Robotica (2010) volume 28, pp. 929–936. © Cambridge University Press 2010
doi:10.1017/S0263574709990804

Parameter self-adaptation in biped navigation employing nonuniform randomized footstep planner

Zeyang Xia†,‡,*, Jing Xiong† and Ken Chen†,§

†*Department of Precision Instruments and Mechanology, Tsinghua University, Beijing 100084, China.*
‡*Mechanical and Aerospace Engineering, Nanyang Technological University, 639798 Singapore.*
§*State Key Laboratory of Tribology, Tsinghua University, Beijing 100084, China.*

(Received in Final Form: December 3, 2009. First published online: January 15, 2010)

SUMMARY

In our previous work, a random-sampling-based footstep planner has been proposed for global biped navigation. Goal-probability threshold (GPT) is the key parameter that controls the convergence rate of the goal-biased nonuniform sampling in the planner. In this paper, an approach to optimized GPT adaptation is explained by a benchmarking planning problem. We first construct a benchmarking model, in which the biped navigation problem is described in selected parameters, to study the relationship between these parameters and the optimized GPT. Then, a back-propagation (BP) neural network is employed to fit this relationship. With a trained BP neural network modular, the optimized GPT can be automatically generated according to the specifications of a planning problem. Compared with previous methods of manual and empirical tuning of GPT for individual planning problems, the proposed approach is self-adaptive. Numerical experiments verified the performance of the proposed approach and furthermore showed that planning with BP-generated GPTs is more stable. Besides the implementation in specific parameterized environments studied in this paper, we attempt to provide the frame of the proposed approach as a reference for footstep planning in other environments.

KEYWORDS: Biped navigation; Randomized sampling; Parameter self-adaptation; Rapidly exploring random trees; Neural network.

1. Introduction

Humanoid robotics research has been one of the most exciting topics in the field of robotics. With its rapid progress in recent years, people concentrate more and more on their application in human services and other task executions besides their laboratory implementations. To realize the biped locomotion in existing unstructured human-living environments, the global navigation is an unavoidable and a new issue to be resolved.

Global path planning and navigation strategies for mobile robots require constructing boundary representation of obstacles in the configuration space, and they always make the robots circumvent obstacles.[1] However, biped robots, as well as other legged robots, have the ability of stepping over or upon certain kinds of obstacles, which makes it impossible to give exact obstacle boundary representations. In contrast, the sampling-based motion-planning approach, which only uses obstacle information by retral collision checking,[2,3] is a practical resolution for the global biped navigation. Kuffner and Chestnutt[4–8] realized the sampling-based footstep planning for global biped navigation using forward dynamic programming to compute footstep placement sequences. Ayaz[9,10] improved the smoothness of trajectories for posture transitions; Michel[11] applied this approach in some dynamic environments; Chestnutt[12,13] applied dynamic adjustment of the footstep transition model and furthermore implemented this approach to navigation for multilegged robots. Xia and Chen[14] implemented a compound footstep transition model that decreases the planning complexity.

All the above-mentioned methods are realized by using deterministic-sampling strategies, in which the expansion of the search trees is directed by deterministic functions. These functions are designed with the aim of generating relatively optimal sequences of footstep placements to reach the goal, such as smooth sequences with least numbers of footsteps and least unnecessary footsteps to step over or upon obstacles. The deterministic-sampling-based approaches are very practical in environment with open areas. However, a critical problem with the existing approaches is that they rely on the design of the sampling set in the footstep transition model and sampling-directing functions, which is usually referred to as resolution complete. This character may result in planning failures: the search tree may be not able to converge, i.e., may get trapped, in the required planning duration or sampling number of times. This situation is especially visible during the implementation in areas with local minima or/and narrow passages, which exist in human-living environments, such as homes or office buildings (see refs. [7, 15] for case studies). Besides the improvement on the design of footstep transition model or the sampling-directing function, another option regarding this problem is to improve the ability to randomly avoid obstacles of the footstep planner by employing random-sampling strategies instead of deterministic-sampling strategies.

In our previous works, a random-sampling-based footstep planner has been proposed.[15] In this planner, a goal-biased nonuniform sampling strategy is employed to improve the rate at which the sampling tree converges to the goal.

* Corresponding author. E-mail: zeyang.xia@ieee.org

Goal-probability threshold (GPT) is the key parameter that controls the convergence rate. In previous motion-planning implementation employing nonuniform sampling strategies, some parameters have to be manually and empirically preset according to individual planning problems. Considering (a) that it is difficult to preset an optimized GPT manually and empirically and (b) that it is impossible to preset a fixed GPT that is an optimized one for a planning problem with changing specifications, a self-adaptive approach to GPT optimization for our proposed footstep planner is quite necessary.

The current paper proposes a self-adaptive approach to GPT optimization. We first construct a benchmarking model, in which the biped navigation problem is described in selected parameters, to study the relationship between these parameters and the optimized GPT. Then, a back-propagation (BP) neural network is employed to fit this relationship. With a trained BP neural network modular, the optimized GPT can be automatically generated according to the specifications of a planning problem.

The rest of the paper is organized as follows: Section 2 introduces the randomized-footstep-planning approach and the employed goal-biased sampling strategy; Section 3 studies the characters of GPT basing on a benchmarking problem of planning in environments with local-minima areas; Section 4 employs a BP neural network to realize the GPT self-adaptation; Section 5 implements the proposed parameter self-adaptive approach and verifies its performance with comparison experiments; Section 6 discusses the further implementation of the proposed approach and concludes the present paper.

2. Randomized Footstep Planning

2.1. Sampling-based footstep planning

The sampling-based footstep planner is a biped navigation algorithm.[15,16] It builds a search tree originated from the initial footstep placement of the biped robot. The search tree is expanded by footstep placement sampling in the planning space. Footstep placements resulting in collision are pruned from the tree by collision checking based on the robot state and environment information. The planning continues until some footstep placement in the search tree reaches the goal region. Figure 1 gives the block diagram of the footstep planner. The footstep placement sampling is directed according to the footstep transition model that predefines a discrete set of feasible footstep locations for the swing foot (see Fig. 2).

2.2. Multi-RRT-GoalBias footstep planner

To resolve the problems of existing approaches using deterministic-sampling strategy stated in the previous section, we propose a randomized-sampling strategy based on rapidly exploring random trees (RRTs).[15–17]

The key issue of the RRT-based footstep planner is to randomly produce a temporary goal region, in place of the actual goal region, at each step of footstep placement sampling. Directed by these temporary goal regions randomly distributed in the planning environment, the search tree is provided with a better ability of avoiding obstacles as well as traversing areas in which the sampling tree may get trapped. Figure 3 explains the single-step extension of the randomized footstep placement sampling.

We modified the RRT-based footstep planner in two aspects. The first one was to add all footstep placements of the footstep transition model to the search tree during

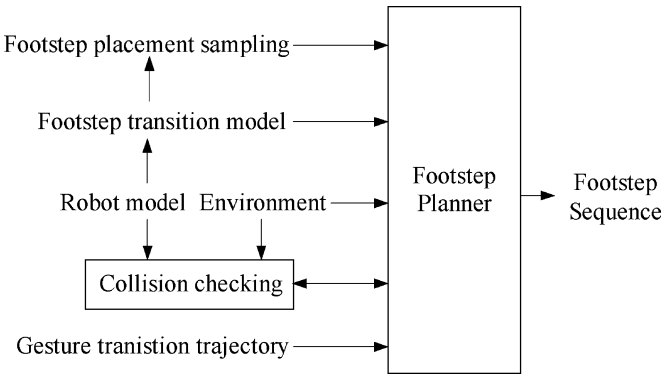

Fig. 1. Sampling-based footstep planner for humanoid robots.

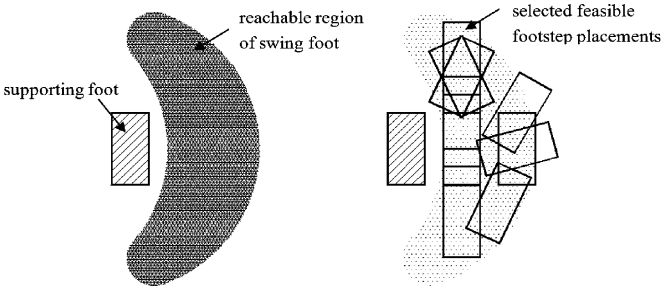

Fig. 2. Footstep transition model. When the robot is supported by one foot, we can obtain a region which can be reached by the swing foot (left); if we consider the height offsets, the 2-D region is expanded to a 3-D space. A set of footstep placements are selected from the reachable region, which configures the footstep transition model (left). Different footstep placements are used to perform different locomotion functions, such as walking forward/backward and turning left/right.

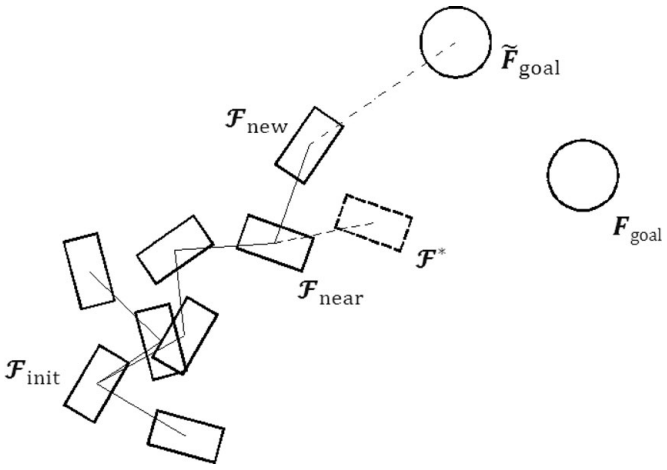

Fig. 3. The single-step expansion of the search tree of the RRT-based footstep planner: F_{init} denotes the initial footstep placement; F_{goal} and $\tilde{F}_{goal}$ denote the goal region and the temporarily produced goal region respectively; F_{new} denotes the footstep placement newly added to the search tree and is added to the search tree because it is the nearest one to the temporarily produced goal region $\tilde{F}_{goal}$ in the existing search tree. Note that the distance refers not to the Euclidean distance but to a distance in the footstep placement space.

https://doi.org/10.1017/S0263574709990804 Published online by Cambridge University Press

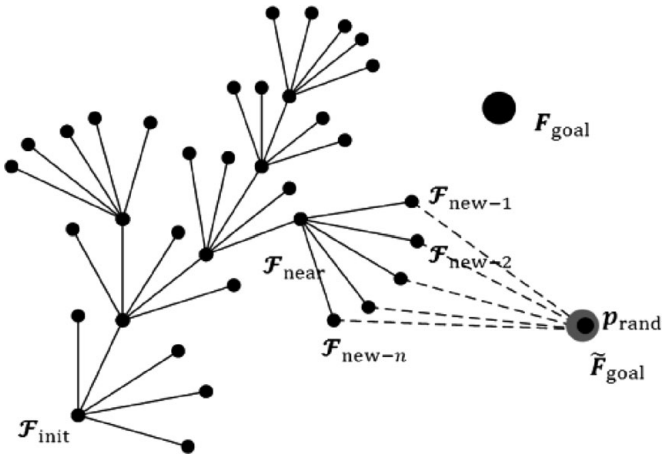

Fig. 4. A single-step expansion of the search tree of the Multi-RRT footstep planner. We use points to denote footstep placements in order to make understanding the figure easier.

Multi-RRT algorithm

1 Initial random tree T, add $\mathcal{F}_{init}$ to T;
2 **repeat while**(sampling time < required largest sampling time),
3 produce a random point $\boldsymbol{p}_{rand}$ and create $\widetilde{F}_{goal} = (\boldsymbol{p}_{rand}, d_{min})$;
4 find $\mathcal{F}_{near}$ nearest to $\widetilde{F}_{goal}$;
5 **if** $\rho_2(\mathcal{F}_{near}, \boldsymbol{p}_{rand}) < d_{min}$, jump to 3
6 **otherwise** $\mathcal{F}_{new-i}(i = 0,1,\ldots,n-1) \leftarrow \text{Extend}(\mathcal{F}_{near})$
7 **if** $\mathcal{F}_{new-i} \in \boldsymbol{F}_{goal}$, **jump** to 9
8 **otherwise** add $\mathcal{F}_{new-i}$ to T
9 search succeeds, output sequence from $\mathcal{F}_{init}$ to $\mathcal{F}_{new-i}$ in 7
10 search fails in required computing time or sampling time

Fig. 5. Algorithm of footstep planner using the Multi-RRT: ρ_2 denotes the measure function from a footstep placement to a region, which considers not only the Euclidean distance but also the height offset and orientation, among others.

Create Temporary Goal Region of Multi-RRT-GoalBias

1 Produce a probability number $P = \text{rand}() \in [0,1]$
2 **if** $P < P_{goal}$
3 return actual goal $\boldsymbol{F}_{goal}$ as temporary goal region $\widetilde{F}_{goal}$
4 **otherwise**
5 Return randomized $\widetilde{F}_{goal} = (\boldsymbol{p}_{rand}, d_{min})$

Fig. 6. Goal-biased random sampling controlled by P_{goal}: $P = \text{rand}()$ is a computer-generated probability number uniformly distributed in [0,1].

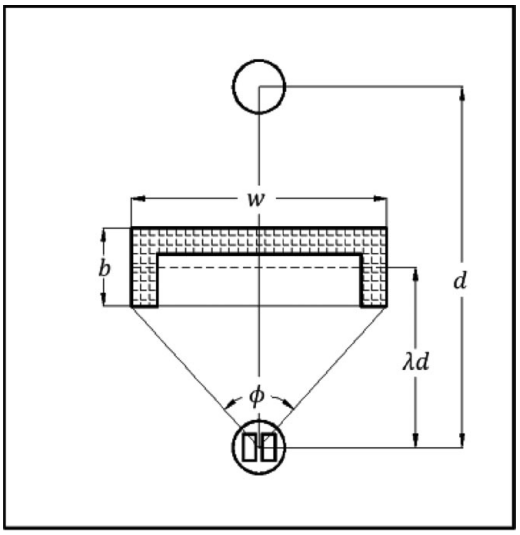

Fig. 7. A benchmarking model for a typical footstep planning problem in an environment with local minima. The planning is processed in a $4 \times 4\,m^2$ area. The bottom and top circles denote the original and goal regions of the biped robot separately, and the distance between them is d. The obstacle is $w \times b$ in dimension, and its center is in the line linking the original and goal regions of the robot. The distance between the obstacle center and the robot original region is λd. The field angle of the obstacle to the original region is ϕ.

the single-step expansion, since our previous studies verified that the character that the RRT only adds one footstep placement to the search tree may result in ill-conditioned footstep placement sequences (see refs. [15, 16] for details). Figure 4 shows the single-step expansion of the search tree that adds multiple footstep placements. Figure 5 gives the algorithm of the Multi-RRT footstep planner. Numerical experiments showed that the Multi-RRT inherits the ability of quick expansion in the planning space as well as the characteristic of probabilistic completeness from the basic RRT algorithm.[18]

Another modification was to apply a nonuniform sampling strategy: the sampling distribution is biased to the goal region controlled by a probability parameter, termed as GPT (denoted as $P_{goal} \in [0, 1]$). A goal-biased strategy can improve the rate at which the random tree converges to the goal region, compared with the way in which randomly sampled footstep placements are uniformly distributed in the planning space. While producing the temporary goal region (see step 3 of the Multi-RRT algorithm), the planner returns the actual goal region, instead of a randomized goal region, at a probability of P_{goal} (see Fig. 6). We termed the footstep planner with the above-mentioned improvements as "Multi-RRT-GoalBias footstep planner."

3. Goal-Probability Threshold: Case Study of a Benchmarking Problem

Now, P_{goal} is a parameter that controls the expansion of the random tree and dominantly affects rate of its convergence of random tree. Our previous sampling experiments provided some empirical guidelines for setting P_{goal} values.[15,16] However, a manually defined value cannot be assured to be an optimized one. In addition, in order to realize the planning in dynamic environments, it is necessary to have a solution for P_{goal} self-adaptation. Toward this objective, we investigate the characters of P_{goal} and self-adaptation of optimized P_{goal} (denoted as $O\,P_{goal}$), using the benchmarking footstep planning problems in an environment with local minima.

3.1. Benchmarking planning problem

A typical footstep planning problem in an environment with local minima is defined as showed in Fig. 7. To

https://doi.org/10.1017/S0263574709990804 Published online by Cambridge University Press

obviate unnecessary coupling effects, the original and goal regions of the robot are defined as constants. The field angle and the distance from the obstacle to the robot, ϕ and λd (see Fig. 7 for parameter descriptions), are two key parameters considered in robot motion decision for obstacle avoidance. The parameters in the benchmarking problem have a geometry constraint as

$$\phi = 2\tan^{-1}\left(\frac{\omega}{2\lambda d - b}\right). \tag{1}$$

So, instead of ϕ and λd, two independent parameters λ and w are selected to describe the planning problem[1]. A benchmarking planning problem, denoted as $\mathbf{P}$, can be parameterized as

$$\mathbf{P} = (\lambda, w) \in \mathrm{R}^2. \tag{2}$$

To ensure the universality, we have geometric constraints for benchmarking problem as follows: (a) A resolution exists; i.e., the original and goal regions of the robot are legal. (b) The robot cannot walk straightly to the goal; i.e., there must be obstacles between the original and goal regions. (c) No narrow-passage situation exists; i.e., there should be a passage wide enough between the obstacle edges and the margin of the planning area[2]. Considering these constraints, we have

$$\lambda \in [0.2,\ 0.8],\ w \in [0.4\,\mathrm{m},\ 3\,\mathrm{m}]. \tag{3}$$

In the implementation of sampling-based approaches, the sampling number of times is related to the time complexity of the planning algorithms and is independent of other planning modules and the computing hardware as well; so it is a logical index to indicate the convergence rate of the footstep planner. The analysis on randomized-sampling approaches is also based on statistical data of a number of experimental trials. So the average sampling number of times of a number of trials is used as the reference index of the benchmarking studies.

Let $\mathbf{P}|P_{\mathrm{goal}}$ be a planning object of planning $\mathbf{P}$ with P_{goal}. For each $\mathbf{P}|P_{\mathrm{goal}}$, we have 2×10^2 trials. The sampling upper limit for each trial is 1×10^3. We compute the average sampling number of times and its standard deviation of k times of successful trials as

$$\left.\begin{aligned} \bar{N} &= \frac{1}{k}\Sigma_{i=1}^{k} N_i, \\ \sigma &= \sqrt{\tfrac{1}{k-1}\Sigma_{i=1}^{k}(N_k - \bar{N})^2}, \end{aligned}\right\} \tag{4}$$

where $N_i (1 \le i \le k,\ k \le 200)$ is the sampling number of times of ith trial. Those trails with a deviation of over 3σ, i.e., $|N_i - \bar{N}| > 3\sigma$, are eliminated. The final average sampling number of times for reference, denoted as $\bar{N}(\mathbf{P}|P_{\mathrm{goal}})$, is computed with the rest of the trials by (4).

Table I. Benchmarking planning problems.

$\mathbf{P}$	(λ, w)	$\bar{N}_{\min}$	$P_{\mathrm{goal}}(\bar{N}_{\min})^a$
$\mathbf{P}_{N1}$	(0.25, 1.60)	235	0.05
$\mathbf{P}_{N2}$	(0.50, 1.60)	132	0.15
$\mathbf{P}_{N3}$	(0.75, 1.60)	135	0.15
$\mathbf{P}_{W1}$	(0.25, 2.40)	373	0.1
$\mathbf{P}_{W2}$	(0.50, 2.40)	200	0.1
$\mathbf{P}_{W3}$	(0.75, 2.40)	236	0.2

[a] $P_{\mathrm{goal}}(\bar{N}_{\min})$ denotes the P_{goal} value of planning $\mathbf{P}|P_{\mathrm{goal}}$ with minimal average sampling number of times.

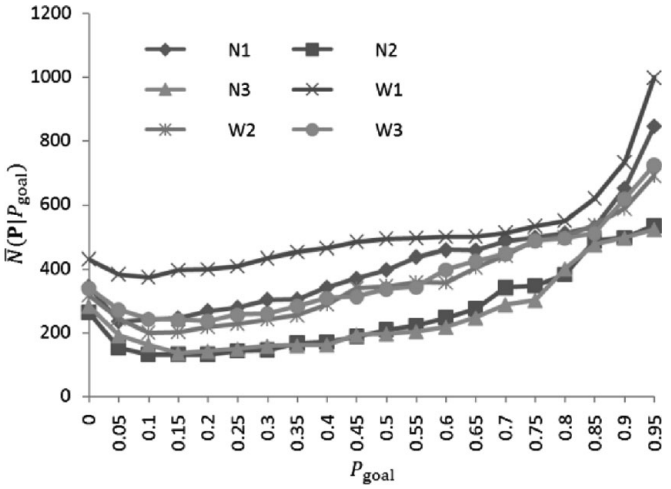

Fig. 8. The average sampling number of times continuously changes with P_{goal}.

3.2. Characters of Goal-Probability Threshold

3.2.1. $\bar{N}(\mathbf{P}|P_{\mathrm{goal}}) \sim P_{\mathrm{goal}}$ relationship. We experiment with six benchmarking problems in Table I, $\{\mathbf{P}_{N1}, \mathbf{P}_{N2}, \mathbf{P}_{N3}, \mathbf{P}_{W1}, \mathbf{P}_{W2}, \mathbf{P}_{W3}\}$, with continuously varying GPTs, $P_{\mathrm{goal}} \in \{0, 0.05, 0.1, 0.15, \ldots, 0.95\}$. Figure 8 shows how the average sampling number of times changes with P_{goal}, from which we can see the following: (a) $\bar{N}(\mathbf{P}|P_{\mathrm{goal}})$ changes smoothly and continuously with P_{goal}; (b) there is only one minimum of $\bar{N}(\mathbf{P}|P_{\mathrm{goal}})$ for each planning problem $\mathbf{P}$; (c) $\bar{N}(\mathbf{P}|P_{\mathrm{goal}})$ changes obviously with P_{goal}. The above phenomenon brings us to the conclusion that there is a P_{goal} value that makes the planner complete with a low average

[1] Independent parameters are preferred to describe benchmarking-planning problems because of the following two considerations: (a) The BP neural network will be employed that normally requires independent input parameters, and the parameter constraint relationship may lead to coupled effects. (b) The numerical planning experiments that study the characters of GPT and its relationship with the parameters of the benchmarking-planning problems also demand independent parameters to describe the benchmarking-planning problems.

[2] The third geometric constraint is given to avoid coupled effects due to both local minima and narrow passages. The narrow passage would occur should w approach 4 m.

https://doi.org/10.1017/S0263574709990804 Published online by Cambridge University Press

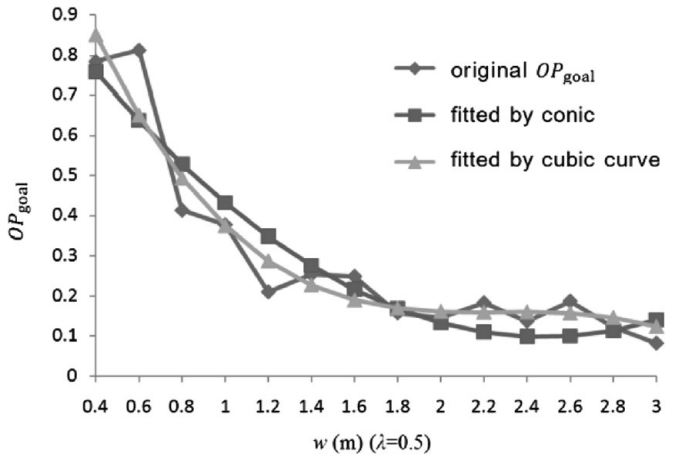

Fig. 9. OP_{goal} changes with w and fitting by conic and cubic curves. The fitting errors by conic and cubic curves reach 0.174 and 0.161 separately.

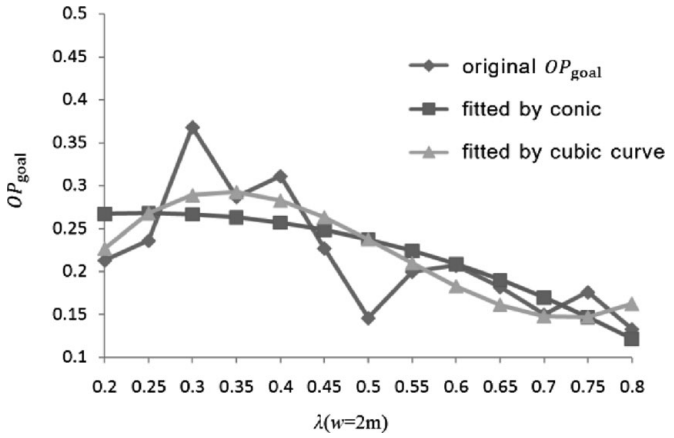

Fig. 10. OP_{goal} changes with λ and fitting by conic and cubic curves. The fitting errors by conic and cubic curves reach 1.101 and 0.079 separately.

sampling number of times, i.e., optimized P_{goal}, denoted as OP_{goal}.

3.2.2. $OP_{goal} \sim \mathbf{P}(\lambda, w)$ *relationship.* Based on the above-given conclusion, a critical issue is to obtain OP_{goal} for individual planning problem $\mathbf{P}(\lambda, w)$. We first study the relationship between OP_{goal} and (λ, w), the parameters describing the benchmarking planning problem. Note that actually it is impossible to know the exact value of OP_{goal}, since there is no analytical solution for how it should be computed. However, we can numerically compute a P_{goal} value with a limited error, which is termed as fitted OP_{goal}. In the following, we do not specifically discriminate fitted OP_{goal} and OP_{goal}. See ref. [16] for how to compute fitted OP_{goal} for individual problem $\mathbf{P}(\lambda, w)$.

Figures 9 and 10 show how OP_{goal} changes with (λ, w) and fitted curves of two sets of benchmarking planning problems, from which we can see that for a given planning problem $\mathbf{P}(\lambda, w)$, OP_{goal} changes continuously with the problem specifications.

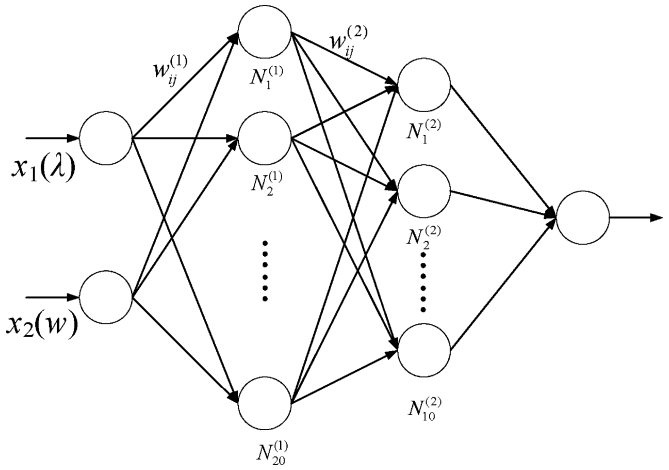

Fig. 11. The employed "2-20-10-1" BP neural network.

4. OP_{goal} Fitting by BP Neural Network

On the basis of the observations in Section 3.2, this section discusses the approach to self-adaptation of OP_{goal} for a planning problem ρ_2.

4.1. Problem description

To obtain OP_{goal} for each $\mathbf{P}(\lambda, w)$, a mapping is defined as

$$\mho: \quad \mathbf{P}(\lambda, w) \rightarrow OP_{goal}. \tag{5}$$

Since the analytical description of $\mho$ is not available, an alternative solution is to employ an approximating function.

Let $\tilde{\mho}$ be the mapping using an approximating function

$$\tilde{\mho}: \quad \mathbf{P}(\lambda, w) \rightarrow \widetilde{OP}_{goal}, \tag{6}$$

where $\widetilde{OP}_{goal}$ is the approximate OP_{goal}. For a planning problem $\mathbf{P}(\lambda, W)$, if the approximate error $\Delta OP = |\widetilde{OP}_{goal} - OP_{goal}|$ is within an acceptable error, $\tilde{\mho}$ can be considered a successful approximating mapping of $\mho$. With a $\tilde{\mho}$ modular employed in the footstep planner, $\widetilde{OP}_{goal}$ can be automatically generated regarding each individual planning problem. Note that for each type of planning problems, only one benchmarking model is needed for self-adaptation of the optimized GPT.

4.2. Fitting by BP neural network

However, the fitting errors by analytical functions are not acceptable (see Figs. 9 and 10), considering a fitting error of over 0.1 will result large fluctuations of average sampling number of times (see Fig. 8). A BP neural network is employed to approximate mapping $\tilde{\mho}$ (see Fig. 11 and Table II for specifications).

The BP neural network is trained by the Powell–Beale conjugate gradient method with a number of learning samples in $\mho$. The samples are obtained by orthogonally designed planning problems, such as

$$\mathbf{S}\,(\mathbf{P}(\lambda^{(i)}, w^{(j)}), OP_{goal}^{(i,j)}), \tag{7}$$

where $\lambda^{(i)} = 0.2 + 0.02i \in \{0.2, 0.25, \ldots, 0.8\} (i = 0 \sim 30)$ and $w^{(j)} = 0.4 + 0.1j \in \{0.4, 0.5, \ldots, 3\} (j = 0 \sim 26)$.

https://doi.org/10.1017/S0263574709990804 Published online by Cambridge University Press

Table II. BP neural network specifications[a].

Item	Descriptions
Input	$(x_1, x_2) = (\lambda, w), \lambda \in [0.2, 0.8], w \in [0.4, 3]$
Layer I	$s_i^{(1)} = \sum_{j=1}^{2} w_{ij}^{(1)} x_j + b_i^{(1)}, y_i^{(1)} = \tanh\left(s_i^{(1)}\right),$
Layer II	$s_i^{(2)} = \sum_{j=1}^{20} w_{ij}^{(2)} y_j^{(1)} + b_i^{(2)}, y_i^{(2)} = \tanh\left(s_i^{(2)}\right),$
Output	$s_i^{(3)} = \sum_{j=1}^{10} w_{ij}^{(3)} y_j^{(2)} + b_i^{(3)}, y^{(3)} = \left(1 + e^{-s^{(3)}}\right)^{-1}$ $\widetilde{OP}_{goal} = y^{(3)}$

[a] $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$; $w_{ij}^{(k)} (k = 1, 2, 3)$ denotes the weight of the jth input of ith neuron in layer k; $s_i^{(k)}$ denotes the linear summary of inputs of ith neuron in layer k; $y_i^{(k)}$ denotes the output of ith neuron in layer k.

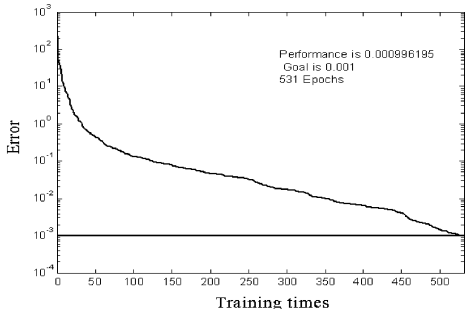

Fig. 12. The BP neural network converges after 531 trainings. The learning rate is 0.01, and the error limit is 0.001.

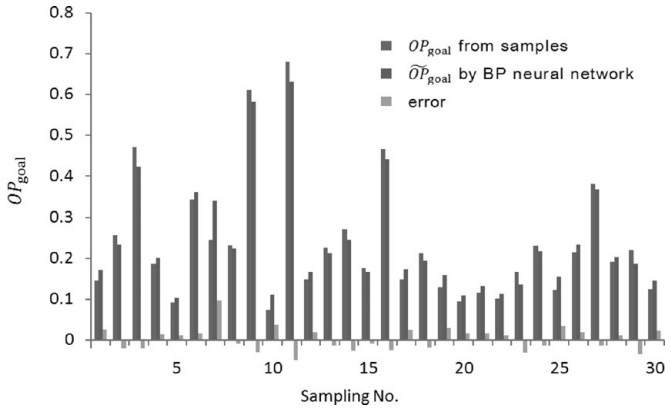

Fig. 13. Comparison between OP_{goal} from the samples and $\widetilde{OP}_{\text{goal}}$ generated by BP neural network.

With total $31 \times 27 = 837$ samples, the BP neural network converges. The training specifications are given in Fig. 12.

The trained BP neural network is used to generate $\widetilde{OP}_{\text{goal}}$ for 30 samples randomly extracted out of the 837 ones. Figure 13 compares OP_{goal} of the samples and $\widetilde{OP}_{\text{goal}}$ generated by BP neural network. The maximal error is 0.047 and the error deviation is 0.031, which are quite satisfying, considering that the average sampling number of times changes flatly while P_{goal} is in the neighborhood of OP_{goal} (see Fig. 8).

So far, for each planning problem $\mathbf{P}(\lambda, W)$, the BP neural network employed in the Multi-RRT-GoalBias footstep planner can generate the corresponding $\widetilde{OP}_{\text{goal}}$. And

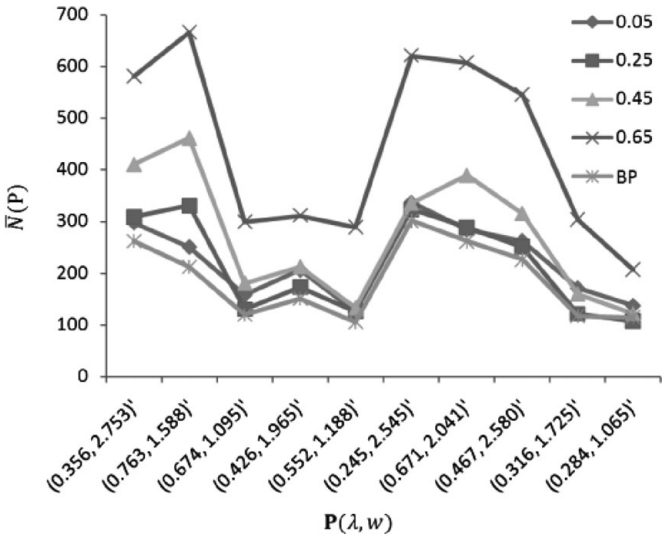

Fig. 14. Average sampling number of times with different values of P_{goal}.

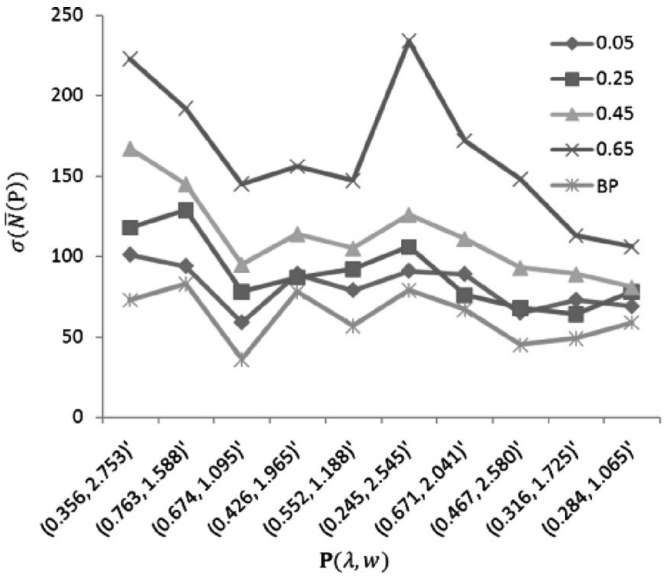

Fig. 15. Standard deviation of average sampling number of times with different values of P_{goal}.

with planning in dynamic environments, the BP module can generate real-time $\widetilde{OP}_{\text{goal}}$ according to the planning problem specifications. Note that for different types of planning problem, alternative benchmark models and the corresponding BP neural networks should be constructed.

5. Implementation

5.1. Verification of self-adaptation approach

In order to compare the planner with BP neural network and with manually given P_{goal}, we planned with Multi-RRT-GoalBias footstep planner for a set of randomly generated benchmarking problems by the computer. Considering the empirical observation that OP_{goal} is normally small except in the cases of planning in clear environments, four manual P_{goal} are set as $\{0.05, 0.25, 0.45, 0.65\}$. Figures 14 and 15 compare the average sampling numbers of times and their

https://doi.org/10.1017/S0263574709990804 Published online by Cambridge University Press

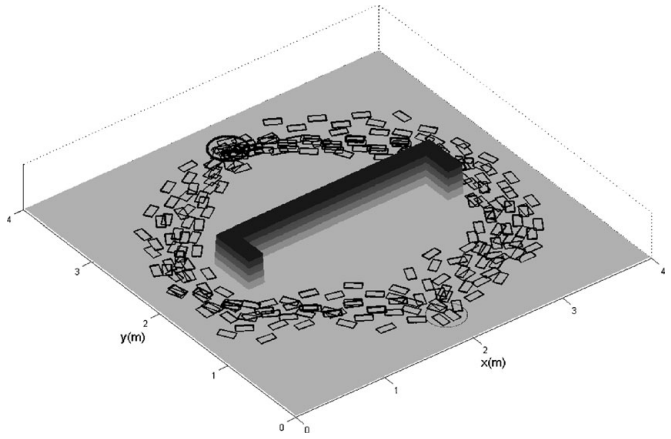

Fig. 16. Footstep sequences of 10 trials by the Multi-RRT-GoalBias footstep planner with the BP-generated P_{goal}.

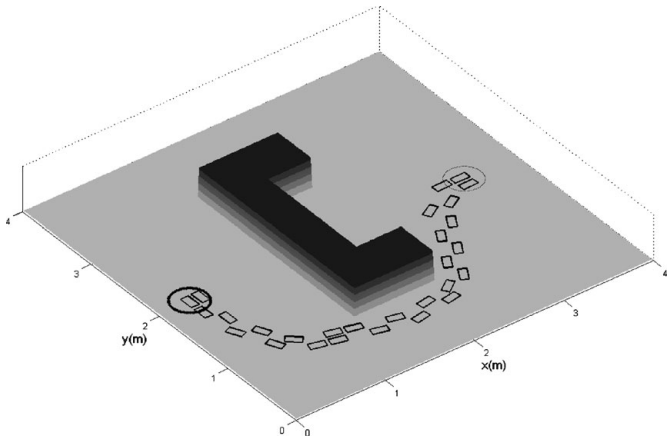

Fig. 17. A footstep sequence by the Multi-RRT-GoalBias footstep planner with the benchmark model discussed in this paper.

standard deviations of planning with BP-generated P_{goal} and manual P_{goal}. Comparison of average sampling numbers of times verified that planners with self-adapted P_{goal} by BP neural network can complete with less complexities.

Figure 14 shows that the distribution of average sampling numbers of times of planners with parameter self-adaptation is more concentrative than planners with manual P_{goal}, which shows that goal-biased randomized planning with optimized P_{goal} is much stable. Figure 16 shows the footstep placement sequences of 10 trials with the BP-generated P_{goal}.

5.2. Planning by Multi-RRT-GoalBias footstep planner

We implemented the Multi-RRT-GoalBias footstep planner for a planning problem reported in ref. [7]. Compared with the results by deterministic planners,[7] the Multi-RRT-GoalBias footstep planner can generate desirable footstep sequences within quite acceptable planning duration and sampling number of times (see Fig. 17 for one footstep placement sequence). The $\widetilde{OP}_{\text{goal}}$ generated by BP neural network is 0.173. Statistical data of 200 trials include average sampling number of times of 338, average searched nodes of 3387, and average planning duration of 0.332 s. Implementation of the Multi-RRT-GoalBias footstep planner in different planning problems relies on alternative parameterized benchmark models with the corresponding BP neural networks. Figure 18 shows one footstep sequence generated with a benchmark model describing a planning problem in a typical environment with narrow passages (see ref. [16] for more discussion).

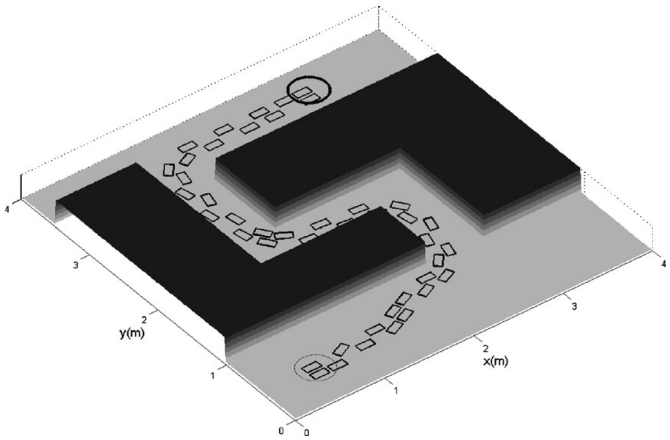

Fig. 18. A footstep sequence by the Multi-RRT-GoalBias footstep planner with a benchmark model describing planning problems in a typical environment with narrow passages.

6. Discussion and Conclusion

GPT is a key parameter controlling the convergence rate and affecting the feasibility of the goal-biased nonuniform sampling in the randomized footstep planner. Self-adaptation of GPT regarding the planning problem specifications is an important issue. In this paper, we proposed an approach to GPT self-adaptation by parameterizing of planning problems and presented how it was realized in the cases of footstep planning in environments with local minima. Three key steps of implementation of the proposed method include (a) construction of parameterized model of planning problems, (b) construction and training of BP neural network, and (c) integration of BP module in the planner. The comparison experiments verified the feasibility and performance of the proposed approach.

Besides the implementation in specific parameterized environments, such as environments with local minima stated in the current paper, we attempt to provide this method as a reference for implementation of planning in other environments. The critical issue of further implementations is to construct the parameterized model. We are in the process of using the proposed approach for biped navigation for humanoid robot soccer competition, and the related publication will follow.

References

1. J. Latombe, *Robot Motion Planning* (Kluwer Academic, Boston, 1991).
2. S. M. LaValle, "Sampling-Based Motion Planning," **In**: *Motion Planning Algorithms* (Cambridge University Press, 2006).
3. S. R. Lindemann and S. M. LaValle, "Current Issues in Sampling-Based Motion Planning," *Proceedings of the International Symposium on Robotics Research* (P. Dario and

https://doi.org/10.1017/S0263574709990804 Published online by Cambridge University Press

R. Chatila, eds.) (Springer-Verlag, Berlin Heidelberg, 2005) pp. 36–54.
4. J. J. Kuffner, K. Nishiwaki, S. Kagami *et al*., "Motion planning for humanoid robots," *Trans. Adv. Rob.* **15**, 365–374 (2005).
5. J. J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba and H. Inoue, " Online Footstep Planning for Humanoid Robots," *Proceedings IEEE International Conference Robotics and Automation*, Taipei (Sep. 2003) pp. 932–937.
6. J. J. Kuffner, K. Nishiwaki, K. Kagami *et al*., "Footstep Planning Among Obstacles for Biped Robots," *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii (Oct. 2001) pp. 500–505.
7. J. Chestnutt, J. J. Kuffner, K. Nishiwaki and S. Kagami, "Planning Biped Navigation Strategies in Complex Environments," *Proceedings of IEEE International Conference on Humanoid Robots*, Munich, Germany (2003). [CD-ROM]
8. J. Chestnutt and J. J. Kuffner, "A Tiered Planning Strategy for Biped Navigation," *Proceedings IEEE International Conference on Humanoid Robots*, California (Nov. 2004) pp. 422–436.
9. Y. Ayaz, K. Munawar, M. B. Malik, A. Konno and M. Uchiyama, "Human-like approach to footstep planning among obstacles for humanoid robots," *Int. J. Human. Rob.* **4**(1), 125–149 (2007).
10. Y. Ayaz, A. Konno, K. Munawar, T. Tsujita and M. Uchiyama, "Planning Footsteps in Obstacle Cluttered Environments," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2009)*, Singapore (Jul. 14–17, 2009) pp. 156–161.
11. P. Michel, J. Chestnutt, J. J. Kuffner *et al*., "Vision-Guided Humanoid Footstep Planning for Dynamic Environments," *Procceedings of the IEEE/RAS International Conference on Humanoid Robots*, Tsukuba, Japan (2005) pp. 13–18.
12. J. Chestnutt, Navigation Planning for Legged Robots *Ph.D. Thesis* CMU-RI-TR-56-23 (Pittsburgh, PA: Robotics Institute, Carnegie Mellon University, Nov. 2007).
13. J. Chestnutt, M. Lau, J. J. Kuffner *et al*., "Footstep Planning for the Honda ASIMO Humanoid," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Tsukuba, Japan (2005) pp. 629–634.
14. Z. Y. Xia and K. Chen, "Modeling and algorithm realization of footstep planning for humanoid robots," *Robot* **30**(3), 231–237 (2008).
15. Z. Xia, G. Chen, J. Xiong, Q. Zhao and K. Chen, "A Random Sampling Based Approach to Goal-Directed Footstep Planning for Humanoid Robots," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2009)*, Singapore (Jul. 14–17, 2009) pp. 168–173.
16. Z. Y. Xia, Sampling-Based Footstep Planning for Humanoid Robots *Ph.D. Thesis* (Beijing: Tsinghua University, 2008).
17. S. M. LaValle and J. J. Kuffner, "Rapidly-Exploring Random Trees: Progress and Prospects," *Workshop on the Algorithmic Foundations of Robotics* (B. R. Donald, K. M. Lynch and D. Rus, eds.) (AK Peters, Wellesley, MA, 2001) pp. 293–308.

https://doi.org/10.1017/S0263574709990804 Published online by Cambridge University Press