2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Adaptation of Quadruped Gaits Using Surface Classification and Gait Optimization

Jeong-Jung Kim[1] and Ju-Jang Lee[2], *Fellow, IEEE*

***Abstract*— An evolutionary computational approach for a gait generation of a quadruped robot autonomously generates a gait that adapts in an environment. In this approach, a fitness function that measures a performance of the gait is defined and parameters are optimized by maximizing or minimizing the function with evolutionary computation algorithms. However the previous research only has considered the optimization on an environment. In this paper, we suggest a gait adaptation method for a quadruped robot using a terrain classification and a gait optimization for an adaptation on various surfaces. The surfaces for the adaptation are learnt with a classification algorithm and a gait parameter on each surface is optimized with Particle Swarm Optimization (PSO). After the learning and the optimization, the classifier is used for classifying a surface that a robot is located and an optimized gait parameter is selected based on the classification result for the adaptation. The adaptation framework, a feature design and a filtering method for a classifier and a gait design for a quadruped robot are proposed in this paper. The proposed method was verified in a realistic 3D simulator and it successfully classified surfaces and selected optimized gaits for adaptations.**

I. INTRODUCTION

Most mammals live on land have four legs and this kind of structure has high mobility. In robotics, a quadruped robot was developed inspired from it. It has much higher mobility than a wheel-type robot and produces a faster and more stable gait than a biped robot. It has been conducted many research related to a generation of a gait for a quadruped robot due to these advantages. The research about the gait adaptation can be divided into three categories, planning a foot hold on a rough terrain, a momentary adaptation of gait shape on a surface with small obstacles, and an optimization of a gait on a flat surface.

Recently DARPA (Defense Advanced Research Projects Agency) supported several research groups for the first approach and it has been shown a significant improvement [1]–[4]. The approach hierarchically divides the control architecture and designs each level for safely crossing a terrain. However it has limitations that it needs many sensors such as a laser sensor, a stereo camera and a tracking system to plan gait on the terrain and it does not produce optimized gait on a flat surface. The [13]–[15] conducted research for the second approach. It generates a gait with Central Pattern Generator (CPG) and a gait is changed immediately by using the sensor information such as an accelerometer and a gyro sensor for the adaptation. The approach can modify the gait when an obstacle on a surface is small and the surface is similar to the pre-designed one. However, the adaption of the gait on a surface where it is different from the pre-designed one is tough.

The third approach generates an optimized gait on a pre-defined surface [5]–[10]. In this approach, a fitness function that measures a performance of a gait is defined and parameters are optimized for maximizing or minimizing the function with evolutionary computation algorithms such as genetic algorithms (GAs) and particle swarm optimization (PSO). The advantage of the approach is that it optimizes parameters without any mathematical model of the robot and the environment. However, the approach has considered only one environment or surface for the optimization. It limits the autonomous gait adaptation of the quadruped robot in various surfaces.

In this paper, a gait adaptation method for a quadruped robot using a surface classification and a gait optimization is suggested. The proposed approach is an extension of the previous research [5]–[10] for a gait optimization on a surface to gait adaptations on various surfaces. The surfaces for the adaptations are classified with a machine learning algorithm and a gait on each surface is optimized with PSO. In detail, a feature vector and a filtering method for the classification and the gait design for the optimization are suggested. The feature vector is generated with a relative position, velocity, and acceleration of the robot and the filtering method eliminates a fluctuation of the classification result for improving the classification accuracy. The designed gait is simple but produces various gaits for adaptations on surfaces.

The paper is organized as follows. The gait adaptation with a surface classification and an optimization are shown in Section II. Conditions and results for simulations are presented in Section III. Finally, the conclusions and the future works are given in Section IV.

II. GAIT ADAPTATION WITH SURFACE CLASSIFICATION AND OPTIMIZATION

The proposed gait adaptation method consists of two stages. In the first stage, a surface classifier is trained using a machine learning algorithm and gaits are optimized on surfaces using Particle Swarm Optimization (PSO). And in the second stage, the trained classifier detects the surfaces and the optimized gait parameter is selected for adapting on a surface.

[1]J.-J. Kim and [2]J.-J. Lee are with the Department of Electrical Engineering, KAIST, Dajeon, the Republic of Korea {rightcore@, jjlee@ee.}kaist.ac.kr

```
procedure DetectSurface(LS)                    ▷ LS is an array of classification labels from t-N×TS to t
    label_count[i] ← 0, i = 1, ..., C                                    ▷ C is a number of classes
    n ← 0
    for n = 1, ..., N do                 ▷ Cumulate the number of labels for each class during N×TS
        L ← LS[n]
        label_count[L] ← label_count[L] + 1
    end for
    FL_t ← arg max_1 (label_count[i])
    return FL_t                         ▷ Return final class label that has maximum label count
end procedure
```

Fig. 1. Surface detection algorithm with filtering. It eliminates the sudden change of classification label and returns stable class label.

A. Surface Classification for Quadruped Robot

We suggest a method that classifies surfaces by using a machine learning algorithm for a quadruped gait. A machine learning algorithm builds a classifier from data. A surface classification for wheel-type robot using a machine learning algorithm was proposed in [20], [21]. In this paper, we follow the similar procedure in the previous study but a feature vector for a quadruped robot is suggested. We assume that a robot can measure its acceleration and speed. A surface change is reflected to the robot's movement and it is measured by an internal sensor of the robot or an external sensor. After walking a robot with a user-designed gait on surfaces for adaptations, the sensor values are used as an input of a classifier. The sensor value on a quadruped robot fluctuates than wheel-type robot because of a counteraction from a ground and a leg change for the gait. Series of sensor data should be considered to capture the property. However when we consider the series, number of the input for an algorithm is large and needs many data to train an algorithm. Seven features from the series are selected to reduce the size of the input and classify surfaces effectively. The features are speed of the robot, mean of the acceleration and standard deviation of the acceleration in x, y and z axis during N sampling time, where N is a number of sampling time. A speed of the robot $robot_{speed}$ is estimated by

$$robot_{speed}(t) = \frac{||robot_{pos}(t) - robot_{pos}(t - N \times TS)||}{N \times TS} \tag{1}$$

where $robot_{pos}(t)$ and $robot_{pos}(t - N \times TS)$ are position of the robot at time t and $t - N \times TS$ and TS is a sampling period. Mean value of the acceleration in x, y and z axis are estimated by

$$acc_{mean_{x,y,z}}(t) = \frac{\sum_{i=t-N\times T}^{t} acc_{x,y,z}(i)}{N} \tag{2}$$

where $acc_x(t)$, $acc_y(t)$ and $acc_z(t)$ are acceleration at time t in axis x, y and z respectively. The standard deviation value of the acceleration in x, y and z axis are estimated by

$$acc_{std_{x,y,z}}(t) = \frac{\sum_{i=t-N\times T}^{t} (acc_{,y,zx}(i) - acc_{mean_{x,y,z}}(t))^2}{N-1} \tag{3}$$

The features are scaled range from -1.0 to 1.0 before they are used as the input of the classification algorithm. Final features for a classification algorithm are

$$\begin{aligned} \boldsymbol{x_t} \;=\; & (robot_{speed}(t), \\ & acc_{mean_x}(t), acc_{mean_y}(t), acc_{mean_z}(t), \\ & acc_{std_x}(t), acc_{std_y}(t), acc_{std_z}(t)). \end{aligned} \tag{4}$$

To classify surfaces with the feature vector, Support Vector Machine (SVM) is used as machine learning algorithm. The SVM is one of the popular method for a pattern classification that uses a concept maximum margin [22]. The trained SVM produces the surface label L_t at each sampling time by using the feature vector defined at (4). Even though the continuous characteristic of the sensor data is considered in the feature vector, the classifier occasionally produces mismatched a label because of the movement of a robot. To reduce the effect, a filtering algorithm is suggested and it is shown in Fig. 1. The algorithm considers instant classification results between $t - N \times TS$ and t sampling time and produces the final label FL_t that has a maximum count number of instant result label during the period. The final label FL_t indicates the surface and a gait is selected based on the label for an adaptation.

B. Gait Design for Quadruped Robot with Cubic Splines

The cubic spline generates a smooth trajectory between an initial position and a final position, when an initial position p_0, an initial velocity $\dot{p}_0$, a final position p_f, a final velocity $\dot{p}_f$ and a total time t_f are specified [19]. A continuous trajectory is generated with the cubic spline in Cartesian coordinate and the position is converted into target joint angles of each leg by inverse kinematics at each sampling time. A total trajectory of each leg for a period is composed with three points and those are connected with two cubic splines as shown in Fig. 2. It starts from $\vec{p}_s = (x_s, y_s)$ to $\vec{p}_v = (x_v, y_v)$ to $\vec{p}_f = (x_f, y_f)$ and goes back to the p_s. The backward trajectory is generated by setting start position as $\vec{p}_f$ and final position as $\vec{p}_s$. The forward trajectory is composed of two splines. By setting a continuous acceleration at the via point, two-segment spline is connected and produces a smooth trajectory that goes through the via point. The equations for the first segment

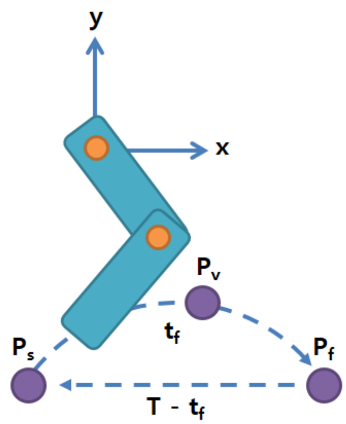


Fig. 2. Trajectory generation with cubic spline. A total trajectory of each leg for a period is composed with three points and those are connected with two cubic splines. Each leg follows the same trajectory, but has a phase to generate a gait of quadruped robot.

and the second segment are

$$\begin{aligned}
a_{10} &= p_0\\
a_{11} &= 0\\
a_{12} &= \tfrac{12p_v-3p_f-9p_0}{4t_f^2}\\
a_{13} &= \tfrac{-8p_v+3p_f+5p_0}{4t_f^3}\\
p_1(t) &= a_{10}+a_{11}t+a_{12}t^2+a_{13}t^3\\
a_{20} &= p_v\\
a_{21} &= \tfrac{3p_f-3p_0}{4t_f}\\
a_{22} &= \tfrac{-12p_v+6p_f+6p_0}{4t_f^2}\\
a_{23} &= \tfrac{8p_v-5p_f-3p_0}{4t_f^3}\\
p_2(t) &= a_{20}+a_{21}t+a_{22}t^2+a_{23}t^3.
\end{aligned} \tag{5}$$

The time duration for the total trajectory is T, for the forward trajectory is t_f and the backward trajectory is $T-t_f$. The gait of the robot is realized by assigning the four total trajectories to legs and coordinating them. By changing the start time of each leg trajectory, various gaits can be generated. The crawl gait is considered in the paper. In the case of the gait, each leg has the same parameters, $\vec{p}_s$, $\vec{p}_v$, $\vec{p}_f$, T and t_f, and left hind leg, left front leg, right hind leg, and right front leg start to move at $0T$, $0.25T$, $0.5T$, and $0.75T$, respectively. We use fixed value for the phases and optimize the parameters $\vec{p}_s$, $\vec{p}_v$, $\vec{p}_f$, T and t_f with PSO for generating a gait of a quadruped robot. The proposed gait design is simple but can produce various gaits that adapt on surfaces.

C. Gait Optimization of Quadruped Robot by PSO

PSO is a population based stochastic optimization method and is inspired by a social behavior of populations that move to a goal by observing movements of neighborhood individuals in the nature [16], [17]. Gaussian Swarm [18] is used for optimizing the parameter of the gait at each surface. Each parameter for the gait is sent to a simulator or a physical robot and the parameters that satisfies constraints are optimized. The constraints are

TABLE I
THE PARAMETERS FOR QUADRUPED GAIT

Parameters	Range
T	$0.1 \sim 2$ s
t_f	$0.1 \sim 2$ s
x_s	$-(l_1+l_2) \sim (l_1+l_2)$ m
y_s	$-(l_1+l_2) \sim 0$ m
x_v	$-(l_1+l_2) \sim (l_1+l_2)$ m
y_v	$-(l_1+l_2) \sim 0$ m
x_f	$-(l_1+l_2) \sim (l_1+l_2)$ m
y_f	y_s

$$\begin{aligned}
&t_f < T,\\
&x_s < x_v < x_f,\\
&y_s < y_v,\\
&y_f < y_v.
\end{aligned}$$

The constraints prevent an abnormal movement of the robot. The fitness evaluation is conducted for a predefined duration with the parameter encoded in each particle and it returns the fitness value to the overall PSO procedure. The fitness is absolute distance from the origin position R_s to final position R_f of the robot during the predefined time and the high fitness value represents that the parameter generates a gait that can produce a fast movement. The function is defined as

$$f = |R_f - R_s|. \tag{6}$$

When an unreachable gait is generated, the fitness evaluation terminates and returns a distance of the robot at that time. The parameters and their range for optimizing the gait are summarized in Table I. The dimension of the parameters for the optimization is only seven. The low dimension is important for an application of the evolutionary algorithms to robotics because usually lower dimensions require less iteration for the optimization. The procedure is repeated as the number of surfaces for the adaptations. The optimized parameter is selected based on the classification labels of a surface for adaptation.

III. SIMULATION

A model of the quadruped robot used in simulations, surface classifications, optimizations of the robot and the gait adaptation results are shown in this section. The gait adaptation of a quadruped robot was conducted in a simulator implemented with Open Dynamics Engine (ODE) [24]. The sampling time for the simulator was set as 0.02s and properties of surfaces were changed by varying the friction parameter μ on the simulator. The proposed method for the adaptation of gaits was applied on four surfaces.

A. Model of Quadruped Robot

The robot has 12 joints and 3 degrees of freedoms (DoFs) were assigned to each leg. Only 2 DoFs for each leg were used in the paper because we considered forward walking movement. The model of the quadruped robot is shown in Fig. 3 and the parameter values are summarized in Table II. The parameters $body_{width}$, $body_{height}$, $body_{length}$, l_1, and l_2 represent width of the body, and height of the body, length

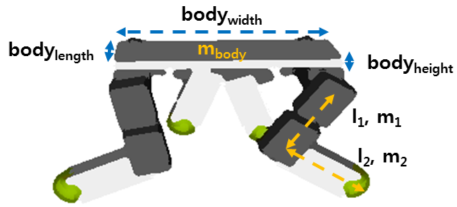


Fig. 3. Model of quadruped robot

TABLE II
THE PARAMETERS FOR EACH RIGID BODY

Parameters	Value
$body_{width}$	0.225 m
$body_{height}$	0.01 m
$body_{length}$	0.105 m
l_1	0.1342 m
l_2	0.102 m
m_{body}	0.1 kg
m_{l_1}	0.134 kg
m_{l_2}	0.11 kg

TABLE III
THE USER-DESIGNED PARAMETERS FOR QUADRUPED GAIT

Parameters	Value
T	1.0 s
t_f	0.5 s
x_s	-0.05 m
y_s	-0.15 m
x_v	0 m
y_v	-0.12 m
x_f	0.0.5 m
y_f	-0.15 m

of the body, length of the upper leg, length of the lower leg, respectively. And the parameters m_{body}, m_{l_1}, and m_{l_2} represent the mass of the body, upper leg, and lower leg, respectively.

B. Surface Classification Result

We used the libsvm [23] to implement the SVM that classifies surfaces. A data set for training SVM was taken after walking on four surfaces with a user-designed gait. The parameters for the gait is shown in Table III. The robot was walked on surfaces that has different friction parameter $\mu = 0.25$, $\mu = 0.5$, $\mu = 0.75$ and $\mu = 1.0$ and each classification result was labeled as $label_1$, $label_2$, $label_3$ and $label_4$, respectively. The parameter N was set as 10. The acceleration and velocity of the robot for the feature vector were calculated by numerical differential from position of the robot body. The number of data for each surface was 1000 and the SVM was trained with the data by 10 fold cross validation for selecting the best parameters of the SVM. The classification accuracy result with the setting was 98.88% which shows the proposed classification feature vector is suitable for classifying the surface changes. The trained SVM

TABLE IV
CLASSIFICATION ACCURACY

Friction parameter	Accuracy	
	Instant classification	Final classification
$\mu = 0.25$	100%	100%
$\mu = 0.5$	88.05%	98.41%
$\mu = 0.75$	82.87%	98.01%
$\mu = 1.0$	86.06%	97.21%

TABLE V
THE OPTIMIZED PARAMETERS OF QUADRUPED GAIT ON SURFACES

Parameters	Optimized value			
	$\mu = 0.25$	$\mu = 0.5$	$\mu = 0.75$	$\mu = 1.0$
T	0.5499 s	0.6440 s	0.5148 s	0.8324 s
t_f	0.2517 s	0.3364 s	0.2019 s	0.3928 s
x_s	-0.1477 m	-0.1301 m	-0.1341 m	-0.1482 m
y_s	-0.094 m	-0.0996 m	-0.1096 m	-0.0984 m
x_v	-0.020 m	-0.0247 m	0.0603 m	-0.0243 m
y_v	-0.087 m	-0.0830 m	-0.0801 m	-0.0866 m
x_f	0.1357 m	0.1305 m	0.07881 m	0.1310 m
y_f	-0.094 m	-0.0996 m	-0.1096 m	-0.0984 m

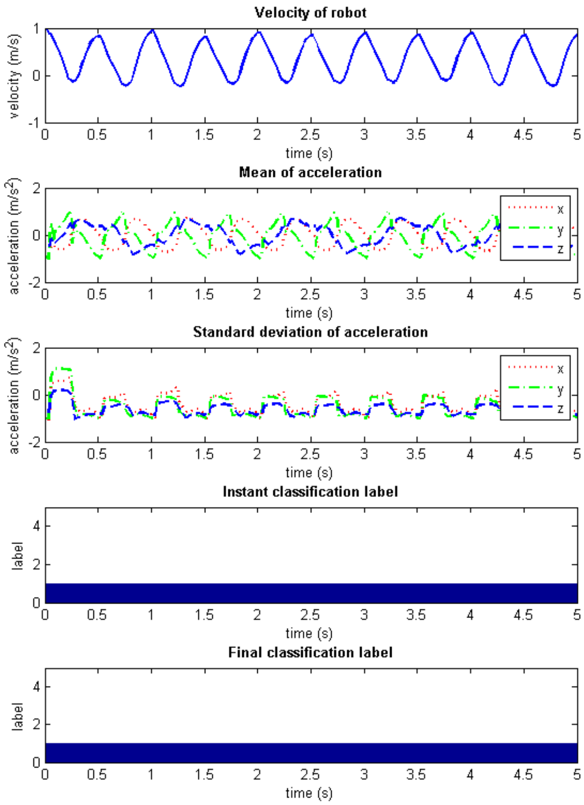
was used for classifying the surface with the response of the robot at every sampling time. The classification results on the surfaces are shown in Fig. 4. First, second, and thirds rows show the input vector for the classifier and fourth row shows the instant classification result and the last row shows the final classification result on the surface. Even though there were frequent fluctuations at the instant classification results, final class labels showed steady classification result. This result represents that proposed classification filter method effectively eliminates fluctuation on the instant classification labels without degrading the accuracy. The result on the instance classification accuracy and the final classification accuracy for the classifier design is shown in Table IV. The instant classification result shows over 80 % accuracy and final classification result filtered by proposed method shows over 97 % accuracy that both are enough to detect surfaces for adaptations.

C. Gait Optimization Result

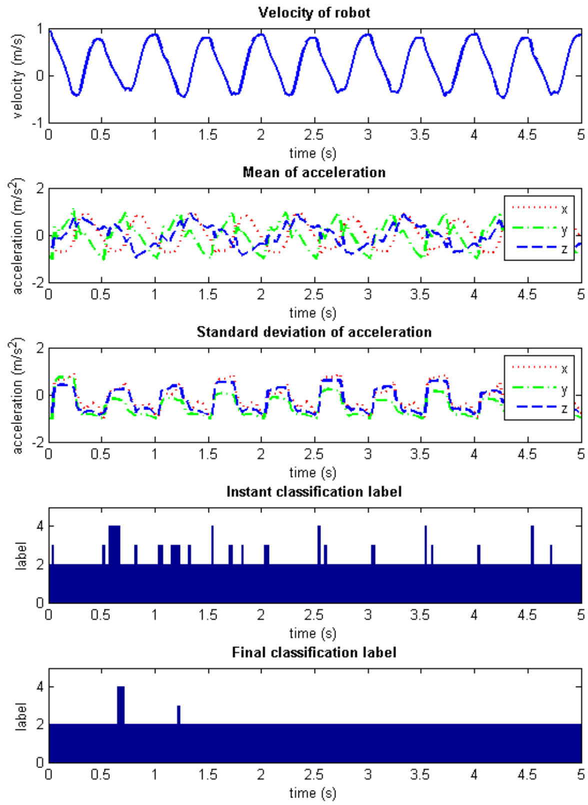
The gait was optimized on the each surface with PSO algorithm. The swarm size was set as 50 and the maximum generation was set as 50. Fitness evaluation for each particle was conducted for during 5 seconds. The gait parameters after the optimizations are summarized in Table V. The initial fitness for each surface was 0 or 0.5, but it reached about 2.5 at the final generation. It represents that the optimization process found the parameters that can move to a position apart 2.5 meters away from the initial position during the predefined 5 seconds on different surfaces. The optimization process found the parameter that was suitable for each surface and it is used for generating a gait for an adaptation.

D. Gait Adaptation Result

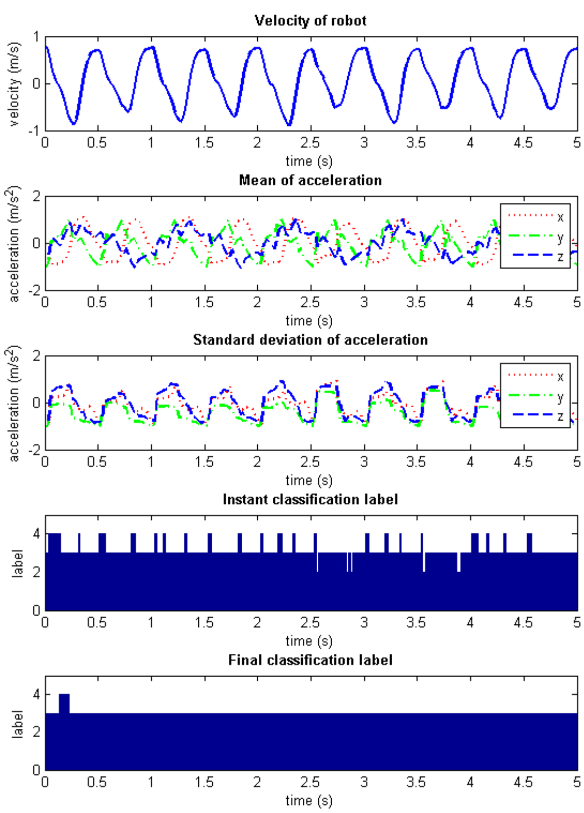
With the trained classifier and the optimized gait parameter from the previous sections, the gait adaption simulation was



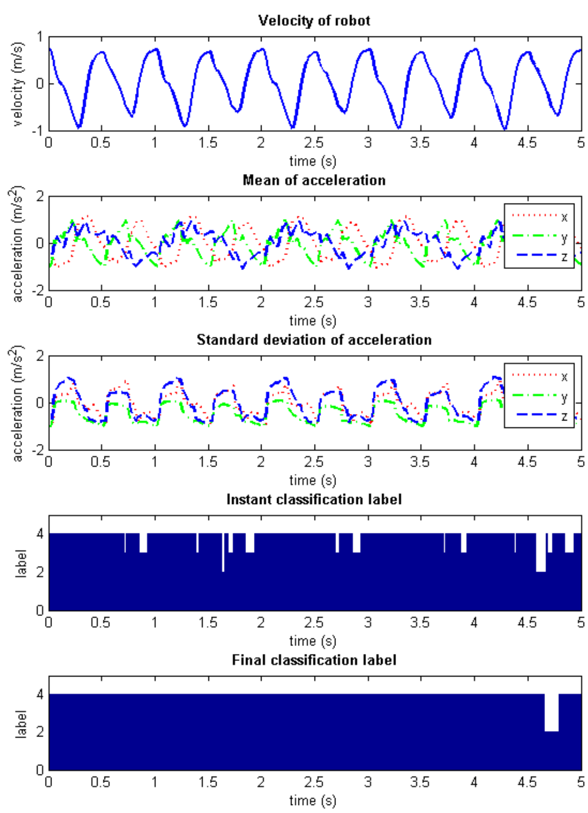
(a) $\mu = 0.25$



(b) $\mu = 0.5$

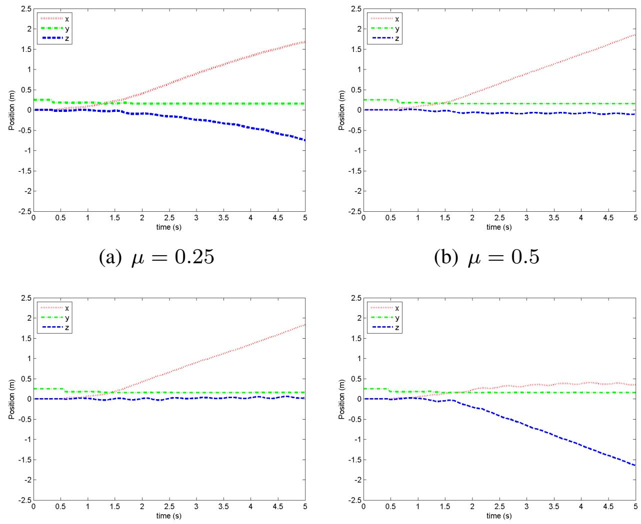


(c) $\mu = 0.75$



(d) $\mu = 1.0$

Fig. 4. The classification result on surfaces



(a) $\mu = 0.25$ (b) $\mu = 0.5$

(c) $\mu = 0.75$ (d) $\mu = 1.0$

Fig. 5. Adaptation result on the surfaces

TABLE VI

THE AVERAGE VELOCITY

Friction parameter	**Average velocity**	
	Before adaptation	After adaptation
$\mu = 0.25$	0.1582 m/s	0.4655 m/s
$\mu = 0.5$	0.2484 m/s	0.4562 m/s
$\mu = 0.75$	0.1909 m/s	0.4566 m/s
$\mu = 1.0$	0.1466 m/s	0.4326 m/s

conducted. The robot was walked on each surface with the user-designed gait parameter, the surface was classified at 1.2 seconds and the gait parameter was changed to the optimized parameter based on the classification result for the adaptation. The simulation results are shown in Fig. 5. The parameters of the gait were adapted on each surface after 1.2 seconds. To see the performance improvement after the adaptation, the average velocity of the robot was considered. The average velocity before the adaption was calculated by dividing the distance from the initial position of the robot to its position at 1.2 second by the 1.2. The average velocity after the adaption was calculated by dividing the distance from the position of the robot at 1.2 second to its position at 5 second by 3.8. The average velocity comparisons between before adaptation and after adaptation are summarized in Table. VI. The average velocity of the robot after adaptation shows around 0.45 m/s which is two or three times faster than before adaptation. The result tells us that the proposed method using the classification and the gait optimization is suitable for adaptation for the surfaces.

IV. CONCLUSIONS

In this paper, we suggested the gait adaptation method for a quadruped robot using a surface classification and a gait optimization with PSO. The surfaces for the adaptation were classified with a machine learning algorithm and a gait on each surface was optimized with PSO. The method was tested with gait adaptations on four different surfaces and it produced the gaits that were suitable for walking on each surface. We are planning to improve the proposed system to an incrementally learning adaptation system for unseen surfaces.

REFERENCES

[1] D. Pongas, M. Mistry and S. Schaal, “A Robust Quadruped Walking Gait for Traversing Rough Terrain”, *in Proc. IEEE Int. Conf. Robotics and Automation*, 2007, pp. 1474-1479.
[2] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor and S. Schaal “Compliant Quadruped Locomotion over Rough Terrain,” *in Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2009, pp. 814-820.
[3] J. Z. Kolter, M. P. Rodgers and A. Y. Ng, “A Control Architecture for Quadruped Locomotion Over Rough Terrain,” *in Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2008, pp. 811-818.
[4] M. Zucker, J. A. Bagnell, C. G. Atkeson and J. Kuffner, “An Optimization Approach to Rough Terrain Locomotion,” *in Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2010, pp. 1474-1479.
[5] G. S. Hornby, M. Fujita, S. Takamura, T. Yamamoto and O. Hanagata, “Autonomous Evolution of Gaits with the Sony Quadruped Robot,” *in Proc. Genetic Evolutionary Computation Conference*, 1999, pp. 1297-1304.
[6] G. S. Hornby, S. Takamura, T. Yamamoto and M. Fujita, “Autonomous Evolution of Dynamic Gaits With Two Quadruped Robots,” *IEEE Trans. Robotics*, vol. 21, pp. 402-410, 2005.
[7] C. Rong, Q Wang, Y. Huang, G. Xie, and L. Wang, “Autonomous Evolution of High-Speed Quadruped Gaits Using Particle Swarm Optimization,” *in RoboCup 2008: Robot Soccer World Cup XII*, Springer, pp. 259-270, 2009.
[8] S. Chernova and M. Veloso, “An Evolutionary Approach to Gait Learning for Four-legged Robots,” *in Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2004, pp. 2562-2567.
[9] T. Röfer, “Evolutionary Gait-optimization Using a Fitness Function based on Proprioception,” *in RoboCup 2004: Robot Soccer World Cup VIII, Lecture Notes in Computer Science*, Springer, pp. 310-322, 2005.
[10] G. S. Hornby, S. Takamura, J. Yokono and O. Hanagata, T. Yamamoto and M. Fujita, “Evolving robust gaits with AIBO,” *in Proc. IEEE Int. Conf. Robotics and Automation*, 2000, pp. 3040-3045.
[11] D. P. Krasny and D. E. Orin, “Generating High-speed Dynamic Running Gaits in a Quadruped Robot using an Evolutionary Search,” *IEEE Trans. Syst.,Man, Cybern. B*, vol. 34, no. 4, pp. 1685-1695, 2004.
[12] N. Kohl and Peter Stone, “Machine Learning for Fast Quadrupedal Locomotion,” *in Proc. 9th AAAI*, pp. 611-616, July, 2004.
[13] Y. Fukuoka, H. Kimura and A. H. Cohen, “Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain based on Biological Concepts,” *Int. J. Robotics Research*, vol 22., pp. 187-202, 2003.
[14] K. Tsujita, K. Tsuchiya and A. Onat, “Adaptive Gait Pattern Control of a Quadruped Locomotion Robot,” *in Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2001, pp. 2318-2325.
[15] C. Liu, Q. Chen, and D. Wang, “CPG-Inspired Workspace Trajectory Generation and Adaptive Locomotion Control for Quadruped Robots,” *IEEE Trans. Syst.,Man, Cybern. B*, vol. 41, no. 3, pp. 867880, 2012.
[16] J. Kennedy and R.C Eberhart, “Particle Swarm Optimization,” *in Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1942-1948.
[17] Y. Shi and R. C. Eberhart, “A Modified Particle Swarm Optimizer,” *in Proc. IEEE Cong. Evolutionary Computation*, 1998, pp. 69-73.
[18] R. A. Krohling, “Gaussian swarm: a novel particle swarm optimization algorithm,” *in Proc. IEEE Int. Conf. Cybernetics and Intelligent Systems*, 2004, pp. 372-376.
[19] J. J. Craig, *Introduction to Robotics 3rd edition*, Prentice Hall, 2004.
[20] C. Weiss, H. Fröhlich and A. Zell, “Vibration-based Terrain Classification Using Support Vector Machines,” *in Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2006, pp. 4429-4434.
[21] P. Komma, C. Weiss, and A. Zell, “Adaptive Bayesian Filtering for Vibration-Based Terrain Classification,” *in Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3307-3313, Mar. 2009.
[22] E. Alpadyn, *Introduction to Machine Learning*, MIT Press, 2004.
[23] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” [Online]. Available:*http://www.csie.ntu.edu.tw/~cjlin/libsvm*, 2001.
[24] R. Smith, “Open Dynamics Engine (ODE),” [Online]. Available: *http://www.ode.org/*, 2012