Modeling and Control of Unstable Physical Human-Machine Interactions: A Rider-Bikebot Example

Professor Jingang Yi

Department of Mechanical and Aerospace Engineering
Rutgers, The State University of New Jersey
Piscataway, New Jersey 08854

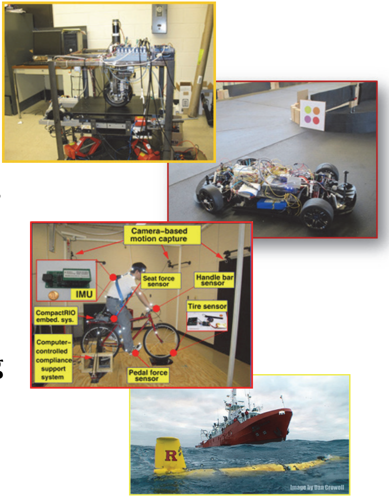
April 16, 2018

Research Interests

- **Research Areas**
  - Applied control theory (nonlinear, adaptive, and hybrid control systems)
  - Mechatronic systems
  - Design and automation
  - Dynamic systems
- **Applications Areas**
  - Robotics and vehicle systems
  - Biomedical and biological systems
  - Civil infrastructure, transportation, and oceanic systems
  - Micro- and nano-manufacturing systems

Recent Research Projects

1. **Robotic/vehicular systems**
   - Autonomous robots and vehicles
   - Contact modeling/tactile sensing
   - Underwater robotics
2. **Physical human-robot interactions**
   - Human-bikebot interactions
   - Robotic assistive slip-and-fall
   - Wearable robotic assistive devices
3. **Automation science & engineering**
   - Civil infrastructure automation
   - Micro-/nano-manufacturing automation



Autonomous Motorcycles/Bicycles

Why Are We Interested in Rider-Bicycle Interactions?

Biking but no walking

- Snijders and Bloem, "Freezing of gait", New England J. of Medicine, vol. 362, e46, 2010.

Rutgers rehabilitation bicycle

Motivation

- Rider-bicycle (or bicycle-based robot, i.e., bikebots) interaction is used as a new paradigm to study unstable physical human-machine interactions (pHMI)
  - Physically unstable, especially under a slow moving velocity
  - Coordinated multi-limbs and body movements for balancing
  - Multiple contact points for complex interactions and constraints
- Bicycling can potentially serve as a rehabilitation tool for postural balance disabilities
  - Possibly treating Parkinson's disease patients (Aerts *et al*. (2011) and Ridgel *et al*. (2009))

Motivation

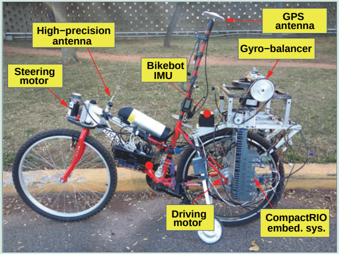
- Rider-bicycle (or bicycle-based robot, i.e., bikebots) interaction is used as a new paradigm to study unstable physical human-machine interactions (pHMI)
  - Physically unstable, especially under a slow moving velocity
  - Coordinated multi-limbs and body movements for balancing
  - Multiple contact points for complex interactions and constraints
- Bicycling can potentially serve as a rehabilitation tool for postural balance disabilities
  - Possibly treating Parkinson's disease patients (Aerts *et al.* (2011) and Ridgel *et al.* (2009))

Challenges

- Capturing and modeling high-dimensional pHMI
- Analyzing and quantifying the human balance motor skills in pHMI
- Tuning and controlling the pHMI

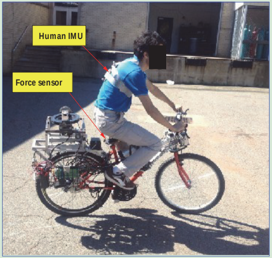
Bikebot Systems Development

Autonomous Bikebot System



Actuators & Sensors

Rider-Bike System



Actively Controlled Physical Human-Bicycle Interactions

- A flywheel gyro-balancer to create perturbation balancing torques
- Independently controlled steering and pedaling mechanisms
- Wearable/onboard sensors to estimate the human and bicycle poses

Objectives

1. Developing a modeling framework to capture the high-dimensional human motion in pHMI with applications to pose estimation
2. Quantifying human balance motor skills and comparing with autonomous control design
3. Stability and balance control of the physical rider-bicycle interactions

Objectives

1. Developing a modeling framework to capture the high-dimensional human motion in pHMI with applications to pose estimation
2. Quantifying human balance motor skills and comparing with autonomous control design
3. Stability and balance control of the physical rider-bicycle interactions

Outline

- **Part I:** Physical-learning modeling framework of the physical rider-bikebot interactions
- **Part II:** Quantifying human balance motor skills and autonomous control design
- **Part III:** Stability and control of the rider-bikebot interactions

Part I

Physical-learning modeling framework of physical rider-bicycle interactions

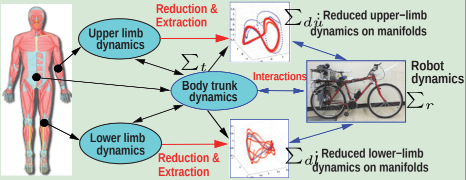
Background

- **Physical human-machine interactions (pHMI) play a critical role for many human-centered design, e.g., robotic assistive and rehabilitation devices**
- **Modeling and control of pHMI is challenging**
  - Highly-dimensional human motion and complex interactions
  - Lack of effective modeling tools
  - Sophisticated human-in-the-loop neuro-control

Background

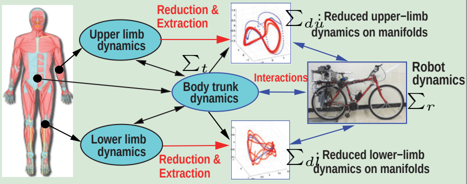
- **Physical human-machine interactions (pHMI) play a critical role for many human-centered design, e.g., robotic assistive and rehabilitation devices**
- **Modeling and control of pHMI is challenging**
  - Highly-dimensional human motion and complex interactions
  - Lack of effective modeling tools
  - Sophisticated human-in-the-loop neuro-control
- **Some existing approaches**
  - Physical principles-based modeling ➙ Complex for controller design
  - Learning from demonstration (e.g., data-driven models) ➙ Lack of physical interpretation

Physical-Learning Modeling Framework

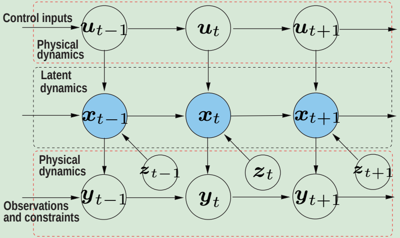


(a) Physical-learning modeling framework.

Physical-Learning Modeling Framework



(a) Physical-learning modeling framework.

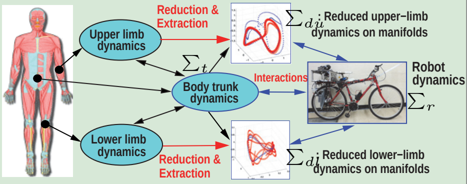


(b) The modeling schematic.

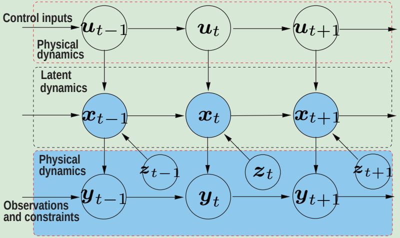
Basic Modeling Approach

- The model $\Sigma_{pl} = \Sigma_{tr} \times \Sigma_{du} \times \Sigma_{dl}$, Σ_{tr}: trunk-robot dynamics, Σ_{du}, Σ_{dl}: upper- and lower-limb dynamics

Physical-Learning Modeling Framework



(a) Physical-learning modeling framework.

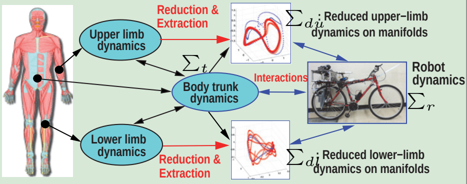


(b) The modeling schematic.

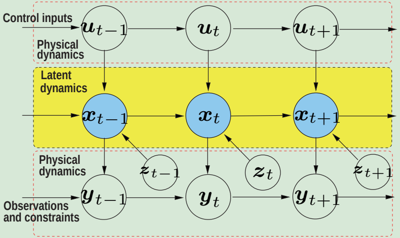
Basic Modeling Approach

- The model $\Sigma_{pl} = \Sigma_{tr} \times \Sigma_{du} \times \Sigma_{dl}$, Σ_{tr}: trunk-robot dynamics, Σ_{du}, Σ_{dl}: upper- and lower-limb dynamics
- Physical models for $\sum_{tr}$ → motions with physical sensing information

Physical-Learning Modeling Framework



(a) Physical-learning modeling framework.

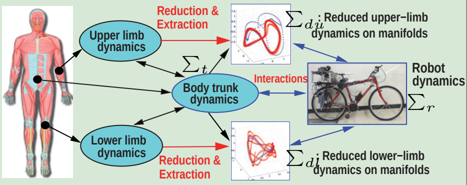


(b) The modeling schematic.

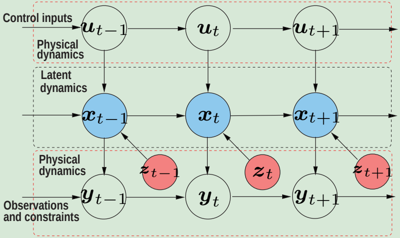
Basic Modeling Approach

- The model $\Sigma_{pl} = \Sigma_{tr} \times \Sigma_{du} \times \Sigma_{dl}$, Σ_{tr}: trunk-robot dynamics, Σ_{du}, Σ_{dl}: upper- and lower-limb dynamics
- Physical models for $\sum_{tr}$ ➙ motions with physical sensing information
- A learning model for $\sum_{du}/\sum_{dl}$ ➙ reduced-dimensional dynamics on manifolds without sensing

Physical-Learning Modeling Framework



(a) Physical-learning modeling framework.



(b) The modeling schematic.

Basic Modeling Approach

- The model $\Sigma_{pl} = \Sigma_{tr} \times \Sigma_{du} \times \Sigma_{dl}$, Σ_{tr}: trunk-robot dynamics, Σ_{du}, Σ_{dl}: upper- and lower-limb dynamics
- Physical models for $\sum_{tr}$ ➙ motions with physical sensing information
- A learning model for $\sum_{du}/\sum_{dl}$ ➙ reduced-dimensional dynamics on manifolds without sensing
- Integrating the physical-learning models with interactions/constraints

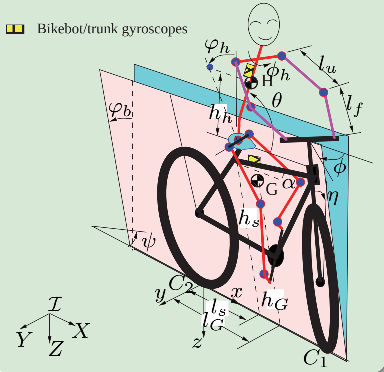
Physical Trunk-Bikebot Dynamics

- Generalized coordinates: the trunk $\boldsymbol{q}_t = [\varphi_h \ \theta \ \phi_h]^T$, bikebot $\boldsymbol{q}_r = \varphi_b$, and $\boldsymbol{q}_{tr} = [\boldsymbol{q}_t^T \ \boldsymbol{q}_r^T]^T$
- The trunk-bikebot dynamics is obtained by Lagrange equations as

$$\boldsymbol{M}(\boldsymbol{q}_{tr})\ddot{\boldsymbol{q}}_{tr} + \boldsymbol{C}(\boldsymbol{q}_{tr}, \dot{\boldsymbol{q}}_{tr})\dot{\boldsymbol{q}}_{tr} + \boldsymbol{G}(\boldsymbol{q}_{tr}) = \boldsymbol{u}_t$$

- Input $\boldsymbol{u}_t = \left[(\sec^2 \varphi_b) g u_s - \tau_h \quad \tau_h \quad \tau_\theta\right]^T$, τ_h and τ_θ: torques applied by the rider to the trunk

Schematic of rider-bikebot systems



- Nonholonomic constraint at rear wheel contact point C_2
- Velocity v_{rx} and yaw angle ψ as time-varying model parameters
- Steering control $u_\psi := \ddot{\psi} = \frac{v_r \, c_\xi}{l \, c_\varphi}\left(\sec^2_\phi \dot{\phi} + \tan\phi \tan\varphi \dot{\varphi}\right) + \frac{\dot{v}_r \tan\phi \, c_\xi}{l \, c_\varphi}$.

Learning Model for Limb Motion

- **Key idea:** high-dimensional human joint angle motion $\boldsymbol{y} \in \mathbb{R}^D$ to low-dimensional **latent space dynamics** $\boldsymbol{x} \in \mathbb{R}^d$, $D \gg d$
  - Linear latent dynamics not work for human motion
  - Commonly used dimension reduction method (e.g., principal component analysis (PCA), locally linear embedding (LLE))
  - The latent dynamics can incorporate and integrate with physical models

Learning Model for Limb Motion

- **Key idea:** high-dimensional human joint angle motion $\boldsymbol{y} \in \mathbb{R}^D$ to low-dimensional **latent space dynamics** $\boldsymbol{x} \in \mathbb{R}^d$, $D \gg d$
  - Linear latent dynamics not work for human motion ➝ Need nonlinear latent dynamics
  - Commonly used dimension reduction method (e.g., principal component analysis (PCA), locally linear embedding (LLE)) ➝ Cannot preserve the physical interpretation
  - The latent dynamics can incorporate and integrate with physical models ➝ Interconnect as inputs and constraints to the model

Learning Model for Limb Motion

- **Key idea:** high-dimensional human joint angle motion $\boldsymbol{y} \in \mathbb{R}^D$ to low-dimensional **latent space dynamics** $\boldsymbol{x} \in \mathbb{R}^d$, $D \gg d$
  - Linear latent dynamics not work for human motion ➙ Need nonlinear latent dynamics
  - Commonly used dimension reduction method (e.g., principal component analysis (PCA), locally linear embedding (LLE)) ➙ Cannot preserve the physical interpretation
  - The latent dynamics can incorporate and integrate with physical models ➙ Interconnect as inputs and constraints to the model

Main Approach

- **Gaussian process dynamic model (GPDM)** for low-dimensional limb motion with inputs/constraints from the physical model
- A novel **axial linear embedding (ALE)** dimensional-reduction method preserving physical interpretation

Nonlinear Learning Model for Human Limb Motion

- **Gaussian process dynamical model (GPDM):** A nonlinear dynamics between the high-dimensional joint angles $\boldsymbol{y} \in \mathbb{R}^D$ and the low-dimensional latent variables $\boldsymbol{x} \in \mathbb{R}^d$ $(d \ll D)$ for limb motion

  $$\delta \boldsymbol{x}(k) = \boldsymbol{x}(k) - \boldsymbol{x}(k-1) = \boldsymbol{f}(\boldsymbol{x}(k-1), \boldsymbol{u}_h(k-1), \boldsymbol{\alpha}) + \boldsymbol{w}_p,$$
  $$\boldsymbol{y}(k) = \boldsymbol{g}(\boldsymbol{x}(k), \boldsymbol{\beta}) + \boldsymbol{w}_o,$$

  $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ are parameters, input $\boldsymbol{u}_h(k-1)$ from the trunk-robot subsystems.
- $\boldsymbol{X} = \{\boldsymbol{x}(k)\}^N$, $\boldsymbol{Y} = \{\boldsymbol{y}(k)\}^N$ and $\boldsymbol{U} = \{\boldsymbol{u}_h(k)\}^N$ as training data sets, estimate $\boldsymbol{f}$ and $\boldsymbol{g}$ by identifying $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and maximizing a-posterior distribution

  $$P(\boldsymbol{X}, \boldsymbol{\alpha}, \boldsymbol{\beta} | \boldsymbol{Y}, \boldsymbol{U}) \propto P(\boldsymbol{Y} | \boldsymbol{X}, \boldsymbol{\beta}) P(\boldsymbol{X} | \boldsymbol{U}, \boldsymbol{\alpha}) P(\boldsymbol{\alpha}) P(\boldsymbol{\beta}).$$

Dimensional Reduction and Initialization – ALE Algorithm

- Axial linear embedding (ALE) for
  - Preserving the physical interpretation of the latent variables
  - Maintaining performance and avoiding trapping at local minimums
- Limb motion is influenced by trunk and steering movements
- Determine three limb motion **primitives** by three **templates** of trunk-robot motion $\boldsymbol{q}_{tr}$ (`Lines 2-3`)
- Arbitrary limb motion is decomposed along three-motion-primitive directions in the latent space (`Lines 4-6`)

ALE Algorithm Flowchart

Algorithm 1: Axial Linear Embedding (ALE)

for $j = 1$ *to* n_{rt} **do**

1 $\quad \mathbf{q} = \mathbf{q}^e_{rt}$; $\mathbf{y} = \mathbf{y}^e$;

2 $\quad$ Perturb $\mathbf{q}$ along the jth template of $\mathscr{T}_{rt}$ and obtain limb motion $\{\mathbf{y}_l\}_{\mathbf{q}^j_{rt}}$;

3 $\quad \{\mathbf{x}_l\}_{\mathbf{q}^j_{rt}} = \mathbf{0}$; $\{\mathbf{x}^j_l\}_{\mathbf{q}^j_{rt}} = \mathrm{PCA}_1(\{\mathbf{y}_l\}_{\mathbf{q}_j})$;

end

for $k = 1$ *to* N **do**

4 $\quad$ Find M points $\mathbf{y}_{k_i}$ s.t. $\|\mathbf{y}_{k_i} - \mathbf{y}(k)\|_2$ are the M smallest value for $\mathbf{y}_{k_i} \in \cup_{j=1,\cdots,n_{rt}} \{\mathbf{y}_l\}_{\mathbf{q}^j_{rt}}$;

5 $\quad \mathbf{w}_{k_i} = \arg\min_{\mathbf{w}_{k_i}} \|\mathbf{y}(k) - \Sigma^M_{i=1} \mathbf{w}_{k_i} \mathbf{y}_{k_i}\|^2_2$, $\Sigma^M_{i=1} \mathbf{w}_{k_i} = 1$;

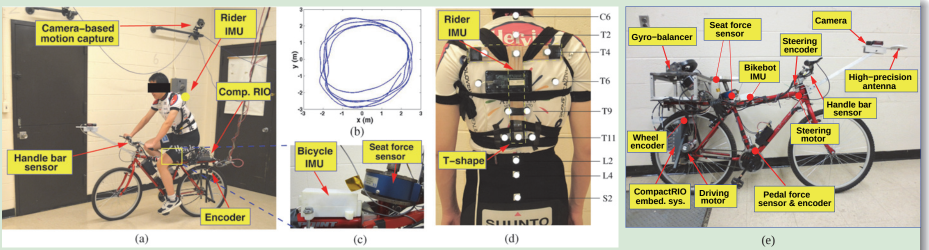
6 $\quad \mathbf{x}(k) = \Sigma^M_{i=1} \mathbf{w}_{k_i} \mathbf{x}_{k_i}$, $\mathbf{x}_{k_i}$ is the latent coordinate of $\mathbf{y}_{k_i}$;

end

Modeling Application: Limb Motion Estimation

- Each upper-limb is modeled as five degree-of-freedom (DOF): shoulder joint angles (y_1, y_2, y_3) (left)/(y_6, y_7, y_8) (right) and elbow joint angles (y_4, y_5) (left)/(y_9, y_{10}) (right), i.e., $D = 10$.
- The upper-limb poses primarily influenced by three trunk-bikebot motion templates: trunk roll, pitch motions and the bikebot steering motion, i.e., $d = 3$.
- Latent dynamics input $\boldsymbol{u}_h = [\phi(t)\ \dot{\phi}(t)\ \boldsymbol{q}_t^T(t)\ \dot{\boldsymbol{q}}_t^T(t)]^T$
- Only trunk and bikebot gyroscope sensors are used in the design (No wearable sensors on limbs)
- Extended Kalman filter (EKF) is used to fuse the gyroscope measurements and a set of geometric and dynamic constraints

Experiments and Motion Kinematics/Constraints

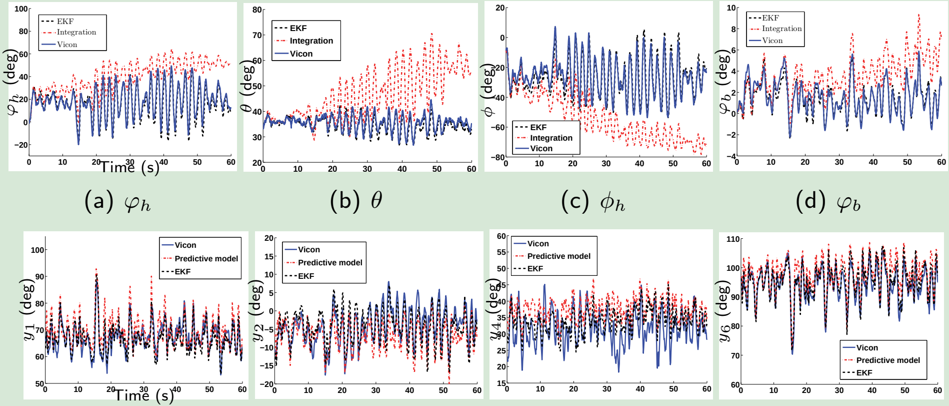


(a)-(b) Riding experiment and trajectory. (c)-(e) Bikebot/trunk gyroscopes and optical markers.

Motion Kinematics and Rider-Bikebot Geometric/Dynamic Constraints

- **Gyroscopes motion**: $\dot{\varphi}_b = [\mathrm{c}_\alpha \;\; 0 \;\; s_\alpha]\boldsymbol{\omega}_r$, $\dot{\boldsymbol{q}}_t = \boldsymbol{f}_2(\boldsymbol{q}_{tr};\boldsymbol{\omega}_r,\boldsymbol{\omega}_t)$
- **Dynamic constraint**: first 2 equations in dynamics $z_1(\boldsymbol{q}_{tr},\dot{\boldsymbol{q}}_{tr},\ddot{\boldsymbol{q}}_{tr}) = 0$
- **Geometric constraints**: human anatomical and pHMI interactions
  - Trunk, limb and bikebot forms closed structure $\boldsymbol{z}_2(\boldsymbol{q}_t,\boldsymbol{y}) = \boldsymbol{0} \in \mathbb{R}^6$
- **EKF design**: state: $\boldsymbol{v} = [\boldsymbol{q}_{tr}^T \; \boldsymbol{x}^T]^T \in \mathbb{R}^7$, output: $\boldsymbol{y} \in \mathbb{R}^{10}$ and $\hat{\boldsymbol{q}}_{tr}$.

Results – Single-Subject Experiment



(a) φ_h (b) θ (c) ϕ_h (d) φ_b

(e) y_1 (f) y_2 (g) y_4 (h) y_6

Comparison among EKF, IMU direct integration and the ground truth.

Results – Single-Subject Experiment

Root mean square (RMS) errors (in degs) of EKF design

GPDM	φ_h	θ	ϕ_h	φ_b	y_1	y_2	y_3
ALE	5.0 ± 1.4	1.4 ± 0.5	5.1 ± 0.3	0.5 ± 0.1	2.2 ± 0.3	2.8 ± 0.1	4.9 ± 1.1
PCA	5.2 ± 1.9	1.5 ± 0.5	5.1 ± 0.6	0.5 ± 0.1	2.3 ± 0.4	2.9 ± 0.5	4.6 ± 0.5
LLE	5.4 ± 0.1	1.5 ± 0.3	5.8 ± 1.2	0.5 ± 0.2	3.3 ± 1.2	3.3 ± 0.8	6.1 ± 1.8
IMU	3.2	4.6	3.5	0.7	4.9	4.7	3.0

GPDM	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
ALE	4.5 ± 0.4	4.2 ± 0.8	1.9 ± 0.7	3.9 ± 0.3	4.1 ± 0.5	5.0 ± 1.2	3.1 ± 0.3
PCA	5.0 ± 0.2	5.3 ± 0.9	2.4 ± 0.8	4.5 ± 0.9	4.7 ± 0.5	5.1 ± 0.8	4.1 ± 0.4
LLE	9.1 ± 0.3	7.3 ± 0.5	2.8 ± 0.6	5.6 ± 2.5	8.5 ± 3.1	6.2 ± 1.3	5.0 ± 1.5
IMU	4.7	4.8	4.7	5.2	4.3	4.8	5.2

Observations

- The ALE algorithm outperforms both the PCA and LLE
- The model predictions have the similar error level as using wearable IMUs

Results – Five-Subject Experiments

RMS errors (in degs) over 1-minute indoor/outdoor experiments

	φ_h	θ	ϕ_h	φ_b	y_1	y_2	y_3
Indoor	6.4 ± 1.8	1.5 ± 0.2	7.6 ± 3.1	0.6 ± 0.2	2.6 ± 0.3	4.2 ± 0.9	5.9 ± 0.9
Outdoor	7.7 ± 1.0	2.0 ± 0.7	9.6 ± 0.4	0.8 ± 0.2	2.9 ± 0.3	4.9 ± 0.7	6.4 ± 0.9

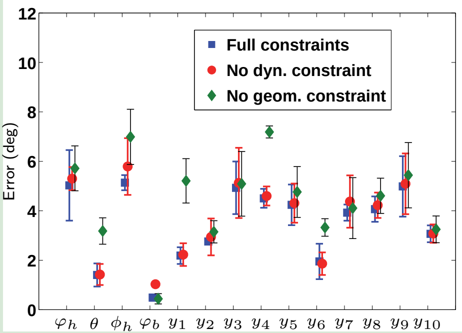
	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
Indoor	5.3 ± 0.8	5.1 ± 0.7	2.8 ± 0.5	4.4 ± 1.2	3.9 ± 0.2	5.6 ± 0.1	3.0 ± 0.4
Outdoor	6.0 ± 2.0	5.2 ± 0.5	2.9 ± 0.4	4.7 ± 0.8	4.5 ± 1.3	6.3 ± 1.1	3.1 ± 0.5

Remarks

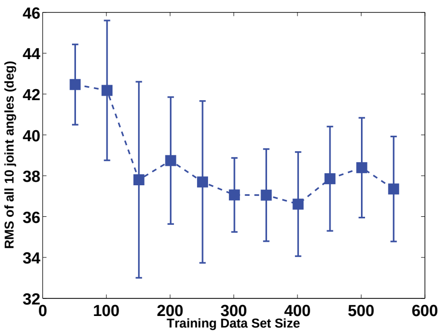
- The results are still comparable with those by wearable sensors [a]
- Indoor experiment results are slightly better than these of outdoor

[a] Zhang *et al.*, IROS'2014; Lu *et al.*, IROS'2014.

Result Analysis



(a)



(b)

(a) The role of the pHMI constraints. (b) The role of the training data size.

Observations

- Dynamic/geometric pHMI constraints enhances the estimation results
- A small size of training data is enough for building the model

Summary

1. **Built a nonlinear learning model to capture the high-dimensional limb motion on the lower-dimensional manifolds**
2. **Proposed an integrated physical-learning model for studying pHMI interactions**
   - The compact model is attractive for pHMI control design
   - The ALE algorithm preserves the physical meaning of the latent variables
3. **Validated and demonstrated pose-estimation performance by extensive indoor/outdoor experiments**

Part II

Quantifying human balance motor skills and autonomous control design

Objectives and Approaches

Objectives

- Analyze the contributions of balance by upper-body movement and steering actuation
- Define metrics to quantify tracking and balancing performance
- Model human path-following strategy and compare with autonomous control

Objectives and Approaches

Objectives

- Analyze the contributions of balance by upper-body movement and steering actuation
- Define metrics to quantify tracking and balancing performance
- Model human path-following strategy and compare with autonomous control

Approaches

- Rider-bikebot dynamics and external/internal convertible (EIC) structure
- Define and use the balance equilibrium manifold (BEM)
- BEM-based path-following controller design and comparison with human rider control
- Introduce BEM-based balance and path-following metrics

Nearly EIC Nonlinear Dynamic Systems

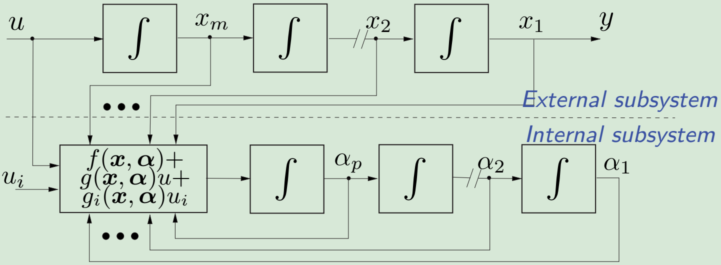
- An $n(=m+p)$-dimensional nonlinear system is called in a *nearly external/internal convertible (EIC) form* if the system is of the form

$$\Sigma: \begin{cases} \dot{x}_i = x_{i+1},\ \dot{x}_m = u, i = 1, \cdots, m-1, \\ \dot{\alpha}_j = \alpha_{j+1}, \\ \dot{\alpha}_p = f(\boldsymbol{x}, \boldsymbol{\alpha}) + g(\boldsymbol{x}, \boldsymbol{\alpha})u + g_i(\boldsymbol{x}, \boldsymbol{\alpha})u_i, \\ y = x_1,\ j = 1, \cdots, p-1, \end{cases}$$

  with $u, u_i, y \in \mathbb{R}$, $\boldsymbol{x} = [x_1 \ \cdots \ x_m]^T \in \mathbb{R}^m$ and $\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_p]^T \in \mathbb{R}^p$.
- The *external* and *internal* subsystems are
  $\Sigma_{\text{ext}}:\ \dot{x}_i = x_{i+1},\ \dot{x}_m = u,\ i = 1, \cdots, m-1$, and
  $\Sigma_{\text{int}}:\ \dot{\alpha}_i = \alpha_{i+1},\ \dot{\alpha}_p = f(\boldsymbol{x}, \boldsymbol{\alpha}) + g(\boldsymbol{x}, \boldsymbol{\alpha})u + g_i(\boldsymbol{x}, \boldsymbol{\alpha})u_i$ for
  $i = 1, \cdots, p-1$.
- System Σ is **convertible** because Σ_{int} is nearly converted to Σ_{ext} (with additional u_i), and Σ_{ext} is nearly converted to Σ_{int} (with u_i term) under $u = g(\boldsymbol{x}, \boldsymbol{\alpha})^{-1}\left[v - f(\boldsymbol{x}, \boldsymbol{\alpha})\right]$

Nearly EIC Property for the Rider-Bikebot System

Schematic of a nearly external/internal convertible system.



- Nearly EIC form for the rider-bikebot system

$$\Sigma_{\text{ext}} : \boldsymbol{r}_{C_2}^{(3)} = \boldsymbol{u}_N,$$
$$\Sigma_{\text{int}} : \ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})[\boldsymbol{B}(\boldsymbol{q})\boldsymbol{R}_\psi^{-1}\boldsymbol{\Psi} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{u}_i + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{R}_\psi^{-1}\boldsymbol{u}_N].$$

- External subsystem: planar motion dynamics; Internal subsystem: balance dynamics.

BEM Under the Rider's Control

BEM under rider control

An 8-dimensional (X, Y)-subspace in $\mathbb{R}^8$ of Σ_{ext} under $\boldsymbol{u}^h$ and $\boldsymbol{\tau}$.

$$\mathcal{E}(\boldsymbol{u}^h, \boldsymbol{\tau}) = \left\{ (\boldsymbol{x}, \boldsymbol{\alpha}) \mid \boldsymbol{q}_e = \boldsymbol{q}_e(\dot{\psi}, v_r, \boldsymbol{u}^h, \boldsymbol{\tau}), \dot{\boldsymbol{q}} = \boldsymbol{0} \right\}, \tag{1}$$

where $\boldsymbol{x} = [\boldsymbol{r}_{C_2}^T \; \dot{\boldsymbol{r}}_{C_2}^T \; \ddot{\boldsymbol{r}}_{C_2}^T]^T$ and $\boldsymbol{\alpha} = [\boldsymbol{q}^T \; \dot{\boldsymbol{q}}^T]^T$.

Consider the internal (roll angles) equilibria, denoted as $\boldsymbol{q}_e$, by setting $\dot{\boldsymbol{q}} = \ddot{\boldsymbol{q}} = 0$ in the balance dynamics, are the solutions of the algebraic equation

$$\boldsymbol{F}(\boldsymbol{q}_e, \dot{\psi}, v_r, \boldsymbol{u}, \boldsymbol{\tau}) = \boldsymbol{0},$$

where

$$\boldsymbol{F}(\boldsymbol{q}, \dot{\psi}, v_r, \boldsymbol{u}, \boldsymbol{\tau}) := \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\tau} - \boldsymbol{C}_q(\boldsymbol{q}) - \boldsymbol{G}(\boldsymbol{q}).$$

Balance Effects by Body Movement and Steering Actuation

- **Aim**: Quantify the influence of the upper-body movement and the steering actuation on the balance task
- **Method**: Perturb the rider-bikebot systems around the BEM and compute the sensitivity factors
- Taking the total derivatives of $\boldsymbol{F}(\boldsymbol{q}, \dot{\psi}, v_r, \boldsymbol{u}, \boldsymbol{\tau})$, we obtain

$$dF_1 = \frac{\partial F_1}{\partial \varphi_b} d\varphi_b + \frac{\partial F_1}{\partial \varphi_h} d\varphi_h + \frac{\partial F_1}{\partial \dot{\psi}} d\dot{\psi} + \frac{\partial F_1}{\partial u_\psi} du_\psi.$$

For steering actuation:

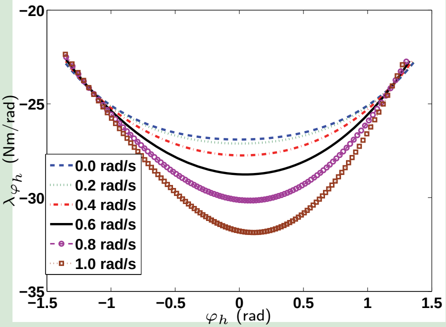
$$\lambda_\phi = \frac{\partial F_1}{\partial \phi} = -\frac{v_r^2 \, \mathrm{c}_\xi}{l \, \mathrm{c}_{\varphi_b} \, \mathrm{c}_\phi^2} \left(A_2 + A_4 \frac{\mathrm{c}_\xi \, \mathrm{s}_\phi}{2l \, \mathrm{c}_{\varphi_b} \, \mathrm{c}_\phi} \right).$$

For upper body leaning actuation:

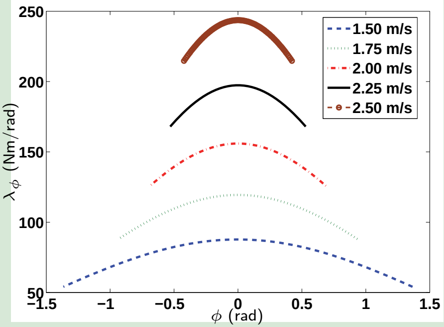
$$\frac{d\varphi_b}{d\varphi_h} = -\frac{M_{12}}{M_{11}} \Longrightarrow \lambda_{\varphi_h} = \left(-\frac{\partial F_1}{\partial \varphi_b} \frac{M_{12}}{M_{11}} + \frac{\partial F_1}{\partial \varphi_h} \right).$$

Balancing by Body Movement and Steering Actuation

Sensitivity calculation at different motion states



(a)



(b)

(a) λ_{φ_h} with varying yaw rate $\dot{\psi}$. (b) λ_ϕ with varying bikebot velocity v_r.

The torque generated by one unit of the steering angle is about five times than that by one unit of the upper-body leaning angle.

Riding Performance Metrics

- Quantify only the balancing performance by using the BEM concept

Balancing metric BM_1

$$BM_1 = F_1(\boldsymbol{q}, \dot{\psi}, v_r, \boldsymbol{u}, \boldsymbol{\tau}).$$

- Quantify both the balancing skills and the path-following performance

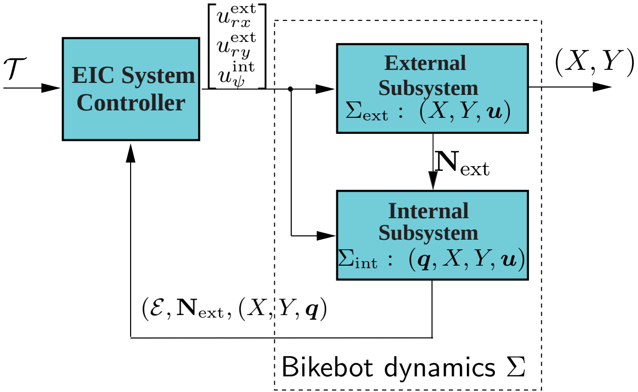
Balancing-tracking metric BM_2

$$BM_2 = E_p\left(\boldsymbol{e}_1\right) + E_q\left(\boldsymbol{e}_2\right) = \boldsymbol{e}_1^T \boldsymbol{M}_1 \boldsymbol{e}_1 + \boldsymbol{e}_2^T \boldsymbol{M}_2 \boldsymbol{e}_2.$$

where $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are the position and the roll angle error vectors, respectively, and $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are positive definite symmetric matrices obtained by solving the Lyapunov equations.

EIC-Based Path-Following Design

- The control goal is to follow the desired trajectory $\mathcal{T}$: $(X_d(t), Y_d(t))$ while keeping the platform balanced and stable
- A two-step control system design process: (1) trajectory tracking control design and (2) internal system stabilization design



Bikebot Autonomous Control w/o Assistive Gyro-Balancer

- Linear feedback control for path-following:

$$\boldsymbol{u}_N^{\text{ext}} = \boldsymbol{r}_d^{(3)} - b_2\ddot{\boldsymbol{e}}_p - b_1\dot{\boldsymbol{e}}_p - b_0\boldsymbol{e}_p,\ \boldsymbol{u}^{\text{ext}} = \boldsymbol{R}_\psi^{-1}\left(\boldsymbol{\Psi} + \boldsymbol{u}_N^{\text{ext}}\right),$$

  where $\boldsymbol{u}^{\text{ext}} = [u_r^{\text{ext}}\ u_\psi^{\text{ext}}]^T$ and $\boldsymbol{e}_p(t) = \boldsymbol{r}_{C_2} - \boldsymbol{r}_d$.
- Balancing equilibrium manifold (BEM) concept

BEM under the external trajectory-tracking control $\boldsymbol{u}^{\text{ext}}$

$$\mathcal{E} = \left\{(\boldsymbol{x}, \varphi_{be})\,\middle|\,\varphi_b = \varphi_{be}\left(\dot{\psi}, v_r, \boldsymbol{u}^{\text{ext}}, \varphi_w\right), \dot{\varphi_{be}} = \ddot{\varphi_{be}} = 0\right\}$$

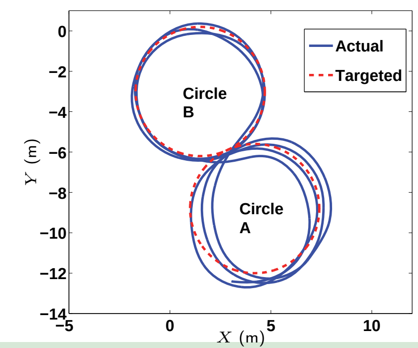
- Linear feedback control for the BEM stabilization

$$v_\psi^{\text{int}} = \bar{L}^2_{\mathbf{N_{ext}}}\varphi_{be} - a_1\dot{e}_{\varphi_b} - a_0 e_{\varphi_b}, u_\psi^{\text{int}} = g_\psi^{-1}(\varphi_b)(J_t v_\psi^{\text{int}} - f(\varphi_b)).$$

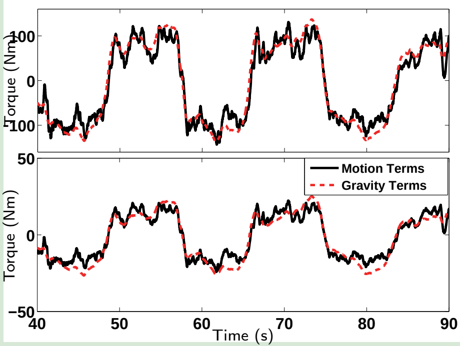
  where $e_{\varphi_b} = \varphi_b - \varphi_{be}$, $\dot{e}_{\varphi_b} = \dot{\varphi}_b - \dot{\varphi}_{be}$.
- The controller is: $\mathcal{C}:\ u_r = u_r^{\text{ext}}, u_\psi = u_\psi^{\text{int}}, u_w = 0.$

Experiments: Dynamics Model Validation

Experimental trajectory and model validation results



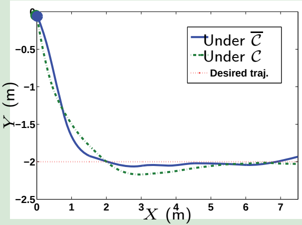
(a)



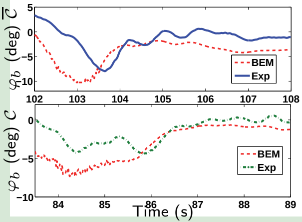
(b)

(a) Bikebot riding of an "8"-shape path-following experiments. (b) Validation of the rider-bikebot balancing subsystem dynamics.

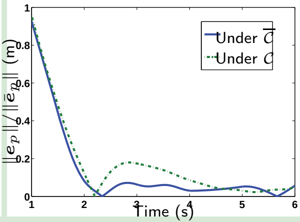
Autonomous Controllers: Straight-Line Trajectory



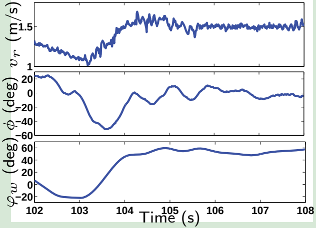
(a) Position trajectories



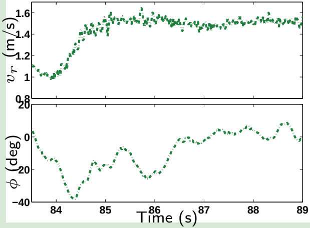
(b) φ_{be} tracking errors



(c) Tracking errors



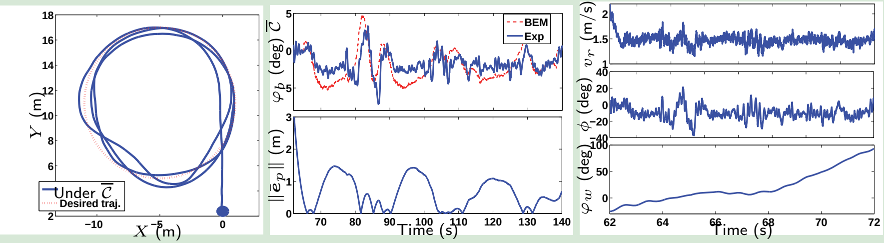
(d) Control inputs under $\bar{\mathcal{C}}$



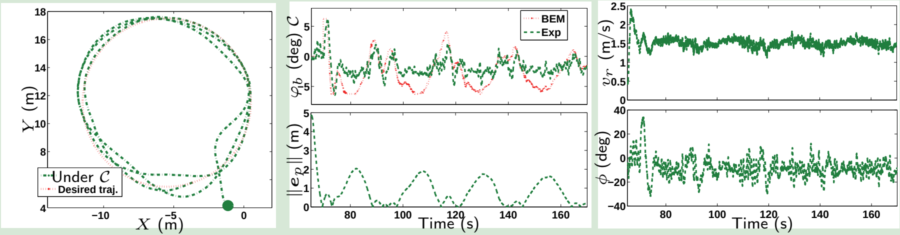
(e) Control inputs under $\mathcal{C}$

Autonomous controllers: $\bar{\mathcal{C}}$ – w/ gyro-balancer; $\mathcal{C}$ – w/o assistive gyro-balancer.

Autonomous Controllers: Circular Trajectory



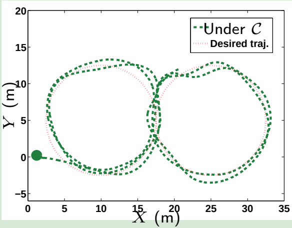
(a) Position trajectories (b) φ_{be} and tracking errors (c) Control inputs



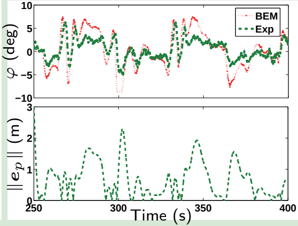
(d) Position trajectories (e) φ_{be} and tracking errors (f) Control inputs

(a)-(c) for controller $\bar{\mathcal{C}}$ and (d)-(f) for controller $\mathcal{C}$.

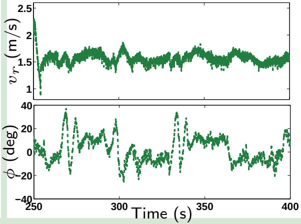
Autonomous & Human Experiments: "8"-shape Trajectory



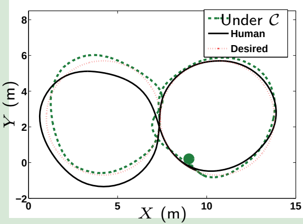
(a) Trajectory under $\mathcal{C}$



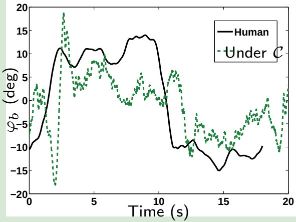
(b) φ_{be} and tracking errors



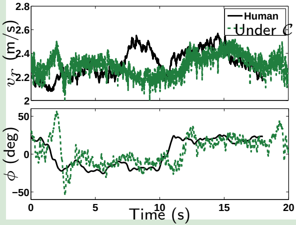
(c) Control inputs



(d) Human riding trajectory



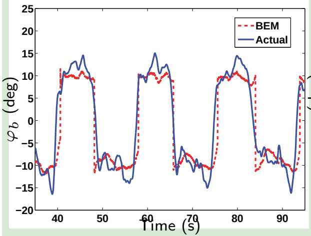
(e) Rider φ_b profile



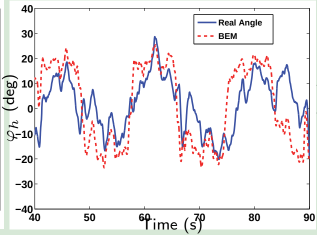
(f) Human rider control inputs

Rider Path-Following Experiments and Performance

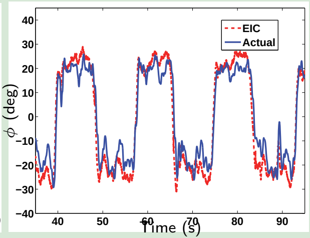
Rider-bikebot motion during the “8”-shape trajectory following



(a) Bikebot roll angle φ_b



(b) Rider trunk roll angle φ_h

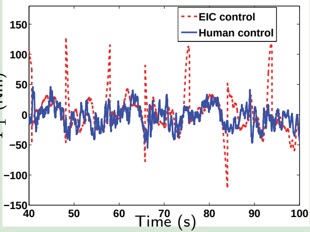


(c) Bikebot steering angle ϕ

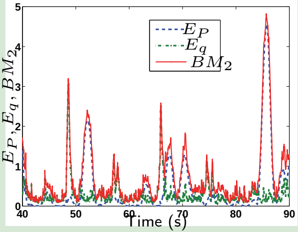
- Rider’s behavior closely tracks the desired equilibrium points (φ_b, φ_h)
- Similar behaviors between the human control and the EIC design

Rider Path-Following Performance

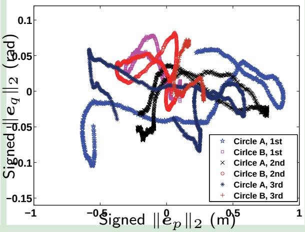
Comparison of the riding performance and metrics



(a) Metric BM_1



(b) Metric BM_2



(c) Error distributions

- The performance metric under the rider control is much smaller than that by the EIC control
- The overall errors in circle trajectory are much smaller than those in the "8"-shape path

Summary

1. Introducing the BEM concept to capture the human balance motor skills in interactions with bikebot
2. Designing EIC-based balance controller with stability analysis
3. Introducing BEM-based performance metrics to quantify the balance and path-following skills
4. Conducting extensive experiments to demonstrate and validate the control design and analyses

Part III

Stability and control of the physical rider-bikebot interactions

Objectives and Approaches

- **Analyze stability of rider-bicycle interactions under human neuro-balancing control**
  - Explain the clinical observation from control systems viewpoints
  - Provide possible design guidance for bikebot-assisted rehabilitation
  - Experimentally validate and demonstrate the human control models

Objectives and Approaches

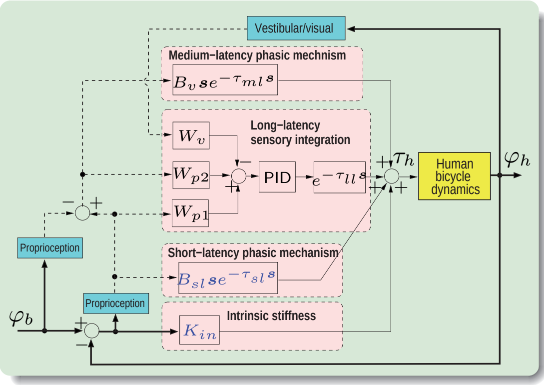
- **Analyze stability of rider-bicycle interactions under human neuro-balancing control**
  - Explain the clinical observation from control systems viewpoints
  - Provide possible design guidance for bikebot-assisted rehabilitation
  - Experimentally validate and demonstrate the human control models

Approaches

- Use a human neuro-balancing control model
- Stability analysis by the results of time-delay dynamical systems
- Conducting experiments for human control models estimation and the systems stability analyses

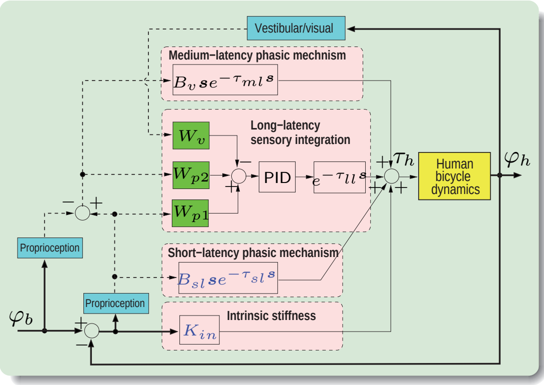
Human Neuro-Balancing Controller

- Capture sensorimotor mechanisms due to proprioception, vestibular and visual sensory



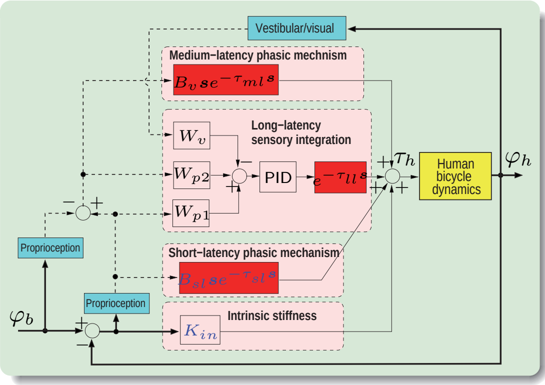
Human Neuro-Balancing Controller

- Capture sensorimotor mechanisms due to proprioception, vestibular and visual sensory
- W_{p1}, W_{p2} – proprioceptive weights; W_v – weight of the vestibular/visual sensory



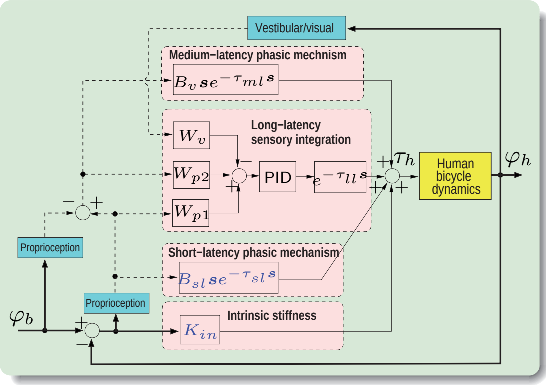
Human Neuro-Balancing Controller

- Capture sensorimotor mechanisms due to proprioception, vestibular and visual sensory
- W_{p1}, W_{p2} – proprioceptive weights; W_v – weight of the vestibular/visual sensory
- Three time delays: the short-, medium- and long-term phasic mechanisms



Human Neuro-Balancing Controller

- Capture sensorimotor mechanisms due to proprioception, vestibular and visual sensory
- W_{p1}, W_{p2} – proprioceptive weights; W_v – weight of the vestibular/visual sensory
- Three time delays: the short-, medium- and long-term phasic mechanisms

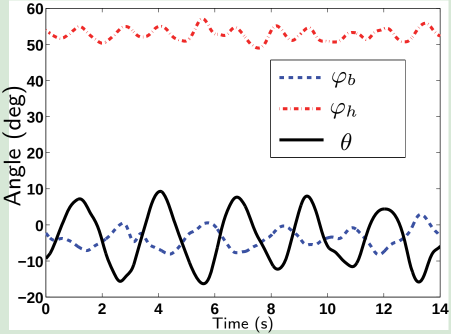


- The human neuro-balancing torque is (s: the Laplace operator)

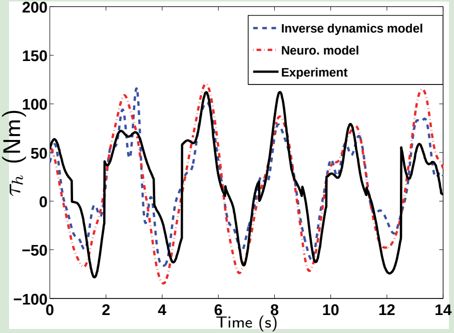
$$\boxed{\begin{aligned}\tau_h \; = \; & K_{in}(x_1 - x_3) + B_{sl}(x_2 - x_4)e^{-\tau_{sl}s} - B_v x_4 e^{-\tau_{ml}s} + \big(W_{p1}x_1 - \\ & W_t x_3\big)\left(K_p + K_d s + \frac{K_i}{s}\right) e^{-\tau_{ll}s},\end{aligned}}$$

Human Neuro-Balancing Control – Experiments

Rider-bicycle poses



Torque comparisons

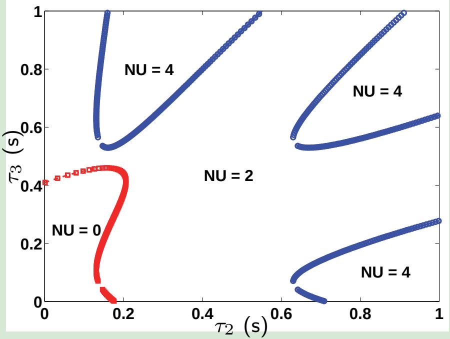


Indoor riding experiments

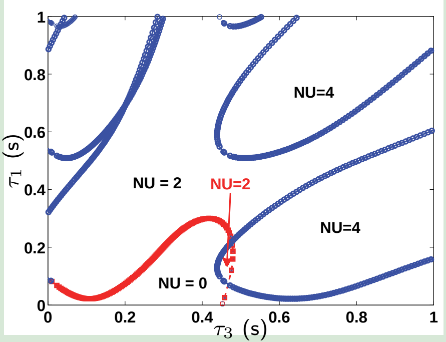
Stability Analysis of the Rider-Bicycle Systems

- The rider-bicycle dynamics under the human neuro-balancing control is a three-delay system
- Steering control is needed for a stable system. A proportional steering control $u_s := \tan\phi = k_{pd}\varphi_b$ is considered (Soudbakhsh et al. (2012))
- Stability charts to show the stability regions in the time-delay space

$\tau_1 = 21$ msec



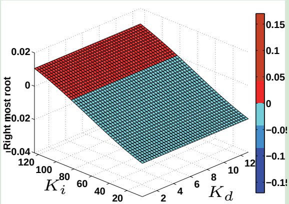
$\tau_2 = 131$ msec



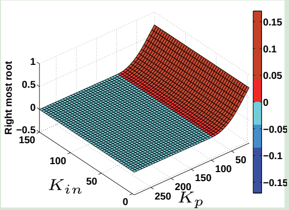
Clinical Implications and Discussions

- For PD patients (Kim et al. (2009), *J Neurophysiol.*)
  - Larger K_{in} and B_{sl} than healthy control subjects
  - Stability effect: a larger B_{sl} helps the subject stabilize the bicycle while the stability is insensitive to the stiffness K_{in}

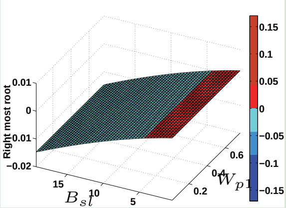
Neuro-PID gains



Stiffness and P-gain



Gain B_{sl} and weight W_{p1}



Explanation for the FOG clinical observation

- Larger damping gain in short-latency for PD/FOG patients helps stabilize bicycle riding
- Larger intrinsic stiffness does not de-stabilize the bicycle balancing

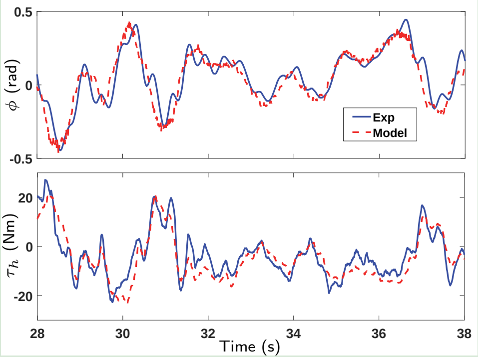
Human Balance Riding Modeling

Upper-body torque and steering angle control models

$$\tau_h(t) = k_{h0}\varphi_h(t) + k_{h1}\varphi_b(t-\tau_1) + k_{h2}\dot{\varphi}_b(t-\tau_2) + k_{h3}\varphi_h(t-\tau_1) + k_{h4}\dot{\varphi}_h(t-\tau_2)$$

$$\phi(t) = \frac{1}{v_r^2}\left[k_{b1}\varphi_b(t-\tau_3) + k_{b2}\dot{\varphi}_b(t-\tau_4) + k_{b3}\varphi_h(t-\tau_3) + k_{b4}\dot{\varphi}_h(t-\tau_4)\right]$$

Experimental validation

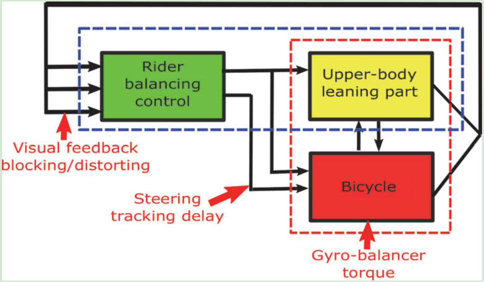


- Both control models follow the PD-structure with two time delays
- Simplified from the complete model in neuroscience literature
- The perturbed experimental data for model parameter estimation and validation

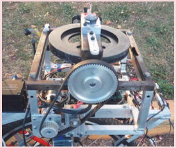
Human Balance Riding Experiments

Experiments with sensorimotor perturbation

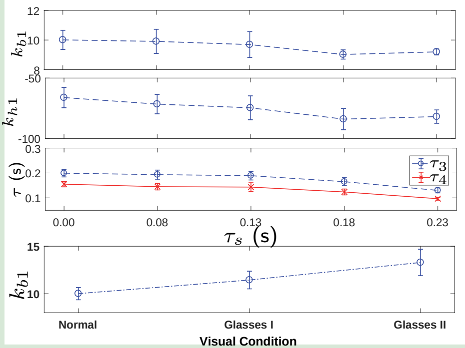
- Three types of sensorimotor disturbances
  - Randomly balance torque disturbance
  - Visual disturbance (partially)
  - Steering time delay disturbance τ_s
- Five subjects with 2-3 trails for each disturbance



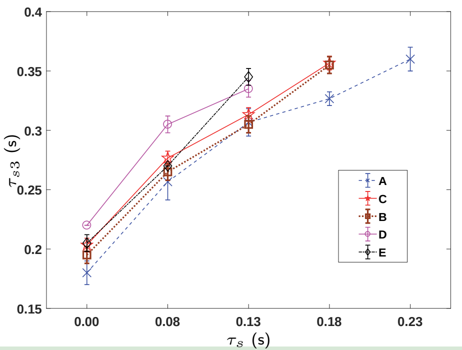
Disturbance sources



Experimental Results: Human Control Parameters



(a) Overall parameter changes

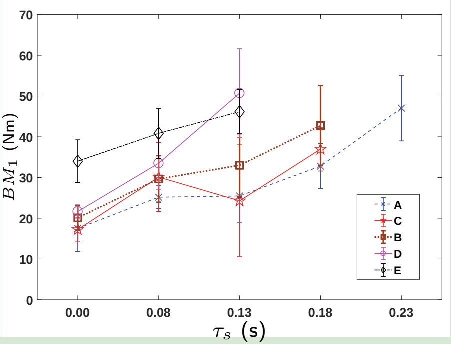


(b) Delay $\tau_{s3} = \tau_3 + \tau_s$

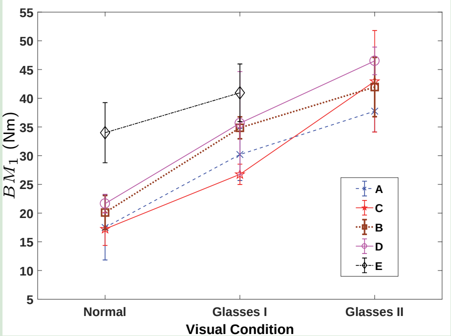
Observations

- Under disturbances, parameters k_{b1} (increasing) and τ_3 (decreasing) (steering) and k_{h1} (decreasing) (torque) change significantly
- Total steering delay $\tau_{s3} \leq 0.36$ sec (human balance delay limit)

Experimental Results: Human Balance Metrics



(a) BM_1 vs. delay τ_s

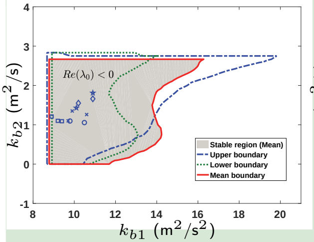


(b) BM_1 vs. visual disturbance

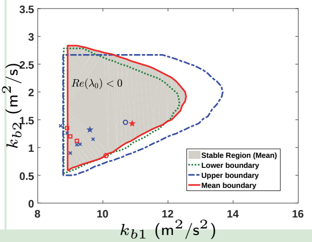
Observations

- Longer steering delay τ_s, more difficult to balance (larger BM_1)
- More severe visual disturbance, more difficult to balance (larger BM_1)

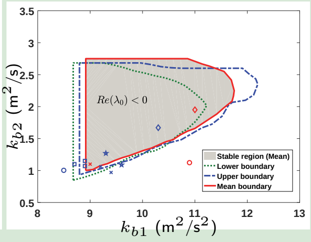
Experimental Results: Stability Regions



(a) $\tau_{s3} = 0.2$ s



(b) $\tau_{s3} = 0.3$ s



(c) $\tau_{s3} = 0.35$ s

Observations

- The shapes of the stability regions changes significantly with delays τ_{s3}
- The changes of stable k_{b1} match the experiments
- Almost all experiments lie in the stable regions
- Longer steering delay results closer to the stability region boundary

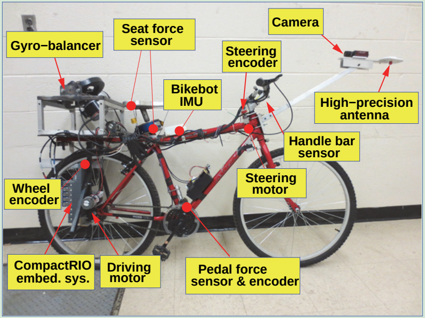
Conclusions and Ongoing Work

Conclusions

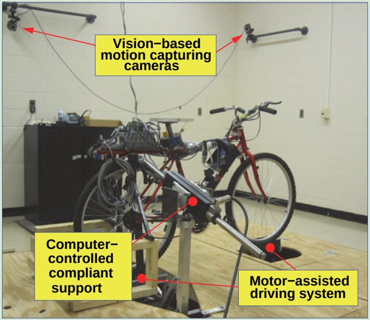
1. Developed a bikebot experimental platform to study the unstable physical human-machine interactions (pHMI)
2. A physical-learning model was presented to capture the physical rider-bikebot interactions in low-dimensional space
3. Analyzed balance skills and defined motor skills metrics in rider-bikebot interactions
4. A neuro-balancing control model was developed and integrated with the rider-bicycle model
5. Stability results were obtained and the sensitivity of the model and control parameters was found to explain the striking clinical observations
6. Extensive experiments were conducted for model estimation and performance demonstration

Ongoing Work

- Conducting experiments to further refine the BEM-based metrics to quantify the balance-tracking motor skills
- Tuning and controlling human sensorimotor skills with the actuated bikebots
- Extensive pre-clinic experiments and testing with human subjects



Outdoor bikebot



Indoor bikebot

Acknowledgments

- Partially supported by the U.S. National Science Foundations under awards CMMI-0954966 and CMMI-1334389
- Rutgers Center of Advanced Infrastructure and Transp. (CAIT)
- National Instruments Inc. for equipment donation and support
- Dr. Guang Yue of Kessler Rehabilitation Institute for collaborative support
- Professor Dezhen Song of Texas A&M University for discussions and suggestions
- Many graduate and undergraduate students in the Rutgers Robotics, Automation, and Mechatronics (RAM) Lab
- Professors Yizhai Zhang of Northwestern Polytechnical University and Tao Liu of Zhejiang University, P. R. China for helpful discussions and suggestions

Thank You!

GPDM for Human Limb Motion

- The first term $P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\beta})$ is factorized as D GP regressions

$$P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\beta}) = \prod_{i=1}^{D} P(\mathrm{y_i}|\boldsymbol{X},\boldsymbol{\beta}_{\mathrm{i}}) = \prod_{\mathrm{i=1}}^{\mathrm{D}} \mathcal{N}(\mathrm{y_i}|\mathbf{0},\boldsymbol{K}_{\mathrm{y_i}} + \sigma_{\mathrm{n_i}}^2 \mathrm{I}),$$

  where $\boldsymbol{K}_{y_i}$ is a Gaussian kernel function, $\boldsymbol{W}_i$ is a diagonal matrix, $\boldsymbol{\beta}_i = \{\boldsymbol{\sigma}_{f_i}, \boldsymbol{W}_i, \boldsymbol{\sigma}_{n_i}\}$ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_i\}^D$.
- The GP model to estimate the second term

$$P(\boldsymbol{X}|\boldsymbol{U},\boldsymbol{\alpha}) = \prod_{i=1}^{d} P(\boldsymbol{x}_i|\boldsymbol{U},\boldsymbol{\alpha}_i) = \prod_{i=1}^{d} \mathcal{N}(\delta\boldsymbol{x}_i|\mathbf{0},\boldsymbol{K}_{x_i} + \sigma_{x_{n_i}}^2 \boldsymbol{I}),$$

  where $\boldsymbol{x}_i = \{\boldsymbol{x}_i(k)\}^N$, $\delta\boldsymbol{x}_i = \{\delta x_i(k)\}^N$, $\boldsymbol{s}(k) = [\boldsymbol{x}^T(k)\ \boldsymbol{u}_h^T(k)]^T$, $\boldsymbol{K}_{x_i}$ is a Gaussian kernel function, $\boldsymbol{\alpha}_i = \{\sigma_{x_{f_i}}, \boldsymbol{W}_{x_i}, \sigma_{x_{n_i}}\}$ and $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_i\}^d$.

GPDM: Mapping from Latent Space to Physical Space

- Given latent state $\boldsymbol{x}(k)$ and observation $\boldsymbol{T}_o = \{\{\boldsymbol{x}(i)\}^N, \{\boldsymbol{y}(i)\}^N\}$, the physical joint angles satisfy

$$P(\boldsymbol{y}(k)|\boldsymbol{x}(k), \boldsymbol{T}_o) = \mathcal{N}(\boldsymbol{y}(k)|GP_\mu(\boldsymbol{x}(k), \boldsymbol{T}_o), GP_\Sigma(\boldsymbol{x}(k), \boldsymbol{T}_o)),$$

  where $GP_\Sigma(\boldsymbol{x}(k), \boldsymbol{T}_o)$ and $GP_\mu(\boldsymbol{x}(k), \boldsymbol{T}_o)$ are GPs with $\boldsymbol{k}_* = \sigma_f^2 e^{-\frac{1}{2}(\boldsymbol{s}(k-1)-\boldsymbol{s}_q)^T W(\boldsymbol{s}^*-\boldsymbol{s}_p)}$.

- The Jacobians of output $\boldsymbol{y}(k)$ and the latent state (needed for EKF design) are respectively as

$$\frac{\partial \boldsymbol{y}(k)}{\partial \boldsymbol{x}(k)} = \frac{\partial GP_\mu(\boldsymbol{x}(k), \boldsymbol{T_o})}{\partial \boldsymbol{k_*}} \frac{\partial \boldsymbol{k_*}}{\partial \boldsymbol{x}(k)},$$

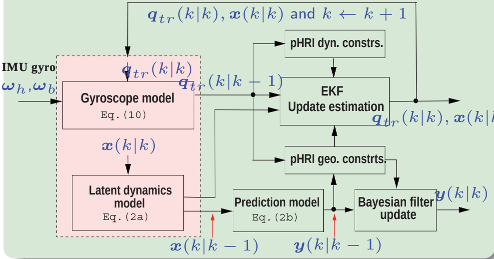
$$\frac{\partial \delta \boldsymbol{x}(k)}{\partial \boldsymbol{x}(k-1)} = \frac{\partial GP_\mu(\boldsymbol{s}(k-1), \boldsymbol{T_p})}{\partial \boldsymbol{k_*}} \frac{\partial \boldsymbol{k_*}}{\partial x(k-1)},$$

  where prediction $\boldsymbol{T}_p = \{\{\boldsymbol{s}(i-1)\}^N, \{\delta \boldsymbol{x}(i)\}^N\}$.

EKF Design for Limb Pose Estimation

- Bayesian filter is used to predict the output $\boldsymbol{y}(k|k)$
- Instead of constant covariances, $\boldsymbol{Q}(k)$ and $\boldsymbol{R}(k)$ are updated by the GP generated by the GPDM (`Lines 3 & 12`)

EKF design flowchart



EKF Implementation

Algorithm 2: EKF implementation

Input: $\mathbf{q}_{rt}(0|0)$, $\mathbf{x}(0|0)$, ω_b and ω_h
Output: Estimates $\mathbf{q}_{rt}(k)$, $\mathbf{x}(k)$, and $\mathbf{y}(k)$

1 Initialize variance matrices $\mathbf{Q}(0)$, $\mathbf{R}(0)$, $\Sigma(0)$;
while $k \leq N$ **do**
2 Update $\mathbf{v}(k|k-1)$ (i.e., $[\mathbf{q}_{rt}^T\ \mathbf{x}^T]^T$) by (9) and (2a);
3 $\mathbf{Q}(k)_{[5:7,5:7]} = \mathrm{GP}_\Sigma(\mathbf{s}_{t-1}, \mathbf{T}_p)$;
4 $\mathbf{F} = [\mathbf{f}_1^T\ f_1\ \mathbf{f}_2^T]^T$; $\mathbf{G}(k) = \frac{\partial \mathbf{F}}{\partial \mathbf{v}}\big|_{\mathbf{v}(k-1)}$ with (5);
5 $\Sigma(k|k-1) = \mathbf{G}(k)\Sigma(k-1)\mathbf{G}^T(k) + \mathbf{Q}(k)$;
6 Update $\hat{\mathbf{z}}(k)$ with $\mathbf{v}(k|k-1)$, $\mathbf{y}(k-1)$ by (10)-(12) ;
7 $\mathbf{M}(k) = \mathrm{GP}_\Sigma(\mathbf{x}(k|k-1), \mathbf{T}_o)$;
8 $\mathbf{H}(k) = \frac{\partial \mathbf{Z}}{\partial \mathbf{v}}\big|_{\mathbf{v}(k|k-1)}$; $\mathbf{N}(k) = \frac{\partial \mathbf{Z}}{\partial \mathbf{y}}\big|_{\mathbf{v}(k|k-1)}$;
9 $\mathbf{K}(k) = \Sigma(k|k-1)\mathbf{H}^T(k)\big[\mathbf{H}(k)\Sigma(k|k-1)\mathbf{H}^T(k) + \mathbf{N}(k)\mathbf{M}(k)\mathbf{N}^T(k) + \mathbf{R}(k)\big]^{-1}$;
10 $\mathbf{v}(k) = \mathbf{v}(k|k-1) + \mathbf{K}(k)(\mathbf{0} - \hat{\mathbf{z}}(k))$;
$\Sigma(k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\Sigma(k|k-1)$;
11 $\mathbf{y}(k)$=`BayesianFilter` $(\mathbf{v}(k), \Sigma(k), \hat{\mathbf{z}}(k))$;
end
function $\mathbf{y}$=`BayesianFilter` $(\mathbf{v}, \Sigma, \hat{\mathbf{z}})$;
12 $\hat{\mathbf{y}} = \mathbf{W}(\mathbf{v})$; $\mathbf{G}_w = \frac{\partial \mathbf{W}}{\partial \mathbf{v}}\big|_{\mathbf{v}}$ with (7); $\mathbf{Q}_y = \mathrm{GP}_\mu(\mathbf{x}, \mathbf{T}_o)$;
13 $\hat{\Sigma}_w = \mathbf{G}_w \Sigma \mathbf{G}_w^T + \mathbf{Q}_y$; $\mathbf{H}_w = \frac{\partial \mathbf{z}}{\partial \mathbf{y}}$;
$\mathbf{K} = \hat{\Sigma}_w \mathbf{H}_w^T (\mathbf{H}_w \hat{\Sigma}_w \mathbf{H}_w^T + \mathbf{R})^{-1}$;
14 $\mathbf{y} = \hat{\mathbf{y}} + \mathbf{K}(\mathbf{0} - \hat{\mathbf{z}})$;

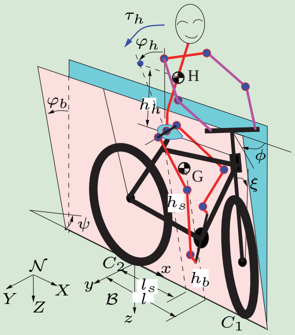
Rider-Bikebot Interaction Dynamic Model

- Steering (i.e., yaw motion)

$$u_\psi := \ddot{\psi} = \frac{v_r \, \mathrm{c}_\xi}{l \, \mathrm{c}_{\varphi_b}} \sec^2 \phi \dot{\phi} + \frac{v_r \, \mathrm{c}_\xi}{l \, \mathrm{c}^2_{\varphi_b}} \mathrm{s}_{\varphi_b} \tan \phi \dot{\varphi}_b.$$

- Upper-body leaning (i.e., gravity/centripetal torques) and torque input $\boldsymbol{\tau} = [0 \ \tau_h]^T$ - Velocity control: rear wheel speed v_r
- Balance coordinates φ_b and φ_h (bikebot and human upper-body rolling angles). Position coordinates X, Y, ψ

Schematic of the rider-bikebot interactions model



- Balancing dynamics model (i.e., internal model)

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{B}\boldsymbol{u},$$

where $\boldsymbol{q} = [\varphi_b \ \varphi_h]^T$, control input $\boldsymbol{u} = [u_r \ u_\psi]^T$, $u_r = \ddot{v}_r$.

Assistive Gyro-Balancer Control Design

- Closed-loop position error dynamics:

$$\boldsymbol{e}_p^{(3)} + b_2\ddot{\boldsymbol{e}}_p + b_1\dot{\boldsymbol{e}}_p + b_0\boldsymbol{e}_p = \boldsymbol{d}_p := \boldsymbol{R}_\psi \begin{bmatrix} 0 \\ u_\psi^{\text{int}} - u_\psi^{\text{ext}} \end{bmatrix} = \boldsymbol{R}_\psi \begin{bmatrix} 0 \\ d_0 \end{bmatrix}.$$

$\boldsymbol{e}_p$ is driven by the difference $d_0 = u_\psi^{\text{int}} - u_\psi^{\text{ext}}$!

- The use of assistive gyro-balancer is to reduce $\|\boldsymbol{d}_p\|$. The new steering control $\bar{u}_\psi^{\text{int}}$ and the gyro-balancer control u_w

$$J_t v_\psi^{\text{int}} = f(\varphi)w + g_\psi(\varphi)u_\psi^{\text{int}} = f(\varphi) + g_\psi(\varphi)\bar{u}_\psi^{\text{int}} + g_w u_w.$$

where $|d_0| - |\bar{d}_0| > 0$ and $\bar{d}_0 = \bar{u}_\psi^{\text{int}} - u_\psi^{\text{ext}}$.

- Considering the physical limits, the gyro-balancer-assisted controller $\bar{\mathcal{C}} : (u_r^{\text{ext}}, \bar{u}_\psi^{\text{int}}, u_w)$ with

$$\begin{aligned} u_w &= \operatorname{sign}(u_w) \min\left(\left|g_w^{-1} g_\psi(|d_0| - b_\psi)\right|, \left|f_{wc}^{\pm}(\varphi_w)\right|\right), \\ \bar{u}_\psi^{\text{int}} &= g_\psi^{-1}(\varphi_b)\left[J_t v_\psi^{\text{int}} - f(\varphi_b) - g_w u_w\right]. \end{aligned}$$

Assistive Gyro-Balancer Control Design

- Closed-loop position error dynamics:

$$\boldsymbol{e}_p^{(3)} + b_2\ddot{\boldsymbol{e}}_p + b_1\dot{\boldsymbol{e}}_p + b_0\boldsymbol{e}_p = \boldsymbol{d}_p := \boldsymbol{R}_\psi \begin{bmatrix} 0 \\ u_\psi^{\text{int}} - u_\psi^{\text{ext}} \end{bmatrix} = \boldsymbol{R}_\psi \begin{bmatrix} 0 \\ d_0 \end{bmatrix}.$$

$\boldsymbol{e}_p$ is driven by the difference $d_0 = u_\psi^{\text{int}} - u_\psi^{\text{ext}}$!

- The use of assistive gyro-balancer is to reduce $\|\boldsymbol{d}_p\|$. The new steering control $\bar{u}_\psi^{\text{int}}$ and the gyro-balancer control u_w

$$J_t v_\psi^{\text{int}} = f(\varphi)w + g_\psi(\varphi)u_\psi^{\text{int}} = f(\varphi) + g_\psi(\varphi)\bar{u}_\psi^{\text{int}} + g_w u_w.$$

Stability and Convergence Results

- Under $\mathcal{C}$ and $\overline{\mathcal{C}}$, the position tracking errors $\boldsymbol{e}_p(t)$ and $\bar{\boldsymbol{e}}_p(t)$ exponentially converge to regions near the origin.
- $\|\boldsymbol{e}_p(t)\| \leq e_p^b(t)$ and $\|\bar{\boldsymbol{e}}_p(t)\| \leq \bar{e}_p^b(t)$, then $\bar{e}_p^b(t) \leq e_p^b(t)$, for $\forall t \geq t_a$

Stability Analysis for the EIC Design

Closed loop errors dynamics:

$$\boldsymbol{e}_p^{(3)} + b_2\ddot{\boldsymbol{e}}_p + b_1\dot{\boldsymbol{e}}_p + b_0\boldsymbol{e}_p = \boldsymbol{R}_\psi \begin{bmatrix} p_p' \\ 0 \end{bmatrix} =: \boldsymbol{p}_p, \tag{2}$$

$$\ddot{\boldsymbol{e}}_q + a_1\dot{\boldsymbol{e}}_q + a_2\boldsymbol{e}_q = \boldsymbol{p}_q. \tag{3}$$

Driving disturbances:

$$\begin{aligned} p_p' = p_p(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{q}_e) &= \left(\boldsymbol{B}_\tau^{-1}(\boldsymbol{q}_e)\left[\boldsymbol{C}_q(\boldsymbol{q}_e) - \boldsymbol{G}(\boldsymbol{q}_e)\right]\right)_1 \\ &- \left(\boldsymbol{B}_\tau^{-1}(\boldsymbol{q})\left[\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + C_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q})\right]\right)_1, \end{aligned}$$

$$\boldsymbol{p}_q = \ddot{\boldsymbol{q}}_e - \bar{L}^2_{\boldsymbol{N}_{\text{ext}}}\boldsymbol{q}_e + a_1(\dot{\boldsymbol{q}}_e - \bar{L}_{\boldsymbol{N}_{\text{ext}}}\boldsymbol{q}_e). \tag{4}$$

with the boundaries

$$\begin{aligned} \|\boldsymbol{p}_p\|_2 &\leq c_1 + c_2\|\boldsymbol{e}_1\|_2 + c_3\|\boldsymbol{e}_2\|_2, \\ \|\boldsymbol{p}_q\|_2 &\leq c_4 + c_5\|\boldsymbol{e}_1\|_2 + c_6\|\boldsymbol{e}_2\|_2. \end{aligned}$$

Stability analysis for EIC design (cont'd)

For the feedback linearized dynamics matrices

$$\boldsymbol{A}_p = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -b_0 & -b_1 & -b_2 \end{bmatrix}, \ \boldsymbol{A}_q = \begin{bmatrix} 0 & 1 \\ -a_1 & -a_0 \end{bmatrix}. \tag{5}$$

and for given positive definite symmetric matrices $\boldsymbol{Q}_p$ and $\boldsymbol{Q}_q$, we can find out $\boldsymbol{M}_p \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{M}_q \in \mathbb{R}^{2\times 2}$ such that

$$\boldsymbol{M}_p\boldsymbol{A}_p + \boldsymbol{A}_p^T\boldsymbol{M}_p = -\boldsymbol{Q}_p, \ \boldsymbol{M}_q\boldsymbol{A}_q + \boldsymbol{A}_q^T\boldsymbol{M}_q = -\boldsymbol{Q}_q.$$

We consider the Lyapunov function candidate

$$V = \boldsymbol{e}_1^T\boldsymbol{M}_1\boldsymbol{e}_1 + \boldsymbol{e}_2^T\boldsymbol{M}_2\boldsymbol{e}_2, \tag{6}$$

with $\boldsymbol{M}_1 = diag(\boldsymbol{M}_p, \boldsymbol{M}_p)$ and $\boldsymbol{M}_2 = diag(\boldsymbol{M}_q, \boldsymbol{M}_q)$.

Stability analysis for EIC design (cont'd)

For any given positive $d_1, d_2 > 0$, we have

$$\dot{V} \leq -\left(\eta_1 - \frac{\alpha_p^2 c_1^2}{d_1}\right)\|\boldsymbol{e}_1\|_2^2 - \left(\eta_2 - \frac{\alpha_q^2 c_4^2}{d_2}\right)\|\boldsymbol{e}_2\|_2^2 + d_1 + d_1,$$

with $\eta_1 = \beta_p - 2\alpha_p c_2 - \alpha_p c_3 - \alpha_q c_5$ and
$\eta_2 = \beta_q - 2\alpha_q c_6 - \alpha_p c_3 - \alpha_q c_5$.
Therefore, if existing $\eta_1 > \alpha_p^2 c_1^2/d_1$, $\eta_2 > \alpha_q^2 c_4^2/d_2$, as $t \to \infty$,
$V = \boldsymbol{e}_1^T(t)\boldsymbol{M}_1\boldsymbol{e}_1(t) + \boldsymbol{e}_2^T(t)\boldsymbol{M}_2\boldsymbol{e}_2(t) \leq b$,with

$$b := \arg_{k>0}\sup\left\{k = V(\boldsymbol{e}_1, \boldsymbol{e}_2), (\boldsymbol{e}_1, \boldsymbol{e}_2) \in \Omega^*\right\}.$$

and the bounded closed set

$$\Omega^* = \left\{(\boldsymbol{e}_1, \boldsymbol{e}_2) : \left(\eta_1 - \alpha_p^2 c_1^2/d_1\right)\|\boldsymbol{e}_1\|_2^2 + \left(\eta_2 - \alpha_q^2 c_4^2/d_2\right)\|\boldsymbol{e}_2\|_2^2 = d_1 + d_2\right\}$$