REINFORCEMENT LEARNING IN CONTROL SYSTEMS FOR WALKING HEXAPOD ROBOTS

HRDLICKA IVO, KUTILEK PATRIK

University of Defense, Department of Technical Cybernetics and Military Robotics.

Kounicova 65 Brno.

Tel. +420 973 44 25 11. E-mail: kutilek.patrik@seznam.cz

Abstract : We have developed a controller for a six-legged robot that allows it to walk on rough terrain or where mines have been detected. Intention of this article is to describe application of reinforcement learning methods for implementation of an acyclic walking. Cyclic methods of a walking are not feasible in difficult terrain because doesn't consider subsequent undercarriage states of the robot in future. For basic (not so rough) terrain we used new designed heuristic algorithm. In complex environment, computational more intensive approach (i.e. Q-learning) is applied. Overall control algorithm of the walking undercarriage combines these two presented techniques. Results of the new algorithm of control to the problem of learning to walk with a six legged robot are presented by demining robot.

Keywords : Q-learning, acyclic walking, hexapod, cognitive, robot.

1. Moving of six legged walking undercarriage

Six legged walking undercarriage moving in discontinuous terrain is a result of coordinated motion of its effectors. For control of robot in such complex terrain like a battlefield we have to consider only method of locomotion by acyclic walking. In acyclic walking each leg or a group is autonomous in its control. Therefore it is necessary to use overall motion control. The discussed design of the solution applies cognitive approach.

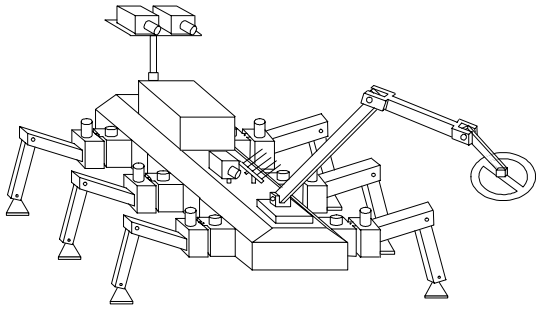
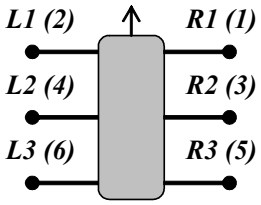


Fig.1 : Concept of the demining walking robot was designed by Department of Technical Cybernetics and Military Robotics in Brno.

1.1 Stability of walking undercarriage

Primary condition [1], [2] for proper control of the walking is undercarriage stability. To maintain stability of undercarriage during the walking we defined states of the steady-state stability [3], [4], [5], figure 2.



T	two legs motion					
3	0	0	1	1	0	0
4	0	1	1	0	0	0
5	0	0	0	1	1	0
6	1	0	0	1	0	0
7	0	0	1	0	0	1
8	1	0	0	0	0	1
9	0	1	0	0	1	0

T	one leg in motion					
10	1	0	0	0	0	0
11	0	1	0	0	0	0
12	0	0	1	0	0	0
13	0	0	0	1	0	0
14	0	0	0	0	1	0
15	0	0	0	0	0	1

	R1	L1	R2	L2	R3	L3
T	three legs motion					
1	1	0	0	1	1	0
2	0	1	1	0	0	1

Fig.2 : Available combinations to maintain stability of body during walking.

We can define the vector $t \in T$ which denotes action of undercarriage

$$t = (l_{R1}, l_{L1}, l_{R2}, l_{L2}, l_{R3}, l_{L3}), \quad (1)$$

where binary (i.e. true/false) variables in vector $l_{R1}, \ldots, l_{L3}$ indicate requests for moving of legs. If $t \in T$ is *1*, it means that robot wants to move the legs. In other case, zero represents stance phase of the legs.

1.2 Working space for legs movements

Motions of legs are possible only in construction workspaces. Potential footprint is allowed only in the leg's workspace (LW), Fig.3.

The value d_{kSmax} presents maximal side size of one steps of leg. If walking of undercarriage is acyclical and motion of every leg is autonomous then we can describe position of legs by value n_{Xi}. The value n_{Xi} is requirement for leg motion in LW [1], [2], [3]. Maximal front position in LW matches the value *0* requirement for leg motion. Value *1* represents maximal requirement for leg motion of the leg in back position, Fig.2. Vector $r \in R$ represents overall requirements for leg motion.

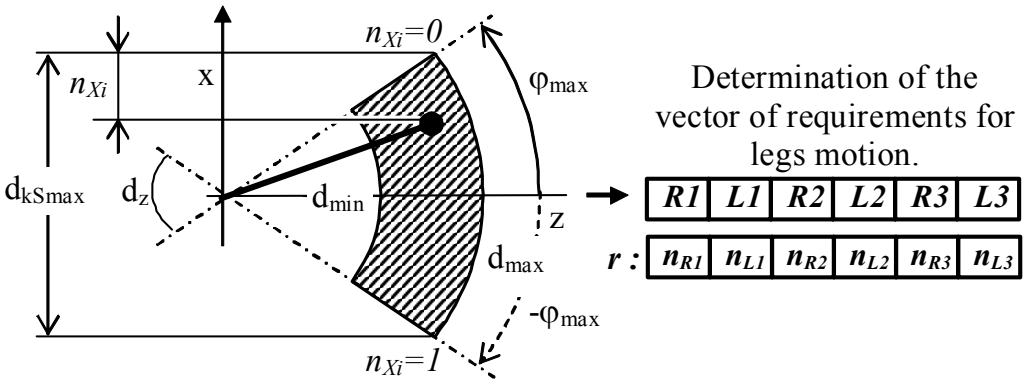


Fig.3 : Possible area of footprint in cutting plane $y = -z_s$.

1.3 Selection of footprints

Task is to select optimal destination for next step of the leg in its working space. The leg will has to serve as a buttress for the body and guarantees stability in-process of shift to forward. Control system localizes possible destinations of footprints and determines relevant step sizes for each leg from its actual and expected locations. The selected legs have to remain at the exact location and have to support undercarriage stability during the following movements [1], [2].

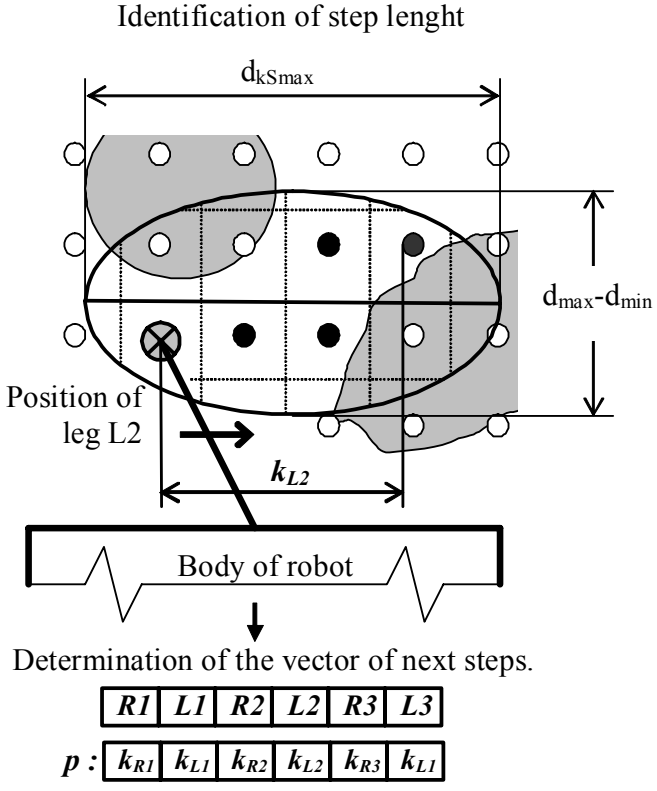


Fig.4 : Example identification of steps for acyclical locomotion.

Vector $p \in P$ represents information about maximal next step size k_{Xi} of the undercarriage legs, which is necessary and sufficient condition for proceeding in acyclic walking. It expresses fundamental dimensional feature for each leg of the walking undercarriage in actual LW.

2. Heuristic control system using Q-learning methods

Control system of the walking undercarriage uses new designed technique. This approach combines several stand-alone methods to solve problems of acyclic walking. First a straightforward method will be used in less complex terrain. Selection of subsequent action is based on maximization of actual rewards that represents evaluation of state transitions of the undercarriage. Another method will be applied in a complex terrain. It is reinforcement learning method based on maximization of rewards of estimated subsequent states of robot model. Block diagram is on figure 5.

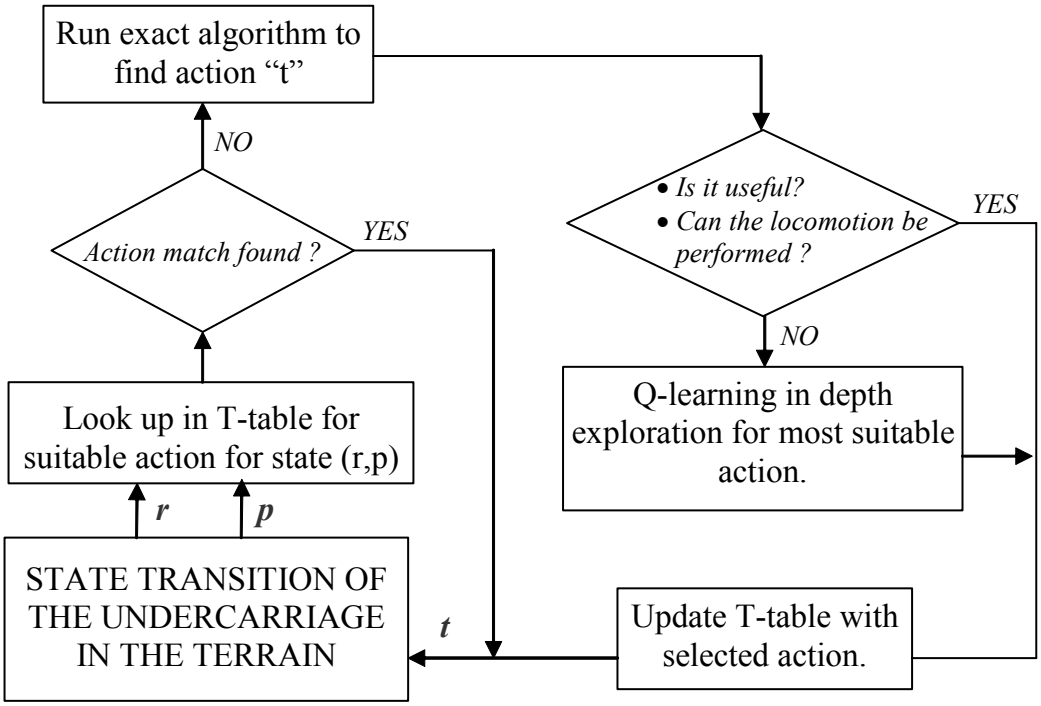


Fig.5 : Block diagram of the composed controller.

To achieve control improvements it is essential to define actual reward *c* that represents feedback response to taken action $t \in T$ after each step. Maximal distance of forward movement l of the undercarriage in desired direction is assumed as suitable option. ***Exact algorithm*** evaluates all fifteen possible actions that preserve undercarriage stability. The highest rated action (i.e. with maximal reward for

intended step) is selected after the evaluation. Relevant position adjustments of the legs are propagated to executive mechanical unit of the undercarriage.

2.1 Knowledge representation and inference

As action decision it is based on actual state of an environment (i.e. surrounding terrain) and the legs, knowledge is represented in one production table of states and relevant actions. It provides us easy way for decision process. It is possible to reuse information from the table about states and relevant actions that were evaluated already at least once by the exact heuristic algorithm. This method sufficiently satisfies our needs for fast and easy inference of appropriate action without additional intensive computing by heuristic algorithm.

Production table is consisted of action-state T-map as follows figure 6.

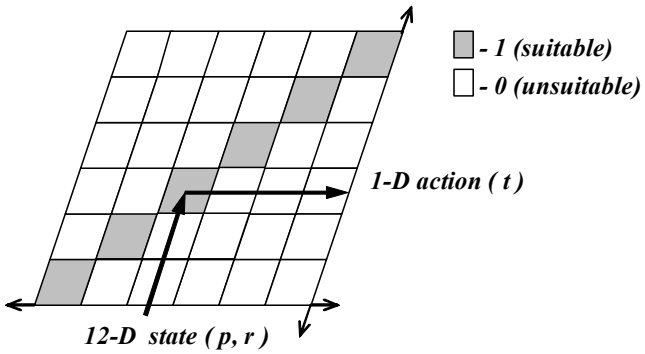


Fig.6 : Production T-table.

Production system consists of knowledge base and base of rules. Production rule syntax is :

IF $r \in R$ *AND* $p \in P$ *THEN* $t \in T$,

where *R* and *P* denote state sets a *T* represents action set.

R and *P* are sets of states represented by *12-D* space. State of each leg denotes variables $n_a \in N_i$ and $k_b \in K_j$ that represents coordinates in two dimensional space. Action set $t \in T$ form additional dimension of the production table map (T-map). 13-D vector space (T-map) is able to hold information of mutual relations between states *p*, *r* and proper actions of the undercarriage.

1. *Initialize 13-D table T(p,r,t)*
2. *Observe the current state of terrain ($p \in P$) and state of undercarriage ($r \in R$)*
3. *Look up in T-map for action t*
4. *If action $t \in T$ already exists in T-map*
5. *Execute the action t*
6. *Else*
7. *Determine the action t by exact heuristic algorithm*
8. *Store result in the T-map : T(p,r,t) = T(p,r,t) + 1*
9. *Execute undercarriage action $t \in T$*
10. *Repeat 2*
11. *Finish locomotion if destination in terrain is reached*

Alg.1 : The algorithm of the production system.

2.2 Walking on rough terrain

Common Q-learning requires discreet Q-function, usually represented by Q-table (i.e. Q-values table). Values of discreet states and actions are used as indexes in the Q-table.

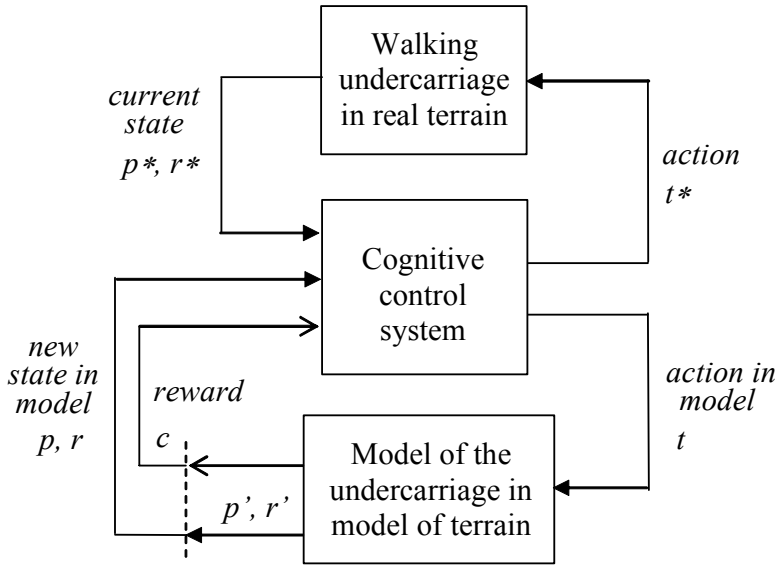


Fig.7 : Cognitive controller utilizes Q-learning and model of the environment and the undercarriage.

Intended cognitive system will approach the problem by using off-line reinforcement learning evaluating modeled environment terrain and using multiple Q-value updates for actions *t* in each state *p** a *r**. For safe locomotion in the real environment it will be evaluated with the modeled system first. Command variables controlling the movement of the real undercarriage will correspond to the taken action of the adapted system. Q-learning process in the cognitive system can be diagrammed as in figure 7.

Q-learning with discrete Q-function is described by fundamental relation [6], [8], [9] :

$$Q(p,r,t) = Q(p,r,t) + \alpha (c + \gamma \max_t Q(p', r',t) - Q(p,r,t)), \quad (2)$$

Formula describes an iteration step of Q-table update [6], [9].

1. ***Initialize 13-D table Q(p,r,t).***
2. ***Set the terms : learning rate α_0, discount rate γ and parameters of exploration rule (L_0,L_{min}, β)***
3. ***Observe the current state of terrain ($p \in P$) and current state of undercarriage ($r \in R$)***
4. ***Update L of Boltzmann exploration rule***
5. ***Determine the action $t \in T$ by exploration rule.***
6. ***Execute undercarriage action t***
7. ***Observe the new state of terrain ($p' \in P$) and new state of undercarriage ($r' \in R$)***
8. ***Observe reward c***
9. ***Update learning rate α***
10. ***$Q(p,r,t) = Q(p,r,t) + \alpha (c + \gamma \max_t Q(p', r',t) - Q(p,r,t))$***
11. ***Repeat 3***
12. ***Finish locomotion if destination in terrain is reached***

Alg.2 : The Q-learning algorithm.

State-action pairs *(p,r,t)* are evaluated in each iteration. State transitions are chosen on the basis of the state-action pair values. *maxtQ(p',r',t)* is the maximum Q-value from all possible actions *t* for actual state. *c* value represents

reward/punishment for the selected transition from state *(p,r)* to *(p',r')* by taking action *t*.

Algorithm 2 modification extends capability with stochastic learning strategy of Q-learning [8]. It performs multiple explorations on presented terrain model in each step.

1. ***Initialize 13-D table Q(p,r,t).***
2. ***Set the terms : learning rate α_0, discount rate γ and parameters of exploration rule (ε_0, n_0)***
3. ***Observe the current state of terrain ($p \in P$) and current state of undercarriage ($r \in R$)***
4. ***Update the parameter ε of exploration rule (i.e. ε-greedy)***
5. ***Determine the action $t \in T$ by exploration rule***
6. ***Execute undercarriage action t in model of terrain***
7. ***Observe the new state of terrain ($p' \in P$) and new state of undercarriage ($r' \in R$) in mode of terrain***
8. ***Observe reward c.***
9. ***Update learning rate α***
10. ***$Q(p,r,t) = Q(p,r,t) + \alpha (c + \gamma \max_t Q(p', r',t) - Q(p,r,t))$***
11. ***Repeat 5 by the number of exploration***
12. ***Execute undercarriage action $t \in T$ in model of terrain***
13. ***Repeat 3***
14. ***Finish locomotion if destination in model of terrain is reached***

Alg.3 : The Q-learning algorithm with stochastic strategy.

Exploration method performs multiple state transitions to states $p' \in P$ a $r' \in R$ on modeled terrain. Number of iterations is usually chosen on empirical basis. Increasing number of exploration iterations within each state shorten duration of adaptation to a certain extent. It increases the duration after due to arising redundancy. Number of iterations can be as many as it is number of possible actions *t*. In that case all fifteen actions are explored successively. It is completely stochastic strategy [8], [9].

There is a relation between number of iteration and vastness of the operating terrain. Parameters of the sensors (i.e. accuracy of the environment model, computational and time intensiveness of Q-learning exploration) limit the number.

2.3 Q-learning parameters determination

Q-learning has to satisfy some conditions to ensure convergence of the method. Essential precondition to assure convergence is decreasing learning rate $\alpha(p,r,t)$ [6], [9].

$$\alpha(p,r,t) = \frac{\alpha_0 n_0}{n_0 + n_\alpha(p,r,t)}. \tag{3}$$

$n_\alpha(p,r,t)$ is number of Q-table value updates realized until actual iteration. α_0 is initial value of the learning rate and n_0 controls decreasing rate of $\alpha(p,r,t)$.

Definition of reward *c* is another important precondition of Q-learning effective applicability. Movement distance l of the undercarriage in desired direction after taking action *t* appears as the right value that will determine intensity of the reinforcement.

$$c_{p,r}(t) \approx l^n, \tag{4}$$

where $0 \le l \le l_{\max}$.

Two methods are suitable for exploration and selection of appropriate action as a response on the undercarriage and environment state. They are ε-greedy [9], and Boltzman‘s exploration [6], [8].

ε-greedy method selects actions $t_a \in T(p_a,r_a)$ on probabilistic basis. Preselected ε value denotes likelihood of exploration. In practice use it is chosen to ensure high probability of selecting optimal action t_a according to strategy $\mu(p_a,r_a)$ that follows :

$$\mu(p_a,r_a) = \underset{t \in T(p_a,r_a)}{arg\ max} Q(p_a,r_a,t) \tag{5}$$

Second method utilizes Boltzman’s distribution for action selection. The method assigns to each action $t_a \in T(p_a,r_a)$ in the state specific preference for selecting it :

$$P(t) = \frac{e^{\frac{Q(p_a, r_a, t)}{L}}}{\sum_{t \in T(p_a, r_a)} e^{\frac{Q(p_a, r_a, t)}{L}}}, \qquad (6)$$

where *L* denotes parameter determining equiprobability of preference for each action in the givens state p_a a r_a. Drawback is insufficient exploration of unexplored terrain state space due to weak or none evaluation of actions leading to the in depth subsequent states. Therefore it is suitable for on-line Q-learning (Alg. 2), i.e. trial and error method of learning in the real-time. An application of ε-greedy method is suggested to satisfy desirable depth of explored state space if Alg.3 is executed (i.e. complete stochastic strategy). It would increase probability of low or/and null Q-function value actions for selection.

3. Experimental results of proposed robot's locomotion control method

Proposed methods performing locomotion were tested with modeled terrain in MATLAB. Illustrative example of used terrain is on Fig. 8.

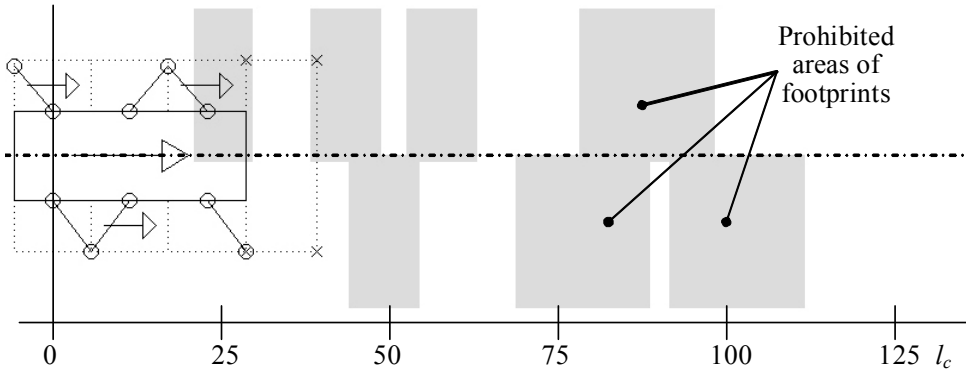


Fig.8 : Example of simplified model of robot in experimental terrain for system testing.

Obtained result indicates how far the robot is able to traverse over the modeled terrain (on Fig. 8). Fig. 9 demonstrates how experience from locomotion over similar terrain in previous cycles increases the reached distance.

Vertical axis of the graph denotes distance achieved during the selected eleven step cycles (i.e. eleven iterations - evaluations of eleven subsequent state-action pairs). Expected distance is limited due to construction dimensions of the

walking undercarriage. Horizontal axis represents number of passes through the operating terrain. Fig.9 illustrates development of the regression growing trend. Fig. 10 demonstrates undercarriage adaptation to pass through the modeled terrain within 3000 cycles.

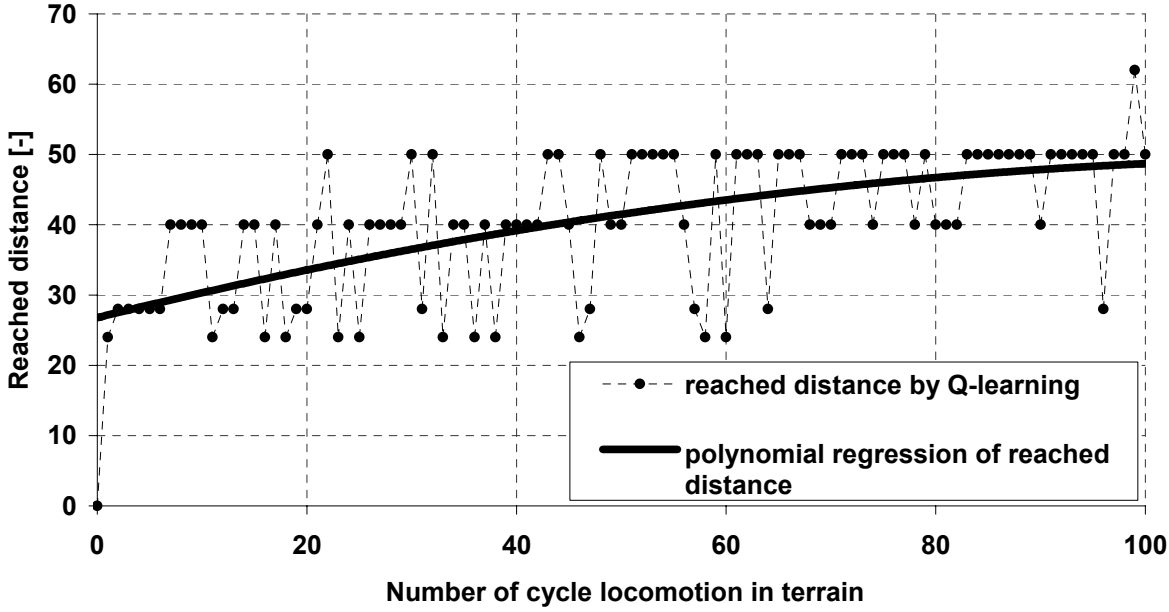


Fig.9 : Performance improvements during adaptation.

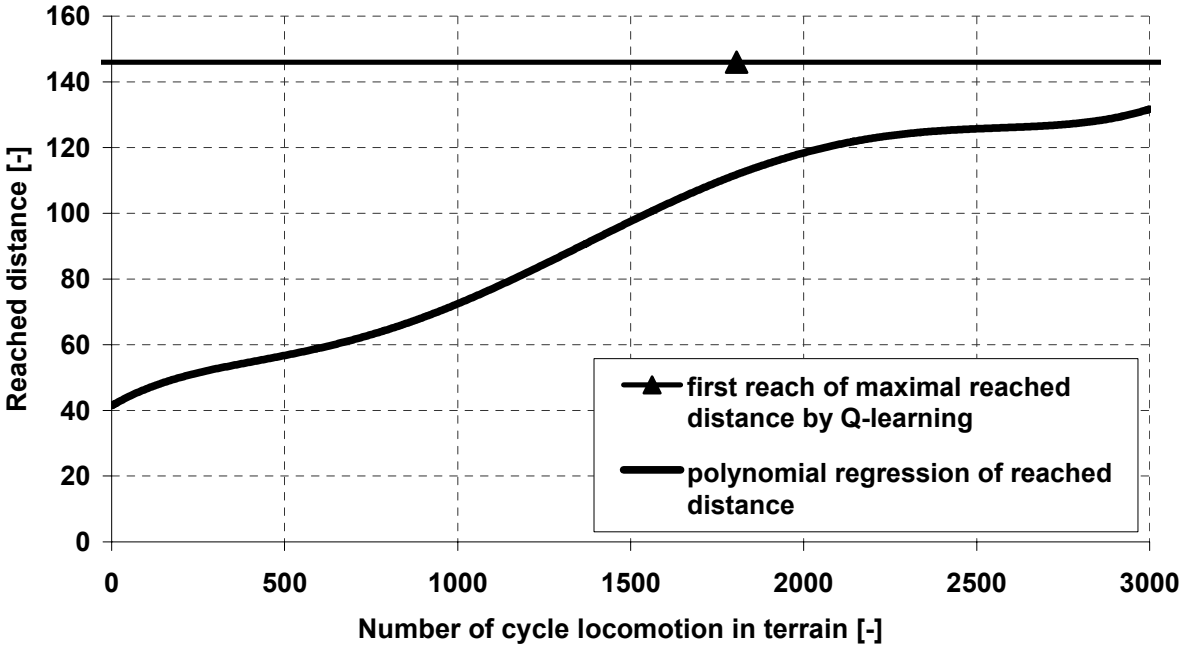


Fig.10 : Distance reached over discontinuous terrain by on-line Q-learning.

Graph illustrates that the outermost accessible distance was reached and QL system can be considered as adapted for the new terrain. Implementation of multiple exploration extended capability Q-learning (i.e. Alg 3) provides assumption of better adaptation performance. Result of the experiment is on Fig.11.

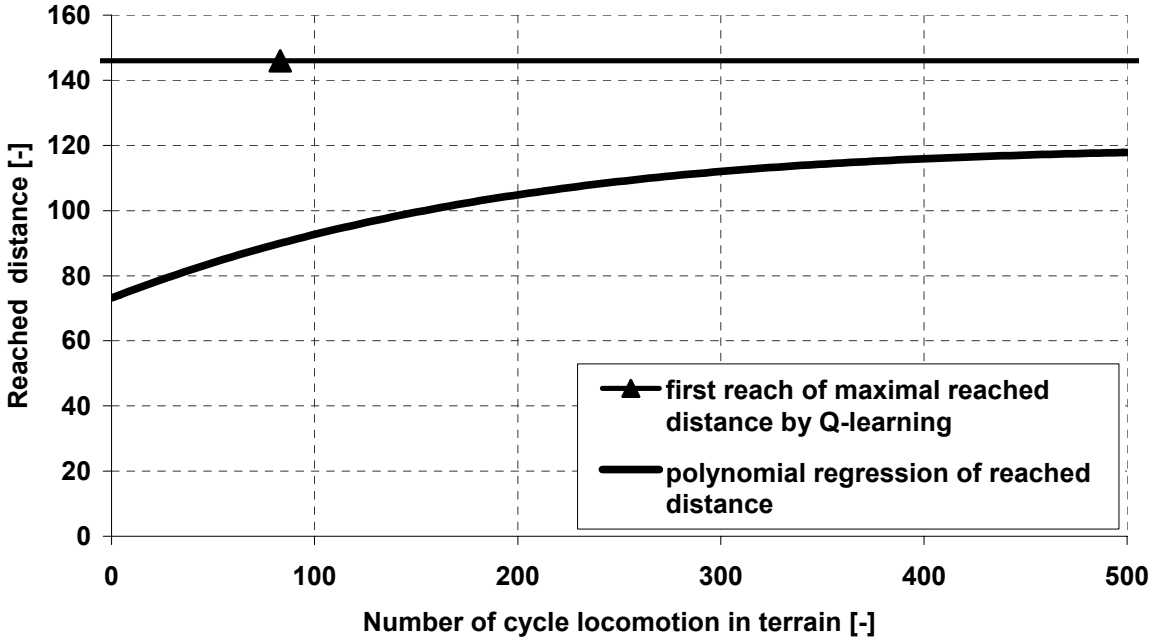


Fig.11 : Distance reached over discontinuous terrain by Q-learning uses completely stochastic strategy.

Increasing number of Q-learning iterations positively affects adaptation rate of walking in operating terrain. Higher achieved performance indicates better suitability of Q-learning, due to its ability of reflection expected subsequent rewards.

4. Conclusion

Presented technique synthesizes methods based on actual rewards and methods solving Markov Decision Processes and thus provides the best of both approaches. Computational intensive (i.e. time intensive) determination of appropriate actions based on Q-learning is used only in a very complicated terrain. In case robot operates in basic terrain it applies the quicker exact heuristic algorithm but no always the algorithm defines the best action of undercarriage.

Designed method of six legged robot control can be utilized in the military area, where are high safety requirements. Nevertheless more general applicability for similar technical topics is available as well.

References :

[1] ŘEŘUCHA, V. *Inteligentní řízení kráčení robota, docentská habilitační práce*. Brno: Vojenská akademie, 1997.

[2] KUTÍLEK, P.; DUFKOVÁ, A. *The Control of the Walking Undercarriage, Proceedings of The 6th International Scientific - Technical Conference Process Control 2004,* Kouty nad Děsnou: Univerzita Pardubice, 2004.

[3] KUTÍLEK, P., KACER, J. *Řízení lokomoce kráčejícího podvozku s necyklickou chůzí, Mezinárodná konferencia SSKI "Kybernetika a informatika",* Dolný Kubín: SAV-STU Bratislava, 2005.

[4] KUTÍLEK, P.; KACER, J. *The Locomotion Control of the Concyclically Walking Carriage, International Internet Journal - Cybernetics Letter*, Brno: Univerzita obrany, 2005.
http://stefek.cz/cyletter/pdf/TheLocomotionControlOfTheConcyclicalWalkingRobot.pdf (2005)

[5] PORTA, J.; CELAYA, E. *Force-based control of six-legged robot on abrupt terrain using the subsumption architecture*. Barcelona: Institut de Cibernetica, 2000.

[6] WATKINS, J. *Learning from delayed rewards, Ph.D. Thesis.* Cambridge: Cambridge University, 1989.

[7] BERGHEN, F. *A tutorial on q-learning algorithms. Technical report.* Bruxelles: Institut de Reacherches Interdisciplinaires et de Developpements en Intelligence Artificielle, 2003.

[8] BŘEZINA, T. *Efektivní metoda Q-učení: simulační posouzení použitelnosti pro řízení aktivního magnetického ložiska, docentská habilitační práce*. Brno: VUTIUM, 2003.

[9] GASKETT, C. *Q-Learning for Robot Control, Ph.D. Thesis.* Canberra: Australian National University, 2002.