Wochenbericht KW 27/28

24.6. - 7.7.2013

Projektwochen: 10/11

Erstellt durch
Christoph Gnip

Extern

Fachbereich Elektrotechnik und angewandte Naturwissenschaften
Westfälische Hochschule

Juni 2013

1 Allgemeines

In diesem Wochenbericht werden die Projektwochen 10 und 11 zusammengefasst. Das ermöglicht eine übersichtlichere Darstellung der Arbeitsergebnisse dieser Wochen, da es vor Allem um die Erstellung von Software ging.

2 Projektfortschritt

Das Programm ermöglicht es nun automatisiert alle Antennenkonfigurationen mit allen benötigten Informationen zu erstellen und eine Lösung durch evolutionäre Strategien zu finden. Weiterhin wird automatisiert die Kalibierung des Messaufbaus anhand des in dieser Arbeit entwickelten Modells bestimmt.

2.1 Programmierung

Das Programm wurde in mehrere Libraries aufgeteilt. Zum Einen erhöhen sich Wartbarkeit und Wiederverwendbarkeit und zum Anderen können diese vor-compiliert werden und reduzieren die Compilezeit erheblich. Einen Überblick über die wesentlichsten Merkmale gibt der Anhang A.
Die Verwendung von *C++*11 erleichtert eine Menge der Entwicklung. Die anfängliche Einarbeitung in diesen Standard hat sich bereits mehr als ausgezahlt. Im Besonderen sind zur Zeit die Möglichkeiten für Thread-Programmierung und Verwaltung großer Datenmengen zu nennen. Diese sind anderen aktuellen Programmiersprachen, wie Java oder C♯, überlegen und beschleunigen die Entwicklung.

2.2 Performance

Das Programm läuft in der Entwicklungsumgebung in etwas 2 Minuten durch. Dabei werden alle Permutationen, die Kalibrierung und 12k Lösungen berechnet. Die Ausgabe des Programms ist im Anhang C zu finden. Das Programm erstellt für jede Lösung einen eigenen Task, der asynchron und auf unterschiedlichen Prozessorkernen ausgeführt werden kann. Dadurch verbessert sich die Ausführungszeit erheblich. Das Programm nutzt für die Parallelisierung die neuen Methoden des *C++*11 Standards. Insbesondere std::future und std::async sind in diesem Zusammenhang von großem Nutzen und erleichtern die Verwaltung und die Erstellung von Parallel ausführbaren Code erheblich. Die Entwicklungszeit des aktuellen Stands wäre ohne diese Funktionalitäten erst in einer Woche erreicht worden.

2.3 Neue Modelle von Susanne Winter

In der KW 26 hat Frau S. Winter ihre Ergebnisse vorgestellt und eine Dokumentation [1] übergeben, mit der es möglich ist ihre Ansätze nachzuvollziehen und zu implementieren. Ihre Ansätze unterscheiden sich von denen im Rahmen dieser Arbeit entwickelten grundsätzlich.
Ihre Ansätze werden in jedem Fall umgesetzt und sollte es der Rahmen dieser Arbeit es Zeitlich erlauben wird die Performance gegen diese Modelle gebenchmarkt werden.

2.4 Berechnungsergebnisse nach Normierung

Die im Wochenbericht der KW 25 vorgestellte Visualisierung der Ergebnisse wurde gegen die Kondition der zur Lösung verwendeten Matrix gestellt. Im Anhang B zeigt die Abbildung 1

diese Gegenüberstellung. Eine genauere Betrachtung der Abbildung 1b zeigt, dass die alle Konfigurationen der Antennen 0 und 7 keine sinnvollen Ergebnisse (trotz guter Kondition vgl. 1a) liefern. Das legt den Schluss nahe dass es noch einen systematischen Fehler in der Messdatenaufnahme und/oder anschließender Berechnungsschritte gibt. Dieser wird in weiteren Schritten untersucht.

3 Probleme

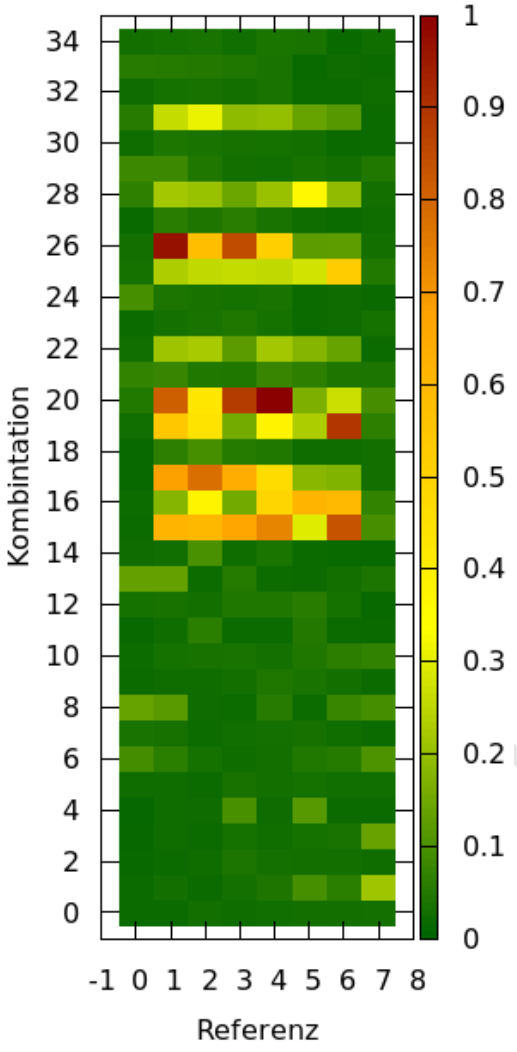
Anhänge

A Dokumentation Libraries

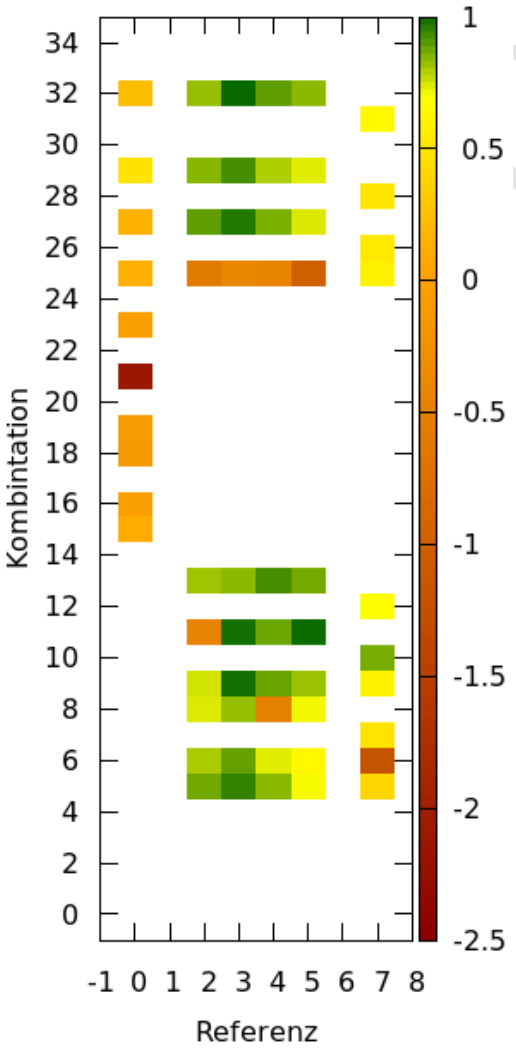
1. libCalibration
   (a) Enthält die Calibration-Klasse
   (b) Bestimmt aus den Entfernungsmessungen von dem Kalibrierphantom die Position der Antennen.
2. libNormalizer
   (a) Sammelt verschiedene Methoden für die Normierung der gemessenen Werte
3. libPermutate
   (a) Erstellt alle möglichen Permutationen die für den Antennenaufbau möglich sind
   (b) Stellt diese Permutationen für spätere Berechnungen komfortabel zur Verfügung
4. libPRPSSystem
   (a) Ließt die Systemkenndaten (z.B. Messfrequenz etc.) aus der entsprechenden Eingabedatei und berechnet die sich daraus ableitenden Faktoren (z.B. Wellenlänge etc.)
5. libSolve
   (a) Das eigentliche Herzstück des Projekts
   (b) Ermöglicht das Lösen mehrerer unterschiedlicher Modelle
   (c) Implementiert unterschiedliche Lösungsstrategien, wie z.B. (1+1)-ES, $(\mu+\lambda)$-ES etc.
   (d) Ist Threadsicher um die Performance erheblich zu verbessern
   (e) Die Dimension des Problem lässt sich leicht anpassen

B Konditionen vs. Berechnete Ergebnisse

Abbildung 1: Analyse der Konditionszahlen gegen die Berechnungsergebnisse eines Messpunktes; Auffällig ist, dass in b) für Referenzantenne 0 und 7 praktisch **keine** korrekte Lösung existieren.



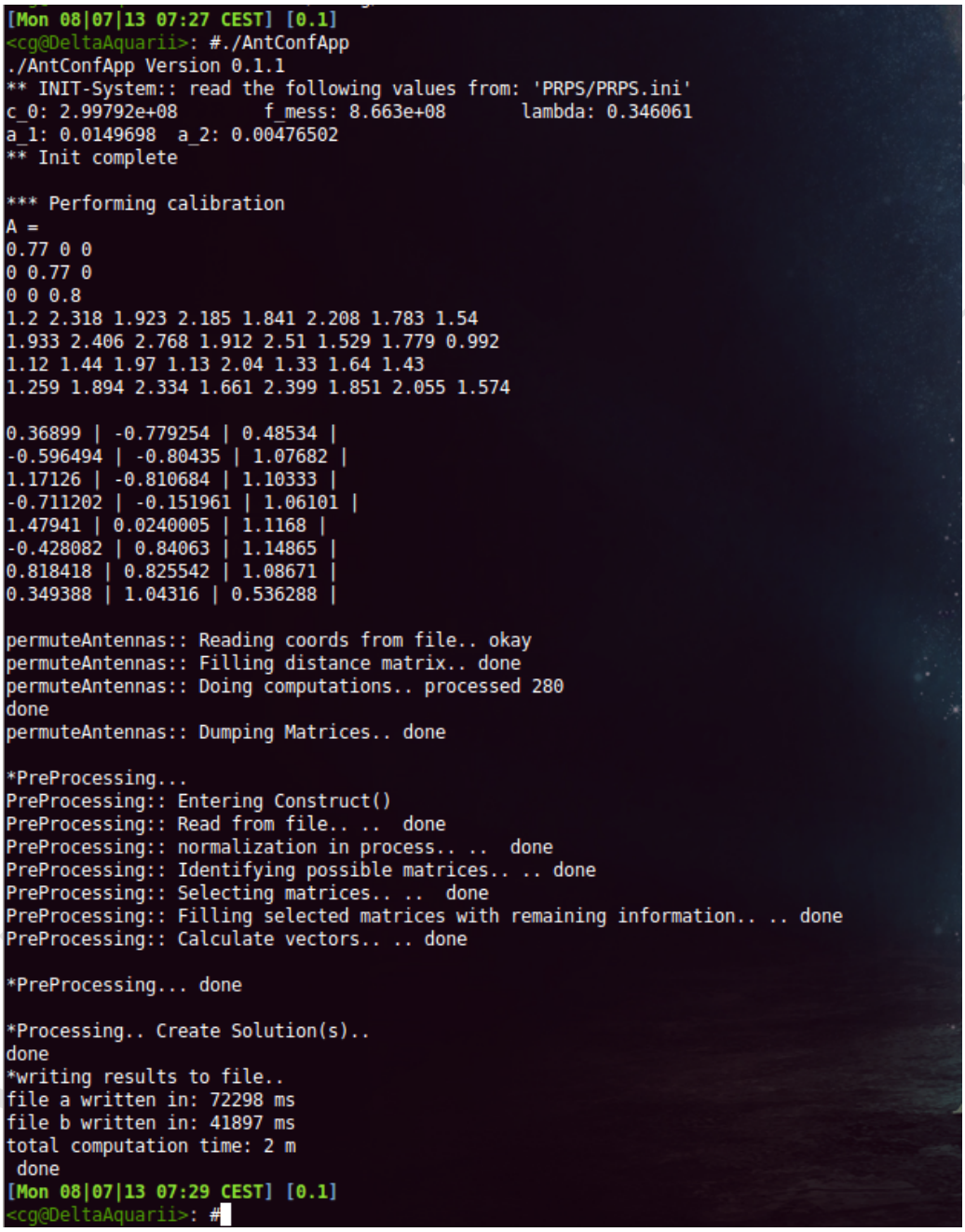
(a) Farbkodierte Matrix der Kondition aller möglichen Kombinationen (skaliert auf den höchsten vorkommenden Wert) B=> grün := gute -, rot= schlechte Konditionierung



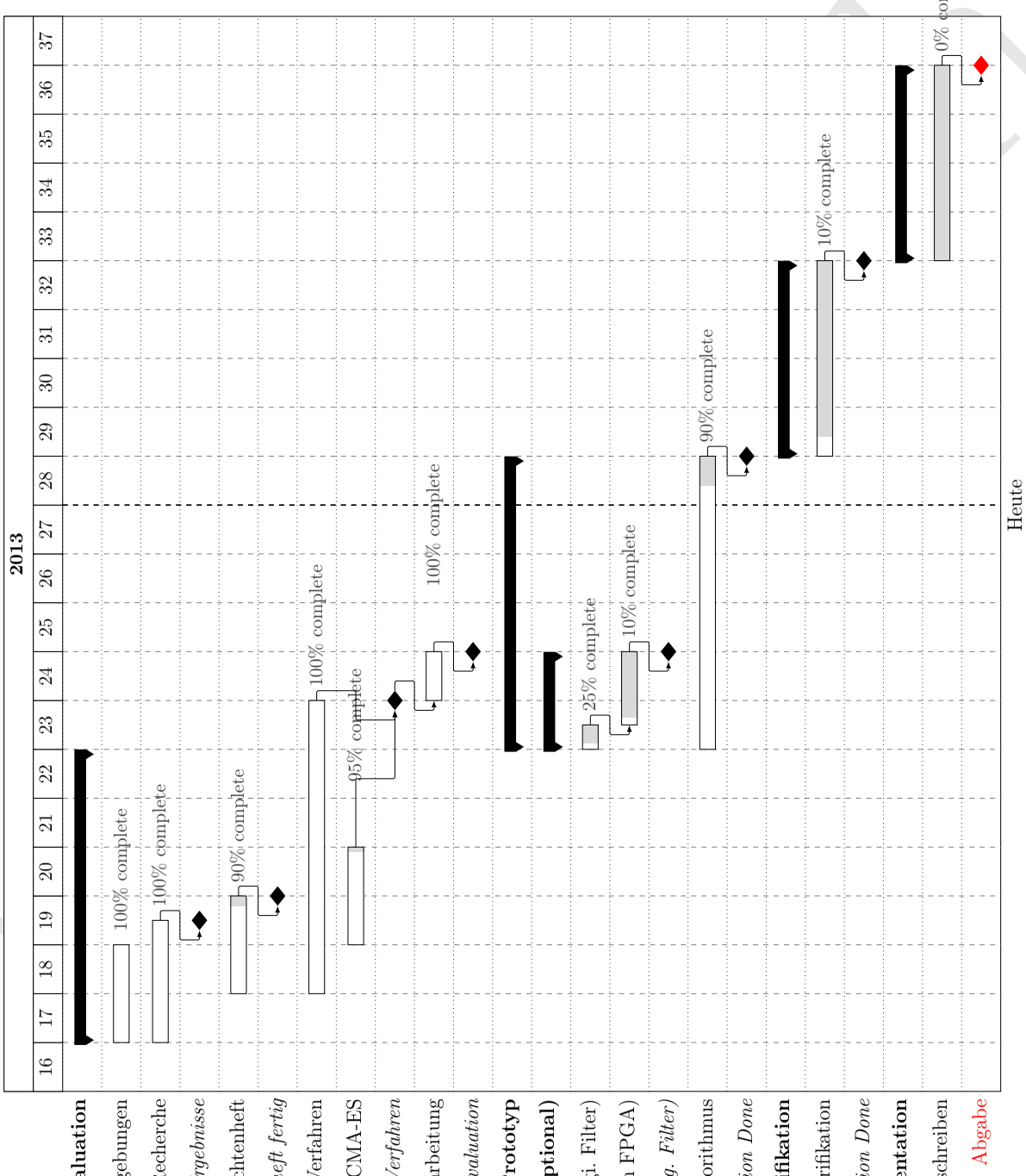
(b) Dargestellt ist Übereinstimmung mit dem wahren Wert, anders ausgedrückt nimmt die Abweichung von oben nach unten zu; Die Position des Ergebnisses in der Matrix entspricht der in der Konditionsmatrix

C Ausgabe des Programms

Abbildung 2: Ausgabe des Programms; Die Ausführungszeit in diesem Beispiel beträgt ca. 2 Minuten, dabei werden 12.000 Lösungen bestimmt und gespeichert.



D Projektlaufplan KW 27



Literatur

[1] S. Winter, "Ansätze zur kalibrierung der wellenzahl im rfid-trackingsystem der firma amedo,"