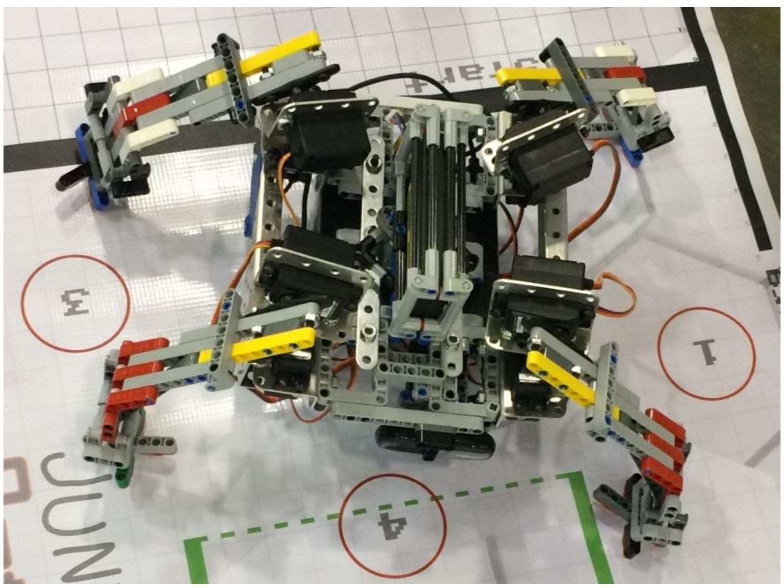
First Steps: An Experiment in Insectoid Movement

Ryan Sizemore

Sizemore Academy (sizemoreacademy@gmail.com)

First Steps: An Experiment in Insectoid Movement

Learning does not have to stay within a certain box consisting of structure, rules, and limitations. It's when we step outside of the box that we can flex our brains and gather the most information. I've participated in Botball competitions during the last four years, which has vastly increased my knowledge in mechanical design and programming. There's nothing wrong with having rules and regulations, limitations on parts, and specific goals to accomplish. However, due to the requirements each year, my design stayed within specific and predictable boundaries. This year, Botball offered high school challenge days. Since it wasn't a competition, I decided to take a small step outside the box. I had a deadline to consider, but I was free to create two robot designs that truly deviated from the norm. While most Botball creations utilize wheels or treads, my bots walked.

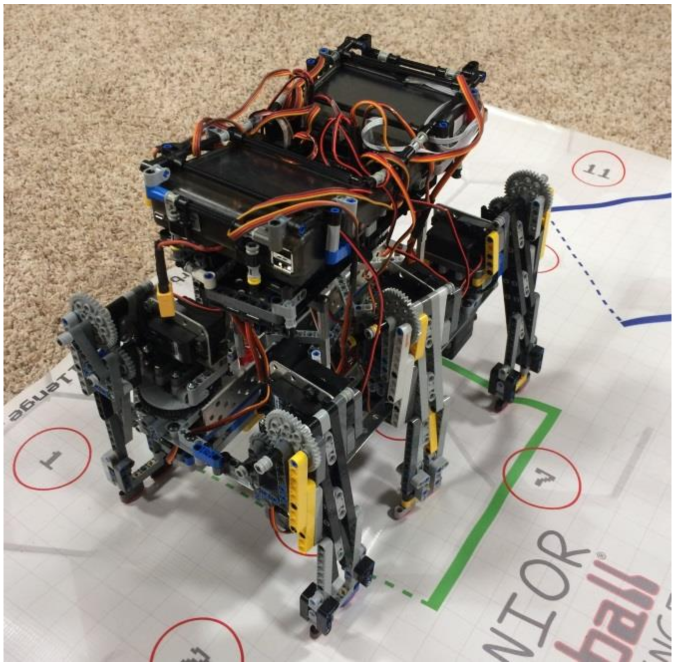


The first model used a quadruped design with a servo and motor controlling vertical movement and rotation, respectively. I originally didn't use metal in the chassis because of weight concerns, but it soon became necessary due to the entire bot folding. For traction, I utilized rubber pieces from Mindstorms. I secured the controller using a thin frame on the underside of the bot. Its gait was moving each leg up-forward-down one at a time, and then moving all back simultaneously.

This design had two core problems. First, it was closer to sliding than to crawling. The bot couldn't balance on three legs; therefore, it would tip when any of the legs rose. However, it wouldn't tip far before coming to rest on the low-hanging controller cage. Eventually, it would be fully supported by the controller while dragging itself along.

The second problem was that the bot wouldn't move straight. This problem stemmed from three separate issues. First, I originally programmed the front left leg to rise first, which resulted in the bot sliding slightly left. This caused the bot to curve noticeably left over time. An obvious fix would have been to alternate the starting leg. However, I didn't think of it at the time. The second issue was inherent in the leg design. The leg shape was fixed, causing the feet to rarely be level with the ground. Finally, the rotation of the legs was controlled by motors. This was very inconsistent because I wasn't using ticks, another item I didn't think about at the time. Eventually, the legs would be significantly out of position.

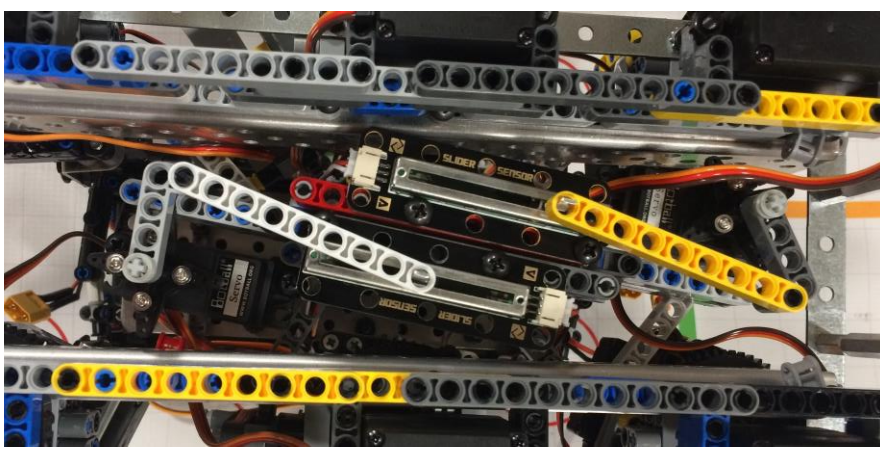
While this first design wasn't able to truly walk, it succeeded in shuffling a moderate distance. Through all this, I gathered several ideas for a better design. First of all, it would need the ability to fully hold itself up to be considered walking instead of dragging. Second, the legs would need to rotate via servos for necessary consistency. Third, it would be important to design the legs to keep the feet level with the ground.



Because Botball 2019 was starting, I had to dismantle the first bot for parts. However, once the Botball season ended, I had ample parts for an improved design. The layout was much more ambitious, sporting two controllers and six legs. Two controllers allowed for double the servo and motor ports, which in turn allowed for more legs. However, this compounded the complexity of building and programming.

A communication system is necessary for two controllers to work in tandem, which is why I created a two-way communication device using slide sensors and servos. This allowed the controllers to communicate a wide range of values. However, the servos aren't precise enough to give exact values. To solve this, I divided the range of the slide sensors into five sections. These sections allowed for each side to send five separate signals while remaining consistent.

Communication Sensors Shown Below



The extra pair of legs allowed the bot to hold up its 7.5 lb. weight. In addition, they allowed the bot to remain balanced and standing when a leg was raised. The legs were much more consistent with the use of servos for rotation. The feet were kept level by utilizing a shifting parallelogram in each leg. These changes fixed many of the original bot's problems but had numerous construction issues along the way.

The first difficulty was mounting the servos, which are perfectly designed to not line up with anything. This was solved by using Lego mountings instead of metal, which would flex slightly and allow the axles to line up. The second problem was when one back leg was moved forward and the second was raised, the entire bot would tip backwards. To fix this problem, I added a support strut on the stern to hold weight.

The next issue was one of programming; the bot was utilizing the same movement pattern as the first bot (move all and then rotate back). This method, while working, inched forward torpidly. After some research, the speed problem was fixed by adopting a tripod gait. This was an improvement because it utilized half the legs for the same motion and effectively doubled the speed.

Unfortunately, this put half the bot's weight on a single middle leg repeatedly, which caused the leg (held on by a small army of pegs) to rip off with every attempted step. My solution was to add screws to the critical connection points.

The last issue is inherent in Legos: everything flexes. Under the considerable weight of the bot, the legs would bend outward and up. They would also not lower fully because the motors weren't strong enough to lift the weight. To fix this, I added a brace that would put additional tension on the leg when supporting weight. I also ran the motors longer when lowering the legs in an effort to fully lower them, which helped immensely. A more extensive fix existed, but I was working under a deadline.

During design, I considered two ideas that I unfortunately did not implement due to time. The first idea was to 3D print parts or find non-Lego parts to reduce flexing. The second idea was to use touch sensors or reflectance sensors for knowing when the leg was fully down. An idea I did test was reducing the contact area of the feet. This failed because the legs couldn't get enough traction to avoid sliding under the weight.

Ultimately, my final design met my goals. It walked consistently straight and was able to turn. While it was relatively slow, it was much better than the first non-tripod speed. It made a 90^o turn and walked around 3.5 feet before a leg finally tore off from the stress. The communication functioned perfectly, allowing the controllers to coordinate their movements. The system was actually underutilized; although it can communicate four unique signals, I only used it similar to a binary sensor.

As I worked through my project, opportunities to learn new things sprouted and sometimes even grew into complicated challenges. I had to come up with new strategies and brainstorm with more knowledgeable people to find solutions. I was able to implement all this knowledge and eventually end up with a robot that wasn't quite perfect but definitely stepped outside the box.