SoftNeuroAdapt: A 3-DoF Neuro-Adaptive Healthcare System

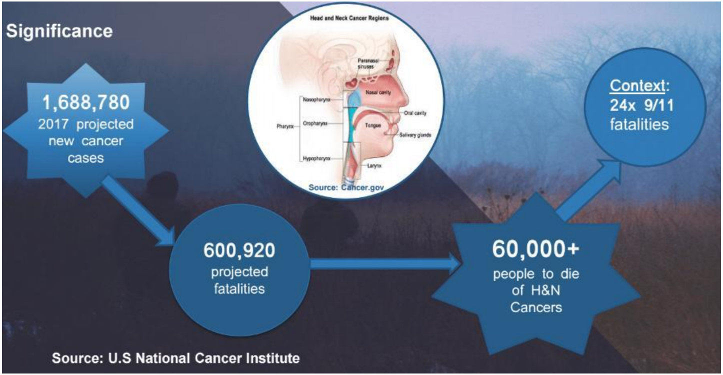
Nicholas Gans & Olalekan Ogunmolu

Department of Electrical Engineering,
The University of Texas at Dallas, Richardson, TX
Work in partnership with
The Department of Radiation Oncology,
UT Southwestern Medical Center, Dallas, TX

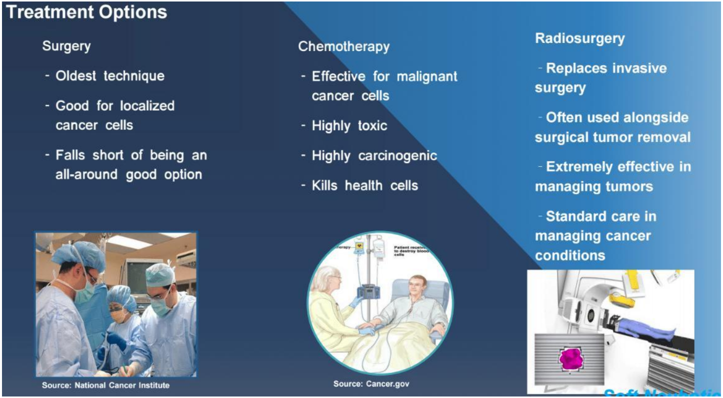
February 5, 2018

Background

- Cancers of the head and neck (H&N)



Treatment Options



- Radiotherapy treatment is the most effective

Radiotherapy Treatment

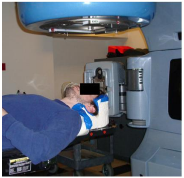
- 3D CT/MRI scans too determine treatment plan
- Requires head landmarks + mechanical immobilization objects
- Employs ionizing radiation to destroy abnormal tissues
- Involves less precise radiation dose over many days/weeks

Radiation Therapy Drawbacks

- Rigid immobilization
  - LINAC angle misalignments cause negative dosimetry effects [Xing (2000)]
  - treatment discomfort from long hours of surgery
  - requires inconvenient face masks
- Cyberknife
  - incompatible with conventional LINACs used at the majority of cancer centers.
  - real-time x-ray motion tracking introduces extra radiation-dose
- Novalis ExacTrac Robotic Tilt Module
  - only uses pre-treatment images
  - only provides rigid motion compensation

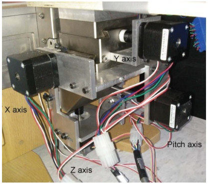
Frameless/Maskless Radiotherapy

(Cervino, L. 2010)



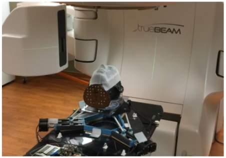
Feasibility evaluation

(Liu, X. 2015)



4-D robotic stage couch

(Ostyn, M. 2017)



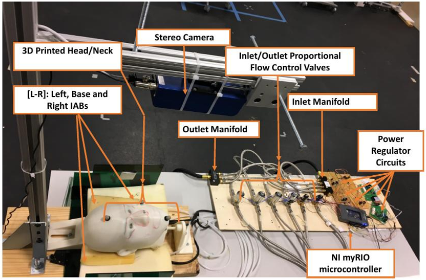
6-DoF rigid robotic couch

Solution: Soft-Robot Position Correcting Systems

- Eliminate rigid frames and metallic rings
  - Maskless and frameless positioning system ✓
- Eliminate attenuation of X-Ray beams
  - Separate electromechanical components from patient actuation ✓
  - Explore soft polymer actuators ✓
- Control design
  - Feedback control + optimal regulation + robustness to disturbance ✓
- Adapt for changing parameters in system's dynamics within control law
  - Neural network function approximator ✓

Hardware set-up

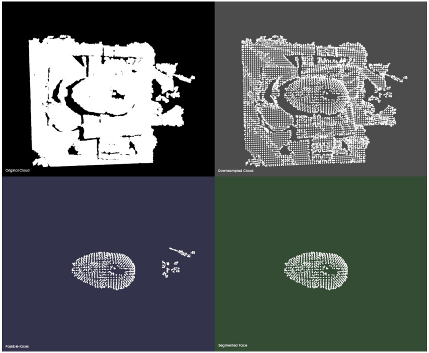
- **System Components**
  - 3D printed head
    - ≈ size of an average adult male
  - 3 custom-designed inflatable air bladders (IABs)
    - made from elastomeric polymers
    - encased in a breathable foam pad for soft actuation
  - A regulated air canister supplied constant air pressure at 15 psi
  - Proportional solenoid valves regulate amount of air supply through silicone hoses



3-DoF Motion Correction System

Vision-based head pose estimation

- Acquire face point cloud from stereo camera
- Find edges of 2D planar regions in scene structure (Torr and Zisserman, 2000)
  - bound resulting plane indices with their 2D convex hull
- Extract face and face-height neighbors into a predefined 2D prismatic model (Rusu, R. et. al, 2008)
- Cluster extracted points based with Euclidean Clustering (Rusu Thesis)



[Top-left]: Dense point cloud of the scene.
[Top-right]: Downsampled cluttered cloud of the scene.
[Bottom-left]: Using RANSAC, we searched for 2D plane candidates in the scene and compute the convex hull of found planar regions. We then extrude point indices in the hull to a prismatic polygonal model to give the face region.
[Bottom-right]: An additional step clusters the resultant cloud based on a user-defined Euclidean distance.

Head Pose Estimation

- Goal: Compute optimal translation and rotation of the head
  - from a model point set $\mathbf{X} = \{\overrightarrow{x}_i\}$ to a measured point set $\mathbf{P} = \{\overrightarrow{p}_i\}$, where $N_x = N_p = 3$,
  - point $\overrightarrow{x}_i \in \mathbf{X}$ has the same index as $\overrightarrow{p}_i \in \mathbf{P}$
  - set three points on the table scene as model points
- Obtain centroid of segmented cloud from the previous scene
  - set this as measured point
- Compute the covariance matrix, Σ_{px}, of $\mathbf{P}$ and $\mathbf{X}$
  - extract cyclic components of the skew-symmetric matrix as Δ
  - form symmetric 4×4 matrix $Q(\Sigma_{px})$
    - $\mathbf{Q}(\boldsymbol{\Sigma}_{px}) = \begin{bmatrix} \mathbf{tr}(\boldsymbol{\Sigma}_{px}) & \boldsymbol{\Delta}^T \\ \boldsymbol{\Delta} & \boldsymbol{\Sigma}_{px} + \boldsymbol{\Sigma}_{px}^T - \mathbf{tr}(\boldsymbol{\Sigma}_{px})\mathbf{I}_3 \end{bmatrix}$

Head Pose Estimation (Besl & McKay, '92)

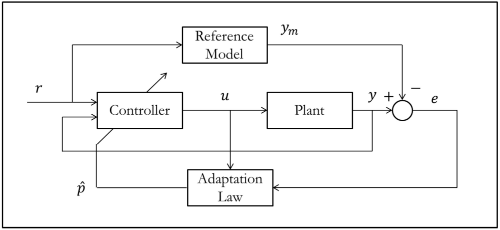
- The unit eigenvector, q_R, that corresponds to the maximum eigenvalue of $\mathbf{Q}(\Sigma_{px})$ is selected as the optimal rotation quaternion;
- Optimal translation vector given by
  - $\overrightarrow{q}_T = \overrightarrow{\mu}_x - \mathbf{R}(\overrightarrow{q}_R)\overrightarrow{\mu}_p$
- μ_x and μ_p are the mean of point sets **X** and **P** respectively
- Face pose defined by tuples $[q_T, q_R] = \{x, y, z, \theta, \phi, \psi\}$ with respect to the world frame
- We control z, θ, ϕ (i.e. z, roll, and pitch states)

Control Overview

- State feedback and feedforward regulation problem
- Adaptation mechanism based on an estimation of the head pose given a priori information about the system's states and past control actions
  - $Z^N = \{u(k), u(k-1), \ldots u(k-n_u), y(k), \ldots, y(k-n_y)\}$
  - Fix a persistently exciting input signal $u_{ex} \in L_2 \cap L_\infty$ to excite the nonlinear modes of the system
    - Approximate nonlinear system by a long short-term memory, Hochreiter (1997)
    - Parameterize network's last layer with a fully connected layer that outputs control torques to the valves

Control Design Objectives

- Stabilize system states, $\mathbf{y} = \left[x, y, z, \theta, \phi, \psi\right]^T$
- Provide closed loop tracking given a desired trajectory, $\mathbf{r}$
- Robustify system to (non-)parametric uncertainties
    - changing head shapes, size and other anatomic/tumor variations



Indirect MRAC system. (Source mdpi.com)

Model Reference Adaptive Control

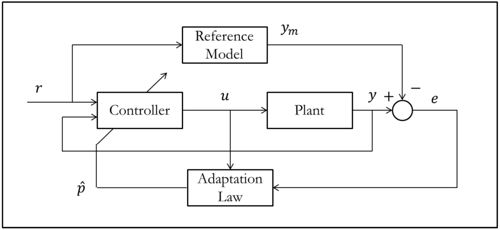
- Model head and bladder dynamics as
  - $\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} + \mathbf{B}\boldsymbol{\Lambda}\left(\mathbf{u} - f(\mathbf{y})\right) + \mathbf{w}(k)$
    - $\mathbf{A}$, $\boldsymbol{\Lambda}$ unknown, $\mathbf{B}$, $\text{sgn}\boldsymbol{\Lambda}$ known
    - $f(\mathbf{y}) \triangleq$ nonlinear function to be adapted for
    - $\mathbf{x} \triangleq$ tuple containing past controls and current outputs
- Approximate $f(\mathbf{y})$ by a neural network with continuous memory states
  - $\hat{f}\left(\mathbf{u}(k-d), \mathbf{y}(k), \mathbf{w}(k)\right)$ is realized with a *long-short term memory* cell (Horchreiter and Schmidhuber, '91, '97)
  - **purpose**: remember good adaptation gains

Assumptions

- A dynamic RNN with N neurons, $\varphi(\mathbf{x})$, exists
  - maps from a compact input space $\mathbf{u} \subset \mathbb{U}$ to $\mathbf{y} \subset \mathbb{Y}$ on the Lebesgue integrable functions within $[0, T]$ or $[0, \infty)$
- $f(\mathbf{x})$ is exactly $\Theta^T \Phi(\mathbf{x})$
  - f has coefficients $\Theta \in R^{N \times m}$ and a Lipschitz-continuous vector of basis functions $\Phi(\mathbf{x}) \in R^N$
- Inside a ball $\mathbf{Y}_R$ with known, finite radius R,
  - an ideal neural network (NN) approximation $f(\mathbf{x}) : R^n \to R^m$, is realized to a sufficient degree of accuracy, $\varepsilon_f > 0$;
- Outside $\mathbf{Y}_R$,
  - the NN approximation error can be upper-bounded by a known unbounded, scalar function $\varepsilon_{max}(\mathbf{x})$;
  - $\|\varepsilon(\mathbf{x})\| \leq \varepsilon_{max}(\mathbf{x}), \quad \forall \quad \mathbf{x} \in \mathbf{Y}_R$;
- There exists an exponentially stable reference model
  - $\dot{\mathbf{y}}_m = \mathbf{A}_m y_m + \mathbf{B}_m \mathbf{r}$

Control Design Objectives

- Stabilize system states, $\mathbf{y} = \left[x, y, z, \theta, \phi, \psi\right]^T$
- Provide closed loop tracking given a desired trajectory, $\mathbf{r}$
- Robustify system to (non-)parametric uncertainties
    - changing head shapes, size and other anatomic/tumor variations



Indirect MRAC system. (Source mdpi.com)

Adaptive Neuro-Control Scheme

- Set control law in terms of parameter estimates from the neural network weights and Lipschitz basis functions
  - $\Phi(\mathbf{y}) = \{\mathbf{y}(k-d), \cdots, \mathbf{y}(k-d-4), \mathbf{u}(k-d) \cdots \mathbf{u}(k-d-5)\}$
  - *i.e.* network looks back in time by 5 time steps at every instant, and then makes a prediction
- Derive adaptive adjustment mechanism from Lyapunov analysis for Adaptive Control (Parks, P., 1966)

Controller formulation

- $\mathbf{u} = \underbrace{\hat{\mathbf{K}}_y^{\mathbf{T}} \mathbf{y}}_{\text{state feedback}} + \underbrace{\hat{\mathbf{K}}_r^{\mathbf{T}} \mathbf{r}}_{\text{optimal regulator}} + \underbrace{\hat{f}(\mathbf{y}, \mathbf{u})}_{\text{approximator}}$
  - $\hat{\mathbf{K}}_y$ and $\hat{\mathbf{K}}_r$ are adaptive gains to be designed

Term Contributions

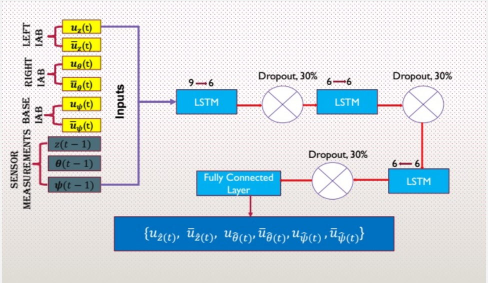
- $\hat{\mathbf{K}}_y^{\mathbf{T}} \mathbf{y}$ term keeps the states of the approximation set $\mathbf{y} \in \mathbf{B}_R$ stable,
- $\hat{\mathbf{K}}_r^{\mathbf{T}} \mathbf{r}$ term causes the states to follow a given reference trajectory
- Function approximator $\hat{f}(\cdot)$ ensures states that start outside the approximation set $\mathbf{y} \in \mathbf{B}_R$ converge to $\mathbf{B}_R$ in finite time

Adaptive Control Formulation

- Assume model matching conditions
  - such that $\hat{\mathbf{K}}_y = \mathbf{K}_y$, and $\hat{\mathbf{K}}_r = \mathbf{K}_r$ (ideally)
- Realize the approximator as $\hat{f}(\mathbf{x}) = \hat{\mathbf{\Theta}}^T \mathbf{\Phi}(\mathbf{y}) + \varepsilon_f(\mathbf{y})$
  - $\hat{\mathbf{\Theta}}^T$ denotes the vectorized weights of the neural network
  - $\mathbf{\Phi}(\mathbf{y})$ denotes the vector of lagged inputs and output,
  - $\varepsilon_f(\mathbf{y})$ is the approximation error.
  - $\mathbf{\Phi}(\mathbf{x}) = \{\mathbf{x}(k-d) \cdots \mathbf{x}(k-d-4), \mathbf{u}(k-d) \cdots \mathbf{u}(k-d-5)\}$

Neural Network Model

Neural Net Architecture



- input: lagged vector of past observations and current control actions
- repeat $\times$ 3
  - pass input through an lstm cell
  - followed by 30% dropout
- output will be control predictions directly fed as valve voltages

Closed-loop dynamics

- State closed loop dynamics is
  $\dot{\mathbf{y}} = \mathbf{A} + \mathbf{B}\Lambda(\hat{\mathbf{K}}_y^T \mathbf{y} + \mathbf{B}\Lambda(\hat{\mathbf{K}}_r^T \mathbf{r} + \varepsilon_f))$
  - $\mathbf{A}$ and Λ are unknown matrices.
  - sign of Λ is known
  - $\hat{\mathbf{K}}_y^T$ and $\hat{\mathbf{K}}_r^T$ are adaptation gains to be determined
- Generalized error state vector: $\mathbf{e}(k) = \mathbf{y}(k) - \mathbf{y}_m(k)$
  - with dynamics $\dot{\mathbf{e}}(k) = \mathbf{A}_m \mathbf{e}(k) + \mathbf{B}\Lambda[\tilde{\mathbf{K}}_r^T \mathbf{r} + \tilde{\mathbf{K}}_y^T \mathbf{y} - \varepsilon_f]$
  - $\mathbf{A}_m$ is Hurwitz and known, $\mathbf{B}$ is known.
  - $\mathbf{y}_m$ is assumed to be a linear model-following model of the form, $\dot{\mathbf{y}}_m = \mathbf{A}_m \mathbf{y}_m + \mathbf{B}\,\mathbf{r}$

Lyapunov Redesign

- **Theorem:**
  - Given correct choice of adaptive gains $\hat{\mathbf{K}}_y$ and $\hat{\mathbf{K}}_r$, the error state vector, $\mathbf{e}(k)$ with closed loop time derivative $\dot{\mathbf{e}}$, is **uniformly ultimately bounded**, and the state $\mathbf{y}$ will converge to a neighborhood of $\mathbf{r}$.
- **Proof**:
  - Choose a Lyapunov function candidate $\mathbf{V}$ in terms of the generalized error state space $\mathbf{e}$, gains, $\tilde{\mathbf{K}}_y^T$, $\tilde{\mathbf{K}}_r^T$, and parameter error $\varepsilon_f(\mathbf{y}(k))$ space

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

Stability proof

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

$$= \left[\mathbf{A}_m \mathbf{e} + \mathbf{B}\mathbf{\Lambda}[\Delta \hat{\mathbf{K}}_r^T \mathbf{r} + \Delta \hat{\mathbf{K}}_x^T \mathbf{x}]\right]^T \mathbf{P}\mathbf{e} + \ldots$$

$$\mathbf{e}^T \mathbf{P} \left[\mathbf{A}_m \mathbf{e} + \mathbf{B}\mathbf{\Lambda}[\Delta \hat{\mathbf{K}}_r^T \mathbf{r} + \Delta \hat{\mathbf{K}}_x^T \mathbf{x}]\right] + \ldots$$

$$2\,\mathbf{tr}(\Delta \mathbf{K}_x^T \Gamma_x^{-1} \dot{\hat{K}}_x |\mathbf{\Lambda}|) + 2\,\mathbf{tr}(\Delta \mathbf{K}_r^T \Gamma_r^{-1} \dot{\hat{\mathbf{K}}}_r |\mathbf{\Lambda}|)$$

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,

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

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

- 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

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

Control gains

Therefore,

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

- $\lambda_{low}, \lambda_{high} \equiv$ minimum and maximum characteristic roots of Q and $\boldsymbol{\Lambda}$ respectively.
- $\dot{\mathbf{V}}(\cdot)$ is thus negative definite outside the compact set
- $\chi = \left(\mathbf{e} : \|\mathbf{e}\| \leq \frac{2\|\mathbf{P}\mathbf{B}\|\lambda_{high}(\boldsymbol{\Lambda})\varepsilon_{max}(\mathbf{y})}{\lambda_{low}(\mathbf{Q})}\right)$
- thus, we conclude that the error $\mathbf{e}$ is uniformly ultimately bounded.
    - i.e. $\mathbf{y}(t) \to 0$ as $t \to \infty$

Experimental Results

- Head 3-DOF pose made up of tuples $\{z(k), \theta(k), \phi(k)\}$
  - *i.e.* roll, pitch and yaw
  - sample from the parameters of the network
- Set $\hat{f}(\cdot)$ to the fully connected layer of samples from the network
- Control law is implemented on the ROS middleware
- Gains $\hat{\mathbf{K}}_y$ and $\hat{\mathbf{K}}_r$ found by solving the ODEs iteratively
  - using a single step of the integral of the solutions to $\dot{\hat{\mathbf{K}}}_y(t)$, $\dot{\hat{\mathbf{K}}}_r(t)$
  - implemented using Runge-Kutta Dormand-Prince 5 ODE-solver available in the Boost C++ libraries

Results

- Sample from trained network parameters
- Set $\hat{f}(\cdot)$ to fully connected layer of network
- $\mathbf{y}_m$ is computed based on
  - $\mathbf{y}_m(t) = e^{\mathbf{A}_m t}\mathbf{y}_m(0) + \int_0^t e^{\mathbf{A}_m(t-\tau)}\mathbf{B}_m\mathbf{r}(\tau)d\tau$
- Set $\mathbf{y}_m(0) = \mathbf{y}(0)$ at t = 0
- For a nonnegative $\mathbf{Q}$ and a positive definite $\mathbf{P}$,
  - the pair $(\mathbf{Q}, \mathbf{A}_m)$ will be observable (LaSalle's theorm)
  - so that the dynamical system is globally asymptotically stable.
- Pick a positive definite, $\mathbf{Q} = diag(100, 100, 100)$ for the dissipation energy
- Set $\boldsymbol{\Lambda} = \mathbf{I}_{3\times 3}$

Results

- Solving the general form of the lyapunov equation, we have

$$\mathbf{P} = \begin{bmatrix} -\frac{170500}{2668} & 0 & 0 \\ 0 & -\frac{170500}{2668} & 0 \\ 0 & 0 & -\frac{170500}{2668} \end{bmatrix}$$

- Solenoid valves operate in pairs
  - two valves create a difference in air mass within each IAB at any given time
  - set

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

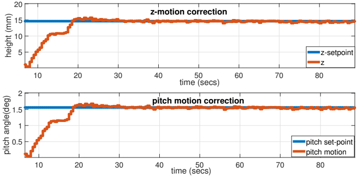
- B maps to the 3-axes controllers

$$\begin{bmatrix} u_z & u_\theta & u_\psi \end{bmatrix}^T$$

- non-zero terms are the max. duty-cycle to valves based on the software configuration of the NI RIO PWM generator

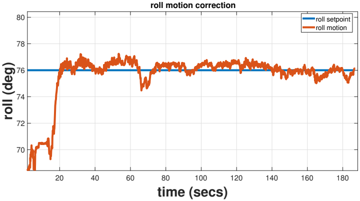
Results

- Performance of the controller when commanded to move head from $[z, \theta, \phi]^T = [2.5mm, .25^o, 35^o]^T$ to $[14mm, 1.6^o, 45^o]^T$.



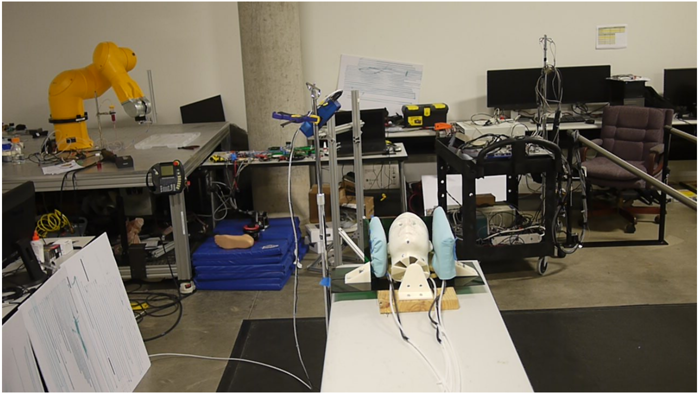
- strong steady-state convergence along z and pitch axes
- 20 second rise time

- Performance on head roll angle



- Need for separation of coupled dynamics
- Results show effectiveness of control law along the physically realizable axes of motions

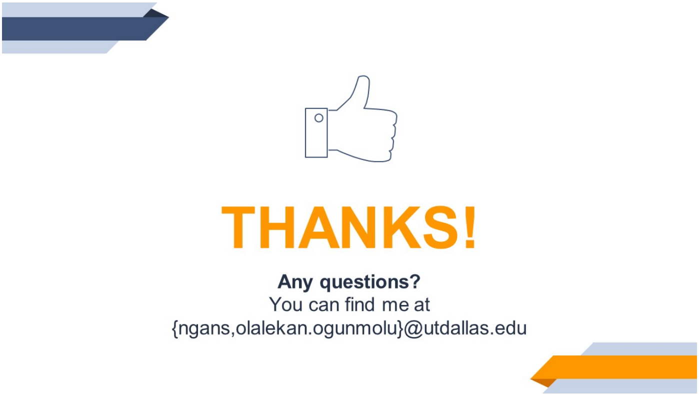
Video of Results



Results and Analysis

- Desired actuation axes coupled
  - there is a limited reachable space with the IABs
  - perform open-loop experiments to ensure pose is feasible
- Performance of the controller when commanded to move the head from $[z, \theta, \phi]^T = [2.5mm, .25^o, 35^o]^T$ to $[14mm, 1.6^o, 45^o]^T$.
  - strong steady-state convergence along z and pitch axes
  - 20 second rise time

Thanks!


THANKS!
Any questions?
You can find me at
{ngans,olalekan.ogunmolu}@utdallas.edu

- Cerviño Laura I., et al. "Frame-less and mask-less cranial stereotactic radiosurgery: a feasibility study." Physics in medicine and biology 55.7 (2010): 1863.

Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.

Liu, X., Belcher, A.H., Grelewicz, Z. and Wiersma, R.D., 2015, July. Robotic stage for head motion correction in stereotactic radiosurgery. In American Control Conference (ACC), 2015 (pp. 5776-5781). IEEE.

Ostyn, M., Dwyer, T., Miller, M., King, P., Sacks, R., Cruikshank, R., Rosario, M., Martinez, D., Kim, S. and Yeo, W.H., 2017. An electromechanical, patient positioning system for head and neck radiotherapy. Physics in Medicine and Biology.

Xing L. et. al., 2000. Dosimetric effects of patient displacement and collimator and gantry angle misalignment on intensity modulated radiation therapy. Radiotherapy and Oncology, 56(1), pp.97-108.