An Investigation into Alternative Control Methods for a Robotic Arm

Author: Leo Morgan
Supervisor: Dr Guido Herrmman
University of Bristol
Department of Mechanical Engineering

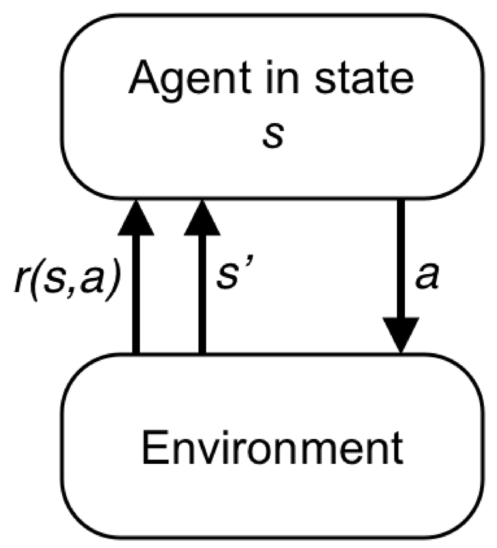
Project Brief

To develop an obstacle avoidance scheme for the BERT II robot arm at the Bristol Robotics Laboratory through the use of reinforcement learning algorithms and posture control.

Reinforcement Learning

Reinforcement learning is a broad term describing the learning of an agent when interacting with an environment. With every action taken by the learner, they will be rewarded or punished as a consequence of this action. After a necessary of amount of experience in the environment, the agent learns which actions are preferable by the association of an action to a reward or punishment.

***Q* Learning** is a simple form of reinforcement learning, where the problem of learning a task is formalised as a **Markov Decision Process** (MDP).



Here, there is finite set of states, S, and a finite set of actions, A. At each state, s, the agent can take an action, a, which will reap a numerical reward, $r(s,a)$ and move the agent into the next state, s'.

Each action a at state s has an associated Q value

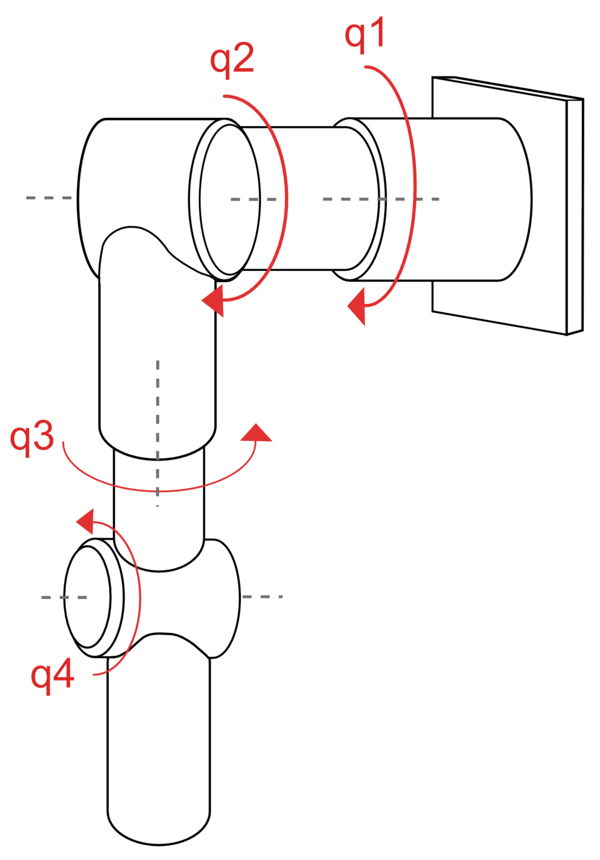
$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a')$$

where a' and s' denote the next action and the next state, and γ is the discounted reward factor, $0 < \gamma \leq 1$, which affects how future rewards are valued against immediate ones. Once every action at every state has a Q value, the optimum sequence of actions is found by choosing the action at each state with the greatest Q value.

This Q learning rule was used to create a MATLAB program which iteratively calculates the optimum path through a two dimensional, deterministic environment by *learning* the best path through trial and error offline.

The BERT II Arm

The BERT II arm is a humanoid robotic arm consisting of a 7 degree of freedom (DOF) arm at the Bristol Robotics Laboratory, mirroring the joints in a human arm.



As this project concerns the position of the of the end-effector (and not the orientation), we can simplify the system by ignoring the wrist joint. Hence, the arm effectively functions with 4 DOF.
The arm's controller is modelled in Simulink and is interfaced via dSpace.

Posture Control

If a robot arm has redundant degrees of freedom, the posture of the arm can be controlled. By decoupling the control of the arm into two schemes, **task** and **posture**: we can develop a posture controller which works independently from the task controller. Γ is the vector of control torques to the arm's joint actuators.

$$\Gamma = \Gamma_{task} + \Gamma_{posture}$$

To enhance the obstacle avoidance of the arm, a cost function describing proximity of the arm's elbow (x_e, z_e) with an obstacle (x_o, z_o) was created:

$$U_o = K_o \left(\frac{1}{\delta_o + |x_e - x_o|^2 + |z_e - z_o|^2} \right)$$

Including this term with the existing cost function due to gravity terms and joint limits, the desired vector of joint torques, Γ_p is calculated through cost minimisation:

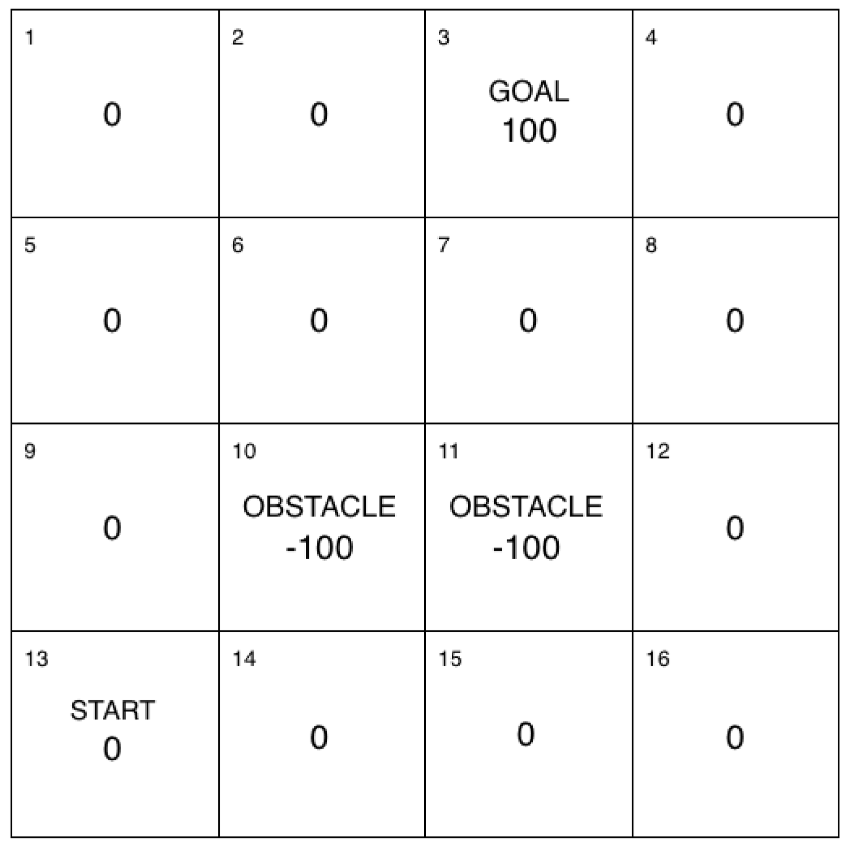
$$\Gamma_p = -K_p(\Delta U)^T - K_d\,\dot{q}, \qquad \Delta U = \left(\frac{\partial U}{\partial q}\right)$$

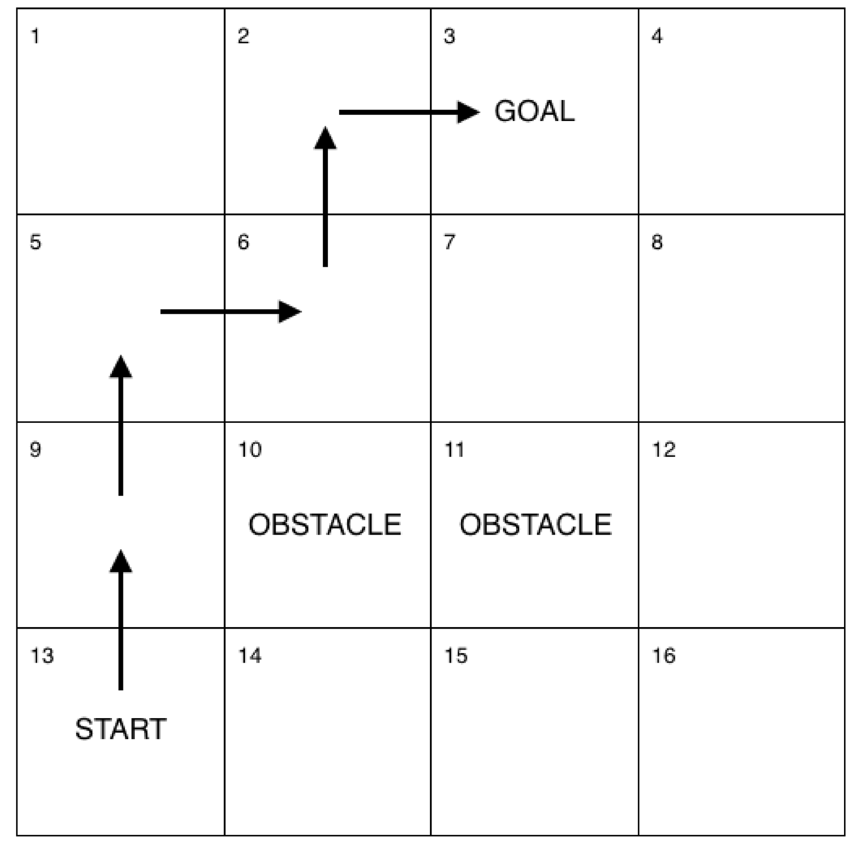
and is integrated into the task space by:

$$\Gamma_{\text{posture}} = N^T \Gamma_p \qquad N^T = (I - J^T \hat{\bar{J}}^T)$$

Q Learning Environment

At first, a theoretical environment was created, with 16 states. A goal state is represented by a reward of +100, while the two obstacles are represented by -100. Using the Q learning algorithms with MATLAB, the optimum path was calculated as illustrated in the second figure. This path was then achieved by the arm's end-effector through integrating the Q learning algorithm with the arm's controller in Simulink.





The VICON System

A development in the project was the use of an optical motion capture system to identify real-world, tangible objects and then use this source as a environment for the Q learning task.

Final Results

The VICON optical motion capture system was used to visually recognise the position of three objects: a starting position, an obstacle and a goal. Using these co-ordinates, the Q learning guided the arm through this virtual environment by sending the demands as shown:

For this experiment, the state length was 4cm and the state transition time 6 seconds. These results illustrate acceptable tracking from the end-effector, however any end-effector error at a state transition is dangerous as it introduces the possibility that the end-effector may move through an undesired state.

While the posture control produced promising results in Simulation, further testing is required for its integration into the practical setup.