Improving Driving Performance in *FIRST*® Robotics Competition

FIRST® **is a global robotics community that prepares young people for the future.**

www.firstinspires.org

Improving Driving Performance

A standard part of almost every FIRST Robotics Competition game is the gameplay loop of acquiring and scoring game pieces. This loop, going from having just scored, to acquiring game pieces, to scoring again, is often referred to as a "cycle". For most games, the majority of a team's match time is spent cycling so optimizing cycles is crucial. This guide provides some ideas about how to practice, what to practice, and how to further analyze and improve your cycle times.

In addition to this guide, you may wish to search out some of the many community resources that have been written on similar topics. Two resources that were particular inspirations for this guide are *Cycling Optimization* by Team 610 and Team 2168's *Drive Team Manual*.

How to Practice

Identifying and implementing processes and techniques to improve driver skills is important for optimizing the robot's performance. One of the most impactful ways to improve driving skills is practice. Practice doesn't have to wait for the robot to be completed; a previous robot, test drivetrain, or the bare drivetrain of the current robot can all be used to start practicing before the robot is complete. While any practice (either physically or virtually through simulators or cooperative video games) is generally effective, how you practice can be very important. There are several techniques that can help make driving practice more effective – a few of these include:

- **Start Slowly and Work Up** – When learning a new skill, it's best to begin slowly. Successful drivers often start by driving and completing tasks slowly. Once familiarity with the controls, the task, and the sequence of events is established, speed up the completion of each task. If practice at a specific speed or level needs to be extended to build confidence or proficiency, that's okay. Eventually the driver develops muscle memory and actions become "second nature," allowing the driver to begin to "practice like you play" and further refine their abilities.
- **Don't Be Afraid of Crutches** – One useful method for learning new skills is to use assistive elements. Some examples include the following:
    - arrows on the floor to help learn how to navigate a turn sequence
    - lines taped on the floor or onto field elements to assist with location or accuracy, or
    - making field elements different colors to improve contrast while learning.

  Over time, the driver picks up cues from places other than the assistive elements, such as how the robot looks when it's a specific distance from the driver or the alignment of the robot's bumpers with a field element. At that point, the assistive add-ons can slowly be removed.
- **Drive with Finesse** – Teams have different ways of saying this, including "Slow is Smooth, Smooth is Fast", but they all have the same underlying goal: approach driving with finesse, not brute force and raw speed. Completing the task carefully the first time is generally faster than having to do it twice after rushing too much. Moving too fast can cause avoidable robot damage, wasted time, or penalties.
- **Practice with Purpose** – The driver will often feel a sense of ownership and responsibility to the team to "not break the robot," causing them to not push the robot to its full potential and miss possible weak points in the robot early in the design and build phase. In contrast, following the cliche "Drive it like you stole it" helps identify and address issues before an event. Rely on the team's technical skills to keep the robot in tip-top shape.

What to Practice

To the extent that you are able, set up a practice environment as representative of the real field as possible. The Layout and Marking diagram and Team Element build guides on the Playing Field page can be a helpful reference for this:

- **Start Simple** – When starting practice with a new robot, start simple. Focus on the basic fundamentals of your robot. Test out the various functionalities you expect to use in a match to

see how they work. Drivers need to become familiar with the robot and its controls. Look for things that feel awkward or difficult as they could be opportunities for iteration.

- **Practice Drills** – Musicians know an impactful way of learning to play an instrument is learning how to play scales, or specific ordered sequences of notes. Scales helps the musician develop speed, dexterity, and muscle memory. For drivers, drills are to the robot as scales are to the instrument. Drills can include driving slalom or other obstacle/patterned courses, picking up and releasing game pieces, and repeatedly scoring game pieces into/onto a practice apparatus. Some of these drills can even be used during driver selection, as talented drivers can be identified through drills and then their skills enhanced through practice.
- **Cycle Tests** – Practice the full loop of game piece acquisition, travel, and scoring. It will feel repetitive, but the more familiar you are with this process, the easier it will be to replicate it under pressure. Consider timing cycles or recording practice so you have objective data to review and understand how different techniques impact results. Make sure to practice the different paths, actions and situations you expect to encounter in a match.
- **Add Precision** – Try drills that test and improve your driver precision. For example, use objects to enforce a narrow and optimal drive path or repeat an object pickup or delivery drill until you consistently achieve 100% accuracy.
- **Add Variables** – A driver needs to be prepared for the various things that can happen in a match. Think about all of the "what ifs" that could happen during a match and start planning and practicing for the most likely ones:
    - If there's an optimal path or scoring location, what if it's obstructed?
    - What if certain functionalities of your robot break down during a match?
    - How do you respond to defense?
    - How can you share the space you are using with your partners?

Analyze and Upgrade Cycles

While practice alone does help with improving cycles, analyzing what's is and isn't happening during those cycles can inform iteration and result in further improvements. Specific ways to do that include:

- **Think about the controls** – WPILib comes with default implementations for arcade, curvature and tank drive. You can learn more about the differences between these in the WPILib documentation. Other permutations have been released within the community that may or may not feel more intuitive to your driver. Another aspect to consider is robot-oriented vs field-oriented drive. In field-oriented driving, the driver no longer considers the orientation of the robot when moving the joystick; pushing the joystick away from the driver moves the robot away from the driver, regardless of robot orientation. Field-oriented driving is most often used with robots with omni-directional movement such as mecanum or swerve drive robots, but is possible to implement using differential drive as well. Drive controls should be figured out early, as it will take time for a driver to adapt to any changes. Make sure the controls are intuitive. It's okay if the incorrect buttons are being pressed at times while first learning, but monitor this closely as it could be a sign that the layout doesn't feel right to the driver.
- **Think about the controller** – Another thing to look at and test early can be the actual controller the driver is using. Different teams have found success with joysticks, gaming controllers, and event steering wheel & throttle combinations. Find an option that is intuitive and has the right range of motion for the drivers.
- **Analyze the cycles** – Analyze the cycles as they're performed. Break down the timing into three buckets: game piece acquisition, field traversal, and game piece delivery or scoring. Which of these three components uses the largest proportion of your cycle? How can you improve each component? Are there design or software tweaks that could improve your speed? Is there wasted time and movement? How does accuracy effect cycle efficiency?
- **Add automation** – One way to decrease cycle time is by adding automation using sensors and software. Examples include:
    - use computer vision to aid lineup with certain targets and destinations.
    - automatically position robot mechanisms, or
    - automate movement of game pieces within the robot.