

SHK gesucht

Optimierung einer Tomographieanlage

Die Computertomographie (CT) – den meisten aus der Medizin bekannt – hat sich in den letzten Jahren zu einem etablierten Verfahren bei der Prozess- und Materialcharakterisierung entwickelt. Das Team der Professur für Magnetoﬂuiddynamik, Mess- und Automatisierungstechnik, die sich schwerpunktmäßig mit der Entwicklung von Messtechnik beschäftigt, hat bereits mehrere Tomographieanlagen designt und in Betrieb genommen. Mit diesen Anlagen werden Serien von Röntgenbildern einer Probe unter verschiedenen Projektionswinkeln aufgenommen, woraus im Anschluss das 3D-Bild dieser Probe errechnet wird. Eine dieser Anlagen müsste jetzt optimiert werden. Dafür suchen wir Dich und darum geht's:

Ziel der geplanten Anwendung ist es, mittels schneller Tomographie Spuren von Partikeln zu detektieren, die durch eine poröse Struktur strömen. Um das zu verwirklichen, muss die Hard- und Software einer Tomographieanlage optimiert werden, sodass möglichst viele Röntgenbilder in einer möglichst kurzen Zeit aufgenommen werden können. Die Anlagensteuerung, die dafür angepasst werden muss, ist in C++ geschrieben. Auch die Elektronik zur Ansteuerung der Schrittmotoren müsste überarbeitet werden. Gemeinsam mit einer Doktorandin sollen im Team Optimierungsansätze aufgespürt und systematisch umgesetzt werden.

Für diese Arbeit braucht man solide C++ - Kenntnisse, genauso wie Spaß am Programmieren und möglichst auch am Elektronikbasteln. Wenn sich zeigt, dass Dir diese Aufgaben liegen, lassen sich im Haus sicher noch ein paar weitere Geräte finden, die ein kleines Tuning vertragen könnten.

Kontakt:

Judith Marie Undine Siebert, M.Sc.
judith_marie_undine.siebert@tu-dresden.de
+49 351 463-35584
Mollier-Bau – Zimmer 303



```
1 #include <iostream>
2 #include "MotorControl.h"
3
4 using namespace std;
5
6 int main()
7 {
8     cout << "Hello Motor-Controll!" << endl;
9
10    bool test;
11
12    MotorControl* motCon = new MotorControl();
13    int comPort = motCon->FindComport();
14    std::cout << "Comport Index: " << comPort << endl;
15    test = motCon->OpenComport();
16    if ( test == true )
17    {
18        std::cout << "Port " << comPort << " opened ... (bestätigt)" <<
19        endl;
20    }
21
22    motCon->AddStepper( "D:\\CodeBlocks\\Projekt\\Motorsteuerung2018\\
23    std::cout << "Anz der Motoren: " << motCon->Motors.size() << endl;
24    test = motCon->Motors[0]->ReadParameter();
25    std::cout << "Parameter einlesen: " << test << endl;
26
27    int test1 = motCon->GetIsNode();
28    std::cout << "test1: " << test1 << endl;
```

