Robotics Competitions

Design, Building, and Management

Revision 1.05
24 December 2017

Team Magma
FRC Team 3008
Kalani High School

Table of Contents

Revision History

Date	Version	Writer	Changes
4/17/2016	1.00	Wayland Kwock	Initial document
4/17/2016	1.01	Wayland Kwock	Expanded the Appendix to include more cognitive conditions
4/19/2016	1.02	Team Magma	Added lessons learned from 2016 competition
4/16/2017	1.03	Wayland Kwock	Added lessons learned from 2017 competition
12/6/2017	1.04	Wayland Kwock	Added Dunning-Kruger effect in Appendix A
12/24/2017	1.05	Wayland Kwock	Moved and improved design considerations.

Purpose

This document is intended to provide guidance in the execution of a single robotics competition event from the analysis of a new competition to the completion of the build of the robot.

The actual strategy of the playing of the game is left open as the game is constantly changing, however, the strategy will always be reflected in the design of the robot.

This document is based upon observations garnered from observing FRC competitions, but the analysis and project management tasks should be applicable to any robotics competitions.

Game Analysis

This game analysis phase is an adaptation of the "Systems Engineering" powerpoint presentat from Beach Cities Robotics - Team 294 provided by Dr. Richard Wagner. The original document is available at:

http://rjwagner49.com/Robotics/Instruction/SystemsEngineeringv25.ppt

Additional resources are available at: http://rjwagner49.com/Robotics/Kalani/

Upon receipt of the actual robotics competition, the following tasks should be performed:

Analysis of the Game

Read and understand the rules of the game. If you don't know what you have to do, can do, and cannot do, you cannot play the game effectively.

Note the negative spaces in the rules. Non-prohibited practices are, by definition, permitted. This area may prove useful.

Analysis of the scoring of the game

The maximum possible score of the game should be determined.

> In addition to the number of points associated with a particular task (hitting a goal, completing a task), this should also take into account the number of scoring events that can be done within the time limitations of the game (i.e. autonomous period vs. scoring goals for the duration of the game).

Time limitations should also examine the number of potential scoring cycles possible. For instance, each method of scoring points will normally have a different scoring value based upon the difficulty of the scoring method and method will have a different time cost based upon the tasks involved: chasing down a game piece, transporting it, locating the goal, aiming, shooting.

The number of possible scoring cycles should be estimated for each scoring method and the maximum number of points should be based upon this estimate.

From the maximum possible score determined previously, a reasonable number of scoring events should be established for each scoring method. This is different than the maximum score estimate. This estimate should reflect the difficulties of driving in a crowded arena, confusion from noise and conflicting instructions, trouble wrangling game elements, etc.

A 40% probability of scoring a 10 point goal is equivalent to a 100% probability of scoring a 4 point goal - the difference is the difficulty and complexity of hitting the 10 point goal. If the number of scoring cycles for the lower difficulty goal is higher, the potential payoff of the easier goal is higher.

At this point, beware of optimism bias. It is attractive to posit that scoring difficult goals is a certainty, but that has to be supported by the design and implementation of the robot. (i.e. 100% accuracy on a difficult goal is attainable with video goal detection by a camera, wheel encoders on shooting motors, and precise angle control, but if those elements are not implemented, estimating 100% accuracy isn't a realistic assessment of the situation).

Analysis of the Playing of the Game

The purpose of this task to to explore the dynamics of the game. There will always be differences in how people/teams will interpret how best to play a game. This procedure is designed to allow varied opinions to express themselves.

To complete the game playing analysis, the game playing field should be mocked up in a reasonable scale that will allow individuals to role-play as a team. Alternatively, a scale desktop model can be created with markers used to represent robots - like a game of monopoly.

Each mock team is allowed to simulate a member of the robot alliance and is allowed to express their robot design as part of a game.

Possible simulated roles are:

- Offensive robot - scoring goals
- Helper robot - moving game elements to offensive robots

- Defensive robot - getting in the way of opponents
- All in one robot - can do everything. Doesn't need help from others

The point here is to establish whether playing a game in a certain role is 1) valid, and 2 advantageous.

Treat this effort as playing chess, but you get to make up the moves up to a certain point.

Game Nuances

An additional consideration at this point is how the playing of the game affects the design of the robot.

For instance:

- Is the game a fast paced game? That would put an emphasis on speed. It may even be a penalty for mobility.
- Does the game reward mobility? That would put an emphasis on how the robot moves. This may penalize speed (if you can't put the robot where you want it, it doesn't matter how fast it can potentially move.)
- Are there a lot of scoring events available during the game? That would put an emphasis on being able to maximize the number of scoring opportunities during the game.

With insight in how the team wants to play the game, it's time to design the robot.

Designing the Robot

What does the Robot have to Do?

First determine what the robot would need to do to accomplish its goals. Possible tasks are:

- Move
- Gather and hold a game element
- Gather and hold several game elements
- Pass a game element to another robot
- Accept a game element from another robot
- Accept a game element from a human station
- Move autonomously
- Score a goal autonomously
- Shoot into a low goal
- Shoot into a high goal
- Climb a game element
- Block other robots from shooting
- Locate a goal using a camera

- Stay shorter than a threshold height
- Lift a game element
- Push a game element

Once all the tasks are identified each task should be ranked as to its importance to the playing of the game as determined by the team during the analysis of the game.

The higher the importance, the higher the number rank of the task. Using a 1 to 10 scale, moving would be a 10.

Ranking number can be re-used to establish the importance of each task relative to another task.

How is the Robot going to do it?

At this point, it’s time for all the ideas about how the robot should be made to be heard from all the team members.

From the announcement of the game till this point, some ideas have been incubated and refined. This is when all those ideas are evaluated.

Each person with a robot design should present their design along with their solution to address the requirements of the game.

How well does the Robot meet its goals?

Each robot should be evaluated and scored on each of the tasks identified as what the robot must do using a range where the higher number is better.

In the following example, the scoring is from 0 to 5 with 5 being the best.

Task	**Task Weight**	Robot A Score	**Robot A Weighted Score**	Robot B Score	**Robot B Weighted Score**
Move	10	5	50	4	40
Shoot Low	7	5	35	4	28
Shoot High	5	3	15	4	20
Defend	3	2	6	5	15
Total			**106**		103

By multiplying how well each robot does a task against the weight of the task gives a score that can be totalled. The higher scoring robot would be the preferred design.

Designing the Robot - CAD

Design Guidelines

- Shafts should be in double shear - no unsupported ends. Shafts unsupported at one end can shear (Recycle Rush)
- Heavy pieces of metal should not be joined by lighter pieces without a good reason - i.e. don’t have ⅛” pieces held together by 1/16” pieces.
- If an accurate design element depiction of a part is not available, CAD a solid that would take up the same volume as the part. A motor can be represented by a block. This will enable you to see where parts can interfere with each other or interfere with playing the game. (Stronghold)
    - Motors
    - Compressor
    - Wheels
    - Transmissions
    - Air accumulators
- If there’s a rotating element to the robot like a flap wheel, CAD a solid where the travel of the wheel would encompass to ensure that it doesn’t interfere with anything else.

Mocking up the robot in CAD as best as possible, even sacrificing pretty pictures of robots for ugly blocks, allows other teams to start mocking up their designs - specifically the Electrical team.

Design Considerations

During the design of the robot as well as design reviews and peer reviews, consider the following:

1) Does the robot fit in the size constraints? Will it be difficult for the design to fit at all?
2) Would there be an advantage to making the robot bigger?
    a. Easier to maintain / service
    b. Easier to manufacture
    c. Could reducing the perimeter allow for robot expansion during the game?
3) Would there be an advantage to making the robot smaller?
    a. Easier to ship
    b. Faster to manufacture
    c. Less mass to drive around
4) Have all the electronic / control elements been accounted for?
    a. Electronics
    b. Battery
    c. Wheels
    d. Compressor
    e. Air accumulators
    f. Drive train
    g. Motors
    h. Wiring harness and routing
    i. RoboRIO
    j. Robot light
    k. Power switch
    l. Power distribution board
    m. Bridge
    n. Sparks / Talons
    o. Voltage regulator
    p. Pneumatic controller
5) Are all the electronic / control elements set up in a way to facilitate installation, troubleshooting. If not, consider redesigning the robot to make it more modular.
    a. Do you have room to add a layer to the robot?
    b. How about an additional vertical surface?

6) Are all electronic / control elements easy to access but protected from projectiles?

Do the Math!

During the design, calculate the forces necessary to accomplish goals instead of guessing. Prototyping is a helpful exercise, but actually knowing the real numbers is more professional and comforting. (Imaging how you would feel in an airplane if the designers only guessed at how much lift the wings had.)

Some of the calculations that should be done include:

- Torques on arms that have to lift things
- Masses of arms that have to lift things
- Drive ratios of motors that have to move things (is it fast enough? Is it powerful enough?)
- Force necessary to move a robot part
  - Mass of the part to be moved
  - The moment of the piece to be moved (moment - force of rotation - look it up, you're engineers)
- Net force (and advantage) of a pulley system

Peer Review

In any large collaborative project or scientific endeavour, the work of each person is subjected to scrutiny by their peers to ensure that:

- Their design isn't overlooking something (like a battery mount)
- Their math is correct
- Any parts aren't interfering with other parts
- There's enough room for all the other parts of the robot (like all the electrical parts)
- Any parts aren't interfering with any game elements (does the frisbee even FIT there?)
- Any assumptions are realistic

It is difficult to allow one's work to be subject to scrutiny by others that haven't put some if any effort into the work, but any errors and corrections made at this stage will save huge amounts of time vs. fixing them on a fully welded robot.

Designing the Robot - Programming

Before the CAD design is done, the programming team can assemble all the parts that are expected to make it onto the robot and start programming all the different elements.

- Motor controls
- Sensors

- Encoders

Even if the parts are assembled on a piece of cardboard, it is enough to program.

- A motor may move the wrong way in the prototype, but it moves. It's not hard to change the polarity
- The proper speed for a motor to spin may not be known, but if it spins according to a supplied parameter, it's enough that it spins. The correct speed can be determined during testing.

Designing the Robot - Electrical

Once the CAD design is done, immediately start on the electrical design. With all the elements of the robot like motors, compressors, accumulators, and travel paths of robot arms and rotors, a mock up of the electrical boards can be done.

This mock up will allow the team to layout the parts of the electronics on a mock up board (cardboard) so it can be laid out and wired.

This can be a single board or several separate board, but the dimensions of the board should be obtained from the CAD drawing.

The cardboard mock up will be transferred to its final substrate after the chassis is done, but the mockup allows for planning and layout early in the build process.

Building the Robot

Prototyping

Prototyping various elements as a proof of concept is always a good idea, but the design of the prototype should be as close to the final product as possible.

The prototype may be made of wood, but the dimensions should be as close to the final product as possible.

A prototype that is built as the final product and can be inserted as a module into the final robot is preferred.

Building

Steps not integral to the game should be avoided if they delay the build process. For instance, if the weight limit of the robot is 120 pounds and the CAD estimate is 100 pounds, spending 3 days to cut lightening holes to save 1 pound is wasting 3 days of work.

Managing the Build

The previous few seasons have seen the implementation of various management tools including Gantt Charts and the task board.

Gantt Charts

This is a charting method that shows each task, the duration of the task, when the task can begin, and when it is scheduled to end.

The uses for a Gantt chart are:

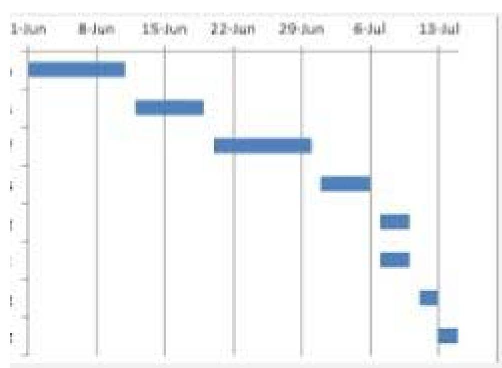
- Allows everyone to see what comes next
- Shows when things are supposed to be done
- Shows dependencies of one task to prior tasks

Of these tasks, the last one is the most important in gaining an advantage.

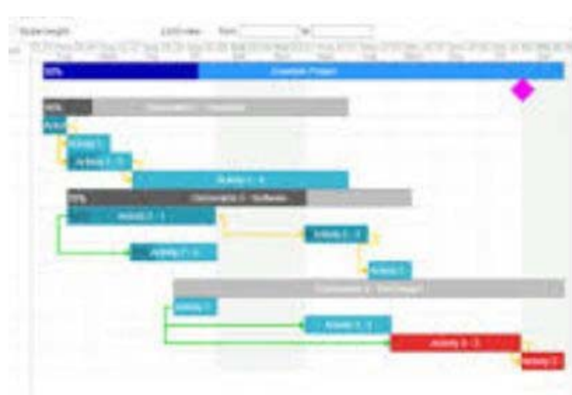
If the chart looks like a set of stairs, multitasking is not being done effectively.

Look for tasks that can be done in advance and create a fork for that task in the chart. An example is the mock up of the electrical board onto cardboard.

This will make use of your resources more effectively and shorten your build time.



A Gantt chart should show several forks indicating a good division of labor.



Task Board

The task board has been in place for three years as of this writing and was introduced a way to manage each task of the season for each team division.

Programmers will recognize this methodology as an adaptation of the Agile development methodology.

Team divisions can be, but are not limited to:
- Design
- Electrical
- Programming
- Fund-Raising
- Promotions
- Video
- Chairman's Presentation

Each team division is assigned a different color of Post-It note and writes each sub-task they need to complete on separate notes. Each note should also list the start and due date of the task.

A task board is created with three sections that reflect the time line of the tasks:
- Not Yet Started
- In Progress
- Completed

Each divisions' tasks are queued up as a group in the Not Yet Started section in the order they need to be completed.

As each task is started, it gets moved from the Not Yet Started queue to the In Progress section and then to the Completed section.

Tasks that were unanticipated can have a Post-It note created for it at any time.

A quick glance at the board will reveal how many tasks have been completed and how many have yet to be started.

A look at the In Progress notes will show which team division is ahead of schedule and which divisions are behind schedule.

Most importantly, this analysis of what's ahead of schedule and what's behind schedule allows:
- Leadership to allocate resources to divisions that need help

- Everybody can see which division needs help and can volunteer what they can to benefit the entire effort.

Lessons Learned 2016 - Stronghold

Institutional Knowledge

Prior years' experiences are not being passed on as seniors graduate. This lack of continuity leads to the team not following what worked and abandoning what didn't work.

This document is a result of this lesson.

Know When to Give Up

Individually and as a team, learn to recognize when a course of action is taking up too many resources or is not going to be successful and commit to ending that endeavor.

The Best Qualified / Most Knowledgeable Should be in Charge

Experts, regardless of rank/class, should be respected and listened to. Informed dissent and discussion is encouraged, but rank is secondary to expertise.

https://www.youtube.com/watch?v=ixDn8Ak4kz8

The pilot is a Bell helicopter executive and felt he should make the first flight instead of the inventor or a test pilot. He was never seen on the field again.

Multi-Task Where Possible

Make a Gantt Chart and optimize the available build time.

Prototype

Create prototype as close to the final design as possible - as if it will actually be going on the robot. Otherwise, prototyping isn't effective.

Rationales

Have reasons to do something, and be able to communicate it to reduce confusion and dissent. This helps everybody work as a team.

Peer Review

Everyone should be critical thinkers and peer review designs.

There should be a formal peer review of the robot design during the design phase so others can examine the design for problems - part interference, insufficient clearances, rule violations, etc.

Only actual problems should be addressed.

Try New Things

The robots for the past few years have all followed the same basic pattern. In order to build new capacity, we should explore new technologies where appropriate: tank treads, frameless construction, direct drive, etc.

Good Leaders Must Also be Good Followers

Good leaders must be able to receive and act appropriately to constructive criticism. A leader must make the final decision, but ignoring evidence and good counsel isn't in the best interest of the team.

Time is Our Enemy

Take time to evaluate where you are and what you're doing and how you can effectively work within the time constraints.

Time is a finite resource that can't be recovered.

Lessons Learned 2017 - Steamworks

Design

The robot constraints included two different configurations, a tall one and a shorter one. Our original design attempted to put three elements, a gear funnel, a climbing winch, and a ball waterfall gatherer, in the taller configuration. A back of the envelope calculation of the size of the elements involved would have revealed that their combined footprint exceeded our selected configuration.

This led to a team meeting to review the game analysis and the economy of the points. As a result, the chosen design was modified to eliminate the ball gatherer as the point value of the balls were deemed minor compared to the gears and climbing.

Lesson(s)

Perform a reality check early on to avoid expending time and resources on an impossible design.

During game play, the ball points became a deciding factor at the top level of play. Discounting low value points can relegate your robot to a lower level of play.

Design Review

We did a formal design review after week 2. The result was that the basic design had a flaw in how the robot approached the climbing rope as the top support interfered with the rope.

Possible solutions:

- Move the top support to the bottom of the winch mechanism
- Create a second layer of robot chassis to house the electronics and rotate the winch 90 degrees.

We decided to move the top support as it was faster than redesigning the entire chassis.

During the build it was discovered that the bolt heads on the motor mounts interfered with each other as the mounting holes were aligned with each other. The motor mounts had to be redesigned and re-manufactured.

Lesson(s)

Perform a design review earlier to avoid being trapped into a sub-optimal design.

The design review should be more detailed in order to catch such design flaws.

Scheduling

During the first 4 weeks of the build, the task board appeared to show us on schedule. However, a mental inventory of the remaining tasks indicated that we were not on target.

The flaw in the task board was that it wasn't fully populated with all the remaining tasks, only those that were due within the next week. Once the task board was fully populated, the sense of urgency returned.

Related to this, many students were distracted by their smartphones - either with games or streaming videos.

Lesson(s)

Populate the task board completely as soon as possible to get a better sense of remaining tasks.

Cell phones use for entertainment purposes was strictly banned.

Build

During the first regional competition, a CIM motor detached from the robot by vibrating loose.

Lesson(s)
Check all fasteners before declaring the robot completed.

Use thread locking compound on all critical fasteners that are not designed or desired to come loose.

Competition

At our first regional competition, we were hampered by the lack of:

- 5/32" Allen wrench
- Long Ethernet cable
- USB A to B cable

Lesson(s)

If you used a hand tool to program or assemble the robot, make sure it's on the packing list. Yes, they can be borrowed during the competition, but it is inconvenient.

Appendix A - Flaws in How We Think

The Abilene Paradox

When a group of people collectively decide to pursue a bad course of action that none of them wants to pursue as an individual. This occurs because of a desire to conform to their perception of the group's desired course of action.

To help prevent the Abilene Paradox, speak your mind instead of speaking what you think is on another's mind. This is an example where the problem of managing agreement is harder than managing disagreement.

Confirmation Bias

Confirmation bias is the tendency to search for, interpret, favor, and recall information in a way that confirms one's beliefs or hypotheses, while giving disproportionately less consideration to alternative possibilities.

It prevents us from rejecting personal beliefs when presented with overwhelming evidence that we are wrong.

Explanatory Attribution

The search for why things happened. People have a high need for an answer, any answer. Answers are person dependent as some are prone to attribute things to internal sources while others are prone to identify external sources as the explanation for events.

We are very bad at accepting that nothing may be to blame. We need an explanation, but one may not exist. Just deal with it.

Groupthink

When a group of people collectively decide to pursue a bad course of action. This occurs because of a desire to conform to the group's desired course of action.

To help prevent groupthink, have somebody propose a counter course of action, becoming the devil's advocate. This will force the group to consider alternatives. This is another example where the problem of managing agreement is harder than managing disagreement.

Negativity Bias

Overweighting of negative outcomes. Spending more resource to determine why something bad happened vs. spending resources to determine why nothing bad happened.

This was useful when life and death situations were a daily occurrence - gathering food while avoiding hungry tigers. It's not so useful now when people are terrified of sharks but don't fear vending machines when both kill about the same number of people annually.

Be aware that risk perception is always skewed to give more weight to negative events. This bias can also be addressed by requiring evidence based proof as to the possible negative outcome.

Omission Bias

Harmful actions are worse than the harmful inactions or omissions. The easiest thing to do is to do nothing. Again, this is a cognitive shortcut.

Optimism Bias

Optimism bias is a cognitive bias that causes a person to believe that they are less at risk of experiencing a negative event compared to others. Las Vegas was built on the backs of these winners.

Just as with negativity bias, evidence based proof as to the probable outcome of an event should be used to address this type of bias.

Skewed Risk Perception

Vague future problems are less problematic than potential immediate problems so it becomes easier to accept the future risk as it avoids having to deal with the immediate risk.

Dunning – Kruger Effect

A type of cognitive bias where low ability people believe they perform at a much higher level than they actually are. This stems from low ability people not knowing what they don’t know. When educated, these people typically recognize their shortfalls. Mitigation of this effect is to become more self-aware on what you do actually know and, more importantly, making the effort to educate oneself.

A corollary is where high performing people cannot recognize their ability and cannot understand that understand why what they do is difficult for other people. This corollary is less important for this discussion.