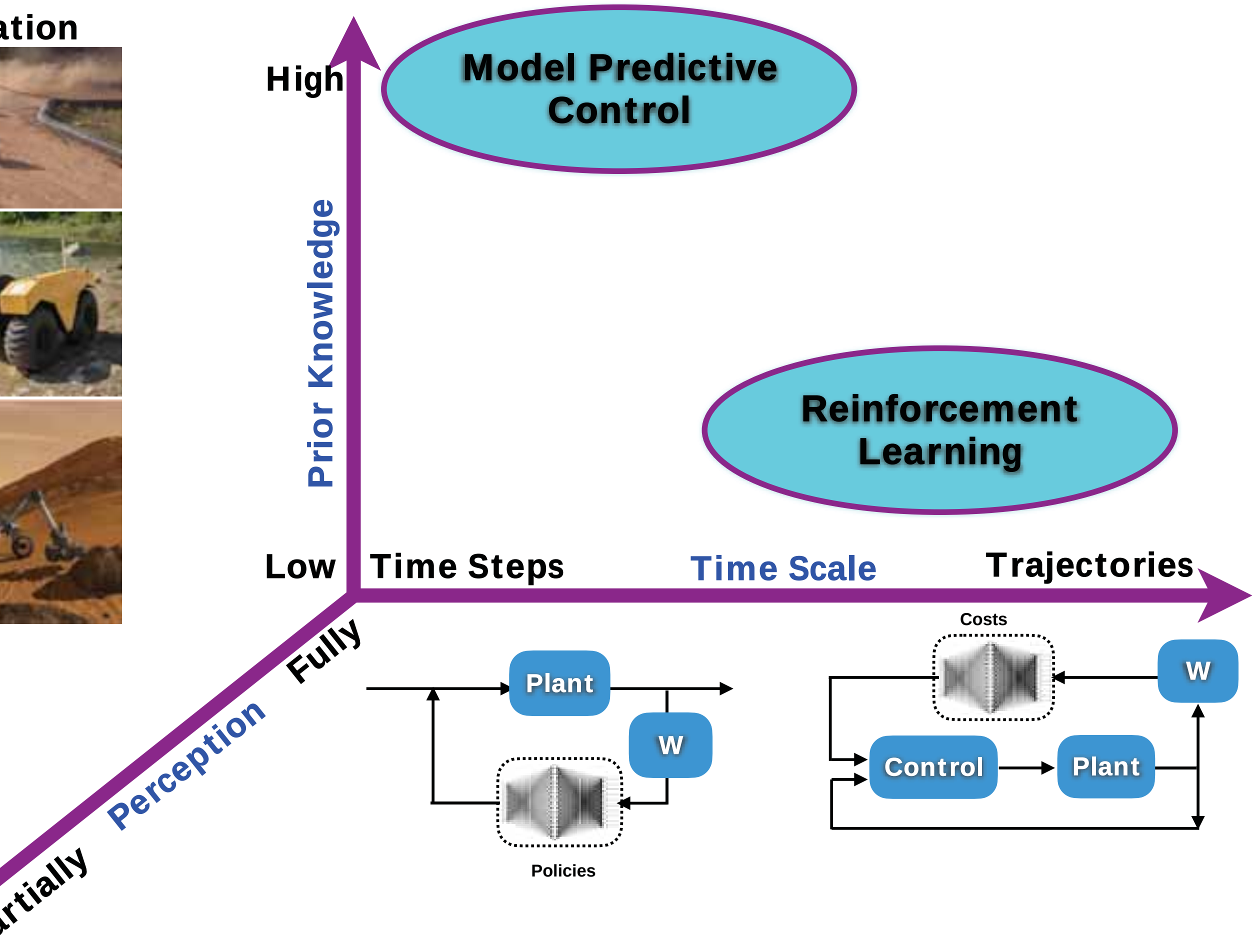
Decision Making Architectures for Safe Planning and Control of Agile Autonomous Vehicles

Evangelos A. Theodorou
Autonomous Control and Decision Systems Lab
Georgia Institute of Technology

Perceptual Decision Making

Terrestrial Navigation



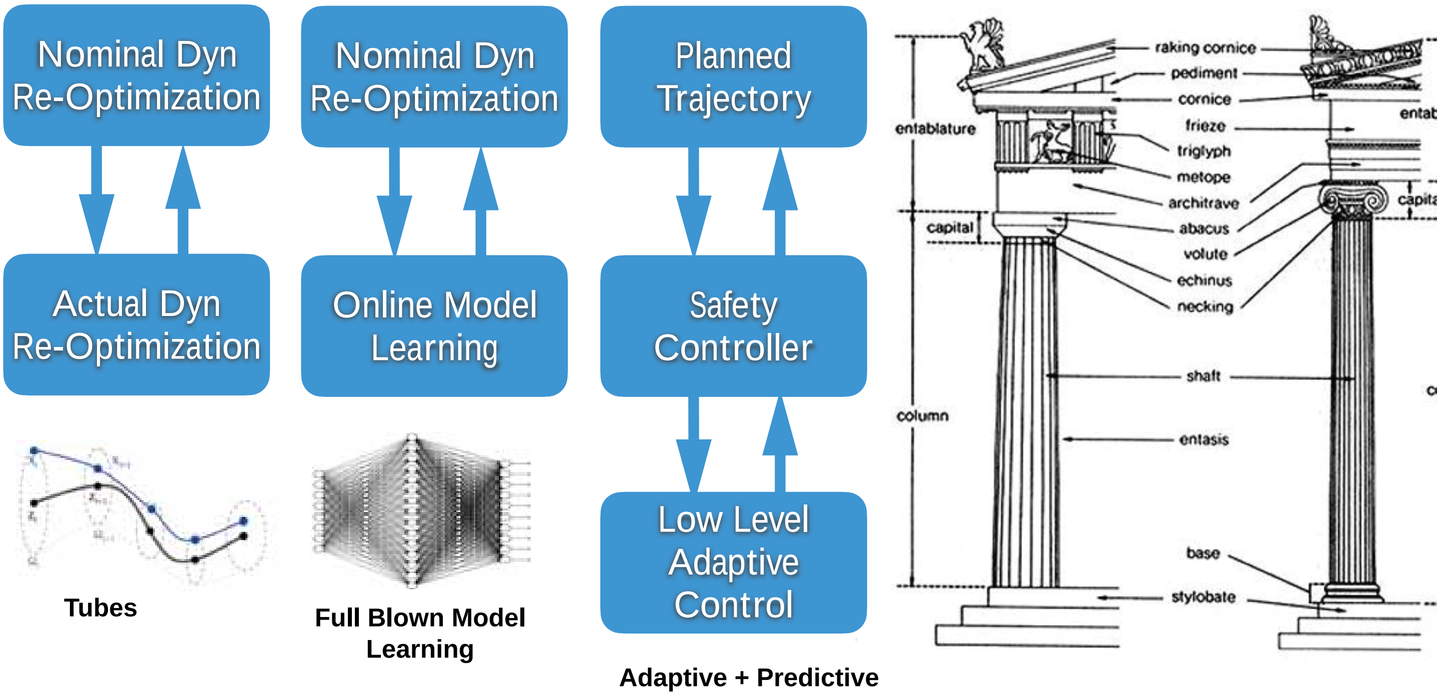
Outline

- ✦ Intro & Motivation
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusions and Future

What happens when uncertainty is not considered?

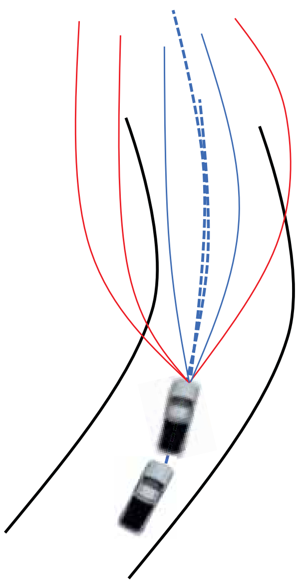
Information Processing Architectures

- ✦ **How would you architect your stack?**
- ✦ **Where should learning be incorporated?**
- ✦ **What notions of robustness we have?**



Model Predictive Path Integral (MPPI) Control

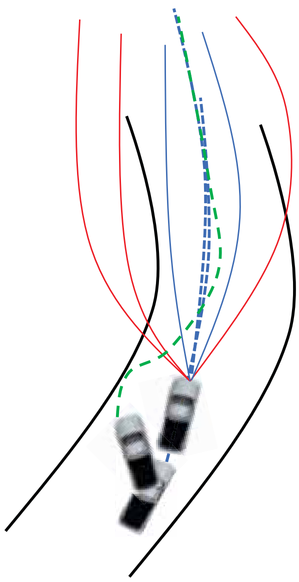
MPPI



(-) Importance Sampler may get stuck to a local minima.

(-) Robustness issues when Large disturbances.

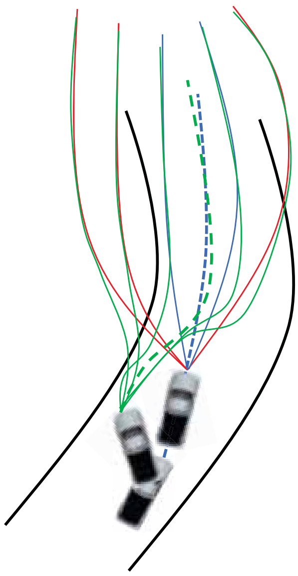
Tube-MPPI



(-) Nominal State is chosen independent of Actual State.

(-) Importance Sampling is unaware of the underlying ancillary controller.

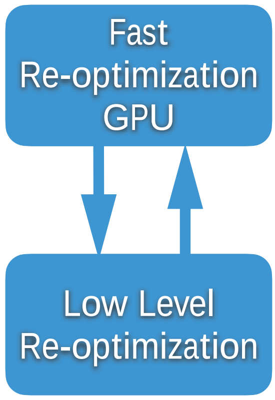
Robust MPPI



(+) Augmented Importance Sampling.

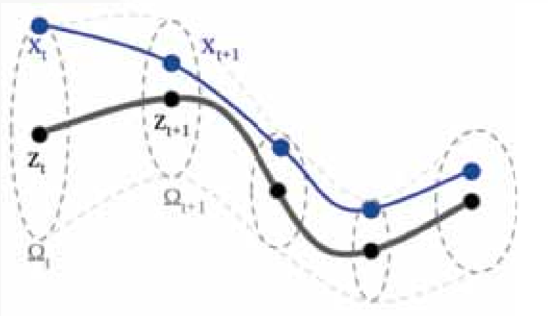
(+) Nice Trade-off between agility and robustness.

Robust MPPI



- ✦ Fast re-planning on GPU on nominal dynamics/Fast Tracking on a CPU
- ✦ Free Energy Diff < Levels Constraint Satisfaction + Tracking/Uncertainty + Sampling Error

Learning Deep Tubes for Robust MPC



$$z_{t+1} = f_z(z_t, v_t)$$
$$\omega_{t+1} = f_\omega(\omega_t, z_t, v_t, t)$$
$$P(d(x_t, z_t) \leq \omega_t) \geq \alpha, \quad \forall t \in \mathbb{N}$$

$$\pi(x,z) : \mathbb{X} \times \mathbb{Z} \to \mathbb{U}$$
$$d(x,z) = |P_{\mathbb{Z}}(x) - z| \in \mathbb{R}^{n_z}$$

$$\Omega_\omega(z) := \{x \in \mathbb{X} : d(x,z) \leq \omega\}$$

$$\Omega_\omega(z) := \{x \in \mathbb{X} : \|P_{\mathbb{Z}}(x) - z\|_\omega \leq 1\}$$

$$\begin{aligned}
&\min_{v_{\cdot|t} \in \mathbb{V}} J_T(v_{\cdot|t}, z_{\cdot|t}, \omega_{\cdot|t}) \\
&s.t. \forall k = 0, \cdots, T: \\
&\quad z_{k+1|t} = f_z(z_{k|t}, v_{k|t}) \\
&\quad \omega_{k+1|t} = f_\omega^\theta(\omega_t, z_t, v_t, t) \\
&\quad \omega_{0|t} = d(x_t, z_{0|t}) \\
&\quad z_{T|t} = f_z(z_{T|t}, v_{T|t}) \\
&\quad \omega_{T|t} \geq f_\omega^\theta(\omega_{T|t}, z_{T|t}, v_{T|t}, T) \\
&\Omega_{\omega_{k|t}}(z_{k|t}) \subseteq \mathcal{C}
\end{aligned}$$

Theorem III.1. *Suppose that the MPC problem (13) is feasible at $t = 0$. Then the problem is feasible for all $t > 0 \in \mathbb{N}$ and at each timestep the constraints are satisfied with probability α.*

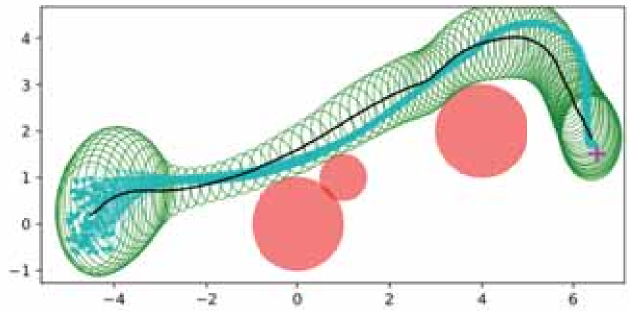
Learning Deep Tubes for Robust MPC

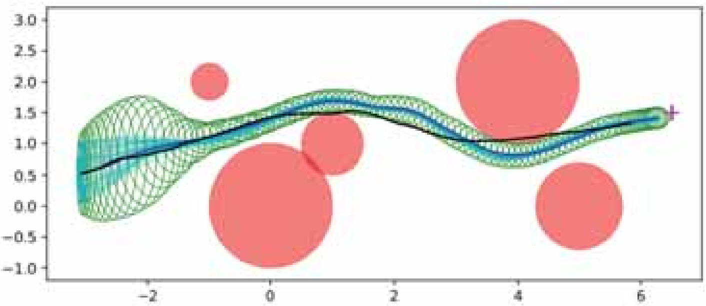
$$\dot{p}_x = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -k_f \end{bmatrix} p_x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} w$$

$$\dot{z}_x = \begin{bmatrix} 0 & 1 \\ 0 & -k_f^z \end{bmatrix} z_x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_x \quad w \sim \mathcal{N}(0, \epsilon I_{2\times 2})$$

$$e_x = k_e(z_x - p_x)$$

$$\pi_x(p_x, z_x) = k_p(e_x - \dot{p}_x + \dot{z}_x) + k_d(-\ddot{p}_x)$$

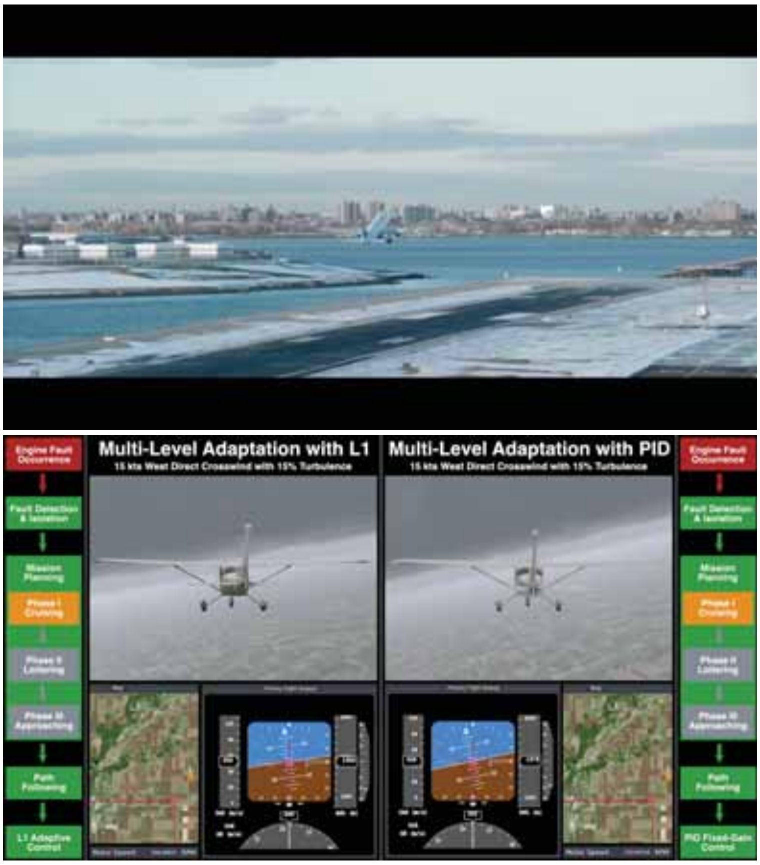


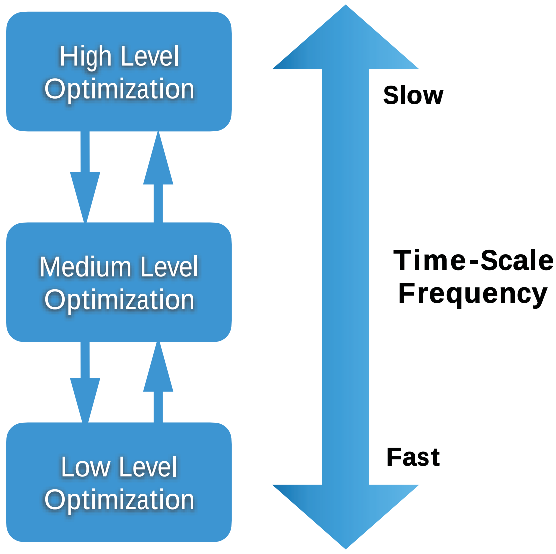


Georgia Tech

Sully Miracle on the Hudson

- Airbus 320 lost both engines shortly after takeoff due to bird strike.

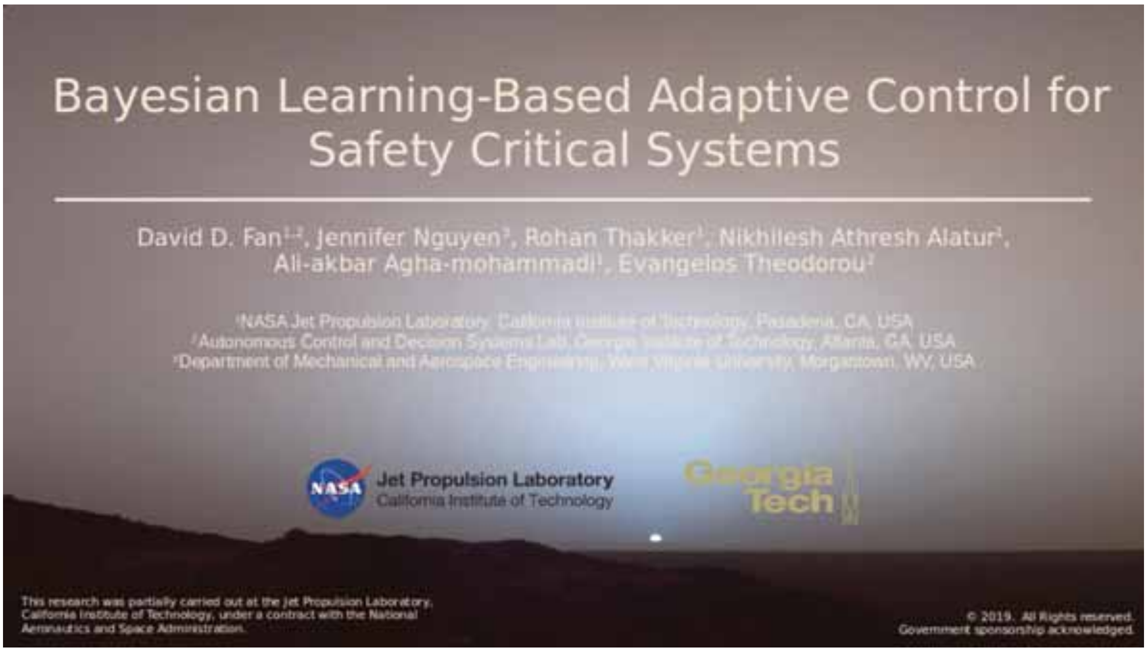


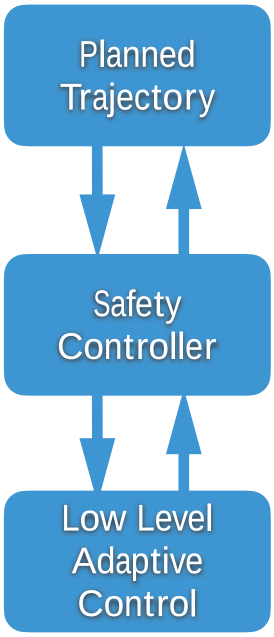


Courtesy: NASA Langley Aerodrome

NSF NASA

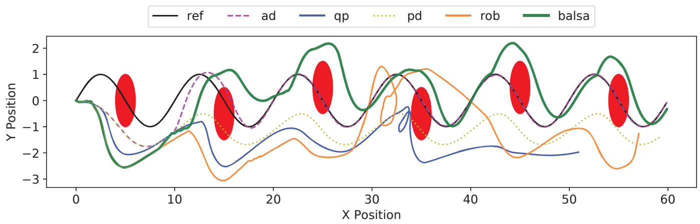
Adaptation and Online Learning

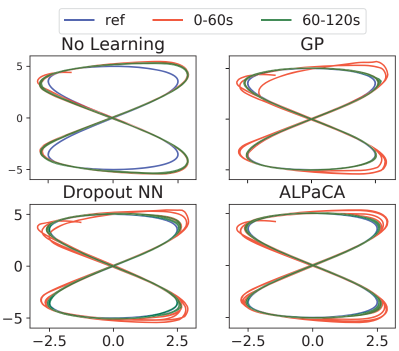




- ✦ Stochastic Control Barrier Functions
- ✦ Stochastic Control Lyapunov Functions
- ✦ Bayesian Neural Networks

How do we bring adaptation?



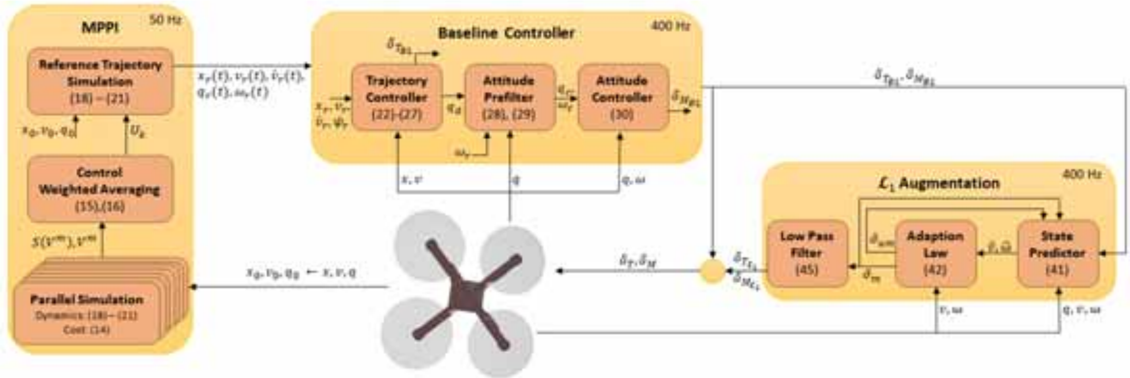


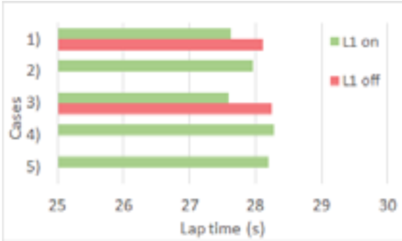
Algorithm 1: BAyesian Learning-based Safety and Adaptation (BALSA)

Require: Prior model $\hat{f}(x)$, known $g(x)$, reference trajectory x_{rm}, choice of modeling algorithm $\bar{\Delta}_i(x) \sim \mathcal{N}(m_i(x), \sigma_i(x))$, dt, A, $Hu \leq b$.
Initialize: $i=0$, Dataset $\mathcal{D}_0 = \emptyset$, $t=0$, solve P
while *true* **do**
  Obtain $\mu_{rm} = \dot{x}_{2rm}(t)$ and compute μ_{pd}
  Compute model error and uncertainty $\mu_{ad} = m_i(x(t))$, and $\sigma_i(x(t))$
  $\mu_{qp} \leftarrow$ Solve QP (17)
  Set $u(t) = g(x)^{-1}(\mu_{rm} + \mu_{pd} + \mu_{qp} - \mu_{ad} - \hat{f}(x))$
  Apply control $u(t)$ to system.
  Step forward in time $t \leftarrow t + dt$.
  Append new data point to database:
  $\bar{X}_t = [x(t)]$, $\bar{Y}_t = (x_2(t+dt) - x_2(t))/dt - (\hat{f}(x(t)) + g(x(t)u(t))$.
  $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{\bar{X}_t, \bar{Y}_t\}$
  if *updateModel* **then**
    Update model $\bar{\Delta}_i(x, \mu)$ with database $\mathcal{D}_i$
    $\mathcal{D}_{i+1} \leftarrow \mathcal{D}_i$, $i \leftarrow i+1$

- ✦ Stochastic Control Barrier Functions
- ✦ Stochastic Control Lyapunov Functions
- ✦ Bayesian Neural Networks

Adaptation and Online Learning





1) known dynamics model (since drag is not modeled in the nominal dynamics, some drag compensation is expected with $\mathcal{L}_1$ augmentation);
2) mass increase by 50%;
3) moment of inertia increase by 100% in all axes;
4) constant nose-up pitching moment disturbance of 0.1 Nm (equivalent to center of gravity offset);
5) reduction in motor thrust control power by 40% (reduction in both $\bar{T}_{\delta_T}$ and $\bar{\mathbf{M}}_{\delta_M}$).

Case	$\mathcal{L}_1$ off	$\mathcal{L}_1$ on
1)	✓	✓
2)	✗	✓
3)	✓	✓
4)	✗	✓
5)	✗	✓

Adaptation and Online Learning

1) known dynamics model (since drag is not modeled in the nominal dynamics, some drag compensation is expected with $\mathcal{L}_1$ augmentation);
2) mass increase by 50%;
3) moment of inertia increase by 100% in all axes;
4) constant nose-up pitching moment disturbance of 0.1 Nm (equivalent to center of gravity offset);
5) reduction in motor thrust control power by 40% (reduction in both $\bar{T}_{\delta_T}$ and $\bar{\mathbf{M}}_{\delta_M}$).

Case	$\mathcal{L}_1$ off	$\mathcal{L}_1$ on
1)	✓	✓
2)	✗	✓
3)	✓	✓
4)	✗	✓
5)	✗	✓

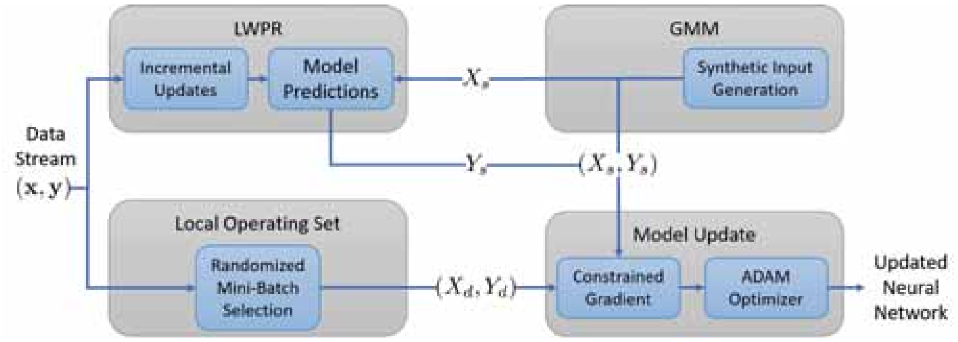
Adaptation and Online Learning

System ID Distribution: $\mathbf{x} \sim \mathcal{P}_{ID}(\mathcal{X})$ **Local Distribution:** $\mathbf{x} \sim \mathcal{P}_{L}(\mathcal{X})$

Update Scheme: $\theta_{i+1} = \theta_i - \gamma G_L(\theta_i) \rightarrow \theta_{i+1} = \theta_i - \gamma(G_L(\theta_i) + G_{ID}(\theta_i))$

Proposed Scheme:
$$\theta_{i+1} = \theta_i - \gamma\left(\alpha G_L(\theta_i) + G_{ID}(\theta_i)\right)$$
$$\alpha = \max_{a \in [0,1]} \quad s.t \quad \langle a G_L(\theta_i) + G_{ID}(\theta_i), G_{ID}(\theta_i) \rangle \geq 0$$

LWPR:
$$\mathbf{y} = \sum_{i=1}^{L} w_i \cdot f_i(\mathbf{x} - \mathbf{c}_i), \quad w_i = \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_I)^{\mathrm{T}} D_i (\mathbf{x} - \mathbf{c}_i)\right)}{\sum_{j=1}^{L} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_j)^{\mathrm{T}} D_j (\mathbf{x} - \mathbf{c}_j)\right)}$$



G. Williams et all, arXiv:1905.05162, Submitted

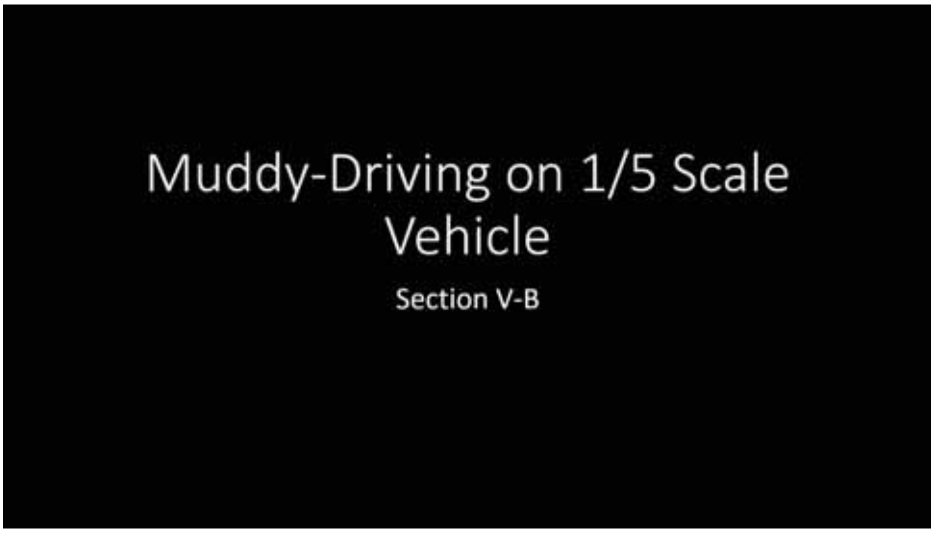
Adaptive Model Predictive Control

Computation →

	Size	FLOPs/Prediction
LWPR	5,645 (Receptive Fields)	$> 141,125$
Neural Network	1,412 (Weights and Biases)	$2,688$

Performance →

	Base	SGD	LW-PR^2	LWPR
Roll Rate	0.01	0.01	0.01	0.01
Long. Acc.	2.73	2.28	2.30	**2.06**
Lat. Acc.	1.71	1.29	**1.24**	1.28
Head. Acc.	8.28	**4.48**	4.87	4.54
Total MSE	3.18	2.10	2.11	**1.97**
Active MSE	N/A	N/A	2.54	N/A



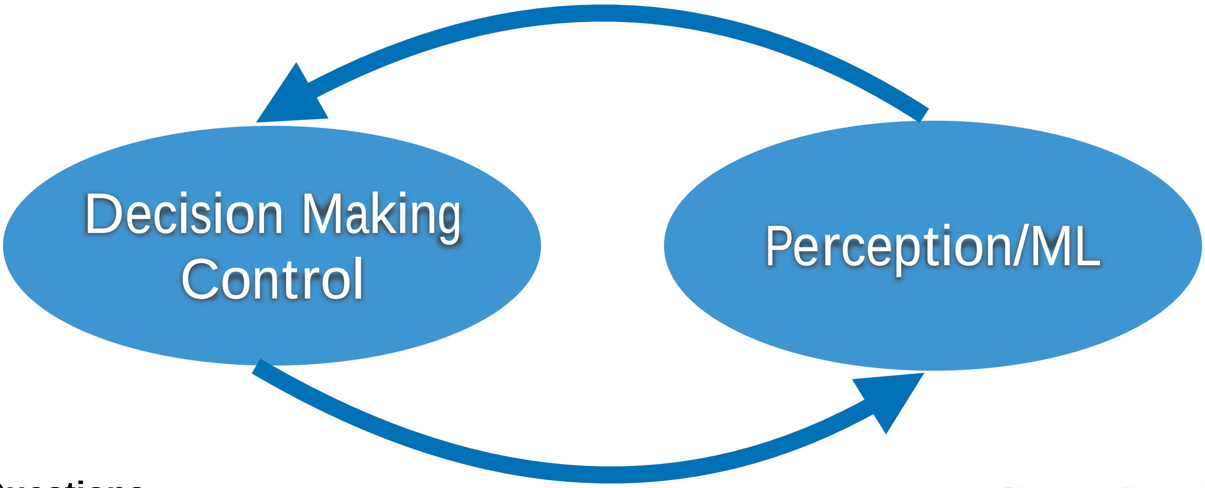
Outline

- ✦ Motivation & Intro
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusiona and Future

Outline

- ✦ Motivation & Intro
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusion and Future

Information Processing Architecture (IPA) for Perceptual control



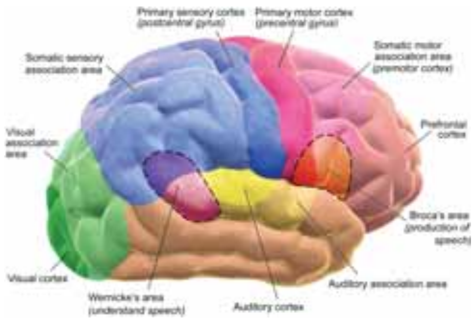
Questions:

What is the optimal **IPA** for perceptual control?

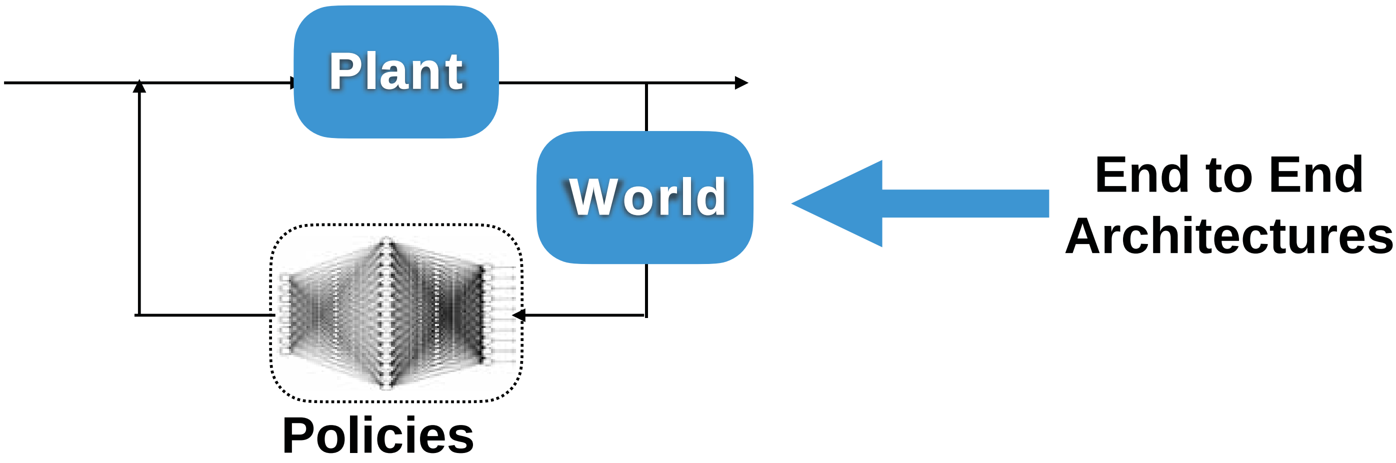
Is the design of **IPA** imposed by the nature of the data?

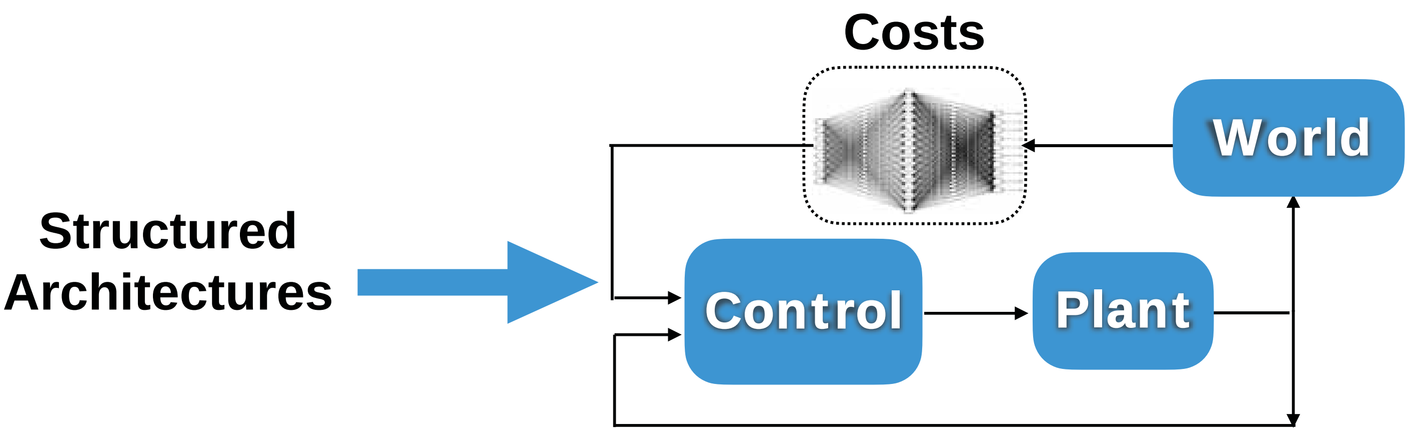
Do we have any priors for designing **IPAs**?

How important is the structure of **IPAs** for **safety** in AI?



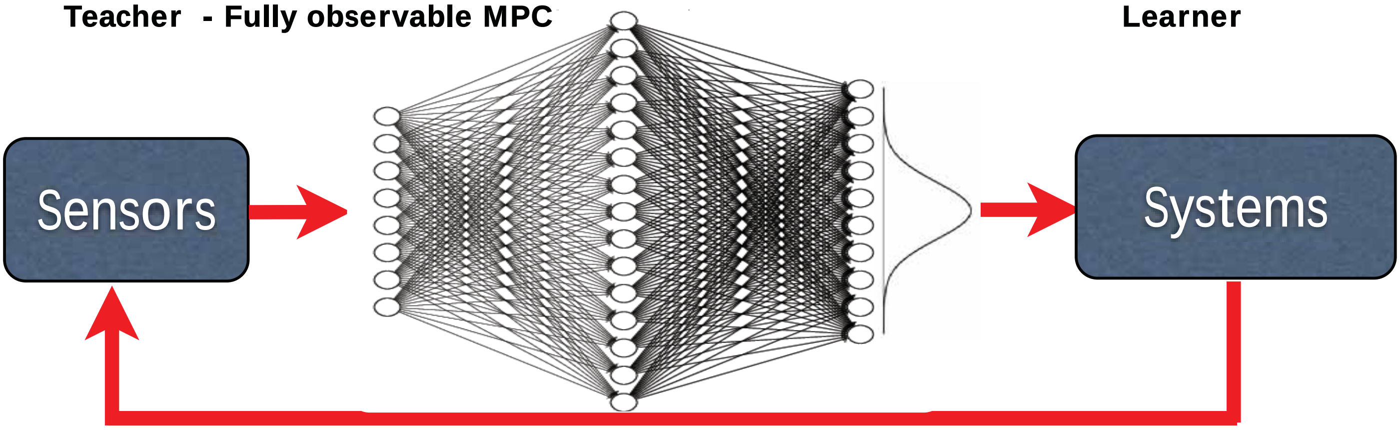
Information Processing Architectures : IPA





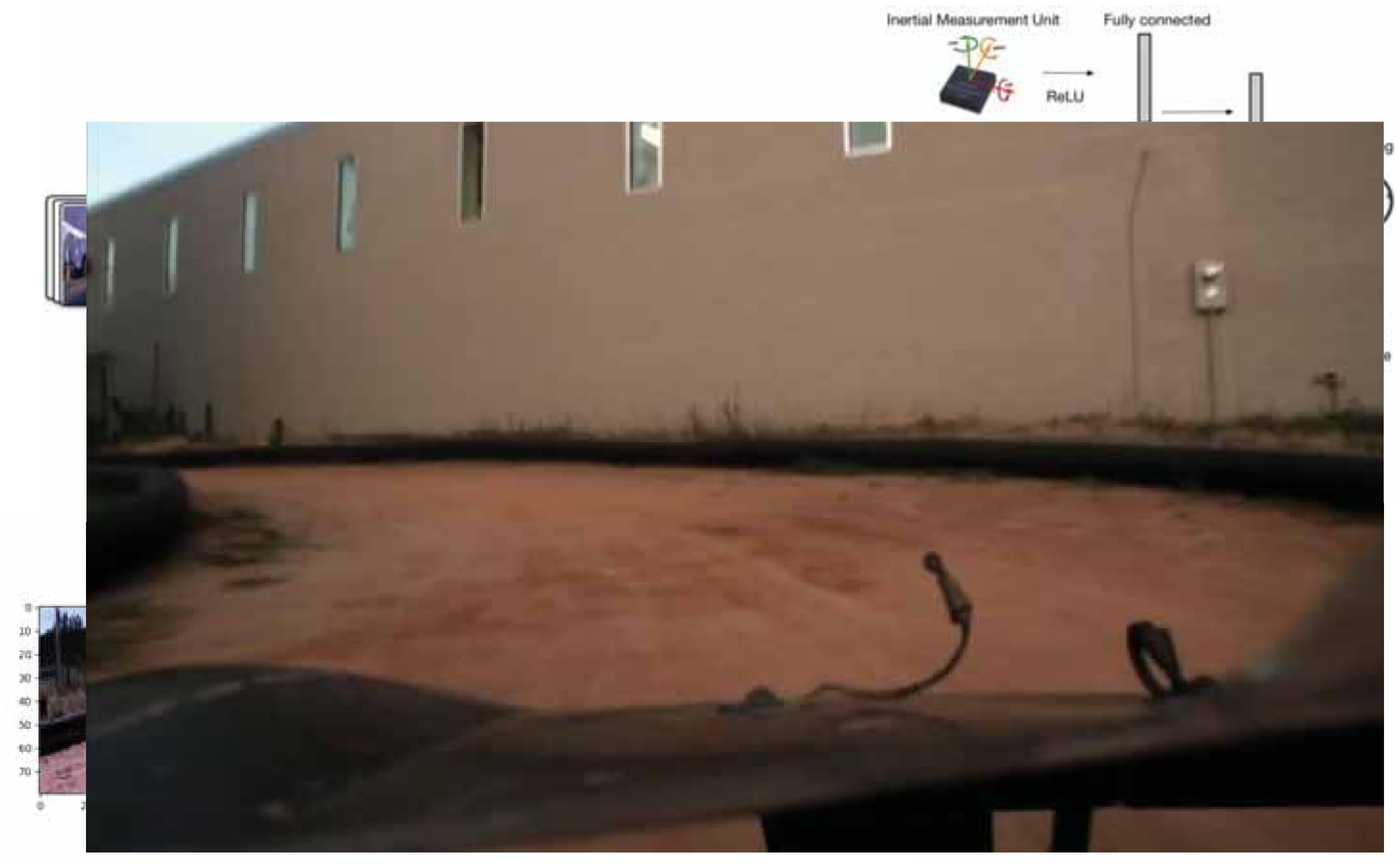
IPA-I

Teacher - Fully observable MPC **Learner**



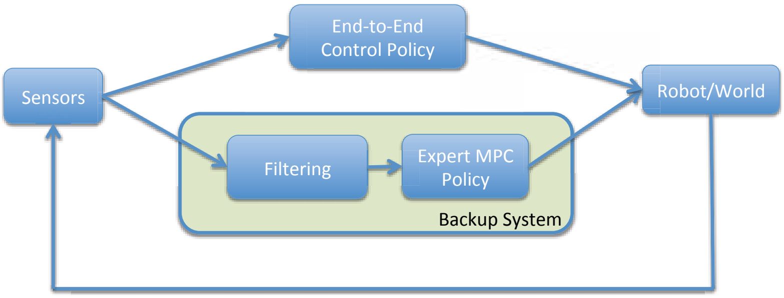
Y. Pan et all RSS 2018.

IPA-I



Y. Pan et all RSS 2018.

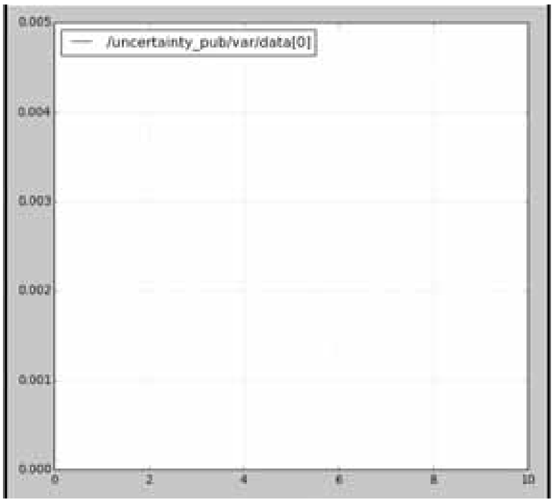
IPA-I & Uncertainty Quantification

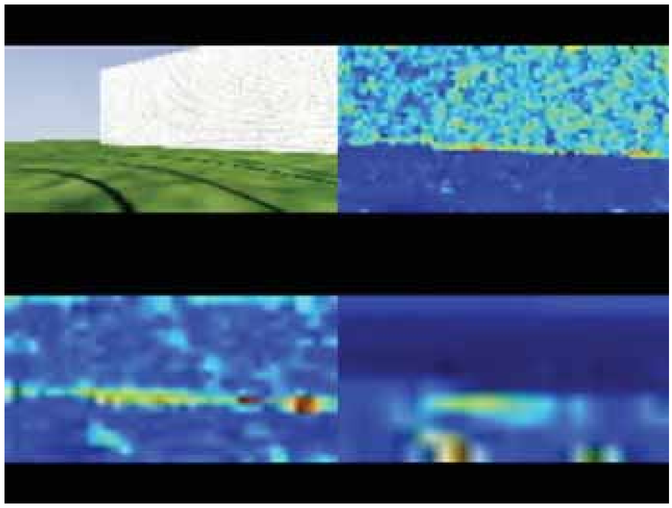


Types of Uncertainty in ML Models

Aleatoric - Incomplete data

Epistemic- Incomplete knowledge of the environment.



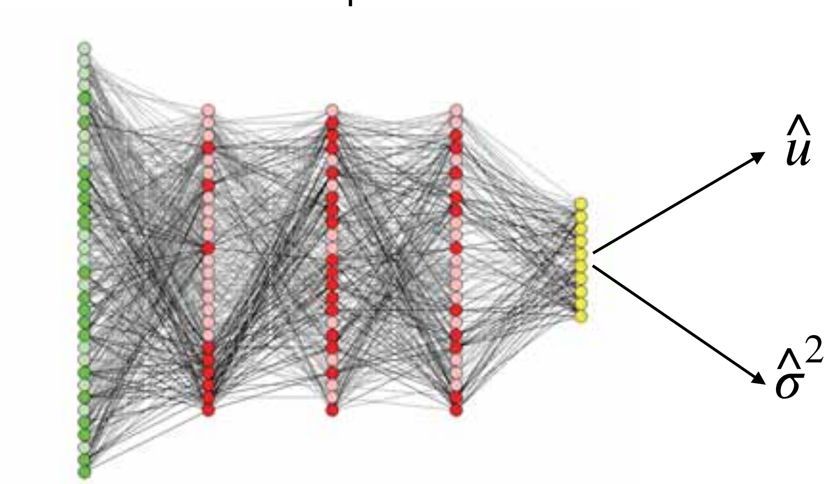


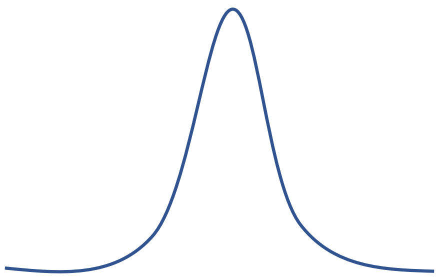
K. Lee et all ICRA 2019.

IPA-I & Uncertainty Quantification

At Training Time Minimize the Loss: $\mathcal{L}(\pi) = \frac{1}{2\hat{\sigma}^2}||u^* - \hat{u}||^2 + \frac{1}{2}\log(\hat{\sigma}^2)$

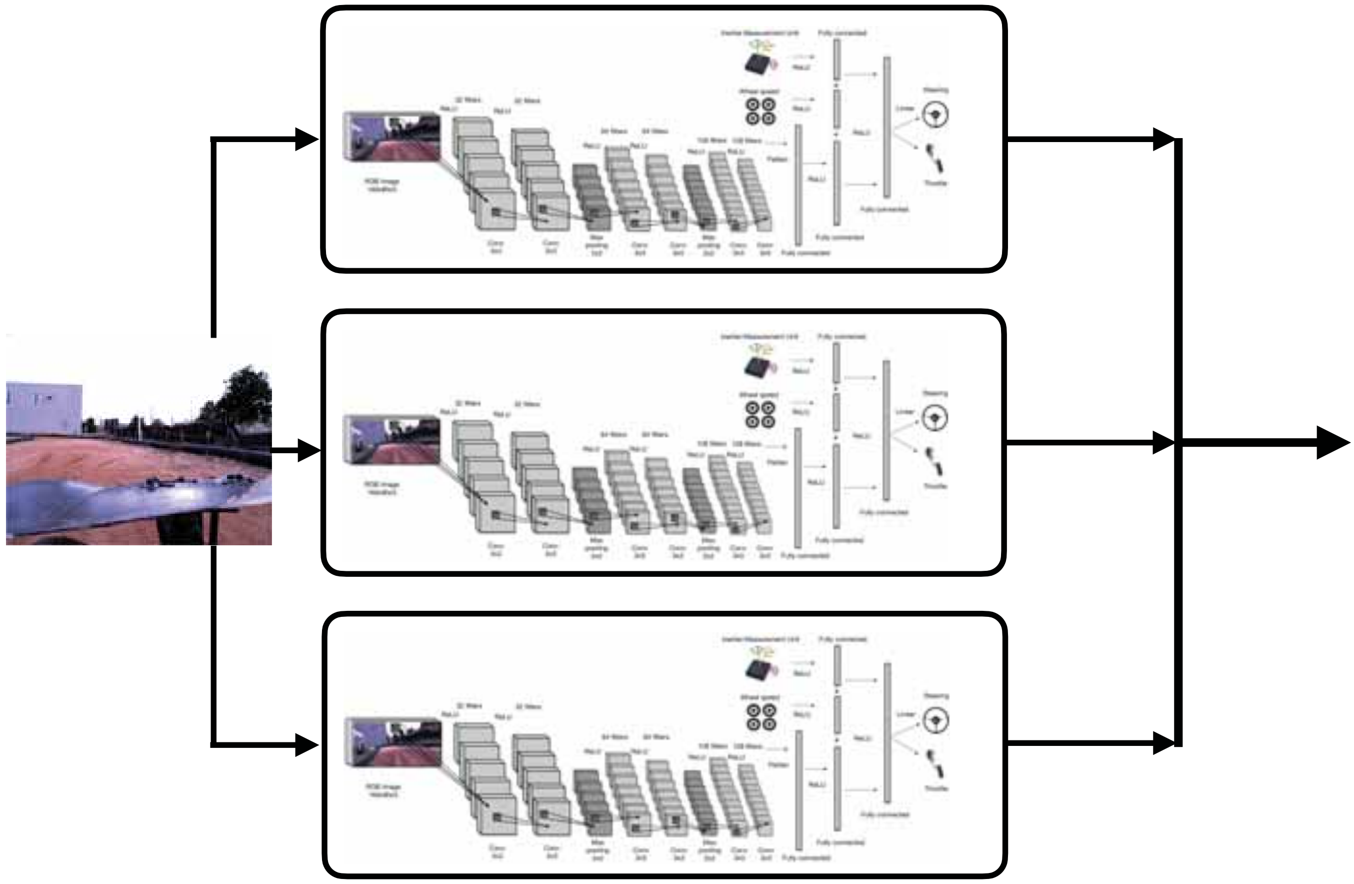
At Test Time Sample the structure of the Network:



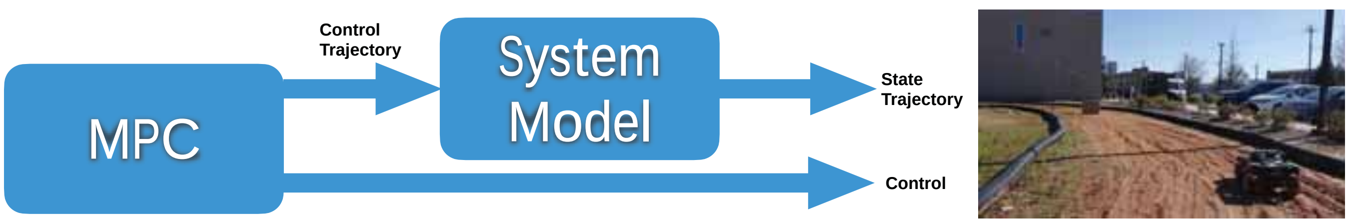


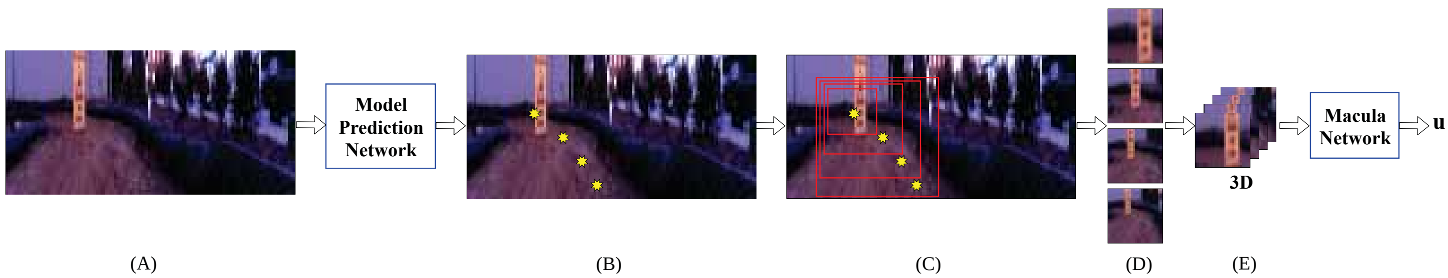
Total Uncertainty: $\sigma_i^2 = Var(u_i) \approx \underbrace{\frac{1}{K}\sum_{k=1}^{K}\hat{u}_{i_k}^2 - \left(\frac{1}{K}\sum_{k=1}^{K}\hat{u}_{i_k}\right)^2}_{epistemic} + \underbrace{\frac{1}{K}\sum_{k=1}^{K}\hat{\sigma}_{i_k}^2}_{aleatoric}$

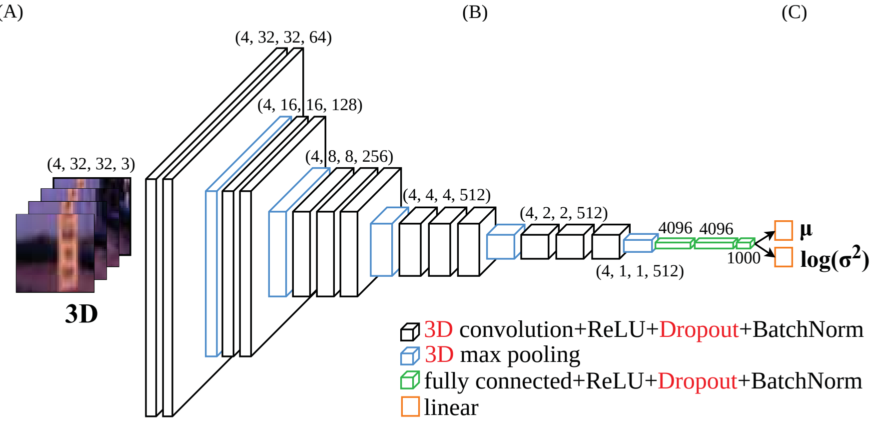
Uncertainty Quantification & Redundancy



IPA-II: The Macula-Net

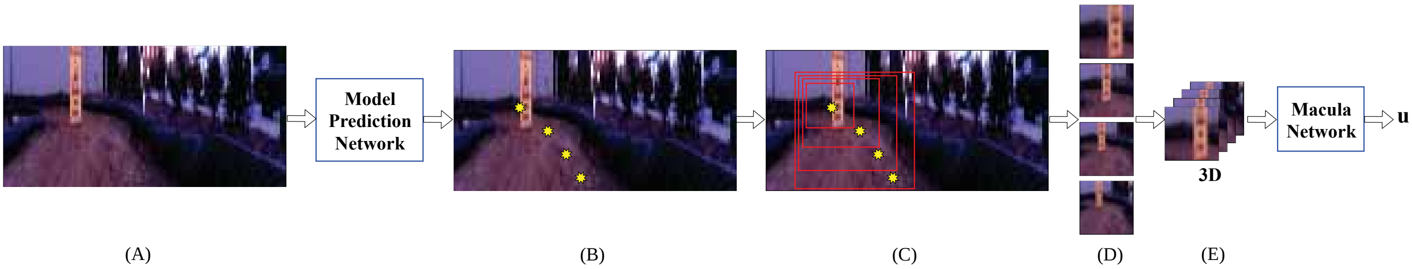






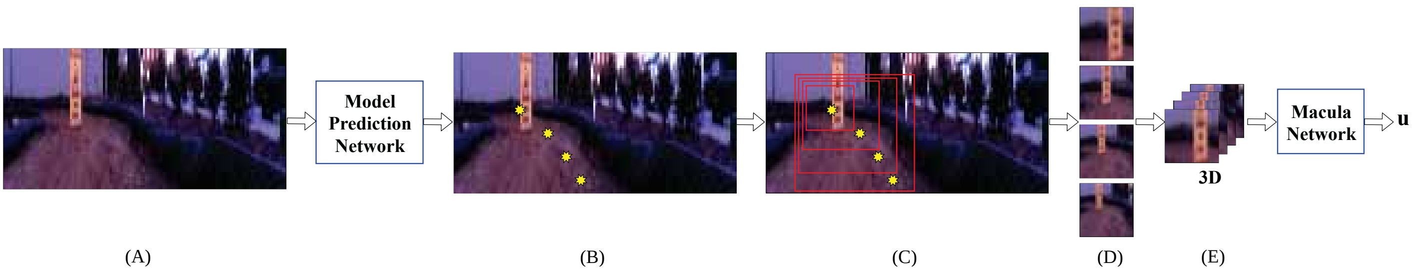
K. Lee et all, arXiv:1904.11898, Submitted

IPA-II: The Macula-Net

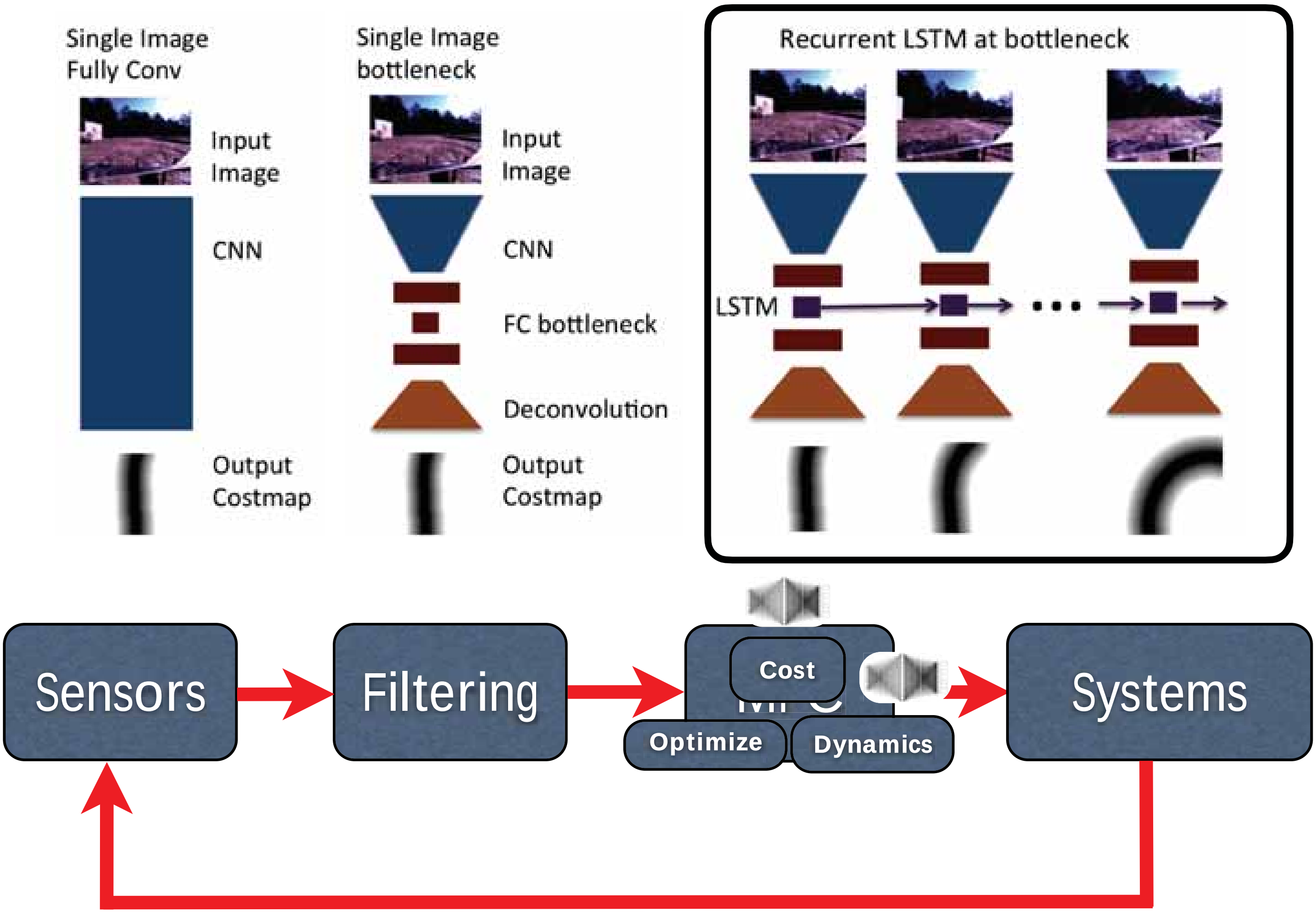


New Objects	DropoutVGG [9]	PAPC [Ours]
	Min: 0.37 m Avg: 0.39 m Max: 0.42 m	Min: **4.28** m Avg: **6.87** m Max: **9.22** m
	Min: 2.20 m Avg: 2.54 m Max: 2.86 m	Min: **4.81** m Avg: **5.48** m Max: **6.25** m
	Min: 0.00 m Avg: 0.00 m Max: 0.00 m	Min: **6.80** m Avg: **7.25** m Max: **7.83** m
	Min: 2.12 m Avg: 2.25 m Max: 2.44 m	Min: **7.62** m Avg: **6.87** m Max: **8.33** m
	Min: 1.28 m Avg: 2.06 m Max: 2.44 m	Min: **6.55** m Avg: **7.51** m Max: **8.17** m
	Min: 0.00 m Avg: 0.63 m Max: 2.51 m	Min: **10.58** m Avg: **11.28** m Max: **14.96** m
	Min: 0.00 m Avg: 0.26 m Max: 1.29 m	Min: **6.91** m Avg: **12.55** m Max: **14.63** m
	Min: 0.00 m Avg: 0.67 m Max: 4.11 m	Min: **6.17** m Avg: **10.09** m Max: **13.25** m

IPA-II



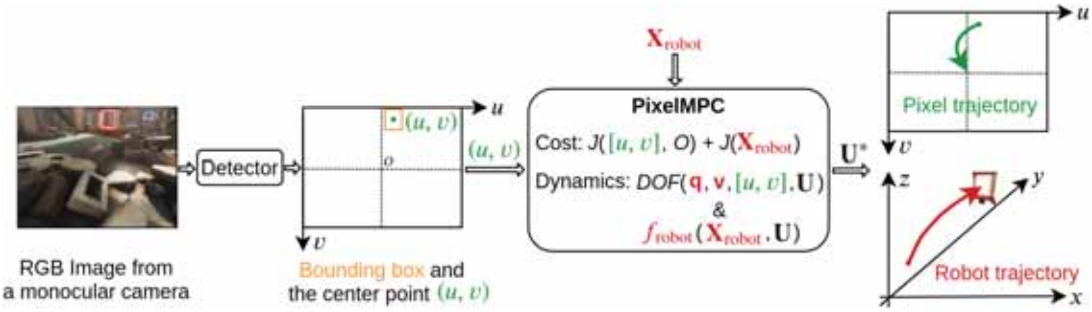
IPA-III



IPA-III

AI in Aerospace Systems

IPA-IV: PixelMPC



Augmented Dynamics

Drone Dynamics

$$\dot{\mathbf{p}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = \mathbf{g} + m^{-1}(\mathbf{R}_b^{\omega}\mathbf{f}_T + \mathbf{f}_D + \mathbf{w}_\mathbf{f})$$

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Pixel Dynamics

$$\begin{aligned}\dot{\mathbf{X}}_{\text{pixel}} &= F_{\text{pixel}}(\mathbf{q}, \mathbf{X}_{\text{pixel}}, \mathbf{U}) \\ &= \text{PolarToEuler}(DOF(\mathbf{q}, \mathbf{X}_{\text{pixel}}, \mathbf{U}))\end{aligned}$$

IPA-IV: PixelMPC

Outline

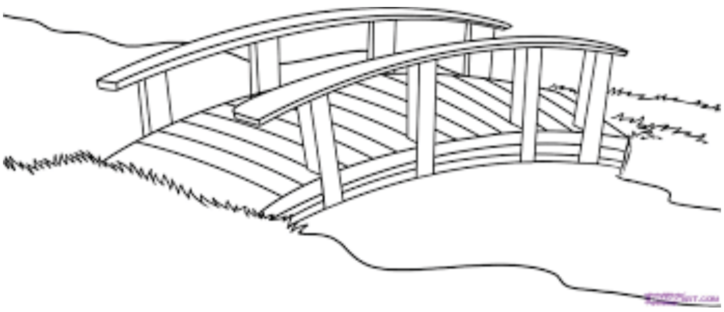
- ✦ Motivation & Intro
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ **Conclusions and Future**

Outline

- ✦ Motivation & Intro
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusions and Future

Decision Making Architectures

Partial Differential Equations



Stochastic Differential Equations

Deep Neural Network Architectures

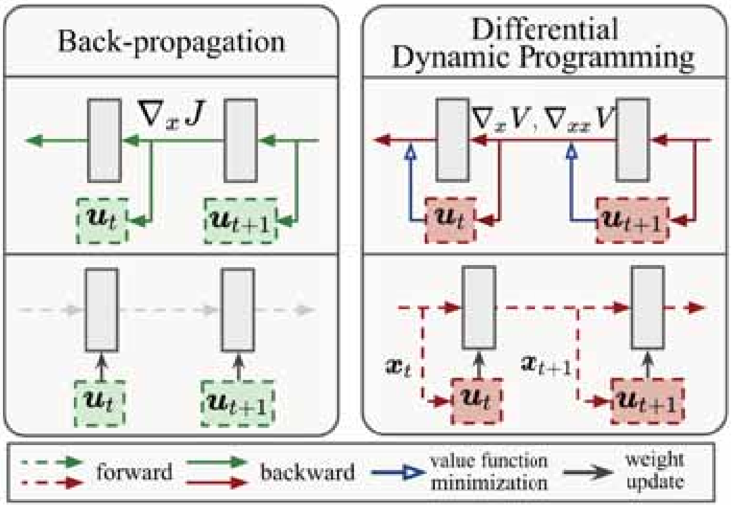
Stochastic Optimal Control ↔ Forward/Backward Stochastic Differential Equations (FBSDEs)

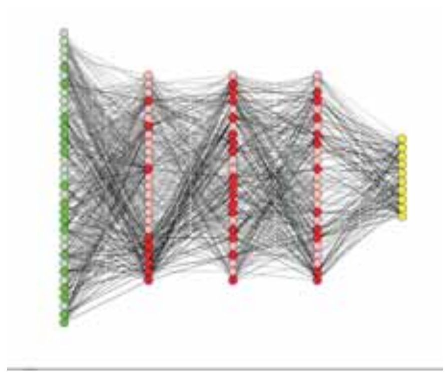
Perceptual Decision Making ↔ Risk Measures and Stochastic Differential Games

Perceptual Decision Making ↔ Control Barrier Functions & Barrier Certificates

Perceptual Decision Making ↔ Adaptive Control & Contraction Theory

Safety & Deep Learning Theory





Cost Function

$$\min_{\mathbf{u}} J(\bar{\mathbf{u}};\mathbf{x}_0) = \min_{\mathbf{u}} \left[\phi(\mathbf{x}_T) + \sum_{t=0}^{T-1} \ell_t(\mathbf{x}_t,\mathbf{u}_t)\right]$$

Dynamics

$$\mathbf{x}_{t+1} = f_t(\mathbf{x}_t,\mathbf{u}_t)$$

Optimal Control	Deep Learning
State	Output Activation
Controls	Weights
Time Horizon	Number of Layers
Terminal Cost	Loss Functions

Autonomous Control and Decision Systems Lab

Students:

Collaborators:

Jim Regh - Georgia Tech

Naira Hovakimyan - UIUC

Ali-akbar Agha-mohammadi
JPL - NASA

NSF

Sandia National Laboratories

amazon web services

NASA

QUALCOMM

Vertical Lift Research
Center of Excellence

Georgia Tech Robotics & Intelligent Machines

KOMATSU