Game Strategy (Advanced)

Fall 2016

Training & Development

POWERED BY DRAGON FIRE

Outline

- Game Strategy
- The Scoring Calculator
- How To Use The Calculator Throughout The Season
- Match Breakdown
- Success Rates And Their Impact
- Putting It All Together

Game Strategy Overview

Game Strategy Overview

- After the Kickoff presentation each team begins their build season plan
- What is the first step towards the Robot Design?
- Game Strategy!!!
- How is the strategy Developed
- For Team 302 it is the scoring calculator
- What will the scoring calculator provide
  - Simple answer -- the way to score the most points per second of the match!!

The Scoring Calculator

The Scoring Calculator

- Why: To provide an objective understanding for the value of each scoring method within the game
- **<u>To establish performance requirements for design elements of the robot</u>**
- How: Derive or Estimate the points per second (or seconds per point) for each method of scoring for the match
  - List Scoring Actions
  - Define Scoring Cycles
  - Detail Scoring Actions
  - Detail Connecting Actions
  - Make Assumptions and Estimate
- Compare each scoring method to develop the strategy to consistently score the most points per match
- Simple Version: $Points\ per\ Second = \frac{Points\ per\ Cycle}{Seconds\ Per\ Cycle}$

How to Develop the Calculator

- List Scoring Actions
- Define Scoring Cycles
- Detail Scoring Actions
- Detail Connecting Actions
- Make Assumptions and Estimate
- … repeat throughout build and competition season

Listing The Scoring Actions

- Why: To create a set list of all ways to gain or lose or prevent points in a match
- How: Examine the manual and find all of the actions and times that cause your score to change
  - Understand the action that causes the score
  - Understand the time in the match that the points are added to the score
    - Automatic field scoring
    - Ref score entry
    - End of match assessment
    - End of autonomous
  - Understand the penalty actions and how consequential they are, ie: how many actions does a penalty cancel out?
- Document each item and its point value

Example: How to List the Scoring Actions 2016 Stronghold

- High goal 5 pts
- low goal 1 pt
- cross defense 5 pt

- Auton high goal….
- Batter
- Hang

- Others...

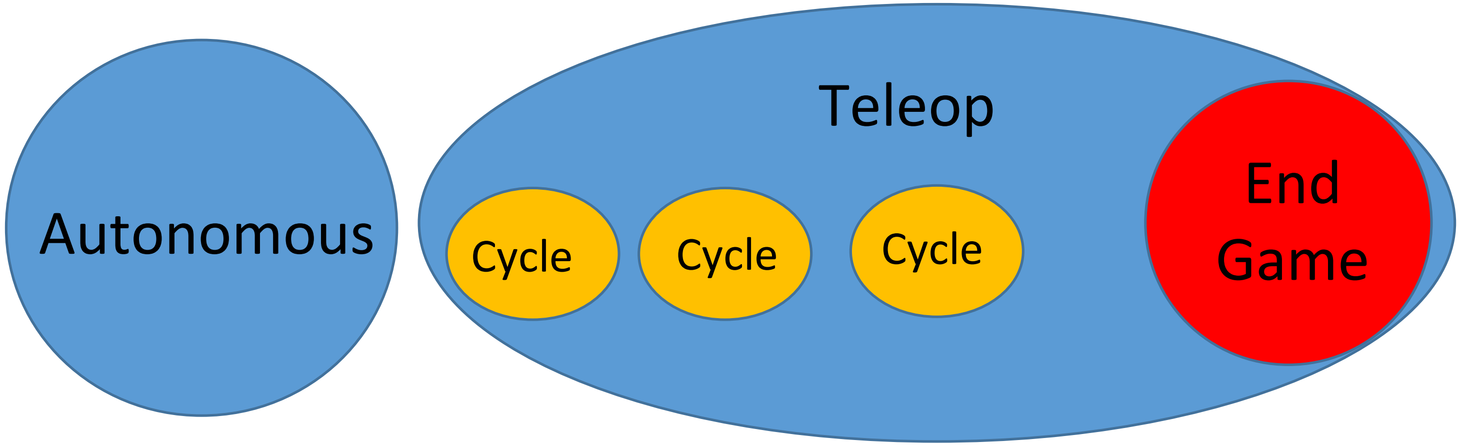
- more than 1 robot defending the castle
- controlling more than 1 boulder

- …..

Defining Scoring Cycles

Within a match there are typically many options for scoring cycles;

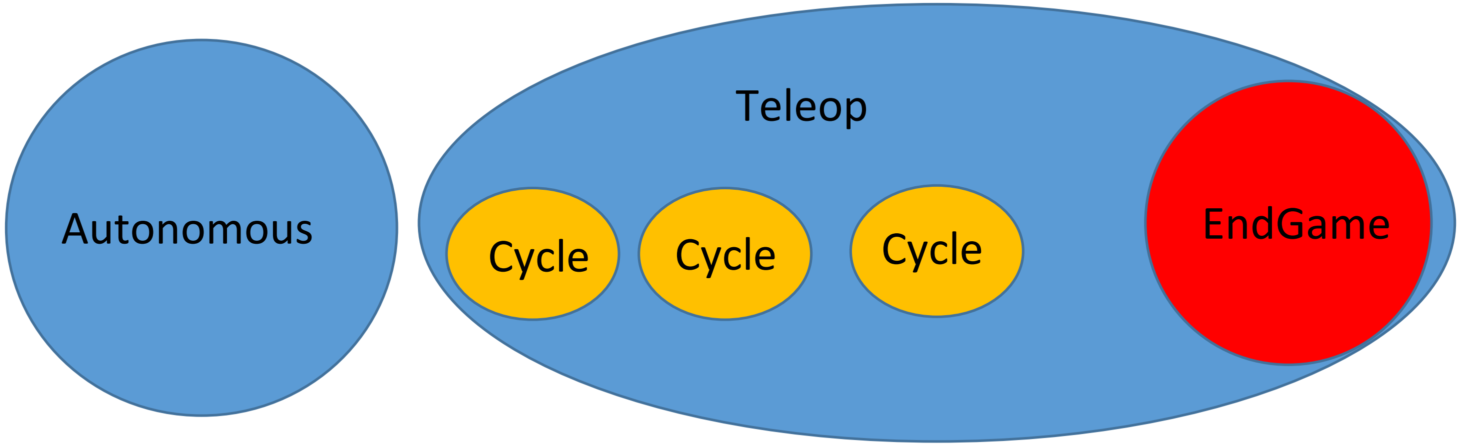
- Autonomous Cycles
- Teleop cycles
- Endgame cycles
- To define Game strategy each must be analyzed



Defining Scoring Cycles - Key Terms

Within a match there are typically many options for scoring cycles; A few Key Terms are listed below

- Cycle Score: The number of points scored during, or as a result of the completed cycle (Points per Cycle)
- Cycle Success Rate: What is the percent chance that you will successfully complete the cycle
- Cycle Time: How long the cycle takes to complete



Define Scoring Cycles

- What is a Scoring Cycle: A set of actions needed for the robot to score
- Scoring Cycle components
  - Starting point
  - Connecting actions
  - Scoring Actions
  - End point
- Connecting Action: any action that does not directly score but is needed in order to create a cycle, fewer and quicker connecting actions means more of the cycle is made up of Scoring Actions! (Cycle Efficiency)
- How to create a Scoring Cycle: Use list of Scoring and Connecting Actions to create a cycle!
- Example of Scoring Cycle:
  - Acquire the game piece (connecting action)
  - Drive to the scoring area (connecting action
  - Execute Scoring Action
  - Get to End Point (connecting action)

Example: Scoring Cycle
2016 Stronghold - after Autonomous

High Goal Score - point value = 5 pts

- Start point - end of Autonomous with boulder (did not shoot the boulder in auton) after crossing the outer works
  - •Connecting Actions -
    - •turn robot towards goal- how many degrees
    - •drive to correct/desired shot distance
    - •Adjust mechanisms to shooting position - is this parallel to the drive
    - •left /right aim
    - •launch angle (up down) aim …
- Scoring Action - the shot
  - •End Point… drive to end point (and turn around robot?)

  - •Quick exercise - what are some quick ways to improve this cycle?
  - •How many actions have 100% success rate

Detailing A Connecting Action

- What is a Connecting Action: Any action taken that does not directly score but is necessary to complete a cycle
- Examples:
  - Acquire a game piece
  - Drive to scoring area
  - Drive to human player
  - Pass ball to alliance member
  - Aim
  - Turn Robot
  - And Many More…
- Makeup of a Connecting Action:
  - Time of Action
  - Success Rate (%)
- $Expected\ Number\ of\ Attempts = \frac{Desired\ Number\ of\ Successes}{Success\ Rate}$
- If our desired number of successes is equal to 1 that indicates that we only need to succeed once before moving on (may differ depending on game rules, ie: can hold more than one game piece)
- $Expected\ Time\ of\ Action = Expected\ Number\ of\ Attempts \times Time\ of\ Action$

Examples of Success Rate of a Connecting Action affecting Cycle Time:
2013 Pickup Mechanism

Detailing A Scoring Action

- What: Any action that results in scoring
- Examples:
  - Shooting a ball at the goal
  - Placing a tote on the scoring platform

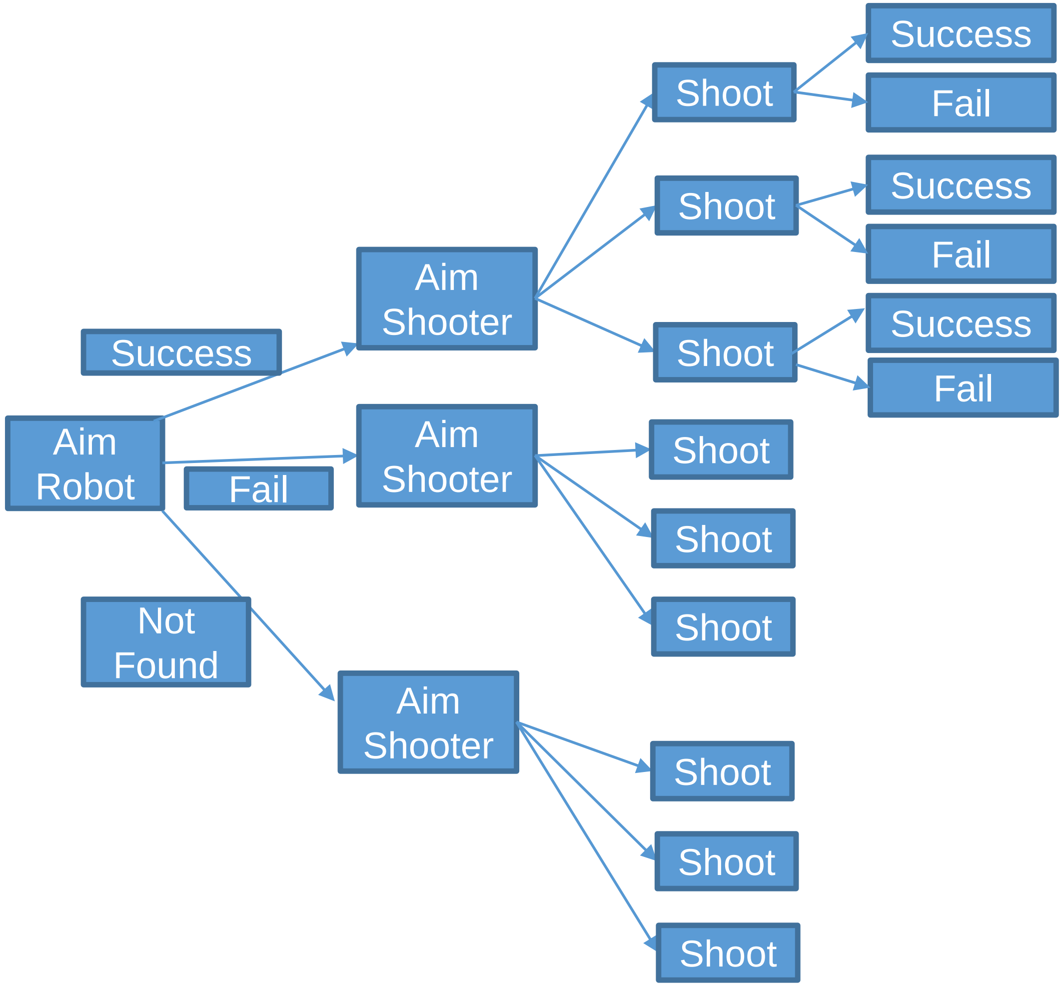
- Makeup of a scoring action:
  - Time of Action
  - Success Rate (%)
  - Number of Points
- $Expected\ Points\ per\ Action\ = Number\ of\ Points \times Success\ Rate$

A Special Case: Coupled Actions

- What: An action whose outcome drastically changes the Success Rate of an action that follows it
- Example: Consider the following sequence of actions from 2016
  - Aim the robot at the tower (side to side)
  - Aim the shooter at the tower (up and down)
  - Shoot the ball
- This is a group of actions that are “Coupled” because the completion of either of the first actions drastically affects the Success Rate of the following actions
- How do we approach coupled actions???

Determining The Success Rate Of Coupled Actions

- Draw out a probability tree
- Fill out the probability tree based on **<u>estimates</u>** (or data)
- Calculate the success rates of the actions that are affected by prior actions



Example: The Hidden Cost Of Aiming

- Consider the Coupled Action Sequence
  - Aim Robot: SR—60%
  - Aim Shooter: SR—Dependent
  - Shoot Ball: SR—Dependent
- Using reasonable estimates the Success Rate of the Action Shoot Ball comes out as 25.9%
- Now If we take the action Aim Shooter out of the sequence and recalculate; the Success Rate of Shoot Ball becomes 40% (large increase!)
- Finally if we take both aiming actions out and instead have a sequence of uncoupled actions {drive to spot, shoot}
  - The Success Rate of Shoot Ball is now only dependent on how well we perform the Action Shoot Ball!
  - In the estimates used the success rate of the Independent Action Shoot Ball was 60%

Cycle Equations

- $Expected\ Points\ of\ Action(EPA) = Points\ of\ Action\ \times Success\ Rate$
- $Expected\ Number\ of\ Attempts\ (ENA) = \frac{Desired\ Number\ of\ Successes}{Success\ Rate}$
- $Expected\ Time\ of\ Action(ETA) = ENA\ \times Time\ of\ Action$
- $Expected\ Cycle\ Time(ECT) = \sum ETA$
- $Expected\ Cycle\ Points(ECP) = \sum EPA$
- $Expected\ Cycle\ Points\ per\ Second\ (ECPS) = \frac{ECP}{ECT}$
- $Expected\ Match\ Points\ per\ Second(EMPS) = \frac{\sum ECP}{Match\ Time}$
- $Cycle\ Success\ Rate\ (CSR) = \prod(Success\ Rate)$

E→ Expected
N→ Number
A→ Action
T→ Time
C→ Cycle
P→ Points or Points per
S→ Second
M→ Match

Scoring Cycle Equation

- A scoring cycle is made of scoring actions and connecting actions
- $Expected\ Cycle\ Points\ per\ Second = \dfrac{\sum(Points\ of\ Action \times Success\ Rate)}{\sum\frac{(Desired\ Number\ of\ Successes \times Time\ of\ Action)}{Success\ Rate}}$

Scoring Action:
I shot 3 times and made 1 goal worth 6 points
Points of action = 6
Success Rate = 0.33
Numerator = 6 * 0.33 = 2

Cycle Time:
I took 9 seconds to shoot 3 times and make one goal, I took 5 seconds to retrieve a game piece each time, I took 3 seconds to aim each time

Using the Calculator to Establish Strategy Decisions that Drive Design Criteria

- Outputs of the Calculator:
  - List of Actions the robot needs to perform
  - Target Success Rates for Actions
  - Target Times for Actions
- These Cycle times need to be the basis for the design expectations of the robot mechanisms
  - If we don’t succeed in matches, did we have a bad strategy or bad implementation???

Scoring Action Simple Example: 2016 Endgame Hanging v. Batter

- Hanging:
  - Points: 15pt
  - Time of Action: 14s
  - Success Rate: 50%
  - EPpS: 0.5
- Batter:
  - Points: 5pt
  - Time of Action: 4s
  - Success Rate: 80%
  - EPpS: 1.0

$$\textit{Expected Points per Second (EPpS)} = \frac{\textit{Number of Points} \times \textit{Success Rate}}{\textit{Elapsed Time}}$$

- Opportunity cost for the 10s lost in hanging to consider as well!!!!
  - What Points per Second are possible in the difference in time?
- Opportunity cost for package and weight needed to hang v. other functions?
  - What % improvement to Points per Second are you getting compared against the % of robot weight and the % of robot development time used to achieve a cycle?

Detailed Example: Comparing Cycles 2016 Endgame

Cycle 1	Hanging	ENA	ETA	EPA
Action	drive to tower			
SR	0.9			
Time	7			
pts	0	1.1111	7.7778	0
Action	get on batter			
SR	0.85			
Time	3			
pts	0	1.1765	3.5294	0
Action	Aim at bar			
SR	0.6			
Time	3			
pts	0	1.6667	5	0
Action	Elevate			
SR	0.6			
Time	8			
pts	15	1.6667	13.333	9
ECP	9			
ECT	29.64052288			
ECPS	0.303638368			

Cycle 2	Batter	ENA	ETA	EPA
Action	drive to tower			
SR	0.9			
Time	7			
pts	0	1.1111	7.7778	0
Action	get on batter			
SR	0.85			
Time	3			
pts	0	1.1765	3.5294	0
Action	Twist			
SR	0.85			
Time	2			
pts	5	1.1765	2.3529	4.25
ECP	4.25			
ECT	13.66013072			
ECPS	0.311124402			

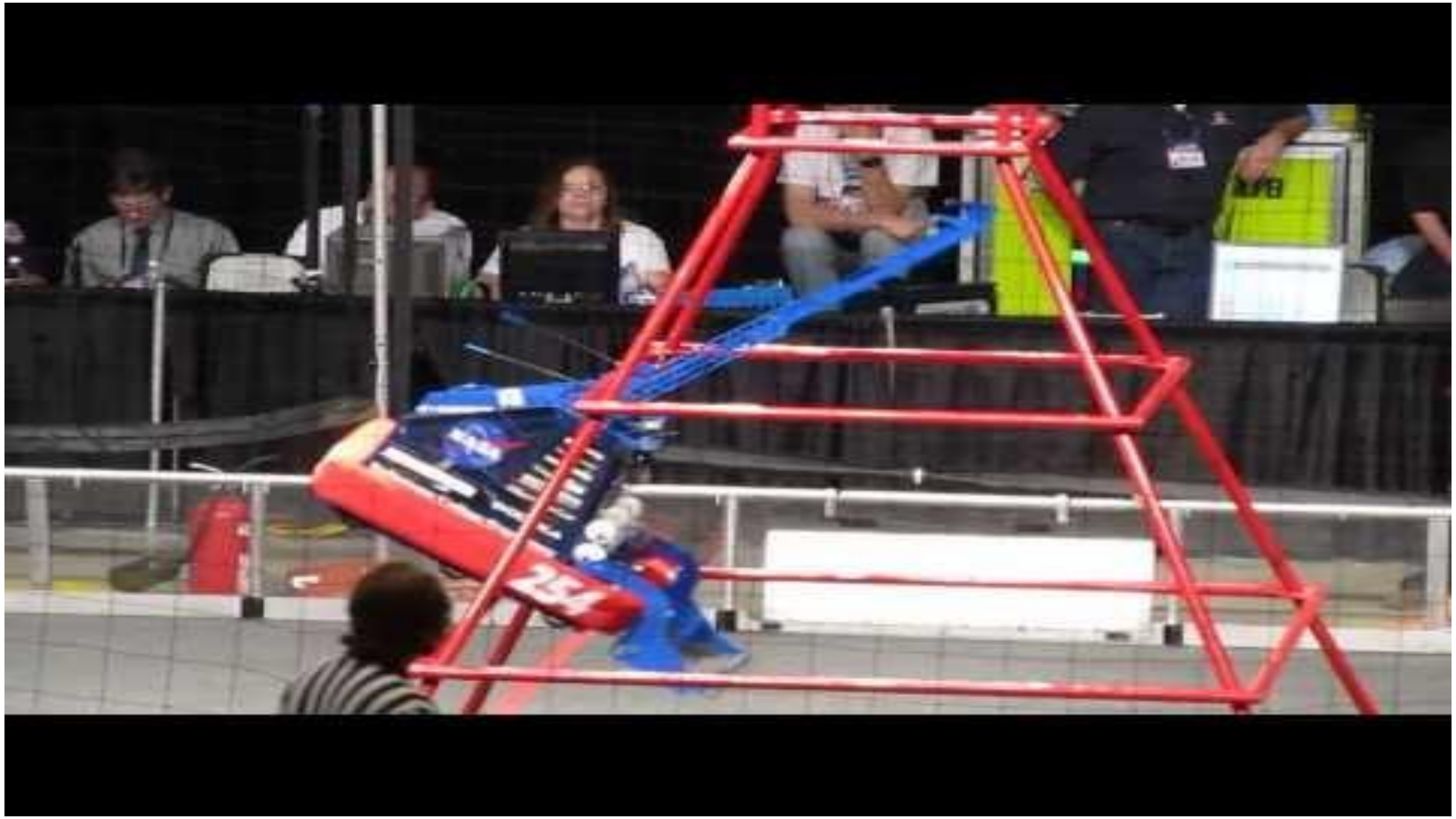
Hang to batter ECPS ratio
0.9759

Cycle minimization

- What: Reducing the number of actions in a cycle
- How:
  - Cut out unnecessary actions
  - Reduce time of actions
  - Increase success rate of actions
  - Avoid coupled actions
- Parallel v Series Cycle actions
  - Can actions be performed in parallel?
- Look at how to design cycles that are efficient
- Examples:
  - 2013 254 pyramid
  - 2013 full field shot

Minimized Cycle

- Team 254: 2013 Pyramid Climb



Cycle Minimization: Connecting Action Examples

- 2016 HOT Bot, never turned.
- 2011 1503, Efficient cycle
- 2016 254, didn’t aim.
- 2012 HOT Bot, didn’t need to turn
- 2011 27 didn’t need to turn
- 2014 HOT Bot didn’t need to turn

- Wait a second…
- All these robots won districts, states, or were on Einstein!

Making Assumptions and Estimations for a Scoring Cycle

- Why: We won't have any actual numbers to use on kick off day
- How Do We Use This: Iterative Estimation
  - Estimate Scoring Action and Connecting Action Time of Actions based on experience
  - Vary the Success Rates to calculate a table of ESTIMATED Cycle times
    - Calculate a table of overall cycle times with the varied success rates
    - Decide on a minimum acceptable success rate to make the Cycle worth doing
    - Decide on a maximum acceptable Cycle Time to make the Cycle worth doing
  - Use simple motion studies (kinematics) to improve Time of Action estimates
    - Do the math to check if estimates are realistic
  - Repeat the first two bullets of Iterative Estimation
  - **Make a DECISION**: Is this cycle worth doing!
  - Decide on which cycle(s) to move forward with as a strategy

How To Use The Calculator Throughout The Season

The Calculator's Uses Through The Season

- Decide on strategy
- Compare concepts
- Compare prototype performance
- Review mechanism performance
- Assess robots at competition

Match Breakdown

Match Breakdown

- Tele-operated Scoring
- Autonomous/Hybrid Scoring
- End Game Scoring
- Co-opertition Scoring

- Identify Cycles for each game period
- Look at expected times and expected points
- Prioritize what will earn the most Points Per Match, given the time to perform the tasks

Autonomous Strategy Considerations

- Look at what is a standard autonomous mode
- Look at the “improved autonomous” mode
- Example: 1 game piece v 2 game pieces
- Where do you want the robot to end up?
- Analyze the Points per Second possible during Autonomous and how to connect it to Teleop
- Does any Autonomous action directly contribute to gaining more points in Teleop (i.e.: grabbing a can from the middle of the field 2015)

Success Rates And Their Impact

Success Rate And Their Impact

- What effect do the success rates of actions have on the success rate of the cycle, and how long the cycle take?
- The Cycle Success Rate is the product of all the independent Actions success rates. Example:
  - Action Success Rate (100%, 100%, 100%, 100%, 10%)
  - Cycle Success Rate is 10%
- How is Cycle Time affected?
  - Failing to complete an action means repeating it
  - Every repeated action adds to cycle time!
  - Low Success Rates lead to long cycle times
- Driving and defense built into the success rate
- Success Rate decreases as number of ways to fail increases
  - KISS principle

Putting It All Together

Putting It All Together

- Goal: Selecting the Match Strategy and turning that into Robot Design
- How: Selecting the best cycle using the ECPS (Expected Cycle Points per Second) analysis for the autonomous and tele operated periods of the game
  - For the end game compare the ECPS and ECT (Expected Cycle Time) of the endgame to the ECPS and ECT of the tele operated cycle
  - Example: Remember the 2016 Stronghold example
    - The opportunity cost

Putting It All Together

- Selecting the Match Strategy and turning that into Robot Design
- How much robot resources, and how much time should the team put into different parts of the robot based on points per second comparison
- Point value tradeoff—Opportunity Cost
  - What Points per Second could you be getting in the time you are performing an action
- The more accurately we can assess our team's capabilities the more accurate the calculator will be
- We can input scouting data into calculator to evaluate robots at competition

Questions?

Auxiliary Slides

Exercises and Examples

- 2056 Every Year
- 1114 2010, 2015
- 2013 Pyramid rung tradeoff
- 2013 Cross field shot
- 2010 Gorillas
- 2012 Mid v High goal
- 2014 Truss to human player
- 2015 Human loader v mid tote collection
- 2011 Create a Logo or next available
- 2011 minibot race
- 2009 Super Cell (2834)
- 2009/2011/2014 v 2010/2013 Design Choices (302)

Min/Max Scoring Analysis

Min/Max Scoring Analysis

- What: Calculate the minimum and maximum theoretical scores for the game
- How:
  - Minimum: usually 0, check the rules!
  - Maximum: Is there a ceiling to the score??
    - If there is, what is it and what must you do to achieve it, and derive time targets for all actions from there
    - If not, use time estimates to find a reasonable maximum score within the allotted game time, improve estimates, then use them as time targets for actions
- Why:
  - Establish if a defensive strategy can prevent a maximum score
  - Shortcut our analysis. If there is a ceiling then there is a determined sequence of actions that must happen to get there. Can we design to complete them

Min/Max Scoring Analysis

- How to find the maximum possible score, if there is no absolute ceiling
  - Use scoring cycles with a success rate of 100%
  - Calculate how many of the fastest scoring cycle can be done within the time limit
  - Calculate how many of the largest point scoring cycle can be done within the time limit
  - Gives you a reasonable range of target match scores
  - Is there a combination of cycles that results in a higher score?