Good morning everyone. My project is the control of a rolling-balancing mechanical system: 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 implementation process. ||

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.

In the context of robotic manipulators, constraining the system to fully actuated scenarios severely limits their ability to achieve the performance they are capable of. Without exploiting the dynamics of the system during control actions, the full kinematic envelope can never be utilised, which motivates research into developing further understanding of underactuated systems.||

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 grab 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 use grasping techniques, such as the gripper manipulator shown on screen.

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.

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.

The layout of the system used for modelling is shown on screen. The upper disk is called the object, and the lower disk is called the hand. 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.

By rearranging the Euler-Lagrange model to form an equation with the second derivative of position as the subject, numerical integration tools can be used to determine the state of the system for any time. Anyone 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 accuracy. 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 test in simulation. The test is started with the system near the balancing position and no input is provided. Due to an assumption that the two disks are always in contact, we would expect the smaller disk to fall and swing about the larger disk. In the absence of any dissipation forces, we would expect this movement to go on indefinitely, which can be seen in the plot 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 setpoint regulation via a reference feedforward structure. ||

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, as shown in the animation. The initial position of the top disk is offset from the balancing position and the controller attempts to balance the disk. The controller then attempts to regulate the velocity of the bottom disk to 1 revolution per second in each direction whilst maintaining balancing.||

Using the simulation developed, the experiment apparatus can be designed. By simulating conditions likely to occur during control operations, the requirements of the motor and behaviour of the system with respect to parameters can be determined, as shown on screen. The two main design considerations are the speed of the poles of the system and the level of input force required. System pole speed is influenced by the physical size of the system, which in turn affects the magnitude of the input force required. Using the simulation, a hand diameter of 240mm and object diameter of 120mm are selected, which results in a potential input torque of 1.3Nm. This system exhibits stability with control frequencies as low as 10Hz, which should be easily achievable. DC motors capable of providing this level of input torque can be found, typically utilising large ratio gearboxes. ||

With the parameters determined from the simulation, the apparatus can be designed. Here we can see the 3D CAD model on the left and the final product in the right two images. The disks are made from clear polycarbonate to reduce weight and cost, whilst the frame is made from 20mm aluminium square hollow section. The enclose plates are also made from clear polycarbonate as they must be transparent for the computer vision measurement system. The motor and electronics are housed under the U channel section which runs the length of the frame in the centre. The blue dots in the right image are markers for computer vision feature detection.||

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. Using a video camera, measurements of the disk position can be obtained from the video feed, with a bit of calibration. ||

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. We can correct for these errors using a series of photos of the checkerboard pattern shown here. The pattern is photographed in several positions in the 3d workspace 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 are 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 on the right. 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 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 left is the undistorted test image in its HSV version. We can see the blue circles are 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 system is known, as is the region the 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 process of saturation thresholding and blob detection can be carried out in MATLAB at upwards of 50Hz, making the limiting factor in measurement frequency the frame acquisition rate.||

Two cameras were used to capture images. The first was a Microsoft vx3000 webcam, which ironically was not compatible with windows 10 and had to use 3rd party software drivers. These drivers were terribly inefficient and introduced significant delay into the video stream. To replace this camera, a raspberry pi camera was acquired, which can deliver 640x480 video at 90fps without a reduction in field of view. The camera interfaces with MATLAB using the image acquisition toolbox, which allows capturing of images from a video stream, but provides minimal control over the method of capture. This ultimately became a major issue, with a 100millisecond delay between event and image availability becoming apparent with testing, regardless of frame rate. ||

Much like the computer vision system, the DC motor underwent a calibration process, known as system identification. This process involves developing a model that captures the electrical and mechanical behaviour of the system and determining the parameter values using numerical optimisation. System identification allows for any non-ideal behaviour in the physical system to modelled and accounted, enabling precise control over the motors behaviour. With the standard DC motor model shown in the upper picture, we expect a sinusoidal input voltage to result in sinusoidal output velocity and current, which is clearly not the case, as shown in the lower plot. There is a phase offset between the velocity and voltage, along with a significant deadzone. ||

Through the system identification process, it was identified this behaviour is a result of several different types of friction in the motors gearbox, and a friction profile was developed, combining Stribeck, coulomb, viscous and static friction models. The plot on screen shows the friction profile as a function of motor shaft velocity.||

Using the developed friction profile, we can simulate the system and determine the difference between predicted outputs and measured outputs for a given input voltage. A plot of the simulated velocity and current for a sinusoidal input voltage using the friction profile developed is shown on screen. There is excellent agreement between the measured and simulated velocity profiles, and relatively poor agreement between the measured and simulated current profiles. The poor agreement between current profiles resulted in extremely poor performance using current based control allocation. ||

Proportional-integral control is a model invariant method of control, which can be implemented as a control allocation method. By implementing a PI control loop at a frequency far higher than the control frequency, extremely good performance can be achieved without any requirement to accurately capture any non-ideal behaviour of the system. As shown in this plot, there is extremely good agreement between the demanded and measured velocity using a PI control allocation scheme.||

Implementation of these components has been complete for several weeks, and extensive research is ongoing into reducing image acquisition input latency, which is causing instability. At this stage, the system is behaving as expected given the control frequency, which is limited to less than 10Hz due to a greater than 100ms delay in the video stream.

sysID and camera issues

questions for people to ask me:

* More on camera
* Explain sysID process – smooth vs non-smooth functions
* Velocity control on DC motor – modelling?