Good morning everyone. My project is the control of a rolling-balancing mechanical system, in particular the disk-on-disk system. ||

I’m going to go through the motivation for this project and why we are researching this area. Then I will speak about the modelling and simulation process. Finally, I will go through the current control method, which will involve the computer vision process being used. ||

The disk on disk system is an example of an under-actuated system. The technical definition of an under-actuated system is one which cannot be commanded to follow an arbitrary trajectory in a configuration space. What this really means is one of two things: that the system has less actuators than degrees of freedom, or there is not an actuator associated with every degree of freedom. An inverted pendulum robot or Segway is a great example of an underactuated system. It can translate along the ground and rotate about the wheel axis, giving two degrees of freedom, but there is only an actuator for the translational degree of freedom.

Under actuation makes it difficult to control a system, and most real-world systems are underactuated. Cars, boats, aeroplanes and even animals are all underactuated systems. Studying these systems and developing a generalised understanding of them allows the potential of robotic systems to be realised. Constraining robots to tasks for which they are fully actuated severely limits their ability to achieve the performance they are capable of. Exploiting the system dynamics when controlling underactuated systems allows this performance to be realised.

The way humans naturally interact with a system has influenced the preferred method of robotic manipulation. If we need to move an object, we pick it up and place it down in the new position using our hands. Normally we grasp objects using our thumbs, fingers and palm to enclose it. This allows us to very accurately control the position and orientation of the object. This is called a prehensile manipulation or grasping manipulation. Prehensile manipulation is the most straightforward way we have to interact with our environment, which has led to the design of robotic systems that exclusively using grasping techniques.

Non-prehensile manipulation is any type of interaction that doesn’t involve grasping. A large portion of the things we do in a day are non-prehensile manipulation. Stirring food in a pan, pushing open a door, carrying a tray of wine glasses, even walking are all non-prehensile interactions. Using robotic manipulators to perform these actions is far more difficult than prehensile ones. The manipulation is not a closed kinematic chain and the changes of state are generally non-smooth. Additionally, as the object is free to move relative to the manipulator, the majority of these systems become underactuated.

To extend the capabilities of robotic systems beyond direct manipulation using grasping actions, we need to better understand these non-prehensile manipulations. A robotic manipulator capable of non-prehensile manipulation is useful in a wide range of applications. Most importantly, robotic walking is a non-prehensile action subject to a large amount of research at this time.

Understanding the interaction between the dynamics of the manipulator and object during a non-prehensile manipulation is critical to determine the appropriate control to be applied. As these interactions are usually complex for even a seemingly simple task, we tend to break the motion down into sub-tasks called primitives. Rolling, sliding, pushing and throwing are examples of primitives. Generally, each type of primitive has a defined control strategy. A complex task involving multiple primitive actions can be controlled using a combination of primitive control strategies. Control of non-prehensile primitives extends the kinematic workspace of a robotic manipulator beyond its physical region of influence and allows a far greater set of actions to be completed.

The disk on disk system isolates the rolling type primitive. In particular, the balancing action using only rolling primitives. By developing a model, simulation, and experiment apparatus to test control methods we can gain further insight into the general method for controlling rolling primitive movements.

Simulation and control of a mechanical system requires a mathematical model. There are several different approaches for developing these models, each with their own benefits and drawbacks. Newton’s equations of motion can be used to develop a first principles approach, which is usually more suitable for non-complex two-dimensional systems. Bond graphs allow a more visual approach to modelling, with the flow of power throughout the system modelled using junctions and signal interconnections. This approach is usually quite systematic, with a defined modelling procedure built up from assigning modelling elements to system elements. The main drawback of the bond graph approach is the requirement of strong foundational knowledge.

Energy based approaches, such as the Euler-Lagrange method, are highly suitable for the disk on disk system. The Euler-Lagrange method focuses on energy storying elements, such as inertias and capacitors, which describe how the energy of the system is stored and transferred. This approach is intuitive and elegantly captures the dynamics of rotational elements. The Euler-Lagrange method requires an intelligent choice of generalised coordinates for the system to generate a model where the number of degrees of freedom aligns with the number of coordinates.

Developing a model with the Euler-Lagrange method is pretty systematic and has a few desired outcomes. We are looking to determine the mass matrix, the damping matrix, the gradient of potential energy, and the centripetal-Coriolis matrix. These matrices can easily be rearranged into forms suitable for simulation and control.

The layout of the system used for modelling is shown on screen. The upper disk is called the hand, and the lower disk is called the object. Modelling occurs in two dimensions as the disks are not free to move in or out of the page. The generalised coordinates are the rotation of the object, theta, and the rotation of the hand about the object centre, phi.

To briefly step through the modelling process, first we identify the energy storing elements, which in this case is the two disks. This allows us to form the Lagrangian, which is the difference between the kinetic co-energy and the potential energy functions.

Next, we factor the Lagrangian into quadratic form and extract the mass matrix, which is the first of our four key components. The Rayleigh dissipation function captures forces like friction which remove energy from the system and is factored into quadratic form to extract the damping matrix. The gradient of potential energy is simply the derivative of the potential energy function, while the centripetal Coriolis matrix is a collection of the terms that are left over after this process.

For the purpose of modelling a few assumptions have been made, shown on screen. These assumptions simplify the problem by removing some dynamics are that very hard to model and control. The last assumption allows for a linearization to occur, which actually doesn’t introduce a significant error.

By rearranging to form an equation with the second derivative of position as the subject, we can use numerical integration tools to determine the state of the system for any time. Anyone here studying mechatronics is probably familiar with Simulink, which will perform this numerical integration when provided with a block diagram of the system.

Before the model can be used for control purposes, it needs to be verified for correctness. As this is a common research problem, there have been models developed using other methods, two of which agree with the result of the Euler-Lagrange method used in this project.

The model is also somewhat verified by conducting a free swing simulation test. The test is started with the system near the balancing position and no input is provided. Due to assumption three, we would expect the smaller disk to fall and swing about the larger disk. Assumption two sees some transfer of energy between the two disks, causing the larger one to rotate as well. In the absence of any dissipation forces, we would expect this movement to go on indefinitely, which can be seen in the plots on the screen.

Since the model is now verified to be accurate, the design of the controller can occur. The control method currently being used is the Linear-Quadratic Regulator, which is a type of state feedback controller. The LQR method works by minimising a specified cost function, with the requirement that the system is expressed by a set of linear differential equations. There are two tuning handles so to speak, which are the state penalty and the input penalty. These handles allow us to specify how the controller determines the control law, and whether it should place emphasis on the state deviation from the desired position, or the amount of input effort being used. LQR also allows for a setpoint regulation via a reference feedforward structure, shown on screen.

This controller, whilst not particularly sophisticated, is suitable to determine the input force required for balancing.

LQR has shown promising results in simulation, with balancing achieved and good response time to set point regulation. Here we can hopefully see a 15 second simulation looping. The initial position of the top disk is offset from the balancing position. The controller then attempts to regulate the velocity of the bottom disk to 1 revolution per second in each direction whilst maintaining balancing.

State feedback control like LQR is not possible without an accurate estimation of the state. The disk on disk system is complicated in this area, as it is difficult to measure the position of the top disk. Typically an IMU or rotary encoder is used to measure angle offsets, but these require a physical connection between the sensor and the controller. This is okay for the bottom disk which is driven by a motor, but not suitable for the top disk.

Instead, state estimates for the position of the top disk can be made using a computer vision process, which removes the need for any wires. Using a low quality webcam, measurements of the disk position can be obtained from the video feed, with a bit of set up.

Cameras introduce distortion to images they capture through two methods. The first is the lens distorting the light as it passes through, and the second is the projection of a 3d scene onto a 2d plane. Fortunately in this experiment the position of the camera relative to the plane of the disks is fixed, so the corrections for these errors only need to be calculated once.

We can correct for these errors using a series of photos of a checkerboard pattern shown here. The pattern is photographed in several positions in the 3d workspace the camera will be used in with the camera fixed in place. This patten is used as it provides well-defined cross corners for use with corner detection algorithms. The size of the grid squares also gives a known length reference.

Using the detected corner points and grid square size, the camera parameters initially determined in closed form, which is then used as the starting point for a least-squares minimisation solution.

The images obtained can be undistorted using the camera parameters. Here we have the original image on the left, and the undistorted image using the estimated camera parameters. We can see the right image has the corners pulled in, which corrects the projection and lens errors. The projection error is similar to the one we see on world maps that causes some countries to be disproportionately stretched out. ||

We can also measure the magnitude of the reprojection errors in the original image, which in this case are quite small. A reprojection error below 1 pixel is typically considered acceptable, and we can see here the mean error is far below this threshold.

Now that the camera is calibrated we can do feature recognition and extract information about certain objects in the image. The disk on disk system is well suited to less sophisticated methods of feature recognition, unlike the impressive neural network-based methods of Tesla for their self-driving cars.

On the screen is the undistorted test image in its HSV version. We can see the blue circles at the top of the image are still well defined from their surroundings. By extracting the saturation channel, and thresholding the image based on the mean saturation value, we are left with a binary representation of the image, shown here ON THE RIGHT?.

This binary representation is suitable for use with a blob detection algorithm. By comparing the value of a pixel to its surroundings, regions of connected pixels can be determined, called blobs. As you can see, there is a significant amount of noise introduced by the checkerboard pattern. Fortunately, the area in which the disks can exist in the disk on disk system is known, as is the region the two circles inhabit in this image. By constraining the blob detection to this region and specifying minimum and maximum size constraints, the circle locations can be determined. The location of the circles in world coordinates is now known.

This process can be carried out far quicker than the camera can capture images in a video stream, making the extension to video streams rather simple.

These measurements are unfortunately subject to uncertainty. In order to account for the uncertainty, a Kalman filter has been used. The Kalman filter assumes the noise on the state transition model and measurements are normally distributed and reduces the impact of incorrect measurements on the state estimation. It also provides an estimate of the velocity of the two disks, which is required for the state feedback controller.

The construction of the experiment apparatus is currently underway and will allow the testing of more sophisticated control methods in real world scenarios, which will be covered in the part B conference.

. Normally on a Segway, we want the person to be standing upright with the wheels underneath them while they move around. Moving the Segway backwards and forwards is simple as the actuator, the motor, acts in this degree of freedom. Keeping the person upright is not simple, but highly desirable. The motor turning the wheels does not allow direct control over the angle the Segway makes with the ground.

Surgical robots could mimic the ability to gently push away an artery, much like a human would. Industrial applications where small and fragile parts are used becomes possible.

The world is designed for humans, and we use non-prehensile actions all the time. Without modifications to its an environment, a robotic system only capable of prehensile actions cannot freely interact with its surroundings. A humanoid type robot with non-prehensile movements can freely interact with world without modifications