MSE 222 FINAL PROJECT

Silicon Whalley

Name	Student Number
Ashley Powell	301293252
Ataur Rehman	301253848
Muhammad Talha Mansoor	301290686
Parshant Bombhi	301255126

Introduction

For this project, we require to design a dynamic system that provides the fastest path for a marble to reach the endpoint. We were also required to predict the total time for the marble to reach the endpoint.

The marble ball track is meant to serve as a representation of the engineering knowledge acquired over the years. Therefore, the design must incorporate and optimize on the key ideas of the project. Mainly the design must;

1. Minimize travel time
2. Meet Minimum design criteria;
    a. At Least one sloped, flat and curved section
    b. Vertical and horizontal direction changes
    c. Passive System (no electronics)
3. Constructed from recycled material
4. Be creative and new

Furthermore, the predicted outcome of the simulation must be within the performance outcome of the track ball run. The prediction is very important as they demonstrate the group's mechanics skills, and will account for final mark. Minimizing travel time and making accurate simulations were the main objectives of the project and were heavily time-funded

Design

The design of the track was begun via brainstorming several different ideas. These were scrutinized with the design objective with a heavy emphasis on limiting traveling time. The main ideas of the track were prematurely simulated in Powder Game[1] which is a rudimentary physics simulator. The track in the game was made of metal and dropped in a gravity environment

Our first idea was to use a tunnel to direct the ball to a drop where it would fall on a series of slanted pieces of wood, the marble would then bounce from one piece of wood to another till falling on the endpoint. The tunnel would be flat at the start would curve near the end to get the marble moving the right direction for the bounces. The bounces would account for our down to up and our right to left movement. The reason we didn’t choose this as our final design was because we wanted to reduce the complexity, we thought that the bounces would be very hard to angle correctly.

[1] http://dan-ball.jp/en/javagame/dust/

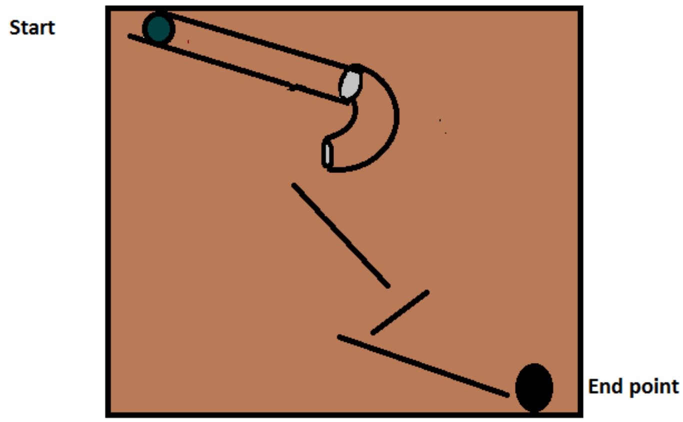


Figure 1 Design One

The second design was purely a tunnel as we thought this would be the simplest track to build. We would let the ball roll down the tunnel and launch it up a ramp at the end to reach the end. The tunnel would be flat at the start and curved throughout, the ball would experience a change of direction from right to left in the tunnel and from down to up at the ramp. In the end we didn't choose this design because we thought the marble would take too long to reach the ramp if we used the tunnel to change directions.

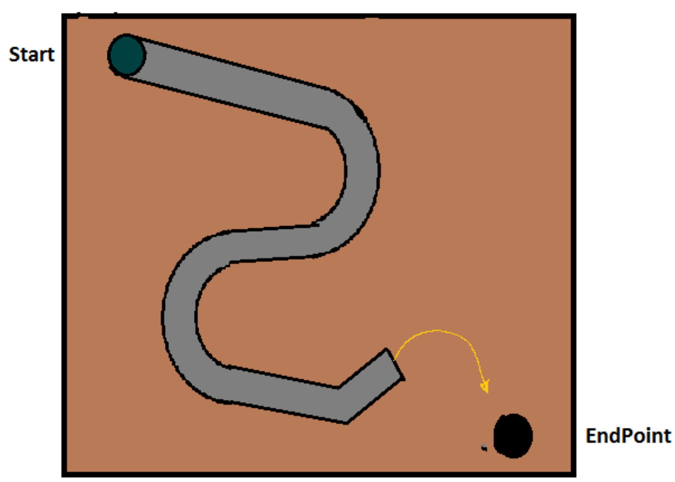


Figure 2 Design Two

The third design and the selected design includes a sloped flat path which leads to curve and then a bounce off two sections till it falls into the endpoint. The main idea behind this design is the brachistochrone curve which is mathematically known for being the fastest path between two points. The curve covers a large portion of the space because we wanted to move the marble from the start point to the ramp as fast as possible[2]. The ramp launches the marble to a flat piece where the marble bounces off onto another flat piece and finally onto the endpoint.

[2] https://en.wikipedia.org/wiki/Brachistochrone_curv

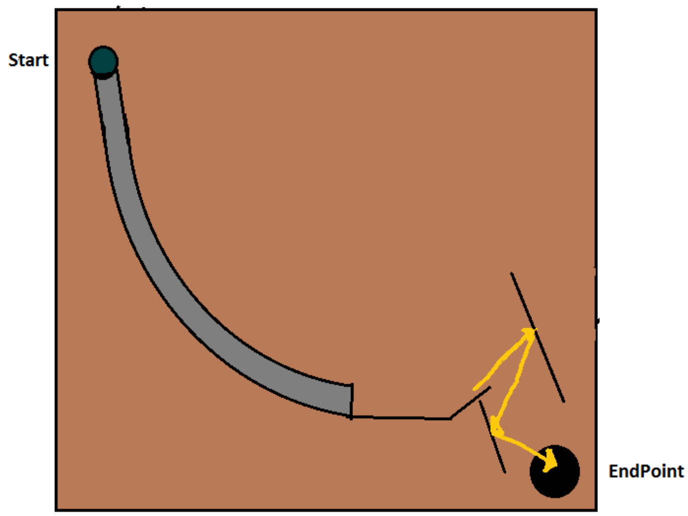


Figure 3 Design Three / Final Design

Using the Powder Game application the simulations of the tracks were run in turn leading to times of 4.01, 3.47, and 2.16 respectively. The third design was selected for further research/simulation in Matlab. The design superiority is due to the brachistochrone curve used, which allows the ball to minimize time across the horizontal plane.

Final Design

The brachistochrone gives us the optimum path between two points. In order to make the maximum use of this curve, we need to initially start with a slope to ensure that the marble is completely rotating when it enters the brachistochrone curve. When the marble exists the brachistochrone curve, we added a flat part to ensure a smooth transition towards the slope for a bounce. We decided to have bounce in the design to reduce friction caused by surface of the wood while incorporating the change in x direction and y direction. A detailed sketch and coordinates of the components is the figure below.

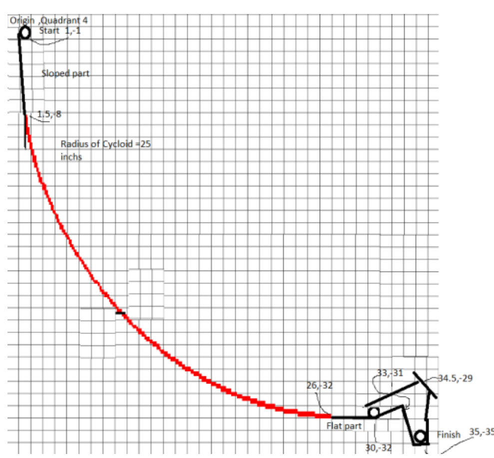


Figure 4Final Design

Matlab

The track was modeled using Matlab and all the code was based off hand calculations. The structure follows a simple 5 while-loop plan, the 3 main loops calculate all the major parameters of each section (i.e. slipping, rolling without slipping, bounces). Then these are plotted using Matlab's built in functions. All values calculated values (i.e. velocity, angular velocity, positon) are stored in an array, which are later manipulated and then plotted. All the assumed constants are based on either research papers, online data, or are determined using experiments. However, the constants have intrinsic error, which is negated by varying the error to +/- 10 percent, and later graphing to determine the effect of the constant on the system. Using Matlab to simulate the complex system was difficult to implement, however the program modelled the real-world system extremely closely.

Graphs

X and Y position of the ball

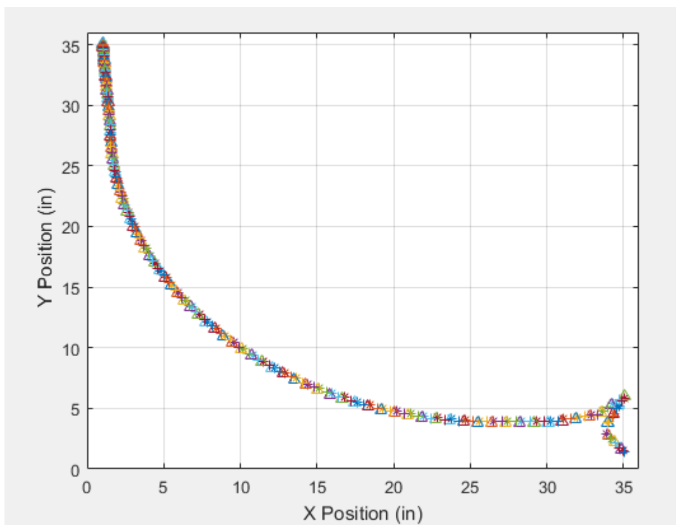


Figure 5 X and Y Position of the ball

The X and Y position of the center of gravity of the ball is given by the graph. There is a fixed time increment of 0.083s between each point. As expected, the ball follows the designed path starting with slope, then brachistochrone, flat part, slope and bounce respectively. We can also see that the plots are closer at the initial starting slope and then get further away. This shows that there is acceleration as the marble travels through the path. The '*' is the position of the ball without account for variance, '^' shows the +10% error and '+' shows the -10%error. The results are expected as well and are very close to the actual value for the position of the ball.

Magnitude of Linear Velocity vs Time

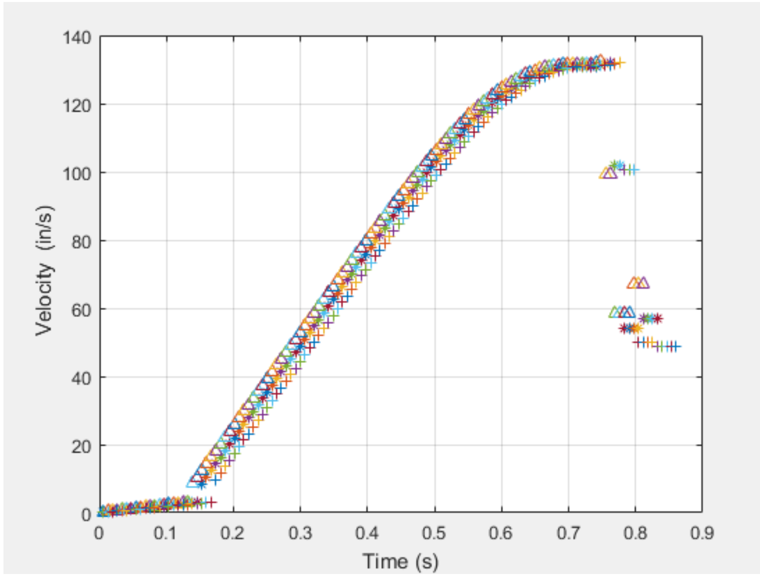


Figure 6 Linear Velocity vs time

As noticed in the Linear Velocity vs Time graph, the velocity is low initially as it starts from rest and then it accelerates rapidly. After 0.15 s, the ball stops slipping and then enters the brachistochrone. This is where there is a rise in slope since the brachistochrone provide the path with maximum acceleration. At 0.76s, the ball encounters its first bounce and about 0.8s, it encounters its second bounce. This can also be affirmed by the demo test which gives us a total time taken to be 0.81 s.

Linear Acceleration vs Time

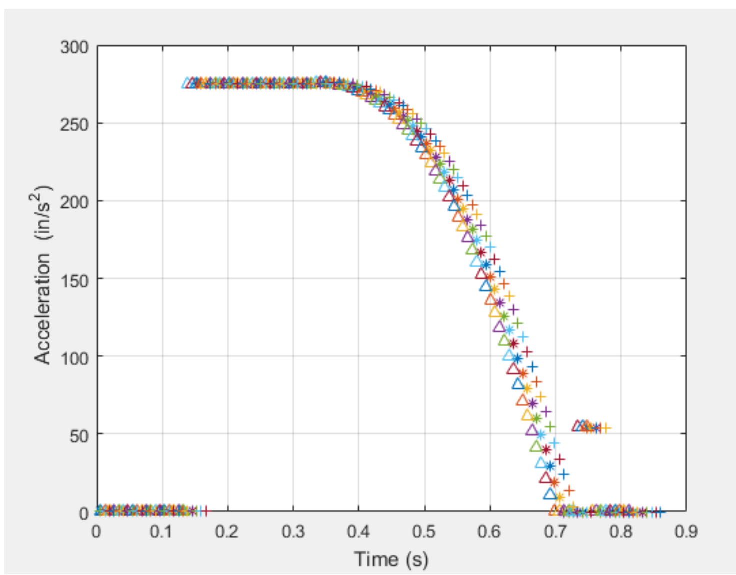


Figure 7 Acceleration vs Time

As noticed in the acceleration vs time graph, the initial acceleration is about 0, which suggests that the marble is at a state of rest and then slowly accelerates. At 0.12 s, we can observe that the marble has the maximum velocity because it is close to a free fall state due to the steep initial slope and the acceleration is high for the moment the marble enters the brachistochrone curve and as it exits the brachistochrone curve, the acceleration decreases. As it encounters the bounces, the acceleration is mostly caused as a change in direction of velocity which is again affirmed by the graph

Angular Velocity vs Time

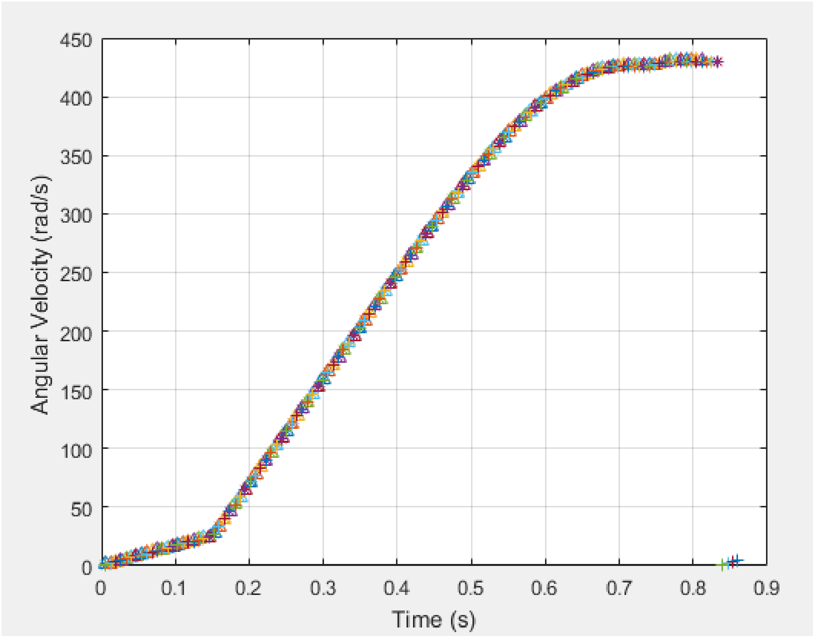


Figure 7 Angular velocity vs Time

As noticed from angular velocity vs time graph, it is similar to the linear velocity vs time graph but it initially starts with a slope which means that initially, it does not start with angular velocity and the angular velocity gradually increases which suggests that the marble is sliding. When the slope of the angular velocity graph increases linearly at 0.5s, it means that the angular velocity is now a function of velocity.

Angular Acceleration vs Time

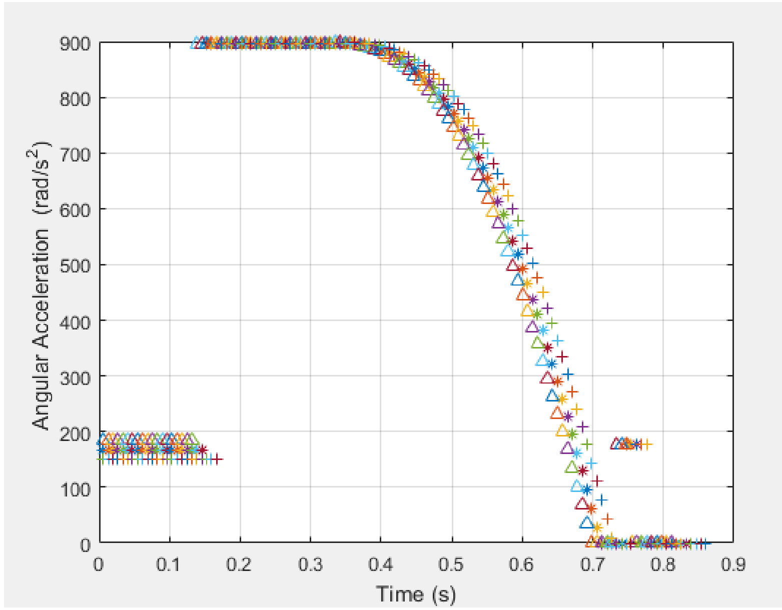


Figure 8 Angular acceleration vs time

In comparison with the acceleration vs time graph, the angular acceleration vs time graph looks similar which concludes that the tangential acceleration contributes mostly towards the motion of the ball and that the normal acceleration does not contribute towards the motion of the ball. The angular velocity is also a function of radius. This suggests that $\alpha = \frac{a_t}{r}$.

Sensitivity Study

Theoretical Time	0.8379s	
Parameters	10%	-10%
Friction	0.8239	0.8547
Coefficient of restitution	0.8309	0.8477
Moment of Interia	0.861	0.8134
Mass Ball	0.8169	0.8645
Radius Ball	0.7945	0.8932
Curvature of Slope	0.8369	0.8379
Average	0.82735	0.8519
Standard Deviation	0.022074	0.026714

Figure 9 Sensitivity Study Data

The parameters assumed of the track system play a role in determining the final travel time. In turn, it is important to determine their impact on the characteristics of the ball. Varying the main assumed parameters and running Matlab simulations allowed the standard deviation and the mean to be revealed. Mainly, the average for adding error is .82374s and removing the standard deviation shows the time to be .80533s which is extremely close to the actual value of .81s. However, doing other combination extends the difference between the actual and the theoretical. Even more, the initial parameters derived from experiments and online data may have been largely incorrect, causing a snowball effect in error. Some of the limitations are caused by the inherent lack of equipment and time available to model the system correctly.

Component Sourcing and Characterization

All materials used in the building of the system we reused from old construction lumber or materials we already had at home.

Source/Duration of prior usage:

- **Plywood and Facia board:** Leftover plywood and facia board from home construction, at least a year old
- **Pipe Insulation:** Leftover from plumbing of house, at least a year old
- **2x4:** Scrap lumber leftover from framing of house, at least a year old
- **Cardboard:** From box a new T.V came in, more than 3 years old
- **Coffee Sleeve:** From Tim Horton's, less than an hour old
- **Duct Tape and Hot Glue:** General use materials from home, more than 3 years old

Disposal:

- **Plywood and Facia board:** Recycled for future projects
- **Pipe Insulation:** Put in recycling bin
- **2x4:** Recycled for future projects
- **Cardboard:** Put in recycling bin
- **Coffee Sleeve:** Put in recycling bin
- **Duct Tape and Hot Glue:** No way to recycle, put in garbage bin

Discussion

Various members of the group did hand calculations to get the total time and the final velocity. These values were calculated using the equations of motion, kinematics equation, conservation of energy and conservation of momentum. The purpose of these calculations was to not only help the team members with coding on MATLAB but to also figure out what design changes could improve or reduce the total travel time from start to finish. However, after constructing our final design, there was a discrepancy of about 0.1 to 0.2 seconds between the calculated and the total experimental time. The main reason between the differences was due to the different values like coefficient of restitution for the material, coefficient of kinetic and static friction between the ball and our base material etc. As mentioned in the project description, 3 different values (with an increment of 10%) for material were used for the hand calculations. These values for material properties were found from an online resource[3]. Using the smallest value for coefficient of kinematic and static friction yields a total travel time of about 0.8379 seconds which is very close to our experimental or real time.

After the in class demo, we realized that although the design was one of the fastest, it wasn't very creative. Some of the other groups had almost the same design, we could have done more to make the marble track more unique. We also observed that the groups with more complicated design tended to take long to finish. In general the groups that had a simple track finished much quicker than most.

[3] http://www.engineeringtoolbox.com/friction-coefficients-d_778.html

The total time from the start to finish was about 0.81 seconds, which was the fifth fastest design. We think that this time is very approximate because the stopwatch relies on a fast reaction time.

Error

While the time calculated using Matlab was 0.8379, the time measured during the demo was 0.81 which has an error percentage of 3.3% error percentage which concludes that the theory and calculated values were accurate and close.

Conclusions

The key to success and hence to get the lowest time on this project is to make sure that your design is simple. For example using a 360 degree turn or a mere bounce off a wall to incorporate a change in direction saves a lot of time. Using the appropriate material like plywood or foam is also a good idea. Craftsmanship skills also play a major role in determining the final travel time. However, one should always keep in mind that the demo is only 5% of the entire project whereas the MATLAB plots are worth the most. Therefore most of the time should be allocated toward the coding part of the project. Finally, the most important aspect for this project is the discrepancy in the real time and the calculated (MATLAB) travel time. As mentioned above, the travel time we got on MATLAB is about ***0.8379 seconds*** whereas experimentally we got ***exactly 0.81 seconds***. This shows that our MATLAB and hand calculations were almost **100% correct** as they are extremely close to the experimental time.

Recommendations

The first recommendation would be to make the design as simple but creative as possible. As mentioned above, the groups which tried extremely fancy designs like a huge 360 degree turn etc, got the most time. Many groups also suffered in incorporating the change of direction from left to right. However, the main reason behind the success of our group was that this change in direction was done by a mere bounce of a wall near to the end. Another recommendation would be to make sure that your design never lets the ball go off track. Another aspect of this project that could be improved is that some mechanism other than a person manually timing the whole thing should be used.

Reference

[1]K. Louth, "Optimization of Modified Brachistochrone Curves Subject to Polynomial Constraints", 2011.

[2]"Friction and Friction Coefficients", *Engineeringtoolbox.com*, 2017. [Online]. Available: http://www.engineeringtoolbox.com/friction-coefficients-d_778.html. [Accessed: 20- Mar- 2017].

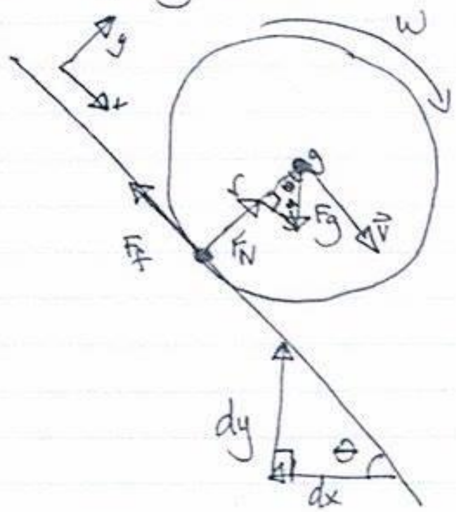
[3]"Coefficients of Restitution - The Physics Factbook", *Hypertextbook.com*, 2017. [Online]. Available: http://hypertextbook.com/facts/2006/restitution.shtml. [Accessed: 25- Mar- 2017].

[4]"Pipe Insulation Roller Coasters: Rolling Friction", *Re:thinking*, 2017. [Online]. Available: http://blog.benwildeboer.com/2011/roller-coaster-rolling-friction/. [Accessed: 02- Apr- 2017].

Appendix A

APPENDIX A.

Slipping



① $\Sigma F_x = ma, \ \Sigma F_y = \cancel{m}a^{0}, \ v_y = 0$

② $\Sigma M_g = I\alpha$

① $-F_f + (F_g)\sin\theta \overset{\cos\theta}{} = ma_x$

$-(F_n)(u_k) + (F_g)\sin\cos\theta = ma_x$

$-(F_g\sin\theta)(u_k) + (F_g)\cos\theta = ma_x$

② $-(F_f)(r) = I\alpha$

$-(F_N)(u_k)(r) = I\alpha$

$-(F_g)(\sin\theta)(r)(u_k) = I\alpha$

$-(F_g)(\sin\theta)(u_k) + (F_g)(\cos\theta) = ma_x, \quad a_x = \frac{dv}{dt}$

$-\cancel{m}g\sin\theta\, u_k + \cancel{m}g\cos\theta = (m)(a_x)$

$\boxed{(-g\sin\theta u_k + g\cos\theta)\,dt = dV_x}$ in frame 1

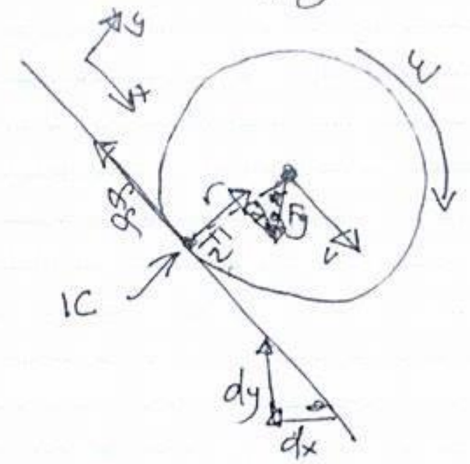
$(-g\sin\theta u_k + g\cos\theta)\,dt(\sin\theta) = dV_y$

$(-g\sin\theta u_k + g\cos\theta)(dt)(\cos\theta) = dV_x$ } in general frame

$\int\left(\frac{-mg\sin\theta(r)(u_k)}{I}\right)dt = d\omega$ → in general frame

Keep continuing calculation until $v = \omega r$

Roll without Slipping



$v = \omega r$

$$\Sigma M_{IC} = I\alpha$$

$$F_f(0) - (F_g)(r) = I\alpha$$

$$\frac{-(F_g)(r)}{I} = \alpha$$

$$\frac{-(F_g)(r)}{I} = \frac{d\omega}{dt}$$

$$\boxed{(dt)\frac{-(F_g)(r)(\cos\theta)}{I} = d\omega}, \boxed{dv = \frac{-(dt)(F_g)(r)(\cos\theta)}{I}\, r} \rightarrow \text{in frame 1}$$

in general frame. $dV_x = \cos\theta\, dV$, $dV_y = \sin\theta\, dV$

$$\boxed{\frac{(-dt)(F_g)(r^2)(\cos\theta)}{I}(\sin\theta) = dV_y,\ dV_x = \frac{(-dt)(F_g)(r^2)(\cos\theta)}{I}}$$