Online Learning for Characterizing Unknown Environments in Ground Robotic Vehicle Models

Alec Koppel[⋆], Jonathan Fink[†], Garrett Warnell[†], Ethan Stump[†], Alejandro Ribeiro[⋆]

[⋆]Dept. of ESE, University of Pennsylvania, Philadelphia, PA

[†]U.S. Army Research Laboratory, Adelphi, MD

IEEE Conference on Intelligent Robots and Systems (IROS)

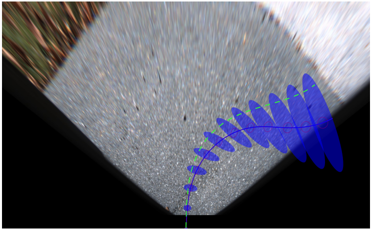
Daejeon, South Korea, Oct. 11, 2016

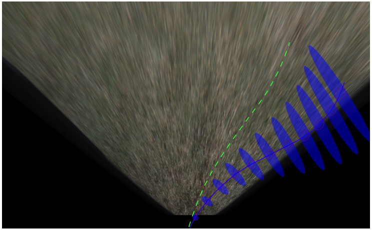
Learning in autonomous systems

- What role should learning play in deploying autonomous robots?
- Simplified physics models used for control due to complexity issues
  $\Rightarrow$ Models are available. They're not perfect but not useless either
  $\Rightarrow$ Replace mechanical models with learned models
- Learn mismatch between model and reality when
  $\Rightarrow$ This mismatch has variability across different terrains
- Use sensory input to learn uncertainty in execution of control actions

Model-based navigation of a wheeled robot

- Robot making a turn in pavement (left) and grass (right)
- Actual trajectory (green) $\neq$ Trajectory predicted by model (red)



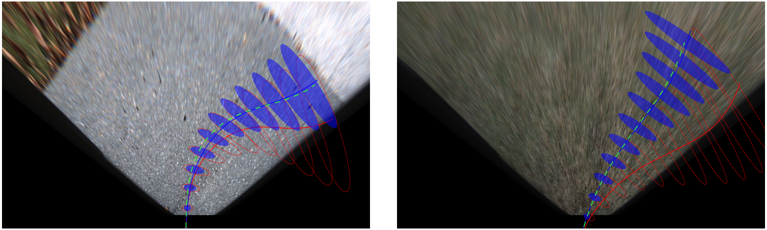


- Add uncertainty ellipses to mechanical model

  $\Rightarrow$ But uncertainty ellipses can't capture difference in environments

Learning-based navigation of a wheeled robot

- Robot making a turn in pavement (left) and grass (right)
- Actual trajectory (green) $\neq$ Trajectory predicted by model (red)



- Learn uncertainty during online field operation (using camera input)
  $\Rightarrow$ Learnt uncertainty ellipses are different in grass and pavement
- Learning $\Rightarrow$ online task-driven dictionary learning

Formulating Disturbance

- Consider a discrete nonlinear state-space system of equations

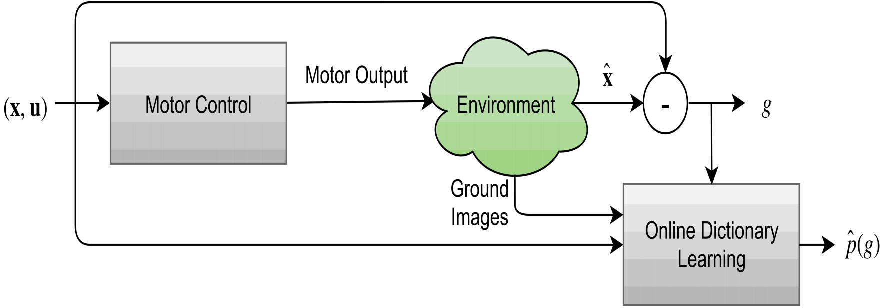
$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + g(\mathbf{a}_k) = f(\mathbf{x}_k, \mathbf{u}_k) + g(\mathbf{u}_k, \mathbf{z}_k)$$

- $\mathbf{x}_k \Rightarrow$ state vector, $\mathbf{u}_k \Rightarrow$ control input, $\mathbf{z}_k \Rightarrow$ sensory input
- Kinematic model $f(\mathbf{x}_k, \mathbf{u}_k)$ not exact $\Rightarrow$ add mismatch term $g(\mathbf{a}_k)$
  - $\Rightarrow$ Want to learn $g(\mathbf{a}_k)$ to use as input to robust control block
- Measure estimate $\hat{\mathbf{x}}_k$ of state $\mathbf{x}_k$ (with on-board IMU, for instance)

$$\hat{g}(\mathbf{a}_{k-1}) = \hat{\mathbf{x}}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \ .$$

- Learning $\hat{g}(\mathbf{a}_{k-1})$ is challenging
  - $\Rightarrow$ Captures difficult-to-model physics we typically ignore
  - $\Rightarrow$ Dictionary leaning approach

Dictionary-Based Learning Architecture



- Platform's state $\mathbf{x}$, control $\mathbf{u}$ intended by a kinematic planner
  $\Rightarrow$ differ from measured ground truth $\hat{\mathbf{x}}$ by disturbance g
- This difference, as well as state, control, and visual features
  $\Rightarrow$ Fed into dictionary learning method $\Rightarrow$ disturbance prediction $\hat{g}$
  $\Rightarrow$ Dictionary is a statistical model using sparse approximation

Parametric representation of disturbance

- Model disturbance $\hat{g}(\mathbf{a})$ as Gaussian conditional on feature vector $\mathbf{a}$

$$\mathbb{P}[\hat{g}(\mathbf{a}) \mid \mathbf{a}] = \frac{1}{\sqrt{2\pi\sigma^2(\mathbf{a})}} \exp\left[-\frac{(\hat{g}(\mathbf{a}) - \mu(\mathbf{a}))^2}{2\sigma^2(\mathbf{a})}\right] .$$

- Distribution parameterized by unknown mean $\mu(\mathbf{a})$, var. $\sigma^2(\mathbf{a})$
  $\Rightarrow$ which depend on control $\mathbf{u}_k$ and sensory input $\mathbf{z}_k$
- Realizations of $(\mathbf{a}, \hat{g}(\mathbf{a}))$ available online
- Sequentially obtained while exploring feature space
- Utilize to learn parametric representation of the distribution

Disturbance Prediction as Dictionary Learning

- Learn online mean, variance $\Rightarrow$ introduce regressors $\mathbf{w}_1$, $\mathbf{w}_2$
  $\Rightarrow$ predict first, second-order stats. $\mu(\mathbf{a})$ and $\sigma^2(\mathbf{a})$, given signal $\mathbf{a}$

$$\hat{\mu}(\mathbf{a}) = \mathbf{w}_1^T \mathbf{a}\ , \quad \hat{\sigma}^2(\mathbf{a}) = \sigma^2_{\text{min}} + \left(\mathbf{w}_2{}^T \mathbf{a} + \sigma^2_{\text{init}}\right)^2$$

- Rather than use $\mathbf{a}$ directly, use a sparse code $\boldsymbol{\alpha}^*(\mathbf{D};\mathbf{a})$

$$\hat{\mu}(\mathbf{a}) = \mathbf{w}_1^T \boldsymbol{\alpha}^*(\mathbf{D};\mathbf{a})\ , \quad \hat{\sigma}^2(\mathbf{a}) = \sigma^2_{\text{min}} + \left(\mathbf{w}_2{}^T \boldsymbol{\alpha}^*(\mathbf{D};\mathbf{a}) + \sigma^2_{\text{init}}\right)^2$$

  $\Rightarrow$ Regress on sparse approximation $\boldsymbol{\alpha}^*(\mathbf{D};\mathbf{a})$ w.r.t. dictionary $\mathbf{D}$

- Motivation for using sparse code $\boldsymbol{\alpha}^*(\mathbf{D};\mathbf{a})$ and learning dictionary $\mathbf{D}$:
  $\Rightarrow$ $g(\cdot)$ relates robotic sensory perception and unexpected dynamics
  $\Rightarrow$ Relationship between $(\mathbf{a}, \hat{g}(\mathbf{a}))$ expected to be highly nonlinear
  $\Rightarrow$ Estimation accuracy $\Rightarrow$ boosted via alternative feature encoding

Task-Driven Dictionary Learning

- Dictionary $\mathbf{D} = \{\mathbf{d}_l\}_{l=1}^m$, $\mathbf{d}_l \in \mathbb{R}^k$ composed of m basis elements
- Estimate $\hat{\mathbf{a}}_k = \mathbf{D}\boldsymbol{\alpha}_k$ as linear combo. of dictionary elements
- Select coefficients that yield a sparse code (elastic net)

$$\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{a}_k) := \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^k} \|\mathbf{a}_k - \mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \nu\|\boldsymbol{\alpha}\|_2^2$$

- Jointly learn dictionary and regressors $\mathbf{w}_1$ and $\mathbf{w}_2$

$$(\mathbf{D}^*, \mathbf{w}_1^*, \mathbf{w}_2^*) := \operatorname*{argmin}_{\mathbf{D} \in \mathcal{D}, \mathbf{w}_1, \mathbf{w}_2} \mathbb{E}_{\mathbf{a}, \hat{g}(\mathbf{a})}\Big(-\log \mathbb{P}[\hat{g}(\mathbf{a}) \,|\, \mathbf{a}, \mathbf{D}, \mathbf{w}_1, \mathbf{w}_2]\Big).$$

- Nonconvex but convex w.r.t. $\mathbf{D}$ and $\mathbf{w}_1$ and $\mathbf{w}_2$ separately
- Objective is an expectation over dataset $\Rightarrow$ use stochastic gradients

Learning Algorithm Summary

- Observe signals $\mathbf{z}_k$, use past control $\mathbf{u}_k$ to compute coding

$$\boldsymbol{\alpha}_k^* := \underset{\boldsymbol{\alpha}_k \in \mathbb{R}^s}{\text{argmin}} (1/2)\|\mathbf{a}_k - \mathbf{D}\boldsymbol{\alpha}_k\|_2^2 + \lambda\|\boldsymbol{\alpha}_k\|_1 + \nu\|\boldsymbol{\alpha}_k\|_2,$$

$\Rightarrow$ Update dictionary using stoch. grad. step w.r.t. dictionary

$$\mathbf{D}_{k+1} = \mathbf{D}_k - \epsilon_k \left(\nabla_{\mathbf{D}} \log \mathbb{P}[\hat{g}(\mathbf{a}_k) \,|\, \mathbf{a}_k, \mathbf{D}_k, \mathbf{w}_{1,k}, \mathbf{w}_{2,k}]\right)$$

- Update regressors along regressor gradient of loss function

$$\mathbf{w}_{1,k+1} = \mathbf{w}_{1,k} + \epsilon_k \left(\nabla_{\mathbf{w}_1} \log \mathbb{P}[\hat{g}(\mathbf{a}_k) \,|\, \mathbf{a}_k, \mathbf{D}_k, \mathbf{w}_{1,k}, \mathbf{w}_{2,k}]\right) \,,$$
$$\mathbf{w}_{2,k+1} = \mathbf{w}_{2,k} + \epsilon_k \left(\nabla_{\mathbf{w}_2} \log \mathbb{P}[\hat{g}(\mathbf{a}_k) \,|\, \mathbf{a}_k, \mathbf{D}_k, \mathbf{w}_{1,k}, \mathbf{w}_{2,k}]\right) \,,$$

- Converges to locally optimal dictionary and regressors

Implementation on a Ground Robot

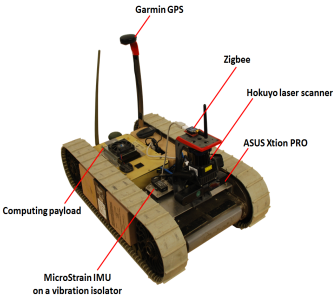


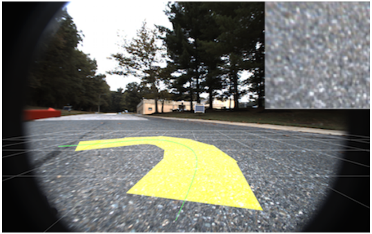
Figure: An iRobot *Packbot* was used in our experiments. It was additionally configured with a high-resolution camera.

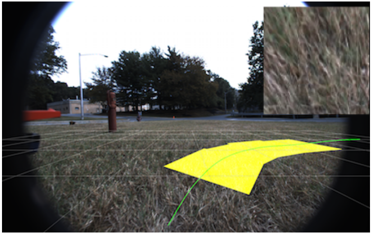
- We consider a differential drive model of a skid-steer robot

$$f(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) & 0 \\ \sin(\theta_k) & \cos(\theta_k) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A(\theta)} \begin{bmatrix} \boldsymbol{\nu}_k \\ \boldsymbol{\omega}_k \end{bmatrix}$$

- Disturbance $\Rightarrow$ difference of commanded & actual angular velocity

- Visual patch $\mathbf{z}_k$ $\Rightarrow$ associated with the portion of ground
  - $\Rightarrow$ Collect images over time horizon of planned robot trajectory





- From the raw patch we construct statistical visual features $\mathbf{c}_k$
  - $\Rightarrow$ mean, variance, skewness, kurtosis of each RGB color channel
  - $\Rightarrow$ Textures $\mathbf{h}_k$ via texton histogram (Leung '99)
- Concatenated with average linear, angular velocity in slot $[k, k+1]$

Empirical Performance Comparison

- Model fit of disturbance to task driven dictionary learnt distribution
- Compared to (windowed) recursive average of mean and variance

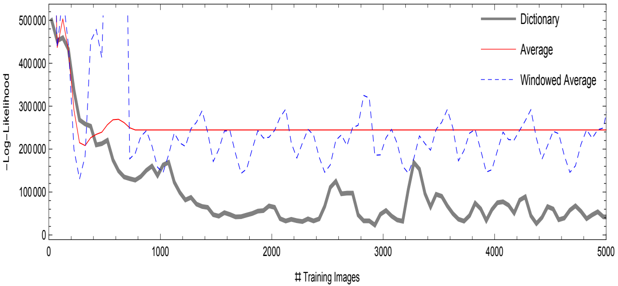


Figure: Comparison of dictionary learning vs. classical alternatives.

- Superior model fit to Gaussian approximation of model disturbance
- Exploits sensory input to identify terrain type

  $\Rightarrow$ Pavement or grass essentially. But more granular than that

Empirical Adaptivity

- Terrain-specific performance as training switches between terrains.
- Prediction of disturbance on grass (green) and pavement (gray)

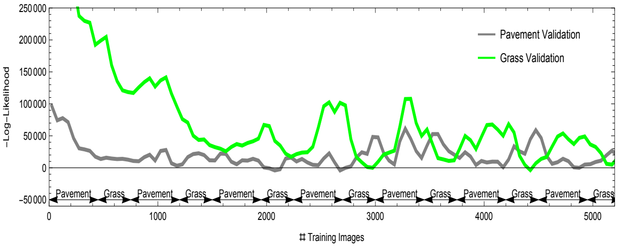


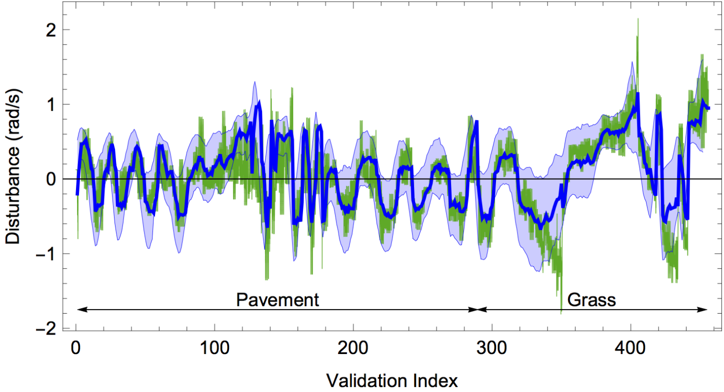
Figure: Comparison of dictionary learning vs. classical alternatives.

- Performance decreases on one terrain when other is encountered
- But this happens only initially until a descriptive dictionary is learnt.

  ⇒ Task-driven dictionary allows for prediction on either terrain

Predicting Future Uncertainty

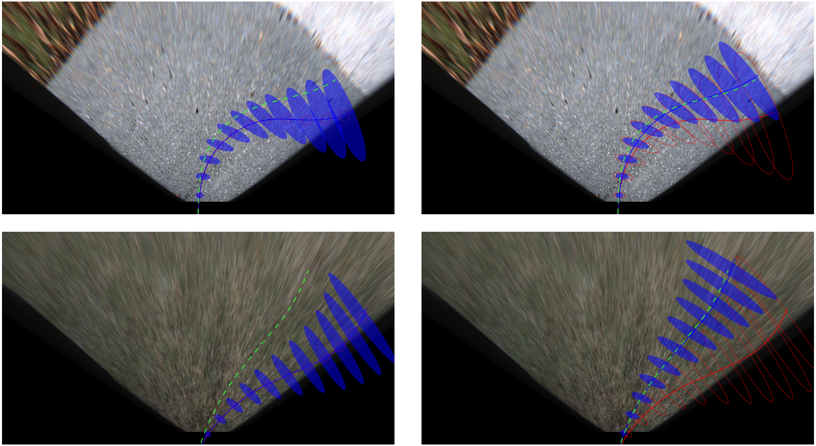
- Test trajectory with predicted & actual disturbance statistics overlaid
- Measured dist. (green) and predicted dist. (blue) for trajectory
  - $\Rightarrow$ Predicted mean and $\pm 2\sigma$ envelope shown



- Observations are mostly contained within confidence envelopes

Future Uncertainty Cones

- Actual trajectory not contained within cones for initial dictionary
  - ⇒ But contained within cone after dictionary is properly learnt



Conclusion

- Online dictionary learning technique
  - $\Rightarrow$ predictions of model disturbance distribution
  - $\Rightarrow$ generate these predictions from control signals, visual features
- Bypass the need for terrain classification
- Disturbance prediction $\Rightarrow$ important to modern/robust controllers
- Promising empirical results
  - $\Rightarrow$ encourage implementation on field robotic systems
  - $\Rightarrow$ motivate experimentation with more aggressive controllers

References

- A. Koppel, J. Fink, G. Warnell, E. Stump, and A. Ribeiro, "Online Learning for Characterizing Unknown Environments in Ground Robotic Vehicle Models," in Proc. Int. Conf. Intelligent Robotics and Systems, Daejeon, Korea, Oct9-Oct14 2016

- IJRR version of this work in preparation