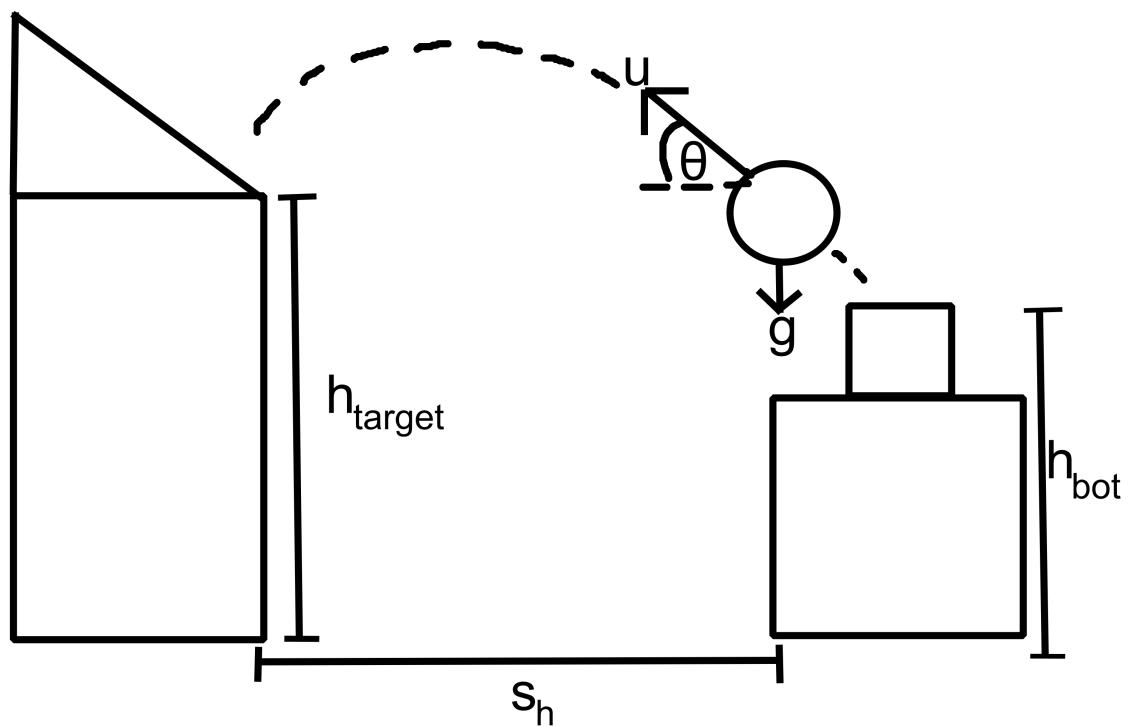


Shooter design: Change-mechmasters (FTC18494-FTC25102)

Initially, we considered using a (stationary) flywheel-based shooter mechanism as our scoring mechanism. To assess whether such a design would be successful, and to design it such that it would be effective & accurate, we had to consider the mathematical & physics principles behind it.

Defining the problem

A flywheel-based shooter functions by rotating at a high RPM, such that when the artifact makes contact, the flywheel's kinetic energy would be transferred to it by friction. Hence, we control the artifact's initial velocity, u . Then, based on the distance to the target, what is the required initial velocity for the artifact to land in the target? We can figure this out using projectile motion physics.



Therefore, we have the following quantities:

- Launch angle: θ . It is constant by the mechanism's design.
- Displacement: $\Delta h := h_{\text{target}} - h_{\text{bot}}$, so $s_v = \Delta h$, and s_h can be determined by odometry or otherwise.
- Initial velocity: $u_h = u \sin \theta$ and $u_v = u \cos \theta$.
- Acceleration: $a_v = -g$ and $a_h = 0$. We can conclude that $a_h = 0$ because there is no horizontal resultant force acting on the artifact (Newton's Second Law). This is only (approximately) true because air resistance is negligible, due to the artifact having good aerodynamics (round shape with holes) travelling at relatively low velocities.

Note also that this assumes that the bot is stationary, and that the shooter is exactly perpendicular to the goal. Compensation for a moving bot can be later considered.

Since acceleration is constant, we can use the equations of projectile motion, namely the following law:

$$s = ut + \frac{1}{2}at^2$$

This law can be used to find the following relation, in the vertical dimension:

$$\implies \Delta h = ut \sin \theta - \frac{1}{2}gt^2$$

However, we don't have a value for t . We can find an expression for it using the same law, applied horizontally:

$$\begin{aligned} &\implies s_h = ut \cos \theta \\ &\implies t = \frac{s_h}{u \cos \theta} \end{aligned}$$

By substituting for t :

$$\implies \Delta h = u \left(\frac{s_h}{u \cos \theta} \right) \sin \theta - \frac{1}{2}g \left(\frac{s_h}{u \cos \theta} \right)^2$$

Through some trivial (and cumbersome) algebraic manipulation, we can find our desired function:

$$\therefore u(s_h) = s_h \sqrt{\frac{g}{(2 \cos^2 \theta)(s_h \tan \theta - \Delta h)}} \quad \{u \leq u_{\max}\}$$

However, notice that the function is undefined when the denominator is less than or equal to zero. This may seem like a problem, but it is an advantage. An undefined point represents a point where it is impossible for the shooter to get the artifact into the target. Hence, this gives the bot an advantage, instead of missing, it will know when it has to reposition itself before shooting.

Note also that the shooter mechanism is limited to some maximum velocity, u_{\max} .

Translating initial velocity into RPM

Now, we have the desired value of u , which represents the initial velocity of the artifact, but how does this translate into the flywheel's angular velocity, ω ? Note that the FTC SDK takes in angular velocity, not RPM.

First, let v represent the instantaneous velocity of the motor, then, given the flywheel's radius r :

$$\omega = \frac{v}{r}$$

Then, consider how a flywheel fundamentally works. The rotating wheel accelerates an artifact, through the transfer of kinetic energies. The masses remains constant, so the only thing that changes is the velocities. The change in kinetic energy in each would be equal in magnitude but opposite in direction, since one gains and the other loses energy. However, some energy is lost, and only a portion is transferred, so the two quantities are actually only proportional:

$$-\frac{1}{2}m_{\text{wheel}}\Delta v^2 \propto \frac{1}{2}m_{\text{artifact}}\Delta u^2$$

Simplifying a little:

$$\implies m_{\text{wheel}}v^2 \propto m_{\text{artifact}}u^2$$

Hence:

$$km_{\text{wheel}}v^2 = m_{\text{artifact}}u^2$$

Making v the subject:

$$\begin{aligned} \implies kv^2 &= \frac{m_{\text{artifact}}}{m_{\text{wheel}}}u^2 \\ \implies \sqrt{kv} &= \sqrt{\frac{m_{\text{artifact}}}{m_{\text{wheel}}}}u \end{aligned}$$

This k is a proportionality constant, denoting the *energy transfer proportion*. This constant could, theoretically, be found mathematically; but it is much easier to simply find it by experimentation, by tuning. All in all, we have the following equation for angular velocity:

$$\omega = \sqrt{\frac{m_{\text{artifact}}}{m_{\text{wheel}}}} \frac{u}{\sqrt{kr}}$$

Further, from the above equation, we can make some conclusions about the design of the flywheel itself. First, an increase in either the flywheel's radius or an increase in its mass would result in a higher maximum velocity, u_{\max} . (See below.) Second, an increase in the flywheel's mass would result in a slower rate of acceleration & deceleration, which implies a higher throughput at the cost of a longer initial acceleration period (since more time is needed to accumulate the required kinetic energy for the desired angular velocity).

Alternatively, if the flywheel's mass cannot be calculated for any reason, the following equation can be used instead, where k would also encompass the effect of the masses. However, this approach may result in a less accurate shooter.

$$\omega = \frac{u}{\sqrt{kr}}$$

Tuning procedure

The tuning procedure for the shooter mechanism is then quite simple. One only has to measure & define the needed constants.

First: Measure the bot height and target height, where $\Delta h := h_{\text{target}} - h_{\text{bot}}$. The DECODE competition manual (pp. 64-65) states that $h_{\text{target}} = 98.45\text{cm}$. The latter value, however, will have to be measured. The measure should not be the entire height of the robot, but only from the ground up to the point where artifacts are launched.

Second: Measure the flywheel's radius, r .

Third (Optional): Measure the flywheel's mass, m_{wheel} . The artifact's mass, m_{artifact} , according to Andymark, is 0.165lbs ($\approx 0.075\text{kg}$). If this step is not possible, then the alternative equation may be used instead.

Fourth: The angle θ must be chosen and defined. The choice of angle is not too important right now; put anything to allow for tuning the rest of the mechanism. A better choice of θ can be found later.

Fifth: Tune for the energy transfer proportion, k . First, set $k = 1$. Then, try to shoot a number of artifacts some distance, s_{desired} ; then, measure (a mean of) the actual distance that the artifacts travel, s_{actual} . Repeat this for a number of distance. For each pair, calculate:

$$u_{\text{desired}} = u(s_{\text{desired}}) \Big|_{h_{\text{target}}=0}$$

$$u_{\text{actual}} = u(s_{\text{actual}}) \Big|_{h_{\text{target}}=0}$$

Plot a graph of u_{actual} against u_{desired} . ($u_{\text{actual}} = \frac{1}{\sqrt{k}} u_{\text{desired}}$, by the angular velocity equation, ignoring masses.) Draw a line of best fit, calculating the gradient, G . Then:

$$k = \frac{m_{\text{artifact}}}{m_{\text{wheel}}} \frac{1}{G^2}$$

Finally: The maximum velocity, u_{max} . This can be easily calculated given the motor's maximum RPS/frequency, f_{max} . (For reference, given RPM instead, $f = \frac{\text{RPM}}{60}$.)

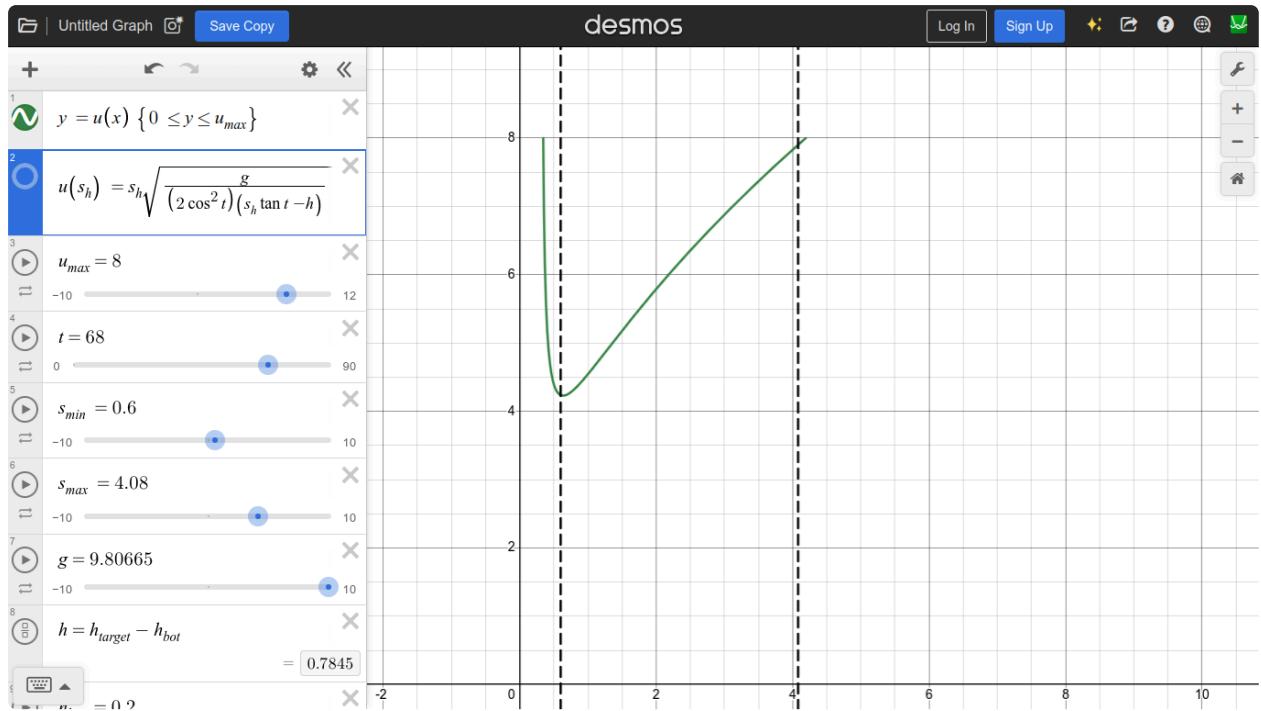
$$u_{\text{max}} = \sqrt{\frac{m_{\text{wheel}}}{m_{\text{artifact}}}} 2\pi r \sqrt{k} f_{\text{max}}$$

Choosing the angle

To choose the angle, it is possible to graph $u(s_h)$ on something like Desmos, with all of the tuning constants defined correctly:

$$y = u(x) \quad \{0 \leq y \leq u_{\max}\}$$

After, the angle θ can be adjusted until all of the function's range is defined for all of the desired distances.



This shows that, for any adequate flywheel, a mechanism with a fixed launch angle is sufficient to launch from all, or almost all, positions on the field.