Automating Treatment Planning in Radiation Therapy

Olalekan Ogunmolu

Perelman School of Medicine

University of Pennsylvania, Philadelphia, PA

Presented by **Lekan Molu** (`Lay-con Moh-lu`)

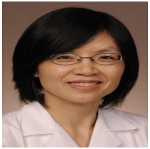
March 10, 2021

Acknowledgments

Steve Jiang, UTSW

Nick Gans, UTARI

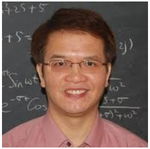
Xuejun Gu, UTSW



Dan Nguyen, UTSW

Rod Wiersma, Penn

Xinmin Liu, Penn



Tyler Summers, UTD

Yonas Tadesse, UTD

Funding Sources

NIH NATIONAL CANCER INSTITUTE

CANCER PREVENTION & RESEARCH INSTITUTE OF TEXAS

Talk Outline

- Beam Orientation Optimization (BOO)
  - Monte Carlo Tree Search and Neuro-Dynamic Programming for BOO
  - Column Generation as Pretraining for MCTS for BOO
- Patient Head Motion Correction in External Beam Radiation Therapy (RT)
  - Intensity-Modulated RT (IMRT): Earlier PhD Work
  - Magnetic Resonance Imaging and Linear Accelerator Systems (MRI-LINACs)
- Robustness Margins and Robust Deep Policies for Nonlinear Control

Research Significance

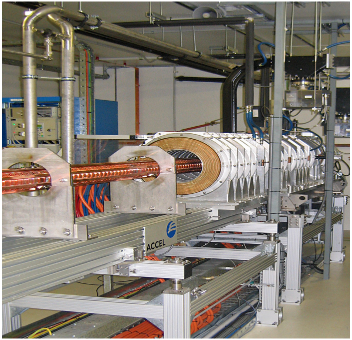
2021 → 1,898,160 U.S. cancer cases[1] → 608,570 deaths → $246 billion in costs by 2030[1]

Cancer Treatment
- Chemotherapy/Radiosurgery.
- Radiation Therapy (RT).

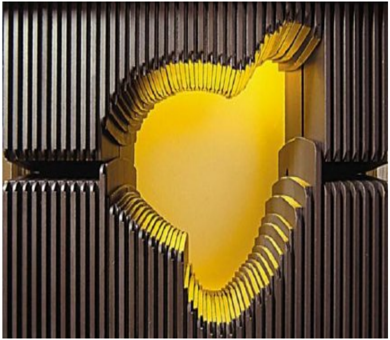
RT Signif.
- Approx. 40% of all curative cancer treatment modality are performed with RT[1].
- Cost-effective -- accounting for only 5% of the total cost of cancer care[3].

Imaging
- CT images in RT = Low soft tissues contrast ⇒ OARs toxicity during treatment planning.
- MRI of brain, and head and neck (H&N) cancers provides exceptional soft tissues contrast and allows greater delineation of cancerous from non-cancerous tissues.

IMRT Treatment Planning (Beam Delivery)

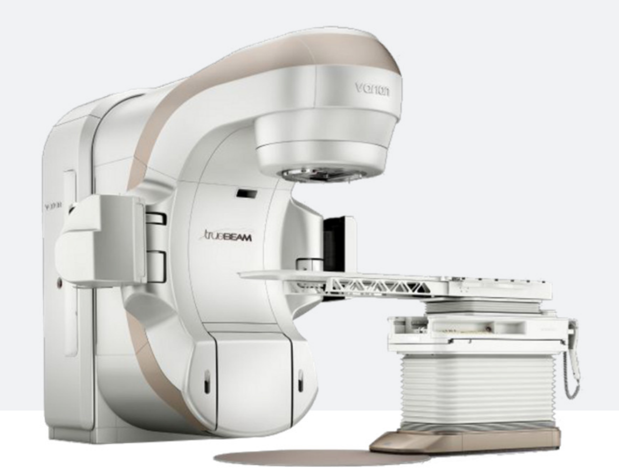


The Australian Synchrotron.



A Multi-leaf collimator, ©Varian.

Radiation Delivery Couch and Gantry



Varian's TrueBeam Radiotherapy System.

Part I.A: Beam Orientation Optimization (BOO)

- Beam Orientation Optimization (BOO)
  - Monte Carlo Tree Search and Neuro-Dynamic Programming

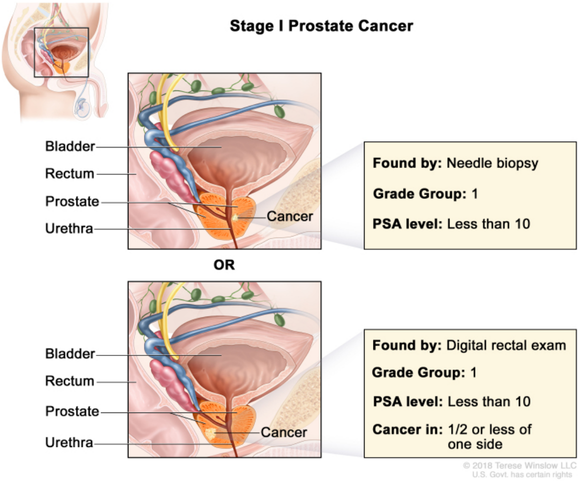
Contributions

Relevant Publications

Ogunmolu, Olalekan, Michael Folkerts, Dan Nguyen, Nicholas Gans, and Steve Jiang. "Deep BOO! Automating Beam Orientation Optimization in Radiation Therapy." In *Algorithm Foundations of Robotics XIII*, Merida, Mexico. Published in *Springer's Proceedings in Advanced Robotics (SPAR) Book*, 2020.

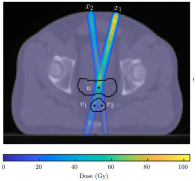
- A sparse tree lookout strategy for games with large state spaces guides transition between beam angle sets
- Tree lookout strategy guided by a deep neural network policy

Protate Cancer Example

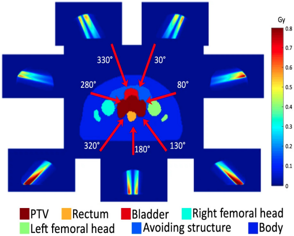


BOO Process: Fluence Map Optimization

Prostate CT slice



Prostate before BOO



Fluence Map

BOO Workflow

Manually Selection/Protocols Adoption

Laborious process; could take up to 5 days for head and neck cancer treatment.

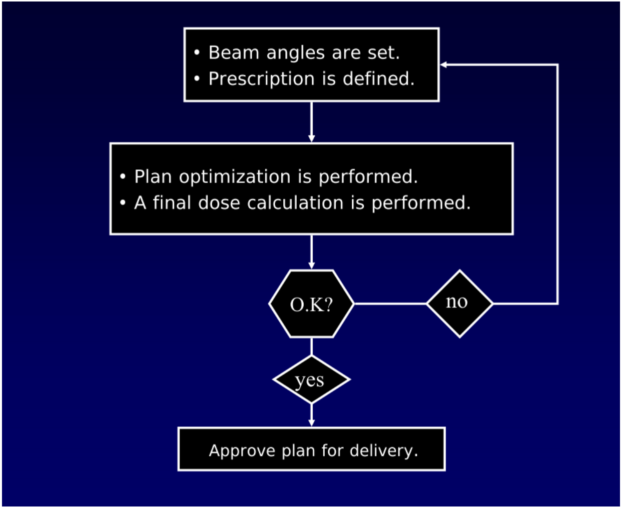
Pre-solve Large Sparse Dose Influence Matrix

Takes hours to solve for a single patient. Days/months for multiple patients.

Solve Fluence Map Optimization

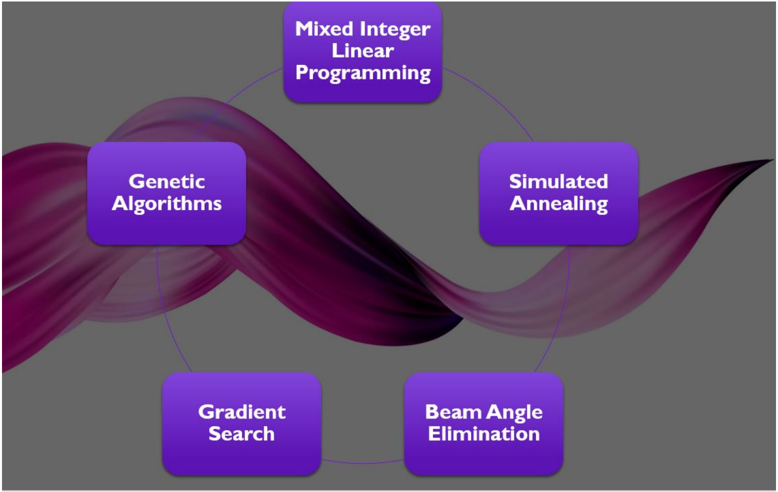
Time-consuming: Often takes minutes.

Treatment Plan Flowchart



Reprinted from "IMRT Optimization Algorithms. David Shepard. Swedish Cancer Institute. AAPM 2007."

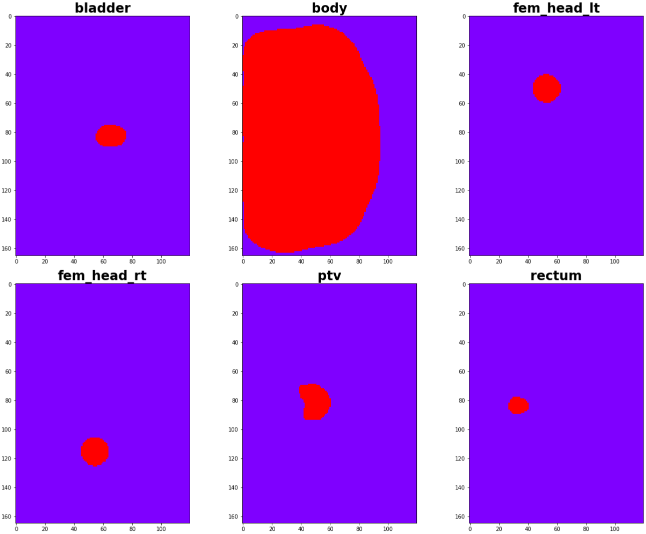
Current Approaches and Limitations



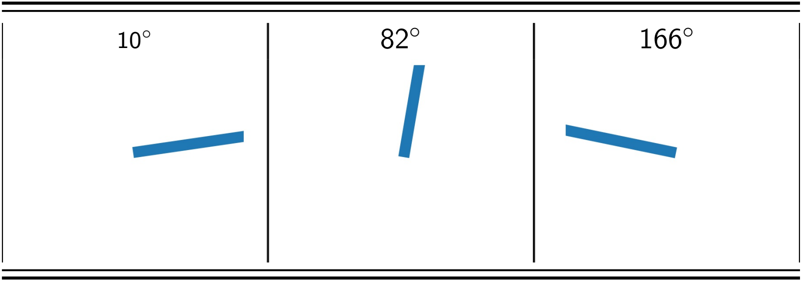
Innovation

- A Tower Neural Network generates a policy that guides MCTS simulations for two players in a zero-sum Markov game
  - Produces a *utility (value) function* & a subjective *probability distribution*
- Each player in a two-player Markov game finds an alternating best response to the current player's average strategy
  - driving the neural network policy's weights toward an approximate **saddle equilibrium** [Heinrich et al. (2015)].
  - aids network in finding an *approximately optimal* beam angle candidate set that meets a dosimetric requirements.

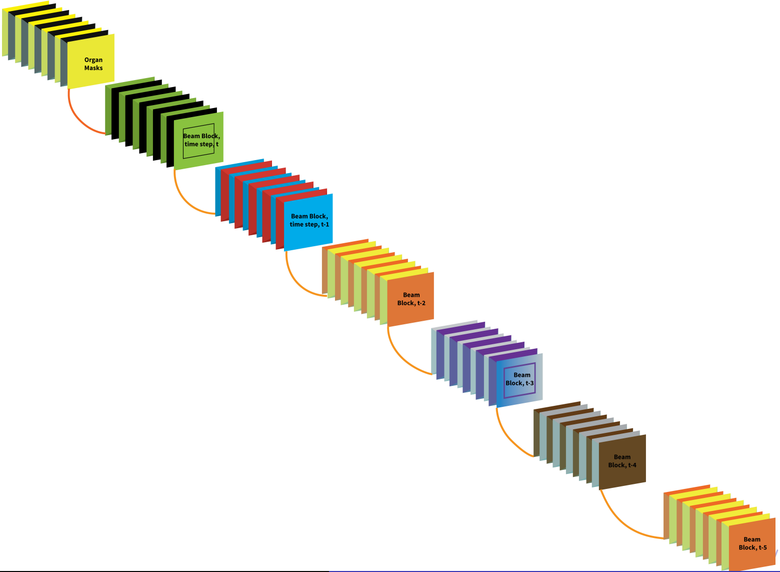
State Encoding: Prostate Organ Masks



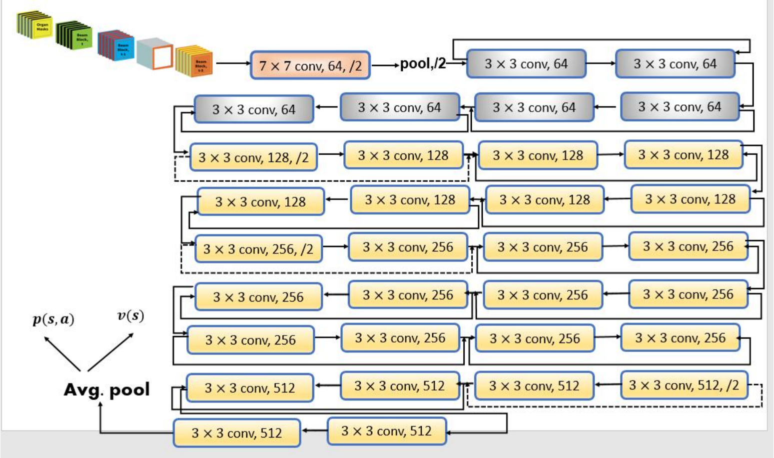
State Representation: Beam Angles



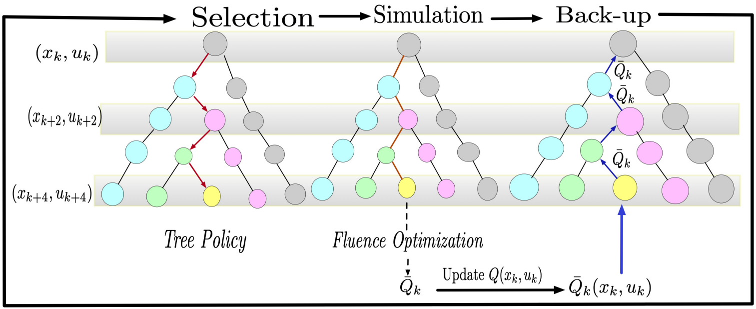
State Representation



Two-player Fictitious network play with ResNet



Tree Representation and Game Simulation



Tree Composition

Every **node** of the tree, $\boldsymbol{x}$, has the following fields:

- a pointer to the parent that led to it, $\boldsymbol{x}.p$;
- the beamlets, $\boldsymbol{x}_b$, stored at that node; $b = \{1, \ldots, m\}$;
- a set of move probabilities prior, $p(s, a)$;
- a pointer $\boldsymbol{x}.\boldsymbol{r}$, to the reward r_t, for the state $\boldsymbol{x}_t$;
- a pointer to the state-action value $Q(s, a)$ and its upper confidence bound $U(s, a)$;
- a visit count $N(s, a)$, that indicates the number of times that node was visited; and
- a pointer $\boldsymbol{x}.child_i$ to each of its children nodes.

Saddle Point Strategy Formulation

- **Saddle point strategies** for optimal control sequence pair $\{a_t^{p_1^*}, a_t^{p_2^*}\}$ recursively obtained by optimizing, $V_t(s, a)$

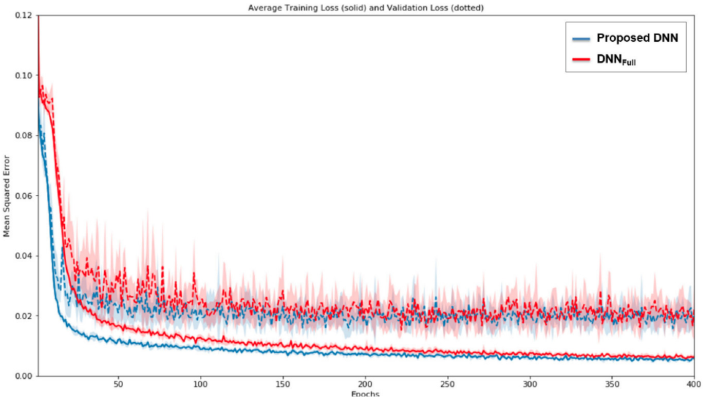
$$V_t^*(s) = Q_t^*(s_t, \pi_t^{p_1}, \pi_t^{p_2}) = \min_{\pi^{p_1} \in \prod^{p_1}} \max_{\pi^{p_2} \in \prod^{p_2}} Q_t^\star(s_t, \pi^{p_1}, \pi^{p_2})$$
$$\forall s_t \in \mathcal{S}, \pi^{p_1} \in \prod^{p_1}, \pi^{p_2} \in \prod^{p_2}.$$

such that

$$v_{p_1}^\star \leq v^\star \leq v_{p_2}^\star \quad \forall \{\pi_t^{p_1}, \pi_t^{p_2}\}_{0 \leq t \leq T}.$$

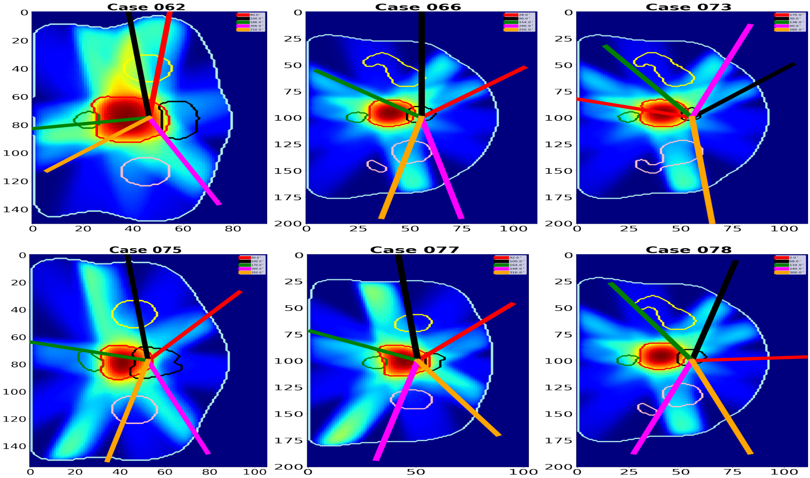
- p_1, p_2 respectively generating a **mixed strategy** via **averaging the outcome** of individual plays.

Training and Validation Loss

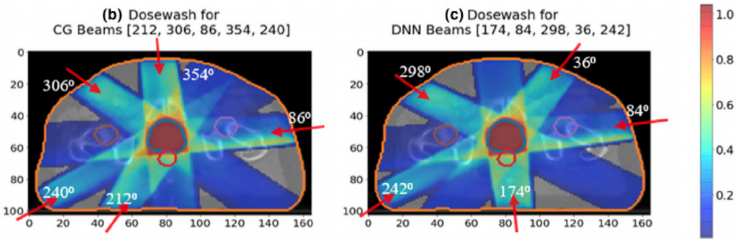


Average training (solid) and validation (dotted) loss function (MSE) values across six cross-validation folds for the network (blue) and full network.

BOO Results: Testing of self-play network



Column Generation vs Neural Network



Dose-Volume Histogram of CG vs DNN architectures [Sadeghnejad Barkousaraie, Azar and Ogunmolu, Olalekan and Jiang, Steve and Nguyen, Dan (2019)].

Conclusions

- Deep Neural Network optimizes network weights in a separate multiprocessing thread; Network outputs probabilities used to guide search;
- Sparse lookahead search builds tree with nodes labeled by state-action pairs in an alternating manner; sample rewards stored on edges connecting state-action with state nodes;
- Beam angles prediction takes between 2-3 minutes with MCTS vs. ~ 60 seconds with Column Generation Pre-training.

Head Stabilization in Radiation Therapy (RT)

- Head Stabilization in Cancer Radiation Therapy
  - Intensity-Modulated RT (IMRT)

Robotic Radiosurgery

UNIVERSITY *of* PENNSYLVANIA

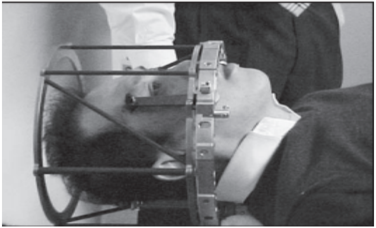
A Patient Head Motion-Correction Mechanism for MRI-LINAC RT

OLALEKAN OGUNMOLU

DEPARTMENT OF RADIATION ONCOLOGY, PENN SCHOOL OF MEDICINE

- Current Collaborators: Rodney Wiersma & Xinmin Liu (UChicago → UPenn)
- Past Collaborators: Steve Jiang, Xuejun Gu, (UT Southwestern); Nick Gans (UT Dallas, UT Arlington)

Correcting Head Motion: RT and MRI-LINACs

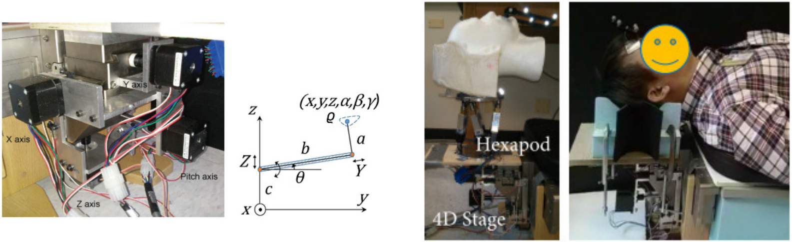


(a) The BRW SRS Frame [Chelvarajah et al. (2004)]

(b) Thermoplastic masks

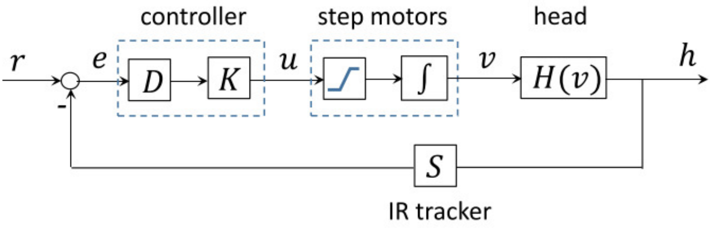
(c) Frame With MRI Coils (PSOM)

4-D Motion Correction Stage



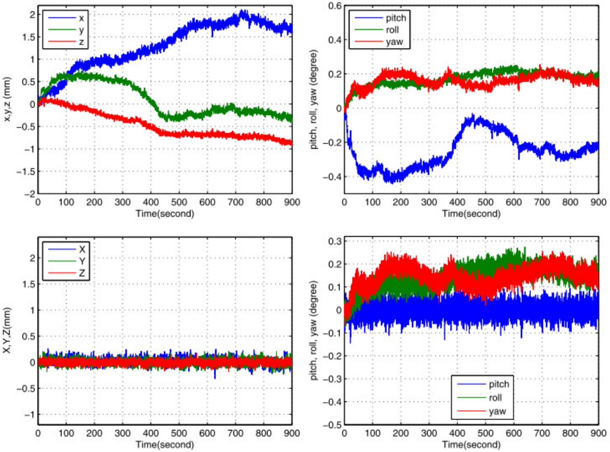
Liu et al. (2015)

4-DOF Motion Controller Block Diagram



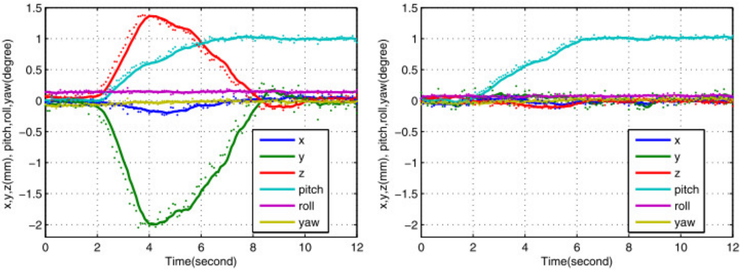
Liu et al. (2015)

Phantom Feedback Motion Correction Results



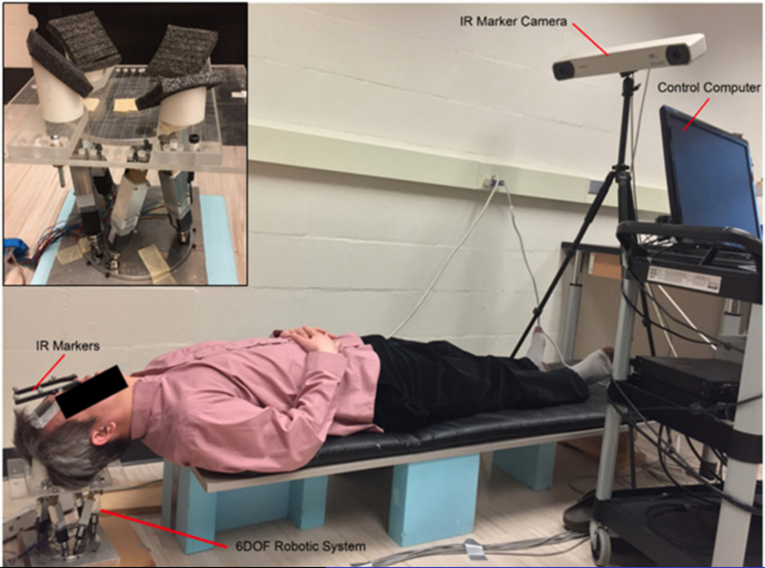
Time response of feedback control without (left) and with (right) decoupling control [Liu et al. (2015)].

Human Volunteer Feedback Motion Correction Results

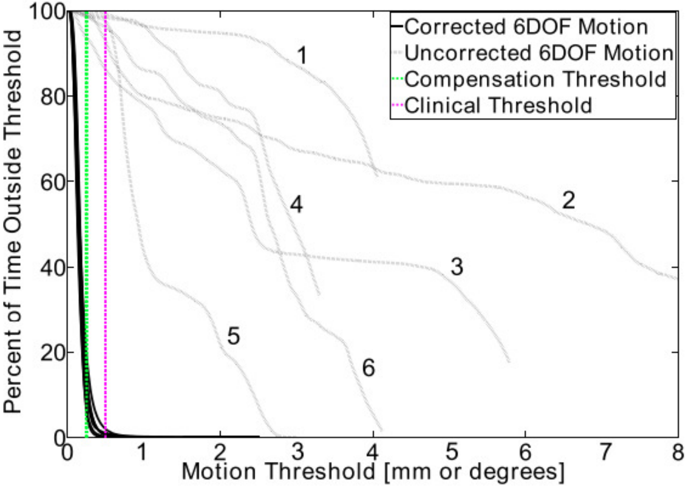


Head Motion Without and With Motion Correction. Left: Coupled Axes; Right: Decoupled Axes.

SRS: Wiersma Stewart-Gough Platform



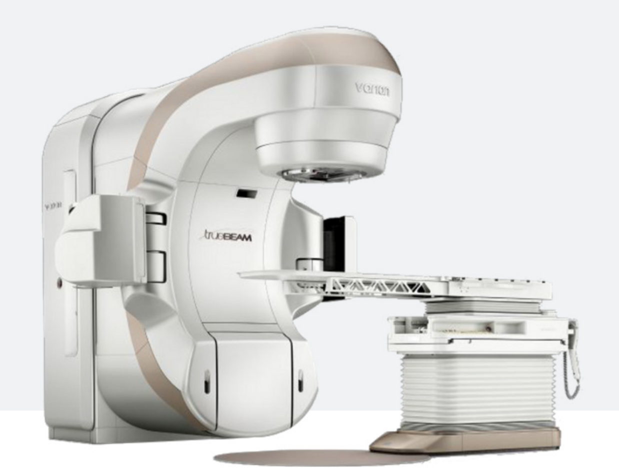
6-DOF Motion Correction Results



Drawbacks of current solutions

- Rigid patient's body assumption
- Non-compliant immobilization devices
- Invasiveness during radiosurgery/RT
- Attenuation of photon beams

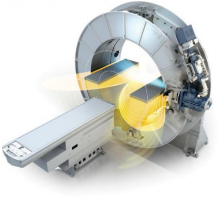
Radiation Delivery Couch and Gantry



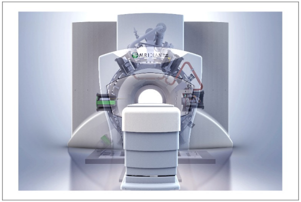
Varian's TrueBeam Radiotherapy System.

Next-Gen RT Treatment with MRI-LINACs

Elekta AB's (Sweden)

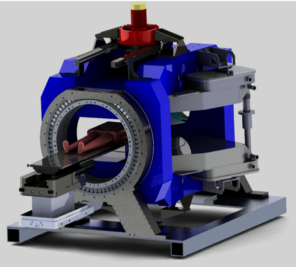


ViewRay's MRIdian



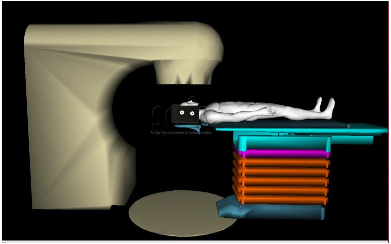
Elekta AB's (Sweden)

MagnetΓx (Canada) Aurora RT

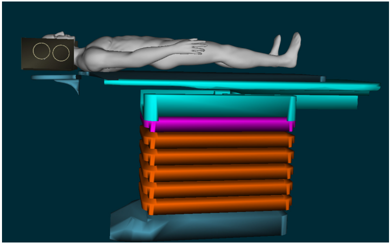


Soft Robots for Head Motion Compensation

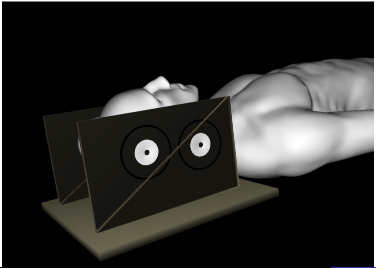
IMRT Setup



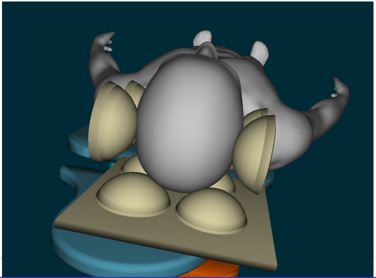
IMRT/MRI



Head and Robot Panel



Head-Robot Closeup



Morphing in Cephalopods

Morph Stage 1

Morph Stage 2

Morph Stage 3

Morph Stage 4

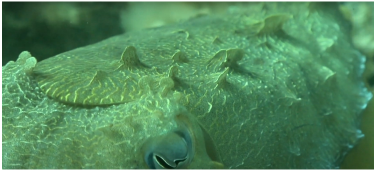
Morph Stage 5

Morph Reversal

©Roger Hanlon, YouTube.

Cephalopods Neural-Controlled Physical Texture

Raises Periscope



Papillae

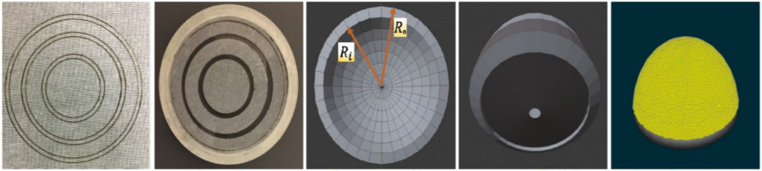
Papillae Up

Papillae $\times$ Papillae

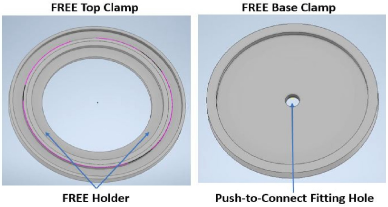
Cuttlefish' Morphin ©Roger Hanlon, YouTube.

Cephalopods-inspired Actuator Design

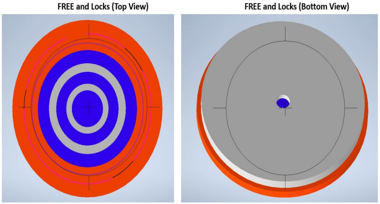
Circumferentially Constrained And Radially Symmetric Elastomers (CCOARSE).



(a) Laser-Cut Fiber (b) Fiber+Rubber (c) Deformation Model (d) Actuator+PVC Base (e) Finite Element Model



(f,g)

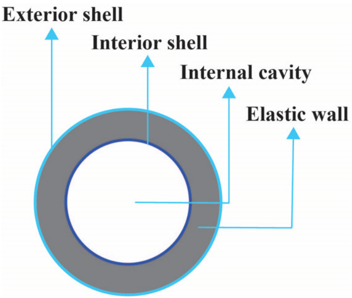


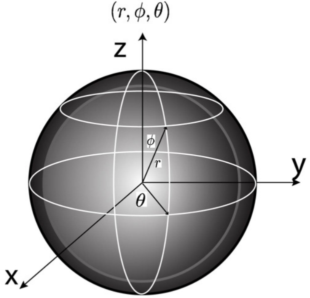
(h)

[Pikul et al. (2019)]

Nonlinear Elastic Deformation Analysis

IAB SHELLS AND AIR CAVITY/DEFORMATION ANALYSIS





$$r_i \le r \le r_o, \qquad 0 \le \theta \le 2\pi, \qquad 0 \le \phi \le \pi$$

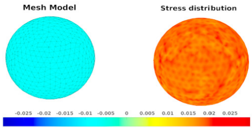
Soft IK via Boundary Value Problem

- With Cauchy's laws of motion, solve boundary volume problem of traction, and find that

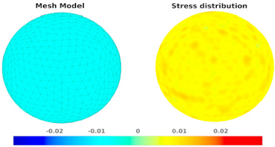
$$P(r) = \int_{r_i}^{r_0} \left[2C_1 \left(\frac{r}{R^2} - \frac{R^4}{r^5} \right) + 2C_2 \left(\frac{r^3}{R^4} - \frac{R^2}{r^3} \right) \right] dr \tag{1}$$

- *i.e.* Given a prescribed radius, find pressure to deform actuator between configurations

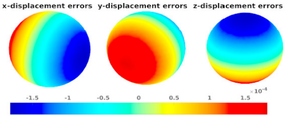
Volumetric Deformation Results (Simulation)



(*a*) **Left**: Mesh model. **Right**: Stress distribution on outer skin.



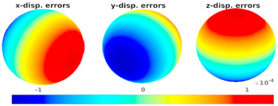
(*a*) **Left**: Mesh model. **Right**: Stress distribution on outer skin.



(*b*) Displacement errors along x, y, z coordinates.

Inputs					Outputs		
C_1	C_2	R_i	r_i	R_o	r_o	P	ΔV
$1.1e4$	$2.2e4$	.027	.03	.03	.033	.76	≈ 0

Fig. 6: Volumetric Deformation (Expansion).

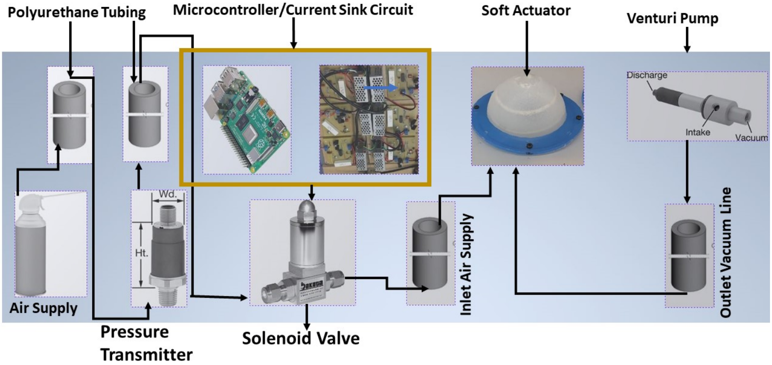


(*b*) Displacement errors along x, y, z coordinates.

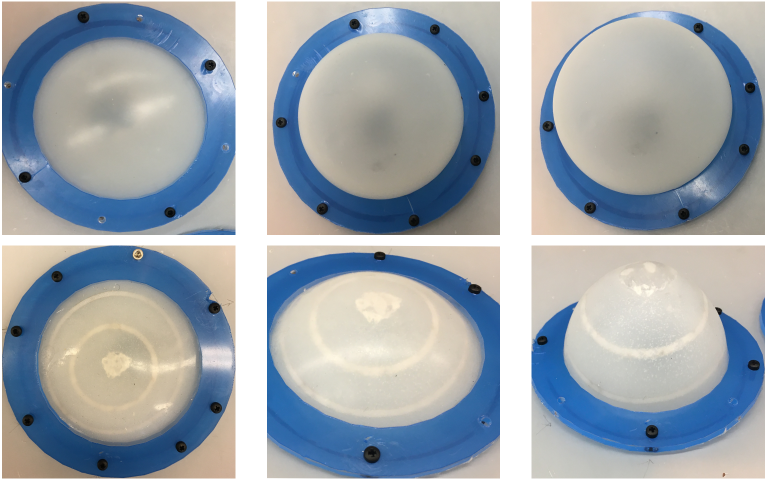
Inputs					Outputs		
C_1	C_2	R_i	r_i	R_o	r_o	P	ΔV
$1.1e4$	$2.2e4$	.025	.03	.03	.028	$-.34$	≈ 0

Fig. 7: Volumetric Deformation (Compression).

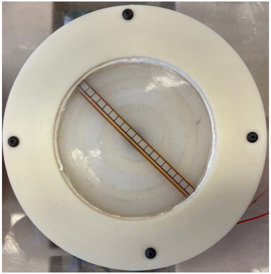
Pneumatic Control and Deformation Scheme



Volumetric Deformation Results (Actual)

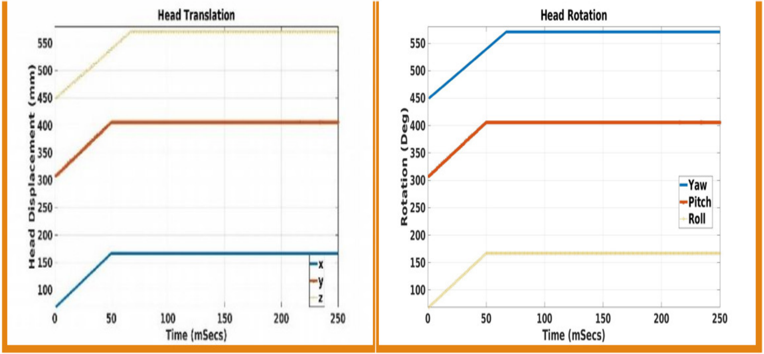


Actuator and Overall Mechanism with Phantom



Head Motion Open Loop Control

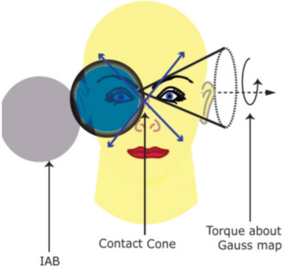
Independent Actuation

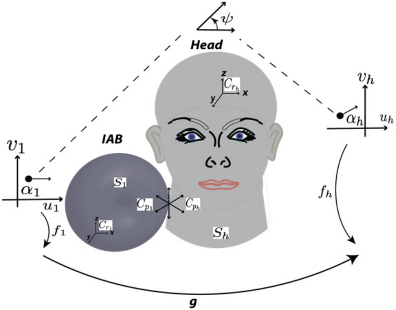


Head Translation along x, y, z for a task of raising the head by a certain threshold above the table	Head rotation in Euler angles for a task of tilting the head about the x, y, z axes on the treatment table.

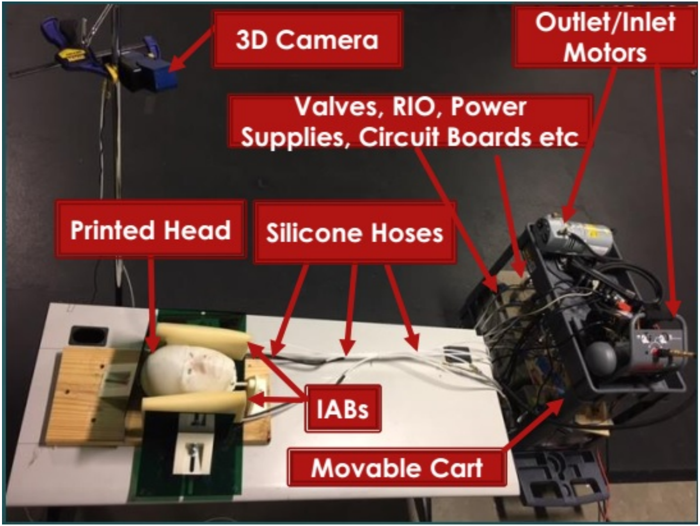
Ongoing Work: 6-DOF Closed-loop Control

Continuum Mechanical Model Validation/Differential Geometry/Newton-Euler Dynamics

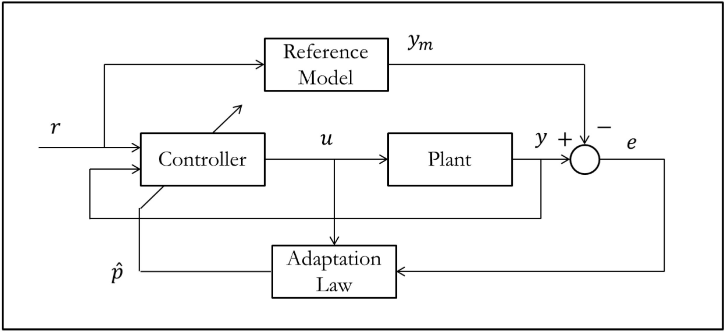




3-DOF Simulation Testbed



Model Reference Adaptive Control



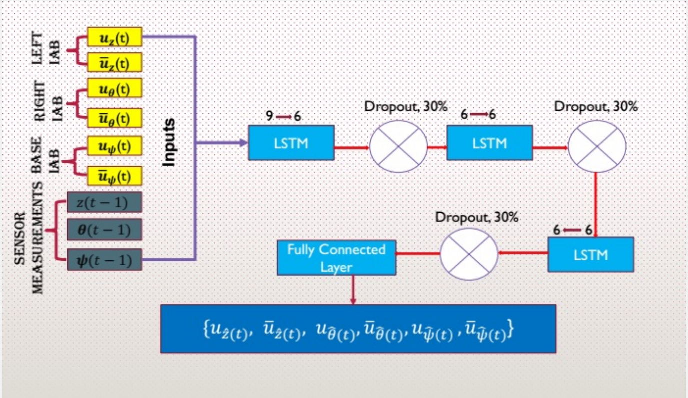
Indirect MRAC system. (Source mdpi.com)

3-DOF Model Reference Adaptive Control

- Model head and bladder dynamics as
  - $\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}\Lambda\left(\mathbf{u} - f(\boldsymbol{x}, \mathbf{u})\right) + \mathbf{w}(k)$
- Approximate $f(\boldsymbol{x}, \mathbf{u})$ by a neural network with continuous memory states
- Derive adaptive adjustment mechanism from Lyapunov analysis for Adaptive Control Parks (1966)
- $\mathbf{u} = \underbrace{\hat{\mathbf{K}}_x^{\mathbf{T}}\boldsymbol{x}}_{\text{state feedback}} + \underbrace{\hat{\mathbf{K}}_r^{\mathbf{T}}\mathbf{r}}_{\text{optimal regulator}} + \underbrace{\hat{f}(\boldsymbol{x}, \mathbf{u})}_{\text{approximator}}$

Neural Network Architecture

Neural Net Architecture

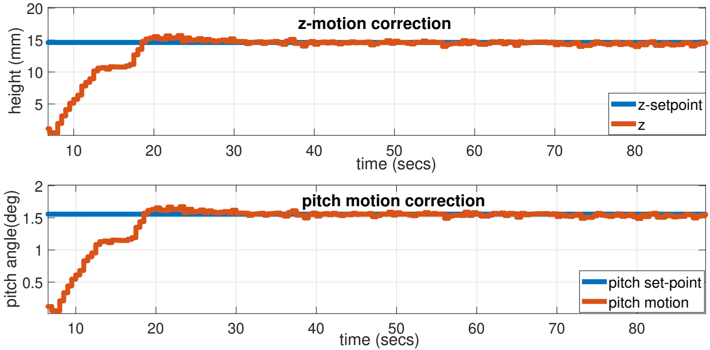


Lyapunov Redesign: Theorem

- For correct adaptive gains, $\hat{\mathbf{K}}_x$ and $\hat{\mathbf{K}}_r$, $\mathbf{e}(k)$ is ***uniformly ultimately bounded***, and the state $\boldsymbol{x}$ converges to a neighborhood of $\mathbf{r}$.
- Choose a $\mathbf{V}$ in terms of $\mathbf{e}$; $\tilde{\mathbf{K}}_x^T$, $\tilde{\mathbf{K}}_r^T$; and parameter error $\varepsilon_f(\boldsymbol{x}(k))$ space

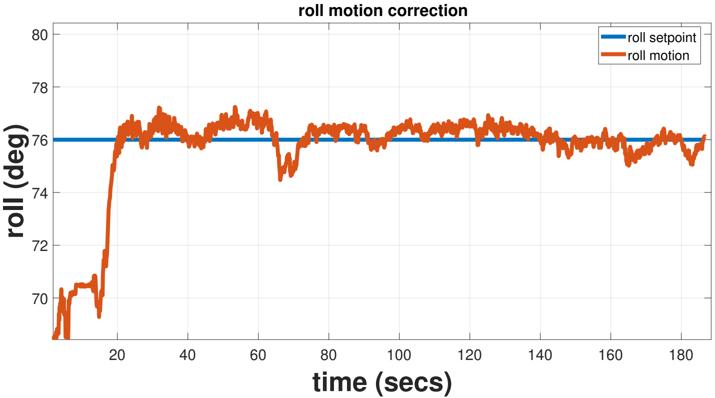
$$\mathbf{V}(\mathbf{e}, \tilde{\mathbf{K}}_x, \tilde{\mathbf{K}}_r^T) = \mathbf{e}^T\mathbf{P}\mathbf{e} + \mathbf{tr}(\tilde{\mathbf{K}}_x^T\boldsymbol{\Gamma}_x^{-1}\tilde{\mathbf{K}}_x^T|\boldsymbol{\Lambda}|) + \mathbf{tr}(\tilde{\mathbf{K}}_r^T\Gamma_r^{-1}\tilde{\mathbf{K}}_r|\boldsymbol{\Lambda}|)$$

Results: Z and Pitch Motions



Goal command: $(z, \theta, \phi) = (14mm, 1.6^\circ, 45^\circ)^T$.

Results: Roll Motion



Head roll tracking.

Conclusions

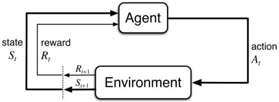
- Non-invasive soft robot for head motion compensation ✓
- Photons-transparent as opposed to rigid/electro-mechanical devices/robots ✓
- Adaptible under MRI coils for newer MRI-LINACs ✓

Part III: Robustness Margins and Robust Deep Policies

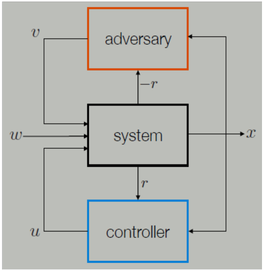
- Robustness Margins and Robust Deep Policies for Nonlinear Control

The robustness conundrum

- How to know *a priori* a policy's robustness limits?



- How to inculcate robustness into multistage decision policies?



Innovation

- To quantify the brittleness, we optimize the stage cost

$$\max_{\mathbf{v}_t \sim \psi \in \Psi} \left[\sum_{t=0}^{T} \underbrace{c(\boldsymbol{x}_t, \mathbf{u}_t)}_{\text{nominal}} - \gamma \underbrace{g(\mathbf{v}_t)}_{\text{adversarial}} \right]$$

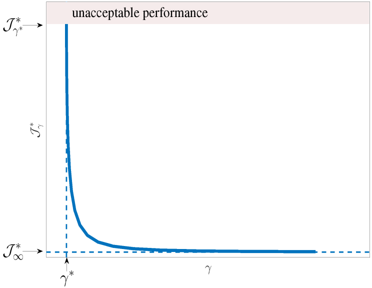
- To mitigate lack of robustness, we optimize the *cost-to-go*

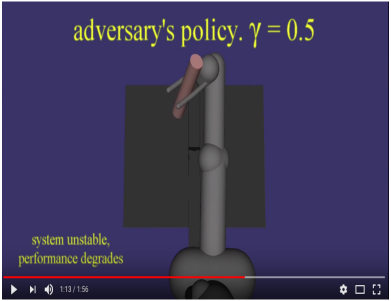
$$\mathcal{J}_t(\boldsymbol{x}_t, \pi, \psi) = \min_{\mathbf{u}_t \sim \boldsymbol{\pi}} \max_{\mathbf{v}_t \sim \psi} \left(\sum_{t=0}^{T-1} \ell_t(\boldsymbol{x}_t, \mathbf{u}_t, \mathbf{v}_t) + L_T(\boldsymbol{x}_T) \right),$$

- and seek a saddle point equilibrium policy that satisfies

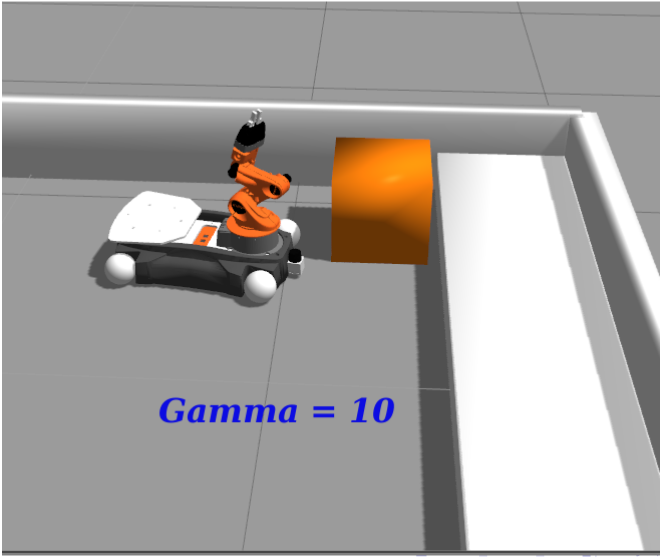
$$\mathcal{J}_t(\boldsymbol{x}_t, \pi^*, \psi) \le \mathcal{J}_t(\boldsymbol{x}_t, \pi^*, \psi^*) \le \mathcal{J}_t(\boldsymbol{x}_t, \pi, \psi^*),$$

Results: Brittleness Quantification





Results: Iterative Dynamic Game



Future Work: MRI/RT Immobilization

- Explore multiple parallel robot mechanisms for head motion correction.
- Adopt iterative dynamic game approach [Ogunmolu et al. (2018)] for solving robust controller for head stabilization.
- Build on Freeman and Kokotovic's point-wise min-norm robust control lyapunov function to realize a meaningful value function in deep policies [Freeman and Kokotovic (1996)].

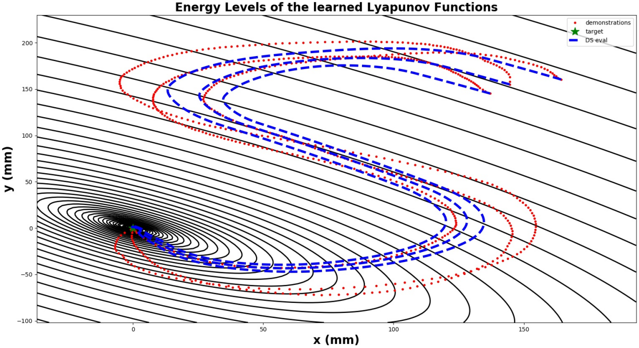
Constrained Robust Control Lyapunov Function (RCLF) Motion Planning

Problem Definition

Find RCLF collision-free path $\sigma^\star : [0,1] \rightarrow \mathcal{M}_{free}$ given a path planning problem $\left(\mathcal{Q}_{free}, \xi_i, \mathcal{Q}_{goal}\right)$, manipulation constraint, G, and cost function V such that $V(\sigma^\star) = \min_{\sigma \in \Sigma_{\mathcal{M}_{free}}} V(\sigma)$ if one exists.

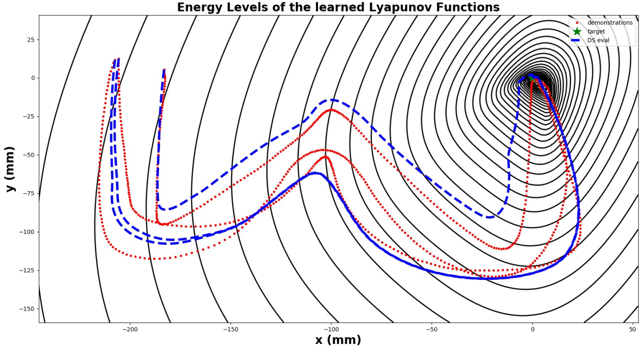
- Leverage [Freeman and Kokotovic (1996),Ogunmolu et al. (2018)].
- Greedy approach using L-BFGS optimization algorithm with box constraints, in contrast to the quadratic nonlinear constrained optimization e.g. Khansari-Zadeh and Billard (2014).

Reproducing a Nonlinear Motion with GMM



A CLF motion executor (red curves) that shows convergence to local attractors (green asterisks) and follows 3 different set trajectories (blue curves) for 2D nonlinear motion-trajectory problems on the WAM robot [Reproduced from Ogunmolu et al. (2020)].

Reproducing a Nonlinear Motion with GMM



A CLF motion executor (red curves) that shows convergence to local attractors (green asterisks) and follows 3 different set trajectories (blue curves) for 2D nonlinear motion-trajectory problems on the WAM robot [Reproduced from Ogunmolu et al. (2020)].

Conclusions

- Designed a non-invasive soft robot for head motion compensation in IMRT/emerging MRI-LINACs ✓
- Photons-transparent; Adaptible under MRI coils for newer MRI-LINACs ✓
- Fast inference of beam orientations in treatment planning: Approx 60 secs beams prediction time✓
- Adapted H-∞ control methods for quantifying the brittleness of deep policies✓
- Devised a min-max-trained deep saddle policy for mitigating model mismatch, transfer errors, and policy sensitivity e.t.c. ✓

End of Slides/More Acknowledgments

James Pikul, Asst. Prof. Penn

Kevin Turner, Prof. Penn

Audrey Sedal, Res. Asst. Prof. TTIC

Rachel Thompson, Student MIT

Ireti Akinola, PhD Student, Columbia

George Pappas, Prof. Penn

Jethro Tan, SW Eng. PFN Tokyo

MAIA Lab Members, UTSW

Publications

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *International Conference on Machine Learning*, pages 805–813, 2015.

Sadeghnejad Barkousaraie, Azar and Ogunmolu, Olalekan and Jiang, Steve and Nguyen, Dan. A Fast Deep Learning Approach for Beam Orientation Selection Using Supervised Learning with Column Generation on IMRT Prostate Cancer Patients. *Medical Physics, American Association of Physicists in Medicine*, 2019.

Radhini Chelvarajah, Brigid Leighton, Linda Martin, Wayne Smith, and Rachael Beldham-Collins. Cranial immobilisation—is there a better way? *Radiographer*, 51(1):29–33, 2004.

Xinmin Liu, Andrew H Belcher, Zachary Grelewicz, and Rodney D Wiersma. Robotic stage for head motion correction in stereotactic radiosurgery. In *2015 American Control Conference (ACC)*, pages 5776–5781. IEEE, 2015.

Andrew Belcher. *Patient Motion Management with 6-DOF Robotics for Frameless and Maskless Stereotactic Radiosurgery*. PhD thesis, The University of Chicago, 2017.

Olalekan Ogunmolu and Rodney Wiersma. A Real-Time Patient Head Motion Correction Mechanism for MRI-Linac Systems. In *Joint AAPM | COMP Meeting, Virtual*, 2020.

James Pikul, Itai Cohen, and Robert Shepherd. Stretchable surfaces with programmable texture, May 2 2019. US Patent App. 16/161,029.

PC Parks. Liapunov Redesign of Model Reference Adaptive Control Systems. *IEEE Transactions on Automatic Control*, 11(3):362–367, 1966.

Olalekan Ogunmolu, Nicholas Gans, and Tyler Summers. Minimax Iterative Dynamic Game : Application to Nonlinear Robot Control. *IEEE International Conference on Intelligent Robots and Systems*, 2018.

Randy A Freeman and Petar V Kokotovic. Inverse optimality in robust stabilization. *SIAM journal on control and optimization*, 34(4):1365–1391, 1996.

S Mohammad Khansari-Zadeh and Aude Billard. Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

Olalekan Ogunmolu, Rachel Skye Thompson, and Rodrigo Pérez Dattari. Learning Control Lyapunov Functions in Python. `https://github.com/lakehanne/lyapunovearner`, 2020. Accessed February 10, 2020.

Olalekan Ogunmolu, Adwait Kulkarni, Yonas Tadesse, Xuejun Gu, Steve Jiang, and Nicholas Gans.

BOO: Search

- Formulated as a bandit search that imposes a regret term on the Q-value
- During the planning process, we estimate a *value*, $v(\mathbf{y}_t)$, that estimates the optimality of a beam block;
- In parallel, we refine the deep neural network policy by optimizing its weight in a separate thread.
  - Network parameters updated by a **mixed strategy** which combines its **pure strategy**,
  - It is a best response to the fictitious opponent's **average pure strategy**.

BOO: Fluence Map Optimization

- Q-value defined as

$$\bar{Q}(s,a) = Q_j(s,a) + c\sqrt{\frac{2\ \ln\ n(s)}{N(s,a}}, \tag{2}$$

$$a^\star = \arg\max_a \bar{Q}(s,a) \tag{3}$$

Fluence Map Optimization

- $\mathcal{X} \implies$ total discretized voxels of interest (VOI's) in a target volume
- $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \ldots \cup \mathcal{B}_n \subseteq \mathcal{B} \implies$ beam partition subset
- $\mathcal{D}_{ij}(\theta_k) \implies$ matrix that describes each dose influence, d_i
  - delivered to a discretized voxel, i, in a volume of interest, VOI_h ($h = 1, \ldots, \mathcal{X}$), from a beam angle, θ_k, $k \in \{1, \ldots, n\}$

BOO: Fluence Map Optimization

- Suppose further that $\mathcal{D}_{ij}(\theta_k)$ is the matrix that describes each dose influence, d_i
  - delivered to a discretized voxel, i, in a volume of interest, VOI_h $(h = 1, \ldots, \mathcal{X})$, from a beam angle, θ_k, $k \in \{1, \ldots, n\}$
- We compute the matrix $\mathcal{D}_{ij}(\theta_k)$ by calculating each d_i for every bixel, j, at every φ°, resolution, where $j \in \mathcal{B}_k$

BOO: FMO problem definition

- The fluence problem is to find the values of decision variables, x_j, for which d_i to the tumor is maximized, while simultaneously minimizing the d_i to critical structures
- For the voxels in a target volume,
  - let a weighted quadratic objective minimize the l_2 distance between a pre-calculated dose $\boldsymbol{Ax}$, and a doctor's prescribed dose, $\mathbf{b}$
  - let a weighted quadratic objective maximizes the l_2 distance between $\boldsymbol{Ax}$ (where $\boldsymbol{x}$ represents the vectorized bixels, x_j) and $\mathbf{b}$

BOO: FMO problem definition

- Cost

$$\frac{1}{v_s} \sum_{s \in \text{OARs}} \|(b_s - \underline{w}_s \mathcal{D}_{ij}^s \boldsymbol{x}_s)_+\|_2^2 + \frac{1}{v_s} \sum_{s \in \text{PTVs}} \|(\bar{w}_s \mathcal{D}_{ij}^s \boldsymbol{x}_s - b_s)_+\|_2^2 \tag{4}$$

- Pre-calculated dose term: $\boldsymbol{Ax} = \{\sum_s \frac{w_s}{v_s} \mathcal{D}_{ij}^s \boldsymbol{x}_s \,|\, \mathcal{D}_{ij} \in \mathbb{R}^{n \times l}, n > l\}$, which is a combination of the dose components that belong to OARs and those that belong to PTVs.
- Let $w_s = \{\underline{w}_s, \bar{w}_s\}$ be the respective underdosing and overdosing weights for the OARs and PTVs
  - v_s represents the total number of voxels in each structure.

BOO: FMO

- Rewriting the objective

$$\min \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2 \quad \text{subject to } x \geq 0.$$

- With Lagrangian:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \min \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2 - \boldsymbol{\lambda}^T \boldsymbol{x}.$$

- Introducing an auxiliary variable $\mathbf{z}$, we have

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \mathbf{b}\|_2^2, \quad \text{subject to } \boldsymbol{z} = \boldsymbol{x},\ \ \boldsymbol{z} \geq 0,$$

BOO: FMO by way of ADMM

- Solving either the $\boldsymbol{x}$ and $\boldsymbol{z}$ sub-problems, we have

$$\boldsymbol{x}^{k+1} = \left(\boldsymbol{A}^T\boldsymbol{A} + \rho\boldsymbol{I}\right)^{-1}\left(\boldsymbol{A}^T\boldsymbol{b} + \rho\boldsymbol{z}^k - \boldsymbol{\lambda}^k\right). \tag{5}$$

- And using the soft-thresholding operator, $S_{\boldsymbol{\lambda}/\rho}$, we find that

$$\boldsymbol{z}^{k+1} = S_{\boldsymbol{\lambda}/\rho}\left(\boldsymbol{x}^{k+1} + \boldsymbol{\lambda}^k\right), \tag{6}$$

where $S_{\boldsymbol{\lambda}/\rho}(\tau) = (x - \boldsymbol{\lambda}/\rho)_+ - (-\tau - \boldsymbol{\lambda}/\rho)_+$. $\boldsymbol{\lambda}$ is updated as

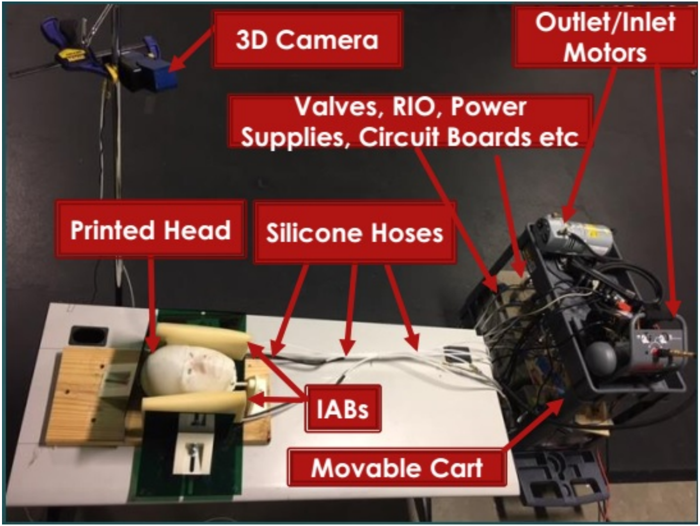
$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \gamma(\boldsymbol{z}^{k+1} - \boldsymbol{x}^{k+1}), \tag{7}$$

where γ is a parameter that controls the step length.

App A: 3-DOF Closed-loop Control

Closed-loop Phantom motion control along 3 DoFs with an adaptive neuro-controller.

3-DOF Simulation Testbed



Hardware Description

Control Design Goals

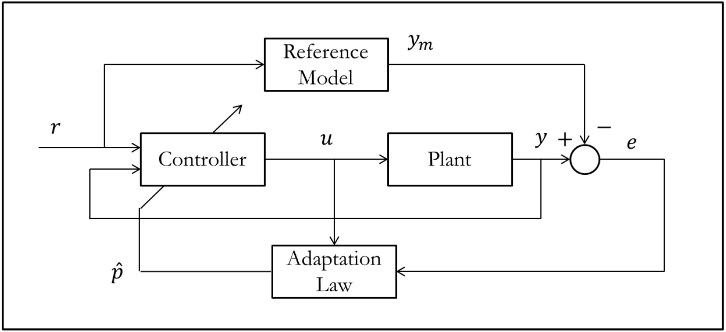
- Stabilize z, pitch, and roll states, *i.e.*

$$\boldsymbol{x} = \begin{pmatrix} z \\ \theta \\ \phi \end{pmatrix}$$

- By solving an adaptive state feedback controller, optimal regulation, and minimize parametric uncertainties

Control Design Goals

- Provide closed loop tracking given a desired trajectory, **r**
- Robustify system to (non-)parametric uncertainties



Indirect MRAC system. (Source mdpi.com)

Lyapunov Redesign: Theorem

- For correct adaptive gains, $\hat{\mathbf{K}}_x$ and $\hat{\mathbf{K}}_r$, $\mathbf{e}(k)$ is ***uniformly ultimately bounded***, and the state $\boldsymbol{x}$ converges to a neighborhood of $\mathbf{r}$.
- Choose a $\mathbf{V}$ in terms of $\mathbf{e}$; $\tilde{\mathbf{K}}_x^T$, $\tilde{\mathbf{K}}_r^T$; and parameter error $\varepsilon_f(\boldsymbol{x}(k))$ space

$$\mathbf{V}(\mathbf{e}, \tilde{\mathbf{K}}_x, \tilde{\mathbf{K}}_r^T) = \mathbf{e}^T\mathbf{P}\mathbf{e} + \mathbf{tr}(\tilde{\mathbf{K}}_x^T \boldsymbol{\Gamma}_x^{-1} \tilde{\mathbf{K}}_x^T |\boldsymbol{\Lambda}|) + \mathbf{tr}(\tilde{\mathbf{K}}_r^T \Gamma_r^{-1} \tilde{\mathbf{K}}_r |\boldsymbol{\Lambda}|)$$

Stability proof

$$\dot{\mathbf{V}}(\mathbf{e}, \tilde{\mathbf{K}}_y, \tilde{\mathbf{K}}_r) = \dot{\mathbf{e}}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}} + 2\mathbf{tr}(\tilde{\mathbf{K}}_y^T \boldsymbol{\Gamma}_{\boldsymbol{y}}^{-1} \dot{\hat{\mathbf{K}}}_y |\boldsymbol{\Lambda}|) \\ + 2\mathbf{tr}(\tilde{\mathbf{K}}_r^T \boldsymbol{\Gamma}_r^{-1} \dot{\hat{\mathbf{K}}}_r |\boldsymbol{\Lambda}|)$$

$$= \left[\mathbf{A}_m \mathbf{e} + \mathbf{B}\boldsymbol{\Lambda}[\Delta\hat{\mathbf{K}}_r^T \mathbf{r} + \Delta\hat{\mathbf{K}}_x^T \mathbf{x}]\right]^T \mathbf{P}\mathbf{e} + \ldots \\ \mathbf{e}^T \mathbf{P} \left[\mathbf{A}_m \mathbf{e} + \mathbf{B}\boldsymbol{\Lambda}[\Delta\hat{\mathbf{K}}_r^T \mathbf{r} + \Delta\hat{\mathbf{K}}_x^T \mathbf{x}]\right] + \ldots \\ 2\,\mathbf{tr}(\Delta\mathbf{K}_x^T \Gamma_x^{-1} \dot{\hat{K}}_x |\boldsymbol{\Lambda}|) + 2\,\mathbf{tr}(\Delta\mathbf{K}_r^T \Gamma_r^{-1} \dot{\hat{\mathbf{K}}}_r |\boldsymbol{\Lambda}|)$$

Stability Analysis

$$= \mathbf{e}^T(\mathbf{P}\,\mathbf{A}_m + \mathbf{A}_m^T\mathbf{P})\mathbf{e} + 2\mathbf{e}^T\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\left(\tilde{\mathbf{K}}_y^T\mathbf{y} + \tilde{\mathbf{K}}_r^T\mathbf{r}\right)$$
$$+2\,\mathbf{tr}\left(\tilde{\mathbf{K}}_y^T\boldsymbol{\Gamma}_{\mathbf{y}}^{-1}\dot{\hat{\mathbf{K}}}_y|\boldsymbol{\Lambda}|\right) + 2\,\mathbf{tr}\left(\tilde{\mathbf{K}}_r^T\boldsymbol{\Gamma}_r^{-1}\dot{\hat{\mathbf{K}}}_r|\boldsymbol{\Lambda}|\right)$$

$$= -\mathbf{e}^T\mathbf{Q}\mathbf{e} - 2\mathbf{e}^T\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\varepsilon_f(\mathbf{y}) + 2\mathbf{e}^T\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\tilde{\mathbf{K}}_y^T\mathbf{y}$$
$$+\,2\,\mathbf{tr}\left(\tilde{\mathbf{K}}_y^T\boldsymbol{\Gamma}_{\mathbf{y}}^{-1}\dot{\hat{\mathbf{K}}}_y\right) + 2\mathbf{e}^T\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\tilde{\mathbf{K}}_r^T\mathbf{r} + 2\,\mathbf{tr}\left(\Delta\mathbf{K}_r^T\boldsymbol{\Gamma}_r^{-1}\dot{\hat{\mathbf{K}}}_r\right)$$

Notice $x^T y = \mathbf{tr}\left(y\,x^T\right)$ from trace identity

Stability Analysis Cont'd

Therefore,

$$\begin{aligned}\dot{\mathbf{V}}(\cdot) &= -\mathbf{e}^T\mathbf{Q}\mathbf{e} - 2\mathbf{e}^T\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\varepsilon_f \\ &\quad + 2\,\mathbf{tr}\left(\tilde{\mathbf{K}}_y^T(\boldsymbol{\Gamma}_{\boldsymbol{y}}^{-1}\dot{\hat{\mathbf{K}}}_y + \mathbf{y}\mathbf{e}^T\mathbf{P}\mathbf{B}sgn(\boldsymbol{\Lambda})\right)|\boldsymbol{\Lambda}| \\ &\quad + 2\,\mathbf{tr}\left(\tilde{\mathbf{K}}_r^T(\boldsymbol{\Gamma}_r^{-1}\dot{\hat{\mathbf{K}}}_r + \mathbf{r}\mathbf{e}^T\mathbf{P}\mathbf{B}sgn(\boldsymbol{\Lambda})\right)|\boldsymbol{\Lambda}|\end{aligned}$$

where for a real-valued x, we have $x = \text{sgn}(x)|x|$.

- first two terms will be negative definite for all $\mathbf{e} \neq 0$
  - since $\mathbf{A}_m$ is Hurwitz
- other terms will be identically null if we choose the adaptation laws

$$\dot{\hat{\mathbf{K}}}_y = -\boldsymbol{\Gamma}_{\boldsymbol{y}}\mathbf{y}\mathbf{e}^T\mathbf{P}\,\mathbf{B}\text{sgn}(\boldsymbol{\Lambda}), \quad \dot{\hat{\mathbf{K}}}_r = -\boldsymbol{\Gamma}_r\mathbf{r}\mathbf{e}^T\mathbf{P}\,\mathbf{B}\,\text{sgn}(\boldsymbol{\Lambda})$$

Stability Results: Ogunmolu et al. (2017)

$$\begin{aligned}\dot{\mathbf{V}}(\cdot) &= -\mathbf{e}^T\mathbf{Q}\mathbf{e} - 2\mathbf{e}^T\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\varepsilon_f \\ &\leq -\lambda_{low}\|\mathbf{e}\|^2 + 2\|\mathbf{e}\|\|\mathbf{P}\mathbf{B}\|\lambda_{high}(\boldsymbol{\Lambda})\varepsilon_{max}\end{aligned}$$

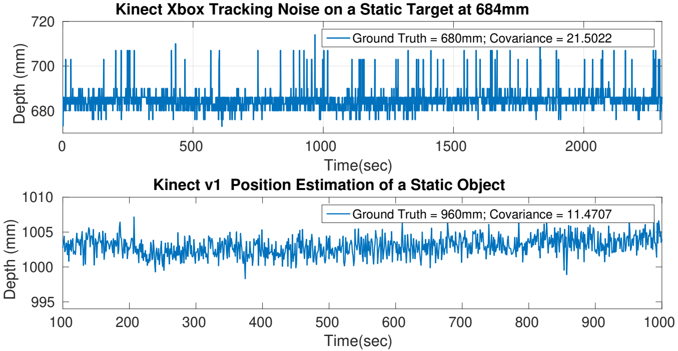
- $\{\lambda_{low}, \lambda_{high}\} \equiv$ min/max eigenvalues of Q and $\boldsymbol{\Lambda}$.
- $\dot{\mathbf{V}}(\cdot)$ is thus negative definite outside the compact set: $\chi = \left(\mathbf{e} : \|\mathbf{e}\| \leq \frac{2\|\mathbf{P}\mathbf{B}\|\lambda_{high}(\boldsymbol{\Lambda})\varepsilon_{max}(\mathbf{y})}{\lambda_{low}(\mathbf{Q})}\right)$
  - *i.e.* $\mathbf{e}$ is **uniformly ultimately bounded**, or $\mathbf{y}(t) \to 0$ as $t \to \infty$.

Appendix C: 1-DOF Closed-Loop Control

Left blank intentionally

Head Pose Estimation: Sensor Fusion

Sensors' Noise Floor



Case for Sensed Observation Filtering

Optimal head state estimation and sensor fusion

- Find the observation estimates $\hat{\mathbf{x}}(i)$ that minimize the mean-square error from true measurement, *i.e.* ,

$$\hat{\mathbf{x}}(i|j) = \arg \min_{\hat{\mathbf{x}}(i|j) \in \mathbb{R}^n} \mathbb{E}\{(\mathbf{x}(i) - \hat{\mathbf{x}})^T (\mathbf{x}(i) - \hat{\mathbf{x}}) | z(1), \cdots, z(j)\}$$

- Define the estimate error's covariance as

$$\mathbf{P}(i|j) \triangleq \mathbb{E}\{(\mathbf{x}(i) - \hat{\mathbf{x}}(i|j))^T (\mathbf{x}(i) - \hat{\mathbf{x}}(i|j)) \, | Z^j\}. \tag{8}$$

State estimation with Kalman filters

- Assume state model, $\mathbf{F}(k)$
- For a discretized time interval ΔT between measurements, we define the state

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}_k + \mathbf{G}_k\mathbf{w}_k \tag{9}$$

  - with

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \tag{10}$$

Kalman filters

- (9) $\implies \mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{G}_k \mathbf{w}_k$
- $\mathbf{G}_k :=$ uncontrolled forces accelerating on head
- Head's acceleration, $a_k \sim \mathcal{N}(0, \sigma_a)$
- Setting $\mathbf{G}_k = \mathbf{I}_{2x2}$ and $\mathbf{w}(k) \sim \mathcal{N}(0, \mathbf{Q}(k))$
  - set $\mathbf{Q}(k)$ to a random walk sequence, $\mathbf{W}_k = [\frac{\Delta T^2}{2}, \Delta T]^T$

Kalman Filters Design

- Such that

$$\mathbf{Q} = \mathbf{W}\mathbf{W}^T {\sigma_a}^2 = \begin{bmatrix} \frac{\Delta T^4}{4} & \frac{\Delta T^3}{2} \\ \frac{\Delta T^3}{2} & \Delta T^2 \end{bmatrix} {\sigma_a}^2. \quad (11)$$

- Set the transfer matrix from the estimates, $\mathbf{x}(k)$, to observations, $z_1(k)$ and $z_2(k)$ according to

$$z_s = \mathbf{H}_s(k)\mathbf{x}(k) + v_s(k) \qquad s = 1, 2 \quad (12)$$

  - where $\mathbf{H}_s(k) = \begin{bmatrix} 1, & 0 \end{bmatrix}^T$ and $v_s(k) \sim \mathcal{N}(0, \sigma_{rs}^2)$

KF Priori and Posteriori State Estimates

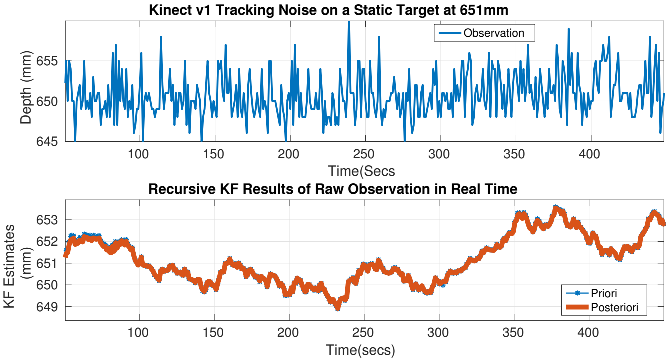
- **Prediction Phase:**

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k{}^T + \mathbf{Q}_k \end{aligned} \tag{13}$$

- **Update Phase:**

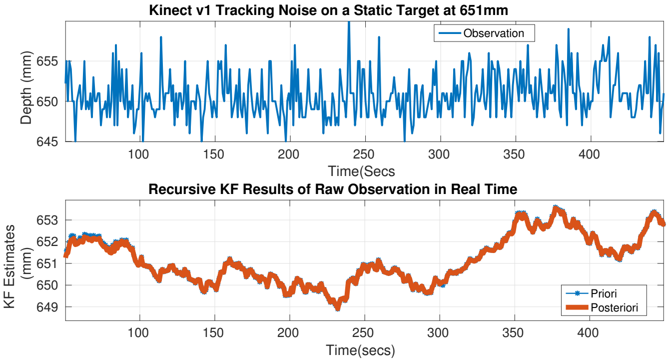
$$\begin{aligned} \mathbf{K}(k) &= \mathbf{P}(k|k-1)\mathbf{H}(k)^T[\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}(k)^T + \mathbf{R}(k)]^{-1} \\ \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1)) \\ \mathbf{P}(k|k) &= (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}(k|k-1) \end{aligned} \tag{14}$$

Xbox Filtering Results



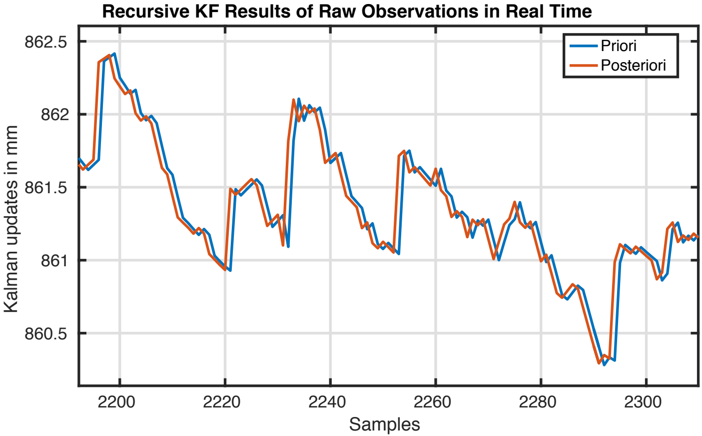
KF results of Xbox's observations

Kinect 2 Filtering Results



KF results of Kinect v1's observation

Filtering Results



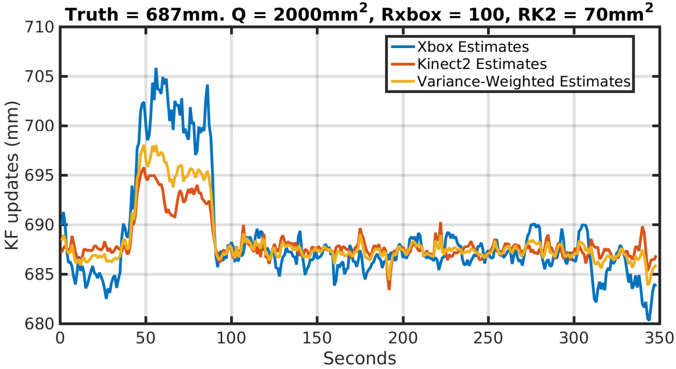
Global fusion of local tracks

- Fuse both updates with a variance-weighted average of each local track as follows,

$$\hat{\mathbf{x}}(F)(k|k) = \mathbf{P}(F)(k|k)\sum_{s=1}^{N}\left[\mathbf{P}(s)^{-1}(k|k)\hat{\mathbf{x}}(s)(k|k)\right]$$

$$\text{where } \mathbf{P}(F)(k|k) = \left[\sum_{s=1}^{N}\mathbf{P}(s)^{-1}(k|k)\right]^{-1}.$$

Track-to-track Fusion



Track-to-track fusion of both sensors' local track estimates.

Vision-based Control

This page is left blank intentionally.

System Identification

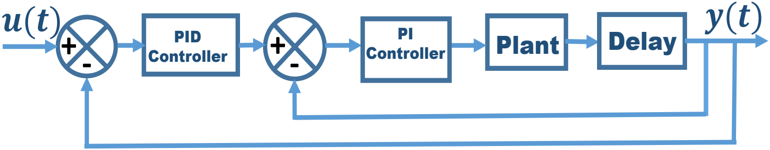
- From I/O data, estimate a model
- Find set of optimal model parameters via the minimization,

$$G(t) = \arg \min_{\theta} V_N(\theta, Z^N)$$

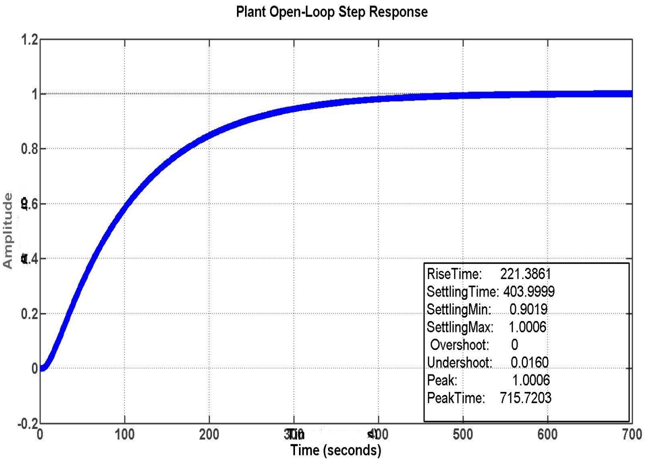
  - where $V_N(\theta, Z^N) = \sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} \frac{1}{2}(\hat{y}_i(k) - y_i(k))^2$,
  - and $Z^N = \{u(1) \cdots u(N),\ y(1) \cdots y(N)\}$
- After a least squares minimization, we derive a state-space realization,

$$\begin{aligned} \mathbf{x}(k + Ts) &= \mathbf{Ax}(k) + \mathbf{Bu}(k) + \mathbf{Ke}(k) \\ \mathbf{y}(k) &= \mathbf{Cx}(k) + \mathbf{Du}(k) + \mathbf{e}(k) \end{aligned} \tag{15}$$

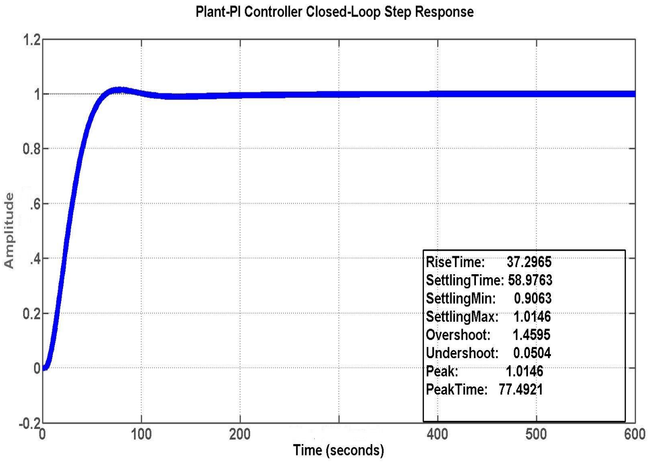
PID-PI Cascaded Controller



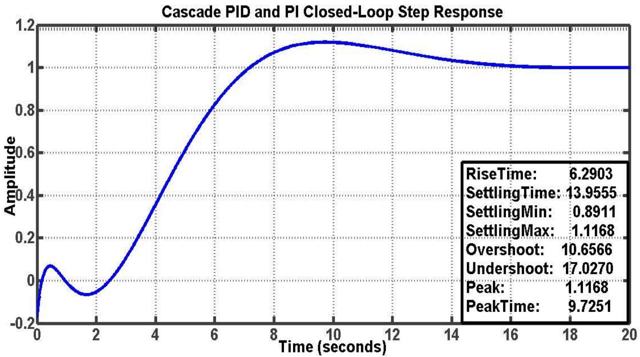
Open Loop Step Response



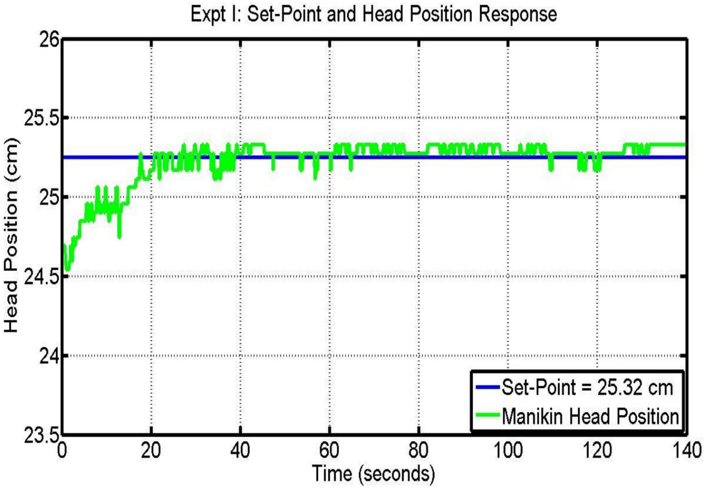
PI Closed-loop Step Response



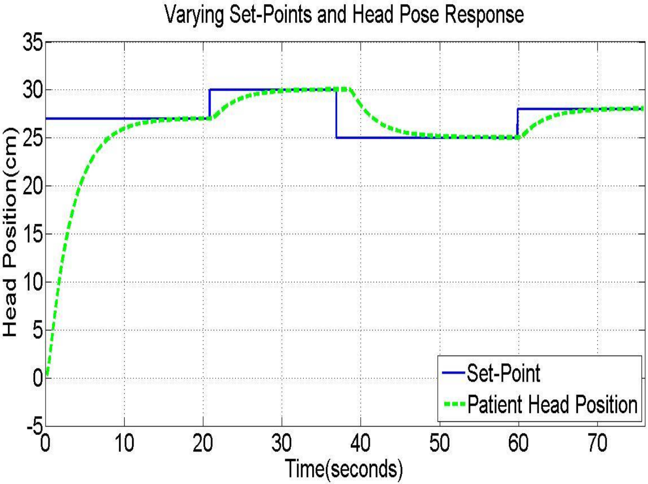
Cascaded PID-PI Closed-loop Step Response



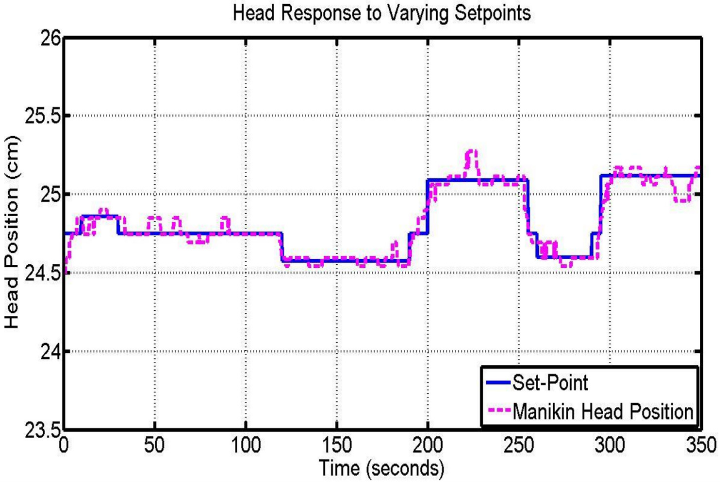
PI-PD Cascaded Controller Experimental Results



Varying Setpoint Simulation



Varying Setpoint Experiment



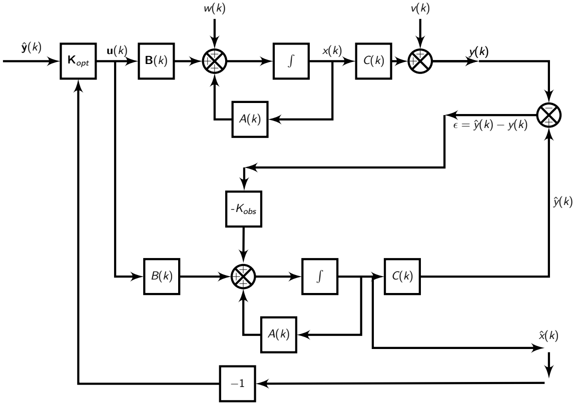
3-DOF Controller Design (IROS 2017)

- Posing the cost

$$J = \sum_{k=0}^{\mathcal{K}} x^T(k)\, Q\, x(k) + u(k)^T\, R\, u(k) + 2x(k)^T\, N\, u(k)$$

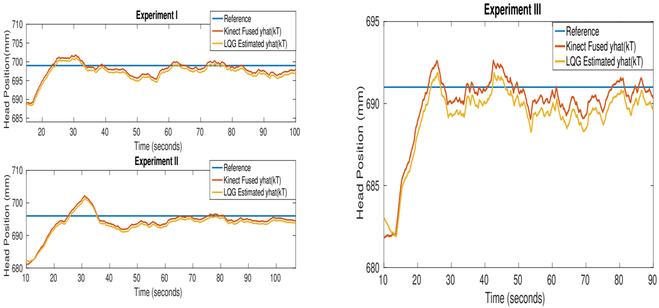
- we can obtain u as $\Delta u = \arg\min_{\Delta u} J$
- Δu is a future control sequence

Closed-loop control (Full state observer).



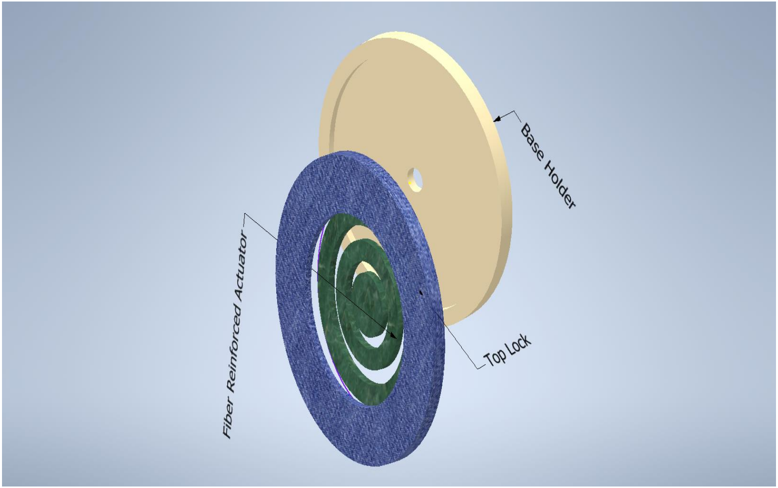
$$\hat{x}(k+1) = A(k)\hat{x}(k) - K_{obs}[C(k)\hat{x}(k) - y(k)] + B(k)u(k).$$

1-DOF Control Results



LQG Controller on mannequine head.

Appendix D: CCOARSE Actuator Schematic



Soft IK via Boundary Value Problem

- With Cauchy's laws of motion, solve boundary volume problem of traction
  - Using the following boundary conditions for the Cauchy Stress,
    - $\sigma_{rr|R=R_0} = -P_{atm}, \sigma_{rr|R=R_i} = -P_{atm} - P$
  - And together with Cauchy's first law, we find that
    - $\sigma_{rr}(r) = -\int_{r_i}^{r_o}[2C_1(\frac{r}{R^2} - \frac{R^4}{r^5}) + 2\,C_2\left(\frac{r^3}{R^4} - \frac{R^2}{r^3}\right)]dr$
    - $\sigma_{rr}(r) = \int_{R_i}^{R_o}[2C_1\left(\frac{1}{r} - \frac{R^6}{r^7}\right) - 2C_2\left(\frac{R^4}{r^5} - \frac{r}{R^2}\right)]\,dr$
  - With $\sigma_{rr|R=R_i} = -P_{atm} - P$ and setting $P_{atm} = 0$, we find
    - $P(r) = \int_{r_i}^{r_o}[2C_1(\frac{r}{R^2} - \frac{R^4}{r^5}) + 2\,C_2\left(\frac{r^3}{R^4} - \frac{R^2}{r^3}\right)]dr$
    - $P(r) = \int_{R_i}^{R_o}[2C_1\left(\frac{1}{r} - \frac{R^6}{r^7}\right) - 2C_2\left(\frac{R^4}{r^5} - \frac{r}{R^2}\right)]\,dr$