
$$\vec{\mathbf{x}} = \mathbf{F}(\vec{\mathbf{q}})\vec{\mathbf{p}}_{tip}$$

53 601.455/655 Fall 2021

Engineering Research Center for Computer Integrated Surgical Systems and Technology

53

Example: Hands-on Guiding with Forbidden Half Space



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min}\left\|\mathbf{K}_v\vec{\mathbf{f}} - \mathbf{J}_{kins}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}\right\|^2$$

Such that

$$d \leq \vec{\mathbf{n}}\bullet(\Delta\mathbf{F}(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\bullet\mathbf{F}(\vec{\mathbf{q}})\bullet\vec{\mathbf{p}}_{tip})$$

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 Jacobeans

$$\Delta\mathbf{F}\left(\mathbf{J}_{kins}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}\right)\bullet\mathbf{F}(\vec{\mathbf{q}}) = \mathbf{F}(\vec{\mathbf{q}})\bullet\Delta\mathbf{F}(\mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}})$$
$$\Delta\mathbf{R}\left(\mathbf{J}_{kins}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}\right)\bullet\mathbf{R}(\vec{\mathbf{q}}) = \mathbf{R}(\vec{\mathbf{q}})\bullet\Delta\mathbf{R}(\mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}})$$

Define

$$\mathbf{J}_{kins} = \begin{bmatrix}\mathbf{J}^{\alpha}_{kins}\\ \mathbf{J}^{\varepsilon}_{kins}\end{bmatrix} \quad \vec{\alpha}_{kins} = \mathbf{J}^{\alpha}_{kins}\Delta\vec{\mathbf{q}} \quad \mathbf{J}_{rhs} = \begin{bmatrix}\mathbf{J}^{\alpha}_{rhs}\\ \mathbf{J}^{\varepsilon}_{rhs}\end{bmatrix} \quad \vec{\alpha}_{rhs} = \mathbf{J}^{\alpha}_{rhs}\Delta\vec{\mathbf{q}}$$

so

$$\Delta\mathbf{R}(\mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}) = \mathbf{R}(\vec{\mathbf{q}})^{-1}\Delta\mathbf{R}\left(\mathbf{J}_{kins}\Delta\vec{\mathbf{q}}\right)\bullet\mathbf{R}(\vec{\mathbf{q}})$$
$$\mathbf{I} + sk(\alpha_{rhs}) = \mathbf{I} + \mathbf{R}^{-1}sk(\vec{\alpha}_{kins})\mathbf{R}$$
$$sk(\vec{\alpha}_{rhs}) = sk(\mathbf{R}^{-1}\vec{\alpha}_{kins})$$
$$\mathbf{J}^{\alpha}_{rhs} = \mathbf{R}^{-1}\mathbf{J}^{\alpha}_{kins}$$

and one can do something similar for the $\Delta\vec{\mathbf{p}}$ parts.

LHS versus RHS Jacobeans

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

Define

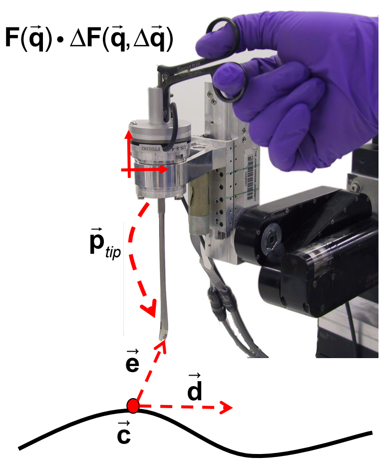
$$\mathbf{J}_{kins} = \begin{bmatrix} \mathbf{J}^{\alpha}_{kins} \\ \mathbf{J}^{\varepsilon}_{kins} \end{bmatrix} \quad \vec{\alpha}_{kins} = \mathbf{J}^{\alpha}_{kins}\Delta\vec{\mathbf{q}} \quad \mathbf{J}_{rhs} = \begin{bmatrix} \mathbf{J}^{\alpha}_{rhs} \\ \mathbf{J}^{\varepsilon}_{rhs} \end{bmatrix} \quad \vec{\alpha}_{rhs} = \mathbf{J}^{\alpha}_{rhs}\Delta\vec{\mathbf{q}}$$

so

$$\begin{aligned}
\mathbf{R}(\vec{\mathbf{q}})\vec{\varepsilon}_{rhs} + \vec{\mathbf{p}}(\vec{\mathbf{q}}) &= \Delta\mathbf{R}(\vec{\alpha}_{kins})\vec{\mathbf{p}}(\vec{\mathbf{q}}) + \vec{\varepsilon}_{kins} \\
\mathbf{R}(\vec{\mathbf{q}})\vec{\varepsilon}_{rhs} &= \vec{\mathbf{p}}(\vec{\mathbf{q}}) + \vec{\alpha}_{kins}\times\vec{\mathbf{p}}(\vec{\mathbf{q}}) + \vec{\varepsilon}_{kins} - \vec{\mathbf{p}}(\vec{\mathbf{q}}) \\
\vec{\varepsilon}_{rhs} &= \mathbf{R}(\vec{\mathbf{q}})^{-1}\left(\vec{\alpha}_{kins}\times\vec{\mathbf{p}}(\vec{\mathbf{q}}) + \vec{\varepsilon}_{kins}\right) \\
&= \mathbf{R}(\vec{\mathbf{q}})^{-1}\left(\vec{\varepsilon}_{kins} - sk\left(\vec{\mathbf{p}}(\vec{\mathbf{q}})\right)\vec{\alpha}_{kins}\right) \\
&= \mathbf{R}(\vec{\mathbf{q}})^{-1}\vec{\varepsilon}_{kins} - \mathbf{R}(\vec{\mathbf{q}})^{-1}sk\left(\vec{\mathbf{p}}(\vec{\mathbf{q}})\right)\vec{\alpha}_{kins}
\end{aligned}$$

$$\mathbf{J}_{rhs} = \begin{bmatrix} \mathbf{R}^{-1} & \mathbf{0} \\ -\mathbf{R}^{-1}sk(\vec{\mathbf{p}}) & \mathbf{R}^{-1} \end{bmatrix}\mathbf{J}_{kins}$$

Example: Hands-on Guiding to Follow a Path



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min}\left\|\mathbf{K}_v\vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}\right\|$$

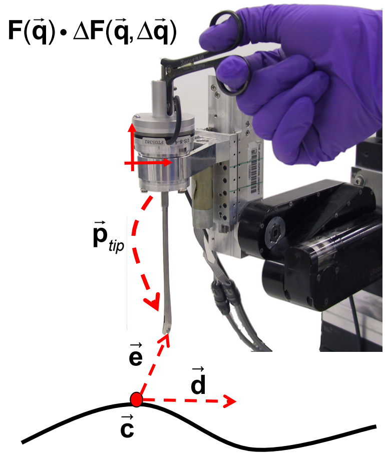
Such that

$$\vec{\mathbf{e}} = (\mathbf{F}(\vec{\mathbf{q}})\Delta\mathbf{F}_{rhs}(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\bullet\vec{\mathbf{p}}_{tip}) - \vec{\mathbf{c}}$$

$$\delta \geq \left\|\vec{\mathbf{e}} - \left(\vec{\mathbf{d}}\bullet\vec{\mathbf{e}}\right)\vec{\mathbf{d}}\right\|$$

Note: δ is the maximum deviation allowed from the path

Example: Hands-on Guiding to Follow a Path



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min} \left\| \mathbf{K}_v \vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}} \right\|^2$$

Such that

$$\vec{\mathbf{e}} = (\mathbf{F}(\vec{\mathbf{q}})\left(\vec{\alpha} \times \vec{\mathbf{p}}_{tip} + \vec{\varepsilon} + \vec{\mathbf{p}}_{tip}\right)) - \vec{\mathbf{c}}$$

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

$$\delta \geq \left\| \vec{\mathbf{e}} - \left(\vec{\mathbf{d}} \bullet \vec{\mathbf{e}}\right)\vec{\mathbf{d}} \right\|$$

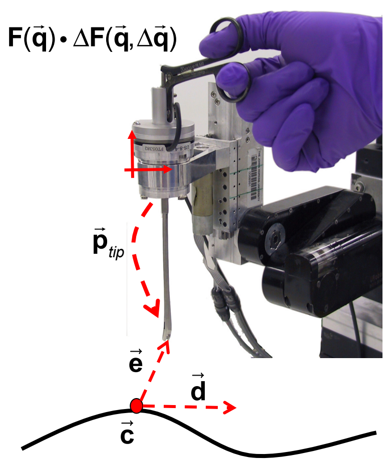
Note: δ is the maximum deviation allowed from the path

58 601.455/655 Fall 2021

Engineering Research Center for Computer Integrated Surgical Systems and Technology

58

Example: Hands-on Guiding to Follow a Path



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min} \left\| \mathbf{K}_v \vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}} \right\|^2$$

Such that

$$\vec{\mathbf{e}} = \mathbf{R}(\vec{\mathbf{q}})\left(\vec{\alpha} \times \vec{\mathbf{p}}_{tip} + \vec{\varepsilon} + \vec{\mathbf{p}}_{tip}\right) + \vec{\mathbf{p}}(\vec{\mathbf{q}}) - \vec{\mathbf{c}}$$

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

$$\delta \geq \left\| \vec{\mathbf{e}} - \left(\vec{\mathbf{d}} \bullet \vec{\mathbf{e}}\right)\vec{\mathbf{d}} \right\|$$

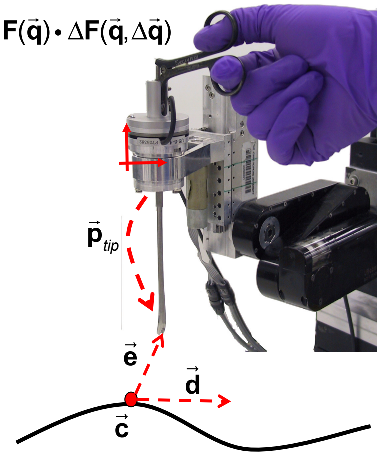
Note: δ is the maximum deviation allowed from the path

59 601.455/655 Fall 2021

Engineering Research Center for Computer Integrated Surgical Systems and Technology

59

Example: Hands-on Guiding to Follow a Path



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min} \left\| \mathbf{K}_v \vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}} \right\|^2$$

Such that

$$\vec{\mathbf{e}} = \vec{\mathbf{p}}(\vec{\mathbf{q}}) + \mathbf{R}(\vec{\mathbf{q}})\vec{\mathbf{p}}_{tip} + \mathbf{R}(\vec{\mathbf{q}})\vec{\varepsilon} - \mathbf{R}(\vec{\mathbf{q}})sk(\vec{\mathbf{p}}_{tip})\vec{\alpha}$$

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

$$\delta \geq \left\| \vec{\mathbf{e}} - \left(\vec{\mathbf{d}} \bullet \vec{\mathbf{e}}\right)\vec{\mathbf{d}} \right\|$$

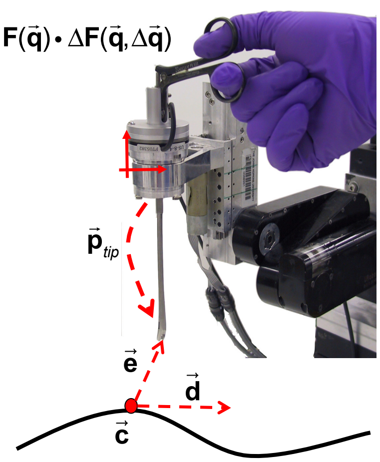
Approximate this by

$$-\delta \leq \left(\vec{\mathbf{e}} - \left(\vec{\mathbf{d}} \bullet \vec{\mathbf{e}}\right)\vec{\mathbf{d}}\right) \bullet \left(Rot(\vec{\mathbf{d}}, k\pi / N)\vec{\mathbf{g}}\right) \leq \delta$$

for $0 \leq k \leq N-1$

and some $\vec{\mathbf{g}}$ perpendicular to $\vec{\mathbf{d}}$

Example: Hands-on Guiding to Follow a Path



$$\Delta\vec{\mathbf{q}} = \underset{\Delta\vec{\mathbf{q}}}{\arg\min} \left\| \mathbf{K}_v \vec{\mathbf{f}} - \mathbf{J}_{rhs}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}} \right\|^2 + \eta \left\| \vec{\mathbf{e}} - \left(\vec{\mathbf{d}} \bullet \vec{\mathbf{e}}\right)\vec{\mathbf{d}} \right\|^2$$

Such that

$$\vec{\mathbf{e}} = \vec{\mathbf{p}}(\vec{\mathbf{q}}) + \mathbf{R}(\vec{\mathbf{q}})\vec{\mathbf{p}}_{tip} + \mathbf{R}(\vec{\mathbf{q}})\vec{\varepsilon} - \mathbf{R}(\vec{\mathbf{q}})sk(\vec{\mathbf{p}}_{tip})\vec{\alpha}$$

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

$$\delta \geq \left\| \vec{\mathbf{e}} - \left(\vec{\mathbf{d}} \bullet \vec{\mathbf{e}}\right)\vec{\mathbf{d}} \right\|$$

Approximate this by

$$-\delta \leq \left(\vec{\mathbf{e}} - \left(\vec{\mathbf{d}} \bullet \vec{\mathbf{e}}\right)\vec{\mathbf{d}}\right) \bullet \left(Rot(\vec{\mathbf{d}}, k\pi / N)\vec{\mathbf{g}}\right) \leq \delta$$

for $0 \leq k \leq N-1$

and some $\vec{\mathbf{g}}$ perpendicular to $\vec{\mathbf{d}}$