Learning to Fire at Targets by an iCub Humanoid Robot

Vishnu K. Nath and Stephen E. Levinson
University of Illinois at Urbana-Champaign
405 North Mathews Avenue
Urbana, IL 61801

Abstract

In this paper, we present an algorithm that integrates computer vision with machine learning to enable a humanoid robot to accurately fire at objects classified as targets. The robot needs to be calibrated to hold the gun and instructed how to pull the trigger. Two algorithms are proposed and are executed depending on the dynamics of the target. If the target is stationery, a least mean square (LMS) approach is used to compute the error and adjust the gun muzzle accordingly. If the target is found to be dynamic, a modified Q-learning is used to best predict the object position and velocity and to adjust relevant parameters, as necessary. The image processing utilizes the OpenCV library to detect the target and point of impact of the bullets. The approach is evaluated on a 53-DOF humanoid robot iCub.

This work is an example of fine motor control which is the basis for much of natural language processing by spatial reasoning. It is one aspect of a long term research effort on automatic language acquisition [3].

Introduction

Acquiring new motor skills requires various forms of learning that involves a mixture of imitation and self-learning and improvisation techniques. Humanoid robots should also be able to pick up new skills after being shown the initial step to perform certain tasks. The complexity of the task decides the ideal mixture of the two techniques that would yield the best results. Some tasks can be performed only by manually programming each parameter and some can be learnt by the robot itself using Reinforcement Learning.

In this paper, we present an integrated way with which a humanoid iCub robot would be able to fire at targets using a commercial NERF gun. After manually programming the robot to hold the gun, the robot learns by itself to shoot the NERF bullets in such a way that it hits the center of the bulls eye.

The implementation is modular in design, with the major modules being robot kinematics, computer vision and learning. The computer vision module picks out the targets from the field of vision and computes the desired target 'sweet spot' i.e. the central part of the target. The algorithm would alter the D-H parameters of the robot's arm so that the point of impact of the bullet gets closer to the 'sweet spot' on every iteration. Two algorithms are proposed and their usage would be decided based upon the dynamics of the target. If the target is stationery, a least mean square approximation algorithm is used, while a modified Q-learning algorithm is proposed for moving targets.

Computer Vision

The computer vision module is required to determine the location of the target and the point of impact of the bullet. The target is a bulls eye that consists of a yellow circular center, circumscribed by red, blue, black and white circles, all concentric in nature. Due to the nature of the target being used, we use the Hough circle transform to detect objects that are circular in nature. The center of every circular object is determined by using moments computed for that object. An object that consists of five circles sharing a common center point is classified as a target.

The camera feed from the iCub has a resolution of 648x488 at 60 FPS, with 3-channels of 8 bits each. The feed is converted into grayscale frames and the Hough circle transform is applied on the current frame. All objects that have been classified as circular would have their moments computed in order to determine their respective centers. The i,j moment for an object k in the object is given by

$$\mathbf{m_{ij}(k)} = \sum_{r,c} \mathbf{r^i c^j I_k(r,c)},$$

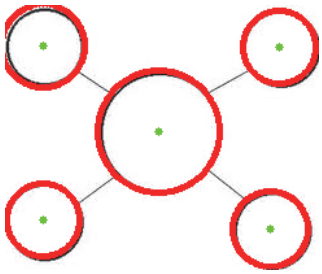
where (r,c) refers to the pixel whose row number is r and column number is c. I_k is the indicator function, and outputs 1 when the pixel (r,c) is in the object, and 0 otherwise[4]. The center of the circular objects are computed by

$$r_i' = \frac{\sum_{r,c} rI_i(r,c)}{\sum_{r,c} I_i(r,c)} = \frac{m_{10}(i)}{m_{00}(i)}$$

$$c_i' = \frac{\sum_{r,c} cI_i(r,c)}{\sum_{r,c} I_i(r,c)} = \frac{m_{01}(i)}{m_{00}(i)}$$

Figure 1(a) shows the vision module correctly identifying a target that consists of five concentric circles. The red circle indicates the boundary of the target circle and the green dot is the computed center point. The green dot would be the ideal point of impact for the bullet. Figure 1(b) shows an example where multiple circles might be present in the frame of vision. In this figure, each circle is considered an individual target circle and their individual center points are computed. However, none of them overlap and therefore this arrangement was not considered as a target.

(a) Bulls eye target correctly identified



(b)Multiple circles not identified as target

Figure 1: Performance of the computer vision module using Hough Circle Transform to classify objects as targets

Robot Kinematics

The entire iCub robot has 53 degrees of freedom. To accomplish our goal, we decided to use the right hand to hold the gun and fire at the target. The DH parameters for the right hand at home position were computed and are given in table 1.

Link	a	d	α	θ
1	32	0	π/2	0
2	0	-5.5	π/2	- π/2
3	- 23.467	-143.3	π/2	- π/2
4	0	-107.74	π/2	- π/2
5	0	0	- π/2	- π/2
6	-15	-152.28	- π/2	- π/2
7	15	0	π/2	π/2
8	0	-137.3	π/2	- π/2
9	0	0	π/2	π/2
10	62.5	16	0	π

Table 1: D-H parameters of right hand at home position

Figure 2 shows the position vector of all 10 joints of the right arm in the home position.

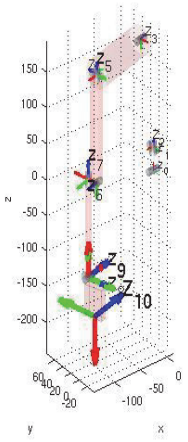


Figure 2: Position vector of the joints in the right arm

The right arm needs to be brought to a default firing position which is characterized by the arm being orthogonal to the z-axis of the robot. The transformation matrices have been calculated and are given as

$$T_1^0 = \begin{vmatrix} 0 & 0 & -1 & 32 \\ 0 & -1 & 0 & 5.5 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_2^0 = \begin{vmatrix} 0 & -1 & 0 & 175.3 \\ 1 & 0 & 0 & -17.967 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_3^0 = \begin{vmatrix} 1 & 0 & 0 & 175.3 \\ 0 & 0 & -1 & -17.967 \\ 0 & 1 & 0 & -107.74 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_4^0 = \begin{vmatrix} 0 & 0 & 1 & 175.3 \\ 0 & 1 & 0 & -17.967 \\ -1 & 0 & 0 & -107.74 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_5^0 = \begin{vmatrix} 0 & -1 & 0 & 175.3 \\ -1 & 0 & 0 & -17.967 \\ 0 & 0 & -1 & -107.74 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_6^0 = \begin{vmatrix} 1 & 0 & 0 & 160.3 \\ 0 & 0 & -1 & -17.967 \\ 0 & 1 & 0 & 44.54 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_7^0 = \begin{vmatrix} 0 & 0 & 1 & 160.3 \\ 0 & -1 & 0 & -17.967 \\ 1 & 0 & 0 & 59.54 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_8^0 = \begin{vmatrix} 0 & 1 & 0 & 23 \\ 1 & 0 & 0 & -17.967 \\ 0 & 0 & -1 & 59.54 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_9^0 = \begin{vmatrix} 1 & 0 & 0 & 23 \\ 0 & 0 & 1 & -17.967 \\ 0 & -1 & 0 & 59.54 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_{10}^0 = \begin{vmatrix} -1 & 0 & 0 & -39.5 \\ 0 & 0 & 1 & -1.967 \\ 0 & 1 & 0 & 59.54 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

The execution of these 10 transformation matrices would have the shoulder roll and the elbow and wrist yaw set to π/2 radians. It is this position that is considered the default firing position and is entered into by the iCub robot when a target has been detected in its field of vision. The learning algorithm would attempt to modify the D-H parameters of the arm from this point onwards to maximize the probability of the bullet hitting the target with high accuracy. Figure 3 illustrates the position vectors of all the joints at this position as well as the position of the arm after all the transformation matrices have been applied.

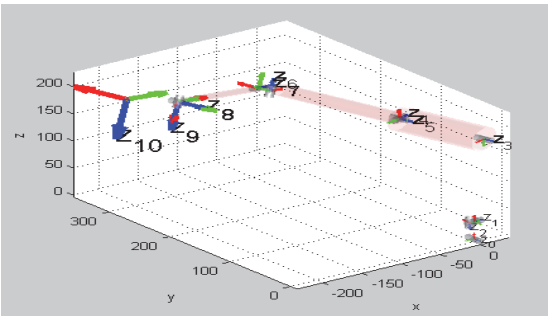


Figure 3: Position vector of the joints in the right arm after the transformations

Experimental Setup

The experiment was conducted using commercially available NERF darts as bullets. In order to understand the behavior of the bullet and its scalability to live ammunition bullets, certain kinematic parameters of the bullet had to be measured. Also, the effect of air resistance was not provided by the manufacturer and had to be modeled before using it for testing purposes. Figure 4 shows the horizontal displacement of the bullet over time[1].

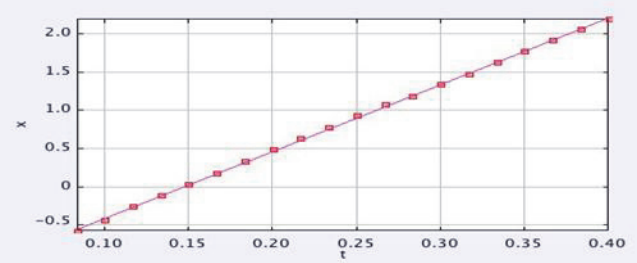


Figure 4: Horizontal displacement of the bullet with time

From figure 4, it is evident that the displacement curve is linear, thereby making it safe to ignore the effects of air resistance on the bullet. Figure 5 shows the horizontal component of velocities over different iterations[1].

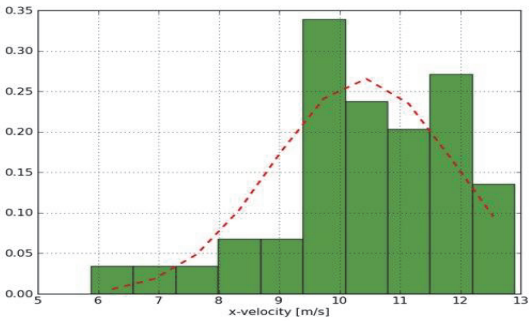


Figure 5: Distribution of the x-velocities of the bullets

From figure 5, the average velocity was computed to be 10.4 m/s and the standard deviation was computed to be 1.5 m/s. From this information, we concluded that it was best to place the target at a distance of ~2 meters away from the iCub robot to allow maximum flight time for the bullet and still make an impact on the target with sufficient force.

Learning Algorithms

The learning algorithm module has been bifurcated to contain two different algorithms, based on the target's dynamics. If the target is stationery, we feel that the least mean squares approximation (LMS) approach would be

best, while if the target is moving, a modified Q-learning approach would work best.

Least Mean Square Approximation

At the beginning of the first attempt, the vision module would compute the target's position. The robot would calibrate its dynamics to hit the given target. However, there is a high probability that the bullet would not hit the target accurately. The error functions are given by

$$\varepsilon_\theta = \left\|\theta' - \theta\right\|$$
$$\varepsilon_r = \left\|r' - r\right\|$$

where ε_θ and ε_r are the errors caused by angular and translational differences respectively. The least mean square approximation is used to iteratively alter the D-H parameters of the right arm so that the magnitude of both errors converge to 0, implying that the bullet would hit the target with high accuracy.

Modified Q-Learning Algorithm

For dynamic targets, the least mean square approximation method would not work simply because the target would not be at the same place at a given point in time. The problem becomes significantly more complex if the robot has no prior information about the kinematics of the target.

The first approach was to use a temporal difference Q-Learning algorithm. The update equation for the temporal difference Q-Learning algorithm is given by

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,(R(s) + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

where α is the learning rate and γ is the discount factor[3].

However, this algorithm is an off-policy learning algorithm and therefore, the probability of this algorithm converging upon an optimum policy to shoot down a moving target is unsatisfactorily low.

A learning algorithm that does utilize a policy to maximize the probability of successfully hitting the target is the State-Action-Reward-State-Action, or SARSA algorithm. The update equation for SARSA is given by

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,(R(s) + \gamma\, Q(s',a') - Q(s,a))$$

The advantage of SARSA over temporal-difference Q-Learning is that SARSA waits until an action as actually taken place and backs up the Q-value for that action[4]. State space exploration is also required to obtain the ideal policy, and SARSA explores this policy, while Q-Learning does not. The optimum policy is given by

$$\pi^* = \operatorname{argmax}_\pi \sum_h P(h|e)\, u_h^\pi$$

where the posterior probability P(h|e) is obtained by applying Bayes' rule. This creates a feedback loop that allows constant improvisation. Due to the feedback loop and the constant convergence towards an optimum policy, the robot is guaranteed to find the best way to fire at a moving target. During the experimental simulations, the average number of times a bullet needed to be fired before the robot attained high probability of hitting the target was 5.

Future Work

Currently the iCub robot can fire at stationery targets and targets that are moving at a constant velocity in a fixed direction. The success rate of firing at targets that move in a random pattern is not high. The future direction of this research project would be to analyze patterns of movement of the target and predict its position after a time interval Δt.

References

1. Allain, Rhett. "How Fast Are These Nerf Bullets?" *Wired.com*. Conde Nast Digital, 30 June 2011. Web. 05 Apr. 2012. <http://www.wired.com/wiredscience/2011/06/how-fast-are-these-nerf-bullets/>.

2. Kormushev, Petar, Sylvain Calinon, Ryo Saegusa, and Giorgio Metta. "Learning the Skill of Archery by a Humanoid Robot ICub." 2010 IEEE-RAS International Conference on Humanoid Robots (2010): 417-23. Print.

3. Majure, Lydia, Logan Niehaus, and Alex Duda. "Integrating Language and Motor Function on a Humanoid Robot." *IEEE/RSJ Proceedings of The International Conference on Intelligent Robots and Systems (IROS 2010)* (2010): n. pag. Print.

4. Russell, Stuart, and Peter Norvig. "Reinforcement Learning." *Artificial Intelligence: A Modern Approach*. Third ed. New Jersey: Prentice Hall, 2010. N. pag. Print.

5. Spong, Mark, Seth Hutchinson, and M. Vidyasagar. *Robot Modelling and Control*. New Jersey: John Wiley & Sons, 2006. Print.