LAB ONE - APPLIED RECURSIVE INTEGRATION

Recursive Integration

Recursive Integration is my engineering methodology. A methodology is an approach to solving a problem. An engineer's job is solving problems with data. My boss once asked me to teach engineers how to debug. I learned by working in lab, on the job. I never thought about my work as methodology. I had a mentor named Steve Fairchild, who was the best engineer I've ever worked with. My methodology is based on what I learned from him.

James Harden said, "*I am a system*," in the context of his job, playing basketball. A system is an implementation of a methodology. His system solves bucket problems. Put your buckets problem in Harden's lab; he'll use his methodology to optimize his system to get you more buckets.

James Harden is the greatest player of his generation because every move he makes is a complex calculation, based on data gathered from all the time he's been in the gym playing and watching basketball. His brain has accumulated petabytes of data about basketball. James has been in the gym hooping for more hours than most people spend sleeping – since he was a kid. His brain is a data accumulation machine of basketball knowledge.

I am good at debugging because I've debugged a lot of problems. James Harden is a basketball system because he's debugged a lot of basketball problems. I've spent most of my career in the lab. James has spent most of life in the gym.

The power of recursive integration is the value it creates for the future. I'm a good debugger because when I started working in a lab in the 1990s, it was the dawn of the internet age. I used search engines to find how others solved similar problems. When Steve Fairchild was that age, he had to either find a book, ask another engineer, or he just kept coding until he figured it out.

Michael Jordan and Hakeem Olajuwon did not have access to taped basketball games as children. If they saw basketball live on TV, it was rare, like the playoffs. They might have watched a local team play, but it wasn't every day. Even as top college players, they couldn't watch a breakdown while they were eating breakfast in their dorm. They had to wake up early, wait for the tech guy to rewind tapes, watch a few sets before they had to go to class. In the NBA in the 80s and 90s, they had access to tapes of other NBA games – but it was limited by how many VHS tapes the video guy could cram into his luggage. Compare that with today.

James Harden is highly leveraged by past basketball genius and technology.

Would James Harden have developed his system without having seen Kobe and Jordan play? James is tall, strong, athletic, extremely intelligent and left-handed; These traits give him an edge in most sports. He would have been a great athlete; but not the unbeatable system we know today. The foundation of James Harden's system is past basketball genius. Technology allows this genius, as well as Harden's own, to be appreciated by any human with a screen.

Today, you can see every basketball game, at any level, from any country, on your phone, wherever you are and whenever you want. Does this mean that there will be better players than James Harden in the future? Certainly. All the best guards in the NBA already copy his game; none of

them have the southpaw hardware James has been training for thirty years. The James Harden system works so well because it was engineered for Harden's custom hardware.

Can a kid today become the next James Harden? If you're willing to put in the work and spend most of your life in the gym, it is possible. James must train his body relentlessly daily to make it perform at the level his system demands. In athletics, there is no getting around putting in the work.

In engineering, we can train while we sleep. We can let the machines do all the hard work, missing all the shots and running drills in the gym, while we do the fun stuff, like working out new plays and scrimmaging with teammates. The better we get at this, the faster we'll be able to solve the most challenging problems. This is the power of applied recursion. Even huge problems, like transportation in Houston, can be solved through recursive integration.

Leverage, Recursion, Integration and Black Boxes

Leverage is the primary force of engineering. Silicon chips are the highest leverage machines man has ever created. Other tools help us exert physical leverage. Digital tools allow us to leverage our minds. Recursion is applying leverage upon itself, accelerating change. Integration is simply breaking things down and putting the pieces together. Black Boxes are machines where you know the expected inputs and outputs, but not necessarily how things work inside the box.

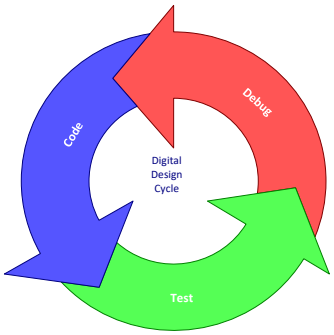
Systems

In James Harden's system, his brain is the software, and his body is the hardware. His system includes many other components; coaches, teammates, trainers, doctors, nutritionists, friends and family, are part of his system.

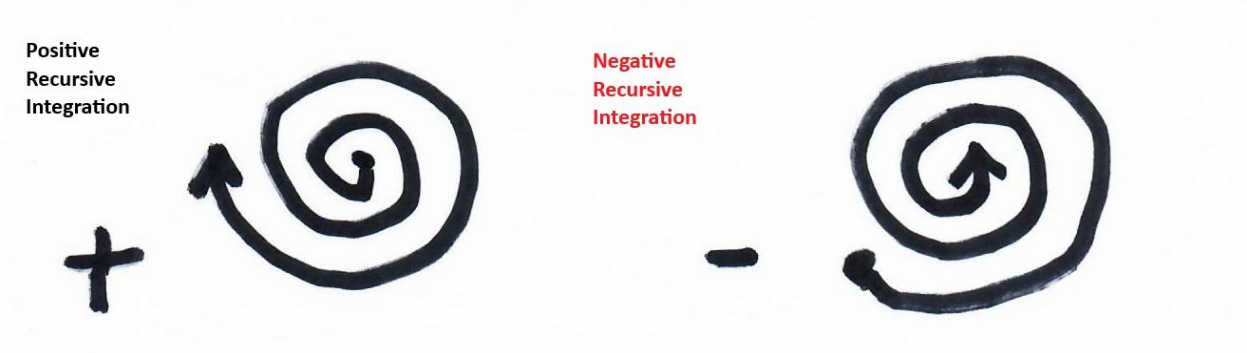
A transportation system includes anything that helps move people around. Almost everybody has the most basic system, legs and feet. In the modern world, all engineering is system engineering.

Design Cycle – Design, Debug, Test

Design, Test, Debug. That is the design cycle. In basketball, you set up a shot (design), shoot it (test), and if you miss you adjust and try again (debug). We learn more about solving a problem by the shots we miss (failed test) than the shots we make. Failure is key to success in both sports and engineering. Each time we go through the design cycle loop, we call it an iteration. The faster we can iterate, the quicker we can resolve the cycle into a solution, no matter the size of the problem.



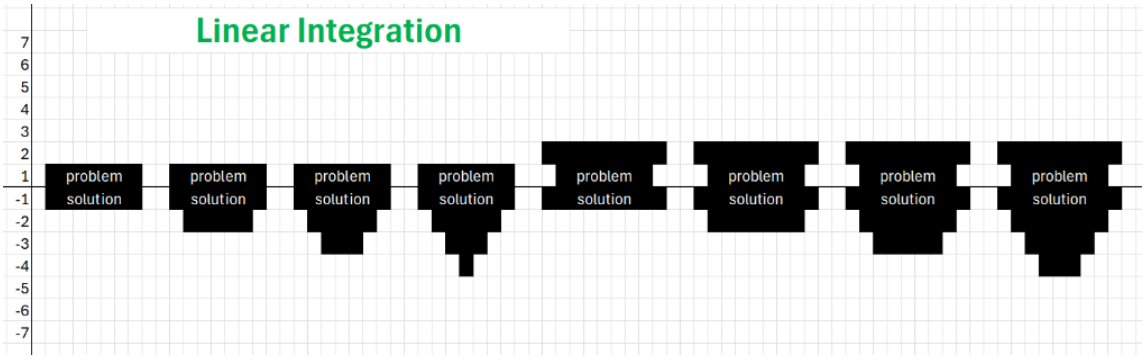
Debug is a word engineers use to mean, "figure out what is broke." In the 40s and 50s, computers were huge rooms with thousands of miles of wire. When a program failed, technicians would comb through all the components looking for insects that had shorted a circuit. From the beginning, smart engineers blamed the bugs even when it was their bad code that was at fault.



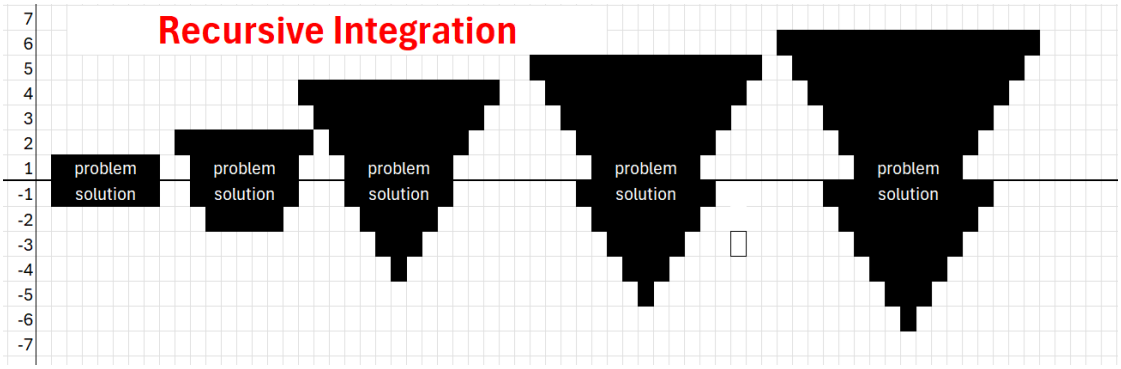
We use positive and negative in the mathematical sense, not in the good or bad sense. Negative recursive integration is making a solution better. In electronics, this means making something smaller, faster and more efficient. Positive recursive integration is expanding the scope of the solution, solving more or different problems with your system.

Applied Recursive Integration

System architecture usually focuses on positive recursive integration, and systems design usually focuses on negative recursive integration. In that case, we have a series of innovation steps that looks like this.



Linear Integration focuses on either negative or positive integration with each integration.



Recursive Integration is working on both the problem and solution at the same time, leveraging what we learn each iteration. We can solve bigger and more complex problems by constantly accelerating each iteration in both directions.

Lab One

Build a dash-cam system for ride share drivers.

Lab One: https://github.com/fairchildlabs/Lab1

Recursive Learning

We cover the lab material as fast as our skills allow, the team working together to move to the next level. We expect the first class to teach the next class, and to get their teams through the lesson in shorter amounts of time. The time that is reduced is the rate of recursion. This is recursive integration applied to an education system.

Key Skills Covered

- Computer Hardware
- SoC (System on a Chip)
- Linux
- Digital Cameras
- Data Storage
- SSH
- Github/Git
- Python
- Field Testing
- Problem Characterization
- Data Analysis
- Scripting/Shell
- C/Make

Lab One Validation Criteria

- Beta-test working camera unit
- Questions: Why are we building a dash-cam system? What is the value of video data? How can we use this system to solve a bigger problem?

Lab One Exam

Make an engineering Mount Rushmore. Create a list of the 4 greatest engineering achievements in the 100 years before you were born.

Example: I was born in 1970.

1. 1969 – NASA Apollo – Man on the Moon
2. 1945 – Manhattan Project
3. 1887 - Electrification (Nikola Tesla)
4. 1914 – Panama Canal

Contact Information:

Brandon Awbrey
scuzzydude@hotmail.com
(713) 305-5620