Autonomous Learning of Stable Quadruped Locomotion

Manish Saggar, Thomas D'Silva, Nate Kohl, **Peter Stone**

Department or Computer Sciences
The University of Texas at Austin

Policy Gradient RL for fast walk (Kohl '04)

Goal: Enable an Aibo to walk as fast as possible

Policy Gradient RL for fast walk (Kohl '04)

Goal: Enable an Aibo to walk as fast as possible

- Start with a **parameterized walk**
- **Learn** fastest possible parameters

Policy Gradient RL for fast walk (Kohl '04)

Goal: Enable an Aibo to walk as fast as possible

- Start with a **parameterized walk**
- **Learn** fastest possible parameters
- **No simulator** available:
  - Learn entirely on robots
  - Minimal human intervention

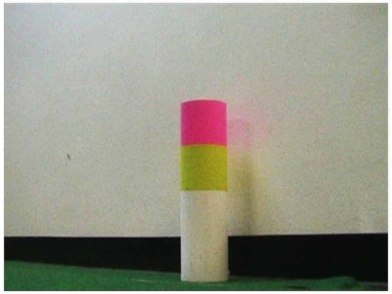
Policy Gradient RL for fast walk (Kohl '04)

Goal: Enable an Aibo to walk as fast as possible

- Start with a **parameterized walk**
- **Learn** fastest possible parameters
- **No simulator** available:
  - Learn entirely on robots
  - Minimal human intervention

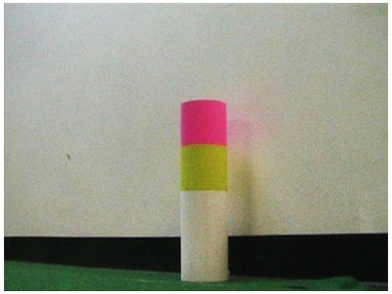
New Goal: fast walk with a stable camera

Usefulness of stability

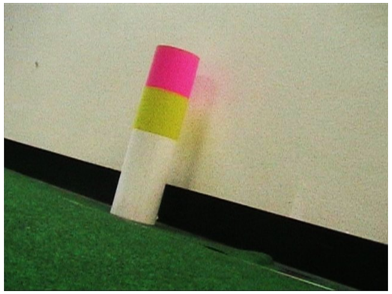


- Keep objects centered, **in the image**

Usefulness of stability

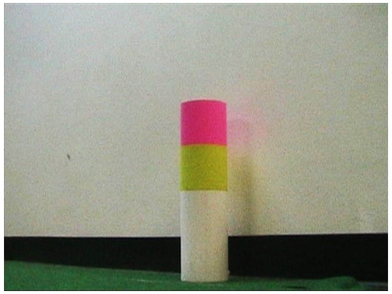


- Keep objects centered, **in the image**

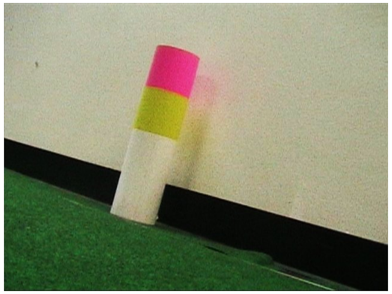


- Reduce rotations:

Usefulness of stability



- Keep objects centered, **in the image**



- Reduce rotations:
  - Eliminate need to transform image
  - Object detection: better speed/accuracy tradeoff

Two possible approaches

1. Learn a walk that **naturally keeps head stable**

Two possible approaches

1. Learn a walk that **naturally keeps head stable**
2. **Actively rotate** the neck to compensate for walk motion

Two possible approaches

1. Learn a walk that **naturally keeps head stable**

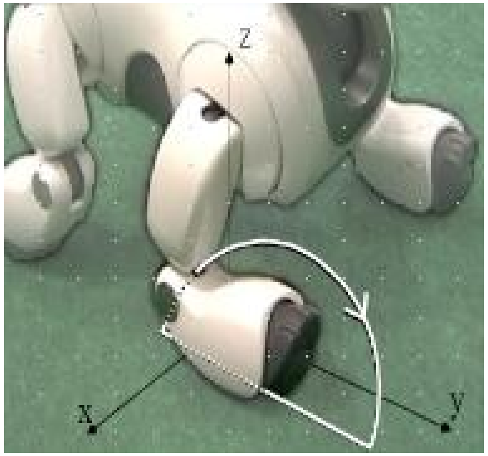
2. **Actively rotate** the neck to compensate for walk motion

May need to trade off against speed

A Parameterized Walk

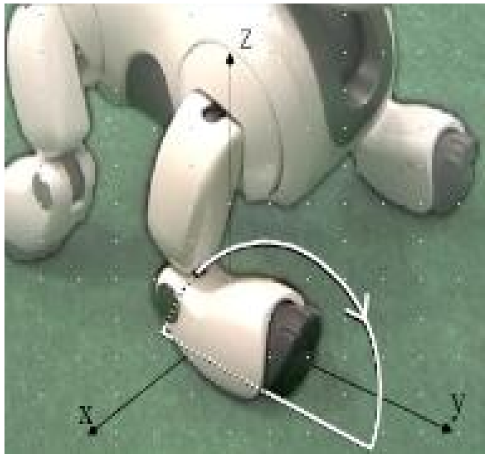
- Developed from scratch as part of **UT Austin Villa** 2003
- **Trot gait** with elliptical locus on each leg

Locus Parameters



- Ellipse length
- Ellipse height
- Position on x axis
- Position on y axis
- Body height
- Timing values

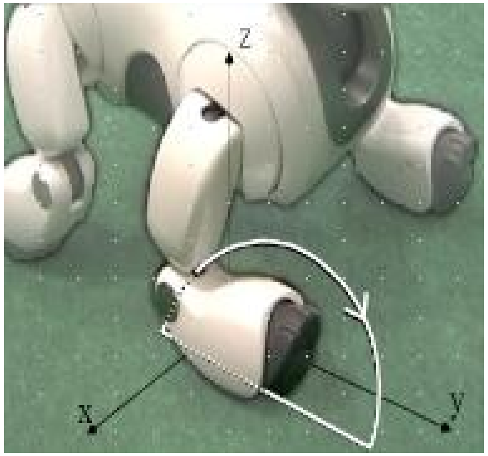
Locus Parameters



- Ellipse length
- Ellipse height
- Position on x axis
- Position on y axis
- Body height
- Timing values

- Head pan/tilt motion

Locus Parameters



- Ellipse length
- Ellipse height
- Position on x axis
- Position on y axis
- Body height
- Timing values

- Head pan/tilt motion

12 (16) continuous parameters

Full parameterization

- Front ellipse height, x-pos., y-pos. **(3)**
- Rear ellipse height, x-pos., y-pos. **(3)**
- Ellipse length **(1)**
- Ellipse skew multiplier in the x-y plane (for turning) **(1)**
- Front/rear body height **(2)**
- Time for each foot to complete ellipse **(1)**
- Fraction of time each foot spends on the ground **(1)**

Full parameterization

- Front ellipse height, x-pos., y-pos. **(3)**
- Rear ellipse height, x-pos., y-pos. **(3)**
- Ellipse length **(1)**
- Ellipse skew multiplier in the x-y plane (for turning) **(1)**
- Front/rear body height **(2)**
- Time for each foot to complete ellipse **(1)**
- Fraction of time each foot spends on the ground **(1)**
- Head pan limit and increment **(2)**
- Head tilt limit and increment **(2)**

Previous Experimental Setup

- Policy $\pi = \{\theta_1, \ldots, \theta_{12}\}$, $V(\pi)$ = walk **speed** when using π

Previous Experimental Setup

- Policy $\pi = \{\theta_1, \ldots, \theta_{12}\}$, $V(\pi)$ = walk **speed** when using π
- Training Scenario
  - Robots **time themselves** traversing fixed distance
  - Off-board computer **collects results**, **assigns policies**

Previous Experimental Setup

- Policy $\pi = \{\theta_1, \ldots, \theta_{12}\}$, $V(\pi)$ = walk **speed** when using π
- Training Scenario
  - Robots **time themselves** traversing fixed distance
  - Off-board computer **collects results**, **assigns policies**

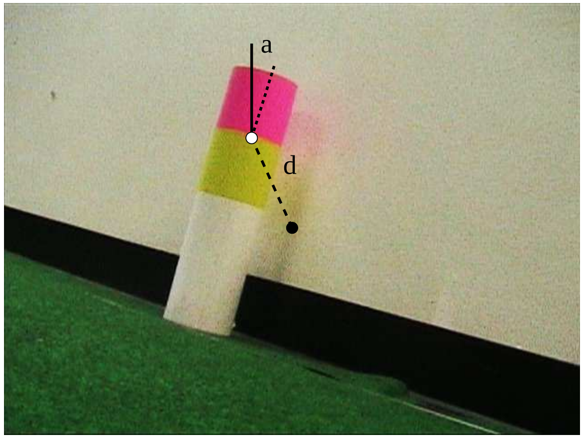
No human intervention except battery changes

Modified Objective Function

- M_t : time to walk fixed distance
- M_a : stddev. of 3 accelerometers

Modified Objective Function

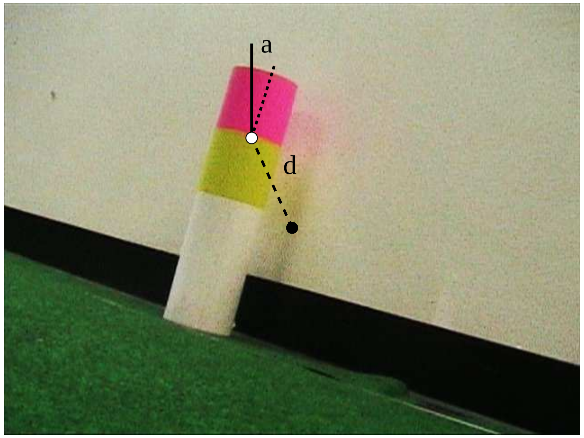
- M_t : time to walk fixed distance
- M_a : stddev. of 3 accelerometers
- M_d : dist. of landmark centroid to image center



- M_θ : landmark tilt angle

Modified Objective Function

- M_t : time to walk fixed distance
- M_a : stddev. of 3 accelerometers
- M_d : dist. of landmark centroid to image center



- M_θ : landmark tilt angle

$$V(\pi) = 1 - (w_t M_t + w_a M_a + w_d M_d + w_\theta M_\theta)$$

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$

Policy Gradient RL

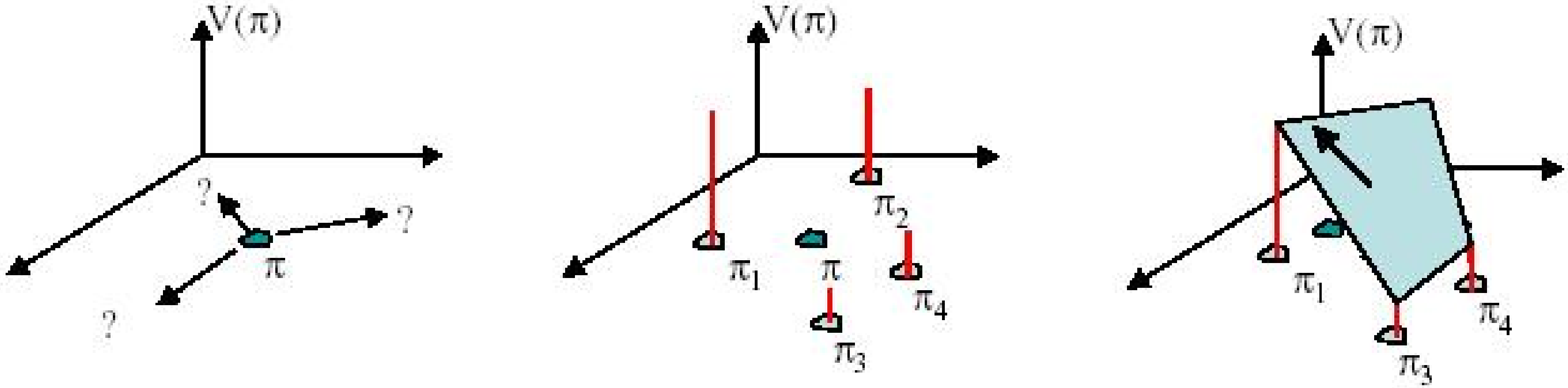
- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically
- Evaluate **neighboring policies** to estimate gradient
- Each trial randomly varies **every parameter**

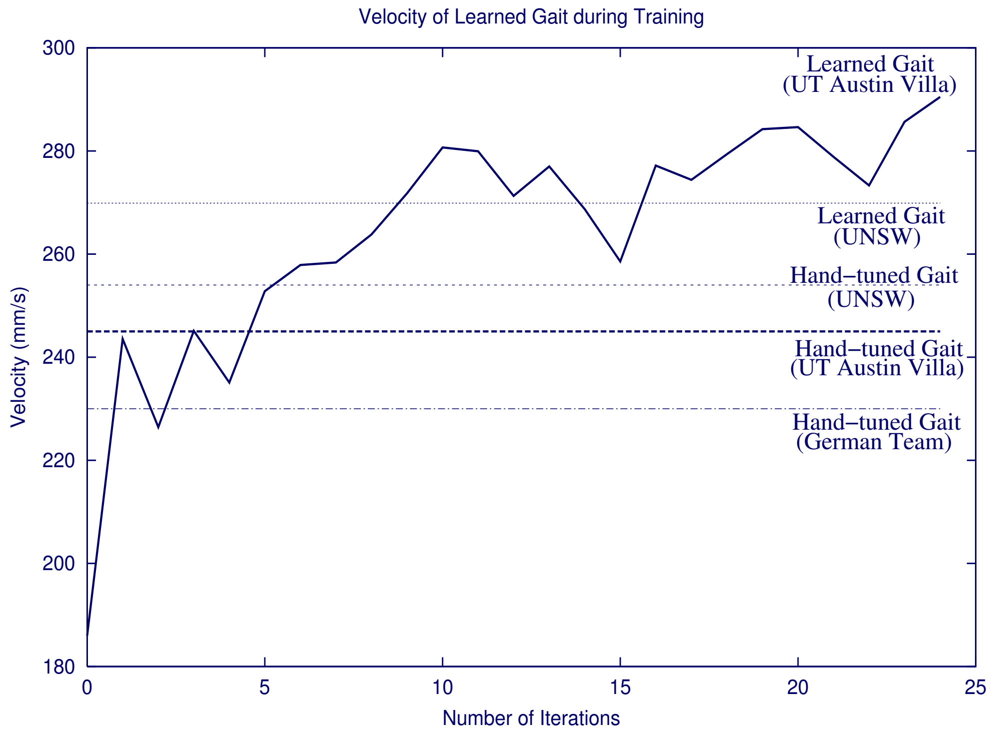
Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically
- Evaluate **neighboring policies** to estimate gradient
- Each trial randomly varies **every parameter**



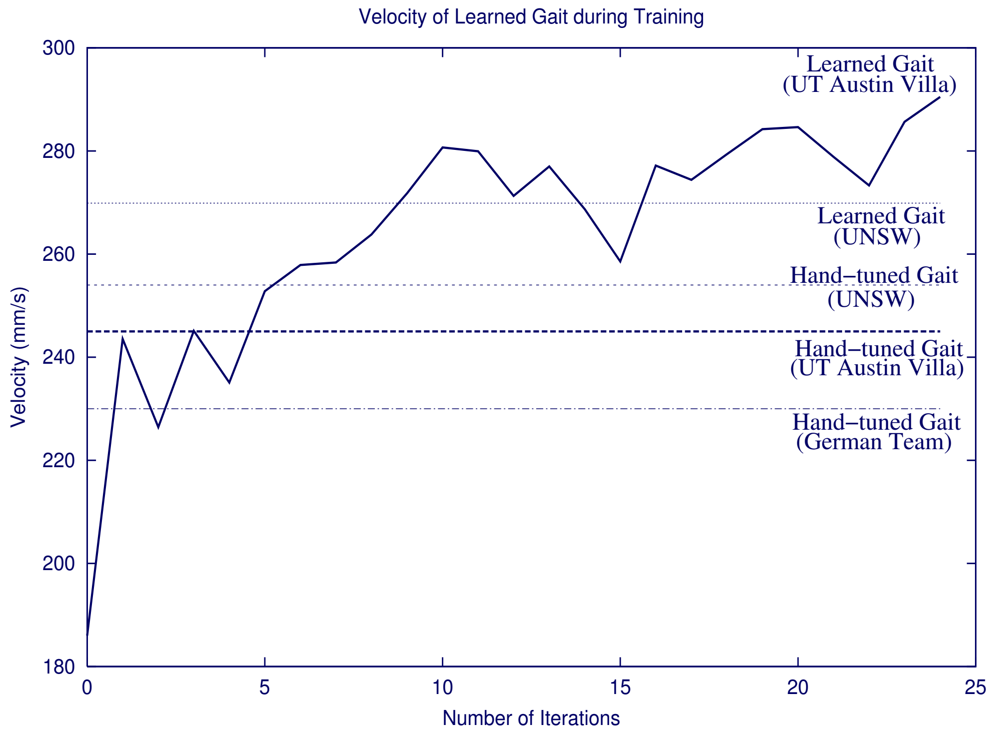
Fast Walk Results

$$V(\pi) = 1 - M_t$$



Fast Walk Results

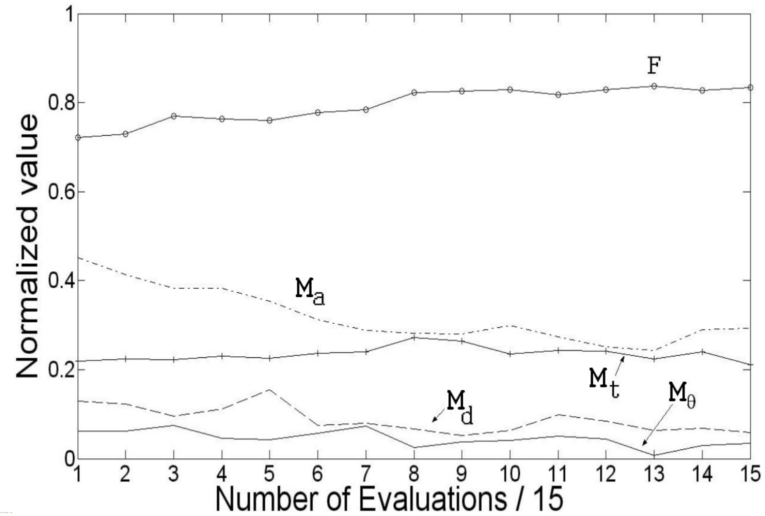
$$V(\pi) = 1 - M_t$$



$$V(\pi) = 1 - (w_t M_t + w_a M_a + w_d M_d + w_\theta M_\theta)$$

Approach 1: Learning a Stable Gait

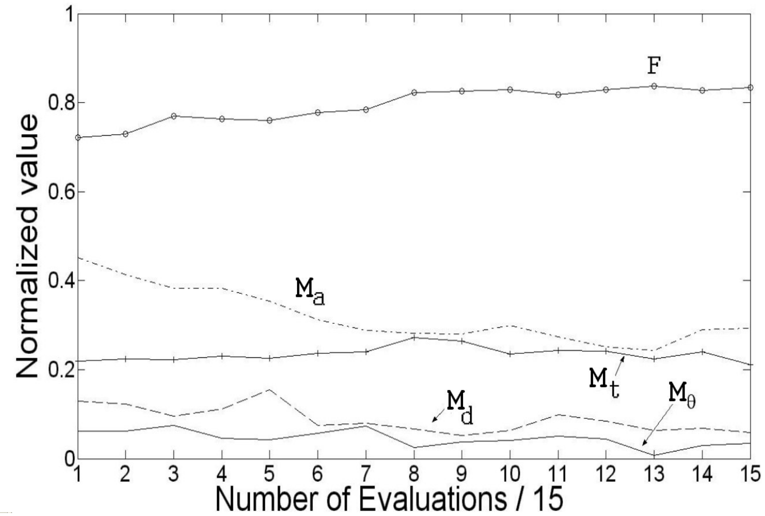
Favor speed



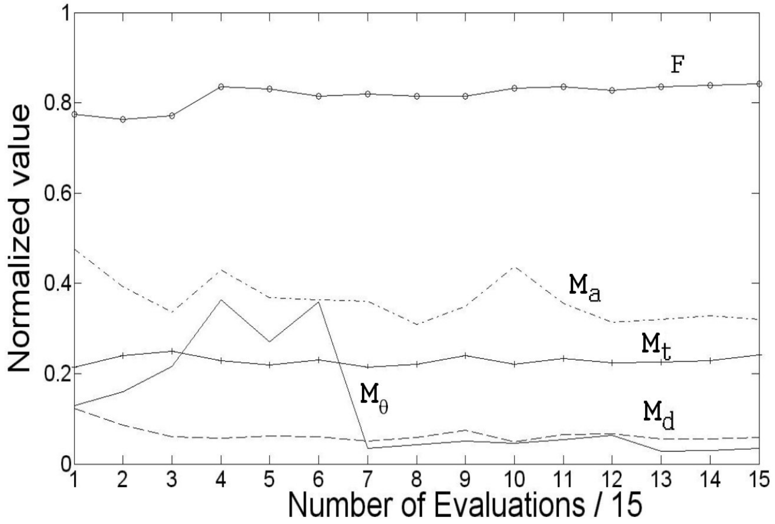
Equal weights

Approach 1: Learning a Stable Gait

Favor speed

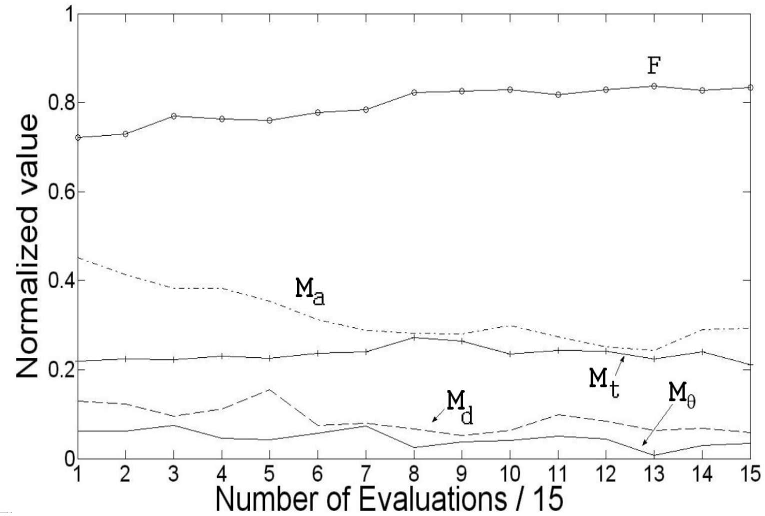


Equal weights

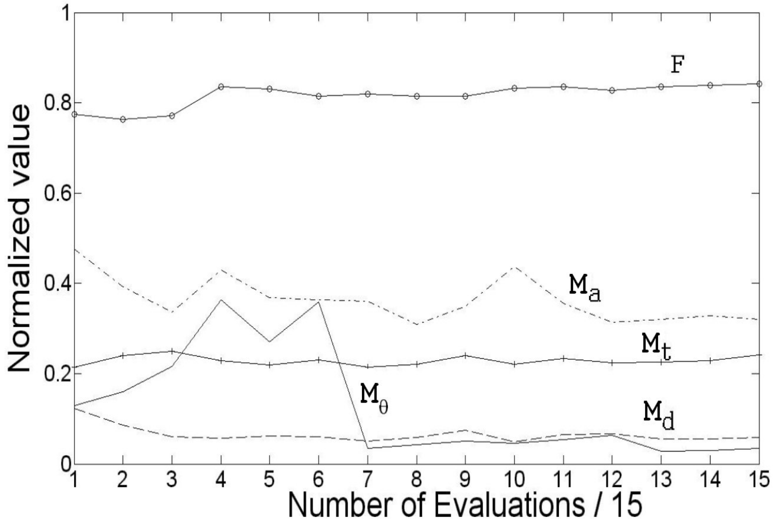


Approach 1: Learning a Stable Gait

Favor speed



Equal weights



Reduction Percentage

	Favor speed	Equal weights
M_t	-4.76%	-4.5%
M_a	34.7	32.6
M_d	60	57.14
M_θ	76.9	51.2

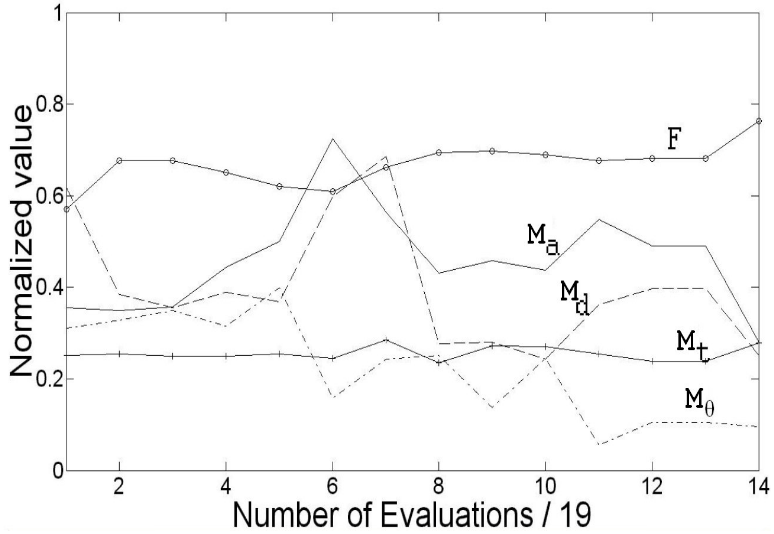
Segmentation Videos

Fast Walk

Stable Walk

Approach 2: Head Compensation

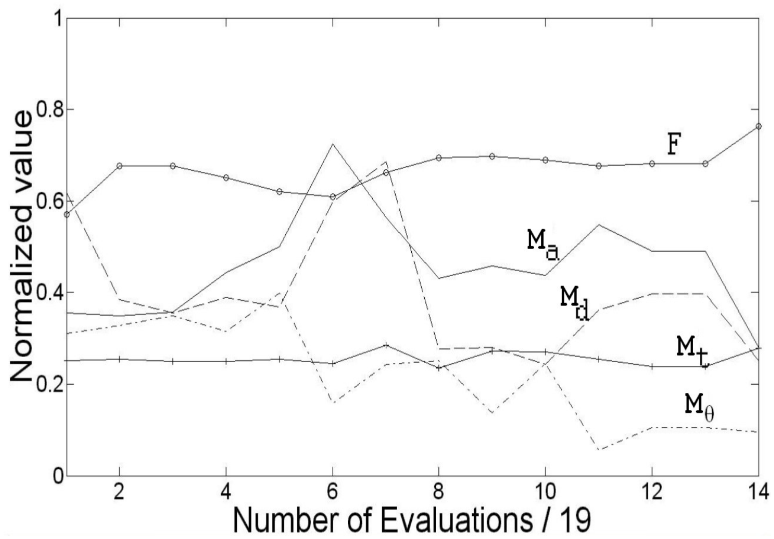
Favor speed



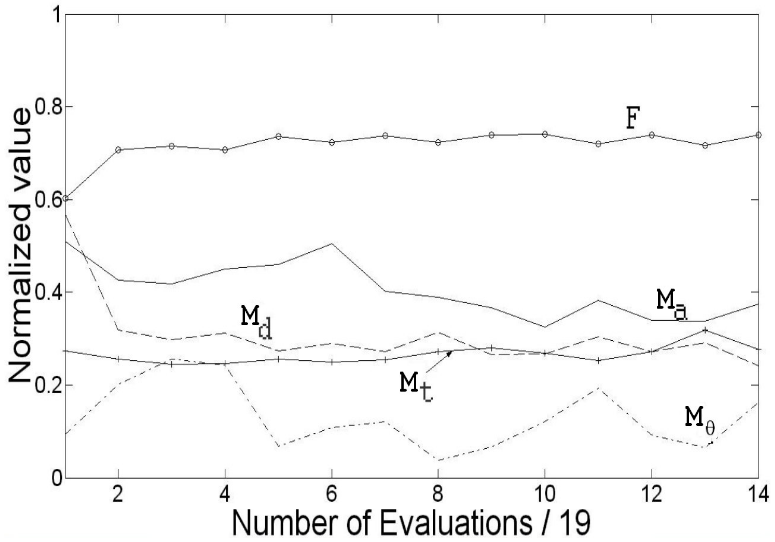
Equal weights

Approach 2: Head Compensation

Favor speed

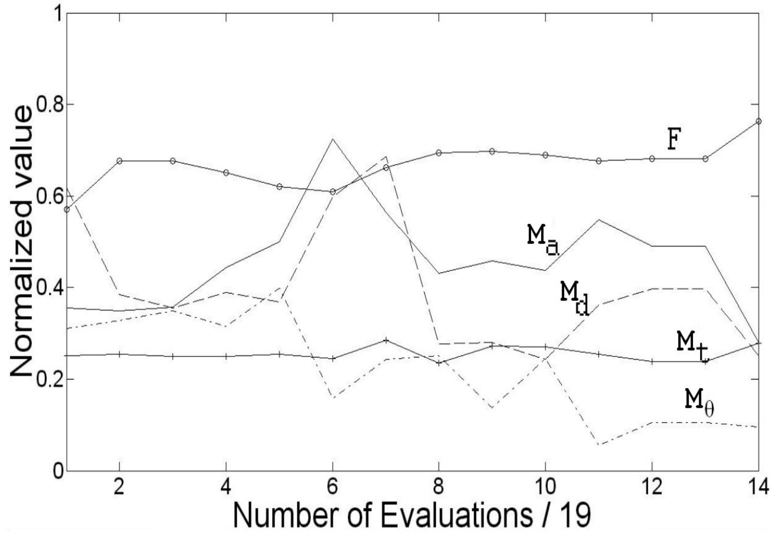


Equal weights

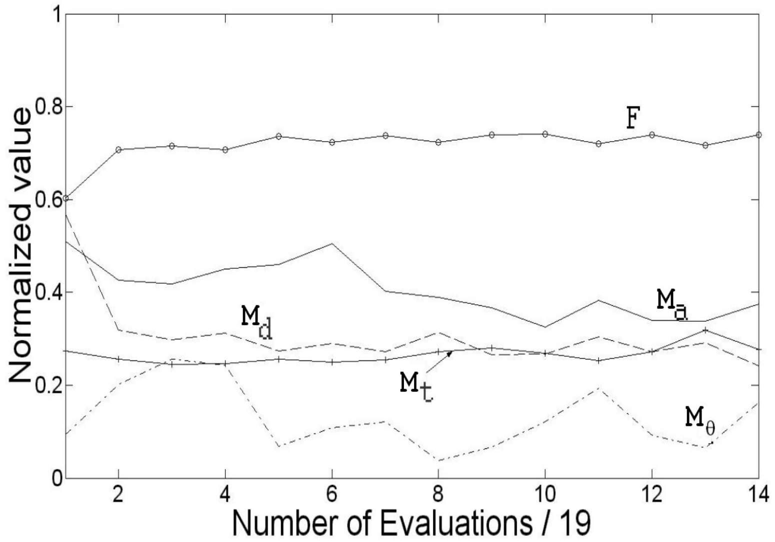


Approach 2: Head Compensation

Favor speed



Equal weights



Stability Improves Object Detection

- Traverse field while **identifying beacons**
- Results scored based on **ground truth**

Stability Improves Object Detection

- Traverse field while **identifying beacons**
- Results scored based on **ground truth**

	True Positives	False Positives
Fast Gait	0.33	0.052
Stable Gait	0.46	0.028

Stability Improves Object Detection

- Traverse field while **identifying beacons**
- Results scored based on **ground truth**

	True Positives	**False Positives**
Fast Gait	0.33	0.052
Stable Gait	0.46	0.028

- 39% more true positives; 54% fewer false positives
- **Statistically significant**

Summary

- Policy gradient learning of **stable Aibo walk**
- All learning done **on real robots**
- Stability **helps vision**

Summary

- Policy gradient learning of **stable Aibo walk**
- All learning done **on real robots**
- Stability **helps vision**

with Manish Saggar, Thomas D'Silva, Nate Kohl