

# Optimizing geodetic VLBI Simulation Parameters using Swarm Intelligence

**Bachelor Thesis**

**Author(s):**

Wicki, Juliette

**Publication date:**

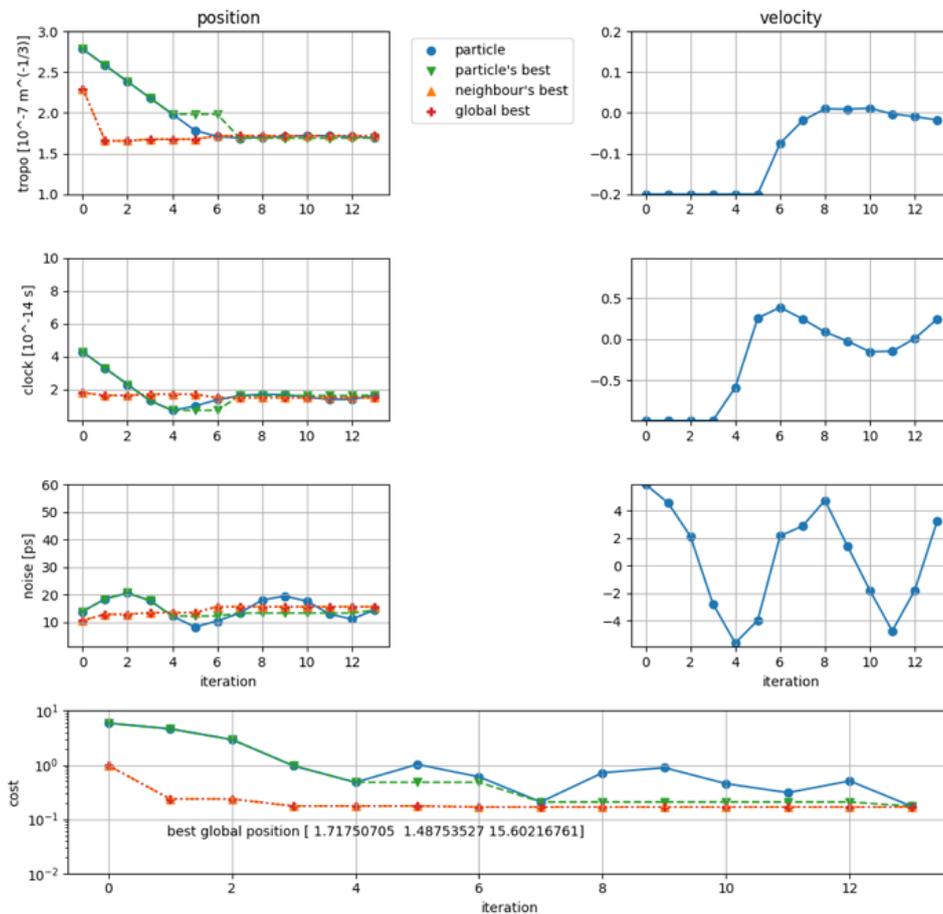
2021-06

**Permanent link:**

<https://doi.org/10.3929/ethz-b-000573140>

**Rights / license:**

[In Copyright - Non-Commercial Use Permitted](#)



# Optimizing geodetic VLBI Simulation Parameters using Swarm Intelligence

Juliette Wicki  
 wickij@student.ethz.ch  
 Juni 2021

Betreuung:  
 Matthias Schartner, mschartner@ethz.ch  
 Prof. Dr. Benedikt Soja, benedikt.soja@geod.baug.ethz.ch

**Bachelorarbeit**  
 Studiengang Raumbezogene Ingenieurwissenschaften

## Abstract

Mit Very Long Baseline Interferometry (VLBI) kann die Differenz der Ankunftszeit eines Signals gemessen werden, um geodätische Parameter wie die Erdorientierungsparameter zu bestimmen. Das empfangene Signal, das von extragalaktischen Quellen ausgesendet wird, ist sehr schwach, weshalb grosse Teleskope und eine hohe Samplingrate notwendig sind. Aus diesem Grund sind VLBI-Messungen zeitintensiv und teuer sowie Simulationen unumgänglich. Das Ziel der Simulationen ist es, Beobachtungen zu berechnen, welche der Realität möglichst nahekommen. Zu diesem Zweck müssen neben den Beobachtungen selbst auch die größten Fehlereinflüsse simuliert werden. Im Falle der VLBI sind das die troposphärische Verzögerung, der Uhrfehler sowie das Messrauschen. Zum Simulieren dieser Fehlereinflüsse werden Modelle verwendet, welche mittels Simulationsparametern angepasst werden können. Bislang werden für alle Messstationen dieselben Simulationsparameter angenommen.

Diese Arbeit zeigt auf, wie mittels Schwarmintelligenz die Simulationsparameter von VLBI-Stationen verbessert werden können, um realistischere Simulationen zu erhalten. Dafür wird der Particle Swarm Optimizer untersucht und dessen Implementierung beschrieben. Er soll die Simulationsparameter bestmöglich an die Realität anpassen. Zusammen mit Beobachtungen der CONT17-Kampagne werden die optimierten Parameter in die Simulationssoftware VieSched++ eingelesen. Mittels eines Vergleiches der Ergebnisse basierend auf den simulierten Beobachtungen und jenen basierend auf den tatsächlichen Beobachtungen kann bestimmt werden, wie gut die Parameter die Realität abbilden. So kann festgestellt werden, dass mit diesem Verfahren für jede Messstation individuelle Simulationsparameter gefunden werden können. Es lässt sich aber auch aufzeigen, dass mithilfe des Particle Swarm Optimizers bessere Ergebnisse als mit den Standardwerten erzielt werden, wenn wie bis anhin für alle Stationen dieselben Parameter angenommen werden.

## Danksagung

Ich möchte folgenden Personen, die zum Gelingen dieser Arbeit beigetragen haben, danken:

Vielen Dank Herr Matthias Schartner, dass Sie mich so kompetent betreut und meine Fragen stets beantwortet haben.

Besten Dank an Beat Rupp, dass du mir deinen Windows-Laptop geliehen hast und ich so auf die benötigte Software zugreifen konnte.

Ein grosses Dankeschön geht auch an meine Eltern für ihre mentale Unterstützung und die wunderbare Möglichkeit, an der ETH studieren zu dürfen.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	VLBI	1
1.2	Simulation	1
1.2.1	Simulationsparameter	2
1.2.2	Simulierte Beobachtungen	2
1.3	Optimierung	3
1.4	Aufgabenstellung	3
<b>2</b>	<b>Methodik</b>	<b>4</b>
2.1	Aufbau des Algorithmus	4
2.2	Simulationssoftware	4
2.3	Particle Swarm Optimizer	4
2.3.1	Partikel	5
2.3.2	Initialisierung	6
2.3.3	Run-Funktion	8
2.3.4	Score-Funktion	9
2.3.5	Update-Funktion	10
2.3.6	Nachbarschaft	12
2.3.7	Hyperparameter	13
2.3.8	Stoppkriterium	14
2.3.9	Speicherung	14
2.3.10	Plots	14
<b>3</b>	<b>Daten</b>	<b>16</b>
<b>4</b>	<b>Resultate</b>	<b>18</b>
4.1	Baseline Lösung	18
4.2	Standard-Durchlauf	18
4.3	Durchlauf mit verschiedenen Nachbarschaften	21
4.3.1	Globale Topologie	21
4.3.2	Ring Topologie	22
4.3.3	Wheel-Topologie	23
4.4	Durchlauf ohne Uhrfehler	25
4.5	Durchlauf mit verschiedenen Simulationsparametern für jede Station	27
4.6	Durchlauf einer anderen Session	29
4.7	Weitere Durchläufe	31
<b>5</b>	<b>Diskussion</b>	<b>33</b>
5.1	Standard-Durchlauf	33
5.2	Nachbarschaften	34
5.3	Ohne Uhrfehler	34
5.4	Verschiedene Simulationsparameter pro Station	34
5.5	Andere Session	34
5.6	Weitere Durchläufe	34
5.7	Auffälligkeiten	34
5.8	Empfehlung	36
<b>6</b>	<b>Fazit</b>	<b>37</b>
	<b>Literaturverzeichnis</b>	<b>38</b>
	<b>Selbstständigkeitserklärung</b>	<b>40</b>
	<b>Anhang</b>	<b>41</b>

# Abbildungsverzeichnis

Abbildung 1.1: VLBI-Teleskop in Wettzell (Bundesamt für Kartographie und Geodäsie (BKG), 2021) ..	1
Abbildung 1.2: Simulation als Blackbox (eigene Darstellung) .....	1
Abbildung 2.1: stark vereinfachter Ablauf des Algorithmus (eigene Darstellung).....	4
Abbildung 2.2: detaillierter Ablauf des Algorithmus (eigene Darstellung).....	5
Abbildung 2.3: Untersuchungsbereich, Startposition und Startgeschwindigkeit eines Partikels im dreidimensionalen Fall (eigene Darstellung) .....	7
Abbildung 2.4: Anpassung der Position eines Partikels anhand seiner besten Position und der besten Position eines anderen Partikels (eigene Darstellung).....	11
Abbildung 2.5: Verhalten des Schwarms, zwei Iterationen in blau und violett (eigene Darstellung) ....	12
Abbildung 2.6: Ring-Topologie (Kennedy et al., 2001) .....	13
Abbildung 2.7: Wheel-Topologie (Kennedy et al., 2001) .....	13
Abbildung 3.1: Messstationen ersten beiden Netzwerke der CONT17-Kampagne, Netzwerk 1 in blau, Netzwerk 2 in rot (Behrend et al., 2020) .....	16
Abbildung 3.2: Messstationen des dritten Netzwerkes der CONT17-Kampagne (Behrend et al., 2020) .....	17
Abbildung 4.1: Vergleich der EOP der Baseline Lösung und der Nachbarschaften zu den Referenzwerten (eigene Darstellung).....	19
Abbildung 4.2: Position, Geschwindigkeit, Kosten eines Partikels beim Standard-Durchlauf (eigene Darstellung) .....	20
Abbildung 4.3: EOP eines Partikels im Vergleich zu den Referenzwerten beim Standard-Durchlauf (eigene Darstellung) .....	20
Abbildung 4.4: Position, Geschwindigkeit, Kosten eines Partikels mit der globalen Nachbarschaft (eigene Darstellung) .....	21
Abbildung 4.5: Annäherung an den Referenzwert DUT1 eines Partikels bei der globalen Topologie (eigene Darstellung) .....	22
Abbildung 4.6: Annäherung an den Referenzwert DUT1 eines Partikels bei der Ring-Topologie (eigene Darstellung) .....	22
Abbildung 4.7: Annäherung an den Referenzwert DUT1 des Partikel 0 bei der Wheel-Topologie (eigene Darstellung) .....	22
Abbildung 4.8: Annäherung an den Referenzwert DUT1 eines anderen Partikels bei der Wheel-Topologie (eigene Darstellung).....	22
Abbildung 4.9: Position, Geschwindigkeit, Kosten eines Partikels mit der Ring-Nachbarschaft (eigene Darstellung) .....	23
Abbildung 4.10: Trägheit eines Partikels in der Wheel-Topologie (eigene Darstellung).....	23
Abbildung 4.11: Position, Geschwindigkeit, Kosten des Partikels 0 bei der Wheel-Topologie (eigene Darstellung) .....	24
Abbildung 4.12: Position, Geschwindigkeit, Kosten eines anderen Partikels der Wheel-Topologie (eigene Darstellung) .....	24
Abbildung 4.13: Uhrfehler wird schnell 0 (eigene Darstellung).....	25
Abbildung 4.14: Position, Geschwindigkeit, Kosten eines Partikels mit der Adaptive Random Topologie ohne Uhrfehler (eigene Darstellung).....	25

Abbildung 4.15: Position, Geschwindigkeit, Kosten eines Partikels mit der globalen Topologie ohne Uhrfehler (eigene Darstellung).....	26
Abbildung 4.16: Annäherung an den Referenzwert DUT1 eines Partikels bei der Adaptive Random Topologie ohne Uhrfehler (eigene Darstellung).....	26
Abbildung 4.17: Annäherung an den Referenzwert DUT1 eines Partikels bei der globalen Topologie ohne Uhrfehler (eigene Darstellung).....	26
Abbildung 4.18: Vergleich der EOP der Durchläufe ohne Uhrparameter mit der Baseline Lösung und den Referenzwerten (eigene Darstellung).....	27
Abbildung 4.19: Annäherung an den Referenzwert DUT1 bei der Adaptive Random Topologie, wenn für jede Station individuelle Simulationsparameter bestimmt werden (eigene Darstellung).....	29
Abbildung 4.20: Annäherung an den Referenzwert DUT1 bei der globalen Topologie, wenn für alle Stationen individuelle Simulationsparameter bestimmt werden (eigene Darstellung).....	29
Abbildung 4.21: Position, Geschwindigkeit, Kosten eines Partikels beim Standard-Durchlauf der Session v1702 (eigene Darstellung).....	30
Abbildung 4.22: Annäherung an den Referenzwert DUT1 beim Standard-Durchlauf der Session v1702 .....	30
Abbildung 4.23: Vergleich der EOP der Durchläufe der Session v1702 mit der Baseline Lösung und den Referenzwerten (eigene Darstellung).....	31
Abbildung 4.24: EOP eines Partikels im Vergleich zu den Referenzwerten bei der globalen Topologie, wenn die Wiederholbarkeit zum Vergleich verwendet wird (eigene Darstellung).....	31
Abbildung 4.25: Position, Geschwindigkeit, Kosten eines Partikels bei der globalen Topologie, wenn die Wiederholbarkeit zum Vergleich verwendet wird (eigene Darstellung).....	32
Abbildung 4.26: Position, Geschwindigkeit, Kosten bei Standard-Durchlauf mit Social und Cognitive Parameter auf 2, score ist mit cost gleichzusetzen (eigene Darstellung).....	32
Abbildung 5.1: Vergleich PSO, Baseline Lösung und Referenzwerte (eigene Darstellung).....	33
Abbildung 5.2: "springende" Partikel (eigene Darstellung), score ist mit cost gleichzusetzen .....	35
Abbildung 5.3: anfangs abnehmende Genauigkeit der EOP .....	35
Abbildung 5.4: Kosten des Nachbarpartikels sind höher als die Kosten des Partikels, score ist mit cost gleichzusetzen (eigene Darstellung).....	35

## Tabellenverzeichnis

Tabelle 4.1: Simulationsparameter pro Station mit der Adaptive Random Topologie .....	27
Tabelle 4.2: Simulationsparameter pro Station mit der globalen Topologie .....	28
Tabelle 5.1: empfohlene Charakteristika des PSO zur Optimierung der VLBI-Simulationsparameter .	36

## Abkürzungsverzeichnis

$C_n$	Troposphärische Verzögerung
CONT17	Continuous VLBI Campaign 2017
csv	Comma Separated Values
DUT1	Differenz UT1 und UTC
EOP	Erdorientierungsparameter
IVS	International VLBI Service for Geodesy and Astronomy
ICRF	International Celestial Reference Frame
$\mu\text{s}$	Mikrobogensekunden
$\mu\text{s}$	Mikrosekunden
nutx	Nutation in x-Richtung
nuty	Nutation in y-Richtung
ps	Picosekunden
PSO	Particle Swarm Optimizier
skd	Autodesk Autosketch Drawing Database
sta_coord	Stationskoordinaten
UTC	Universal Time Coordinated
UT1	Universal Time
VGOS	VLBI Global Observing System
VLBI	Very Long Baseline Interferometry
XML	Extensible Markup Language
xpo	Polschwankung in x-Richtung
ypo	Polschwankung in y-Richtung

# 1 Einleitung

Bereits der Titel dieser Arbeit enthält viele Fachbegriffe. In diesem Kapitel sollen einige davon erläutert und die Aufgabenstellung geklärt werden.

## 1.1 VLBI

VLBI ist die Abkürzung für Very Long Baseline Interferometry und beschreibt eine Technik, mit welcher geodätische Parameter bestimmt werden können. Dabei werden grosse Radioteleskope eingesetzt, um eingehende Signale von extragalaktischen Radioquellen zu empfangen (Abbildung 1.1). Um die Genauigkeit der Messmethode zu erhöhen und ein interferometrisches Messverfahren durchführen zu können, werden mehrere dieser Teleskope zusammengefasst. Sie bilden sogenannte Basislinien, welche sich über mehrere Kontinente erstrecken können. Die Teleskope beobachten jeweils für eine gewisse Zeit dieselbe Radioquelle, damit mittels Kreuzkorrelation die Differenz der Ankunftszeit der Signale quantifiziert werden kann. Die Kreuzkorrelation ermöglicht die Detektion des Signals, obwohl dieses oftmals viel schwächer als das Hintergrundrauschen ist. Dies liegt daran, dass die Radioquellen weit entfernt von der Erde sind, was dazu führt, dass diese Radioquellen im Vergleich zur Erde keine feststellbare Bewegung aufweisen. Daher können aus VLBI-Messungen die Erdorientierungsparameter (EOP) bestimmt werden, welche für die Positionierung und Navigation auf der Erde sowie im All benötigt werden (Kapitel 1.2.2). Zudem werden die Messungen für die Realisierung des International Celestial Reference Frame (ICRF) verwendet, welches die Grundlage für sämtliche Koordinatensysteme bildet. (Bundesamt für Kartographie und Geodäsie (BKG), 2021; Schuh & Böhm, 2013)



Abbildung 1.1: VLBI-Teleskop in Wettzell (Bundesamt für Kartographie und Geodäsie (BKG), 2021)

## 1.2 Simulation

Wie aus Kapitel 1.1 hervorgeht, ist das empfangene Signal der VLBI-Stationen sehr schwach. Daher benötigt man für die Messungen eine hohe Samplingrate, welche viele Daten produziert, die gespeichert und verarbeitet werden müssen. VLBI-Messungen sind folglich zeitintensiv und teuer, weshalb oftmals Simulationen durchgeführt werden.

Bei Simulationen werden Modelle verwendet, um Beobachtungen inklusive ihrer grössten Fehlereinflüsse zu simulieren. Dabei können die Modelle mittels Simulationsparameter angepasst werden, um die Realität möglichst exakt abzubilden (Abbildung 1.2).

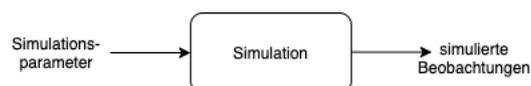


Abbildung 1.2: Simulation als Blackbox (eigene Darstellung)

Anschliessend können die simulierten Beobachtungen wie echte Beobachtungen analysiert werden. Daher kann eine Simulation Genauigkeiten von zukünftigen Messungen vorhersagen.

### 1.2.1 Simulationsparameter

Simulationsparameter beschreiben die Werte, die angenommen werden, um die Simulation zu berechnen. Für VLBI-Simulationen werden die drei Simulationsparameter *troposphärische Verzögerung*, *Uhrfehler* und *Messrauschen* benötigt (Pany et al., 2011).

#### **Troposphärische Verzögerung**

Die troposphärische Verzögerung beschreibt die Verzögerung des Signals aufgrund der Atmosphäre. Das heisst, es wird das Verhältnis der Lichtgeschwindigkeit im Vakuum zur Ausbreitungsgeschwindigkeit in der Atmosphäre betrachtet, weshalb es sich um einen Brechungsindex handelt. In der Fachliteratur wird die Abkürzung  $C_n$  verwendet. Es wird zwischen einem trockenen und einem feuchten Anteil unterschieden, wobei letzterer schwieriger zu modellieren ist. (Rothacher et al., 2021)

Die Verzögerung ist auch unter dem Begriff Refraktion bekannt und gilt als grösste Fehlerquelle für VLBI. Die Korrektur ist folglich massgeblich für die Genauigkeit der Simulationen und Messungen. Diese Korrektur ist aber nicht genau bekannt und muss daher während der Analyse geschätzt werden. Im Artikel von Nilsson und Haas (2010) wird die Simulation der troposphärischen Verzögerung genauer beschrieben.

Gemessen wird die troposphärische Verzögerung in der Einheit  $10^{-7} m^{-1/3}$  (Schartner et al., 2020).

#### **Uhrfehler**

Aufgrund der allgemeinen und spezifischen Relativitätstheorie laufen Uhren nicht gleich schnell abhängig davon, ob sie sich bewegen und wie stark das Gravitationsfeld auf sie wirkt (Rothacher et al., 2021). Des Weiteren haben auch die genauesten Atomuhren gewisse Fehler, welche zu Unregelmässigkeiten führen können. Daher ist es wichtig, dass auch die Uhr in der Simulation von VLBI berücksichtigt wird. Der Uhrfehler wird dabei wie im Artikel von Herring et al. (1990) beschrieben simuliert. Angegeben wird der Fehler in  $10^{-14} s$ , was der Allan Standardabweichung nach 50 min entspricht (Schartner et al., 2020).

#### **Messrauschen**

Wie in Kapitel 1.1 erwähnt, ist das Messrauschen bei VLBI-Messungen wegen der weiten Entfernung der Radioquellen von der Erde besonders gross. Die Simulation des Rauschens ist daher zentral. Messrauschen ist unkorreliert und wird daher weisses Rauschen genannt. Es kann mittels Monte Carlo simuliert werden (Guillaume, 2018). Die Masseinheit des Rauschens ist Picosekunden (ps) (Schartner et al., 2020).

### 1.2.2 Simulierte Beobachtungen

Simulierte Beobachtungen sind die Ergebnisse einer Simulation und sollen die Realität möglichst gut wiedergeben. Mit Hilfe der VLBI-Simulationen können geodätische Parameter wie EOP oder Stationskoordinaten geschätzt und deren erwartete Genauigkeit berechnet werden.

Die EOP sind zentrale Elemente der Positionierung und Navigation auf der Erde und im All. Sie beschreiben Unregelmässigkeiten der Erdrotation wie die Polschwankung oder die Nutation. Die Polschwankung und die Nutation werden in der x- sowie der y-Achse angegeben. Die

x- und y-Richtungen sind wie üblich in der Geodäsie definiert. Das heisst, bei der Nutation zeigt die x-Achse zum Frühlingspunkt und die y-Achse ergänzt das Rechtssystem, weil es sich um das Inertialsystem handelt. Die Polbewegung dagegen wird im erdfesten System angegeben. Die x-Komponente beschreibt daher die Bewegung in Nord-Süd-Richtung und die y-Komponente die Bewegung in Ost-West-Richtung. Ein weiterer EOP ist die Zeitkorrektur DUT1. Diese Korrektur ist die Differenz der Universal Time (UT1) und der Universal Time Coordinated (UTC) und beschreibt die unregelmässigen Tageslängen aufgrund der schwankenden Rotationsperiode der Erde. (Rothacher et al., 2019)

Folglich gibt es bei VLBI-Simulationen sechs simulierte Beobachtungen: die *Polbewegung in x- und y-Richtung*, die *Nutation in x- und y-Richtung*, die *Zeitkorrektur DUT1* und die *Stationskoordinaten*.

### 1.3 Optimierung

Optimierung beschreibt den Prozess des Verbesserns. Mathematisch gesprochen werden dafür Entscheidungsvariablen verwendet, die das Optimierungsproblem umschreiben. Diesen Variablen sollen durch die Optimierung die idealen Werte zugeteilt werden. Die Beschreibung des Problems wird Modellierung genannt. Mit Hilfe der Zielfunktion kann schliesslich die Qualität der Modellierung festgestellt werden. Die Optimierung selbst wird hauptsächlich durch Algorithmen durchgeführt. Diese Algorithmen sind meist iterativ. Das heisst, sie finden schrittweise die beste Lösung. In dieser Arbeit wird der Particle Swarm Optimizer (PSO) verwendet, welcher iterativ angewendet wird (Kapitel 2.3). (Grimme & Bossek, 2018)

### 1.4 Aufgabenstellung

Ziel dieser Arbeit ist es, Simulationsparameter für VLBI mittels eines Optimierungsalgorithmus zu finden. Diese sollen Beobachtungen produzieren, die die Realität möglichst exakt widerspiegeln. Bis lang ist es so, dass VLBI-Simulationen pro Messstation dieselben Simulationsparameter annehmen. Es soll untersucht werden, ob mit Hilfe eines Optimierungsalgorithmus für jede Station eigene Simulationsparameter gefunden werden können, welche die Realität besser abbilden.

Dies ist entscheidend, weil sich beispielsweise die Eigenschaften der troposphärischen Verzögerung je nach Feuchtigkeitsgehalt in der Atmosphäre ändern. Es kann daher nicht davon ausgegangen werden, dass Teleskope in trockenen Gebieten, im hohen Norden oder im Regenwald dieselbe Refraktion haben. Der Optimierungsalgorithmus soll daher gezielt Parameter finden, die dieser Ortsabhängigkeit Rechnung tragen.

Als Optimierungsalgorithmus wird der Particle Swarm Optimizer gewählt (Kapitel 2.3).

## 2 Methodik

Wie in der Aufgabenstellung (Kapitel 1.4) beschrieben, geht es nicht um die Simulation von VLBI-Messungen. Die Ergebnisse der Simulation dienen aber dazu die Performanz des Optimierungsalgorithmus zu bewerten (Abbildung 2.1). Denn dieser steht im Fokus dieser Arbeit, um die Simulationsparameter zu verbessern.

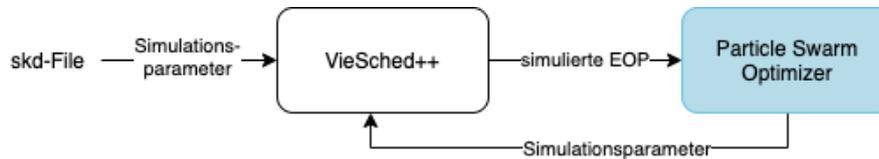


Abbildung 2.1: stark vereinfachter Ablauf des Algorithmus (eigene Darstellung)

### 2.1 Aufbau des Algorithmus

Die Beobachtungsdaten, die für die Simulation verwendet werden, sind in skd-Files gespeichert (Kapitel 3). Für jedes dieser Files läuft der Algorithmus vollständig durch. Dafür wird anfangs der Particle Swarm Optimizer initialisiert (Kapitel 2.3.2), woraufhin die Simulationssoftware aufgerufen wird (Kapitel 2.2 und Kapitel 2.3.3). Das skd-File sowie ein XML-File mit den initialen Simulationsparametern werden der Software übergeben. Diese fungiert als Black-box, die ein csv-Dokument mit den simulierten EOP zurückgibt. Das Dokument wird vom Algorithmus eingelesen und der PSO kommt zum Zug. Dazu werden zuerst die Werte im csv-Dokument mit Referenzwerten verglichen (Kapitel 2.3.4). So kann ermittelt werden, wie gut die Simulationsparameter die Realität widerspiegeln. Anhand ihrer Qualität werden die Simulationsparameter schliesslich angepasst (Kapitel 2.3.5). Die verbesserten Parameter können wieder der Simulationssoftware übergeben werden und erneut werden die daraus resultierenden EOP bewertet. Das Verfahren wiederholt sich so lange, bis das Stoppkriterium erreicht wird (Kapitel 2.3.8). Dies bedeutet, der PSO hat die numerisch optimalen Simulationsparameter gefunden. Alle Ergebnisse werden abgespeichert und in verschiedenen Plots dargestellt (Kapitel 2.3.9 und 2.3.10). Abbildung 2.2 zeigt den Ablauf schematisch auf.

Den Code für den Algorithmus bilden mehrere Python-Skripte.

### 2.2 Simulationssoftware

Die Simulation von VLBI wird in dieser Arbeit mit der Software VieSched++ gerechnet (Schartner & Böhm, 2019), beruhend auf den Arbeiten von Pany et al. (2011) und Petrachenko et al. (2009). Der Software wird ein XML-Dokument mit den Simulationsparametern übergeben (Kapitel 2.3.3). Daraus simuliert sie Beobachtungen, welche in einem csv-File gespeichert werden. Konkret bekommt man die Polschwankung in x- und y-Richtung in Mikrobogensekunden ( $\mu\text{s}$ ), die Nutation in x- und y-Richtung in  $\mu\text{s}$ , die Zeitkorrektur DUT1 in  $\mu\text{s}$  sowie die Mittelung der 3d-Stationskoordinaten in mm.

### 2.3 Particle Swarm Optimizer

Der Particle Swarm Optimizer wird in dieser Arbeit als Optimierungsalgorithmus verwendet. Er gehört zu den Algorithmen, welche auf der Schwarmintelligenz beruhen. Diese haben in den letzten 20 Jahren stark an Popularität gewonnen, weil sie nicht-lineare Optimierungsprobleme

effizient lösen und flexibel einsetzbar sind. Inspiriert werden die Algorithmen von natürlichen Vogel- oder Fischeschwärmen. Das heisst, das Verhalten des Einzelnen in einer Population ist abhängig vom Verhalten seiner Umgebung, sprich der Kolonie. Der Algorithmus ist daher populationsbasiert, was bedeutet, dass viele Agenten miteinander interagieren. Wie der Name schon sagt, werden die Agenten Partikel genannt. Jedes Partikel sucht dabei den Untersuchungsbereich des Optimierungsproblems, der durch die Entscheidungsvariablen definiert ist, nach der besten Lösung ab und informiert andere Partikel darüber. Damit dies gelingt, muss die Art des Informationsaustauschs festgelegt werden (Kapitel 2.3.6). (Yang, 2013)

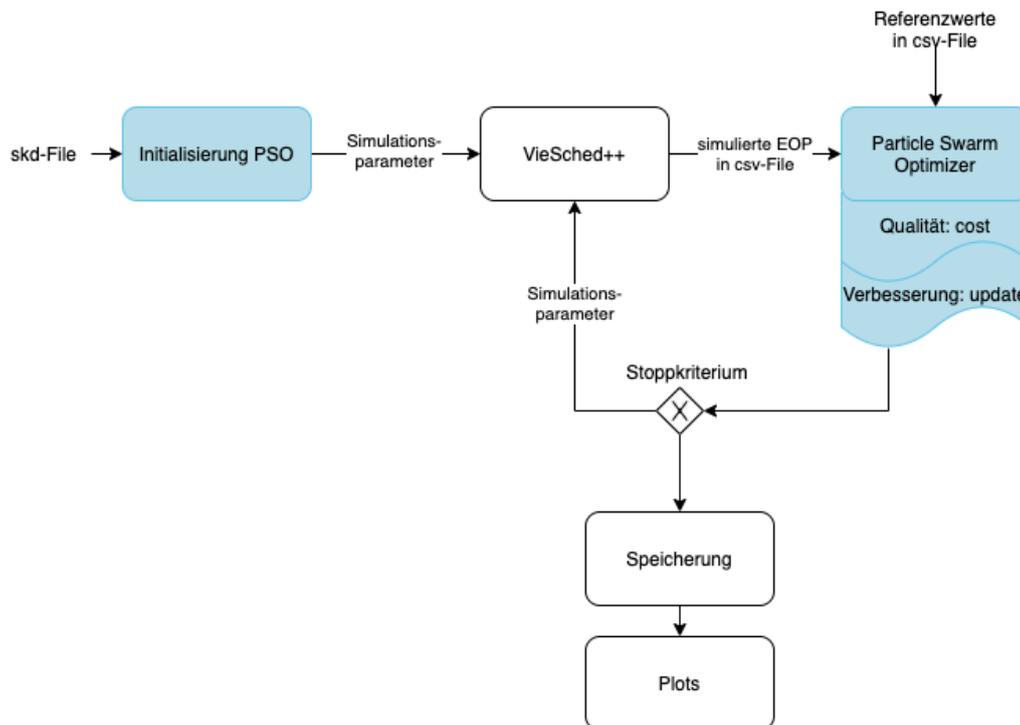


Abbildung 2.2: detaillierter Ablauf des Algorithmus (eigene Darstellung)

Der PSO gehört zu den metaheuristischen Algorithmen. Im Gegensatz zu heuristischen Methoden wird dabei nicht mit «trial and error» die optimale Lösung gesucht, sondern es werden kompliziertere Selektionsmechanismen angewendet. Den verbreitesten Selektionsmechanismus kennt man aus der Natur unter dem Schlagwort «Survival of the Fittest». Dabei überleben nur die bestangepassten Individuen. Im PSO sterben keine Partikel, aber sie orientieren sich jeweils an den Besten (Kapitel 2.3.5). Metaheuristische Algorithmen sind effizient, weil sie sich auf die Natur beziehen und diese imitieren. Dennoch gelingt es metaheuristischen Algorithmen nicht immer, erfolgreich das globale Optimum zu finden. (Yang, 2013)

### 2.3.1 Partikel

Die Partikel werden jeweils durch drei Kenngrößen beschrieben: die *Position*, die *Geschwindigkeit* und die *Eignung* (Yang, 2013). Da der Algorithmus iterativ das globale Optimum sucht, ändern sich diese drei Größen pro Iteration. Wie sie sich gegenseitig beeinflussen, wird in Kapitel 2.3.4 und Kapitel 2.3.5 beschrieben.

Zudem ist jedes Partikel mit einer Nummer versehen, damit sie unterschieden werden können. Wie in der Informatik typisch beginnt die Nummerierung mit 0.

## Position

Die Position ist durch die Entscheidungsvariablen des Optimierungsproblems definiert (Yang, 2013). Hier sind dies die drei Simulationsparameter, da diese optimiert werden sollen. Folglich definiert jede Position einen Satz an Simulationsparameter und das Ziel des Algorithmus ist es, jene Position bzw. Simulationsparameter zu finden, welche die Realität am besten abbilden. Die Entscheidungsvariablen, hier die Simulationsparameter, bilden auch die Grundlage für den Untersuchungsbereich. Dieser und daher auch die Position der Partikel ist dreidimensional, wenn für alle Messstationen dieselben Parameter angenommen werden, weil es genau drei Simulationsparameter gibt (Abbildung 2.3). Wenn wie in Kapitel 1.4 angedeutet für alle Stationen eigene Parameter gefunden werden sollen, ist das Problem höher-dimensional. Die Dimension ist dann abhängig von der Anzahl Stationen, die das Netzwerk bilden (Formel (2.1)):

$$d = s \cdot n \quad (2.1)$$

wobei

- d      Dimension
- s      Anzahl Stationen
- n      Anzahl Simulationsparameter, hier 3

## Geschwindigkeit

Die Geschwindigkeit des Partikels zeigt die Änderung der Position auf, weshalb sie gleich viele Dimensionen wie die Position aufweist. Abhängig von der Geschwindigkeit, die das Partikel bereits hat, der besten Position des Partikels, der Position anderer Partikel sowie der jetzigen Position des Partikels ändert sich die Geschwindigkeit (Kapitel 2.3.5).

## Eignung

Die Eignung des Partikels charakterisiert die Qualität der Position. In dieser Arbeit ist sie ein Mass dafür, wie gut die Position, beziehungsweise die simulierten Ergebnisse basierend auf den korrespondierenden Simulationsparameter, die Realität widerspiegelt. Hier wird sie aus den Ergebnissen der Simulationssoftware bestimmt (Kapitel 2.3.4).

### 2.3.2 Initialisierung

In der Initialisierung werden die Partikel erstellt. Sie erhalten dabei eine individuelle Startposition sowie eine individuelle Startgeschwindigkeit.

#### Startposition

Für jede Dimension des Partikels wird die Startposition anhand der Formel (2.2) bestimmt (Clerc, 2012):

$$p_d(0) = U(\min_d, \max_d) \quad (2.2)$$

wobei

- $p_d(0)$       initiale Position der Dimension d

$U(\min_d, \max_d)$  uniform verteilte Zufallszahl im Bereich von  $\min_d$  bis  $\max_d$

Die Grenzen des Untersuchungsbereichs,  $\min_d$  und  $\max_d$ , sind dabei abhängig von der Dimension, sprich den erlaubten Werten der einzelnen Simulationsparameter. Die troposphärische Refraktion darf anfangs beispielsweise Werte von  $1 \times 10^{-7} m^{-1/3}$  bis  $3 \times 10^{-7} m^{-1/3}$ , der Uhrfehler Werte von  $0.1 \times 10^{-14} s$  bis  $10 \times 10^{-14} s$  und das weisse Rauschen Werte von  $1 ps$  bis  $60 ps$  annehmen (M. Schartner, persönliche Kommunikation, 2021). Folglich ist bei der troposphärischen Refraktion  $\min_d$  1 und  $\max_d$  3.

Wenn alle Stationen dieselben Simulationsparameter haben und ein dreidimensionales Problem betrachtet wird, sieht der Untersuchungsbereich und eine mögliche Startposition wie in Abbildung 2.3 aus.

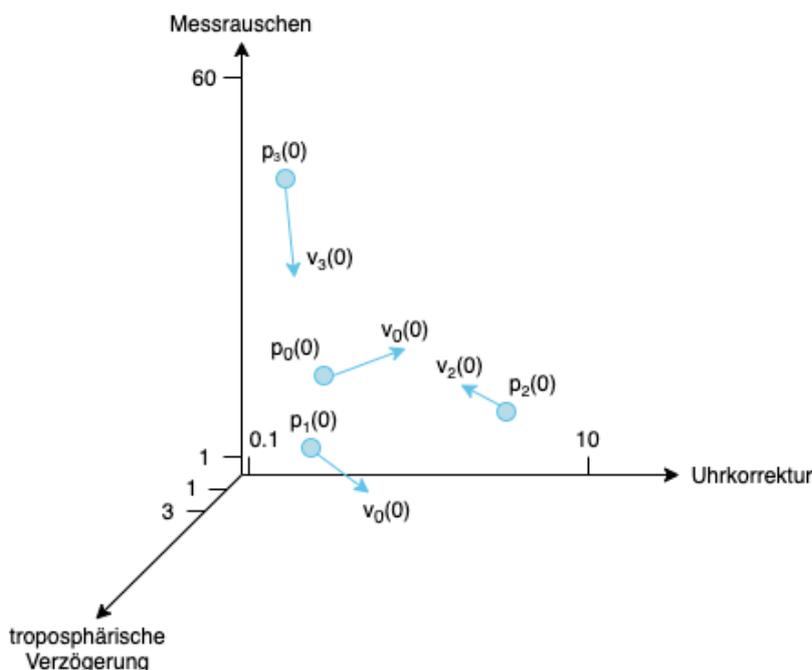


Abbildung 2.3: Untersuchungsbereich, Startposition und Startgeschwindigkeit eines Partikels im dreidimensionalen Fall (eigene Darstellung)

### Startgeschwindigkeit

Die initiale Geschwindigkeit jeder Dimension ist ebenfalls abhängig von den Grenzwerten des Untersuchungsbereichs, wird aber zusätzlich noch von der Startposition beeinflusst. Die Formel (2.3) beschreibt diese Abhängigkeit. (Clerc, 2012)

$$v_d(0) = \frac{U(\min_d, \max_d) - p_d(0)}{2} \quad (2.3)$$

wobei

$v_d(0)$  initiale Geschwindigkeit der Dimension  $d$

$U(\min_d, \max_d)$  uniform verteilte Zufallszahl im Bereich von  $\min_d$  bis  $\max_d$

$p_d(0)$  initiale Position

Damit die Geschwindigkeit nicht zu gross wird, wird sie mit einer maximalen Geschwindigkeit  $v_{\max}$  begrenzt. Das heisst, wenn die Startgeschwindigkeit die maximale Geschwindigkeit übersteigt, wird sie auf diese limitiert, behält aber ihr Vorzeichen (Formel (2.4)). (Yang, 2013)

$$v_d(0) = \text{sign}(v_d(0)) \cdot \min(|v_d(0)|, |v_{\max,d}|) \quad (2.4)$$

Wichtig ist dabei, dass der Vergleich mit  $v_{\max}$  komponentenweise passiert. Dazu wird pro Dimension die Geschwindigkeit der maximal erlaubten Geschwindigkeit gegenübergestellt. Ein blosser Normvergleich der Vektoren wäre ungeschickt, weil sich die einzelnen Komponenten aufgrund der verschiedenen Grenzwerte des Untersuchungsbereichs stark unterscheiden. Bei einem Vergleich der Norm der Vektoren kämen diese Unterschiede nicht zum Tragen.

Die maximale Geschwindigkeit für jede Dimension wird gemäss Formel (2.5) gebildet. Der Skalierfaktor 0.1 ist dabei aufgrund des Verhaltens des Algorithmus so gewählt worden.

$$v_{\max,d} = 0.1 \cdot (\max_d - \min_d) \quad (2.5)$$

Es gäbe diverse andere Varianten, die Startgeschwindigkeit zu definieren (vgl. Kennedy et al., 2001). Die hier gewählte garantiert aber im Fall der Optimierung von VLBI-Simulationsparameter ein erfolgreicher Durchlauf des Algorithmus.

### Schwarmgrösse

Wie viele Partikel der Schwarm enthält, kann frei gewählt werden. Es gibt dabei keinen optimalen Wert, weshalb Werte zwischen 10 und 50 empfohlen werden (Kennedy et al., 2001). Hier wird der Schwarm mit 16 Partikeln initialisiert. Dies hat den Grund, dass genau so viele parallellaufende Prozesse stattfinden können und der Algorithmus gute Ergebnisse liefert (Kapitel 2.3.3 und Kapitel 4). Mit einem Parser-Argument kann die Anzahl Partikel aber bei jedem Durchlauf des Algorithmus nach Wunsch angepasst werden (Python Software Foundation, 2021a).

Eine weitere Möglichkeit, die Schwarmgrösse zu bestimmen, bietet dabei folgende Formel:

$$S = 10 + [2 \cdot \sqrt{d}] \quad (2.6)$$

mit

S      Schwarmgrösse

d      Dimension

$[2 \cdot \sqrt{d}]$     ganzzahliger Anteil der reellen Zahl  $2 \cdot \sqrt{d}$

Untersuchungen zeigen, dass die so definierte Schwarmgrösse nicht zwingend passend ist, weshalb sie in dieser Arbeit nicht angewendet wird. (Clerc, 2012)

### 2.3.3 Run-Funktion

Nach der Initialisierung wird für jedes Partikel die Run-Funktion aufgerufen. In dieser wird der Simulationssoftware die berechnete Position, sprich die bestimmten Simulationsparameter, in einem XML-File übergeben. Als Ergebnis der Simulation erhält man eine csv-Datei, die die

simulierten EOP enthält. Dies läuft parallelisiert ab, was bedeutet, dass der Prozess für alle Partikel gleichzeitig abläuft.

Damit dafür genügend Rechenleistung zur Verfügung steht, läuft der gesamte Algorithmus über einen Server an der ETH Zürich. Für den Serverzugriff wird die Software PuTTY genutzt (Tatham, 2021), wobei mittels GIT die aktuelle Version des Codes abgefragt wird (Git, 2021).

### 2.3.4 Score-Funktion

Nachdem für jedes Partikel die Simulationssoftware ausgeführt worden ist, kann die Qualität der Position des Partikels bestimmt werden. Dies geschieht, indem die EOP, die die Simulation berechnet hat, mit Referenzwerten verglichen werden. Die Referenzwerte stammen aus Beobachtungen und sind für jedes Netzwerk an jedem Tag bekannt (Kapitel 3). Je besser die simulierten EOP den Referenzwerten entsprechen, desto besser stimmen die Positionen des Partikels mit den tatsächlichen Gegebenheiten überein; oder anders gesagt, desto realistischer sind die Simulationsparameter, die mit dem PSO bestimmt und in die Software eingegeben worden sind.

Die EOP werden aus dem csv-File, das die Simulationssoftware generiert, herausgelesen. Dabei gibt es zwei mögliche Werte, die betrachtet werden können: entweder der mean formal error, der mittlere, formale Fehler, oder die repeatability, die Wiederholbarkeit. Aufgrund besserer Ergebnisse wird in dieser Arbeit vorwiegend der mittlere, formale Fehler verwendet (Kapitel 4.7 und Kapitel 5.6).

Der Vergleich der einzelnen EOP mit den entsprechenden Referenzwerten basiert auf deren Verhältnisse. Dabei muss darauf geachtet werden, dass die Werte dieselbe Masseinheit haben. Gegebenenfalls müssen daher die Werte skaliert werden.

Von jedem Verhältnis wird nun 1 abgezogen, da 1 dem optimalen Ergebnis entspricht. Es werden die Absolutbeträge der reduzierten Verhältnisse gebildet und aufaddiert (Formel (2.7)). Da es sechs EOP gibt, wenn die Stationskoordinaten dazugezählt werden, werden sechs Terme aufaddiert. Dabei wird der Absolutbetrag genommen, damit sich die Abweichungen der einzelnen EOP vom optimalen Verhältnis nicht rausmitteln. Die gebildete Summe ist die Bewertung der Position des Partikels. Je kleiner die Bewertung ist, desto besser passt die Position mit der Realität überein. Die Bewertung entspricht daher eigentlich einer Kosten- und nicht einer Score-Funktion. Denn nach Definition sollen Kosten im Gegensatz zu Scores möglichst klein sein. Im Code wird aber immer von Scores gesprochen, weshalb das Kapitel diese Überschrift trägt.

$$c_p = \sum_{i=1}^{i=6} \left| \frac{e_i}{e_{i,ref}} - 1 \right| \quad (2.7)$$

wobei

- $c_p$  Kosten eines Partikels
- $e_i$  ein EOP
- $e_{i,ref}$  Referenzwert desselben EOP

Es sollen nicht nur die Mittelwerte der Verhältnisse angeschaut werden, weil sich so das eben Geschilderte ergibt und sich grosse Abweichungen rausmitteln können. Dies ist nicht das Ziel, da die Position alle EOP gut simulieren soll und nicht nur deren Mittel.

Die berechnete Bewertung wird als Eignung gespeichert, damit man später den Verlauf der Kosten sehen kann (Kapitel 2.3.10). Ausserdem wird die Bewertung mit der bisher besten Bewertung des Partikels verglichen. Wenn sich die Kosten verringert haben, wird die neue Bewertung als beste Bewertung abgespeichert und dementsprechend die beste Position angepasst. So kann man später ebenfalls die Entwicklung der besten Position und Bewertung pro Partikel sehen (Kapitel 2.3.10). In der ersten Iteration ist dies noch nicht möglich, weshalb die beste Bewertung und die beste Position der ersten Bewertung sprich Position entsprechen.

### 2.3.5 Update-Funktion

Anhand der Bewertung des Partikels wird als nächstes die Position des Partikels angepasst. Die Geschwindigkeit definiert diese Veränderung der Position (Formel (2.8)). Dabei wird das Partikel von seiner ehemaligen Geschwindigkeit, seiner eigenen besten Position und der besten Position eines anderen Partikels beeinflusst. Dies ist anhand der drei Terme in Formel (2.9) zu sehen. Von welchem Partikel es dabei gelenkt wird, hängt von der gewählten Nachbarschaft ab (Kapitel 2.3.6). (Clerc, 2012; Yang, 2013)

Die Abbildung 2.4 verdeutlicht die Formeln (2.8) und (2.9) schematisch. Abbildung 2.5 zeigt auf, wie sich der Schwarm verhält.

$$p_{k+1} = p_k + v_{k+1} \quad (2.8)$$

mit

$p_{k+1}$  neue Position des Partikels

$p_k$  aktuelle Position des Partikels

$v_{k+1}$  neue Geschwindigkeit des Partikels

welche folgendermassen bestimmt wird

$$v_{k+1} = \underbrace{w \cdot v_k}_{\text{Trägheit}} + \underbrace{c_1 \cdot r_1 \cdot (p_{best} - p_k)}_{\text{Erfahrung des Partikels}} + \underbrace{c_2 \cdot r_2 \cdot (p_n - p_k)}_{\text{Einfluss des Schwarms}} \quad (2.9)$$

wobei

$w$  Trägheitsfaktor

$v_k$  aktuelle Geschwindigkeit des Partikels

$c_1$  Cognitive Parameter

$c_2$  Social Parameter

$r_1$  Zufallszahl im Intervall [0, 1]

$r_2$  Zufallszahl im Intervall [0, 1]

$p_{best}$  beste Position des Partikels

$p_n$  beste Position in der Nachbarschaft des Partikels

Der Trägheitsfaktor sowie der Cognitive und Social Parameter werden Hyperparameter genannt (Yang, 2013) und im Kapitel 2.3.7 genauer behandelt.

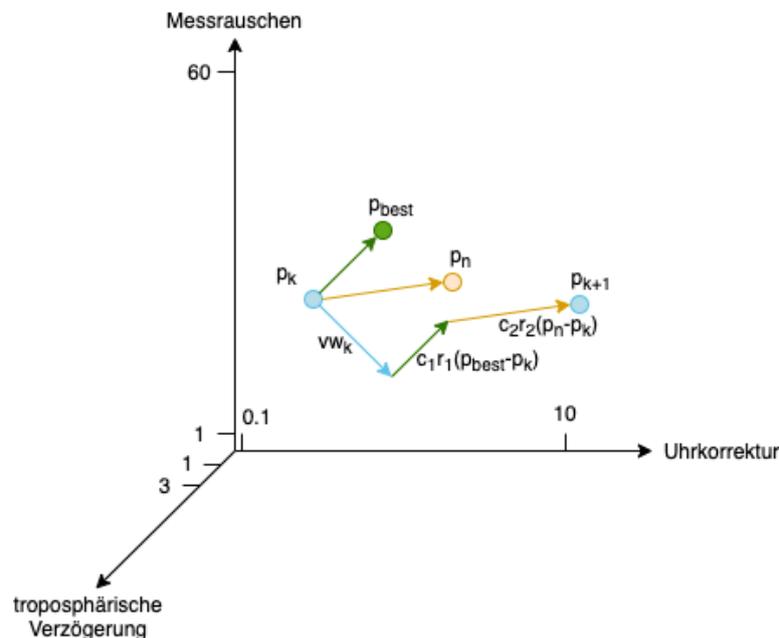


Abbildung 2.4: Anpassung der Position eines Partikels anhand seiner besten Position und der besten Position eines anderen Partikels (eigene Darstellung)

Damit die Geschwindigkeit nicht massiv anwächst, hat sich gezeigt, dass es sich lohnt, die Geschwindigkeit wie die initiale Geschwindigkeit mit  $v_{max}$  zu begrenzen. Es wird wie in Formel (2.4) vorgegangen, wobei die initiale Geschwindigkeit  $v(0)$  durch die neue Geschwindigkeit  $v_{k+1}$  ersetzt wird.

Wenn eine Dimension der Position nach der Anpassung mit der Geschwindigkeit negative Werte aufweist, wird sowohl die Position wie auch die Geschwindigkeit in dieser Dimension auf 0 gesetzt. Dies liegt an der physikalischen Natur der Simulationsparameter, die positive Werte voraussetzt. Die Geschwindigkeit wird ebenfalls auf 0 gesetzt, damit sich die Position erholen kann und in der nächsten Iteration nicht noch mehr ins Negative gezogen wird.

Wie die beste Bewertung und die beste Position des Partikels werden auch die Geschwindigkeit und die Position gespeichert, damit deren Verlauf aufgezeigt werden kann (Kapitel 2.3.10).

Da die Position durch die Simulationsparameter charakterisiert wird, ist die beschriebene Änderung der Position mit der Verbesserung der Simulationsparameter gleichzusetzen. Folglich können die verbesserten Simulationsparameter wieder in die Simulationssoftware eingelesen werden, um so deren Qualität zu bestimmen. Das heisst, der Algorithmus beginnt die nächste Iteration, indem zuerst die Run-Funktion (Kapitel 2.3.3) aufgerufen wird. Der Algorithmus läuft wie oben beschrieben weiter, bis er das Stoppkriterium erreicht (Kapitel 2.3.8, Abbildung 2.2).

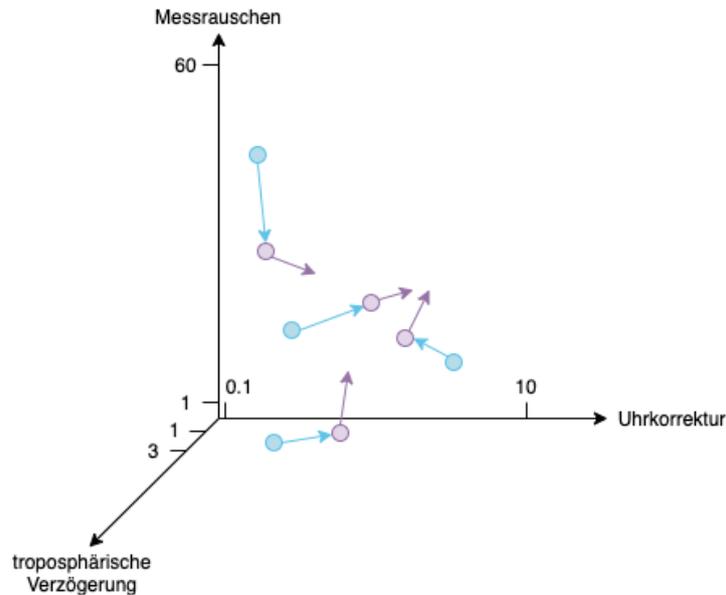


Abbildung 2.5: Verhalten des Schwarms, zwei Iterationen in blau und violett (eigene Darstellung)

### 2.3.6 Nachbarschaft

Die Nachbarschaft, auch Topologie genannt (Clerc, 2012), bestimmt, welche Partikel sich gegenseitig informieren und wie die Interaktion innerhalb des Schwarms geregelt ist. Sie definiert, welches Partikel im letzten Term der Formel (2.9) für den Einfluss des Schwarms eingesetzt wird. In der Literatur werden diverse Nachbarschaftsbeziehungen vorgeschlagen, welche implementiert und auf ihre Anwendbarkeit zur Optimierung der Simulationsparameter getestet worden sind. Welche Nachbarschaft bei einem Durchlauf des Algorithmus verwendet wird, kann selbst bestimmt werden, indem das entsprechende Parser-Argument angepasst wird (Python Software Foundation, 2021a).

#### Globale Topologie

Die globale Nachbarschaft ist simpel und intuitiv. Das Partikel, das von allen die niedrigsten Kosten hat, wird als Nachbar von jedem Partikel definiert und in Formel (2.9) eingesetzt. Das heisst, alle Partikel haben denselben Nachbarn. Dies hat den Vorteil, dass jedes Partikel das derzeit beste Partikel imitiert und sich so schnell dem Optimum annähert. Dies kann aber auch ein Nachteil sein, weil der Algorithmus so eher zu lokalen als globalen Optima konvergiert. (Kennedy et al., 2001)

Die globale Topologie bewirkt von allen Nachbarschaften aufgrund ihrer Definition die stärkste Regularisierung.

#### Adaptive Random Topologie

Bei der Adaptive Random Topologie werden anfangs pro Partikel zufällig drei Partikel als Nachbarn definiert. Dabei spielt es keine Rolle, wenn ein Partikel zwei Mal denselben Nachbarn oder gar sich selbst als Nachbarn hat. Die Eignungen aller Nachbarn werden verglichen und derjenige Nachbar, der die niedrigsten Kosten hat, wird als definitiver Nachbar dieser Iteration festgelegt. Dessen Position wird dann in Formel (2.9) verwendet. (Clerc, 2012)

Eine Besonderheit dieser Nachbarschaftsbeziehung ist, dass sich die Nachbarschaft, wie der Name schon sagt, im Verlaufe des Algorithmus ändern kann. Dies geschieht, wenn sich die

Bewertung des besten Nachbarn nicht verbessert hat und die Iteration daher als nicht erfolgreich gilt (Clerc, 2012). Maximal neun Mal wird eine neue, zufällige Nachbarschaft generiert und überprüft, ob sich so eine erfolgreiche Iteration feststellen lässt (Kwok et al., 2007).

Diese Nachbarschaft wird als Standard-Nachbarschaft definiert.

### Ring-Topologie

Bei der Anwendung der Ring-Topologie wird ein Partikel jeweils von denjenigen Partikeln informiert, die gemäss der Nummerierung am nächsten sind. Partikel 8 wird so beispielsweise von Partikel 7, 8 oder 9 beeinflusst. Welches dabei gewählt wird, ist wiederum von den niedrigsten Kosten abhängig. Dieses wird dann in Formel (2.9) verwendet. Wichtig ist dabei zu beachten, dass das erste Partikel und das letzte Partikel Spezialfälle sind. Das erste Partikel wird nämlich von Partikel 0 und dem letzten, hier Partikel 15 und das letzte Partikel vom ersten und vom vorletzten, hier Partikel 14 geprägt. Abbildung 2.6 verdeutlicht die Topologie. (Clerc, 2012; Kennedy et al., 2001)

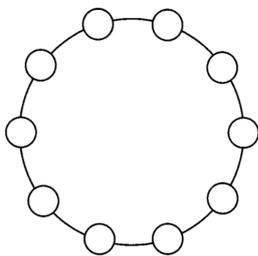


Abbildung 2.6: Ring-Topologie (Kennedy et al., 2001)

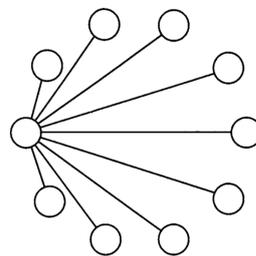


Abbildung 2.7: Wheel-Topologie (Kennedy et al., 2001)

### Wheel-Topologie

Bei der Wheel-Nachbarschaft wird ein Partikel vom besten Partikel, dem global Besten beeinflusst. Hier ist dies gemäss Definition Partikel 0. Es könnte aber theoretisch auch jedes andere Partikel gewählt werden. Dieses Partikel informiert alle übrigen Partikel, indem es seine beste Position weitergibt. Das heisst, in der Formel (2.9) wird für das Partikel 0 das global Beste als Nachbarn eingesetzt und für alle übrigen die beste Position von Partikel 0. Zur Veranschaulichung dient Abbildung 2.7. (Kennedy et al., 2001)

#### 2.3.7 Hyperparameter

Die Hyperparameter sind die «Stellschrauben» jedes metaheuristischen Algorithmus. Ihre Wahl ist sehr komplex und bildet daher oft ein eigenes Optimierungsproblem. Der PSO hat drei Hyperparameter; namentlich den *Trägheitsfaktor*, den *Social Parameter* und den *Cognitive Parameter*. (Yang, 2013)

### Trägheitsfaktor

Der Trägheitsfaktor  $w$  bewirkt eine Dämpfung des Einflusses der alten Geschwindigkeit (Formel (2.9)). So werden die Partikel automatisch langsamer, je länger der Algorithmus läuft, weil davon ausgegangen wird, dass sie sich je länger je mehr dem Optimum annähern. Er nimmt abhängig von den maximal erlaubten Iterationen von 0.9 bis 0.4 linear ab (Kapitel 2.3.8). (Kennedy et al., 2001)

## Cognitive Parameter und Social Parameter

Der Cognitive Parameter skaliert, wie stark die eigene Erfahrung des Partikels in der Formel (2.9) miteinfließt. Im Gegensatz dazu bestimmt der Social Parameter den Einfluss des Schwarms auf das Partikel. In der Literatur werden die Parameter auf denselben Wert, in der Nähe von 2 gesetzt. Hier bekommen beide den Wert 1. Die Resultate der Arbeit zeigen, dass dies besser passt (Kapitel 4.7 und Kapitel 5.6). Zusätzlich werden beide Parameter mit einer Zufallszahl gewichtet, da nicht bekannt ist, ob die individuelle oder soziale Komponente wichtiger ist. Die zufällige Gewichtung erlaubt, damit zu spielen. (Kennedy et al., 2001; Yang, 2013)

### 2.3.8 Stoppkriterium

Bei einem iterativen Algorithmus wie dem PSO ist das Stoppkriterium zentral, damit der Algorithmus nicht endlos weiterläuft. In der Literatur werden kaum Kriterien für den PSO vorgeschlagen (Clerc, 2012; Kennedy et al., 2001; Yang, 2013). Daher wird die global beste Bewertung betrachtet und untersucht, ob sich diese im Vergleich zur letzten Iteration um weniger als 5% verändert hat. Wenn dies in neun aufeinanderfolgenden Iterationen gegeben ist, wird der Algorithmus terminiert (Kwok et al., 2007). Dabei müssen Kosten, die genau 0 sind, gesondert behandelt werden, damit es keine Division durch 0 gibt. Tritt dies ein, zählt diese Iteration ebenfalls zu den neun Relevanten, weil Kosten von 0 optimal sind.

Für den Fall, dass dieses Stoppkriterium nicht greift, gibt es eine weitere Bedingung. Wenn der Algorithmus nach einer definierten Anzahl maximaler Iterationen die 9-mal-5%-Regelung nicht erfüllt, wird er automatisch gestoppt. Hier wird die maximale Anzahl Iterationen auf 100 gesetzt. Gemäss den Resultaten dieser Arbeit sollten nie mehr Iterationen benötigt werden (Kapitel 4).

### 2.3.9 Speicherung

Alle Informationen, die für die Interpretation und Visualisierung der Ergebnisse benötigt werden, werden für jedes Partikel einzeln gespeichert. Das Python Package pickle wird zur Serialisierung verwendet (Python Software Foundation, 2021b). So können die Informationen im Python-Skript, welches die Plots erstellt, einfach wieder eingelesen werden.

Die finalen Werte der EOP werden separat noch in eine csv-Datei gespeichert, damit die genauen Werte auch ohne Python-Skripte gelesen und verglichen werden können.

Wenn für alle Stationen eigene Parameter verwendet werden, wird zusätzliche eine csv-Datei erstellt, in welcher pro Station die global beste Position gespeichert wird (Kapitel 4.5).

### 2.3.10 Plots

Zur Erstellung der Plots wird das Python-Package matplotlib verwendet (The Matplotlib development team, 2021). Die Plot-Funktion kann unabhängig vom PSO laufen, weil sie in einem externen Python-Skript gecoded ist. Dies hat den Vorteil, dass Änderungen im Code sofort umgesetzt und angeschaut werden können, ohne dass der PSO erneut gestartet werden muss. Es kann dabei aber nur auf die zuletzt erstellten Ergebnisse des PSO zurückgegriffen werden.

Für jeden Durchlauf des Algorithmus wird pro Partikel ein Plot erstellt, welcher die simulierten EOP und die Referenzwerte aufzeigt. Pro EOP wird ein Subplot erstellt, wobei auf der x-Achse

die Iterationen und auf der y-Achse die Werte eingetragen werden. Die Abkürzungen der Achsenbeschriftungen können im Abkürzungsverzeichnis nachgeschaut werden. Dieser Plot kann aufzeigen, wie nahe man mithilfe der vom PSO generierten Simulationsparameter an die Referenzwerten kommt (bspw. Abbildung 4.3).

Wenn für alle Stationen die gleichen Parameter angenommen werden und das Problem daher dreidimensional ist, können zudem Plots erstellt werden, in welchen die Entwicklung der Simulationsparameter ersichtlich ist. Drei Subplots zeigen dabei den Verlauf der Position, drei den Verlauf der Geschwindigkeit und einer den Verlauf der Kosten. Es werden für die Position und die Geschwindigkeit drei Subplots benötigt, weil jeder eine Dimension abbildet. Damit nicht nur die Position des Partikels, sondern auch die beste Position dieses Partikels, die Position des Nachbarn und die global beste Position sichtbar sind, werden diese in verschiedenen Farben im selben Subplot abgebildet. Im Subplot, der den Verlauf der Kosten aufzeigt, wird der Vollständigkeit halber die global beste Position noch ausgeschrieben aufgeführt. Die Plots werden wieder pro Partikel generiert und die x-Achse zeigt die Iterationen und die y-Achse die Werte an. Mit diesem Plot kann gezeigt werden, wie der PSO funktioniert und wie sich die Kosten, die Position und die Geschwindigkeit verhalten (bspw. Abbildung 4.2).

### 3 Daten

Für die Simulationen werden Daten aus der CONT17-Kampagne (Continuous VLBI Campaign 2017) des International VLBI Service for Geodesy and Astronomy (IVS) verwendet. Die Kampagne beobachtete vom 28. November 2017 bis am 12. Dezember 2017 kontinuierlich extragalaktische Radioquellen. Die Teleskope, welche dabei dieselben Radioquellen beobachteten und daher Basislinien bildeten, wurden in sogenannten Netzwerken zusammengefasst. Insgesamt wurde mit drei unabhängigen Netzwerken von VLBI-Stationen gemessen, wovon das dritte Netzwerk fünf Tage lang ununterbrochen aktiv war. Dies ist eine Besonderheit, weil VLBI-Messungen normalerweise nicht kontinuierlich sind, sondern wegen den zurzeit zur Verfügung stehenden Instrumenten maximal 24 Stunden dauern. Messungen finden üblicherweise auch nicht jeden Tag statt sondern nur an ausgewählten Tagen der Beobachtungswoche. Daher sind die CONT-Kampagnen eine erste Umsetzung der Zukunftsvision von VLBI-Messungen, die 7\*24-Stunden Messungen an 365 Tagen vorsieht und unter dem Namen VLBI Global Observing System (VGOS) bekannt ist. (Behrend et al., 2020)

Eine weitere Eigenart der CONT17-Kampagne ist, dass mehr als nur ein Netzwerk für die Messungen verwendet worden ist. Die Messung mit mehreren Netzwerken ermöglicht den Vergleich der Resultate und somit die Überprüfung der Genauigkeit der EOP. Zudem können Abhängigkeiten aufgrund der Netzwerkwahl untersucht werden. (Behrend et al., 2020)

Jeder Tag in jedem Netzwerk bildet eine Session. So wird beispielsweise der erste Tag im ersten Netzwerk Session c1701 genannt. Pro Session sind alle wichtigen Informationen in einem sogenannten skd-File abgespeichert. (Behrend et al., 2020)

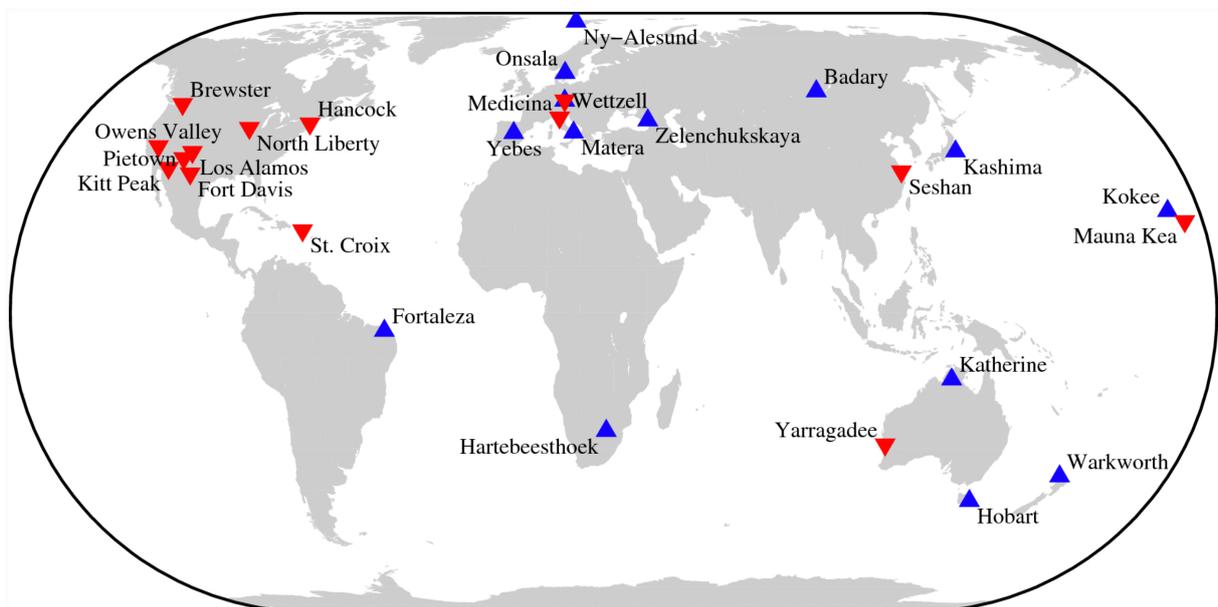


Abbildung 3.1: Messstationen ersten beiden Netzwerke der CONT17-Kampagne, Netzwerk 1 in blau, Netzwerk 2 in rot (Behrend et al., 2020)

Für die beiden Netzwerke, die 15 Tage lang gemessen haben, sind je 14 Stationen eingesetzt worden, welche die ältere S/X-Messmethode anwenden. Die Genauigkeit der Beobachtungen hängt dabei neben der Qualität der Radioquellen von der räumlichen Verteilung der Stationen

ab. Die Wahl der Stationen wiederum ist abhängig von den erwarteten EOP und betrieblichen Einschränkungen (Abbildung 3.1). (Behrend et al., 2020)

Das dritte Netzwerk wurde nur vom 4. Dezember 2017 bis am 8. Dezember 2017 eingesetzt und diente zur Demonstration der zukünftigen Messmethode VGOS. Sechs Stationen, welche die technischen Anforderungen erfüllten, bildeten dieses Netzwerk (Abbildung 3.2). (Behrend et al., 2020)

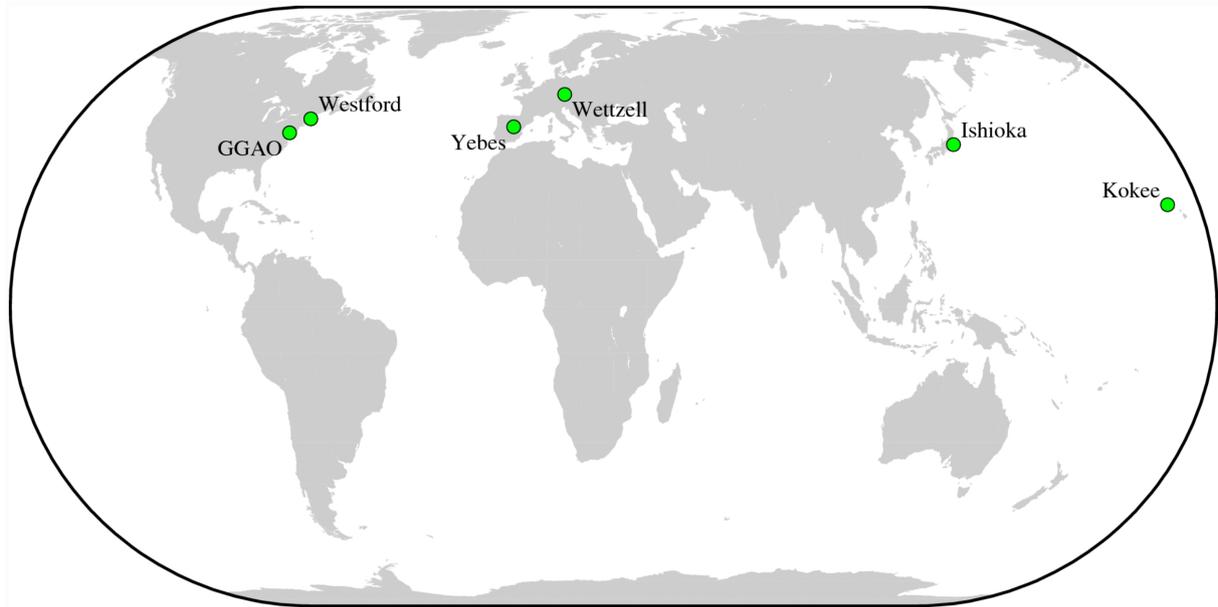


Abbildung 3.2: Messstationen des dritten Netzwerkes der CONT17-Kampagne (Behrend et al., 2020)

## 4 Resultate

Im folgenden Kapitel werden die Resultate präsentiert, die sich mittels des geschilderten Algorithmus ergeben. Diese sind prinzipiell reproduzierbar, indem der Python-Code laufen gelassen wird. Aufgrund diverser zufälliger Einflüsse, beispielsweise in der Initialisierung, ergeben sich jedoch bei jedem Durchlauf etwas andere Ergebnisse. So erhält auch jedes Partikel etwas andere Resultate.

Wenn nicht anders vermerkt, gelten diese Resultate für Simulationen des ersten Netzwerkes am ersten Tag, sprich Session c1701 (Kapitel 3), wobei für alle Stationen dieselben Parameter angenommen werden.

### 4.1 Baseline Lösung

Die Baseline Lösung gilt als Referenzlösung, da sie für die Simulationsparameter Standardwerte verwendet. Das heisst, die troposphärische Verzögerung wird auf  $1.8 \times 10^{-7} m^{-1/3}$ , der Uhrfehler auf  $1 \times 10^{-14} s$  und das Messrauschen auf  $17.68 ps$  gesetzt (M. Schartner, persönliche Kommunikation, 2021). Zudem wird die Anzahl maximaler Iterationen auf 1 gesetzt, um die Genauigkeit der Standardwerte zu sehen.

Mit Hilfe dieser Lösung kann aufgezeigt werden, ob mittels PSO bessere Ergebnisse als mit Standardwerten erzielt werden können. Dazu werden Säulendiagramme erstellt, die die Ergebnisse aus den Kapiteln 4.2 und 4.3 der Baseline Lösung und den Referenzwerten gegenüberstellen (Abbildung 4.1). Wichtig zu beachten ist, dass jeweils nur die Resultate eines Partikels dargestellt werden. Die anderen Partikel eines Durchlaufs verhalten sich jedoch sehr ähnlich.

### 4.2 Standard-Durchlauf

Beim Standard-Durchlauf werden für alle Stationen dieselben Simulationsparameter angenommen. Die Nachbarschaft ist gemäss der Adaptive Random Topologie definiert.

In Abbildung 4.2 wird aufgezeigt, wie sich die drei Eigenschaften (Position, Geschwindigkeit und Eignung) eines Partikels über die Iterationen verändern. Wie in Kapitel 2.3.10 beschrieben, erhält dabei jede Dimension einen eigenen Subplot. In verschiedenen Farben werden zudem auch die Ergebnisse des Nachbarpartikels, des global besten Partikels und die besten Eigenschaften des Partikels geplottet. Man kann eine Annäherung an die die global beste Position beobachten (Abbildung 4.2), obwohl anfangs die Simulationsparameter für die troposphärische Verzögerung und den Uhrfehler weit vom global besten Wert entfernt sind. Bei genauem Hinschauen fällt auf, dass teilweise in einer Dimension der Position der momentane Wert des Partikels besser ist als der beste Wert des Partikels (Abbildung 4.2 in der Dimension der troposphärischen Verzögerung bei Iteration 5). Die Art der Anpassung des besten Werts des Partikels verursacht diese Beobachtung. Denn jene geschieht über die Kosten der Partikel, die auf den simulierten EOP basieren, und betrachtet nicht den Wert der einzelnen Dimensionen (Kapitel 2.3.4).

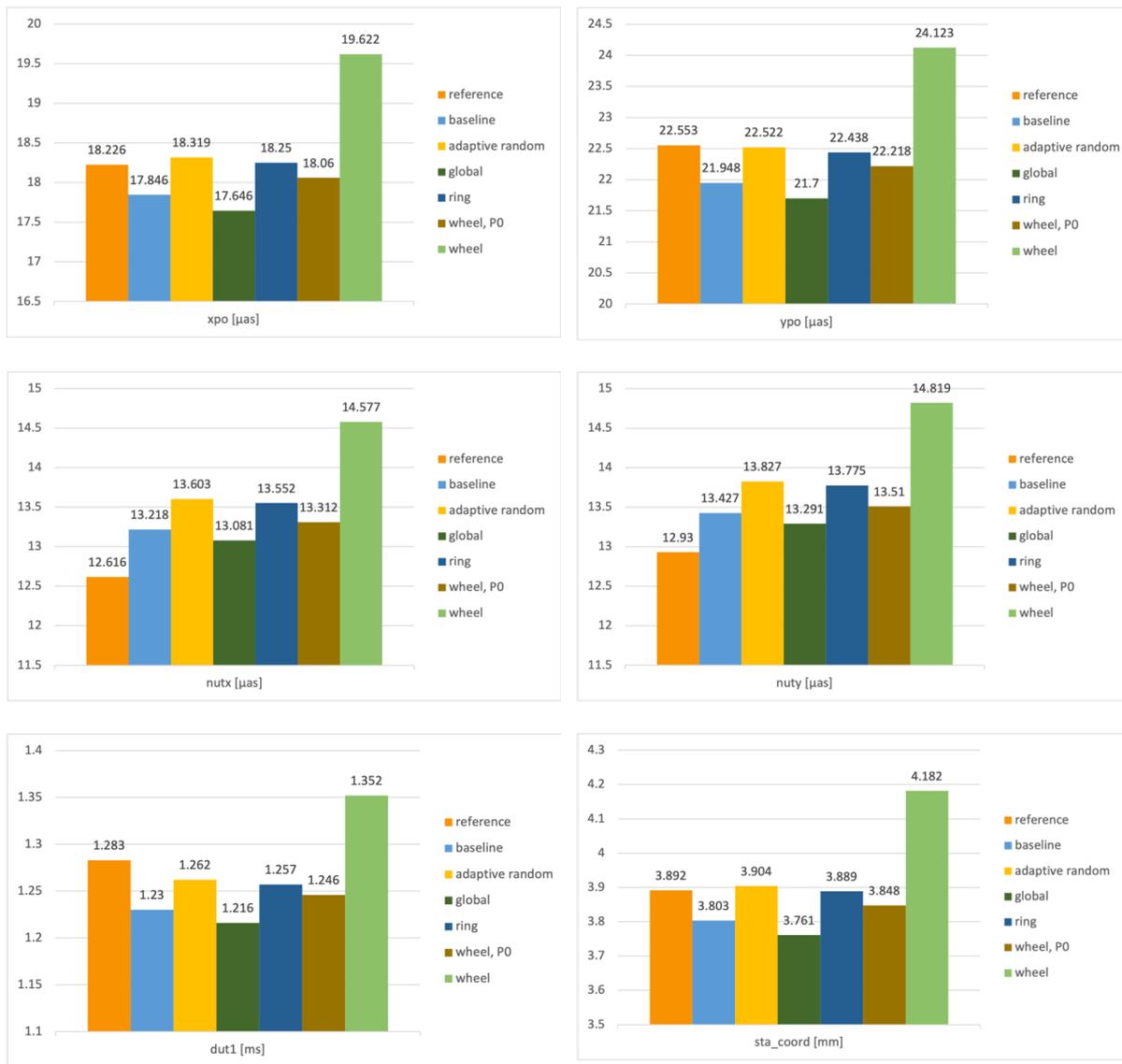


Abbildung 4.1: Vergleich der EOP der Baseline Lösung und der Nachbarschaften zu den Referenzwerten (eigene Darstellung)

Abbildung 4.3 stellt die Annäherung der simulierten EOP an die Referenzwerte dar und zeigt, dass zu Beginn die simulierten Genauigkeiten schlechter sind als die Realität. Über die nächsten Iterationen passen sich die Werte an und nähern sich den Referenzwerten. Es fällt auf, dass für alle EOP eines Partikels die gleiche Kurve mit einer etwas anderen Skalierung geplottet ist. Dies gilt nicht nur für den Standard-Durchlauf, sondern auch für alle anderen getesteten Fällen. Dies liegt daran, dass die Werte durch die Simulation skaliert werden und auf denselben Beobachtungen beruhen.

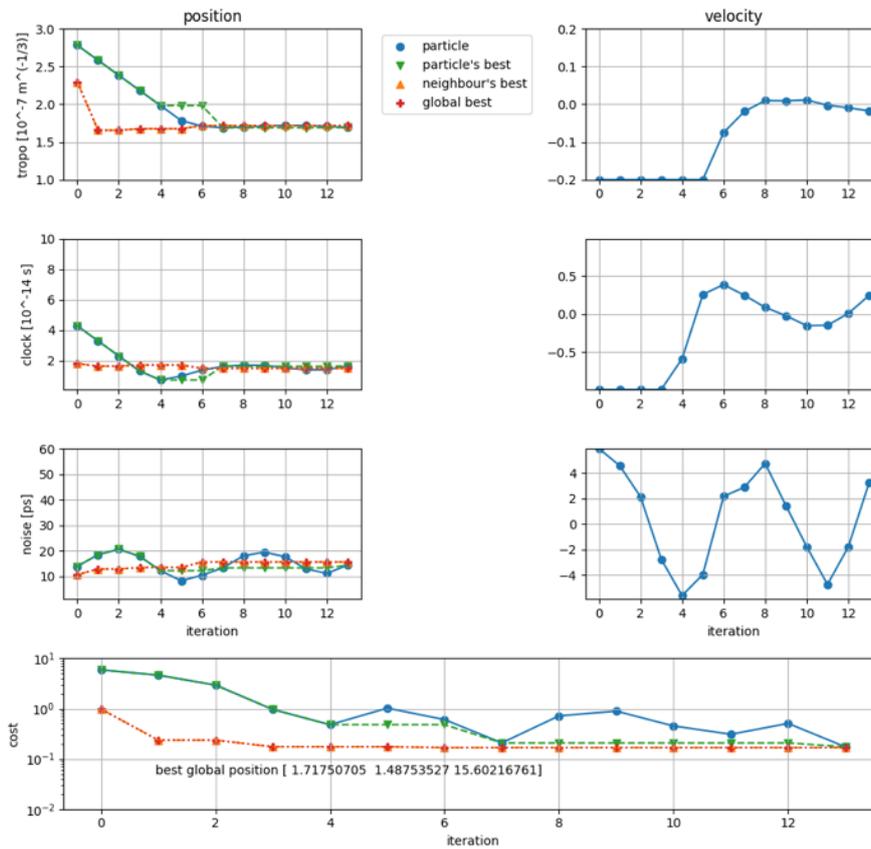


Abbildung 4.2: Position, Geschwindigkeit, Kosten eines Partikels beim Standard-Durchlauf (eigene Darstellung)

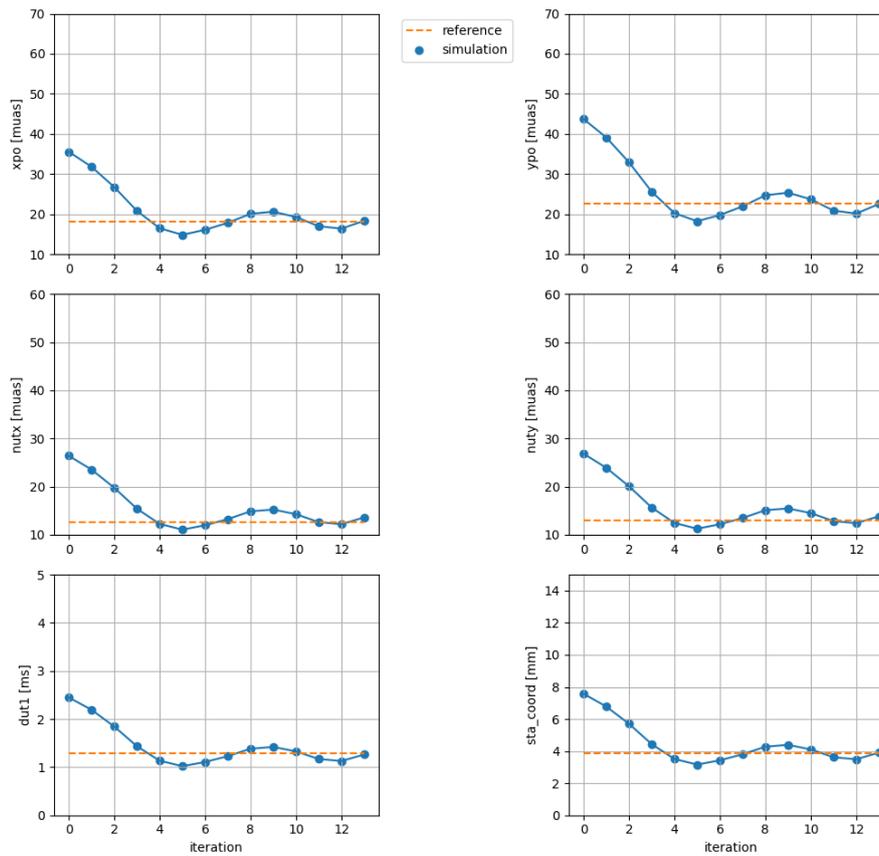


Abbildung 4.3: EOP eines Partikels im Vergleich zu den Referenzwerten beim Standard-Durchlauf (eigene Darstellung)

### 4.3 Durchlauf mit verschiedenen Nachbarschaften

Für Durchläufe mit verschiedenen Nachbarschaften können ebenfalls die zwei bekannten Plots generiert werden. Aus den Plots, die den Vergleich der simulierten Beobachtungen zu den Referenzwerten zeigen, wird jeweils nur ein Ausschnitt abgebildet. So kann das Verhalten exemplarisch anhand eines EOP gezeigt werden. Die anderen Parameter verhalten sich wie in Kapitel 4.2 beschrieben gleich. Die vollständigen Abbildungen sind im Anhang zu finden.

#### 4.3.1 Globale Topologie

In diesem Durchlauf kann die globale Topologie getestet werden.

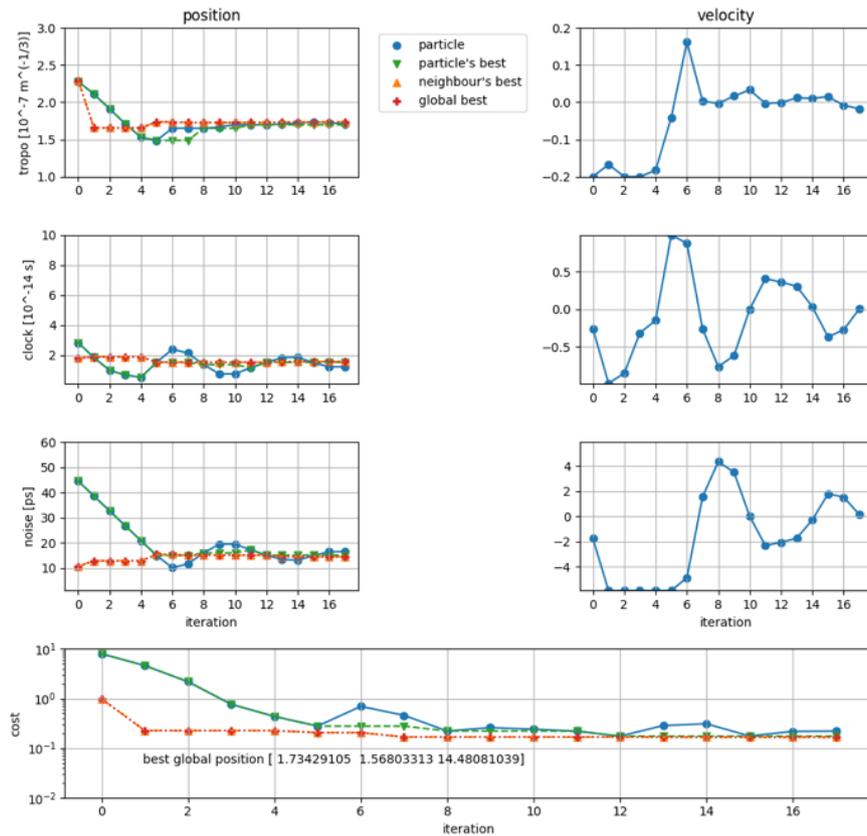


Abbildung 4.4: Position, Geschwindigkeit, Kosten eines Partikels mit der globalen Nachbarschaft (eigene Darstellung)

Bei der globalen Topologie fällt als erstes auf, dass die orange und die rote Kurve in Abbildung 4.4 identisch sind. Dies liegt an der Definition der Topologie, da das Nachbarpartikel dem global besten Partikel entspricht.

Die simulierten Beobachtungen nähern sich schnell und passend an die Referenzwerte an (Abbildung 4.5).

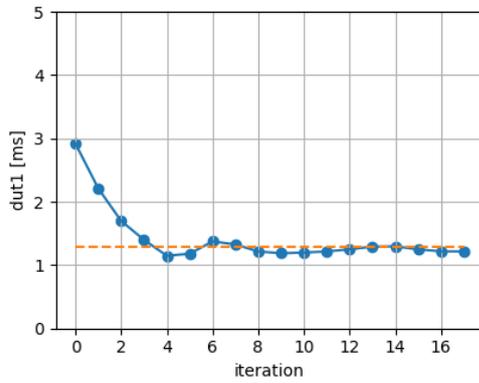


Abbildung 4.5: Annäherung an den Referenzwert DUT1 eines Partikels bei der globalen Topologie (eigene Darstellung)

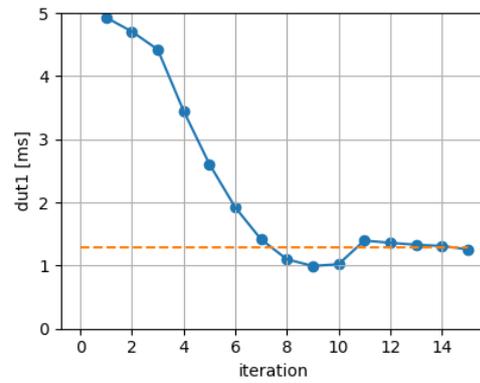


Abbildung 4.6: Annäherung an den Referenzwert DUT1 eines Partikels bei der Ring-Topologie (eigene Darstellung)

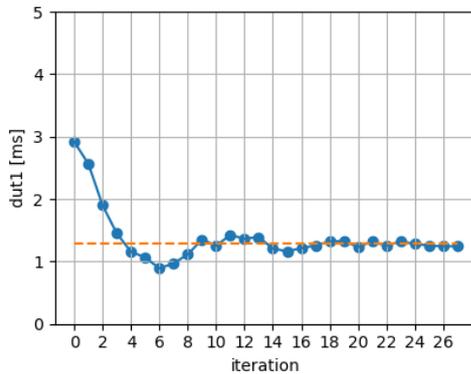


Abbildung 4.7: Annäherung an den Referenzwert DUT1 des Partikel 0 bei der Wheel-Topologie (eigene Darstellung)

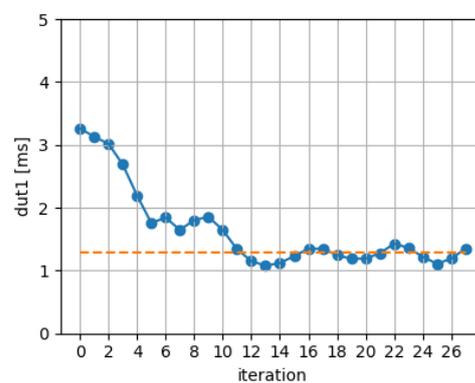


Abbildung 4.8: Annäherung an den Referenzwert DUT1 eines anderen Partikels bei der Wheel-Topologie (eigene Darstellung)

### 4.3.2 Ring Topologie

In einem weiteren Durchlauf wird die Ring-Nachbarschaft untersucht. Abbildung 4.6 und Abbildung 4.9 verdeutlichen wiederum das Verhalten des Algorithmus.

Die Ring-Nachbarschaft zeigt eine schlechtere Annäherung der Position des Partikels an die global beste Position (Abbildung 4.9). Dennoch werden die Referenzwerte gut angenähert (Abbildung 4.1) und anfangs schlechte EOP können korrigiert werden (Abbildung 4.6).

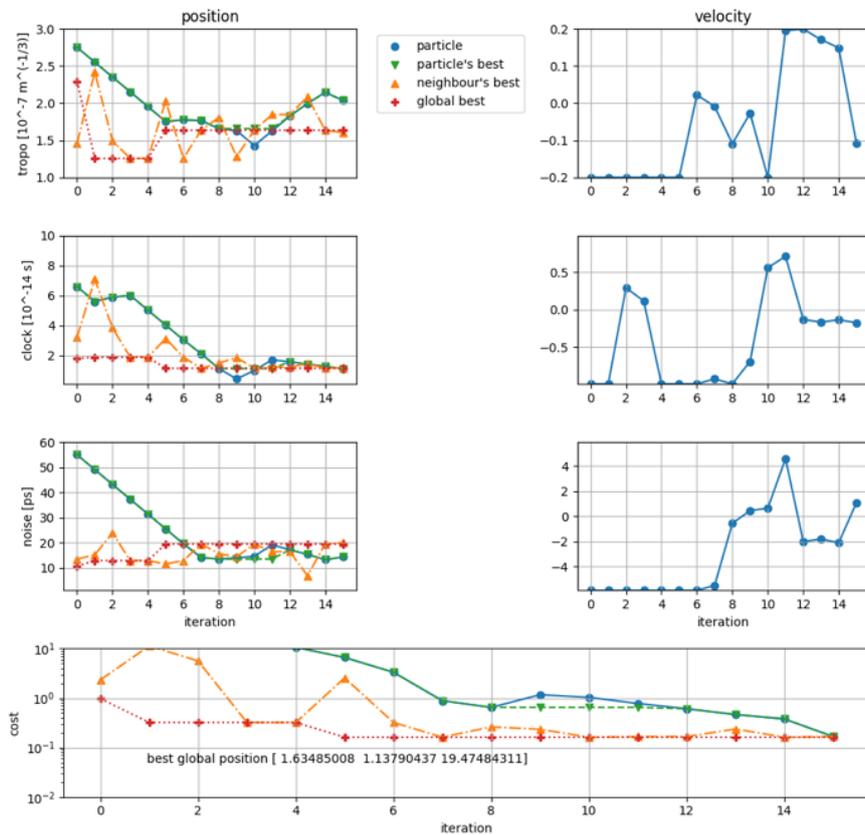


Abbildung 4.9: Position, Geschwindigkeit, Kosten eines Partikels mit der Ring-Nachbarschaft (eigene Darstellung)

### 4.3.3 Wheel-Topologie

Bei der Anwendung der Wheel-Topologie muss zwischen dem Verhalten von Partikel 0 und demjenigen der anderen Partikel unterschieden werden.

Zwischen dem informierenden Partikel der Wheel-Topologie (hier Partikel 0) und den anderen Partikeln lassen sich klare Unterschiede aufzeigen. Das Partikel 0 kann sich viel schneller als die übrigen Partikel an die global beste Position und die Referenzwerte anpassen. Diese Trägheit der anderen Partikel kann anhand der Abbildung 4.10 am besten aufgezeigt werden. Zudem ist auch ersichtlich, dass sich die anderen Partikel bis zum Schluss mit grossen Geschwindigkeiten versuchen anzunähern, um diese Trägheit auszugleichen (Abbildung 4.12). Ein Problem der Topologie zeigt sich, wenn die beste Position des Partikels 0 (Abbildung 4.11 in grün) und die Nachbarschaftsposition der anderen Partikel (Abbildung 4.12 in orange) verglichen wird. Gemäss Definition müssten die beiden Kurven identisch sein, weil das Partikel 0 der Nachbar aller anderen Partikel ist. Weshalb dies nicht stimmt, ist unklar.

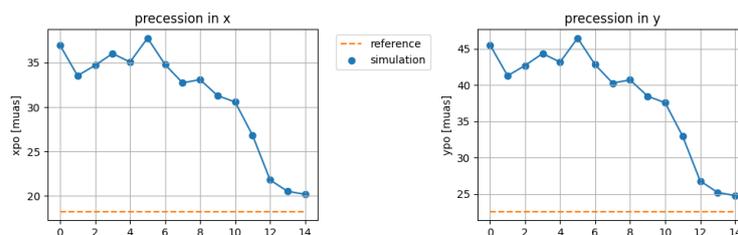


Abbildung 4.10: Trägheit eines Partikels in der Wheel-Topologie (eigene Darstellung)

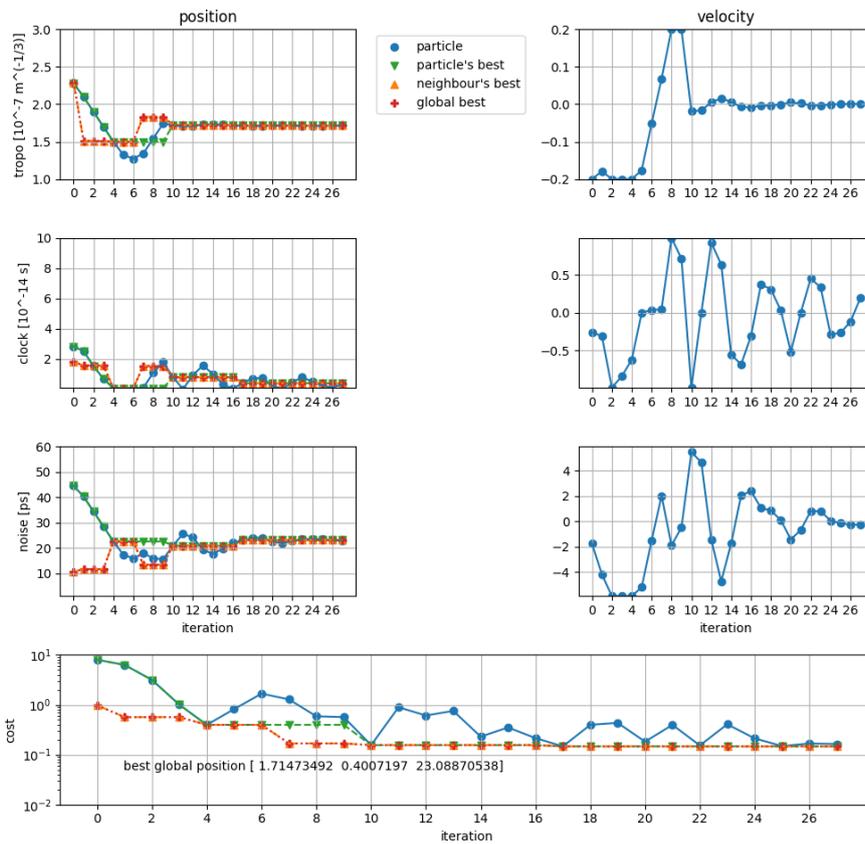


Abbildung 4.11: Position, Geschwindigkeit, Kosten des Partikels 0 bei der Wheel-Topologie (eigene Darstellung)

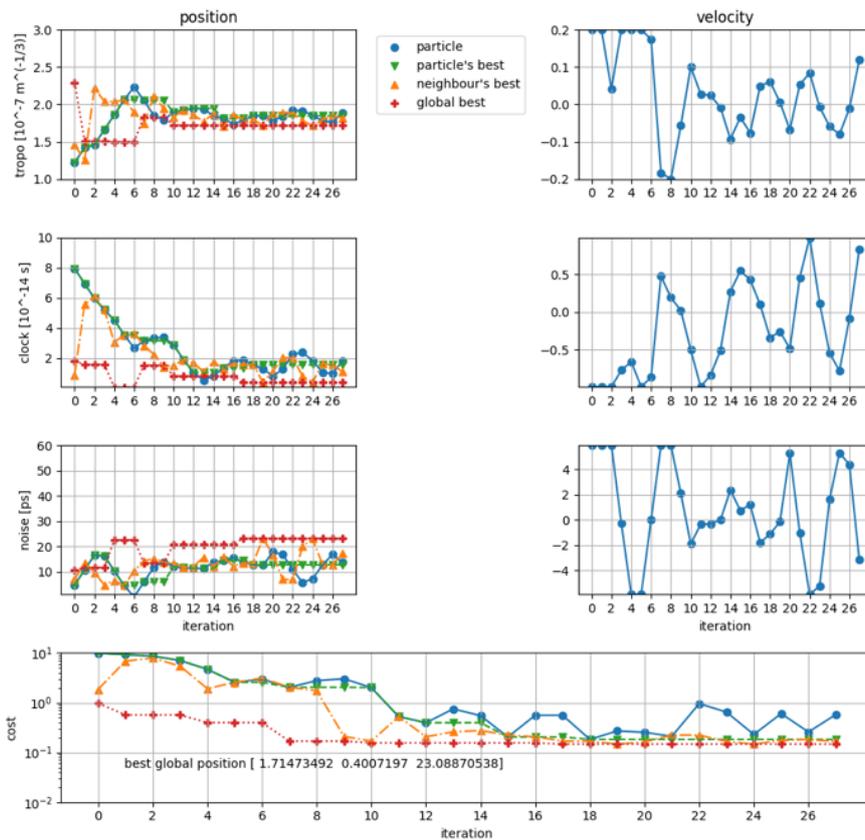


Abbildung 4.12: Position, Geschwindigkeit, Kosten eines anderen Partikels der Wheel-Topologie (eigene Darstellung)

## 4.4 Durchlauf ohne Uhrfehler

Da in einigen Durchläufen der Uhrfehler gegen 0 tendiert (Abbildung 4.13), wird untersucht, wie sich die Ergebnisse verhalten, wenn der Uhrparameter nicht geschätzt wird. Er wird wie in der Baseline Lösung daher auf 1 gesetzt und behält diesen Wert.

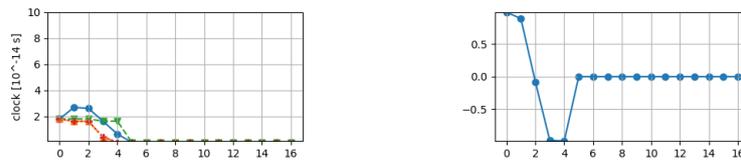


Abbildung 4.13: Uhrfehler wird schnell 0 (eigene Darstellung)

Wenn der Uhrfehler nicht mitgeschätzt wird, können ebenfalls verschiedene Nachbarschaften ausprobiert werden. Beispielhaft wird hier die Adaptive Random und die globale Topologie gezeigt. Es entstehen die bereits bekannten Plots, wobei bei der Annäherung an die Referenzwerte wiederum nur exemplarisch das Verhalten für den Parameter DUT1 gezeigt wird (Abbildung 4.14, Abbildung 4.15, Abbildung 4.16, Abbildung 4.17). Im Anhang befinden sich die vollständigen Plots. Die Annäherung an die Referenzwerte kann mittels Säulendiagrammen noch besser quantitativ betrachtet werden (Abbildung 4.18).

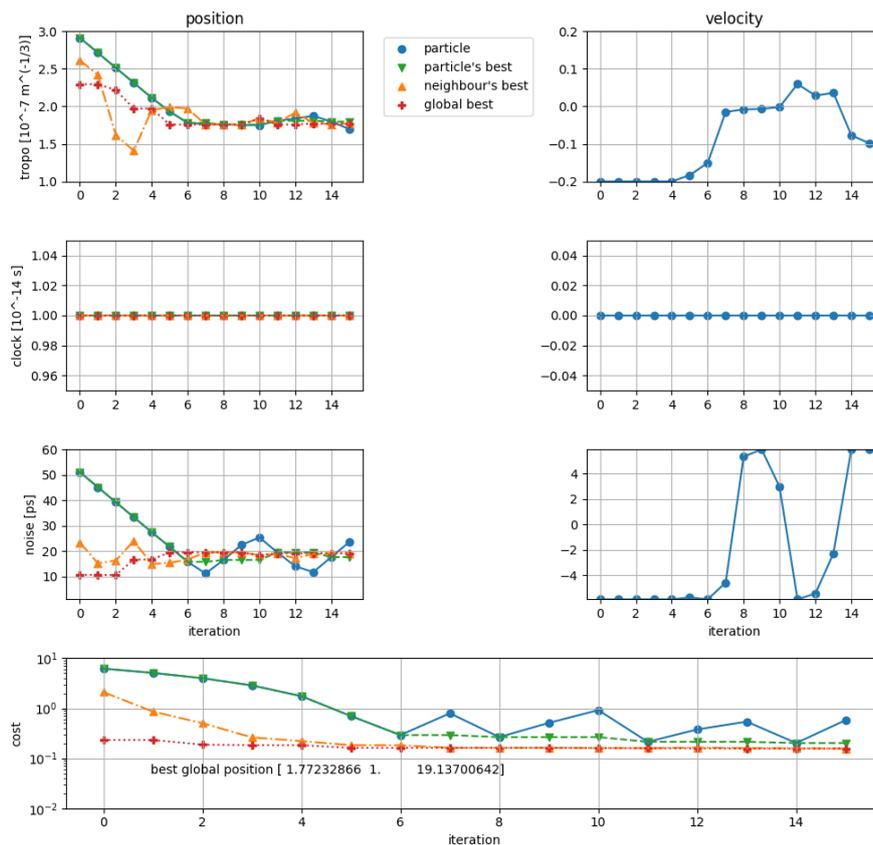


Abbildung 4.14: Position, Geschwindigkeit, Kosten eines Partikels mit der Adaptive Random Topologie ohne Uhrfehler (eigene Darstellung)

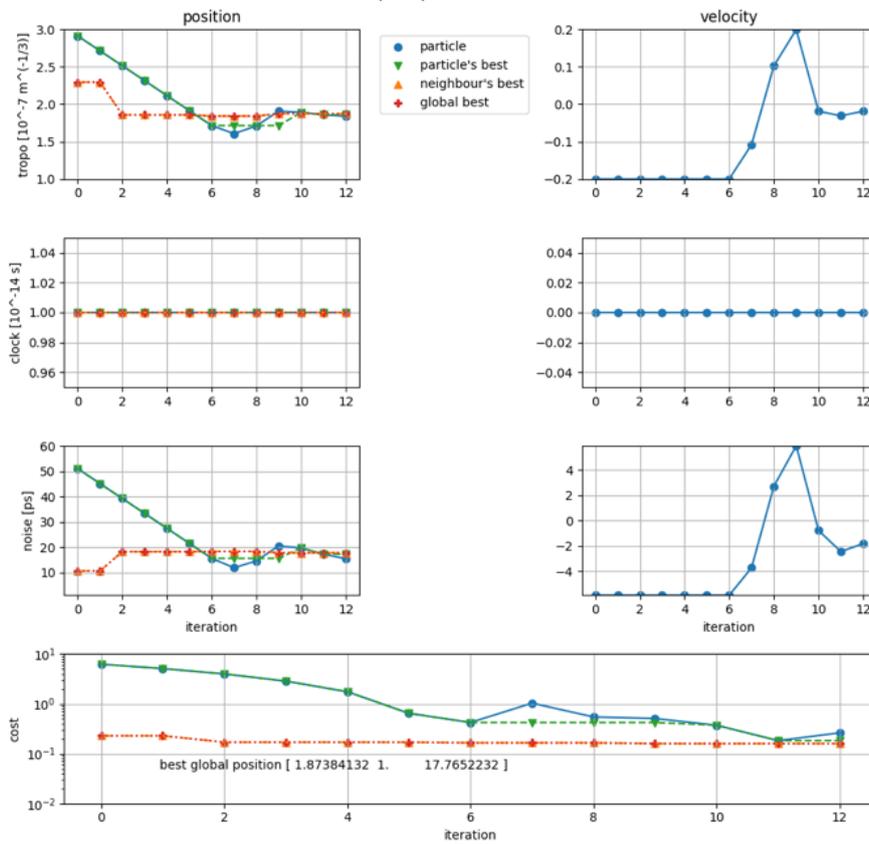


Abbildung 4.15: Position, Geschwindigkeit, Kosten eines Partikels mit der globalen Topologie ohne Uhrfehler (eigene Darstellung)

Betrachtet man Abbildung 4.14 und Abbildung 4.15, sieht man, dass die Geschwindigkeiten anfangs konstant gross sind.

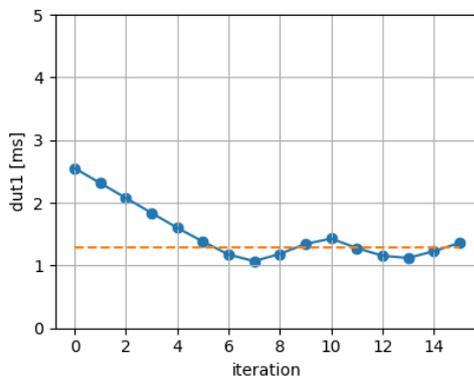


Abbildung 4.16: Annäherung an den Referenzwert DUT1 eines Partikels bei der Adaptive Random Topologie ohne Uhrfehler (eigene Darstellung)

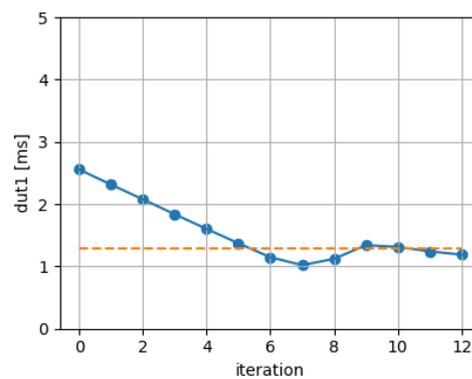


Abbildung 4.17: Annäherung an den Referenzwert DUT1 eines Partikels bei der globalen Topologie ohne Uhrfehler (eigene Darstellung)

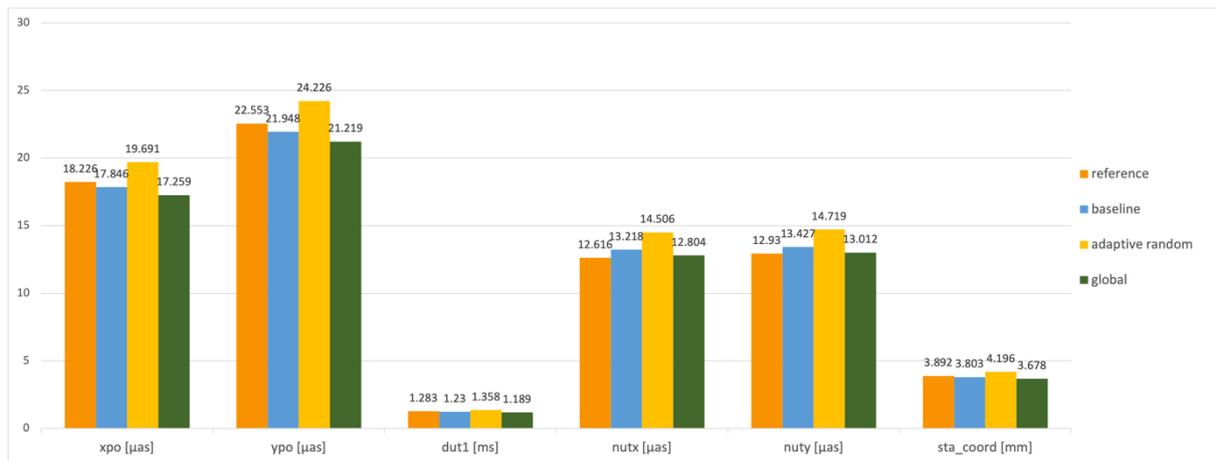


Abbildung 4.18: Vergleich der EOP der Durchläufe ohne Uhrparameter mit der Baseline Lösung und den Referenzwerten (eigene Darstellung)

## 4.5 Durchlauf mit verschiedenen Simulationsparametern für jede Station

Bisher wurde immer ein dreidimensionales Problem betrachtet, da für alle Stationen dieselben Simulationsparameter verwendet worden sind. Der PSO ist auch auf höher-dimensionale Probleme anwendbar. Daher wird im Folgenden getestet, ob es möglich und besser ist, jeder Station eigene Simulationsparameter zuzuordnen und diese zu optimieren. Da das erste Netzwerk mit 14 Stationen untersucht wird, betrachtet man ein 42-dimensionales Optimierungsproblem (vgl. Kapitel 2.3.1, Formel (2.1) und Kapitel 3).

Wenn für die Messstationen verschiedene Simulationsparameter angenommen werden, ergeben sich pro Messstation<sup>1</sup> folgende Ergebnisse (Tabelle 4.1, Tabelle 4.2). Dabei sind beispielhaft die Adaptive Random und die globale Topologie verwendet worden.

Tabelle 4.1: Simulationsparameter pro Station mit der Adaptive Random Topologie

Messstationen	Troposphärische Verzögerung in $10^{-7} m^{-1/3}$	Uhrfehler in $10^{-14} s$	Messrauschen in $ps$
BADARY	2.890	1.896	6.444
FORTLEZA	3.238	0.678	35.423
HART15M	0.816	0.902	3.092
HOBART26	2.317	0	20.635
KASHIM11	2.603	1.484	7.918
KATHI12M	1.385	0	20.924

<sup>1</sup> Die Bezeichnungen der Messstationen basieren auf den technischen Namen aus den skd-Files. Sie sind aber eindeutig den Beschriftungen in Abbildung 3.1 und Abbildung 3.2 zuzuordnen.

KOKEE	0.601	0.437	5.963
MATERA	0.785	2.006	46.204
NYALES20	1.030	0.088	12.360
ONSALA60	1.840	0	5.937
WARK12M	1.816	1.181	26.480
WETTZELL	1.522	0.598	2.393
YEBES40M	0.821	0.738	39.835
ZELENCHK	1.652	0.634	13.499

Tabelle 4.2: Simulationsparameter pro Station mit der globalen Topologie

Messstationen	Troposphärische Verzögerung in $10^{-7} m^{-1/3}$	Uhrfehler in $10^{-14} s$	Messrauschen in $ps$
BADARY	1.963	3.346	14.981
FORTLEZA	2.390	3.106	2.928
HART15M	0.467	0.100	4.002
HOBART26	1.458	1.715	1.762
KASHIM11	1.490	2.556	21.802
KATHI12M	4.174	2.597	8.614
KOKEE	0.779	0	4.839
MATERA	2.332	0	22.039
NYALES20	3.027	1.169	3.445
ONSALA60	3.602	8.210	72.542
WARK12M	1.446	3.552	25.070
WETTZELL	2.723	0.365	47.416
YEBES40M	1.570	2.360	16.598
ZELENCHK	1.229	1.101	25.049

Die resultierenden, optimalen Simulationsparameter liegen nicht immer im erwarteten Wertebereich. So ergeben sich beispielsweise fürs Messrauschen hohe Werte wie 72.5 ps, aber auch viel zu niedrige Werte in der Nähe von 0. Ebenfalls tritt das bekannte Phänomen ein, dass der Uhrfehler teilweise auf 0 gesetzt wird.

Ausschnitte der Annäherung an die Referenzwerte sind in Abbildung 4.19 und Abbildung 4.20 gegeben. Wie erwähnt, sind im Anhang die vollständigen Plots zu sehen. Weil es sich um ein höher-dimensionales Problem handelt, kann der Verlauf der Position und der Geschwindigkeit nicht mehr dargestellt werden.

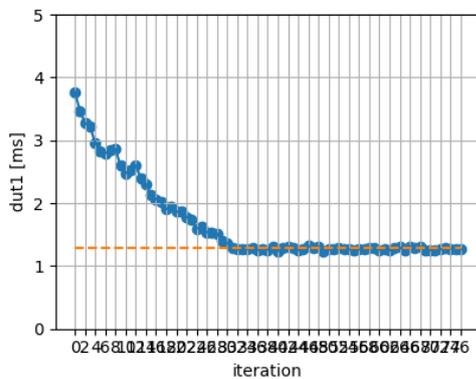


Abbildung 4.19: Annäherung an den Referenzwert DUT1 bei der Adaptive Random Topologie, wenn für jede Station individuelle Simulationsparameter bestimmt werden (eigene Darstellung)

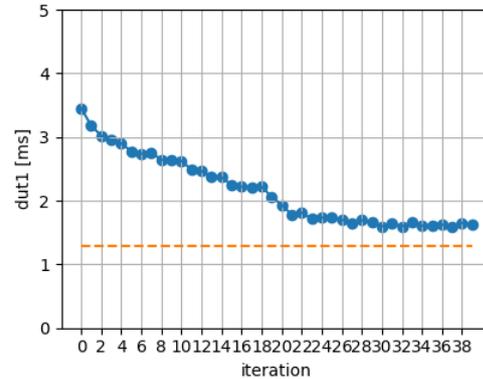


Abbildung 4.20: Annäherung an den Referenzwert DUT1 bei der globalen Topologie, wenn für alle Stationen individuelle Simulationsparameter bestimmt werden (eigene Darstellung)

Es zeigt sich, dass die Adaptive Random Topologie die Referenzwerte annähern kann, wenn verschiedene Stationsparameter angenommen werden. Dazu sind jedoch viele Iterationen, 76 bei der Zahl, nötig. Bei der globalen Topologie erreicht der PSO keine vergleichbar gute Annäherung.

Damit die Approximation auch für die globale Topologie funktioniert, sind verschiedene Anpassungen des Algorithmus ausprobiert worden. Zum einen ist getestet worden, wie sich der Algorithmus verhält, wenn die Hyperparameter, Social und Cognitive Parameter, auf 2 gesetzt werden. Dies ermöglicht den Partikeln, höhere Geschwindigkeiten zu erreichen und so gegebenenfalls näher an die Referenzwerte zu kommen. Ebenfalls ist überprüft worden, ob ein strengeres Stoppkriterium bessere Ergebnisse erlaubt, weil der Algorithmus länger läuft. Dafür ist die 5%-Schranke auf 2% herabgesetzt worden. Beide Anpassungen sind wirkungslos. Eine Aufhebung der maximal erlaubten Geschwindigkeit wird nicht in Betracht gezogen, weil dies nicht garantieren kann, dass sich die Partikel wirklich annähern. Vielmehr erhöht sich das Risiko, dass sich die Partikel unkontrolliert im Untersuchungsbereich bewegen (Kennedy et al., 2001).

## 4.6 Durchlauf einer anderen Session

Die vorangehenden Ergebnisse sind immer für das erste Netzwerk am ersten Tag gerechnet worden. Der Algorithmus kann aber auch auf die anderen Netzwerke und andere Tage angewendet werden. Als Beispiel sieht man hier Ausschnitte des Standard-Durchlaufs für das

zweite Netzwerk am zweiten Tag, namentlich Session v1702 (Abbildung 4.21, Abbildung 4.22). Die vollständigen Plots sind im Anhang zu finden.

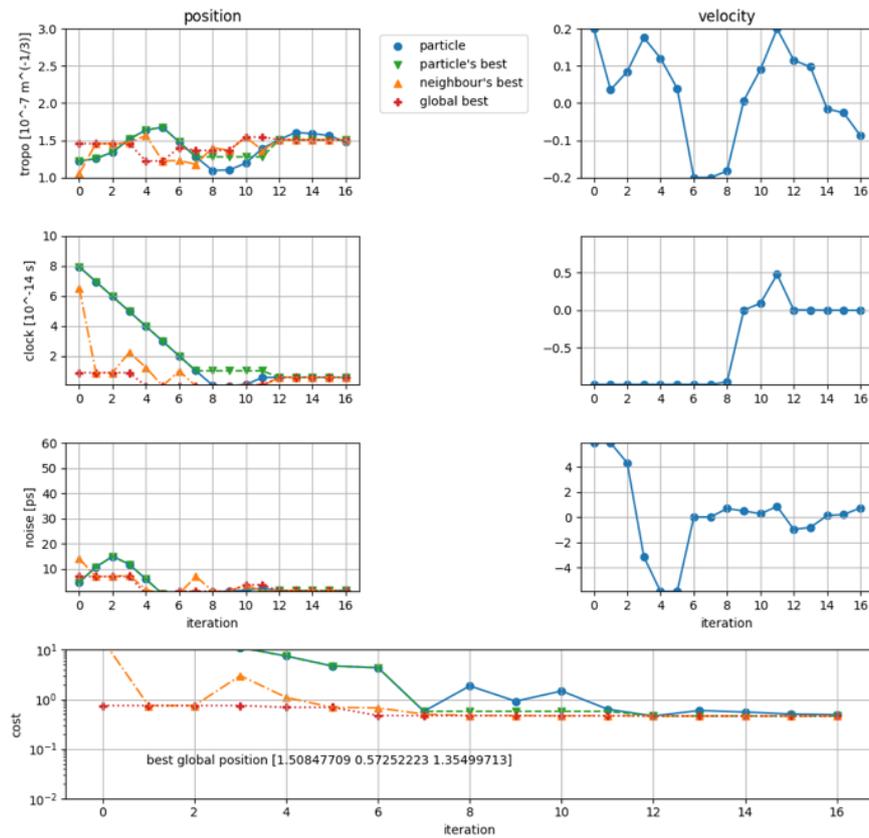


Abbildung 4.21: Position, Geschwindigkeit, Kosten eines Partikels beim Standard-Durchlauf der Session v1702 (eigene Darstellung)

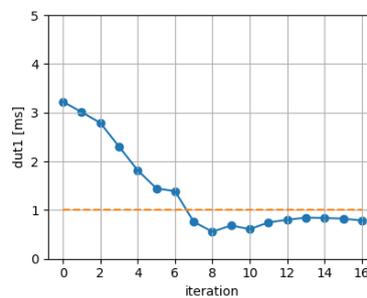


Abbildung 4.22: Annäherung an den Referenzwert DUT1 beim Standard-Durchlauf der Session v1702

Sowohl die Annäherung an die global beste Position und die global beste Bewertung als auch die Annäherung an die Referenzwerte ist gegeben (Abbildung 4.21, Abbildung 4.22). Ein Säulendiagramm kann die Performanz noch genauer aufzeigen (Abbildung 4.23).

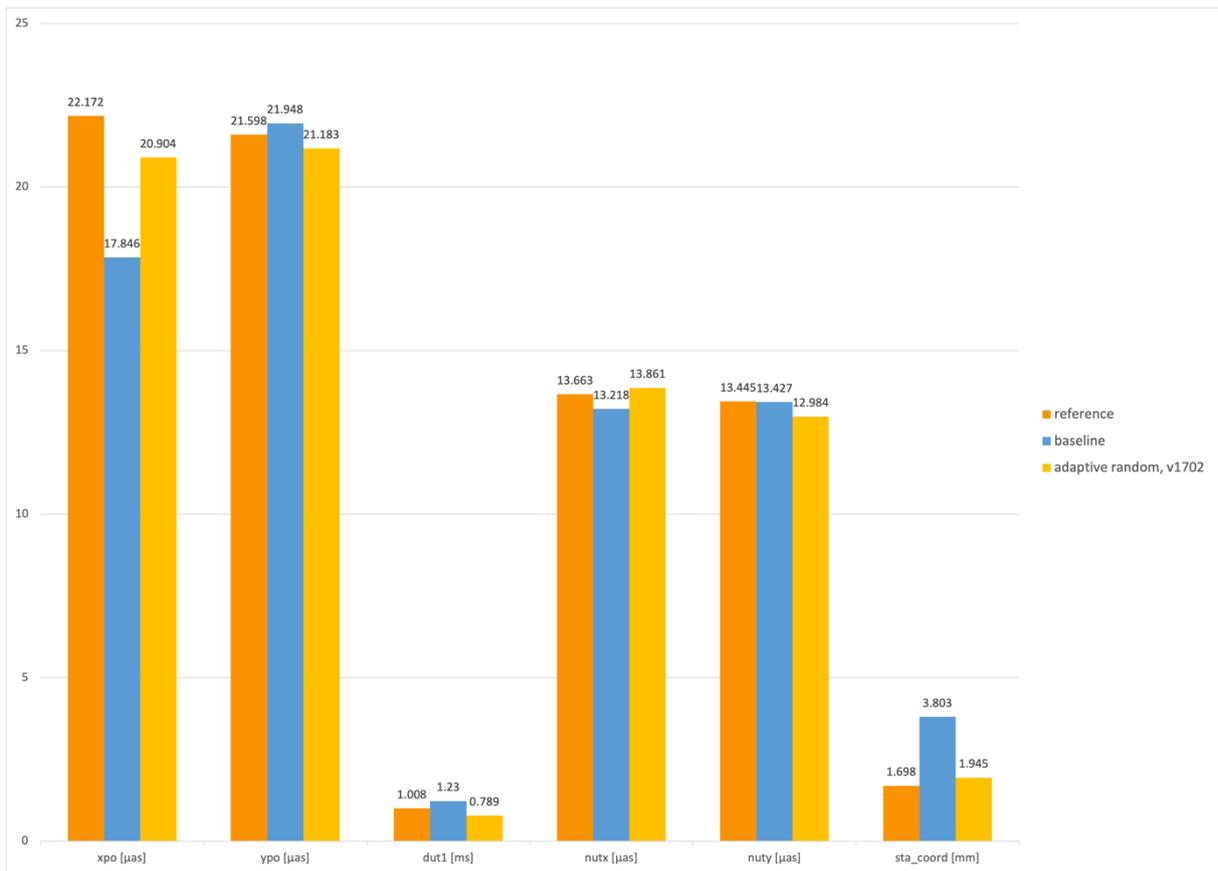


Abbildung 4.23: Vergleich der EOP der Durchläufe der Session v1702 mit der Baseline Lösung und den Referenzwerten (eigene Darstellung)

## 4.7 Weitere Durchläufe

Wie in Kapitel 2.3.4 beschrieben, können für den Vergleich verschiedene Werte herangezogen werden. Wenn dabei die Wiederholbarkeit anstatt des mittleren, formalen Fehlers gewählt wird, ergeben sich folgende Plots (Abbildung 4.24, Abbildung 4.25). Abbildung 4.24 ist dabei bloss ein Ausschnitt und der vollständige Plot ist im Anhang zu finden. Es ergeben sich kleine Werte für die Position, sprich für die Simulationsparameter. Dies ist physikalisch gesehen nicht realistisch. Numerisch funktioniert aber auch hier der PSO einwandfrei, weil sich die simulierten EOP an die Referenzwerte annähern (Abbildung 4.24).

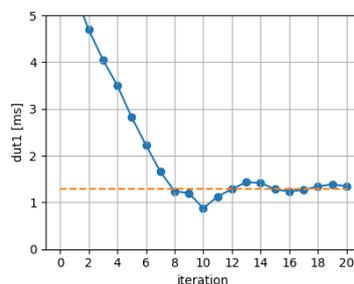


Abbildung 4.24: EOP eines Partikels im Vergleich zu den Referenzwerten bei der globalen Topologie, wenn die Wiederholbarkeit zum Vergleich verwendet wird (eigene Darstellung)

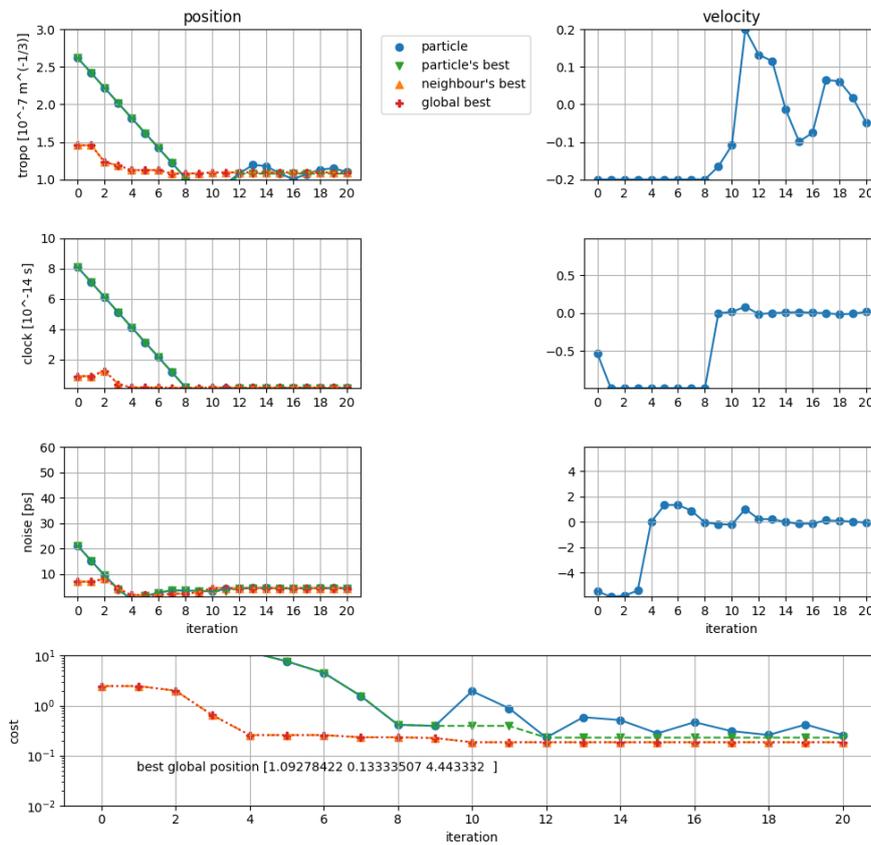


Abbildung 4.25: Position, Geschwindigkeit, Kosten eines Partikels bei der globalen Topologie, wenn die Wiederholbarkeit zum Vergleich verwendet wird (eigene Darstellung)

Ebenfalls können die Hyperparameter angepasst werden. Wenn für den Social und Cognitive Parameter wie in der Literatur vorgeschlagen der Wert 2 gesetzt wird (Kapitel 2.3.7), kann ein Plot gemäß Abbildung 4.26 generiert werden. Die Position des Partikels konvergiert dabei nicht und die Geschwindigkeit ist unkontrolliert.

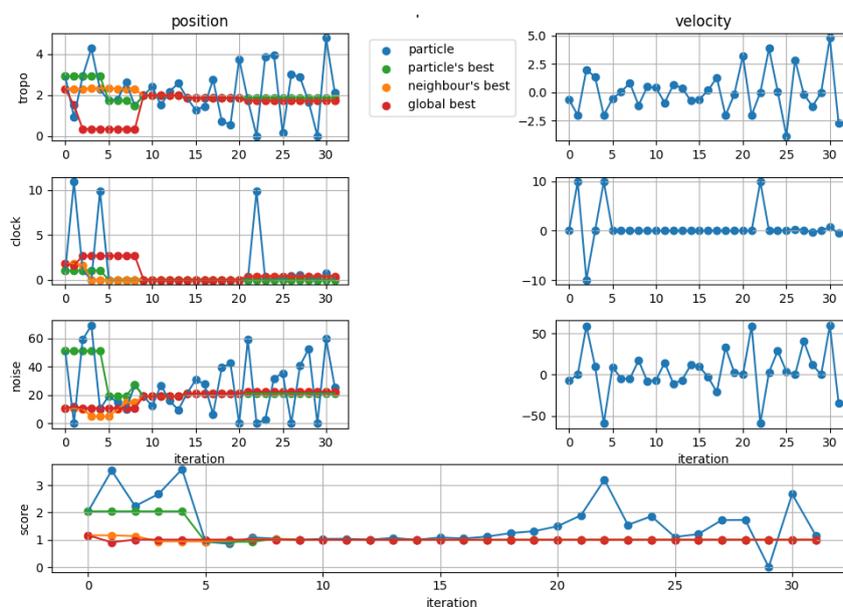


Abbildung 4.26: Position, Geschwindigkeit, Kosten bei Standard-Durchlauf mit Social und Cognitive Parameter auf 2, score ist mit cost gleichzusetzen (eigene Darstellung)

## 5 Diskussion

Allgemein lässt sich feststellen, dass in allen getesteten Fällen der Algorithmus sein Ziel erreicht. Sowohl die Position und die Bewertung wie auch die simulierten EOP nähern sich dem global besten Ergebnis beziehungsweise den Referenzwerten an. Bei den meisten Durchläufen erreicht der PSO bessere Ergebnisse als die Baseline Lösung. Das heisst, mit dem PSO können Simulationsparameter bestimmt werden, die die Realität besser abbilden als die bis anhin verwendeten Standard-Simulationsparameter (Abbildung 5.1).

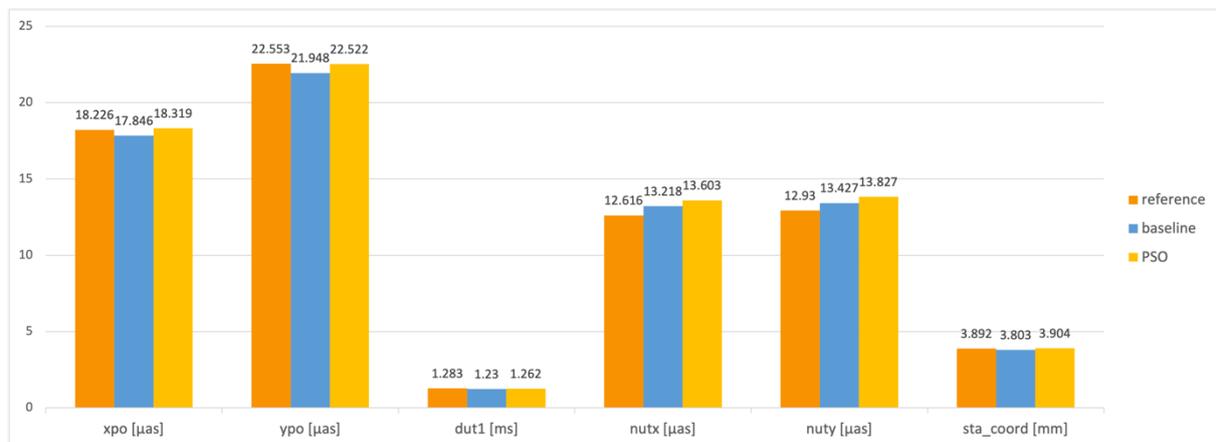


Abbildung 5.1: Vergleich PSO, Baseline Lösung und Referenzwerte (eigene Darstellung)

Aufgrund der zufälligen Einflussfaktoren ergeben sich diverse Resultate, die aber alle passend sind. Folglich kann mit einer Vielzahl an Kombinationen von Parametern ähnlich gute Ergebnisse erzielt werden. Es ist daher auch möglich, dass ein Simulationsmodell mittels eines anderen ausgedrückt werden kann, z.B. dass ein geringer Uhrfehler durch erhöhtes weisses Rauschen kompensiert wird. Der PSO als Ansatz zur Optimierung der Simulationsparameter scheint daher plausibel. Dennoch sind die numerisch optimalen Werte nicht zwingend die physikalisch Besten. So werden beispielsweise Parameter teilweise 0. Dies liegt daran, dass die Optimierung ein rein numerisches Verfahren ist und die physikalische Korrektheit nicht garantiert werden kann. Aus diesem Grund müssen die berechneten Werte nicht zwingend realistisch sein und bei einer Weiterverwendung kritisch hinterfragt werden (Kapitel 5.3 und Kapitel 5.4).

Des Weiteren ist auch nicht gegeben, dass jedes Partikel die Realität gleich gut abdeckt. In Summe kann aber beobachtet werden, dass die Partikel zufriedenstellende Ergebnisse liefern.

### 5.1 Standard-Durchlauf

Der Standard-Durchlauf ist eine gute Wahl zur Optimierung der Simulationsparameter, da bei vielen EOP im Vergleich zur Baseline Lösung passendere Werte gefunden werden (Abbildung 4.1). Dies gilt für die Polschwankungen, die Stationskoordinaten und die Zeitkorrektur DUT1. Einzig für die Nutationen kann mittels der globalen Nachbarschaft bessere Ergebnisse erzielt werden.

## 5.2 Nachbarschaften

Auf den ersten Blick scheinen alle Nachbarschaften bis auf die Wheel-Topologie ähnlich gut zu funktionieren. Diese These lässt sich auch durch Abbildung 4.1 bestätigen. Man sieht, dass das informierende Partikel der Wheel-Nachbarschaft zwar einigermaßen nahe an die Referenzwerte kommt, aber alle übrigen Partikel weit weg sind. Es lässt sich auch feststellen, dass immer mindestens ein durch den PSO generiertes Ergebnis den Referenzwerten näherkommt als die Baseline Lösung. Die Adaptive Random und Ring-Nachbarschaft erreichen dabei meist ähnlich passende Resultate, wobei die Ring-Nachbarschaft meist mehr Iterationen benötigt. Ausser für die Nutationen generieren sie die besten Werte. Für die Approximation der Nutation ist die globale Nachbarschaft am besten geeignet.

## 5.3 Ohne Uhrfehler

Beide untersuchten Topologien liefern beim Durchlauf ohne Uhrfehler ähnlich gute Ergebnisse. Unter Berücksichtigung des Säulendiagramms ist zu erkennen, dass die Baseline Lösung mehrheitlich am besten funktioniert (Abbildung 4.18). Folglich können bessere Simulationsparameter bestimmt werden, wenn der Uhrfehler mitberücksichtigt wird, auch wenn er dadurch teilweise auf 0 gesetzt wird. Dies verdeutlicht die Problematik des numerischen Verfahrens in einem Setting, in welchem physikalische Korrektheit erwartet wird.

## 5.4 Verschiedene Simulationsparameter pro Station

Die Resultate verdeutlichen, dass der PSO ebenfalls gute Ergebnisse erzielt, wenn nicht für alle Stationen dieselben Parameter angenommen werden. Mit der Adaptive Random Topologie funktioniert der Algorithmus dabei deutlich besser als mit der globalen Topologie. Allgemein werden jedoch für die höher-dimensionale Optimierung mehr Iterationen benötigt, da das Problem komplexer ist.

## 5.5 Andere Session

Der Algorithmus funktioniert auch für verschiedene Sessions. Im Vergleich zur Baseline Lösung ist zu beobachten, dass im Falle der Polschwankung in x-Richtung und den Stationskoordinaten mit dem PSO massiv bessere Ergebnisse erzielt werden können (Abbildung 4.23).

## 5.6 Weitere Durchläufe

Im Vergleich zum mittleren, formalen Fehler ergibt die Wiederholbarkeit physikalisch unpassendere Werte. Dies sieht man in Abbildung 4.25 an den kleinen Werten für die Position, sprich für die Simulationsparameter.

Bei der Wahl der Hyperparameter, Social und Cognitive Parameter, lässt sich aufzeigen, wie diese das Verhalten der Partikel beeinflussen. Bei höheren Werten bewegen sich die Partikel zu unruhig und konvergieren nicht (Abbildung 4.26). Der Einfluss der Parameter ist zu gross.

## 5.7 Auffälligkeiten

Bei einigen Läufen des Algorithmus treten spezielle, teils unerklärte Phänomene auf. In diesem Kapitel werden diese kurz aufgegriffen.

Immer wieder zu beobachten sind die «springenden» Kosten der Partikel (Abbildung 5.2). Dies kann mit der Formel der Geschwindigkeit erklärt werden (Formel (2.9)). Die beiden letzten Terme, die die Erfahrung des Partikels und den Einfluss des Schwarms beschreiben, werden bei der Annäherung an die global beste Bewertung 0. Die Geschwindigkeit bleibt nicht 0 wegen des ersten Terms. So bewegen sich die Partikel wieder weg, werden aber später zurückgezogen. Es könnte getestet werden, ob sich dies ändert, wenn die maximal erlaubte Geschwindigkeit anhand der Nähe zum global besten Partikel skaliert wird. Ein ähnliches Verhalten ist auch bei der Annäherung an die Position oder die Referenzwerte zu beobachten (bspw. Abbildung 4.2 und Abbildung 4.3). Dabei gleicht die Approximation einer gedämpften Schwingung, sprich einer sinusoidalen Kurve, was auch in der Literatur beschrieben wird (Kennedy et al., 2001). Dies kann mit der Trägheit der Partikel erklärt werden.

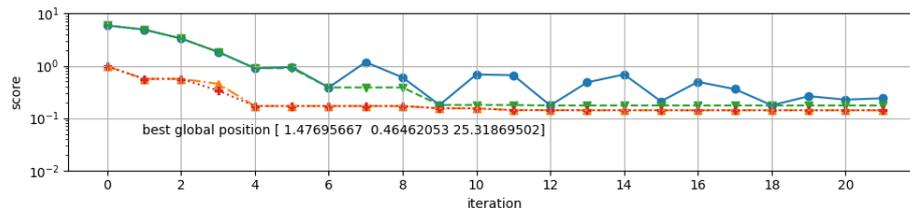


Abbildung 5.2: "springende" Partikel (eigene Darstellung), score ist mit cost gleichzusetzen

Teilweise ergeben sich auch Plots, bei welchen die Genauigkeit der EOP zuerst abnimmt. Dies hängt ebenfalls mit der Trägheit der Partikel zusammen, hindert aber die Annäherung an die Referenzwerte nicht (Abbildung 5.3).

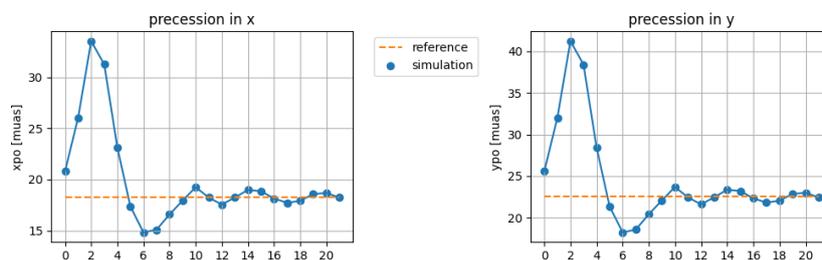


Abbildung 5.3: anfangs abnehmende Genauigkeit der EOP

Bei einem Durchlauf der Ring-Typologie zeigt der Algorithmus ein Verhalten, das so nicht auftreten sollte. Die Bewertung des Nachbarpartikels in orange ist schlechter als die momentane Bewertung des Partikels in blau (Abbildung 5.4). Wieso dies ein Mal passiert, ist unklar.

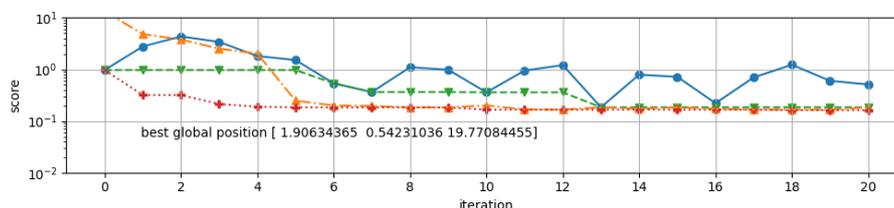


Abbildung 5.4: Kosten des Nachbarpartikels sind höher als die Kosten des Partikels, score ist mit cost gleichzusetzen (eigene Darstellung)

## 5.8 Empfehlung

Wie die Resultate in Kapitel 4 und die obige Diskussion zeigen, können mithilfe des PSO Simulationsparameter bestimmt werden, die bessere Beobachtungen simulieren als die Standardparameter. Daher wird folgende Empfehlung zur Anwendung des PSO zur Optimierung von VLBI-Simulationsparameter gemacht (Tabelle 5.1). Dabei spielt es keine Rolle, ob für alle Stationen dieselben oder individuelle Parameter definiert werden.

Tabelle 5.1: empfohlene Charakteristika des PSO zur Optimierung der VLBI-Simulationsparameter

Charakteristika	Beschreibung
Schwarmgrösse	16 Partikel
Initialisierung der Position	Formel (2.2)
Initialisierung der Geschwindigkeit	Formeln (2.3) und (2.4)
Initialer Wertebereich der troposphärischen Verzögerung	$1 \times 10^{-7} m^{-1/3}$ bis $3 \times 10^{-7} m^{-1/3}$
Initialer Wertebereich des Uhrfehlers	$0.1 \times 10^{-14} s$ bis $10 \times 10^{-14} s$
Initialer Wertebereich des weissen Rauschens	1 ps bis 60 ps
Eignung des Partikels / Kosten	Formel (2.7)
Simulierte Beobachtung zum Vergleich	Mittlerer, formaler Fehler
Anpassung der Position	Formel (2.8)
Anpassung der Geschwindigkeit	Formeln (2.9) und (2.4)
Nachbarschaft	Adaptive Random Topologie
Trägheitsfaktor	Von 0.9 bis 0.4 linear abnehmend abhängig von der Anzahl maximal erlaubter Iterationen
Cognitive Parameter	1
Social Parameter	1
Anzahl maximal erlaubter Iterationen	100
Stoppkriterium	In neun aufeinanderfolgenden Iterationen verbessert sich die global beste Position weniger als 5% oder die Anzahl maximal erlaubter Iterationen ist erreicht

## 6 Fazit

Die Arbeit zeigt, dass mithilfe des PSO die Simulationsparameter von VLBI optimiert werden können. Numerisch funktioniert das Verfahren sowohl, wenn für alle Stationen dieselben Parameter angenommen werden, wie auch wenn jede Station eigene Parameter hat. Von der physikalischen Seite betrachtet werden nicht immer optimale Ergebnisse produziert. Dies kann aber von einem rein numerischen Prozess nicht erwartet werden.

Es zeichnet sich ab, dass gewisse Topologien besser funktionieren als andere und die Schätzung des Uhrfehlers die Ergebnisse positiv beeinflusst. Ebenfalls zeigt sich, dass es nicht nur einen Satz an optimalen Simulationsparametern gibt, sondern eine Vielzahl an Kombinationen ähnlich gute Ergebnisse liefert.

In einem weiteren Schritt sollen die aufgestellten Thesen über das Verhalten des Algorithmus gefestigt werden, indem viele Durchläufe gestartet werden. Die kleinen Unstimmigkeiten, die noch ungeklärt sind, sollen behoben werden. Ebenfalls kann noch mehr mit den Hyperparametern experimentiert werden, um die Resultate weiter zu verfeinern. Es können auch diverse weitere Varianten des Algorithmus getestet werden, beispielsweise die Skalierung von  $v_{max}$  abhängig von der Distanz zum global besten Partikel oder eine andere Art des Vergleichs in der Score-Funktion. Wie anfangs vorgesehen, können nicht nur die Schwarmintelligenz, sondern auch Evolutionsstrategien zur Optimierung untersucht werden.

## Literaturverzeichnis

Behrend, D., Thomas, C., Gipson, J., Himwich, E., & Le Bail, K. (2020). On the organization of CONT17. *Journal of Geodesy*, 94(10), 100. <https://doi.org/10.1007/s00190-020-01436-x>

Bundesamt für Kartographie und Geodäsie (BKG). (2021). *Radioteleskop Wettzell*. [https://www.bkg.bund.de/DE/Observatorium-Wettzell/Messsysteme-Wettzell/Radioteleskop-Wettzell/radioteleskop\\_cont.html](https://www.bkg.bund.de/DE/Observatorium-Wettzell/Messsysteme-Wettzell/Radioteleskop-Wettzell/radioteleskop_cont.html)

Clerc, M. (2012). *Standard Particle Swarm Optimisation*. <https://hal.archives-ouvertes.fr/hal-00764996>

Git. (2021). <http://git-scm.com/download/win>

Grimme, C., & Bossek, J. (2018). *Einführung in die Optimierung: Konzepte, Methoden und Anwendungen*. Springer Vieweg.

Guillaume, S. (2018). *Geodetic Networks and Parameter Estimation*. Vorlesungsskriptum.

Herring, T. A., Davis, J. L., & Shapiro, I. I. (1990). Geodesy by radio interferometry: The application of Kalman Filtering to the analysis of very long baseline interferometry data. *Journal of Geophysical Research*, 95(B8), 12561. <https://doi.org/10.1029/JB095iB08p12561>

Kennedy, J. F., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers.

Kwok, N. M., Ha, Q. P., Liu, D. K., Fang, G., & Tan, K. C. (2007). Efficient particle swarm optimization: A termination condition based on the decision-making approach. *2007 IEEE Congress on Evolutionary Computation*, 3353–3360. <https://doi.org/10.1109/CEC.2007.4424905>

Nilsson, T., & Haas, R. (2010). Impact of atmospheric turbulence on geodetic very long baseline interferometry. *Journal of Geophysical Research*, 115(B3), B03407. <https://doi.org/10.1029/2009JB006579>

Pany, A., Böhm, J., MacMillan, D., Schuh, H., Nilsson, T., & Wresnik, J. (2011). Monte Carlo simulations of the impact of troposphere, clock and measurement errors on the repeatability of VLBI positions. *Journal of Geodesy*, 85(1), 39–50. <https://doi.org/10.1007/s00190-010-0415-1>

Petrachenko, B., Niell, A., Behrend, B., Corey, B., Böhm, J., Charlot, P., Collioud, A., Gipson, J., Haas, R., Hobiger, T., Koyama, Y., MacMillan, D., Malkin, Z., Nilsson, T., Pany, A., Tuccari, G., Whitney, A., & Wresnik, J. (2009). *Design Aspects of the VLBI2010 System*. *Progress Report of the IVS VLBI2010 Committee*. <https://hal.archives-ouvertes.fr/hal-00582342>

Python Software Foundation. (2021a). *Argparse – Parser for command-line options, arguments and sub-comments*. <https://docs.python.org/3/library/argparse.html>

Python Software Foundation. (2021b). *Pickle – Python object serialization*. <https://docs.python.org/3/library/pickle.html>

Rothacher, M., Hugentobler, U., Steigenberger, P., & Schmid, R. (2021). *Globale Satellitennavigationssysteme – Vorlesungsskriptum*.

Rothacher, M., Meindl, M., & Müller, M. (2019). *Satellitengeodäsie – Vorlesungsskriptum*.

Schartner, M. (2021). *Quelle: Standardwerte und Grenzen des Untersuchungsbereichs* [Persönliche Kommunikation].

Schartner, M., & Böhm, J. (2019). VieSched++: A New VLBI Scheduling Software for Geodesy and Astrometry. *Publications of the Astronomical Society of the Pacific*, 131(1002), 084501. <https://doi.org/10.1088/1538-3873/ab1820>

Schartner, M., Böhm, J., & Nothnagel, A. (2020). Optimal antenna locations of the VLBI Global Observing System for the estimation of Earth orientation parameters. *Earth, Planets and Space*, 72(1), 87. <https://doi.org/10.1186/s40623-020-01214-1>

Schuh, H., & Böhm, J. (2013). Very Long Baseline Interferometry for Geodesy and Astrometry. In G. Xu (Hrsg.), *Sciences of Geodesy—II: Innovations and Future Developments* (S. 339–376). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-28000-9\\_7](https://doi.org/10.1007/978-3-642-28000-9_7)

Tatham, S. (2021). *PuTTY*. <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

The Matplotlib development team. (2021). *Matplotlib: Visualization with Python*. <https://matplotlib.org>

Yang, X.-S. (Hrsg.). (2013). *Swarm intelligence and bio-inspired computation: Theory and applications* (1st ed). Elsevier.

# Selbstständigkeitserklärung



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

## Eigenständigkeitserklärung

Die unterzeichnete Eigenständigkeitserklärung ist Bestandteil jeder während des Studiums verfassten Semester-, Bachelor- und Master-Arbeit oder anderen Abschlussarbeit (auch der jeweils elektronischen Version).

Die Dozentinnen und Dozenten können auch für andere bei ihnen verfasste schriftliche Arbeiten eine Eigenständigkeitserklärung verlangen.

Ich bestätige, die vorliegende Arbeit selbständig und in eigenen Worten verfasst zu haben. Davon ausgenommen sind sprachliche und inhaltliche Korrekturvorschläge durch die Betreuer und Betreuerinnen der Arbeit.

**Titel der Arbeit** (in Druckschrift):

Optimizing geodetic VLBI Simulation Parameters using Swarm Intelligence

**Verfasst von** (in Druckschrift):

*Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich.*

**Name(n):**

Wicki

**Vorname(n):**

Juliette

Ich bestätige mit meiner Unterschrift:

- Ich habe keine im Merkblatt „Zitier-Knigge“ beschriebene Form des Plagiats begangen.
- Ich habe alle Methoden, Daten und Arbeitsabläufe wahrheitsgetreu dokumentiert.
- Ich habe keine Daten manipuliert.
- Ich habe alle Personen erwähnt, welche die Arbeit wesentlich unterstützt haben.

Ich nehme zur Kenntnis, dass die Arbeit mit elektronischen Hilfsmitteln auf Plagiate überprüft werden kann.

**Ort, Datum**

Zürich, 03.06.21

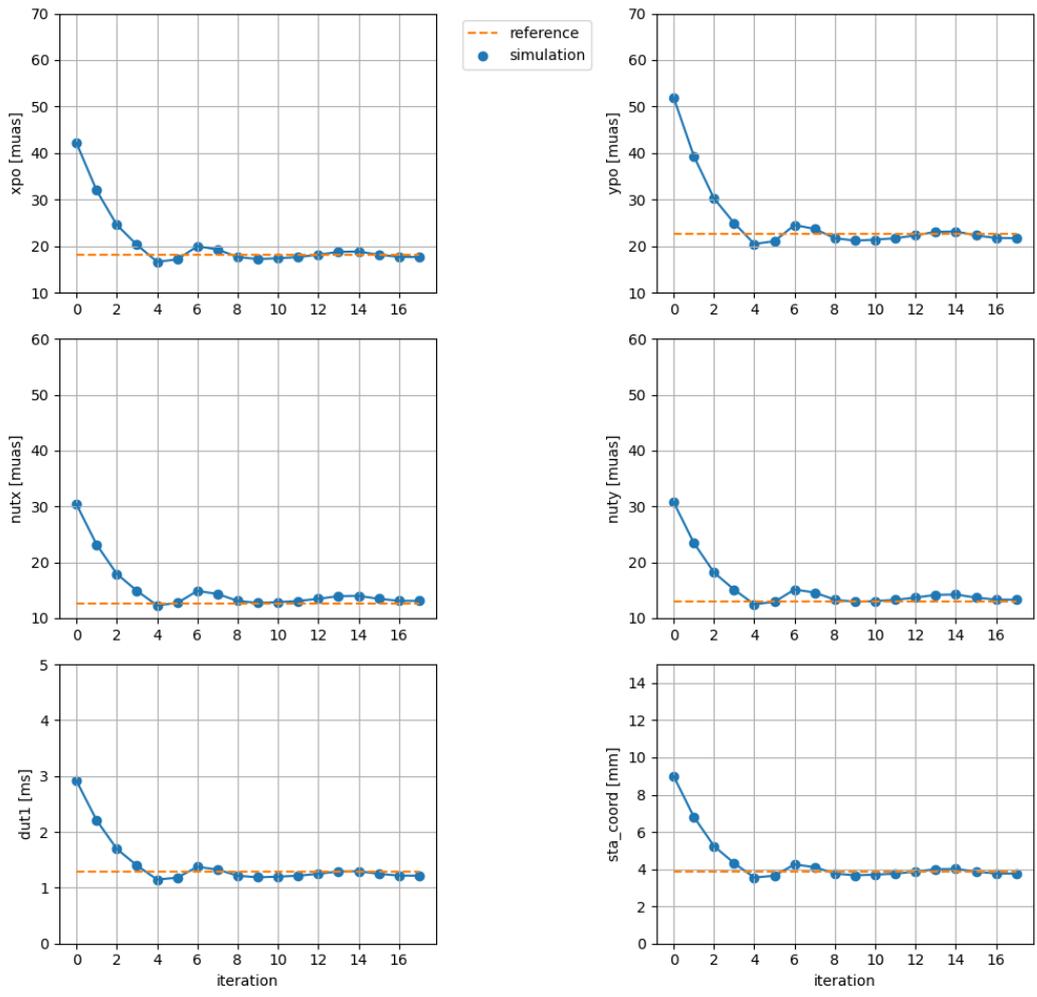
**Unterschrift(en)**

JWicki

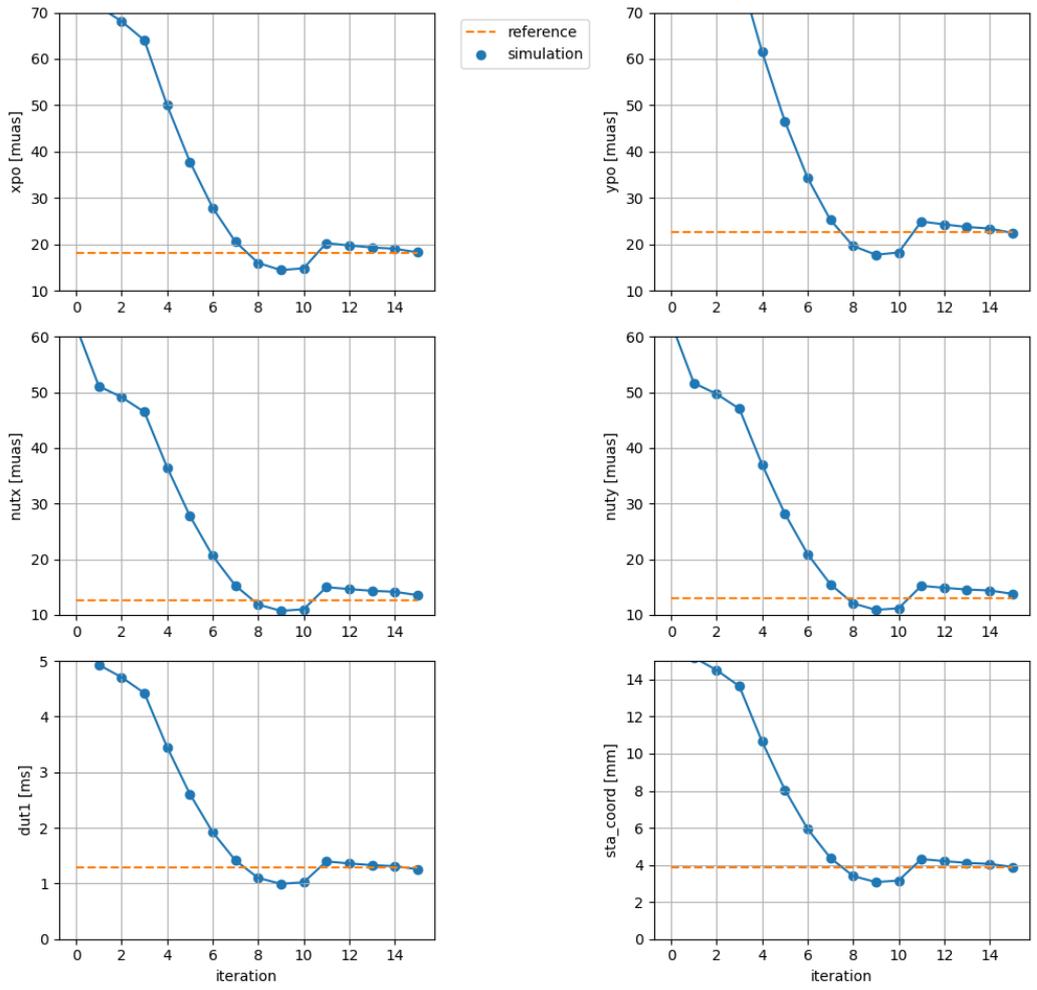
*Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich. Durch die Unterschriften bürgen sie gemeinsam für den gesamten Inhalt dieser schriftlichen Arbeit.*

# Anhang

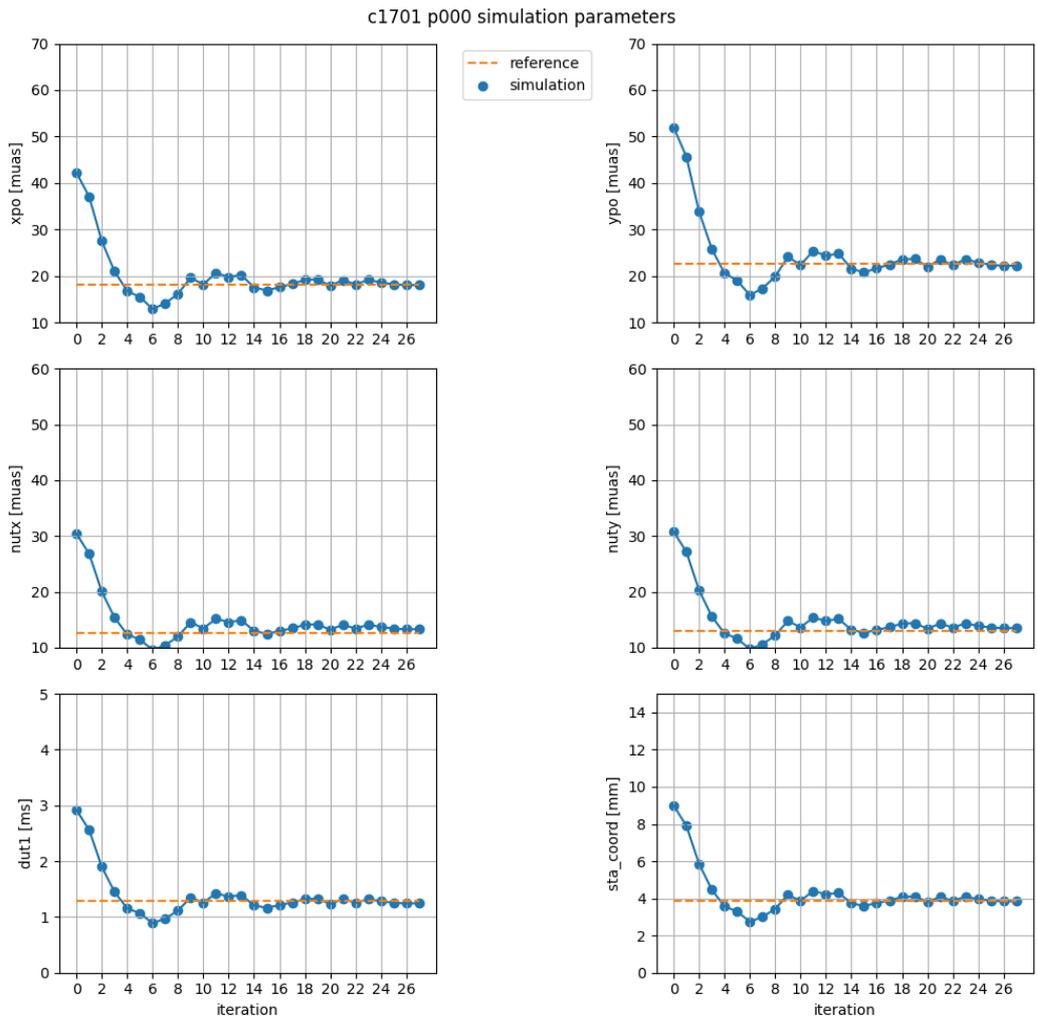
Vollständiger Plot zu Abbildung 4.5



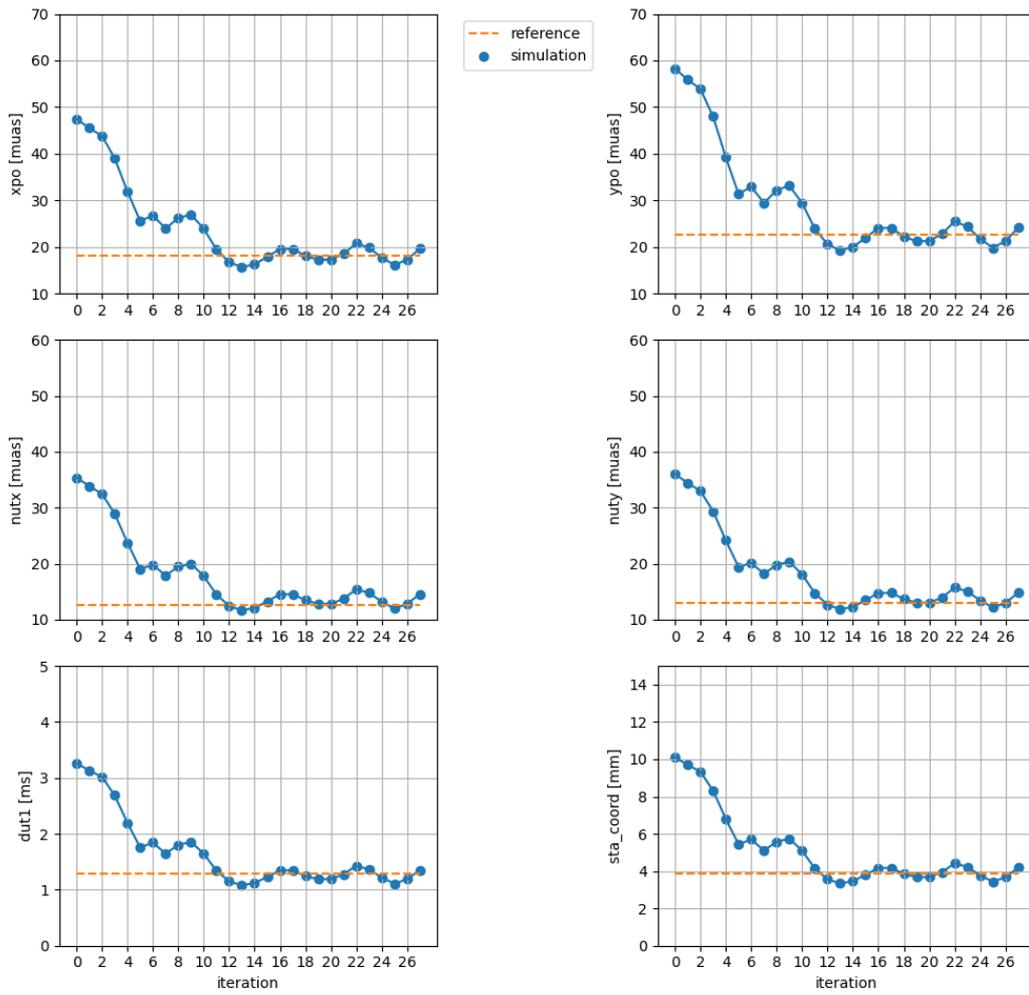
# Vollständiger Plot zu Abbildung 4.6



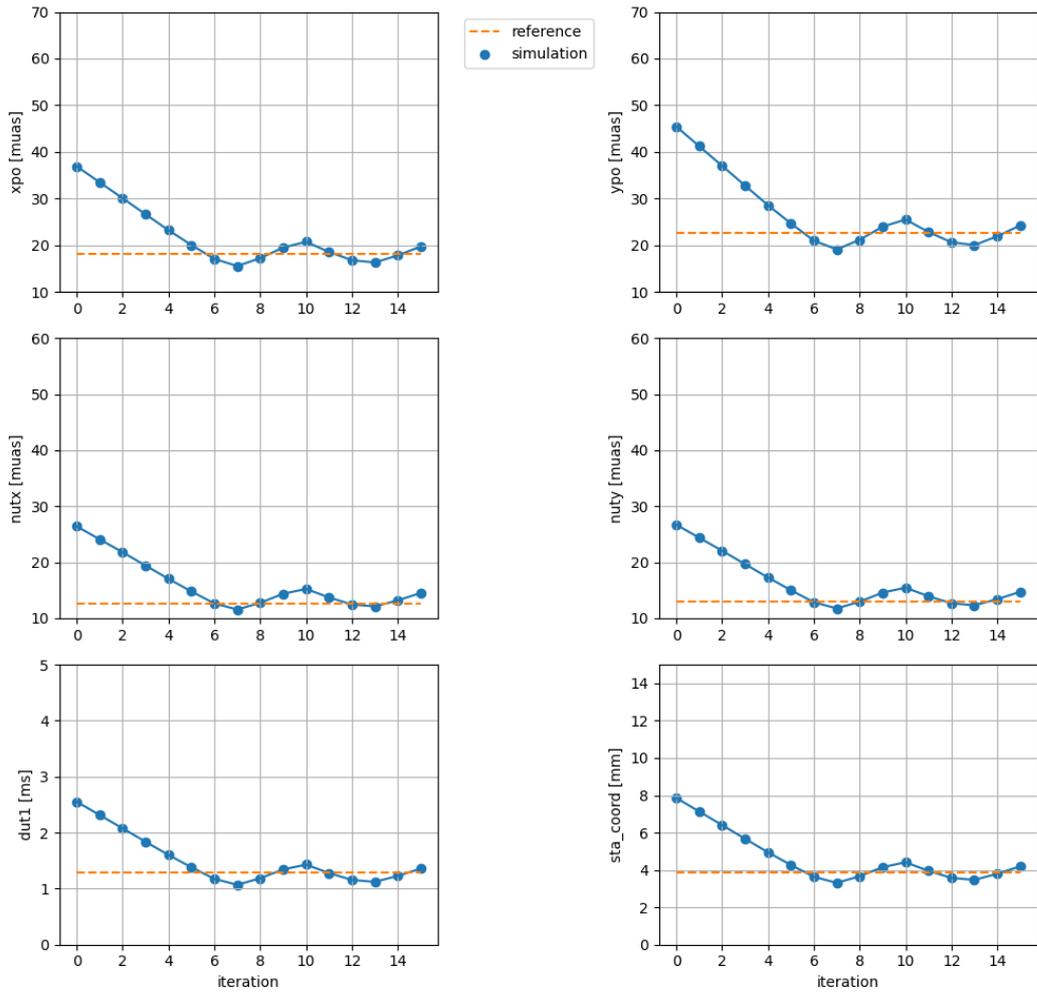
# Vollständiger Plot zu Abbildung 4.7



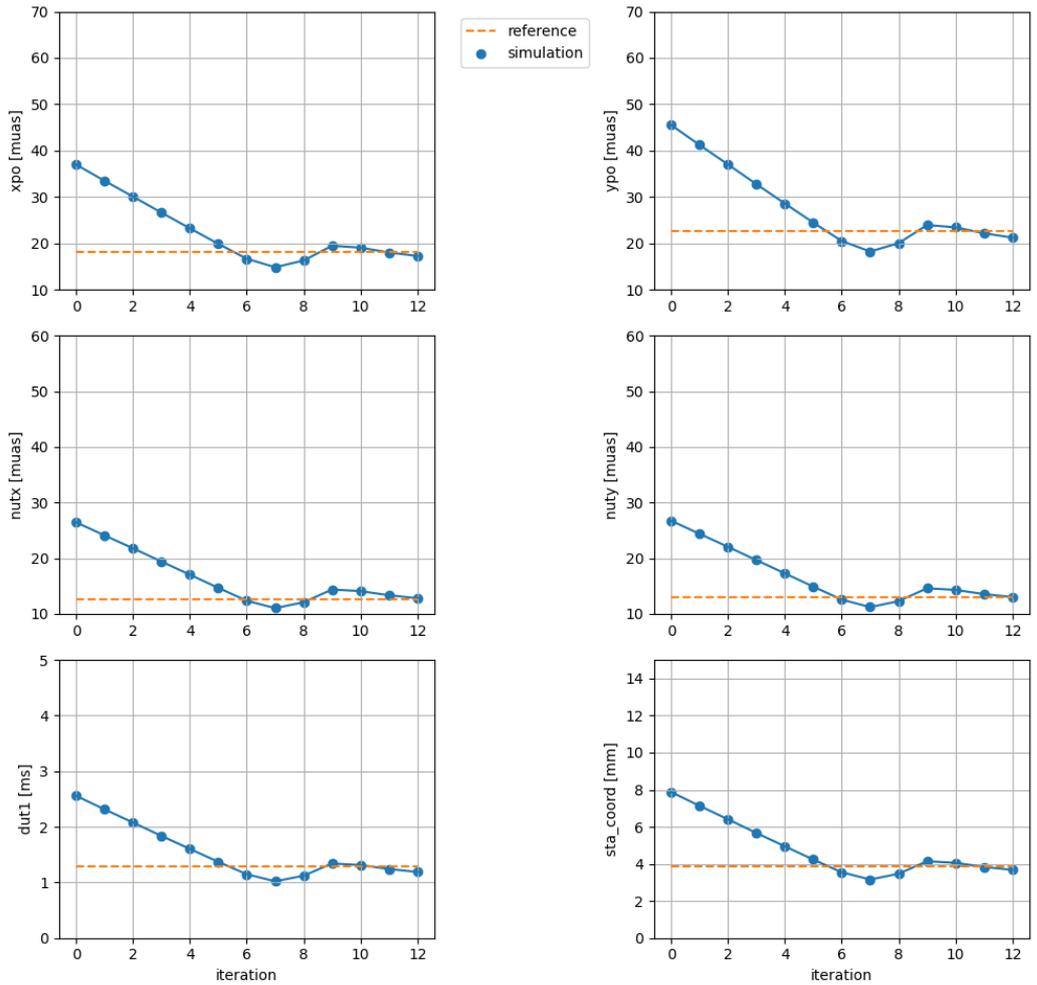
# Vollständiger Plot zu Abbildung 4.8



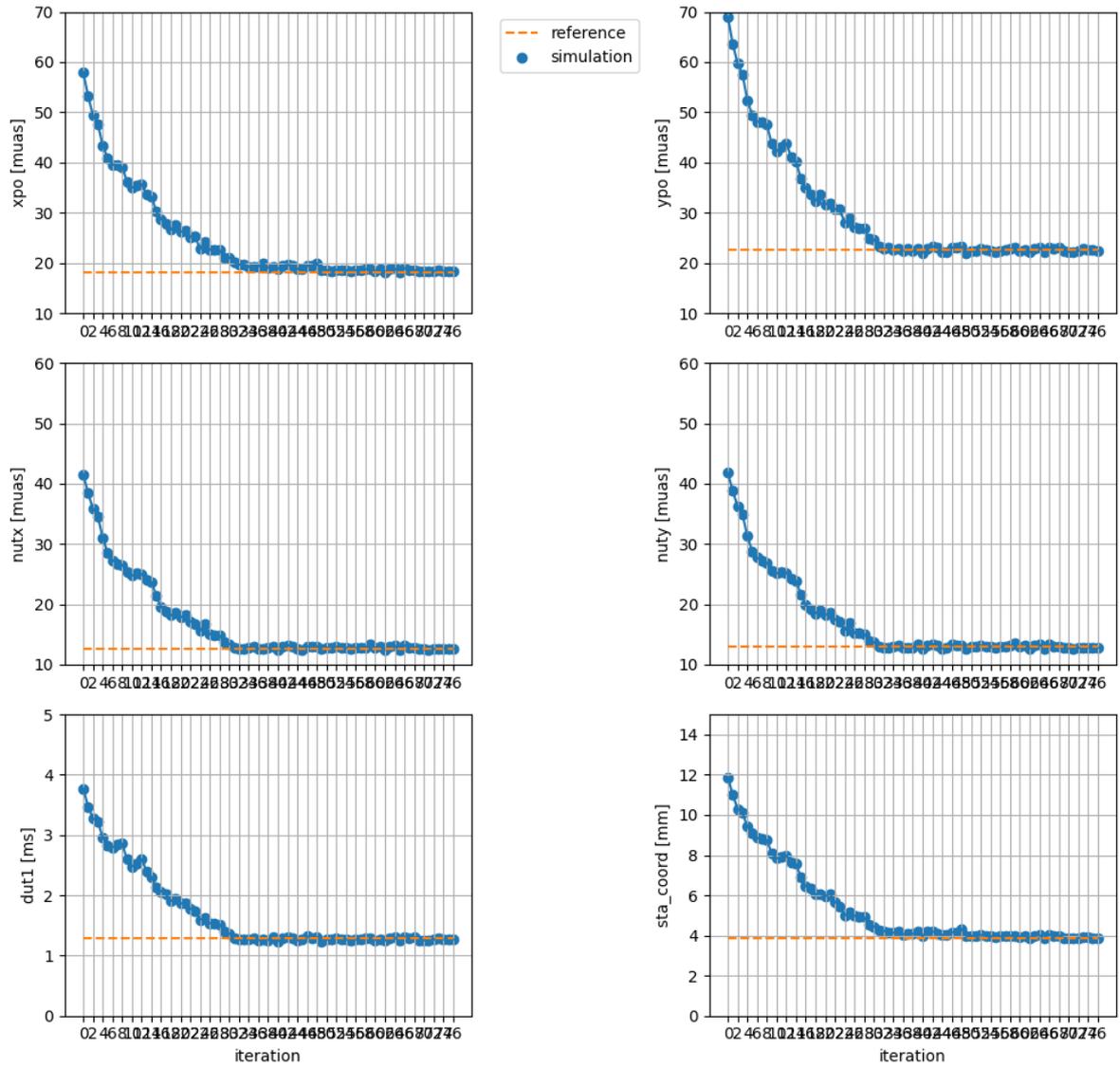
# Vollständiger Plot zu Abbildung 4.16



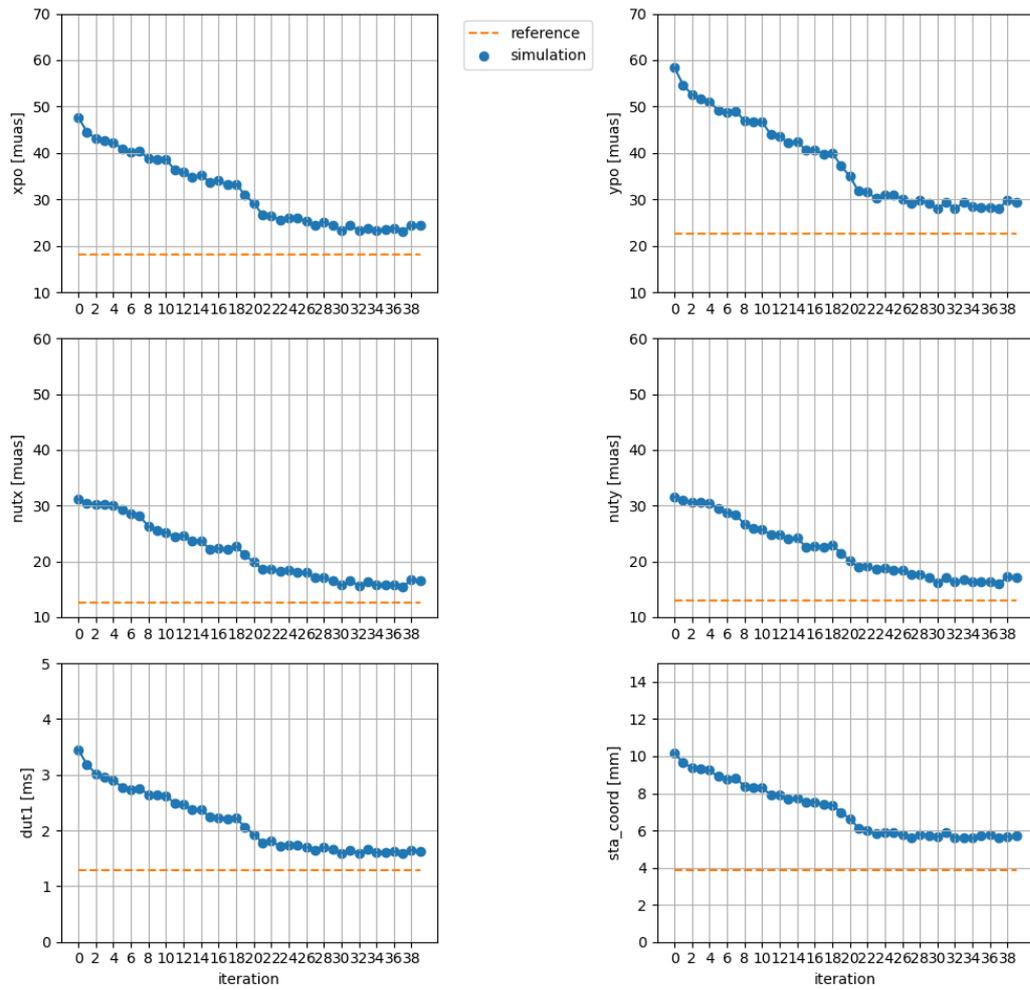
# Vollständiger Plot zu Abbildung 4.17



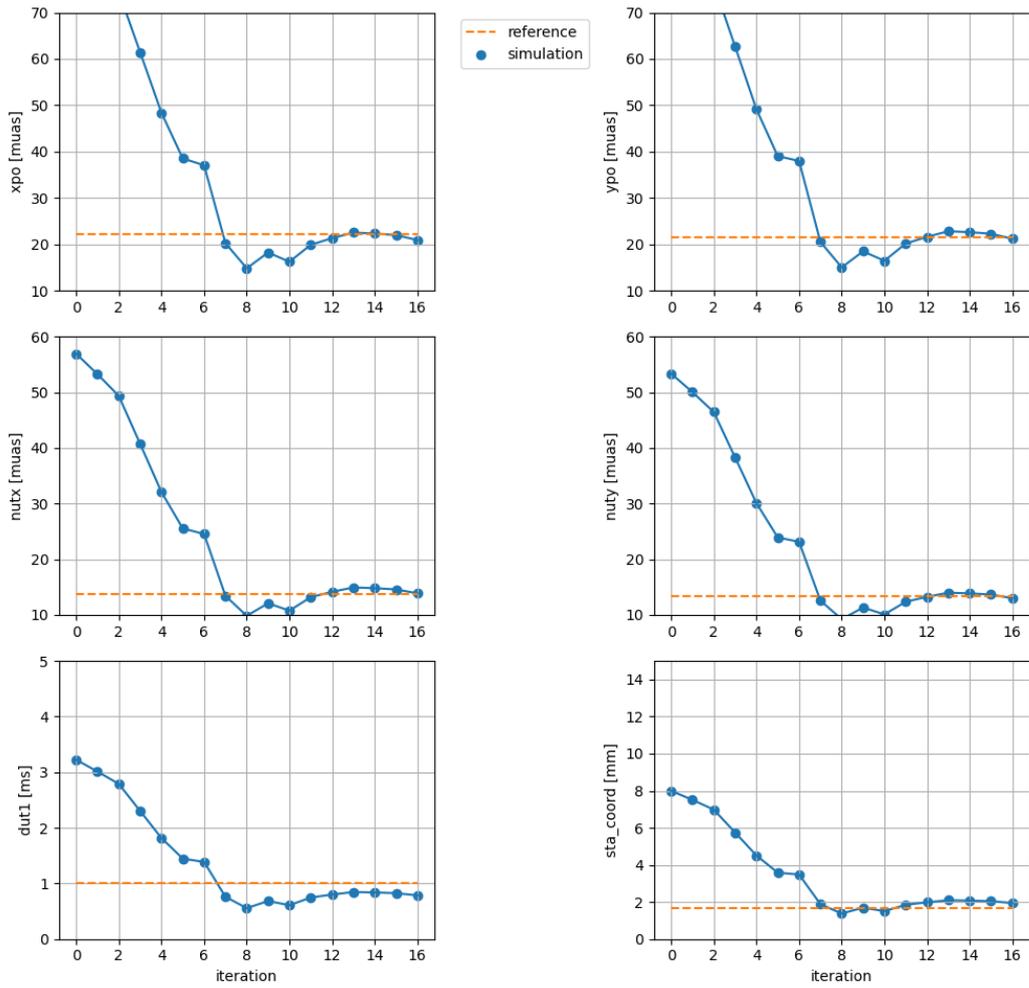
# Vollständiger Plot zu Abbildung 4.19



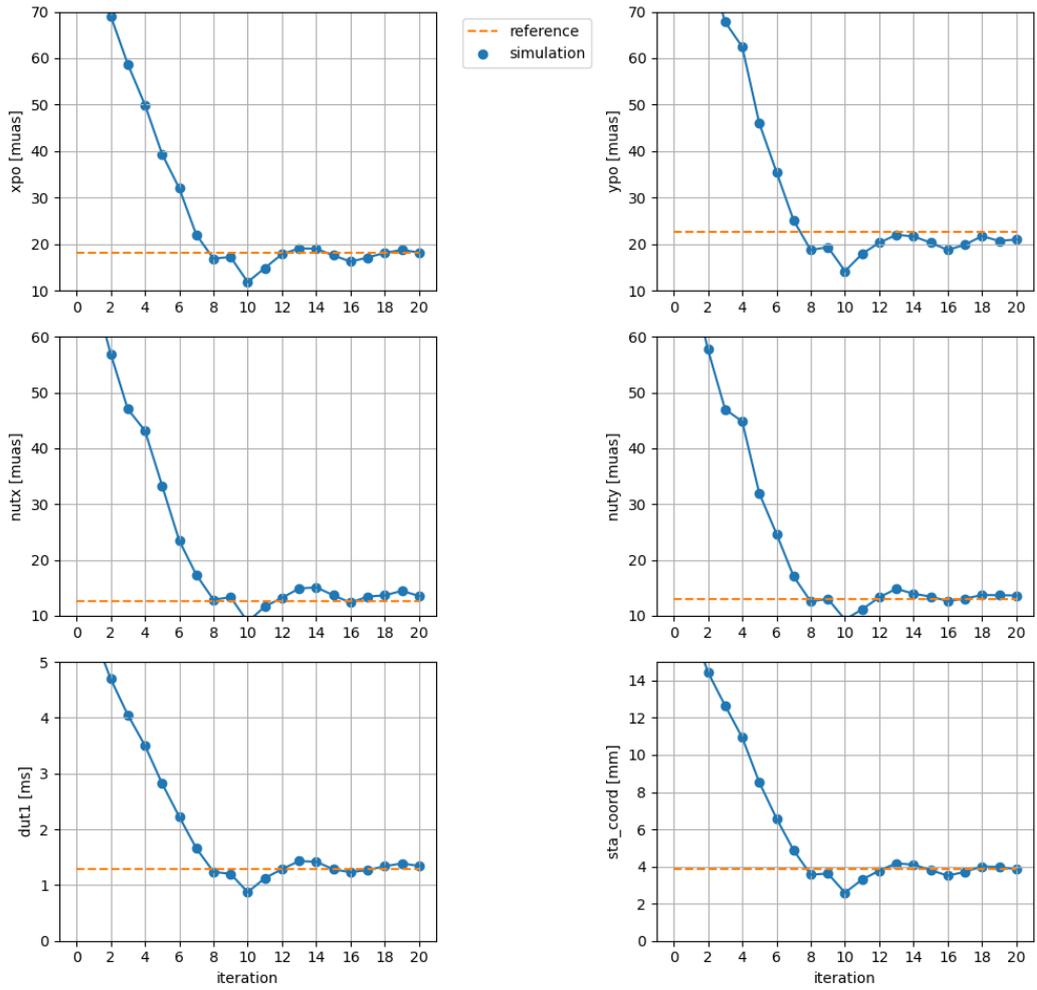
## Vollständiger Plot zu Abbildung 4.20



# Vollständiger Plot zu Abbildung 4.22



# Vollständiger Plot zu Abbildung 4.24



ETH Zürich  
Professur für Weltraumgeodäsie  
Robert-Gnehm-Weg 15  
8093 Zürich

<https://space.igp.ethz.ch>

**Herausgeber:** Professur für Weltraumgeodäsie  
**Redaktion:** Juliette Wicki  
**Gestaltung:** Juliette Wicki  
**Fotos:** Juliette Wicki  
**Druck:** ETH Store Zürich  
**Auflage:** 1

© ETH Zürich, Juni 2021