FRC Team 987 Shooter Design Process

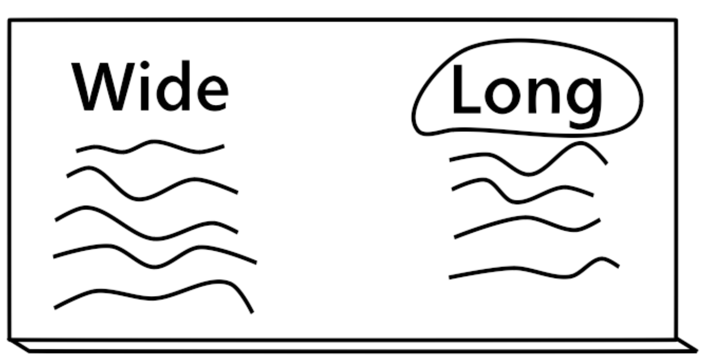
Brandon Hjelstrom, 987

Introduction

The 2012 FRC game was perhaps the most difficult yet. Making a robot that could reliably shoot basketballs into a hoop, even from a single position was the most time consuming challenge among bridge balancing, crossing the field barrier, etc. This document outlines the design process team 987 used to arrive with their final shooter.

Basic Process

When faced with multiple solutions to a problem, team 987 uses a simple process to come up with the final design solution. The possibilities are listed side-by-side, then evaluated by their pros, cons and effects on other design solutions. For example, making a shooter that can elevate variably makes turret implementation much more difficult. Making a stationary shooter means autonomous driving will become even more complex to compensate for an angled robot. This ends up looking similar to a classic design matrix by the end of the process. Note that these decisions are not always permanent – the prototyping done afterward plays a huge role in the final robot design. In fact in our process is much more hectic in reality.



Wheel vs. Launcher (catapult, kicker, etc.)

Wheel	Launcher
Multiple scoring positions (variable adjustment)	Reliable action
Experience (2006)	Works despite varying ball texture
Fed from below	Single scoring position
Torn balls, inconsistency in ball design could influence reliability	Fed from above (if launched, not kicked)
Verdict: After long discussion – Wheel	

Further prototyping showed that using a wheel was quite reliable even before advanced programming was added to help it maintain absolute commanded speeds. Several Wheel types were tested, which led to our final design making use of two (more mass + ball-centering) 8″ AndyMark wheels with rough top tread. With a second wheel on the shooter, the use of a dual 775 transmission was now completely plausible for use as well. Our final design internalized the transmission with the use of an external belt to power the wheel.

High vs. Low

High	Low
Thought to be more reliable – for our wheel type, compression, etc., it was not	No tipping, ease for driver when speeding over bridge and bump
Elevator was considered – weight, and adding a turret would be even harder	
Mentors from 2006 very worried about tipping	
Verdict: Low	

The majority of our team favored a high robot – the decision to go low came from both prototyping, and our mentors from '06 warning us about tipping.

Luckily our height did not seem to influence reliability at the time. Tipping also influenced the decision to make a long robot.

Turret vs. Stationary

Turret	Stationary
More versatile scoring and driving	Simplicity
Scoring under defense	Easily fed consistently
Heavy, Complex	Must use drivetrain to aim
Unique feeding solution needed	
Verdict: Turret	

Being able to score without lining up the base of the robot was a must for our team. In the end it was crucial – our matches were won because we could stop driving regardless of the orientation our robot was in and start scoring.

Straight Fed vs. "Popper"

Straight Fed	Popper
Extremely simple	Consistent feeding to shooter
Different feeding speeds	Hard to implement
Different feeding angles (turret decided)	
Unique feeding solution needed	
Verdict: Popper	

"Popper" – Vertically launching pneumatic device

This decision was actually one of our last as it came up during the late stages of our prototyping. Our team's main goal was to eliminate excess variables though, so we accepted the complexity to allow for a more consistent shot.

Adjustable Hood

Adjustable Hood
Our chosen angle could launch angled fender shots as well as far shots
Adds another variable to shooter
Not necessary
Verdict – No need

An adjustable hood was not necessary due to our chosen angle and height – it would only add another variable.

RPM Control

Using a machined "gear" and a Hall Effect sensor, we coded our shooter wheel to approach a commanded RPM using a P-loop with feed forward. We actually spent a large portion of our time getting this to work - we went through multiple gears and sensors without success. Once it began functioning properly, we found through prototyping that the comparison between RPM and the distance to the 3pt hoop was practically linear – there was only a slight dip in the middle of our graph.

Kinect

After the Kinect was implemented, we ran a test with promising results. Our robot could score 25 completely random consecutive shots using manual aim and Kinect RPM control. See our other whitepaper outlining how the Kinect was implemented for RPM control and later turret aiming.