Autonomous Learning of Ball Passing by Four-legged Robots and Trial Reduction by Thinning-out and Surrogate Functions

Hayato Kobayashi$^1$, Kohei Hatano$^2$, Akira Ishino$^1$, and Ayumi Shinohara$^1$

$^1$Tohoku University, Japan
$^2$Kyushu University, Japan
Contents

- Background
  - Autonomous learning of ball passing skills
  - Hybrid method for trial reduction
- Experimental results
  - Minimization of test functions
  - Learning of ball passing skills
- Conclusions
For robots to function in the real world, learning abilities are essential

- To adapt to unknown environments
- Legged robots must learn many basic skills
  - E.g., walking, running, pushing, pulling, jumping, catching, kicking, hitting, ...

Learning of ball passing skills by AIBO
RoboCup soccer

Competition for autonomous robots that play soccer

- Small size league
- Middle size league
- Standard platform league (four-legged robot league)
- Simulation league
- Humanoid league

https://www.robocup.org/
Experimental costs using real robots

- Trial
  - Human intervention
  - Time consuming
  - Motor failure

Ex. Learning process of goal saving skills

Initial phase

Later phase

https://youtu.be/9oHA-GH9JT8

https://youtu.be/3Pluuk20xqs
Our result: reduction of the experimental costs

- Autonomous learning method of passing skills
  - For reducing human intervention
  - Application of the idea of autonomous learning of ball trapping skills [Kobayashi et al. 2006]

- Hybrid method for trial reduction
  - For reducing all costs of each trial
  - Improvement of thinning-out [Kobayashi et al. 2007] utilizing surrogate functions
Contents

- Background
- Autonomous learning of ball passing skills
- Hybrid method for trial reduction
- Experimental results
  - Minimization of test functions
  - Learning of ball passing skills
- Conclusions
Accurate shooting motions that move and stop a ball to a specific area
- Neither too strong nor too weak

Shooting motions
- Generated by key-frames (seq. of joint angles)

Ex. Forward shooting motion pushing a ball with its chest
Autonomous learning method of ball passing skills

Robots can acquire passing skills on their own

Related work
- Learning of walking skills [Kim and Uther 2003][Kohl and Stone 2004]
  [Hornby et al. 2005][Saggar et al. 2007]
- Learning of ball acquiring skills [Fidelman and Stone 2004][Fidelman and Stone 2007]
- Learning of ball trapping skills [Kobayashi et al. 2007]
Maximization of the following score function

Score function $f: X \rightarrow \mathbb{R}$ on $X \subseteq \mathbb{R}^{8K}$  
(K=#key-frames)

- Generate a motion from $x \in X$
- Make the robot kick the ball using the motion
- Return the distance to the kicked ball
  - Using the median of 5 evaluations

Each key-frame is indicated by 8 joint angles
(= head 2 + fore leg 3 + rear leg 3) using symmetry
Meta-heuristics

- Heuristic algorithms that are independent of problems
  - Genetic Algorithm
  - Simulated Annealing
  - Policy Gradient
  - Hill Climbing
  - ...

- We choose Genetic Algorithm (GA)
**Idea**: Make the resampling process of new candidates more efficient using meta-heuristics instead of random perturbation.

**Thinning-out** [Kobayashi et al. 2007]
To skip over the evaluation of unpromising candidates selected by meta-heuristics.

Our hybrid method combining **thinning-out** and **surrogate functions**

<table>
<thead>
<tr>
<th>Candidate $x \in X$</th>
<th>Random perturbation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thinning-out cond.</td>
<td>YES (unpromising)</td>
</tr>
<tr>
<td>NO (promising)</td>
<td></td>
</tr>
<tr>
<td>Score func. $f(x)$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate $x \in X$</th>
<th>Thinning-out cond.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO (promising)</td>
<td>YES (unpromising)</td>
</tr>
<tr>
<td>Score func. $f(x)$</td>
<td>Surrogate func. $a(x)$ approx. of $f(x)$</td>
</tr>
</tbody>
</table>
Thinning-out
[Kobayashi et al. 2007]

- To reduce unpromising trials
  - The same concept as “pruning” in search trees
- Based on the assumption
  - The score function is $g$-Lipschitz continuous

- Memory-based learning
  - Memory-based fitness evaluation GA [Sano et al. 2000]
  - Locally weighted regression [Schaal and Atkeson 1994]
  - Acceleration by function approximation [Ratle 1998]

We can easily combine the other methods with thinning-out
Lipschitz condition

\[ \exists g : \mathbb{R} \rightarrow \mathbb{R} \quad \forall x_1, x_2 \in X \quad |f(x_1) - f(x_2)| \leq g(d(x_1, x_2)) \]

\( f \) is said to be \( g \)-Lipschitz continuous
\( g \) is said to be a Lipschitz function

\( X \): Search space
\( f \): Score function
\( d \): Metric of \( X \)
The Lipschitz condition is given by:

\[ f(x_1) - g(d(x_1, x_2)) \leq f(x_2) \leq f(x_1) + g(d(x_1, x_2)) \]

where:
- \( f(x) \) is the score function
- \( g \) is a Lipschitz function
- \( d \) is the metric of \( X \)
- \( X \) is the search space

A function \( f \) is said to be \( g \)-Lipschitz continuous if there exists a real-valued function \( g \) such that:

\[ |f(x_1) - f(x_2)| \leq g(d(x_1, x_2)) \]

for all \( x_1, x_2 \in X \).
\[ f(x_1) - g(d(x_1, x_2)) \leq f(x_2) \leq f(x_1) + g(d(x_1, x_2)) \]

**Lipschitz condition**

\[ \exists g : \mathbb{R} \to \mathbb{R} \quad \forall x_1, x_2 \in X \quad |f(x_1) - f(x_2)| \leq g(d(x_1, x_2)) \]

\( f \) is said to be **\( g \)-Lipschitz continuous**

\( g \) is said to be a **Lipschitz function**

**Possible range of score**

\( X \): Search space

\( f \): Score function

\( d \): Metric of \( X \)
\( f(x_1) - g(d(x_1, x_2)) \leq f(x_2) \leq f(x_1) + g(d(x_1, x_2)) \)

Lipschitz condition

\[ \exists g : \mathbb{R} \rightarrow \mathbb{R} \ \forall x_1, x_2 \in X \ |f(x_1) - f(x_2)| \leq g(d(x_1, x_2)) \]

\( f \) is said to be \( g \)-Lipschitz continuous
\( g \) is said to be a Lipschitz function

**X**: Search space

**f**: Score function

**d**: Metric of \( X \)
Thinning-out condition

\[ f(x_n) + g(d(x_c, x_n)) \leq f(x_b) \]

The upperbound of the score range of \( x_c \)

\[ \Rightarrow f(x_c) \leq f(x_b) \]

\( f(x) \)

\( X \): Search space

\( f \): Score function,

\( g \)-Lipschitz continuous

\( d \): Metric of \( X \)

Best score \( f(x_b) \)

\( x_b \) (Best point so far)
Thinning-out condition

\[ f(x_n) + g(d(x_c, x_n)) \leq f(x_b) \]

The upperbound of the score range of \( x_c \)

\[ \Rightarrow f(x_c) \leq f(x_b) \]

\( f(x) \)

\( x \)

\( f(x_b) \) (Best point so far)

\( x_c \) (Candidate)

\( x_n \) (Nearest neighbor)

\( X: \) Search space

\( f: \) Score function, 

\( g: \) Lipschitz continuous

\( d: \) Metric of \( X \)
Thinning-out condition

\[ f(x_n) + g(d(x_c, x_n)) \leq f(x_b) \]

The upperbound of the score range of \( x_c \)

\[ \Rightarrow f(x_c) \leq f(x_b) \]

X: Search space
f: Score function,
g-Lipschitz continuous
d: Metric of X

\[ f(x_n) + g(d(x_c, x_n)) \]

\( x_n \) (Nearest neighbor)

\( x_b \) (Best point so far)

\( x_c \) (Candidate)
Thinning-out condition

\[ f(x_n) + g(d(x_c, x_n)) \leq f(x_b) \]

The upper bound of the score range of \( x_c \)

\[ \Rightarrow f(x_c) \leq f(x_b) \]

\[ f(x) \]

\( f(x_b) \) (Best point so far)

\( x_n \) (Nearest neighbor)

\( f(x_n) + g(d(x_c, x_n)) \)

Score range

\( x_c \) (Candidate)

\( X \): Search space

\( f \): Score function,

\( g \): Lipschitz continuous

\( d \): Metric of \( X \)
Thininning-out condition

\[ f(x_n) + g(d(x_c, x_n)) \leq f(x_b) \]

The upperbound of the score range of \( x_c \)

\[ \Rightarrow f(x_c) \leq f(x_b) \]

\[ X: \text{Search space} \]
\[ f: \text{Score function, } \]
\[ g: \text{Lipschitz continuous} \]
\[ d: \text{Metric of } X \]

\( x_n \) (Nearest neighbor)

\( x_b \) (Best point so far)

\( x_c \) (Candidate)
The thinning-out condition is given by:

\[ f(x_n) + g(d(x_c, x_n)) \leq f(x_b) \]

which implies \( f(x_c) \leq f(x_b) \).

- **X**: Search space
- **f**: Score function
- **g**: Lipschitz continuous
- **d**: Metric of \( X \)

The upper bound of the score range of \( x_c \) is given by:

\[ f(x_n) + g(d(x_c, x_n)) \]

And the best score is given by:

\[ f(x_b) \]

The point \( x_b \) is the best point so far, and \( x_n \) is the nearest neighbor to \( x_c \).
Thinning-out condition

\[ f(x_n) + g(d(x_c, x_n)) \leq f(x_b) \]

The upperbound of the score range of \( x_c \)

\[ \Rightarrow f(x_c) \leq f(x_b) \]

X: Search space
f: Score function, g-Lipschitz continuous
d: Metric of X

Score range

\[ f(x_n) + g(d(x_c, x_n)) \]

Evaluated

Best score \( f(x_b) \)

\( x_b \) (Best point so far)

\( x_n \) (Nearest neighbor)

\( x_c \) (Candidate)
Inferring methods of Lipschitz functions

- Max Gradient method (MG)
  - Using the max gradient in the history
  - Naïve method
  - Thin-out correctly

- Gathering Differences method (GD)
  - Using the weighted average of gradients in the history
  - Heuristic method
  - Thin-out a lot
  - Useful in high dimension
Function interpolation method [Matheron 1963]
  Initially developed in geostatistics
  Recently used as surrogate functions

Ordinary kriging
  Most common type of kriging
  Related studies used as surrogate functions
    [Martin and Simpson 2003]
    [Jouhaud et al. 2007]
    [Glaz et al. 2008]
Interpolated value of \( x^* \) is represented by

\[
\hat{f}(x^*) = \sum_{i=1}^{n} w_i f(x_i)
\]

where \( f(x_i) \) is the observed score of \( x_i \in X \)

\( w_i \) is the weight of \( f(x_i) \)

The weights for \( x^* \) are calculated by minimizing the error variance

\[
V_e = Var\left[ \hat{f}(x^*) - f(x^*) \right]
\]

subject to

\[
\sum_{i=1}^{n} w_i = 1
\]

Given by the unbiased condition and second-order stationarity
Contents

- Background
- Autonomous learning of ball passing skills
- Hybrid method for trial reduction
- Experimental results
  - Minimization of test functions
  - Learning of ball passing skills
- Conclusions
Minimization problems of test functions

Multiple peaks

Single peak with a global view

The shape of test functions in 2 dimensions

Dependency of variables

(Sano et al. (2000) also utilized these test functions for evaluating distributed GA)
## Comparison of trial rates and error rates

### Trial rates and error rates in 100 candidates (lower = better)

<table>
<thead>
<tr>
<th>Function in 10 dim.</th>
<th>GAT [Kobayashi et al. 2007]</th>
<th>GATS (this work)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial rate (%)</td>
<td>Error rate (%)</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>54.20</td>
<td>0.80</td>
</tr>
<tr>
<td>Schwefel</td>
<td>62.84</td>
<td>0.87</td>
</tr>
<tr>
<td>Griewank</td>
<td>48.24</td>
<td>0.09</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>54.75</td>
<td>0.06</td>
</tr>
<tr>
<td>Ridge</td>
<td>55.42</td>
<td>0.04</td>
</tr>
<tr>
<td>Ackley</td>
<td>60.37</td>
<td>0.92</td>
</tr>
</tbody>
</table>

(Each value is the average over 100 experiments)

**Trial rate** = \[
\frac{\#(\text{tried candidates})}{\#(\text{all candidates})} \times 100
\]

**Error rate** = \[
\frac{\#(\text{wrongly thinned out candidates})}{\#(\text{thinned out candidates})} \times 100
\]

SGA: Simple GA  
GAT: SGA with Thinning-out  
GATS: GAT with Surrogate func.

The trial rate of SGA is always 100%
### Comparison of trial rates and error rates

#### Trial rates and error rates in 100 candidates

(Each value is the average over 100 experiments)

<table>
<thead>
<tr>
<th>Function in 10 dim.</th>
<th>GAT [Kobayashi et al. 2007]</th>
<th>GATS (this work)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial rate (%)</td>
<td>Error rate (%)</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>54.20</td>
<td>0.80</td>
</tr>
<tr>
<td>Schwefel</td>
<td>62.84</td>
<td>0.87</td>
</tr>
<tr>
<td>Griewank</td>
<td>48.24</td>
<td>0.09</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>54.75</td>
<td>0.06</td>
</tr>
<tr>
<td>Ridge</td>
<td>55.42</td>
<td>0.04</td>
</tr>
<tr>
<td>Ackley</td>
<td>60.37</td>
<td>0.92</td>
</tr>
</tbody>
</table>

The trial rate of SGA is always 100%.

SGA: Simple GA
GAT: SGA with Thinning-out
GATS: GAT with Surrogate func.
### Comparison of minimum scores

#### Minimum scores in 100 trials (lower = better)

<table>
<thead>
<tr>
<th>Function in 10 dim.</th>
<th>SGA</th>
<th>GAT</th>
<th>GATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rastrigin</td>
<td>260</td>
<td>165</td>
<td>152</td>
</tr>
<tr>
<td>Schwefel</td>
<td>3583</td>
<td>1817</td>
<td>1305</td>
</tr>
<tr>
<td>Griewank</td>
<td>621</td>
<td>211</td>
<td>112</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>17472</td>
<td>&gt; 3326</td>
<td>&gt; 2265</td>
</tr>
<tr>
<td>Ridge</td>
<td>5.7e9</td>
<td>6.4e8</td>
<td>2.3e8</td>
</tr>
<tr>
<td>Ackley</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

(Each value is the average over 100 experiments)

SGA: Simple GA  
GAT: SGA with Thinning-out  
GATS: GAT with Surrogate func.
Learning of Passing skills

- Initial motion: Forward chest shooting
  - Search space: 48 dim. (=8 joints × 6 key-frames)
  - Shooting distance: 1500 mm

- Distance to the objective: 800 mm

- Min. of passing distances in the passing challenge

Passing challenge in RoboCup

Initial phase of the experiment

https://youtu.be/QKuRUwwTUbQ
Learning results

<table>
<thead>
<tr>
<th>Number of Trials</th>
<th>Score (higher = better)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Proposed hybrid method)</td>
</tr>
<tr>
<td></td>
<td>(Previous method)</td>
</tr>
<tr>
<td></td>
<td>(Normal method)</td>
</tr>
</tbody>
</table>

SGA: Simple GA
GAT: SGA with Thinning-out
GATS: GAT with Surrogate func.
Learned passing skills

Later phase of the experiment
(accuracy of about 3 cm)
https://youtu.be/WiDadAzfasg
Conclusions and future work

- Autonomous learning of ball passing skills
- Hybrid method for trial reduction combining thinning-out and surrogate functions
- The first application of thinning-out in the real world

Future work
- Extension to two-dimensions
- Adaptation to arbitrary distances
Thank you for your attention