The Low-Down on Low-Level Control

Here are, in outline form, the key issues you need to think about and toy with to stop being irritated by the low-level control of our scout via Java. Hope this helps!

1. OS Overhead/Hardware

Processor speed setting (set it to High)
Virtual memory (we've got it on, but it may play a role)
Java memory size - defaults to 1 MB, bigger could theoretically help (to a point)

2. Threads

When Java is multi-threaded, that's when the guarantees*really* disappear.
Because: there is no starvation control in Java.

3. Robot Side

It is conceivable that communicating too quickly with the robot can cause the 68332 to thrash. It's happened to some other model robots (Pioneer by RWI) before, and famously so. In our case, the true communication rate you can achieve with the Scout, come what may, is about 8 Hertz maximum. You may want to experiment, especially in your Turn-to commands and elsewhere where there is a tight loop, sleeping for about 50 or 100 milliseconds per cycle. If your code works with a Satellite Pro and fails with aTecra, this is probably why.

4. Memory

You all know that you can cause a garbage collect to happen using System.gc().

But there's more. Runtime.getRuntime().freeMemory() will tell you how much heap remains. This number starts out around 700,000 or 800,000 (1 MB to be precise, before all allocation). It goes down slowly and when it reaches about 30,000, that's when the system garbage collector kicks in. I wrote some ugly code, and it takes about 40 right turns to kick the garbage collector into action. It's *not* an incremental garbage collect. That is, when the system callsgc, then your 40,000 or 30,000 of space turns right back into 700,000 or so.

Now, using freeMemory, you can tell when a garbage collect happens! This is cool and useful, and it will also help you determine if the reason you're overshooting is because of garbage collection or not. Basically, keep track of the return value of freeMemory. It decreases monotonically. If it increases, then you know that GC just happened! Nifty, eh? I wrote my robot code so that every time there's a garbage collection, the speech synthesizer says so. It's very rare, believe it or not.

Another memory item: printing out takes memory. Try printing out the result of freeMemory. It shrinks every time you print! (Which is okay, because it shrinks quite slowly, even if you print out high volumes).