guarantee a constant current to deliver a constant τ_{max} after the gear reduction, independently of the speed of the motors.

Let's assume now that the robot center of mass C_1 is located along the wheel axis. If the mass and moment of inertia of the entire robot (chassis plus weapon) in the wheel axis direction with respect to C_1 are m_1 and I_1, then the torque τ_{max} would generate an angular acceleration $\alpha = \tau_{max}/I_1$. Note that most of the value of I_1 will come from the weapon, because the rest of the chassis is very close to the wheel axis and do not contribute much to the moment of inertia. For the robot to strike, it needs to turn about 180 degrees (π radians), therefore $\pi = \alpha \cdot t^2/2$, where t is the short time the robot takes to strike.

If the robot is initially at rest, then its chassis will start moving backwards, due to the linear acceleration $f/m_1 = \mu_t \cdot g$ caused by the ground force f. During the strike period t, the distance x_{back} the chassis moves backwards is then $x_{back} = \mu_t \cdot g \cdot t^2/2$. Eliminating t and α from these equations we get $x_{back} = \pi \cdot I_1/(m_1 \cdot r_w)$.

So, if its weapon has a radius r, then it will hit a spot at a distance $(r - x_{back})$ from its position when it started the attack, see the pictures above. The driver has to get a feeling of this distance after some practice, otherwise the weapon will hit short of the opponent's position. Note that x_{back} does not depend on the coefficient of friction μ_t, it is a constant for each robot.

But x_{back} can be compensated for, if the robot starts striking when it is moving with some initial forward speed v_{x1}, not at rest as before. Then $x_{back} = \mu_t \cdot g \cdot t^2/2 - v_{x1} \cdot t$, which would be equal to zero right at the end of the attack if

$$v_{x1,ideal} = \frac{\mu_t \cdot g}{2} \cdot t = \frac{\mu_t \cdot g}{2} \cdot \sqrt{\frac{2\pi}{\alpha}} = \frac{\mu_t \cdot g}{2} \cdot \sqrt{\frac{2\pi \cdot I_1}{\mu_t \cdot m_1 \cdot g \cdot r_w}} = \sqrt{\frac{\mu_t \cdot g}{2} \cdot \frac{\pi \cdot I_1}{m_1 \cdot r_w}}$$

So, if the robot is initially moving towards the opponent at this $v_{x1,ideal}$ speed, it will hit exactly at the distance r from its initial position. If moving slower than that, it will hit short of that distance. If moving faster, it will hit beyond that. This is why overhead thwackbots are so difficult to drive, it is up to the driver to get a feeling of the attack distance as a function of the attack speed. Note that, because this function depends on μ_t, a dirtier arena (leading to a lower μ_t) will result in a lower $v_{x1,ideal}$, which needs to be adaptively controlled by the driver. Poor driver.

Finally, the weapon maximum angular speed ω_b and energy E_b can be calculated from α and t, resulting in

$$\omega_b = \alpha \cdot t = \sqrt{2\pi\alpha} = \sqrt{\frac{2\pi \cdot \mu_t \cdot m_1 \cdot g \cdot r_w}{I_1}} \quad \text{and} \quad E_b = \frac{1}{2} I_1 \omega_b^2 = \pi \cdot \mu_t \cdot m_1 \cdot g \cdot r_w$$

So, since m_1 is basically the mass of the weight class and g is a constant, to maximize the attack energy E_b you must have rubber wheels with large radius r_w, and also with large width to maximize its coefficient of friction μ_t. And, of course, the drivetrain gearmotors should be able to provide altogether at least $\tau_{max} = \mu_t \cdot m_1 \cdot g \cdot r_w$ to reach these maximum ω_b and E_b values.

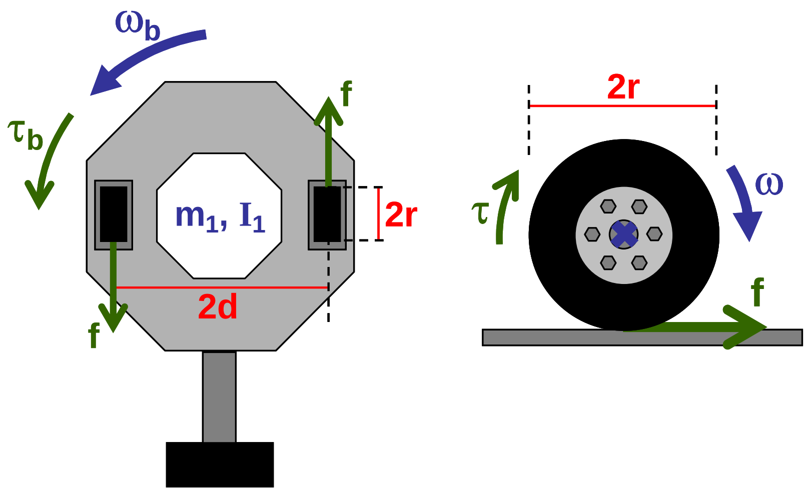
But be careful, because horizontal spinners love large wheels. It's their favorite breakfast.

6.9. Thwackbot Design

A thwackbot can be thought of as a full body spinner. It uses the power of its two (or more) wheels to spin up its entire body. As seen before, its normalized effective mass M_1' can reach very high values, from 1/3 for disk shaped designs tending towards 1/2 for ring shaped designs, being able to store a lot of kinetic energy.

6.9.1. Thwackbot Equations

Let's consider a thwackbot with mass m_1, moment of inertia I_1 in its spin direction with respect to its center of mass, and two wheels with radius r separated by a distance 2·d, as shown in the picture to the right. We'll assume that the robot center of mass is located in the middle of the line between the wheel centers, otherwise the robot would be unbalanced, compromising its maximum angular speed and drivability. So, if it has some asymmetrical feature such as a single hammer, as pictured above, then the wheel axis location must be carefully calculated to guarantee that it ends up balanced.



Each wheel has a variable angular speed ω, with a maximum value ω_{max}, and it receives a torque τ from the gearmotor output. This wheel torque will cause a traction force f, which is equal to τ/r if the wheel does not slip. Assuming that each wheel bears half of the robot weight, the maximum possible value for f is μ_t·g·m/2, where μ_t is the coefficient of friction between the wheel and the ground, and g the acceleration of gravity. If τ/r is greater than μ_t·g·m/2, then the wheel will slip.

Note that it is not a good idea to increase the width of the tires to increase μ_t, because tires with large width tend to waste a lot of energy while making sharp turns, due to the slipping that always occurs along their width. This slip happens because the inner surface of a wheel with width w is closer to the center of the robot, moving along a circle with radius (d – w/2), while the outer surface moves along a circle with radius (d + w/2). The mid-section of the wheel, which moves along a circle with radius d, makes the inner surface waste some energy by slipping forward to catch up with the mid-section, while the outer surface wastes energy slipping backwards. So, while overhead thwackbots should have wide tires, thwackbots should use instead relatively thin tires to avoid this energy loss.

The wheel torque τ used in the above equations depends on the wheel speed ω, as seen in chapter 5 for DC motors. We can then define an effective wheel torque function τ(ω) that includes this dependency, and which is also limited by the value μ_t·g·m·r/2 which would make the wheel slip. This effective torque τ(ω) depends not only on ω, but also on the motor and battery properties (such as K_t, K_v, I_{stall}, I_{no_load} and R_{system} for DC motors, see chapter 5), gear ratio n:1, and the value of $\mu_t \cdot g \cdot m \cdot r/2$. For instance, if min(x,y) is the function that returns the minimum value between x and y, then DC motors would result in an effective wheel torque function

$$\tau(\omega) = \min\left\{ n \cdot K_t \cdot (I_{stall} - I_{no\_load} - \frac{\omega \cdot n}{K_v \cdot R_{system}}), \quad \frac{\mu_t \cdot m \cdot g \cdot r}{2} \right\}$$

Let's first calculate the time Δt_{drive} the robot takes to accelerate to, for instance, 95% of its top speed, while driving on a straight line. In this case, the traction forces f of both wheels would be directed towards the same direction, contrary to the figure above. The forward speed of the robot would be $v = \omega \cdot r$, with a maximum value $v_{max} = \omega_{max} \cdot r$. So, the forward acceleration of the robot dv/dt would be equal to the angular acceleration $d\omega/dt$ of each wheel multiplied by the wheel radius r, thus $dv/dt = r \cdot d\omega/dt$. Both forward forces f cause the robot to accelerate, following the equation $2 \cdot f = m \cdot dv/dt$. We can then use $f = \tau(\omega)/r$ and $dv/dt = r \cdot d\omega/dt$ in this equation, to obtain

$$2 \cdot f = 2 \cdot \frac{\tau(\omega)}{r} = m \cdot r \cdot \frac{d\omega}{dt} \quad \Rightarrow \quad \Delta t_{drive} = \int dt = \frac{m \cdot r^2}{2} \cdot \int_0^{0.95\omega_{max}} \frac{d\omega}{\tau(\omega)}$$

The integral shown above is not difficult to calculate, it was obtained for DC motors in chapter 5.

Let's now calculate the spin up time Δt_{weapon} until the robot reaches 95% of its top weapon speed. The chassis angular speed ω_b is equal to the wheel linear speed $\omega \cdot r$ divided by d, so we have $\omega_b = \omega \cdot r/d$. Thus, their angular accelerations are related by $d\omega_b/dt = (d\omega/dt) \cdot r/d$, and the maximum angular speed of the chassis is $\omega_{b,max} = \omega_{max} \cdot r/d$.

The traction forces f are now in opposite directions, as shown in the figure above, spinning up the chassis with a torque $\tau_b = 2 \cdot f \cdot d = 2 \cdot \tau(\omega) \cdot d/r$, which is equal to the robot moment of inertia I_1 in the spin direction times its angular acceleration $d\omega_b/dt$. We can then use $d\omega_b/dt = (d\omega/dt) \cdot r/d$ to obtain

$$2 \cdot f \cdot d = 2 \cdot \frac{\tau(\omega) \cdot d}{r} = I_1 \cdot \frac{d\omega_b}{dt} = I_1 \cdot \frac{r}{d} \cdot \frac{d\omega}{dt} \quad \Rightarrow \quad \Delta t_{weapon} = \int dt = \frac{I_1 \cdot r^2}{2 \cdot d^2} \cdot \int_0^{0.95\omega_{max}} \frac{d\omega}{\tau(\omega)}$$

So, it is easy to see that decreasing the distance $2 \cdot d$ between the wheels increases the maximum weapon speed $\omega_{b,max}$, but it also increases the weapon spin up time Δt_{weapon}. It is not a good idea to have Δt_{weapon} much higher than 4 to 8 seconds, otherwise the thwackbot won't stand a chance against a very aggressive rammer or wedge, so choose wisely your distance $2 \cdot d$.

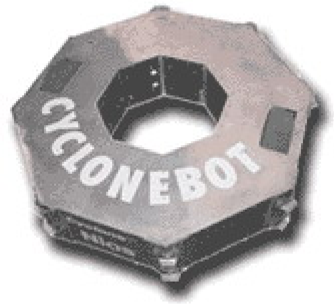
Since both Δt_{weapon} and Δt_{drive} depend on the same integral, they are related by

$$\Delta t_{weapon} = \frac{I_1 \cdot r^2}{2 \cdot d^2} \cdot \int_0^{0.95\omega_{max}} \frac{d\omega}{\tau(\omega)} = \frac{I_1 \cdot r^2}{2 \cdot d^2} \cdot \frac{2 \cdot \Delta t_{drive}}{m \cdot r^2} = \Delta t_{drive} \cdot \frac{I_1}{m \cdot d^2}$$

So, if you've already calculated the acceleration time Δt_{drive} of the drive system, as described in chapter 5, then the weapon spin up time is easily obtained from the above equation.

If your thwackbot has its wheels close to its perimeter, it is very likely that I_1 is equal to or a little lower than $m \cdot d^2$, therefore Δt_{weapon} and Δt_{drive} would be approximately equal. This is not good, because the low Δt_{drive} required to make the robot agile would lower the spin up time Δt_{weapon}, probably resulting in a low weapon energy. Even if you upgrade your drive motors, they won't be able to deliver a very high power to spin up the weapon in a short Δt_{weapon}, because the wheel forces are limited to $\mu_t \cdot g \cdot m/2$ (for a two wheeled thwackbot), they would slip beyond that. And a larger Δt_{weapon} would naturally result in a large Δt_{drive}, compromising the drivetrain acceleration. So, wheels with a large distance $2 \cdot d$ usually result in a relatively slow robot, or one with low weapon energy.

One way to avoid this is to decrease the wheel distance $2 \cdot d$. But they can't be too close together, otherwise any unbalancing in the robot or any attack from a wedge could make the spinning chassis touch the ground and launch itself. Using casters near the perimeter could help making the thwackbot stable, but there's a good chance they'll be knocked off or broken during an angled impact.



Another way to increase the weapon energy without compromising the drivetrain acceleration is to maximize the value of I_1. This is done by concentrating most of its mass in its outer perimeter, trying to approach the upper limit 1/2 of its normalized effective mass M_1', as done in the ring-shaped heavyweight Cyclonebot (pictured to the right).

6.9.2. Melty Brain Control

The main drawback of a thwackbot is the required complexity of its drive system to enable it to move around in a controlled way while spinning. The idea, referred to as *translational drift*, is to somehow oscillate the speed or the steering of each wheel with the same frequency that the entire robot spins. The speed oscillation solution is usually called "melty brain" or "tornado drive" control, while the steering oscillation has been called "wobbly drive" or "NavBot steering" control.

Despite its name, the math behind "melty brain" control is not hard enough to melt someone's brain. For instance, when the chassis of a two-wheeled thwackbot is spinning at ω_b, at every time period $T = 2 \cdot \pi / \omega_b$ one wheel will be facing the desired direction to which you want to move, while the other will be facing the opposite direction. If at this moment the first wheel has a slightly larger speed than the other, then the robot will end up moving in that desired direction. The angular speed of each wheel would need to be $\omega = \omega_b \cdot d/r \pm (v/r) \cdot \cos(2 \cdot \pi \cdot t / \omega_b + \varphi)$, with the plus sign for one wheel and the minus sign for the other, where t is time, v is the desired linear speed, r is the wheel radius, and φ is a phase angle that will define the direction of the movement.

"Melty brain" control is not easy to implement, because the time period T is very short. You need a very fast acting control system, powerful drive motors to be able to change the wheel speeds in such high frequency, and some way to measure both the robot angular speed, to obtain ω_b, and its orientation at the end of each time period, to define φ. In theory, wheel encoders or other angular position sensors could be used to estimate ω_b and φ, using dead reckoning, but they do not work in practice because there is wheel slip. Digital compasses do not work well because the high motor currents usually affect the readings of the Earth magnetic field.

Most successful implementations of “melty brain” control require that the driver emits some light beam, usually infra-red or laser, in the direction of the thwackbot. This beam is detected by a sensor on the periphery of the robot, allowing it to estimate ω_b from the time period T between two sensor readings, $\omega_b = 2\cdot\pi/T$. The robot can also estimate its orientation at each reading, which would be the one facing the driver’s light source. It is a good idea to use two light sensors close together instead of one, to minimize the chance of both picking up random reflections of the beam from the arena or the opponent robot, which would confuse the control system.

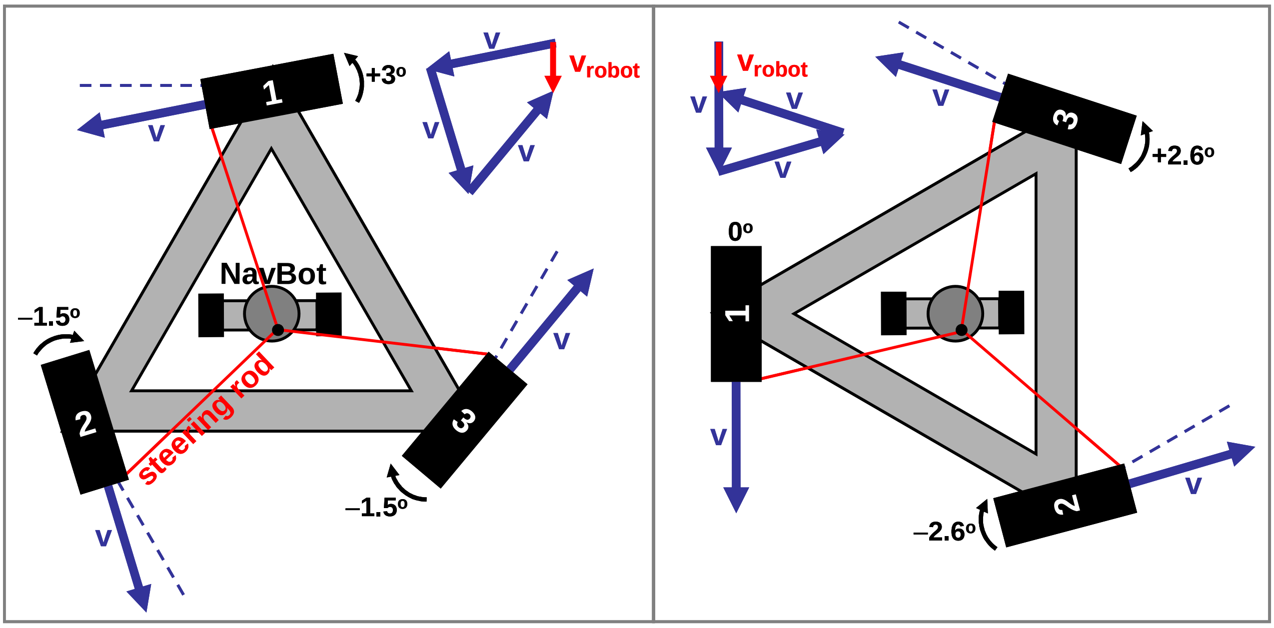
Other successful implementations of “melty brain” control use accelerometers or gyroscopes to measure or infer ω_b, and a led on its periphery that blinks with a period $T = 2\cdot\pi/\omega_b$, calculated in real time from the current ω_b. If the ω_b measurement is accurate enough, the led will only blink once per revolution, apparently at the same position in space, which would point to a nominal direction. It is then up to the driver to look at the position of the led light and use the radio control to change accordingly the phase angle φ of the robot software, allowing the thwackbot to move around.

A few robots have successfully implemented “melty brain” control through electronics, such as the heavyweight Cyclonebot, the middleweight Blade Runner, the lightweight Herr Gepöunden, the featherweight Scary-Go-Round, and the antweight Melty B, as pointed out by Kevin Berry in his “melty brain” Servo magazine articles from February and March 2008.

6.9.3. NavBot Control

Other robots, such as the middleweight WhyNot (pictured below to the left) and the heavyweight Y-Pout (to the right), have followed a mechanical approach to implement steering control. These robots have three wheels in a 120^o configuration, driven by high power motors.

The steering is performed by making each of their three active wheels slightly steer in and out about 3 degrees at every robot revolution. The steering angle of each wheel is approximately given by $\theta_1 = 3^o \cdot \cos(2\cdot\pi\cdot t/\omega_b + \varphi)$, $\theta_2 = 3^o \cdot \cos(2\cdot\pi\cdot t/\omega_b + 2\pi/3 + \varphi)$ and $\theta_3 = 3^o \cdot \cos(2\cdot\pi\cdot t/\omega_b + 4\pi/3 + \varphi)$, mechanically implemented using three steering rods connected to a cam mechanism on a small independent radio-controlled two-wheeled robot called NavBot, as seen in the figures below. The phase angle φ that defines the direction of the movement is mechanically controlled by the direction of the NavBot. The NavBot has only one motor, a low power one, which works as a differential drive. When this motor is locked, a worm gear makes sure that both wheels turn together to make the NavBot move straight, pulled by the main robot, keeping constant the phase angle φ. And when this motor is powered, following a radio control signal, one of the small wheels starts spinning with a different speed than the other, making the NavBot turn, which will change the phase angle φ and thus make the entire robot move in the new direction of the NavBot. The NavBot will then be pulled in this new direction that it is facing.



The figures above show that the differential steering of the wheels make the robot move with a constant speed $v_{robot} = v \cdot \sin 3^o$, where v is the linear speed of each wheel. The direction of v_{robot} does not change while the robot is spinning, as shown in the figures, as long as the NavBot keeps its orientation. The orientation of the cam mechanism must be setup in such a way that the directions of v_{robot} and of the two small NavBot wheels always coincide, allowing the NavBot to easily follow the robot path while being pulled.

Note that these robots can be categorized as a thwackbot, if we consider that its 3-wheeled drive system makes the entire robot spin, with the NavBot being a secondary robot (almost as a Multibot). Or they can be categorized as ring or shell spinners, if we consider that the NavBot is the main robot, with a two-wheeled drivetrain, and the spinning triangle is the ring or shell, powered by three weapon motors embedded in it. It's just a matter of point of view.

The main advantage of this system is that it is possible to implement a thwackbot (or ring or shell spinner) with a normalized effective mass M_1' close to 1/2, using the same high power motors for both the weapon and drive system. Using high diameter wheels, it might be even possible to make the robot invertible, but the NavBot would be exposed to hammer attacks without a top cover.

This system, however, has a few disadvantages, because the drive speed of the robot is a function of the weapon speed, due to the relation $v_{robot} = v \cdot \sin 3^o$. So, when the weapon is at full speed, with the wheels at full linear speed v_{max}, the robot will have to move around at a full speed $v_{robot,max} = v_{max} \cdot \sin 3^o$, all the time. It won't be able to slow down or stop its drive system, it will have to keep moving until it hits something, slowing down the weapon. And, most importantly, after a major hit, the weapon will probably be spinning so slowly that the robot won't be able to move around. An aggressive rammer would only need to survive the first hit, and then keep ramming the thwackbot to prevent its weapon from spinning up. The thwackbot wouldn't be able to run away from the rammer to try to spin up its weapon, because its drive speed would be too slow due to the slow weapon speed.

6.10. Launcher Design

Launchers need to deliver a huge amount of energy during a very brief time. Because of that, they're almost invariably powered by high pressure pneumatic systems. A very simplified estimate can show that a cylinder with piston cross section area A, stroke d, with pressure p, can accelerate a total mass m (including the mass of its piston) with an average power of up to $(p \cdot A)^{1.5} \cdot (0.5 \cdot d/m)^{0.5}$. For instance, a 4" bore cylinder with 8" stroke pressurized at 1000psi would accelerate a 220lb mass with an average power of about 566HP.

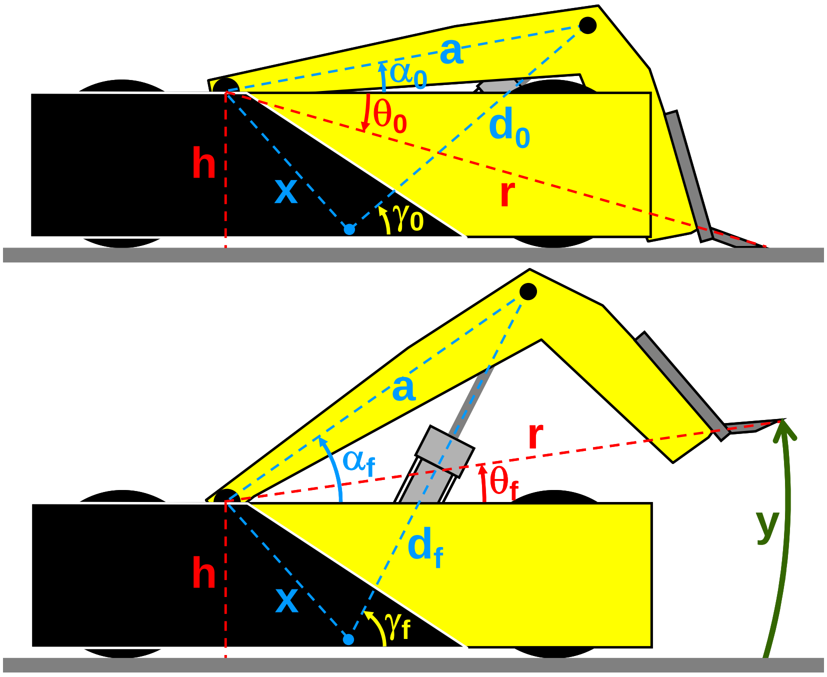
Of course, this power is only delivered during a very brief time, but a light weight electric motor or internal combustion engine cannot supply that. Unless the motor is used to store kinetic energy in a flywheel during a few seconds, with an ingenious and very strong mechanism that suddenly transfers this energy to the launcher arm, as done by the Warrior SKF robot (pictured to the right). But such sturdy mechanism is not simple to build.

Hydraulic systems are not good options either for launchers. They can deliver huge forces and accelerations, but their top speed is relatively low.

Most launchers try to either maximize the height or the range of the throw. The "height launchers" try to launch the opponent as high as possible, trying to flip it while causing damage when it hits the ground. The "range launchers" try to launch the opponent as far as possible, not necessarily high, trying to throw it out of bounds to the arena dead zone.

Against horizontal spinners, especially undercutters, there is a strategy used not only by launchers, but also by lifters and wedges, which is to tilt the spinner chassis so that its spinning weapon hits the arena floor, usually launching it. The picture to the right shows the middleweight launcher Sub Zero tilting the chassis of the spinner The Mortician, to launch it with the help of the additional energy from the spinning bar hitting the ground.

6.10.1. Three-Bar Mechanisms

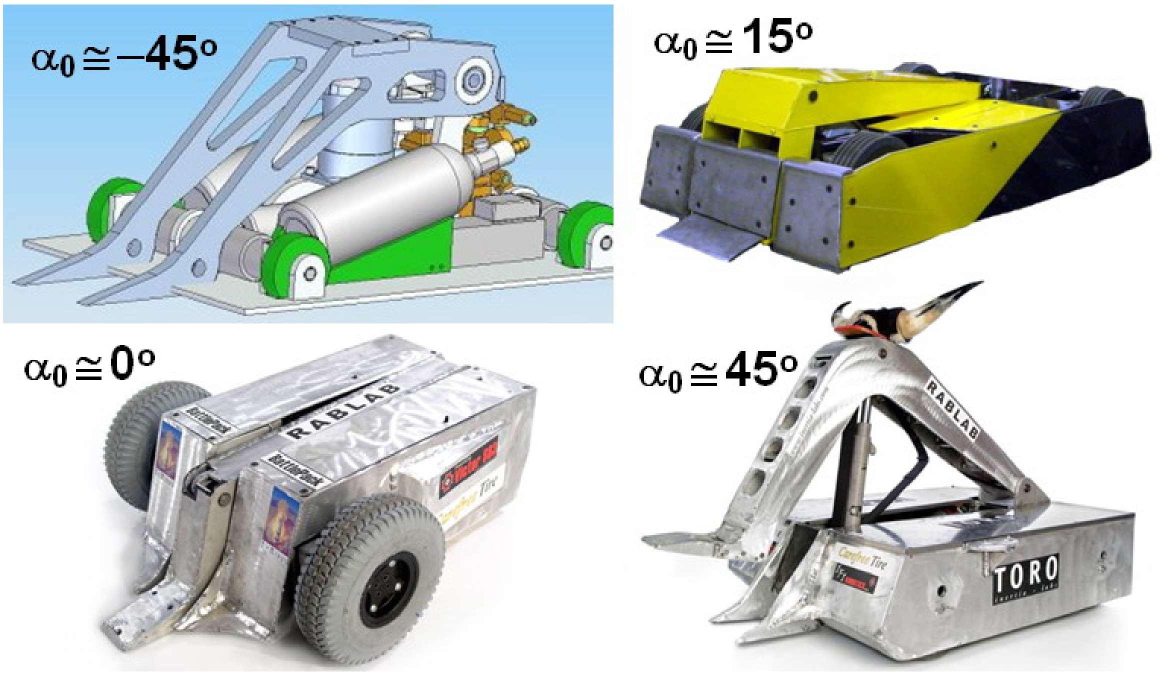


A very popular launcher design uses a so-called three-bar mechanism, as pictured to the right. The "three bars" are the pneumatic cylinder, with total length varying from d_0 to d_f during the launch, the main structure of the launcher arm, with constant length a, and the part of the robot chassis that connects the arm and cylinder pivots, with constant length x. The launcher arm tip features a wedge-like scoop, at a constant distance r from the arm pivot. The initial and final angles of the main structure of the arm are defined as α_0 and α_f, while the angles for the arm tip are θ_0 (which is negative for the particular robot shown in the picture) and θ_f. The angle variation during the launch is then $\alpha_f - \alpha_0$, which is equal to $\theta_f - \theta_0$. The initial and final angles of the cylinder are γ_0 and γ_f, as shown in the picture.

During the launch, the arm tip follows the green circular path shown above, with an arc length y that is usually between h and 2·h, where h is the height of the launcher chassis. The direction of this path is, in average, equal to 90° plus the average angle θ between θ_0 and θ_f, which is roughly the direction of the launching force. In the picture above, $\theta = (\theta_0 + \theta_f)/2$ is approximately zero, leaving in average an almost vertical (90°) force. Note also that a negative θ_0 is a good idea, it makes the arm tip move forward in the beginning of its path, helping to properly scoop the opponent.

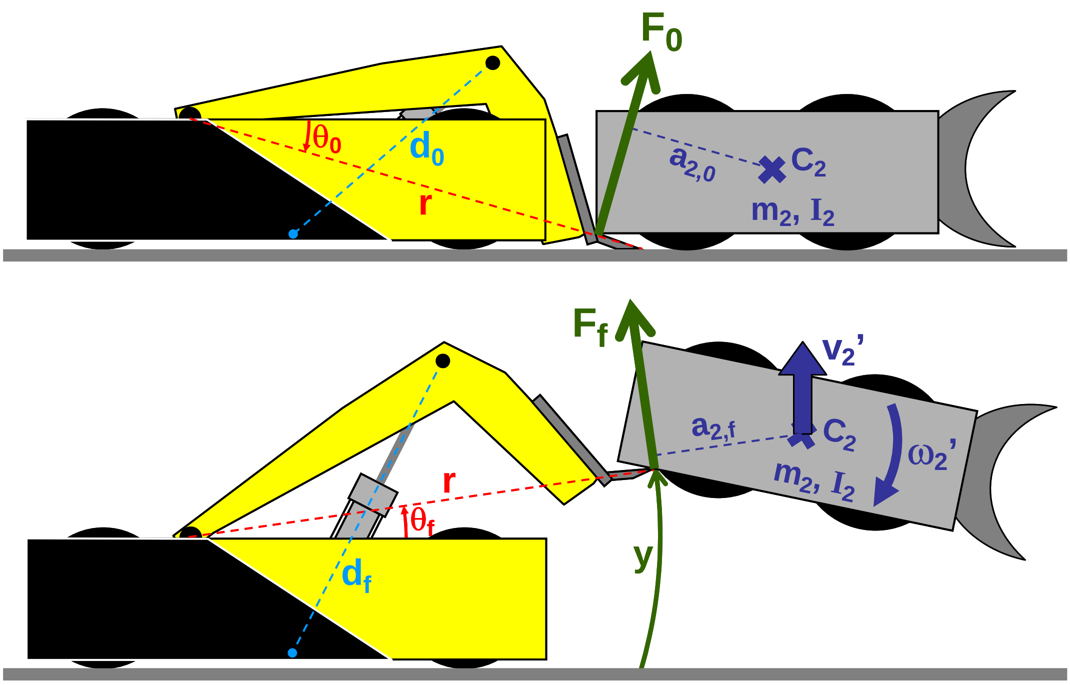
The figures in the next page show 4 different launcher configurations. The lightweight Rocket has both α_0 and θ_0 a little above –45°, making it a good "range launcher" due to the average 45° force it delivers. The only problem with this design is that it requires a high chassis to get a sufficiently long arm with θ_0 = –45°, decreasing its stability and making it more vulnerable to spinners.

The "height launchers" Bounty Hunter and T-Minus were able to lower their height when the arms are retracted, due to their low α_0, about 15° and 0°, respectively. Their average θ during the launch is close to zero, leading to an almost vertical force that allows them to throw the opponents very high in the air. But their low initial cylinder angle γ_0, below 45°, puts a lot of stress on the back pivot joint of the arm, initially trying to push forward the arm with almost the same force used to launch the opponent. This forward force, which tries to rip off the back pivot, is not necessarily wasted because it does not produce work. But this added force increases the friction losses in the joints. This is the price to pay for a low profile launcher.



Toro (pictured above), on the other hand, has γ_0 close to 90^o, not stressing too much the arm pivot. It is also a “height launcher” because its average θ is close to zero. But to be able to accommodate its relatively long cylinders with such high γ_0, it needed to increase its α_0 to about 45^o, as shown in the picture, making its tall launcher arm vulnerable to horizontal spinners. Note the curved strap under Toro’s arm that limits the stroke of the cylinders, avoiding their self-destruction when reaching their maximum stroke.

The launching calculations are more complicated than in the impact problem, because the arm does not hit the opponent, it shoves it. Therefore, the contact point between the launcher arm and the opponent may move during the launch. The figure below shows an opponent being launched.



When the launching starts, the contact point is usually located further in the back of the arm scoop (at a distance smaller than r from the back pivot of the arm), and very far from the opponent's center of mass C_2. The initial launching force F_0 might have a direction θ_0 with respect to the vertical, defining the distance $a_{2,0}$ to C_2. When the contact ends, the contact point of the final force F_f will probably have shifted to the tip of the arm scoop (at a distance r from the back pivot), defining a different distance $a_{2,f}$ to C_2.

The intensities of F_f and F_0 are probably different, even if the force in the cylinder is constant during the entire launch, because of the different mechanism configurations for angles θ_0 and θ_f. Their directions are also different, F_f might have a direction that makes the angle θ_f with respect to the vertical if there's enough contact friction at the arm tip, or it might be perpendicular to the opponent's bottom plate if there's no contact friction, or it may have some direction in between.

Also, finding out at which value of the path length y the contact will end is not simple, it depends a lot on the opponent's mass m_2 and moment of inertia I_2 in the direction it ends up spinning. Only one thing is certain: it is that the kinetic energy you may induce in the opponent cannot be higher than the energy delivered by the pneumatic cylinder, $E_b = p \cdot A \cdot (d_f - d_0)$. In practice, this theoretical value is not reached because of friction and pneumatic losses, gravity effects, and because of the inertia of the launcher arm, which needs to be accelerated as well.

Also, since the opponent usually tends to rotate away from the launcher arm during the launch, it is more efficient to maximize the forces than the path length y, to increase the delivered work. A very large path y is not effective, because the contact between the robots will probably be lost before the end of the stroke of the pneumatic cylinder. It is better to have a cylinder with large diameter and high pressure, to increase the force, than to have a cylinder with very large stroke. The ideal stroke would be the one that ends slightly after the contact between the robots is lost. The straps, or other system that limits the cylinder movement, should also be dimensioned in this way.

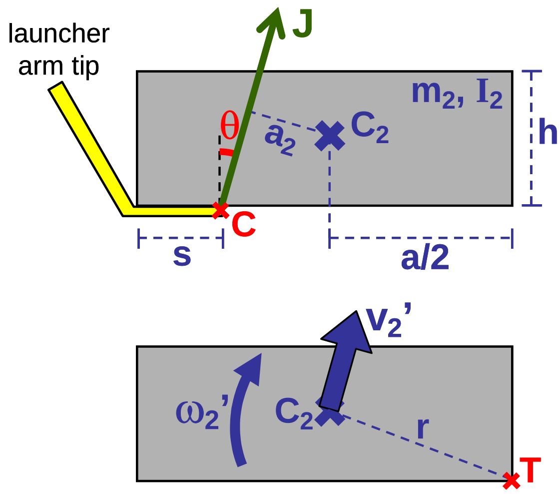
However, a short stroke cylinder does not necessarily mean it has a short overall length. For instance, a typical industrial 4" bore hydraulic cylinder, which can be adapted to pneumatic applications, has an overall retracted length of 9.625" plus the stroke length. Even if the cylinder stroke is only 1", its overall retracted and extended lengths 10.625" and 11.625" are relatively high. This short ratio between stroke and total length would not be effective in a three bar mechanism.

6.10.2. Launcher Equations

To properly simulate the launch, it is likely that you'll need some dynamic simulation software. Or you can use the spreadsheet from www.hassockshog.co.uk/flipper_calculator.htm, which has nice launcher models, as pointed out by Kevin Berry in the March 2009 edition of Servo Magazine. But you can also use a simple approximation to get a feeling of what happens during the launch.

Consider that the impulse J that the launcher inflicts on the opponent has an average direction θ with respect to the vertical, as pictured in the next page. The opponent is assumed as a homogeneous rectangular block with width a and height h. This value of a can be either the opponent's length or width, depending on whether you're launching it from its front/back or from its sides.

We also consider that the contact point between the launcher arm tip and the opponent does not shift during the launch, remaining fixed at a point C. This point is located by the distance s shown in the figure to the right, which is related to the length of the arm scoop.



This s value is also increased if the launcher is able to get under the opponent, as shown before in the action shots from the Ziggy vs. The Judge fight.

It is not difficult to calculate the distance a_2 between the impulse vector J and the opponent's center of mass C_2, and the distance r between C_2 and the edge T, they are obtained from

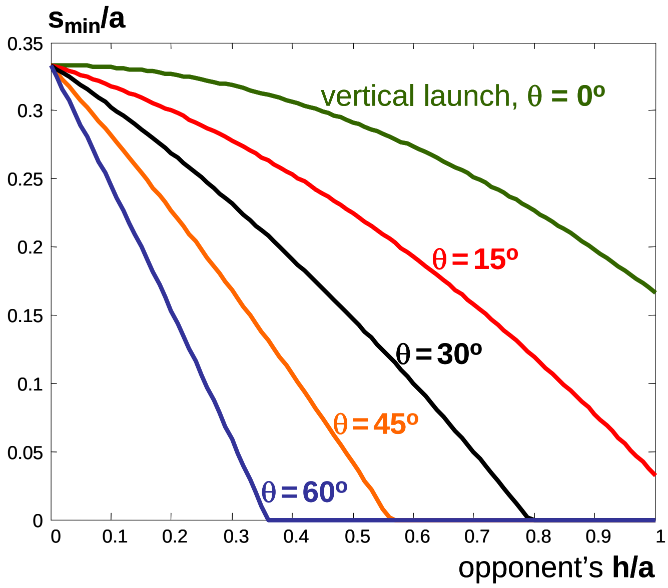
$$a_2 = (\frac{a}{2} - s) \cdot \cos\theta - \frac{h}{2} \cdot \sin\theta \quad \text{and} \quad r = \sqrt{(\frac{a}{2})^2 + (\frac{h}{2})^2}$$

If the opponent's mass is m_2, then its moment of inertia I_2, in the direction it spins due to the launch, can be estimated from the moment of inertia of rectangular bars, $I_2 = m_2 \cdot r^2/3$. If the point T in the figure does not touch the ground during the launch, and if the launching force is much higher than gravity, then we can estimate that the opponent's speed v_2', parallel to J, reaches J/m_2, and the angular speed results in $\omega_2' = J \cdot a_2/I_2$, leading to the relation $\omega_2' = v_2' \cdot a_2 \cdot m_2/I_2$ (see the figure above).

The vertical component of the speed at point T, equal to $v_2' \cdot \cos\theta - \omega_2' \cdot a/2$, must be positive to validate this analysis, making sure that this point does not touch the ground during the launch. So, from the previous equations, the point T does not touch the ground if s is greater or equal than a minimum value s_{min}, where

$$a_2 \leq \frac{2 \cdot I_2 \cdot \cos\theta}{m_2 \cdot a} \quad \Rightarrow \quad s \geq s_{min} = \frac{a}{3} - \frac{h^2}{6a} - \frac{h}{2} \cdot \tan\theta$$

This resulting relationship between s_{min}/a as a function of the opponent's aspect ratio h/a is plotted to the right. Note from the graph that, to launch an opponent without making it touch the ground, a "height launcher" ($\theta \cong 0^o$) would need a higher scoop length s than a "range launcher" (which usually has $\theta = 45^o$).



In theory, from the gravity potential energy equation, the maximum height an opponent would achieve would be, in theory, $H_{max} = E_b/(m_2 \cdot g)$, where E_b is the launching energy discussed before and g is the acceleration of gravity. But this height could only be reached if the opponent was vertically launched (thus $\theta = 0^o$) without spinning ($\omega_2' = 0$). If, ideally, the entire energy E_b is transformed into the opponent's kinetic energy (translational and rotational kinetic energies), we can show from the relation $\omega_2' = v_2' \cdot a_2 \cdot m_2 / I_2$, valid if T does not touch the ground, that

$$E_b = \frac{1}{2} m_2 \cdot v_2'^2 + \frac{1}{2} I_2 \cdot \omega_2'^2 = \frac{1}{2} m_2 \cdot \left(\frac{I_2 + m_2 a_2^2}{I_2} \right) \cdot v_2'^2, \quad \text{if } s \geq s_{min}$$

6.10.3. Height Launcher Equations

A "height launcher" with $\theta = 0^o$ would have $a_2 = a/2 - s$, while v_2' would be in the vertical direction. Since the maximum height H of a vertical launch is $v_2'^2/(2g)$, the opponent would reach

$$H = \frac{E_b}{m_2 \cdot g} \cdot \left(\frac{I_2}{I_2 + m_2 a_2^2} \right) = H_{max} \cdot \left(\frac{r^2}{r^2 + 3 \cdot (a/2 - s)^2} \right), \quad \text{if } s \geq s_{min}$$

The above expressions are only valid if $s \geq s_{min}$. But if $s < s_{min}$, the point T touches the ground, which makes its vertical speed $v_2' \cdot \cos\theta - \omega_2' \cdot a/2$ equal to zero at the end of the impulse, leading to the relation $\omega_2' = 2 \cdot v_2' \cdot \cos\theta / a$.

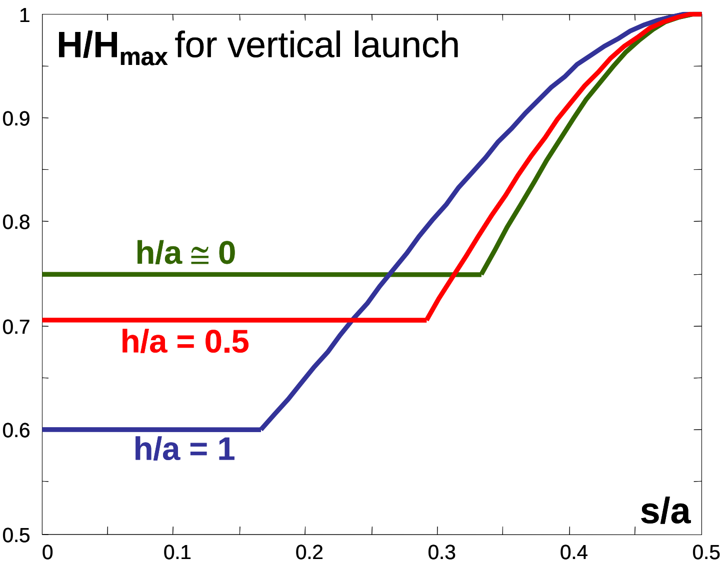
In this case, v_2' may not be parallel to J, its direction will depend on the contact friction between point T and the ground. A frictionless contact would keep the horizontal component of v_2' unchanged, while a very high friction could significantly reduce it, making the direction of v_2' becomes steeper than the direction of the impulse J.

A "height launcher" with $s < s_{min}$ would have then $\omega_2' = 2 \cdot v_2' / a$, leading to

$$v_2'^2 = \frac{2 \cdot E_b}{m_2} \cdot \left(\frac{m_2 a^2/4}{I_2 + m_2 a^2/4} \right) \Rightarrow H = \frac{v_2'^2}{2 \cdot g} = H_{max} \cdot \left(\frac{3}{4 + h^2/a^2} \right), \quad \text{if } s < s_{min}$$

Note that, for such small s, the launch height H only depends on the opponent aspect ratio h/a.

The graph to the right shows the H/H_{max} ratio as a function of the normalized length s/a, against opponents with aspect ratios h/a. Note that the horizontal lines are the values obtained for $s < s_{min}$, while the curved ones reflect the results from $s \geq s_{min}$.



The horizontal lines suggests that scooping the opponent with s/a = 0.15 results in the same height as if s/a was close to zero 0. This is true, at least for the simple model we're using, as long as the contact between the robots is maintained during the launch, which depends not only on s/a but also on the path direction and length y of the arm tip. Obviously, s/a = 0.15 would be better to maintain this contact than s/a close to zero.

So, from the graph we conclude that the most effective launch happens when s is close to a/2. One way to do that is to try to launch the opponent from the direction where it is shortest. So, for instance, against a narrow robot, try to launch it from its side, making the distance a become its width instead of length. Getting under the opponent is also one way to increase s, as done by Ziggy with its front wedge.

A long scoop at the end of the launcher arm (as seen in the middleweight Sub Zero, pictured to the right) also helps to increase the distance s. A long scoop will also make sure that the contact between the robots won't be lost during the entire stroke of the launcher arm. But be careful, a very long scoop will be vulnerable to drumbots and undercutters, which may bend it until it loses functionality, not being able to get under robots with low ground clearance. In addition, the previous graph showed that, against very low profile opponents (small h/a), there's no point in having a very long scoop to increase H unless it has s/a greater than 0.35 or if the weapon tip path y is too large. If below 0.35, the value of s/a would only have to be large enough not to lose contact with the opponent during the launch.

Note that the above calculations assumed that the launcher didn't tilt forward too much during the launch, which could make it get unstable. The requirements to guarantee launcher stability are presented in section 6.10.6.

6.10.4. Range Launcher Equations

A similar analysis can be made for a "range launcher" as follows. If the opponent does not touch the ground while it is launched, which happens when $s \geq s_{min}$, then the launch speed v_2' is parallel to the impulse J. The launch angle, with respect to the horizontal, is then $90^o - \theta$. The horizontal range R of the launch is then

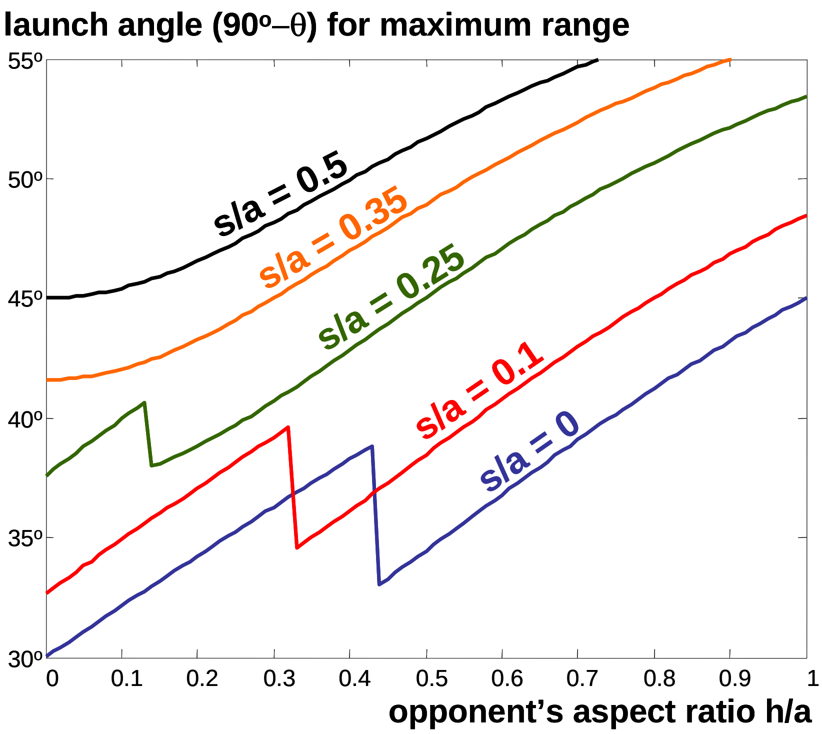
$$R = \frac{{v_2'}^2 \cdot \sin[2(90^o - \theta)]}{g} = \frac{2E_b \sin 2\theta}{m_2 \cdot g} \cdot \left(\frac{I_2}{I_2 + m_2 {a_2}^2} \right) = R_{max} \cdot \sin 2\theta \cdot \left(\frac{r^2}{r^2 + 3 \cdot {a_2}^2} \right), \quad \text{if } s \geq s_{min}$$

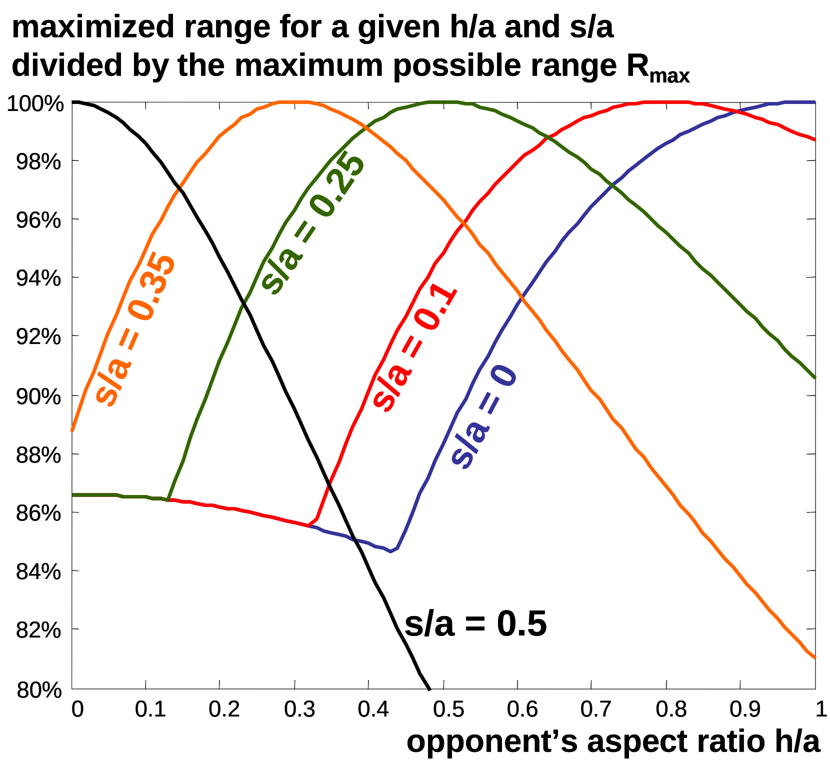
where $R_{max} = 2E_b/(m_2 \cdot g)$ is the maximum possible launch range, which only happens when $a_2 = 0$ (the impulse vector passes through C_2) and $\theta = 45^o$. The launch angle that maximizes R when a_2 is different than zero must be calculated with a numeric method, because a_2 depends on a, s, h and also θ.

On the other hand, if the opponent touches the ground at point T while it is launched (which happens when $s < s_{min}$), then the equations get much more complicated, because the ground reaction at T will cause a vertical impulse, and also a horizontal one if there's ground friction. This will change not only the magnitude of v_2' but also the launch angle. Even without considering ground friction, the equations are too lengthy to be shown here.

But the results are seen in the graphs to the right, obtained neglecting the effect of ground friction at point T during the launch. The neglected friction effect would only be significant if both θ and a_2 were large and if both s and h were very small, therefore it does not significantly influence the conclusions that are presented next.

The top graph shows the ideal launch angle (90°–θ) of the arm to maximize the range for a given s/a and aspect ratio h/a of the opponent. Against very low profile opponents (h/a close to zero), the best launch angle is 45° if you're able to reach s/a = 0.5, as expected, reaching 100% of the maximum possible range R_{max}, as seen on the bottom graph.





But if you can't get under such very low profile opponent and your arm has a very short scoop (s/a close to 0), then the best launch angle to maximize the reach would be 30° (with respect to the horizontal). This might seem strange but it makes sense: in this case, the ground impulse at point T, which happens due to the small s/a, will add up to the launcher's 30° impulse to effectively launch the opponent at 43.7°. As seen in the bottom graph, this best 30° angle will achieve 86.6% of R_{max}, which is still pretty good considering that s/a is so small. It would be impossible to reach 100% of R_{max} with s/a = 0. Unless the opponent was very tall, with h/a = 1, but then the best launch angle would be 45°.

Note that there is a step in the top graph for the curves with s/a < 0.33. This change (or step) is associated with the opponent touching the ground (values to the left of the step) or not touching it (values to the right of the step) during the launch. For instance, if s/a = 0, then an opponent with aspect ratio h/a = 0.4 (a value to the left of the step in the s/a = 0 curve) would be thrown further away if launched at 38^o, using the ground to help in its launch.

But if h/a = 0.5 (to the right of the step), then a shallower angle of 34^o would decrease the angular speed of the opponent, making it not touch the ground, resulting in the optimal throw in this case. So, depending on h/a and s/a, it may be good or not to use the help of the ground to launch the opponent with maximized range. This conclusion is not trivial at all.

Since most combat robots tend to have a low profile, with 0.2 < h/a < 0.4, to keep low their center of gravity, we can draw several conclusions about "range launchers" as follows. If the launcher can only provide a small s/a, then it is better to set its arm such that its impulse is in average at about $(90^o–\theta) = 36^o$ from the horizontal, to reach a maximized range of about 85% of R_{max} (in this case, the opponent would touch the ground during the launch, because $s < s_{min}$).

If the arm scoop allows the launcher to reach s/a = 0.25, then a steeper $(90^o–\theta) = 41^o$ angle would be a better option, typically launching the opponent between 91% and 99% of R_{max} (in this case, the opponent does not touch the ground because $s > s_{min}$).

Finally, if the launcher has a wedge to get under the opponent, or if you decide to use a very long scoop in its arm, then the best choice would be to use s/a = 0.35 and a $(90^o–\theta) = 45^o$ launch angle. With these values, you can launch typical opponents between 99% and 100% of R_{max}. This s/a and $(90^o–\theta)$ combination makes the distance a_2 become very close to zero for most robots, spending most of the launch energy throwing the opponent instead of making it spin.

Note, however, that these ideal launch angles to maximize range are only achievable in practice if they do not cause the launcher to be pushed too much backwards by the reaction force. The requirements to avoid this are presented in section 6.10.6.

Combinations of s/a, h/a and θ that lead to high values of a_2 should be avoided, because they waste too much energy making the opponent spin forward. Also, it is not a good idea to go beyond s/a = 0.35, you'll probably end up with a negative value of a_2, which will waste energy spinning the opponent backwards. The pictures in the next page show three different launch situations. The first one shows the super heavyweight Ziggy launching an opponent with a high average distance a_2 from C_2, resulting in a forward spin. In the second situation, Ziggy is able to launch The Judge with an impulse vector very close to C_2 (therefore a_2 close to zero), resulting in a high range due to the much lower resulting spin. Finally, the lightweight Rocket is able to launch the opponent with a backward spin, because the contact point was beyond the opponent's C_2, making a_2 become negative. Ideally, you should try to keep the opponent's spin as low as possible to maximize the launch range.

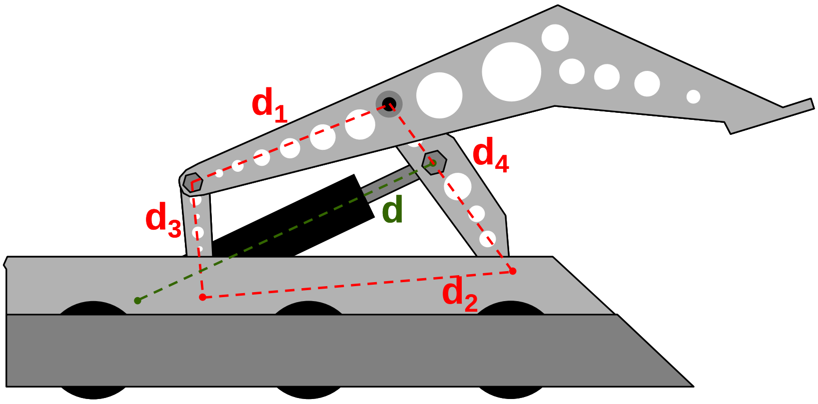
Launching with forward spin, due to the high a_2


High range launch because of the almost neutral spin, due to a_2 close to zero


Launching with backward spin, due to a negative a_2

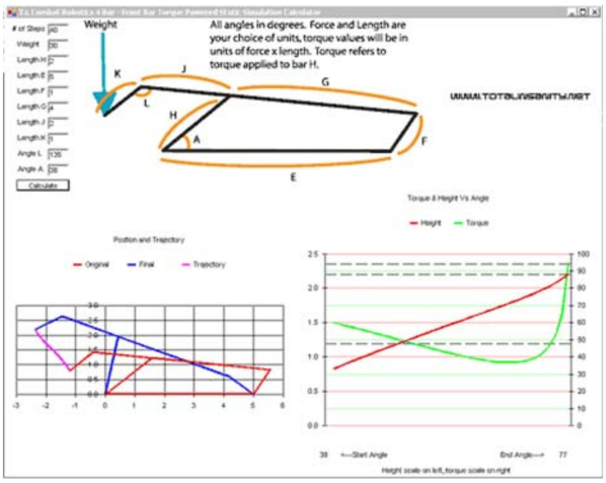
6.10.5. Four-Bar Mechanisms

As discussed before, the problem with most “range launchers” with three-bar mechanisms is that they usually end up with a tall chassis if they want to throw opponents at the ideal angles ($90^o-\theta$) from 36^o to 45^o. The tall chassis is a result of their mechanisms, which would need an average θ between 54^o and 45^o.



One alternative is to use four-bar mechanisms, as pictured to the right. The four bars consist of part of the launch arm (d_1), part of the chassis (d_2) and two auxiliary links (d_3 and d_4). Technically, these launchers have a five-bar mechanism, because the pneumatic cylinder (d) counts as a fifth link.

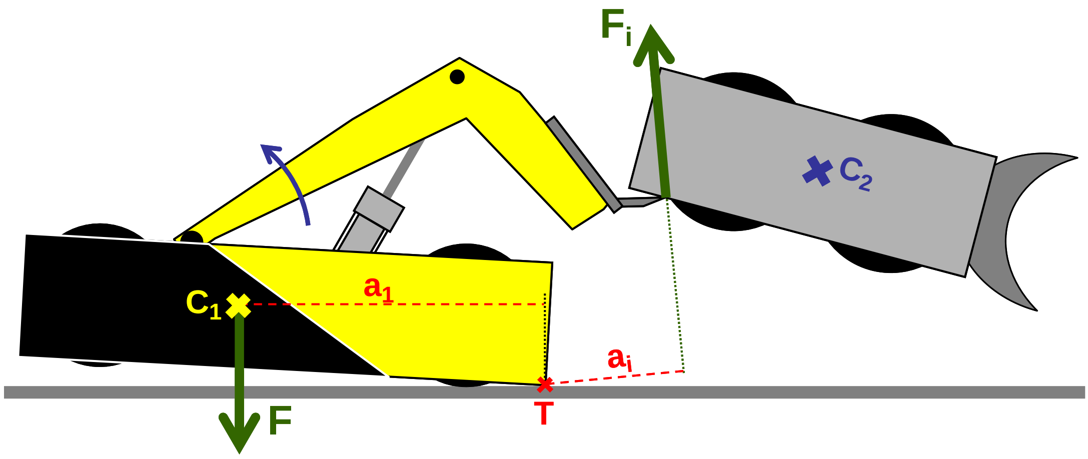
Four-bar mechanisms have two advantages. First, if well designed, they can be completely retracted inside the robot, allowing the use of a low profile chassis. And, if the constant lengths d_1, d_2, d_3 and d_4 are appropriately defined, it is possible to generate optimal trajectories for the arm tip. You can, for instance, make the arm tip trajectory become almost a straight line, with some desired optimal angle (which for “range launchers” would probably be between 36^o to 45^o with respect to the horizontal). The four-bar mechanism calculations are too lengthy to be shown here, but you can make them using, for instance, a free static simulation program (screenshot pictured to the right) that can be found on the tutorials in http://www.totalinsanity.net.



6.10.6. Launcher Stability

During the launch, a launcher should neither tilt forward too much, nor be pushed backwards, otherwise it will lose its effectiveness.

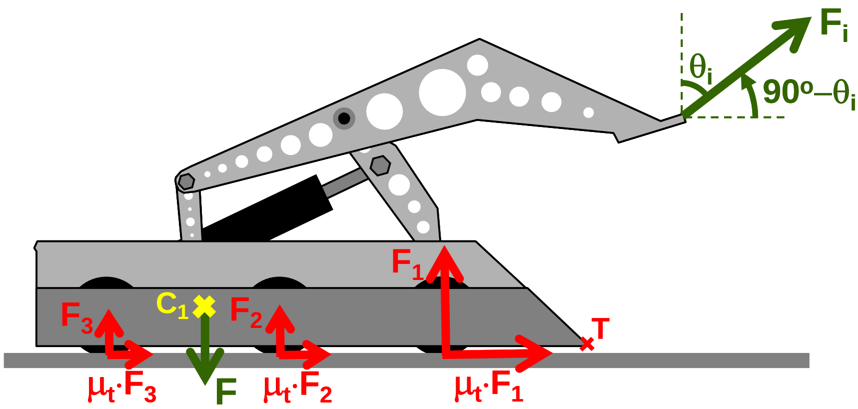
Due to the very high forces involved during the launch, it is very likely that “height launchers” will tilt forward until they touch the ground at their foremost point T, as shown in the figure in the next page as well as in the action shot to the right, featuring Sub Zero launching The Mortician. To avoid tilting forward even more and becoming unstable, it is necessary to locate the launcher’s center of gravity C_1 as far back as possible, to maximize the horizontal distance a_1 to point T, as pictured below. If F is the launcher weight, then the force F_i at any moment during the launch must satisfy $F_i \cdot a_i < F \cdot a_1$, where a_i is the distance between T and the line that contains the vector F_i. It is advisable that this condition is satisfied during the entire launch, for all values of F_i between F_0 and F_f, multiplied by their respective distances to point T.



Another concern with "height launchers" is with the stiffness and damping properties of their front wheels. During the launch, these wheels are very much compressed against the ground, storing a great deal of elastic energy. Towards the end of the launch, when the contact with the opponent is almost lost, these wheels will spring back. If their damping is low, as in foam-filled rubber wheels, they may launch the launcher and even make it flip backwards. The action shot to the right shows the launcher Sub Zero off the ground as soon as it loses contact with its opponent The Mortician.

"Range launchers" may tilt forward as well, but it is very unlikely that they lose stability in this way. This is because the line that contains the launch force vector F_i usually meets the ground within the launcher footprint, or very close to is foremost point T, due to the shallower launch angles ($90^o - \theta_i$) involved, see the picture to the right.

But "range launchers" have a problem with very shallow launch angles, because the horizontal component $F_i \cdot \cos(90^o - \theta_i)$ may become too large for the wheel friction to bear, pushing it backwards. As seen in the figure, if F is the launcher weight, F_1, F_2 and F_3 are normal ground forces on each wheel pair, and μ_t is the coefficient of friction between the tires and the ground that cause the maximum friction forces $\mu_t \cdot F_1$, $\mu_t \cdot F_2$ and $\mu_t \cdot F_3$, then



$$\left.\begin{array}{r} F_1 + F_2 + F_3 = F + F_i \cdot \sin(90^o - \theta_i) \\ \mu_t \cdot F_1 + \mu_t \cdot F_2 + \mu_t \cdot F_3 \geq F_i \cdot \cos(90^o - \theta_i) \end{array}\right\} \Rightarrow \mu_t \geq \frac{F_i \cdot \cos(90^o - \theta_i)}{F + F_i \cdot \sin(90^o - \theta_i)}$$

This equation shows that tire friction is very important to avoid “range launchers” from being pushed backwards, probably decreasing the contact time with the opponent and the effectiveness of the launch. A typical high traction tire with $\mu_t = 0.9$ is usually enough for a $(90^o - \theta_i) = 45^o$ launch angle. But for the 36^o or 41^o angles, which maximize the launch range for small s/a values, you might need a higher μ_t. If this higher μ_t is not achievable, then the best option is to adopt the lowest launch angle $(90^o - \theta_i)$ that satisfies the μ_t condition above.

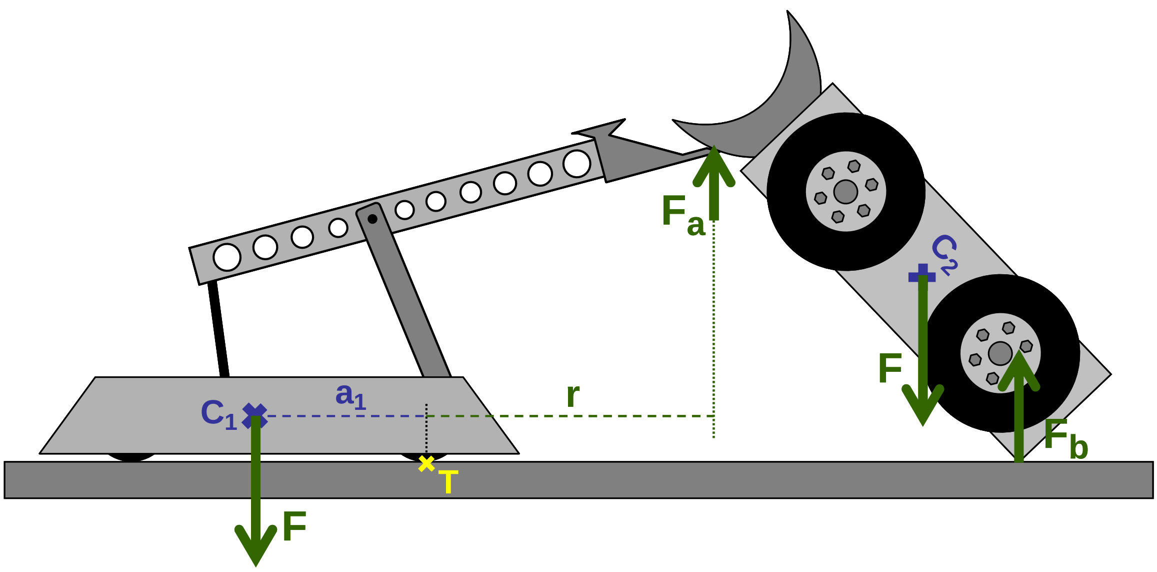
Finally, note that it is also a good idea for “range launchers” to locate C_1 as far back as possible, to prevent them from even touching their foremost point T on the ground, because this point will probably have a coefficient of friction with the ground lower than μ_t. If some tilting is inevitable, then it might be a good idea to install some anti-sliding material on the bottom of the robot beneath point T, such as a rubber strip, to increase friction.

6.11. Lifter Design

Lifters and launchers have a few common design features. The main difference is that lifters have relatively slow lifting mechanisms, allowing them to use, for instance, highly geared electric motors. The use of electric motors instead of pneumatic systems usually results in less weight to the weapon system, allowing them to improve their armor or drivetrain.

Lifters work better against non-invertible robots, or on arena with hazards where they can shove the opponent to. Clearly, they must have enough wheel traction to be able to push the opponent around the arena while it is lifted.

Similarly to launchers, a four-bar mechanism (as pictured below) is a good option to allow the lifting arm to retract inside the robot chassis, becoming less vulnerable to spinners.



To be able to lift their opponents without losing stability, it is important to locate their center of gravity C_1 as far back as possible, to maximize the distance a_1 to its foremost ground support (point T in the figure), usually located under the front wheels.

If F is the lifter weight, r is the maximum horizontal distance between point T and the tip of the lifter arm, and if F_a is the lifting force it applies on the opponent, then the gravity torque $F \cdot a_1$ with respect to point T must be higher than $F_a \cdot r$ to prevent it from tilting forward. The force F_a is a function of the weight of the opponent robot (also assumed as F, since both robots should be from the same weight class), of the force F_b from the opponent's ground support, and of the relative horizontal distances among them.

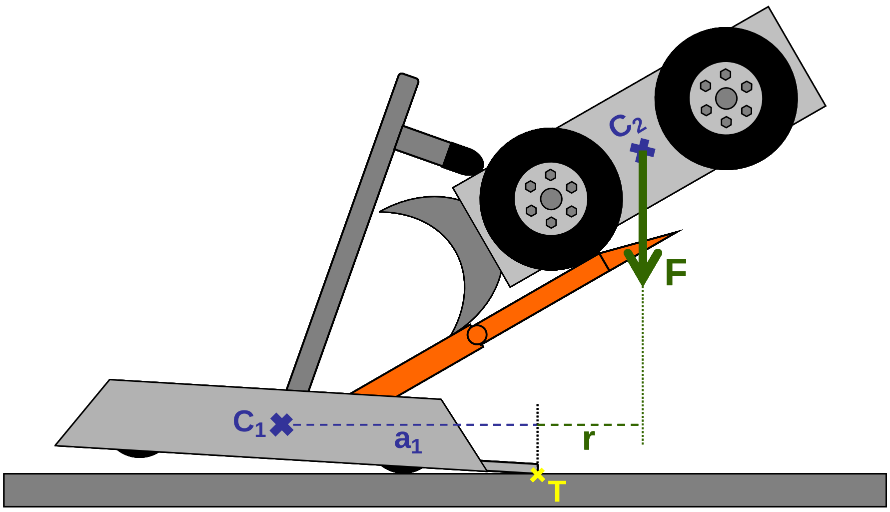
For a symmetric opponent, with a center of mass C_2 at the center of the chassis, it is easy to see that $F_a = F_b = F/2$ when the lifting begins, with the opponent in a horizontal position. As the opponent is lifted, F_b is increased while F_a decreases, as suggested by the picture. This is most noticed on tall opponents, which become easier to lift as they are lifted. So, the worst case scenario would be to consider that F_a is equal to its maximum value F/2, resulting in

$$F \cdot a_1 > \frac{F}{2} \cdot r \quad \Rightarrow \quad r < 2 \cdot a_1$$

It is not difficult to satisfy the above condition for the maximum horizontal reach r, so tilting forward is not a major concern for most lifters. Note that a lot of weight will be concentrated on the front wheels, up to 1.5 times the lifter weight in this example. So, make sure that the front wheels have high torque motors, to prevent them from stalling while pushing around the lifted opponent.

6.12. Clamper Design

Clampers are similar to lifters, except that they need to lift the entire weight F of the opponent, instead of just about half of it. To be able to clamp and lift their opponents without losing stability, they should also locate their center of mass C_1 as far back as possible. They usually need an extension on their front to act at point T as their foremost ground support, to increase the distance a_1 shown in the figure to the right.



If the lifter and opponent have same weight F, and r is the horizontal distance between point T and the opponent's center of mass C_2, then the gravity torque $F \cdot a_1$ with respect to point T must be higher than $F \cdot r$ to prevent the lifter from tilting forward, resulting in

$$F \cdot a_1 > F \cdot r \quad \Rightarrow \quad r < a_1$$

The above condition is much harder to meet than the stability condition for lifters, because here the distance r is not to the clamper tip, but to C_2. This distance to C_2, which is maximum when the opponent is starting to be lifted, can be very large for a long opponent. In addition, r must be smaller than a_1 for clampers, instead of $2 \cdot a_1$ as found for lifters. This is why tilting forward is a major concern for clampers, usually forcing them to use front extensions to increase a_1.

Similarly to lifters, clampers also need high torque on their front wheels to be able to drive around while carrying the opponent. Their front wheels have to bear up to twice the clamper weight, so make sure that their drive system is very sturdy and powerful.

6.13. Rammer Design

Rammers are usually nicknamed BMW, because they're basically made out of Batteries, Motors and Wheels. They must have a lot of traction to shove other robots around, and high top speeds to be effective as a ram.

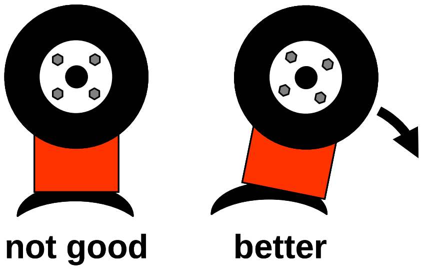
Its shield or armor is usually made out of hard materials, used in traditional armors. Ablative materials would also work, however they would have to be changed more often.

There are two design strategies to make them resistant to spinner attacks. Defensive rammers use shock mounts to attach their shield, trying to absorb and dissipate the energy of the attack. They can also use ablative shields for that.

Offensive rammers, on the other hand, have very hard shields rigidly attached to a stiff chassis, trying to divert the impact energy back to the attacker and break its weapon system. But remember to shock mount internal critical components.

Needless to say that taking the hit is not necessarily the best strategy: if you're able to push around a spinner without getting hit by its spinning weapon, it may self-destruct after hitting the arena wall.

Rammers should always be invertible. A few of them, such as the middleweight Ice Cube (pictured to the right), are even capable of righting themselves using only the power of their wheels, similarly to an overhead thwackbot.

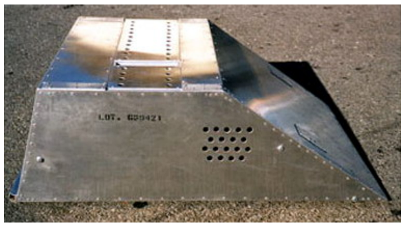


One issue with rammers with large shields is to avoid getting "stuck on their nose" (as pictured to the right). Ice Cube faces this problem, even though it is able to rock back and forth using the inertia of its wheels to flip back on its feet after a few seconds (we wish we had known that before the RoboGames 2006 semifinals!).

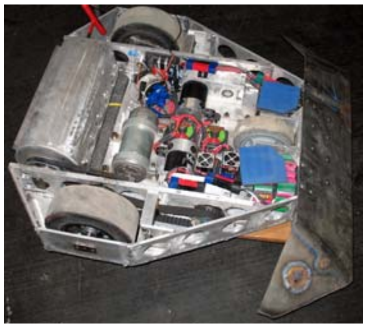
But, during the seconds it is rocking, the (orange) chassis gets exposed to attacks. Also, if the rammer is gently pushed while on its nose against the arena walls, it won't be able to get unstuck by rocking. One possible solution to avoid getting stuck is to mount the shield in a slightly asymmetrical position, as pictured above. With the ground projection of the rammer center of gravity closer to (or beyond) the edge of its shield, it becomes much easier to flip back.

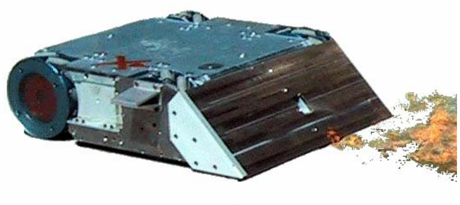
6.14. Wedge Design

Wedges have a very simple and effective design, especially against spinners. They try to use the opponent’s energy against it, with the aid of a ground support and an inclined plane (their wedge). The inclined plane idea is so simple and effective that it was used on one of the first combat wedges in history, Leonardo Da Vinci’s tank, pictured to the right. Its 35^o sloped conical armor, covered with steel armor plates, would work as a wedge to deflect enemy fire from all sides. It was not built back in 1495 because Italian battlefields were not flat enough to allow it to move without getting stuck. We had to wait 500 years, in 1995, to see a combat wedge on a flat “battlefield”, with La Machine (pictured to the right), coincidently with the same slope as Leonardo’s tank, taken from his original XV century drawing.



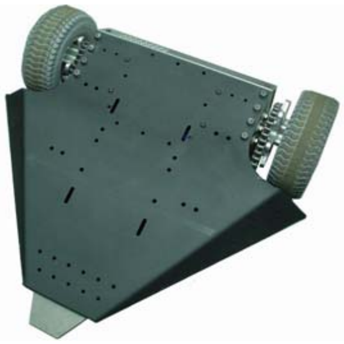
The wedge concept is not limited to wedges. As pictured below, it can be seen in vertical spinners (such as K2), horizontal spinners (Hazard), drumbots (Stewie), lifters (Biohazard), launchers (Ziggy), spearbots (Rammstein), hammerbots (The Judge), overhead thwackbots (Toe Crusher), and even combined with flamethrowers (Alcoholic Stepfather).



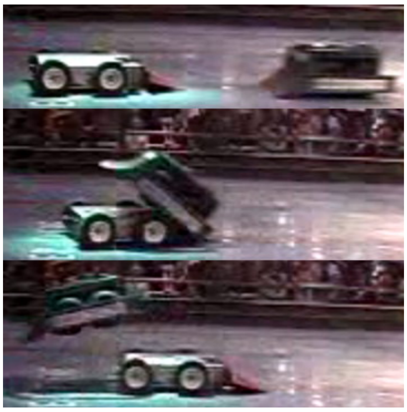


6.14.1. Wedge Types and Shapes

Fixed wedges, without any articulation, can either have some ground clearance, or they can be supported by the ground, scraping it. The first type might work well against most horizontal and vertical spinners, but it is vulnerable to undercutters or to a lower wedge.



The fixed wedges that scrape the floor, on the other hand, do not have this vulnerability, but they need to have a very sharp edge to stay flush to the ground. If they carry a significant portion of the robot's weight, they'll be much more difficult to get under due to the increased downward pressure. But, on the other hand, the robot might lose traction due to the decreased weight under the active wheels. Another way to increase the downward pressure is to decrease the ground contact area of the wedge, as done by the middleweight Emily with its narrow frontal titanium insert (pictured to the right).

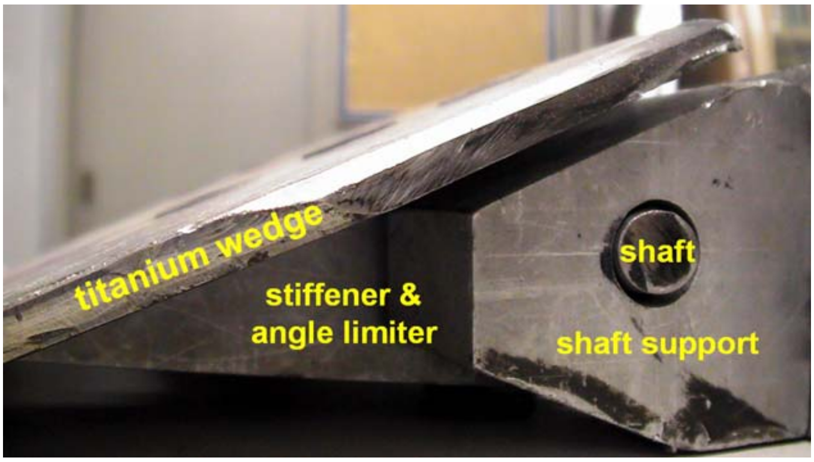


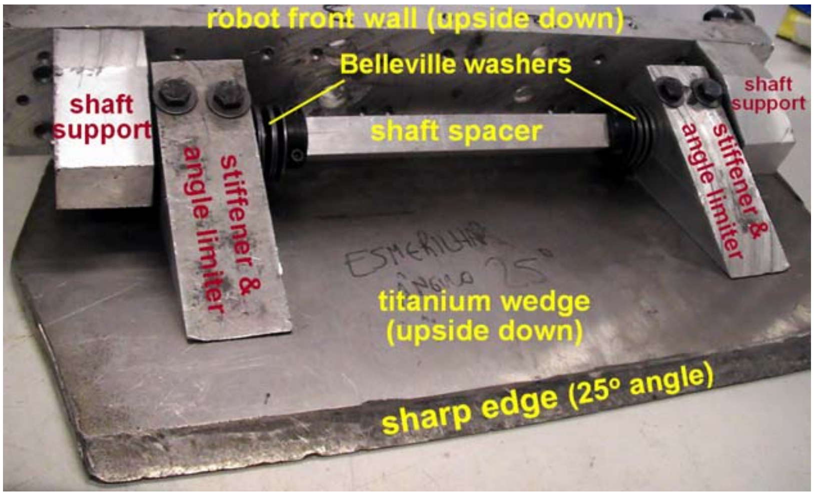
Articulated wedges are probably the most popular, resulting in a virtually zero ground clearance if their edge is sharpened. Floppy wedges, which have articulations that are not actively powered, should be heavy enough to increase the downward pressure at their sharp edge. On the other hand, active wedges, which have a powered articulation to make them work as lifters, can have their downward pressure increased by their motors just before they hit the opponent. This strategy has worked very well for the heavyweight lifter Sewer Snake, as pictured to the right.

Downward pressure can also be achieved by mounting springs to the robot's walls, keeping the wedges spring loaded flush to the ground. The picture to the right shows a nice wedge from the spinner Hazard. Note that, besides the spring, there are also two triangular-shaped supports underneath the wedge. These supports work as angle limiters, preventing the wedge from articulating too much and lifting the robot's own wheels off the ground, as well as working as stiffener brackets. Note also the rubber sandwich mounts used as dampers, improving the resistance to spinner impacts.

Wedge angle limiters are very important, especially if the robot has internal wheels. The picture to the right shows that, if the angle limiters from our hobbyweight Puminha are removed, its titanium wedge may get stuck and prevent the internal rear wheels from touching the ground, immobilizing the robot.

Both pictures to the right show a sturdy Ti-6Al-4V titanium articulated wedge used by our middleweight horizontal bar spinner Titan. Besides the high strength aircraft aluminum angle limiter, which also works as a stiffener, the wedge features Belleville washers on its titanium articulation shaft that work as shock mounts against lateral impacts.





Note that this 32° wedge has a shallower 25° sharp edge, to make sure that only the tip of the edge touches the ground, increasing the downward pressure. Note also that this wedge wouldn't be very effective if upside down, as shown in the bottom picture. This is why we only use it in non-invertible robots such as Titan.

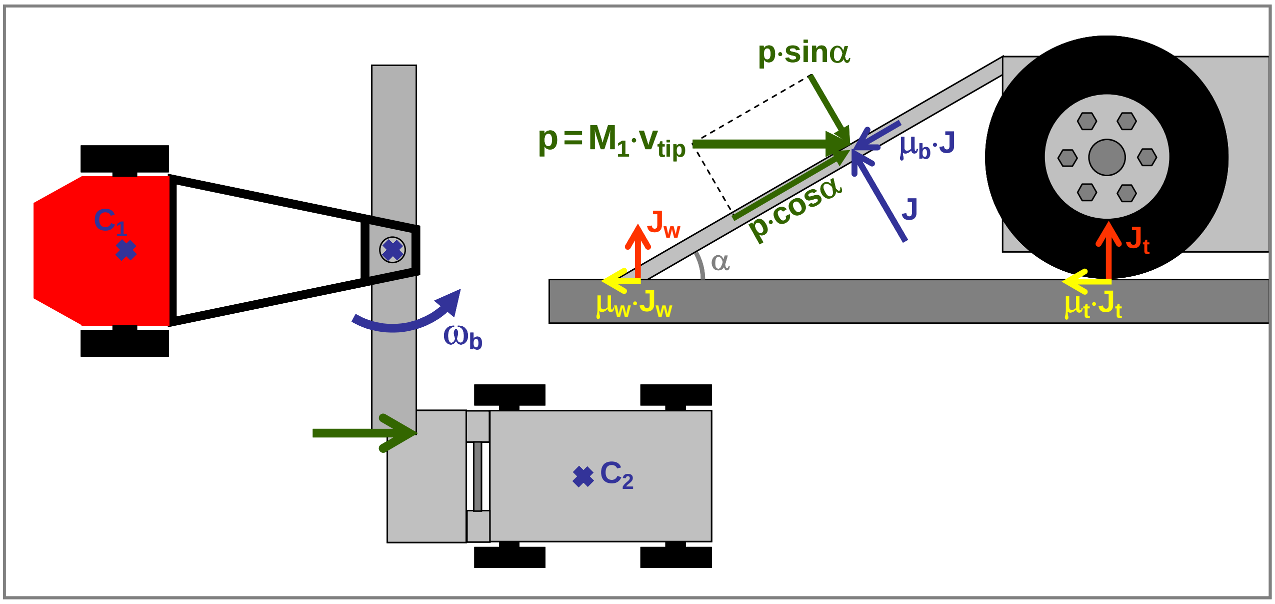
For invertible robots that don't have self-righting mechanisms, it is a good idea to have a symmetric wedge, which has the same effectiveness on either side. For instance, the heavyweight Original Sin has a hollow wedge made out of two rectangular plates separated by triangular spacers/stiffeners (pictured to the right), resulting in a symmetric wedge with a high stiffness to weight ratio.

But, unless your robot has external wheels such as Original Sin, or its wedge is actively powered, avoid using rectangular wedges, they can easily get your robot stuck resting on its side. Instead, use trapezoid-shaped wedges, either with a narrower edge (as pictured to the right) or with a wider edge. It is very unlikely that an internal-wheeled robot gets stuck resting on its side if it has a trapezoid-shaped wedge. A trapezoid-shaped wedge with a narrower edge will probably allow the robot to get unstuck and fall back, as long as there are angle limiters (which were removed before taking the picture to the right). A trapezoid wedge with wider edge will also work, but there's a greater chance that the robot will fall back upside down, which would be a problem if it does not have an invertible design.

Finally, wedges with blunt edges should be avoided, because they’re usually vulnerable to sharp anti-wedge skids, such as the S7 steel ones that support the drum of our hobbyweight Tourinho (pictured to the right). These narrow skids are able to concentrate a significant amount of downward pressure on the ground, in special if they're properly sharpened, easily getting under a blunt wedge.

6.14.2. Wedge Impact

One of the most important features of a wedge is its slope, represented by an angle α with respect to the horizontal. To find optimum α values, it is necessary to study the effect of an impact caused by a weapon that has an effective horizontal linear momentum $p = M_1 \cdot v_{tip}$, where M_1 is the effective mass of the attacker robot and v_{tip} is the speed of its weapon tip. Initially, let’s assume that the momentum p is perpendicular to the direction of the wedge's edge, as if facing a horizontal spinner or thwackbot at the side where the weapon tip approaches the wedge, as pictured below, or if frontally attacked by a spearbot.



The impact of the horizontal spinner (or thwackbot or spearbot) will cause a reaction impulse J normal to the wedge surface, in response to the $p \cdot \sin\alpha$ component, as shown in the figure. If the coefficient of restitution (COR) of the impact is e, and assuming that v_{tip} is much higher than the speed of the wedge (either before or after the impact), then $J \cong (1+e) \cdot p \cdot \sin\alpha$.

The wedge also responds with a friction impulse $\mu_b \cdot J$ parallel to its surface, decreasing the $p \cdot \cos\alpha$ component as the weapon slides during the impact, where μ_b is the friction coefficient between the weapon tip and the wedge. So, instead of $p \cdot \cos\alpha$, the wedge will only feel $\mu_b \cdot (1+e) \cdot p \cdot \sin\alpha$ parallel to its surface. Because of that, the wedge will effectively receive a horizontal impulse J_x that is smaller than p. It will also respond with a vertical impulse J_y that will try to launch the horizontal spinner, where

$$\begin{cases} J_x = J \cdot \sin\alpha + \mu_b \cdot J \cdot \cos\alpha = (1+e) \cdot p \cdot \sin\alpha \cdot (\sin\alpha + \mu_b \cdot \cos\alpha) \\ J_y = J \cdot \cos\alpha - \mu_b \cdot J \cdot \sin\alpha = (1+e) \cdot p \cdot \sin\alpha \cdot (\cos\alpha - \mu_b \cdot \sin\alpha) \end{cases}$$

As seen from these equations, it is very important that the wedge is very smooth, to decrease μ_b and thus minimize the backward impulse J_x, while maximizing the "launch impulse" J_y. And the wedge material must also be very hard to avoid dents, which could stick to the weapon tip and make the wedge suffer along its surface the entire component $p \cdot \cos\alpha$ instead of only $\mu_b \cdot (1+e) \cdot p \cdot \sin\alpha$.

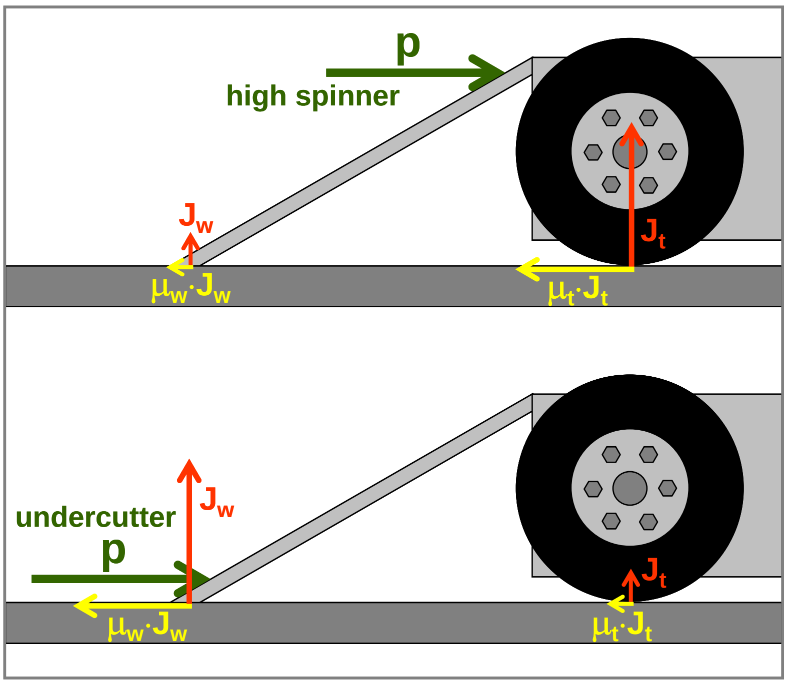
Hardened steels would be a good choice to avoid dents, however their stiffness-to-weight and toughness-to-weight ratios are not nearly as good as Ti-6Al-4V titanium, as seen in chapter 3. Since this grade 5 titanium is also relatively resistant to dents, due to its medium-high hardness, it is the material of choice for wedges. A hardened steel weapon tip would have $\mu_b \cong 0.3$ against a very smooth Ti-6Al-4V wedge, or up to $\mu_b \cong 0.5$ against a very rough and battle-battered Ti-6Al-4V wedge.

6.14.3. Defensive Wedges

A defensive wedge has the objective to resist the attack without being thrown backwards, even if it is standing still and not charging the attacker. This can happen if the horizontal impulse J_x that tries to push the wedge backwards is smaller than the vertical impulse J_y multiplied by the coefficient of friction μ with the ground, resulting in

$$J_x < \mu \cdot J_y \quad \Rightarrow \quad (\sin\alpha + \mu_b \cdot \cos\alpha) < \mu \cdot (\cos\alpha - \mu_b \cdot \sin\alpha) \quad \Rightarrow \quad \tan\alpha < \frac{\mu - \mu_b}{1 + \mu \cdot \mu_b}$$

Note that we've neglected above the effect of the robot weight on the friction impulse, because the impact is usually so fast that its forces are much higher than such weight.



If attacked by a "high spinner" as shown in the picture to the right, with the linear momentum p in the upper part of the wedge, then most of the reaction impulse will be provided by the front tires, resulting in $J_x \cong \mu_t \cdot J_t$ and $J_y \cong J_t$, where μ_t is the coefficient of friction between the front tires and the ground, and J_t is the vertical impulse from the ground to both tires altogether.

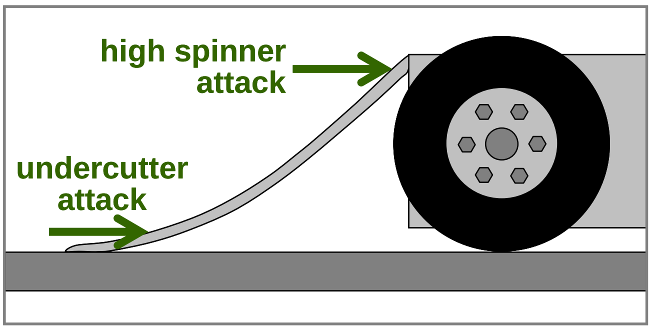
Assuming a typical high traction tire with $\mu = \mu_t = 0.9$, then a smooth titanium wedge with $\mu_b \cong 0.3$ would need to have about $\alpha < 25^o$ to be considered as a defensive wedge, while a rough wedge with $\mu_b \cong 0.5$ would only be defensive if its slope $\alpha < 15^o$. Note, however, that it is not a good idea to use a very small α, such as $\alpha < 20^o$, because it would lower too much the average thickness (and therefore the strength) of the sharp edge of the wedge.

On the other hand, against an undercutter, the linear momentum p is very close to the wedge's edge, making $J_x \cong \mu_w \cdot J_w$ and $J_y \cong J_w$, where μ_w is the coefficient of friction between the wedge and the ground, and J_w is the vertical impulse from the ground to the wedge's edge, as seen in the figure. Assuming a coefficient of friction $\mu = \mu_w = 0.35$ between the titanium edge and the soft steel arena floor, the equation shows that any $\alpha > 3^o$ would allow the robot to be thrown backwards, even if it had a very smooth wedge.

So, in theory, no wedge can be considered defensive against an undercutter: the wedge robot can defend itself from the first attack, but it will be thrown backwards or get spun, making it vulnerable to an immediate second attack. It is up to the wedge driver to keep facing the undercutter at all times.

But it is possible to have a defensive wedge against undercutters. Choosing a very small α is not a good idea, because the resulting low average thickness of the edge would allow the wedge to be easily torn apart by the undercutter, which would hit it at its weakest spot. Perhaps a good idea would be to spread some anti-slip product, such as anti-slip v-belt spray, under the edge where the wedge touches the ground, significantly increasing μ_w. Clearly, the spray should only be applied before matches against undercutters.

Another idea is to use a wedge with variable slope, similar to a scoop, with a lower α near the edge and a higher α near the top, to be effective against both undercutters and high spinners, as pictured below. Due to the variable slope, it would be possible to have an edge with a low α without compromising is thickness and strength.



A defensive wedge may be a good option if the robot also has some active weapon. The wedge could be used to defend the robot from opponents' attacks, or to slow down their spinning weapon, until the active weapon had a chance to get in action against them.

But a defensive wedge by itself would have a hard time winning a fight by knockout. You would need then an offensive wedge.

6.14.4. Offensive Wedges

Offensive wedges have the objective to launch their opponents, as pictured to the right. To most effectively accomplish that, they need to maximize the vertical impulse J_y, which happens for some $\alpha = \alpha_{launch}$ the makes the derivative $dJ_y/d\alpha = 0$, resulting in

$$\frac{dJ_y}{d\alpha} = (1+e)\cdot p \cdot \frac{d}{d\alpha}[\sin\alpha \cdot (\cos\alpha - \mu_b \cdot \sin\alpha)] = 0 \;\Rightarrow \tan(2\cdot\alpha_{launch}) = \frac{1}{\mu_b}$$

For this optimum angle $\alpha = \alpha_{launch}$, the horizontal impulse given by $J_x = J_{x,launch}$ and the maximized vertical impulse $J_y = J_{y,launch}$ would result in

$$J_{x,launch} = \frac{(1+e)\cdot p}{2} \quad \text{and} \quad J_{y,launch} = \frac{(1+e)\cdot p}{2}\cdot[\sqrt{1+\mu_b^2} - \mu_b]$$

For a very smooth titanium wedge with $\mu_b \cong 0.3$, the optimum angle to launch the opponent is $\alpha_{launch} \cong 37^o$, resulting in a maximum $J_{y,launch} = 0.37\cdot(1+e)\cdot p$, while a battle-battered titanium wedge with $\mu_b \cong 0.5$ would have $\alpha_{launch} \cong 32^o$ and $J_{y,launch} = 0.31\cdot(1+e)\cdot p$. Curiously, a hard steel wedge against a hard steel weapon would have $\mu_b \cong 0.4$ and therefore $\alpha_{launch} \cong 34^o$, very close to the slope angle from Leonardo's steel-plated tank. This might not be a coincidence: Leonardo Da Vinci was known for performing simple experiments in several areas before proposing a new design.

From the calculations for defensive wedges, a wedge robot with mass m_2 and an optimum angle $\alpha = \alpha_{launch}$, with initial speed equal to zero, would be thrown backwards with a speed v_2' such that

$$J_{x,launch} - \mu \cdot J_{y,launch} = m_2 v_2' \quad \Rightarrow \quad v_2' = \frac{(1+e)\cdot p}{2\cdot m_2}\cdot[1 + \mu\cdot\mu_b - \mu\cdot\sqrt{1+\mu_b^2}]$$

where $\mu = \mu_t$ against high spinners, and $\mu = \mu_w$ against undercutters, as defined before. So, to avoid being thrown backwards, the offensive wedge just needs to charge with a speed of at least v_2' before the impact.

6.14.5. Example: Offensive Wedge vs. Horizontal Spinner

In the calculation example for Last Rites against Sir Loin, the effective linear momentum before the impact was $p = M_1\cdot v_{tip} = 6.21kg\cdot 106.4m/s = 661Ns$. If Sir Loin had a smooth 37^o sloped titanium wedge aligned with J, with $\mu_b = 0.3$, and if the impact had as well a COR $e \cong 0.13$ with the same effective mass $M_1 = 6.21kg$ calculated before (which is not necessarily true, since the calculated M_1 and the measured e were not obtained for an angled impact), then it would only have to take a horizontal impulse $J_x = J_{x,launch} = (1+0.13)\cdot 661/2 = 373Ns$, while the arena floor would provide the vertical reaction impulse $J_y = J_{y,launch} = 0.37\cdot(1+0.13)\cdot 661 = 276Ns$ to launch Last Rites.

Assuming Last Rites has a mass of 220lb (99.8kg), then this $J_{y,launch}$ would launch its center of mass with a speed $v = 276Ns/99.8kg = 2.77m/s$, reaching a height of $v^2/(2g) = (2.77)^2/(2\cdot 9.81) = 0.39m$ (about 15 inches). In addition, this vertical impulse $J_{y,launch}$ would also make Last Rites roll, with an angular speed that would depend on its moment of inertia in the roll direction and on the gyroscopic effect of the weapon. This roll movement could make its spinning bar touch the ground, probably launching it even higher than that.

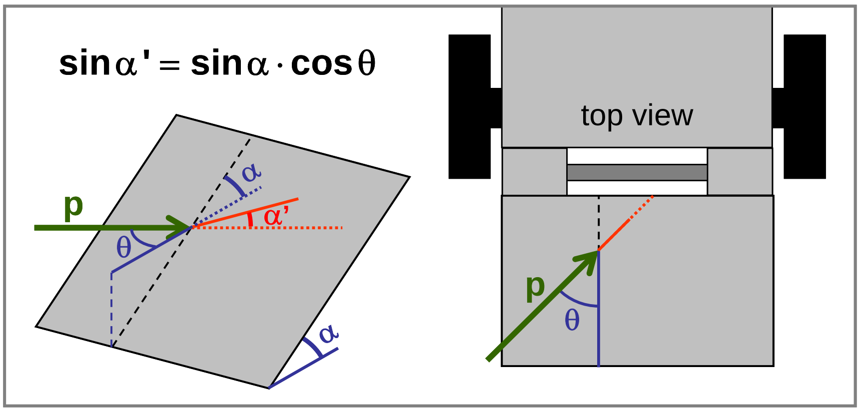
If the wedge version of Sir Loin had almost no speed before the impact, it would be thrown backwards with a speed v_2', calculated next. If the spinning bar from Last Rites was very low to the ground, then assuming $\mu = \mu_w = 0.35$ and m_2 = 220lb (99.8kg) we would get

$$v_2' = \frac{(1+0.13)\cdot 661}{2\cdot 99.8}\cdot[1+0.35\cdot 0.3-0.35\cdot\sqrt{1+0.3^2}] = 2.77\text{m/s}$$

So, to avoid being thrown backwards, the wedge Sir Loin would only need to charge before the impact with a forward speed of at least 2.77m/s (10km/h or 6.2mph), which is not a big deal for most combots. On the other hand, if the bar was spinning higher off the ground (which happens when Last Rites is flipped upside down), hitting near the top of the wedge, then $\mu = \mu_t = 0.9$ would lower v_2' to 1.24m/s (4.5km/h or 2.8mph), which is even easier to reach by Sir Loin.

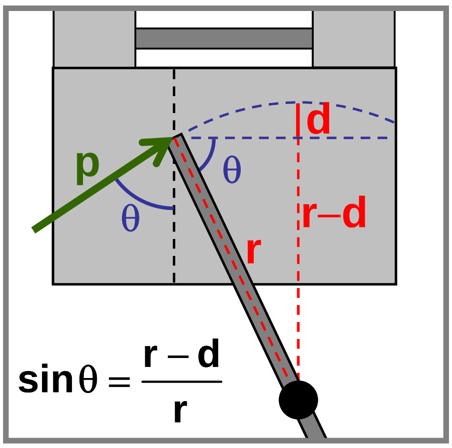
6.14.6. Angled Impacts

The previous equations assumed that the impact direction was perpendicular to the direction of the wedge's edge, meaning $\theta = 0^o$ in the figure to the right.



For angled impacts, with θ between 0^o and 90^o, the wedge works as if it had an effective slope angle α' smaller than α, where $\sin\alpha' = \sin\alpha\cdot\cos\theta$. All previous equations would remain valid, as long as α is replaced by this effective α'.

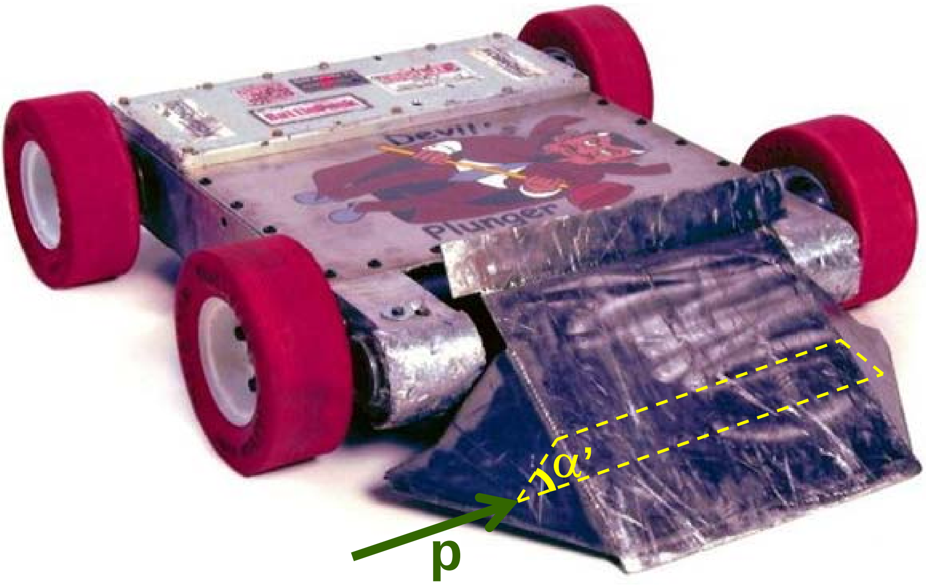
For instance, in the figure below to the left, the wedge from Pirinah 3 is not able to launch the horizontal bar spinner The Mortician, because an aligned frontal attack with a low forward speed has $\theta \cong 90^o$, resulting in $\alpha' \cong 0^o$ and therefore $J_y \cong 0$ and $J_x \cong 0$. As long as the wedge is sufficiently smooth, without dents or bolt heads sticking out, it will work as a defensive wedge if properly aligned to the attacking robot.



But if the robots are moving forward at high speeds, then the tooth travel distance d (related to a tooth bite $d \cdot \sin\alpha$) can be large enough to significantly lower the attack angle θ, because of the relation $\sin\theta = (r-d)/r$, where r is the radius of the spinning weapon, as shown in the previous picture to the right. This lower attack angle θ can then increase α', which can be enough to launch the spinner. So, for an aligned frontal attack of an offensive wedge, forward speed is fundamental.

Another approach is to try to hit the spinner with an offset to the side where the spinning weapon approaches the wedge, as it was assumed in the previous examples, trying to make θ close to zero. But, against a skillful driver, it might be difficult to make the wedge hit the spinner with such offset. The spinner will probably be trying to face the wedge robot at all times.

This is one of the reasons why several wedges have angled side edges, such in the floppy wedge from the middleweight Devil's Plunger (pictured below). Its wedge has angled sides usually fabricated through bending or welding. These angled sides can turn even a perfectly aligned hit at a low forward speed, which would have $\theta \cong 90^o$ leading to $\alpha' \cong 0^o$, into a spinner-launching hit. The effective slope angle α' is a design parameter for the side edges, measured on a vertical section of the wedge as shown in the picture. In addition, these side edges protect the wedge from being knocked off due to a side hit.



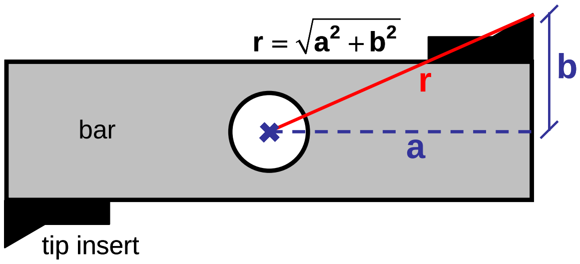
A suggested value for the side angle α' to launch spinners would be, for a smooth titanium wedge, equal to $\alpha' = \alpha_{launch} = 37^o$. And, for a hardened steel wedge, which has a larger coefficient of friction than smooth titanium, a suggested side angle α' to launch spinners would be Leonardo Da Vinci's $\alpha_{launch} = 34^o$.

6.14.7. Wedge Design Against Vertical Spinners

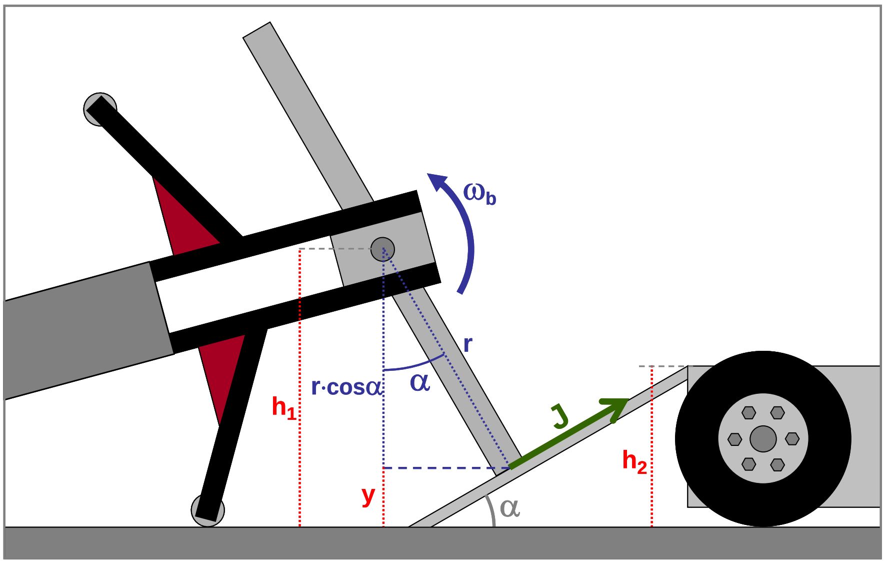
One final concern regarding wedge design is to make sure it will be effective against vertical spinners.

As seen in the picture in the next page, a vertical spinner with weapon radius r and weapon shaft height h_1 will most likely hit a wedge with height h_2 and slope angle α at a height $y = h_1 - r \cdot \cos\alpha$. Clearly, the height h_2 must be larger than y, so α cannot be too high to make sure that

$$y = h_1 - r \cdot \cos\alpha < h_2 \quad \Rightarrow \quad \cos\alpha > \frac{h_1 - h_2}{r}$$

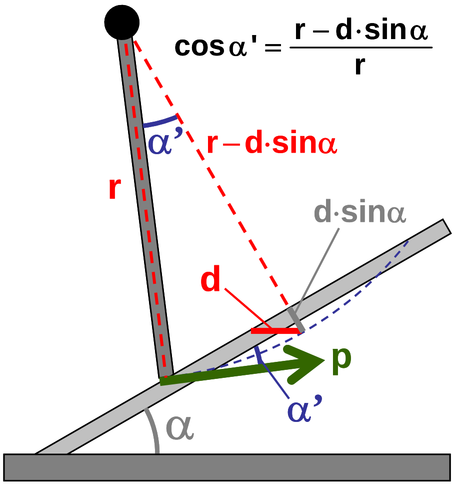


Note that if the spinning weapon has large tip inserts, or if it is a very wide bar, then the radius r used in the above equations must be calculated as in the picture to the right, considering the bar width and tip insert dimensions in the value of b, as well as the bar length 2·a.



But even if the above condition for α is satisfied, there is no guarantee that the vertical spinner won't bounce off and end up hitting the top plate of the wedge robot. This situation is very likely to happen when low profile wedges charge forward at high speeds against tall vertical spinners. To avoid that, the top of the wedge should have an overhanging section, such as the small one on Devil's Plunger or the large one on Pipe Wench, pictured below. A large overhung section is very effective against vertical spinners, working as a scoop, as shown in the action shots below featuring Pipe Wench vs. Terminal Velocity.

Note from the action shots that the wedge should be charging forward towards the vertical spinner to most effectively launch it and eventually flip it. This is because the vertical spinner weapon always hits tangentially to the wedge surface. If the robots are not moving forward and the wedge is sufficiently smooth, you should only see sparks and both robots repelling each other, but no major hits.



But if the robots are moving forward at high speeds, then the tooth travel distance d will result in a tooth bite d·sinα, due to the slope of the wedge. This tooth bite will result in an effective angle α' ≠ 0 between the speed of the weapon tip and the wedge surface, as pictured to the right, where r is the radius of the spinning weapon and $\cos\alpha' = (r - d \cdot \sin\alpha)/r$. So, the higher the forward charge speed, the higher will be the tooth travel distance d and the angle α', resulting in a higher impact that might launch the vertical spinner.

6.15. Gyroscopic Effect

An interesting characteristic of robots with spinning weapons is their gyroscopic effect. Vertical spinners and drums tend to lift off their sides (tilting) when making sharp turns with their weapon turned on. The picture to the right shows our drumbot *Touro*, which is able to make turns with only one wheel touching the ground. This wheel lift-off, besides impressing well judges and audience, works as an excellent "victory dance" at the end of a match. However, if the robot's wheels lift too much off the ground it can be a disadvantage, because you'll have a hard time making turns, risking being flipped over. But what causes the gyroscopic effect?

The gyroscopic effect comes from the fact that bodies tend to remain in their state of motion, as stated by Newton's first law. In this case, they tend to maintain their angular momentum. When *Touro* tries to turn with the weapon turned on, it is forcing the drum to change its spinning orientation, making it harder to maneuver.

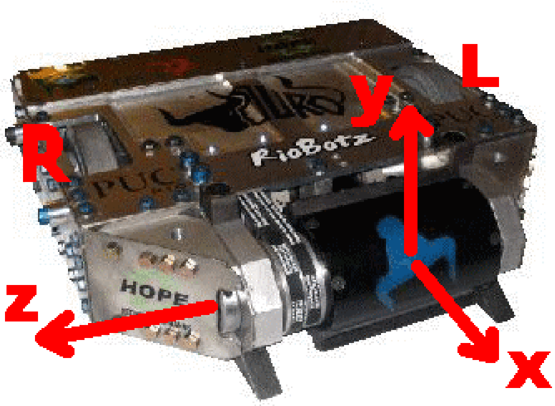
Horizontal spinners don't have this problem, because when turning they don't change the orientation of the weapon axis, which remains vertical. However, spinners have a small difficulty when turning in the opposite direction of the one that its weapon spins, and they turn more easily in the same direction. This doesn't have anything to do with the gyroscopic effect, it is simply caused by the friction between the weapon and the robot structure, which tends to turn the robot in the same sense of the weapon. This effect is usually small. The gyroscopic effect in the horizontal spinners appears when an opponent tries to flip them, because the high speed of the spinning weapon provides them with a certain stability that helps them to remain horizontal. Our spinner *Ciclone* escaped from several potential flips because its weapon was turned on.

There are spinners that have weapon spinning axis that are not vertical, such as the robot Afterthought, pictured to the right. They have a small slope forward, intended to hit lower opponents because the weapon tip gets closer to the ground. Those tilted spinners suffer a little from the gyroscopic effect, which is proportional to the sine of the slope angle between the weapon spinning axis and the vertical. The smaller the angle, the smaller the effect.

This tilted spinner type has a serious problem: there is a chance that the robot gets flipped over during its own attack. If for instance the weapon spins in the sense shown in the picture above, it won't have problems when its opponent is right in front at its left side (the "good" side shown in the figure, which hits like an uppercut). But if the opponent is on its right side (the "bad" side), then the tilted spinner may be flipped over when hitting from top to bottom.

The gyroscopic effect, besides making it harder for vertical spinners and drumbots to make turns, also causes the phenomenon called precession, the same that explains why a spinning top can have its spin axis sloped without falling over. This phenomenon explains the wheel lift off (tilting) during sharp turns. The precession of a robot's weapon can be calculated from the principle of angular momentum (which results in what is known as Euler's equations), which depends on the rotational moment of inertia of the weapon in the spin direction (horizontal, in this case) I_{zz}, and on the moments of inertia in the 2 other directions, I_{xx} and I_{yy}. The weapons of those robots usually have axial symmetry (as in the disks of the vertical spinners or the cylinders of the drums), and therefore $I_{xx} = I_{yy}$.

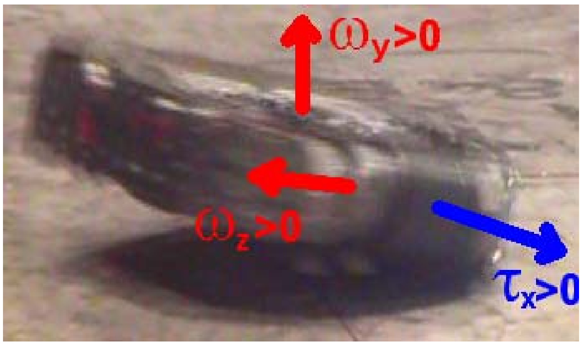
In the figure to the right, the z-axis was chosen in the direction that the weapon spins, with angular speed ω_z, from ground up in such a way to throw the opponents into the air. Notice that the robot's right and left wheels are represented by the letters R and L respectively. The y-axis was chosen as the vertical one, and the x-axis is directed horizontally towards the front of the robot. When the robot is turning around its y-axis with angular speed ω_y and with its weapon turned on spinning with ω_z, the principle of the angular momentum results in

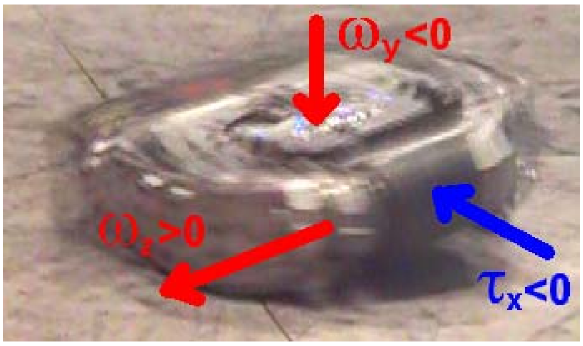


$$\tau_x \cong I_{zz} \cdot \omega_z \cdot \omega_y$$

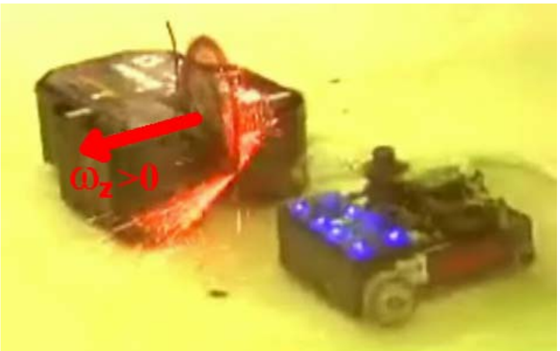
where τ_x is an external torque applied in the direction of x. This equation is a good approximation if the tilting angle of the robot is small.

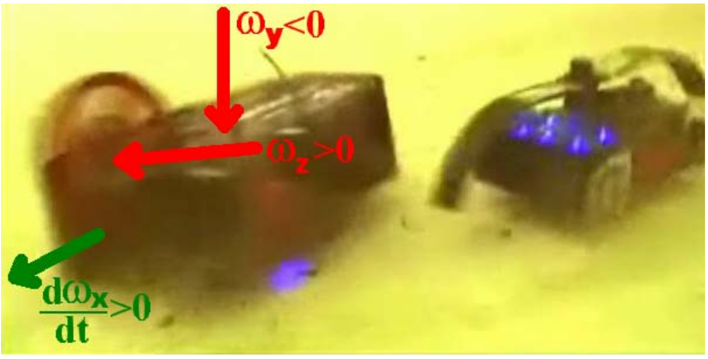
If the robot turns left, then $\omega_y > 0$, and therefore from the above equation we get $\tau_x > 0$. This means that the gravity force needs to generate a positive torque in the x direction to keep the system in balance, which happens when the right wheel (R in the figure above) lifts off the ground, see the left figure below. Similarly, if the robot turns right, then $\omega_y < 0$ and therefore $\tau_x < 0$. This negative torque that the gravity force needs to generate is obtained when the left wheel (L in the figure above) lifts off the ground, see the right figure below.





Those results are the same for drumbots as well as for vertical spinners. For instance, during the final match of the RoboCore Winter Challenge 2005 competition, the middleweight vertical spinner *Vingador* was spinning its disk with $\omega_z > 0$. After receiving an impact from our horizontal bar spinner *Ciclone* (left figure below), *Vingador* began to twirl with a clockwise speed $\omega_y < 0$. To balance this movement, a negative torque $\tau_x < 0$ would be necessary. However, even lifting its left wheel, the gravity force wasn't able to generate enough torque. *Vingador* continued tilting (with an angular acceleration $d\omega_x/dt > 0$, see the right figure below) until it ended up capsizing over its right side.

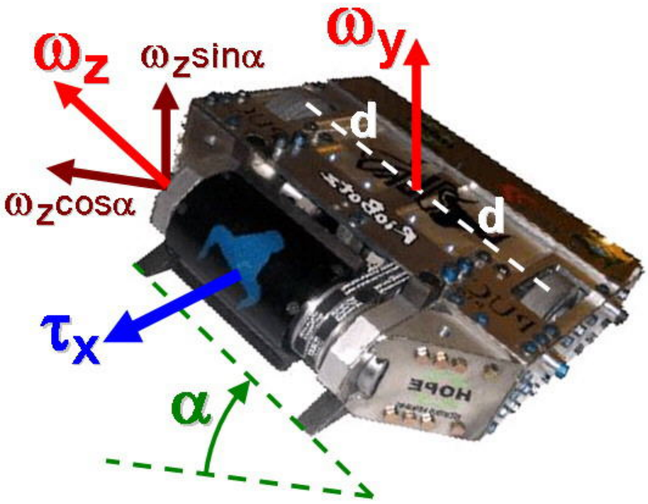




Vertical spinners have more problems with the gyroscopic effect than drumbots. The reason for that is because the gyroscopic effect is proportional to the angular speed of the weapon ω_z, while the kinetic energy depends on ω_z^2. Vertical spinners usually have lower weapon speed ω_z and higher moment of inertia I_{zz} than drums.

Therefore, for instance, a vertical spinner that spins a solid disk of mass m_b = 10kg (22lb) and radius r = 0.3m (almost 1ft) at ω_z = 1,000RPM = 105rad/s has a weapon with moment of inertia $I_{zz} = m_b \cdot r^2/2 = 10 \cdot 0.3^2/2 = 0.45 kg \cdot m^2$ and with kinetic energy equal to $E = I_{zz} \cdot \omega_z^2 / 2 = 0.45 \cdot 105^2 / 2$ = 2,481J. A drumbot that spins a cylinder with the same mass m_b = 10kg (22lb) and external and internal radii r = 0.06m and r_i = 0.04m at ω_z = 4,173RPM = 437rad/s has $I_{zz} = m_b \cdot (r^2 + r_i^2)/2 = 10 \cdot (0.06^2 + 0.04^2)/2 = 0.026 kg \cdot m^2$, and the kinetic energy of the weapon is $E = I_{zz} \cdot \omega_z^2 / 2 = 0.026 \cdot 437^2 / 2$ = 2,483J, practically the same energy of the vertical spinner. Therefore, both robots have similar destruction power.

However, the angular momentum of the vertical spinner weapon is $I_{zz} \cdot \omega_z = 0.45 \cdot 105 = 47.25$, much larger than the one from the drum, $I_{zz} \cdot \omega_z = 0.026 \cdot 437 = 11.36$. Because the gyroscopic effect depends on the product $I_{zz} \cdot \omega_z$, a drumbot usually tilts much less than a vertical spinner while making turns.



It is possible to get a better estimate of the gyroscopic effect, explicitly considering the tilt angle α with respect to the horizontal (as pictured to the right), which had been assumed to be very small in the previous calculations. As the robot turns with speed ω_y and with its weapon spinning with ω_z, the robot tilts by the angle α. The projection of the vector ω_z onto the vertical, $\omega_z \cdot \sin\alpha$, doesn't change direction, but the horizontal projection $\omega_z \cdot \cos\alpha$ does, rotating around the y-axis with speed ω_y, which is responsible for the gyroscopic effect.

The gravity torque τ_x is equal to $m \cdot g \cdot d \cdot \cos\alpha$, where m is the mass of the entire robot, g is the acceleration of gravity, and d is the distance between each wheel and the robot's center of mass, as pictured above. Assuming that ω_z is much larger than ω_y (because the weapon spins much faster than the robot turns), the principle of angular momentum states that the tilting movement is in equilibrium if $\omega_y = \omega_{y,critical}$, where

$$\tau_x = m \cdot g \cdot d \cdot \cos\alpha = I_{zz} \cdot (\omega_z \cos\alpha) \cdot \omega_{y,critical}$$

Canceling the $\cos\alpha$ term from the equation, we get

$$\omega_{y,critical} = \frac{m \cdot g \cdot d}{I_{zz} \cdot \omega_z}$$

In other words, if you turn with speed ω_y equal to $\omega_{y,critical}$, the robot will tilt with an arbitrary angle α (the robot stability does not depend on α, at least in the considered model approximation). If ω_y is smaller than $\omega_{y,critical}$, the robot doesn't lift any wheel off the ground. And if ω_y gets larger than $\omega_{y,critical}$, the robot tilting will become unstable, capsizing over its side. At the final match of the RoboCore Winter Challenge 2005, *Ciclone*'s impact made the robot *Vingador* twirl with a speed ω_y larger than its critical value $\omega_{y,critical}$, capsizing it.

In the previous example, which compared a vertical spinner and a drumbot with same weapon system energy, assuming that m = 55kg (about 120lb) and d = 0.2m (7.9"), the above equation would result in the conclusion that the vertical spinner would not be able to make turns faster than $\omega_{y,critical}$ = 2.28rad/s = 21.8RPM without lifting its wheel and risking capsizing. On the other hand, the considered drumbot would be able to turn even at $\omega_{y,critical}$ = 9.5rad/s = 90.7RPM without lifting off the ground, a much more reasonable value.

Finally, to avoid that the robot capsizes on its side, it is necessary that $\omega_{y,critical}$ has the largest possible value. Therefore, it is important that the base of a vertical spinner or drumbot is wide, because a larger distance 2·d between the wheels increases $\omega_{y,critical}$, as seen in the above equation. This explains, for instance, the reason for the large distance between the wheels of the fairyweight vertical spinner Nano Nightmare, pictured to the right.

If the vertical spinner from the previous example had a wider base with d = 0.3m (due to a distance 2·d = 600mm between the wheels, about 23.6"), then the calculations would result in allowable turning speeds of up to 33RPM. In other words, it would be able to turn 180 degrees in less than 1 second, a reasonable value to keep facing the opponent.

6.16. Summary

In this chapter, it was shown that weapon design can benefit a lot from basic physics calculations. It was concluded that spinning disks have better inertia and in-plane bending strength than bars with same weight, however they suffer from a lower out-of-plane bending strength. The concepts of tooth bite, as well as effective mass, stiffness and damping, were introduced, showing that the effectiveness of an impact depends on properties of both attacker and attacked robot. It was shown that drumbots have one of the highest effective masses M_1, however they usually suffer limitations regarding the speed v_{tip} of the weapon tip. Impact equations were presented for several robot types, including spinners and hammerbots. The difference between defensive and offensive rammers was discussed, including information on how to setup their shields. It was shown why thwackbots and overhead thwackbots are difficult to steer to try to hit an opponent. Lifter and clamper stability equations were also presented. It was seen that "height launchers" can benefit from a long scoop, while "range launchers" should choose average impulse angles between 36^o and 45^o, depending on the opponent's aspect ratio, as long as they have enough tire friction not to be pushed backwards. It was found that defensive wedges have a slope angle of at most 25^o, while an ideal offensive wedge would be made out of Ti-6Al-4V titanium with a smooth surface and a 37^o slope. And, finally, gyroscopic effect equations were presented to help in the design of vertical spinners. In the next chapter, the main electronic concepts to power such weapon systems and the robot drivetrain are introduced.

Chapter

7

Electronics

There are countless electrical and electronic options to use in a combat robot. This subject by itself could result in an entire book. Because of that, in this chapter we try to summarize and limit the discussion to the most used components in combots, with effectiveness verified in practice, in the arena.

Combat robot operation demands a great number of electronic components, among them: radio transmitter, receiver, RC interface, speed controllers, relays/solenoids and on-off power switches, connected by plugs, terminals and wires. These components are described next.

7.1. Radio Transmitter and Receiver

7.1.1. Transmitters

A radio transmitter allows the driver to send commands to a receiver inside the combat robot. There are several radio manufacturers, such as Futaba, Airtronics, JR, Hitec, GWS and Spektrum. There are also other cheaper solutions, such as radios adapted from toys, wireless gamepads, and transceiver circuits, however you must guarantee that these low-cost systems will have enough power to avoid signal loss when the robot is inside the arena, as well as implement failsafe features in all channels, as explained later.

Radio systems are named for their number of channels, which is the number of outputs that a transmitter-receiver set has. For instance, a four-channel set can control four different devices. Most combat robots only use three channels: two for the drive system (forward/backward and left/right) and one to control the weapon, if any. Three and four-channel radios are cheaper and, in general, enough for a combot. However, radios with 6, 7 or more channels usually have more functions, being programmable and including internal memory, such as the Futaba 7CAP 75MHz radio pictured to the right.

Most radios use the 72MHz frequency band, which is reserved for air models only, which makes them prohibited in almost all combot competitions. The usual ground band is 75MHz, but several others are also used: 27MHz, 50MHz, 433MHz (only in Europe, Africa and Middle East), 900MHz (only in the Americas), 2.4GHz and 5.8GHz. All these frequencies are part of the Industrial, Scientific and Medical (ISM) radio bands, originally reserved by an international treaty for the use of radio-frequency in the cited fields other than communications.

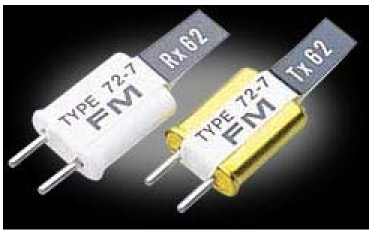
The 27 and 50MHz bands are normally employed in radio-controlled toys and old cordless phones, 433MHz is found in European transceiver pairs, while 900MHz and 2.4GHz are a commonplace in modern life, being present in wireless LANs, Bluetooth, cordless phones, and even microwave ovens. The 5.8GHz band is not very common, but it is also present in wireless network systems and phones.

Each band is divided into channels. There are only 30 channels for the 75MHz band, from channel 61 (75.410MHz) to 90 (75.990MHz). The 72MHz band, which is not allowed in combot competitions, has 50 channels, from 11 (72.010MHz) to 60 (72.990MHz). And the 27MHz band has only 6 channels, while the 50MHz has 10.

In a competition, it is forbidden to have two radios using the same channel at the same time. This is a safety measure, since the radio from one team could accidentally activate the robot from another. With this in mind, in the events it is mandatory to only turn your radio on if it has an appropriate frequency clip. Since there is only one clip available per channel at the event, the problem is solved.

However, if a distracted builder forgets to return a clip after using it, the event might suffer delays. As a result, a few events require the use of single bind systems, which only allow a receiver to follow commands from a single radio, normally using the 900MHz or 2.4GHz frequencies.

Those systems rely on Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS), which use a unique identifier code for the receiver and transmitter. They dynamically change the paired transmitter and receiver channels upon signal quality loss, caused normally by interferences (noise). This bi-directional system is only possible due to the larger bandwidth and higher frequency they use. A few 900MHz and 2.4GHz systems can use the available bandwidth to perform robot telemetry as well, such as in the IFI FRC or Spektrum Telemetry packs.



In 72 and 75MHz systems, channels are defined by a pair of crystals (pictured to the right), usually sold together. The Tx crystal must be installed into the transmitter, while its pair Rx is placed into the receiver. Always have a spare crystal pair from a different channel, in case you have to face an opponent that uses the same channel. Only buy crystals for the same band as your radio - 72MHz crystals do not work on 75MHz radios and vice-versa. It is possible to convert a radio from 72MHz to 75MHz, a procedure that costs between US$20 and US$50 if done by professionals.

Besides frequency bands and channels, there is also modulation, which is the way that information is encoded into the radio wave. The most commonly used modulations among the countless existing standards are AM, FM, PCM and DSM, described next.

AM stands for Amplitude Modulation, in which information is transmitted varying the radio signal amplitude.

FM means Frequency Modulation, it is a standard less prone to noise than AM. Control information is transmitted adding a variable frequency wave into a carrier wave. The receiver then extracts the information from the carrier and sends the appropriate commands to the devices attached to it.

The third modulation type is PCM, which means Pulse Code Modulation. Technically, this transmission is also FM, the difference is that in PCM the information is digitally transmitted. Instead of an analog transmission, signals are send in digital form, coded, becoming a lot more reliable. PCM provides an even greater noise immunity.

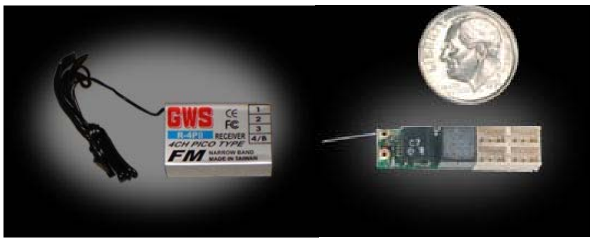
DSM, which stands for Digital Spectrum Modulation, is Spektrum's proprietary modulation for 2.4GHz systems. It divides the 2.4GHz band into 80 channels (slots), using some DSSS and FHSS features, with a unique identifier that only allows communication between a single transmitter and its bound receiver. In the unlikely event of all channels becoming occupied, the link between the Tx and the Rx won't happen. This standard has been updated to DSM2; unfortunately, older radios such as Spektrum DX6 only accept DSM receivers, although DSM2 radios work with DSM receivers.

Besides Spektrum and JR (which also uses DSM), other radio brands have their own 2.4GHz modulation, such as Futaba’s FASST, Airtronics’ FHSS-2, and Hitec’s AFHSS. Their differences are minimal, so choose them keeping in mind your budget and favorite brand.

7.1.2. Receivers

A receiver is the component responsible to demodulate the radio-transmitted signals and direct the commands to servos and other electronic circuits. A typical receiver is pictured to the right, a Futaba 75MHz. They come in several sizes and weights.

For insect robots, such as fairyweights (150g), antweights (1lb, equivalent to 454g) or beetleweights (3lb, equivalent to 1361g), it is a good choice to use the GWS micro receiver (left picture to the right) or the Nano receiver (right picture). They are really small and weigh between 2 and 8 grams without the crystal. Be careful, they do not work with regular crystals, they need special ones.



It is extremely important that the Rx crystal (if needed) is well attached to the receiver. It is usually a good idea to use an adhesive tape around the receiver to prevent the crystal from becoming loose. Many fights have been lost because of a knocked off crystal due to the vibrations from a major hit. Tape as well the connectors, to avoid the cables from becoming loose. It is also a good idea to use some adhesive tape to cover the receiver inputs that are not used, to prevent metal debris from causing a short-circuit during a round. Also, it is fundamental to shock-mount the receiver, using for instance foam, EVA or rubber.

All modern receivers pick up the modulated signal and then decode it, resulting in a Pulsed Position Modulation (PPM) signal that contains the information about all output channels. This information is framed in an envelope of 20ms (or some other fixed value between 18 and 25ms), consisting of a train pulse with several 5V pulses, one for each output. These 5V pulses have periods that vary between 1ms and 2ms, depending on the command sent by the driver. If a radio stick is completely to the left or down, then the period of the associated pulse from this channel is usually 1.0ms. If the stick is centered in neutral position, then it is 1.5ms. And if it is to the right or up, then the pulse lasts 2ms. For switches, as those that are usually placed on the top portion of a radio transmitter, 1ms would be associated to the off position and 2ms to the on position. Most high-end systems allow these configurations to be changed on the transmitter unit. For instance, a channel can be reversed, or it can be mixed with others. PPM is further explained in section 7.8.2.

Among the radio features, *failsafe* is one that is required by all combot events. A few vendors call it *failproof* or *smartsafe*. It consists of a subsystem that allows to pre-program each receiver channel output in case of a signal loss. Program the failsafe to, in the event of signal loss, send a 1.5ms pulse signal to the drive system channels (associated with a centered radio stick, which would stop the robot's translational movement), as well as send a 1ms pulse signal to a channel from a solenoid or relay from the weapon system (if the 1ms pulse is associated with the off condition). This is an obligatory feature, checked in the safety inspection from any combot event. For hobbyweights (12lb) or heavier robots, all used channels must have a failsafe. Usually, robots in insect classes only need to have failsafe on the weapon channel.

AM, FM and a few other radio systems do not have built-in failsafe. Failsafe must then be implemented between the receiver and the commanded devices, which can be accomplished using an appropriate RC (radio-control) interface board, or through a dedicated module such as the Micro Failsafe Dynamite (pictured to the right), which can be bought in R/C hobby stores.

Listed below is a summary of the main features from radio-receiver systems:

- number of output channels: in general from 4 to 9 in air radios, or 3 in pistol-style ground radios; for a robot with an active weapon, at least three channels are needed;
- reversion: ability to program channel output inversion;
- ATV / EPA: adjustable maximum and minimum values that an output can have;
- dual rate / exponential: output sensibility and linearity adjustment;
- mixing: ability to mix channels (very useful), which sometimes can be programmable;
- multiple models: allows the storage of several distinct programs, one for each robot;
- failsafe: allows to program the channel outputs in case of a signal loss; only digital radios have this feature, but a few do not have it in all channels; analog radios must use an external module;
- frequency channel reassignment: ability to automatically switch the transmitter and receiver channels.

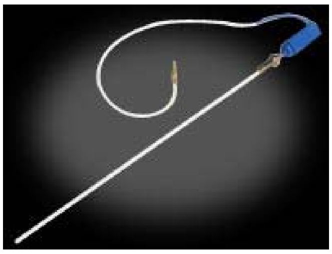
7.1.3. Antennas

A huge problem that strikes combat robots is signal loss. Combots are made out of metal, therefore antennas placed inside them may suffer from the Faraday's Cage effect, where the robot chassis blocks the signal or at least considerably reduces its strength.

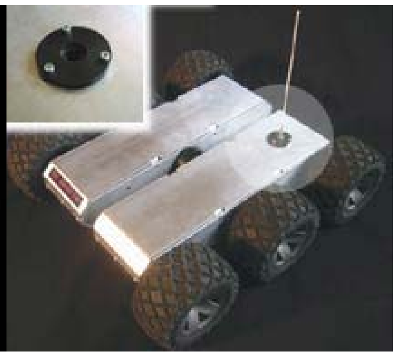
Robots with polycarbonate covers don't suffer from this problem, since this material is radio transparent. Sometimes, if a robot is completely shielded by metal, then the solution is to place the antenna outside it. There is a risk that the antenna will get damaged, impairing signal reception, however it is a better choice than to not have a control signal at all.

The good news is that the higher the frequency, the smaller is the wavelength. Therefore, modern systems that use 2.4GHz have waves that can go through the breaches and gaps on the robot covers (such as the holes around the wheels). This allows you to place the antenna inside the robot. All our bots have internal antennas, since we've changed the radio system from 75MHz to 2.4GHz.

Antennas are usually a conductive wire with one fourth of the signal wavelength. Therefore, a 75MHz signal traveling at light speed (about 300,000 km/s) has a wavelength of $(3 \times 10^8 \text{ m/s}) / (75 \times 10^6 \text{ Hz}) = 4$ meters (about 13 feet). Since a 4 meter long wire would be too long, the 75MHz systems use one with 1/4 of this length, 1 meter (a little over 3 feet). In practice, it is a good idea to place the antenna wire in a zig-zag pattern inside the robot. Often, the longer the antenna, the higher is its gain, so if your robot needs a signal boost then try longer antenna lengths.



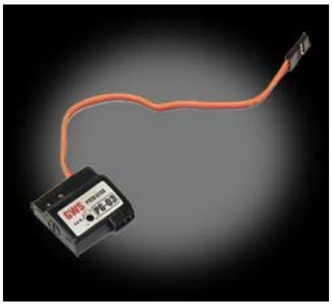
You can replace the 3-foot wire from the receiver antenna by a mini-antenna with less than 8 inches long. But you need an amplified antenna for a good result, such as the Deans Base-Loaded Whip (pictured to the right). Before switching from 75MHz to 2.4GHz, we used this antenna in our middleweights Touro and Titan without any problems, however it had to be placed outside the robot due to Fadaray's Cage Effect.



Avoid placing the antenna directly over metal. Ideally, it is good to have at least 6mm (1/4 inch) between any part of the antenna and a metal part. This can be achieved with a large rubber grommet or with some Lexan or Delrin spacer, such as the Delrin Antenna Mount pictured to the right, sold at The Robot MarketPlace.

Since the wavelength in 2.4GHz systems is only $(3 \times 10^8 \text{ m/s}) / (2.4 \times 10^9 \text{ Hz}) = 0.125\text{m}$ (less than 5 inches), their antennas are really tiny, between 31mm and 62.5mm in length. These receivers (such as the Spektrum 2.4GHz BR6000 model pictured to the right) normally have two antennas, due to wave polarization. It is recommended to place the antennas forming a 90° angle to maximize signal reception.

7.1.4. Gyroscopes



To guarantee that a robot can follow a straight line, you can use gyroscopes (a.k.a. gyros), which are sensors that measure the robot orientation angle. Easy to use, they can be directly connected to most receivers. Almost all R/C hobby stores sell gyros, which are usually used in radio-controlled model helicopters. One of the cheaper models is the GWS PG-03 Micro Gyro (pictured to the right), which costs US$38 at The Robot MarketPlace, which has a great cost-benefit.

The secret to use gyros in combat robots is to adjust the feedback gain to a maximum of 20% of its full scale. This worked very well with our hobbyweight wedge Puminha using a GWS PG-03. Higher gains can make the system become unstable, since the gyroscope will pick up motor vibrations and try to compensate them.

It’s not recommended to use gyros on invertible robots, because when the robot is upside down the control gain is inverted, giving positive feedback and making the robot spin out of control (a.k.a. the “Death Spin”). To solve this problem, it would be necessary to have an electronic system to turn the gyro off when the robot is flipped over, or even better, to invert the gain in this case.

Note that gyros must be very well shock-mounted inside the robot not to break. And they must be well secured because, if they get loose or shift their position inside the robot, it will go crazy or go into “Death Spin” mode.

7.1.5. Battery Elimination Circuit

In several model airplanes, there is a small battery pack used exclusively to power the receiver. If your electric motors generate a lot of noise, it can be a good idea to have separate packs for the motors and for the receiver, to avoid interference or signal loss. However, if the motors have noise suppression capacitors or shielded armature, then it is much simpler to use the robot’s main batteries to power as well the receiver, without using a separate pack.

This small receiver pack is troublesome: it can become loose inside the robot, it increases the robot weight, and it is another battery pack that needs to be charged, hogging your chargers. In addition, there’s a chance you might forget to charge them during a hectic event.

To eliminate the receiver pack, you need a Battery Elimination Circuit (BEC). The BEC is nothing more than a voltage regulator that guarantees a constant supply of, usually, 5V (or other value between 4.8V and 6V).

A few speed controllers, discussed later, already have built-in BECs. But most of them use linear regulators, such as the LM7805, which can overheat if the voltage drop is too high. Our middleweight Touro needs a BEC because the Victor speed controllers that it uses do not have this feature. We once were out of BECs, so we’ve experimented using BaneBots BB-12-45 speed controllers just to work as BECs. These speed controllers have a built-in BEC, however due to the voltage drop from 27.2V (20 NiCd cells in series fully charged) to 5V they overheated, malfunctioned, and sometimes even got burnt. This speed controller is rated for 24V at most, so it is not recommended to power it beyond this voltage, unless some heatsink is attached to it to dissipate the heat from the linear regulator.

If a speed controller has a built-in BEC but you don't need to use it (because you have a separate stand-alone BEC, for instance), then it is a good idea to remove the red wire (the middle one) from the crimp connector that goes into the receiver. This will avoid it from overheating, especially if it uses a linear regulator. This tip is valid for both brushed and brushless speed controllers.

There are commercially available stand-alone Battery Elimination Circuits, named UBEC (U stands for Universal), such as the S-BEC Super BEC 5V model pictured to the right. They are usually cheap, found in R/C hobby stores. A few UBECs use switching regulators, such as the LM2575, which switches the voltage on and off in order to drop its value, instead of dissipating the power from the excess voltage as heat. In this way, they are less prone to overheat or malfunction.

You can also develop your own BEC. We've used our own BECs for years with great results, as discussed in section 7.8. If you decide to build your own BEC, choose switching regulators, they don't overheat as easily as linear regulators.

7.1.6. Servos

Servos, a.k.a. servo-motors, are motors with embedded position control, such as the Hitec standard model pictured to the right. In model aircrafts, small low-power servos are directly connected to the receiver, and powered by it. Servos are very practical and cheap, a few combat robots use them in the throttle system of internal combustion engines, or to mechanically control some electric switch. The problem with this approach is the great risk of servo failure after an impact, which can break them or let them become loose. In combots, always implement your control system electronically (in solid state), avoiding moving parts or servos.

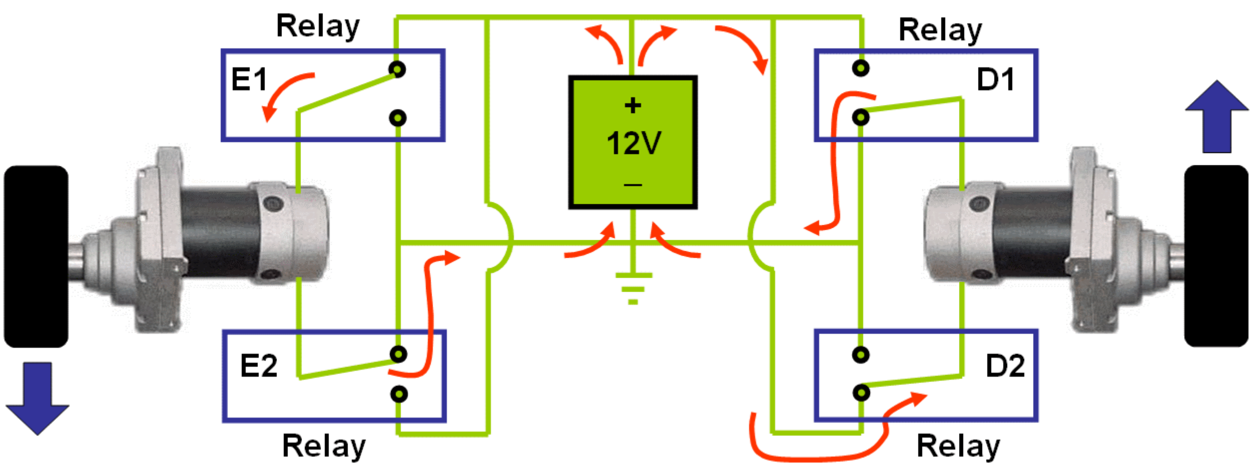
Servos have been used in insect robots, actuating lifter or clamper/grabber mechanisms that need position control to properly function. With a simple modification, servos can become DC motors with continuous rotation, with an embedded electronic control to convert PPM signals into movement. The modification consists of disassembling the servo, exchanging the internal potentiometer by a two-resistor ladder, removing the plastic stop from the largest gear, and reassembling the unit. A good tutorial on this servo modification can be found at http://www.acroname.com/robotics/info/ideas/continuous/continuous.html. The modified servos can be used in the drivetrain of small robots such as insect combots or light sumo bots, however only the spin direction can be controlled, not its speed, as in the bang-bang control discussed later.

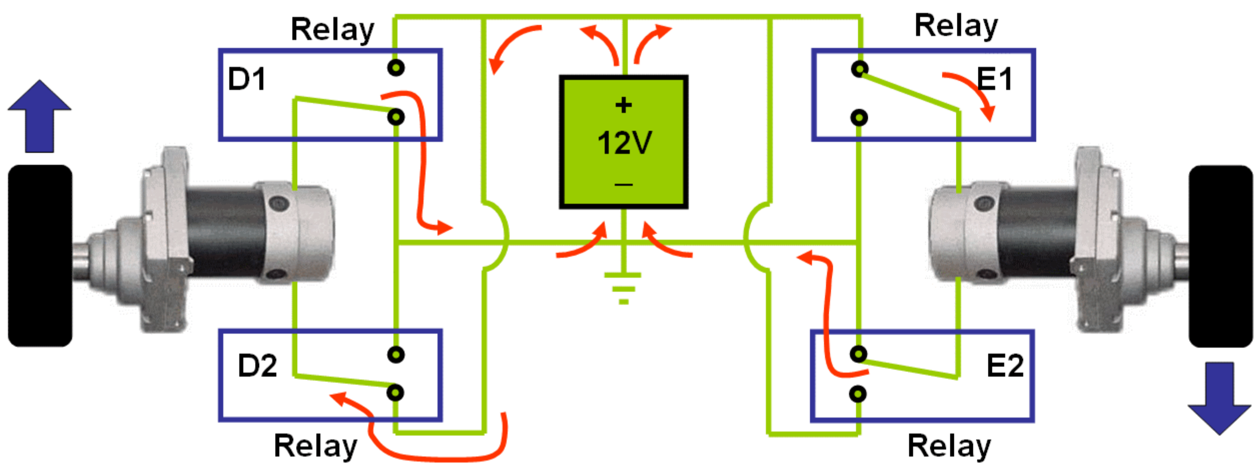
Note that servos included in radio transmitter-receiver packages do not have enough power to actuate most systems in larger bots. High torque servos should be preferred, however most of them will need to be modified if you need continuous rotation. If you need continuous rotation, including speed control, then it is better to stick to DC motors with speed controllers, discussed next.

7.2. Controlling Brushed DC Motors

7.2.1. Bang-Bang Control

The most primitive way to drive a DC motor is through a bang-bang controller. Is consists of using relays to create a basic H-Bridge scheme, in such a way that the motor can spin at full power forward, full power backward, or stop. The figures below show how it is possible to use 4 relays to create a bang-bang controller for two opposite wheel motors.





As shown in the upper figure, when the E1 and E2 relays are closed and the D1 and D2 are open, the robot turns to the left. On the other hand, closing relays D1 and D2 while opening E1 and E2 would make the robot turn to the right, as shown in the lower picture. To move forward, you only need to open all four relays. To move backward, you should close all relays. Finally, to stop the robot, you can choose for instance to close E2 and D2, while opening E1 and D1. In the above example, if a 12V battery was used, then each motor would only receive either +12V, 0V or –12V.

Avoid using this type of controller, because it always provides the maximum voltage (in absolute value) to your motors, abruptly reversing their direction. This sudden reversion of the movement causes premature brush wear and can lead to broken gears because of the associated impacts (which are the origin of the "bang-bang" expression). In addition, the high inductance of the motors will create voltaic arches on the relay terminals, shortening their useful life. This kind of controller is only acceptable in low speed motors, which are not very common in a combat robot, except perhaps for a slow lifting or clamping/grabbing mechanism.

A slightly improved version of a bang-bang control would use the 4 relays in series with a single adjustable linear voltage regulator. As seen in chapter 5, the speed of a brushed DC motor is proportional to the applied voltage, so the linear regulator would control speed, while the relays would control direction. Our first combot, the middleweight Lacrainha, used such bang-bang implementation to control the speed of its wheels. The electronic board that we developed also featured an automatic system that would briefly lower the applied voltage down to zero using the linear regulator during the reversion of the relays, to avoid voltaic arches on the relay terminals.

Although very simple, this version of bang-bang control has serious issues, mainly due to its low efficiency at low speeds, since all the energy from the excess voltage that is not utilized by the motor is dissipated as heat on the linear regulator. In addition, the resulting electronic system may still be unreliable since it depends on mechanical moving parts from the relays.

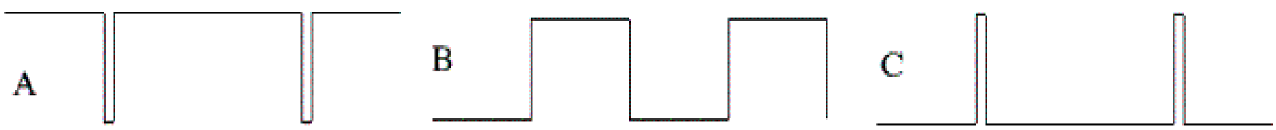
To achieve a higher motor efficiency and more compact and reliable circuits, it is necessary to vary the motor speed in a different way, known as PWM, described next.

7.2.2. Pulse Width Modulation

The Pulse Width Modulation (PWM) method consists of turning the motor voltage on and off, on a fixed frequency basis, through an electronic switch, usually some kind of transistor. The transistor can be, for instance, a Bipolar Junction Transistor (BJT) or a Metal-Oxide Semiconductor Field-Effect Transistor (MOSFET, a.k.a. FET).

The motor speed is then proportional to the ratio between the time interval T_{on} during which the motor is on and the pulse period T. This relation is named Duty Cycle (D) and, if multiplied by the peak voltage supplied to the motor, it results in an average voltage that can be controlled.

The figures below show three PWM signals that use the same frequency. Figure A shows the PWM signal with almost 100% D, because the time T_{on} is almost equal to the period T (the signal is high almost 100% of the time). Therefore, a motor subject to the pulse A would behave as if it was powered by almost the nominal voltage.



The pulse in figure B, on the other hand, is high half of the time, therefore T_{on} is equal to T/2, and the motor would receive about half of the nominal voltage, spinning at half-speed (50% D). Finally, the pulse in figure C would make a motor spin very slowly, since the time T_{off} during which it is off, equal to $T - T_{on}$, is almost equal to T (almost 0% D).

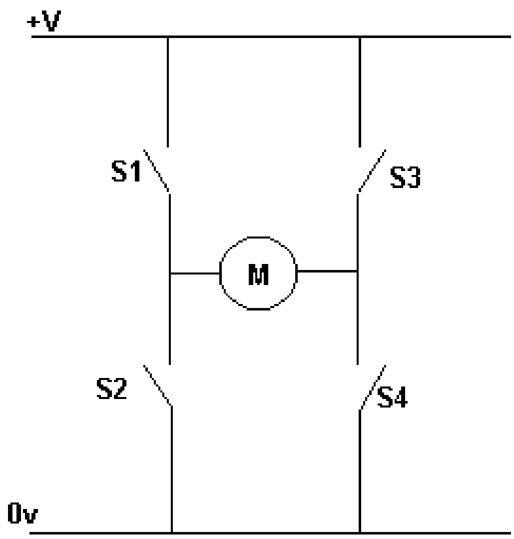
The efficiency of a PWM circuit is, in an ideal case, 100%, since the switching process that would power the motor wouldn't have any losses. But in practice there are losses in the used transistors, due to their resistance. Despite this, the efficiency of a well-designed PWM controller is usually above 90%.

Be careful not to confuse PWM with PPM (or even PCM), several people mix these concepts, saying that a receiver output signal is PWM. Although these signals are similar, this is not true. Both PWM and PPM are analog pulse trains, yet their functionalities are completely distinct. One of the differences is that the PPM needs to use a train pulse with a precisely defined period. The PWM, on the other hand, can have any pulse period T, what really matters is only the Duty Cycle, the ratio between T_{on} and T. The pulse frequency 1/T is an arbitrary value, but it should be large enough to avoid undesired oscillations in the motor. Typical values of 1/T are above 4kHz, sometimes higher than 16kHz. Using 20kHz or higher is a good idea to avoid buzz sounds, since it is above the range of human hearing.

7.2.3. H-Bridge

PWM by itself is only able to control the absolute value of the motor speed. There are a few possible ways to control the motor direction. You can, for instance, physically invert the terminals using mechanical switches such as relays and solenoids, however this would result in a bulky and impact-sensitive system, with lower reliability, as discussed before. Another option is to generate a negative voltage, however this can result in a very complex system since the main power supply is a battery, which provides direct current.

The best option is to use an H-Bridge, as mentioned before, named after the disposition of the switches in the circuit. It does not need to generate negative voltages, or to mechanically disconnect the terminals. The H-Bridge can use transistors, which easily stand high currents.

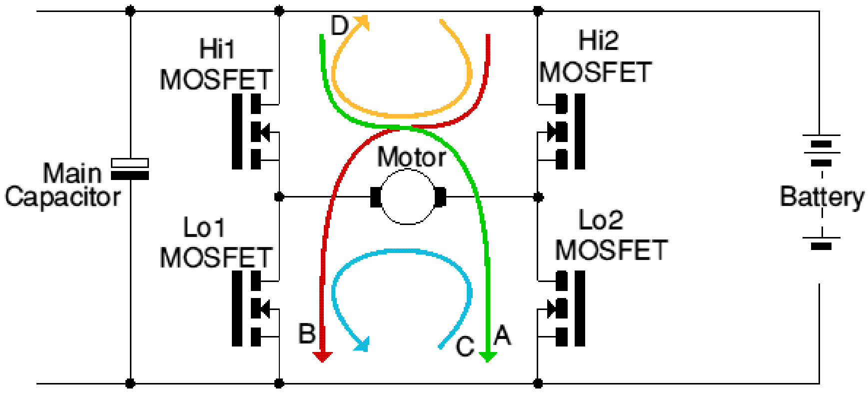


The picture to the right shows a basic H-Bridge, with a motor M and 4 transistors S1, S2, S3 and S4 that work as solid-state switches.

To make the motor spin forward with a voltage V, you just need to activate S1 and S4 and deactivate (open) S2 and S3. To spin backward with voltage V, activate S2 and S3 and deactivate S1 and S4.

To brake the motor, you can either activate only S1 and S3, or activate only S2 and S4, shorting the motor terminals. This braking effect is called motor brake and happens due to the entire energy being dissipated by the motor internal resistance, which is usually very low. This results in quick energy dissipation, which happens until the motor stops. Note that motors with very small internal resistance can generate a lot of heat in this process, so be careful!

In the figure to the right, the switches are implemented using MOSFETs, resulting in a simple H-Bridge circuit.



To spin the motor forward, the current from the battery needs to go through the FETs Hi1 and Lo2, following the path A shown in the figure. To spin backward, Hi1 and Lo2 must be deactivated, and immediately after this Hi2 and Lo1 need to be activated, making the current follow path B. Clearly, to brake the motor you can either activate Lo1 and Lo2 (for the current to follow path C, in either sense), or activate Hi1 and Hi2 (to follow path D, in either sense). Note that paths C and D are only possible because these FETs feature integrated free-wheeling diodes, which allow the current to go from source to drain (the upward direction in the figure). Make sure these diodes are present, otherwise the motor inductance will fry the FETs.

PWM is very easy to be implemented into the H-Bridge. When spinning forward, it is enough to keep Hi2 and Lo1 always deactivated and Hi1 always activated, while Lo2 is only active during the T_{on} interval from the PWM, making the current follow path A.

Due to the motor's inductance, the current tends to keep a constant flow even during the T_{off} period. Therefore, during T_{off} from a forward movement, the current will follow the opposite sense of path D, flowing through the motor in the same sense as it did in path A. Note that Hi2 can remain deactivated during T_{off}, because its free-wheeling diode always allows conduction from source to drain, necessary for the opposite sense of path D (Hi2 only controls the flow from drain to source).

This path during T_{off} shorts out both motor's terminals, however with minimal energy losses. It indeed brakes the motor, but this is not a problem, because it only happens during T_{off}. It is the combination of motor acceleration (during T_{on}) and braking (during T_{off}) that allows the resulting average speed to achieve any value between zero and the top speed. As mentioned before, the period T must be small enough so that the acceleration and braking effect cannot be noticed, avoiding oscillations in the motor.

Analogously, to spin backward, you just need to keep Hi1 and Lo2 always deactivated and Hi2 always activated, while Lo1 is only active during the T_{on} interval from the PWM. During T_{on}, the current will follow path B, while during T_{off} the motor inductance will make the current flow through the motor in the same sense as it did in path B, resulting in path D in the sense described in the figure (not in its opposite sense). Note that Hi1 can remain deactivated during T_{off}, because its free-wheeling diode always allows conduction from source to drain, necessary for path D.

A major issue with H-Bridges is an effect called shoot-through, which occurs if two switches from the same side are active at the same time, for instance Hi1 and Lo1. If this happens, the battery is shorted out, generating a very large current that usually destroys the MOSFETs. To address this issue, resistors are installed in series with the MOSFET gates, delaying their activation, while fast diodes are installed in parallel with those resistors, as described in section 7.8.

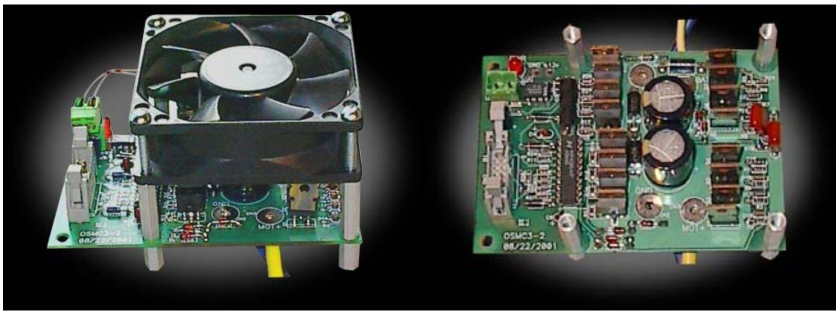
7.3. Electronic Speed Controllers

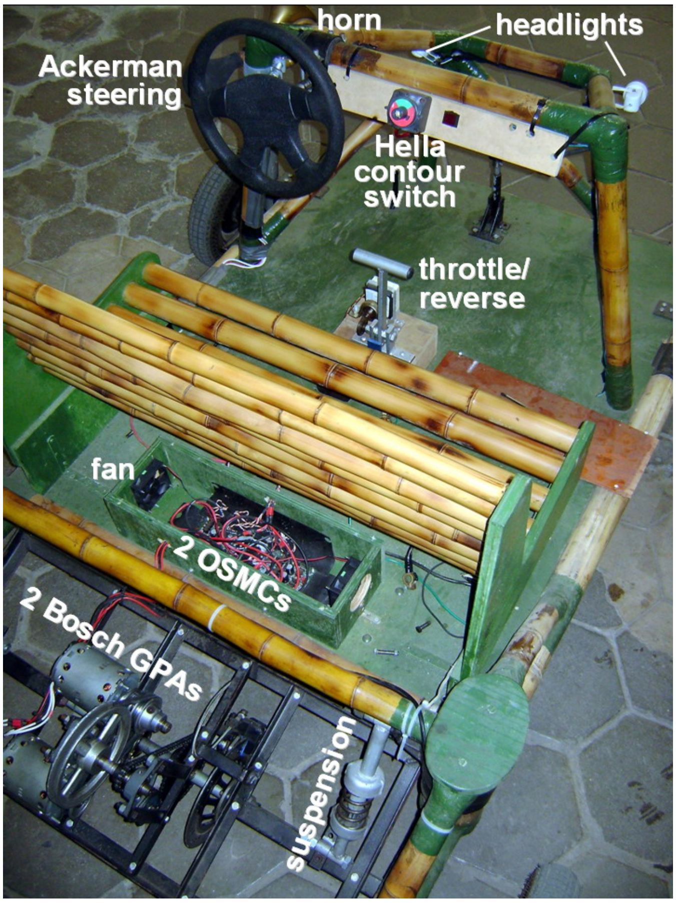
Electronic Speed Controllers, or simply ESC, are electronic power systems that implement an H-Bridge with PWM, to control both spin direction and speed of a motor. There are several ESCs in the market, so we'll focus on the ones that we have already tested in combat: OSMC, Victor, Scorpion and BaneBots, which are all brushed Permanent Magnet DC (PMDC) motor controllers. All these systems can be purchased, for instance, at The Robot MarketPlace, IFI Robotics, BaneBots, Trossen Robotics or Robot Power.

To control Brushless DC (BLDC) motors, a Brushless Electronic Speed Controller (BESC) is required, explained in section 7.3.6.

7.3.1. OSMC – Open Source Motor Controller

OSMC is a speed controller board capable of powering a single DC motor with nominal voltages between 13V and 50V, handling continuous currents of up to 160A, with 400A peaks. They are very robust, using 16 MOSFETs on an H-Bridge (4 transistors per leg), cooled by a fan. The pictures to the right show the OSMC board with and without the fan.





We have used OSMCs in a few of our combots, such as the middleweights Anubis, Ciclone and Titan. We also use 2 OSMCs to power the 2 Bosch GPA motors that drive our bamboo electric vehicle, pictured to the right. This vehicle, built by the students, is useful to carry two people and a middleweight from our lab to the “weapon testing field” from our University (a.k.a. the soccer field).

The OSMCs are a little bulky if compared with other ESCs available in the market, but they are very reliable. The middleweight wedge Max Wedge, honorable mention in the Robot Hall of Fame, used OSMCs to control the speeds of its power hungry D-Pack motors.

The Open Source Motor Controller is the collaborative result of several combot and electric vehicle builders. The OSMC diagrams can be found at http://robotpower.com/osmc_info. In this very site it is possible to buy an assembled and tested OSMC board for US$219, or a full kit including a bare-board and all components needed for the assembly at US$169.

To save some money, our team bought several bare-boards for US$29 each at Robot Power, and looked for information about the needed components and their suppliers in the OSMC Discussion Group (http://groups.yahoo.com/group/osmc) to buy them individually. The savings were not significant if compared to the Robot Power full kit, our cost per board was slightly below US$159. But at least we've learned a lot about each individual component and its vendors.

One disadvantage of the OSMC board for use in combots is the need for a separate electronic RC interface between the OSMC and the receiver, called MOB (Modular OSMC Brain), or µMOB for a smaller version. Unfortunately, the MOB/µMOB interfaces have been discontinued, so we've developed our own RC interface board, which is able to control 2 OSMCs. Our RC interface, which can also trigger a solenoid to be used in the weapon system, is detailed in section 7.8.

7.3.2. IFI Victor

Victor is a family of speed controllers from IFI. These ESCs are extremely compact, almost as robust as the OSMC. We use Victors in our middleweight Touro to save precious space for other components (Touro ended up so compact that OSMCs wouldn't fit inside it). Each Victor only controls one motor, so you'll need at least two in the robot drivetrain.

There are several Victor models, however externally they have the same look, as pictured to the right. A fan is used to cool the MOSFETs from the H-Bridge. There are several different models and prices, listed next.

- Victor 884 – US$ 114.95: single channel, forward and reverse, from 6 to 15V, handling up to 40A continuously; it uses 12 MOSFETs, 6 for each direction;
- Victor 883 – US$ 149.95: single channel, forward and reverse, from 6 to 30V, up to 60A continuously, with a surge current capability of 100A for less than 2s, and 200A for less than 1s; it uses 12 MOSFETs, 6 for each direction;
- Victor 885 – US$ 199.95: single channel, forward and reverse, from 6 to 30V, up to 120A continuously, surge currents of 200A for less than 2s, and 300A for less than 1s; it uses 12 MOSFETs, 6 for each direction;
- Victor HV-36 (the model shown in the picture above) – US$199.95: single channel, forward and reverse, from 12 to 42V, up to 120A continuously, surge currents of 250A for less than 2s, and 275A for less than 1s; it uses 16 MOSFETs, 8 for each direction;
- Victor HV-48 – US$199.95: single channel, forward and reverse, from 12 to 60V, up to 90A continuously, surge currents of 200A for less than 2s, and 225A for less than 1s; it uses 16 MOSFETs, 8 for each direction;

- Victor 883SC – US$169.95: single channel, forward only (therefore it is usually used to power a spinning weapon), from 6 to 30V, up to 90A continuously, surge current capability of 100A for less than 2s, and 200A for less than 1s; it uses 12 MOSFETs, all of them for a single direction.
- Thor 885SC – US$219.95: single channel, forward only, from 6 to 30V, up to 150A continuously, surge current capability of 200A for less than 2s, and 300A for less than 1s; it uses 12 MOSFETs, all of them for a single direction.

Victors can be connected directly to the receiver, without the need for an external RC interface (such as the MOB needed by the OSMC). However, a signal booster cable (US$ 15, pictured to the right) is highly recommended between the receiver and the Victor, improving the quality of the signal. The booster cable is also interesting for robots that suffer from radio signal noise or loss, especially in combots with internal combustion engines.

IFI also sells speed controllers specifically developed to drive spinning weapons, denoted by the SC (Spin Controller) suffix, such as the Victor 883SC and Thor 885SC discussed above. They only work in one direction, but they feature softer acceleration and deceleration ramps that minimize current peaks, saving battery capacity. With them it is possible to control the speed of a spinning weapon, which can be especially useful in three situations:

1. batteries not lasting the entire match: you could spin the weapon at an intermediate speed, saving the batteries, and accelerate to full speed a few moments before striking your opponent;
2. damaged or cracked weapon: if, after a tough match, some structural damage or crack is found on a weapon component that cannot be replaced, then it is possible to avoid a broken weapon in the following fights adopting a lower speed;
3. robot vibration when the weapon is at full speed: slightly slowing down the weapon can minimize this problem, allowing you to control the weapon speed to avoid natural frequencies.

Despite the acceleration and deceleration ramp features from the SC types, most builders prefer to use in their weapon systems ESCs that are capable of forward and reverse spin, resulting in a fully reversible weapon system (which is very useful, for instance, for invertible drumbots). If you don't need a fully reversible weapon, such as in most horizontal bar spinners, and if you don't care about controlling the weapon speed, then it is a good idea to use solenoids, due their lower price tag and higher current limits. Solenoids are studied in section 7.4.

We have been using Victors HV-36 to control the drive system of all our combots from 30lb to 120lb. Our 4-wheel drive hobbyweight wedge Puminha also uses Victors HV-36, each one is used to power both drive motors from each side of the robot. But, since Victors in hobbyweights are usually overkill, we've removed their fans in Puminha without needing to worry about overheating problems, saving precious space.

The Victor HV-36 is a better option than the Victor 885 even for 24V combots, because it uses 16 MOSFETs instead of only 12, better handling high peak currents from high power motors. And the use of the Victor HV-36 also allows future robot upgrades up to 36V.

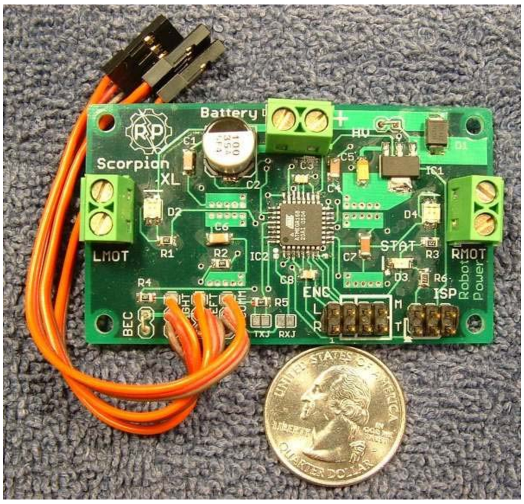
The Victor HV-36 is also a good option for the weapon system. After RoboGames 2008, we've replaced the weapon solenoids from our middleweights Touro and Titan, lightweight Touro Light, and hobbyweight Tourinho, with Victors HV-36, resulting in fully reversible weapons with speed control. One Victor HV-36 is needed for each Magmotor S28-150 or DeWalt 18V motor from the weapon systems of Touro Light and Tourinho. But we use 2 Victors HV-36 for each Magmotor S28-400 from the weapon system of our middleweights, as explained in section 7.7.3, for redundancy and to avoid burning the ESCs due to the high stall current of this motor.

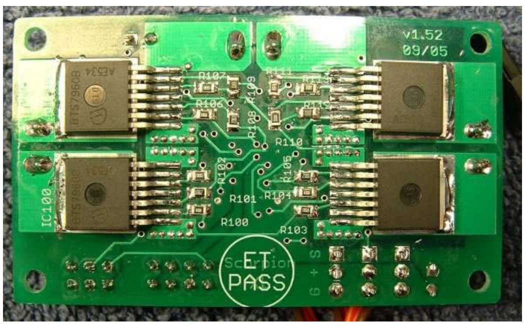
Since the fans are really important to cool down the MOSFETs, it is recommended to protect them. Debris or even loose wires from the electric system can touch the fan blades, making it stop. To avoid this, you can use a fan grill (pictured to the right), which can be made out of steel wire or from an aluminum sheet. Another good idea is to use the IFI Stainless Steel PWM Clip, which locks the signal cables onto the Victors, preventing the connectors from popping loose.

7.3.3. Robot Power Scorpion

Victors and OSMCs are suited for larger robots, typically featherweights (30lb) or heavier. They are expensive and relatively large to be used in robots that weigh 15lb or less. Robot Power sells four ESCs that are a good option for those lighter combots.

Scorpion XL and XXL





The most famous product from the Scorpion line is the Scorpion XL (US$119.99, pictured to the right showing its front and back sides), a controller that offers two PWM output channels, normally used to control both drivetrain sides. It can handle from 4.8 to 28V, up to 12.5A continuous, and 45A peaks per output, which can be combined into a single channel to provide higher current limits.

In 2008, a beefier version was introduced, the Scorpion XXL (US$159.99). It uses the same board from the XL version but, instead of one BTS7960B power IC per H-bridge leg, this version has a pair of them, one on top of the other. This allows a higher continuous current of 20A. The channels can also be combined, resulting in a single PWM output that could handle continuous 40A. To withstand the higher currents, the battery screw terminals were replaced with wire pig-tails.

Since both XL and XXL versions use the same board, most features are the same, such as current and temperature limiting, flip input to allow a radio channel to invert the robot drive system commands (to be used if an invertible robot is flipped over), five signal mixing options, exponential output, built-in BEC, and failsafe. They have a downloadable quick start guide, however they lack a full user manual.

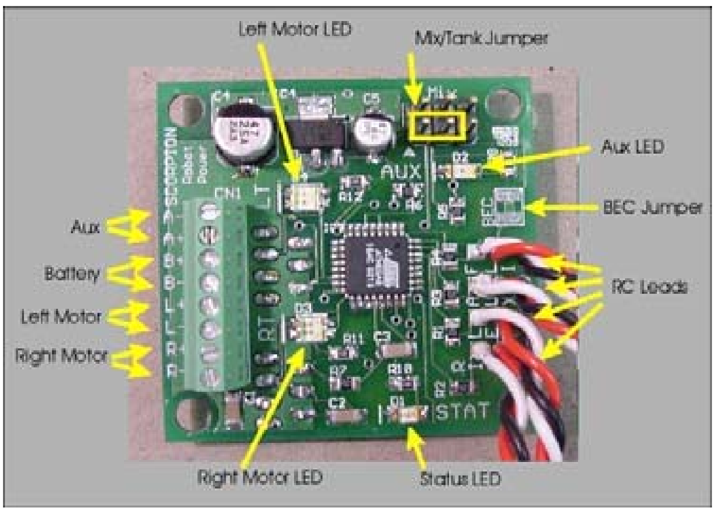
These controllers are a good choice for combots up to 15lb, especially for their size and built-in features. As a matter of fact, until 2007 our hobbyweights Tourinho and Puminha used Scorpions XL. However, in 2008 we've reduced the size of Tourinho's chassis to save weight, which required us to switch the Scorpion XL board to two BaneBots BB-12-45 ESCs, which are smaller in size. In addition, in 2008 we've replaced Puminha's drive motors to higher power models that would greatly exceed the 45A peaks, so its Scorpion XL was switched to a pair of Victors.

Scorpion XL is a very good board, however there are a few tips to make it bulletproof, as we've learned after two years using and abusing its 2006 version in our hobbyweights:

- the large SMD capacitor (silver cylinder with black notch under the battery writing in the top picture) can become loose or even break during an impact, unless the board is very well shock-mounted; to avoid this, you can use epoxy resin without metal additives, hot glue, or even tape, to better secure the capacitor; if the capacitor gets knocked off, the board will present an inconsistent behavior; you can replace the capacitor by carefully soldering it onto the surface contacts, bending it towards the board over the Scorpion writing, and gluing it to the PCB;
- the green screw terminals are prone to become unfastened during combat, enabling the wires to fall off; we've removed these terminals and soldered the wires directly to the board;
- the flip feature doesn't work in the 2006 versions that we've bought; the solution, if available, is to implement this function using mixes in the radio transmitter;
- to improve heat dissipation and stretch the current limits a little bit, it is possible to attach heatsinks onto the BTS chips; old CPUs are a good source for small aluminum heatsinks.

Scorpion HX

The Scorpion HX (pictured to the right) is a great option for lighter combots such as beetleweights and antweights. The board weighs only 0.78oz (22g), with a compact 1.6" × 1.6" × 0.5" size.



Unlike the XL/XXL controllers, this board features 3 channels. Two of them are PWM outputs, usually used to control drive motors, and one on-off switch (Aux/Weapon channel) that can be used to operate a brushed DC motor in a single direction, usually in the weapon system.

The PWM channels accept motors between 4.8V and 22V, delivering each 2.5A continuously with 6A peaks. If you need a few extra amps, it is possible to attach a heatsink onto the drive chips. Regarding the Aux/Weapon channel, it can handle constant 12A and as much as 35A peaks.

Other characteristics include flip (which works very well), current and temperature protection, channel mixing, BEC (which must be deactivated for input voltages above 12V), and safe weapon channel start (where the robot won't start if the weapon switch is activated during power-up), besides failsafe in all channels.

Mini-Touro used a Scorpion HX ESC during RoboGames 2006, with great results. However, in 2007 we’ve decided to use a brushless motor to power its drum, instead of a brushed one, so we had to replace the HX board with two BaneBots BB-3-9 ESCs (for the drive system), to give room for the BESC needed by the weapon motor.

Scorpion Mini

The Scorpion Mini (pictured to the right) is the smallest ESC from Robot Power, weighing only 0.21oz (6g) and measuring 0.625” × 1.6” × 0.4”. It is a single channel PWM H-Bridge, which takes input voltages from 4.8 to 18V, or up to 34V if an external 5V supply is used. Current limits are 2.5A continuous without a heatsink, up to 4A with one, and 6A peaks.

Its version 2.4 includes BEC, which must be disabled if using voltages above 16V, and failsafe. It also features limit switch inputs that stop motion when pressed, which makes it an interesting choice to actuate clamper or lifter weapon systems in insect combots.

7.3.4. BaneBots

BaneBots is another manufacturer with ESCs aimed for light robot classes, from fairyweights to hobbyweights. There are three versions, all featuring a single fully reversible PWM output that can handle inputs from 6 to 24V: the BB-12-45, the BB-5-18, and the BB-3-9, pictured below from left to right, respectively.

The larger unit, the BB-12-45 (US$57), can withstand 12A continuously and 45A peaks (thus its name 12-45), presenting current limiting and thermal protection. It also counts with an integrated

BEC, failsafe, and neutral start. It also includes receiver cables and leads to the motor and power supply. It weighs 0.98oz (26.3g) and measures 1.7” × 1.1” × 0.5”, including the orange shrink wrap. It is a good option for most hobbyweights.

The BB-5-18 (US$ 46.50) is the intermediate model, it can withstand 5A continuously and 18A surges (thus its name 5-18). Similarly to the BB-12-45, it includes all wires, it has an integrated BEC, failsafe, and neutral start. It measures 1.45” × 0.825” × 0.375” and weighs only 0.60oz (17g).

The tiniest model is the BB-3-9 (US$28.75), it only weighs 0.33oz (9.4g). Its dimensions are 1.2” × 0.52” × 0.29”, allowing it to fit in small spaces inside the robot. As its 3-9 name suggests, it can handle 3A continuously and 9A peaks.

One great feature that is common to all BB speed controllers is their modularity. Single channel units are easier to fit inside the robot, because several different orientation combinations can be experimented. In addition, if for instance a wheel locks up and damages your ESC, there is no need to replace the entire electronic system, only replacing the ESC associated with that channel.

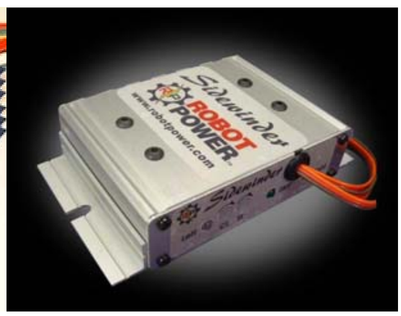
We use BaneBots speed controllers in our lighter combots: the hobbyweight Tourinho uses a pair of BB-12-45 for its drive system, while the beetleweight Mini-Touro, the antweight Micro-Touro, and the fairyweight Pocket, use each a pair of BB-3-9 ESCs.

Despite their advantages, BaneBots ESCs aren’t fail-proof. The BEC circuit heats up very easily, leading to unexpected failures when used at 24V. One of our BB-12-45 caught fire while tested at 24V drawing only 500mA, due to the BEC issue. It’s probably a good idea to disable their BEC if you’re using input voltages near 24V, which might require you to install a UBEC, discussed in section 7.1.5. Another issue is that the names from all of its electronic components have been sanded out by the manufacturer, making any homemade repair impossible.

7.3.5. Other Brushed Motor Speed Controllers

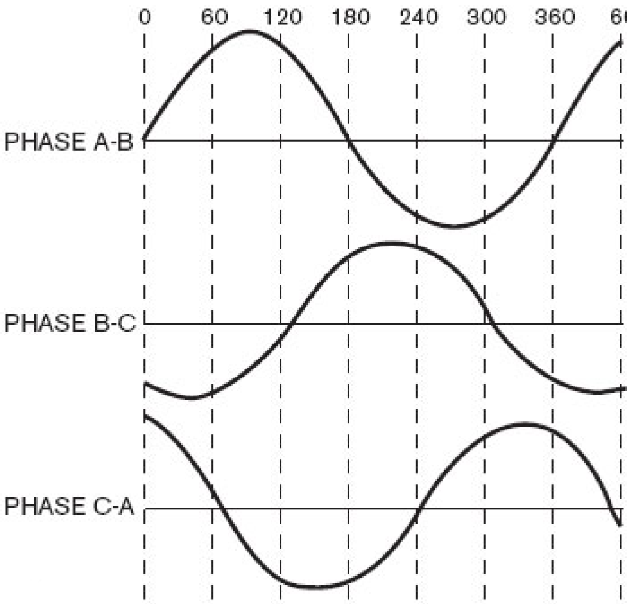
There are several other speed controllers in the market. A few of them are very sophisticated, including several programmable features, input voltages of up to 48V with peak currents exceeding 100A, tachometer and potentiometer inputs for closed-loop speed and position control, regenerative braking, temperature sensors, RC and microcomputer interfaces, and much more.

A few examples of sophisticated controllers are the RoboteQ, AmpFlow and Sidewinder, pictured below. They can independently control 2 brushed DC motors at the same time, which is perfect for the drivetrain of most combat robots. They also have mounting brackets for easy installation. They are good options for middleweights, heavyweights, and super heavyweights.

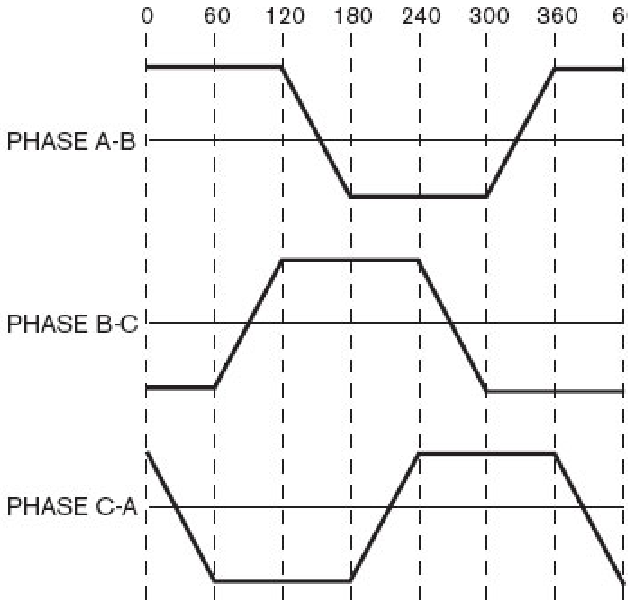


7.3.6. Brushless Electronic Speed Controllers

As seen in chapter 5, brushless DC (BLDC) motors demand a specific controller to be powered. This kind of motor resembles an alternate current (AC) synchronous three-phase motor, in which three waves with 120° shifted phases and same frequency actuate the rotation. Similarly to AC motors, brushless motors also spin due to three 120° phase-shifted waves but, instead of sinusoidal, these waves are three trapezoidal PWM signals, as pictured below. Each PWM signal acts over each one of the three motor windings from the brushless motor.

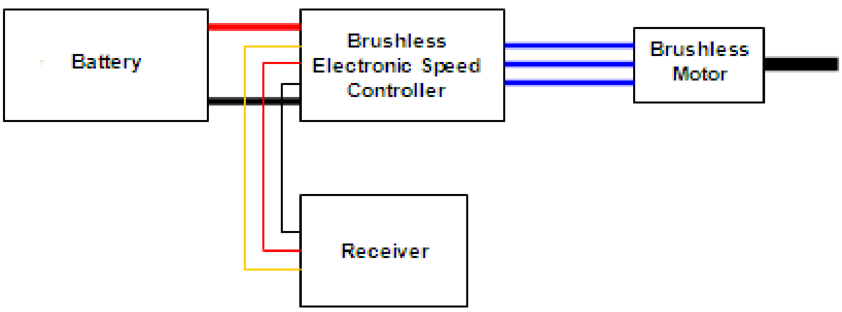


sinusoidal inputs for AC three-phase motors



trapezoidal PWM inputs for brushless motors

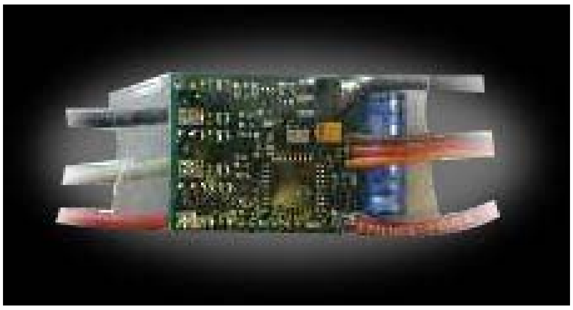
The wiring scheme to control brushless motors using a BESC is a little different from the one from brushed DC motors controlled by an ESC, as pictured to the right. Never connect the BESC to the battery with reverse polarity. Note that the BESC can be connected to the brushless motor using any combination of the motor's 3 wires. However, if the brushless motor spins in the wrong direction, then simply switch connection between any 2 wires. If, when the motor is unloaded, it takes too long to start spinning, you could try other wiring combinations between the BESC and the motor, this might solve the problem.



In order to start the motor, the controller must know which of the three windings should be first triggered. There are two methods to do that. The first is to have position sensors on the motor to measure the angle of the rotor, however brushless motors with sensors usually have high cost and complexity.

The other method is sensorless, where the controller is able to figure out the angle of the rotor by measuring the differences between the windings inductances. This is why you should never shorten brushless motor wires, this would change their inductances and confuse a sensorless system. Sometimes the motor wires are the coil wires extended out, so if you don't cut them at exactly the same length, getting them cleaned up nicely, removing all the varnish, and evenly tinning them, then your motor will single phase and not work, but just chatter. So, leave the motor wires alone.

On the other hand, long wires between the battery and the BESC should be avoided. The inductance of such long wires would cause voltage spikes on the BESC's input power lines due to its voltage switching. This is why most BESCs (and ESCs as well, for brushed DC motors) have a capacitor between their input power lines: its role is to suppress voltage spikes that could fry the speed controller. So, it is a good idea to shorten the wires between the battery and the BESC, placing these components close together inside the robot, and leaving the brushless motor wires with their original length. Long motor wires only lose a bit of power, they don't damage the BESC.



There are several sensorless Brushless Electronic Speed Controllers (BESC) in the market, from various manufacturers, such as Castle Creations, Hextronik, and Dynamite. The picture to the right shows the Castle Creations Phoenix 25 BESC, used in the weapon system of our beetleweight Mini-Touro.

The downside of most BESCs is that they can only spin the motor in one direction. To reverse rotation during combat, you can commute any two of the motor wires, which can be done, for instance, using traditional or solid-state relays. Or you can use a reversible BESC, usually developed for R/C model cars, which can reverse the spin direction of brushless motors using a three phase H-Bridge circuit.

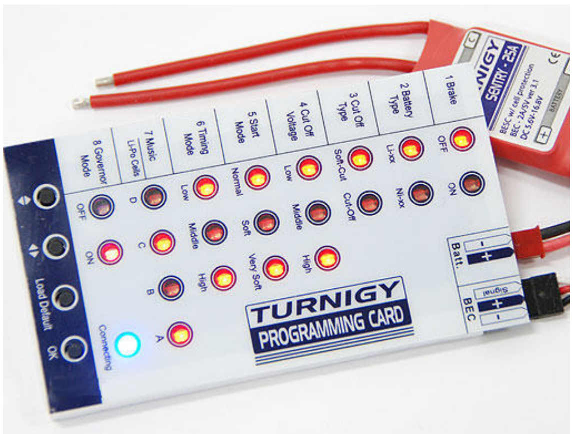
A few important aspects when choosing a BESC are its maximum voltage capacity, maximum continuous and peak currents, and motor reversion ability. Brushless motor reversion is still uncommon in combots, however it is very useful to reverse the direction of a weapon, allowing for instance an invertible drumbot that was flipped over to reverse its drum to continue launching the opponent. Brushless motor reversion has also potential applications in the robot drivetrain, resulting in a very efficient drive system with a high power-to-weight ratio.

A usual feature found in most BESCs is a BEC to power the receiver. But most BECs found in BESCs use linear regulators, which may overheat at high voltages. You may need to disable the BEC if the battery voltage is high. For instance, the HXT120 BESC from Hextronik (pictured to the right) can handle 120A continuously if properly ventilated, at input voltages of up to 24V. The BEC can handle up to 2A, but only if the input voltage is below 12V, otherwise it will overheat. For voltages higher than 12V, the BEC needs to be disabled, which is done by removing the middle red wire from the receiver connector. We use this BESC to power the weapon motors from our featherweight Touro Feather and hobbyweight Touro Jr.

High quality BESCs usually feature soft start, to prevent damages to fragile gearboxes, along with several programmable features such as:

- low-voltage cut-off: if the battery voltage per cell drops below a certain threshold, the motor speed is reduced or even brought to zero, to prevent damaging lithium batteries; the problem with this feature is that combat robot weapons usually require high current bursts for short periods, which can momentarily lower the battery voltage below the threshold, turning your weapon off; the default threshold settings are usually too conservative to use in combat, so program this feature to the lowest allowed value;
- over-current protection: if the current gets higher than a programmed threshold, the motor is turned off; so, choose the highest possible value for this threshold, if your motor can take it;
- brake type: option to brake the motor when the radio transmitter stick is in neutral position; this can be useful to stop spinning weapons in less than 60 seconds, as required in most competitions; note that the entire energy is dissipated in the motor, which might overheat;
- throttle range: option to limit the maximum power output, which is an interesting option in case the motor is overheating and/or drawing too much current from the batteries during combat;
- timing advance: ability to advance motor timing, resulting in a higher top speed in one direction;
- reverse delay: if a BESC is capable of reversing the motor by itself, without the aid of any external system, then it will probably feature this option; it defines a time delay between the moment when the radio control stick is moved to a reverse position and the moment when the spin reversion is fully commanded; this feature is used to prevent damages to gearboxes from the impact caused by a sudden reversion; if you need a very fast reversion, then set this delay to the lowest possible value, however keep in mind that the entire braking energy will need to be dissipated by the motor, which can end up overheating.

BESC programming methods vary among manufacturers and models. High-end models can be programmed via USB (using an appropriate connector, such as the one pictured to the right) in a computer. This is a really nice feature, making it possible to tweak the performance on the fly using computer software with intuitive user interfaces.



Cheaper models are usually programmed using the radio transmitter throttle stick, with the aid of feedback beep sounds emitted by the BESC, which can be quite confusing (and very annoying).

Other BESCs allow the use of a special programming card, such as the one pictured to the right, which significantly eases the task. It is plugged between the receiver and the BESC, allowing you to program it according to a series of LED indicators. The card usually has buttons at the bottom to navigate through the menu, select the desired feature, and then effectively change it in the BESC.

7.4. Solenoids

High power motors will probably require expensive speed controllers to be activated. A few motors, such as the D-Pack, are so powerful that they can even blow tough ESCs such as OSMCs, if care is not taken. Using your ESC in combat near its operational limit is risky, not to mention that its life will be significantly shortened.

So, if you don't really need speed or direction control, such as in the spinning weapon motor of most combots that have self-righting mechanisms, then solenoids might be a good option. Solenoids, a.k.a. contactors, are basically relays on steroids, capable of handling high currents to operate powerful weapons. Two of the most famous solenoids used in combots are the White-Rodgers 586 SPDT, and Team Whyachi's TW-C1, discussed next.

7.4.1. White-Rodgers 586 SPDT

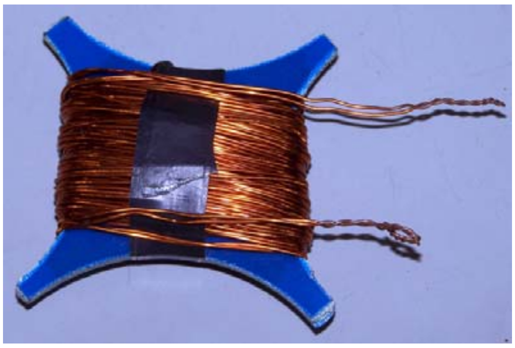
The White-Rodgers 586 SPDT (pictured to the right, sold at The Robot MarketPlace for US$96) handles 200A continuous and withstands peaks higher than 600A, being, therefore, appropriate for almost any combot weapon even in super heavyweights. It is the solenoid used to power the Etek motor from our middleweight horizontal bar spinner Ciclone.

The SPDT in its name stands for Single Pull Double Throw, in other words, a single signal can switch between the Normally Open (NO) and Normally Closed (NC) terminals, necessary to activate or brake the weapon.

Most competitions require a robot's weapon to stop in less than 1 minute, therefore spinners with large weapon inertia might need some sort of braking system - it is not enough to just turn the motor off. A very simple braking system can be implemented in a brushed DC motor by shorting its leads. A spinning motor that is shorted out will become a generator, the same principle used in power plants, producing a current that is dissipated by the internal resistance of the motor, converting its kinetic energy into heat. This will effectively brake the system, as it can be easily verified by turning by hand the shaft of a motor with or without its leads shorted out. It will be much harder to turn the shaft when the leads are shorted.

Therefore, to implement a braking system, it is enough to connect the motor to the solenoid NO terminals, while shorting the terminals from the NC pair passing through the motor. When the solenoid is activated, current will flow through the NO terminals and the motor will accelerate. When deactivated, the motor leads will be shorted out at the NC terminals, braking the motor.

However, the kinetic energy of a few weapons, such as spinning bars or disks, is so high that shorting out the motor would simply fry it, due to the entire energy being dissipated as heat by the motor's usually small internal resistance. To prevent this, a power resistor must be placed between the NC leads, instead of just shorting them. Its value can be, for instance, 9 times the internal resistance of the motor, which would result in 90% of the weapon's energy being dissipated by this resistor, and only 10% by the motor. Note, however, that higher resistance values mean a longer braking time, because the dissipated power is inversely proportional to the resistance of the circuit. Therefore, choose a power resistor with a high enough resistance to avoid frying the motor, but not too high to guarantee that the weapon will stop in significantly less than 60 seconds.



Instead of a power resistor, it is possible to use a long coated copper wire that is wound up on some heat resistant material (as pictured to the right, wound up around a blue piece of garolite). Knowing the wire resistivity, you can calculate the needed length for the desired resistance. Then, wind it up as a reel to keep its size very compact. The advantages of this braking system are its low cost, its low weight, and the efficiency of the heat dissipation provided by the long wire length. It is also easy to change its resistance: if the weapon is taking too long to stop, then just cut a few feet of wire from the reel to lower its resistance. It is important to use a heat resistant material such as garolite, because the temperatures while braking a powerful weapon can reach very high values. The blue garolite from our braking system usually comes out with a burnt-black color after a competition. In addition, never mount this braking reel close to the electronic system or to any flammable material.

We have used this wire braking solution in our middleweight horizontal bar spinners Ciclone and Titan. It works really well. Touro’s drum doesn’t need a brake, despite the weapon’s high kinetic energy, because the moment of inertia of the drum is not as high as the one from the spinning bars. Even if two spinning weapons have the same kinetic energy, the one with lower moment of inertia will most likely stop faster, because it will have a much higher spin speed (to result in the same energy) that will result in larger bearing friction. Touro’s drum stops in a few seconds, without the need for a braking system. Drumbots in general do not need a braking system to stop under 60 seconds. But horizontal and vertical spinners, which have weapons with very large moments of inertia, might need it. A few powerful spinners hit the arena walls after the end of a match to slow down and stop their weapon in much less than 60 seconds, to avoid delaying the event - but you must check if this is allowed, especially with the arena’s owner!

7.4.2. Team Whyachi TW-C1

Team Whyachi’s TW-C1 contactor (pictured to the right) is also SPDT, however it is smaller, lighter and cheaper than White-Rodger's 586. It tolerates, at 48V, currents of 80A continuous, 240A for 3 minutes, and 500A for 25 seconds.

Until 2008, both Touro and Titan had TW-C1s powering their weapons. The only issue with both presented solenoids is their plastic casing, which can break due to the high impact accelerations. Therefore, always shock-mount solenoids inside the robot structure.

7.5. Wiring

To connect the previously presented components, high quality wires and connectors are needed. The wires must bear high currents, while the connectors can't become loose during impacts. These components are presented next.

7.5.1. Wires

Wires must be very flexible, making it easy to route them through the robot's inside without rupturing the solders during impacts. Therefore, never use cables with a solid metallic core, use instead cables with multiple wires. A good example is the Deans Wet Noodle (pictured to the right), formed by over a thousand extremely thin wires. Also, it is important to leave a little slack in all wires, in order to avoid them from getting stretched, ruptured or disconnected during combat, especially if the robot suffers structural deformations or if internal parts slightly move.

It is important to keep in mind the current ratings, which depends on the wire diameter (gauge, usually measured in AWG, which stands for American Wire Gauge) and the isolation material. The higher the AWG, the smaller the wire diameter is. When it comes to isolation, there are two usual types: PVC, which withstands temperatures up to 221°F (105°C), and silicone, withstanding up to 392°F (200°C). The highest current ratings for typical wire gauges are the following:

- 8 AWG: 70A to 80A continuous (PVC); 100 to 110A continuous and 500A peaks (silicone);
- 10 AWG: 50 to 60A continuous (PVC); 75 to 85A continuous and 350A peaks (silicone);
- 12 AWG: 35 to 45A continuous (PVC); 55 to 65A continuous and 200A peaks (silicone);
- 14 AWG: 30 to 35A continuous (PVC); 45 to 50A continuous (silicone).

The picture to the right shows a typical device used to measure wire gauges, ranging from the 0.3249" (8.252mm) diameter 0 AWG to the 0.005" (0.127mm) diameter 36 AWG.

Note that most wires can withstand without problems very brief and sporadic current peaks that are 4 times higher than the continuous limits.

A good tip is to use zip-ties to organize the wiring inside the robot. This can save a lot of time during a pitstop.

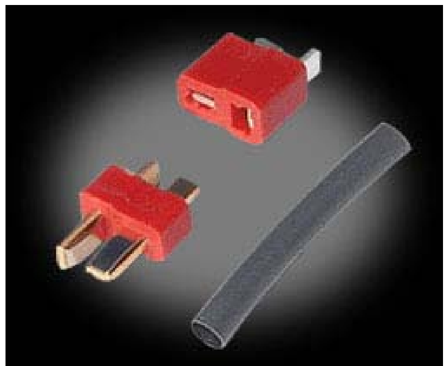
Also, always use rubber grommets (pictured to the right) to protect wires that need to go through metallic plates: if not protected, the friction between the wires and a hole in a metallic part can cut the isolation layer and cause short-circuits. Smoothing out with a metal file the borders of the hole is also a good idea.

7.5.2. Terminals, Plugs and Connectors

Terminals, plugs and connectors are critical in combat, because they must withstand impact accelerations and high currents. Avoid using fork, slide or quick connection terminals, since they are prone to disconnect upon impacts. Always use ring terminals (pictured below to the left). Fasten tightly the connectors with nuts, along with a pressure washer. Never place a washer between the contacts, because it usually has a large electric resistance. Also, apply liquid electrical tape (as pictured in white to the right, applied to a White-Rodgers 586 SPDT solenoid) to avoid shorted contacts due to metal debris.

It is a good idea to stick several pieces of double face adhesive tape inside the robot, especially on the inner part of the bottom cover, near the electronics. If the tape is very sticky, such as the VHB4910 tape, it works as a “flypaper” to pick up any debris that enters the robot, which could cause problems such as shorting out the electronics or getting stuck in the clearances between the wheels and the structure. We always use this technique: at the end of a competition, our VHB4910 tapes are filled with metal chips, small bolts and dirt. The tapes are replaced before every event.

Connectors must have a very low resistance and lock in well. An excellent connector is the Deans Ultra (pictured to the right), which withstands continuous 80A. Their maximum peak current is much higher: Touro’s weapon motor draws almost 300A for a couple of seconds in the beginning of the drum acceleration, as we’ve measured, which goes through a single Deans Ultra without problems. We use these connectors on the batteries, motors and ESCs from all our hobby, feather, light and middleweight combots. An extra protection is, after connecting them, to duct tape them together to make sure they won’t get disconnected. Be careful with knock-offs with cheap plastic (not nylon) housings that easily melt during soldering, letting their contacts come loose.



Another high power connector we’ve used is the Anderson PowerPole. The most common are the 45A version (pictured to the right) and the 75A. The 75A tolerates higher current peaks than the Deans Ultra, however it is a little bulky.

For lower currents, we use the Deans Micro Plug (pictured to the right), which is much smaller but even so tolerate currents higher than 20A. We use them as fan connectors in the bigger bots, as well as to connect the brushed DC motors from our beetleweight Mini-Touro.

7.6. Power Switches

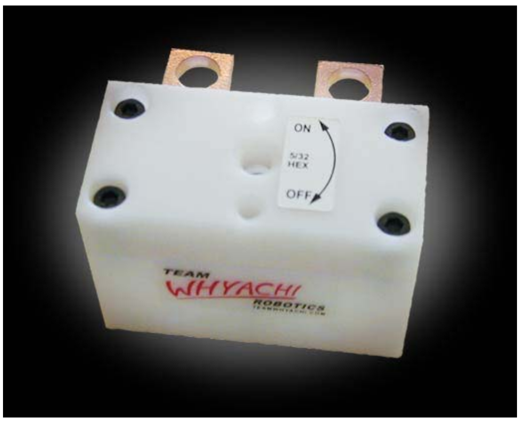
Power switches, or other on/off mechanisms, are mandatory in every combot (except, in a few cases, in insect class robots without active weapons). The switch must interrupt any current flux inside the robot, even low power ones. It is not enough to be able to switch off the RC interface or the receiver, even if this is enough to stop your robot, because any malfunction could still activate your weapon or drive system. It is necessary to be able to completely disconnect all batteries from every system.

For this reason, such switches must handle the entire amperage that goes through the robot, including the current required by the drivetrain and weapon systems. In the case of our middleweight Touro, current peaks can reach up to 400A.

There are automotive switches that can handle this level of current, however they are usually bulky and heavy. Two small and light weight switches, popular with combot builders, are the Hella Master Power Switch and the Team Whyachi MS-2.

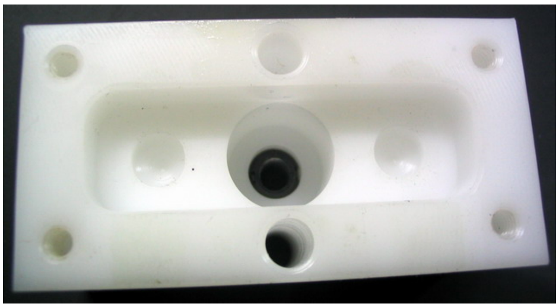
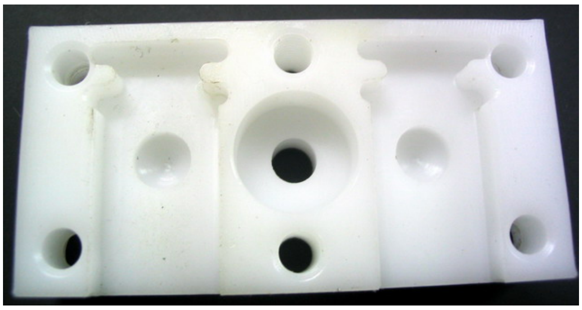
The Hella switch (US$18, pictured to the right) can take continuous 100A, 500A for 10 seconds, and 1,000A current peaks. It is turned on or off by a red key that is not very convenient to use in combat. Most builders cut off the head of the red key, and file a notch on the remaining stub (as pictured to the right), allowing it to be switched on or off using a flathead screwdriver. We have used this switch in our middleweights Ciclone and Titan.

Team Whyachi's MS-2 switch (pictured to the right) is better and smaller than the Hella switch, however it is more expensive (US$65). It withstands continuous 175A, 500A for 3 minutes, and 1,000A for 25 seconds. To turn the robot on or off, you need to insert an Allen wrench (also pictured to the right) and turn it four times in the appropriate direction. These four turns make it almost impossible to have the switch disconnected due to vibrations during combat.



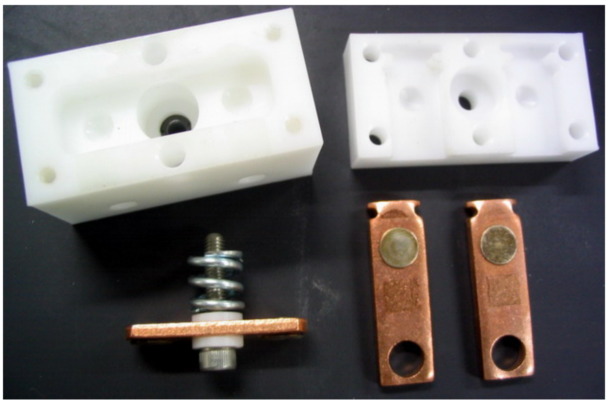
There is a smaller version that is rated for a lower current, the MS-05, good for hobbyweights (12lb), however it isn't much cheaper, at US$45. It takes continuous 40A, 140A for 3 minutes, and 250A for 25 seconds. In high power featherweights or heavier combots, it is better to use the MS-2, due its higher current rating. Our featherweight Touro Feather, which has a PolyQuest lithium-polymer battery capable of delivering up to 225A, had a few problems with the MS-05; since we've replaced it with the MS-2 switch, we haven't had any more problems.

The MS-2 and MS-05 switches seem a little pricey, but they have an excellent cost-benefit relation considering that they are a vital part of the robot. And their manufacturing is not so simple, it involves the milling of two Delrin blocks, one for the body of the switch and the other for its cover, as pictured below.

milled Delrin body of the switch (inner and outer views)	milled Delrin cover of the switch (inner and outer views)
 	 

The Delrin body has an embedded threaded nut (in black at its center), to which a long Allen screw is attached, along with a copper bar with gold contacts, two plastic non-conductive washers, and a spring, pictured below to the left. After the switch is assembled, the long Allen screw will have the function to position the copper bar to close or open the circuit. As it is screwed, it will drive the copper bar away from the two copper terminals (pictured below in the center), opening the circuit and turning the robot off. When the long Allen screw is unscrewed, the spring-return will make the copper bar touch the two copper terminals, turning the robot on.

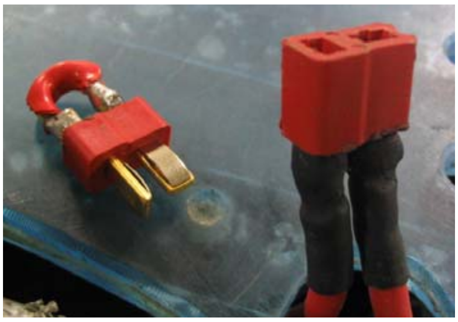
The picture below to the right shows the assembled body and cover, before they are attached together using 4 small Allen screws. The switch itself is attached to the robot with the aid of two threaded holes in its Delrin body.

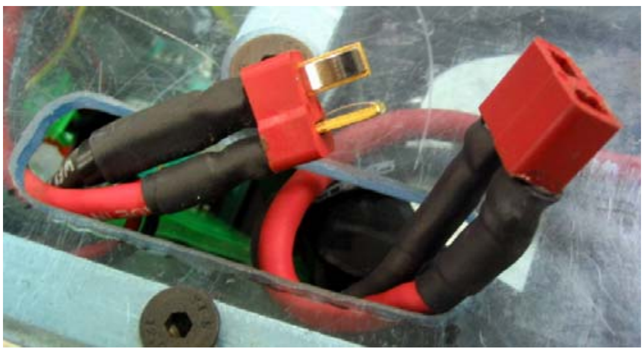
Note that the non-conductive washers are important to electrically isolate the long screw from the copper bar, besides reducing friction between them.

Note also that the spring is important to guarantee contact between the copper bar and the terminals. It also works as a spring-lock to avoid the screw from turning due to vibration. And, if the switch breaks due to an impact, it is likely that it will remain in the "on" position due to the spring, keeping the robot alive during a violent match. This feature has saved us during RoboGames 2006: the violent impacts during the match against the undercutter The Mortician managed to detach the long Allen screw from the threaded nut in the Delrin body of Touro's MS-2 switch; however, Touro continued to fully function because the spring was able to guarantee contact between the terminals.

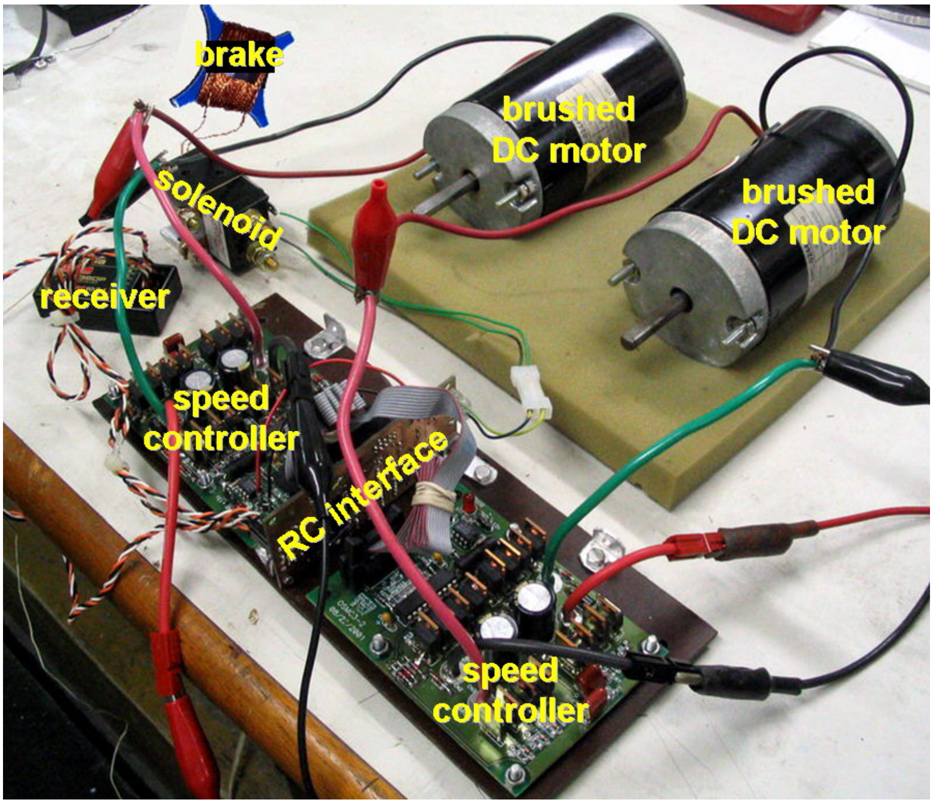
There are also even simpler and cheaper switches that can be made. One of them is the one used in the drive system of our middleweight Ciclone. A Hella switch controls the weapon, at 24V, but the drive motors use an 18V cordless drill battery pack, which needs a second switch. The adopted solution costs only a few bucks: a pair of Deans Ultra connectors wired as a jumper, as shown to the right while turned off (left picture) or on (right picture). The wire that is connected to the battery positive was cut in two, opening the circuit. Then, the two cut pieces were soldered to a female Deans Ultra plug. Next, a short wire was soldered to both terminals of a male Deans Ultra plug. Don't forget to isolate well both plugs (the Deans jumper in the pictures is shown without its isolation tape). Connecting the male into the female plug closes the circuit and turns the robot on. It's important to insert the connected plugs well inside the robot, to avoid them from being knocked-off by an opponent. The bar spinner The Mortician was able to win a RoboGames 2006 match by knock out after knocking off a jumper switch from the launcher Sub Zero (as pictured to the right), even though the jumper was almost completely inserted into the robot.



An even simpler solution is to not use a switch. The robot must have an opening that allows the driver to directly connect or disconnect the battery (or batteries). This is the solution used in our hobbyweights (as pictured to the right) and smaller robots. A Deans Ultra female connector is soldered to the battery leads (never solder the male connector to a battery, to avoid accidental shorts). The robot electronics uses a male Deans Ultra. To turn the robot on, just connect the plugs, insert them into the robot opening, and cover it to protect against debris. Since this must be done by the driver in the arena, make sure that the cover can be easily attached to the bot.



7.7. Connection Schemes



In this section, it is shown how to connect the presented components in a combat robot. A typical configuration is pictured to the right.

In this picture, the receiver converts the radio signals into a PPM signal, which is then interpreted by the RC interface (which is described in details in section 7.8.2).

This RC interface then generates low power PWM signals that are sent to the speed controllers, such as OSMCs (as shown in the picture), which amplify them to a high power PWM output to control the speed of the brushed DC motors used in the robot drivetrain.

The RC interface is also able to activate the solenoid that triggers the weapon motor (not shown in this picture), using a copper wire reel connected to the Normally Closed (NC) terminals to brake the weapon.

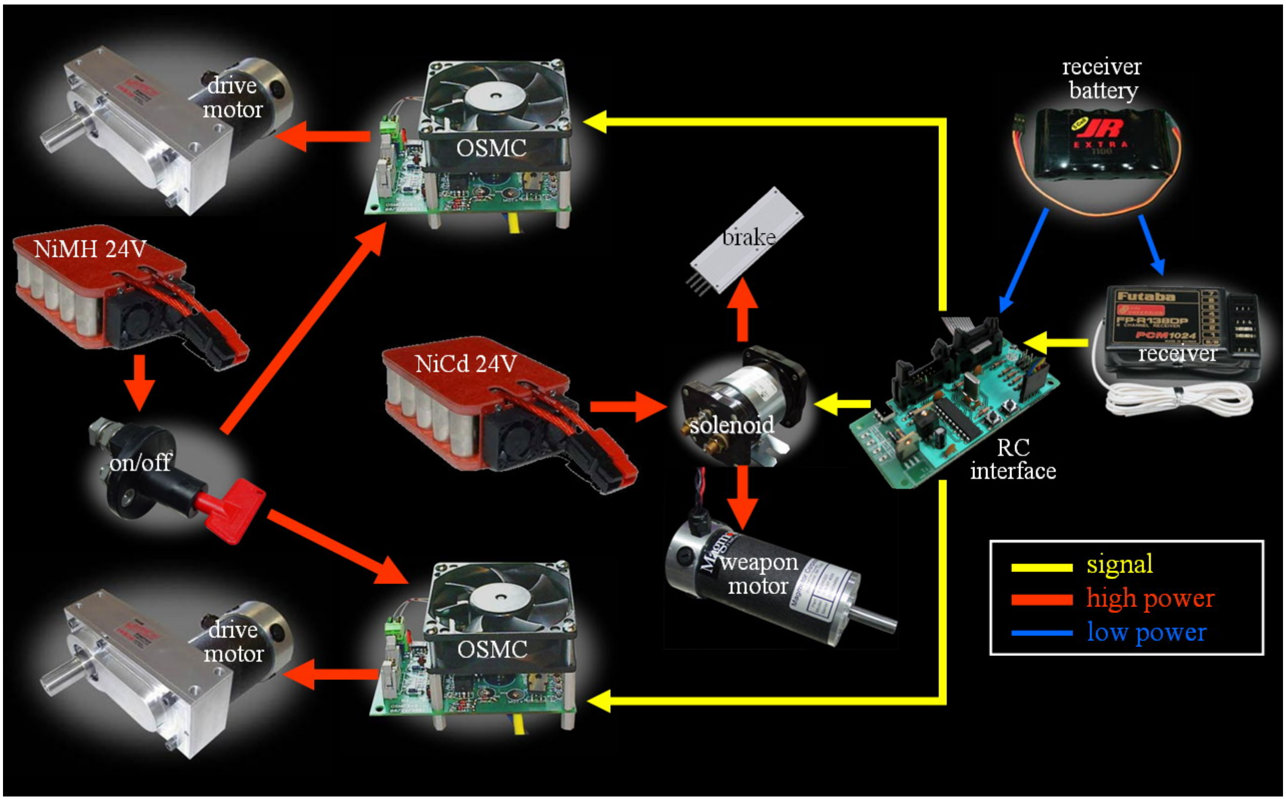
Three connection schemes are presented next. The first is a classic scheme, often used by beginners, which will surely not pass safety inspection. The second one is an improved version, which addresses all the issues from the first one. Finally, a third connection scheme is shown, better than the second one if you need a fully reversible weapon with speed control.

7.7.1 Classic Connection Scheme

The figure in the next page shows a classic connection scheme, including components that are sized to a middleweight combot.

It uses one Nickel-Cadmium (NiCd) battery pack to power the weapon, which is a good option due to its ability to provide very high peak currents (as it will be studied in chapter 8).

And it uses one Nickel-Metal Hydride (NiMH) battery pack for the drivetrain, which is also a good option because it has more capacity than NiCd, lasting longer (the current peaks from the drivetrain are usually much lower than the ones from the weapon system, because wheel slip acts as a torque limiter).



In the configuration shown in the scheme, the NiMH pack is connected to a Hella switch, which powers the OSMC speed controllers. The NiCd pack is connected to a White-Rodgers 586 SPDT solenoid that activates the weapon, while a power resistor is used for braking. A small battery pack powers both the receiver and the RC interface. The RC interface interprets the PPM signal from the receiver, sending a low power PWM signal to each OSMC. Each OSMC amplifies the received signal, sending a high power PWM output to the drive motor it is connected to. The RC interface is used as well to trigger the solenoid that powers the weapon motor.

Note that this RC interface only works if powered by both 5V from the small battery pack and 12V from the OSMCs. Therefore, if the Hella power switch is off, the NiMH pack won't provide 24V to the OSMCs, which in turn won't provide 12V to the RC interface, which in turn won't be able to keep the solenoid active, turning the weapon off. In theory it would work, but it is unsafe.

The above scheme looks good, including the battery optimization feature: NiCd for the weapon, to deliver high peak currents, and NiMH for the drivetrain, for improved capacity. However, it has serious flaws:

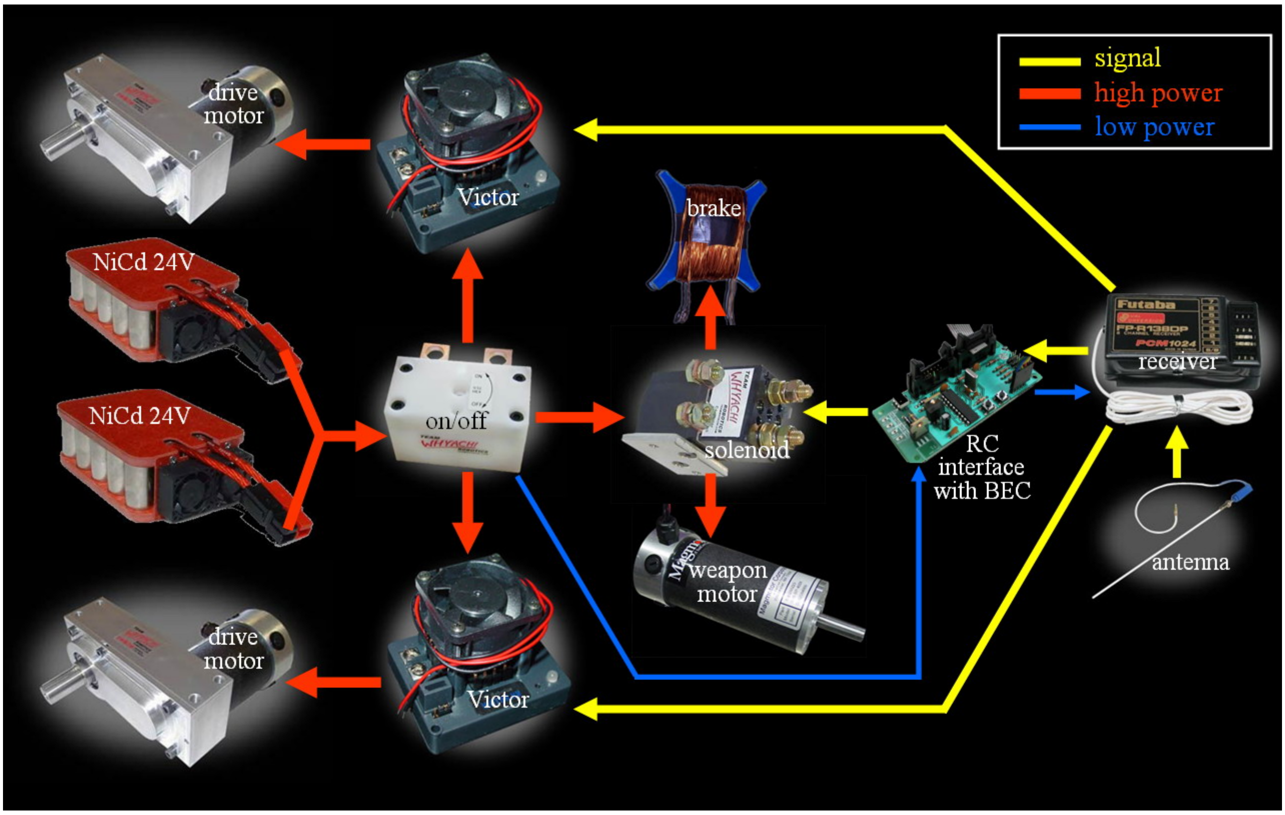
- if the single NiMH pack breaks, the robot will stop working and, therefore, lose the match;
- if the small battery pack voltage is too low, the robot will become unresponsive;
- there isn't a power (on/off) switch between the NiCd pack and the weapon solenoid, thus if due to a surge current the solenoid terminals get soldered, the weapon won't stop, even with the main switch off – therefore this scheme won't pass safety inspection; you would need to include another switch between the NiCd pack and the solenoid;
- there isn't an on/off switch between the small pack and the receiver and RC interface, which is

required to pass safety inspection; note that even the fans used in the robot must be turned off by the robot switch(es); therefore, the robot would need 3 on/off switches, because you can't connect the three battery packs in parallel due to their different types or voltages; the driver would need then to remember to turn on all three switches in the beginning of every match;

- the White-Rodgers 586 SPDT Solenoid is relatively large and heavy;
- OSMC speed controllers are not very compact, occupying a lot of the internal space;
- a resistor with both low-resistance and high-power, typically with less than 1Ω and more than 1kW for middleweights, needed to brake the weapon, isn't cheap and it can burn.

7.7.2. Improved Connection Scheme

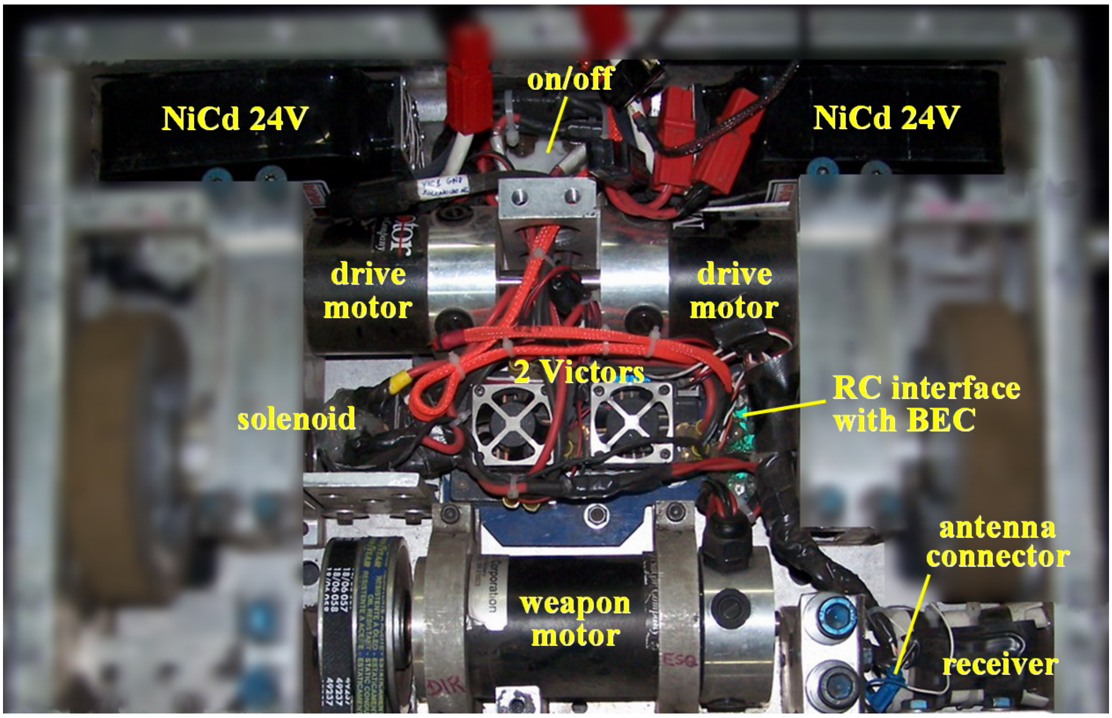
To solve the problems presented above, you should use an improved scheme, such as the one pictured below. It includes 2 (or more) identical battery packs in parallel (NiCd in this example), connected to a single MS-2 power switch. A second power switch could be used in parallel to both packs, as a redundancy measure in case one of the switches breaks in the off position. Both packs need to be exactly the same, with same type, voltage and capacity, to be connected in parallel without any problems. This is why we use 2 identical NiCd packs. This switch powers the Victors, the TW-C1 solenoid, and the RC interface. This interface, needed to activate the solenoid, has a built-in BEC to power the receiver. The Victors can be directly connected to the receiver without an RC interface. A copper wire reel is connected to the solenoid to act as a weapon brake. A Deans Base-Loaded Whip antenna is attached to the 75MHz receiver, enhancing reception quality.



This improved scheme addresses all the issues from the classic scheme, because:

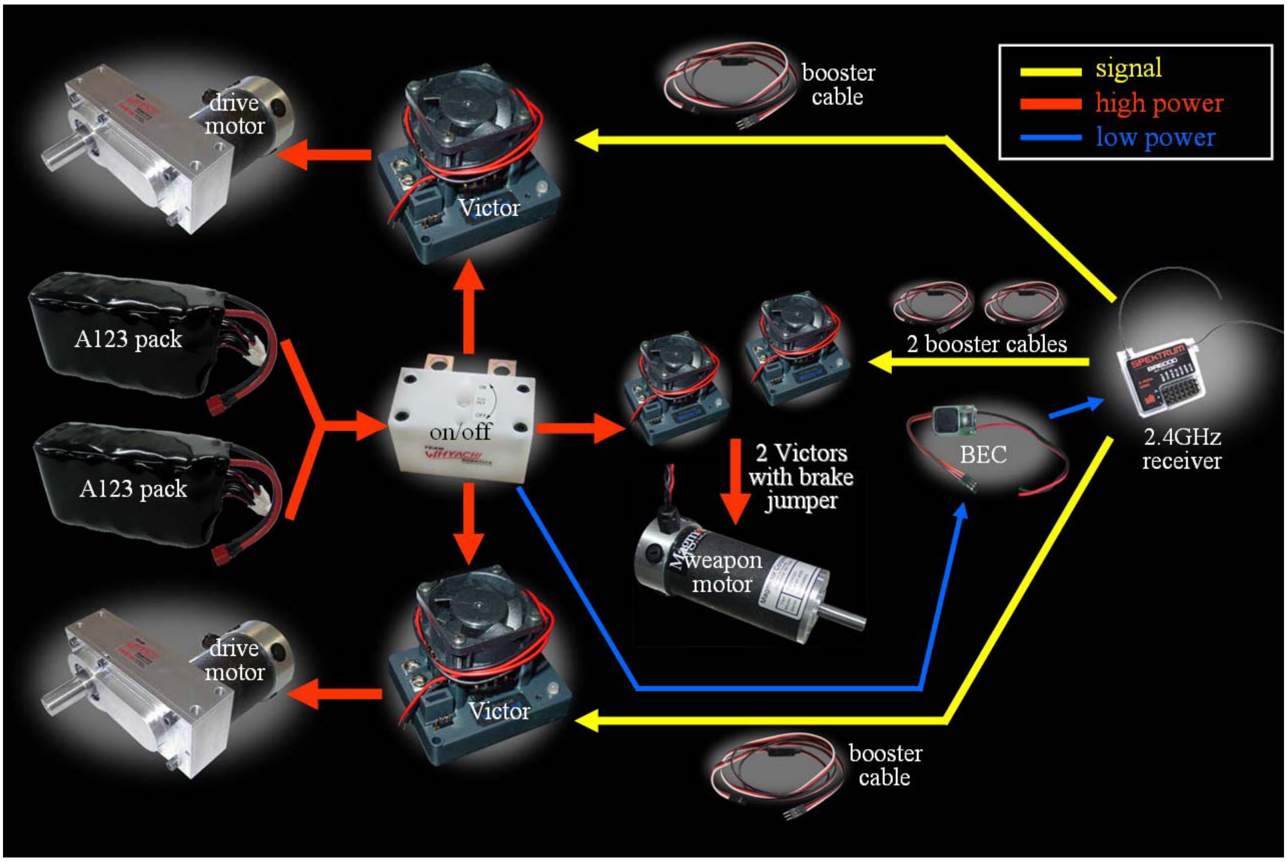
- if one of the batteries fails, due to a broken solder or connector malfunction, the robot will continue to fully function using the remaining pack(s), however with lower total capacity;
- the weapon will accelerate faster, since two packs in parallel can supply twice the current, assuming that the drivetrain isn't demanding too much from them during this acceleration;
- there is no need to have a small battery pack for the receiver, due to the RC interface BEC;
- a single on/off switch can power down the entire robot, including drivetrain and weapon motors, as well as the receiver and RC interface, which is required by safety inspections in the events;
- the TW-C1 solenoid is smaller, lighter and cheaper than the White-Rodgers 586 SPDT;
- Victor speed controllers, besides smaller than the OSMC, can be directly connected to the receiver, without needing the RC interface (which is only used above to power the solenoid and to work as a BEC to power the receiver);
- Victors have a brake/coast jumper, used to set its action during a neutral condition from the radio; the brake setting sets the output to a short-circuit during neutral, while the coast setting sets an open circuit; when used in the drive system, the brake setting will stop your robot when you release the radio control stick, while the coast setting will let your robot continue moving due to its inertia; the brake setting is a good option for sharp turns in agile robots, while the coast setting is good to prevent the drive motor from overheating due to the short-circuits;
- the RC interface can become smaller, because it only needs to actuate the solenoid and to work as a BEC, without any need for PWM outputs for the drive system;
- the copper wire reel is cheaper and it dissipates heat better than the power resistor.

The presented scheme is also pictured below, showing a close up of the electronic system of the 2006 version of our middleweight Touro. Note that all components from this improved scheme are included below, except for the weapon brake, usually not needed in drumbots.



7.7.3. Connection Scheme for Reversible Weapons

After RoboGames 2008, we've decided to try a new connection configuration, one capable to reverse Touro's drum rotation and to solve minor issues from the previous schemes. To improve the battery capacity and voltage, the NiCd packs in parallel were replaced with A123 packs. Instead of using an RC interface and solenoid, two Victors are used to control the weapon motor, one for each pair of brushes. Using 2 ESCs to power the same DC motor is only possible if it has independent circuits for each pair of brushes, such as the Magmotors (which have 4 brushes).

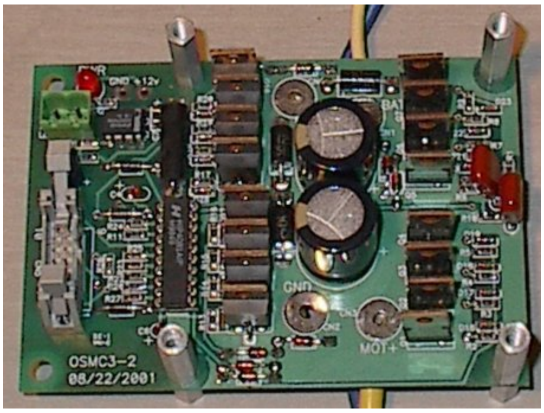


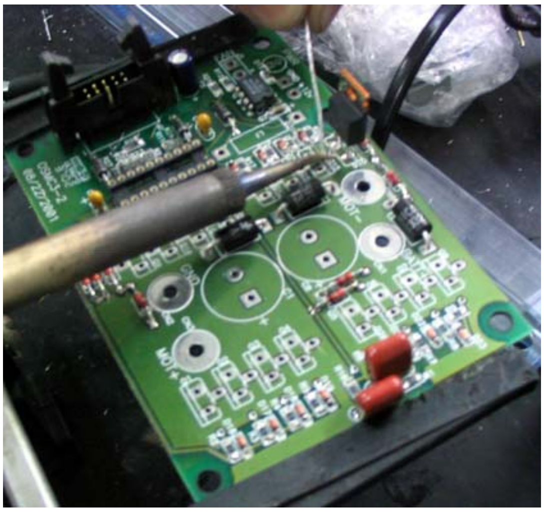
This scheme has improvements that are important for fully reversible spinners:

- the two Victors HV-36 powering the weapon allow its spin sense to be inverted if the robot is flipped, maximizing weapon effectiveness;
- one Victor for each pair of brushes from the weapon motor improves reliability, since if one of them fails the other will be able to spin the motor, although with less power;
- Victors weigh less than the TW-C1 solenoid, therefore their inertia is lower, reducing the risk of mechanical damage due to high impact accelerations;
- if the weapon is taking too long to brake, then simply set the jumpers from its Victors to the brake setting; otherwise, choose the coast setting to prevent the motor from overheating;
- Victors are less prone to lock-up than solenoids, therefore safety is also improved;
- a dedicated Universal BEC is used to power the receiver, instead of one integrated in an RC interface that also controls other devices, improving reliability; note that this BEC features a switching voltage regulator instead of a linear regulator, avoiding overheating problems.

7.8. Developing your Own Electronics

The speed controllers presented in this chapter and their radio-control interface (RC interface, a.k.a. logic interface, which interfaces the controllers with the receiver) are not cheap. But even so they are an off-the-shelf solution with very good cost-benefit, considering their complexity. Since most good quality electronic components needed by these systems are expensive, developing your own speed controller or assembling yourself an existing one doesn't save you too much money. For instance, a fully assembled and tested OSMC (pictured to the right) costs US$219 at www.robotpower.com, while its bare board (US$29) and components (about US$140) will set you back about US$169. You're basically paying US$50 for assembly and testing, which is quite reasonable, in special considering that you could burn out the entire controller if it's not carefully soldered.





But developing your own speed controller or RC interface, or even just assembling an OSMC (as pictured to the right), is a great learning experience. In addition, you'll be able to tailor the RC interface to your needs. For instance, Robot Power sells a very high-performance microcontroller-based closed-loop interface that controls two OSMCs, named Dalf. It is great for an autonomous robot, but it would be overkill for a radio-controlled combot, in special considering its US$250 price tag. The simpler MOB (Modular OSMC Brain) interface, or its more compact version µMOB, would be a much better option for robot combat, however both have been discontinued. This was the motivation for us to create a compact 3-channel RC interface to control two OSMCs (for the drivetrain motors) and one heavy duty relay or solenoid (such as the White-Rodgers 586 solenoid, for the weapon system), including a BEC to power the receiver, as explained in section 7.8.2.

Note, however, that bulletproof speed controllers and RC interfaces are not trivial to build. So, if you're planning to use your own system in combat, it is fundamental to perform several benchmark tests to avoid any surprises. There are a lot of things that can go wrong with these systems.

7.8.1. Speed Controller Development

Before we discuss the RC interface, it is important to understand the speed controller it will interface with. In this section, we'll introduce the main features of a typical high power speed controller, based on the OSMC design.

A typical speed controller is basically an H-Bridge (introduced in section 7.2.3) used to power DC motors with a controllable voltage. To do that, the H-Bridge uses one or more transistors in parallel at each of its four legs.

To activate the H-Bridge, do not use a Bipolar Junction Transistor (BJT), it is not efficient when dealing with the high electric currents needed in combat. Instead, use a MOSFET (Metal-Oxide Semiconductor Field-Effect Transistor, a.k.a. FET), such as the IRF1405 used in our OSMC boards. It has several advantages, despite its relatively high cost.

The first advantage of FETs is that they are voltage-activated (instead of current-activated such as in a BJT), making it easy to activate them. It is enough to guarantee that its input voltage (at the gate) is higher than its threshold voltage V_{th}, to allow the high currents to go through the drain and source.

When the FET is activated, in saturation mode, it behaves as a resistor, with resistance R_{on}. Very good quality FETs can have R_{on} as low as 5mΩ. To continuously supply, for instance, 160A to a motor, you will need more than one FET at each leg of the H-Bridge. If 4 FETs are used in parallel at each leg (totaling 4 × 4 = 16 FETs for all 4 legs from the H-Bridge), then the power dissipated by each FET is about

$$P = I^2 \cdot R_{on} = \left(\frac{160A}{4}\right)^2 \cdot 5m\Omega = 8W$$

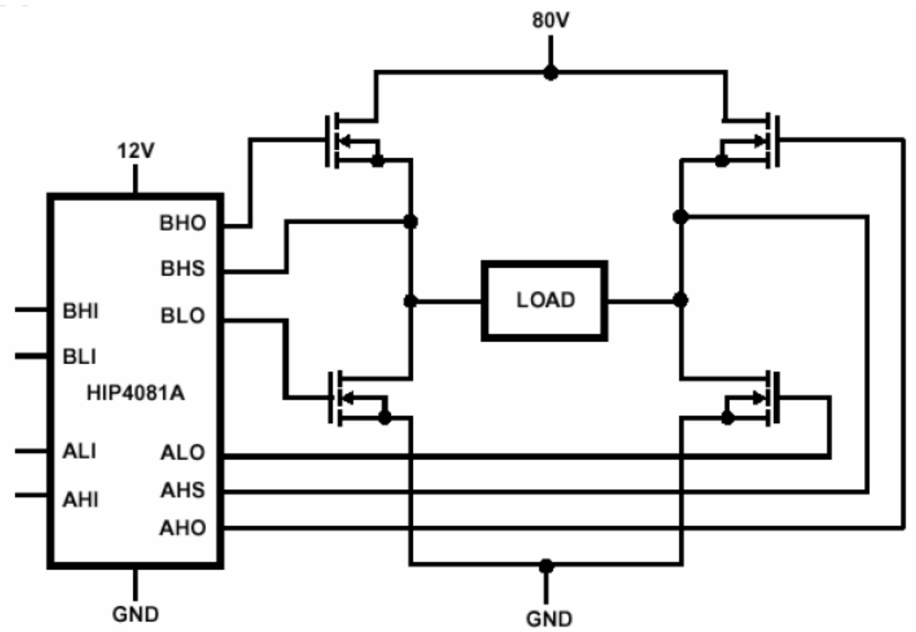
which is an acceptable value for use with small heatsinks coupled with a fan to actively cool down the FETs. If the heatsinks were not used, then the maximum continuous current acceptable going through a system with 4 FETs in parallel would be approximately 100A. Another great advantage of the FETs is that they don't have any current limitation, as long as their maximum temperature is not exceeded. Therefore, FETs can easily take very high current peaks, as long as they are brief enough not to overheat them. And the FET commutation usually takes only a few dozen nanoseconds, keeping low the energy losses from this process.

The fact that FETs are activated by voltage helps a lot in the development of an activation circuit. But there's a catch that can cause a few problems. For the FET to conduct (in saturation mode), an electric charge must be injected at the gate of the FET to make the voltage between the gate and the source reach approximately 10V. This 10V is, in general, the voltage required to completely enter saturation mode, minimizing the value of the resistance R_{on}. Such need to charge the FET, called parasitic capacitance effect, can be modeled as a capacitor in parallel with the gate of the FET.

To charge this large capacitance, the integrated circuit HIP4081A can be used. It is a high frequency H-Bridge driver, capable of supplying up to 2A for the four FETs connected in parallel to each output. To avoid shoot-through, which could happen for instance if an upper FET turns on while a lower FET from the same side of the H-Bridge is still conducting, a resistor is connected in series with the gate of each FET, limiting the total current and making the FETs take longer to be activated. In this way, the resistors help to balance the T_{on} and T_{off} times from all FETs in parallel, by equalizing their resistor-capacitor constants.

Despite the presence of the resistors, there would still be a chance of happening a shoot-through. Two protection measures exist to avoid this condition. The first is a programmable time in the HIP4081A when both FETs get turned off (in cut-off mode). The second is the addition of extremely fast diodes in parallel with the resistors, so that during the T_{off} time of the PWM the entire current is drained by them, eliminating any chance of happening a shoot-through in the circuit.

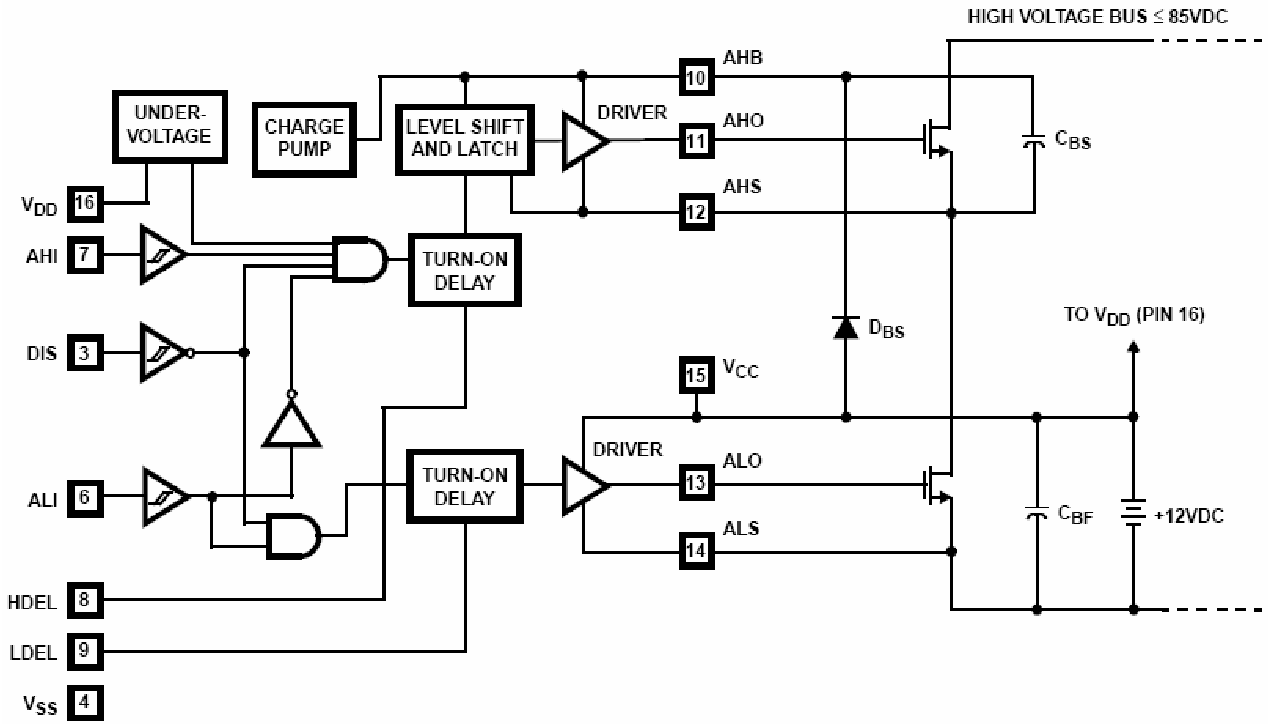
The HIP4081A has, therefore, the function to activate the upper and lower FETs, including a circuit to increase the voltage to the levels required by the FETs. In the application example pictured to the right, the HIP4081A is powered by 12V (allowable values are between 9.5V and 15V), while a battery voltage of 80V (or any other value between 12V and 80V) is applied to the load.



The load in this example can be, for instance, a brushed DC motor. If the voltage supplied to the HIP4081A is lower than 9.5V, then an internal protection turns off the upper FETs. On the other hand, if such voltage is higher than 16V, the HIP4081A can be damaged. In addition, to protect the FETs against voltage peaks, two Zener diodes are used to limit their voltage to 15V.

The HIP4081A has four digital inputs, AHI, ALI, BHI and BLI, each one corresponding to the outputs that power each group of FETs from a leg, respectively AHO, ALO, BHO and BLO, as pictured above. In other words, when an input is enabled, the FETs connected to the corresponding output are activated. The RC interface, which in the case of the OSMCs is an external electronic system (not an integrated one as in Victors), needs to send PWM and direction signals to the HIP4081A to control the H-Bridge. These digital signals are compatible with the TTL logic, but any input voltage above 3V, such as 5V or 12V, is recognized as a high ("1") logic state.

Also, the HIP4081A has a protection in its internal logic against shoot-through, which shuts down the upper FETs connected to AHO (or BHO) when the lower FETs from the same side of the bridge, connected to ALO (or BLO), are activated, independently of the state of the upper inputs AHI and BHI. This protection is implemented using AND logic gates in all the HIP4081A inputs, as seen in the picture on the next page, which shows the functional diagram of half of a HIP4081A driver. The AND gates have as input the values of AHI, ALI, BHI and BLI, in addition to the complement from the DIS (Disable) pin, deactivating all FETs if DIS has a low logic level.



Each upper gate also features two other inputs. One of them is used for protection against low supply voltages, while the other is the complement of the lower gates, to guarantee that the upper gate output AHO (or BHO) will be turned off if the lower gate inputs ALI (or BLI), from the same side, are activated.

Resistors are also connected to the HIP4081A, between the input and ground pins. These resistors guarantee that all FETs will be turned off if no RC interface is connected to the power board. This is an additional protection to make sure that the motors will be turned off if the connection with the RC interface is lost.

Due to the nature of the used FETs, the gate voltage should be approximately 10V higher than the battery voltage to activate the upper FETs. To generate such higher voltage, the HIP4081A has a charge-pump system that, with the aid of a diode and an external capacitor, generates the necessary voltage at the outputs AHO and BHO, making it possible to activate the FETs connected to them.

To protect the circuit from voltage peaks caused by the DC motor brushes and commutators, a Transient Voltage Suppressor (TVS) is used. It works exactly as a Zener diode, in other words, when the voltage on the TVS is above a specified value, it starts conducting, “absorbing” the excess voltage. The TVS is optimized to tolerate voltage peaks with high currents. It is used to absorb the voltage peaks between the battery terminals, and to protect the FETs.

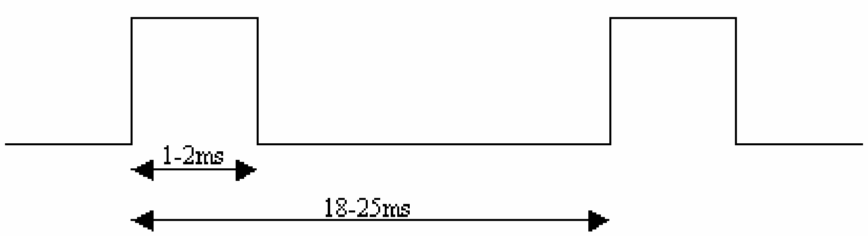
In addition to the TVS, resistor-capacitor circuits between the motor terminals provide an additional protection against high frequency peaks generated by the brushes. Also, large electrolytic capacitors are placed as close as possible to the H-Bridge to reduce the effects caused by the inductances of the wires that connect the battery to the circuit.

The last part of the circuit comprises the switched-mode power supply, which converts the battery voltage (between 12V and 80V) down to 12V, using a high efficiency regulator that does not need a heatsink. These 12V will also be used to power the RC interface, described next.

7.8.2. RC Interface Development

The power circuit discussed above cannot receive directly the signals from a radio-control (RC) receiver. These signals need to be treated first. This signal conditioning is made through a RC interface circuit, which is an interface between a receiver and a speed controller (or solenoid).

There are several off-the-shelf RC interfaces that you can buy, for instance, at www.robotmarketplace.com. But if you want to build one yourself, then you'll probably need to use a micro-controller, such as a PIC, dsPIC or AVR, capable of executing several million instructions per second. The RC interface that we've developed uses a PIC, capable of decoding the signals from up to four receiver channels, to use them to command power circuits, solenoids or relays.



The input signal in the RC interface comes from the receiver, it is a pulse train following the PPM standard, as pictured to the right. This pulse train has a period that can vary between 18 and 25ms, with each pulse lasting between 1ms (low) and 2ms (high).

As mentioned before, PPM and PWM are two completely different signals, even though they are both pulses. In our application, PPM is a low power pulse train from the receiver that carries the commands from several channels through a code that is based on the absolute width of each pulse, bringing not only the information about the desired speed of several motors, but also their direction. PWM, on the other hand, as explained in section 7.2.2, is a pulsed signal that carries, in our application, the information about the absolute speed (but not the direction) of a single motor, determined from the *ratio* between the periods T_{on} and T (and not from the absolute value of T_{on}). So, the job of our RC interface is to take the single PPM signal from the receiver, decode it, and send one PWM and one directionality signal to the power board of each motor.

When, for instance, a stick of the radio control is completely to the left, the PPM pulse from the associated channel has a 1ms width; if the stick is in the middle, then the width is 1.5ms; and if the stick is moved completely to the right, the pulse will take 2ms. All other stick positions will translate to a pulse width between 1ms and 2ms. The pulse width is either directly or exponentially proportional to the stick position, depending on the radio settings.

Therefore, to control a bi-directional motor, the 1ms pulse is usually associated with a command to move back at full speed, while a 2ms pulse would mean move forward at full speed, and a 1.5ms pulse means that the motor should stop. Also, for instance, a 1.9ms pulse would mean that we want to go forward with $2 \times (1.9 - 1.5) = 0.80 = 80\%$ of the top speed, while a 1.2ms pulse would mean that we want to go back at 60% of the top speed, because $2 \times (1.2 - 1.5) = -0.60 = -60\%$.

We've programmed our PIC to validate the PPM signal from the receiver, and then to count the width (time interval) of each pulse. Clearly, each pulse is associated with one receiver channel.

There are five output signals that need to be sent to the HIP4081A driver: AHI, ALI, BHI, BLI, and the DIS (Disable) signal. The DIS signal is only used in case you want to turn off the H-Bridge. Due to the protection against shoot-through in the HIP4081A, it is possible to simplify the involved logic, keeping both AHI and BHI signals in the high logic level, all the time. The suggested signals for the correct operation of the power circuit are shown in the table below.

AHI	BHI	ALI	BLI	DIS	Function
1	1	0	PWM	0	Forward
1	1	PWM	0	0	Back
1	1	0	0	0	Brake
1	1	1	1	0	Brake
×	×	×	×	1	Off

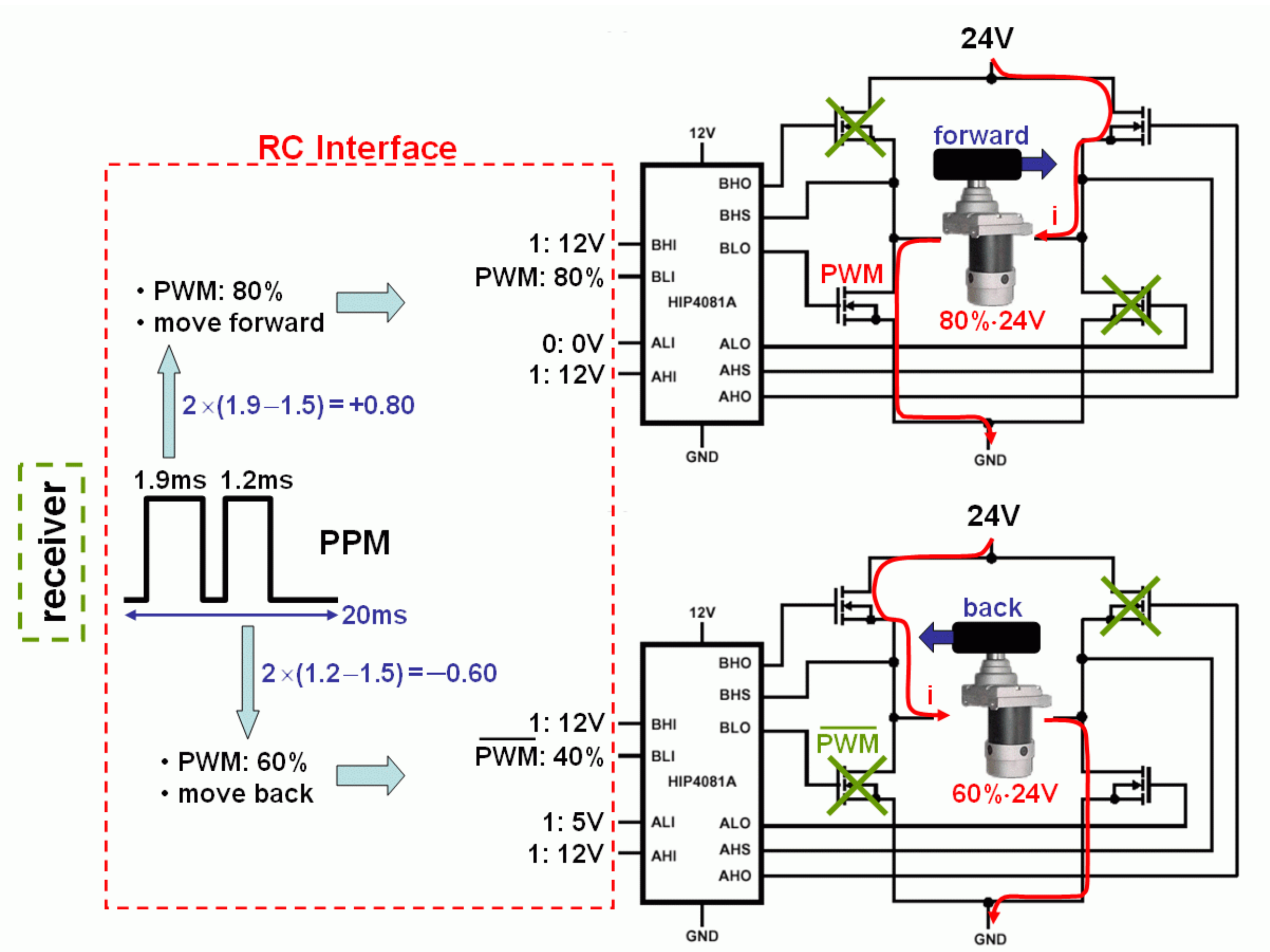
×: the state doesn't matter; 1: means 5V (or 12V); 0: means 0V;

So, we only need to deal with two signals, ALI and BLI, because AHI and BHI are always kept at the high ("1") logic level. But both ALI and BLI deal with the PWM signal, as seen in the table above, which is not good because you would need to use two PWM output pins from the PIC to control a single motor. We need to modify the table above such that only one signal takes care of the PWM (for instance, the BLI, which can be used to define the absolute speed of the controlled motor) while the other defines the direction of the movement (the ALI signal, in this example). In this way, only one PWM output pin from the PIC will need to be used per motor (to carry the BLI signal, in this example). The modification is shown in the table below.

AHI	BHI	ALI	BLI	DIS	Function
1	1	0	PWM	0	Forward
1	1	1	$\overline{\text{PWM}}$	0	Back
1	1	0	0	0	Brake
1	1	1	1	0	Brake
×	×	×	×	1	Off

×: the state doesn't matter; 1: means 5V (or 12V); 0: means 0V;

With this new table, the BLI will be the PWM signal, while the ALI will control the direction in such a way that a low ("0") logic level means forward, and a high ("1") logic level means backward. So, when moving forward, the current goes through the FETs connected to the HIP4081A outputs AHO and BLO, and when moving backward the current goes through the FETs at BHO and ALO. However, when moving back, the PWM that goes to BLI must be inverted, either through software or hardware, resulting in the $\overline{\text{PWM}}$ signal as shown in the table. The reason for that can be understood through the example in the next page.



The figure above shows an example of a PPM pulse train with 20ms period, generated by a receiver, used to control two 24V permanent magnet brushed DC motors. The RC interface uses its PIC controller to measure the pulse widths, resulting in 1.9ms for the first motor (the top motor in the figure) and 1.2ms for the second. Then, the RC interface figures out, from the calculations shown in the figure, that the first motor has been commanded to move forward with 80% of the provided 24V, while the second motor needs to move back with 60% of 24V.

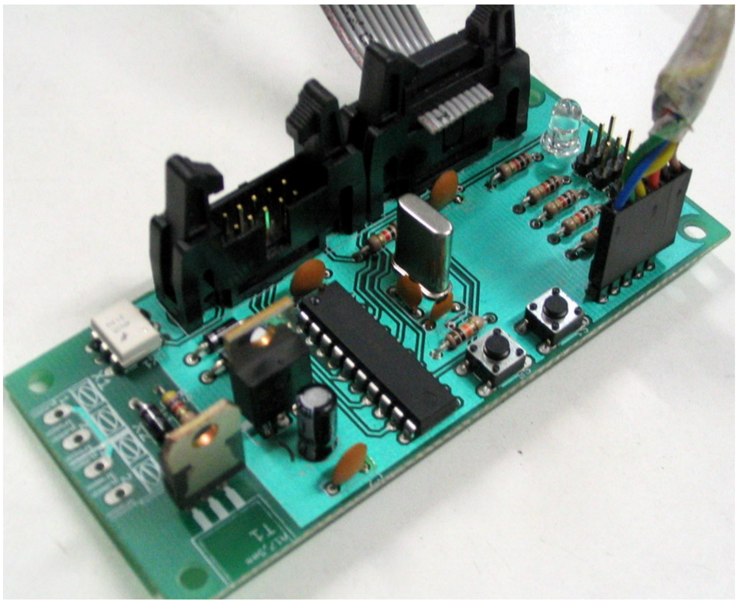
For the first motor, the RC interface sets the ALI voltage to 0V (low level "0") to indicate that the motor should move forward, and the BLI receives a PWM signal that is high ("1") during 80% of the time. Note that the BHI and AHI are always at a high ("1") logic level, by default, which can be obtained using 12V hard-wired from the HIP4081A 12V pin (as discussed before, any voltage above 3V translates to "1" in the HIP4081A). With ALI at 0V, the FETs connected to ALO do not conduct. Even though BHI is at the high logic level, the FETs at BHO do not conduct because of the shoot-through protection that prevents a short-circuit of the 24V battery through BHO and BLO.

So, the current from the 24V battery has to flow to the first motor through AHO (which conducts because AHI is always set to "1") and BLO. But BLO only conducts 80% of the time, because of the 80% PWM signal at BLI, so the resulting motor voltage will be, in average, about 80% of the 24V input voltage, as desired.

For the second motor, ALI is set to 5V (at the high logic level "1") to indicate that the motor should move backward, while BHI and AHI are always at 12V (also at the high logic level "1"), by default. But the RC interface, instead of sending the desired 60% PWM signal to BLI, sends the inverse signal $\overline{\text{PWM}}$, which is low (0V, instead of high) during 60% of the time. Therefore, this inverted signal is high (logic level "1") during only 40% of the time. With ALI set to the high logic level, the FETs at ALO will always conduct. Even though AHI is at the high logic level, the FETs at AHO do not conduct because of the shoot-through protection that prevents a short-circuit of the 24V battery through AHO and ALO.

During 60% of the time, when BLI is low ("0"), the FETs at BLO will not conduct, and the current from the 24V battery will flow to the second motor through BHO (which conducts because BHI is always set to "1") and ALO. But during the remaining 40% of the time, when BLI is high ("1") and hence the BLO FETs conduct, the BHO FETs will stop conducting due to the shoot-through protection that prevents a short-circuit through BHO and BLO. Without the BHO FETs to conduct, the second motor won't be powered during 40% of the time. So, the resulting motor voltage will be, in average, about 60% of the 24V input voltage, while moving back, as desired.

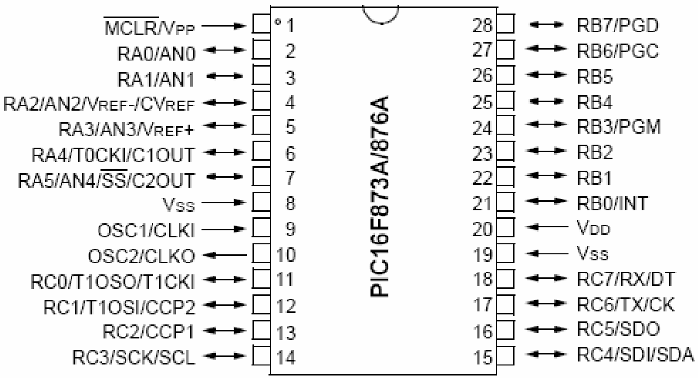
In summary, the PWM signal must be inverted at BLI to move back because we make use of the shoot-through protection, which only allows the motor to be powered when BLO is not conducting.



The hardware of the developed RC interface, pictured to the right, is relatively simple and compact in size, measuring 4" × 1.75". It includes a micro-controller PIC16F876A, and a buffer to isolate the signals generated by the PIC from the power board signals, to avoid any problems.

The interface board also features an independent circuit used to activate a high power relay or solenoid, which is completely isolated with the aid of an optocoupler.

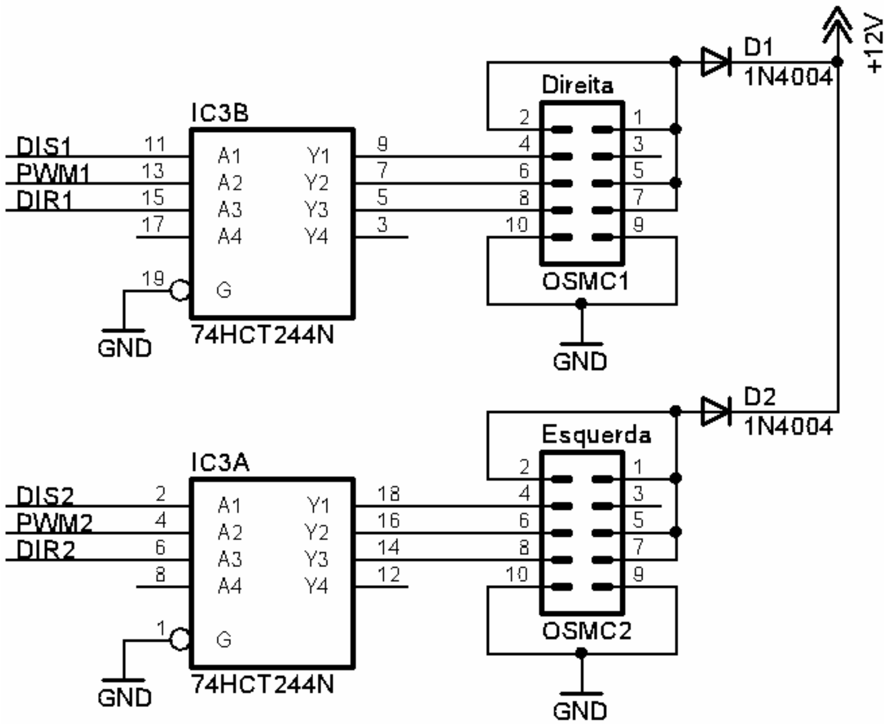
In addition, the developed board includes a BEC that takes the 12V from the power board (used in the HIP4081A driver) and converts it into 5V to power both the receiver and the RC interface itself, using a linear regulator. The developed board also includes two buttons, one to reset the PIC and the other to enter into calibration mode.



The micro-controller PIC16F876A (pictured to the right) features in-circuit serial programming, which allows it to be programmed without the need to remove it from the board. The input signals from the receiver are connected through resistors to the pins RB4 through RB7 from the PIC.

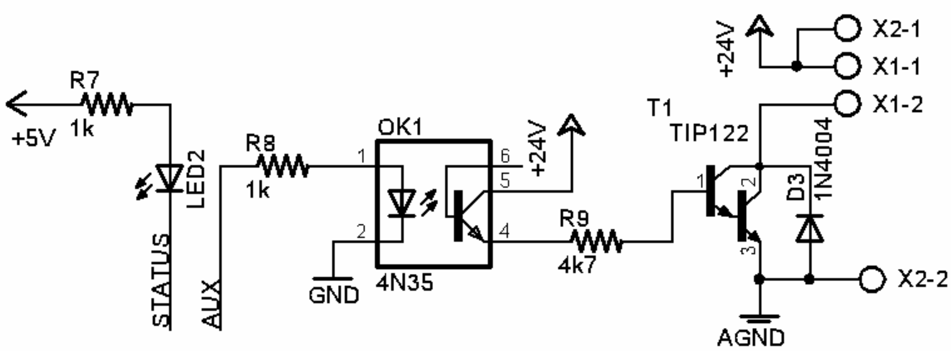
Those pins generate an interruption when the input changes its state, which is perfect to read PPM receiver signals.

The buffer used in both PWM outputs from the RC interface is the chip 74HCT244, consisting of two sets of four buffers each. It is possible to use as well other chips equivalent to 74HCT244, such as the 74HC244, as long as their output voltages are high enough to be used with the HIP4081A from the power board. For instance, the 74LS244 chip is not recommended in this case, because it associates any voltage beyond 2V to a high logic level, while the HIP4081A requires a minimum value of 2.5V.



The figure to the right shows two connectors, named OSMC1 and OSMC2, which are used to connect the developed RC interface to two OSMC speed controllers. The output signals to AHI and BHI (pins 5 and 7, respectively) need to be permanently set at the high logic level "1" as discussed before. This is accomplished with any voltage higher than 3V, not necessarily 5V, so in our case we've connected these pins 5 and 7 to the pins 1 and 2, which provide the 12V supplied by the HIP4081A.

Note also in the figure above that there are two diodes D1 and D2 between the RC interface and both power boards. These diodes are a redundancy measure to ensure that the RC interface, which is powered by the 12V lines from both power boards, will still be functional even if one of the power boards burns out.

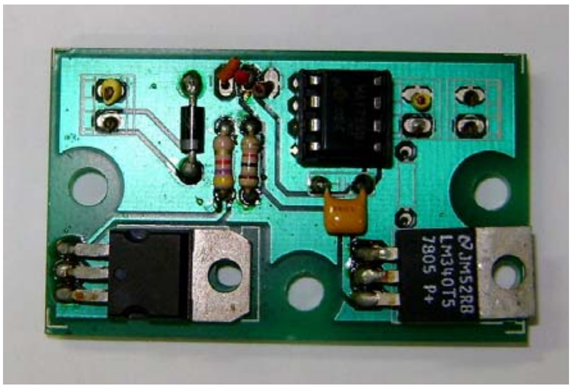


The developed RC interface is also able to activate a relay (or a solenoid), usually used in the weapon system, including as well a status LED, as pictured to the right. This circuit uses an optocoupler that, when enabled, makes the T1 transistor conduct, activating the relay. To do that, the relay terminals must be positioned at X1-1 and X1-2, while 24V should be applied to X2-1 and X2-2 (assuming a 24V relay). The transistor T1 can handle up to 3.5A with a heatsink, or 1.0A without one. The transistors TIP120, TIP121 or TIP122 can be used in this circuit. The figure also shows the status LED, which is used to inform the state of the program running in the PIC. In this case, the LED is used to indicate whether the circuit is under normal mode or calibration mode.

The software used by the PIC16F876A from the RC interface was written in the programming language C. The entire program, together with more details about our RC interface board, can be found in the undergraduate thesis (in Portuguese) from the RioBotz team member and former advisee *Felipe Maimon*, which can be downloaded at www.riobotz.com.br/en/tutorial.html. I've tried to summarize and translate to English the main points from this thesis here, in section 7.8. Note that the program, which is relatively lengthy, is very specific to the hardware of the developed RC interface. Our RC interface is nicknamed MOB, in honor of the discontinued "Modular OSMC Brain" interface, however here it stands for "*Maimon*'s OSMC Board."

Our RC interface board was successfully used in all our OSMC-powered middleweight combots: the overhead thwackbot Anubis (controlling the speed of both NPC T74 drive motors), as well as the horizontal bar spinners Ciclone (controlling the speed of two DeWalt gearmotors and activating its Etek weapon motor through a White-Rodgers 586 SPDT solenoid) and Titan (controlling 4 Magmotors S28-150, two of them for the drive system using PWM, and the other two for the weapon through a single TW-C1 solenoid). The board withstood well the rigors of combat.

We haven't been using our RC interface board described above since we migrated from OSMCs to Victors in all our middleweights. Victors don't handle as much current as the OSMCs do, but they're more compact and they can be connected directly to the receiver without an RC interface.



However, we still needed an RC interface to activate the solenoid from the weapon system of our middleweights. We've then designed another more compact RC interface, measuring 2" × 1.25" (pictured to the right), without the PWM outputs, featuring 1 output for a high power relay or solenoid and a BEC to power the receiver. It was used until 2008 by Touro and Titan to power the TW-C1 solenoid from their weapon systems.

Finally, we've considered designing a third RC interface board, featuring 2 solenoid outputs controlled by independent channels. Having two solenoid outputs, instead of one, would be useful to power a single weapon (such as a drum) in both directions using two SPDT solenoids arranged in the bang-bang configuration shown in section 7.2.1 (or it could be used in combots with more than one weapon). However, the added weight and volume of 2 solenoids, in addition to the risk of shorting out the battery if both of them are accidentally switched to a "shoot-through" configuration, made us choose instead to use Victors to power the weapon in both directions, as explained in section 7.7.3.

To power the presented electronic systems, you'll need batteries that are capable to deliver high currents. The main battery types, along with their advantages and disadvantages, are studied in the next chapter.

Chapter

8

Batteries

Batteries are components that usually limit a lot the autonomy of a mobile robot, besides representing a significant part of its weight. Usually, batteries are the heaviest component of a mobile robot. Humanoid robots, for instance, have reached an impressive level of sophistication in the last 20 years. Powerful motors were miniaturized, high performance computational systems became even more compact, however the components that less evolved until now were batteries. In 2000, the most sophisticated humanoid robots needed to be recharged every 30 minutes, even though their batteries accounted for a significant portion of their weight, about 15%. But recent advances in lithium battery technologies, such as the development of A123 batteries, are starting to change this.

Fortunately, combat robots only need an autonomy of about 3 minutes. Combots still need about 15% of their weight in batteries, similar to several humanoid robots, however it is possible to extract from them a much higher power during this short period. But most batteries were designed to be slowly discharged, in 20 hours, in 1 hour, not in 3 minutes. Therefore, it is necessary to know the advantages and disadvantages of each type.

The main battery types are: lead-acid (Sealed Lead Acid, SLA), nickel-cadmium (NiCd), nickel-metal hydride (NiMH), alkaline, and lithium, presented next.

8.1. Battery Types

8.1.1. Sealed Lead Acid (SLA)

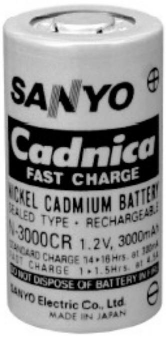
SLA batteries have lead-based electrodes, and electrolyte composed of sulfuric acid. Each electrode inside the battery contributes with about 2V, therefore a typical 12V battery has 6 cells connected in series. The SLA types usually used in automobiles cannot be used in combat, because the acid can spill if they are flipped over or perforated by an opponent’s weapon.

Competitions only allow SLA batteries in which the electrolyte is immobilized, which could work upside down without risk of spilling. The most common technologies to immobilize the electrolyte are gel, where silica is added to generate a semi-solid gel, and AGM (Absorbed Glass Matte), where a fibrous and porous material absorbs the acid and keeps it suspended.

SLAs are usually available in up to 12V, therefore it is necessary to use at least 2 of them connected in series to reach usual combot voltages of 24V or more. They are the cheapest type of battery, however they are the heaviest ones, therefore it is usually better to replace them for NiCd, NiMH, or lithium batteries, which will be discussed next. Another disadvantage is that most of them take several hours to charge.

8.1.2. Nickel-Cadmium (NiCd)

NiCd batteries use nickel as cathode, and cadmium as anode. They supply high currents without significant voltage drops, and because of that they are an excellent choice to power the robots’ weapons. They are more expensive than SLAs, however they can last several years if properly handled, returning their investment. Each cell (pictured to the right) provides about 1.2V. The cells are usually soldered in series to form battery packs (also pictured to the right), with voltages that are a multiple of 1.2V. The packs used in combat usually have 12V, 18V, 24V and 36V, with respectively 10, 15, 20 and 30 cells.



8.1.3. Nickel-Metal Hydride (NiMH)

NiMH batteries also use nickel as cathode, however the anode is composed by a metallic alloy capable to absorb hydrates, replacing cadmium, which is poisonous. NiMH batteries store 30% more energy per weight than NiCd, however they can consistently supply about half the peak currents of a NiCd with same capacity. They are a good choice for the robot's drive system, resulting in a high capacity to avoid having a slow robot at the end of a match (drive systems usually don’t require very high current peaks due to wheel slip). A significant problem is that these batteries lose naturally about 30% of their charge every month (self-discharge), therefore they are not appropriate for applications with sporadic use, such as TV remote controls. Even if the remote is not used, in about 2 months the battery would probably need to be recharged again.

8.1.4. Alkaline

Alkaline batteries are the most common, storing a great amount of energy. They don't suffer as much from the self-discharge problem as NiMH batteries, therefore they are the best option for sporadic use (although they are prone to suffer long term corrosion, which may cause cell rupture and electrolyte leakage – this is why they should be removed if not used for several months). The problem with alkaline batteries is that they are not able to supply high currents, and because of that they are not used in combat. Besides, they are not rechargeable, and therefore it would be very expensive to use new batteries in every match. There is a rechargeable version, called RAM (Rechargeable Alkaline Manganese, pictured to the right), however it doesn't supply high currents as well, and the number of recharge cycles is relatively low.

8.1.5. Lithium

Very used in cellular phones, portable computers and several other gadgets, lithium batteries (pictured to the right) currently are the ones with the highest charge capacity with lowest weight. However, they are more expensive and, sometimes, dangerous.

The lithium-ion type is the oldest one, and it suffers risk of explosion if perforated and exposed to oxygen, shorted out, or improperly charged, hence it is not recommended for combat robots. This risk is reduced in the lithium-ion-polymer type (a.k.a. LiPo or lithium-polymer), due to its polymeric layer, but it still exists. Newer lithium battery chemistries, such as lithium-manganese and lithium-iron-phosphate, are much safer, although great care and attention is needed when handling this kind of battery, as discussed in section 8.3.

In addition to safety issues, the models that are capable to supply high currents are still expensive, and they require some electronic system in the robot to guarantee that they won't be discharged below a critical voltage, to avoid permanent damage. But the cost-benefit is still very good. The main lithium technologies for use in combat are described next.

Lithium-Ion-Polymer

Most lithium-ion-polymer batteries have discharge rates higher than 20C, in other words, it is possible to completely discharge them in less than 1/20 of an hour, which is exactly the 3 minutes that we need during a combot match. For 2 minute matches, common in insect classes, a discharge rate of 30C or higher would be better.

The nominal lithium-polymer cell voltage is 3.7V, but when fully charged it provides up to 4.2V per cell. It is not recommended to let the battery voltage drop bellow 3.0V per cell. If this happens, the pack can swell and become permanently damaged, which is also known as puffing or ballooning. This is why it is a good idea to use LiPo battery sets with at least twice the capacity you might think your robot will need during a match, making sure they’ll not be completely drained.

Similarly to other battery types, more than one cell is usually needed to power a combat robot. Most manufacturers use the number of cells connected in series or in parallel to describe their products. For instance, if each cell has a nominal voltage of 3.7V and a 500mAh capacity, then a 3S LiPo pack would stand for three cells in series, resulting in 11.1V and 500mAh, and a 3S2P pack would stand for two parallel arrays of three cells in series, resulting in 11.1V and 1,000mAh.

An inconvenience of LiPo batteries is that, despite their short discharge time, usually the charge time is much longer, up to 2 hours in the oldest models, which can be critical between combats. However, newer models can be safely charged at a 1C rate (in other words, in 1 hour), while a few vendors state that their packs can handle a 2C charge rate (charged in 30 minutes). The use of an adequate charger is mandatory, never charge lithium-polymer batteries on lead-acid, NiCd or NiMH chargers, otherwise it will ignite on a strong fire, releasing toxic fumes.

As it can be seen in the picture to the right, LiPo batteries usually have two sets of wires. The twisted pair cable, with black and red wires, is the main power cable. The other cable, with five wires and a white connector in this case, is used for cell-balancing. This process consists of equalizing the cells after charging. There’s some controversy on that, because a few manufacturers claim that their battery packs don’t need to be balanced, while others recommend to balance the cells regularly. As a rule of thumb, cell balancing is only needed when a fully charged pack presents significant disparities between cell voltages. This can be checked using a voltmeter between the black (negative) and the other wires of the balancing connector. Most vendors recommend balancing if there’s a difference higher than 0.1V.

LiPo batteries are mostly used in combat in the lightest (up to 3lb) weight classes, however new technologies are emerging to allow their widespread use. Lithium batteries have a great potential to become the best choice even for the heaviest classes. Nowadays, there are quite a few heavier robots that use this kind of battery, such as all our hobbyweights and featherweight combots, as well as Kevin Barker's lightweight vertical spinner K2, which uses two LiPo 6S1P 5,000mAh packs.

Lithium-Manganese

Lithium-manganese batteries (LiMn, pictured to the right), developed in 2005 by Apogee High Performance Lithium Polymer Technology, use a safer chemistry that can sustain perforation without exploding.

Since the voltage of each cell is 3.7V, these batteries are entirely compatible with lithium-polymer chargers. They can provide the same peak currents of NiCd and the same capacity of NiMH with about half the weight. In addition, they can be charged at a 2C rate. Apogee states that their batteries don't need balancing, due to their cell matching process.

An interesting feature of these batteries is their polycarbonate shielding, which can minimize cell damage during rough handling in combat.

Super Charge ion Battery

Toshiba started shipping in 2008 its Super Charge ion Battery (SCiB, pictured to the right), which can be recharged to 90% of its capacity in only 5 minutes, with a life span of over 10 years.

SCiB Cell

Charging can be performed with currents as high as 50A, which is a real breakthrough. This battery can sustain more than 3,000 rapid charge cycles, with less than 10% capacity loss. It adopts a new negative-electrode material technology that is safer and more stable, being virtually resistant to punctures and short-circuits.

Unfortunately, SCiB batteries are currently only available to industrial markets, in either 2.4V/4.2Ah/0.150kg or 24V/4.2Ah/2.0kg versions.

Lithium-Iron-Phosphate

One of the most promising battery technologies is the lithium-iron-phosphate ($LiFePO_4$ or LFP), discovered in 1996 at the University of Texas. In addition to its high peak currents (over 100C pulsed discharge rates) and high capacity, it is a safe technology: it will not catch fire or explode with overcharge. Its charge time is very low compared to other lithium battery types, sometimes as low as 15 minutes. It is also environmentally friendly.

The most famous brand of $LiFePO_4$ battery is A123. Originally, only the M1 cell model (pictured to the right) was available, but now A123 is also producing other models with higher capacity, such as the M1HD and M1Ultra cells. A123 cells are also sold assembled in battle-ready packs, as pictured to the right, available in several configurations at www.battlepack.com.

Nominal voltage varies amoung manufactures, from 3.0V (K2) to 3.3V (A123) per cell, but when fully charged the cells can provide up to 3.6V. To avoid damaging them, these batteries must not be discharged below 2.8V per cell.

Note that $LiFePO_4$ batteries must be charged with a specific charger, such as the Astroflight 109 A123, iCharger 1010B+, or Robbe Power / MegaPower Infinity SR. There are also adapters, such as the Dapter123, which allows the use of most NiCd chargers.

8.2. Battery Properties

Several battery characteristics need to be considered: price, weight, voltage, shelf life, number of recharge cycles, charge time, self-discharge, discharge curve, internal resistance (which determines the peak current and voltage drop), capacity, de-rating factor and discharge rate, described next.

8.2.1. Price

Price is the first factor in the choice of batteries. SLAs are the cheapest, followed by alkaline, NiCd, NiMH, and finally the lithium batteries. Prices vary a lot depending on the technology, manufacturer, quality and capacity.

8.2.2. Weight

The weight of the battery is crucial in robot combat. More specifically, it is important to know the power-to-weight, energy-to-weight, and capacity-to-weight ratios of each type, the higher the better. SLAs are the worst ones in this requirement, they store less energy per pound than any other type. NiCd and NiMH are much better, while lithium is the best, see sections 8.2.10 and 8.2.12.

8.2.3. Voltage

Battery voltage depends on the number of cells and the electrode chemistry. SLA electrodes nominally supply 2V, usually combined to provide 12V. Alkaline electrodes supply 1.5V, while each NiCd or NiMH cell provides 1.2V. The nominal voltage of lithium batteries depends on their type: lithium-ion-polymer (LiPo) and lithium-manganese (LiMn) provide 3.7V per cell, lithium-ion (Li-Ion) 3.6V, and lithium-iron-phosphate ($LiFePO_4$ / A123) between 3.0 and 3.3V.

Note that the values above are nominal voltages. In practice, the voltage is usually higher than this value, when the battery fully charged, or it can be lower, if it is supplying very high currents, which lead to significant voltage drops due to their internal resistance.

8.2.4. Shelf Life

Shelf life depends a lot on the use and mainly on the storage temperature. In a few cases the batteries can last more than 20 years without significant capacity loss, such as in the case of NiCd stored at 40^{o}F (about 5^{o}C) in a refrigerator. If stored at 100^{o}F (about 38^{o}C), these same batteries would last less than 2 years.

8.2.5. Number of Recharge Cycles

The number of recharge cycles during the useful life of a battery goes from zero (alkaline), up to 300-800 (SLA and NiMH), 500-1200 (lithium-ion), 1500-2000 (NiCd), 5000 (SCiB), and even up to 10,000 recharge cycles in a few special lithium batteries. Note that, as the technology develops, these numbers can be outdated, however they are a good reference for comparison purposes. Forum posts and manufacturer websites are a good source of information to find out more accurate values.

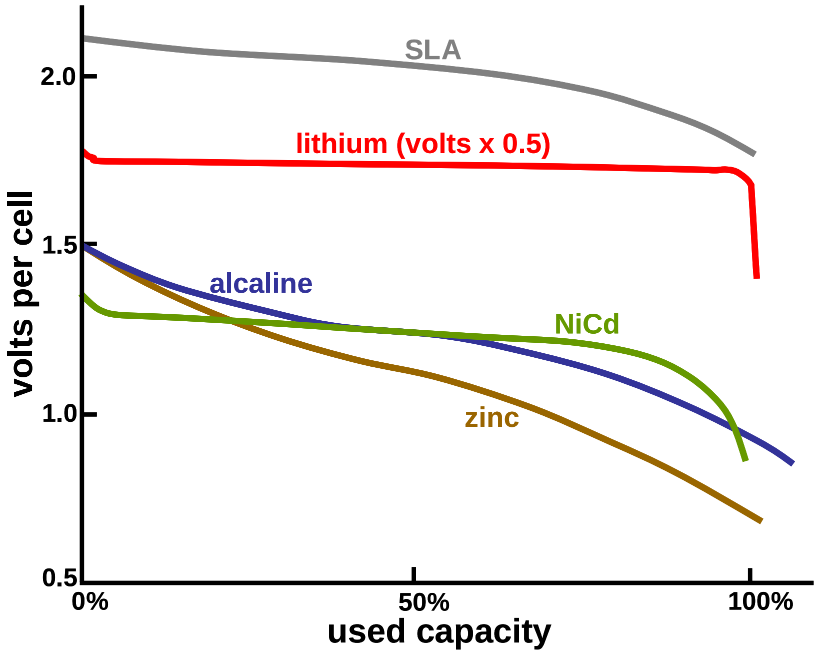
8.2.6. Charge Time

Charge time is another important factor, it determines the minimum time necessary to fully charge a battery without damaging it. The longer it is, the higher will be the number of spare battery sets you will need in a competition. SLAs are the worst ones in that sense, they usually need several hours to fully charge. The Li-Ion and LiPo types usually need at least 1 hour (1C charge rate), however a few of the newer technologies may charge must faster than this, such as A123 (in 15 minutes, with a 4C charge rate) and SCiB (in 5 minutes, 12C). The NiCd is one of the best types, it takes much less than 1 hour to fully charge, in a few cases in only 15 minutes, without permanent damage. Some newer NiMH batteries are reaching similar charge times as NiCd.

8.2.7. Self-Discharge

Self-discharge quantifies which percentage of its capacity a battery naturally loses per month (or per day). Lithium batteries lose about 5% of their capacity per month. NiCd and SLA batteries, if stored at room temperature, may lose about 10% of their capacity per month, while NiMH may lose about 30% per month. Therefore, if you use NiMH batteries in a combat robot, always recharge them again at the day of the competition, to compensate for this loss. This is also a good idea for the other types.

8.2.8. Discharge Curve



The discharge curve of a battery shows its voltage level as it drops off during use. For instance, the graph to the right shows that each electrode (cell) of a SLA battery supplies about 2.1V (up to 2.2V) when fully charged, a value that is gradually reduced until reaching about 1.7V. Therefore, a SLA battery with nominal voltage 12V (with 6 lead electrodes) would have up to 6 × 2.2V = 13.2V when fully charged, and 6 × 1.7V = 10.2V when discharged. This noticeable drop has only one advantage, it could be used to indirectly measure the remaining capacity of the battery. But this voltage drop, on the other hand, will make the system lose power and become slower. Also, robot combat judges would be able to tell that the batteries where dying from this sluggishness, awarding damage points to the opponent. This significant voltage drop happens not only with SLA, but also with (disposable) zinc and alkaline batteries, as seen in the graph above.

Lithium and NiCd cells have an almost horizontal discharge curve, keeping constant their voltage level during the entire combat (except during voltage drops due to high currents). The (rather abrupt) voltage drop is only noticeable towards the end of the battery capacity. NiMH curves are not as horizontal as in NiCd, they are slightly sloped, however not nearly as much as in SLA.

8.2.9. Internal Resistance

The internal resistance of a battery is added to the total resistance of your electronic system. Therefore, the smaller the resistance, the larger will be the current peaks that the battery can deliver. SLA and NiCd batteries have very low internal resistance, allowing them to generate very high currents. The problem with SLA is that those current peaks reduce a lot the battery capacity, due to the de-rating factor, which will be discussed later. NiMH batteries have larger resistance than NiCd, and therefore they are not able to deliver such high current peaks (if compared to NiCd batteries with same capacity, of course). The first lithium batteries had high internal resistance, however in the most recent versions, such as the A123, this value is much lower.

The internal resistance is also related with the voltage drop in the battery caused by very high currents. This is simply due to the energy loss caused by the resistance, which is significant under high currents. This energy is converted into heat, which can also cause thermal failure of the battery due to overheating. A123 batteries, due to their very low internal resistance, can deliver very high currents without significant increase in their temperature.

8.2.10. Capacity

Capacity measures the total amount of current that a battery can deliver until it is fully discharged. It is measured in A·h, calculated from the product between the total discharge time and the average delivered current (if the current isn't constant, then it is calculated integrating it along the discharge time). For instance, a 3.6A·h battery would theoretically supply a current of 3.6A, continually, during 1 hour, or 36A for 1/10 hour (6 minutes). Note that if two identical batteries are connected in parallel, the total capacity is doubled.

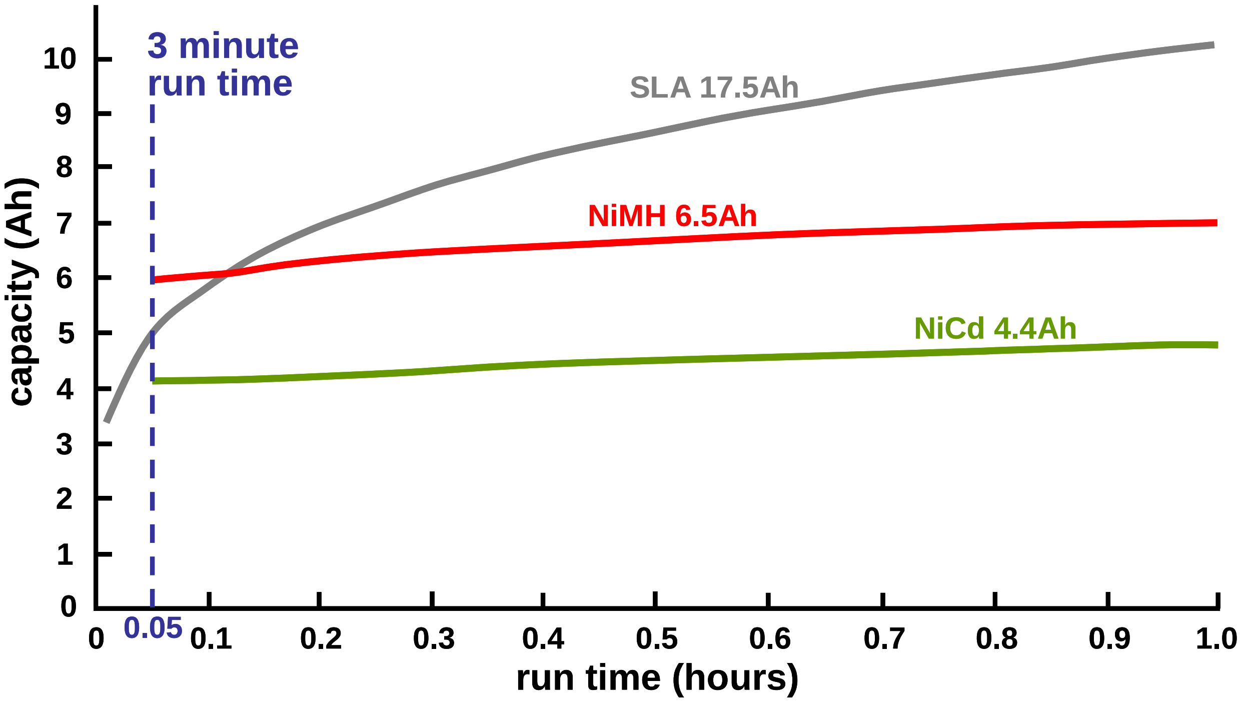
In theory, the capacity of a 24V SLA battery would be about 1.25A·h per kilogram (about 0.57A·h per pound). This is a relatively low capacity, leading to a low energy density of about 24V × 1.25A·h = 30V·A·h/kg = 30W·h/kg. A 24V NiCd pack would have from 1.7 to 2.5A·h/kg (0.77 to 1.13A·h/lb, with energy density between 40 and 60W·h/kg), a good quality NiMH would have 2.5 to 3.3A·h/kg (1.13 to 1.5A·h/lb, with energy density between 60 and 80W·h/kg), and finally lithium batteries would go beyond 4.2A·h/kg (1.9A·h/lb, with energy densities between 100 and 200W·h/kg).

Regarding energy per volume, which is also relevant if you want to build a compact robot, then SLAs only have between 60 and 75 W·h per liter, while NiCd between 50 and 150, NiMH between 140 and 300, lithium-ion about 270, and LiPo around 300W·h/liter.

But those capacity and energy numbers are theoretical, because in practice it is not so simple, the effect of the de-rating factor must be considered, as discussed next.

8.2.11. De-Rating Factor

The total capacity of a battery depends on the discharge time. The nominal capacity that is written on SLA batteries, for instance, is related to a discharge time of 20 hours. Therefore, if you discharge a 17.5A·h SLA with a constant current of only 17.5/20 = 0.875A, it will really last 20 hours. But if you discharge it at 17.5A, it won't last 1 hour. This is because the real capacity of this battery in 1 hour would be only about 10A·h, see the graph below. Therefore, the correct value for a 1 hour discharge would be 10A instead of 17.5A. As it is discharged faster, its capacity decreases. That same battery would only supply 5.8A·h if totally discharged in 6 minutes (0.1 hours), and less than 5A·h during a 3 minute combat (0.05 hours). Those values, obtained experimentally, are represented in the graph below.



Note in the graph that the capacity of the SLA battery depends a lot on the total discharge time (run time). The value that must be multiplied to the nominal capacity to generate the actual battery capacity is called de-rating factor, a number that is usually between 0 and 1. For instance, the de-rating factor of a SLA battery that is required to be discharged in only 6 minutes (0.1h) is worth 0.33, which would give 0.33 × 17.5 = 5.8A·h for the 17.5A·h SLA, which agrees with the graph above, delivering continuous 5.8A·h / 0.1h = 58A.

If you still need more current than that, to the point of fully discharging the battery during a 3 minute combat (0.05h), the de-rating factor will be even lower, about 0.28. In this case, the capacity would be 0.28 × 17.5 = 4.9A·h for the 17.5A·h SLA, which also agrees with the graph above, delivering continuous 4.9A·h / 0.05h = 98A.

The special SLA Hawker-Odyssey (also known as Hawker-Genesis, pictured to the right) has higher de-rating factors than regular SLA batteries, reaching values between 0.4 and 0.5 for the 6 minute run time (instead of 0.33).

One of the greatest advantages of NiCd and NiMH batteries is that their capacity is almost insensitive to the total discharge time (run time). This can be seen in the previous graph, which shows NiCd and NiMH capacity curves that are almost horizontal. Note that their nominal capacity is measured in a 1 hour discharge time, instead of 20 hours as with SLA (lithium batteries are also measured in 1 hour). Even so, there is a de-rating factor for NiCd and NiMH, which is about 0.9 for run times between 3 and 6 minutes. That de-rating factor is 3 times better than the one from regular SLA, and almost 2 times better than in Hawker-Odyssey.

Therefore, for instance, 2 regular SLA batteries with 12V and 18A·h each, when in series, are able to supply 24V with a combined weight of 6.2kg × 2 = 12.4kg (27.3lb). If their desired run time is 3 minutes (in practice it is safer to design your robot with run times of at least 4 minutes, so it can safely endure a 3 minute combat), their actual capacity is 0.28 × 18 = 5A·h. On the other hand, two 24V NiCd packs with 3A·h each, when in parallel, can supply the same 24V with a nominal capacity of 3A·h × 2 = 6A·h. Their actual capacity in 3 minutes is 0.9 × 6 = 5.4A·h, larger than the one from the SLA, and with a total weight of only 1.8kg × 2 = 3.6kg (7.9lb). This is less than a third of the weight of the SLA set, with an equivalent capacity!

The performance of SLA batteries only approaches NiCd when at least 10A·h is needed by the robot, such as in heavyweights and super heavyweights. Even so, this only happens for special batteries such as Hawker-Odyssey. For instance, two 12V Hawker-Odyssey batteries with 26A·h each, when in series, supply 24V, with a combined weight of 6.1kg × 2 = 12.2kg (26.9lb), with an actual capacity of 0.42 × 26 = 10.9A·h (the 0.42 de-rating factor was experimentally measured). It would be necessary to use four 24V NiCd packs, with 3A·h each, in parallel, to achieve those values, resulting in a combined weight of 7.2kg (15.9lb). The NiCd packs would still be lighter, however the weight difference decreased to 5kg (11lb), a small value if compared to the total weight of a heavyweight or super heavyweight. The advantages of using special SLA batteries are their price (about one third the price of equivalent NiCd) and the achievable peak currents, which would be about 800A for this NiCd arrangement (with high discharge cells) but almost 2400A for the Hawker-Odyssey (watch out not to burn your motors and electronics!).

The previous calculation always assumed that the discharge current was constant, which is certainly not true during combat. To estimate with better accuracy the capacity of a SLA battery, you need to use different values of the de-rating factor. For instance, consider the 17.5A·h SLA battery from the previous graph, and assume that your robot needs about 15A to drive around with the weapon turned off, and 100A when it is on. How many minutes would it last with that battery, assuming that it spends 80% of its time with the weapon powered? The answer is obtained calculating the capacity considering the different values of the de-rating factor. From the previous graph, a run time of 0.6 hours (36 minutes) would result in 9A·h, with a continuous current of 9A·h/0.6h = 15A. But a run time of 0.05 hours (3 minutes) would result in only 5A·h, with a continuous current of 5A·h/0.05h = 100A. If the number of minutes to be calculated is t, then the robot spends 0.8·t minutes drawing 100A (3 minute run time), and 0.2·t minutes drawing 15A (36 minute run time), so to completely discharge the battery we would have (0.2·t)/36 + (0.8·t)/3 = 1, thus t = 3.67 minutes.

Let's check the calculations: during the 0.2·t = 0.734 minutes at 15A the robot drains 0.734min/36min = 2% of the battery capacity, and during the remaining 0.8·t = 2.936 minutes at 100A it drains the other 2.936min/3min = 98%. These more sophisticated calculations are not necessary for nickel or lithium batteries, because their de-rating factor varies very little, between 0.9 and 1.0.

8.2.12. Discharge Rate

Finally, the last relevant battery property is the discharge rate, which measures how much current can be continually drawn from the battery without letting it become significantly hot. It is represented by a number followed by the letter C. For instance, 8C means that the battery tolerates without problems a current of 8 times its measured capacity (C, hence the name 8C) in A·h. For instance, a 3.6A·h battery with 8C tolerates continuous 8 × 3.6 = 28.8A without overheating. This is the same as to say that it can be fully discharged in 1/8 of an hour (because 28.8A × 1/8 h = 3.6A·h), which is equivalent to 7.5 minutes.

In practice, most NiCd batteries can withstand more than twice the calculated current from their discharge rate, however they will significantly warm up (but not necessarily overheat, in special if the cells are spaced inside the pack to keep down their temperature). In other words, a NiCd 8C battery could be continuously discharged in only 3.75 minutes, compatible with the duration of a typical combat. Avoid using NiCd batteries rated below 8C, they will very likely overheat during combat.

A few lithium batteries, such as the Polyquest (LiPo) and A123 ($LiFePO_4$), can continuously deliver over 30C and sustain 50C (or higher) peaks without overheating. But, different from NiCd packs, lithium batteries usually do not tolerate current peaks that are more than twice the calculated value from the discharge rate.

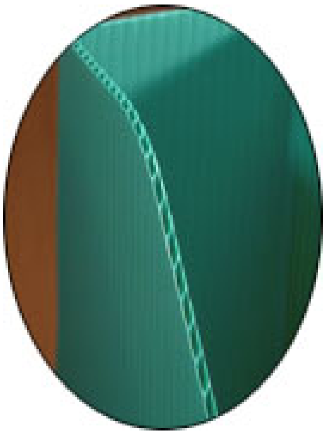
Another way to evaluate discharge capacity is through the power-to-weight ratio of a battery, in W/kg. It evaluates the power that a battery can continuously deliver divided by its weight. SLAs can only deliver about 180W/kg, while nickel batteries between 150 and 1,000W/kg, lithium-ion about 1,800W/kg, and LiPo beyond 2,800W/kg.

8.3. Battery Care and Tips

To make your batteries last longer, it is important to follow several procedures, described next.

8.3.1. Shock Mounting

Make sure that the batteries are very well mounted inside your robot, with some cushioning to avoid impact damages. For instance, the instantaneous accelerations that a robot might suffer during an impact from a violent spinner can reach up to 800G, in other words, 800 times the acceleration of gravity. Only for reference, at 10G a person would faint, and at 100G one's brain would detach from the skull, causing instantaneous death. Therefore, 800G is somewhat frightening, even considering that this acceleration lasts only a small fraction of a second. Quick calculations show that a 4.4lb (2kg) battery would suffer an equivalent inertial force of 2kg × 800 = 3,520lb in its support. Of course this would be an extreme case, but even for much smaller impacts it is evident that zip ties are not appropriate (unless it is a very light pack such as the ones used in receivers). Besides, zip ties might also melt due to the high temperatures that the batteries can reach.



Good materials to shock mount your batteries are hook-and-loops and neoprene. Corrugated plastic, cut from file cases or other office supplies, is also an inexpensive and effective shock mounting material, as pictured to the right.

Be careful not to cause short-circuits, you must isolate very well any metal parts that get in touch with the battery. And you must guarantee in your robot design that the batteries can be quickly replaced, to speed up pitstops.

Note that LiPo batteries expand almost 10% in size during use, so make sure that there's extra room inside your robot not to let them get squeezed too much. Using very compliant shock mounts is a good way to accomplish that.

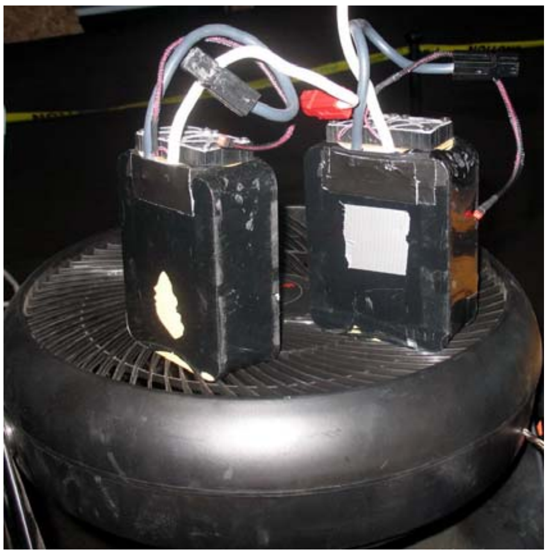
8.3.2. Recharging

To recharge batteries, especially the nickel and lithium types, you must use an electronic charger. They are an indispensable investment. Without them, the chances of damaging batteries are very high. Triton 2 (pictured to the right) is one of the best and easiest chargers to use. It automatically charges or discharges most battery types, with several programming options. It eliminates the infamous "memory effect" that happens when NiCd batteries are not properly charged. It costs a little over US$100 in the US. It is really worth investing in an electronic charger such as Triton 2, or several famous others such as Astroflight, Thunderpower or Dynamite models. It only takes one damaged 24V NiCd pack to set you back more than the price of the charger.

Due to lithium batteries being prone to ignite when mishandled, they need a lot of attention when charged. They need an intelligent charger to charge them following the correct algorithm. This kind of battery cannot be overcharged under any circumstances. In addition, to prevent damages to your robot (and people), always remove the batteries from the robot and charge them inside a fireproof container, such as LP-Guard or LipoSack. The picture to the right shows a LipoSack and the ignited battery that it withstood inside.

If your robot uses more than one pack, you might want to have more than one charger, to be able to charge the entire set of batteries at the same time, in time for the next match. We’ve built a wooden box, which is a good electric insulator, to mount 4 Triton chargers (pictured to the right), together with 12V power supplies taken from old personal computers. The box also carries our battery packs, which is a practical solution for ground transport and use in Brazilian competitions. For international competitions, which require air travel, we have a more compact version of the charger box, shock-mounted inside a suitcase.

Always have at least 2 sets of batteries for your robot. Nickel and lithium batteries can take up to 1 hour to fully charge, however the pit time between rounds can be as low as 20 minutes. If you can afford it, get 3 or more battery sets. Besides solving the charging time problem between rounds, having a third (or fourth) set is an additional insurance in case a pack gets damaged or shorted out during combat. If you use regular SLA batteries, you might need up to 6 or 7 sets during a competition, because they charge very slowly, sometimes taking several hours.



Another important tip is: never charge hot packs. Its useful life would be very much reduced if charged while still hot after a match. As soon as a combat ends, immediately remove the battery packs and put them over a large fan to cool down (as pictured to the right). Only begin to charge them after they get close to room temperature.

8.3.3. Battery Storage

Always store SLA batteries fully charged. If they are stored discharged for a long time, they can be damaged. Completely recharge SLA batteries every 6 months if they're stored.

Unlike SLA, NiCd and NiMH batteries should be stored fully discharged. Be careful not to discharge them too much, nickel batteries should never get below 0.9V per cell. Lithium-polymer and lithium-manganese batteries must never get below 3.0V per cell, while $LiFePO_4$ should never be allowed to drop below 2.8V per cell.

Also, have in mind that heat kills: even if properly discharged, nickel batteries can last less than 2 years if stored at 100°F (38°C). At 77°F (25°C) they usually last 5 years, at 59°F (15°C) they can reach 10 years, and at 41°F (5°C) up to 20 years. Therefore, store NiCd and NiMH batteries inside a refrigerator, fully discharged. Put them inside a sealed plastic bag to protect them from moisture. Never freeze the batteries. Every 6 months or less, make sure to fully charge and discharge them, and put them back in the refrigerator.

To store lithium batteries, first place the packs into a LiPo sack or equivalent, then charge (or discharge) them to 40%~60% of their capacity (LiPo: 3.8~3.9V per cell; $LiFePO_4$: 3.2~3.3V per cell). If your charger does not support terminal voltage configuration, then fully charge the battery and then discharge it monitoring the voltage value with a voltmeter. Finally, isolate the connectors with tape, place the batteries into separate sealed plastic bags, and store them in the refrigerator at about 41°F (5°C). Once a month, check the voltage of your lithium batteries and, if needed, recharge to keep them between 40% and 60% of their capacity.

Be very careful not to short-circuit your batteries, especially the lithium ones. Be attentive when handling screws near the pack, they can fall inside and cause a short-circuit that can permanently damage the battery. There is also the risk of metal debris entering your pack during combat. This may result in the famous "magic smoke" (pictured to the right), which will either disable your robot or result in damage points to your opponent. Some people say that magic smoke is the robot's soul leaving its metal body.

To avoid this problem, you can wrap up each battery in the pack with Kapton tape, a polymer that resists high temperatures, up to 750°F (400°C), besides being a good electric insulator.

If your pack is getting too hot, an option is to install one or two fans to blow air inside it, helping it cool down. There are a few ready solutions in the market. One of the best NiCd and NiMH packs in the market are the ones from Robotic Power Solutions (www.battlepack.com), they sell both the traditional battlepacks and the intercooled ones (pictured to the right).

8.3.4. Assembling Your Own Pack

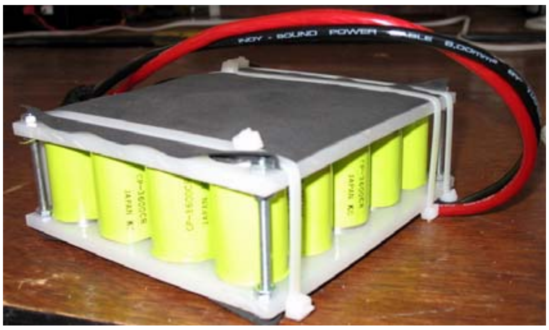
NiCd and NiMH packs are not cheap, so it is possible to save some money if you assemble them yourself. First, buy individual cells, making sure that their discharge capacity is at least 8C, and wrap each of them with Kapton to avoid shorts.

To assemble the pack, weld the cells using flexible copper braids. Rigid connections can break during combat. At the RoboGames 2006 semi-finals, one of the two 24V battery packs from our middleweight drumbot Touro stopped working right after the first impact against the tough rammer Ice Cube. Touro had to fight the entire 3 minutes with only one pack, which made it slow down near the end of the match. This counted as damage, which was decisive in our split decision loss by 17-16. Back in the pits, we've realized that a rigid connection had broken off inside the battery. The pack was cold, indicating that it had broken in the beginning of the match, with almost full charge. This was confirmed from the 2 minutes it later took to be recharged, after the solder was fixed. Since then, we've only used copper braid connections in our batteries.

Use very fine sandpaper on the battery contacts to remove oxidation, which would compromise the mechanical and electrical resistance of the solder. Use a high power solder iron, with at least 100W. Tin as much as possible the battery contacts and the wire, and weld them quickly to avoid heating up and damaging the cell. An important tip is, before welding, to put o-ring spacers among the cells, as pictured to the right. These spacers can be, for instance, cardboard rings held together with shoemaker's glue. By doing so, you will leave gaps among the cells that will allow air to flow, making the heat exchange much more efficient, avoiding overheating them.

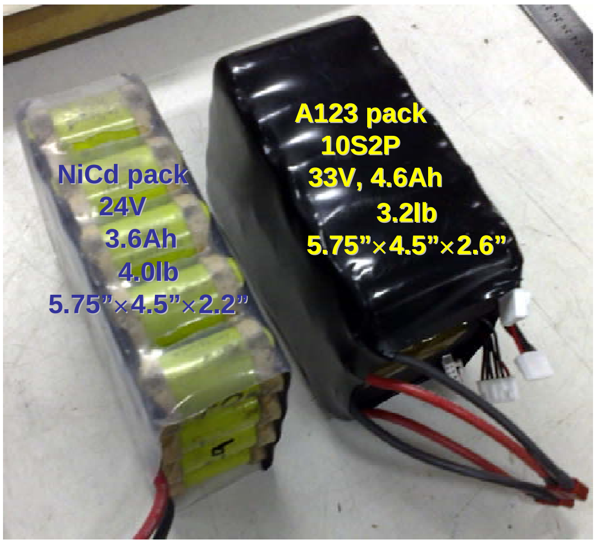
After having soldered all the cells using copper braid, weld the connector and its wires. The connector can be, for instance, a Deans Ultra or the Anderson Powerpole. Remember that the battery should always use the female connector, never the male, to avoid any chances of short-circuit if the connector accidentally touches some metallic part.

Optionally, you can use nylon plates (or some other insulating and resistant material) covering the top and bottom of the pack, as pictured to the right. It will be necessary to use a mill to carve slots in the nylon to accommodate each of the cells. You can then secure the nylon plates with long screws, as shown in the picture.



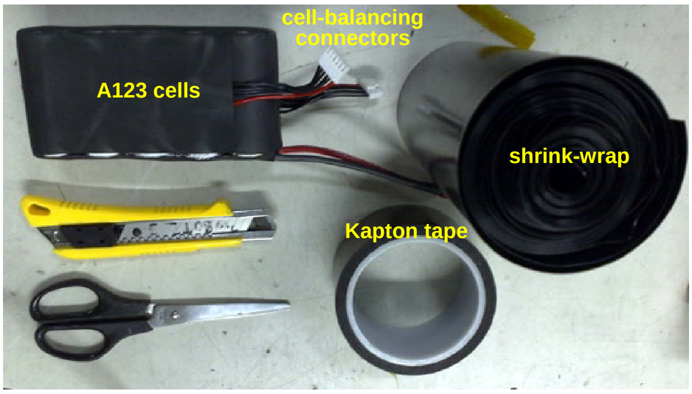
With or without nylon plates, it is advisable to protect the pack with shrink-wrap. If you don't have shrink-wrap, a very cheap alternative is to fit the pack inside a cut Coke bottle, and use a heat blower to shrink it to hold the pack with a snug fit. Using a hot soldering iron, you can make a few openings in the shrink wrap (or Coke bottle) at the spaces between the cells, to improve cooling.

A very similar process can be used to assemble packs made out of A123 packs, which also have a cylindrical shape. But don't forget to solder separate cables and connectors for cell-balancing.



The pictures to the right show a 20-cell 10S2P A123 pack including cell-balancing connectors. The 10S2P configuration features 2 parallel arrays of 10 A123 cells each in series, resulting in a nominal capacity of 2 × 2.3Ah = 4.6Ah and a nominal voltage of 10 × 3.3V = 33V.

This pack resulted in the exact same width 4.5" and length 5.75" of a 24V 3.6Ah NiCd battlepack (including o-ring spacers, as pictured to the right), which allows both to be interchanged without having to modify existing robots to fit them.



The height of the A123 pack, however, is a little higher: 2.6" instead of 2.2" from the NiCd pack. But the A123 pack is actually lighter, because its 20 cells weigh 20 × 70g = 1.4kg, instead of 20 × 88g = 1.76kg from the NiCd CP-3600CR cells. In addition, this A123 pack has higher nominal voltage and capacity than the equivalent sized NiCd battlepack, 33V and 4.6Ah instead of 24V and 3.6Ah. Not to mention the improved properties of A123 cells over NiCd.

An important advice: you should only assemble your own pack if you know what you’re doing. It is not difficult to damage cells by overheating them while they’re soldered. It pays off to have them professionally assembled, for instance, at www.battlepack.com.

8.3.5. Billy Moon’s Rules for LiPo

Finally, a battery care section would not be complete without the rules from famous builder Billy Moon for handling LiPo batteries:

1. NEVER, NEVER, NEVER charge in your robot;
2. NEVER, NEVER, NEVER charge them hot;
3. always charge them in a LiPo sack or steel tool box;
4. check balance on your batteries before each charge;
5. charge them as slow as you can afford to;
6. never short their leads (if you do, toss them out);
7. never remove them from your bot by the leads;
8. allow extra room for them to expand by 10% in all dimensions while in use;
9. never fully discharge them: plan for at least 50% more capacity than you need;
10. bring or arrange to have a 'class D' fire extinguisher on hand.

Chapter

9

Combot Events

This chapter presents several tips related to combat robot events, and how to get ready for them.

9.1. Before the Event

The first step is to find out an event. In the www.buildersdb.com website you'll find all the information on most of the incoming as well as past events. There you'll also be able to register your team, builders and robots, as well as search for other teams and robots. In addition, the Robot Fighting League events can also be checked at http://botleague.net. Don't forget about the registration deadlines, the organizers need to know as soon as possible how many teams will attend to plan accordingly. It's important to register in advance for the events, because there might be a limited number of robots in each weight class. Read carefully all the event rules, to make sure that there are no problems with your robot.

9.1.1. Test and Drive Your Robot

Finish your robot before you travel to the event. There's nothing more stressful than going for an all-nighter on the eve of the event. Especially if you'll be waking up all other hotel guests with grinding noises and the smell of burnt rubber mixed with Dremel disks, as we unfortunately did during our first event back in 2003. Guarantee that your robot will pass safety inspection.

Train driving your robot. A lot. Several matches are won or lost because of the driver's ability. Train slalom using traffic cones. Wendy Maxham suggests a practice technique from Grant Imahara's book [10], in which you mark out a square on the floor and then drive the robot as close to the edges as possible. You'll learn how to drive straight, and how to make sharp turns. Start out slow, then go faster and faster until you reach full speed. Don't forget to train in both directions, to practice both left and right sharp turns.

Another great practice move, suggested by Matt Maxham, is the "James Bond turn." While driving forward on a straight line, quickly spin your bot 180^o and reverse the wheels to keep driving in the same sense (you'll be driving backwards, but in the same original sense). Then spin again, to make the bot face forward, always moving in the same sense. It is a good maneuver to make your weapon face the opponent while you're escaping from it. It is also a great maneuver to shoot your pursuer during a car chase, if you're James Bond.

During a combat, you can't waste time thinking about which way to steer, left or right, which can be tricky if your robot is moving towards you. As Matt Maxham says, you (the driver) need to imagine that you're sitting on top of the bot, then you'll naturally steer in the correct direction.

Buy a cheap remote control car to play cat and mouse. Actually, buy more than one, they usually don’t survive when you catch them. In early 2003 we created a toy overhead thwackbot out of a plastic remote control car (pictured to the right). The robot itself, while driven, was useful to improve our skills controlling overhead thwackbots in general, which are very tricky to handle. And this toy robot also doubled as a very fast and efficient “mouse” when chased by our first middleweight combot *Lacrainha*.

Always test your robot. Test it under real conditions, drive it against a wall, several times. Use its weapon (if any) to hit on junk parts with up to the same weight as your robot. Try to hit solid one-piece blocks, to avoid having small parts flying all over. A low hardness sparring is a good idea to avoid blunting your weapon, such as the 7” diameter solid aluminum block that our combots like to play with, pictured to the right. Use the hit-break-fix it technique, until your robot does not break anymore. It is important to test the robot well in advance, to make sure that there will be time to fix it before the event.

Drop your robot from 3 feet (about 1 meter) in the air over a rigid floor. It needs to resist this fall, no matter which weight class it belongs to. From fairyweights to super heavyweights, all of them are usually thrown higher than that during battle. Drop it several times and always verify if something broke or got loose. If you really trust in your robot’s resistance, try dropping it from 6 feet (about 2 meters). Most well designed combat robots can survive such 6 foot fall. During RoboGames 2006 we were able to verify that: the heavyweight Sewer Snake (pictured to the right) was still functional even after it was launched several feet into the air by the super heavyweight Ziggy.

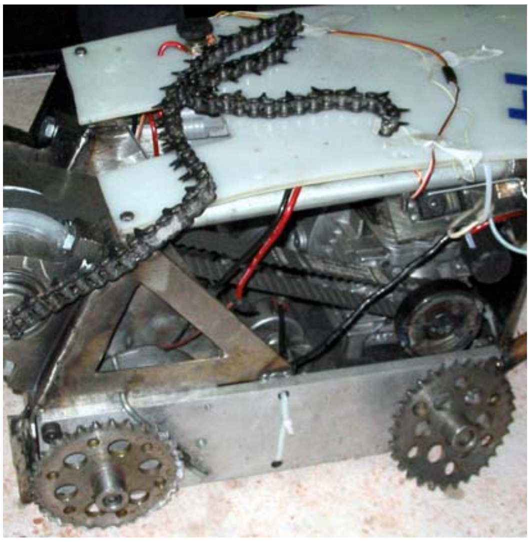
Tests will expose the robot’s weak points. It is usually possible to correct them without changing too much the original design. Even a tiny flaw can sometimes be enough to make you lose a match. Therefore, it is very important to have redundancy. For instance, if your robot uses 2 or more batteries in parallel, guarantee that it will keep moving if one of them fails. If your robot has 4 active wheels, guarantee that if one of them is destroyed all the other three will still be able to drive it. If you use belts or chains in a critical component such as the weapon, consider the possibility of using a double pulley or double sprocket. During Touro’s first match ever, at RoboGames 2006, its opponent was able to tear one of the drum V-belts. However, Touro’s weapon continued to spin because of the redundancy from the second V-belt, allowing it to win the match by knockout.

9.1.2. Prevent Common Failures

The 20 most common causes for a combat robot to lose a match, according to the website http://www.solarnavigator.net/robot_building_tips.htm, and our proposed solutions, are:

1. Battery connectors or other wires getting loose – always use good quality connectors such as Deans or Powerpole, always use ring terminals (not fork terminals), tighten each terminal connection using pressure washers (but never place them between the electric contacts, since their electric resistance may be high), and use liquid electrical tape or hot glue;

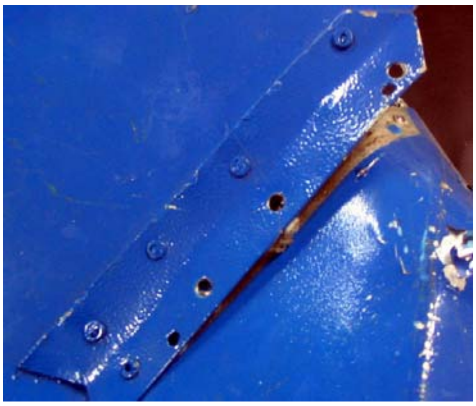
2. Motors, batteries or other components getting loose – avoid using nylon ties or clamps, even the metal ones, always verify any loose parts and tighten critical screws before each match, use threadlockers or spring locks;



3. Chains coming off from the sprockets – make sure that the sprockets are well aligned, avoid exposed sprockets (as in the special drive system for ice arenas from the robot pictured to the right); if possible, replace the chains with timing belts (such as in the weapon system of the same robot to the right), which can withstand larger misalignments;

4. Radio interference or signal loss – if using 75MHz or lower frequency radios, place the antenna outside the robot, without touching the metal surfaces, and use an amplified antenna such as the Deans Base-Loaded Whip (chapter 7); if using 2.4GHz or higher radio frequencies, or if the covers are not metallic, then the antenna can be left inside the robot;

5. Improperly charged or low capacity batteries – always use electronic chargers such as Triton, check the battery voltage before each match, and calculate and test the robot's consumption under real conditions; before an event it is a good idea to apply several discharge-charge cycles to the batteries;



6. Smoking speed controllers – always match the maximum acceptable current in the controller with the motor specs; if a wheel or a spinning weapon gets stuck during a match, turn it off to avoid stalling the motor;

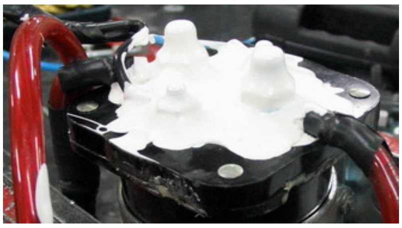
7. Rupture of rivets, screws, nuts – never use rivets (seen in the picture to the right), always use hardened steel screws and nuts, class 8.8 or 10.9 for hex screws and 12.9 for Allen, and with appropriate diameters as discussed in chapter 4;

8. Low clearance robots getting stuck in the arena – from hobbyweights to super heavyweights, make sure that you have a ground clearance of at least 1/4”, preferably 3/8” or more; don’t forget to consider the wear and tear of the wheels, and use flat head screws on the robot's bottom cover to make sure they won’t get stuck on the arena floor;

9. Low power wheel and weapon motors – use motors with enough power to guarantee that your robot’s drive system acceleration is high enough for your strategy, preferably taking less than 2 seconds to reach top speed; if you have a spinning weapon, make sure that it can reach damaging speeds in less than 4 seconds;

10. Burning fuses – do not use fuses in combat, otherwise you can lose a match only because of a brief current peak; use a current limiting circuit if needed, but not a fuse;

11. Shorting of wires and electric components – always isolate the terminals with electrical tape (either regular or liquid, as pictured to the right), heat shrink and/or hot glue; always protect any electronic component that can be shorted out if metal debris enters the robot;



12. Overheating motors – avoid overvolting too much the motor; several motors can take up to twice their nominal voltage, but it might be necessary to use a current limiting circuit; in a few cases it is possible to mount fan blades onto the rear end of the motor shaft to improve cooling; avoid stalling the motors during combat;

13. Broken gears – all gearboxes need to be well designed and built, with well aligned and precisely spaced gears; the gear thickness and teeth dimensions must be proportional to the torque it carries; therefore, to optimize weight, use heavier duty gears in the last stage and lighter ones in the first; always use hardened steel gears instead of mild steel or cast iron ones;

14. Internal combustion engines that die or won’t start – use an automatic ignition system, controlled by a separate radio channel (see chapter 5);

15. Shaft mounted components getting loose – never use set screws or pins, either in shaft couplings (as pictured to the right) or in other shaft mounted components such as pulleys or sprockets; always use keys and keyways (or keyless bushings such as Trantorque) to transmit torque, and very tight shaft collars to avoid axial displacement;

16. Broken or bent shafts – never use shafts made out of mild steel (also pictured to the right) or aluminum, always use hardened steel or titanium grade 5; make sure that the shaft diameter is large enough to keep the stresses below the material yield strength;

17. Wheels getting stuck in the robot’s bent structure or armor (also pictured to the right) – leave a significant clearance between the wheels and any armor or structural part of the robot that could get bent;

18. Flat tires – use solid wheels such as Colsons or, if using pneumatic wheels, make sure that they're filled with polyurethane foam, such as the NPC Flat Proof wheel pictured to the right; a few pneumatic kart wheels, aimed for rental karts, are so sturdy that they could be used in a robot without having to be inflated or filled with foam;

19. Robot failure due to arena hazards – this only applies to arenas with hazards, such as saws coming out of the floor or large sledgehammers; against saws, make sure that your robot has a thick bottom plate or cover it with alumina tiles; against sledgehammers, use a shock mounted top cover;

20. Home-made speed controllers and electronics – building a reliable speed controller that can withstand hundreds of amps is not a simple task, do your research and thoroughly test your system if you plan to develop it by yourself; see chapter 7 for more information.

9.1.3. Lose Weight

Make sure that your robot is not over its weight limit. When designing it, estimate the weight of all the components, to avoid unpleasant surprises. CAD programs can provide very precise calculations if you feed them with the correct part weights and material densities.

And don't forget to include the weight of the screws. We forgot to include the screws when carefully designing and calculating Touro's weight back in 2006, just to find out after it was built that it was almost 6.5lb (almost 3kg) overweight. Just because of the screws. To lose weight, there are a few techniques, as described next.

Rearrange your robot's components

If you're still in the design phase, try to rearrange the robot's internal components to reduce the chassis dimensions. If your robot has several empty spaces in it, it won't be difficult to make it smaller.

Consider all possible component arrangements, but don't forget to leave enough space for the wiring. Try placing the batteries in different orientations.

If it's a 4 wheel-drive design with 4 motors, try using only 2 motors with a timing belt or chain transmission to drive all wheels. If your design does not depend too much on traction, such as with powerful spinners, try using only 2 wheels, with the robot's center of mass located close to the line that joins their centers.

Change the battery type

Switch SLA batteries to NiCd or NiMH. Most 24V SLA batteries have a capacity density of about 1.25A·h/kg, however this number does not consider the de-rating factor (see chapter 8) for a 3 minute run time. It does not consider the worst case scenario, where it will be fully discharged at the end of a 3 minute match. The de-rating factor in this case is about 0.28, which would result in a capacity density of only 0.28 × 1.25A·h/kg = 0.35A·h/kg.

The de-rating factor of nickel batteries in 3 minutes is much better, about 0.9, therefore a typical 24V NiCd pack would have 0.9 × 2.1A·h/kg ≅ 1.9A·h/kg, while a typical 24V NiMH pack would have an even better 0.9 × 2.9A·h/kg ≅ 2.6A·h/kg. In this way, without decreasing the robot's battery capacity, you can lose 80% of the battery weight when changing from SLA to NiCd. Changing from NiCd to NiMH will result in an additional weight loss of about 30%. But be careful with NiMH packs because, despite their greater capacity, they cannot supply the high current peaks that a NiCd pack with same capacity can, which makes a big difference especially for the weapon acceleration.

To lose weight even more, you could migrate to lithium batteries (see chapter 8), such as Li-Po, Li-Mn, or Li-Fe-PO_4 (A123 or K2), however with a higher cost.

Reduce shaft dimensions

To lose weight, try reducing the diameter of the robot's shafts. This can make a difference especially if they're made out of steel, which has a high density. This will also reduce the size and weight of other components such as bearings and their mounts. Check if the shaft length can be reduced. Using a lathe, drill an internal hole through the entire shaft, as long as it hadn't been tempered, to transform it into a cylinder. If the shaft has diameter D and the hole d, the weight will be decreased by a factor d^2/D^2, while the bending and torsion strengths (which usually are the most important in shafts) will decrease by only a factor of d^3/D^3. In this way, for instance, if a hole with diameter d = D/2 is drilled, the shaft weight will decrease in $(D/2)^2/D^2 = 0.25 = 25\%$, while the bending strength will only be $(D/2)^3/D^3 = 0.125 = 12.5\%$ lower.

Change the shaft material

An excellent technique to lose weight, although costly, is to switch all steel shafts to titanium grade 5 (Ti-6Al-4V), without increasing their diameter. We had to do this with Touro, its 1.5" diameter weapon shaft was originally made out of tempered 4340 steel, weighing 6lb (2.7kg). Since it had already been tempered, it would be very hard to drill it, as explained above, to lose weight. The solution was to replace the shaft with a Ti-6Al-4V one, which only weighed 3.5lb (1.6kg), resulting in a 2.5lb saving. And the shaft strength was not significantly lowered, because this titanium alloy has excellent mechanical properties. The cost was not too expensive, considering that it is a critical part of the robot: about US$150, in the US.

Avoid the temptation to switch steel to aluminum in shafts. If the shaft diameter is maintained, low and medium strength aluminum alloys will easily yield, and most aerospace alloys will possibly break due to their lower impact toughness. Aerospace aluminum could result in a lighter shaft with the same strength as a steel version, but the increased diameter needed by the resulting shaft would significantly increase the weight of its bearings, collars, and all other shaft mounts. So, as extensively discussed in chapter 3, choose titanium and hardened steel shafts instead of aluminum or magnesium ones.

Change the material and dimensions of robot components

Wisely changing the material of a robot component is not a simple task. This was thoroughly discussed in chapter 3. The best material choice to reduce weight depends on the functionality of the component. For instance, if a robot's armor is shock-mounted to its structure, then most structural parts could have their stiffness maximized without worrying too much about impact toughness, while the armor should withstand impacts without worrying too much about its stiffness. In this case, very thick magnesium or aluminum alloys would be a good choice for a light structure, while thinner titanium Ti-6Al-4V would make a tough and light traditional armor.

But there are several other cases and options. See chapter 3 for a more detailed discussion on weight saving techniques based on changing both the material and dimensions.

Reduce the thickness of plates

If after optimizing the materials of the entire robot it is still too heavy, then it might be necessary to decrease the thicknesses of its plates.

The first idea that comes to mind is to drill holes in the plates, turning them into Swiss cheese. This should only be considered in an emergency, during the event. Holes are a bad choice, because they let debris enter the robot, which can short out the electronics, not to mention the higher vulnerability against hammerbots, spearbots or overhead thwackbots with thin weapon tips, which could reach internal parts, as well as against flamethrowers. Besides, circular holes have a stress concentration factor of about 3 under tension and 2 under bending. In other words, even a small hole will locally multiply the mechanical tensile stresses by 3 and bending stresses by 2 (the stress concentration factors of several geometries can be seen in the Appendix C). These higher stresses make it easier to initiate cracks at the borders of the hole.

In addition, you would need too many holes to significantly lose weight, as seen in the next example. Consider, for instance, a 0.5m × 0.5m cover plate made out of 1/4" (6.35mm) thick aluminum. Its mass is approximately $2800\text{kg/m}^3 \times 0.5\text{m} \times 0.5\text{m} \times 0.00635\text{m} = 4.45\text{kg}$ (9.8lb). Let's try to lower its mass in 25%, to 3.33kg (7.3lb), using a hole saw to drill several 1" (25.4mm) diameter holes. Each hole would only relieve $2800\text{kg/m}^3 \times \pi \times (0.0254\text{m})^2/4 \times 0.00635\text{m} = 0.009\text{kg}$ (0.02lb). In other words, to lose $4.45 - 3.33 = 1.12$kg, you would need $1.12/0.009 \cong 124$ holes! In addition to the hours spent drilling 124 holes, the robot would suffer from the problems discussed above regarding debris, piercing opponents and flamethrowers.

A better solution is to mill the aluminum plate. In the previous example, we could decrease the plate thickness down to 3/16" (4.76mm) through milling, resulting in a 25% lighter plate. The bending stress of a plate depends on the square of its thickness, therefore it would be multiplied by $(6.35/4.76)^2 = 1.78$ in the thinner plate, which is lower than the factor 2 that would be obtained by drilling holes. And, since the tensile stress along the plate depends directly on its thickness, it would be multiplied by a factor $6.35/4.76 = 1.33$, much smaller than the tensile factor 3 of the holed version. Therefore, the milled plate would have a higher strength than the holed one.

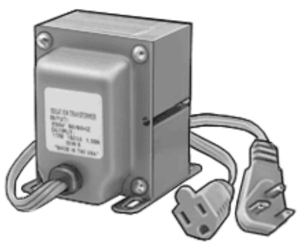
An even better solution is to selectively mill the plate. In other words, to reduce the thickness of the plate only in a few areas, leaving it with the original thickness at the most stressed areas. This was the procedure adopted in the 5/16" thick top cover plate of our middleweight Touro, to lose weight. We've selectively milled 1/8" deep pockets on its outer surface, as pictured to the right. The thickness was neither reduced near the screws (not to compromise strength) nor where the weapon motor is mounted (by the RioBotz logo in the picture). The strip-shaped area between the pockets was also kept with its original thickness, acting as a rib to keep high the plate bending stiffness.

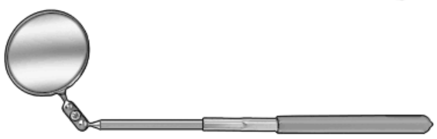
9.1.4. Travel Preparations

Once your robot is built and tested, making sure that it is not overweight and that it complies with all the event rules, then the next step is to make travel arrangements. Plan the trip well in advance, you'll get cheaper fares and hotel rates.

What to bring

Before the trip, make a list of tools. Avoid the temptation (which I have) of bringing your entire machine shop to the event. Choose wisely which tools you'll really need, among the ones listed in chapter 2. A few very useful items, but usually forgotten, are a portable vacuum cleaner (pictured below, to clean the robot interior in between matches, because small metal debris can cause shorts), a large fan (to cool down the batteries after each match, before charging them), a 220V/110V transformer if needed (pictured below, rated to at least 1kVA if using several power tools and chargers at the same time), heavy-duty electric extension cord, plug strip (pictured below), flashlight (for repairs inside the robot, preferably with a swivel head, as seen below) or headlight (for hands-free operation), telescopic mirror (to inspect the robot's interior without disassembling it, see picture), telescopic magnet (to pick up screws or nuts that fall inside the robot, see picture), and J.B.Weld and duct tape (for desperate emergency repairs in the robot). And don't forget the battery chargers and their power supply.


Have at least 2 sets of batteries, 3 or more if possible, and bring spare parts. It's a good idea to have robots that share parts, in this way you'll need to carry fewer spares. For instance, because both Touro's drivetrain and Touro Light's weapon system use Magmotors S28-150, it might be enough for both robots to only bring one spare. They also use the same front skids, battery packs, receivers, Victor speed controllers, TW C1 solenoids and MS-2 switch, not to mention several of their 8mm diameter screws. This helps a lot with spare part management and transport, in special if competing in overseas events.

Bring spare screws, in special if they're oddly sized or difficult to find. Remember that it will be more difficult to find metric screws to borrow in US events, and vice-versa, few Brazilian builders will have inch sized screws to lend. This also applies to tools that come in different systems of measurement, such as wrenches or sockets.

Traveling by plane

If you're traveling by plane, remember that most robot parts won't be allowed in the cabin, they'll need to be checked. Since your checked luggage will most likely be X-ray inspected and opened, it is a good idea to write down on each checked robot part what it is, such as "discharged dry cell battery pack" or "aluminum plate."

When traveling to RoboGames, we also include in every luggage a copy of Dave Calkins' invitation letter, explaining that the robots and parts are for competition purposes only. We also carry a picture of each robot (or its trading card, if it has one), in case the luggage is opened in our presence. In this way, it is easier to explain to the TSA officer why we're checking a sharp spinning bar or so many aluminum plates. If you're lucky, the TSA officer may even know your robot, as it once happened with Touro, making the inspection process very fast and friendly.

We've never had problems traveling by plane with NiCd packs, as long as they are discharged and placed into the checked luggage. Since no wet cell batteries are allowed in the plane, it's a good idea to write "discharged dry cell battery pack" on every pack.

Also, make sure that all the electric wires are very well organized and placed inside a different luggage than the one with the battery packs. Trust me, a luggage full of NiCd packs and random electric wire will draw a lot of unnecessary attention in the X-ray.

Apparently, lithium batteries such as LiPo or A123 can be carried with you inside the cabin, we've never had problems with them even when they were inspected. Otherwise, notebooks and their batteries would need to be forbidden as well. But they can be dangerous if shorted, so we always carry them partly discharged (not too much to avoid damaging them) and inside a fireproof LiPo sack such as LP-Guard (pictured to the right) or LipoSack.

You can ship your robot fully assembled in a crate. However, for international flights, you might need to apply for a temporary export of your robot if it is shipped fully assembled in a large crate. This is usually expensive, and it involves a lot of bureaucracy. Shipping the robot by sea is also risky, because even if sending it well in advance it might arrive at the event after it is over.

The cheapest solution is to carry your robot in your checked luggage, not in crates. If your robot is a middleweight or from a heavier weight class, you will need to partly disassemble it if you want to split it into two or more pieces of luggage. Lightweights or lighter robots can be checked fully assembled if the weight limit allows and if they fit inside the luggage. Our lightweight Touro Light is checked inside a 10lb luggage, reaching exactly the 70lb allowance for each checked item in international flights originated in Brazil.

A very good investment is to buy a digital scale, bringing it with you to weigh all pieces of luggage before each flight, using up the entire weight allowance. The 150lb capacity Pelouze digital scale with remote display, pictured to the right, is a good option.

Note that fees are less expensive to check in an extra luggage than to have several overweight bags. For instance, international tickets bought in Brazil have a 70lb allowance per bag, allowing 2 checked bags, with a US$100 fee for a third piece of luggage. Checked items between 70 and 100lb are subject to a US$100 fee per item, and bags over 100lb are forbidden. Therefore, if you're carrying for instance 180lb worth of robot parts, it is better to pay US$100 for an extra 70lb bag (which will allow you to carry 60lb worth of parts in each of the three bags, as long as each empty bag weighs up to 10lb), instead of paying US$200 for two overweight bags with 100lb each (90lb worth of parts in each plus the own weight of the luggage).

Get to the airport well in advance, in special because of the odd and heavy luggage you'll be carrying. For international flights, register the robot parts at the customs office from your airport of origin before leaving the country, it will simplify your reentry with them. You can do this in the same day of your departure, before checking your bags, but you need to arrive early. You only need to register foreign parts, but it is also a good idea to register custom-made parts such as the robot itself or large disassembled parts of its chassis. Parts might need to have a serial number to be registered. If they don't have it, check with the customs officers if they accept serial number plaques issued from a University, for instance. Most manual and electric tools don't need to be registered, they're considered as tools for professional use. But it is always a good idea to register any expensive part of tool.

Finally, if you're carrying a lot of weight, then rent a car or van at your destination. Choose to pick it up and also to drop it off at the airport. It's less expensive and more practical than riding a taxi carrying heavy luggage full of robots all over town.

9.2. During the Event

Finally, the great day(s) has arrived. How will the event be? We will describe the typical procedures based on our personal experience at RoboGames, in the US.

9.2.1. Getting Started

After getting your badge and the ones from your other team members, you will be assigned a table in the pits, where you'll place all your tools and robots. Unless you are competing with a single featherweight, you'll probably have to manage well the pit space to store all the robots and tools, as pictured to the right. Try to place the robots, all important and frequently used tools, radios, batteries and chargers on the table. Place all electronic equipment (such as soldering iron, electronic board support), radios, batteries and chargers close together on one end of the table, and the robots closer to the other end. In this way, there's less chance of dropping some heavy component on a delicate electronic board. Make sure you have plug strips on both ends of the table, one for the delicate electronic equipment, and the other to be used in power tools while working on the robots. The remaining items, such as large or infrequently used tools, should be placed under the table in an organized way, easily accessible.

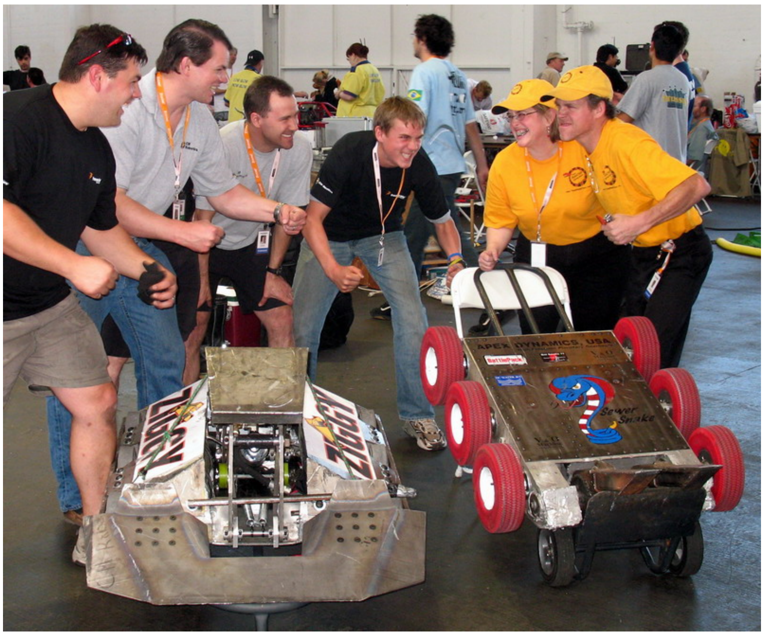
Do not place any items on the neighboring tables, used by other teams, even during a frantic pitstop, unless they allow you to do that. I'm not a good example of this, I'm sorry if RioBotz ever invaded your table...

Arrive early. Try to pass safety inspection on the first try, and as early as possible. "This will give you time to relax and socialize with the rest of the competitors," as pointed out by Mr. Tentacle in his webpage http://architeuthis-dux.org/tips.asp.

Organize the team members so that each one of them has a defined function. Label your tools. Make sure that everybody knows where each tool is stored, either on top or under the table, and always return this tool to its place after using it. This can make a difference during a quick pitstop.

Have a notepad to write down all the ideas that you have during the event. Ideas will come either from what you've learned talking with other builders, or from what your robot learned while struggling in the arena. This information will be very useful later. Several important upgrades in our robots came from crumpled pieces of paper covered in grease and pizza sauce written during the event. Also, don't forget to tape and to photograph the entire event, several ideas will come up while reviewing the pictures and videos.

During the event, it is important to keep in mind that rivalry should stay inside the arena. Unlike these famous builders on the right, do not tease your opponents! Unless if it’s playfully, of course.



Talk with other builders, show them your robots, exchange information, lend tools. This sport is still relatively small, it is fundamental to help other teams and to learn from them, to improve the level of the competition, attracting spectators and sponsors.

Don't be afraid to show the interior of your robot to other builders, even if you'll face them in the next match. There aren't many secrets in this sport that haven't been revealed, in special if you search through the great number of websites, posts, build reports, tutorials and books on the subject. If you don't show your robot to other builders, you'll probably waste the chance of learning from their comments about your robot or from exchanging information by looking at theirs.

Let other people take pictures or tape your robot. This is good for your sponsors, in special if their logo shows up in the pictures or videos. Let them take pictures from your robot's interior. Even if another builder discovers a small weak point in your robot, he/she won't be able to explore it in the middle of a fight, there's not enough precision in combots to deliver a surgical strike. If your robot has a serious weak point, any experienced builder will figure it out even if you try to hide your bot. So, let them look at it, thoroughly if they ask to. In the next chapter, all RioBotz robots are exposed in details, including their interior components, through pictures and CAD drawings.

Walk along the pits to check the robots from the other teams, as pictured below. Unless the other builders are too busy repairing their robot, try to talk with them. But always ask for their permission to take pictures from their robots, in special if you want to touch some part of the bots. If you need to borrow a tool, these teams will most certainly be much more helpful if you have been polite with them.

A nice picture from the RoboGames pits is shown in the next page.

9.2.2. Waiting for Your Fight

Pay attention to the schedule of your next fight, not to get caught by surprise. Even if your fight will be much later that day, have your robot ready and checked. If you check your robot well in advance, you'll have more time to fix any eventual problems. In addition, if your opponent is also ready beforehand, you both can ask the event organizers for an earlier fight. Win or lose, this will leave more time after the match for you and your opponent to fix your robots.

About 30 to 40 minutes before the scheduled time of your fight, charge the robot's batteries one last time, to compensate for any self-discharge, which can be significant in nickel batteries. After this brief charging period, check the battery voltage with a voltmeter and close the robot.

If you're using wheels with polyurethane treads, such as Colsons, it is a good idea to clean their treads using WD-40. Just spray a little bit on the tread, all around the wheel, and wipe it off with a dry cloth or paper towel. Even though WD-40 is a lubricant, it will start to react with the polyurethane tread surface, making it very sticky and improving the robot's traction. The downside is that the arena dirt will also tend to stick to the treads, meaning that you'll need to clean the wheels before every match. But it is worth it. Another great suggestion to improve traction is to engrave grooves on the polyurethane treads. In addition to the use of WD-40, we manually carved in our hobbyweights the Z-shaped grooves seen on the right, improving wheel traction a lot, in special in dirty arenas.

It is also a good idea to mark the robot’s bolt heads with, for instance, a Sharpie. Then it will be easy to know if one of them got loose and needs to be retightened. After applying threadlockers and tightening each bolt, you’ll just need to draw a short straight line starting on the bolt head, and extend it onto the robot structure. Before the match, the very existence of the markings will help you make sure that all bolts have been tightened and have threadlockers. And, after the match, it will be easy to spot any loose bolts just by checking the alignment between the markings on the bolt head and on the robot structure, such as in the middle screw from the picture to the right. We’ve developed this technique after riding too many roller coasters and observing their similar bolt head markings.

If your robot does not use a spread spectrum radio system such as a 2.4GHz one, then you have to pick up the appropriate transmitter clip featuring the channel you're using, as pictured to the right. For instance, for a 75MHz radio system, you'll need to pick up a clip corresponding to one of the channels between 61 and 90 (see Appendix D). It is forbidden to turn on any radio without the clip, to avoid accidents that could happen if another robot uses the same channel as you. More recently, several competitions have required the use of radio systems featuring some sort of binding, such as the 2.4GHz ones, eliminating the need for radio clips.

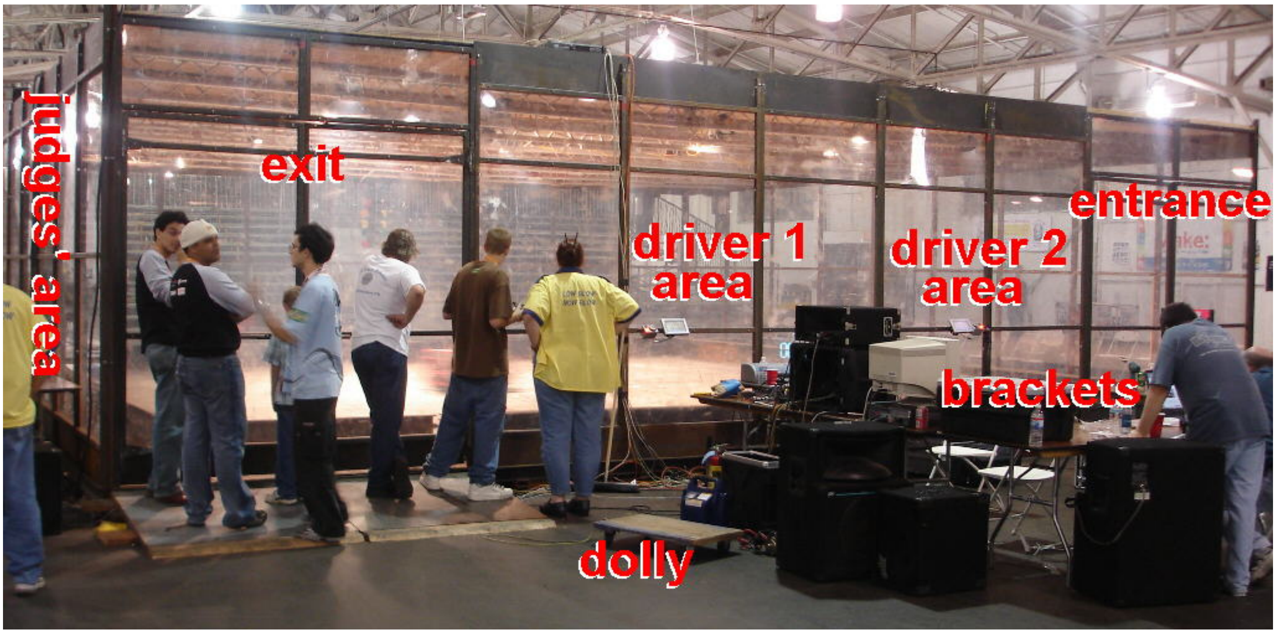
If the event staff allows, make a quick drivetrain test at the pits, but with your robot well secured and with its wheels lifted off the ground. Do not test the weapon, and use some weapon restraint at all times. The restraint should only be removed inside the arena, after you've been told to do so.

Go to the queue with your robot as soon as you're called. Lightweights or heavier robots should be carried on a dolly (as pictured to the right) or pushcart, to avoid accidents such as dropping them on the floor. Once at the queue, you will be standing beside your opponent (as seen in the picture). Exchange conversation, show your robot. Do not be afraid of answering any questions about your robot. At this point it won't make any difference, it is just a way to talk and relax.

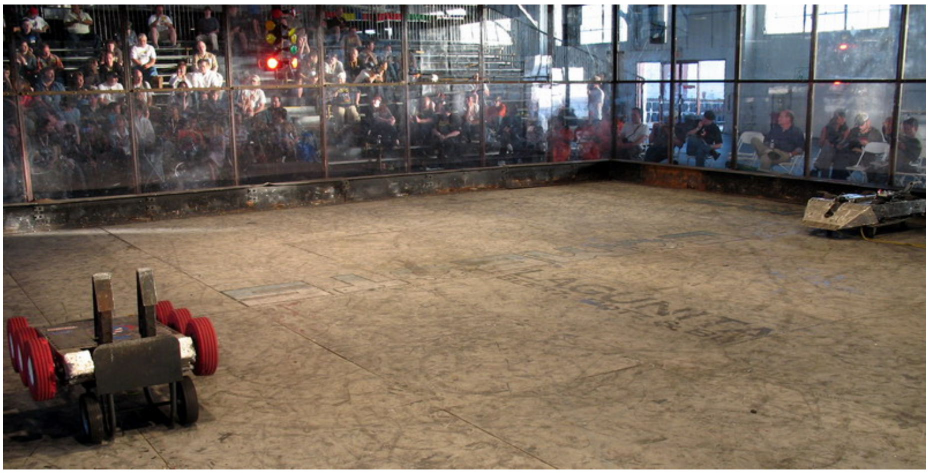
If your next opponent asks for the match to be postponed, and the event staff allows it, then don't hesitate to agree. You came all the way here to fight, not to win by WO. The spectators, pictured to the right, came here to see exciting combats. They might boo you and your robot if you don't agree to grant a brief postponement.

9.2.3. Before Your Fight

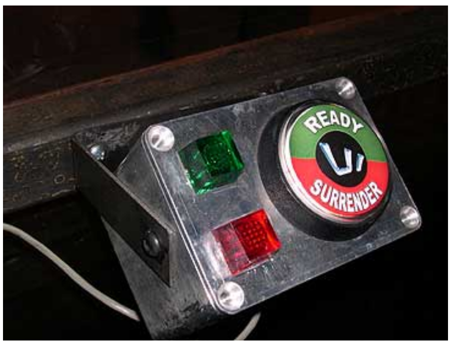
A typical arena in the US has two doors, one for the robots from the next match to enter, and another one for them to leave, as pictured below. Next to the arena there is usually a table with a computer that allows you to check in real time the fight brackets and schedule. Next to the arena there is also the judges' table. If you're the robot driver, enter the arena from its entry door when your called, carrying your robot with a dolly or pushcart.



After entering the arena, you will take your robot to its starting position, as pictured to the right, which is determined by the event staff. Wait beside your robot. When requested, turn on the robot and remove the weapon restraint (if any).



Get outside the arena and position yourself in the areas reserved for each driver. After the arena is locked, you can touch very briefly the radio control just to see if the robot is responding. When you're ready, press the button "Ready / Surrender" from your driver area, pictured to the right. A few seconds after both drivers have pressed their buttons (which seem to last eternally), a series of lights will be turned on, until the green one is lit, starting the fight.



9.2.4. During Your Fight

The matches usually last up to 3 minutes, except for insect classes in the small arena, where the matches are restricted to 2 minutes. Check the specific rules of your competition. The complete set of RFL rules can be found in www.botleague.net/rules.asp. A few of them are described next.

If a robot does not move for 5 seconds after the opponent has ceased attacking, a 10 second countdown will be issued, at the end of which it will lose by KO if it doesn't show any controlled translational movements.

Pinning or lifting your opponent is allowed, however it is limited to 15 seconds (10 seconds for antweights or lighter). After releasing the opponent, you must move far enough away to let it escape from that pinning position.

If a robot gets stuck on the arena through its own action, not due to some direct action of the opponent, then, depending on the event rules, it may (or may not) be granted one free release per match. If the combatant becomes stuck again during the same match, no intervention will take place: it will have 10 seconds to free itself not to lose the match by KO.

Arenas usually have a Death Zone. The first robot to contact the floor on the Death Zone is declared dead, regardless of which robot initiated the entry.

To surrender during a match, just press again the button "Ready / Surrender." Sometimes it is wise to surrender if your robot has suffered enough damage to make it impossible to win the match, in special if you're still on the winners' bracket in a double elimination competition. This will prevent further damage and allow you to rebuild the robot in time for the next match. But always think twice before throwing in the towel, not to regret it. Even if your robot is barely moving, there's a chance that the opponent robot suddenly dies for some reason, in special in very violent matches. But if your expensive electronics and batteries are hanging out of your robot, and your opponent seems to be in good shape, then don't hesitate to surrender.

At the end of a match, when requested to, put back your robot’s weapon restraints (if any) and turn it off. Greet your opponent, independently of the result. Remember that your opponent was just trying to win, he/she didn't have anything to do with the judges' decision.

9.2.5. Deciding Who Won

Don't argue with the judges, even if you don't agree with their decision. Their decision is final. Sometimes, from the point of view of the loser combatant, a decision on a very close match might seem unfair. This does not mean that it was a wrong decision, it could just be a matter of point of view, since there always is a subjective aspect in the judgment. The proof of that is the very existence of split decisions.

This is why the judges follow very specific and objective guidelines, which are summarized below, extracted from the website www.robogames.net/rules/combat.php. It is important that all combatants are familiarized with these guidelines, so they can better understand the reasons behind the judges' decisions.

An odd number of judges, usually three, decide the winner of the matches where no robot is defeated during its 3 (or 2) minutes. There is also one Judge Foreman, who ensures that all judges are conforming to the guidelines.

In a judges' decision, the points awarded to the combatants by the panel of judges are totaled and the robot with the majority of points is declared the winner. Points are awarded by each judge in two categories: aggression, worth 5 points, and damage, worth 6 points. All 11 points must be awarded by each judge, who determines how many points to award each combatant. Therefore, a 3-judge panel will award a total of 33 points, which must be equal to the sum of the scores of both robots. Therefore, the closest possible win in this case would be by a score of 17-16, which can only happen in a split decision by the judges.

Aggression

Aggression is based on the relative amount of time each robot spends attacking the other, in a controlled way. Attacks do not have to be successful to count for aggression points, but the attacking robot must move towards the opponent, not just wait for it to drive into the attacker weapon.

The distribution of the 5 aggression points from each judge between the robots is of three types:

- a 5-0 (or 0-5) score, if one of the robots never attempts to attack the other, while the other consistently attacks;
- a 4-1 (or 1-4) score, if there's significant dominance of attacks by one robot, with the other only attempting to attack a few times during the match;
- a 3-2 (or 2-3) score, if both robots consistently attack each other, or if both robots only attack each other for part of the match. If both robots spend most of the match avoiding each other, then the judges will decide which one made more attempts to attack, awarding it 3 points and 2 to the other robot. Note that a robot that attacks a full-body spinner, intentionally driving towards it, is automatically considered the aggressor in the attack.

Note that there can be no ties in aggression, since its number of points is odd. Judges must decide which robot is more aggressive than the other.

Damage

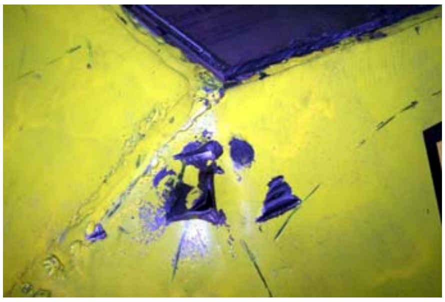
Damage points are awarded to the robot that can make the opponent lose functionality in some way. Damage does not have to be visually striking, it has to do with functionality, with incapacitating the opponent. For instance, titanium will send off bright sparks when hit, but most of the time it will be undamaged. Also, a gash in an armor plate may be very visible, but it only minimally reduces the armor's functionality.

But a bent armor or wedge that prevents the robot from resting squarely on the floor, reducing the effectiveness of the drivetrain, counts as damage. A small bend in a lifting arm or spinner weapon may dramatically affect its functionality by preventing it from having its full range of motion, so it is also considered as damage. A wobbly wheel is also a sign of damage, probably indicating a bent shaft, compromising drivetrain performance.

There are 6 levels of damage:

- trivial: being flipped over causing no loss of mobility or loss of weapon functionality (such as in an invertible drumbot that is able to spin its drum in both directions), direct impacts that do not leave a visible dent or scratch, sparks resulting from strike of opponent's weapon, or being lifted in the air with no damage and no lasting loss of traction.
- cosmetic: visible scratches to armor (as pictured to the right), non-penetrating cut or dent or slight bending of armor or exposed frame, removal of non-structural or non-functional cosmetic pieces (dolls, foliage, foam, or ablative armor), or damage to wheel, spinning blade, or other exposed moving part not resulting in loss of functionality or mobility.

- minor: being flipped over causing some loss of mobility or control or making it impossible to use a weapon (such as in an invertible drumbot with a drum that can only spin in one direction, because while inverted it would not be able to launch opponents), intermittent smoke not associated with noticeable power drop, penetrating dent or small hole (as pictured to the right), slightly warped frame not resulting in loss of mobility or weapon function, or removal of most or all of a wheel or weapon part without loss of functionality or mobility.



- significant: continuous smoke or smoke associated with partial loss of power of drive or weapons, damage or removal of wheels resulting in impaired mobility, damage to rotary weapon resulting in loss of weapon speed or severe vibration, damage to arm, hammer, or other moving part resulting in partial loss of weapon functionality, visibly bent or warped frame, and torn, ripped, or badly warped armor or large hole punched in armor (as pictured to the right).

- major: smoke with visible fire, armor section completely removed exposing interior components, warped frame causing partial loss of mobility or complete loss of functionality of the weapon system, internal components broken free from mounts and resting or dragging on the arena floor, significant leak of hydraulic fluid or pneumatic gases, or removal of wheels, spinning blade, saw, hammer, or lifting arm, or other major component resulting in total loss of weapon functionality or mobility.
- massive: armor shell completely torn off frame, major subassemblies torn free from frame (as pictured to the right), total loss of power, or loss of structural integrity such as major frame or armor sections dragging or resting on the floor.

If your robot is in good shape at the end of a close match, it is a good idea to demonstrate operability of the robot's drivetrain and/or weapon for a few seconds immediately after the end of the match, without touching the opponent. In this way, the judges will ascertain that your robot is still functional, and not sluggish or dead.

Scoring of damage points is based on relative grading of each robot's damage, as described below:

- a 6-0 (or 0-6) score is awarded when one robot suffers nothing more than trivial damage, and the other is at least significantly damaged, or one robot has suffered major or massive damage and the other is no more than cosmetically damaged.
- a 5-1 (or 1-5) score, if one robot suffers at least minor damage and the other suffers major or worse damage, or one robot has suffered cosmetic damage and the other has suffered at least significant damage.
- a 4-2 (or 2-4) score, if both robots have suffered nearly the same level of damage but one is slightly more damaged than the other.
- a 3-3 score, if both robots have suffered the same level of damage, or neither robot has even cosmetically damaged the other.

Damage that is self-inflicted by the robot's own systems, and not directly or indirectly caused by contact with the other robot or an active arena hazard, will not be counted against that robot for scoring purposes. In addition, any pre-existing damage in a robot before the match should not be counted against it.

9.2.6. After Your Fight

After your fight, immediately take your robot back to your pit and service it, even if it doesn't look damaged. Your first priority is to take care of the (probably hot) batteries.

Take care of your batteries

Immediately open the battery compartment. Carefully touch the battery packs to check their temperature. For NiCd packs, it’s normal if they’re warm or even fairly hot. But if they’re too hot even to be briefly touched, this means that their temperature is much higher than 140^{o}F (60^{o}C), which is a cause of concern: the battery life might be significantly shortened. If one pack is much colder than the other(s), this might be an indication that its circuit was open during the match, either due to a broken solder inside the pack or to a connector malfunction, which can be easily checked with a multimeter. After checking their temperatures, immediately place the battery packs over a large fan to be cooled down (see chapter 8). Only start recharging them after they get cold.

Access damage

After taking care of the batteries, inspect the entire robot for any structural damage. Look for any large debris that might have entered the robot, such as metallic parts that could shorten your electronics or pieces of rubber or foam tire treads that could get stuck in the clearances between your wheels and structure.

Turn the wheels by hand to feel if there’s any problem with the drivetrain. A stuck wheel could be either due to debris in the transmission (either foreign debris or from the own robot, such as a broken gear tooth inside a gearbox), or due to bent armor/structure interfering with the wheel. If the wheel gets stuck only in a few positions, this might be an indication of a broken gear tooth or a bent shaft. If the wheel gets stuck once at every turn, then the problem might be in its shaft or in the last stage of the reduction. If it gets stuck once every few turns, the problem might be in the previous stages. On the other hand, if the wheel easily turns by hand without any mechanical resistance from the motor, then you might be facing a broken key or shaft coupling, a stripped gear, a loose pinion, a ruptured belt, or a derailed chain from its sprocket, depending on your transmission system.

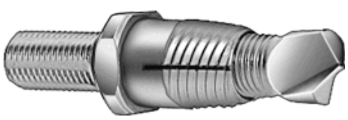
After checking the drive system, look for damage in the weapon, focusing on the most stressed parts, such as on the teeth of a drum, or the center section of a spinning bar. Check the condition of all belts and chains, from both the weapon and drive systems, and change them if necessary. If you previously marked the robot’s bolt heads with a Sharpie, as explained before, it will be easy to spot if any of them got loose and needs to be retightened.

Remove damaged screws

There are several ways to deal with a screw with a stripped or broken off head. On socket head cap screws with hexes that have been stripped out, you can take a slightly larger Allen wrench and grind it just enough for it to be hammered into the stripped recess, and then unscrew them.

For screws with broken off heads, try to unscrew the remaining stub with a vise-grip. If this doesn't work, then use a Dremel (as seen in the left picture) to cut two parallel chamfers in such a way that an open-end wrench could do the job (as seen in the right picture above).

If the stub is entirely embedded into the robot, then you can use a screw extractor, such as the ones pictured below. This tool drills a pilot hole into the screw stub, and then a left-handed thread takes care of unscrewing it. If the damaged screw is a high strength one, you'll need special screw extractors, which can drill even class 12.9 hardened steel bolts.



If you don't have screw extractors, another option is to use a Dremel to cut a channel in the middle of the embedded stub, and then use a large flathead screwdriver to unscrew it. This method is not as good as the previous ones, but it may work if the stub is not too bent.

If everything else fails, then another tip is to weld a nut onto the stub (as pictured to the right), and then use an open-end wrench that matches the nut. If the stub is entirely embedded into the robot, it might be easier to first weld a washer to the stub, and then weld the nut onto the washer.

If the broken stub is deeply embedded into the robot, then another option is to weld a long and thin steel strip (as pictured to the right) and use a vise grip to unscrew it.

If the workpiece is made out of aluminum, then there's one last resort, which is to dissolve away the steel stub with nitric acid (HNO_3). Be very careful, do not immerse the workpiece in the acid, just put a couple of drops in the hole and, when no more gas comes out, wash out and repeat. Nitric acid dissolves the edges of the screw stub (or the stub of a broken tap), reducing its diameter and easing removal. It reacts much faster with hardened steel than with aluminum, so the threads in the workpiece won’t be compromised.

But, if the workpiece has a high value, it might be worth looking for a bolt disintegrator device. In this technique, the workpiece is immersed in water or oil, while the bolt (or tap) is electrically eroded. You will probably have to look for a machine shop that offers this service.

Socialize

If your opponent from the previous match is not busy working on his/her robot, go to his/her pit after the match to check the damages, to talk about the match, to take pictures, and to invite him/her to see the damages caused in your robot.

It is very common to give your opponent unusable parts from your robot that were destroyed by him/her during the match. These are called "trophies," they are memories to keep from the combats. It is an honor to receive them, and giving them away is another way to be polite and to establish friendships with other builders.

The picture to the right shows a few of our most cherished trophies, which we had the honor to receive (or scavenge in the arena, in a few cases) along the years, since RioBotz was born, in 2003.

Our “trophy box” has now more than 50 pounds worth of good memories.

9.2.7. Between Fights

Between fights, do not perform any dirty jobs on your pit table, such as grinding large parts sending sparks everywhere. Most events have a designated area for this. Otherwise, find an isolated area to use such tools. If it is a very small job, then ask for other team members to form a protection barrier to avoid sending sparks to neighboring tables. Several builders are already stressed trying to get their bots ready for combat, so it is wise to avoid conflicts.

Never test your robot's weapon in the pits. Keep the safety restraints on the weapon at all times. It is usually OK to test the robot drivetrain if its wheels are lifted off the ground, but always check with the event staff.

A very useful accessory is a 2-way radio (pictured to the right), which can be used by 2 or more team members, especially to communicate with the driver. This gives more freedom for your teammates, allowing them to wander around the pits between fights, until their presence is required to fix a robot or to drive it. Use a headset (earphone and microphone) for a hands-free experience. It is important for the radios to have a vibrating alert, because loud noises and music from the pits and arena might make it difficult to listen to sound alerts from the incoming calls.

Even if you have been eliminated, try to attend the event until its end. In this way you won't miss the show, you'll watch the fights and championship matches from a privileged position from the pits, and you'll be giving prestige to your peer builders that are still competing. At the end of the event, you will have learned more than you could imagine. And you'll have made many friends and met great builders. After all, it’s not every day that I get to meet legends such as Matt and Wendy Maxham, as pictured to the right.

Attending a combot event is a wonderful experience. And competing is even better, it is not easy to describe with words. You have to experience it yourself. Get ready for a major adrenaline rush.

9.3. After the Event

After each event, get together all your personal notes, and organize them while the information and memories are still fresh in your mind. They will be very useful to improve your robot and its future versions.

9.3.1. Battery Care

It is a good idea to store your batteries adequately, especially if you won't go to an event within the next months. If you'll be practicing driving your robot regularly, which is a good idea to improve your skills, choose perhaps 1 or at most 2 sets of batteries to be used on a daily basis, and store all others to save them for the next event.

SLA batteries should be stored at full charge, keeping their terminals very well isolated to avoid shorts. Recharge them at least every 6 months, even if you don't use them, due to self-discharge. You don't need to keep them in a refrigerator, as long as they're stored below 80^{o}F (27^{o}F).

Nickel batteries such as NiCd and NiMH, on the other hand, should be stored fully discharged, as discussed in chapter 8. But never below 0.9V per cell. It is a good idea to discharge them using an electronic charger such as Triton, see chapter 8. Then, place the batteries in a refrigerator at 5^{o}C (41^{o}F), not a freezer. This is so important that we have a dedicated refrigerator just for that, as pictured to right. But always store the batteries inside a sealed plastic bag such as ziploc, to protect them from humidity. In this way, the batteries can last up to 20 years, but you'll need to completely charge and discharge them at least every 6 months for that. When you remove the batteries from the refrigerator, wait for them to get to room temperature before charging. Never freeze the batteries.

Lithium batteries should also be stored in a refrigerator, inside a sealed plastic bag. But, instead of fully discharged, which could make them permanently unusable, they should be stored at about 40 to 60% of their charge level. Storing the battery at 100% charge level applies unnecessary stress and can cause internal corrosion. Recharge them back to 40 to 60% at least once per year, due to battery self-discharge.

9.3.2. Inspect Your Robot

After taking care of the batteries, disassemble your robot to access damage, several problems are not easy to spot in a fully assembled robot. Switch the screws that are in bad shape, either bent or with stripped heads. If you're having trouble removing a damaged screw, follow the screw removing techniques explained before.

Verify the condition of the belts or chains, this is a good time to change them if needed. Clean up very well your robot, acetone is very good to clean metallic parts (but don't use it on Lexan).

Then, visually inspect critical components, such as shafts, looking for cracks. Several times, visual inspection is not enough to spot a crack, because cracks usually have their mouth closed when the part is not loaded, only leaving a very subtle trace on its surface. One efficient way to detect a crack is to use a low-cost technique called dye penetrant inspection (DPI). DPI is based upon capillary action, where a low surface tension fluid penetrates into surface-breaking cracks.

To perform DPI, you only need two spray cans, one with the penetrant dye and the other with a white developer, found for instance at McMaster-Carr under Dye-Penetrant Detection Kit. The inspection steps are described next, applied to the tempered 4340 steel weapon shaft from our middleweight horizontal bar spinner Ciclone.

(a) (b) (c) (d) (e)

(a) pre-cleaning: the test surface is cleaned to remove any dirt, paint, oil or grease, using for instance acetone; do not leave fingerprints;

(b) application of penetrant: the penetrant is applied to the surface, in general as a spray; the penetrant dye can be colored, usually red as pictured above, or fluorescent, to be later inspected under ultraviolet light; always use gloves, because the dye will penetrate all the way under your fingernails if you're not careful (unless you want to save money on nail polish);

(c) waiting period: wait from 5 to 30 minutes for the penetrant to soak into any cracks or flaws; very small flaws may require a longer waiting time;

(d) application of developer: completely remove the penetrant from the surface using a dry lint-free cloth (do not use acetone in this step, it could remove as well the penetrant absorbed by the cracks), and then apply the appropriate white developer to the entire surface, until the surface looks like it's frozen (as pictured above), forming a semi-transparent, even coating;

(e) inspection: wait for 10 minutes for the blotting action to occur, where the developer will bring any trapped penetrant to the surface, exposing cracks or flaws through the form of thin red (or fluorescent) lines under white (or ultraviolet) light; do not wait too long to inspect, because the thin lines may "bleed out" and make it difficult to evaluate the size of the crack, if any; inspect very carefully near geometry changes such as notches, where it is more likely to find a crack; beware with false positives, because very small harmless scratches (generated either during manufacturing or combat) can result in very thin lines - it is up to the inspector to distinguish between cracks and scratches, depending on the thickness of the developed lines; fortunately, Ciclone's shaft was free of cracks in the above inspection.

Finally, completely assemble your robot as soon as you finish servicing and inspecting it. With your robot fully assembled, it will be impossible to misplace any of its components. Misplaced components will most likely be lost forever if you only look for them several months later, near the date of the next event.

9.3.3. Wrap Up

Update your homepage as soon as possible. During or immediately after an event is the best time to do that, in special if you want to increase the number of hits in your webpage to please your sponsors. Most builders and enthusiasts that didn't attend the event will certainly be searching for photos and videos during it, and mostly everyone will be looking for them right after the event ends. Make sure you post announcements of your updates, for instance, on the RFL Forum in the appropriate topic related to the event.

If you don't have a webpage, make one. It is important that your team has visibility to be able to get sponsors. Nowadays it is very easy to design and upload a webpage. A basic one will take you less than an hour to prepare.

Check the current ranking of your robots at both www.botrank.com and www.buildersdb.com websites, and keep in mind the dates of the next events.

Now relax, and review the pictures and videos from the event. Win or lose, celebrate with your teammates and other builders. Cheers!

Chapter

10

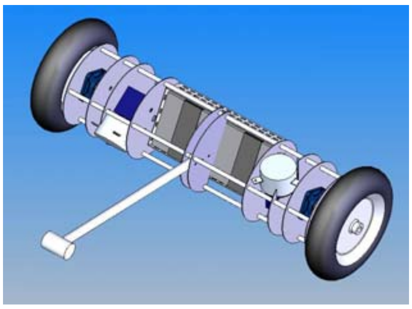
RioBotz Build Reports

This chapter presents the build reports of all combat robots from RioBotz, including the entire Touro family. It also talks about the origins of our team, since our first combot Lacrainha.

10.1. Lacrainha

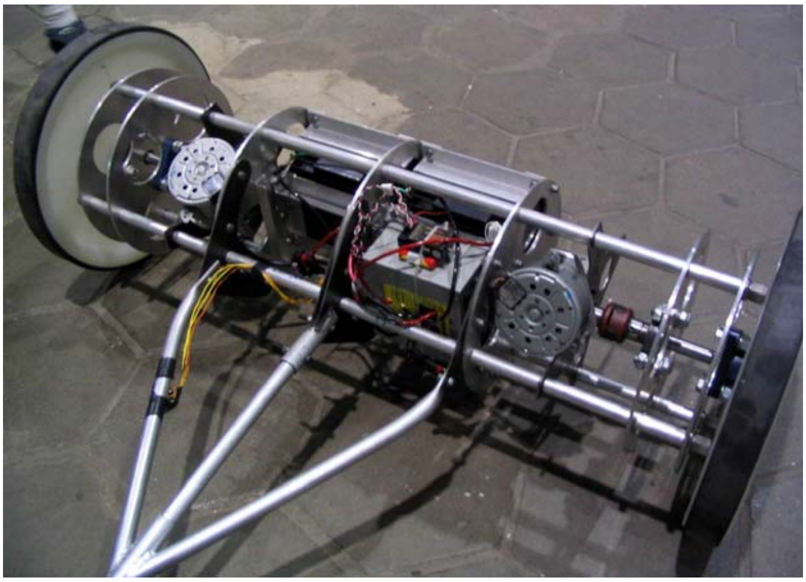
RioBotz was created in January 2003, when six undergraduate students from my University, PUC-Rio, asked me to be their advisor to help them build a combat robot. It didn't take long for me to get hooked. We then started to build our first combot, the overhead thwack robot Lacrainha ("Little Centipede" in Portuguese, pictured to the right), a **middleweight** (120lb) with structure completely made out of scrap metal. Note in the picture the SLA battery holder, made out of scrap perforated cable trays used in our University.

Its cylindrical shape was due to our very low budget: we were able to get several scrap aluminum disks used in the structure of the pigs built in our University. The pigs (pictured to the right) are used in the internal inspection and cleaning of oil pipelines, with the same cylindrical shape of our robot Lacrainha.



Bosch donated us a pair of GPB motors, and we bought two surplus worm gearboxes, made out of cast iron. Two heavy 12V 17A·h AGM SLA batteries, used in electric bikes, powered the robot and also acted as counterweights to help it strike with its hammer. We developed the entire electronics, both controller and power boards, using relays to provide a simple bang-bang control (no speed control at all). The radio was borrowed from the Aerodesign team from our University, and our first robot was born. Lacrainha never saw combat, because it was soon replaced by its bigger brother Lacraia.

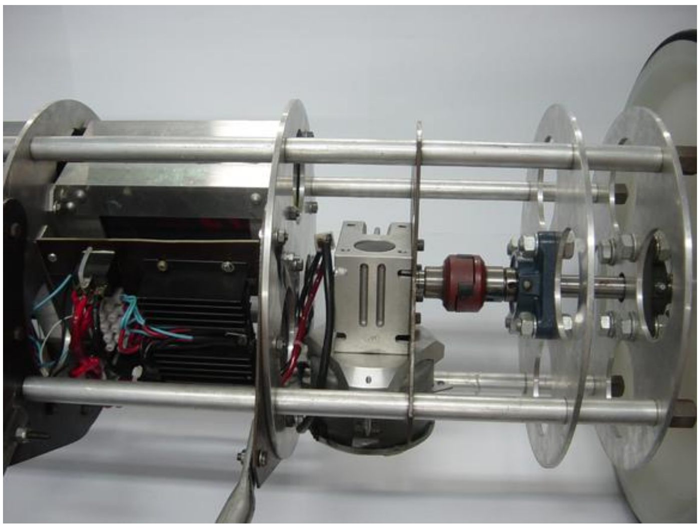
10.2. Lacraia



Still in 2003 we were able to get the support from our first sponsor, EPTCA Medical Devices. We then built an improved version of Lacrainha, the **middleweight** Lacraia (meaning “Centipede” in Portuguese, pictured to the right).

In spite of its better appearance, the robot was still very primitive: its 6063-T5 aluminum armor had a thickness of only 1mm, its electronics used bang-bang control with relays and a single MOSFET per motor, and SLA batteries powered the GPBs with a heavy and inefficient worm gearbox. Nevertheless, it was a competitive robot in the Brazilian competitions at that time. It was one of the only invertible robots. It achieved the 6th place during the III ENECA Brazilian national championship. The steel ball used in the hammer was later replaced by a sharp S1 tool steel piece.

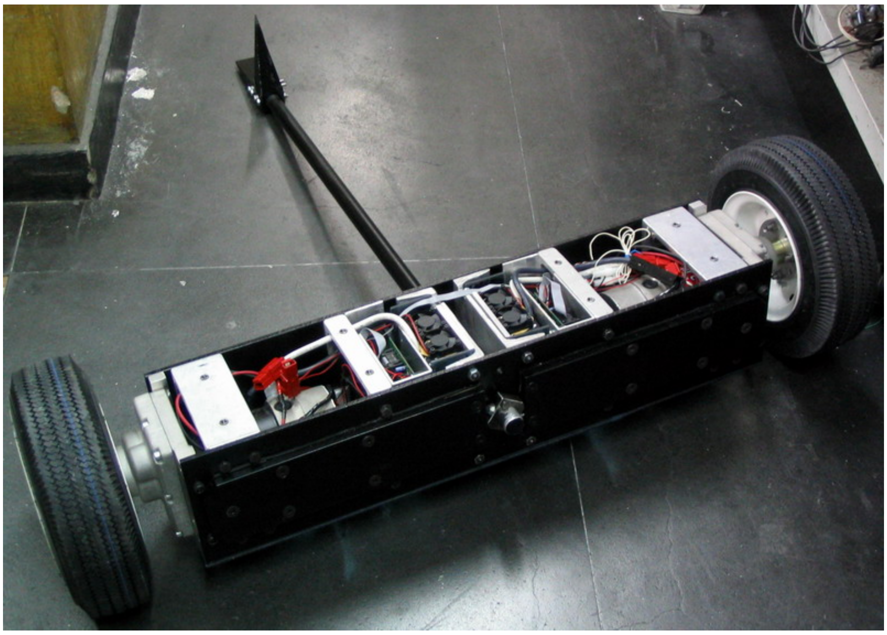
Lacraia is now bolted to the ceiling of our lab (as pictured below), bearing a medieval axe.



10.3. Anubis

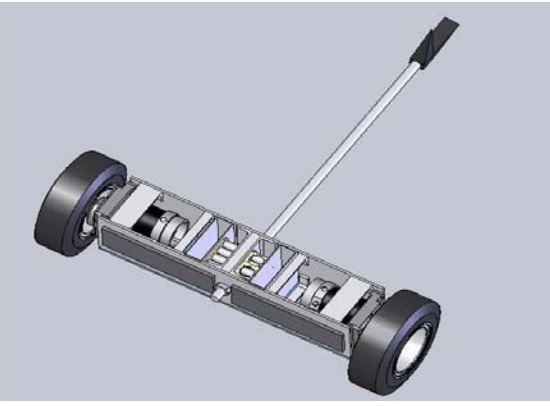
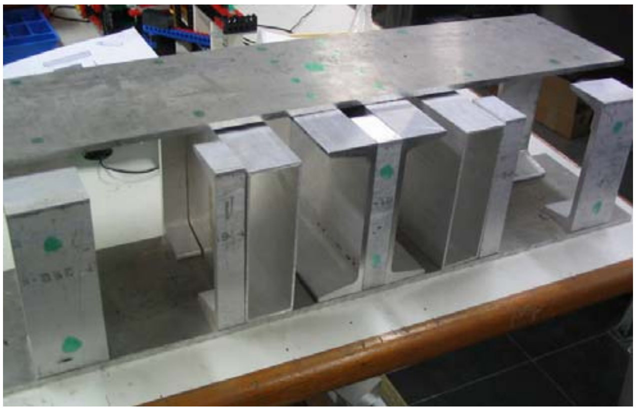
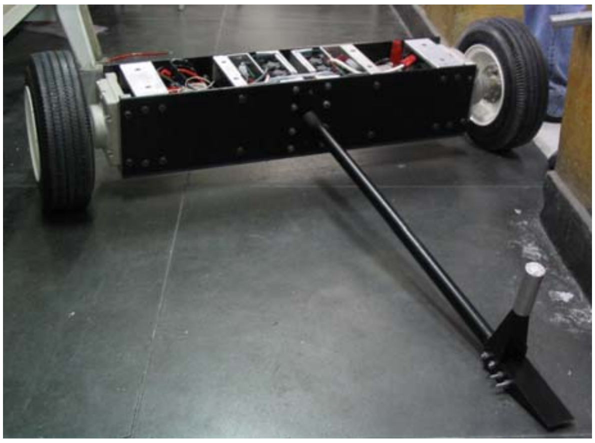
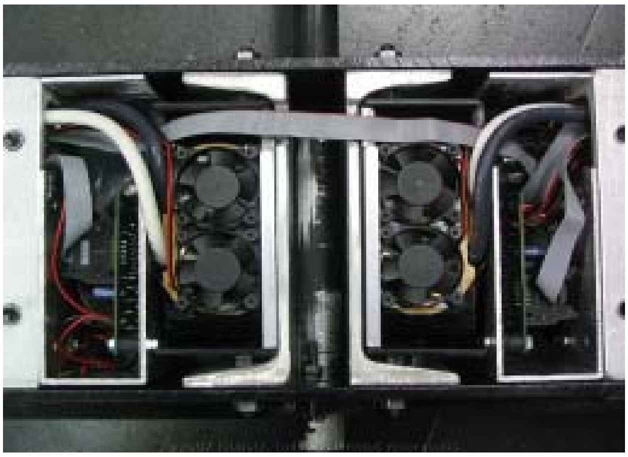
In 2004 we designed our third **middleweight** (120lb), Anubis, another overhead thwackbot. With almost 10 times the power of Lacraia, Anubis is one of the fastest robots that we ever built. Its structure is made out of two 7050 aeronautical aluminum plates, 6061-T6 aluminum extrusions, and Lexan covers.

The tip of its weapon is made out of tempered S1 tool steel, designed to pierce the opponents. The shape of this tip reminded a lot the head of the Egyptian god Anubis (see its logo pictured right) - this is how it got its name. The robot also works as a rammer, since it has two tempered S1 steel plates that act both as counterweight and armor. The two NPC wheels, filled with polyurethane foam, are powered by two NPC T74 motors.



We specially developed a controller board for the two OSMC speed controllers that power the NPC T74 motors. Anubis was the first Brazilian combat robot that used NiCd batteries, it was powered by two 24V 3.6Ah NiCd Battlepacks, providing the necessary high currents and torques to accelerate the weapon and perforate armors.

Anubis only fought once, it won a rumble match against 3 other robots (hey, that's a 100% win record!). However, the gearbox of one of the NPC T74 motors broke towards the end of the match. Today, we're converting Anubis into a Segway-type personal transporter.

 CAD drawing	 7050 aluminum plates and 6061-T6 extrusions
 Anubis (front) is much smaller than Lacraia	 after the black electrostatic powder coating
 compact electronics and NiCd batteries	 practicing against a monitor

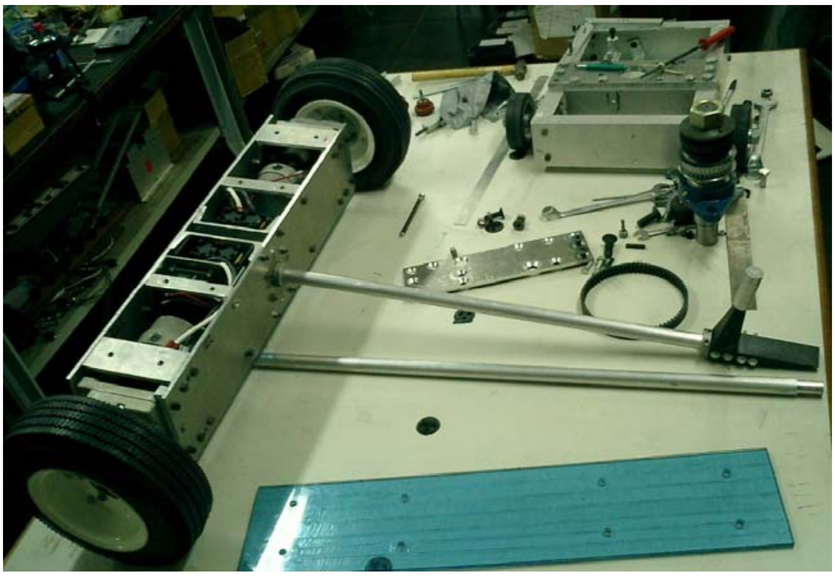
piercing an opponent in a rumble match

piercing once again

Still in 2004, we decided to create a robot to replace Anubis in case it broke in combat. We didn't want to build another thwackbot, we wanted to explore different possibilities. So we decided to follow the dark path known as Spin To Win.

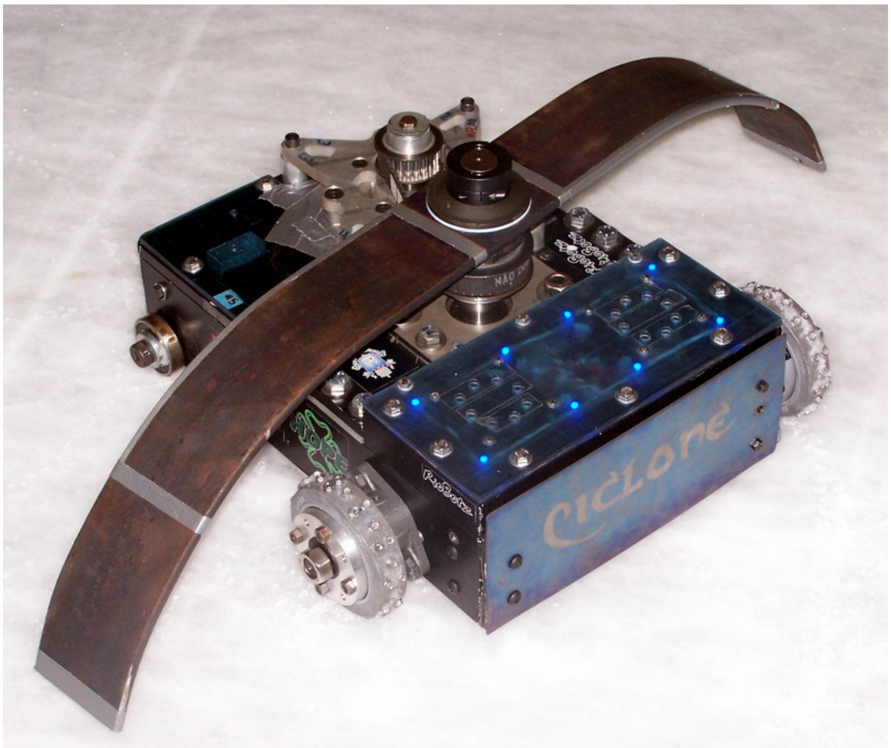
Many people think of spinners (horizontal, vertical or drums) as the robots from the "dark side of the Force." This is because they are the quick and easy path to victory and destruction (and therefore follow the Sith philosophy).

But this is not entirely true: yes, they can generate a lot of destruction, but they are probably one of the most difficult robot types to properly design. It is not at all an easy path. It is easy to spin a heavy bar or disk using a high power motor, but it is very hard to design a robust structure and weapon system that can cause huge damages to the opponent without harming itself. The spinner is always challenging Newton's third law. Besides, vertical spinners and drumbots have directional weapons, and therefore they need as much a skillful driver as any wedge, hammer, lifter, launcher. Even a horizontal spinner involves some strategy, as it needs to maneuver around the opponents to hit their weak points, or to run away from the adversaries while its weapon is still spinning up.



The result from this dark path was our first spinner, Ciclone. It is the little guy standing beside Anubis in the picture. It was meant to be just a spare robot in case Anubis broke, but it ended up so destructive that it was promoted to our main combot.

10.4. Ciclone



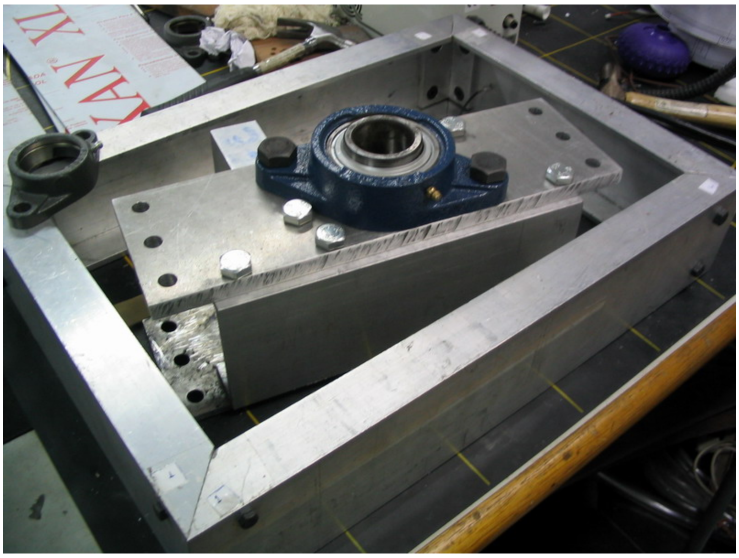
Our 120lb **middleweight** Ciclone ("Cyclone" in Portuguese, pictured to the right) is, essentially, an Etek motor surrounded by a robot. This motor, as already discussed in chapter 5, is extremely powerful. It is an excellent choice for heavyweights and super-heavyweights, but it is too heavy for a middleweight. We had a lot of trouble to fit an Etek into a middleweight without sacrificing the resistance of the structure. The complete weapon system, including the 5160 tempered steel bar, torque limiter, shaft collars, Etek motor with its mount, weapon shaft with mounted bearings, timing pulleys, and belt, added up to almost 50% of the robot's weight. It is not respecting at all the 30-30-25-15 rule, which would suggest that only 30% of the robot weight should be used in the weapon (see chapter 2).

To compensate for that, we had to sacrifice a little the remaining 30-25-15 from the rule. We used only two 18V RS-775 motors with DeWalt gearboxes set at high torque to drive the two wheels, which resulted in only 15% of the robot weight in the drive system, instead of the 30% from the rule. This caused Ciclone to be a little slow for US standards, however it was not too bad for the Brazilian competitions back in 2004. The two 18V gearmotors, powered by a specially developed control board and two OSMC speed controllers, were attached to the robot using shaft collars in the region pictured to the right.

Due to its weight budget, Ciclone only used two NiCd packs: one 24V Battlepack to power the Etek (weapon system) and one 18V pack from our DeWalt cordless drill for the drive system. The batteries accounted for only 7% of the robot's weight, instead of 15% from the 30-30-25-15 rule. Thus, the robot only had left about 100% – 50% – 15% – 7% = 28% of its total weight for the structure and armor, a reasonable value that is compatible with the rule.

The motors and gearbox for the drive system were obtained from disassembling two 18V DeWalt cordless drills. We used not only the motors (number 1 in the picture to the right) and planetary gearboxes (number 2), but also the batteries (number 5) and chargers (number 6). It is an excellent cost-benefit to disassemble cordless drills.

The structure of Ciclone was all made out of 4” high 1/4” thick aluminum extrusions. Unfortunately, we were only able to find 6063-T5 extrusions with those dimensions (the 6063-T5 is a very low strength aluminum alloy).



To compensate for that, the plates used to hold the mounted bearings from the weapon system (in the center of the picture to the right) were all made out of 7050 aeronautical aluminum, which has a much higher strength than the 6063-T5. The covers for the electronics and batteries were made out of Lexan.

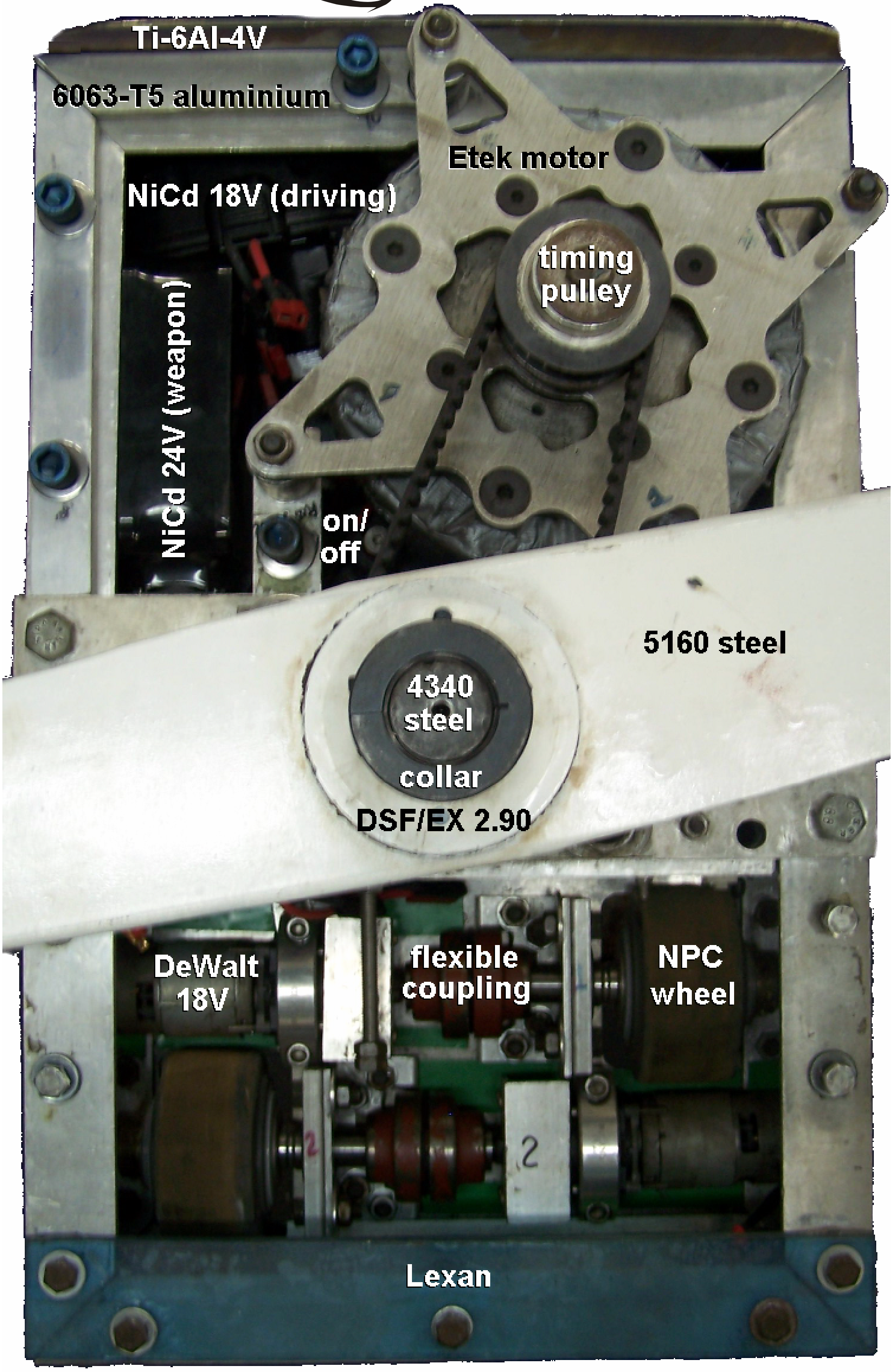
The spinning weapon is a 5160 steel flat bar used as a leaf spring in the suspension system of trucks. The bar was bent using a servo-hydraulic machine from PUC-Rio’s Fatigue laboratory, and later tempered. We used the torque limiter DSF/EX 2.90 (see chapter 5) to connect the bar to the 1.5” 4340 steel weapon shaft. This torque limiter acts as a clutch, to allow slippage during the impacts. The shaft is powered by a pair of timing pulleys, using an 8M size timing belt.

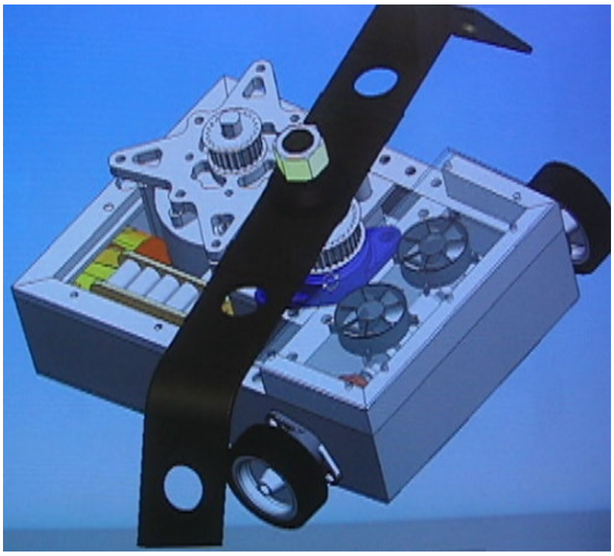
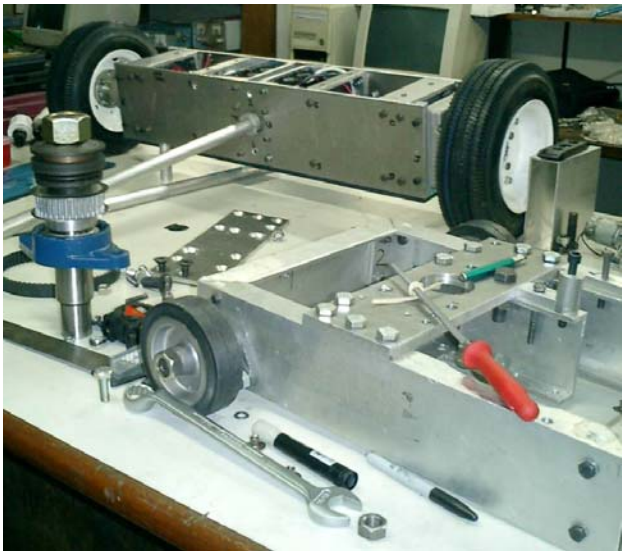
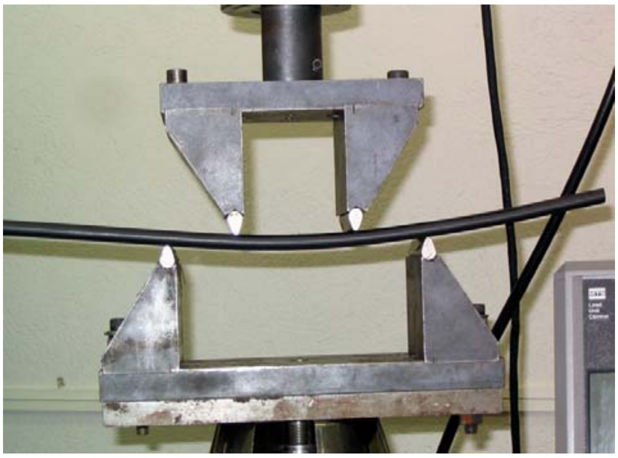
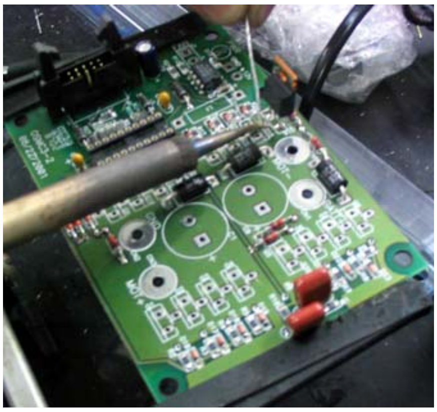
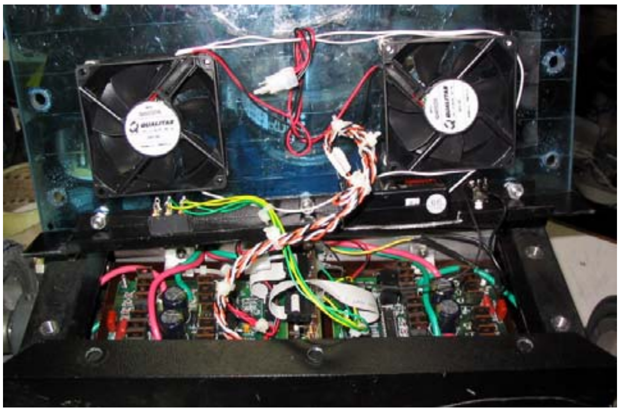
Subsequent improvements included a front armor made out of titanium grade 5, internal wheels, and a Hella key to turn on/off the robot. The cast iron mountings of the weapon shaft bearings survived combat, but not without a few cracks. So we later press fitted the bearings directly into the aluminum plates of the robot structure, and never had another problem. Cast iron is very brittle and heavy, it is not a good option in combat robot designs (except for the fact that cast iron bearing mounts are very easy to attach to a robot).

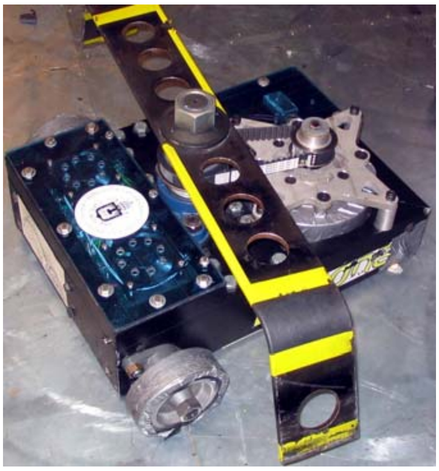
Ciclone became the Brazilian champion in 2004 and 2005. The 2005 event was particularly interesting, it was held on an ice arena, we had to develop special wheels to guarantee traction (see chapter 2). In its third competition, Ciclone was flipped over by a wedge. This was when we realized that we needed an invertible robot. Suddenly, drumbots started to sound like a good idea...

The pictures that follow show a detailed anatomy of Ciclone, as well as photos taken during its building, tests and combats.

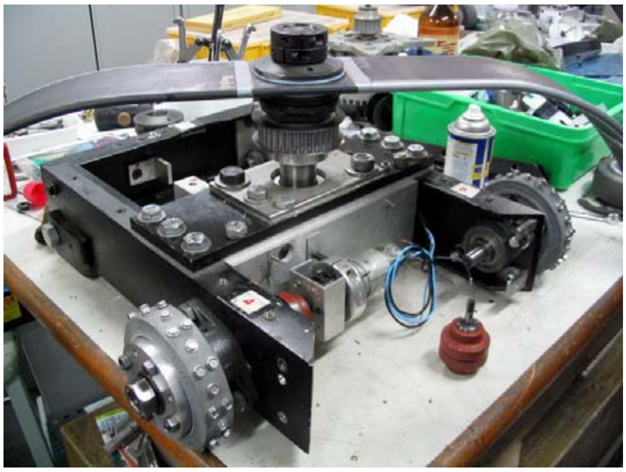
Ciclone


Ti-6Al-4V
6063-T5 aluminium
Etek motor
NiCd 18V (driving)
timing pulley
NiCd 24V (weapon)
on/off
5160 steel
4340 steel
collar
DSF/EX 2.90
DeWalt 18V
flexible coupling
NPC wheel
Lexan

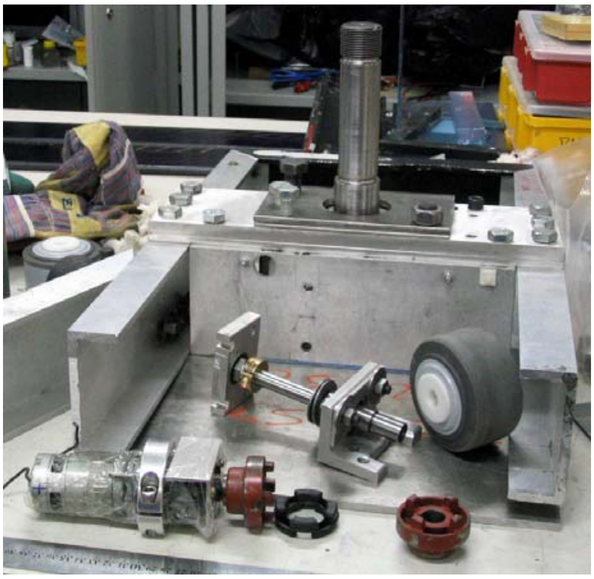
 CAD drawing of the 2004 version	 Ciclone and Anubis being built
 nice black electrostatic powder coating	 bending the weapon bar
 assembling the OSMC speed controller	 control board, OSMCs, and fans



Ciclone, 2004 version

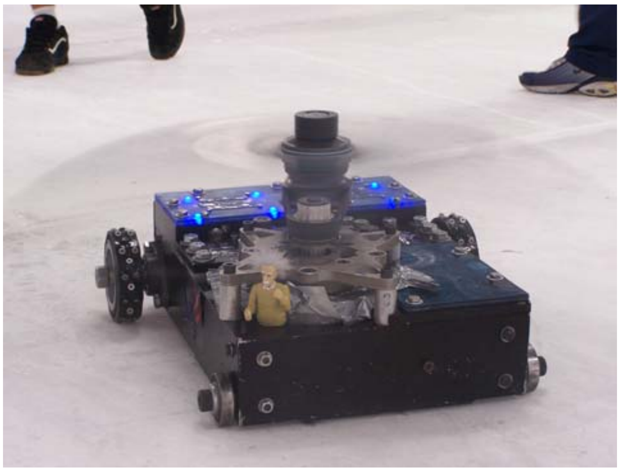


modifications in 2005 to drive on ice



moving the wheels inside the robot

Ciclone, late 2005 version



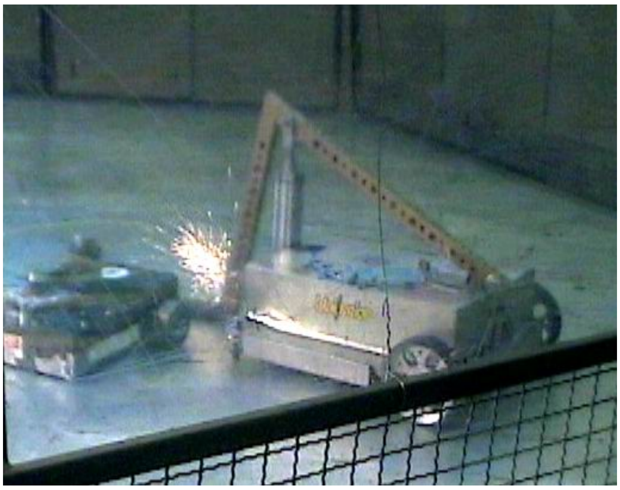
you don’t have to be Einstein to drive on ice

Ciclone’s very first opponent

challenging a concrete block

smashing a monitor

spinner vs. spinner in the 2004 finals



2004 semifinal against a pneumatic flipper

flipping Vingador in the 2005 finals, on ice

getting a piece of Donatello

10.5. Titan

In 2005 we designed the **middleweight** Titan (pictured to the right), a horizontal spinner that incorporated a series of improvements over Ciclone.

Our first concern in the design was the 30-30-25-15 rule, which had been severely violated by Ciclone. This time, we designed the entire weapon system keeping in mind to use in it less than 25% of the robot's weight. We used two Magmotors S28-150 to drive a single 90^o conical gear that powered the weapon shaft up to 3000 RPM. The spinning bar, a tempered 5160 steel leaf spring, was attached to the weapon shaft using a large Belleville washer and a threaded shaft collar. In the 2006 version, we added a Ti-6Al-4V titanium wedge to make it effective against wedges or very low robots. The total weight of the weapon system, if we consider the wedge as part of it, reached about 30% of the robot's total weight, as recommended by the rule.

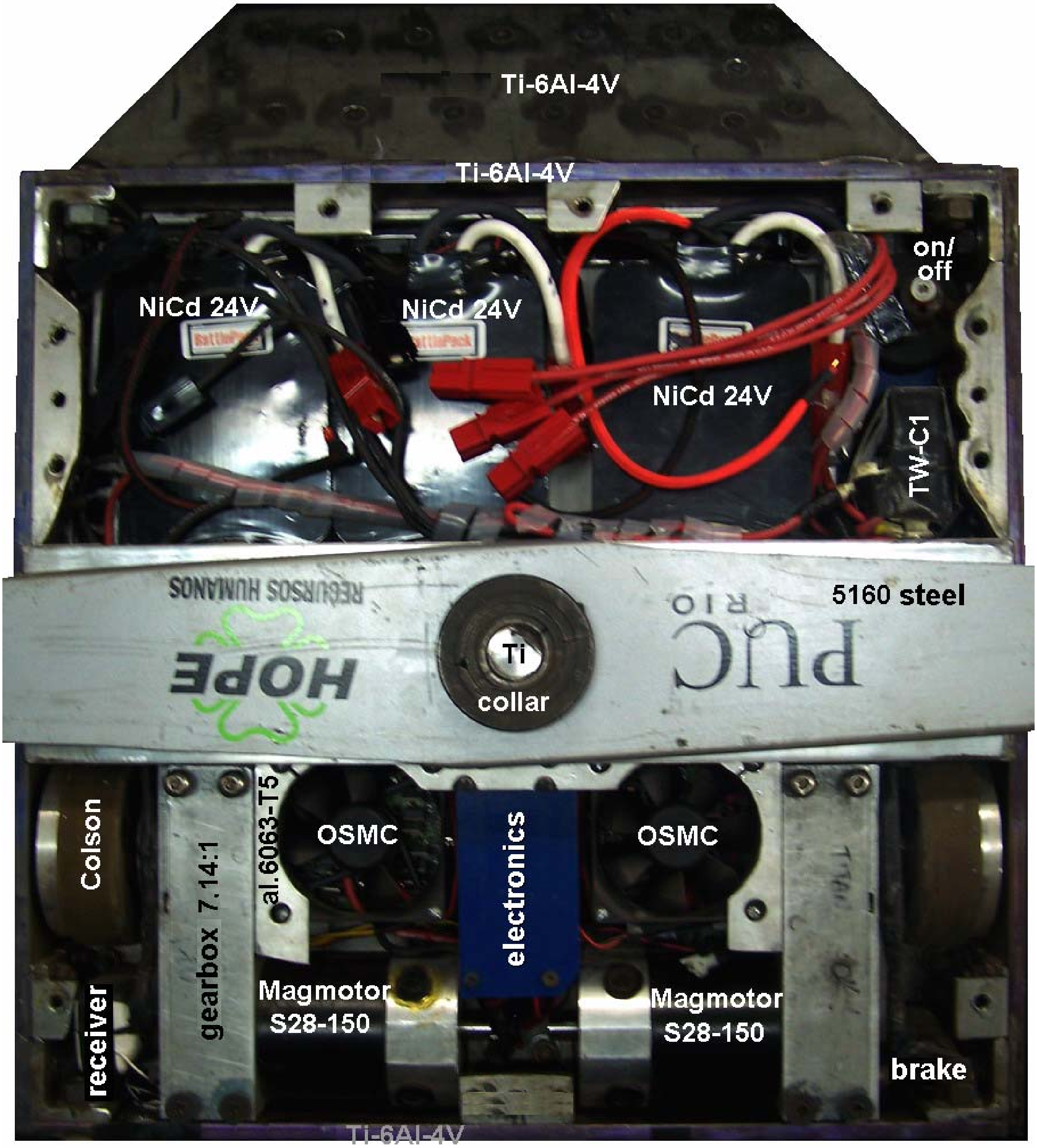
The robot's drive system used two Magmotors S28-150 with TWM3M gearboxes and 4" diameter Colson wheels. The drive system used only 15% of the robot's weight, half of the recommended value, but even so its speed was already much higher than Ciclone's. Traction was also better, because in Titan the two wheels were closer to the robot's center of mass.

We used two OSMCs to control the drive system, which were later replaced by Victors to gain some space. The weapon motors were powered by a TW-C1 solenoid, with a Hella key as on/off switch. The three 24V NiCd battery packs that Titan uses in parallel, together with the entire electronics, resulted in less than 15% of the robot's weight, well in agreement with the 30-30-25-15 rule.

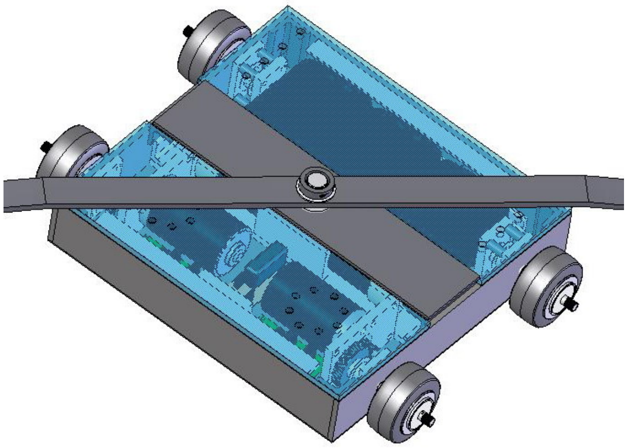
Titan's integrated structure/armor was entirely made out of 8mm (5/16") thick Ti-6Al-4V titanium, with a 2mm thick titanium bottom cover. The top covers were made out of 8mm thick 7050 aeronautical aluminum. The top plate where the weapon shaft is attached to was made out of 304 stainless steel. Angle extrusions, also made out of 304 stainless steel, were used inside the robot to join the titanium walls. These heavy steel reinforcements were only possible thanks to the 15% weight savings from the drive system. Therefore, the structure/armor ended up using respectable 40% of the robot's weight, well above the 25% from the rule. This sturdy structure was important to help the robot survive its own reaction forces from the inflicted blows.

The pictures below show a detailed anatomy of Titan, as well as building and testing photos.

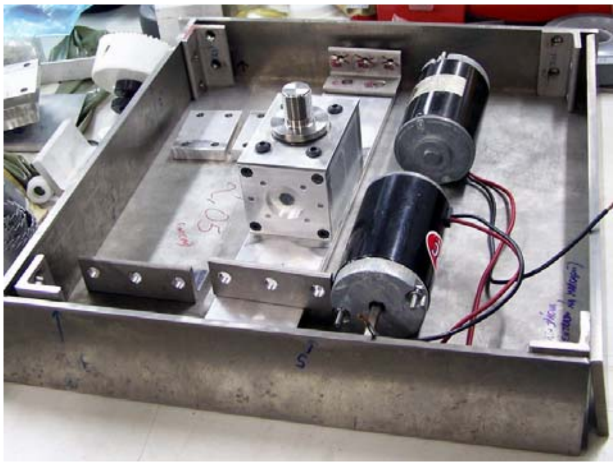
TiTAN2



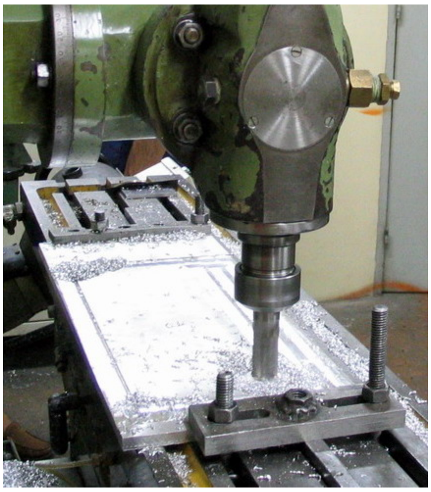
[redacted] Ti-6Al-4V
Ti-6Al-4V
NiCd 24V
NiCd 24V
NiCd 24V
on/
off
TW-C1
5160 steel
Ti
collar
Colson
gearbox 7.14:1
al.6063-T5
OSMC
electronics
OSMC
Magmotor
S28-150
Magmotor
S28-150
receiver
brake
Ti-6Al-4V



original CAD design, with 4 exposed wheels: it was soon changed to 2 internal wheels

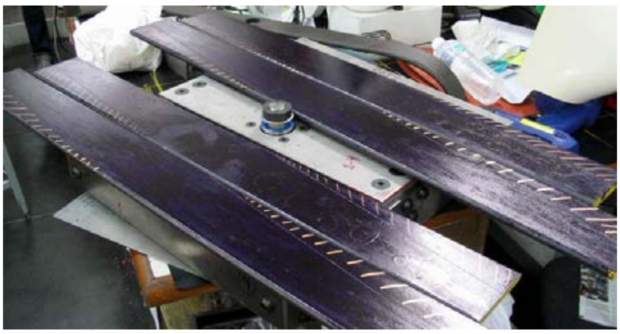


titanium walls and bottom, and steel angle extrusions: the original NPC-02446 drive motors were soon changed to short Magmotors



milling the top covers, to reduce weight

Titan with the milled aluminum top covers

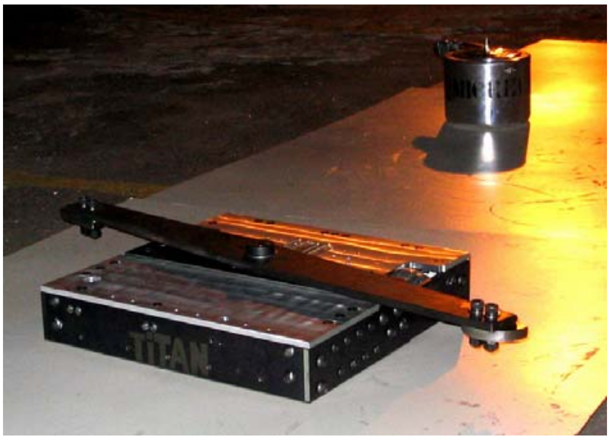


5160 steel bars for the weapon, before drilling and tempering

the center of gravity of each bar was found balancing them on the tip of a center punch

the weapon bars originally included a pair of S7 steel claws, later removed

Michelangelo, Leonardo and Rafael: 30kg (66lb) sparring ninja turtle-bots



weapon test: Titan vs. Michelangelo, at dawn, in the parking lot of PUC-Rio

poor Michelangelo after a whole night of beating: “shell shock”

Titan vs. the combat arena

10.6. Touro

The design of our **middleweight** Touro ("Bull" in Portuguese, pictured to the right) focused on the idea of a low profile reversible robot with a kinetic energy weapon. Being reversible was a very important factor, because Touro would debut at RoboGames 2006, against several US robots that could easily flip their opponents. We went for the drumbot design, inspired by Falcon's compact size and motor choices, Tekka Maki's sloped front plates beside the drum, and Angry Asp's anti-wedge skids.

We started machining the drum, made out of a 1" thick ST-52 steel cylinder (similar to a 1025 steel, with 0.25% carbon). In 2007 the ST-52 was switched to 304 stainless, which has a much higher impact toughness. We've bolted to the drum two tempered S7 steel bars to work as teeth to catch the opponents.

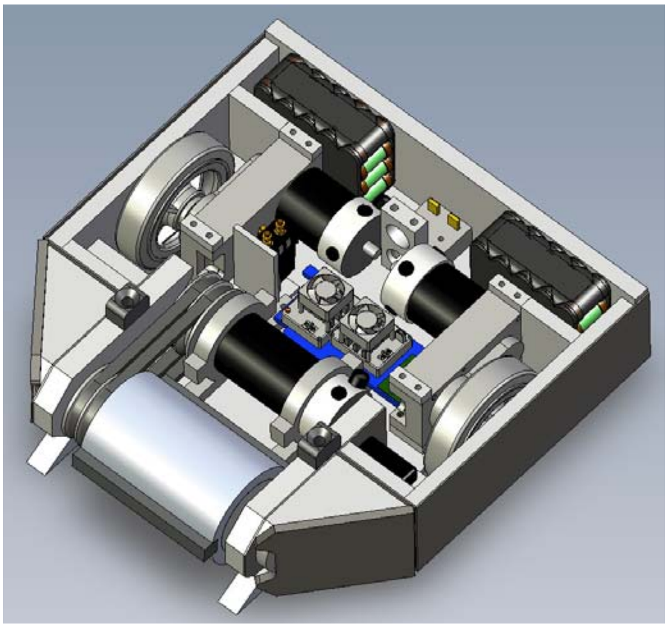
We machined a double pulley to be fitted to the drum, allowing it to be powered by a pair of A-size V-belts. As discussed in chapter 5, V-belts work as a clutch, sliding during the impact. The drum was mounted to a 1.5" diameter solid shaft made out of tempered 4340 steel, which was later replaced with titanium grade 5 to save weight. The drum was powered by a Magmotor S28-400, the longer and more powerful version of the S28-150, at 24V. In 2006, the motor and drum pulleys had the same diameter, resulting in a drum top speed of about 4,900RPM. In 2007, the diameter of the drum pulley was reduced, increasing the weapon speed to 6,000RPM.

The entire weapon system resulted in almost 35% of the robot's weight. That value was a little over the 30% from the 30-30-25-15 rule, but this is not too bad for drumbots, because due to its small radius the drum needs to be heavy to generate a significant moment of inertia. Besides, the drum can also be considered as part of the armor, since sturdy drumbots can also do a great job as rammers. Several drumbots from various weight classes use about 20% of their weight in the drum, and up to 15% to power them (including shaft, pulleys, belts, bearings, motor and its mounts), adding up to 35% as in the case of Touro.

Touro's drive system is similar to Titan's, it used 2 Magmotors S28-150 with TWM3M gearboxes. We used 6" diameter Colson wheels, instead of Titan's 4" wheels, increasing Touro's top speed in 50%. The drive system ended up using about 15% of the robot's weight, well below the 30% value from the rule. Note that most robots spend something closer to 30% of their weight, rather than 15%, in their drive system. We were only able to reach 15% because we only used 2 wheels, powered by motors and gearboxes with very high power to weight ratios. Any rammer, wedge, thwackbot or overhead thwackbot, which depend a lot more on a robust and powerful drive system, as well as any robot with 4 (or more) active wheels, will need to get closer to the 30% value to be efficient. Therefore, the 15% value would probably be a lower limit for the drive system weight, which could be enough only for robots with very powerful weapons.

A MS-2 switch (more compact than the Hella key) was used to turn the robot on or off. The weapon motor was controlled by a TW-C1 solenoid. To keep Touro compact, we used Victors instead of OSMCs for the drive system. We developed a small electronic control board specifically to power the TW-C1 and to act as BEC (Battery Elimination Circuit, see chapter 7) for the receiver.

Note in the following pages that we used a braided mesh (in light red, in the center) to organize and to protect the wiring, avoiding shorts due to friction with metal parts from the structure. The entire robot was powered by two 24V NiCd Battlepacks connected in parallel. The entire battery and electronic system added up to about 10% of Touro's weight.



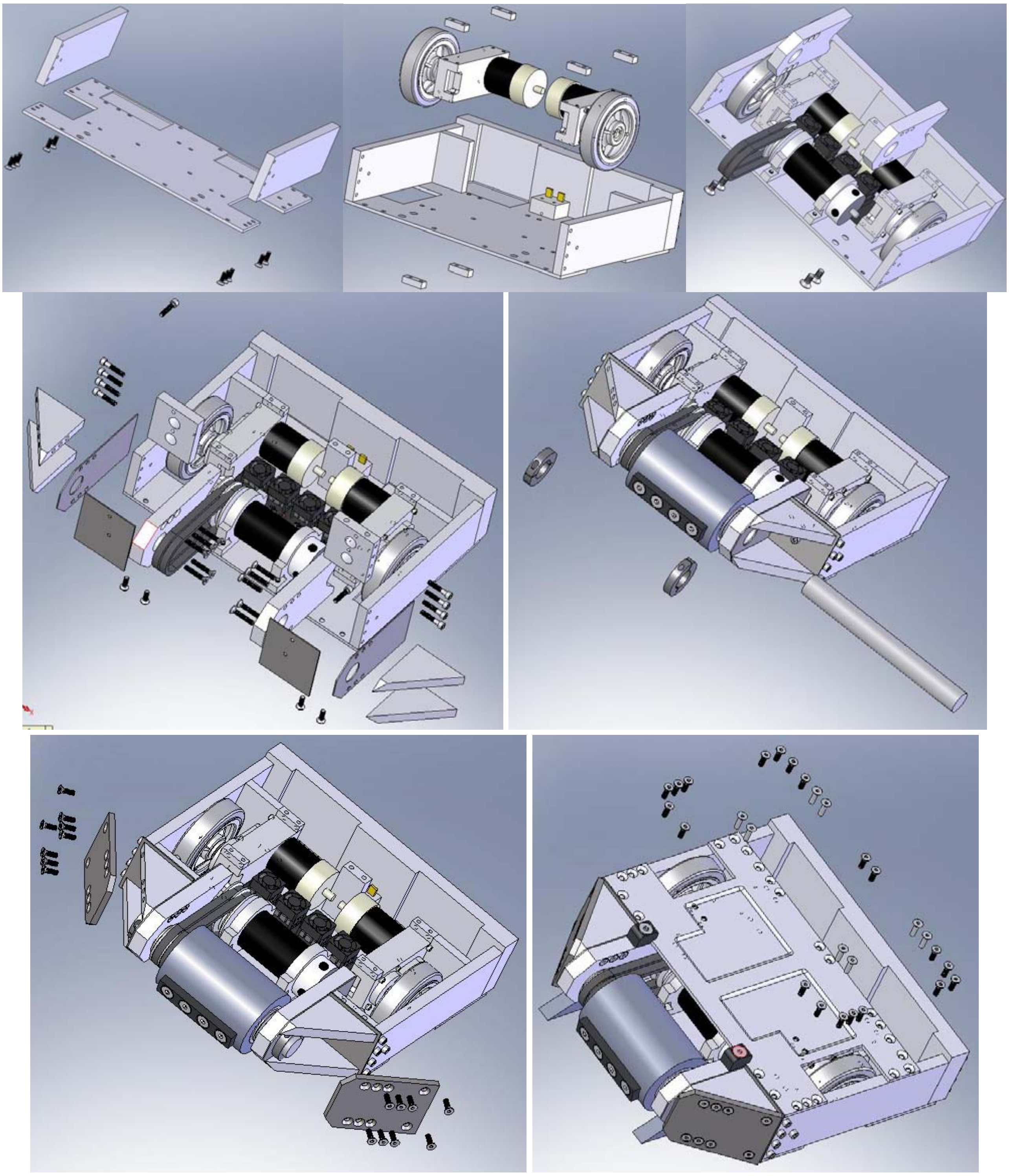
Touro's integrated structure/ armor (see CAD to the right) used 3/4" thick 7050 aluminum walls, covered by a layer of Kevlar and another of titanium Ti-6Al-4V. The wall sections that hold the drum were 1" thick. The top and bottom covers were made out of 1/4" thick 7075-T6 aluminum. Pockets had to be selectively milled in all walls and covers to relieve weight, see chapter 9. A few internal mounts that required high stiffness, but not high strength, were made out of 6063-T5 aluminum extrusions, which were easier to find than 6061-T6 (all aluminum alloys have roughly the same stiffness and density, but very different strengths).

Almost all the screws made use of threaded holes along the thickness of the plates, which simplifies a lot the assembly task, without the need for nuts. Touro has 423 screws, but only 10 nuts (which are used in the Victors and MS-2 switch contacts).

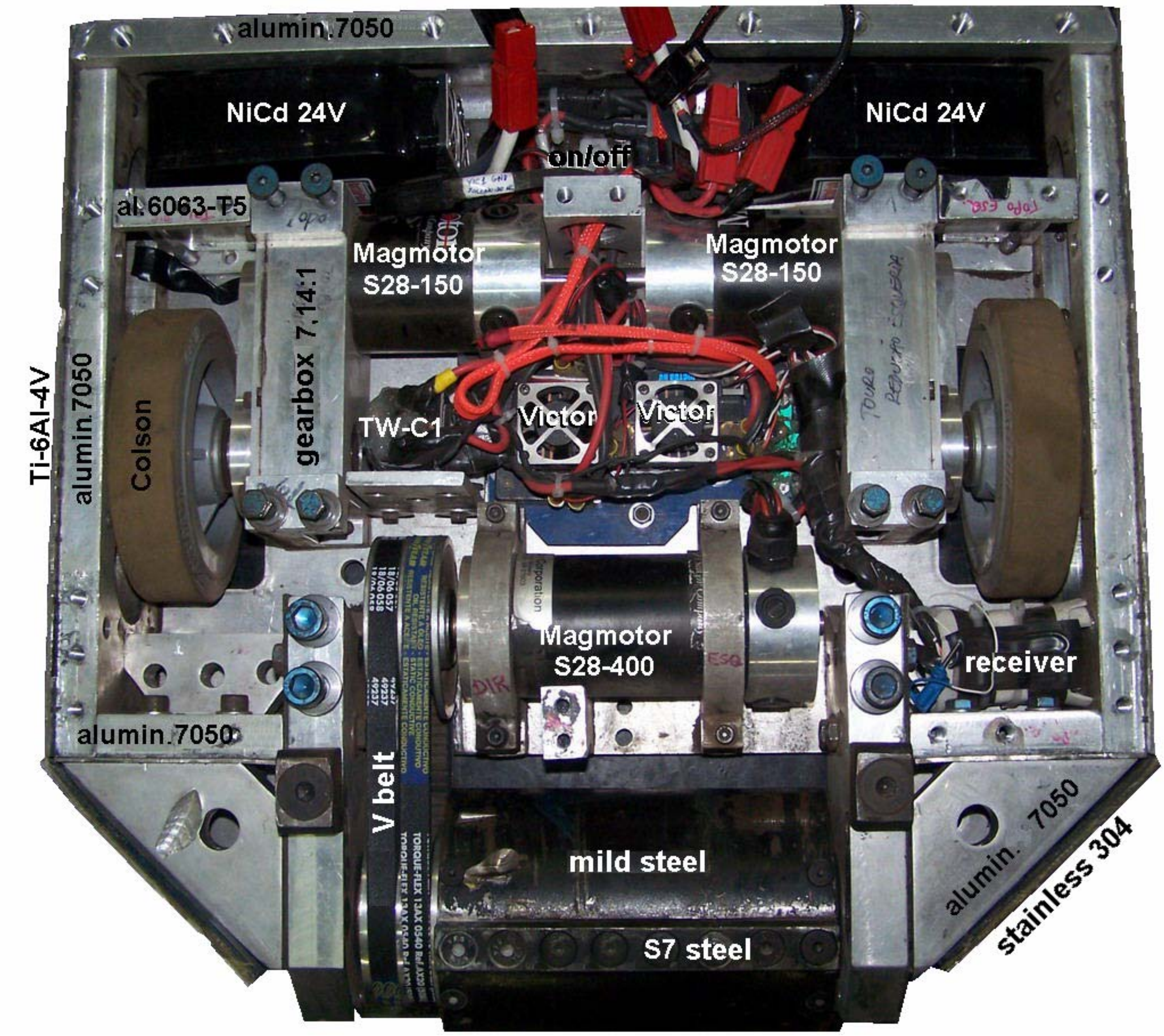
Similarly to Titan, Touro's integrated structure/armor ended up with respectable 40% of the robot's weight, way above the 25% value from the rule. To be able to reach this 40% value, without compromising too much the drive system, weapon and batteries, is not an easy task. These 40% caught the attention of several US builders when they first saw Touro in 2006. A few builders asked us back then if Touro was a lightweight, judging from its size, and a couple asked if it was a heavyweight, looking at the 1" thick walls near the drum. These 40% also help to explain how Touro survived the violent fights against The Mortician in 2006 and Prof. Chaos in 2008.

In summary, a good drumbot might follow a small variation of the weight rule, which would be 20-35-30-15: 20% of the robot's weight in the drive system (a little more than Touro), 35% in the weapon, 30% in the structure and armor (or a little more, but always between 25% and 40%), and 15% in the batteries and electronics (a little more than Touro's 10% to be able to use more batteries). As for other types of robots, certainly other more specific rules can be proposed, however the original 30-30-25-15 rule is always a good starting point.

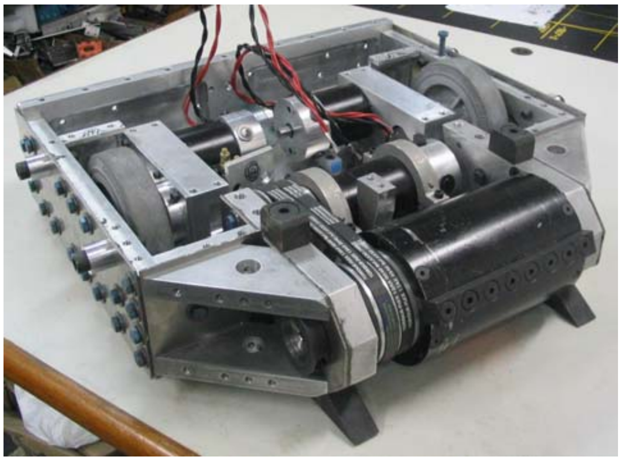
To make it easier for a new team member to get used with (and service) our robots, we've generated exploded assembly views of most of them using Solidworks, as pictured below for Touro.



Touro got bronze, gold and silver medals at, respectively, RoboGames 2006, 2007 and 2008. It had also won 5 Brazilian championships until 2008, including the 2006, 2007 and 2008 editions of the RoboCore Winter Challenge. The pictures in the next pages show a detailed anatomy of Touro, as well as several action shots in combat.


alumin.7050
NiCd 24V
NiCd 24V
on/off
al.6063-T5
Magmotor
S28-150
Magmotor
S28-150
gearbox 7.14:1
Ti-6Al-4V
alumin.7050
Colson
TW-C1
Victor
Victor
Magmotor
S28-400
receiver
alumin.7050
V belt
mild steel
S7 steel
alumin. 7050
stainless 304

steel drums, just turned in the lathe



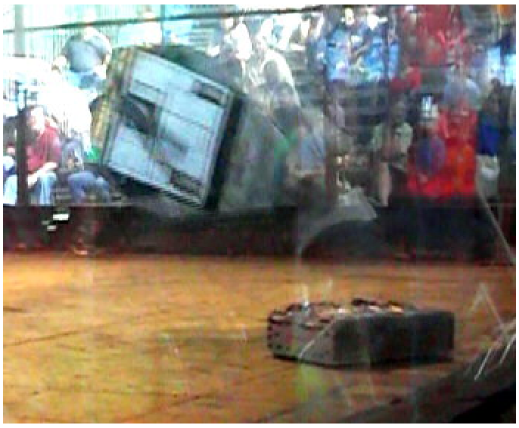
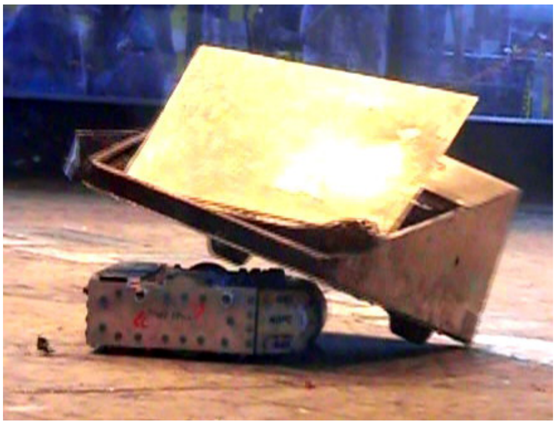
mechanical structure ready, now let's wire it

just born in our San Francisco hotel

repairs during a pitstop

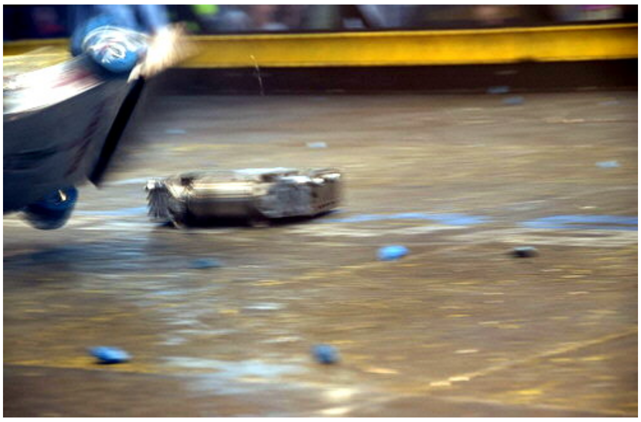
our overwhelmed driver learning that Touro had won the RoboGames 2007 gold medal

Touro had to beat Pipe Wench and Sub Zero to get gold in 2007

 testing the welds from Wiz	 destroying the Destroyer
 sending The Mortician to the graveyard	 telling Pirinah 2 that “size matters not”
 making Stewie look like an UFO	 playing around with Ice Cube

 beautiful titanium sparks against Sub Zero	 Pipe Wench righting itself
 breaking Terminal Velocity’s bar	 flipping Dolly
 Vingador’s retirement	 Orion 3 getting airborne

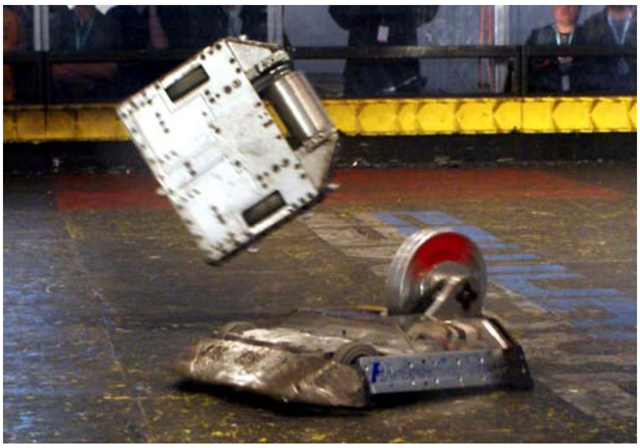
making Emily fly



chewing Pirinah 3’s blue tires

damaging Argus’ flamethrower

making TSA Inspected pop a wheelie

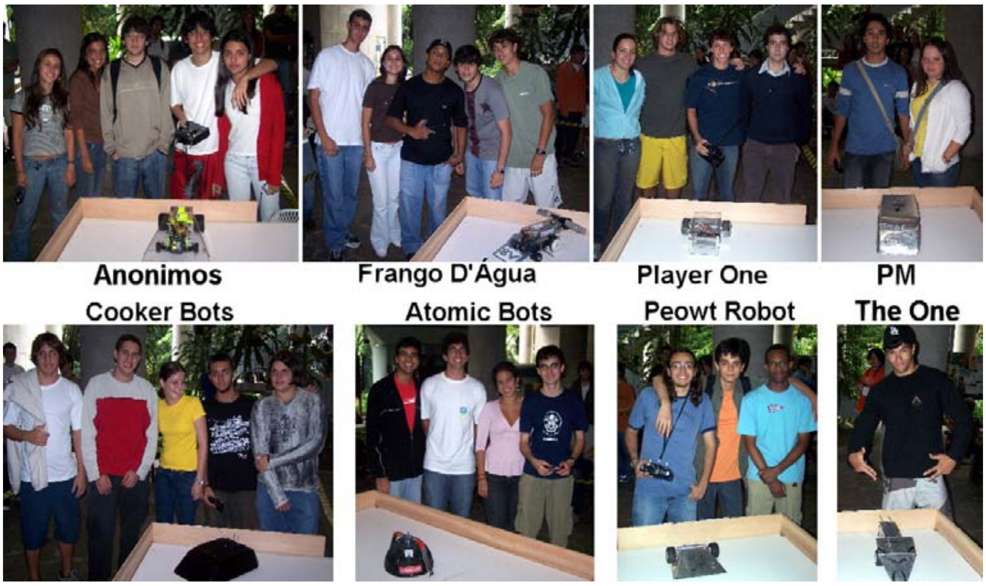


getting some air time from Prof. Chaos…

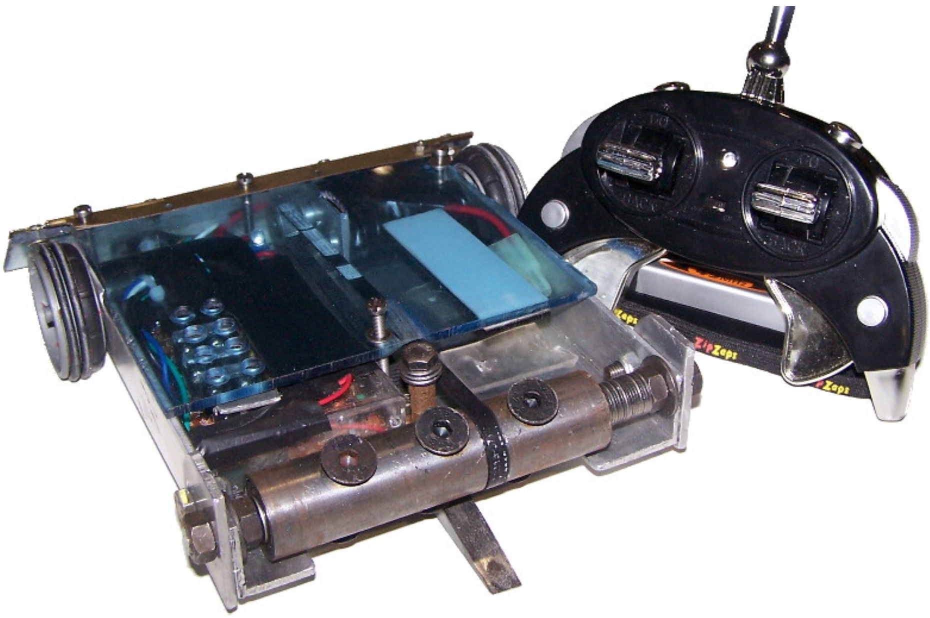
…and giving some too

10.7. Mini-Touro

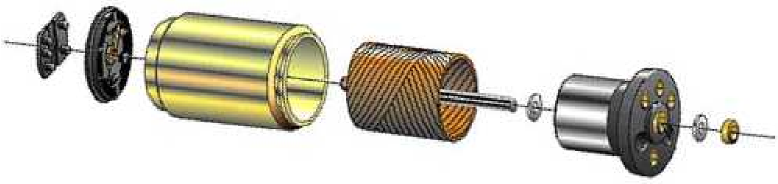
Touro has a father. And it only weighs 1.5kg (3.3lb). That was the (non-official) weight class in an internal combat robot competition at PUC-Rio University, organized by RioBotz for 30 freshman students that were enrolled in the subject “Introduction to Engineering.” Each one of the 8 teams (pictured to the right, with their robots) developed during the term a 1.5kg radio controlled combot using aluminum extrusions, Lexan, toy components, and scrap metal. The competition gathered several students around a small arena in the campus of the PUC-Rio University in November 2005.



RioBotz also developed then a 1.5kg combot, to entertain the audience during the intervals between the fights. So, at the end of 2005, the very first member of the Touro family was born. Named Tourinho (“Little Bull” in Portuguese), this almost-beetleweight was made out of a single 6063-T5 aluminum rectangular extrusion, with a Lexan top cover (pictured below). The radio-control, electronics, NiCd battery and wheels were all adapted from toys. The drum was a scrap piece of pipe with 6 flat head allen screws. The weapon shaft was simply a long 8mm diameter hex screw. The same name Tourinho was later used in our hobbyweight developed in 2006.



The motors used in Tourinho were quite unique. They were coreless (or ironless) DC motors from Faulhaber, meaning that their rotor did not have an iron core. The structural integrity of the rotor only depended on its windings. In this way, the rotor was hollow, allowing the permanent magnets to be mounted within the windings (as pictured to the right). Without the iron core, the motor inductance was extremely low, increasing the life of the brushes and commuter, and the rotor inertia can get so low that a few very small models could reach accelerations of up to 1 million rad/s^2 (with an unloaded shaft, of course). The energy loss was very low, almost as low as in brushless motors. Their disadvantages were the tendency to overheat, because of the absence of the iron core to improve the heat exchange, and their high cost.



These motors had been used in a robotic rover project, until their embedded encoders were damaged. It would be more expensive to have them repaired then to buy new motor-encoder systems, so RioBotz basically got them for free. We used 2 of them to drive Tourinho's two wheels, and a third one to spin the drum.

The robot was a crowd pleaser, and with the lessons learned we were able to design Touro, using the scale factor principles described in chapter 2. Certainly it was much cheaper and faster to build our first drumbot with only 1.5kg, learn from it, and only then face the costs and challenges of creating a middleweight version. We've learned several things from building the 1.5kg version. For instance, the use of a single front ground support under Tourinho's drum seemed like a good idea to guarantee that both wheels would always touch the ground. But this made the robot tilt diagonally whenever it hit an opponent, so Touro was later designed with 2 front ground supports. Tourinho also had traction problems with the wheels so far behind, which helped us place Touro's two wheels close to its center of mass. Tourinho certainly saved a lot of redesign time for Touro.

During the building of Touro, we also decided to generate an improved version of its 1.5kg father. Instead of a 6063-T5 aluminum extrusion, we milled a unibody (pictured to the right, see chapter 2) from a solid 7050 aluminum block. The Lexan cover was replaced by black garolite. We machined a new drum, and replaced the toy NiCd packs with two 11V LiPo batteries connected in series, to generate 22V. From the original Tourinho, we only took the Faulhaber motors, to power the drum and the 1.75" diameter DuBro rubber wheels. This is how our 3lb **beetleweight** Mini-Touro was born, in 2006.

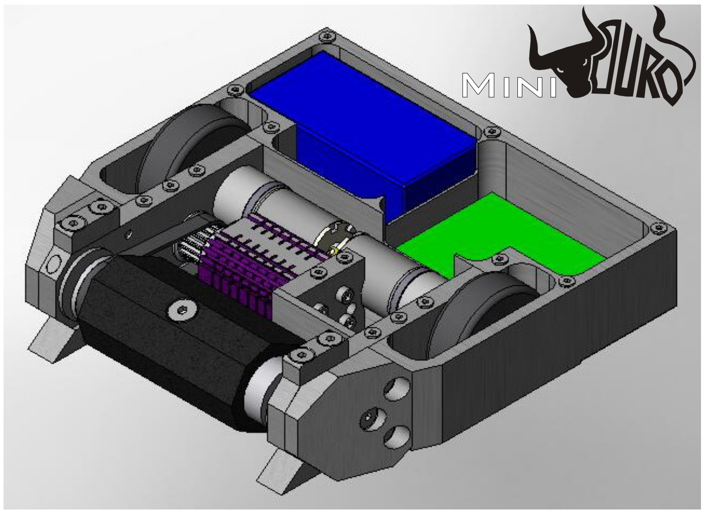
But the Faulhaber motors we had did not include gearboxes. This meant that the drive system top speed was too high (even though its acceleration was not too bad), and that the drum was not as fast as it could get. So in 2007 we replaced them with B-Series 16:1 gearmotors for the drive system, and with a HXT 2835 2700kv (2700RPM/V) inrunner brushless motor, with a Phoenix 25 speed controller, to power the drum. A 3M-size timing belt was attached to the brushless motor through a timing pulley. This belt fitted inside a smooth groove in the robot's drum (pictured to the right), which allows it to slide during impacts.

The batteries, which were originally connected in series to deliver the required 22V for the Faulhaber motors, were then wired in parallel to provide 11V, in order not to burn the new lower voltage motors, and 1,450mAh. The resulting top speed of drum, considering the speed reduction from the belt, was about 10,000RPM.

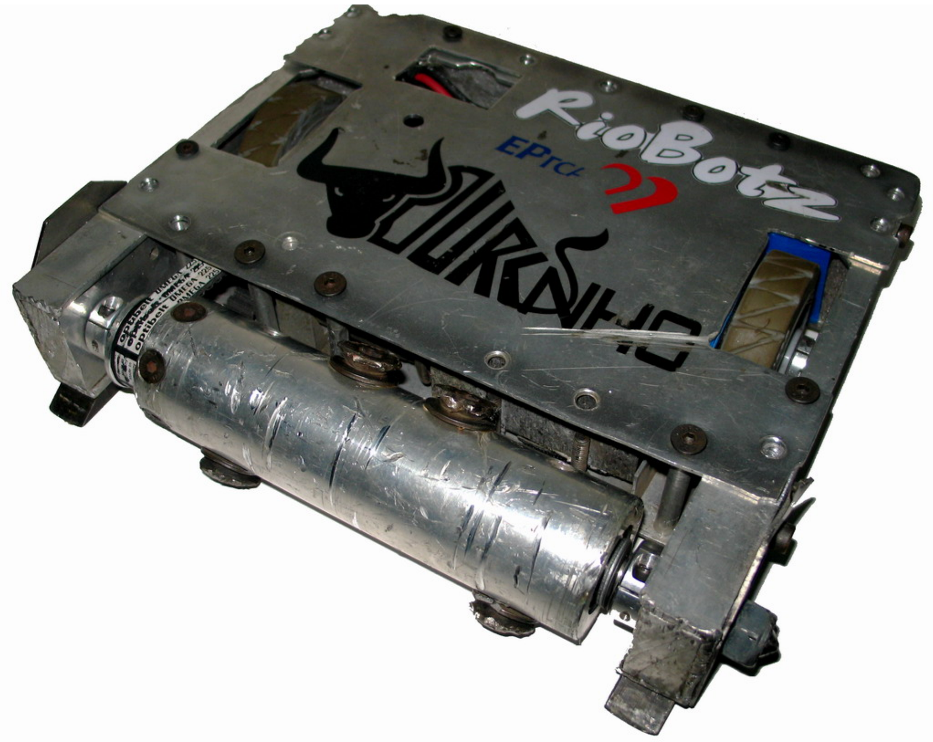
Mini-Touro was able to get the gold medal at RoboGames 2006, still in its Faulhaber version. The final match was against the powerful spinner Itsa (pictured to the right).

Mini-Touro faced Itsa again two years later. After a tough final match against the undercutter One Fierce Weed Wacker, Mini-Touro was able to get another gold medal at RoboGames 2008.

Mini-Touro, so full of itself, was later spotted subjugating super-heavyweight Ziggy.

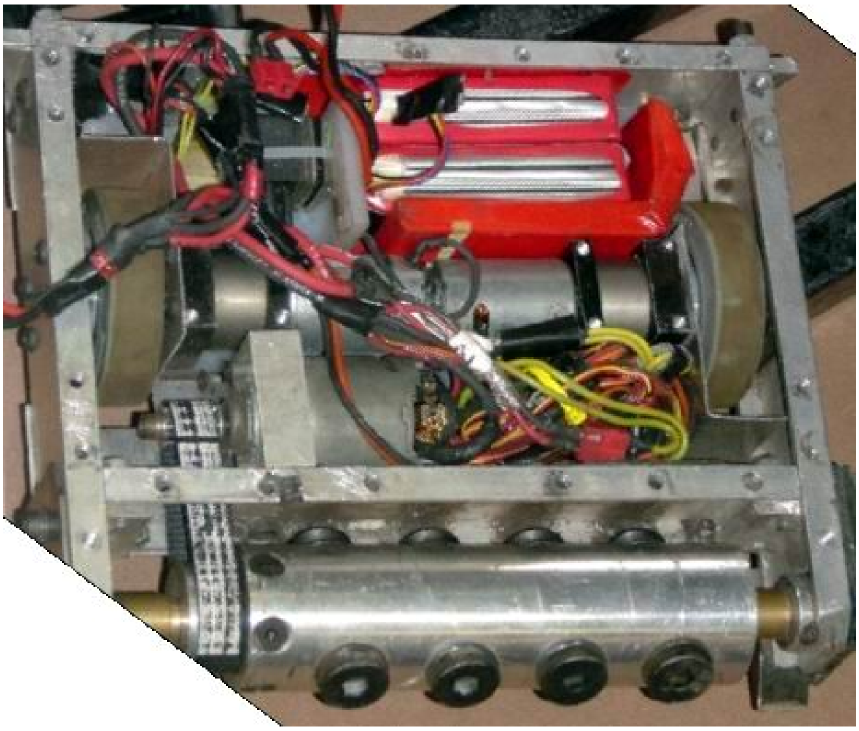


10.8. Tourinho



The RoboCore Winter Challenge featured in 2006, for the first time in Brazil, a **hobbyweight** (12lb) competition. So we decided to create a bigger brother to Mini-Touro. Tourinho (pictured to the right) was born. It is a hobbyweight drumbot, with walls made out of 1/8” thick 2” high 6061-T6 aluminum extrusions, a 2mm (about 5/64”) thick 2024-T3 aluminum bottom, and a 12mm (almost 1/2”) diameter titanium weapon shaft.

Tourinho originally had a Lexan top cover and an 8” wide ST-52 steel drum spinning at 5,700RPM. The Lexan top cover was later replaced with a 2024-T3 aluminum sheet, to avoid cracking around the countersunk holes. The ST-52 steel was replaced with 6351-T6 aluminum to increase the drum diameter to 2”, which was then overvolted to reach an 11,000RPM top speed.



Two surplus Buehler gearmotors had been used in 2006 to drive two 3” diameter Colson wheels, controlled by a Scorpion XL board.

These drive motors were replaced in 2007 with 16:1 36mm planetary gearboxes from Banebots, powered by RS-540 motors, as pictured to the right.

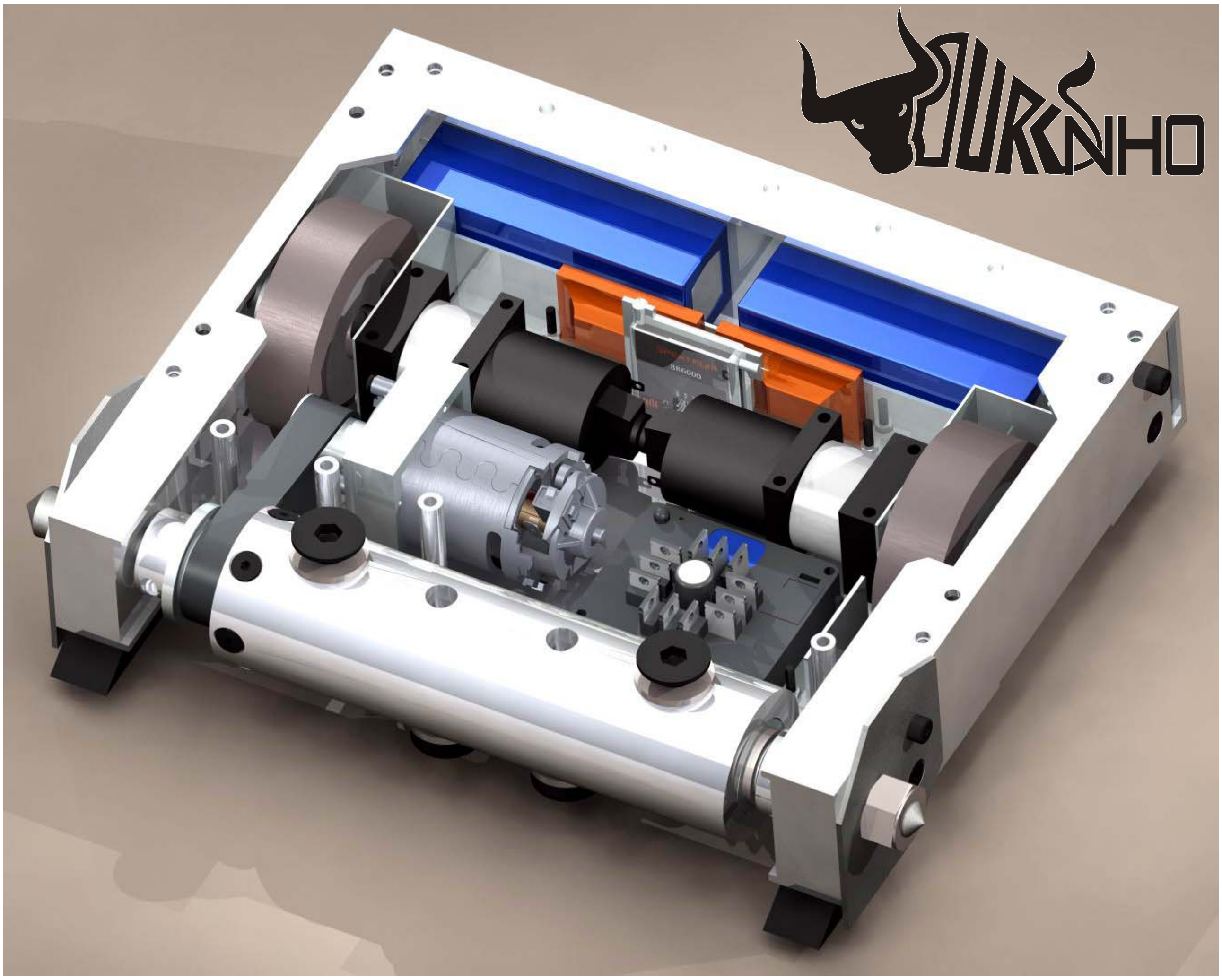
In 2008, the not-so-reliable RS-540 was replaced with Integy Matrix Pro Lathe motors, adapted to the same Banebots gearboxes.

The weapon motor was an 18V DeWalt, powered by a 40A automotive relay. The 16.8V NiCd pack was obtained from removing the top cell from a DeWalt cordless drill battery, reducing its height to less then 2” to fit inside the robot.

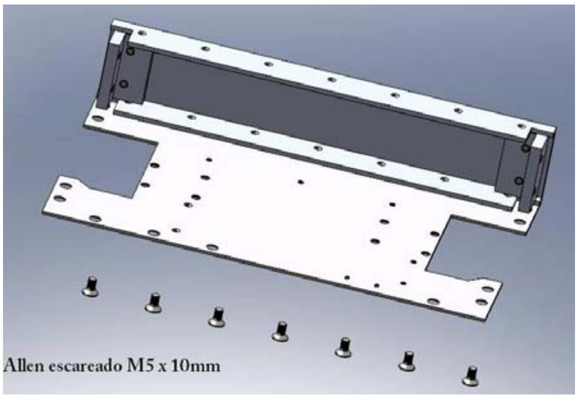
In 2007, the NiCd battery pack was changed to two 2,100mAh 3S LiPo batteries (shown in red in the previous picture), connected in series to increase the weapon motor voltage to 22.2V.

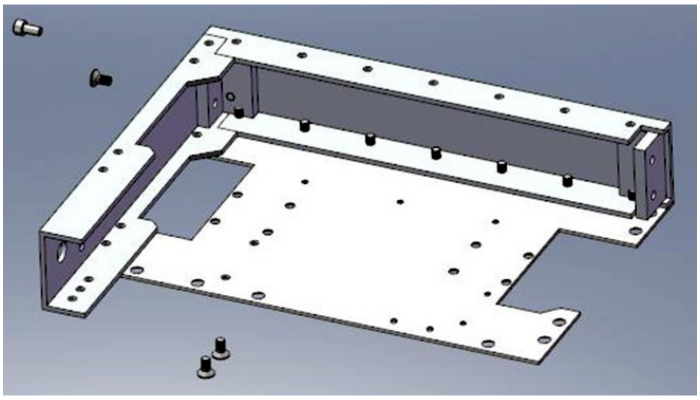
In 2008, the relay was switched to a Victor speed controller. The Victor can not only control the speed of the weapon motor, but it also allows the drum to reverse its spin direction if the robot gets flipped over.

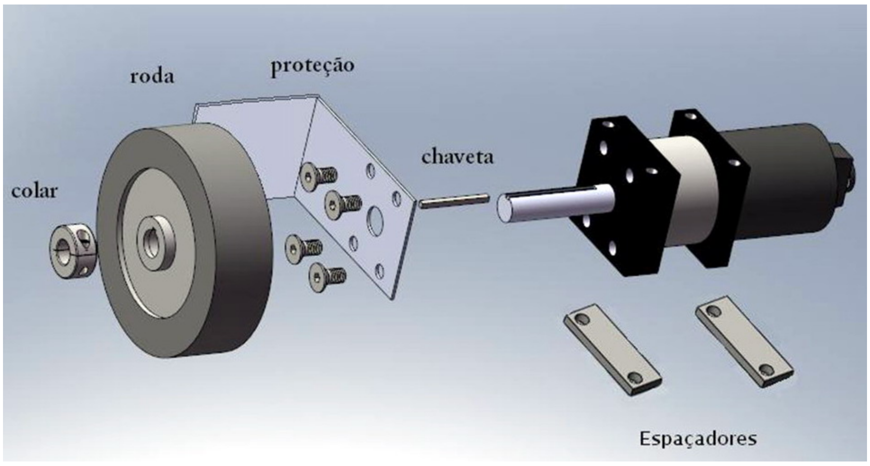
Also in 2008, the 2,100mAh LiPo batteries were replaced with new 2,200mAh LiPo with higher discharge rate (in blue in the CAD rendering below). The new batteries were repositioned inside the robot to allow the chassis to be sized down to 11” wide × 9.15” length. With the saved weight, it was possible to increase the thickness of the side walls to 1” in the region where the weapon shaft is supported, as shown below in a Solidworks rendering.

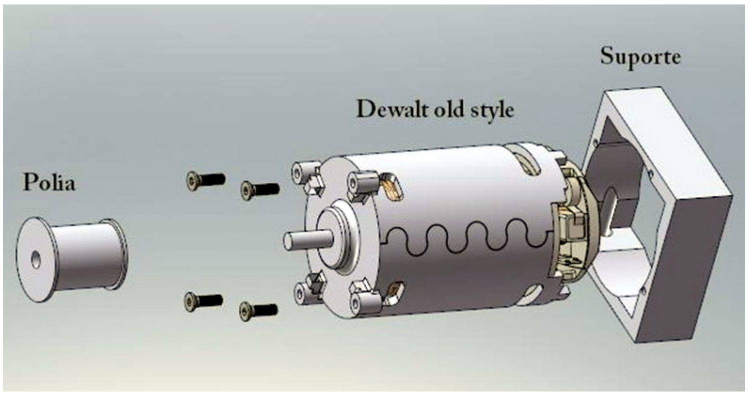


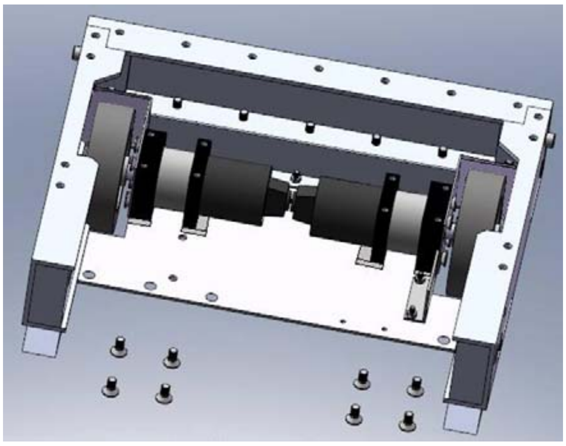
Similarly to Touro, Tourinho also has an assembly guide featuring several exploded views, which is summarized below.

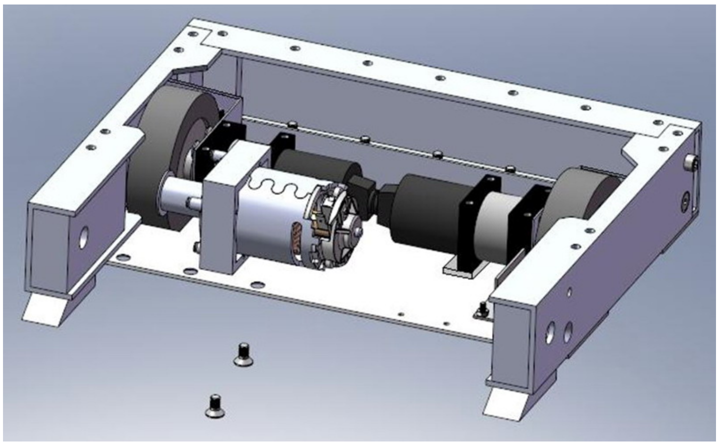


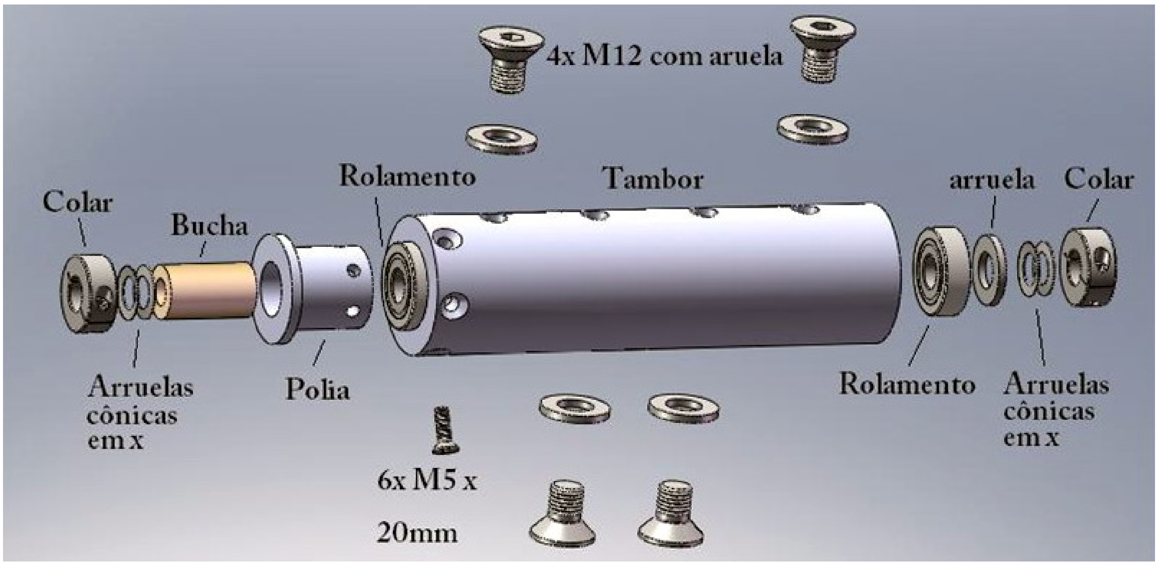


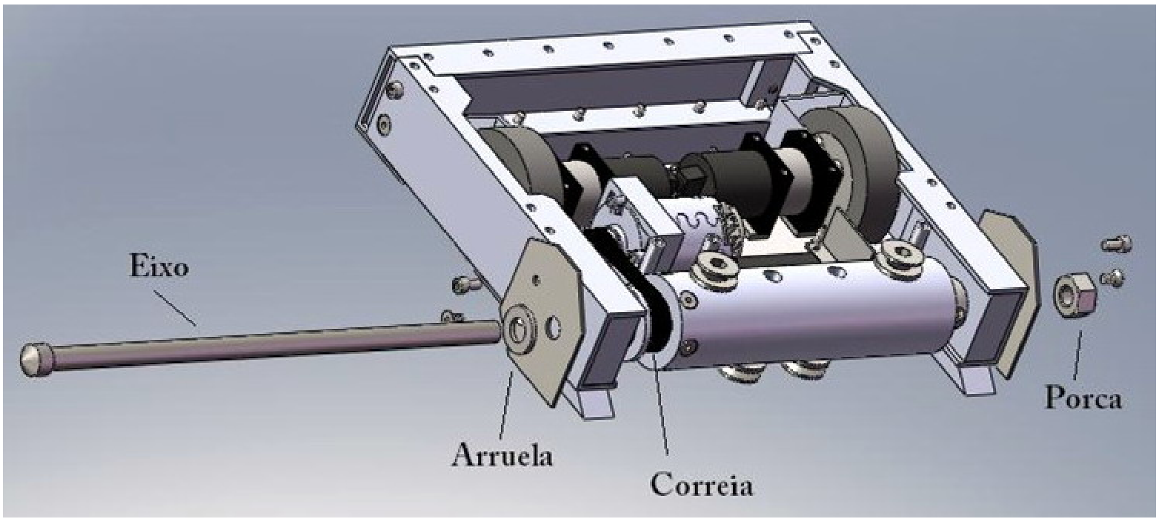


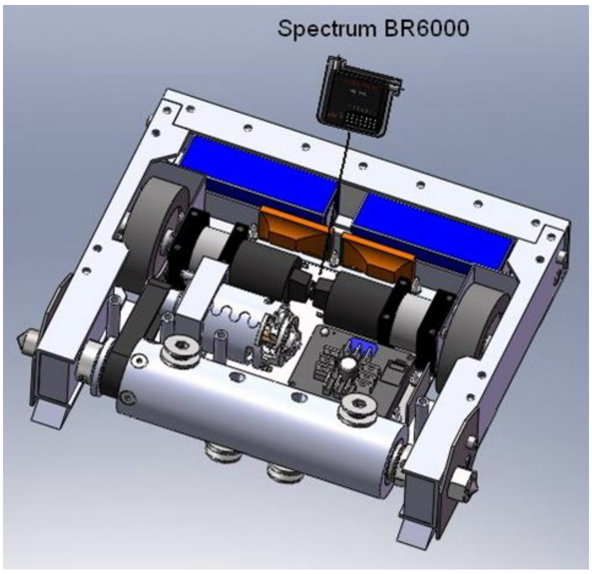


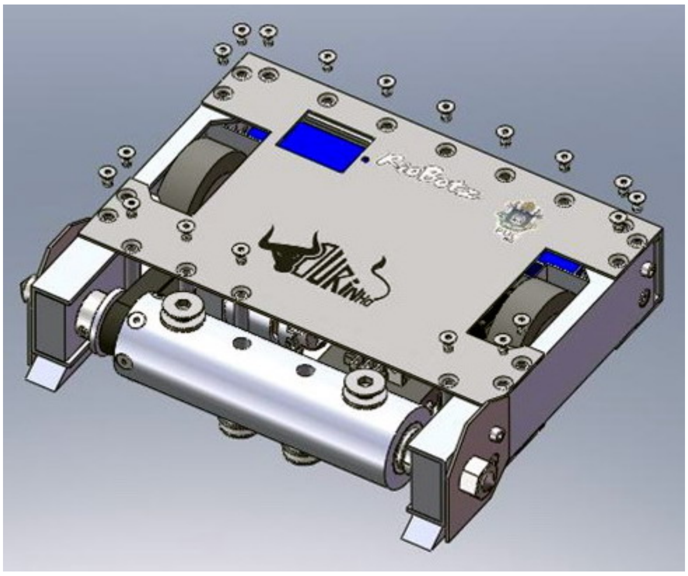












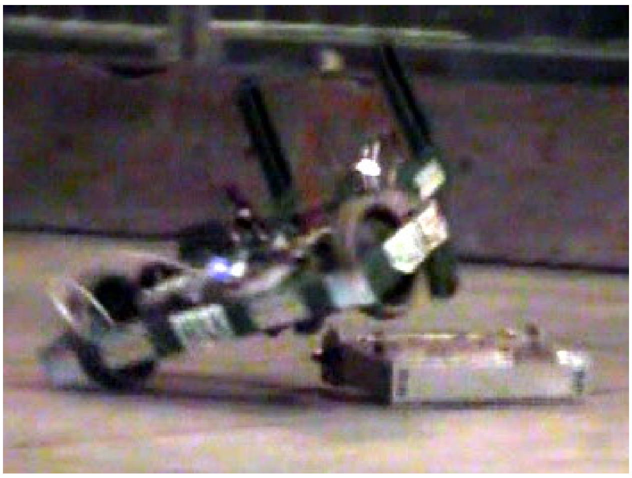
Tourinho became the champion of the 2006 and 2007 editions of the RoboCore Winter Challenge. More pictures from Tourinho are shown next.

flipping the vertical spinner Agressor

hitting Lasca Bit from Team Proteus

drum vs. drum, against Xpow

launching the shell spinner Butcher

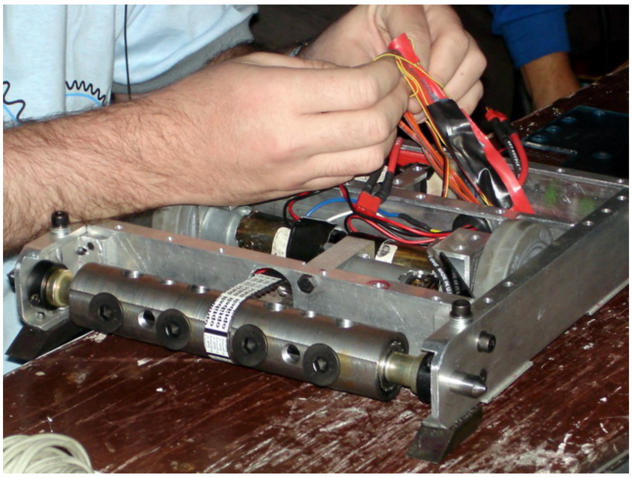


challenging a powerful featherweight

finishing Catatau

LTFD – Little Tourinho’s Flipper Drum

in mid-air, trying to self-right

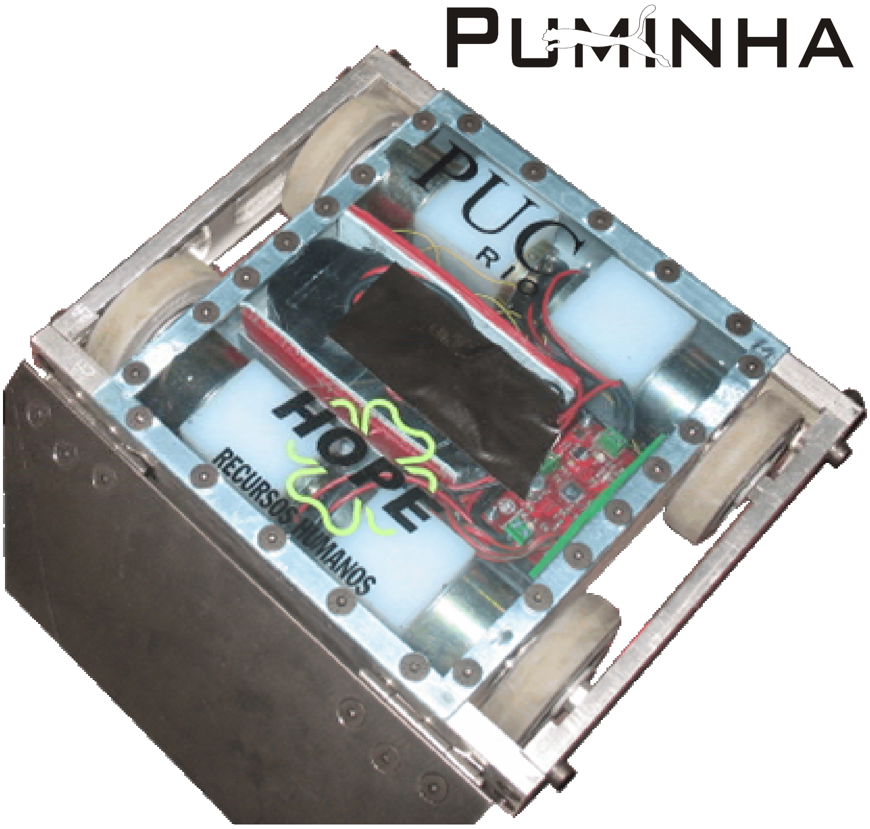


untangling wet noodles

getting my daughter hooked since she was 3 y.o.

active RioBotz robots back in 2006: middleweights Titan, Touro and Ciclone in the back, and Tourinho (hobby), Mini-Touro (beetle) and Puminha (hobby) in the front

10.9. Puminha



As soon as we finished building Tourinho, we decided to build another **hobbyweight** (12lb) robot to compete at the RoboCore Winter Challenge 2006. We then built, in about a week, our first wedge, called Puminha ("Little Puma" in Portuguese, pictured to the right).

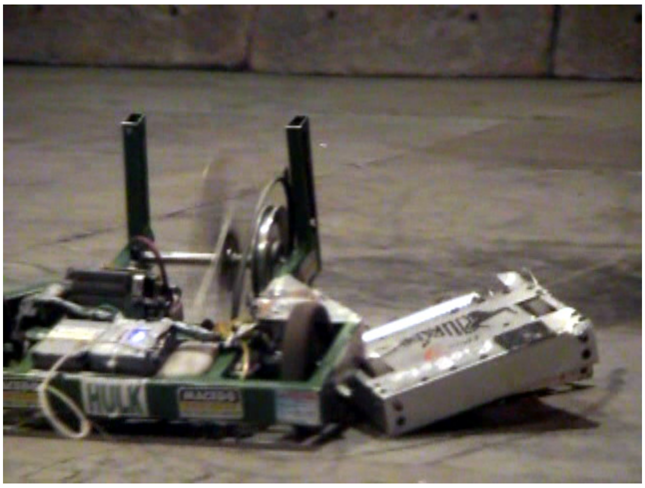
Originally, four surplus Pittman gearmotors (pictured to the right, partially covered by nylon mounts) were attached to 3" diameter Colson wheels, controlled by a single Scorpion XL board and powered by a 16.8V NiCd battery pack. The speed controller was later switched to a pair of Victors.

The side walls were made out of 1/4" thick 6061-T6 aluminum extrusions, and 1/8" thick for the front and rear. Lexan was used in both top and bottom covers, which was later replaced by 2024-T3 aluminum. The wedge was originally a 2mm thick titanium grade 5 plate, borrowed from Touro's armor, attached to the front wall using two stainless steel door hinges.

After a few broken gears from the Pittman gearmotors, we decided in 2007 to switch them to four 16:1 36mm planetary gearboxes from Banebots, powered by RS-550 motors, a little faster than the RS-540 previously used in Tourinho.

In 2008, we switched the RS-550 to even better motors, the Integy Matrix Pro Lathe, using the same Banebots gearbox, as pictured to the right. In addition, the gearboxes were modified following Nick Martin's recommendations, described in the March 2008 edition of Servo Magazine, to avoid any broken last stage pin.

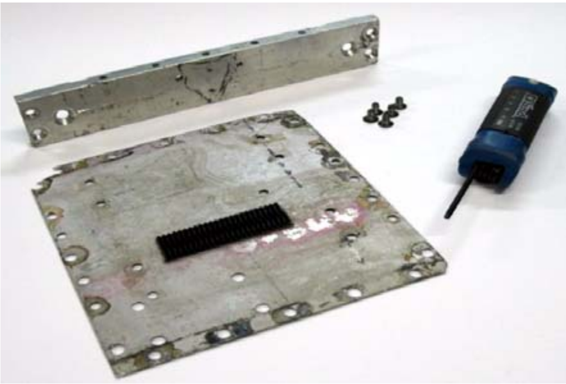
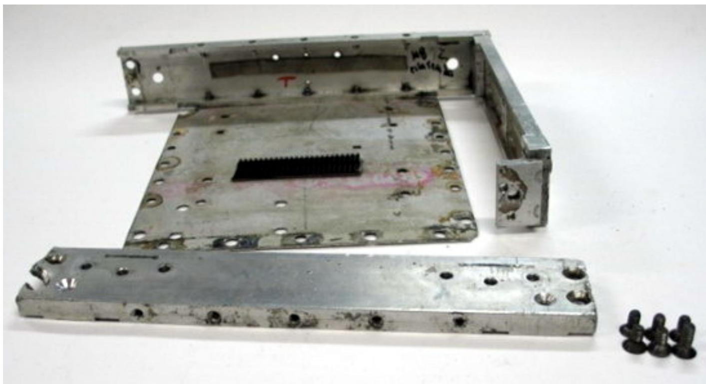
The new drive motors made Puminha become even faster and with more traction. In fact, during the first Brazilian multi-bot match ever, in 2008, Puminha was able to carry both its hobbyweight fellow Tourinho and its featherweight opponent Hulk all over the arena, as pictured to the right (Puminha is hidden under both robots).

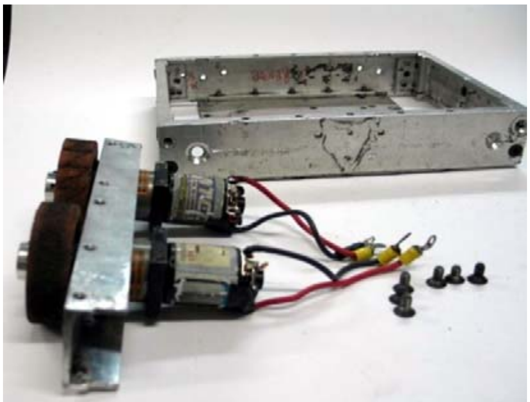


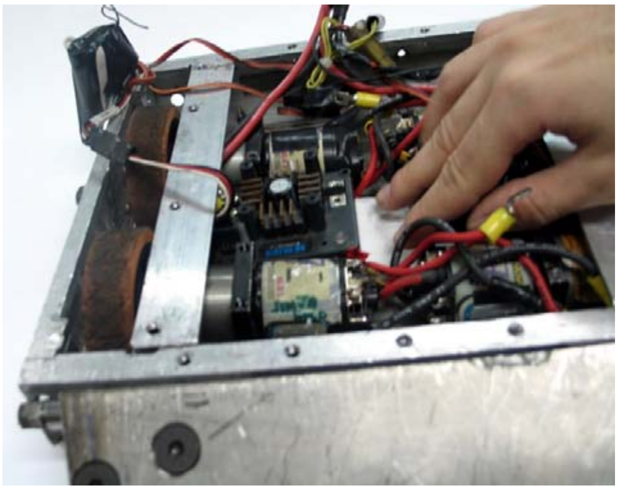
By 2008, Puminha had already won two Brazilian championships organized by RoboCore: the 2007 ENECA and the 2008 Winter Challenge. The picture to the right shows Puminha launching the shell spinner Butcher all across the arena during the 2007 final match.

As seen below, the 2mm thick wedge was in a very bad shape by the end of the 2007 season. In 2008, the wedge was upgraded to a 1/4” thick titanium grade 5 plate, attached using heavier duty door hinges. In 2008, the battery was also upgraded to a 4S (14.8V) LiPo Polyquest with 4,500mAh, making Puminha so fast that only little Anakin can drive it.

Puminha also features step-by-step assembly instructions, aimed to help new team members. The main steps are shown below, in an old-school style using photographs, instead of using exploded-view Solidworks images such as in Touro's and Tourinho's assembly instructions. Note in the pictures the 2008 version of the wedge, with thickness increased from 2mm to 1/4".



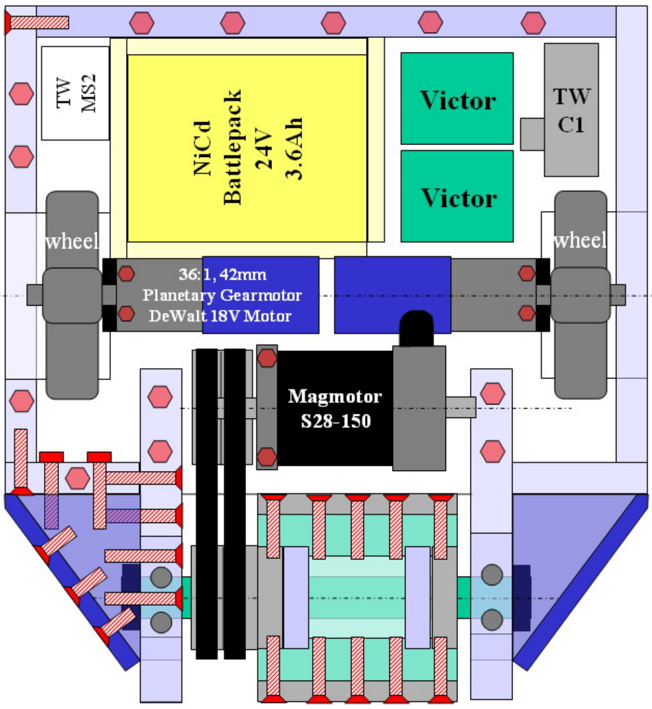



10.10. Touro Light

After Touro's bronze medal at RoboGames 2006, we decided to build another robot to compete in one of the upper weight classes. We decided to test once again our scale factor theories to see if we could create an effective **lightweight** (60lb) based on Touro's design. Touro Light ended up so similar to Touro that many people sometimes get confused about which is which (even ourselves).

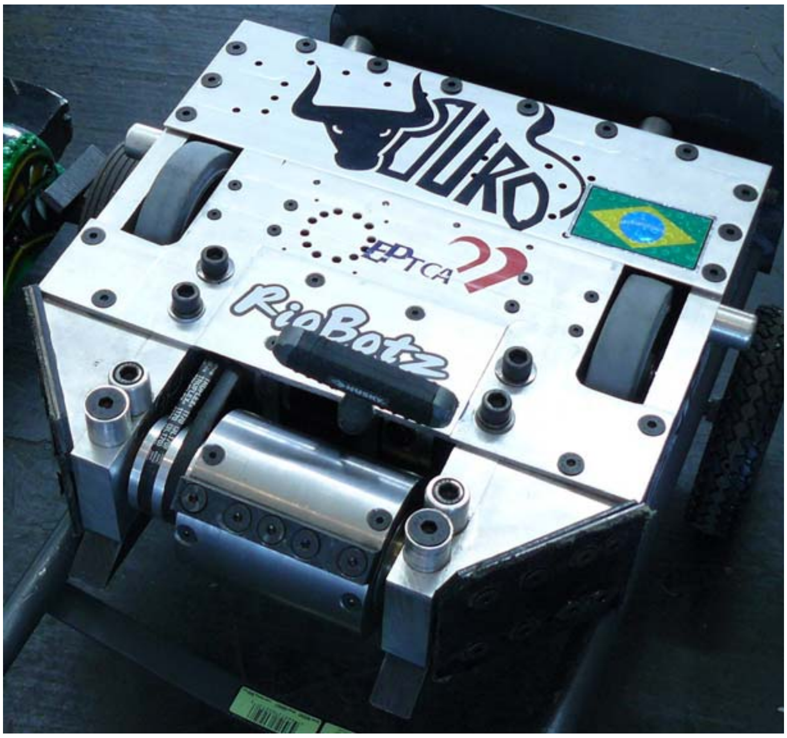
Since Touro is 2 times heavier than Touro Light, their scale factor should be the cube root of 2, which is 1.26. The actual external dimensions of Touro's chassis, without the drums, wheels, and front ground supports, are 20.3" (width) × 19.25" (length) × 4.5" (height). Considering the 1.26 scale factor, these values would translate to 16.1" × 15.3" × 3.57", very close to Touro Light chassis' actual dimensions 15.6" × 15.9" × 3.50".

Touro Light used two 36:1 42mm Banebots gearboxes, powered by 14.4V Mabuchi RS-775 motors, to drive its two 5" diameter Colson wheels. Even with the speed controllers trimmed to a 20V limit, the overvolting of the 14.4V RS-775 motors caused them to overheat. This overheating forced us to replace them almost every 2 matches during RoboGames 2007.



In 2008, the RS-775 were upgraded to 18V DeWalt motors, adapted to the same Banebots gearbox, as seen in the PAD (Powerpoint Aided Design) drawing on the right. However, the more powerful DeWalt motors ended up causing the planetary gear pins from the last stage of the gearbox to break at almost every match during RoboGames 2008. This problem should be solved in the future by making modifications to the gearbox, increasing in about 1mm the diameter of the last stage pins, as recommended by Nick Martin from Team Overkill.

The robot's drive system ended up so light that it was possible to beef up the structure. Touro Light's integrated structure/armor had basically the same thicknesses as its bigger brother Touro. It used 3/4” thick 7050 aluminum walls, and the top and bottom covers were made out of 1/4” thick 7075-T6 aluminum. The wall sections that held the drum were 1” thick. Similarly to Touro, pockets had to be selectively milled in the walls and covers to relieve weight.

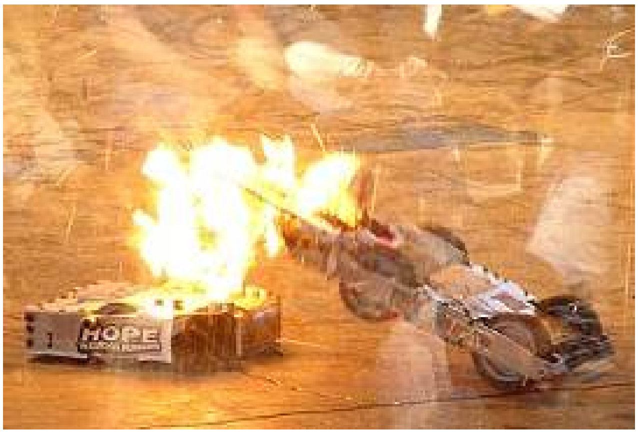
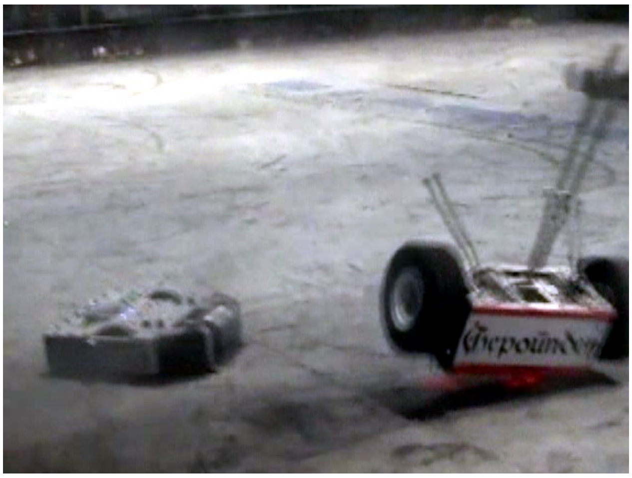


The drum was made out of a 1” thick 410 stainless steel cylinder, with two tempered S7 steel bar teeth. We machined a double pulley to be fitted to the drum, allowing it to be powered by a pair of 3L-size V-belts. The drum was mounted to a 1” diameter titanium grade 5 shaft. It was powered by a Magmotor S28-150 to reach about 6000RPM.

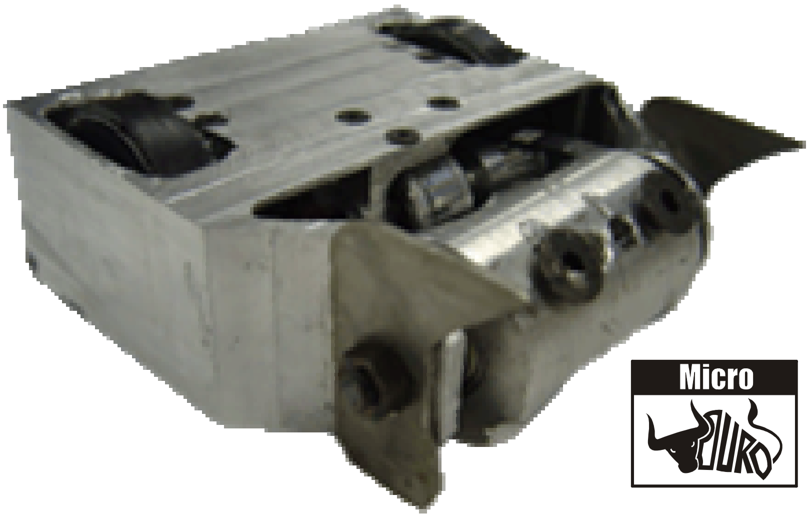
To keep down Touro Light’s development cost, we tried to use in its design several spare parts from Touro. They both used identical front ground supports, two Victors to control the drive motors, a TW-C1 solenoid to power the weapon, a MS-2 switch, and the same 24V NiCd Battlepacks (though Touro Light only used one pack instead of two). In addition, both used Magmotors S28-150, Touro Light for the weapon and Touro for the drive system. These shared components helped to keep low the number of spare parts needed in a competition, saving us a lot of excess baggage fees when traveling to overseas events.

Touro Light ended up getting the gold medal at RoboGames 2007, together with its big brother Touro (pictured to the right), both undefeated in their weight classes.

The photos below show Touro Light in action in 2007 and 2008.

 cooling down Texas Heat	 flipping Crocbot
 ripping off Conniption’s drivetrain	 pounding Herr Gepoünden
 facing K2 in the RoboGames 2007 final	 launching the Rocket

10.11. Micro-Touro



In early 2007 we decided to build Micro-Touro (pictured to the right), the first Brazilian **antweight** (1lb) combot. We tried to incorporate in its design several features from our beetleweight Mini-Touro, such as a unibody chassis, milled from a single 7050 aluminum block, brushless weapon motor, and LiPo batteries. However, we were not so careful with the scale factor between beetles and ants, which is 1.44 (cubic root of 3). Micro-Touro's chassis ended up with basically the same height as Mini-Touro's, instead of 1/1.44 = 69.4% of that value. Such big height makes it easier to get hit by opponents and to get flipped over. A drum with a large diameter was necessary to match the robot's height, which forced us to use aluminum in it (6351-T6 alloy), instead of steel, in order to save weight. The drum's teeth were class 12.9 flat head allen screws. The weapon shaft was simply a long 6mm diameter hex screw. Two titanium strips were used as front ground supports, working as well as armor. The top cover was made out of black garolite.

Two Sanyo 50:1 micro-geared motors were used to drive two 1.5” diameter rubber foam Lite Flite wheels, controlled by two Banebots BB-3-9 speed controllers and a micro-receiver. One 700mAh 3S (11.1V) LiPo battery was used to power the entire robot.

The weapon used a LittleScreamers “DeNovo” micro outrunner brushless motor (pictured to the right), with 1,250kv (RPM/V), capable of sustaining 11A, powered by a hexTronik PRO 10A speed controller. A 2M-size timing belt was attached to the brushless motor through a timing pulley. This belt fitted inside a smooth groove on the side of drum, which allowed it to slide during impacts.

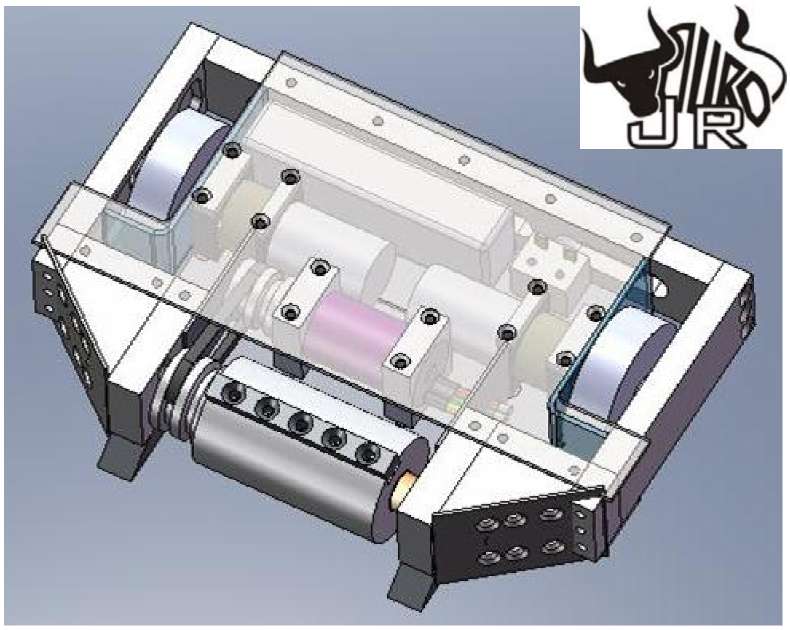
Micro-Touro didn't do well in the 2007 RoboGames. The 1.5” diameter wheels ended up leaving a very low ground clearance. This caused Micro-Touro to frequently get stuck due to the floor deflections of the small combat arena. This will be taken care of in the future using a shorter chassis, which will also help lowering the robot's center of mass, making it more difficult to get flipped over.

10.12. Touro Jr.

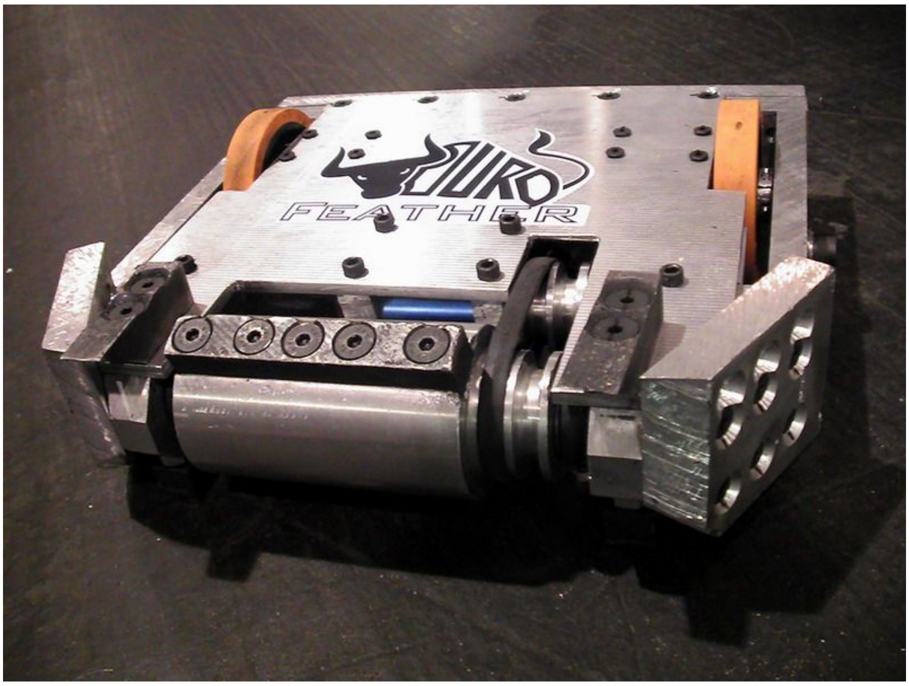
As the Brazilian hobbyweight robots were getting increasingly powerful, we've realized that Tourinho's 1/8” thick side walls started to look very thin. Keeping that in mind, we designed in mid-2007 our third **hobbyweight** (12lb), named Touro Jr. To be able to increase the thickness of the structure/armor, we faced the challenge to reduce the overall size of the robot. The result was a drumbot chassis with very short length (about 8”) and height (1.75”), and respectable 5/8” thick 7050 aluminum walls. Nevertheless, pockets had to be selectively milled in the walls to relieve weight. The robot ended up a little wide, about 12.5”, to be able to fit the drivetrain gearmotors.

Two 16:1 36mm planetary gearboxes from Banebots, powered by RS-540 motors, were used to drive two 3” diameter Colson wheels. Two Banebots BB-12-45 speed controllers were used in the drive system.

The 6351-T6 aluminum drum used flat head allen screws as teeth. It was powered, through a pair of 2L V-belts, by a Feigao 540-06XL 2779kv (RPM/V) inrunner brushless motor. The brushless speed controller was a hexTronik PRO 120A, powered by a 4S (14.8V) 4,500mAh Polyquest LiPo battery. Considering the V-belt speed reduction, this 2,779RPM/V brushless motor, in theory, would be able to spin the drum up to 30,000RPM. The actual top speed was certainly lower due to bearing friction and air resistance, but it was still so high that our tachometer was not able to measure it. This very high speed makes it difficult to launch the opponents. The drum teeth, instead of biting into the other robots, end up just grinding them. During the weapon tests, it was not easy to bite into aluminum blocks. But when it did bit, the impact was so high that the windings from the Feigao brushless motor detached from the can and broke off the speed controller contacts. We're currently looking for a replacement motor for the weapon, one with a lower RPM/V value.



10.13. Touro Feather

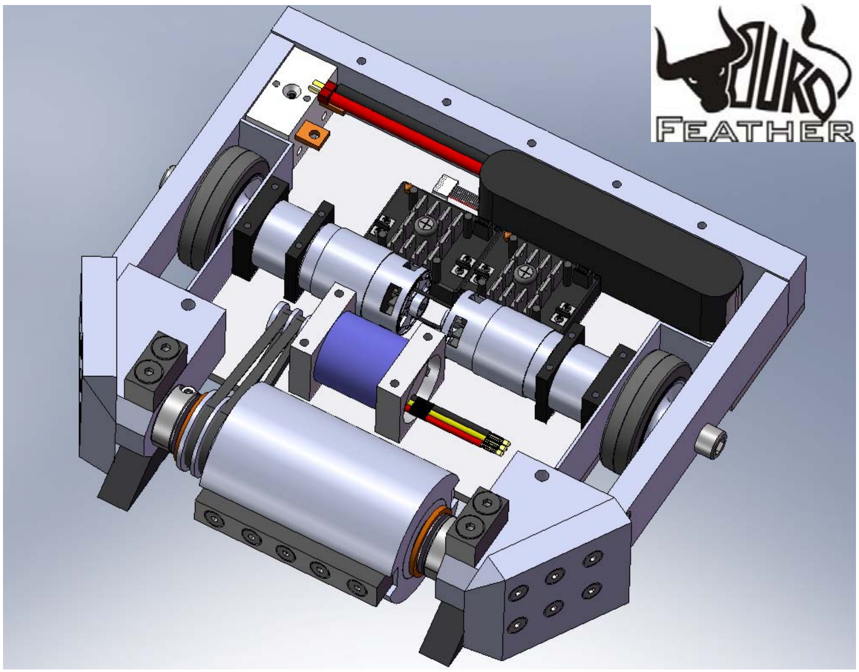


We started to build Touro Feather (pictured to the right) in early 2008, as soon as it was announced that the RoboCore Winter Challenge would debut a **featherweight** (30lb) class. Touro Feather was basically a longer and heavier version of our hobbyweight Touro Jr. Its structure/armor was made out of 5/8” thick 7050 aluminum, with 3/16” thick 2024-T3 top and bottom covers. Two 3/4” thick 7050 aluminum plates were mounted diagonally in the front to work as ablative armor.

The 6351-T6 aluminum drum used tempered S7 steel bars as teeth, the same ones from Touro Light. It was spun by a KB45-08L 2300kv (RPM/V) inrunner brushless motor using a pair of 3L V-belts. The brushless speed controller was a hexTronik PRO 120A, powered by a 4S (14.8V) 4,500mAh Polyquest LiPo battery. Considering the V-belt speed reduction, the theoretical top speed of the drum would be a little under 30,000RPM. Even including the bearing friction and air resistance, the actual top speed was still so high that the drum teeth ended up grinding the opponents instead of biting them. We had to slow down the drum to less than half of its top speed to be able to bite and launch the other robots. Even at this lower speed, there was enough energy to launch the wedge Titanick a few feet up in the air during RoboGames 2008, as seen on the right.

Another problem we had with the high RPM/V weapon motor was regarding its low starting torque. Brushless motors inherently have a lower starting torque than DC motors, a problem that is exacerbated when their RPM/V is high. Since the 3L V-belts were relatively stiff and their pulleys had small radii, sometimes the drum would simply not start spinning due to such lack of starting torque. We hope to solve this problem in the future by switching the 3L V-belts to the thinner and more compliant 2L type.

Touro Feather used two modified 12:1 42mm Banebots gearboxes with RS-775 motors for its drive system, controlled by two Victors, see the CAD drawing to the right. These RS-775 motors from Mabuchi were not the common 14.4V version. They were an 18V model, used in very old DeWalt cordless drills, borrowed from the drivetrain of our retired middleweight Ciclone. A MS-2 switch was used to turn the robot on or off.

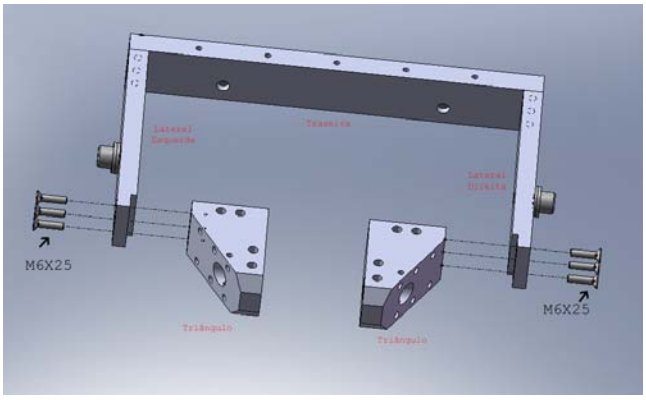


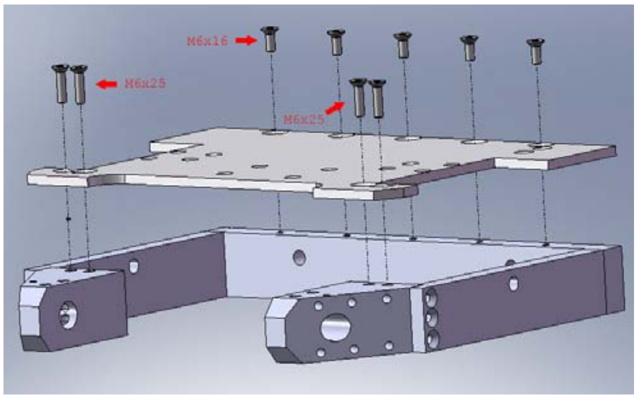
Touro Feather used two 4” diameter orange wheels from Banebots. The color denotes the wheel hardness: green for 30 Shore A (too soft), orange for 40 Shore A (soft) and blue for 50 Shore A (a little soft). Their low cost and finished keyed bore make them an attractive option. We've tested the three types. On a clean floor, the green type provided the best traction, followed by the orange and then the blue.

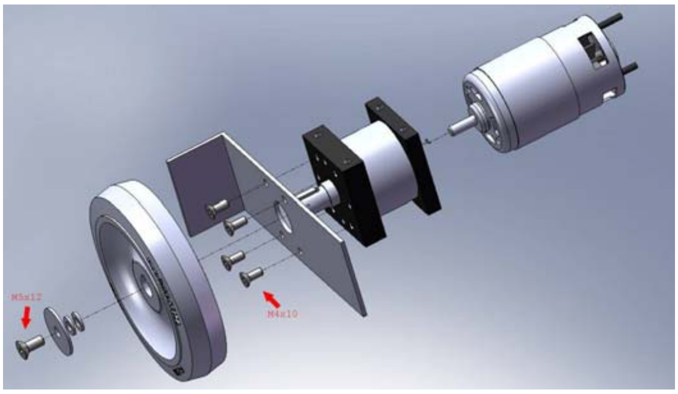
However, there are a few catches if using them in combat. Because both green and orange types were very soft, the dirt on the combat arena floor would stick very easily to them, compromising traction. They would also get worn out very quickly during aggressive driving tests. In all three types, the polypropylene cores were thinner than Colson wheel cores, which could lead to a low resistance to direct hits.

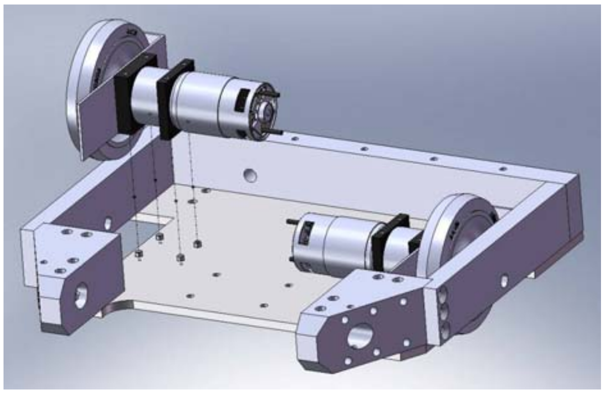
Finally, a critical feature was the poor bonding of their rubber tread to the polypropylene core. A shallow cut from the blade of the vertical spinner Hulk, at the RoboCore Winter Challenge 2008 final, was enough to rip off the entire orange rubber tread from the black core. The remaining rigid black core (pictured to the right), besides barely touching the ground, was not able to provide any traction. Fortunately, we could count on Daniel's ability to drive with only one wheel to win the match. Colson wheels under similar conditions would only break locally (as pictured to the right) and still be functional, as we've experienced several times against other spinners. Until those issues are corrected, we'll be using Colson wheels in Touro Feather, not only due to their better resistance, but also due to the higher 60 Shore A hardness, better suited for dirty combat robot arenas.

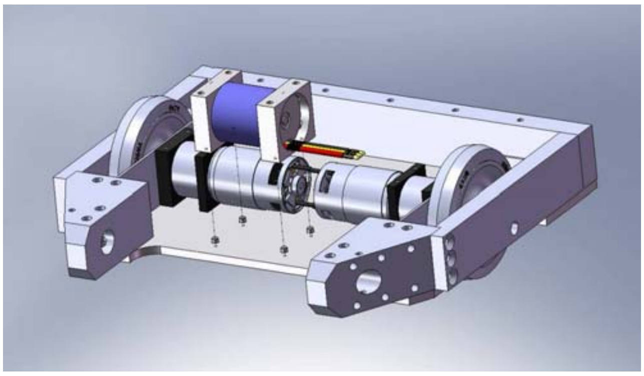
Despite these drivetrain and weapon problems, Touro Feather was able to get fourth place in its debut at RoboGames 2008. One month later, it became the champion of the RoboCore Winter Challenge 2008. The figures below show the exploded assembly views of Touro Feather.

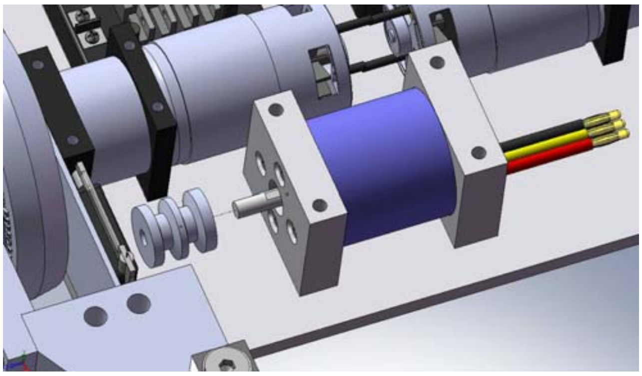


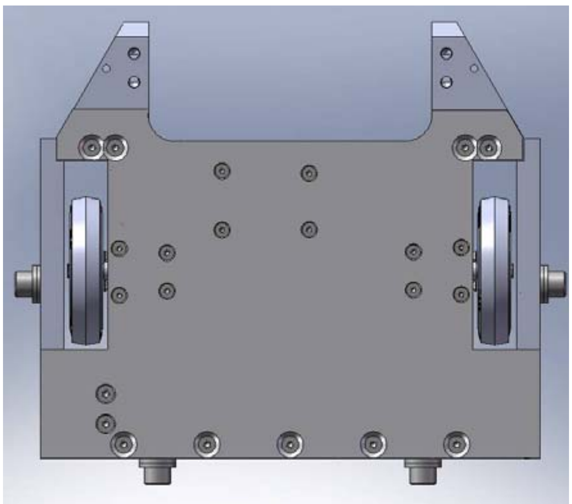


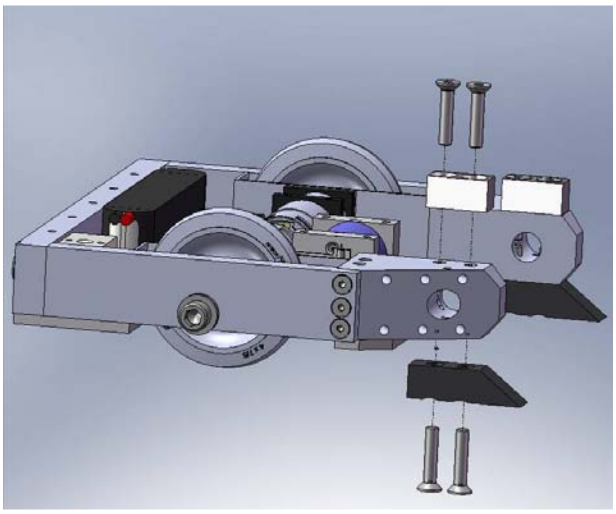


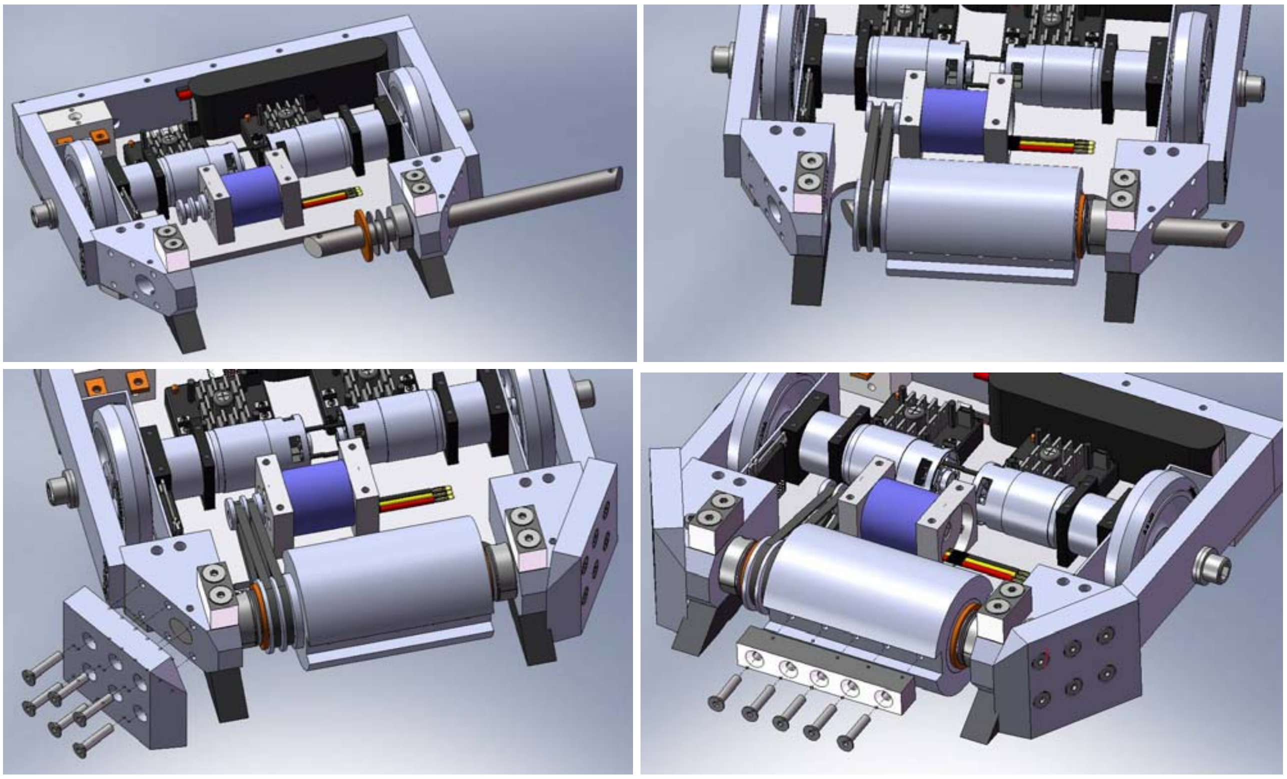


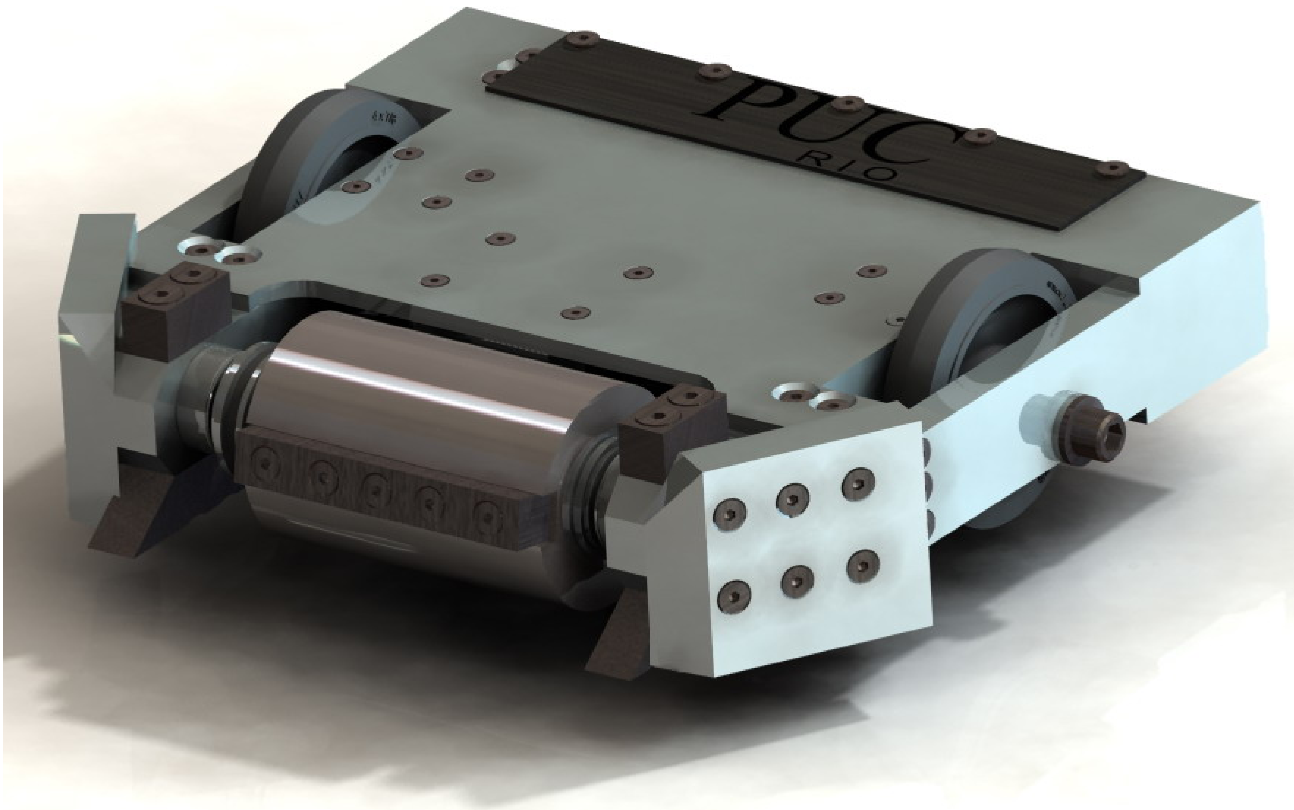










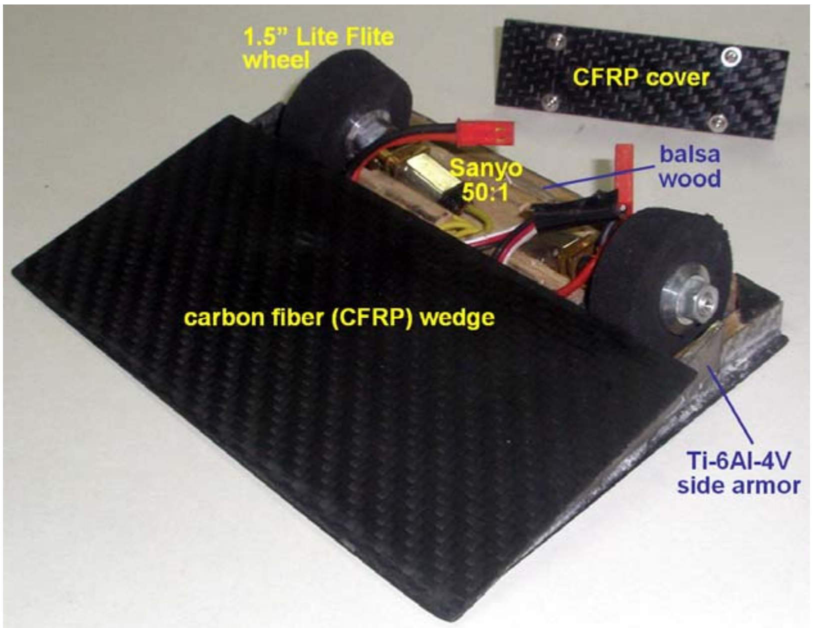

PUC
RIO

10.14. Pocket

A couple of months before RoboGames 2008, we decided to build our first **fairyweight** (150 grams). In its original design, Pocket Touro was supposed to be a drumbot. Its design was extremely compact to avoid going over the 150 gram weight limit. But this made it impossible to find commercial DC motor speed controllers that could fit inside the robot. Increasing the chassis would surely make it go over the weight limit. We then aimed to develop our own speed controller, small enough to fit inside our vaporbot.

However, the building of Touro Feather was demanding most of our spare time in the University. We finally realized we wouldn't be able to develop the speed controller in time for RoboGames 2008. So, one week before the competition, we completely changed our design, from drumbot to wedge. Since the robot would not be a drumbot anymore, we dropped the "Touro" from its name, calling it simply Pocket.

The fairyweight wedge Pocket (pictured to the right) was built from two carbon fiber (CFRP) rectangles, joined together in a V-shape using four triangular pieces of balsa wood and some Gorilla glue. A strip of CFRP was used as the robot's rear wall, glued as well to the wood. Two triangular pieces of 1/16" thick titanium grade 5 sheet were glued to the outer wooden triangles to work as side armor, as pictured to the right.



Pocket used two Sanyo 50:1 micro-geared motors to drive two 1.5" diameter rubber foam Lite Flite wheels. The motors were controlled by two Banebots BB-3-9 speed controllers, powered by a 250mAh 2S (7.4V) LiPo battery and connected to a 75MHz nano-receiver. The drive system was held in place with the aid of a CFRP cover, pictured above.

Pocket didn't do well at RoboGames 2008. It had a hard time getting under other fairyweights, because its carbon fiber wedge was not sharp enough. Anyway, it was a great learning experience. Building such a light robot is an extremely challenging task. Every part must be carefully planned and weighed, even the wires and the amount of solder used in the electronics.

Conclusions

I hope this tutorial is useful for the entire combat robot community, as well as for other builders of competitive robots in general. A great thing about such competitions is that they promote hands-on learning and exchange of information among the teams, bringing together people with completely different backgrounds. This is the spirit behind this tutorial. We would appreciate receiving any corrections, suggestions and contributions, through the "RioBotz Combot Tutorial" RFL Forum post, so that the text can evolve into next versions. Every contribution will be acknowledged. This is just version 2.0, we'll always try to go deeper into the most interesting subjects and keep the content updated with recent technology advances.

Try not. Do, or do not. There is no try.

FAQ - Frequently Asked Questions

The questions below were taken from the Forum at www.robocore.net

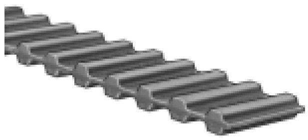
Design Fundamentals

I would like to know the best way to build a very basic combat robot made out of Lego.

You'll probably be able to build at most a hobbyweight or a lighter robot. You would need to bond the pieces with professional (24 hour) epoxy, otherwise the robot would fall apart after the first impact. Even so, the plastic pieces would not resist the tool steel and titanium weapons that most of the teams use nowadays, you would need some metallic armor. As for the electronics, you would need to adapt the RCX or NXT control module to a radio-control system. You'll probably need more powerful motors than the ones from Lego, for that it would be necessary to add power electronics to amplify the outputs of the RCX/NXT. The VEX system, used in the FIRST competition, is similar to Lego, but its parts are made out of metal and joined by screws, and it already comes with radio and receiver. But, for instance, its wheels and gears are made out of plastic, which would be a weak point.

How can I make a tank tread?

There are basically two types of tank treads: the ones used in toys (made out of rubber) and the ones from war tanks (made out of metal). They are completely different. Don't use the toy ones, they fall off easily, and the rubber tread would stretch out too much or even rupture when applying high torques. A possible solution is to use timing belt pulleys as driving wheels and double-sided timing belts (pictured to the right) as the tread. Several timing belts have steel wire reinforcement, which would prevent the stretching issue. The treads used in war tanks are made out of several articulated steel parts, requiring a lot of machining knowledge to build.



They are expensive and, even so, a powerful spinner could easily knock them off. The best bet is to use wheels. If you want more traction, use 6 wheels, with all-wheel drive, using a system of pulleys or sprockets, or even use 6 motors for redundancy.

What are the advantages of using a rubber tire as armor?

A rubber tire creates a good protection against blunt spinners, working as a damper, but not against very sharp blades. There are not many advantages against other types of robots. Wedges usually get underneath tire-robots very easily. Vertical spinners and drumbots have a greater advantage, because their weapons will grip better onto the tire to fling it high. If you intend to use the tire as both armor and structure, then install some metallic protection layer between the tire and the robot's interior, to help shield it against perforating weapons.

What is the maximum weight of a weapon so that the robot does not get too sluggish?
Follow the 30-30-25-15 rule discussed in chapter 2. The total weight of the weapon, including its motor and accessories, should not be much above nor below 30% of the robot's weight. This is not an exact rule because it depends on the robot type (for instance, 35% might be a good value for drumbots), but certainly 10% of the weight would be too little, and more than 50% would be too much.

Is there any limitation regarding the number of weapons that a robot can have?
No, but it is probably better to use only one. Unless the weapons work together on the opponent, at the same time, as it was discussed in chapter 2. Or if the secondary weapon is a wedge.

I wanted to know if there's any middleweight with dimensions 2m × 2m × 2m (6.6ft × 6.6ft × 6.6ft, the size limit for middleweights in Brazilian events) or anything close to that.
No middleweight would get even close to such size, it would be very fragile. A hollow cube made out of aluminum with those dimensions and wall thickness of only 1mm would have 67kg (148lb)! And this would happen without any parts inside it. The robots need to be compact so that their structure and armor can be thick.

Which is the best software to draw robots?
The most used in the US in combat robotics are probably Solidworks and Rhyno3D. The 3D modeling capabilities of those programs are better and easier to use than in Autocad.

Motors and Transmissions

I'd like to know how other teams build such fast robots with such small motors. Which motor do you recommend to power my robot's drive system?
Watch out for the main indicators of motor performance: the ratio between the maximum power and the motor weight is one of them, and the ratio between I_{stall} and I_{no_load} is another. Compare these values with the ones from the motors listed in chapter 5. Depending on the motor, it is possible to double the input voltage, multiplying the power by 4. But there's a chance of overheating, so limit such overvolting to tests that take no longer than 3 minutes, and check if the speed controller can take so much current (especially if the motor stalls during a match). The S28 series Magmotors are an exception, they're already optimized for 24V, so use them at most at 36V, not 48V, unless you have a current-limiting system. Test a lot, and always keep spare motors.

Does anybody know a mechanical trick to increase the power output of a motor?
There is no mechanical trick, the first law of thermodynamics, which deals with the conservation of energy, doesn't allow that. What you can do is to align or lubricate your transmission system to reduce friction, reducing power losses. But there is no way to mechanically increase its power unless you provide more power, for instance, by increasing the input voltage of the motor.

Are the Bosch motors bi-directional? Their datasheet says they only work in one sense.
All permanent magnet DC motors work in both senses without problems, it is enough to invert the input connections. However, these motors usually have advanced timing, where the permanent magnets of the stator are rotated with respect to the brushes, turning faster in one sense than in the other (see chapter 5). In a few cases, the speed difference is so large that the manufacturer recommends that it is used only in the faster direction. You can reverse them, but they will be slower.

What do you need to power up a very high speed spinner bar from a middleweight?
You need a powerful motor, preferably using NiCd batteries instead of NiMH. SLA batteries could be a good option for heavier classes. Lithium batteries such as A123 would be the best choice, however the cost is higher. The stored energy depends on the bar, but typical speeds for middleweight bar spinners can go up to 3,000RPM, such as in our robot Titan.

Which motor can I use to drive a hobbyweight?
The gearmotors from Pittman and Buehler are good and inexpensive choices, if bought second-hand. Power drill motors are also a great choice, especially for the weapon system. Disassemble an 18V drill and you'll get a motor, gearbox, battery and charger for a relatively low cost. Stock motors such as the 540 and 550 series are also a great choice, but you'll need to gear them down.

What does it mean to have a servomotor with 4 kg×cm torque?
This servo is able to bear, for instance, the weight of a 4kg mass hanging 1cm away from the output shaft. Or a 2kg with a leverage arm of 2cm.

I want to know how to use a step motor.
A step motor is a brushless electric motor that can divide a full rotation into a large number of steps. The motor's position can be controlled precisely, with the resolution of one angular step, without the need for position sensors. Hence, they are a good option for open-loop position control. But they are not the best option for speed control, which is what you usually need in combat. Their torque and power are relatively low if compared to a DC motor of same weight. Most electronic systems that power step motors do not tolerate the high currents required in combat. Almost all the electric motors used in combat are either brushed or brushless permanent magnet DC motors.

Can anybody send me a scheme/drawing of how to build a lifting arm for a hobbyweight, using an electric motor instead of a pneumatic cylinder?
To build an electric lifter, you can use a rack and pinion system coupled to an electric motor to convert rotary motion into linear movement. A few gearmotors have an endless screw that drives a cursor, translating it forward and back. You can also use a 4-bar mechanism, such as the one from BioHazard or Ziggy, see chapters 5 and 6. This mechanism could be powered by either linear or rotary electric motors.

Electronics

With the relays I'm using, my motors can only move forward.
Use an H-bridge. It can be implemented, for each motor, using 4 relays with single contacts or with 2 relays with double contacts. The scheme is in chapter 7. However, this "bang-bang" control is not the best option. Try to implement PWM (chapter 7), it will be easier to drive the robot and it will make the motors and electronics last longer.

How can you control a 12V motor with a 5V receiver?
You need to use a speed controller, see chapter 7.

Which speed controller is the best?
Chapter 7 describes OSMC and Victors, they are probably one of the best options available. For smaller robots (hobbyweights or under), the Scorpion XL and HX are also good options. Building your own battle-proof speed controller is very challenging, but you'll learn a lot in the process.

Does the Victor controller brake the motor?
Victor has a jumper to choose between coast (not braking) or brake. In the brake mode, Victor shorts the motor leads, turning them into generators, which will then dissipate energy from its internal resistance in the form of heat.

Does hot glue affect the RC boards?
No, hot glue guns are a great option to protect your electronics against shorts and to avoid loose contacts and screws in the electric system. Avoid using hot glue on components that need to dissipate heat, such as FETs.

Batteries

Which is the perfect and cheap combination for high power batteries? What type of batteries should I use?
Especially for middleweights, perhaps the best solution is to use NiCd batteries. The cheap solution is to use SLA batteries (AGM or gel). You can get high currents from SLA's, but for them to last the entire match you will end up adding a lot of weight to the robot. Good quality SLA's might be a good option for the heaviest robots such as super-heavyweights, especially due to cost issues. Unfortunately there is no powerful, cheap and light battery, you have to choose two of them: powerful and cheap (high capacity SLA's), light and cheap (low capacity SLA's), or powerful and light (NiCd, NiMH or lithium).

In a few technical datasheets it is written "discharge rate: up to 3C." What does the "C" stand for?

Such 3C means that the battery tolerates, without problems, a discharge current in A equal to 3 times its measured capacity (C, then the name 3C) measured in A·h, as explained in chapter 8. If your battery has 3.6A·h, then it tolerates a discharge current of 3 × 3.6 = 10.8A. This is the same as saying that it can be discharged in 1/3 of an hour (because 10.8A × 1/3 h = 3.6A·h). But combot matches don't last 20 minutes (1/3 hour), they usually last 3 or 2 minutes, therefore the discharge rates for use in combat should be at least 8C, a value that you can find in high discharge NiCd and in most lithium batteries. This means that the 8C battery could be discharged without significantly warming up in 1/8 hour = 7.5 minutes, which in practice allows you to fully discharge it in 3 minutes without overheating. The ideal discharge rate would be at least 20C, to be able to discharge the entire pack in 1/20 hours = 3 minutes.

Can I assemble my own battery pack?

Yes, see the chapter 8. But take care not to overheat the batteries when soldering them together, it is necessary to solder them quickly and with localized heat.

Can I use a 12V car battery with a 45A·h capacity in my middleweight? Is it true that lead-acid batteries are not allowed?

Considering that the de-rating factor (see chapter 8) of SLA batteries is about 0.28 for 3 minutes, your actual capacity would be 0.28 × 45 = 12.6A·h. To completely discharge such battery during a 3 minute match (1/20 hours), your motors would need to draw 12.6A·h / (1/20)h = 252A continuously. This is a lot of continuous current for a middleweight. Use a battery with lower capacity, which will save you a lot of weight. The car lead-acid batteries, which can spill the electrolyte if upside down, are forbidden in combat. They need to have an immobilized electrolyte, such as the gel or AGM types.

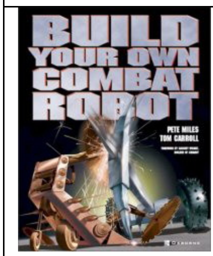
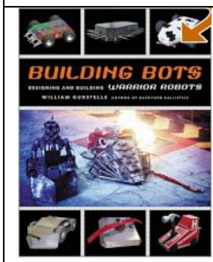
Could anybody write a tutorial on how to build a combat robot?

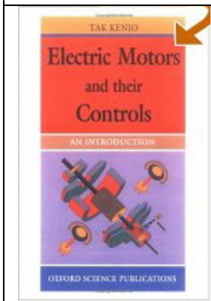
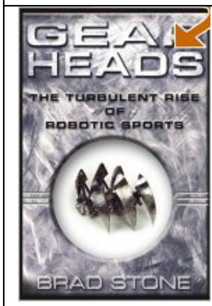
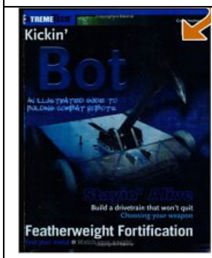
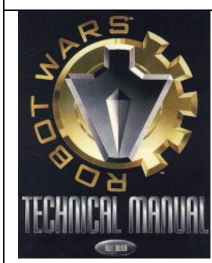
Here it is! I hope this tutorial has helped. Several other tutorials can be found on the internet as well, along with great build reports and other FAQ lists. There’s even a Combat Robot Wiki (http://combots.net/wiki). There are also several great forums for further research, such as:

- RFL Forum (http://forums.delphiforums.com/therfl);
- Antweight Forum (http://forums.delphiforums.com/antweights);
- BattleBots Forum (http://forums.delphiforums.com/BattleBot_Tech);
- RoboWars Australia Forum (www.robowars.org/forum); and
- RoboCore Forum (“Forum” link at www.robocore.net; most topics are in Portuguese).

Bibliography

All the books below are recommended readings if you plan to build a combat robot. A lot of the information in this tutorial was learned from them.

	**[1] Art of Electronics, The.** Horowitz, P., Hill, W., 1125 pages, Cambridge University Press, 1989. This is one of the most complete books on electronics, a must-read if you plan to develop your own electronic system.
	**[2] BattleBots, The Official Guide.** Clarkson, M., 272 pages, McGraw-Hill, 2002. This guide presents in an informative way the most famous robots from the BattleBots league, as well as their builders, in an almanac style. The pictures are very interesting. There are a few very basic building tips as well.
	**[3] Build Your Own Combat Robot.** Miles, P., Carroll, T., 416 pages, McGraw-Hill, 2002. Excellent guide about building combots, with great chapters such as the ones on batteries and another about the main tips to design each type of robot. It also deals with special topics such as autonomous and sumo robots.
	**[4] Building Bots: Designing and Building Warrior Robots. Gurstelle, W.**, 256 pages, Chicago Review Press, 2002. A good book for beginners, being able to teach even builders with little engineering background. It has good sections on materials, radio systems, and internal combustion engines. The book presents several basic physics equations to exemplify the concepts, making it very instructive. It also includes basic tips to organize an event.
	**[5] Combat Robots Complete: Everything You Need to Build, Compete, and Win.** Hannold, C., 311 pages, McGraw-Hill / TAB Electronics, 2002. With 22 chapters, this comprehensive book is able to approach several subjects in robot design. Naturally, since it covers so many subjects, a few of them are not presented with a very high depth. It includes excellent appendices, and it teaches step-by-step how to build specific robots from three different weight classes: antweight, featherweight, and heavyweight.

	**[6] Combat Robot Weapons. Hannold, C.**, 288 pages, McGraw-Hill / TAB Electronics, 2003. It's a follow up of the book Combat Robots Complete, it focuses on the main types of weapons and how to design them. It also presents great strategies related to each weapon type. Since it focuses on weapons, this book is able to go much deeper than the previous one from the same author.
	**[7] Electric Motors and their Controls: An Introduction.** Kenjo, T., 192 pages, Oxford University Press, 1991. Excellent introductory book on electric motors, it presents in a simple and didactic way the principles of operation of nearly all the existing types, and how to control them.
	**[8] Fatigue Under Service Loads.** Castro, J.T.P., Meggiolaro, M.A., 1200 pages, in Portuguese, to be published in 2009. It deals in depth with mechanical behavior of materials, presenting the main materials used in engineering, their properties, and how to select them depending on the application. It also deals with failure mechanisms such as fatigue, fracture, yield, plastic collapse, creep and corrosion, among others. Several sections from chapters 3 and 4 from this tutorial, as well as the appendices A, B and C, were adapted from this book.
	**[9] Gearheads: The Turbulent Rise of Robotic Sports.** Stone, B., 304 pages, Simon & Schuster, 2003. A great informative book about the history behind robot combat, from its beginnings.
	**[10] Kickin' Bot: An Illustrated Guide to Building Combat Robots.** Imahara, G., 528 pages, Wiley, 2003. An excellent combot building guide, it is one of the best combat robot books ever written. It thoroughly approaches several subjects from robot conception to combot events. It has great sections about building tools and how to efficiently use them. The pneumatics appendix is also a must-read.
	**[11] Robot Wars: Technical Manual.** Baker, A., 144 pages, Boxtree, 1998. It is a very good almanac with a lot of information on the Robot Wars league. Despite its title, it is not exactly a technical manual about robot building. The pictures are very interesting, and they reflect the success that robot combat has achieved in the United Kingdom.

Appendix A – Conversion among Brinell, Vickers and Rockwell A, B and C hardnesses

HB 3ton	HV	HRA 60kg	HRB 100kg	HRC 150kg		HB 3ton	HV	HRA 60kg	HRB 100kg	HRC 150kg
100	105	-	-	-		311	327	66.9	-	33.1
105	110	-	-	-		321	337	67.5	-	34.3
111	116	-	65.7	-		331	347	68.1	-	35.5
116	121	-	67.6	-		341	358	68.7	-	36.6
121	127	-	69.8	-		352	370	69.3	-	37.9
126	132	-	72	-		363	382	70	-	39.1
131	138	-	74	-		375	394	70.6	-	40.4
137	144	-	76.4	-		388	408	71.4	-	41.8
143	150	-	78.7	-		401	422	72	-	43.1
149	157	-	80.8	-		415	436	72.8	-	44.5
156	164	-	82.9	-		429	451	73.4	-	45.7
163	171	-	85	-		444	467	74.2	-	47.1
167	175	-	86	-		461	485	74.9	-	48.5
170	179	-	86.8	-		477	502	75.6	-	49.6
174	183	-	87.8	-		495	521	76.3	-	51
179	188	-	89	-		514	541	76.9	-	52.1
183	192	-	90	-		534	562	77.8	-	53.5
187	196	-	90.7	-		555	584	78.4	-	54.7
192	202	-	91.9	-		578	608	79.1	-	56
197	207	-	92.8	-		601	632	79.8	-	57.3
201	211	-	93.8	15		630	670	80.6	-	58.8
207	217	-	94.6	16		638	680	80.8	-	59.2
212	223	-	95.5	17		647	690	81.1	-	59.7
217	228	-	96.4	18		656	700	81.3	-	60.1
223	234	-	97.3	20		670	720	81.8	-	61
229	241	60.8	98.2	20.5		684	740	82.2	-	61.8
235	247	61.4	99	21.7		698	760	82.6	-	62.5
241	253	61.8	100	22.8		710	780	83	-	63.3
248	261	62.5	-	24.2		722	800	83.4	-	64
255	268	63	-	25.4		733	820	83.8	-	64.7
262	275	63.6	-	26.6		745	840	84.1	-	65.3
269	283	64.1	-	27.6		757	860	84.4	-	65.9
277	291	64.6	-	28.8		767	880	84.7	-	66.4
285	300	65.3	-	29.9		779	900	85	-	67
293	308	65.7	-	30.9		790	920	85.3	-	67.5
302	317	66.3	-	32.1		800	940	85.6	-	68

Appendix B – Material Data

Relative prices per weight of several materials, with respect to 1020 carbon steel.

cement	0.10-0.15	aluminum alloys	3.0-11.0
charcoal	0.15-0.20	natural rubber	3.1-3.2
burning oil	0.45-0.50	regular glass	3.2-3.3
gray cast iron	0.55-0.60	copper alloys	4.0-7.7
reinforced concrete	0.60-0.70	glass fiber (GFRP)	5.2-7.2
soft woods	0.90-1.6	polycarbonate	5.5-5.7
laminated 1020 steel	1.00	stainless steels	6.0-13.0
structural A36 steel	1.08	tool steels	6.0-31.0
press.vessel A515 steel	1.3	nylon	7.1-7.3
low alloy steels	1.0-3.0	acrylic (PMMA)	11.5-12.0
PVC	1.7-1.8	titanium alloys	22.0-130
zinc	2.0-3.8	copper-nickel alloys	27.0-35.0
UHMW	2.1-2.3	nickel superalloys	50+
alumina (Al_2O_3)	2.4-3.9	carbon fiber (CFRP)	200+

Prices in US$/kg for several metals (data from 1998, possibly outdated).

ASTM A36 (plate)	0.50-0.90	Nickel 200	19-25
ASTM A36 (bar)	1.15	Inconel 625	20-29
SAE 1020 (plate)	0.50-1.45	Monel 400	15-17
SAE 1040 (plate)	0.75-1.30	Haynes 25	85-104
SAE 4140 (bar)	1.75-1.95	Invar	17-20
SAE 4140H (bar)	2.85-3.05	Super Invar	22-33
SAE 4340 (bar)	2.45-3.30	Kovar	30-40
304 stainless (plate)	2.15-3.50	C11000 copper	4-7
316 stainless	3.00-6.20	C17200 (Be-Cu copper)	25-47
440A stainless (plate)	4.40-5.00	C26000, C36000 (plate)	3.20-4.85
17-7PH stainless (plate)	6.85-10.00	C71500 (Cu-Ni, 30%)	8.50-9.50
gray cast iron	1.20-3.30	C93200 (bar)	4.50-12.20
malleable cast iron	1.45-5.00	AZ31B (plate, extrusion)	8.80-11.00
Al 1100 (plate)	7.25-10.00	AZ91D (cast)	3.80
Al 2024 T3 (plate)	8.80-11.00	lead	1.20-2.70
Al 2024 T351 (bar)	11.35	solder 60Sn-40Pb	5.50-7.50
Al 6061 T6 (plate)	4.40-6.20	tin	6.85-8.85
Al 6061 T651 (bar)	6.10	zirconium 702 (plate)	44-49
Al 7075 T6 (plate)	9.00-9.70	tungsten (pure)	77-135
Al 356 (cast)	4.40-11.65	molybdenum (pure)	85-115
Ti ASTM grade 1 (pure)	28-65	silver	170-210
Ti 5Al 2.5V	90-130	tantalum (pure)	390-440
Ti 6Al 4V	55-130	gold	9,500-10,250
zinc (pure)	1.20-2.45	platinum	11,400-14,400

Typical values of the impact strength of structural materials.

material	$\mathcal{G}_{IC}$(kJ/m^2)	K_{IC}(MPa√m)
pure ductile metals	100-1000	100-450
ductile low carbon steels	100-300	140-250
high strength steels	10-150	45-175
titanium alloys	25-115	55-115
aluminum alloys	6-35	20-50
glass fiber (GFRP)	10-100	20-60
carbon fiber (CFRP)	5-30	32-45
wood, ⊥ to fibers	8-20	11-13
polypropylene (PP)	8	3
polyethylene (PE)	6-7	1-2
reinforced concrete	0.2-4	10-15
cast irons	0.2-3	6-20
wood, // to fibers	0.5-2	0.5-1
acrylic (PMMA)	0.3-0.4	0.9-1.4
granites	~0.1	1-3
Si_3N_4	0.1	4-5
cement	0.03	0.2
glass	0.01	0.7-0.8
ice	0.003	0.2

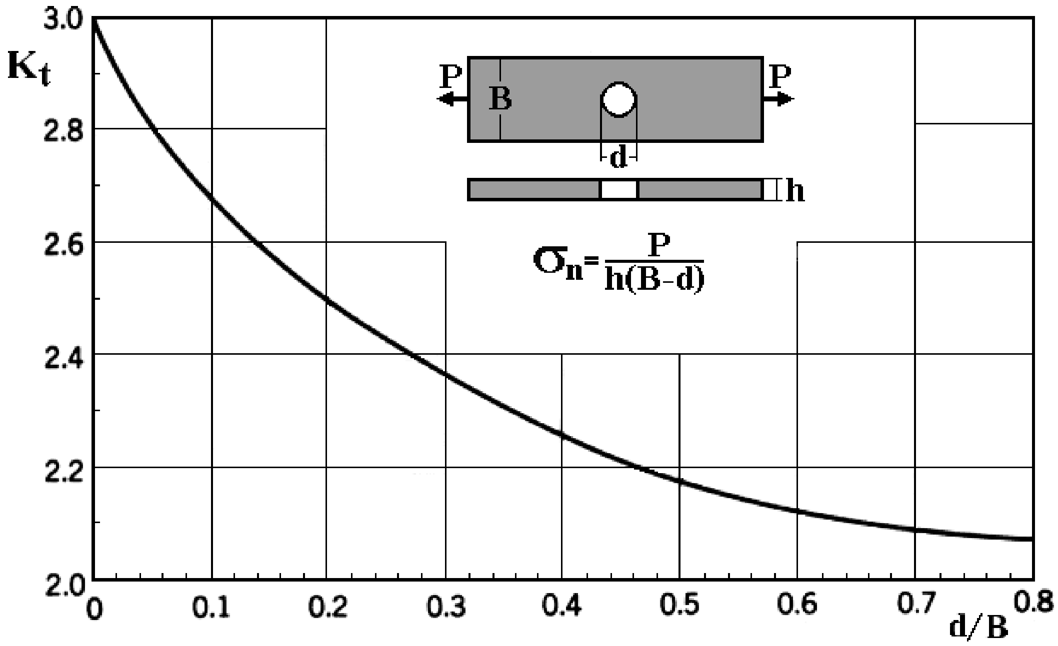
Values of **E**/ρ, **E**$^{1/2}$/ρ, **E**$^{1/3}$/ρ, **S**/ρ, **S**$^{2/3}$/ρ and **S**$^{1/2}$/ρ of a few materials, where **E** is the Young Modulus (in GPa), $\mathbf{S_u}$ is the rupture strength (in MPa), and ρ is the relative density.

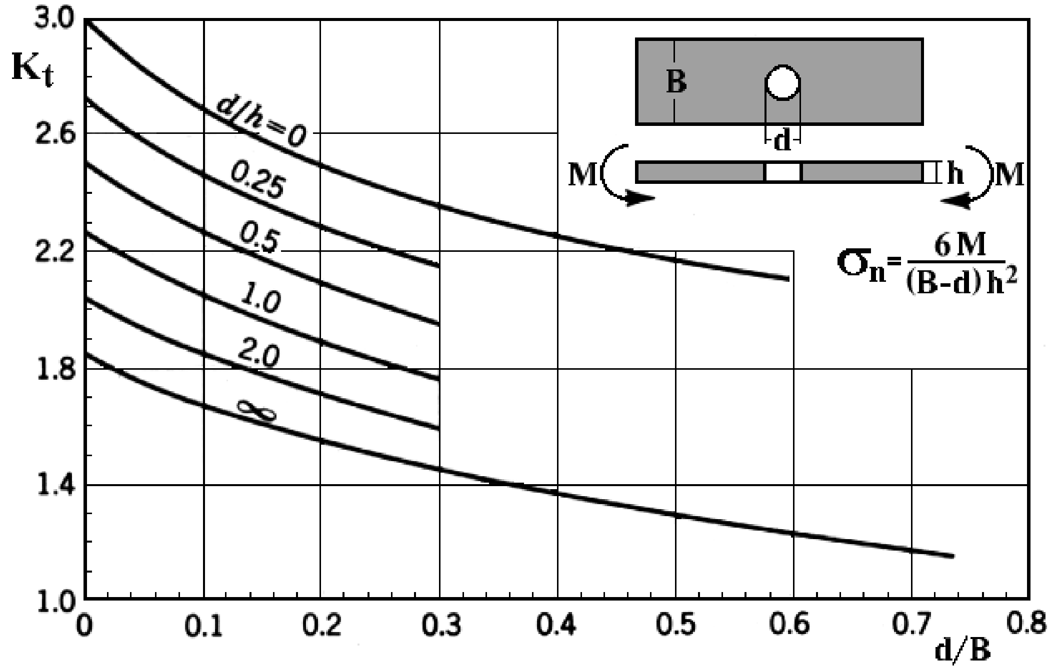
material	E/ρ	$E^{1/2}$/ρ	$E^{1/3}$/ρ	material	S_u/ρ	$S_u^{2/3}$/ρ	$S_u^{1/2}$/ρ
steels	26	1.8	0.8	**1020 steel**	56	7	2.7
				304 stainless	77	9	3.1
				4340 steel	184	16	4.8
				5160 steel	196	17	5.0
				S7 steel	251	20	5.7
				18Ni(350)	305	23	6.1
aluminum alloys	26	3.0	1.5	**Al 2024 T3**	174	22	7.9
				Al 6061 T6	115	17	6.5
				Al 7075 T6	196	24	8.4
titanium alloys	25	2.3	1.0	**Ti-6Al-4V**	224	22	7.1
magnesium alloys	25	3.7	2.0	**AZ31B-H24**	143	23	9.0
				ZK60A-T5	169	25	9.6
beryllium alloys	164	9.4	3.6	**Be S-200**	415	45	15
polycarbonate (PC)	2	1.3	1.1	**PC**	54	13	6.7
Delrin	2	1.3	1.0	**Delrin**	54	13	6.2
UHMW	0.7	0.9	0.9	**UHMW**	43	13	6.8

Appendix C – Stress Concentration Factor Graphs

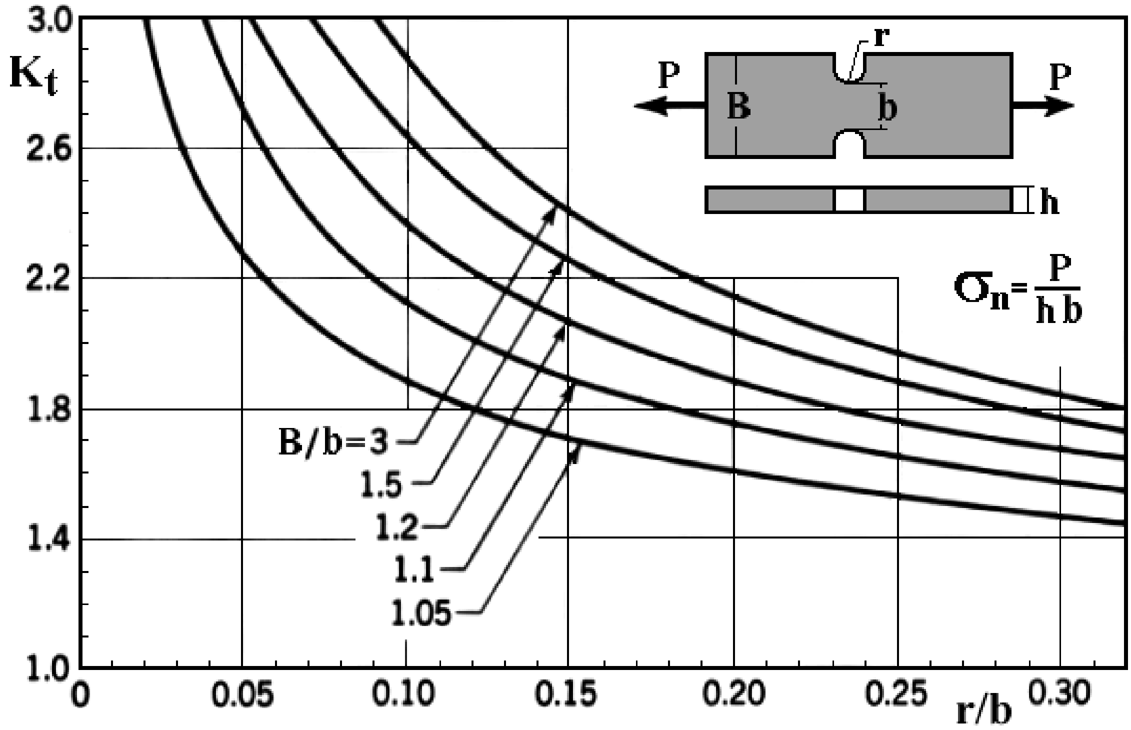
K_t is the ratio between the notch root stress σ_{max} and the nominal stress σ_n (or between the notch root shear stress τ_{max} and the nominal shear stress τ_n). The nominal stresses σ_n are defined in each graph. The notch root stresses are therefore $\sigma_{max} = K_t \cdot \sigma_n$, which can be used for design against yield, fatigue, etc.

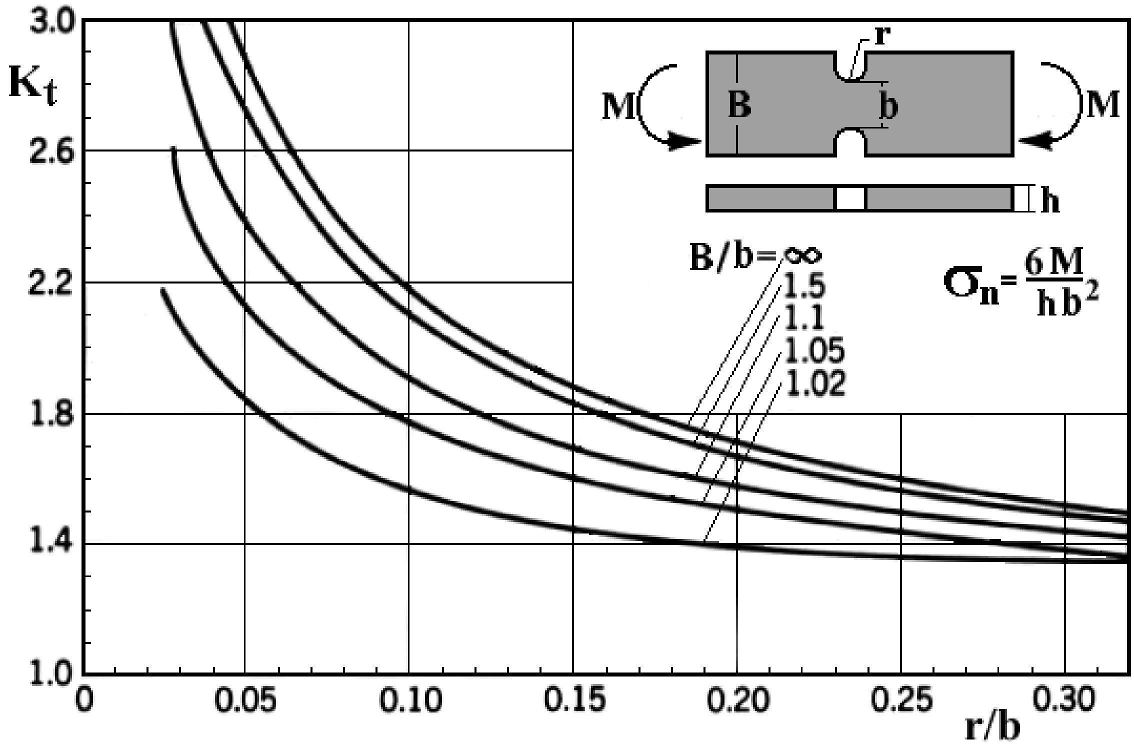
(i) holed plates subject to a traction force **P** or bending moment **M**:



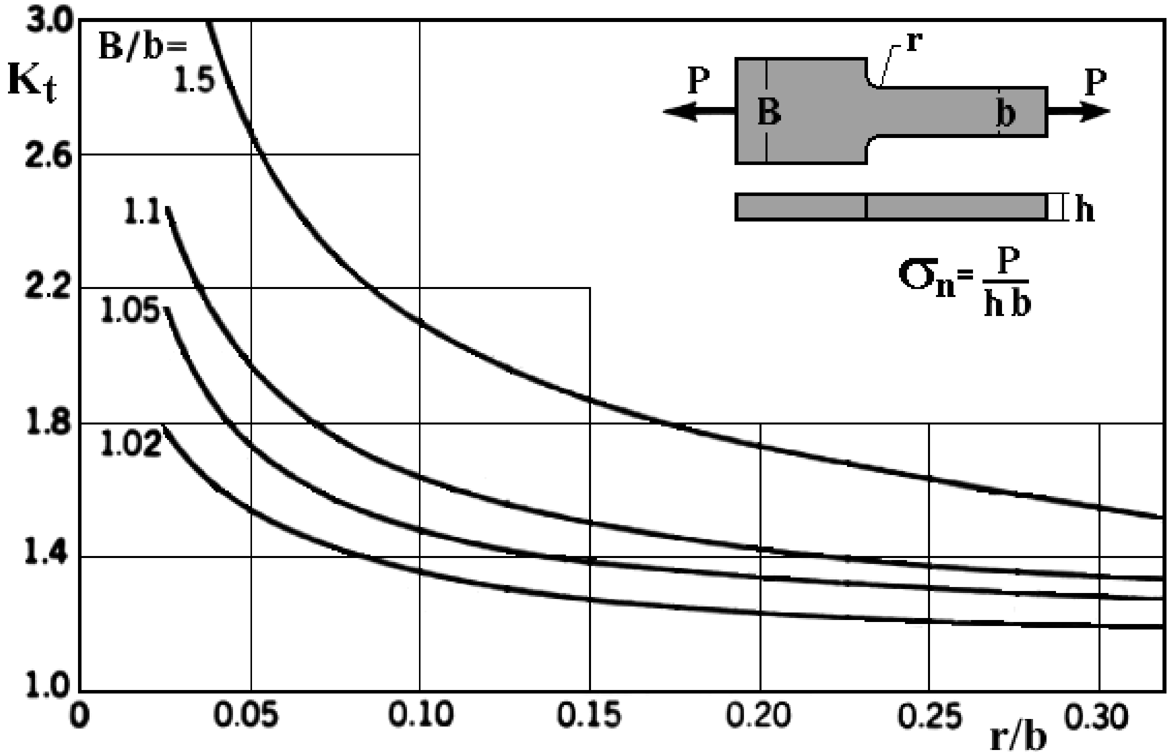


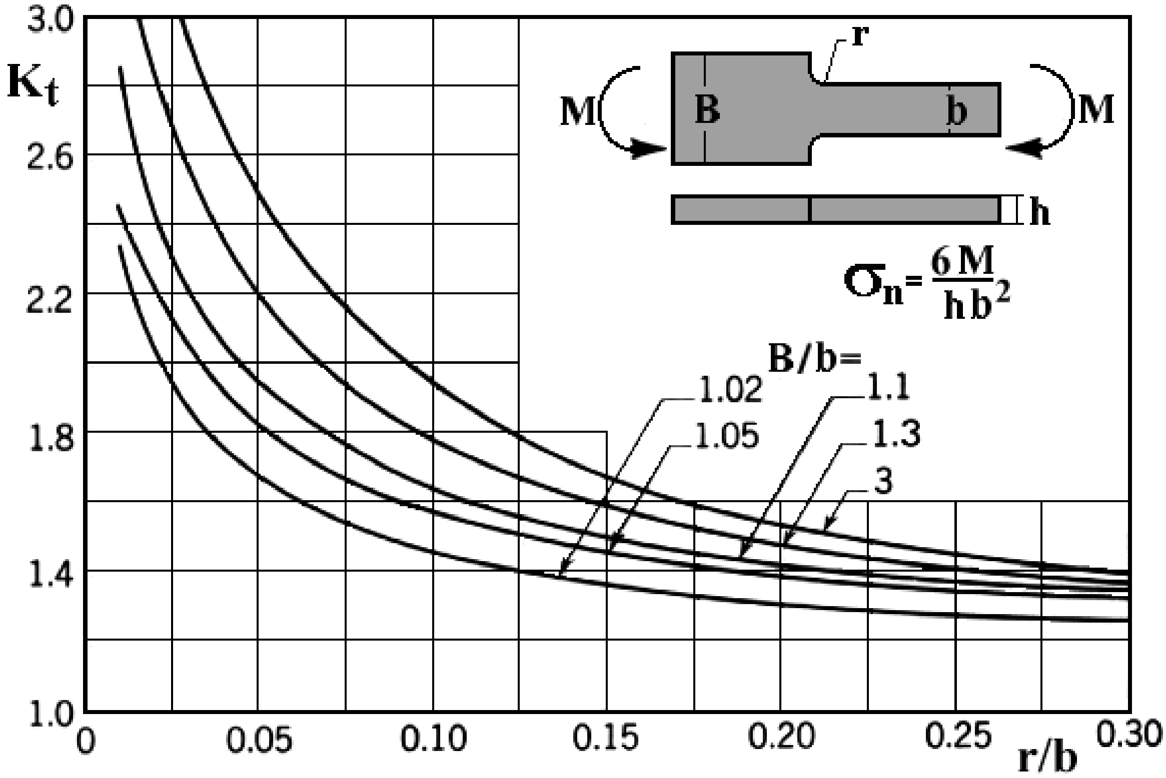
(ii) notched plates subject to a traction force **P** or bending moment **M**:



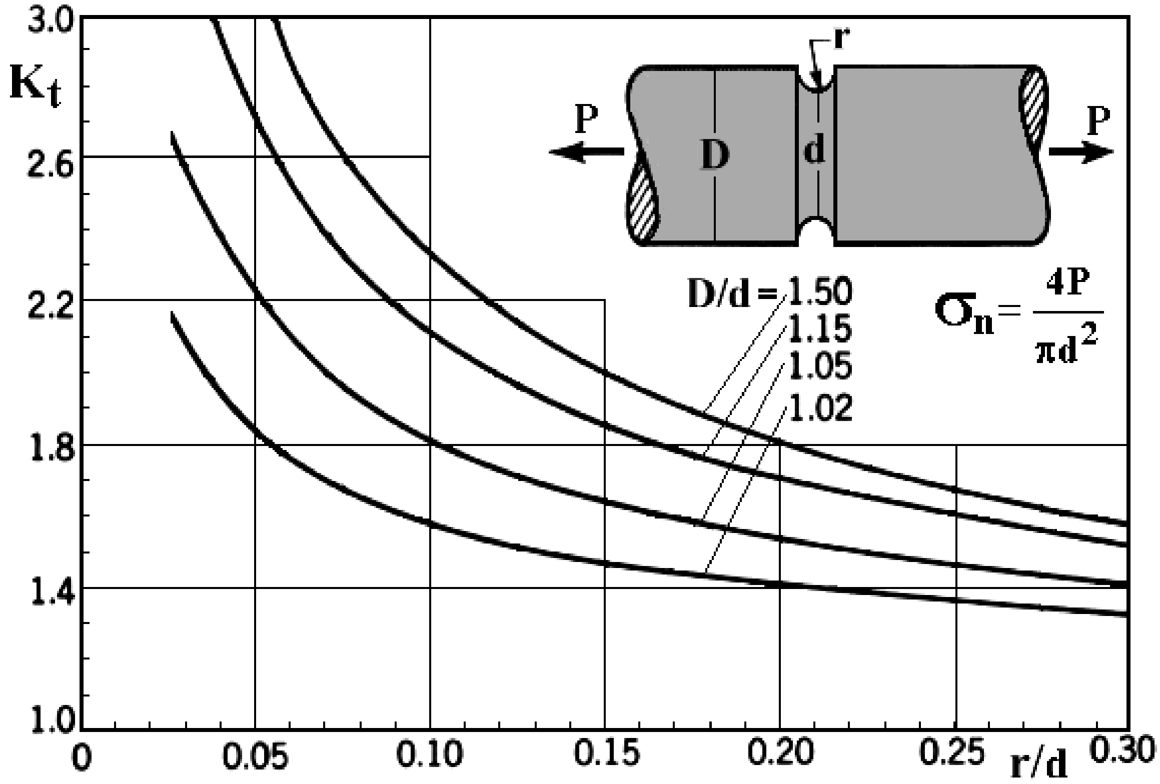


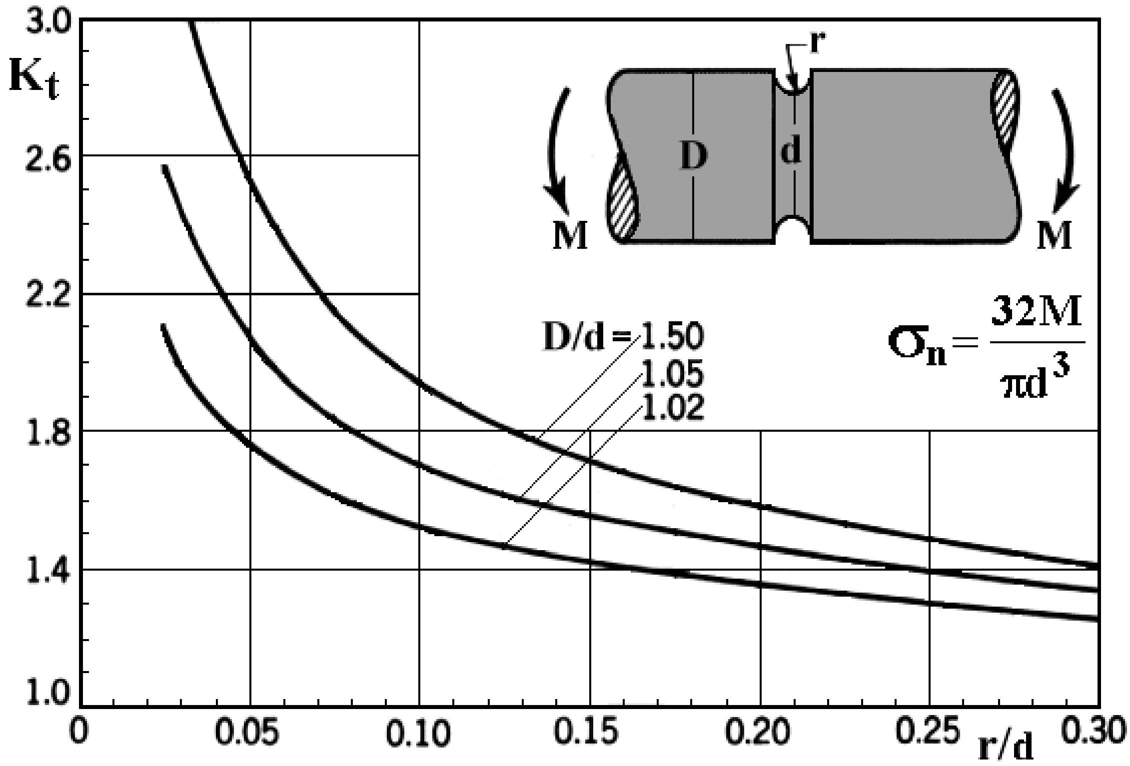
(iii) filleted plates subject to a traction force **P** or bending moment **M**:

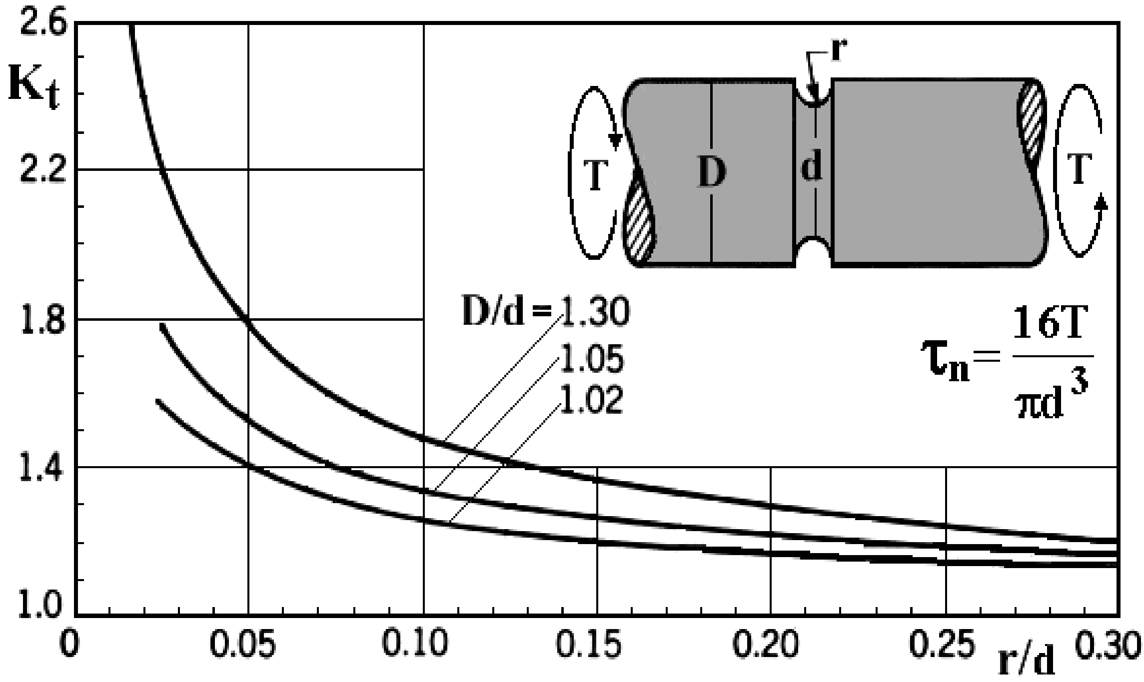




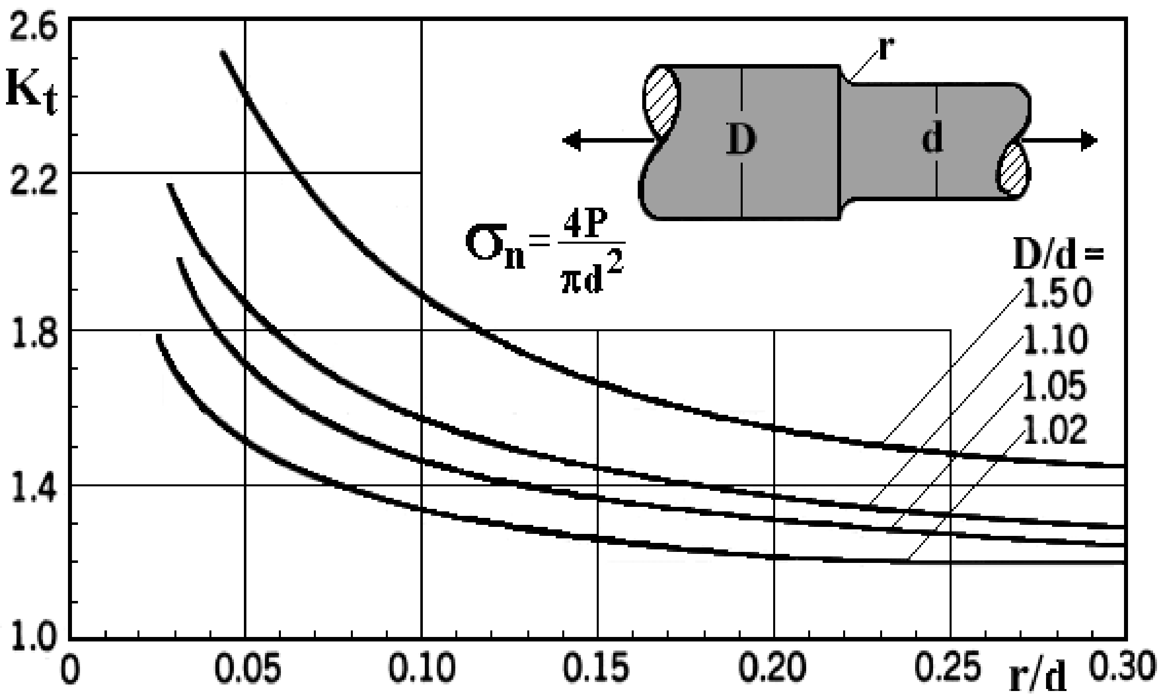
(iv) notched shafts subject to a traction force **P**, bending moment **M**, or torque **T**:

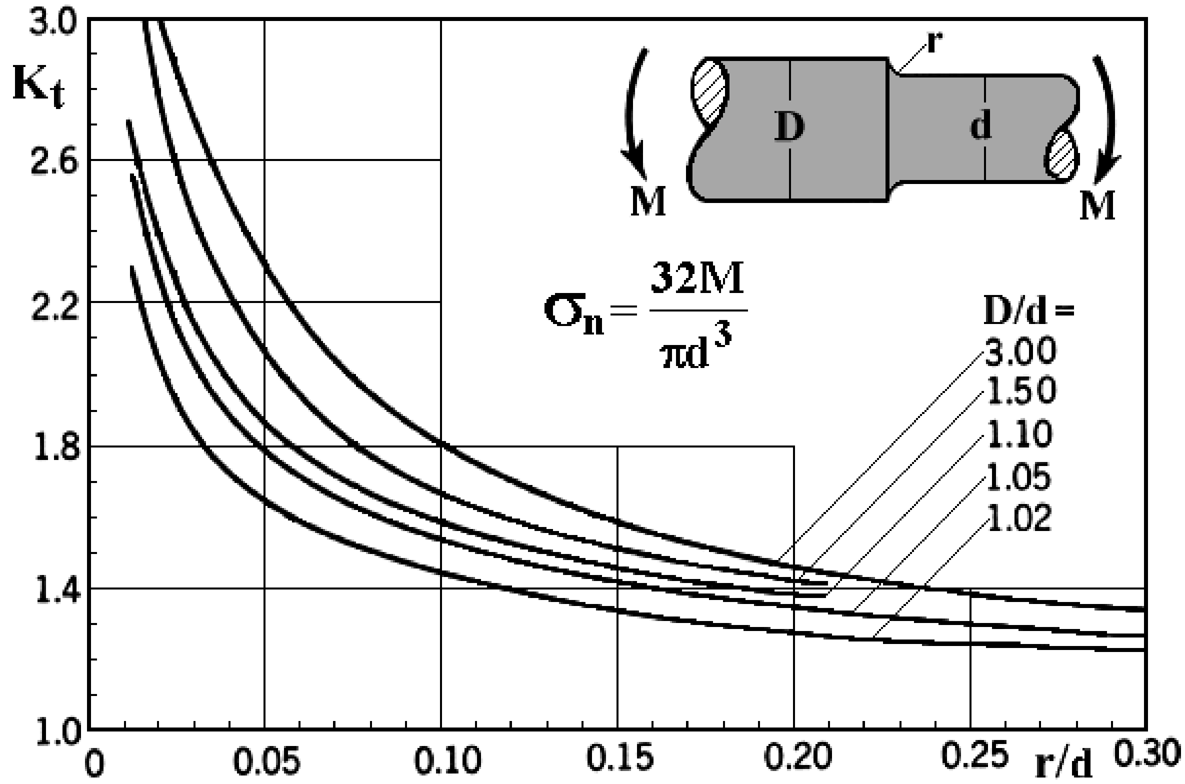




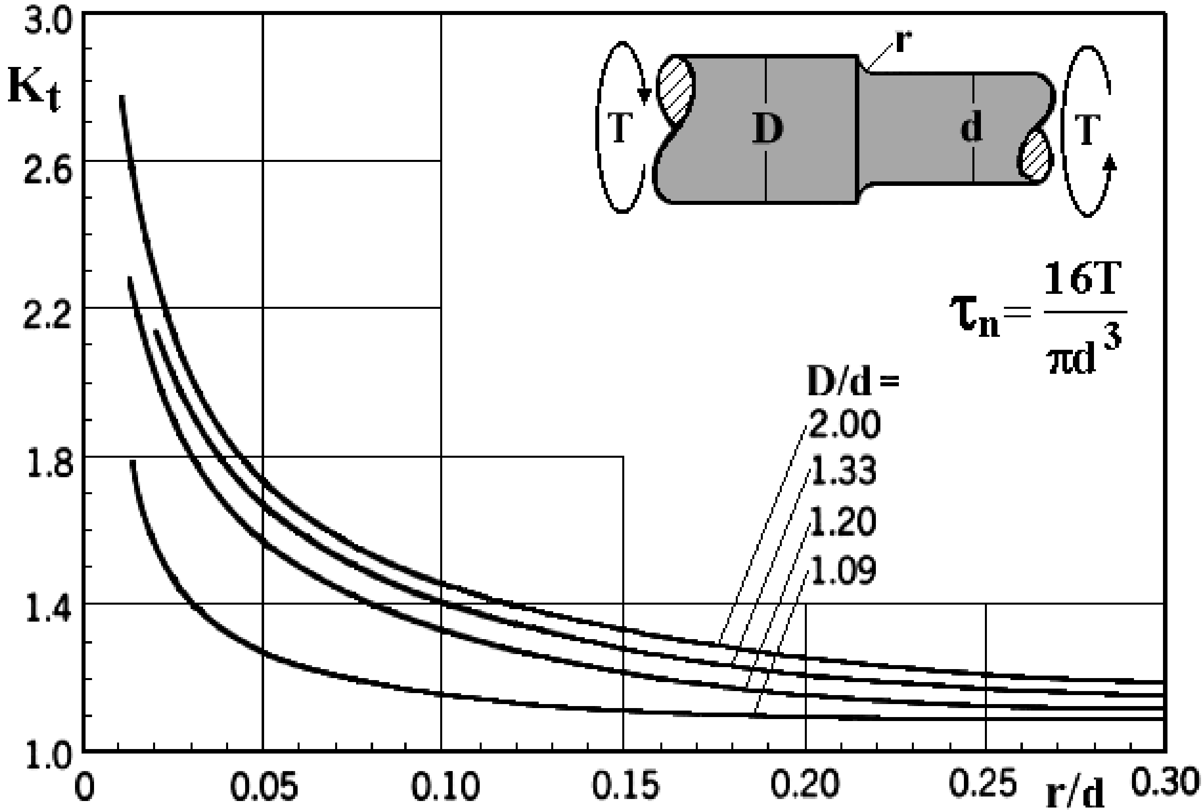


(v) filleted shafts (shoulders) subject to a traction force **P**, bending moment **M**, or torque **T**:



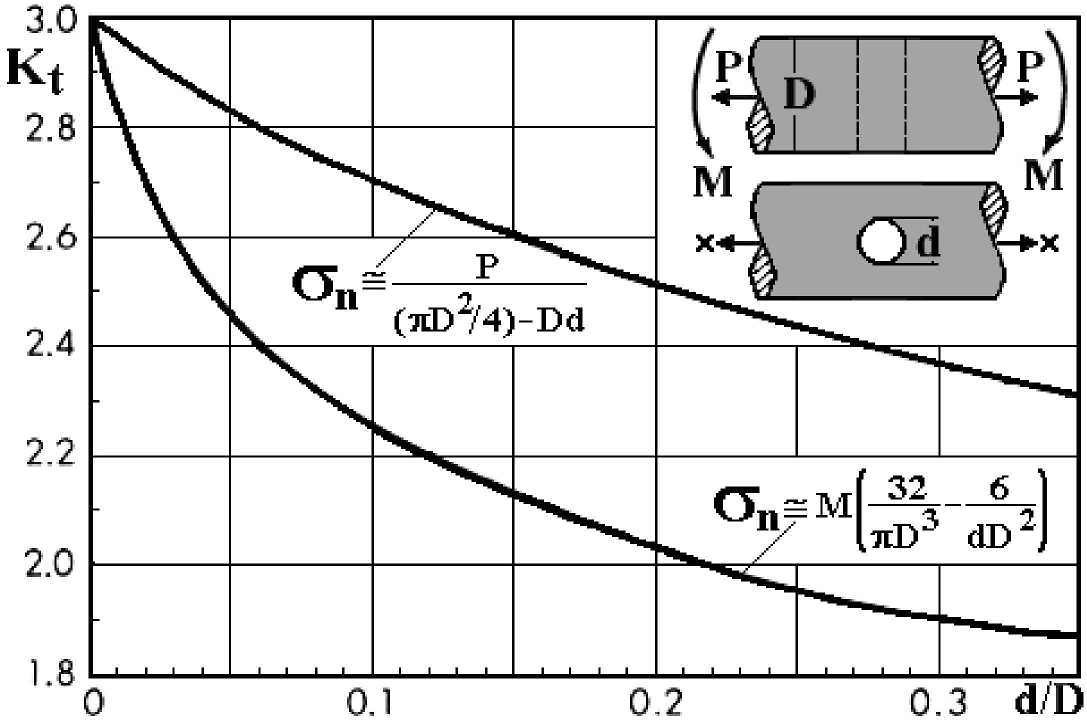


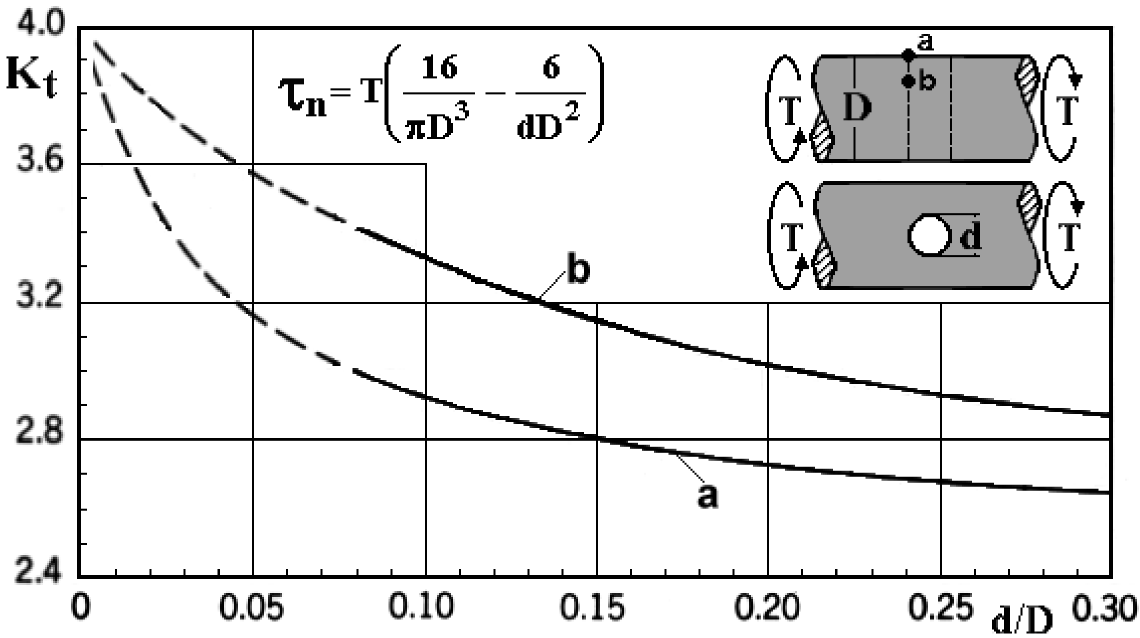
K_t
r
D
d
M
M
$\sigma_n = \frac{32M}{\pi d^3}$
D/d =
3.00
1.50
1.10
1.05
1.02
r/d



K_t
r
D
d
T
T
$\tau_n = \frac{16T}{\pi d^3}$
D/d =
2.00
1.33
1.20
1.09
r/d

(vi) holed shafts subject to a traction force **P**, bending moment **M**, or torque **T**:





Appendix D – Radio Control Channels and Frequencies

27 MHz band (air / cars / boats)
26.995 MHz - Channel 1. Brown
27.045 MHz - Channel 2. Red
27.095 MHz - Channel 3. Orange
27.145 MHz - Channel 4. Yellow
27.195 MHz - Channel 5. Green
27.255 MHz - Channel 6 - Blue

50 MHz band
(air / cars / boats)
50.800 MHz - Canal RC00
50.820 MHz - Canal RC01
50.840 MHz - Canal RC02
50.860 MHz - Canal RC03
50.880 MHz - Canal RC04
50.900 MHz - Canal RC05
50.920 MHz - Canal RC06
50.940 MHz - Canal RC07
50.960 MHz - Canal RC08
50.980 MHz - Canal RC09

72 MHz band
(air only)
72.010 MHz - Channel 11
72.030 MHz - Channel 12
72.050 MHz - Channel 13
72.070 MHz - Channel 14
72.090 MHz - Channel 15
72.110 MHz - Channel 16
72.130 MHz - Channel 17
72.150 MHz - Channel 18
72.170 MHz - Channel 19
72.190 MHz - Channel 20
72.210 MHz - Channel 21
72.230 MHz - Channel 22
72.250 MHz - Channel 23
72.270 MHz - Channel 24
72.290 MHz - Channel 25

72 MHz band
(continued)

72.310 MHz - Channel 26
72.330 MHz - Channel 27
72.350 MHz - Channel 28
72.370 MHz - Channel 29
72.390 MHz - Channel 30
72.410 MHz - Channel 31
72.430 MHz - Channel 32
72.450 MHz - Channel 33
72.470 MHz - Channel 34
72.490 MHz - Channel 35
72.510 MHz - Channel 36
72.530 MHz - Channel 37
72.550 MHz - Channel 38
72.570 MHz - Channel 39
72.590 MHz - Channel 40
72.610 MHz - Channel 41
72.630 MHz - Channel 42
72.650 MHz - Channel 43
72.670 MHz - Channel 44
72.690 MHz - Channel 45
72.710 MHz - Channel 46
72.730 MHz - Channel 47
72.750 MHz - Channel 48
72.770 MHz - Channel 49
72.790 MHz - Channel 50
72.810 MHz - Channel 51
72.830 MHz - Channel 52
72.850 MHz - Channel 53
72.870 MHz - Channel 54
72.890 MHz - Channel 55
72.910 MHz - Channel 56
72.930 MHz - Channel 57
72.950 MHz - Channel 58
72.970 MHz - Channel 59
72.990 MHz - Channel 60

75 MHz band (cars /
boats)

75.410 MHz - Channel 61
75.430 MHz - Channel 62
75.450 MHz - Channel 63
75.470 MHz - Channel 64
75.490 MHz - Channel 65
75.510 MHz - Channel 66
75.530 MHz - Channel 67
75.550 MHz - Channel 68
75.570 MHz - Channel 69
75.590 MHz - Channel 70
75.610 MHz - Channel 71
75.630 MHz - Channel 72
75.650 MHz - Channel 73
75.670 MHz - Channel 74
75.690 MHz - Channel 75
75.710 MHz - Channel 76
75.730 MHz - Channel 77
75.750 MHz - Channel 78
75.770 MHz - Channel 79
75.790 MHz - Channel 80
75.810 MHz - Channel 81
75.830 MHz - Channel 82
75.850 MHz - Channel 83
75.870 MHz - Channel 84
75.890 MHz - Channel 85
75.910 MHz - Channel 86
75.930 MHz - Channel 87
75.950 MHz - Channel 88
75.970 MHz - Channel 89
75.990 MHz - Channel 90

700MHz band

2.4GHz band