SAND2015-20826C

Aerial Coffee Delivery: Preference Appraisal Reinforcement Leaning in Presence of Stochastic Input Disturbances for Control-affine Systems

Aleksandra Faust, Sandia National Laboratories

Lydia Tapia, Department of Computer Science, University of New Mexico

January 2015

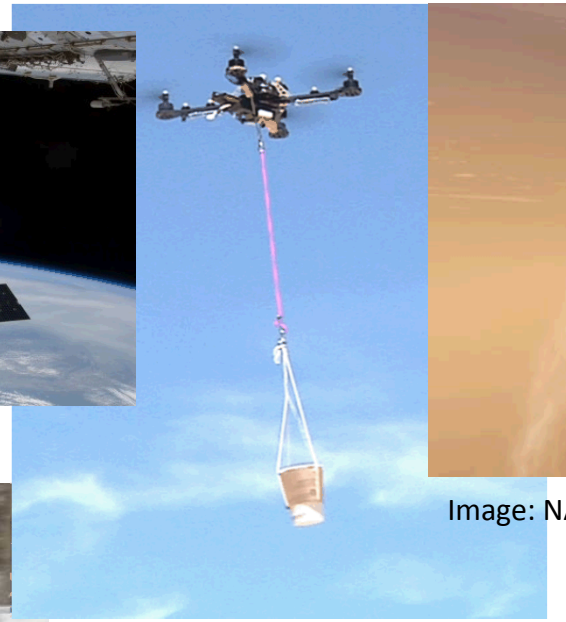
UNM

Sandia National Laboratories

Preference Balancing Tasks (PBTs)

- Characteristics
  - Robotic motion
  - High-stakes
  - Complicated dynamics
  - Described with preferences
- Develop, analyze, and evaluate a solution for robotic PTBs
- System
  - High-dimensional
  - Acceleration-controlled
  - Unknown control-affine dynamics
- Solution
  - Efficient
  - Safe

Image: SpaceX



[Faust et al., 2013]

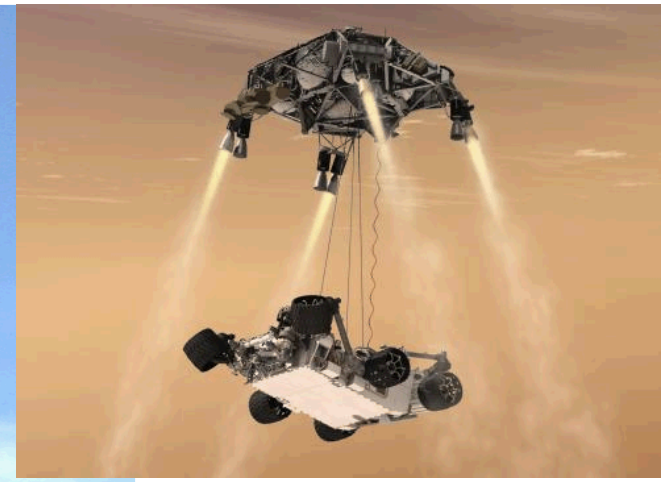
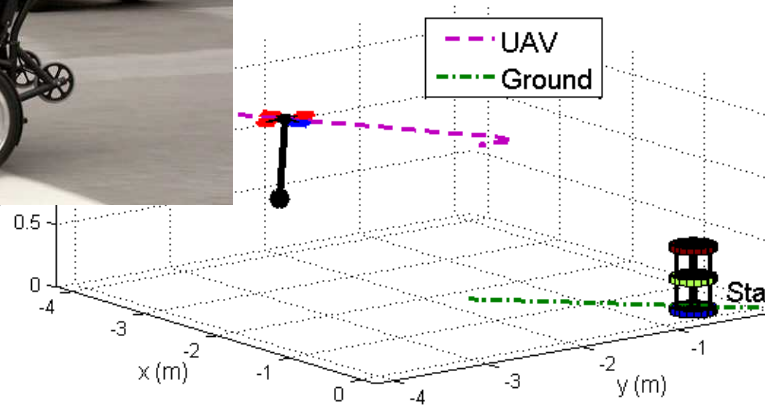
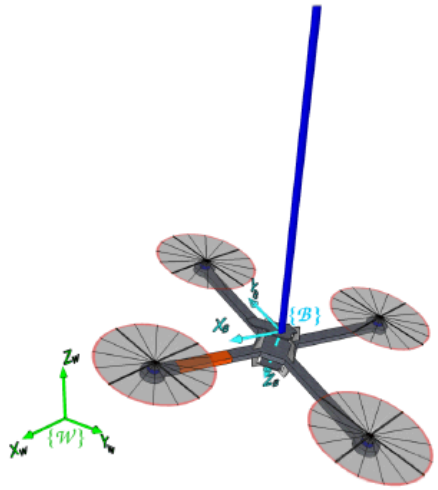


Image: NASA/JPL/Caltech

Image: GM



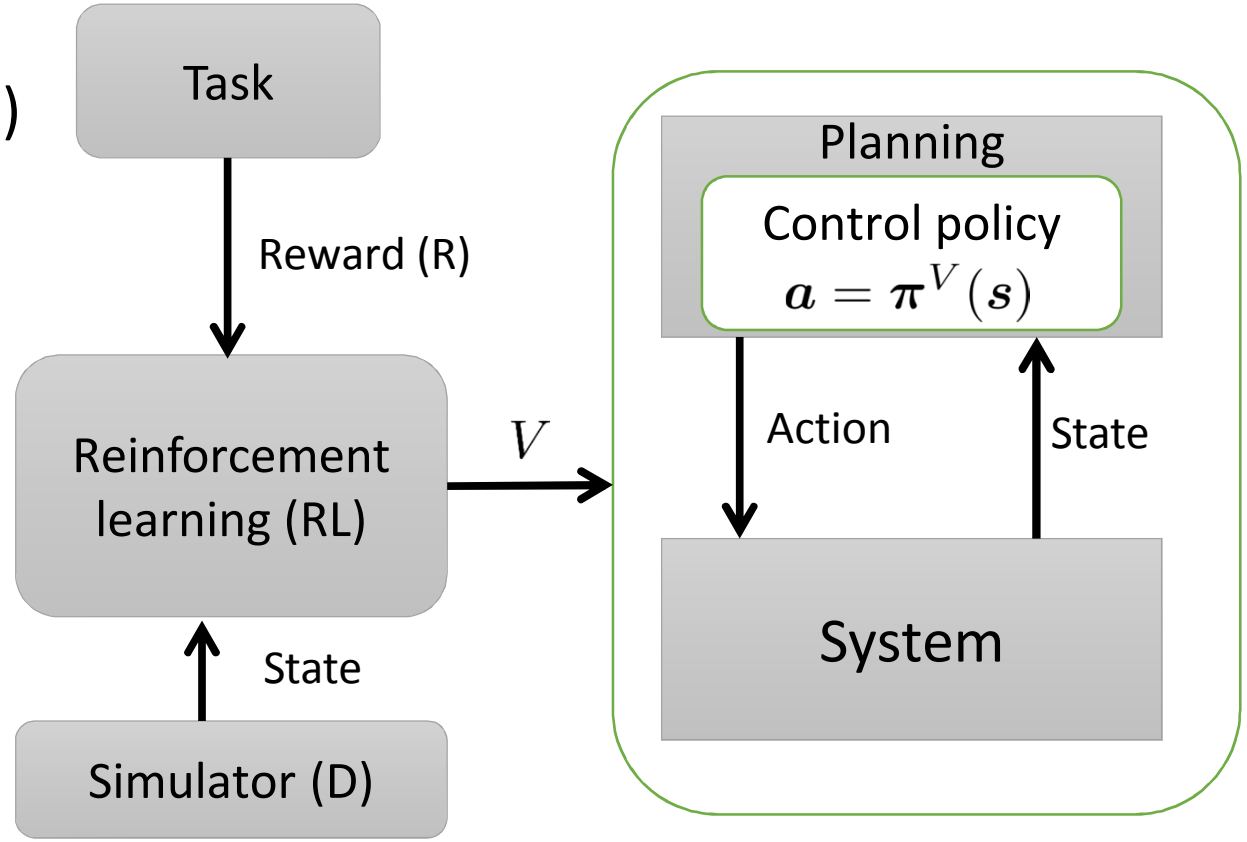
[Faust et al., 2015]



[Figueroa et al., 2014]

Reinforcement learning

- Solves
  - Markov Decision Process MDP (S,A,D,R)
  - Unknown D and R
- State-value function $V : S \rightarrow R$
  - Cost to go
  - Potential for accumulated reward
- Planning $\pi^V : S \rightarrow A$
  - Greedy policy $\pi^V(s) = \text{argmax}_{a \in A} V(s')$
  - Action sequence that maximizes the value
  - s' is result of applying action a to state s



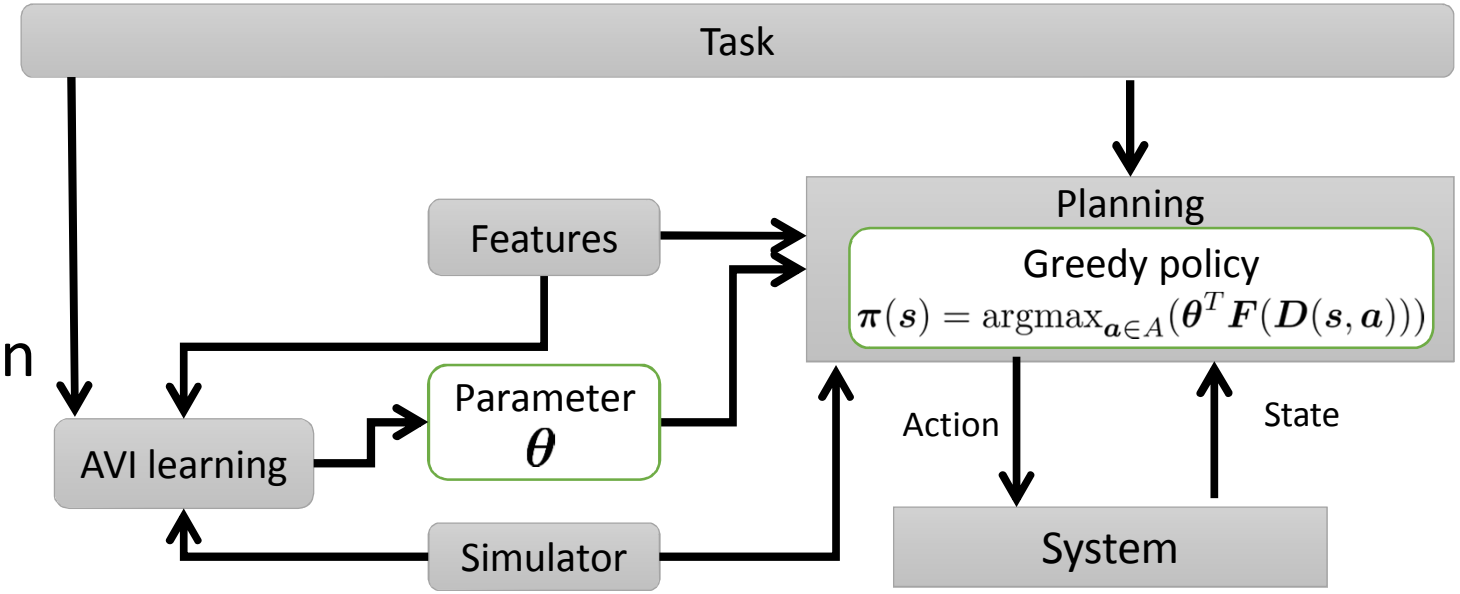
[Barto and Sutton '98]

Approximate Value Iteration (AVI) *[Ernst et al. '95]*

- Batch Reinforcement Learning
  - Deterministic planning
- Continuous state Markov Decision Process (MDP)
- State-value function approximation

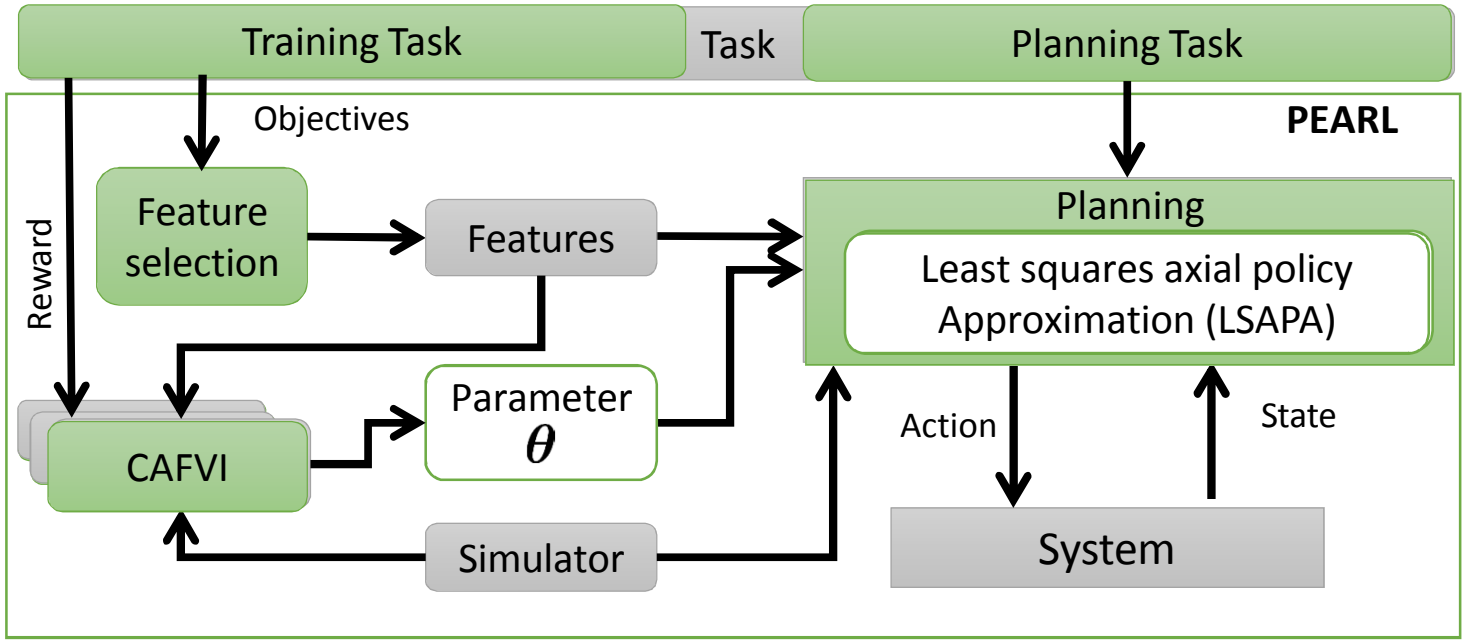
$$V(\boldsymbol{s}) = \boldsymbol{\theta}^T F(\boldsymbol{s})$$

- Learns feature vector weights
- AVI Algorithm
  - Iteratively
    - Sample states and observe rewards $(\boldsymbol{s}_i, r_i),\ i = 1, \ldots, n$
    - Update state value $v_i = v_i + r_i + \gamma \max_{\boldsymbol{a} \in A} \boldsymbol{\theta}^T \boldsymbol{F}(\boldsymbol{s}'_i)$
    - Find new parameter that fits the new observations $\boldsymbol{\theta} \leftarrow \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^{n} (v_i - \boldsymbol{\theta}^T \boldsymbol{F}(\boldsymbol{s}'_i))$



PrEference Appraisal Reinforcement Learning (PEARL)

- How to select features?
- Did we learn the right thing?
- How to generalize learning?
- What about continuous actions?
- How to adapt to external disturbances?

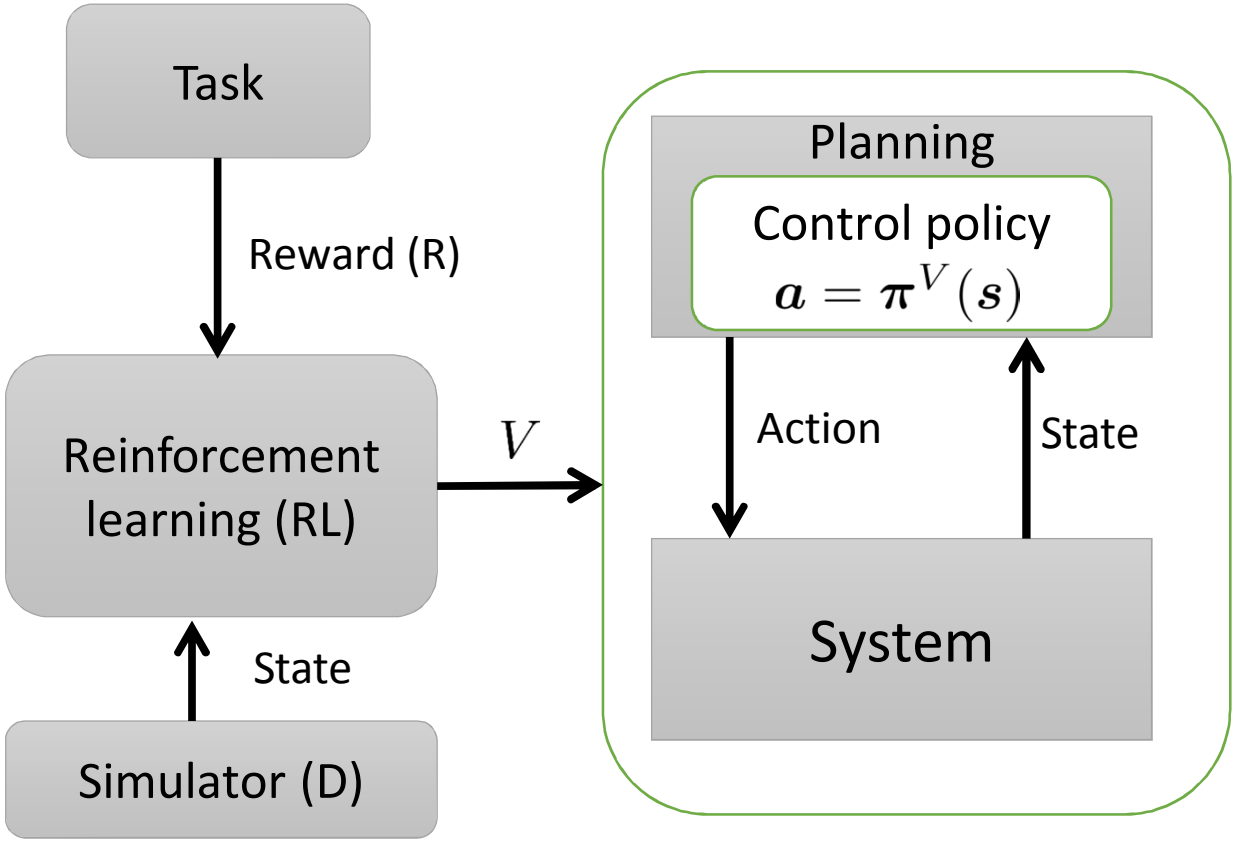


Related Work

- Feature selection
  - Radial bias functions [Busoniu et al. '10]
  - Space partitioning [Kimura '07]
  - Neural networks [Busoniu et al. '10]
- Lyapunov stable RL
  [DeCastro and Krass-Gazit '13]
  [Perkins and Barto '02]
  [Ogren et al. 02]
- Learning transfer
  [Sherstov and Stone '05]
  [Taylor and Stone '09]
  [McMahan et al. '05]
- Continuous action planning
  - HOOT [Mansley et al. '11]
  - Optimistic planning [Walsh et al. '10], [Busoniu et al. '13]
  - Gradient descent [Hasselt '11]
- RL with continuous actions
  [Bubeck et al. '11]
  [Madares et al. '13]
  [Dievks and Jagannathan '12]

PEARL Markov decision process (MDP)

- m agents, d DoFs each
- States, $S = \mathbb{R}^{2md}$
  - Joint agent-position-velocity
- Actions, $A = \mathbb{R}^{md}$
  - Joint acceleration space
- Transition function
  - System dynamics $\boldsymbol{s_{n+1}} = \boldsymbol{f}(\boldsymbol{s_n}) + \boldsymbol{g}(\boldsymbol{s_n})\boldsymbol{a_n}$
  - $\boldsymbol{f}$, $\boldsymbol{g}$ unknown and deterministic
- Reward
  - Measure of immediate state quality

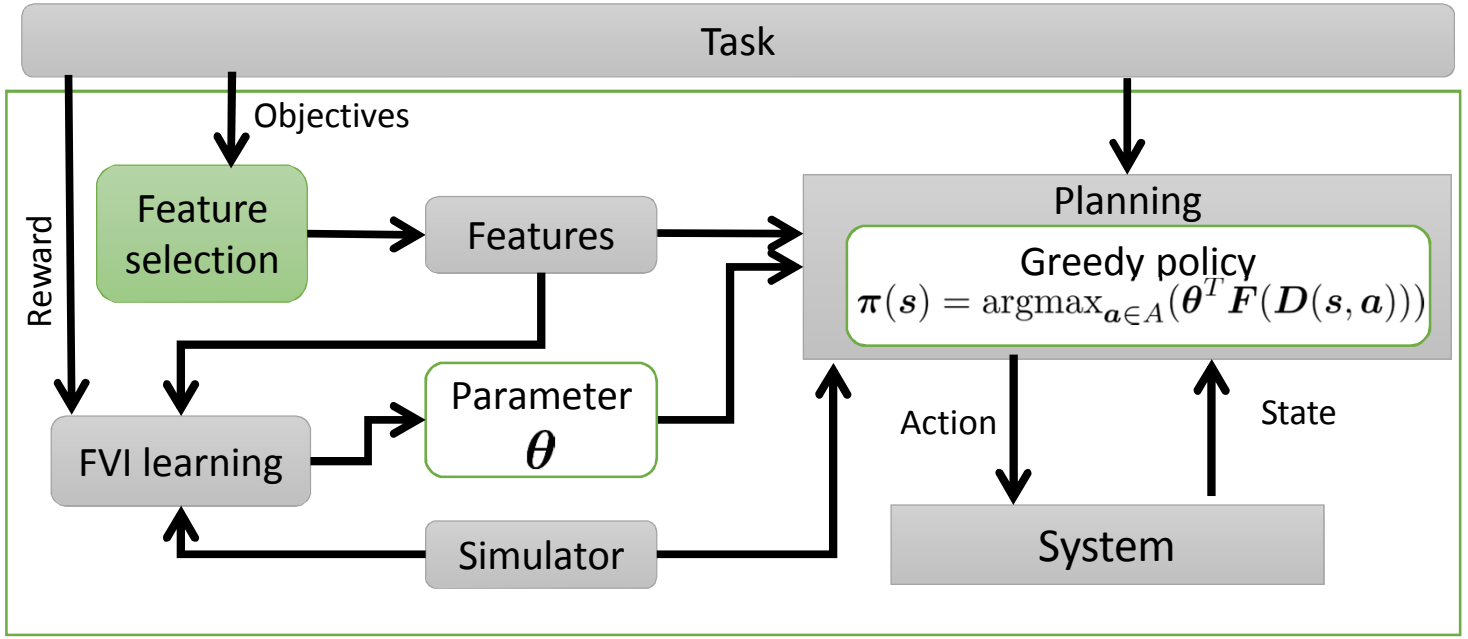


[Faust, 2014]

How to select features?

- Objectives, $\mathbf{o}_i$, applicable to agents $S_i \subset \{1, .., m\}$

- Preferences
  - $\mathbf{F}_i(\mathbf{s}, 2md) = \sum_{j \in Si} \left\| \mathbf{p}_j^{\mathbf{o}_i}(\mathbf{s}) - \mathbf{o}_i \right\|^2$

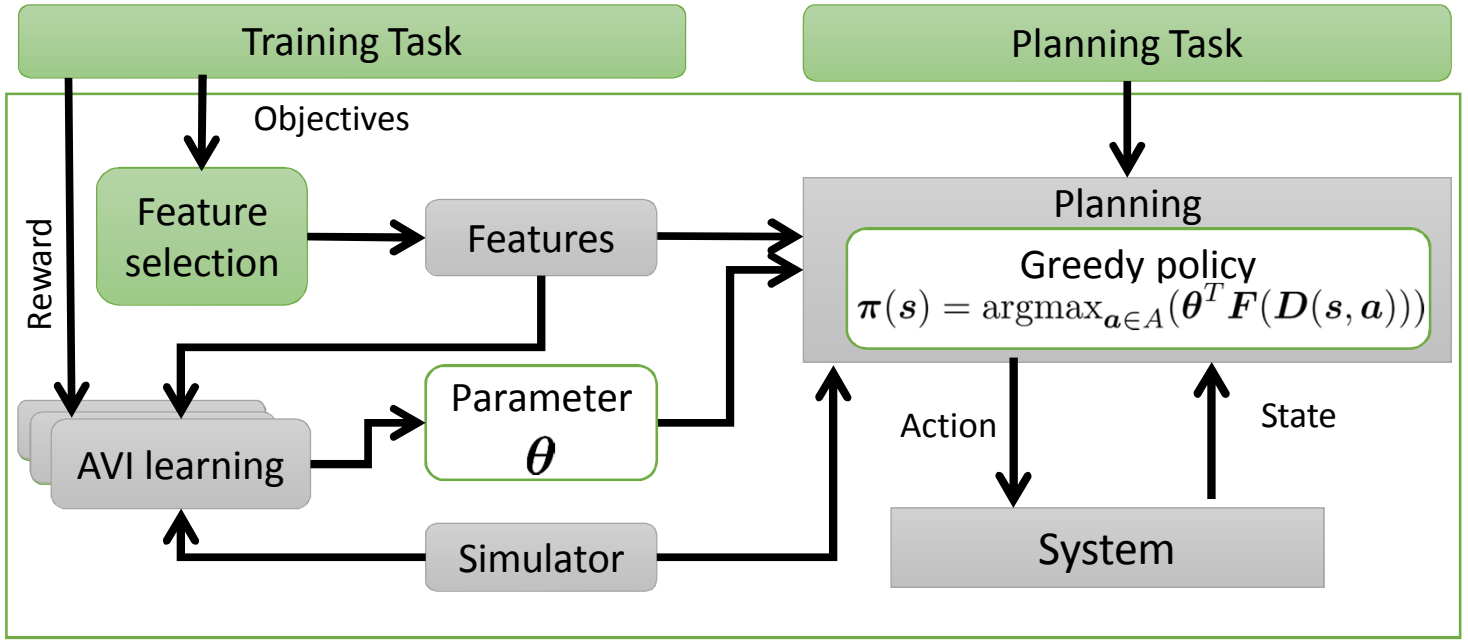


- Preferences
  - Polymorphic to the problem size
  - Polynomial computation time
  - Number does not change with the physical state size

[Faust, 2014]

How to generalize learning?

- Policy leads to the goal if
  - $\theta_i < 0$
  - *Can transition to a higher-valued* state
- State space changes
  - *Learn in small area*
  - *The policy viable starting at arbitrary position*
- Simulator and action space changes
  - *Preserve ability to transition to a higher valued state*

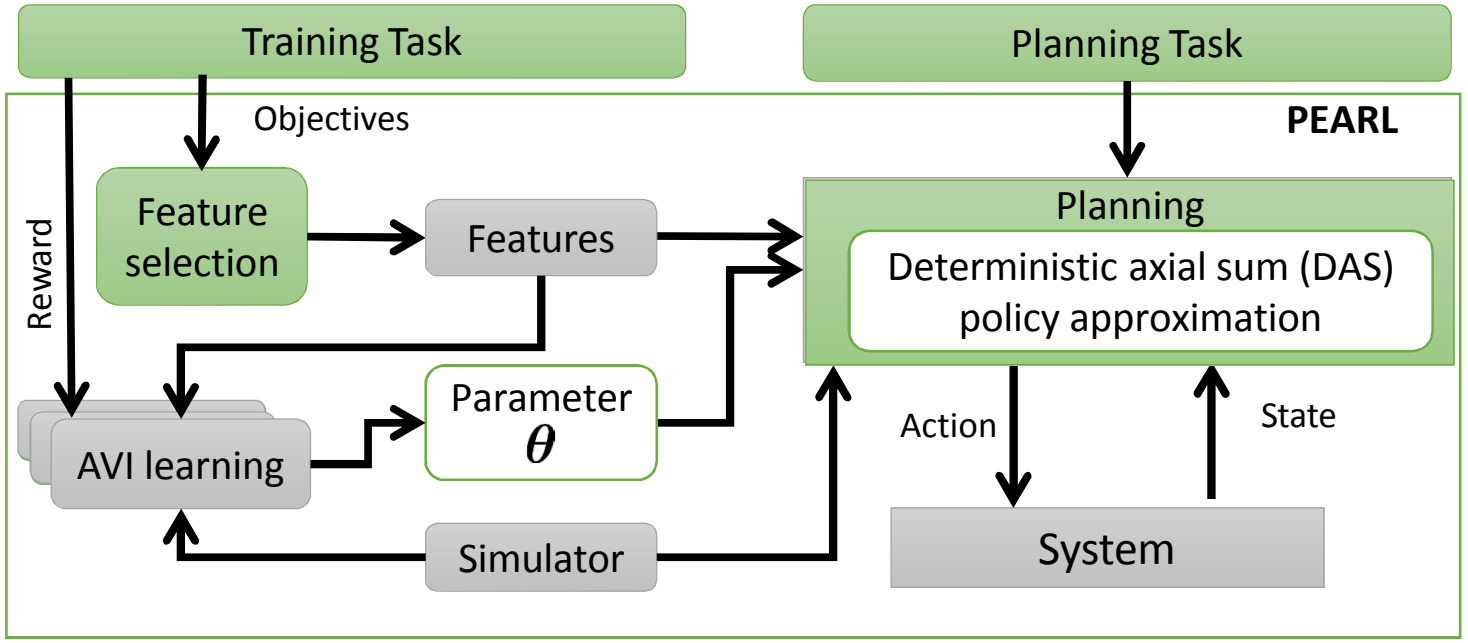


[Faust et al., 2013]

What about continuous actions?

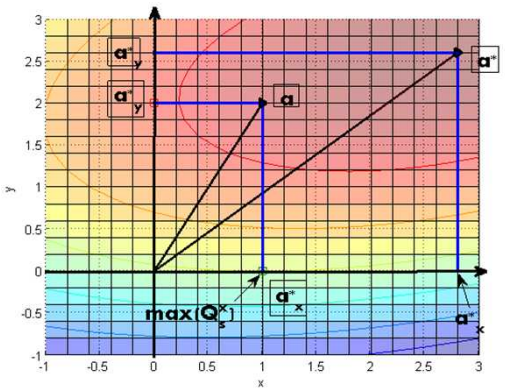
- Greedy policy $\pi(s) = \text{argmax}_{a \in A}(\theta^T F(s'))$

  is optimization problem
  - Unknown quadratic objective function
- Deterministic axial sum (DAS) policy
  - Sample objective function along axes
  - Interpolate and find maximum
  - Combine with a vector or convex sum
- Characteristic
  - Consistent
  - Efficient, $O(d_a d_s)$
  - Parallelizable
  - Negative weights are sufficient for convergence to goal



$$\hat{a}_i = \min(\max(\hat{a^*}_i, a_i^l), a_i^u), \text{ where}$$

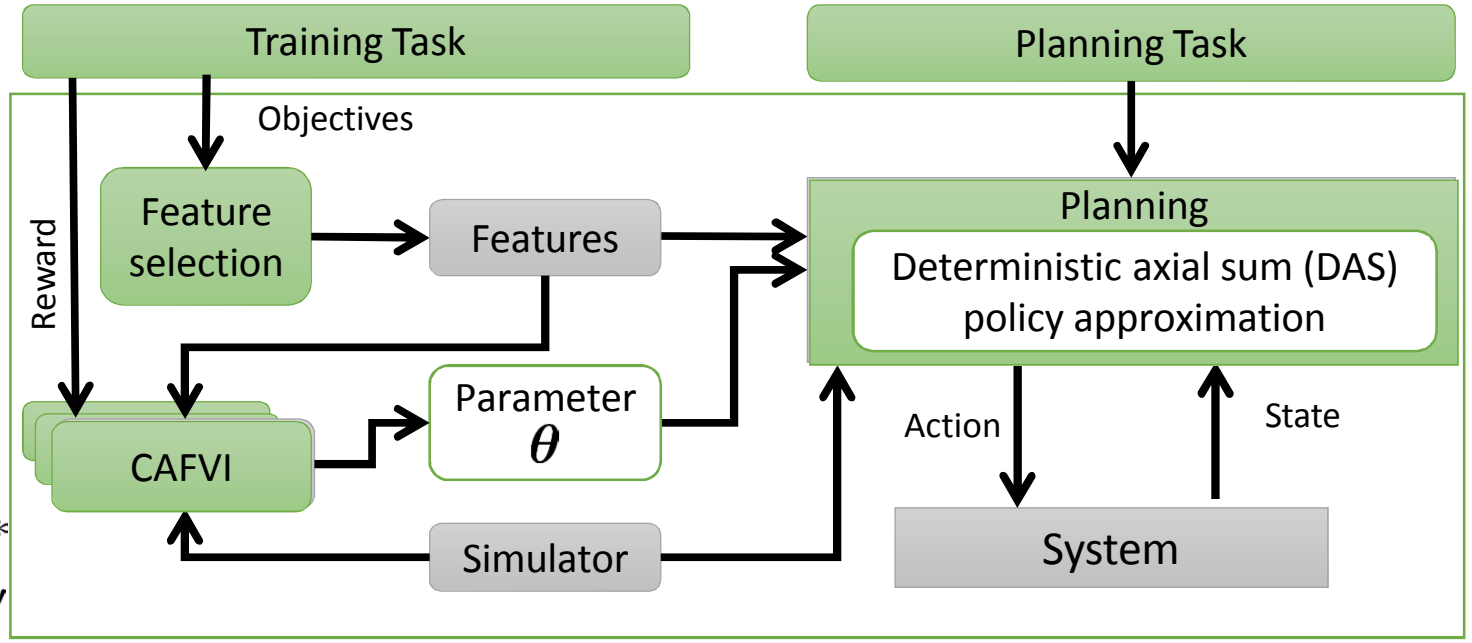
$$\hat{a}_i^* = \frac{\boldsymbol{q}_i^T \cdot ([a_{i2}^2 \quad a_{i3}^2 \quad a_{i1}^2] - [a_{i3}^2 \quad a_{i1}^2 \quad a_{i2}^2])^T}{2\boldsymbol{q}_i^T \cdot ([a_{i2} \quad a_{i3} \quad a_{i1}] - [a_{i3} \quad a_{i1} \quad a_{i2}])^T},$$



[Faust et al., 2015]

Continuous Action Fitted Value Iteration (CAFVI)

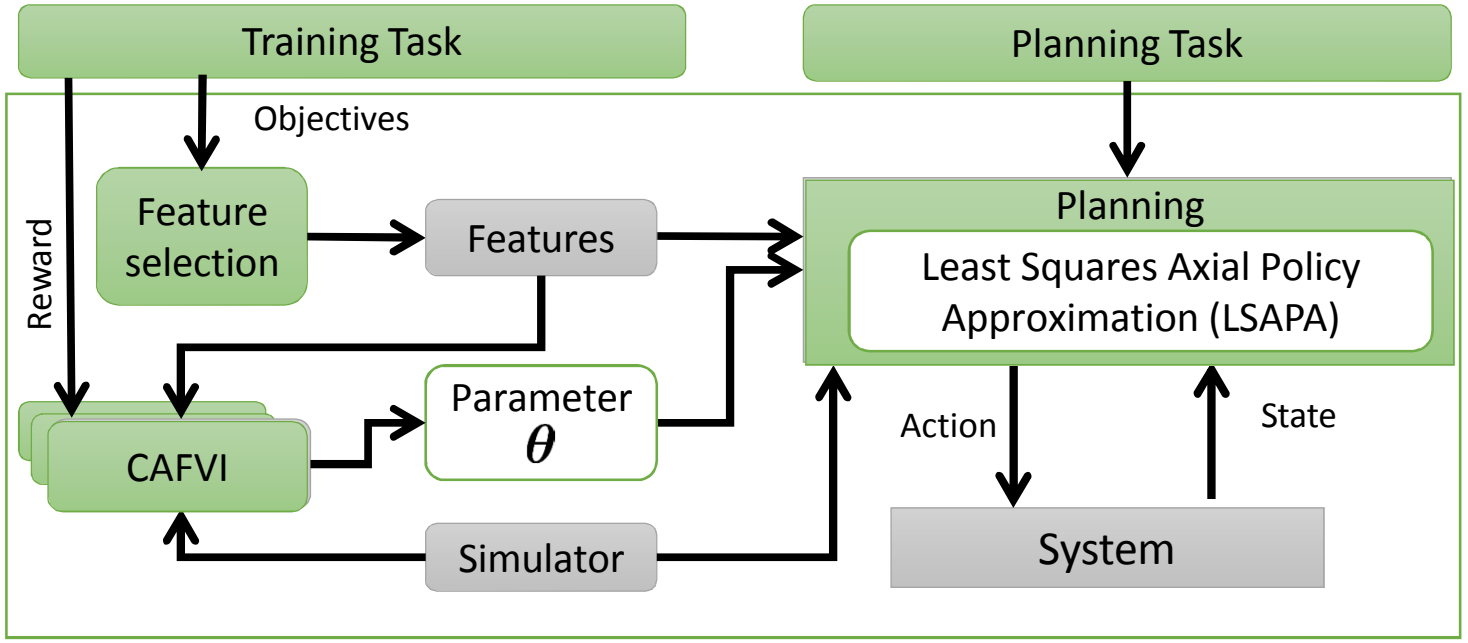
- Sample states
- For each state, s, using current weights
  - Observe reward
  - Find the best action wrt. DAS policy, a^*
  - Query simulator for a resulting state s' of applying a^* to s
  - Update state's value: reward + discounted value of s'
- Fit new weights through new observations

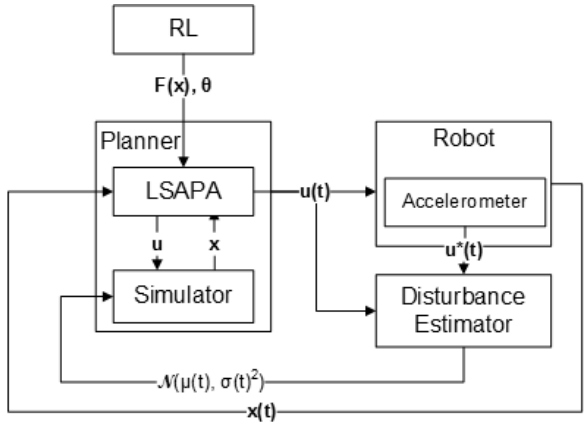


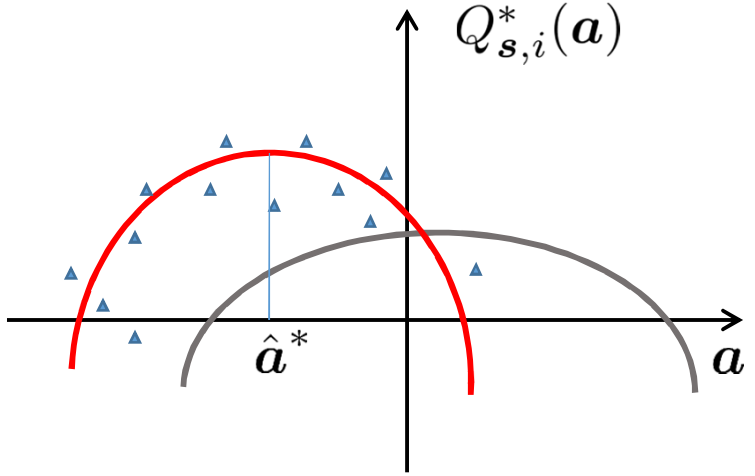
[Faust et al., 2015]

Least Squares Axial Policy Approximation (LSAPA)

- Problem $\pi(s) = \text{argmax}_{a \in A}(\theta^T F(D(s,a)))$
  - PBTs $V(s) = \theta^T F(s)$
  - Control-affine system with external input disturbance
    $s_{k+1} = f(s_k) + g(s_k)(a_k + \eta_k)$
- Learning
  - Deterministic CAFVI
- Planning
  - Estimate disturbance in real-time
  - Least Squares Axial Policy Approximation selects an action at every time step
  - Adapts to observed disturbance



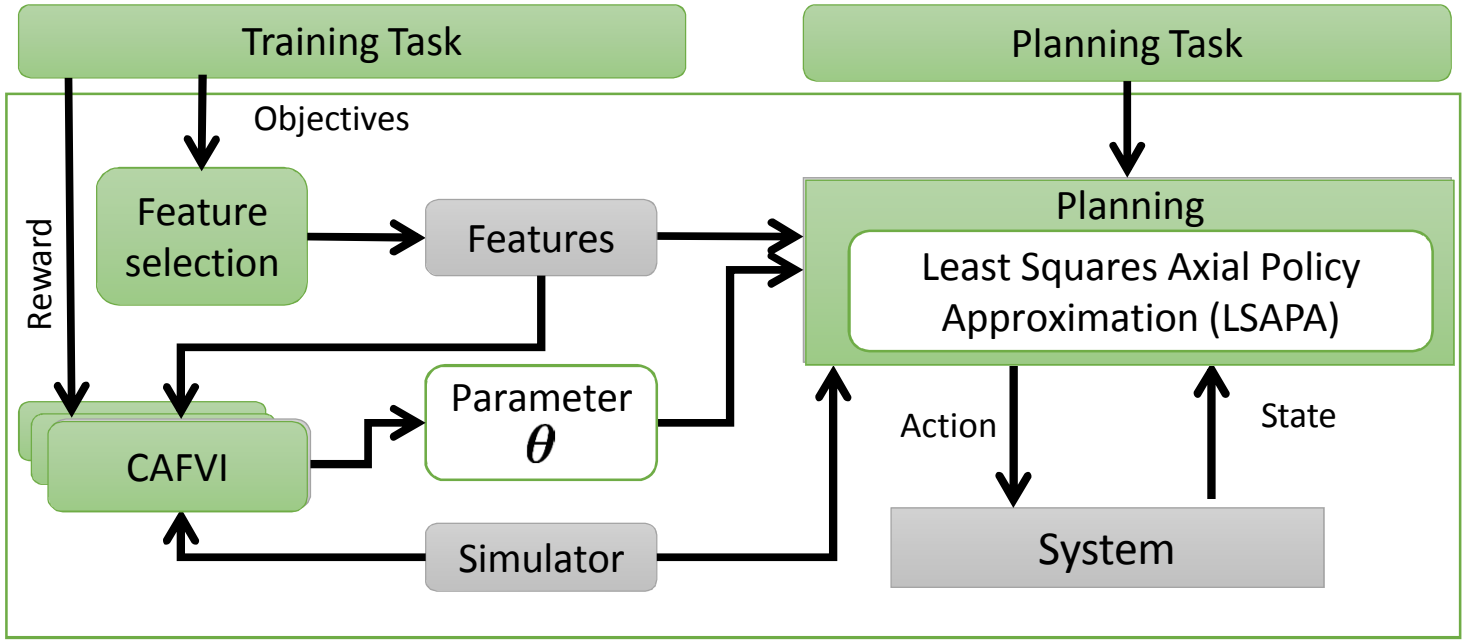




[Faust et al., 2014]

PEARL Summary

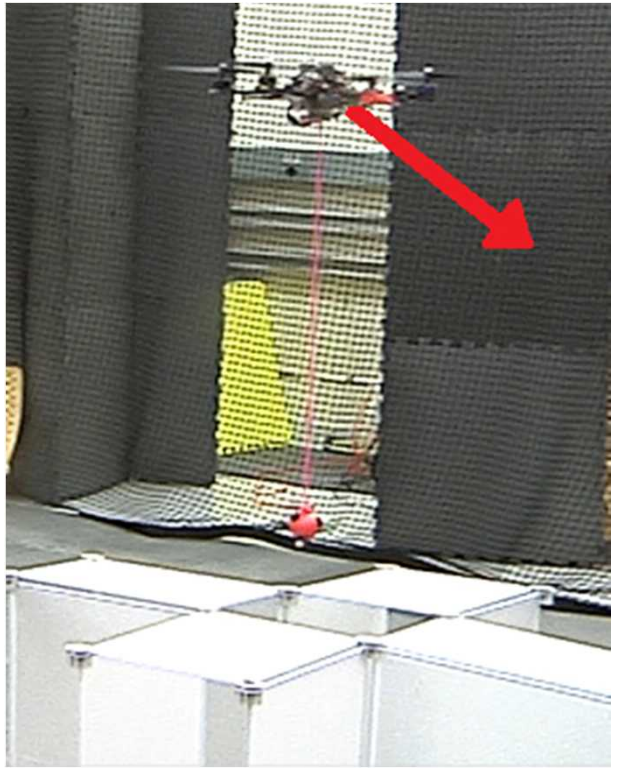
- Learns to perform PBTs* through interactions
- Learns on small problems, plans on large problems
- Efficient
- Autonomous
- Sufficient conditions for convergence to the goal in the deterministic case



*Preference Balancing Tasks

Coffee Delivery: Setup

- Problem
  - Holonomic cargo-bearing UAV
  - Bring the suspended load to the destination
  - Minimal residual load oscillations at arrival
- Previous solutions
  - Dynamic programming [Palunko et al. '12]
  - AVI [Faust et al. '13]
- Preferences, reduce
  - Distance from the destination
  - Vehicle's velocity
  - Load displacement
  - Load's velocity
- MDP
  - States: 10-dimensional vector space
  - Actions: 3-dimenstional vector space



[Faust et al., 2015]

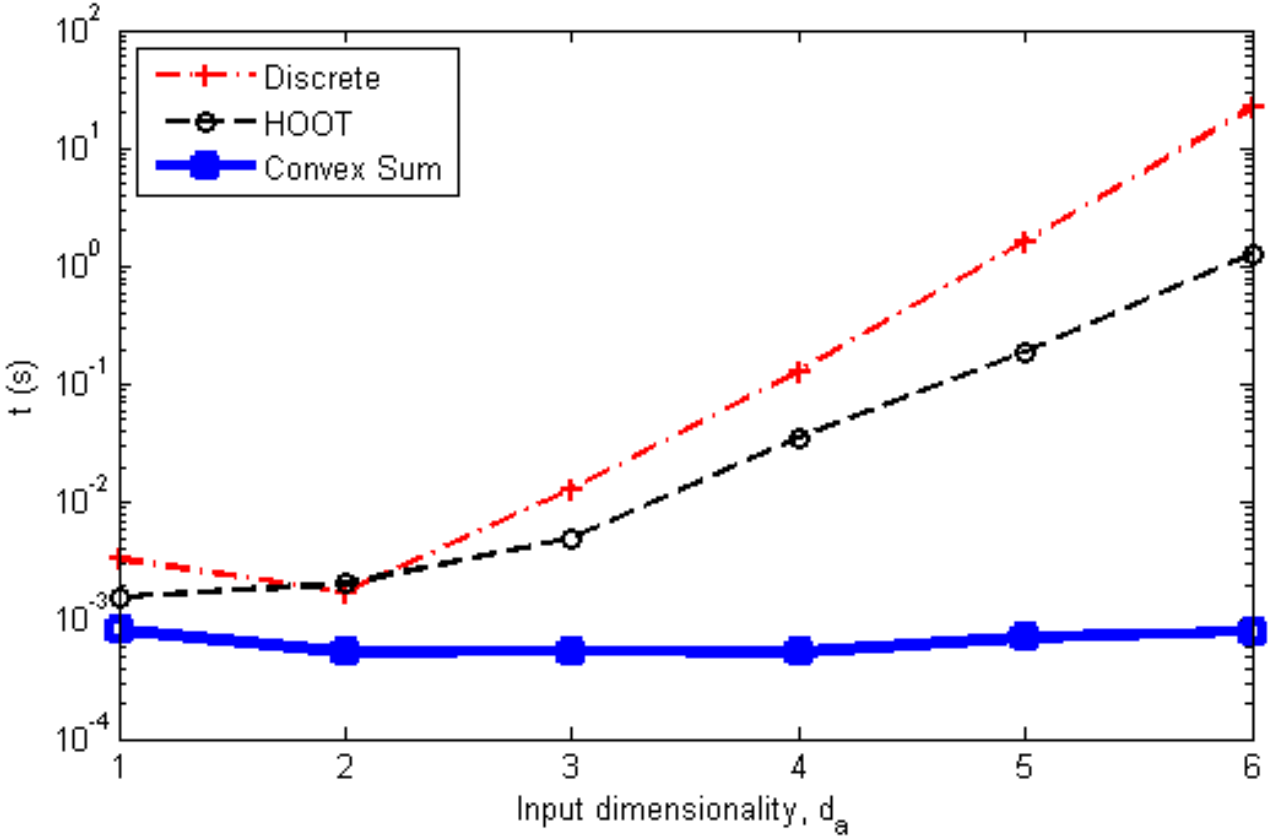
Trajectory without swing control

[Faust et al., 2013]

https://www.youtube.com/watch?v=I8z67xdoS1Q&list=PLNCPLrvktEnsVIR-Z7t5e8xJlrZjNMZxx

Computational Efficiency

Over 1300 times faster action selection in 3D spaces

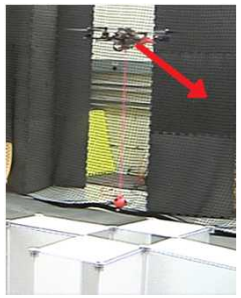


Time to make a decision per action dimensionality

[Faust et al., 2015]

Trajectory Characteristics under Varying Disturbance

Disturbance μ	Disturbance σ	Policy	Planning Time (ms) μ	Planning Time (ms) σ	Distance (cm) μ	Distance (cm) σ	Swing (°) μ	Swing (°) σ	Max. Swing (°) μ	Max. Swing (°) σ
1.00	0.00	LSAPA	1.76	0.04	0.09	0.00	0.01	0.00	15.62	0.00
		DAS	0.77	0.01	14.01	0.00	0.02	0.00	15.99	0.00
		NMPC	77.35	0.21	0.89	0.00	0.33	0.00	15.63	0.00
2.00	0.00	LSAPA	1.78	0.04	0.06	0.00	0.01	0.00	15.45	0.00
		DAS	0.77	0.01	28.11	0.00	0.01	0.00	16.18	0.00
		NMPC	77.57	96.97	4.73	5.91	1.74	2.18	6.19	7.74
0.00	0.50	LSAPA	1.68	0.07	0.61	0.24	0.14	0.05	15.82	0.07
		DAS	0.77	0.01	0.49	0.21	0.11	0.03	15.90	0.06
		NMPC	341.04	4.53	8.40	3.58	1.69	0.70	13.12	1.04
1.00	0.50	LSAPA	1.78	0.17	0.96	0.28	0.13	0.04	15.60	0.07
		DAS	0.77	0.02	14.01	0.37	0.12	0.04	16.00	0.07
		NMPC	318.39	4.91	96.54	31.16	3.64	1.28	14.35	1.71
2.00	0.50	LSAPA	1.65	0.05	0.65	0.25	0.13	0.05	15.42	0.11
		DAS	0.77	0.02	28.18	0.43	0.13	0.04	16.20	0.08
		NMPC	310.83	6.99	7147.63	1744.13	35.63	12.21	55.42	15.30
0.00	1.00	LSAPA	1.79	0.24	0.94	0.35	0.29	0.09	15.80	0.17
		DAS	0.81	0.06	0.79	0.28	0.23	0.07	15.88	0.15
		NMPC	321.59	6.61	14.67	5.74	2.66	1.00	12.79	1.80
1.00	1.00	LSAPA	1.65	0.08	3.14	0.83	0.31	0.11	15.58	0.16
		DAS	0.77	0.00	14.42	0.83	0.27	0.09	15.99	0.13
		NMPC	305.55	4.89	354.79	124.39	5.89	2.01	16.09	3.27
2.00	1.00	LSAPA	1.65	0.04	9.52	2.81	0.56	0.25	15.57	0.28
		DAS	0.76	0.00	30.24	1.34	0.34	0.14	16.16	0.15
		NMPC	309.31	6.03	13287.59	1513.72	42.71	24.56	71.47	39.23



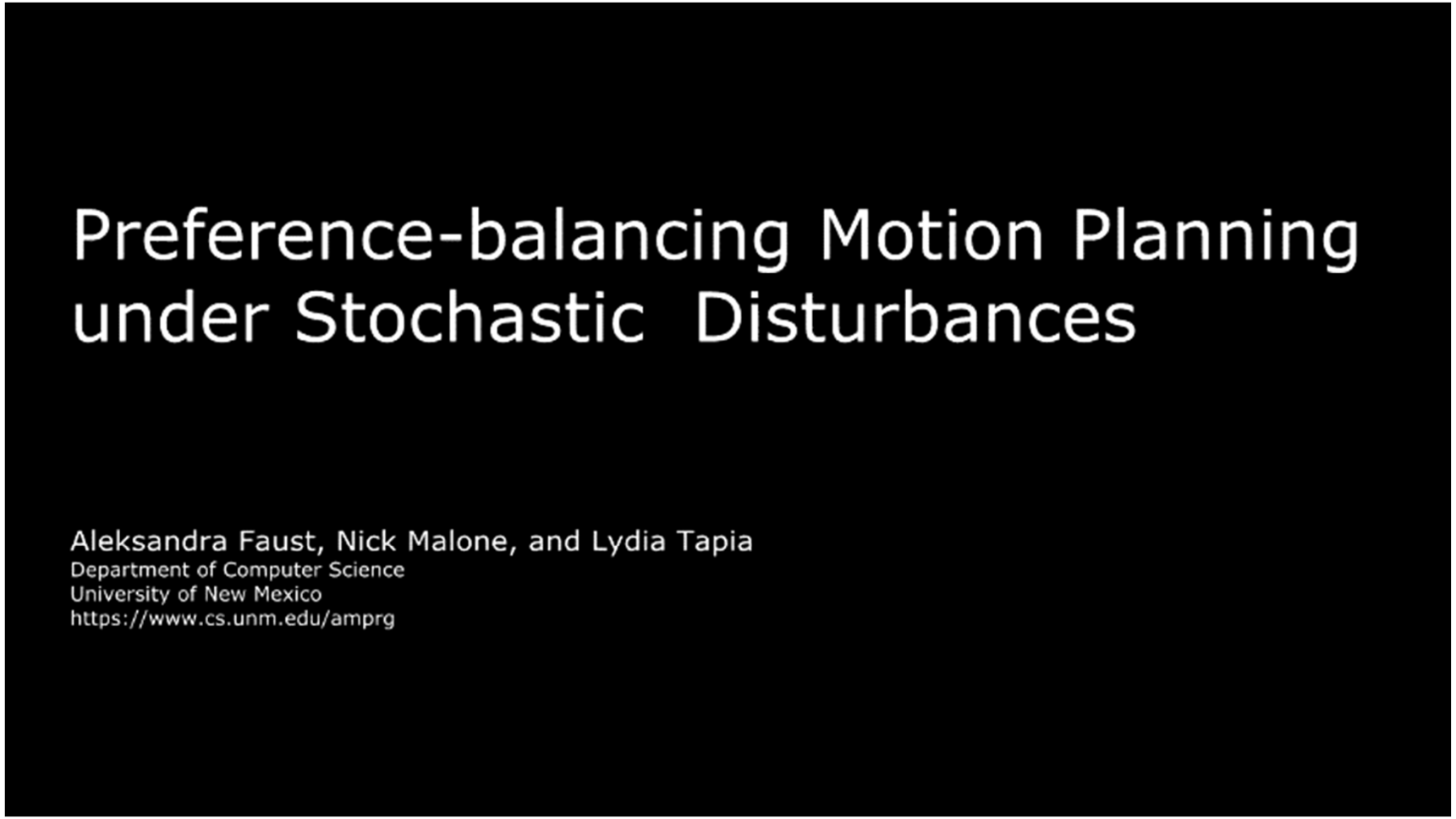
[Faust et al., under submission]

- Least Squares Axial Policy Approximation (LSAPA) [Faust et al., 2014]
- Deterministic Axial Sum (DAS) [Faust et al., 2015]
- Nonlinear Model Predictive Control (NMPC) [Grune and Pannek, 2011]

LSAPA and DAS perform decision-making in real-time.

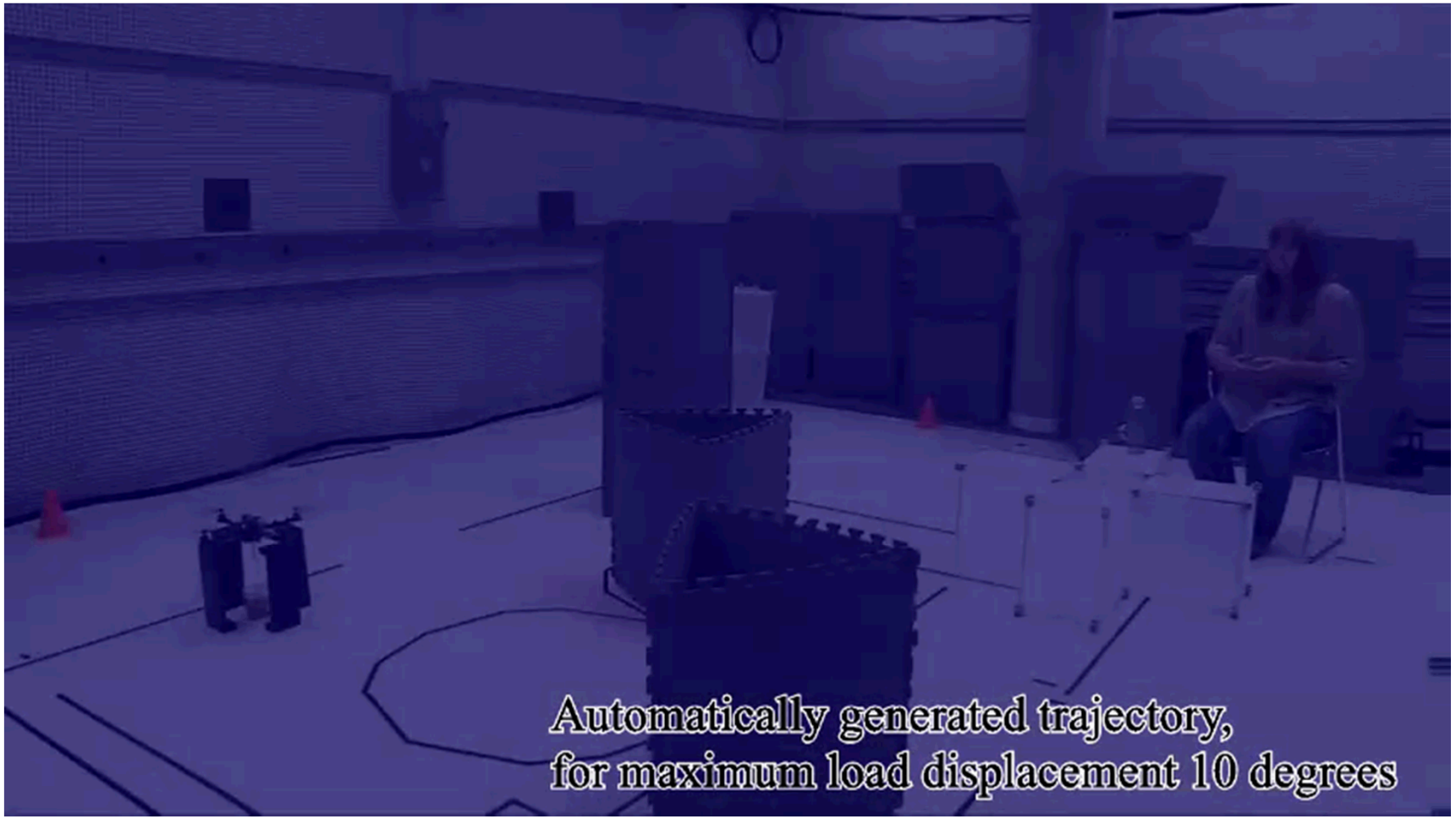
LSAPA reaches the goal for non-zero mean disturbances.

Swing-free aerial cargo delivery with disturbances



http://www.cs.unm.edu/~afaust/movies/afaustIcra15.mp4

Coffee Delivery

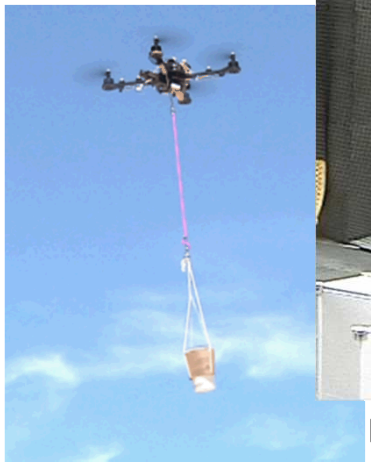


[Faust et al., under submission]

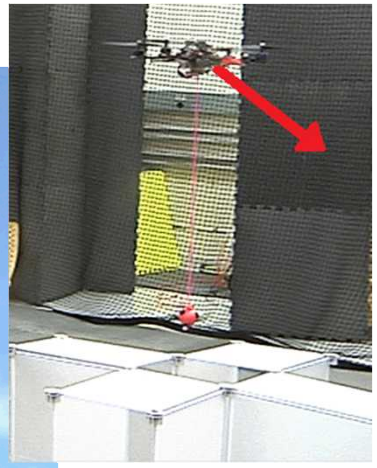
https://www.youtube.com/watch?v=s2pWxgAHw5E&index=5&list=PLNCPLrvktEnsVIR-Z7t5e8xJlrZjNMZxx

Conclusion

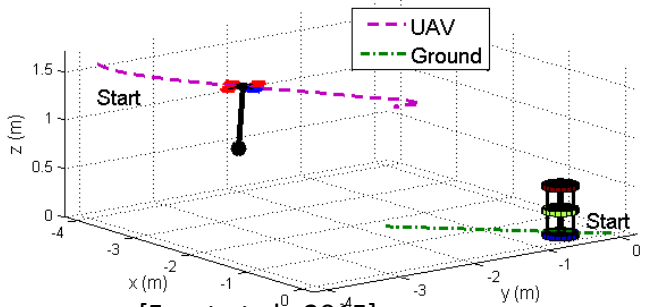
- PEARL
  - Solves robotic PBTs
  - For high-dimensional, acceleration-controlled robotic problems
  - Efficient
  - Safe
- Developed
  - Feature selection
  - Learning in continuous action spaces
  - Continuous action planning under external disturba
- Analysis
  - Task modification conditions
  - Convergence to goal conditions
- Evaluation
  - Additional applications
  - Outperforms previous methods in precision and planning speed



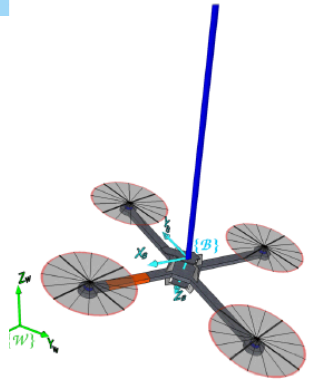
[Faust et al., 2013]



[Faust et al., under submission]



[Faust et al., 2015]



[Figueroa et al., 2014]

Questions

- Thank you!

- Website:
  - https://www.cs.unm.edu/amprg/Research/Quadrotor/

- Acknowledgements
  - Dr. Peter Ruymgaart for discussing disturbance modelling
  - Patricio Cruz for assisting with experiments
  - Reviewers for very helpful and constructive feedback

This work was in part supported by Sandia National Labs, NM Space Grant, and National Institutes of Health (NIH) Grant P20RR018754 to the Center for Evolutionary and Theoretical Immunology.