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 capabilities, these types of manipulator fail to do the task. In such cases the preprogrammed robot does not know what to do. The reinforcement 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 mechanics 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. Reinforcement 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, because when the humanoid is falling on the ground to reach the goal position, humanoid should bring the feet either in forward 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. Reinforcement 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

III. REINFORCEMENT LEARNING (RL) MODEL

Kinematic and dynamic analysis of the system gives real time information to the controller of the humanoid robot.
Development of Reinforcement Control Algorithm of Lower Body of Autonomous 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 proper action. After choosing the action the joint leg 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.

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 certainty equivalence, temporal difference, Q-learning and SARSA (state-action-reward -state-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-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) = (1-α) Q(s,a)+α(r+γmax_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
[~,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)+discount
*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.

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

Result 1: 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 point 10°. During this switching, the ankle joint touches different intermediate positions as shown in figure 6.

Table 1: Q optimal policy for ankle joint

<table>
<thead>
<tr>
<th>State</th>
<th>Reward(State)</th>
<th>Reward(action)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1111</td>
<td>0.1111</td>
</tr>
<tr>
<td>1.11111111111111</td>
<td>0.1111</td>
<td>0.1111</td>
</tr>
<tr>
<td>2.22222222222222</td>
<td>0.2222</td>
<td>0.2222</td>
</tr>
<tr>
<td>3.33333333333333</td>
<td>0.3333</td>
<td>0.3333</td>
</tr>
<tr>
<td>4.44444444444444</td>
<td>0.4444</td>
<td>0.417544</td>
</tr>
<tr>
<td>5.55555555555555</td>
<td>0.5556</td>
<td>0.5556</td>
</tr>
<tr>
<td>6.66666666666667</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>7.77777777777778</td>
<td>0.7778</td>
<td>0.7778</td>
</tr>
<tr>
<td>8.88888888888889</td>
<td>0.8889</td>
<td>0.8889</td>
</tr>
<tr>
<td>10</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Result 2: 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 positions as shown in figure 7.

Table 2: Q optimal policy for knee joint

<table>
<thead>
<tr>
<th>State</th>
<th>Reward(State)</th>
<th>Reward(action)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>-1.0000</td>
<td>-1.0000</td>
</tr>
<tr>
<td>-7.22222222222222</td>
<td>-0.7222</td>
<td>-0.7222</td>
</tr>
<tr>
<td>-4.44444444444445</td>
<td>-0.4444</td>
<td>-0.4444</td>
</tr>
<tr>
<td>-1.66666666666667</td>
<td>-0.1667</td>
<td>-0.1667</td>
</tr>
<tr>
<td>1.11111111111111</td>
<td>0.1111</td>
<td>0.1111</td>
</tr>
<tr>
<td>3.88888888888889</td>
<td>0.3889</td>
<td>0.3889</td>
</tr>
<tr>
<td>6.66666666666667</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>9.44444444444444</td>
<td>0.9444</td>
<td>92.6594</td>
</tr>
<tr>
<td>12.22222222222222</td>
<td>1.2222</td>
<td>1.2222</td>
</tr>
<tr>
<td>15</td>
<td>1.5000</td>
<td>1.5000</td>
</tr>
</tbody>
</table>

Result 3: 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 figure 8.

Table 3: Q optimal policy for hip joint

<table>
<thead>
<tr>
<th>State</th>
<th>Reward(State)</th>
<th>Reward(action)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-20</td>
<td>-2.0000</td>
<td>-2.0000</td>
</tr>
<tr>
<td>-15.55555555555556</td>
<td>-1.5556</td>
<td>-1.5556</td>
</tr>
<tr>
<td>-11.11111111111111</td>
<td>-1.1111</td>
<td>-1.1111</td>
</tr>
<tr>
<td>-6.66666666666667</td>
<td>-0.6667</td>
<td>-0.6667</td>
</tr>
<tr>
<td>-2.22222222222222</td>
<td>-0.2222</td>
<td>-0.2222</td>
</tr>
<tr>
<td>2.22222222222222</td>
<td>2.2222</td>
<td>2.2222</td>
</tr>
<tr>
<td>6.66666666666667</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>11.11111111111111</td>
<td>1.1111</td>
<td>1.1111</td>
</tr>
<tr>
<td>15.55555555555556</td>
<td>1.5556</td>
<td>93.5756</td>
</tr>
<tr>
<td>20</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
</tbody>
</table>

Figure 5: Switching to goal state

Figure 6: Learning of ankle joint
And the corresponding optimal policy for the ankle joint state and action is shown in table 1.

Figure 7: Learning of knee joint
And the corresponding optimal policy for the knee joint state and action is shown in table 2.

Figure 8: Learning of hip joint
And the corresponding optimal policy for the hip joint state and action is shown in table 3.
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 probabilities. While deciding the weight of the transition probabilities, reinforcement algorithm helps to find these values. So that it can feed to model predicting controller. After several experiments, a observation is made that the minimum episodes 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


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