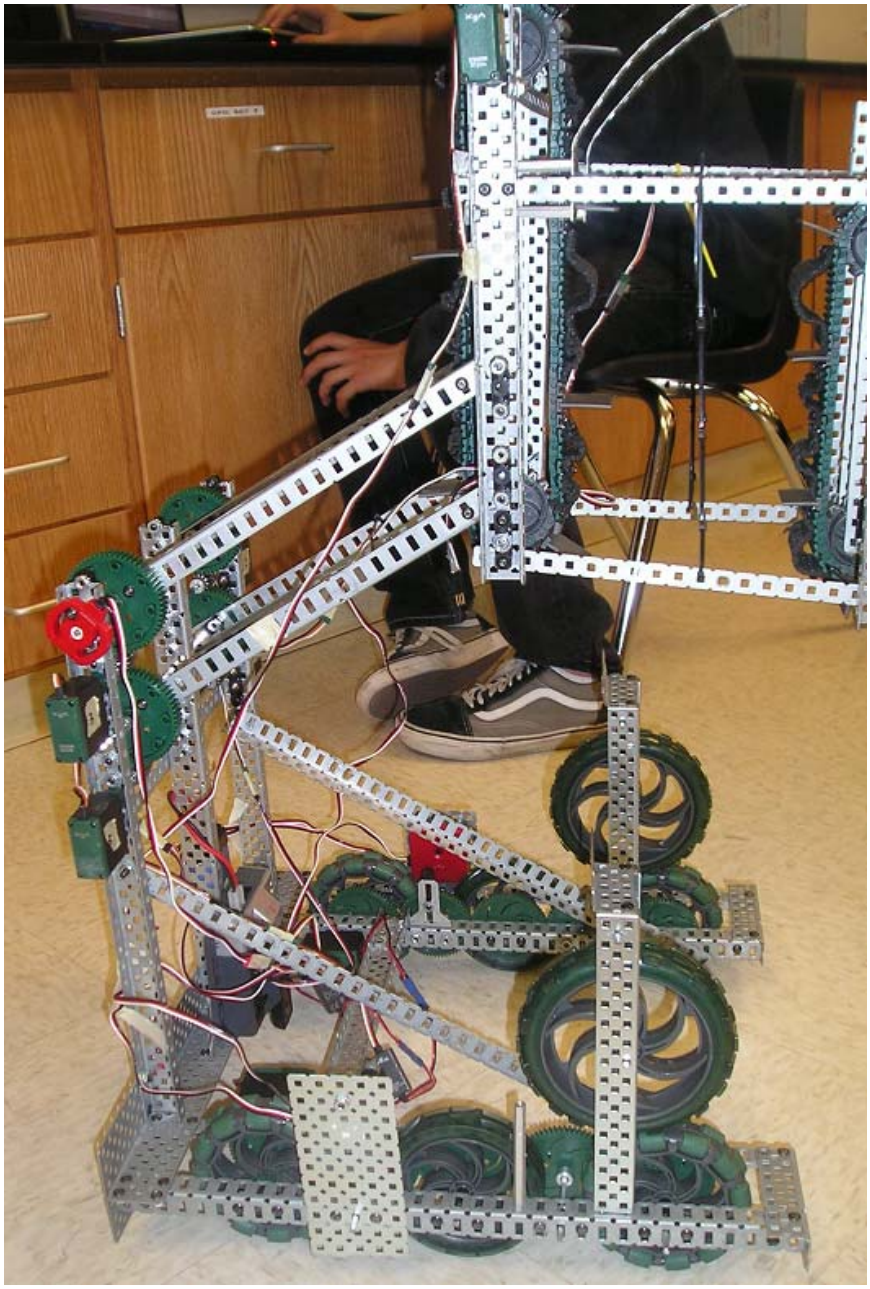
How to Hold Up an Arm/How to Problem Solve

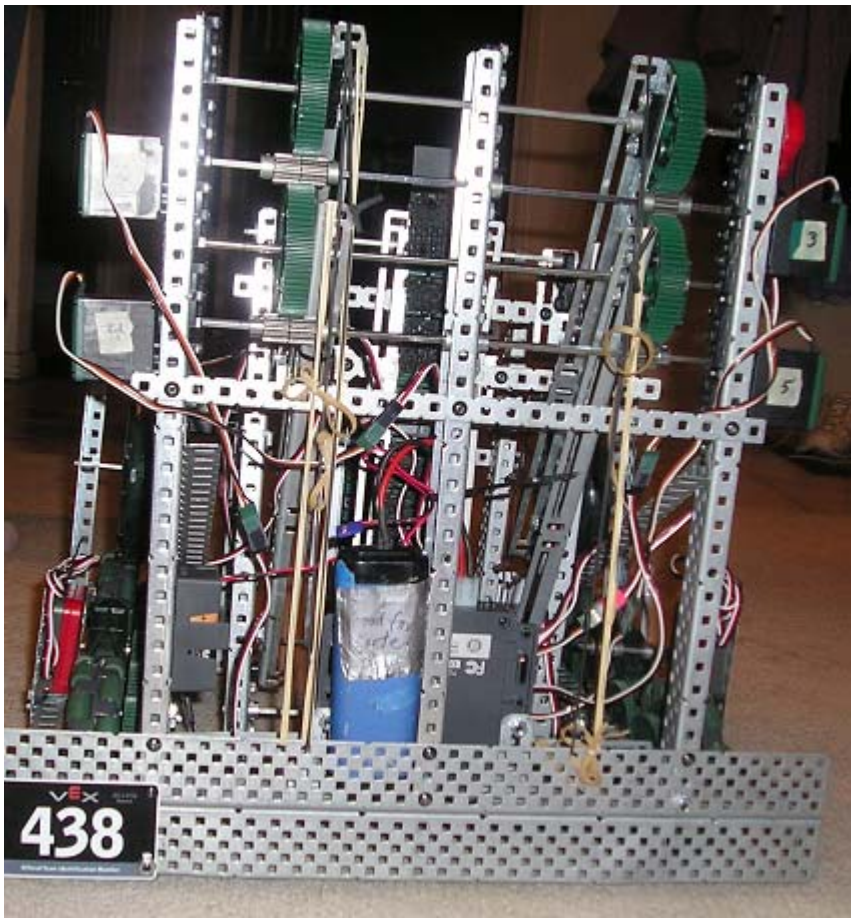
by Team Metal Gear, VRC 438

For the 2011 game, we built a 4-bar lift which lifted to 30". Using 4 motors, the arm lifted, but dropped under its own weight when the "up" button was released.



We wanted the arm to stay up without input, rather than forcing the driver to continually push the "up" button. Our mentor showed us how to search for solutions on vexforum.com (create an account, then use the "search" tab) and found a thread called "Having a motor HOLD its position."

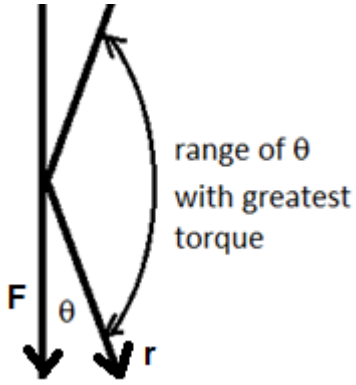
Two common suggestions in the thread were elastics and PID control. Elastics reduced the dropping, but over time they stretched out and were less effective.



We needed a better solution. PID control was popular, but we didn't understand the calculus, so we searched for pre-written code on the Vex forum under the "code" tab. Using a downloaded program, the arm twitched and moved without driver input. The documentation said to "tweak" the gain values for the P, I, and D components, but we couldn't make it work by trial and error, so we looked for a simpler solution, one we could understand.

The following post suggested outputting a constant level of power (the mean of maximum and minimum values).

We tried this with one adjustment. We learned that torque = r f sin θ, where r = radius (in the direction of the arm), F = force (downward) , θ = angle between r and F. So the greatest torque occurs when the arm is in the middle of its range. To prevent the arm from pushing too high or too low from constant power, we decided stop motor power when the angle of the arm (measured by potentiometer) was outside the range with highest torque.



The following program (EasyCv4) worked well to keep the arm from falling during Operator Control:

OperatorControl 1

```
#include "Main.h"
void OperatorControl ( unsigned long ulTime )
{
      int LeftBoost = 80;
      int RightBoost = -80;
      while ( 1 )
      {
            Potentiometer = GetAnalogInput ( 1 ) ; // read arm height (angle)
            DownButtonState = GetJoystickDigital ( 1 , 6 , 1 ) ; // Is either button press¬
            ed?
            UpButtonState = GetJoystickDigital ( 1 , 6 , 2 ) ;
            if ( DownButtonState == 1 ) // down button pushed - map button to drive down n¬
            ormally
            {
                  basebeltdrive ( ) ;
                  liftdrive ( ) ;
            }
            else if ( UpButtonState == 1 ) // up button pushed - map button to drive up no¬
            rmally
            {
                  basebeltdrive ( ) ;
                  liftdrive ( ) ;
            }
            else // no button pushed -- needs "boost"
            {
                  basebeltdrive ( ) ;
                  if ( Potentiometer > 600 ) // no boost if above upper threshold
                  {
                        liftstop ( ) ;
                  }
                  else if ( Potentiometer < 80 ) // no boost if below lower threshold
                  {
                        liftstop ( ) ;
                  }
                  else // boost if arm is in the middle
                  {
                        liftboost ( RightBoost, LeftBoost ) ;
                  }
            }
      }
}
```

In summary:

<u>How to hold up an arm:</u>

1. If possible, design your arm pivot point closer to the center of the bar for better balance.
2. Use elastics to counteract the downward force due to gravity.
3. Programming a low level of power to the arm motors when the arm isn't being driven may be enough to hold up the arm. Don't apply power when the arm is close to vertical or near the point of damage
4. PID control may be needed if it's important to hold up the arm at the upper and lower ends of the arm's range without overshooting.

<u>How to problem-solve</u>

1. It pays to follow the advice of your mentor (sometimes).
2. Search the Vex forum (vexforum.com) for solutions, including code examples. If you need to, post a question.
3. Try simple solutions first.
4. Try solutions you can understand.