Using Dynamic Rewards to Learn a Fully Holonomic Bipedal Walk

Patrick MacAlpine and Peter Stone

Department of Computer Science, The University of Texas at Austin

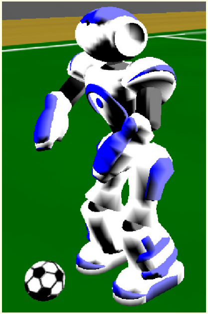
June 4, 2012

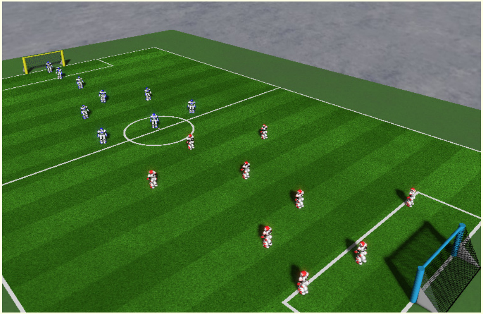
Definitions

- Bipedal locomotion: Walking upright on two legs
- Fully holonomic: Able to move in all directions with equal velocity

RoboCup 3D Simulation Domain

- Teams of 9 vs 9 autonomous agents play soccer
- Realistic physics using Open Dynamics Engine (ODE)
- Agents modeled after Aldebaron Nao robot
- Agent receives noisy visual information about environment
- Agents can communicate with each other over limited bandwidth channel





Competition Results

RoboCup	2010	2011
Goals For:	11	
Goals Against:	17	
Record (W-L-T):	4-5-1	
Place:	Outside Top-8	

Competition Results

RoboCup	2010	2011
Goals For:	11	136
Goals Against:	17	
Record (W-L-T):	4-5-1	
Place:	Outside Top-8	

Competition Results

RoboCup	2010	2011
Goals For:	**11**	**136**
Goals Against:	**17**	**0**
Record (W-L-T):	**4-5-1**	
Place:	**Outside Top-8**	

Competition Results

RoboCup	2010	2011
Goals For:	11	136
Goals Against:	17	0
Record (W-L-T):	4-5-1	24-0-0
Place:	Outside Top-8	

Competition Results

RoboCup	2010	2011
Goals For:	11	136
Goals Against:	17	0
Record (W-L-T):	4-5-1	24-0-0
Place:	Outside Top-8	1st

RoboCup	2010	2011
Goals For:	11	136
Goals Against:	17	0
Record (W-L-T):	4-5-1	24-0-0
Place:	Outside Top-8	1st

BIG IMPROVEMENT!

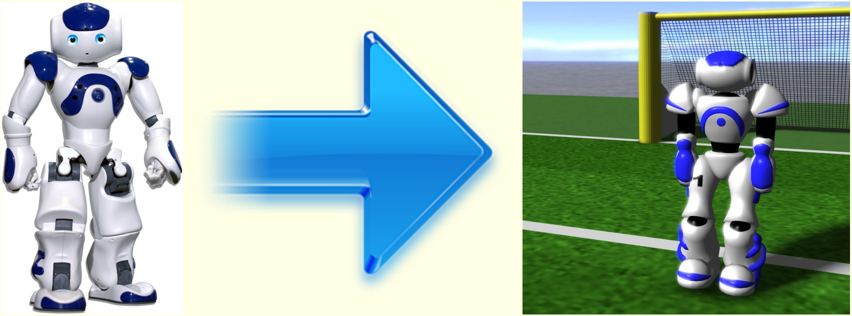
RoboCup	2010	2011
Goals For:	**11**	**136**
Goals Against:	**17**	**0**
Record (W-L-T):	**4-5-1**	**24-0-0**
Place:	**Outside Top-8**	**1st**

BIG IMPROVEMENT!

Optimized omnidirectional walk propelled team from 10th to 1st

Omnidirectional Walk Engine

- Double linear inverted pendulum model
- Based closely on that of walk engine by Graf et al
- Mostly open loop but not entirely
- Designed on actual Nao robot



Walk Engine Parameters

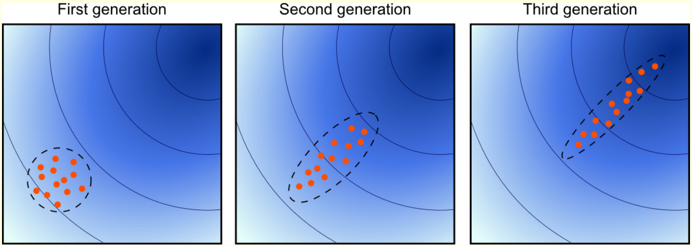
Notation	Description
$\mathbf{maxStep_i}$	Maximum step sizes allowed for x, y, and θ
$\mathbf{y_{shift}}$	Side to side shift amount with no side velocity
$\mathbf{z_{torso}}$	Height of the torso from the ground
$\mathbf{z_{step}}$	Maximum height of the foot from the ground
$\mathbf{f_g}$	Fraction of a phase that the swing foot spends on the ground before lifting
f_a	Fraction that the swing foot spends in the air
$\mathbf{f_s}$	Fraction before the swing foot starts moving
f_m	Fraction that the swing foot spends moving
$\boldsymbol{\phi_{length}}$	Duration of a single step
$\boldsymbol{\delta}$	Factors of how fast the step sizes change
y_{sep}	Separation between the feet
$\mathbf{x_{offset}}$	Constant offset between the torso and feet
$\mathbf{x_{factor}}$	Factor of the step size applied to the forwards position of the torso
$\mathbf{err_{norm}}$	Maximum COM error before the steps are slowed
$\mathbf{err_{max}}$	Maximum COM error before all velocity reach 0

Parameters of the walk engine with the optimized parameters shown in bold

Initial Walk Parameters

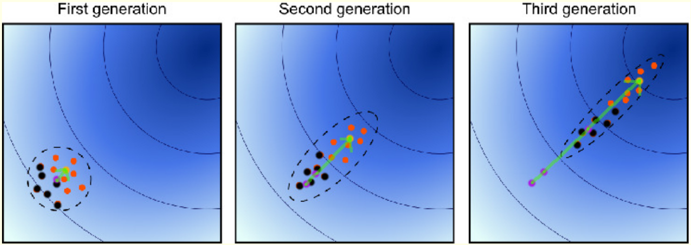
- Designed and hand-tuned to work on the actual Nao robot
- Provides a slow and stable walk

CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy)



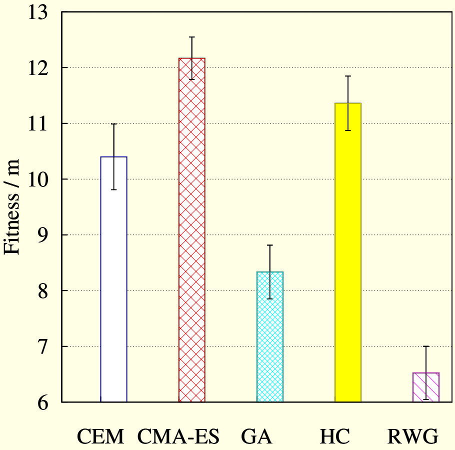
- Evolutionary numerical optimization method
- Candidates sampled from multidimensional Gaussian and evaluated for their fitness
- Weighted average of members with highest fitness used to update mean of distribution
- Covariance update using evolution paths controls search step sizes

CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy)



- Evolutionary numerical optimization method
- Candidates sampled from multidimensional Gaussian and evaluated for their fitness
- Weighted average of members with highest fitness used to update mean of distribution
- Covariance update using evolution paths controls search step sizes

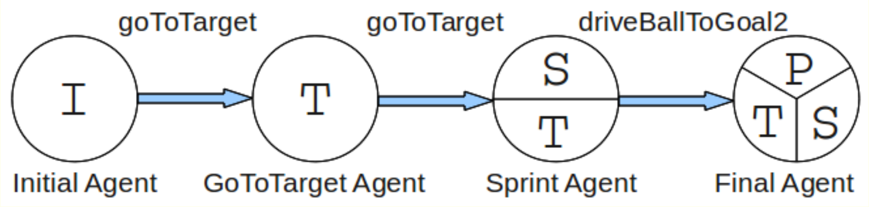
Learning Algorithms Evaluation



CEM	Cross Entropy Method
CMA-ES	Covariance Matrix Strategy Evolutionary Strategy
GA	Genetic Algorithm
HC	Hill Climbing
RWG	Random Weight Guessing

2011 Omnidirectional Walk Optimization

- Agent moves and turns in direction of target at the same time
- When dribbling agent circles while always facing ball
- Learn three different parameter sets for three different tasks
  - Going to a target
  - Sprinting forward
  - Positioning around the ball when dribbling
- Parameters learned through a layered learning approach



I = *initial*, T = *goToTarget*, S = *sprint*, P = *positioning*

Go to Target Optimization

- Agent navigates to a series of target positions on the field
- Also have stop targets where agent is told to stop
- Reward: + for distance traveled toward target,
  - for movement when told to stop

Fall = 5 if robot fell, 0 otherwise
d_{target} = distance traveled towards target
d_{moved} = total distance moved
t_{total} = duration a target is active
t_{taken} = time taken to reach target, or t_{total} if target not reached

$$\begin{aligned} reward_{target} &= d_{target}\frac{t_{total}}{t_{taken}} - Fall \\ reward_{stop} &= -d_{moved} - Fall \end{aligned}$$

Go to Target Optimization Video

Red 'T' = *gotoTarget* parameters, yellow 'S' = *sprint* parameters

Final Agent Video

Red 'T' = *gotoTarget* parameters, yellow 'S' = *sprint* parameters, cyan 'P' = *positioning* parameters

2011 Walk Weaknesses

- Still not all all that fast moving around the ball
- Turning takes time and causes a delay

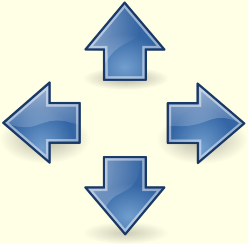
Fully Holonomic Walk

- Want to be able to walk in all directions with equal velocity
- No delays for needing to turn

Problems in Learning a Fully Holonomic Walk

- Kinematics of robot allow for faster walking forward speed
- Speed in one direction dominates speed in other directions
- Agent optimized without turning to target lost on average by .7 goals to agent that does turn

Fully Holonomic Walk Optimization



- Use GoToTarget optimization but agent does not turn toward target
- Only give positive rewards during longs walks in cardinal forward, backward, and sideways directions
- Still penalize for falls in all parts of the optimization
- Dynamically reweight directional rewards to encourage equal velocities in each direction

$$reward = reward_{fw} * weight_{fw} + reward_{bw} * weight_{bw} + reward_{sw} * weight_{sw}$$

Reweighting Rewards

Reweighting Rewards

Separate directional rewards from overall reward
(from top fitness member or weighted average of top half of population)

$$reward_i \quad \Rightarrow \quad reward_{i\{fw,bw,sw\}}$$

Reweighting Rewards

Separate directional rewards from overall reward
(from top fitness member or weighted average of top half of population)

$$reward_i \quad \Rightarrow \quad reward_{i\{fw,bw,sw\}}$$

Get maximum reward for any of the directions

$$reward_{i\{max\}} \quad = \quad max(reward_{i\{fw,bw,sw\}})$$

Reweighting Rewards

Separate directional rewards from overall reward
(from top fitness member or weighted average of top half of population)

$$reward_i \quad \Rightarrow \quad reward_{i\{fw,bw,sw\}}$$

Get maximum reward for any of the directions

$$reward_{i\{max\}} \quad = \quad max(reward_{i\{fw,bw,sw\}})$$

Compute weights (factors) to multiply each directional reward by to equal maximum reward

$$weight_{i+1}\{fw/bw/sw\} \quad = \quad reward_{i\{max\}}/reward_{i\{fw/bw/sw\}}$$

Reweighting Rewards

Separate directional rewards from overall reward
(from top fitness member or weighted average of top half of population)

$$reward_i \quad \Rightarrow \quad reward_{i\{fw,bw,sw\}}$$

Get maximum reward for any of the directions

$$reward_{i\{max\}} \quad = \quad max(reward_{i\{fw,bw,sw\}})$$

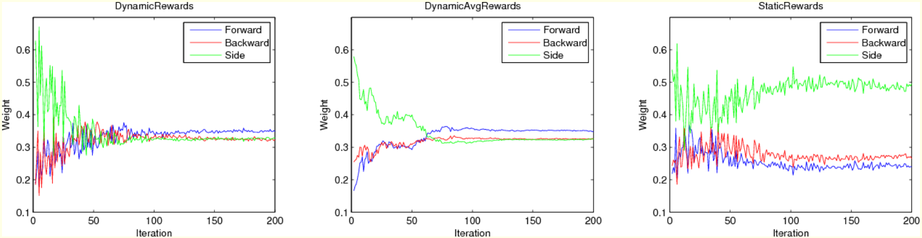
Compute weights (factors) to multiply each directional reward by to equal maximum reward

$$weight_{i+1}\{fw/bw/sw\} \quad = \quad reward_{i\{max\}}/reward_{i\{fw/bw/sw\}}$$

Normalize all weights to sum to 1

$$weight_{i+1\{fw/bw/sw\}} \quad = \quad weight_{i+1\{fw/bw/sw\}}/sum(weight_{i+1\{fw,bw,sw\}})$$

Weights Over Iterations of CMA-ES



- Both dynamic reward agent's weights converge to almost the same value
- Static reward agent's weights (not applied to reward) diverge as forward speed dominates

Directional Speeds

Agent	Forward	Backward	Sideways
DynamicRewards	.42	.53	.48
DynamicAvgRewards	.45	.53	.51
StaticRewards	.58	.52	.37
FaceForward	.74	.35	.03
2011 Walk	.71	.40	.21

- Both dynamic reward agents have close to equal speeds in all directions
- Static reward agent has slower side walking speed
- Face forward agent very biased toward forward walking speed with almost 0 speed for sideways direction

Fully Holonomic Walk

- Can walk in all directions with nearly equal velocity

Fully Holonomic Walk

Circling the Ball

Reaction Time

Game Performance

	2011 Walk	FaceForward	StaticRewards	DynAvgRewards
DynRewards	0.20(.08)	3.27(.09)	3.18(.11)	-0.06(.07)
DynAvgRewards	0.10(.07)	3.49(.11)	2.88(.11)	
StaticRewards	-2.77(.13)	0.22(.06)		
FaceForward	-2.99(.12)			

Game Performance

	2011 Walk	FaceForward	StaticRewards	DynAvgRewards
DynRewards	0.20(.08)	3.27(.09)	3.18(.11)	-0.06(.07)
DynAvgRewards	0.10(.07)	3.49(.11)	2.88(.11)	
StaticRewards	-2.77(.13)	0.22(.06)		
FaceForward	-2.99(.12)			

DynRewards vs 2011 Walk Record: 23-7-70 (29 goals for, 9 against)

Summary

Summary

- Dynamically updating reward weights is an effective means for learning a fully holonomic walk

Summary

- Dynamically updating reward weights is an effective means for learning a fully holonomic walk
- Rebalancing reward weights helps to prevent domination of one component of a reward signal over other components

Summary

- Dynamically updating reward weights is an effective means for learning a fully holonomic walk
- Rebalancing reward weights helps to prevent domination of one component of a reward signal over other components
- In the 3D simulation league quickness is more important than speed

Related Work

- N. Hansen. The CMA Evolution Strategy: A Tutorial, January 2009.
- C. Graf, A. Härtl, T. Röefer, and T. Laue. A robust closed-loop gait for the standard platform league humanoid.
- N. Shafii, L. P. Reis, and N. Lao. Biped walking using coronal and sagittal movements based on truncated Fourier series, January 2010.
- J. E. Pratt. Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2000.
- N. Kohl and P. Stone. Machine learning for fast quadrupedal locomotion, 2004.
- D. Urieli, P. MacAlpine, S. Kalyanakrishnan, Y. Bentor, and P. Stone. On optimizing interdependent skills: A case study in simulated 3d humanoid robot soccer, 2011.
- P. MacAlpine, D. Urieli, S. Barrett, S. Kalyanakrishnan, F. Barrera, A. Lopez-Mobilia, N. Stiurca, V. Vu, and P. Stone. UT Austin Villa 2011: A Winning Approach to the RoboCup 3D Soccer Simulation Competition, 2012.
- P. MacAlpine, S. Barrett, D. Urieli, V. Vu, and P. Stone. Design and Optimization of an Omnidirectional Humanoid Walk: A Winning Approach at the RoboCup 2011 3D Simulation Competition, 2012.

Related Work

- N. Hansen. The CMA Evolution Strategy: A Tutorial, January 2009.
- C. Graf, A. Härtl, T. Röefer, and T. Laue. A robust closed-loop gait for the standard platform league humanoid.
- N. Shafii, L. P. Reis, and N. Lao. Biped walking using coronal and sagittal movements based on truncated Fourier series, January 2010.
- J. E. Pratt. Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2000.
- N. Kohl and P. Stone. Machine learning for fast quadrupedal locomotion, 2004.
- D. Urieli, P. MacAlpine, S. Kalyanakrishnan, Y. Bentor, and P. Stone. On optimizing interdependent skills: A case study in simulated 3d humanoid robot soccer, 2011.
- P. MacAlpine, D. Urieli, S. Barrett, S. Kalyanakrishnan, F. Barrera, A. Lopez-Mobilia, N. Stiurca, V. Vu, and P. Stone. UT Austin Villa 2011: A Winning Approach to the RoboCup 3D Soccer Simulation Competition, 2012.
- P. MacAlpine, S. Barrett, D. Urieli, V. Vu, and P. Stone. Design and Optimization of an Omnidirectional Humanoid Walk: A Winning Approach at the RoboCup 2011 3D Simulation Competition, 2012.

Future Work

- Attempt to apply learned walks in simulation to actual Nao robots
- Extend holonomic walk to use multiple parameter sets (one for each of the cardinal directions)
- Model walk trajectories after those taken by human infants learning to walk

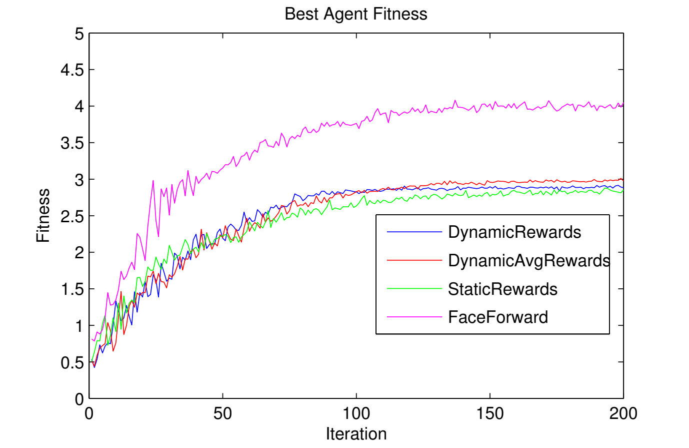
More Information

UT Austin Villa 3D Simulation Team homepage:
www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/

Email: patmac@cs.utexas.edu

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030).

Fitness Over Iterations of CMA-ES



- All non-turning holonomic agents have similar fitness
- Face forward turning agent (similar to 2011 walk agent) has highest fitness

Average Weighted Rewards Calculation

$$
\begin{aligned}
weight_i &= log(popsize/2 + 1/2) - log(i) \\
weights_{sum} &= \sum_{i=1}^{popsize/2} weight_i \\
weight_i &= weight_i / weights_{sum} \\
rew_{avg\{fw/bw/sw\}} &= \sum_{i=1}^{popsize/2} rew_{i\{fw/bw/sw\}} * weight_i
\end{aligned}
$$