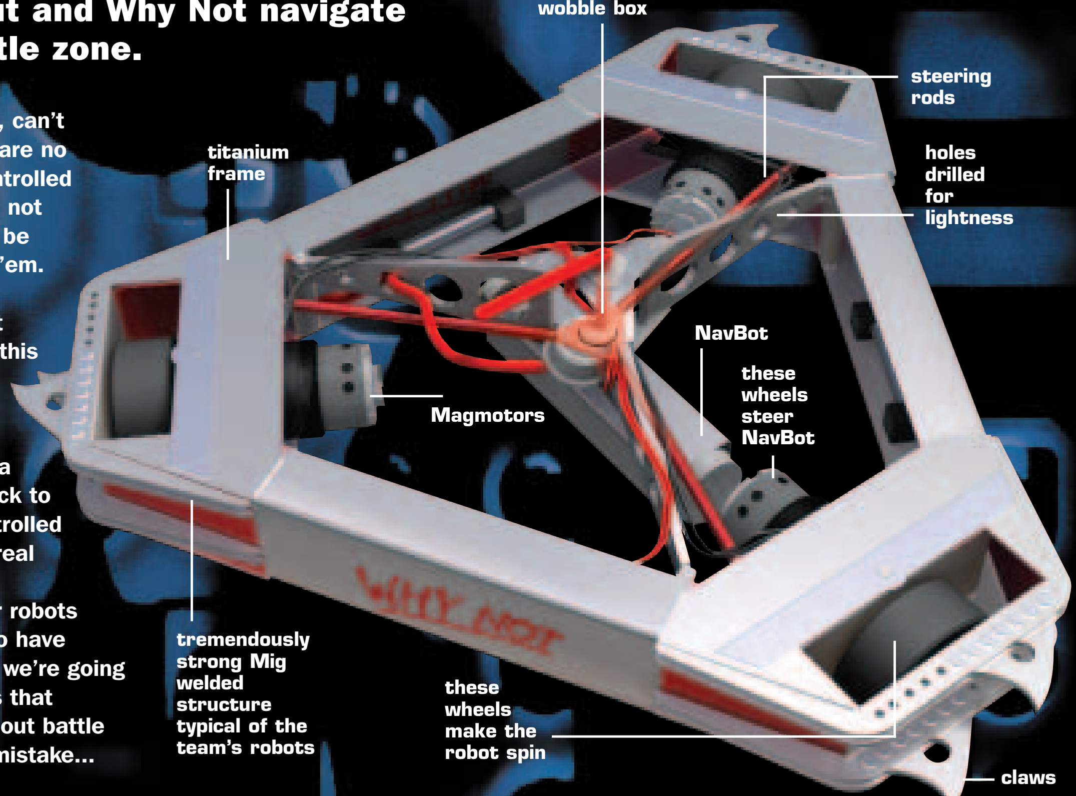
NavBot GUIDANCE

Finding a means of steering his spinners presented mechanical engineer Terry Ewert with a great technical challenge.

Meet the ghost in the machine: a tiny bot that helps Y-Pout and Why Not navigate through the battle zone.



They've no intelligence, can't think for themselves, are no more than remote-controlled vehicles. As such, they're not real robots and shouldn't be included in a mag' about 'em.

This is an objection sometimes raised against putting fighting robots in this magazine. But is it fair?

LET'S TAKE A HIKE!

Well, strictly speaking, if a robot has to have feedback to be real, then remote-controlled fighting machines aren't real robots. (Though there are competitions specially for robots that aren't RC and that do have feedback!) Today though, we're going to meet two US machines that demonstrate why leaving out battle bots would be a big, big mistake...

TRYANGULATION

Y-Pout and Why Not are virtually identical but designed for two different weight classes – Heavyweight and Middleweight, respectively. They're among six of the Ewert clan's entries for BattleBots (the others being: Son of Whyachi, Warrior, Red Square and YU812). And they're triangular.

The machines' designer Terry Ewert likes triangles (he used the same shape for the spinners on Whyachi and SOW). "They're inherently strong," he explains, "self-supporting."

And that's just as well, because the main difference between these robots and Whyachi and SOW is that here the whole machines spin (at around 1000rpm): each robot is in itself the weapon, and each takes an absolute battering!

TECH SPEC

Why Not

Type:	Body spinner, with NavBot steering
Weight:	120lb
Dimensions:	44in diameter
Power:	Three 4 inch Mag motors at 38V
Weapon:	130mph claws

ULTIMATE DESTINATION

But it's the way Y-Pout and Why Not move, not their shape, that's totally revolutionary. Instead of using basic skid steer such as on spinners SOW (Issue 39 Workshop) or Mauler 51-50 (Issues 43 and 44), Terry has devised a new system.

"We wanted to create the ultimate spinner and for that you need lots of horsepower combined with perimeter weighting," he explains. "I wanted to make virtually all the robot spin – not to have just a spinning weapon. But then we needed a way of steering it."

FOREST OF CONFUSION

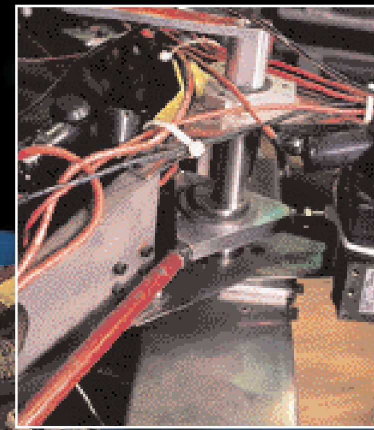
If you're having trouble seeing why you can't steer such a machine in the conventional way, imagine a spinning top (the robot), spinning on the spot. You want to move it (steer it) across the floor, so you give it a shove with your finger and it moves. That's fine here, but in the arena you need some means of applying a sideways (lateral) force from within the bot itself. Trouble is: if it's resting on its wheels to spin, you can't use them in the conventional way to guide it.

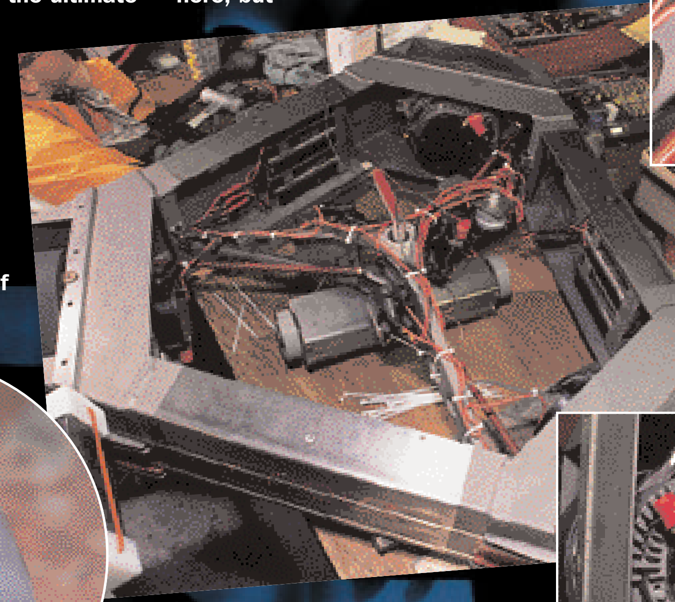
MEETING THE GHOST

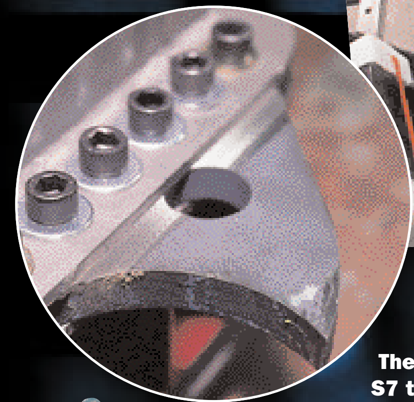
"The answer was to use a NavBot," says Terry, "a small robot weighing a fraction of the overall machine (15lb). And this is it," he says, pointing out a rectangular box with two wheels sitting at the heart of Why Not.

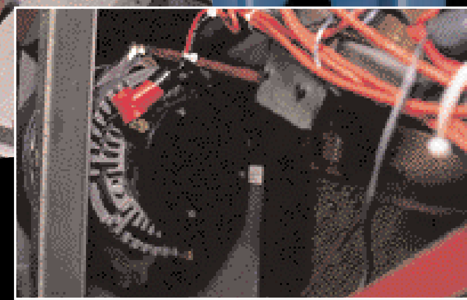
Terry explains that the NavBot receives signals from his transmitter and is steered around the arena by means of a motor-powered gearbox from within.

"See that red pointer? "Which ever way that's pointing, that's the way the robot translates."









Here's Y-Pout with the cover off (*left*): it's steered by a NavBot (*above inset*), powered by three 12hp Etek motors (*below inset*) and has S7 tool steel claws (*inset below left*).

Y-Pout's frame (*above*) is made from welded titanium. The lid (removed) is steel.

The claws are made from hardened S7 tool steel and are set at different heights for max damage.

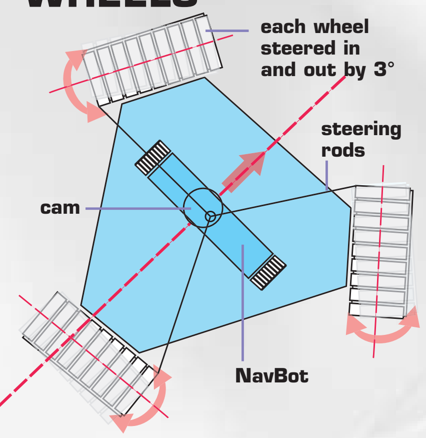
Need To Know

WOBBLY WHEELS

Fast-spinning Y-Pout and Why Not both use the same steering mechanism. It's driven by a small rectangular robot at the centre of the machines – a NavBot.

CAM DOES IT

Whether the big robots are changing direction or not, a cam at the centre (in a wobble box) causes each of the three wheels to be steered in and out by 3° on each rev. To change direction, the NavBot changes the point (on the floor) where this happens.



Hmm okay, but what makes it follow the pointer? Clearly it doesn't just shove directly against the insides of the fast-spinning triangle and push it about. Apart from not having the power, it'd get smashed to pieces! So what's happening?

STRAIGHT AND NARROW

"The NavBot is connected to a wobble box."

Er, hang on. Now we've got a NavBot, and a wobble box?

"Right. The wobble box has a cam inside that pushes these rods in and out 15 times a second."

(Terry indicates three red rods running from the centre of the machine to the three corners.)

"They're connected to the wheel assemblies, so that the wheels themselves are steered out by 3° and back in again by 3° for every revolution of the bot."

Wow. This is tough. It sounds ingenious. (No doubt it is ingenious!) But it's a bit hard to understand. Terry, help!

COMPASS BEARINGS

"By steering each wheel in succession out on the East side and then back in again on the West side of the bot, the whole bot will translate to the East. In this case the NavBot would be pointing East."

Ah right! So the NavBot changes the position – the spot relative to the arena floor, or to N, S, E or W – where the wheels begin to steer out and then back in. (See Inside View.)

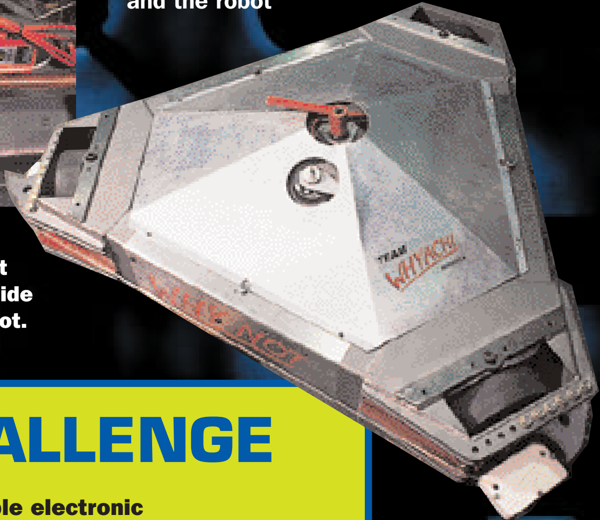
"Right."

But what if you wanted to stay on the same spot, wouldn't the pointer always be pointing in some direction and the robot wanting to go that way?

"Right. So to have zero translation, you have to keep the pointer going in a circle."

Then the robot is spinning fast and at the same time, tracing a small circle – virtually on the spot. "Exactly."

Here's that all-important red pointer – this time on Why Not (*right*) that Terry uses as a guide to steer the NavBot and big bot.



TAKING A SHOT AT IT

Most of the robots here on Treasure Island have some unique design features; all have features borrowed and adapted from elsewhere. That's what 99% of engineering is about – which is fine (after all, why re-invent the wheel?) But Y-Pout and Why Not both work on a principle that is, to the best of our knowledge, totally new. So how did Terry dream it up?

"I started with three sides to make the robot strong and to put the maximum weight out in the perimeter for more energy storage. From this ideal concept came the problem of solving a way to translate. I finally hit on the method of steering the wheels when I was up in my tree stand, deerhunting."

So, like many inventions, necessity was the mother of this one. But did Terry know it would work?

"I had a pretty good idea it would, but I made a full size model to try it. With the model on the floor, each of the wheels were steered manually while the bot was spun very slowly to check for linear translation."

WE'VE ARRIVED!

There you have it: an exciting new steering principle for a spinning robot – be it remote controlled or really REAL (and we're proud to debut it)! But that's not the end of the story. With 120lb of bot (in the case of Why Not – 220lb in the case of Y-Pout) spinning at 1000rpm; those wheels flicking in and out 15 times a second, tremendous torques (turning forces) are put on the NavBot's wheels. To keep itself going in the right direction, the NavBot has to fight back, and we'll find out what's inside it that makes it able to do that, next time.

TRANSLATION

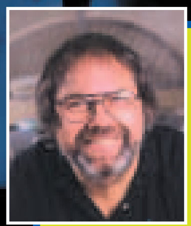
To understand how the robots move, watch the one green wheel here. To steer East, Terry turns the NavBot pointer that way. This makes each wheel steer out (by 3°) when it gets to the far right, and back in (by 3°) when on the opposite side of the robot. The net result: it heads East.

TECH SPEC

Y-Pout

Type:	Body spinner, with NavBot steering
Weight:	220lb
Dimensions:	55in diameter
Power:	Three 12hp Briggs and Stratton Etek motors
Weapon:	147mph claws

REX'S ROBOT CHALLENGE



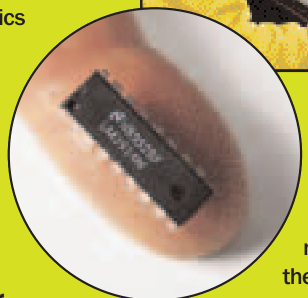
Over the next few issues I'm going to build a simple electronic circuit for indicating battery charge levels. The idea is to show you what's now achievable in 'home' electronics.

Microchips are now the standard building blocks for all sorts of electronic circuits. Each chip comprises an amazing number of components on a tiny integrated circuit (IC). What this means is that with just a chip and a few extra components, you can build a simple but functionally highly sophisticated circuit, for very little expense.

I'll show you how I start with a chip, make an acetate mask, etch the board and build it.

CHIP FUNCTION

Take a look through an electronics catalogue and you'll see chips dedicated to all sorts of jobs: voice record and playback, voice modulation, timing, and temperature sensing, for example. The one I have here is a dot/bar display driver. It can be used to light a row of LEDs to create a display similar to the ones showing record/playback levels on audio equipment. I'm going to use it to make an indicator for showing battery charge levels on Cassius Junior.



The chip I'm using is of the analogue (rather than logic) type – a dot/bar display driver.

USEFUL DATA

For each chip you can download (or print on-line) a manufacturer's data sheet, giving information about the chip's performance, it's limits and so on. Some info is useful – to know the chip can handle inputs of between 0-25 volts, for instance. But most of it you needn't worry about. Although, it's worth looking at the datasheet, you'll find the most helpful info (circuit diagrams, say) comes from the electronics enthusiasts community – particularly over the internet.

Here's the completed circuit (*left*) with a chip, three conventional and one variable resistors, a capacitor, and LEDs in three colours. It can be used to show you when you need to re-charge all your batteries (on Cassius Junior, say).

Coming Next: Look inside the NavBots, and more from Rex's Robot Challenge.