CajunBot: A Case Study of an Embodied System

Istvan S. N. Berkeley,
Philosophy and Cognitive Science,
The University of Louisiana at Lafayette,
(istvan@louisiana.edu)

Two Views, I
(From Cowart, 2006)

- **Classicicist/Cognitivist View**

- 1. Computer metaphor of mind; rule-based, logic driven.

- **Embodied Cognition View**

- 1. Coupling metaphor of mind; form of embodiment + environment + action, constrain cognitive processes.

Two Views, II

- **Classicist/Cognitivist View**
- **2. Isolationist analysis - cognition can be understood by focusing primarily on an organism's internal processes.**

- **Embodied Cognition View**
- 2. Relational analysis- interplay among mind, body, and environment must be studied to understand cognition.

Two Views, III

Classicist/Cognitivist View	Embodied Cognition View
3. Primacy of computation.	3. Primacy of goal-directed action, unfolding in real time.
4. Cognition as passive retrieval.	4. Cognition as active construction based upon an organism's embodied, goal-directed actions
5. Symbolic, encoded representations	5. Sensorimotor representations

Background: The DARPA Grand Challenge 2005

- In June 2004, DARPA announced their Grand Challenge for 2005.
- The goal of the Challenge was for teams to construct autonomous vehicles that "could navigate a challenging course over varying terrain."
- The detailed rules, issued in October 2004, specified that the course would be no more than 175 miles (282Km) long and would consist of roads, trails and off-road desert areas, which contained a variety of obstacles.

Team CajunBot

- The University of Louisiana at Lafayette entered the Grand Challenge as 'Team CajunBot'.
- This team had also participated in the 2004 Grand Challenge.

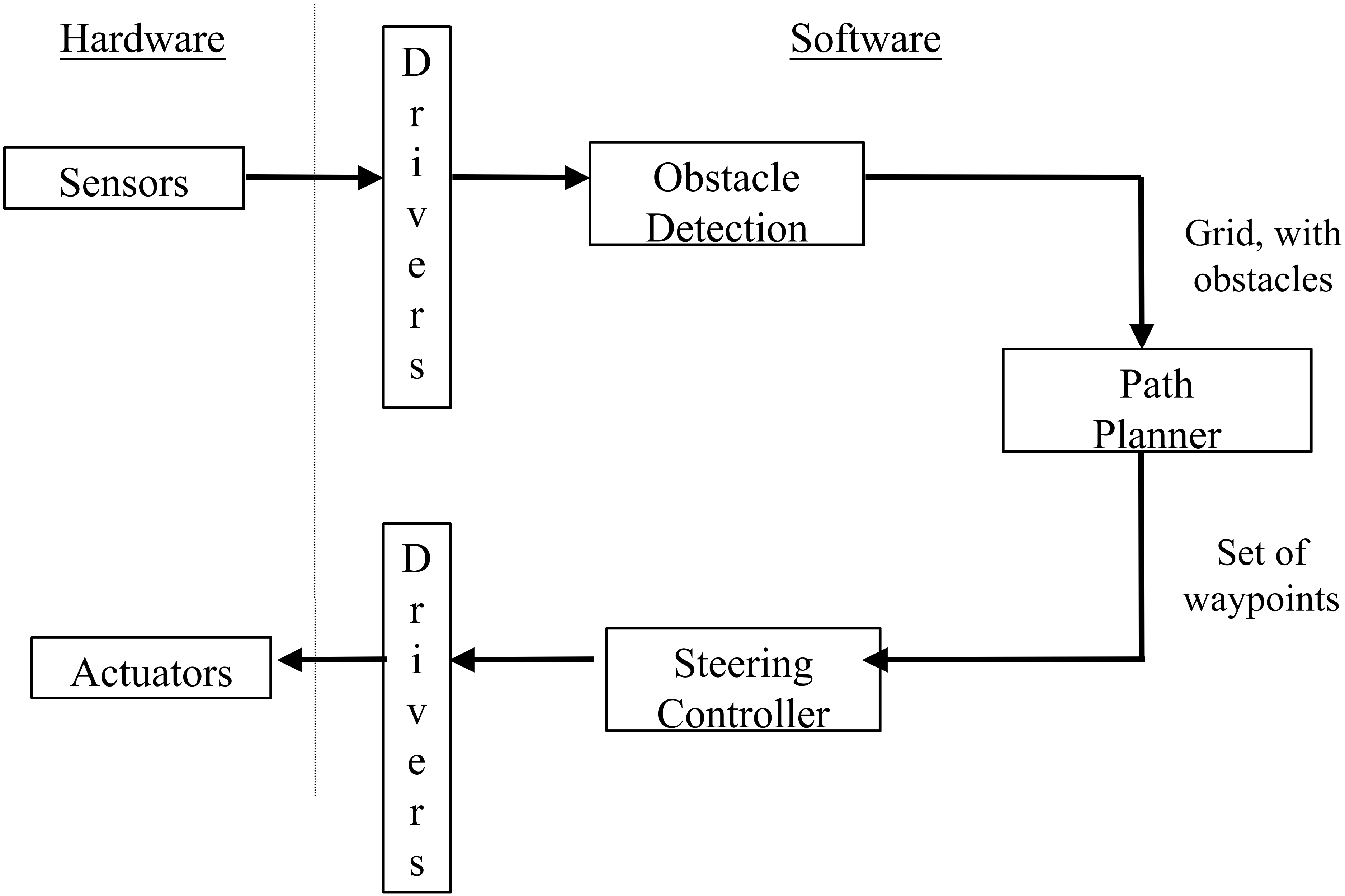
Mechanical Configuration

- Based upon a Recreative Industries MAX IV ATV.
- Six wheel drive, with skid steering.
- Powered by 28hp Kohler engine.
- Electrical power supplied by two Honda generators.

Sensor Systems

- Oxford Technical RT3000 Inertial Navigation System, enhanced by Starfire differential GPS, from a C&C Technology C-Nav receiver.
- Up to Five LIDAR laser obstacle sensors.

Architectural Overview



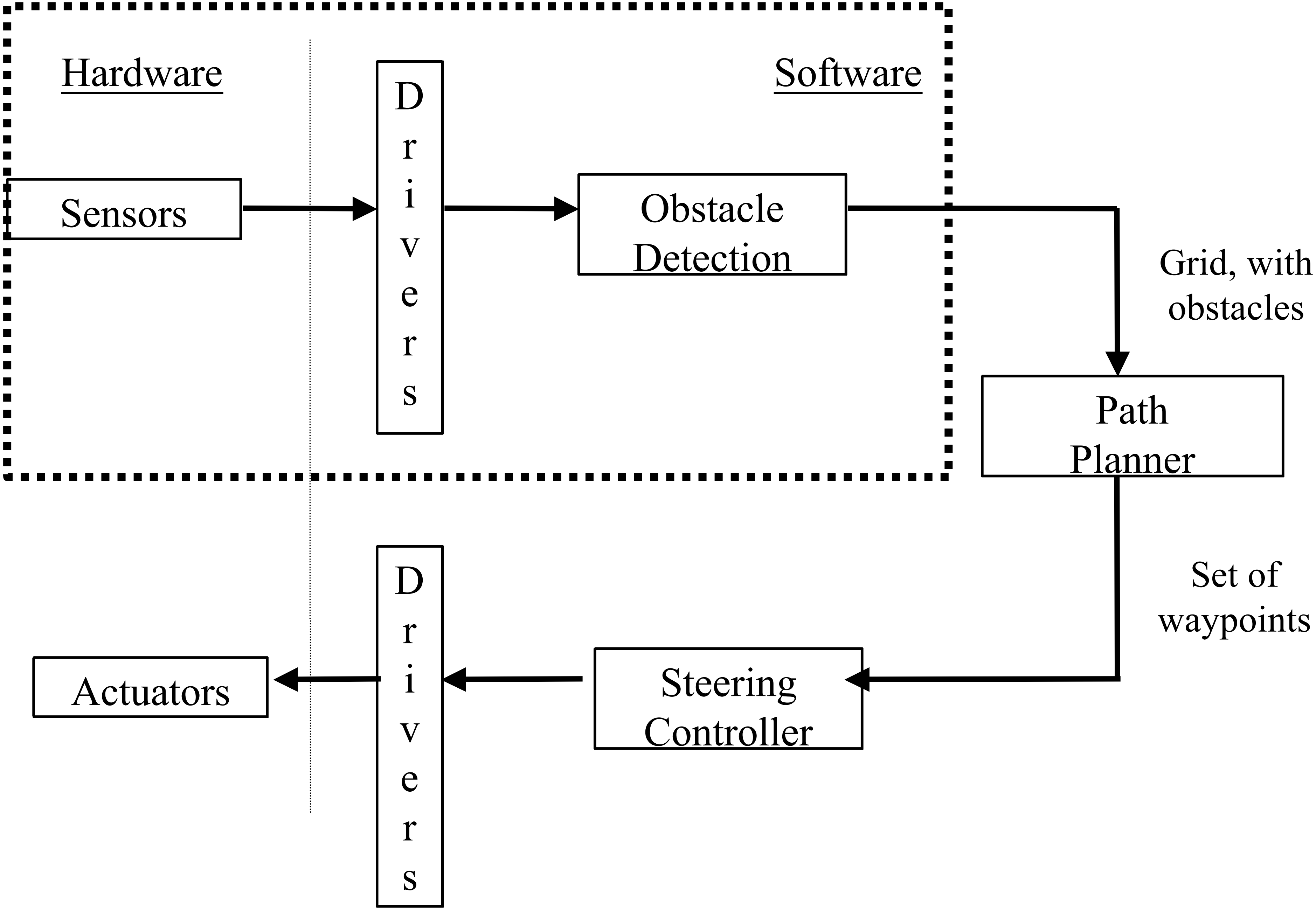
Special Features of The CajunBot Situation.

- One of the things that makes CajunBot interesting in the current context is that the device has to operate in a real world environment, in real time.
- This presents additional constraints upon the kinds of solutions to problems that can be used with CajunBot. For instance;
  - Sensor outputs cannot be 'idealized',
  - The actual manoeuvrability of CajunBot must be taken into account (rather than an idealized version)
  - Processes must operate fast enough to prevent mishaps.

Real World Solutions to Real World Problems

- In what follows, various subsystems of CajunBot will be examined to illustrate how the real world nature of the task at hand influenced the ways that solutions to particular problems were implemented within CajunBot.
- The results show an interesting interaction between 'embodied' type solutions and more traditional solutions.
- The results are not always what one might expect!

Subsystem 1: The Sensors



Subsystem 1: The Sensors

- In the words of one of the CajunBot team members (Tony Maida),
  - *"...the vehicle moves like a brick on wheels."*
- *Given that CajunBot could travel at speeds of up to 28 mph (45 Kph), over rough terrain, this meant that there was going to be an issue of integrating the outputs from the LIDAR laser sensors.*
- *This was because where obstacles were detected by these devices would change, depending upon the vertical angle of the vehicle.*

Sensors: Possible Solutions

- There are roughly three ways that the up and down motion of the LIDAR laser sensors can be handled:
  - (1) Use a vehicle with a very good suspension, so as to dampen motions,
    - This was not an option for CajunBot, due to the hardware available
  - (2) Mount sensors on a platform stabilized by a Gimbel, or other stabilizers,
    - Although this was a strategy used by many teams, cost prohibited it being used on CajunBot.
  - (3) Mount all sensors on a single rigid platform.
    - This was the solution selected. It turned out to provide a surprising advantage.

Sensor Solutions: Pose

- The difficulty caused by the up and down motion of the vehicle was handled by using 'pose' data from the Inertial Navigation System.
- By including this data, it became possible to determine the complete location the LIDAR system was actually reading.
- In fact, this approach actually conveyed a distinct advantage, as it had the effect of increasing the effective range of the LIDAR system
- This also avoided the 'single point of failure' problem that effected Gimbel based solutions.

Two Further Sensor Problems

- Occasionally, the INS/GPS system produced 'spikes', caused by satellite communication problems. This data needed to be filtered out and discarded.
- The INS/GPS system also suffered from a phenomenon know as 'Z-drift'.
- The system often exhibited a drift in Z values, reported over time.
- Even when the vehicle was stationary, a drift in Z values of 10cm – 25cm, could arise. This could cause a flat surface to appear uneven.

Sensor Solutions: Spikes

- Inputs to this sub-system came from both the INS/GPS system and from the LIDAR scans.
- The INS/GPS data was filtered to remove any spikes. This data was then used to compute the global coordinates of the LIDAR scans.
- This avoided corrupted data entering the data stream.
- When the GPS system suffered a mishap, there was still data available from the Inertial Navigation System.

Sensor Solutions: Z-Drift

- The Z-drift issue was handled by taking into account the time that a particular global point was observed.
- Global points were thus represented as a 4-D value. This had the format (X, Y, Z, and time-of-measurement).
- By ensuring that points have a temporal distance between them of less than 3 seconds, when used in further computations, it was possible to overcome the Z-drift problem.

Another Sensor Problem: Data Integration

- A further problem arose from the fact that the different sensor systems generated data at different rates.
- The Inertial Navigation System (INS) generated data at 100Hz, producing data at 10ms intervals.
- The LIDAR laser sensors generated data at 75Hz, producing data at 13ms intervals.
- Thus, the most recent INS reading, when a LIDAR scan is read, may be up to 9ms old.

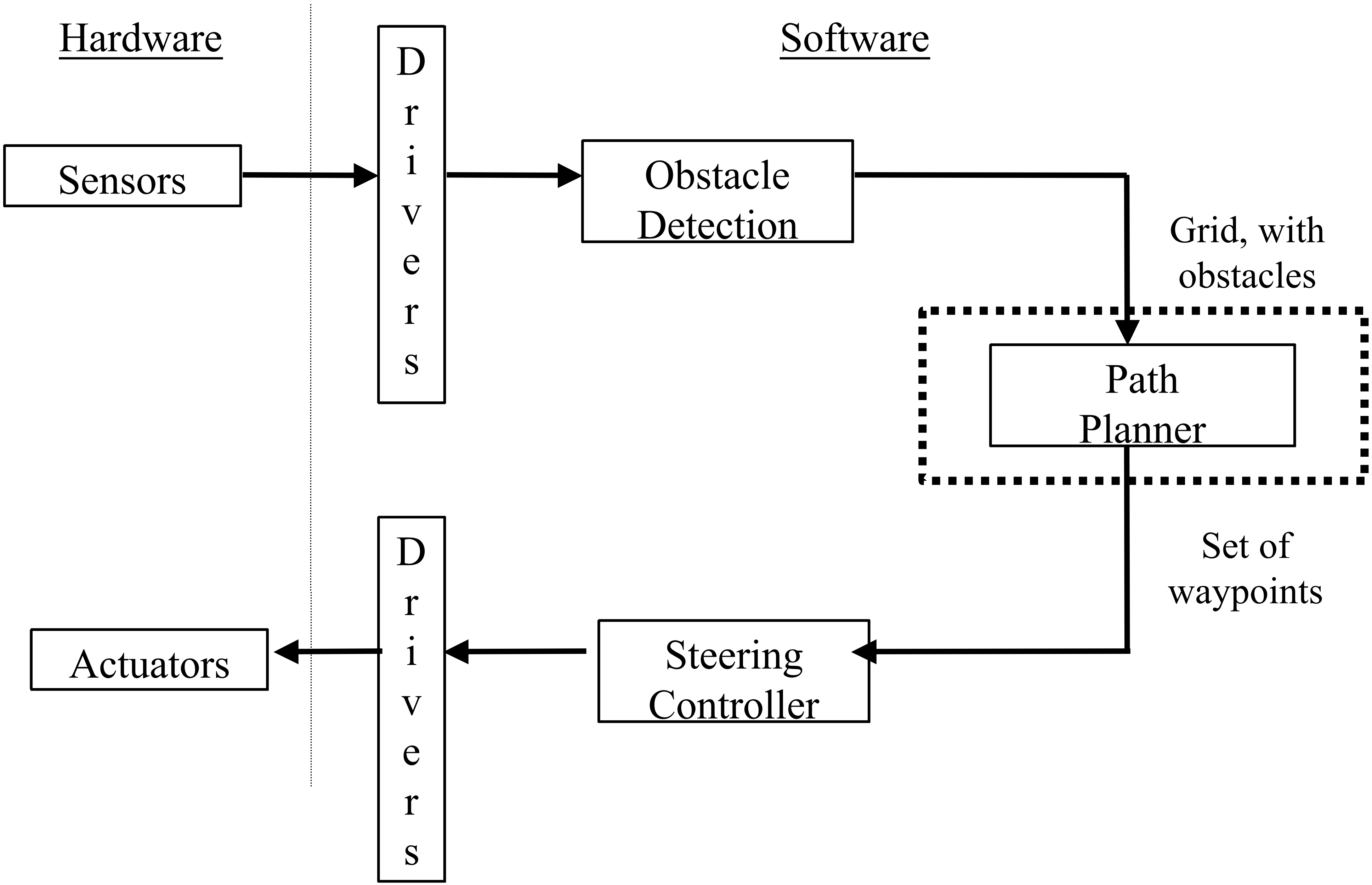
Sensor Solutions: Data Integration

- The data integration issue was handled by the 'blackboard' system, through which different CajunBot modules communicated with one another.
- Instead of just fusing the most recent data from the LIDAR and INS/GPS sensor systems, global points were computed by interpolating the state immediately before and immediately after a LIDAR scan was read.
- In some senses, this blackboard system was somewhat analogous to working memory in a cognitive system.

Sensor Subsystem: Comments

- It is clear that the embedded real world nature of CajunBot presented some interesting difficulties with the sensor subsystems.
- The solutions to these problems were often quite conventional in nature.
- However, the issue concerning the up and down motion of the sensors, was less than entirely conventional and indeed, by taking additional information about the environment into account, CajunBot was actually able to gain an advantage.

Subsystem 2: Path Planning



Subsystem 2: Path Planning

- Path planning, as a type of search, is a venerable topic in traditional Artificial Intelligence research.
- Planning systems can be characterized as being 'deliberative', or 'reactive'.
- *"**Deliberative** systems that embody powerful techniques for reasoning about actions and their consequences often fail to guarantee a timely response in time-critical situations. Reactive systems that respond well in time-critical situations typically do not provide a reasonable response in situations unforeseen by the designer."* Blythe and Reilly, (1993).

Path Planning: Deliberative vs. Reactive

- In order for CajunBot to perform successfully, it needed to employ elements of both strategies.
- A deliberative strategy would be helpful in ensuring that CajunBot reliably reached goals along the specified route.
- A reactive strategy would be helpful in ensuring that CajunBot avoided obstacles that appeared along the way.

Path Planning: G-Nav and L-Nav

- The long range planning system used with CajunBot, G-Nav, was based upon a sequence of static GPS waypoints, that were held in a route description file.
- The local planning system, L-Nav, provided sub-goals that enabled the navigation between the static waypoints.
- The L-Nav system was able to take into account the presence of local obstacles and make the appropriate adjustments to the relevant subgoals.

Path Planning: L-Nav Metaphor

- The path planning method used by the L-Nav system rests upon a metaphor of charged particles.
- The idea is described in Maida *et al.* *(forthcoming) as follows,*
- *"...the robot is (say) positively charged and a desired goal is negatively charged. Obstacles are given the same charge as the vehicle. The simulated force vectors can control the steering of the actual robot in the actual world so that the robot approaches the goal while avoiding obstacles."*

Path Planning: Is L-Nav Deliberative, or Reactive?

- The fact that the L-Nav system enabled CajunBot to escape from dead end canyons has been suggested as evidence that the system has deliberative properties.
- On the other hand, the real-time obstacle avoidance capacity, appears to exhibit reactive properties.
- In fact, there was little agreement (though much debate) over this question amongst members of the CajunBot team!
- L-Nav is probably best thought of as being a blended system.

Path Planning: L-Nav Expansion Regions

- Obstacles were represented in the L-Nav system as having an 'expansion region' around them.
- This expansion region effectively made obstacles larger than they really were.
- The purpose of the expansion region was to provide a margin of safety, so as to allow for "...imperfect steering or other unanticipated physical event[s]." (Maida, *et al., f*orthcoming).
- Given the potentially catastrophic consequences of a collision between CajunBot and a obstacle, this was a prudent and necessary affordance.

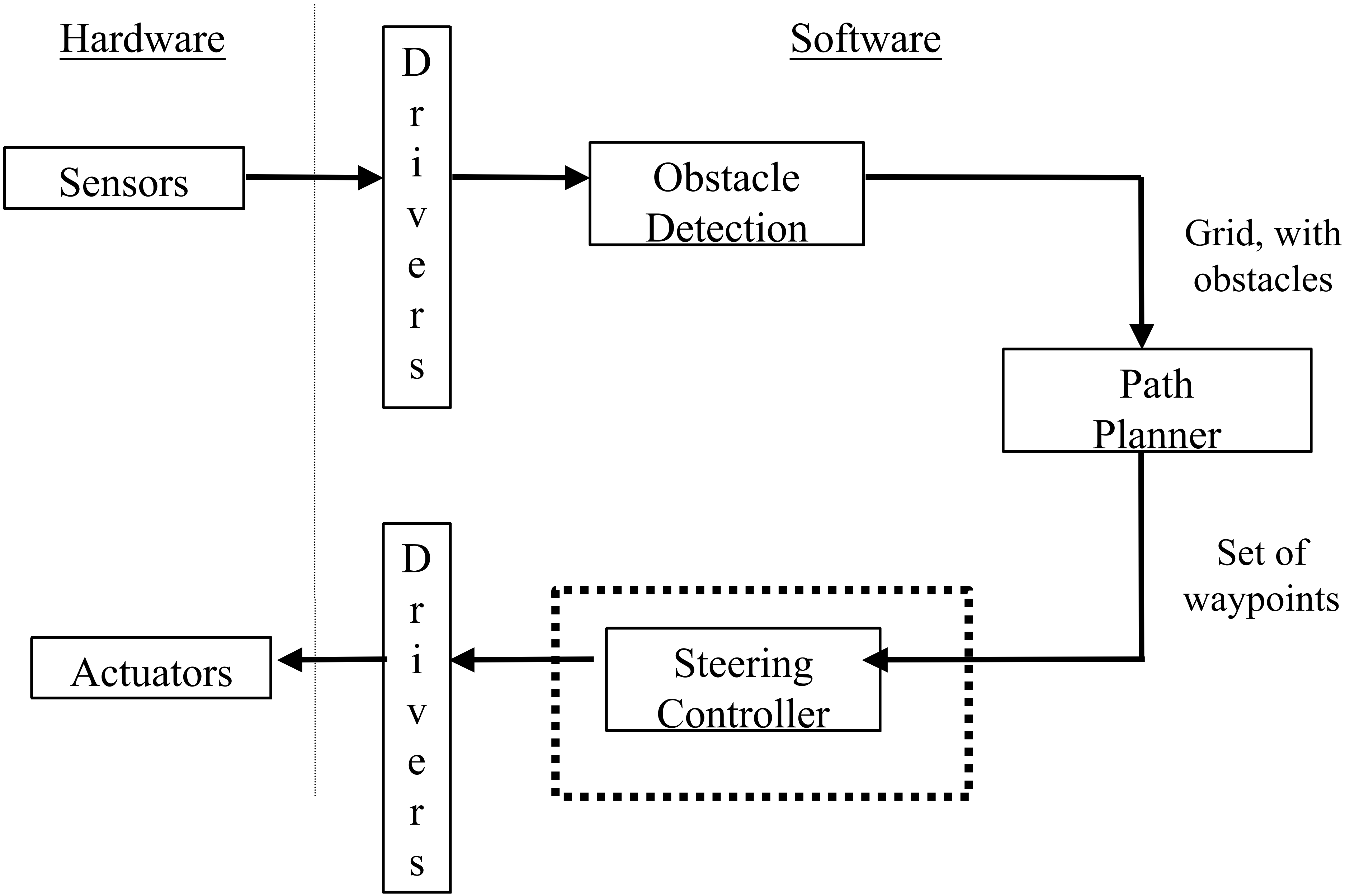
Path Planning: Waypoint Filtering

- A final issue that the path planner had to take into account was the limits of manoeuvrability of CajunBot.
- Particularly at speed, CajunBot was not able to make rapid turns or sharp changes of direction.
- This was handled by first identifying places in a proposed path that would involve a change of direction.
- Potential waypoints in a proposed path were then filtered to ensure that only paths that were consistent with the capabilities of CajunBot were selected.

Path Planning: Comments

- The G-Nav subsystem was manifestly deliberative and traditional in the way that it functioned.
- The L-Nav subsystem, by contrast appears to have had both deliberative and reactive features.
- Furthermore, other aspects of the L-Nav system had to make allowances for the fact that CajunBot had to operate in a real world environment and was subject to real world constraints.

Subsystem 3: Steering Control



Subsystem 3: Steering Control

- Steering CajunBot presented some interesting challenges.
- This was in part, because the steering system presented a classic control system problem.
- As CajunBot turned towards a new heading, it was necessary to stop turning before the exact new heading was reached, in order to prevent over steering.
- This problem arose in large part due to the real time nature of the problem.
- Both software systems and hardware systems suffered from temporal lag.

Steering Control: Solution Part 1

- The steering controller had the *current_heading* of CajunBot and the *desired_heading* as primary inputs.
- There were other inputs, that encode current speed and other information, but they will be overlooked here.
- The first step in the process was to compute *Error, by subtracting present_heading from desired_heading.*
- *Then the Error_rate was computed, by subtracting the Error from the previous_error, with respect to the time since the last control loop execution.*

Steering Control: Solution Part 2

- The *Proportional term (P-term) was then computed by multiplying the Error by a constant K_p.*
- *The Differential term (D-term) was then computed by multiplying the Error_rate by a constant K_d.*
- *The constants K_p and K_d were set by a trial and error, in field trials.*
- *The Proportional term offered a measure of how much error needed to be corrected.*
- *The Differential term gave a metric of the rate of increase, or decrease in Error.*
- *The P-term and D-term were usually of opposite signs, such that they can cancel one another.*

Steering Control: Solution Part 3

- Finally, the value of the steering command was computed as follows:

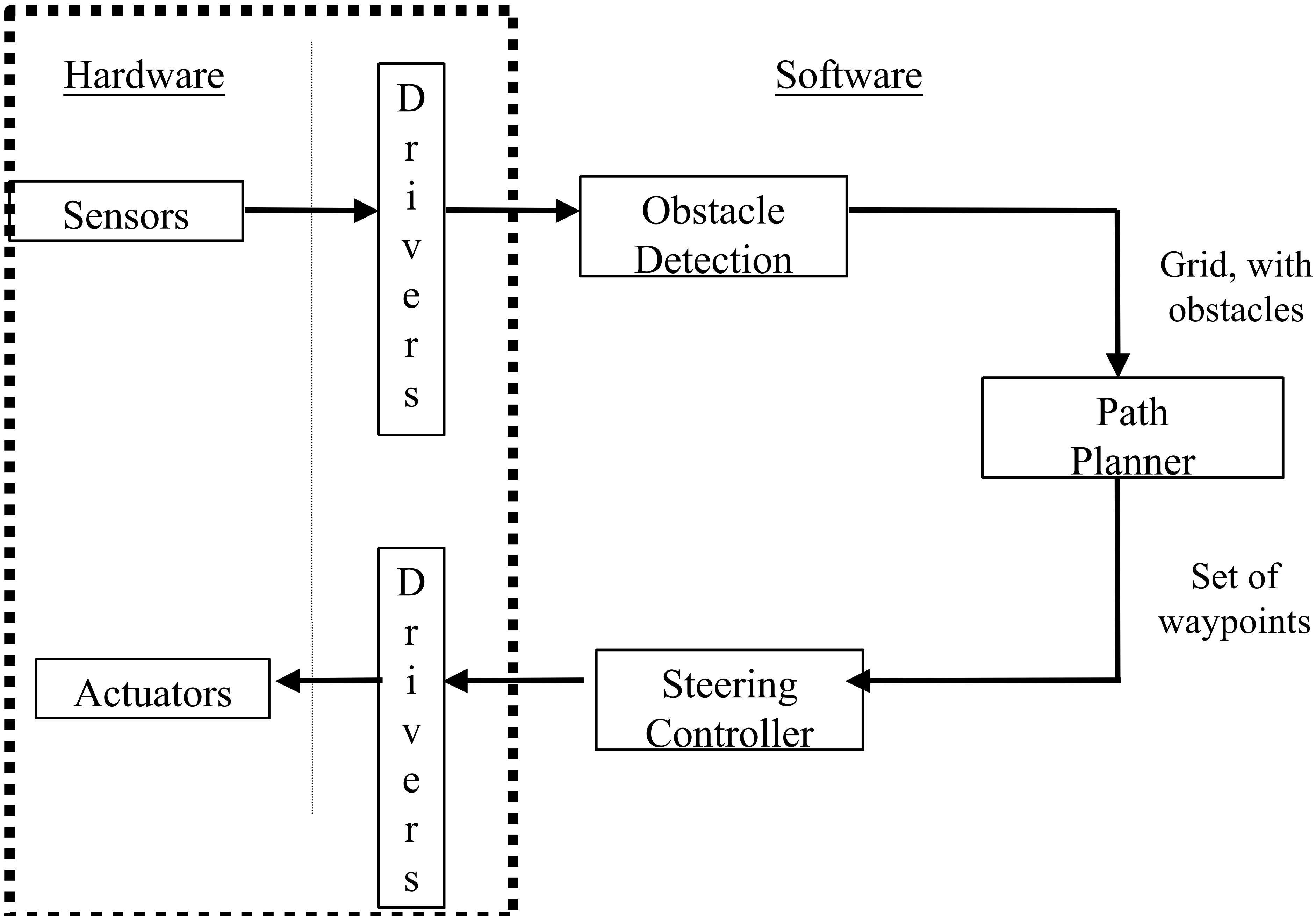
$$Steering = K_p(Error) + K_d(Error\_rate)$$

- Using this method, it was found that the steering problem could be solved satisfactorily for CajunBot.
- When the results of this system were passed to the actuators, CajunBot was able to navigate successfully, without running into problems of over steer, under steer, or falling into oscillatory states.

Steering Control: Comments

- Once again, there were some highly dynamic elements that had to be taken into account in this solution.
- These dynamic elements were a direct consequence of the embodied nature of the CajunBot steering task.
- These dynamic elements could have been abstracted away from, or just ignored if the system just had to operate in a highly abstract domain, which could idealize the environment (i.e. if this system was deployed in a purely classical domain).

Subsystem 3: Simulations



Subsystem 3: Simulations

- Simulation environments were used extensively in the development of CajunBot.
- Given that simulated environments have no direct contact with the real world, they appear to fall into the 'classical/cognitive' approach far more than they do the 'embodied' approach.
- Yet, they played a crucial role.
- There were roughly two strategies used with simulations;
  - *Targeted simulations* were used to answer questions about particular problems.
  - *Comprehensive simulations were used to solve full system integration and testing problems.*

Simulations: Targeted Simulations

- Potential Field Visualizations:
  - These were used extensively in the development of the L-Nav module.
  - Different methods of generating potential field flow maps were tested, to determine which methods gave the best results.
  - This testing led to an abandonment of the neural network based potential field generation strategy that was initially used, in favour of using simple linear potential fields.
- G-Nav and L-Nav interactions:
  - Initially, G-Nav invoked L-Nav whenever an object was detected in sensor range.
  - The method of transforming G-Nav and L-Nav into concurrent processes was perfected by simulation.

Simulations: Comprehensive CajunBot Simulations, Early Stages

- Comprehensive CajunBot simulations were constructed by incrementally adding more and more realism to the simulations, both with respect to CajunBot and to the environment.
- The addition of realistic steering delays brought about something of a crisis, as this revealed that even at low speeds, the direction of travel of the system would oscillate, leading to crashes.
- Fortunately, it was also discovered from these simulations that the adoption of the waypoint interaction system between the G-Nav and L-Nav systems improved these steering issues.

Simulations: Comprehensive CajunBot Simulations, Later Stages

- As the integration of the various systems into the simulations continued, the process continued to provide useful information and to highlight bugs.
  - Coordinate transformation bugs were detected. For example, a failure to translate between meters and centimetres, when reading from the blackboard communication system was discovered.
  - A few waypoint extraction bugs were also found.
  - The L-Nav system was found to give uninformative error messages when it encountered unanticipated types of data from the richer simulation environment.
  - This suggested that a richer simulation environment, that used a broader spectrum of data should have been used in earlier simulations.

Simulations: Simulated Performance

- Using the simulation environment, the testing of CajunBot in various simulated situations became possible. This too was informative.
  - It was possible to determine the performance of the system when an obstacle (for example, a van) was located exactly on top of a G-Nav global waypoint.
  - When this eventuality was tested, CajunBot performed perfectly.
  - It was also possible to determine how the system would perform under various circumstances, such as when CajunBot was in a dead end canyon, with the next way point directly behind the end of the canyon.
  - The simulations showed that CajunBot would also manage this situation effectively, for the most part.

Simulations: Comments

- It is interesting to note that the use of simulations was motivated in large part by real world factors, thus suggesting that embodiment may not really be quite as far away from these otherwise classical approaches as might be initially supposed.
- Actual system testing has the following drawbacks;
  - It is expensive to conduct,
  - It is time consuming to conduct,
  - It carries with it a risk of damage to the hardware components of CajunBot.
- Simulations were used, in part, to mitigate against these drawbacks. They played an important role in developing, refining and testing of CajunBot.

CajunBot in Action

A video of CajunBot in the Qualifying round.

The Outcome

- Having made it through the qualifying rounds, CajunBot competed in the final of the Grand Challenge at Primm, Nevada, with 23 other teams.
- CajunBot ran well for the first 17 miles of the Challenge.
- Then CajunBot was ordered to pause, in order to provide a safe distance between it and other competitors.
- However, after this pause, CajunBot never moved again and was eventually eliminated from the competition.

What Happened?

- Arun Lakhotia, of Team CajunBot described what happened:

"CajunBot was put in pause mode for about fifty minutes to allow other oncoming bots on the track to clear. In the pause mode CajunBot pulls its breaks fully, which means the motors are engaged to their maximum capacity. Normally at this state the motor should lock and not use power. But for some reason, the motor continued to drain power, that too very high amperage. A sustain draw of that level of power for fifty minutes fried the motor."

The Underlying Cause

- A few days prior to the Challenge, CajunBot's transmission failed and had to be replaced.
- When the new transmission was installed it was half an inch out of alignment
- This is what caused the actuator motor to stay powered up and burn out.
- Engineering failures are another peril of embodiment!

Conclusions

- It is clear that CajunBot made used of techniques that were consistent with both the Classical/Cognitive view, as well as techniques that were consistent with the Embodied view.
- This suggests that the two views should not be seen so much as competitors, but rather as complimenting each other.
- Thus, the evidence from studying the subsystems of CajunBot seems to suggest that a compatiblist position on the two views is the correct one to take.
- Furthermore, as the philosopher of science Paul Feyerabend has notoriously argued, employing more than one methodology on a problem has distinct advantages.

Acknowledgments

I wish to thank Anthony Maida, Suresh Golconda, Arun Lakhotia, and the rest of the CajunBot Team for all their assistance in preparing this talk.

For further details on CajunBot, see http://www.cajunbot.com.