**ENGEN582-24X**

**Honours Research and Development Project**

**A Review of the Actuation, Sensing and Control Methods for Humanoid Robotic Arms**

**Kieran Smith**

**Supervisors:**

**Chi Kit Au & Mahonri Owen**

## Table of Contents

[Table of Contents 1](#_Toc165932520)

[1. Introduction 3](#_Toc165932521)

[2. Overview of humanoid robotic arm technology 4](#_Toc165932522)

[2.1 Uses of Humanoid Arms 4](#_Toc165932523)

[2.2 Review of Humanoid Robotic Arm Degrees of Freedom and Strength 4](#_Toc165932524)

[2.3 Review of Humanoid Robotic Arm Accuracy 5](#_Toc165932525)

[3. Humanoid Arm Actuators 5](#_Toc165932526)

[3.1 Actuator Requirements 5](#_Toc165932527)

[3.2 Electric actuators 6](#_Toc165932528)

[3.2.1 Brushed DC Motors 6](#_Toc165932529)

[3.2.2 Brushless DC Motors 6](#_Toc165932530)

[3.2.3 Brushless DC Servo Motors 7](#_Toc165932531)

[3.2.4 Brushed Servo Motors 7](#_Toc165932532)

[3.2.5 Solenoids and Stepper Motors 7](#_Toc165932533)

[3.2.6 AC Motors 7](#_Toc165932534)

[3.3 Hydraulic Actuators 7](#_Toc165932535)

[3.4 Pneumatic Actuators 8](#_Toc165932536)

[4. Humanoid Robotics Arm Sensors 8](#_Toc165932537)

[4.1 Joint Level Sensors 8](#_Toc165932538)

[4.1.1 Joint Positional (Angular) Sensors 8](#_Toc165932539)

[4.1.2 Joint Torque Sensors 10](#_Toc165932540)

[4.2 Tactile Sensors 10](#_Toc165932541)

[4.2.1 Capacitive and Resistive Tactile Sensing 10](#_Toc165932542)

[4.2.2 Accelerometer, Temperature Sensor, Magnetic and Barometric Tactile Sensing 11](#_Toc165932543)

[4.2.3 Optical Tactile Sensing 11](#_Toc165932544)

[5. Humanoid Arm Control Systems 12](#_Toc165932545)

[5.1 Control Hardware 12](#_Toc165932546)

[5.1.1 Low Level Controllers 12](#_Toc165932547)

[5.1.1.1 Use of Arduinos as a Low Level Controller 12](#_Toc165932548)

[5.1.1.2 Use of Raspberry Pis as a Low Level Controller 12](#_Toc165932549)

[5.1.1.3 Other Low Level Control Hardware 13](#_Toc165932550)

[5.1.2 High Level Controllers 13](#_Toc165932551)

[5.2 Software Environment 14](#_Toc165932552)

[5.2.1 Operating System (OS) 14](#_Toc165932553)

[5.2.2 Middleware 15](#_Toc165932554)

[5.2.2.1 Robotic Operating System (ROS) 15](#_Toc165932555)

[5.2.2.2 Open source Robotic Technology (OpenRTM) 15](#_Toc165932556)

[5.2.2.3 Yet Another Robotics Platform (YARP) 15](#_Toc165932557)

[5.2.2.4 Other Robotic Middleware 15](#_Toc165932558)

[5.3 Humanoid Arm Control Methods and Algorithms 16](#_Toc165932559)

[5.3.1 Control Input 16](#_Toc165932560)

[5.3.2 Low Level Control Algorithms 16](#_Toc165932561)

[5.3.3 Higher Level Control Algorithms and Inverse Kinematics 17](#_Toc165932562)

[6. Summary 18](#_Toc165932563)

[References 19](#_Toc165932564)

[Appendix – Project Propossal 22](#_Toc165932565)

## 1. Introduction

This literature review has been undertaken as part of an honours engineering project at the University of Waikato to develop a pair of humanoid robotic arms. The project’s aim is to produce a pair of arms that can play the piano as a demonstration of their dexterity and control accuracy. The project focuses on the mechanical, electromechanical and low level control of the arms, there are no plans to develop high level control of the arm for autonomous or semi autonomous tasks. This is left as a latter exercise so the piano playing will likely be done mostly via record and play functionality using a digital twin. The aim of this literature review is to develop an understanding of the latest advancements in areas relevant to the actuators, sensors and control methods of humanoid robotic arms. A companion review focuses instead on the mechanical design and joint mechanisms. Areas covered in this review include: humanoid arm applications and state of the technology, what actuators are used in humanoid arms and their characteristics, what tactile and joint sensors are used, the control electronics used, the operating systems and middleware used and finally, a brief overview of low level and higher level control algorithms. The literature review only has a minimal review of arm control input methods (such as control gloves) as this is less relevant to the project and only briefly touches on soft robotics where relevant as this is a complex subject beyond the scope of the arm being considered. Other sensors that are not tactile or joint sensors are out of scope (such as cameras and microphones) as there are not plans to implement these on the planned arm. For the same reasons human-robot interaction and autonomous robotic behaviour are not considered, nor is the generation of naturalistic arm movements for aesthetic reasons. Finally, robotic arms are that are not humanoid (made to look and move with the same motions as the human arm) are not considered in order to keep the scope of the review small and relevant.

The following Section 2 provides an overview of the uses and relevance of humanoid arms and focusses on the current state of the art including how accurate, strong and dexterous are the best humanoid arms to date. This is followed by section 3 on humanoid robotic arm actuators, which considers what actuators have been used in the literature and what their relevant characteristics are. Section 4 then considers the types of joint position and torque sensors typically used, including their prevalence and accuracy. Section 5 then considers the control of a humanoid arm, firstly what electrical hardware is used: low level and high level programable controllers, then secondly what software is generally run on this hardware including the operating systems, middlewares, and actual control algorithms. This is concluded by section 6 which provides a summary of the findings in the review, including the gaps in the literature.

## 2. Overview of humanoid robotic arm technology

### 2.1 Uses of Humanoid Arms

Humanoid robotic arms have many uses, but the human arm is not the optimal topology for a robotic arm so they only have uses where the arm’s similarity to a human arm is useful. One of the most obvious is the applications in prosthetics (Aljalal et al., 2020), where the ultimate aim is to allow the user to have control of a robotic arm that replicates a human one to the same degree that humans have control over their own arms. Another key operation is teleoperation such as the teleoperation arm from Procter and Secco (2021). Such teleoperation robots allow a skilled human to operate the robot remotely without having to travel (if there are few personnel with the appropriate skills for the task), which costs money and potentially downtime, and without potentially putting themselves in danger. Potential application areas for the later point include in rescue operations and in the nuclear, mining and heavy industries. There are also robots that have been developed as sign language interpreters or teachers (Meghdari et al., 2018), where dexterity and humanoid looks are paramount, though these appear to currently lack sufficient abilities to be useful. A fourth application is that for robots whose primary purpose is human interaction, humanoid robots can make the robots appear more friendly and personable An example of this is the human interaction robot from Sakhineti & Jayabalan (2020) which is designed to provide companionship and assistance to the elderly (including those with dementia) and those with mental disabilities. Finally, humanoid robots allow for the straightforward replacement of humans in tasks and environments that have already been designed for humans. For example, in low skill factory jobs currently done by humans that require high flexibility (Ali Ahmad Malik et al., 2024). In many of these situations a non-humanoid robot would be more effective, but a humanoid robot has the advantage that it does not need to be custom designed to undertake tasks that were previously done by humans (Ali Ahmad Malik et al., 2024).

### 2.2 Review of Humanoid Robotic Arm Degrees of Freedom and Strength

Extensive research has been undertaken on humanoid arms in recent years and they can be evaluated by various metrics. The first is the degrees of freedom in the joints, and how well the joints correspond to the human joints. Of the well-funded top of the line robots, nearly all have a 6 or 7 degree arm (including the wrist) (Stasse & Flayols, 2018), which very closely matches the joints in the human arm. However almost none come close to the number of actuated (not including flexible joints that can be controlled) degrees of freedom (DOF) in the human hand, generally listed as 21 actuated degrees of freedom (Breen et al., 2003)(Rahman & Al-Jumaily, 2013), which includes 3 extension and flexion motions in the fingers, an abduction/adduction motion in the fingers and 5 DOF in the thumb. The most dexterous robots towards this end are robots such as the Walkman robot from the Istituto Italiano Di Tecnologia (IIT) which has 19 DOF per hand (Stasse & Flayols, 2018) and the ShadowHand which has 18 actuated DOF per hand (Shadowrobot, n.d.)(not counting the DOF in the wrist). There are also non-actuated DOF of which the shadow hand has an additional 4. These are clearly a very similar numbers to the DOF to the human hand however hands like the ShadowHand are considerably bulkier than a human forearm and don't have anywhere near the same strength as a human hand: it can only hold a maximum of 5kg (Shadowrobot, n.d.) and the Walkman robot can only hold a maximum payload of 10kg (Davis, 2018). There is considerable uncertainty in these numbers as well as it is not specified in what grip type the robots can hold these loads. This highlights a technological gap as there are no robotic arms in existence that have both the same dexterity and strength as the human arm, as 10kg is far below what a strong adult human could lift and manipulate. Generally there is a tradeoff between gripper strength and gripper dexterity, so most robotic arms have considerably under actuated hands, which are also cheaper and less complex (Stasse & Flayols, 2018)(Asfour et al., 2019)(Prabhu et al., 2021)(Naoki Fukaya et al., 2013)(Kaneko et al., 2019)(Tatsch et al., 2017). Many of these arms also hold similar payloads like dual arms from Shut & Hollis (2019) which can hold 6.8kg using the fingers of a hand, but up to 10kg if the weight is attached to the end effector. Another robot with similar weight capabilities is the full humanoid robot from Kaneko et al. (2019) which can lift 11kg gypsum boards. The arms with higher DOF at low price points are often sign language robots such as from Meghdari et al. (2018) which has 9 actuated DOF in each hand, in these types of robots there is very little torque in the fingers as they are not designed to grip objects. Most robots at a low price point have actuated DOF in the hand less than 5. This does highlight another gap in the literature where there are few hands with a moderate to high degree of freedom and moderate strength at a low price point, and is something we hope to address with our robotic arm.

### 2.3 Review of Humanoid Robotic Arm Accuracy

Another metric by which robotic arms can be evaluated is by how accurately they can move and manipulate objects. Unfortunately a broad overview of data on this topic for humanoid robots is missing in the literature and data is difficult to come by as most papers on individual robots quantify their accuracy in unique ways that are not easily comparable or entirely neglect to quantify the robot’s accuracy. A review on the accuracy of hydraulic robots has been done by Mattila et al. (2017) in which the most accurate robot was said to have an accuracy of 1.5mm at the end effector, however the reference listed did not provide any support for this. Other hydraulic robot end effector accuracies from this review were listed as between 5.2-120mm. The low cost hobby brushed servo arm from Yang et al. (2019) listed an accuracy of +/-10mm when stacking blocks which was attributed to gears and part precision, which is relatively accurate considering that the actuators used do not appear to be robotics grade. Industrial robotic arms, while not the focus of this review, provide a good indication of achievable accuracy where data for humanoid arms is lacking, and accuracy for these can be as high as +/-0.1mm for some high quality electric actuator arms (Lattanzi et al., 2020). +/-0.1mm is clearly sufficient to replicate the accuracy of a human arm, but most humanoid arms are unlikely to be this accurate due to space constraints limiting the size of the actuators and structure of the robot. Hence it is unclear how common high accuracy is in humanoid robots.

## 3. Humanoid Arm Actuators

Arguably the most important electrical components in a robot are the actuators. They determine the strength of the robot, the size, a considerable fraction of the of the robot’s weight, the positional accuracy of the joints, and the safe duty cycle.

### 3.1 Actuator Requirements

The key requirements are generally a high power to volume ratio and low speed-high torque operation (Stasse & Flayols, 2018). For robots that are mobile, power to weight ratio is also important (Shut & Hollis, 2019) as lower weights allow for a faster movement and lighter, cheaper robots. Another example where weight is important is of course prosthetics. Even for non-mobile robotics such as this project, lower weights are desired as it increases safety when interacting with humans and increases lifting capacity. Backdrivability (which is not possible for very high gear ratio gearboxes or worm drives for example) is also desired characteristic (Stasse & Flayols, 2018): it allows for the robot to be robust to collisions, to interact with humans in a more natural way and for better control of torque when torque sensing on the motor side. It is not however a requirement as collision robustness can be achieved via torque limiting mechanisms or via flexible couplings (Tatsch et al., 2017) and torque can be managed by sensing torque on the output side (where it is unaffected by the inability to backdrive). They should be robust (Shut & Hollis, 2019) and for our purposes relatively low cost.

Leaning towards the control requirements for actuators, the actuators must be capable of accurate position control (Shut & Hollis, 2019) in a robotic arm due to the compounding effect of having so many DOF, which can either be achieved through embedded sensors or through sensor feedback from the endpoint of the actuator. Ideally they should also have some sort of torque control as this allows for differing torques for grasping different objects and for limiting torque when working with humans to increase safety. This review will focus mainly on these control characteristics across the spectrum of different actuators available.

### 3.2 Electric actuators

This review has shown that electric actuators appear to be the most popular choice for humanoid arms. This is due to their simplicity, low cost, high degree of available options and ease of control (Stasse & Flayols, 2018). They are most commonly DC motors, either brushed or brushless, but AC motors are also used (Stasse & Flayols, 2018) as they can offer higher performances despite the need to transform DC battery voltages for mobile robots. DC motors can be either standalone or servo versions (brushed or brushless) that have integrated position feedback and sometimes torque feedback or control. Finally stepper motors are also used, which are fundamentally the same concept as brushless DC motors but with more poles, and solenoids are another option that is rarely used in humanoid robotics due to its limited positional control.

#### 3.2.1 Brushed DC Motors

Brushed DC motors are used in some robots such as Meghdari et al. (2018), a sign language robot and in some joints of the full humanoid robot from Asfour et al. (2019). They are usually purchased as a gearbox-motor combination, where the gearbox is usually planetary. They offer the advantages of low costs and relatively straightforward combined speed-torque control via PWM or voltage adjustment. To get position control they require extra rotational displacement sensors and a position - motor power feedback loop which is commonly PID (Meghdari et al., 2018) which is their main disadvantage. They offer medium power to weight and power to volume ratios, mainly due to space not being wasted with inefficient sensor placement like it is in servos.

#### 3.2.2 Brushless DC Motors

Brushless DC (BLDC) motors as standalone units offer the advantage of the highest power to weight and power to volume ratio of any electric actuator however the trade-off is control difficulties (Procter & Secco, 2021) and higher costs. They are used in robots such as the full humanoid from Asfour et al. (2019), where they allow the robot to have very high power joints. They are also used in the teleoperation robot from Procter and Secco (2021), which allows for the use of high powered arm joints powered by drone motors with added encoders. In this robot they were controlled using an open source motor driver called Odrive. No overheating issues were mentioned in the paper but the use of drone motors under high loads tends to incur heating issues as they are designed for low torque, short (7-30 min) duty cycles and to have large amounts of cooling from the attached propeller forcing air through the motor (Paul, personal communication, March 5, 2024). The lack of heating issues could have been because the motors were only tested for short periods of time as the arm was not fully operational. BLDC motors come in either inrunner or outrunner versions, where outrunners have the rotor on the outside and vice versa for inrunners. Generally for robotics applications outrunners are used as they offer higher torques and lower speeds (Procter & Secco, 2021). They are also sold either with or without embedded hall effect sensors. Control of the motors at stall or low speeds or stall is realistically possible only with sensored versions (though of course sensors can be added), as the back EMF effect needed in sensorless versions to determine which poles to apply a voltage to only works when the motor is spinning. These sensored versions generally cost more and are not available for low power ratings which limits the applications of BLDC motors to the larger joints in a robotic arm. In the full humanoid robot from Asfour et al. (2019) they are controlled using feedback from an absolute encoder, an incremental encoder and a torque sensor, which allows for very precise control, but at a high cost. It is likely the second incremental encoder is just used here to increase accuracy as in theory a single absolute encoder would be sufficient for positioning.

#### 3.2.3 Brushless DC Servo Motors

Likely the most common actuator for robotic arms is instead the BLDC servo motor. These are generally robotics grade versions (though hobbyist, RC vehicle grade servos are also used) and the most common are those from Dynamixel used in robotic arms from Hayosh et al. (2020), Meghdari et al. (2018), Tatsch et al. (2017), and Mick et al. (2019). These motors have high power to weight and power to volume ratios, have position control with integrated PID controllers, have torque controller or torque sensing, and continuous rotation. These are typical characteristics for motors of this type, though not all will have torque control or torque sensing. The Dynamixel motors in particular also can be chained together to minimise cabling and come with extensive robotics software to make them easy to use (Mick et al., 2019). These Dynamixel motors use serial communication and in the literature examples are controlled via a Raspberry Pi (Mick et al., 2019) or a miniature computer (Tatsch et al., 2017). Of course, other brands of similar BLDC servos are also used for robotic arms for example in the arms developed by Shut & Hollis (2019), which make use of large diameter cylindrical servos as the function of their arm is not dependent on the arm appearing humanoid (and these large motors decrease how much the harm looks humanoid).

#### 3.2.4 Brushed Servo Motors

Brushed DC servo motors are a cheaper alternative to BLDC servos that offer many of the same advantages and multiple humanoid robotics projects built on lower budgets use these motors such as a robotic arm from Yang et al. (2019) and a robotic hand from Prabhu et al. (2021). They also come in robotic grade and hobbyist grade versions; however robotics grade servos appear to be more likely to be BLDC. They have a lower power to volume and power to weight ratio than their BLDC counterparts. And while they do have position position control, the hobbyist grade versions used in the mentioned robots lack any sort of torque control, torque sensing or ability for the user to customise the position control algorithm. They don't usually come in versions where the wiring can be chained together either.

#### 3.2.5 Solenoids and Stepper Motors

Other DC electric actuators include solenoids and stepper motors. Solenoids find little use in robotic arms as generally they don't have any position control. Stepper motors are not commonly used either, most likely as they are not precise enough to replace high quality servos, and not powerful enough to replace standalone DC motors.

#### 3.2.6 AC Motors

AC motors can deliver high amounts of power, however the trade-off is more complex control (Stasse & Flayols, 2018). This is because their torque rapidly drops off at lower speeds and they tend to have low torque on startup. Some robots do use them however, such as the TORO humanoid robot, which uses torque sensors to make the control more feasible (Stasse & Flayols, 2018).

### 3.3 Hydraulic Actuators

Hydraulics are another alternative to electric actuators and are used in many high performance robots such as the Boston Dynamics Atlas robot (Stasse & Flayols, 2018). They offer better power and torque per kg and per cm3 than electric motors (Stasse & Flayols, 2018) (Junget al., 2018). They also offer good control of output force through the use of variable pressure (electric) control valves (Stasse & Flayols, 2018). Unfortunately, they also come with significant downsides. The most important of these is their high costs (Saeedvand et al., 2019), their tendency towards leakage problems (Kaneko et al., 2019) and greater difficulty in position control (Mattila et al., 2017). Mattila et al. (2017) postulates that position control can be difficult as in many examples of hydraulic robots the position of a hydraulic actuator is intimately and non-linearly linked to the force on the actuator, which induces difficulties especially when the actuator is interacting with objects (and the force is being changed). Error at the end effector in a range of 6 research arms from 2002-2015 was found to be mostly between 1.5-27mm with one arm having an error of 120mm, though vastly different working velocities are used for many of the arms which would have an impact on the accuracy (Mattila et al., 2017). So while the control may be difficult (indeed many of the arms use nonlinear model based control systems and there are many different controller methods used) moderately high accuracy is clearly achievable, if at a higher cost and complexity. Other downsides to hydraulic arm systems include the large size of hydraulic pumps which is a limitation for mobile robots (Stasse & Flayols, 2018), the loud sound of hydraulic pumps (Stasse & Flayols, 2018)(Suzumori, 2020) and the low energy efficiency (Mattila et al., 2017). Energy efficiency is important to reduce operating costs, environmental impact and for mobile robots: battery life. This efficiency is a key limitation of hydraulics not considered by Stasse & Flayos (2018) in their overview, however they do consider how pump size limitations for mobile robots can be overcome by well designed pumps such as from Alfayad et al. (2011).

### 3.4 Pneumatic Actuators

Similar to hydraulics are pneumatic actuators. These offer the advantage over hydraulics of less concern over leakage but it is not as easy to control thier position (Hashimoto, 2020) and torque (Stasse & Flayols, 2018). They also generally offer faster actuation than either hydraulic or electric actuators (Mori et al., 2018). One of the most common pneumatic actuators for robotic arms is the Mac Kibben muscle (Tondu, 2012), which has a form factor reminiscent of human muscles and thereby can allow for more humanoid movements and actuator placements. Tondu (2012) further highlights the difficulties associated with accurate control of this “muscle” in their paper, due to such issues as the non-linear stress-strain relationship of the elastomer used in the muscle, and the muscle hysteresis. Pneumatic actuators are also the key actuators for soft robotics (Su et al., 2022) which is an important emerging field and especially relevant to humanoid arms that will interact with humans. For these soft robots powered by pneumatics, either muscle type actuators connected to cables are used, or soft fillable voids are used (Su et al., 2022). These soft robots offer superior resistance to damage, and safety over traditional solid robots. Robots such as the badminton robotic arm from Mori et al. (2018) also make use of pneumatics as this allows for the fast actuation needed to play badminton. This team also managed relative accurate control on the order of 10-40mm using fast swings using feedforward control.

## 4. Humanoid Robotics Arm Sensors

In addition to the discussed actuators, another key part of a robotic arm are the sensors implemented. These can be broadly broken into two categories: joint sensors and environmental sensors. Joint sensors are used on each joint and include sensors for position (and thereby speed and acceleration) and sensors for torque. Environmental sensors are widely varying but include sensors such as cameras, microphones, tactile sensors and time of flight cameras or scanners. This review will only focus on joint sensors and tactile sensors as these are the only sensors that will be implemented on the planned robotic arm.

### 4.1 Joint Level Sensors

#### 4.1.1 Joint Positional (Angular) Sensors

Joint sensors for sensing position (rotational displacement) can be either standalone components added in to allow for position control of standalone motors, or they can come preinstalled in servo motors. They are absolutely key to a useful robotic arm as small errors in the position of one joint compound to large errors in the end effector position. They can also allow for damage detection (Stasse & Flayols, 2018). Servo motors will then have a basic controller inside them that will allow for (at a minimum) position control through the use of a joint position sensor feedback loop. Clearly this is a key advantage of using servo motors over other types, and the accuracy and speed of their control will depend on the servo selected. Regardless of whether they come preinstalled on a servo or not, angle sensors generally come in two forms: potentiometer based or encoder (hall effect or magneto resistive) based. These are not the only possibilities however and another commercially available alternative are inductive sensors. Capacitive and optical angle sensors have also been developed to a high accuracy (Kumar et al., 2021)(Figure 1) but their commercial availability appears to be severely limited. Potentiometer based sensors rely on the variable resistance of a sliding connector that rotates to different positions along a wire, whereas encoders rely on the changing magnetic field of one or multiple magnets that rotate. Hall effect based encoders are the most common angle sensor to apply to a BLDC motor to sense its angle (Kumar et al., 2021) and they come in either absolute or incremental versions (incremental versions can only sense the change in angle from the starting angle). Most robots use servo motors which avoids the need for additional sensors, but the custom motor-sensor combinations used in the ARMAR-6 and the HRP-5P full humanoid robots both use an absolute and an incremental encoder on most joints (Asfour et al., 2019)(Kaneko et al., 2019). Another example of encoders in use is on the sign language robot from Meghdari et al. (2018), in order to maximise finger DOF brushed DC motors have been used instead of servos (as servos are larger) and the joint angle sensing on the fingers has been achieved by using a hall effect sensor coupled with a diametrically magnetised magnet. This is essentially a custom encoder and has been made in this way due to the minimal space on the fingers for adding sensors. A similar design using 17 tiny hall effect sensors and ring magnets (Figure 2) is employed for sensing the angles of the phalangeal joints in the highly dexterous iCub robot (Schmitz et al., 2010). Using sensors on the joints, which are actuated by cables in both robots, increases accuracy over rotation sensors on the motors used to pull the cables as joint sensors are not affected by cable flex and slack. On-axis absolute encoders are also used in the custom actuators on the medical teleoperation arm from Procter and Secco (2021). In terms of servo motors, potentiometer sensors are commonly used for hobby brushed servos (Adafruit, n.d.) while BLDC servos are more likely to use absolute encoders, for example on the Dynamaxil range of BLDC servos (Dynamixel, n.d.).

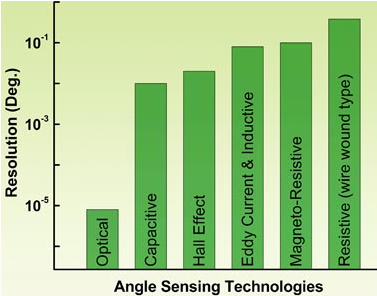


Figure 1 - Best accuracy of different types of joint sensors, from Kumar et al. (2021).

A close up of a coin

Description automatically generated

Figure 2 - Joint angle sensor used on the phalangeal joints on the sign language robot from Schmitz et al. (2010). Left is the neodymium magnet used, middle is the hall effect sensor used and right is a 2 cent Euro coin (18.75mm) for scale.

#### 4.1.2 Joint Torque Sensors

Torque sensing is another important aspect of joint sensors as it allows for the detection of objects and, if coupled with torque control, allows for the robot to apply appropriate torque for interacting with objects. Hobbyist servo motors do not typically have torque sensing, however BLDC servos mostly have torque sensing in built, though they do not necessarily have variable torque output. Torque can be sensed based on a strain gauge such as the custom design used in the ARMAR-6 humanoid (Asfour et al., 2019), and in the BLDC servos commonly used in many robots (Shut & Hollis, 2019). Torsional springs with angular displacement sensors can also be employed as in some of the actuators on the full humanoid from Tatsch et al. (2017). It is not a requirement to have censored torque sensing however and many complex robots such as the humanoid TOCABI do not use them (Schwartz et al., 2022). If specific sensors are not employed then torque can also be estimated based on sensing the current going to the motors (this is less accurate due to non-linear rotational friction and changing temperature response).

### 4.2 Tactile Sensors

Tactile sensors are an area of ongoing research, especially in regards to large area sensors that mimic the capabilities of human skin (Roberts et al., 2021)(Pagoli et al., 2022). While tactile sensors are potentially useful on the external surface of any part of a humanoid robot, the most important locations to consider are the hands, specifically on the top pads of the fingers and thumb and on the top of the palm (closest to the proximal phalanges) (Weiner et al., 2019). They are important as they allow for the force being applied to objects the robot is interacting with to be sensed as well as its compliance to applied force, and more complex sensors even allow the robot to sense when objects are slipping out from the fingers or to sense the temperature and texture of the object. Some sensors can even sense vibrations. Due to the size and form factor requirements they need to be thin and flexible (Hammock et al., 2013), and they also need to be soft to allow objects to be easily grasped (Roberts et al., 2021).

Types of sensors that have been used in or developed for robotic gripper applications include:

* Resistive
* Capacitive
* Magnetic
* Barometric
* Optical
* Accelerometers (used for slip detection)
* Distance sensors
* Temperature sensors

#### 4.2.1 Capacitive and Resistive Tactile Sensing

Of these capacitive touch and piezoelectric resistive type sensors are the most commonly used. (Stasse & Flayols, 2018) Capacitive sensors work by the compression of force applied to the finger bringing one electrode (usually flexible material in this case) of a capacitor closer to the other, thereby changing the capacitor’s capacitance (Roberts et al., 2021). They can also sense directional sliding forces (shear forces) due to the angle of the capacitive electrode changing (Hammock et al., 2013). Such sensors are used in the I-Cub hand (Schmitz et al., 2010) and are implemented using 108 pads on flexible PCBs and then covering these pads with first a silicone foam dielectric, then with a grounded conductive layer (silicone or lycra). In this case however the sensors were not designed to detect shear forces and their accuracy in detecting force was not quantified, only their ability to detect contact. Sensor communication was done over I2C with chips for each group of sensors to minimise wiring. One type of resistive sensor is the piezoelectric type consisting of piezoelectric materials with overlapping electrodes. This type has been demonstrated for haptic feedback robots (Roberts et al., 2021). Another type in the literature is the liquid conductor type that has a liquid metal or ionic conductor contained within channels in a polymer. As pressure is applied to the polymer the channels’ cross sectional area decreases, which increases the resistance through the liquid conductor (Roberts et al., 2021).

#### 4.2.2 Accelerometer, Temperature Sensor, Magnetic and Barometric Tactile Sensing

Magnetic sensors have either an elastomer embedded with magnetic particles on top of a hall effect sensor (Roberts et al., 2021) or a magnet on top of a flexible material on top of a hall effect sensor (Weiner et al., 2019). In the former case pressure is interpreted using machine learning algorithms while in the later case the interpretation is more straightforward but the sensor is presumably thicker. The former case also allows for the measurement of force and location of that force, down to sub millimetre accuracy (Hellebrekers et al., 2019). The latter can be used to measure both normal and shear forces through measuring field orientation as well as stress (Weiner et al., 2019). Another sensor type that is popular is a barometric type that measures fluid pressure from a fluid sealed in an elastomer (Roberts et al., 2021). This type is used in some commercial sensors such as the RightHand Robotics Inc’s TakkTile sensor and the BioTac sensor from SynTouch (Roberts et al., 2021), and the NPA201 from Amphenol (Weiner et al., 2019). Both magnetic and barometric sensors were used on the ARMAR-6 robot by embedding 2 into the distal phalanges and 1 into the middle phalanges of their robot and covering them with moulded silicone (Weiner et al., 2019). They partly used so many to test the effectiveness of the sensors but also because multimethod sensing showed advantages. On the same fingers they also implemented an accelerometer, polled at a high frequency of 1.6KHz, and using the short bursts of detected acceleration they were able to successfully detect when an object was slipping from the robot’s grasp. They also implemented time of flight distance sensors on the fingers which allowed them to conform the grip to the object better by sensing the distance from the base of the middle phalanx to the object. Capacitive sensors can also be used in this way to detect distances up to 60mm if designed well (Gao et al., 2020). Finally, in the ARMAR-6 each of the sensors had an additional temperature sensor to allow for calibration. Obviously all these sensors would have come at significant cost and complexity so there are many projects that would not justify such an array.

#### 4.2.3 Optical Tactile Sensing

The last of these sensors are the optical sensors. These work by having a constant light shine against a deformable reflective surface and the light bouncing off that surface being measured. When force is applied the surface deforms and the light coming into the light sensors changes, a machine learning algorithm can then be used to detect deformation direction and distance (corresponding to force). They are still very much in the research stage (Cirillo et al., 2021) but some robots like the full humanoid NASA Robonaut 2 use them in combination with other sensors (Roberts et al., 2021).

In general not many of the humanoid arms in the literature integrate fingertip sensors, this is likely due to the focus of the research being on one aspect or the other. Nevertheless, this does highlight a gap in terms of the testing of the sensors.

## 5. Humanoid Arm Control Systems

Now that the arm’s functional hardware has been considered the control systems and their associated hardware will be detailed.

### 5.1 Control Hardware

Control of a robotic arm is often split into 2-3 controllers. A faster, higher level controller like a miniature PC deciding on the joint movements, and running more complex tasks like computer vision and machine learning algorithms. This high level controller then sends signals to a lower level controller that controls torque and turns the joints to the right angles. From here either the low level controller signals in turn go to servo motor controllers, or go to motor controllers and position control is done by the low level controller using sensor feedback. There is of course great variance in the options for this setup, and many robots leave out one of these controllers.

#### 5.1.1 Low Level Controllers

##### 5.1.1.1 Use of Arduinos as a Low Level Controller

For low level controllers one common option is the humble Arduino. The use of this controller appears to be common in arms that use brushed DC servos as actuators such as Prabhu et al. (2021), Sakhineti & Jayabalan (2020) and Yang et al. (2019). The vast majority of hobbyist servos are designed to work well with an Arduino so this is a sensible decision for this case. It is very cheap, and the *mega* version has a lot of IO pins, however it is severely limited in terms of its 48MHz clock speed and <32kBytes of SRAM (for the fastest version to date, the R4 (Arduino, 2024)). Provided that the Arduino is only undertaking simple tasks and is backed up by a higher level processor as in Sakhineti & Jayabalan (2020) and Yang et al. (2019) this is of little concern. Though one would need to be confident the Arduino’s limitations would not be a hindrance later in a robotic arm project before committing to using it. Prabhu et al. (2021) used an Arduino as the only controller for the entire arm control system, which was controlled via flex sensors in a control glove that fed into the Arduino’s IO pins. This setup was sufficient in this case as the algorithms implemented for control were basic, however it means that more complex controls would be difficult to implement without rewiring in a new controller or rewriting the code to accommodate an additional high level controller. The Arduino also has sufficient processing to implement basic sensor-actuator control for non-servo motors or to control more complex BLDC servo motors. An example of this is the teleoperation robot from Procter and Secco (2021) where a single Arduino is used in combination with an ODrive BLDC controller for every pair of motors on the arm. The Arduino’s 5V output signals also provide more flexibility than a 3.3V system such as the Raspberry Pi.

##### 5.1.1.2 Use of Raspberry Pis as a Low Level Controller

Another sound option for a low level controller is a Raspberry Pi. This operates at a 3.3V standard and all of its IO pins are GPIO pins with PWM output and analog input, as opposed to the Arduino that only has limited PWM pins and analog input pins. The latest version (Pi 5) also operates at 2.4GHz, around 150 times faster than an Arduino, and with 500,000 times the memory at 8GB. For these reasons the Raspberry Pi is a popular choice and it has enough processing power that it can be used as both a high level and low level controller on one board. The exceedingly capable TOCABI full humanoid robot uses the Raspberry Pi as a low level controller (Schwartz et al., 2022). It is unclear from the paper whether only one is used or multiple are used to control different subsystems in this case, but the higher level control is then done from a embedded PC with an i7-9700 intel CPU. This makes sense in this case as the code for a complete humanoid robot is exceedingly computationally expensive (and high computational power creates opportunities for better robotic capabilities) so a high quality (for the time) PC would be desired. On the other end of the spectrum is a humanoid torso (including arms) designed to be low cost (in contrast to the $100,000+ NZD in parts for TOCABI (Schwartz et al., 2022)). This torso by Hayosh et al. (2020) is designed as a “social interaction robot” whose primary purpose is to provide emotional support and social interaction to users. This robot uses a Raspberry pi as the only controller, which interfaces with Dynamixel BLDC servos. Not using an embedded PC in this case greatly reduces costs and simplifies the programming, while the Raspberry Pi still allows for considerable computing power which is enough for most tasks. There is perhaps an argument that an embedded PC would also be beneficial in this case as the social interactions are likely to be autonomous and driven by machine learning (Sakhineti & Jayabalan, 2020), which is computationally heavy, but an embedded PC obviously comes at a significantly increased cost.

##### 5.1.1.3 Other Low Level Control Hardware

Generally low level controllers are either Raspberry Pis or Arduinos, but other microcontrollers are also used such as the 32 bit Atmel XMega64 microprocessor, 2 of which were used in the sign language robot from Meghdari et al. (2018). Or the custom motor controller board used for the dual arm robot from Shut & Hollis (2019). There are also custom motor-driver-sensor modules such as those used by the 2 armed robot from Asfour et al. (2019) and humanoid robot from Kaneko et al. (2019). In these cases some sort of microprocessor will have been used though exactly what is not specified. The advantages of using these microcontrollers would be additional capabilities they may have such as increased current output or true analog output, as well as their decreased cost and size. The disadvantage though is the increased development work required due to a lack of support and libraries, and as the development work is generally the largest cost of a robot (Schwartz et al., 2022), there is a strong argument to avoid these processors in a robotic arm, with the exception being for mass production arms where development costs are less important than unit costs.

#### 5.1.2 High Level Controllers

In robots that have a higher level controller on top of some amount of lower level controllers, this higher level controller is invariably a miniature PC. This is because if the higher processing power of a PC was not required then the control would simply be handled by a single lower level controller such as in Prabhu et al. (2021). One common setup for a miniature PC is to have one or multiple high level controllers (PCs) integrated with many very low level controllers, one for each actuator, that control torque and position for that actuator. The system has the advantage of having very fast processing of data, but it does increase complexity as it means there is a low level controller for every actuator as opposed to one low level controller controlling many actuators. The use of PCs and their increased processing power allows for greater abstraction and autonomy and one robot that embodied this concept is the wheeled, 2 armed robot from Asfour et al. (2019). This robot uses four miniature PCs and an additional graphics module for controlling the robot at the 3 levels of abstraction: perception and reasoning, movement/pose execution, and actuator control. This higher processing power allows for the use of a real time OS, for machine vision processing and for complex task planning, however this obviously comes at a significant financial investment. These PCs integrate with the sensors and actuators through EtherCAT (Ethernet for Control Automation Technology) protocol ethernet cabling. To allow the PCs to directly interface with the hardware the motors and sensors have some sort of controller directly attached to them that allows them to use the EtherCat protocol (it is not made clear whether this is custom or off the shelf). This adds complexity to the sensors and actuators, likely increasing costs. Another robot that also uses a similar control system is the humanoid robot from Kaneko et al. (2019). In this design two miniature PCs i7 intel processors (one 5th generation and one 7th generation) communicate via EtherCat to motor drivers, and the motor drivers integrate torque and position feedback from joint sensors. Communication to the cameras, inertial management unit and other sensors is then achieved via USB. In both systems the motors drivers control BLDC motors (presumed on Kaneko et al. (2019)) as opposed to servos. The full humanoid robot from Tatsch et al. (2017) and the testbed robotic arm from Mick et al. (2019) also use a similar system, except that the actuators used are off the shelf Dynamixel BLDC servos that be controlled via a PC over USB (with a protocol converter intermediary).

A common alternative to having a controller for each actuator (whether custom made or embedded into a servo motor) is to have one (or a few) lower level controllers that control multiple motors connected to a higher level PC controller. This is the case in the full humanoid robot from Schwartz et al. (2022) where a miniature PC with an i7 9th generation processor controls 3 lower level controllers: 2 Raspberry Pis and 1 secondary PC. Similarly, the robotic arm from Yang et al. (2019) and the dual arm social interaction robot from Sakhineti & Jayabalan (2020) use a PC as a higher level controller and an Arduino low level controller. Other off the shelf microprocessors can also be used as a lower level controller such as in the case of the dual arm robot from Shut & Hollis (2019) which uses one large control board coupled with a Intel Core2 Duo @ 2.4GHz and in the sign language robot from Meghdari et al. (2018) which uses multiple Atmel XMega64 microprocessors. All of these setups offer lower costs than having controllers for every motor if a higher level controller is required for performance reasons, but adds complexity and cost over a single controller such as a PC controlling BLDC servos or standalone raspberry pi.

In most cases the PC is selected to be an embedded part of the robot so a miniature PC is used because of space constraints, as well as because it is an all in one solution. However non-mobile arms that are in the prototyping stage may just use a standard PC or workstation to save on costs, such as may be the case in the arms from Yang et al. (2019) and Mick et al. (2019) (however neither specified what model PC they used).

### 5.2 Software Environment

As well as considering the software to run on the robot the software environment must be considered. The control software environment used on robotic arm depends on a few factors, namely:

* The limitations of the controller running it. E.g. an Arduino can only run compiled C++ programs and can’t run an OS.
* The complexity of the tasks to be undertaken.
* The speed at which tasks must be undertaken.
* Software cost.
* How easily the software integrates with the actuators and sensors.
* The desire for standardisation and interoperability.

For the robots that use an Arduino or other microcontroller there will generally be only a single operating environment: some version of embedded C or Object-C/C++, however a few microcontrollers do allow for microPython wrappers or other languages such as the ESP32, Pyboard and Raspberry Pi Pico (Raspberry Pi, n.d.). The many of these microcontrollers, the Arduino in particular, also have an extensive range of libraries that could be considered as a sort of middleware.

For embedded PCs like the Raspberry Pi (which is not a microcontroller like the Raspberry Pi Pico) and miniature PC’s there is a much wider range of options for software environments. Firstly, there is the actual operating system and then there is the middleware.

#### 5.2.1 Operating System (OS)

The operating system used is generally some version of Linux as this allows for stable systems and overarching control by the programmer. For the robotic systems running on a Raspberry Pi these will commonly use the default Raspbian OS, but if running Robotic Operating System middleware (which is not in itself an OS) it is more common to run a version of Ubuntu on the Pi as it has better integration (The Robotics Back-End, 2020). Ubuntu appears to be the most commonly used OS on miniature PCs in robotics. Ubuntu 14.04 Linux is used on the dual arm assistance robot from Shut & Hollis (2019), and Ubuntu 18.04 is used on the full humanoid robot from (Schwartz et al., 2022). However most papers do not specify what operating system is used, as in many cases they are interchangeable and only minor adjustments need to be made to run the application in a different OS. It is possible that some prototype robotic arms such as the one from Sakhineti & Jayabalan (2020), where the computer only used to communicate small amounts of data to the lower level control board, could run Windows. In this case communication is done via a com port and the computer is simply sending the motions from a digital pre-recording to the Arduino so nothing complex needs to be done on the operating system. It is not specified what OS they use however.

#### 5.2.2 Middleware

There are extensive options for middleware specifically designed for robotics. Middleware is an extension of the operating system’s functionality. In robotics this extension commonly includes such features as support for real-time systems, physics simulation libraries, communication protocol support and integration with machine learning libraries. Generally most complex robots will use some version of middleware

##### 5.2.2.1 Robotic Operating System (ROS)

The most common of these middlewares is ROS (Robotic Operating System), despite the name it is not a standalone operating system but rather a middleware primarily used on Ubuntu, although according to the distributors of the software (Open Robotics, 2020) support for other linux systems and Mac OS X also exist. Both ROS 1 and ROS 2 exist and they are open source projects supported by the commercial company Open Robotics (Open Robotics, 2020). ROS 1 has been criticised as lacking support for real time operations (Stasse & Flayols, 2018) but is commonly used across the board in robotics applications (Macenski et al., 2022). ROS 1 is also lacking in terms of security and reliability (Macenski et al., 2022). ROS 2 is an alternative that was first developed in 2017 to combat these challenges. ROS 2 is fully supported by Ubuntu, Mac OS and Windows 10, and has improved security using the Data Distribution Service (DDS) standard, which is used in such high stakes applications as spacecraft. Importantly it also provides real-time support, which is critical in fast moving, accurate robots. There are few examples in the literature of ROS 2 being used but this is likely as it is such a new software. In contrast, examples of ROS 1 being used include in the medical teleoperation arm from Procter and Secco (2021), and the full humanoid robot from Pal Robotics (Aller et al., 2022). There is a strong argument for the use of ROS in robotics as it allows for programs to be easily built on previous work due to it being open source and the most popular middleware (Saeedvand et al., 2019).

##### 5.2.2.2 Open source Robotic Technology (OpenRTM)

An alternative to ROS is OpenRTM (Open source Robotic Technology), which is heavily oriented towards real time operations (Stasse & Flayols, 2018). This is commonly used in Japan, and is used in the Humanoid Robotics Project competition there (Stasse & Flayols, 2018). It comes with realistic simulators and good support for balancing and walking.

##### 5.2.2.3 Yet Another Robotics Platform (YARP)

YARP (Yet Another Robotics Platform) is, yet another, middleware that can be used for robotics (Stasse & Flayols, 2018). The main use of YARP is to allow for straightforward communication across multiple pieces of hardware and operating systems (Stasse & Flayols, 2018) and for this reason it allows for the use of many different communication protocols (Macenski et al., 2022) and runs on Linux, Mac OS and Windows (Stasse & Flayols, 2018). It is most commonly used on legged and humanoid robots (Macenski et al., 2022).

##### 5.2.2.4 Other Robotic Middleware

These are of course not the only options for robotic middleware, there are also frameworks in Matlab/Simulink and an MCA (Modular Controller Architecture) middleware that are useful for the control aspects, and middlewares such as Orocos and which are similar to YARP and ROS in that the provide many low level communication features (Vahrenkamp et al., 2015). There are also middlewares designed for specific actuator-sensor systems such as Pypot, a python library for controlling Dynamixel motors and sensors. Such a system would normally have little relevance except for the fact that Dynamixel motors are used in so many robots. Not all of these robots use Pypot but some do such as the testbed robotic arm from Mick et al. (2019). The use of python here on a test arm is not an issue, but it should be noted that running a production level or complex robot primarily using the python language and a library such as Pypot would be inadvisable due to python's very slow execution speed. Another option is to develop a custom middleware such as the ArmarX middleware developed and used for the ARMAR series of full humanoid robots (Asfour et al., 2019). However this option comes at considerable expense/resource investment, and comes with the downside of limiting software interoperability (though this could be an advantage for commercial robots). Other robots may use a real-time system (but not a robot specific one) such as Xenomai-3.0.10 which is run on Ubuntu 18.04 in the full humanoid robot TOCABI (Schwartz et al., 2022). Finally, the other option is to use no specific middleware, which may be the simplest option when the robot is not complex and there are a limited number of sensors and controllers. This is likely the case in the social interaction torso robot from Sakhineti and Jayabalan (2020) which only uses an Arduino that is communicated with via a com port, and is of course the case when the robot is controlled entirely by microcontrollers such as in the control glove controlled arm from Prabhu et al. (2021).

### 5.3 Humanoid Arm Control Methods and Algorithms

#### 5.3.1 Control Input

After considering the environment in which to run the software the actault softawrae and algorithsm will now be considered. The software for controlling a robotic arm can be broken down into hierarchical components, depending on what the control input is to the arm. There are many input methods that are minimal and the arm has a high degree of autonomy. For example where the robot is making decisions about how to best pick up an object, or how to climb a ladder etc. This is an important area of research but the focus of this project is on the hardware of the arm and the development of a flexible software/firmware base, so that further work can develop these autonomous methods. Other input methods do not need the robot to make many decisions about joint angles as the input controls have an almost 1 to 1 correspondence with the robot DOF. Examples of these methods include control gloves (for example in the robotic arm from (Prabhu et al., 2021)) and machine vision systems for teleoperation. If well designed for interoperability then all of these input methods can be used to control a general arm model that can change based on the hardware, and that is the aim of this project: to develop a robotic arm and the base software for controlling said arm such that it can be easily implemented with a variety of control methods at a later stage.

Doing this involves implementing a model of the arm in an easy to use software that can control the real world arm. An example of this is the dual arm robot from Sakhineti & Jayabalan, (2020) that is controlled by a Unity 3D GUI with a joystick controller. This robot allows for play and record functionality, but could easily be extended to do more complex functions. A similar digital model was also used for the arm from Yang et al. (2019), where a Xbox Kinect skeleton framework coupled with the Open Dynamics Engine physics simulator are used as a digital twin of the arm.

#### 5.3.2 Low Level Control Algorithms

Any robot needs some level of low level control over the joints in order to get them to the desired position at the desired speed, without exceeding safe torques and potentially while interacting with objects. Arguably the simplest and most well established method for low level optimization control are PID controllers. Within this category there are also many variations such as PD controllers, PI controllers and fractional order PID controllers (Muñoz et al., 2019). If the robot is using servo motors then position control is already handled by feedback controllers in the servos and more expensive BLDC servos may also have speed or torque control. Otherwise, a feedback loop is implemented between the angle sensors installed, any torque sensors and the inputs to the motor. Motor inputs could include PWM duty cycle, current, or operating frequency depending on the type of motor and driver. Another similar method is a fuzzy-tuned PI control method which was found by Yin et al. (2022) to considerably decrease the speed fluctuations and increase the control performance vs a PI controller on a humanoid driven by tendons. In addition to the array of different methods mentioned there are also different methods for tuning the parameters of the controller. For a simple PID controller there are well established methods such as Trial and Error, Zieglar-Nichols Step response, Relay Tuning, Cohen-Coon (Borase et al., 2020). There are also less common methods such as neural network tuned PID controllers which were shown by Ali Talib Jawad et al. (2021) to improve control. Some common approaches to tuning fractional order PID controllers are Monge's Method, the Artificial Bee Colony algorithm or the Counter-slope method (Muñoz et al., 2019), though a novel graphical method that shows robust (and comparable) performance for a robotic elbow joint is also presented by Muñoz et al. (2019). Other low level control methods also exist for robotic arms but there appears to be a gap in the literature on the use of low level control in humanoid arms to base this on.

#### 5.3.3 Higher Level Control Algorithms and Inverse Kinematics

A major aspect of controlling a robotic arm when the control input is not exact in its specifications for the position of every joint (e.g for teleoperations control) is in determining how best to angle the joints of the arm to place the end effector in the desired position (Mick et al., 2019). An example of this is when a robot is programmed to pick up an object and must place the end effector around the object. This is commonly referred to as the inverse kinematics problem (Mick et al., 2019). Typically there are libraries that come with robot middleware for solving these inverse kinematics problems both numerically and analytically (Mick et al., 2019) however robots with 7 degrees of freedom or higher will typically need their inverse kinematics to be solved numerically due to the complexity of the equations and processing involved (Kamil Khusnutdinov et al., 2019). One common such numerical method is local optimization (Mick et al., 2019) and varies libraries exist for this such as the IKPy (Python) library used for controlling the testbed robotic arm from Mick et al. (2019). Neural networks are another approach to the inverse kinematics problem and were to be slightly superior over local optimization when tested on the same robotic arm (Mick et al., 2019). Neural networks also offer the opportunity to add in bais to the training data to make the robot select kinematic solutions that are more useful in the real world (Mick et al., 2019), for example having the robot face the palm down when manipulating objects up high.

## 6. Summary

This review began by considering the applications of the humanoid robotic arms and found that there is a broad range of applications that humanoid arms have been developed and considered for. These include teleoperation for dangerous environments or skilled personal, companionship robots, sign language interpretation, and automation. An analysis of the state of the art then found that there was a lack of research on humanoid arm accuracy, and that the best humanoid arms could replicate almost all the actuated DOF of the human arm but could only handle loads of up to 10kg. It was also found that there was a gap in the literature for arms that were low cost with moderate to high DOF and moderate strength. Actuators for these arms were then considered and this found that the majority of humanoid arms use BLDC servo motors as actuators which provide positional control and torque sensing. There were also a few humanoid robots that used standalone BLDC or brushed DC motors coupled with positional feedback, some cheaper robots that used brushed servo motors, and a few that used hydraulic actuators which provided the best torque to weight ratio. Very few humanoid arms have used pneumatic actuators, AC motors, solenoids or stepper motors.

Joint positional sensors were then considered which were only used on arms that didn’t use servo motors. These sensors tend to be of the encoder (hall effect type) as opposed to the potentiometer type and a few robots have used “custom” sensors based on hall effect sensors and neodymium magnets to create smaller sensors that can fit on the phalangeal joints. Joint torque sensors we also briefly considered at it was found that they are used on a few robots but are not on all, though many of the commonly used BLDC servos come with torque sensing in-built. Tactile sensing was found to be most useful on the fingertips and top of the palm and can be achieved in wide array of methods under active research but the most common are capacitive and piezoelectric resistive type, few of the other types are commonly available.

Arm control systems were also considered, starting with the hardware. Generally this consisted of an embedded PC that would control either actuators with individual controllers (custom or off the shelf) or a lower level controller that controlled multiple actuators. In a few cases however a low level controller would be the sole controller. Low level controllers were most commonly Arduinos or Raspberry Pis. Where the controller allowed for an operating system this system was Linux based in the literature that was found, and this operating system would commonly have a middleware installed to extend its functionality towards robotic applications. Common middleware was ROS, Open RTM and YARP though many other system exist. ROS (v1) is a common standard across the board. Methods for controlling a humanoid arm were summarized such as autonomy, preprograming, control gloves and machine vision (in teleoperation). Low level control methods for robotic arms in the literature include PID control and its variations, fuzzy tuned PID and fractional order PID controllers, but there was a gap in terms of the literature that have used different low level control methods on real world humanoid arms. Higher level control algorithms for solving inverse kinematic problems were found to be commonly numerically solved using local optimization but could also be solved analytically or using machine learning. Overall, it was found there was a gap surrounding a broad review of common joint level control algorithms in humanoid arms.

As a result of this review a broad understanding of the types of actuators, sensors, controllers and algorithms used in humanoid arms has been reached, thereby fulfilling the aim of the review. This allows the project to move forward with the selection of these components based on what has worked previously in the literature as well as based on the characteristics of these components and robots using these components.

## References

Adafruit. (n.d.). *Adafruit Motor Selection Guide*. Adafruit Learning System. <https://learn.adafruit.com/adafruit-motor-selection-guide/rc-servos>

Alfayad, S., Ouezdou, F. B., Namoun, F., & Gheng, G. (2011). High performance integrated electro-hydraulic actuator for robotics – Part I: Principle, prototype design and first experiments. *Sensors and Actuators A: Physical*, *169*(1), 115–123. <https://doi.org/10.1016/j.sna.2010.10.026>

Ali Ahmad Malik, Masood, T., & Brem, A. (2024, March 11). Intelligent Humanoid Robots in Manufacturing. *HRI ’24: Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. 2024 ACM/IEEE International Conference on Human-Robot Interaction. <https://doi.org/10.1145/3610978.3640765>

Ali Talib Jawad, Ali, N. S., Ali Najim Abdullah, & Nahedh Hasoon Alwash. (2021, July 25). *Design of Adaptive Controller for Robot Arm Manipulator Based on ANN with Optimized PID by IWO Algorithm*. 2021 International Conference on Advanced Computer Applications (ACA). <https://doi.org/10.1109/aca52198.2021.9626781>

Aljalal, M., Ibrahim, S., Djemal, R., & Ko, W. (2020). Comprehensive review on brain-controlled mobile robots and robotic arms based on electroencephalography signals. *Intelligent Service Robotics*, *13*(4), 539–563. <https://doi.org/10.1007/s11370-020-00328-5>

Aller, F., Harant, M., & Mombaur, K. (2022). Optimization of Dynamic Sit-to-Stand Trajectories to Assess Whole-Body Motion Performance of the Humanoid Robot REEM-C. *Frontiers in Robotics and AI*, *9*. <https://doi.org/10.3389/frobt.2022.898696>

Arduino. (2024). *UNO R4 Minima*. Arduino.cc. <https://docs.arduino.cc/hardware/uno-r4-minima/#tech-specs>

Asfour, T., Paus, F., Waechter, M., Kaul, L., Rader, S., Weiner, P., Ottenhaus, S., Grimm, R., Zhou, Y., & Grotz, M. (2019). ARMAR-6: A High-Performance Humanoid for Human-Robot Collaboration in Real-World Scenarios. *IEEE Robotics & Automation Magazine*, *26*(4), 108–121. <https://doi.org/10.1109/mra.2019.2941246>

Borase, R. P., Maghade, D. K., Sondkar, S. Y., & Pawar, S. N. (2020). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*, *9*(2), 818–827. <https://doi.org/10.1007/s40435-020-00665-4>

Breen, D., Lin, M., Elkoura, G., & Singh, K. (2003). *Handrix: Animating the Human Hand*. <https://www.dgp.toronto.edu/~gelkoura/noback/scapaper03.pdf>

Cirillo, A., Costanzo, M., Laudante, G., & Pirozzi, S. (2021). Tactile Sensors for Parallel Grippers: Design and Characterization. *Sensors*, *21*(5), 1915. <https://doi.org/10.3390/s21051915>

Davis, S. (2018, March 10). *The New WALK-MAN: A Look at IIT’s Multi-Faceted Robotic Endeavor*. Electronic Design. <https://www.electronicdesign.com/markets/robotics/article/21199514/the-new-walk-man-a-look-at-iits-multi-faceted-robotic-endeavor>

Dynamixel. (n.d.). *DYNAMIXEL MX-28AR*. ROBOTIS. Retrieved May 5, 2024, from <https://www.robotis.us/dynamixel-mx-28ar/>

Gao, D., Wang, J., Ai, K., Xiong, J., Li, S., & Pooi See Lee. (2020). Inkjet‐Printed Iontronics for Transparent, Elastic, and Strain‐Insensitive Touch Sensing Matrix. *Advanced Intelligent Systems*, *2*(7). <https://doi.org/10.1002/aisy.202000088>

Hammock, M. L., Chortos, A., Tee, B. C.-K. ., Tok, J. B.-H. ., & Bao, Z. (2013). 25th Anniversary Article: The Evolution of Electronic Skin (E-Skin): A Brief History, Design Considerations, and Recent Progress. *Advanced Materials*, *25*(42), 5997–6038. <https://doi.org/10.1002/adma.201302240>

Hashimoto, K. (2020). Mechanics of humanoid robot. *Advanced Robotics*, *34*(21-22), 1390–1397. <https://doi.org/10.1080/01691864.2020.1813624>

Hayosh, D., Liu, X., & Lee, K. (2020, June 1). *Woody: Low-Cost, Open-Source Humanoid Torso Robot*. 2020 17th International Conference on Ubiquitous Robots (UR). <https://doi.org/10.1109/ur49135.2020.9144924>

Hellebrekers, T., Kroemer, O., & Majidi, C. (2019). Soft Magnetic Skin for Continuous Deformation Sensing. *Advanced Intelligent Systems*, *1*(4), 1900025. <https://doi.org/10.1002/aisy.201900025>

Kamil Khusnutdinov, Artur Sagitov, Ayrat Yakupov, Lavrenov, R., Edgar Alonso Martínez-García, Hsia, K.-H., & Magid, E. (2019). Household Objects Pick and Place Task for AR-601M Humanoid Robot. *Lecture Notes in Computer Science*, 139–149. <https://doi.org/10.1007/978-3-030-26118-4_14>

Kaneko, K., Kaminaga, H., Sakaguchi, T., Kajita, S., Morisawa, M., Kumagai, I., & Kanehiro, F. (2019). Humanoid Robot HRP-5P: An Electrically Actuated Humanoid Robot With High-Power and Wide-Range Joints. *IEEE Robotics and Automation Letters*, *4*(2), 1431–1438. <https://doi.org/10.1109/lra.2019.2896465>

Kumar, A. S. A., George, B., & Mukhopadhyay, S. C. (2021). Technologies and Applications of Angle Sensors: A Review. *IEEE Sensors Journal*, *21*(6), 7195–7206. <https://doi.org/10.1109/jsen.2020.3045461>

Lattanzi, L., Cristalli, C., Massa, D., Boria, S., Lépine, P., & Pellicciari, M. (2020). Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task. *The International Journal of Advanced Manufacturing Technology*, *111*(7-8), 1813–1829. <https://doi.org/10.1007/s00170-020-06179-9>

Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics*, *7*(66). <https://doi.org/10.1126/scirobotics.abm6074>

Mattila, J., Koivumaki, J., Caldwell, D. G., & Semini, C. (2017). A Survey on Control of Hydraulic Robotic Manipulators With Projection to Future Trends. *IEEE/ASME Transactions on Mechatronics*, *22*(2), 669–680. <https://doi.org/10.1109/tmech.2017.2668604>

Meghdari, A., Alemi, M., Zakipour, M., & Kashanian, S. A. (2018). Design and Realization of a Sign Language Educational Humanoid Robot. *Journal of Intelligent & Robotic Systems*, *95*(1), 3–17. <https://doi.org/10.1007/s10846-018-0860-2>

Mick, S., Lapeyre, M., Rouanet, P., Halgand, C., Benois-Pineau, J., Paclet, F., Cattaert, D., Oudeyer, P.-Y., & de Rugy, A. (2019). Reachy, a 3D-Printed Human-Like Robotic Arm as a Testbed for Human-Robot Control Strategies. *Frontiers in Neurorobotics*, *13*. <https://doi.org/10.3389/fnbot.2019.00065>

Mori, S., Tanaka, K., Nishikawa, S., Niiyama, R., & Kuniyoshi, Y. (2018). High-Speed and Lightweight Humanoid Robot Arm for a Skillful Badminton Robot. *IEEE Robotics and Automation Letters*, *3*(3), 1727–1734. <https://doi.org/10.1109/LRA.2018.2803207>

Muñoz, J., Monje, C. A., Martínez, S., & Balaguer, C. (2019). Joint Position Control Based on Fractional-Order PD and PI Controllers for the Arm of the Humanoid Robot TEO. *International Journal of Humanoid Robotics*, *16*(06), 1950042–1950042. <https://doi.org/10.1142/s0219843619500427>

Naoki Fukaya, Asfour, T., Dillmann, R., & Toyama, S. (2013). *Development of a five-finger dexterous hand without feedback control: The TUAT/Karlsruhe humanoid hand*. pp. 4533-4540. <https://doi.org/10.1109/iros.2013.6697008>

Open Robotics. (2020). *ROS.org | Powering the world’s robots*. Ros.org. <https://www.ros.org/>

Pagoli, A., Chapelle, F., Corrales-Ramon, J.-A., Mezouar, Y., & Lapusta, Y. (2022). Large-Area and Low-Cost Force/Tactile Capacitive Sensor for Soft Robotic Applications. *Sensors*, *22*(11), 4083. <https://doi.org/10.3390/s22114083>

Paul. (2024, March 5). *Use of drone motors for robotic arm* [Email to Kieran Smith]. Combat Robotics New Zealand Representative.

Prabhu, M. K., Sivaraman, P., Vishnukaarthi, M., Shankara Narayanan, V., Raveen, S., & Keerthana, G. (2021). Humanoid gesture control arm with manifold actuation using additive manufacturing. *Materials Today: Proceedings*, *37*, 717–722. <https://doi.org/10.1016/j.matpr.2020.05.724>

Procter, S., & Secco, E. L. (2021). Design of a Biomimetic BLDC Driven Robotic Arm for Teleoperation & Biomedical Applications. *Journal of Human, Earth, and Future*, *2*(4), 345–354. <https://doi.org/10.28991/hef-2021-02-04-03>

Raspberry Pi. (n.d.). *Raspberry Pi Documentation - Raspberry Pi Pico*. Www.raspberrypi.com. Retrieved May 6, 2024, from <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>

Roberts, P., Zadan, M., & Majidi, C. (2021). Soft Tactile Sensing Skins for Robotics. *Current Robotics Reports*, *2*(3), 343–354. <https://doi.org/10.1007/s43154-021-00065-2>

Saeedvand, S., Jafari, M., Aghdasi, H. S., & Baltes, J. (2019). A comprehensive survey on humanoid robot development. *The Knowledge Engineering Review*, *34*. <https://doi.org/10.1017/s0269888919000158>

Sakhineti, M., & Jayabalan, S. (2020). Design and Fabrication of SHRALA: Social Humanoid Robot Based on Autonomous Learning Algorithm. *Procedia Computer Science*, *171*, 2050–2056. <https://doi.org/10.1016/j.procs.2020.04.220>

Schmitz, A., Ugo Pattacini, Nori, F., Natale, L., Metta, G., & Giulio Sandini. (2010, December 1). *Design, realization and sensorization of the dexterous iCub hand*. Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on. <https://doi.org/10.1109/ichr.2010.5686825>

Schwartz, M., Sim, J., Ahn, J., Hwang, S., Lee, Y., & Park, J. (2022, November 28). Design of the Humanoid Robot TOCABI. *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. 2022 IEEE-RAS. <https://doi.org/10.1109/humanoids53995.2022.10000102>

Shadowrobot. (n.d.). *Dexterous Hand Series – Shadow Robot Company*. Shadowrobot. Retrieved May 6, 2024, from <https://www.shadowrobot.com/dexterous-hand-series/>

Shut, R., & Hollis, R. (2019, October 1). *Development of a Humanoid Dual Arm System for a Single Spherical Wheeled Balancing Mobile Robot*. 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids). <https://doi.org/10.1109/humanoids43949.2019.9034999>

Stasse, O., & Flayols, T. (2018). An Overview of Humanoid Robots Technologies. *Springer Tracts in Advanced Robotics*, 281–310. <https://doi.org/10.1007/978-3-319-93870-7_13>

Su, H., Hou, X., Zhang, X., Qi, W., Cai, S., Xiong, X., & Guo, J. (2022). Pneumatic Soft Robots: Challenges and Benefits. *Actuators*, *11*(3), 92. <https://doi.org/10.3390/act11030092>

Suzumori, K. (2020). New Robotics Pioneered by Fluid Power. *Journal of Robotics and Mechatronics*, *32*(5), 854–862. <https://doi.org/10.20965/jrm.2020.p0854>

Tatsch, C., Ahmadi, A., Fabrício Bottega, Tani, J., & Rodrigo. (2017). Dimitri: an Open-Source Humanoid Robot with Compliant Joint. *Journal of Intelligent & Robotic Systems*, *91*(2), 291–300. <https://doi.org/10.1007/s10846-017-0727-y>

The Robotics Back-End. (2020, March 16). *Using ROS on Raspberry Pi: Best Practices*. The Robotics Back-End. <https://roboticsbackend.com/using-ros-on-raspberry-pi-best-practices/>

Tondu, B. (2012). Modelling of the McKibben artificial muscle: A review. *Journal of Intelligent Material Systems and Structures*, *23*(3), 225–253. <https://doi.org/10.1177/1045389x11435435>

Vahrenkamp, N., Wächter, M., Kröhnert, M., Welke, K., & Asfour, T. (2015). The robot software framework ArmarX. *It - Information Technology*, *57*(2). <https://doi.org/10.1515/itit-2014-1066>

Weiner, P., Neef, C., Shibata, Y., Nakamura, Y., & Asfour, T. (2019). An Embedded, Multi-Modal Sensor System for Scalable Robotic and Prosthetic Hand Fingers. *Sensors*, *20*(1), 101. <https://doi.org/10.3390/s20010101>

Yang, H., Yan, Y., Su, S., Dong, Z., & Ul Hassan, S. H. (2019, June 1). *LWH-Arm: A Prototype of 8-DoF Lightweight Humanoid Robot Arm*. IEEE Xplore. <https://doi.org/10.1109/ICRAS.2019.8809073>

Yin, M., Shang, D., Huang, B., Wu, X., & Feng, W. (2022). Modeling and control strategy of flexible joint servo system in humanoid manipulator driven by tendon-sheath. *Journal of Mechanical Science and Technology*, *36*(5), 2585–2595. <https://doi.org/10.1007/s12206-022-0439-5>

## Appendix – Project Propossal