REINFORCEMENT LEARNING ROBOT

AN EXPERIMENT IN SELF-TAUGHT MOBILITY

INTRODUCTION

- This project will construct both a self-contained training environment and robot with an articulated appendage with which the robot will learn to use to move in a specific direction.
- The system will employ software that will utilize a reward to teach the robot which actions result in movement in the desired direction.

ORIGIN

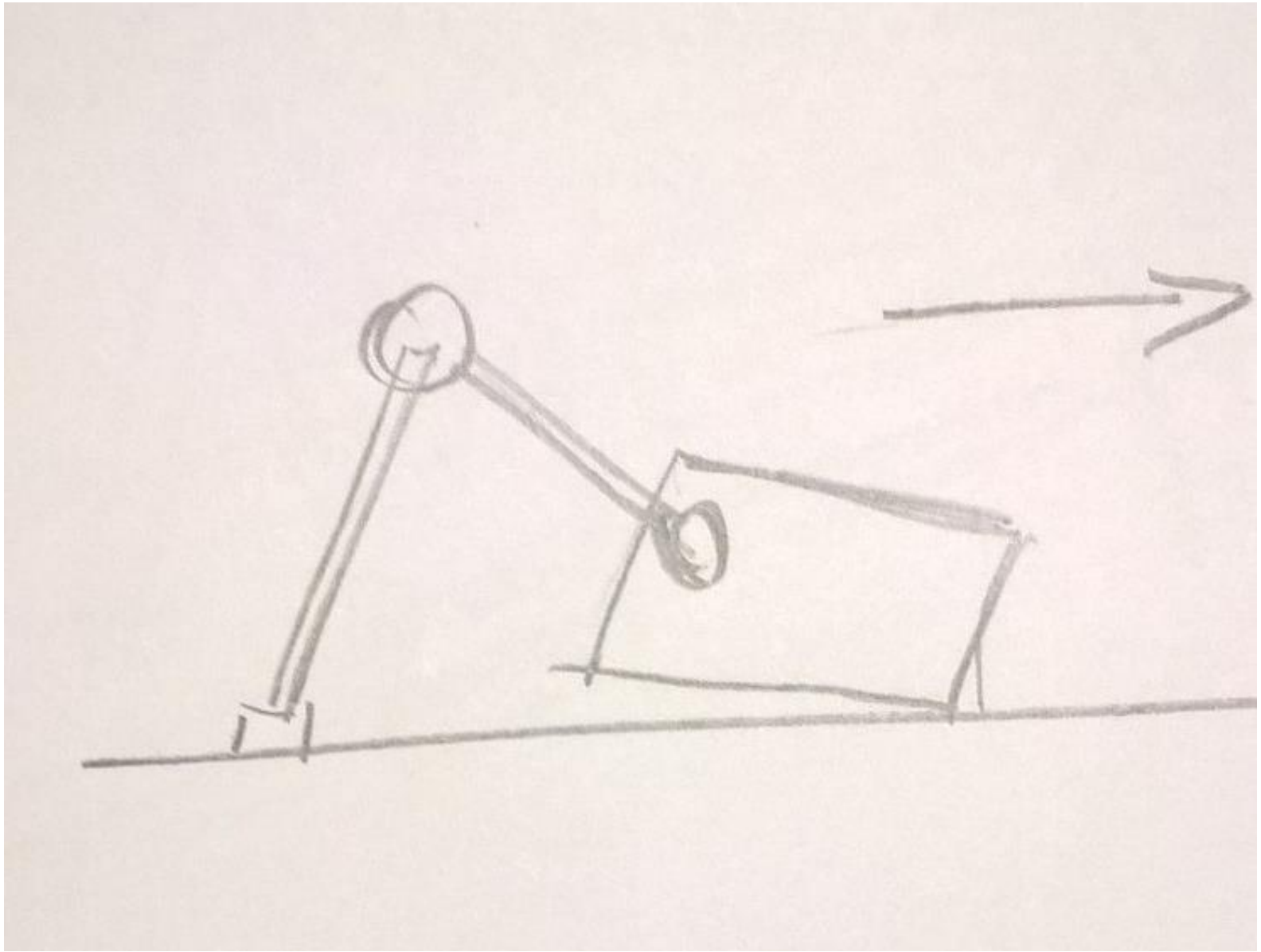
- Scott presented such a 'robot' learning in a virtual environment during a Q-Learning discussion a few months back.
- The presentation described a reward/penalty system in which the *agent* (robot) learned to move through a trial and error process.
- The Metrowest Boston Machine Learning Group had great success in developing, training and demonstrating a model self-driving car project.

VIDEO PROOF

- https://ibm.ent.box.com/s/jkzxleqk1775stk1cyt2mi4jjeq7hy1u/file/385071978673

WHY?

- The physical 'embodiment' of machine learning systems is very engaging (and entertaining).
- Most reinforcement robot examples transfer a 'virtually' trained model into a physical machine.
- Building Something = Learning Something
  - Construction of reliable hardware
  - Latency between the model making a decision and the 'machine' taking action
  - Machine calibration between uses (sensors, servos, etc.)



SKETCH

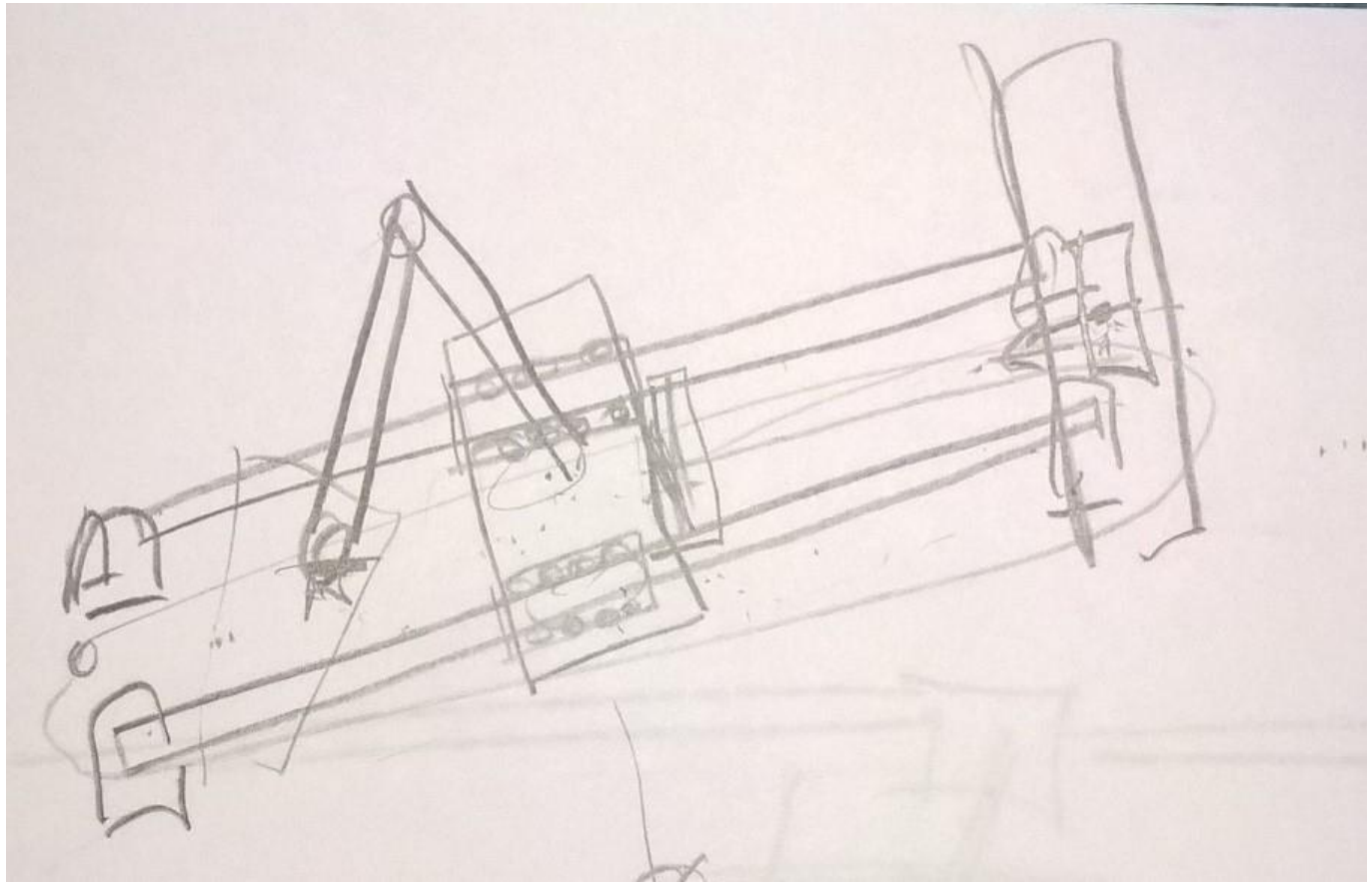
THE ROBOT WILL HAVE AN APPENDAGE THAT IT WILL USE TO PUSH IT IN THE SPECIFIED DIRECTION.

DESIGN CONSTRAINTS

- The robot and training environment must be robust, self-contained, and easy to pick up and transport to demonstrate at meetings.
- It is expected that the training mode does not require supervision. As a reinforcement learning system the demonstrator should be 'turned on' and 'learn' in an environment that can handle all learning attempt outcomes.
- The system allows multiple learned 'plans' to be demonstrated.

FIRST THOUGHTS

- The robot will be constrained to movement along one axis by operating on rails.



RAIL SYSTEM ADVANTAGE

- Limits movement which makes it easier to determine the reward and penalty.
- Locks all components of the system down for transport and demonstration.
- Facilitates the construction of a controlled environment.

RAIL COMPONENTS

- The rails are 300mm (about 1 foot long.) Shoebox size.
- There are standoffs which allow the rails to be mounted above a fixed surface.
- The rail 'trolley' contains bearings to reduce friction and surround the rod to limit any other axis movement.

RAIL ASSEMBLY

ARTICULATING APPENDAGE

- I took inspiration from camera gimble devices.

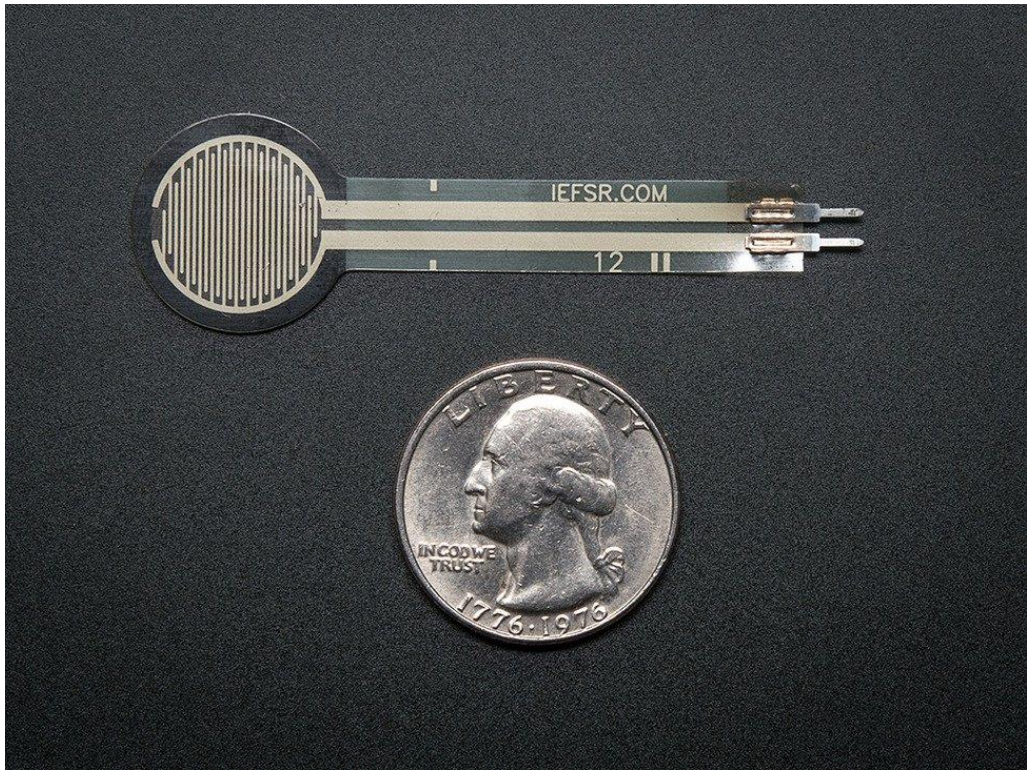
SERVOS IN MOTION

- https://www.adafruit.com/product/1967?gclid=EAIaIQobChMI4bSemoeO5wIVzJ-zCh3A0w-OEAQYAyABEgIFYfD_BwE

SERVO MOUNTING BRACKETS

- High-quality and durable in performance.
- The mount bracket easily attaches to MG995 996 Metal Mechanical Robot Servo

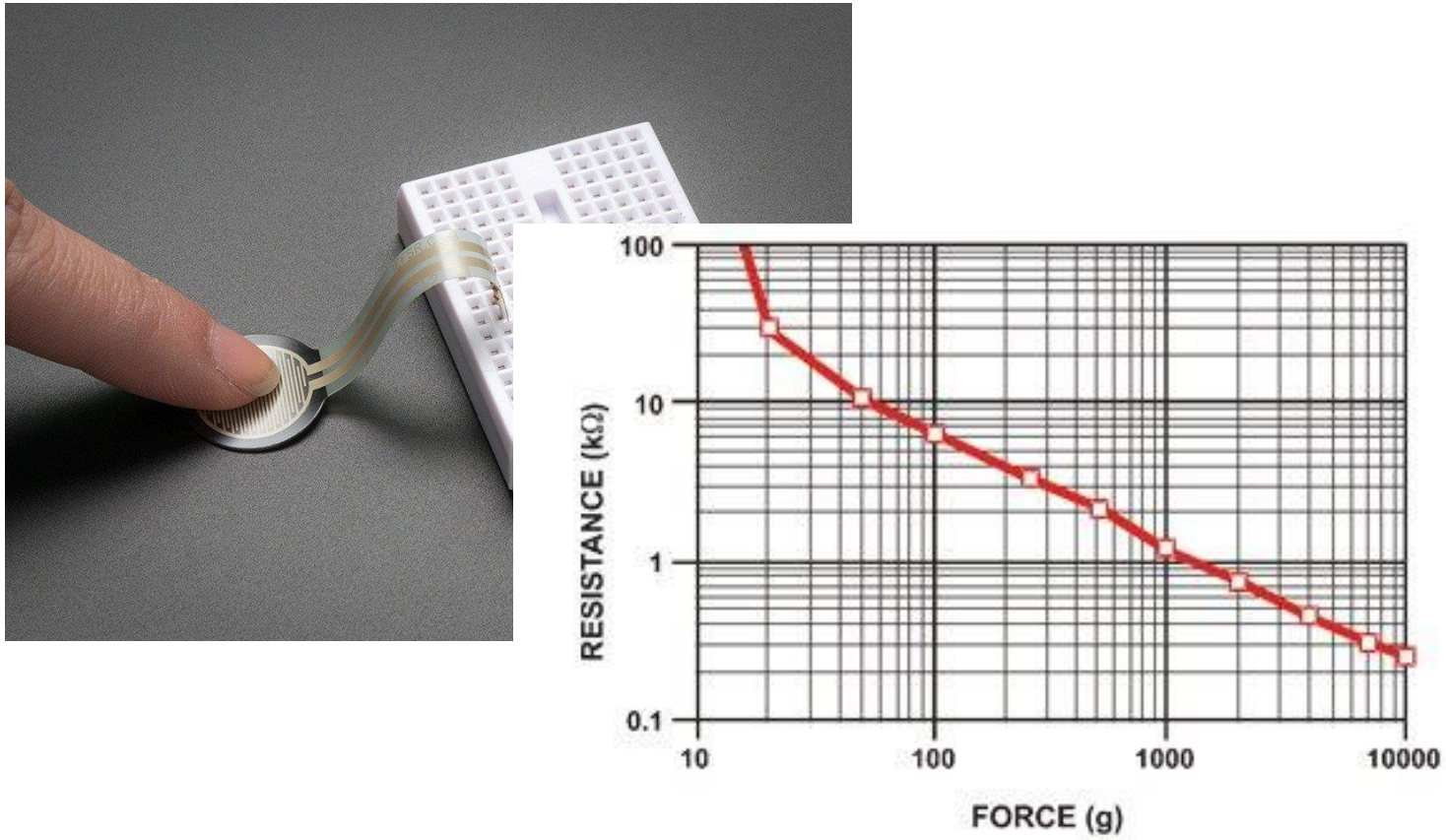
APPENDAGE FORCE SENSOR



- In initial discussions with Scott we identified the need for the robot to be rewarded when the appendage touched the ground.
- We will include a force sensor on the end of the 'foot'.

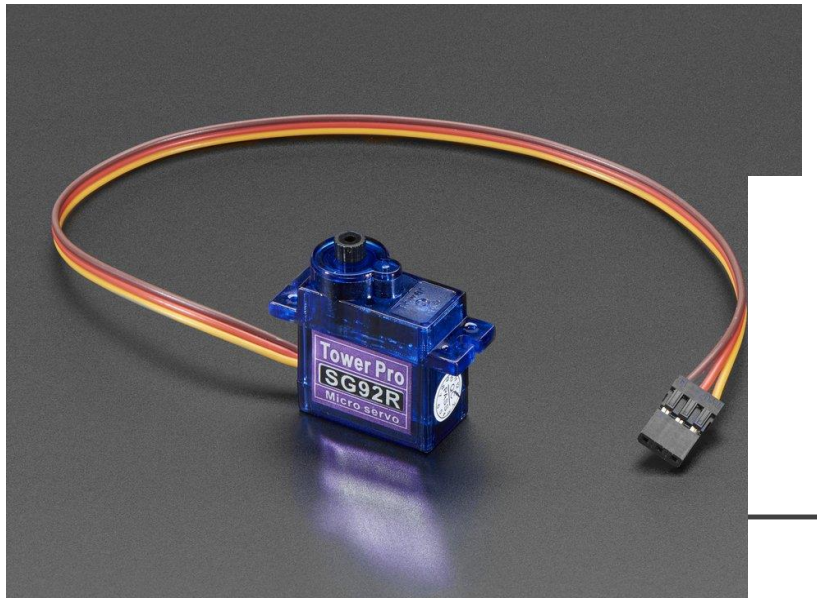
VARIABLE RESISTOR

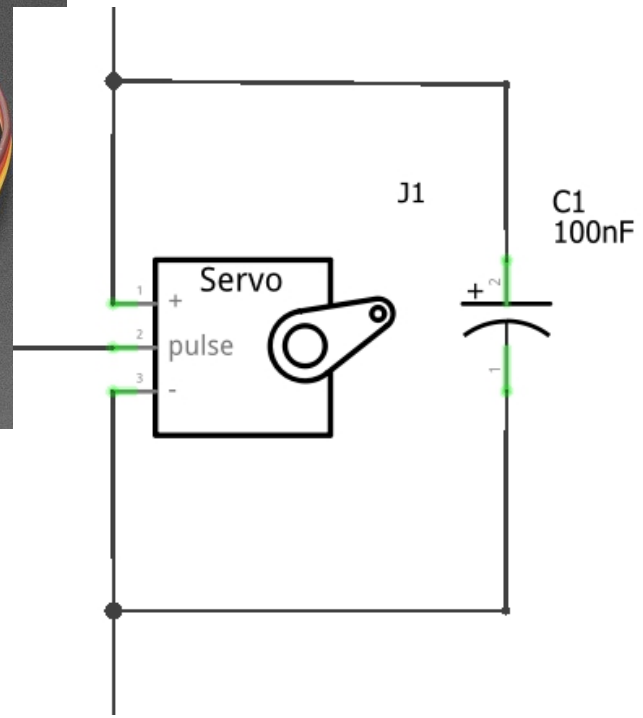
- FSR's are basically a resistor that changes its resistive value (in ohms Ω) depending on how much its pressed.



SERVOS

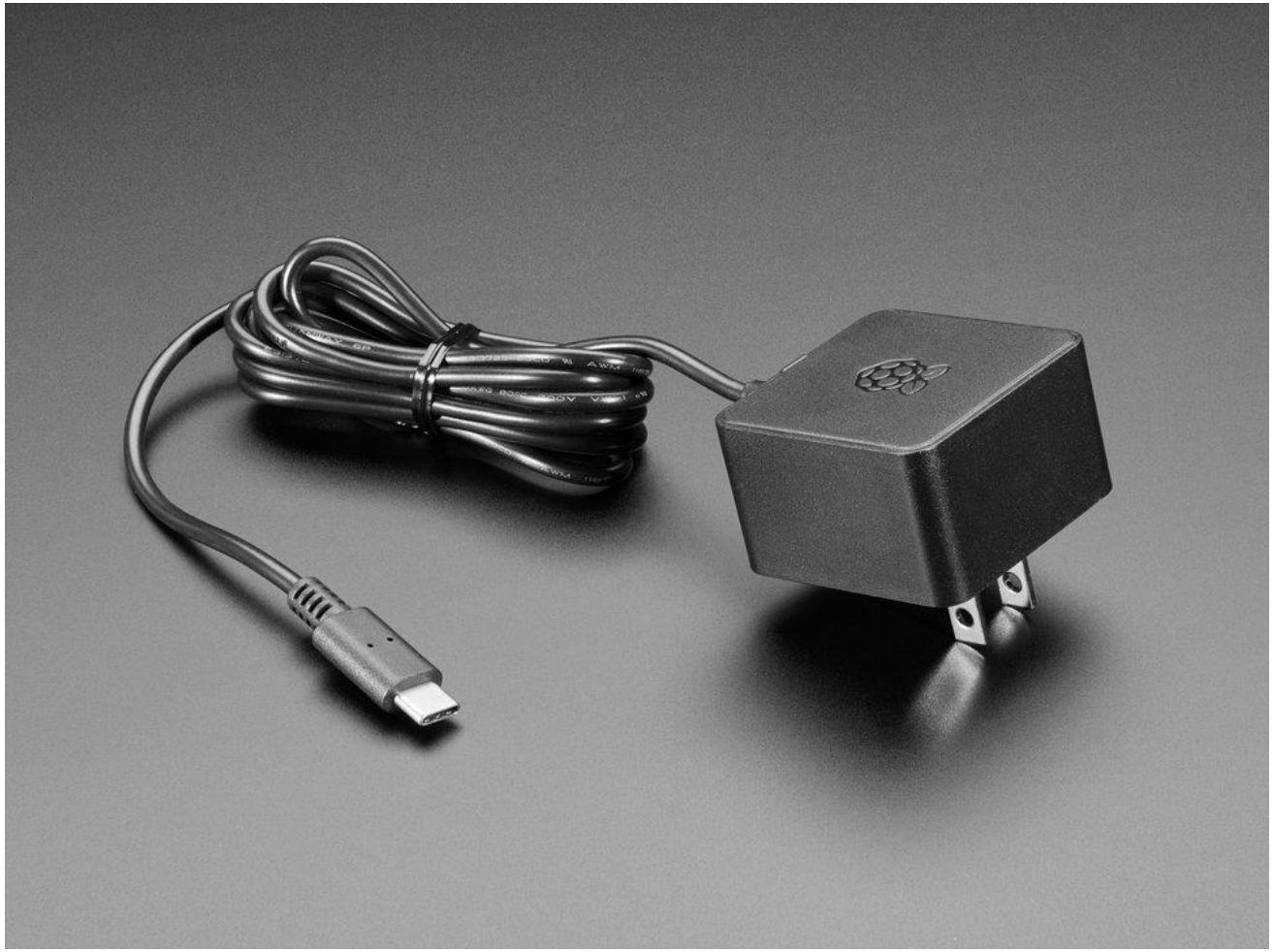
- The servo can rotate approximately 180 degrees (90 in each direction).
- Wired with a decoupling capacitor.
- Adding a decoupling capacitor allows some of the instantaneous current to come from the capacitor during servo startup instead of the power line, thus keeping the power line glitch free.





POWER SUPPLY

- The system will be powered from wall power so that when in 'learning mode' the system can run continuously unattended.



HOW TO DETECT MOVEMENT?

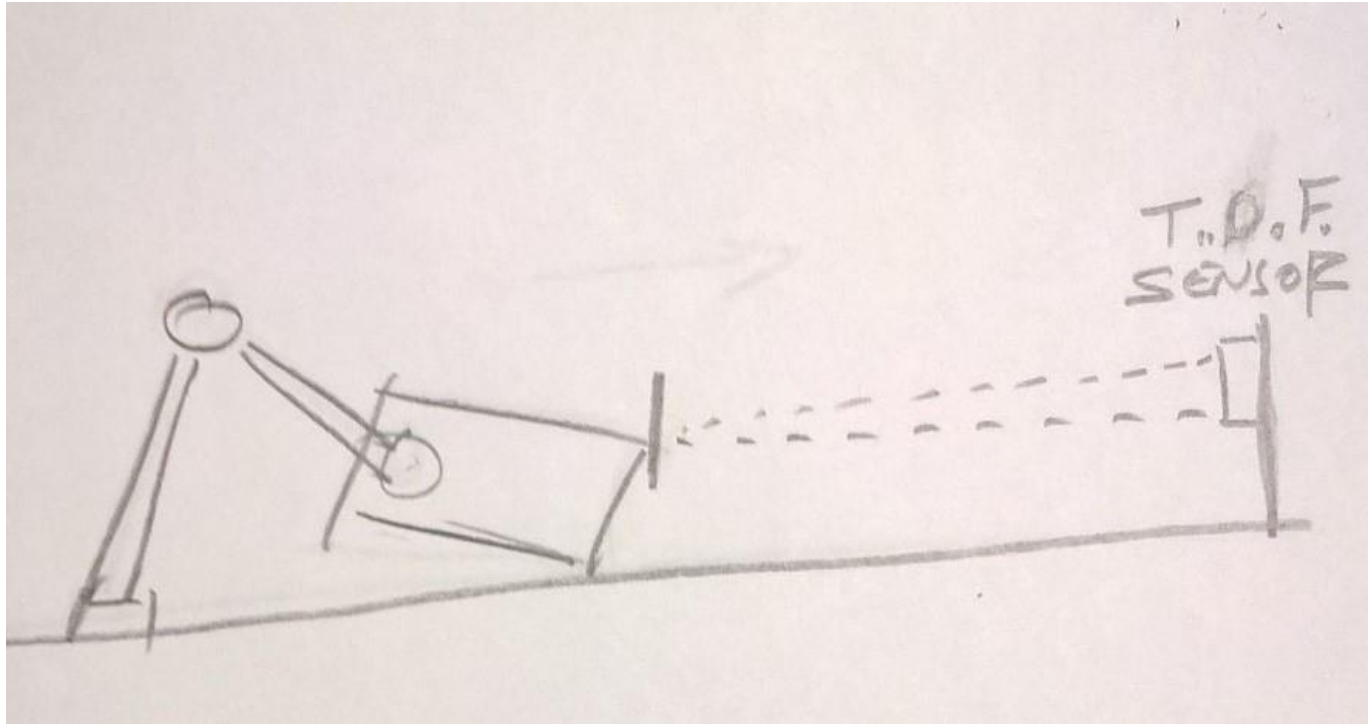
- The robot is constrained to move along a track. A force sensor can be used to detect when the appendage has touched the 'ground'.
- A Time of Flight Distance Sensor can be used to detect robot movement.

TIME OF FLIGHT SENSOR DEMONSTRATION

- https://www.adafruit.com/product/3316
- The sensor contains a very tiny laser source, and a matching sensor.
- The VL6180X can detect the "time of flight", or how long the laser light has taken to bounce back to the sensor. S
- ince it uses a very narrow light source, it is good for determining distance of only the surface directly in front of it.

REWARD SENSOR

- The Time Of Flight sensor will be mounted at the end of the rail system from where it will measure the position of the robot after each 'learning' instance.



RESET

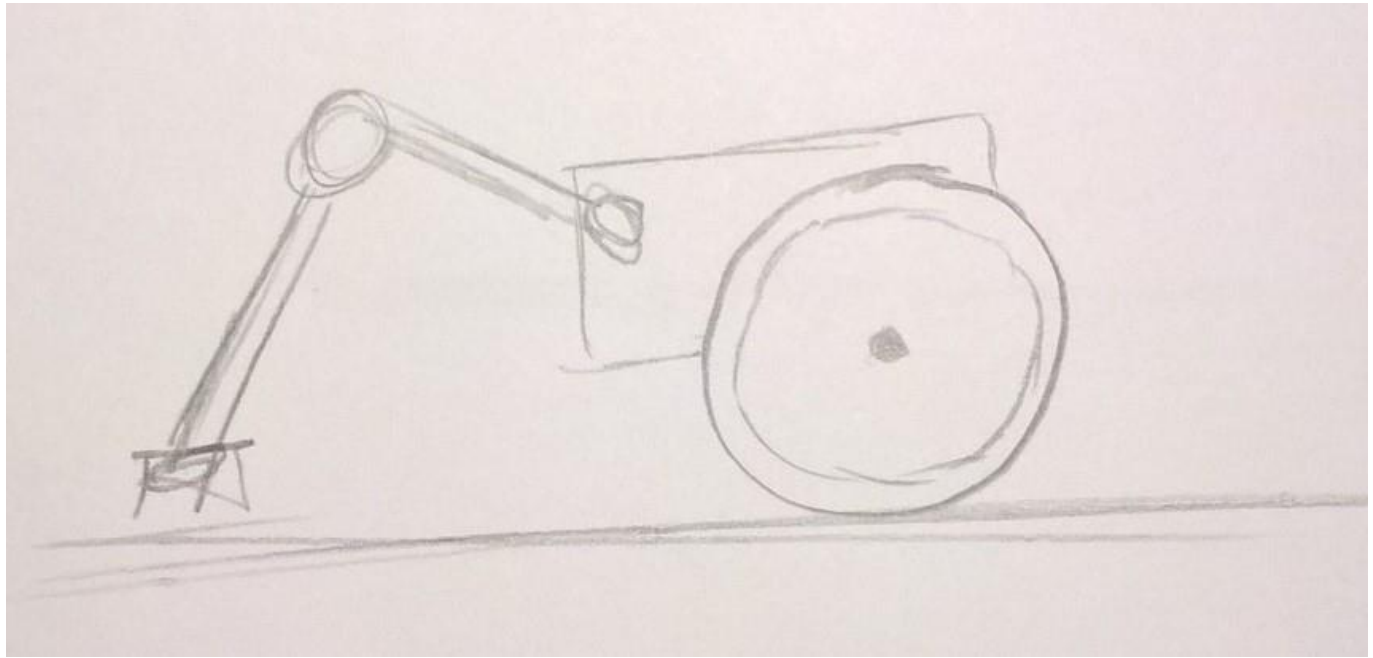
- It may be necessary to include a mechanism to 'center' the robot after some number of movement attempts cause the robot to run out of room for training.

CONVERGENCE

- The T.O.F. sensor will be used to turn the robot off when it moves completely across the training field. (If the reset/centering mechanism is not available.)

WHEELED BASE

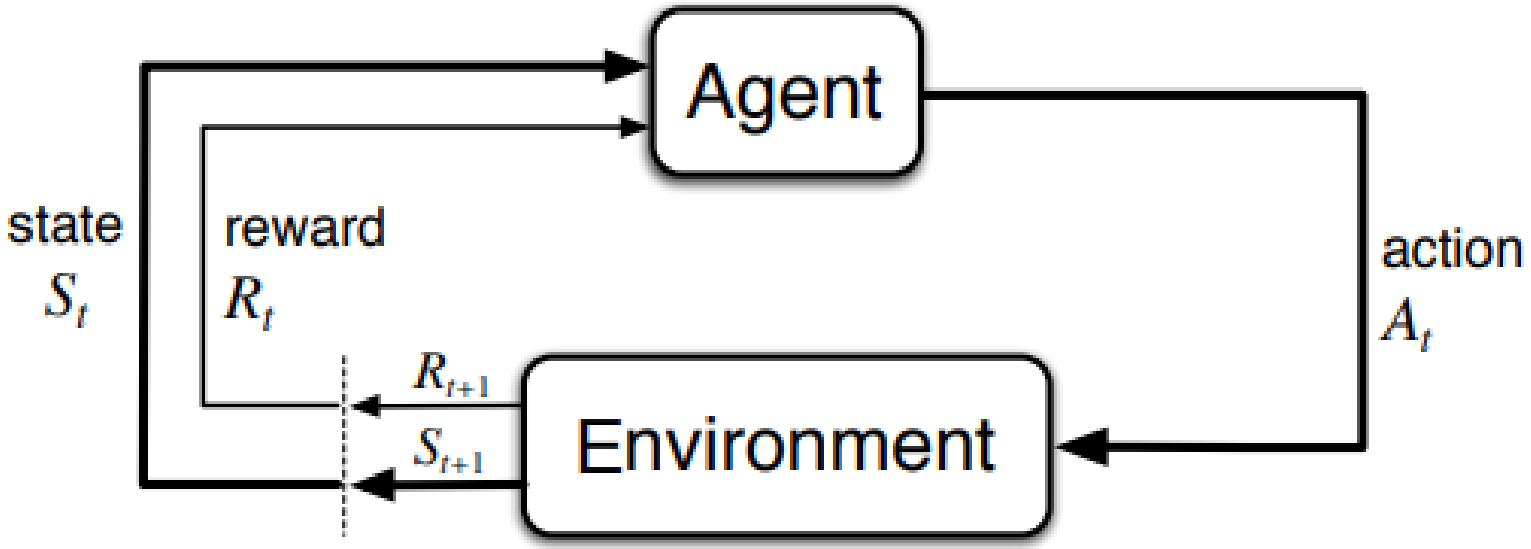
- Perhaps the robot can be removed from the training fixture and secured to a wheeled base to allow it to move without restriction.



REWARD

- Q-learning and reinforcement learning, in general terms, is about selecting an action that gives the maximum reward.
- Key to the reinforcement learning process is that rewards can not be arbitrarily assigned by the environment or changed by the agent (robot) operating in the environment.

REINFORCEMENT LEARNING CYCLE



TERMS

- **Agent** : Software program that make intelligent decisions and continues to 'learn'. The agent interacts with the environment by taking an action and receiving a reward based on the action.
- **Environment**: Contains the demonstration of the problem to be solved.
- **State** : The position of the agent at a specific time in the environment. Whenever an agent performs an action- the environment responds by providing the agent with a reward and a new state. State can be anything which can be useful in choosing actions.

TRAINING SYSTEM COMPONENTS

- Environment
  - The reinforcement learning training system. The robot sitting on a trolley between two guide rails.
- State
  - The measurement obtained from the force sensor at the end of the appendage.
  - The change of distance read by the Time Of Flight distance sensor.
- Action
  - The position of each of the appendage servos.

ACTION REFINED

- Each servo can be programmed to one of 255 possible positions or about .7 of a degree. Considering 2 servos, this yeilds 65,536 possible states.
- We can reduce the number of states by limiting each servo to 90 degrees of movement. (32,768 states)
- If learning doesn't converge on a solution with this restriction, the servo precision can be further reduce to increments of 5 degrees (instead of the default .7). Resulting in 4,681 states.

- After making a move during learning, the Q value for a given state and action is replaced the new value

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

LEARNING RATE (OR STEP SIZE)

- The new value is a sum of two parts. The first part is (1-learning rate)*old value.
- A learning rate of 0 will mean nothing new will be learnt.
- A learning rate of 1 will mean the old value will be completely discarded.
- Note: In practice a learning rate of 0.1 is a typical value.

DISCOUNT RATE

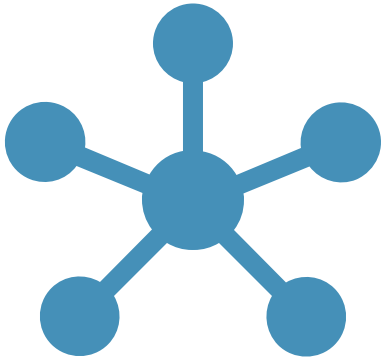
- The discount factor (gamma) determines the importance of future rewards.
- A factor of 0 will make the agent "myopic" (or short-sighted) by only considering current rewards.
- A factor approaching 1 will make it strive for a long-term high reward.

EXPLORATION FACTOR

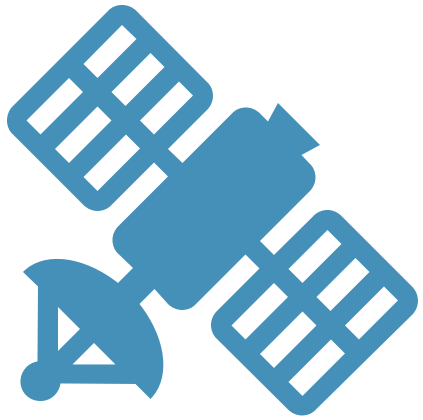
- The exploration are important to reinforcement learning systems. If the system keeps selecting the max reward each time, then it will keep performing the same action and will not try anything else. The system will never learn that another untried action has a better reward than this.
- An exploration factor makes the algorithm select a random action a predetermined % of times.

LEARNING RAGE AND EXPLORATION

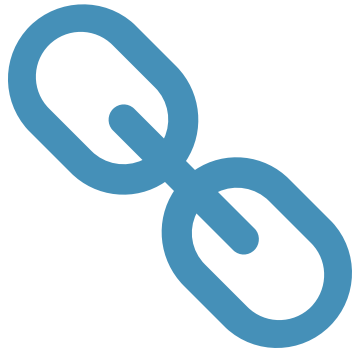
- a



Network



Satellite

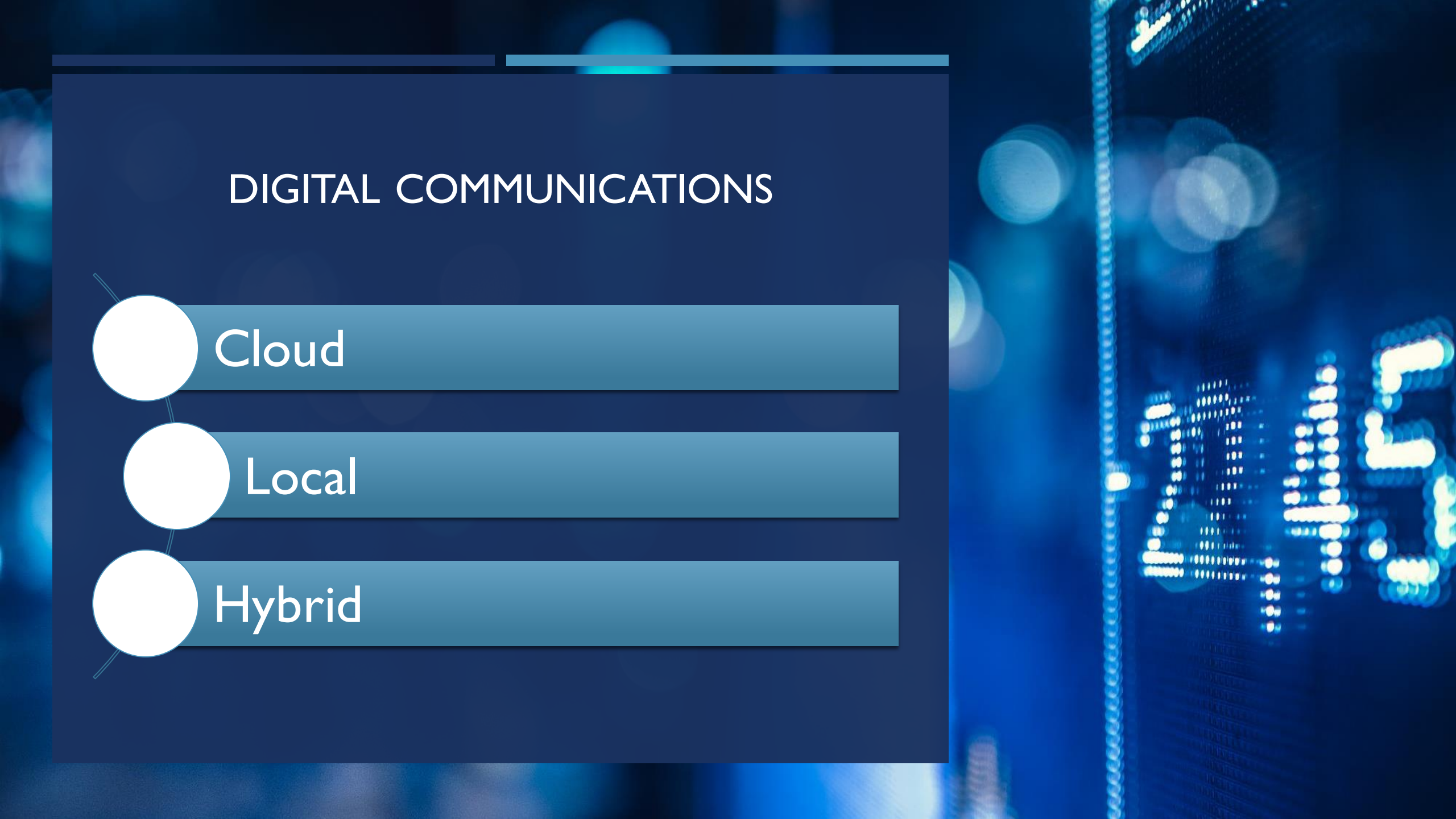


Link

TECH REQUIREMENTS

DIGITAL COMMUNICATIONS

- Cloud
- Local
- Hybrid



THANK YOU

SOMEONE@EXAMPLE.COM