International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

Development of Reinforcement Control Algorithm of Lower Body of Autonomous Humanoid Robot

Deepak Bharadwaj, Manish Prateek, Rashmi Sharma

***Abstract*: This paper presents the reinforcement control algorithm for an autonomous humanoid robot. The proposed humanoid robot is sensing the present state and switching to the goal state without knowing the kinematics and dynamics of the body. Simulation is carried out on the MATLAB platform and Multibody dynamics toolbox to verify the algorithm. Optimal policy is decided based on the result to feed this value in model predative controller.**

***Index Terms*: Reinforcement algorithm, state, reward, Q-learning.**

I. INTRODUCTION

In present scenario research community is trying to developed the self decision capable robot for the dynamic condition. In pre-programmed robots, manipulator is doing the task in known environment. But,with the existence of dynamic condition, due to lack of decision making capabilites,these types of manipulator fail to do the task. In such cases the preprogrammed robot does not know what to do. The rienforcement algorithm helps the robot to perform the required task. Reinforcement algorithm[1] deals with the present state of the robot. It is sensing the present condition of the robot and taking the robot to next state without knowing the mechanics and dynamics of the system. So it is model free based control. Reinforcement control algorithm is fully independent from mechancics and dynamics of the body. But while choosing the action it is partially dependent on the dynamic values. While taking the action to move to the next state, torque on the joints is varying . At that time joint motor is capable to handle these torque.

There are some singularity points while taking the step towards the next state. Singularity is the point where controller does not know what to do. So it needs to bypass that point, otherwise controller will not be able to send any information to the manipulator to perform the next task. Reinforceemnt control algorithm based on the value function[2], transition probabilities and the cost function reward. Reward function is the combination of positive and negative values. In present work, when the manipulator is reaching to the next step successfully, positive value of reward is awarded. Negative reward is not given to the action ,becuause when the humanoid is falling on the ground to reach the goal position, hunamoid should bring the feet either in forwrad direction or negative direction . If negative reward is awarded to the state and action , next time the feet is not doing the action to bring the feet in reverse direction. Present work focuses on the stability of lower body of humanoid robot. When some slipping condition happens due to change in the environment, at that time robot is not capable to come back to the initial state i.e standing position of the humanoid robot. Reinforcemet control algorithm helps to bring the robot in the standing condition without any preprogram of the robot.

II. THEORETICAL MODEL

A theoretical model built in the MATLAB Multibody dynamics toolbox considering the biomechanics parameter of human as shown in figure1. Theoretical MATLAB model is created with the help of body elements, joint and transform blocks. Kinematics and dynamics of the lower body of humanoid robot[3] is obtained for the maximum value of joint movement [4] of hip, knee and ankle joint considering all the dynamics condition. When the leg is taking any action in forward and reverse direction, at that time joint motor is capable to give these values.

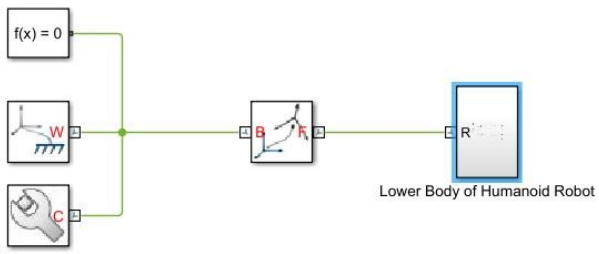

Figure1: MATLAB Model

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author
Deepak Bharadwaj*,Assistant Professor,School Of Engineering, Department Of Mechanical Engineering, University of Petroleum & Energy Studies Dehradun, India
Dr. Manish Prateek, Professor, School Of Computer Science, Engineering Department Of Mechanical Engineering, University of Petroleum & Energy Studies Dehradun, India
Rashmi Sharma, Research Scholar, School Of Computer Science, Engineering Department Of Mechanical Engineering, University of Petroleum & Energy Studies Dehradun, India

III. REINFORCEMENT LEARNING (RL) MODEL

Kinematic and dynamic analysis of the system gives real time information to the controller of the humanoid robot.
If case of any singular position in between the joint movement, the joint trajectory control bypasses that point.
In reinforcement control algorithm, while choosing the action values, the maximum and minimum values of acceleration of the joint are considered[5][6][7]. If the magnitude of action is very small or very large, it takes too much time to come to the next state. Reinforcement learning is a way of programing agents by reward and punishment without needing to specify how the task to be achieved. In reinforcement learning the humanoid robot is sensing the state with the help of gyroscope sensor. Gyroscope gives the real time orientation and current position of the joint to the controller. After observing the current state, the reinforcement controller chooses the appropriate action. After choosing the action the leg joint moves to next state. This iteration happens continuously till the goal point is obtained by the humanoid robot. Figure 2 shows the basic RL model of the lower body of autonomous humanoid robot. This paper considers the fall in the forward direction and reverse direction, if the environment is continuously changing. Currently, reinforcement algorithm is developed for the hip, knee and ankle of humanoid robot of each leg .

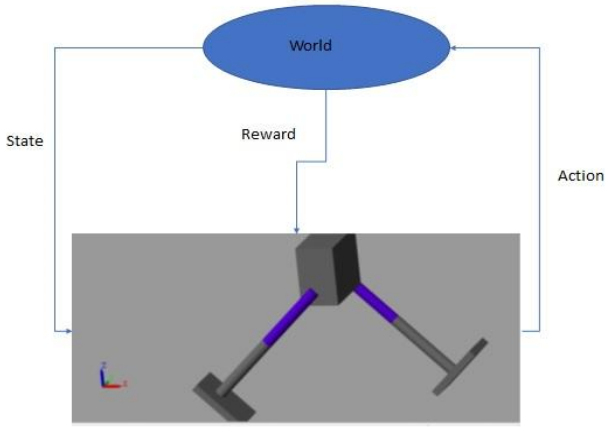

Figure2: Basic RL Model

The first step involved in RL model is observing the state of humanoid robot [8][9][10]. This information is sensed with the help of gyroscope sensor. It gives the actual orientation of the hip, knee and ankle joint. Second step involved in RL is decides an action. DC servomotor is used to actuate the joint. DC motor can move with full speed in the forward and reverse direction. Speed of the DC motor can be controlled with the help of pulse width modulation (PWM) technique. Since the robot is not preprogrammed, the action depends upon the learning rate, epsilon and some randomness values. These parameters will be covered in the next section of this paper. Third step of the RL algorithm involves performing the action. Time taken to perform the action depends upon the motor speed. Since, the reinforcement algorithm is valid for discrete state between present state to goal state, time taken to reach these states depend on the capabilities of the processor. As there is no mathematical equation involved to calculate the time, it is independent property, hence the user cannot control the humanoid robot to reach the goal point on required time. Time is totally dependent on the randomness and probabilities to reach to next step. Fourth step of reinforcement learning includes observing the new state by performing the action. Fifth step in reinforcement learning is observing the reward. After performing the action and observing the new state, a reward is provided. If the humanoid robot successfully obtains the next state, a positive reward is assigned to the action and the state. If it not achieved, then a negative reward is provided. But in the present work, no negative reward is assigned to the state and action. Because, if negative reward is assigned to the state and action, then next time the reinforcement controller will not perform that task. Robot must move the feet either in forward direction or reverse direction to come back to the standup position in case it falls. In sixth step humanoid robot is learning from the past experiences. If the goal point is achieved by the humanoid robot, then the reinforcement controller finds that the error is zero and stops. If it does not achieve the goal it repeats the sequence from step two till it achieves the goal point.

IV. REINFORCEMENT LEARNING ALGORITHM

At present, the present state and goal state of the humanoid robot and action of the joint is known. But, information about system model that is transition function [11] and reward function is unknown. Here it needs to perform actions to generate new experiences. Based on the direct experience of the world, humanoid robot wants to learn. There are four main algorithms of learning such as certainly equivalence, temporal difference, Q-learning and SARSA (state-action-reward -sate-action). This paper adopts the Q-learning approach to learn.

A. Q-learning

Q-learning [12][13] iteratively approximates the state-action value function. Q.Q-learning is an off -policy method. In Q-learning keep an estimate of Q(s,a). Where, state s and a is the action in a table. After gathering more experiences and updating these estimates. Estimates do not depend on the exploration policy. In Q-learning, learning of value function and policy will be done simultaneously. In present condition reinforcement algorithm is developed for the hip joint and same concept is applied for other two remaining joint of knee and hip. The algorithm is developed on the MATLAB platform, afterwards MATLAB converts these algorithms in C, C++ and python programming languages. Then these codes can be easily implemented on embedded programming. The body of Q-learning algorithm is as follows;

1.Initialize Q(s,a) to small random values
2.Observe state ,s
3.Pick an action ,a, do it
4.Observe next state, s', and reward ,r
5. $Q(s,a) \longleftarrow (1-\alpha)\ Q(s,a)+\alpha(r+\gamma \max_{a'} Q(s',a'))$

B. Implementation of Q-learning algorithm on MATALB

While developing the MATLAB code α is taken as epsilon and γ is taken as learning rate, s is taken as state and a is taken as actions and lastly reward r is taken as R.

```
learnRate=.90
epsilon=.6  initial values
epsilonDecay=.8// need to decay the epsilon
discount=.9
successRate=1;
winBonus =100// if the goal point is achieved
startPt=[-40]// this information is coming from the gyroscope sensor
goalPt=[0]// humanoid hip joint to reach the goal point  and it can varies depends on the user
maxEpi=50000;
action=1° // This values can be obtained from the smooth joint trajectory
actions=[0 , action]
x=linspace(startPt,goalPt,10)// controller crates the  eight intermediate step between the startPt and goalPt
states=zeros(length(x),1)
index=1;
for j=1:length(x)
states(index,1)=x(j)
index=index+1;
end
R=states*.1// reward is taken as linear here , it may be non linear //It depend on the probiblitis oh happening the state.
Q=repmat(R,[1,3])// Assigning the values of Q value for state and action
z1=startPt
For episodes=1:maxEpi
[~,SIdx]=min(sum(states-repmat(z1,[size(states,1),1])).^2,2)
)
if(rand()epsilon!!episodes=maxEpi)&&rand()<=successRate
//pick according to the Qmatrix
[~,aIdx]=max(Q(sIdx,:) //best action
else
aIdx=randi(length(actions),1)
end
T=actions(aIdx)
z2=z1+T
z1=z2
if(z2=goalPt)
success=true
bonus=winBonus
[~,snewIdx]=min(sum(states-repmat(z1),[size(states,1),1])).
^2,2))
Q(sIdx,aIdx)=Q(sIdx,aIdx)+learnRate*(R(snewIdx)+discou
nt*max(Q(snewIdx,:) )-Q(sIdx,aIdx)+bonus)
break
else
bonus=0
success=false
end
```

V. REINFORCEMENT CONTROLLER

A reinforcement controller is implemented in MATLAB platform as shown in figure3.

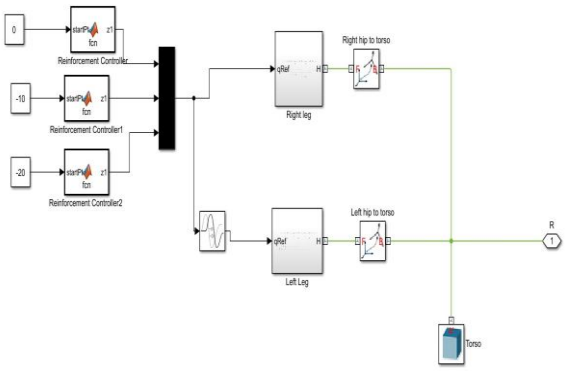

Figure 3: Implementation of reinforcement controller

Since humanoid robot is taking the decision by its own, knowing the present state and switching over to the next state. It does not know the kinematics and dynamics of the system. In the present work lower body of humanoid robot is at certain position and that position is 0° to the ankle joint,-10° to the knee joint and -20° to the hip joint for the right leg and opposite of these vales to the left leg as shown in figure2 of the reinforcement basic model.

When these values are sensed with the help of gyroscope sensor, robot must travel the goal point. The goal point for the ankle joint, knee joint and hip joint are 10°,15° and 20° respectively. Time taking to reach these points are independent from the kinematics and dynamics calculation. In reinforcement controller to reach next state, it is totally dependent on the processor capabilities that are generating the control signal. If it is too fast, then joint motor is unable to sense that signal. If it is too slow, then it is very tough to switch to the next state. Microcontroller and joint motor, both have different scan time.

To match the synchronization, design is totally based on the higher scan time of device. To match these frequencies time delay is being used.

VI. RESULT & SIMULATION

Simulation is carried out for the model as shown in figure4 and figure5 in Multibody dynamics toolbox. Figure4 shows the initial state of lower body of humanoid robot. After sensing the present state, the lower body switches to the goal state without any kinematics and dynamics of the system. It is shown in figure5.

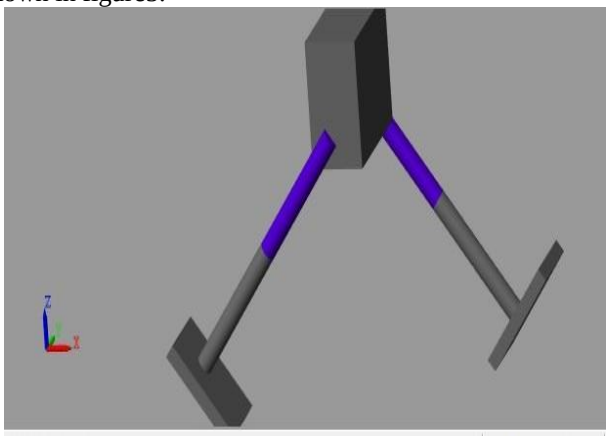

Figure 4: Robot is at present state

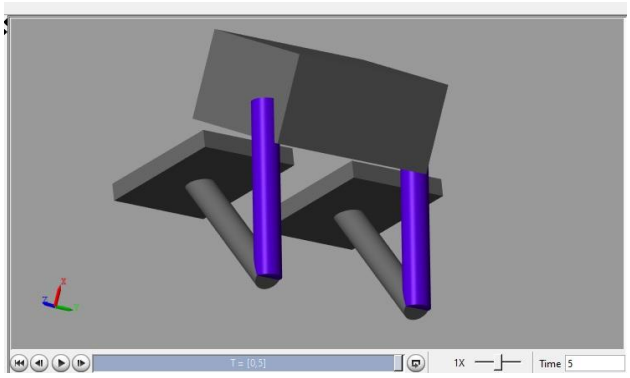

Figure 5: Switching to goal state

Result1: Learning of ankle joint

When the lower body of humanoid root is switched to the goal state, at that moment the ankle joint started from the starting point 0° and reached goal point10°. During this switching, the ankle joint touches different intermediate positions as shown in figure6.

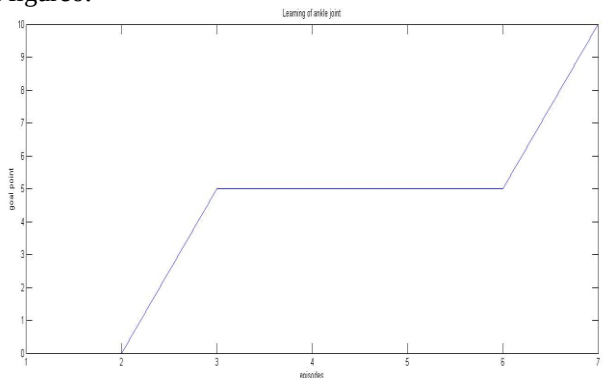

Figure6: learning of ankle joint

And the corresponding optimal policy for the ankle joint state and action is shown in table1.

Table1:Q optimal policy for ankle joint

Sate	Reward(Sate)	Reward(action)
0		0
1.11111111111111	0.1111	0.1111
2.22222222222222	0.2222	0.2222
3.33333333333333	0.3333	0.3333
4.44444444444445	0.4444	91.7544
5.55555555555556	0.5556	0.5556
6.66666666666667	0.6667	0.6667
7.77777777777778	0.7778	0.7778
8.88888888888889	0.8889	0.8889
10	1.0000	1.0000

Result2: Learning of knee joint

The knee joint starts from -10° and reaches the goal point 15°. During this switching the knee joint touches different intermediate position as shown in figure7.

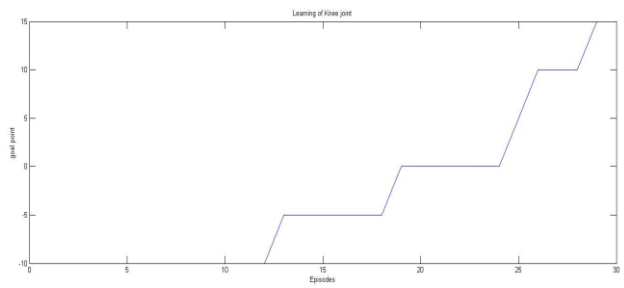

Figure7: Learning of knee joint

And the corresponding optimal policy for the knee joint state and action is shown in table2.

Table2:Q optimal policy for knee joint

Sate	Reward(Sate)	Reward(action)
-10	-1.0000	-1.0000
-7.22222222222222	-0.7222	-0.7222
-4.44444444444445	-0.4444	-0.4444
-1.66666666666667	-0.1667	-0.1667
1.11111111111111	0.1111	0.1111
3.88888888888889	0.3889	0.3889
6.66666666666667	0.6667	0.6667
9.44444444444444	0.9444	92.6594
12.2222222222222	1.2222	1.2222
15	1.5000	1.5000

Result3: Learning of hip joint

The hip joint is started from -20° and reaching to the goal point 20°.During this switching the knee joint touches different intermediate positions as shown in figure8.

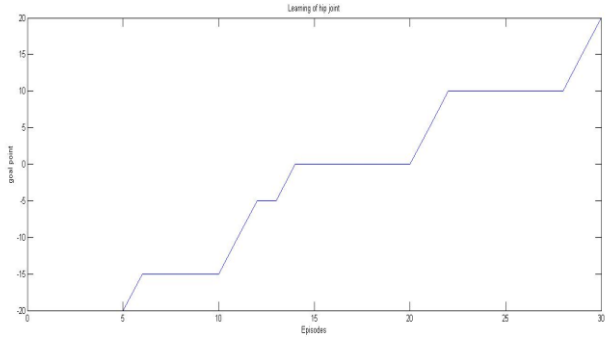

Figure 8: Learning of hip joint

The corresponding optimal policy for the hip joint state and action in table3.

Table3:Q optimal policy for hip joint

Sate	Reward(Sate)	Reward(action)
-20	-2.0000	-2.0000
-15.5555555555556	-1.5556	-1.5556
-11.1111111111111	-1.1111	-1.1111
-6.66666666666667	-0.6667	-0.6667
-2.22222222222222	-0.2222	-0.2222
2.22222222222222	.2222	.2222
6.66666666666667	0.6667	0.6667
11.1111111111111	1.1111	1.1111
15.5555555555556	1.5556	93.5756
20	2.0000	2.0000

VII. DISCUSSION

Result obtained from the reinforcement algorithm is totally depend on the randomness values of the transition probabilities. While switching from the start point to goal point, humanoid is taking action either zero throw of the joint motor or full throw of the motor. Humanoid is deciding the action on the random values of the transition probabilities. Due to this reason a observation made in figure, found that the joint position is at same position and after some time it is switching to the next position.

Each time the joint movement response is changing to reach the goal point.

VIII. CONCLUSION

While implementing the reinforcement controller to switch the humanoid robot from present state to next state , controller understand the only weight of randomness of the transition probabilites.While decidng the weight of the transition probablites, reinforcement algorithm helps to find these values. So that it can feed to model prediciting controller. After several experiments, a observation is made that the minimum epsides is obtained to reach the goal point.At that minimum episode , the transition probability is obtained and these values work as the deciding factor to switch to the next state within minimum time. So that switching happens in humanoid robot will be fast without knowing the mechanics and dynamics of the system.

REFERENCES

1. Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath, [2017] "Deep Reinforcement Learning", IEEE Signal Processing Magazine, November 2017
2. Sudhir Raj, Cheruvu Siva Kumar, [2013] "Q Learning based Reinforcement Learning Approach to Bipedal Walking Control" Proceedings of the 1st International and 16th National Conference on Machines and Mechanisms (iNaCoMM2013), IIT Roorkee, India, Dec 18-20 2013.
3. Deepak Bharadwaj,Manish Prateek,[2019] "Kinematics & Dynamics of Lower Body of Autonomous Humnaoid Biped Robot",IJITEE,Volume-8,Issue-4,February,2019
4. C.Hernandez-santos,E-rodriguez_leal,etal[2012], "Kinematics & Dynamics of a New 16-DOF Humanoid Biped Robot with Active Toe Joint", International Journal of Advanced Robotics System,INTECH,17,August,2012
5. DDuško Kati, Miomir Vukobratovi uško ,etal[2007], "Reinforcement Learning Algorithms In Humanoid Robotics", Humanoid Robots, New Developments, Book edited by: Armando Carlos de Pina Filho ISBN 978-3-902613-02-8, pp.582, I-Tech, Vienna, Austria, June 2007
6. Duško Katić,[2012] "Episodic Reinforcement Learning Control Approach for Biped Walking" Serbian Journal Of Electrical Engineering Vol. 9, No. 2, June 2012, 231-245
7. Robert Platt, Robert Burridge, etal[2006], "Humanoid Mobile Manipulation Using Controller Refinement" Dexterous Robotics Laboratory Johnson Space Center, NASA
8. David Garc´ıa and Leonardo Garrido, [2012] "Generation of Motion Policies Applying Multiagent Reinforcement Learning in Simulated Robotic Soccer", Joint conference on Artificial Intelligence,August 2012
9. Robert Platt, Robert Burridge, etal[2013], "Humanoid Mobile Manipulation Using Controller Refinement" Dexterous Robotics Laboratory Johnson Space Center, NASA,july 2013
10. Riadh Zaier, "The Future Of Humanoid Robots – Research And Applications" Published by InTech, Janeza Trdine 9, 51000 Rijeka, Croatia, ISBN 978-953-307-951-6
11. Ajay Kumar Tanwani[2011] "Optimizing Walking of a Humanoid Robot using Reinforcement Learning"MS Thesis, Warsaw, November, 2011
12. Gyula MESTER1, Aleksandar Rodic,[2011] "Contribution To The Simulation Of Humanoid Kondo Robot", Annals of faculty Engineering Hunedora, International journal of engineering , TOME IX , ISSN 1584-2665.
13. Hirokai Arie, Tetsuya Ogata, etal[2007]"Reinforcement learning of a continuous motor sequence with hidden states" Advanced Robotics, Vol. 21, No. 10, pp. 1215–1229 (2007)

AUTHORS PROFILE

Mr Deepak Bharadwaj, received his M.Tech Degree in 2008-10 from the MDU university Rohtak.. At present he is working as Assistant Professor in mechanical engineering department of UPES Dehradun. His area of research is robotics & Automation. At present he is developing the reinforcement control algorithm for an autonomous humanoid robot.

Professor Manish Prateek, currently working as professor of school of computer science. His area of research is soft computing and wireless network. He had published many research papers on robotics. Currently he is the Dean of School of computer science department Of UPES Dehradun.

Ms. Rashmi Sharma, received her Master's in Computer Application (1999) DAVV Indore (M.P.) and Master's in Technology (2010) UPTU, Lucknow. At present, doing research in field of Reinforcement Learning and Bipedal. Interest area includes - Soft Computing, Artificial Intelligence, Machine Learning and Machine Vision