



Christian-Albrechts-Universität zu Kiel

Gefördert durch:



Bundesministerium
für Wirtschaft
und Klimaschutz



aufgrund eines Beschlusses
des Deutschen Bundestages

Luftfahrtforschungsprogramm VI (LuFo VI-1) Programmlinie “Technologie”

Verbundprojekt

Multisensorsystem zur Automation von Helikoptern und zur
sicheren Integration von UAS in den Luftverkehr mit 360°

Abdeckung

Master360

Teilprojekt

Kommunikation und Peilung basierend auf
Multimoden-Antennen (KommPeilMA)

Schlussbericht

Laufzeit und Berichtszeitraum: 01.06.2020 bis 30.06.2022

Verbundführer: Airbus Defence and Space GmbH

Projektpartner: Christian-Albrechts-Universität zu Kiel

FKZ: 20D19051

Geschäftszeichen: 100 386 733

Datum: 08.11.2022

M.Sc. Sami Alkubti Almasri

M.Sc. Nils L. Johannsen

Prof. Dr.-Ing. Peter A. Höher

Das diesem Bericht zugrunde liegende Vorhaben wurde mit den Mitteln des Bundesministeriums für Wirtschaft und Klimaschutz (BMWK, vormals BMWi) unter dem Förderkennzeichen 16KIS0915 gefördert. Die Verantwortung für den Inhalt dieses Berichts liegt bei den Berichtverfassern.

Inhaltsverzeichnis

I	Kurzdarstellung	5
1	Aufgabenstellung	5
2	Voraussetzungen, unter denen das Vorhaben durchgeführt wurde	6
3	Planung und Ablauf des Vorhabens	6
4	Stand der Wissenschaft und Technik	7
5	Zusammenarbeit mit anderen Stellen	8
II	Eingehende Darstellung	9
1	Verwendung der Zuwendung und erzielte Ergebnisse	9
1.1	HAP1: Optimierung UAV Luftraumnutzung	9
1.1.1	AP1.6: Anforderung Systemkomponenten	9
1.2	HAP6: Cruise-UTM, Datenfusion und Avoid	13
1.2.1	AP6.2: Konzept Advanced A2A Link (AP CAU-2 (E6.2.CAU)) . . .	13
1.2.2	AP6.3: Implementierung und Test Advanced A2A Link (AP CAU-2 (E6.2.CAU), AP CAU-3 (E6.3.CAU))	14
2	Wichtigste Positionen des zahlenmäßigen Nachweises	20
3	Notwendigkeit und Angemessenheit der geleisteten Arbeit	20
4	Verwertbarkeit der Ergebnisse	21
5	Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen	21
6	Erfolgte und geplante Veröffentlichungen der Ergebnisse	21
	Anhang	22
A	Literaturliste	22

B	MATLAB-Software für Simulationen	25
B.1	simDoaDistEstim.m	25
B.2	Demo_main.m	29
B.3	requestRxData.m	31
B.4	simRxData.m	32
B.5	ModeS_Modulate.m	33
B.6	getPattern3D.m	34
B.7	EtoD.m	37
B.8	wm3D.m	38
B.9	coeffMatrix3D.m	40
B.10	simMusicCost3D.m	41
B.11	WMInterpPattern3D.m	42
B.12	CRLB3D.m	43
C	MATLAB-Software für praktische Implementierungen	45
C.1	Demo_main.m	45
C.2	DoaDistEstim.m	47
C.3	pushOutput.m	50
C.4	pushRxData.m	50

Teil I

Kurzdarstellung

Das Verbundprojekt Master360 zielt darauf ab, eine sichere und effiziente Integration von unbemannten Luftfahrzeugen (UAV) und Lufttaxis (UAM) im Luftverkehr zu ermöglichen. Diesbezüglich stellt die Kollisionsvermeidung eine der größten und wichtigsten Herausforderungen dar. Zur Verhinderung einer potentiellen Kollision mit einem bemannten Luftfahrzeug oder anderen UAV ist die kooperative Zusammenarbeit von mehreren Sensorsystemen erforderlich. Durch die gewonnenen Informationen der Sensorsysteme und einen Datenfusions-Prozess sollen Ausweichmanöver und Routenempfehlungen errechnet werden. Eine weitere Herausforderung besteht darin, den Landeprozess auch bei anspruchsvollen und schwierigen Umgebungsbedingungen zu unterstützen. Das wird ermöglicht durch den Einsatz von neuartigen Methoden im Bereich der Künstlichen Intelligenz (KI).

Das Teilvorhaben Kommunikation und Peilung basierend auf sog. Multimoden-Antennen (KommPeilMA) unterstützt das Kollisionsvermeidungssystem und trägt so dazu bei, die Sicherheit der Integration von UAVs in den Luftverkehr zu erhöhen und die Wahrscheinlichkeit einer Kollision zu reduzieren. Basierend auf dem Konzept multimodaler Antennen wurde im Rahmen des Teilvorhabens eine Systemkomponente entwickelt, die die Kommunikation und Peilung anderer Luftverkehrsteilnehmer ermöglicht.

Die Schwerpunkte der Arbeiten an der Christian-Albrechts-Universität zu Kiel (CAU) liegen darin, ein Konzept zur Kommunikation und Peilung mithilfe der Multimoden-Antennen zu entwerfen, geeignete Algorithmik zur Standortbestimmung anderer Luftverkehrsteilnehmer zu entwickeln und diese als ausführbaren Code in einer Hochsprache zu realisieren und zu simulieren. Ein Prototyp der Multimoden-Antennen wurde vom Projektpartner Leibniz Universität Hannover (LUH) entworfen und als Untersuchungsbaustein für die CAU bereitgestellt. Die CAU hat das entworfene Konzept und die Algorithmik dem Partner f.u.n.k.e. AVIONICS GmbH (FAV) zur Verfügung gestellt. Im weiteren Projektverlauf ist vorgesehen, dass der Projektpartner FAV die Implementierung und Realisierung des Gesamtsystems anhand der Multimoden-Antenne des Partners LUH und der entwickelten Algorithmen der CAU übernimmt.

1 Aufgabenstellung

Um die Leistungsfähigkeit des Luftfahrtsystems im Sinne der Kollisionsvermeidung zu verbessern, und damit die sichere Einbindung von UAV im Luftverkehr zu ermöglichen, hat die CAU folgende Aufgaben bearbeitet und gelöst:

- Entwurf eines Konzeptes zur Integration von Multimoden-Antennen auf autonomen Fluggeräten zwecks Kommunikation und Peilung.
- Richtungsschätzung und Laufzeitmessung anhand des vorgeschlagenen Konzepts, um die Bestimmung der Position anderer Luftverkehrsteilnehmer zu ermöglichen.

- Algorithmische Umsetzung der entwickelten Lösungen zur Richtungschätzung und Laufzeitmessung, sowie Tests und Bewertung dieser in einer Simulationsumgebung.
- Bereitstellung der getesteten Algorithmen für den Partner FAV als ausführbarer Code für eine Implementierung und für Praxistests.

2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Das Verbundprojekt Master360 wurde von zwölf industriellen und akademischen Partnern unter Konsortialführung der Airbus Defence and Space GmbH, Freimann, bearbeitet. Das Teilprojekt KommPeilMA wurde von der CAU unter Beteiligung der Projektpartner LUH und FAV bearbeitet. Für die Durchführung der Forschungsarbeiten standen an der CAU bis auf die Grundausstattung weder Mittel noch Personal zur Verfügung. Daher wurde an der CAU für die gesamte Projektlaufzeit eine volle Wissenschaftliche Mitarbeiterstelle geschaffen, welche zu 100 % vom Projektträger finanziert und mit M.Sc. Sami Albusti Almasri besetzt wurde. Dienstreisen zu Projekttreffen wurden ebenfalls über die bewilligten Projektmittel finanziert. Bezüglich der für die Durchführung der Forschungsarbeiten notwendigen technischen Ausstattung in Form von Rechnern und Software wurde auf vorhandene Ressourcen zurückgegriffen. Für die Bearbeitung der einzelnen Aufgaben konnte auf Erfahrungen aus früheren Forschungsprojekten zurückgegriffen werden. Zusätzlich haben Prof. Dr.-Ing. Peter A. Höher (Lehrstuhlinhaber) und M.Sc. Nils Johannsen (Wissenschaftlicher Mitarbeiter auf einer Landesstelle am Lehrstuhl) mit Ihrer Expertise zum Projekterfolg beigetragen.

3 Planung und Ablauf des Vorhabens

Der Arbeitsplan des Vorhabens Master360 ist in Hauptarbeitspakete (HAP) unterteilt. Jedes Hauptarbeitspaket ist in weitere Subarbeitspakete unterteilt. Die CAU war an den folgenden Hauptarbeitspaketen und den darunter organisierten Subarbeitspaketen beteiligt:

- HAP1: Optimierung UAV Luftraumnutzung
 - AP1.6: Anforderungen Systemkomponenten: Ziel dieses Arbeitspakets war, die Systemkomponenten und die Kernparameter zur Kommunikation und Peilung zu identifizieren.
- HAP6: Cruise-UTM, Datenfusion und Avoid
 - AP6.2: Konzept Advanced A2A Link: Im Rahmen dieses Pakets wurde ein Konzept zur Kommunikation und Peilung mittels Multimoden-Antennen erstellt. Geeignete Algorithmen zur Richtungs- und Distanzschätzung wurden bestimmt und die Komplexität der Algorithmen wurde abgeschätzt.

- AP6.3: Implementierung und Test Advanced A2A Link: Um die Leistungsfähigkeit der in AP6.2 bestimmten Algorithmen auszuwerten, erfolgte in diesem Arbeitspaket die Programmierung und die Simulation der Richtungs- und Distanzschätzalgorithmen.

Eine Beschreibung der Arbeitsschritte zur Bearbeitung jedes Arbeitspakets ist in Kapitel II.1 detailliert aufgeschlüsselt.

4 Stand der Wissenschaft und Technik

Methoden zur Navigation und Lokalisierung basieren auf zwei Hauptprinzipien: Einfallswinkel- und Signallaufzeitschätzung. Zur Bestimmung des Einfallswinkels eines empfangenen Signals werden klassischerweise Antennenarrays eingesetzt [1]. Aufgrund der räumlichen Verteilung der Antennenelemente eines Antennenarrays empfängt jedes Antennenelement ein Exemplar des einfallenden Signals mit einer Laufzeitverschiebung im Vergleich zu den anderen Antennenelementen. Die Laufzeitverschiebungen können mit der Kenntnis der Antennenarraygeometrie ausgenutzt werden, um den Einfallswinkel des empfangenen Signal zu schätzen. Im Gegensatz zur Einfallswinkelschätzung kommen für die Signallaufzeitschätzung üblicherweise einzelne Antennen zum Einsatz [2]. Die Kernidee ist es, die Laufzeit zwischen einem Sender und einem Empfänger zu schätzen und auf Basis der Lichtgeschwindigkeit die Distanz zu bestimmen. Somit kann die Position des Empfängers auf einen Kreis beschränkt werden, in dessen Mittelpunkt sich der Sender befindet und dessen Radius der berechneten Distanz entspricht. In Kombination mit dem geschätzten Einfallswinkel kann die Position des Empfängers eindeutig bestimmt werden.

Die innovative Idee der Multimoden-Antennen beruht auf der Theorie der Charakteristischen Moden [3]. Diese Theorie besagt, dass der Oberflächenstrom auf einem Radiator in mehrere Gruppen orthogonaler Stromkomponenten (auch Moden genannt) zerlegt werden kann. Durch räumlich verteilte Anregungspunkte können die entsprechenden Gruppen von Moden unabhängig von einander angeregt werden. Jede angeregte Gruppe von Moden bzw. jede Kombination davon generiert ein elektromagnetisches Feld. Gelingt es, mehrere Moden orthogonal anzuregen, so kann der Radiator mehrere unabhängige, sich überlagernde, elektromagnetische Felder gleichzeitig abstrahlen. In der Realisierung der Multimoden-Antennen werden die Anregungspunkte für jede Mode bzw. jede Gruppe von Moden mit einem Anschluss (auch Port genannt) verbunden. Damit sind die unabhängigen elektromagnetischen Felder auch separat anregbar [4]. Das heißt, eine einzelne Multimoden-Antenne verhält sich ähnlich wie eine Gruppenantenne, allerdings in einer kompakteren und robusteren Form. Damit bieten Multimoden-Antennen eine vielversprechende Lösung für Kommunikations- und Peilungssysteme. Bezüglich der Kommunikationsaspekte ermöglicht eine einzelne Multimoden-Antenne das parallele Senden und Empfangen von mehreren Datensequenzen gleichzeitig über die getrennten Ports. Bezüglich der Peilungsaspekte kann eine einzelne Multimoden-Antenne für die Einfallswinkelschätzung eingesetzt werden, wenn mehrere Ports benutzt werden, und für die Signallaufzeitschätzung, wenn ein einzelner Port benutzt wird.

Aus der Literatur sind verschiedene Entwürfe von Multimoden-Antennen basierend auf der Theorie der Charakteristischen Moden bekannt [5–7]. Eine planare Realisierung einer Multimoden-Antenne wurde in [8] erarbeitet. Die planare Antenne weist in einer kompakten Form vier Ports auf und kann damit vier Datenströme gleichzeitig ausstrahlen und

somit eine 4-Element-Gruppenantenne ersetzen. Eine Weiterentwicklung dieses Antennentyps wurde im Rahmen dieses Projekts verwendet. In [9–13] wurde die Leistungsfähigkeit planarer Multimoden-Antennen für Kommunikationszwecke untersucht. Die Tauglichkeit dieser Multimoden-Antennen zur Verwendung in ultra-hochratigen Kommunikationsanwendungen (bis zu 100 Gbps) wurde in [14] bewiesen. Ferner wurden die Grundlagen der Peilung mittels Multimoden-Antennen im Sinne der Einfallswinkelschätzung in [15–18] erarbeitet. In Kombination mit geeigneten Basisband-Algorithmen bestätigten diese Studien das Potenzial der Multimoden-Antennen hinsichtlich der Einfallswinkelschätzung. Numerische Simulationen zeigten eine hervorragende Leistungsfähigkeit sehr nah an der bekannten unteren Schranke der Schätztheorie.

5 Zusammenarbeit mit anderen Stellen

Die Durchführung der CAU-Arbeitspakete erfolgte in enger Kooperation mit den Verbundpartnern LUH und FAV. Im Rahmen von HAP1 erstellte die CAU nach einem intensiven Austausch mit den Verbundpartnern einen Anforderungskatalog. Projektpartner LUH legte die Randbedingungen für die Antennenentwicklung fest. Anforderungen für die praktische Umsetzung der entwickelten Methoden wurden von Projektpartner FAV definiert. Weiterhin stellte Projektpartner LUH der CAU einen Entwurf der Multimoden-Antenne für die Entwicklung der Kommunikations- und Peilungsalgorithmen im Rahmen von HAP6 zur Verfügung. Die entwickelten Algorithmen und Modelle wurden an Projektpartner FAV weitergeleitet. Die Bereitstellung für Projektpartner FAV erfolgte, wie im Antrag vorgesehen, in Form eines MATLAB-Codes sowie, auf Wunsch des Projektpartners, zusätzlich in Form eines ausführbaren C-Codes. Auf Basis der erzielten Erkenntnisse erfolgte über den Projektumfang hinaus eine Kooperation mit dem Projektpartner HAW Hamburg.

Teil II

Eingehende Darstellung

1 Verwendung der Zuwendung und erzielte Ergebnisse

Einzelne Arbeitspakete und Ziele mit den entsprechenden Ergebnissen

1.1 HAP1: Optimierung UAV Luftraumnutzung

1.1.1 AP1.6: Anforderung Systemkomponenten

Zwecks Bearbeitung dieses Arbeitspakets wurden die Anforderungen an das System in zwei Bereiche unterteilt – die Kommunikation und die Peilung. In jedem der beiden Bereiche wurden die Randbedingungen festgelegt und definiert. Weiterhin wurden die Kernparameter für Kommunikation und Peilung identifiziert und bestimmt.

Systemanforderung zwecks Kommunikation (AP CAU-0 (E1.6.6), AP CAU-1 (E6.2.CAU))

Für die Definition der Systemanforderung ist das Einsatzszenario entscheidend. Es wird erwartet, dass das UAV im unteren Luftraum Golf operiert. Die Notwendigkeit der Peilung erfordert eine omnidirektionale Abfrage anderer Luftfahrzeug-Transponder durch den Interrogator. Daher ergab sich die Notwendigkeit, ausgehend von der Erwartung, je eine planare Antenne auf den Stirnseiten des UAV zu platzieren, eine omnidirektionale Richtcharakteristik mittels Beamforming zu ermöglichen. Desweiteren wurden die verfügbaren Standards und die Literatur im Bereich Luftfahrtransponder zu den Punkten Antennendiversität, Beamforming und Reichweite untersucht.

In [19] werden zum Beispiel einige Informationen der Standards [20] zusammengefasst. Sendeleistungen und Signalentwurf wurden [19] und [21] entnommen. Insgesamt lässt sich festhalten, dass die Sendeleistung je nach Fluggeschwindigkeit des Fluggerätes mindestens entweder 75 W (langsamer als 175 Knoten) oder 125 W (schneller) betragen muss. Es wurden zusätzlich Datenraten und Aufbau des Signals betrachtet, wobei aus Sicht des Beamformings vor allem die Betriebsfrequenzen von 1030 MHz und 1090 MHz sowie die Bandbreite von 6 MHz interessant sind. Der Aufbau des Signals selbst ist festgelegt und bietet daher nur geringen Optimierungsspielraum. Weitere Anforderungen betreffen die Antenne: Diese wird auf die typischerweise verwendeten Monopolantennen von einer Viertel-Wellenlänge bezogen. Der Gewinn einer solchen Antenne beträgt etwa 5,2 dBi [22, S. 218] und darf im Winkelbereich von 5 - 30° Elevation um maximal 3 dB darunter liegen, muss also mindestens 2,2 dBi betragen. Wollte man die Eingangs Idee von planaren Antennen auf den jeweiligen Stirnseiten umsetzen, würde eine Sonderzulassung benötigt, da seitens des Standards Montagen auf und unter dem Rumpf vorgesehen sind [21]. Auch eine

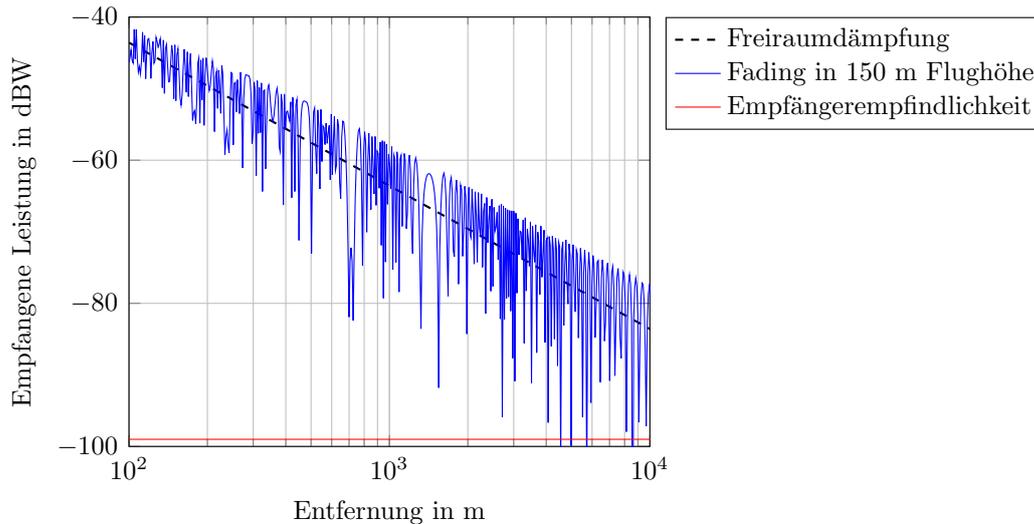


Abbildung 1: Empfangene Leistung unter der Annahme zweier UAVs auf gleicher Flughöhe. Es wird ein Zweiwegemodell angenommen, welches eine LOS und eine Bodenwelle beinhaltet. Zusätzlich sind die Freiraumdämpfung der LOS Komponente sowie die Empfängerempfindlichkeit abgebildet, wie sie in [21] definiert ist.

MIMO-Signalverarbeitung im Sinne der Gewinnoptimierung ist seitens [21] nicht vorgesehen, bzw. beschränkt sich auf die Auswahl der Antenne mit der höheren Empfangsleistung bzw. der früheren Empfangszeit. Dies ändert sich allerdings in Bezug auf den Standard ACAS Xu [23, 24]. Dieser erlaubt zum Beispiel gerichtete Antennencharakteristiken. Um die Notwendigkeit einer gerichteten Transponderabfrage zu evaluieren, wurde ein Zweiwegemodell mit der minimalen Flughöhe von 150 m angenommen. Ergebnisse hinsichtlich der simulierten Empfangsleistung sind in Abb. 1 abgebildet. Es lässt sich erkennen, dass erste Fadingeinbrüche schon vor einer Entfernung von 10 km für Empfangsprobleme sorgen können. Bei einer zugelassenen Höchstgeschwindigkeit von etwa 260 km/h entspricht dies einer Reaktionszeit von etwa 70 Sekunden. Da bereits heute in Gebieten mit hohem Verkehrsaufkommen die Abfragerate und Reichweite reduziert werden, ist eine gerichtete Abfrage in der Zukunft eine vielversprechende Option.

Systemanforderung zwecks Peilung (AP CAU-0 (E1.6.6), AP CAU-1 (E6.2.CAU))

Als Kern des Systems dominiert die angewandte Antenne die Systemanforderungen bezüglich der Peilung. Klassischerweise wird zum Zweck der Peilung ein Antennenarray eingesetzt. Bevorzugt werden hierbei Antennenarrays, die eine regelmäßige Geometrie besitzen (z.B. linear, planar, zirkular). Für diese Art von Antennenarrays wurden in der Literatur präzise und schnelle Algorithmen erforscht und entwickelt. Im Gegensatz dazu verhält sich eine Multimoden-Antenne wie ein Antennenarray, deren einzelne Antennenelemente räumlich unregelmäßig verteilt sind. Infolgedessen sind Peilalgorithmen, die für regelmäßige Arraygeometrien entwickelt wurden, nicht auf Multimoden-Antennen anwendbar. Jedoch eignen sich Schätzer, die unabhängig von der Geometrie des angewandten Antennenarrays auf Basis eines Maximum-Likelihood-Verfahrens den Einfallswinkel schätzen.

Nachteilig von Maximum-Likelihood-Schätzern ist die Berechnungskomplexität, die diese Algorithmen für Echtzeitanwendungen ungeeignet macht.

Demzufolge wurden im Rahmen dieses Arbeitspakets geeignete Algorithmen zur Peilung für beliebige Arraygeometrien untersucht. Der Schwerpunkt der Untersuchungen lag auf der Schätzgenauigkeit verschiedener Algorithmen unter Berücksichtigung folgender Kernparameter: Anzahl der Ziele, Komplexität des Algorithmus, Trennschärfe zwischen benachbarten Zielen und Anzahl der Snapshots. Die Untersuchungen umfassten die folgenden Algorithmen: “Multiple Signal Classification (MUSIC)” [25], “Manifold Separation-Root-MUSIC (MS Root-MUSIC)” [26], “Fourier Domain (FD) Root-MUSIC”, “FD Line-Search MUSIC” und “FD Weighted Least Squares (FWDLS) Root-MUSIC” [27]. Für die Untersuchungen wurde ein Prototyp einer quaderförmigen Multimoden-Antenne verwendet. Die Einfallswinkelschätzung bezog sich auf den Elevationswinkel in einer 2-D Ebene. Zusätzlich wird im folgenden davon ausgegangen, dass die empfangenen Signale unabhängig sind und eine direkte Sichtverbindung (Line of Sight, LOS) vorhanden ist. Weiterhin wird von weißem, gaußverteilterm Rauschen ausgegangen. Als Benchmark für die Schätzgenauigkeit verschiedener Algorithmen wurde die *Cramér-Rao-Schranke* verwendet. Sie ist aus der Schätztheorie bekannt als die untere Schranke für die Varianz eines erwartungstreuen Schätzers [28].

Anzahl der Ziele

Die Anzahl der Antennenelemente des angewandten Arrays ist maßgebend für die Anzahl der Ziele, deren Richtungen mit einer akzeptablen Genauigkeit geschätzt werden können [1]. Das bedeutet, für eine einzelne Multimoden-Antenne muss die Anzahl der Ziele kleiner als die Anzahl der Ports sein. Beispielsweise kann eine Multimoden-Antenne mit $N = 3$ Ports die Einfallswinkel von maximal $L = 2$ Zielen mit akzeptabler Genauigkeit schätzen.

Komplexität des Algorithmus

Im Allgemeinen beruhen alle untersuchten Algorithmen auf dem Prinzip der Eigenzerlegung der Kovarianzmatrix der empfangenen Signale. Sie können in zwei Kategorien eingeteilt werden, nämlich Such-Algorithmen und suchfreie Algorithmen.

MUSIC ist ein klassischer Such-Algorithmus, der die Signal- und Rauschunterräume direkt aus der zerlegten Kovarianzmatrix auswertet. Anschließend wird das sogenannte MUSIC-Nullspektrum berechnet und über einem Winkelraster J evaluiert. Die Einfallswinkel werden aus den Minima der MUSIC-Nullspektrum geschätzt [25]. Zur Kategorie der Such-Algorithmen gehört auch der FD Line-Search MUSIC Algorithmus. Dabei wird das MUSIC-Nullspektrum als eine Fourier-Reihe approximiert und mithilfe einer inversen diskreten Fourier-Transformation (IDFT) effizient berechnet (statt einer Evaluierung für jeden Punkt J des Winkelrasters). Anschließend wird ebenfalls wie bei MUSIC nach den Minima des Nullspektrums gesucht.

Die anderen im Rahmen des Projekts untersuchten Algorithmen gehören zur Kategorie der suchfreien Algorithmen. Bei MS Root MUSIC wird eine Transformation der Manifoldmatrix der Multimoden-Antenne in einer äquivalenten vordefinierten Manifoldmatrix eines Arrays mit einer regelmäßigen Geometrie (z.B. linear, planar, zirkular) durchgeführt. Diese Transformation ermöglicht die Anwendung einer suchfreien Schätzmethode, in der

das Such- und Evaluierungsverfahren durch Bestimmung der Nullstellen eines Polynoms ersetzt wird. Die FD Root-MUSIC und FDWLS Root-MUSIC Algorithmen nutzen die Approximation des MUSIC-Nullspektrums als eine Fourier-Reihe und finden deren Nullstellen, um die Einfallswinkel zu schätzen.

Nachteil der Algorithmen, die eine Transformation (MS Root-MUSIC) oder eine Approximation als eine Fourier-Reihe (FD Root-MUSIC, FD Line-Search MUSIC und FDWLS Root-MUSIC) nutzen, ist die Anfälligkeit für Transformations- oder Approximationsfehler. Diese Fehler können zu einer reduzierten Genauigkeit der Schätzung führen. Die Transformations- und Approximationsfehler sind abhängig von der Anzahl M der Koeffizienten, die jeweils für die Transformation bzw. Approximation genutzt werden. Im weiteren Verlauf des Dokumentes wird der Parameter M der Einfachheit halber als Anzahl der Transformationskoeffizienten bezeichnet. Dieser Wert muss so hoch gewählt werden, dass die Transformationsfehler in einem akzeptablen Rahmen bleiben [27].

Die folgende Tabelle zeigt die Berechnungskomplexität der untersuchten Algorithmen bezogen auf die jeweiligen Kernparameter [27]:

Algorithmus	Berechnungskomplexität
MUSIC	$O(N^3 + JNL)$
MS Root-MUSIC	$O(N^3 + MNL + M^2L + \text{degree-}M \text{ rooting})$
FD Root-MUSIC	$O(N^3 + MNL + M \log_2 M + \text{degree-}M \text{ rooting})$
FD Line-search MUSIC	$O(N^3 + MNL + J \log_2 J)$
FDWLS Root-MUSIC	$O(N^3 + QNL + QM^2 + M^3)$

Hierbei ist N die Anzahl der Antennenelemente (bzw. Ports), L ist die Anzahl der Ziele, M ist die Anzahl der Transformationskoeffizienten, J ist die Anzahl der Suchpunkte im Winkelraster und Q ist die Anzahl der Abtastwerte des MUSIC-Nullspektrums.

Da der Parameter N und darauf folgend der Parameter L in Kooperation mit LUH (Antennenbau) und FAV (Signalverarbeitungshardware) festgelegt wird, haben die Parameter M und J den größten Einfluss auf die Berechnungskomplexität. Deshalb wurde die Komplexität in Abhängigkeit von M und J für die untersuchten Algorithmen analysiert und in Abb. 2 dargestellt.

Der MUSIC-Algorithmus braucht keine Transformation für die Schätzung, deshalb ist seine Komplexität unabhängig von M , wie in Abb. 2(a) zu sehen ist. Die Komplexität der anderen Algorithmen steigt jedoch schnell mit Anstieg des Parameters M . Andererseits weisen die Such-Algorithmen einen schnellen Anstieg der Komplexität mit Anstieg des Parameters J auf, während die suchfreien Algorithmen unabhängig davon sind, siehe Abb. 2(b).

Trennschärfe zwischen den benachbarten Zielen

Um die erreichbare Genauigkeit der untersuchten Algorithmen zu evaluieren, wurde der Einfallswinkel von zwei benachbarten Zielen über einen Bereich des Signal-zu-Rauschleistungsverhältnisses (SNR) von -10 dB bis 60 dB geschätzt. Für jeden SNR-Wert wurde die Wurzel des mittleren quadratischen Fehlers (RMSE) der Einfallswinkelschätzung in Abb. 3 aufgetragen. Abb. 3(a) stellt den Fall von zwei benachbarten Zielen mit 30° Trennung dar und Abb. 3(b) stellt den Fall von zwei benachbarten Zielen mit 5° Trennung dar.

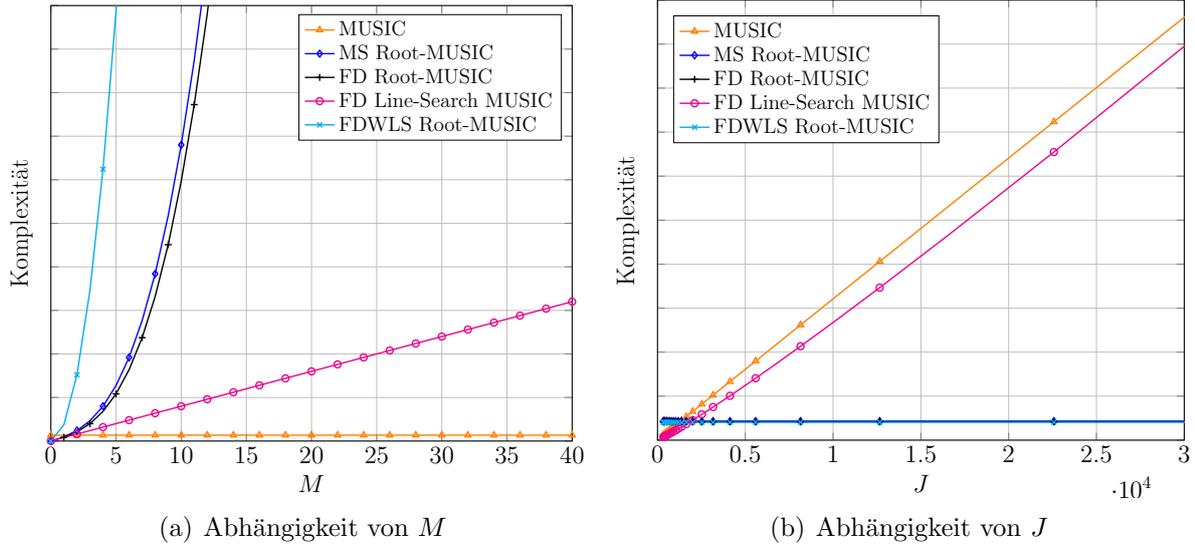


Abbildung 2: Berechnungskomplexität der untersuchten Algorithmen in Abhängigkeit von M und J .

Die Untersuchungen bestätigen, dass einige Algorithmen ab einem bestimmten, hohen SNR keine Verbesserung der Genauigkeit erreichen, sondern eine Sättigung aufweisen. Grund dafür sind Transformations- und Approximationsfehler, die durch eine endliche Anzahl M an Transformationskoeffizienten entstehen. Die Wahl eines größeren M würde die Leistungsfähigkeit verbessern, aber auch die Berechnungskomplexität des Algorithmus steigern, siehe Abb. 2(a).

Anzahl der Snapshots

Im folgenden wurde der Einfluss der Anzahl der Snapshots auf die Genauigkeit der untersuchten Algorithmen verglichen. Bei den Simulationen wurden alle andere Kernparameter festgesetzt. Die Trennung der benachbarten Ziele beträgt 5° und $\text{SNR} = 5$ dB. Abb. 4 zeigt die Schätzgenauigkeit mittels RMSE als Funktion der Anzahl der Snapshots. Ein ähnliches Ereignis der Sättigung geschieht bei MS Root-MUSIC und mit kleinerer Auswirkung als bei den anderen Algorithmen im Fall einer großen Anzahl an Snapshots. Im Allgemeinen erreichen alle Algorithmen mit einer Anzahl der Snapshots zwischen 10^2 und 10^3 eine gute Schätzgenauigkeit.

1.2 HAP6: Cruise-UTM, Datenfusion und Avoid

1.2.1 AP6.2: Konzept Advanced A2A Link (AP CAU-2 (E6.2.CAU))

Im Rahmen der Konzepterstellung für einen Advanced Air-to-Air (A2A) Link wurden neben dem Entwurf der Algorithmen für die Lage- und Distanzschätzung auch die Eigenschaften der Familie der Multimoden-Antennen verwendet, um verbesserte Kommunikation zwischen Interrogatoren auf Basis von Single-Antenna-Element Beamforming [13] zu ermöglichen. Dazu wurde zunächst ein geometrisches Modell entworfen, welches die Lage

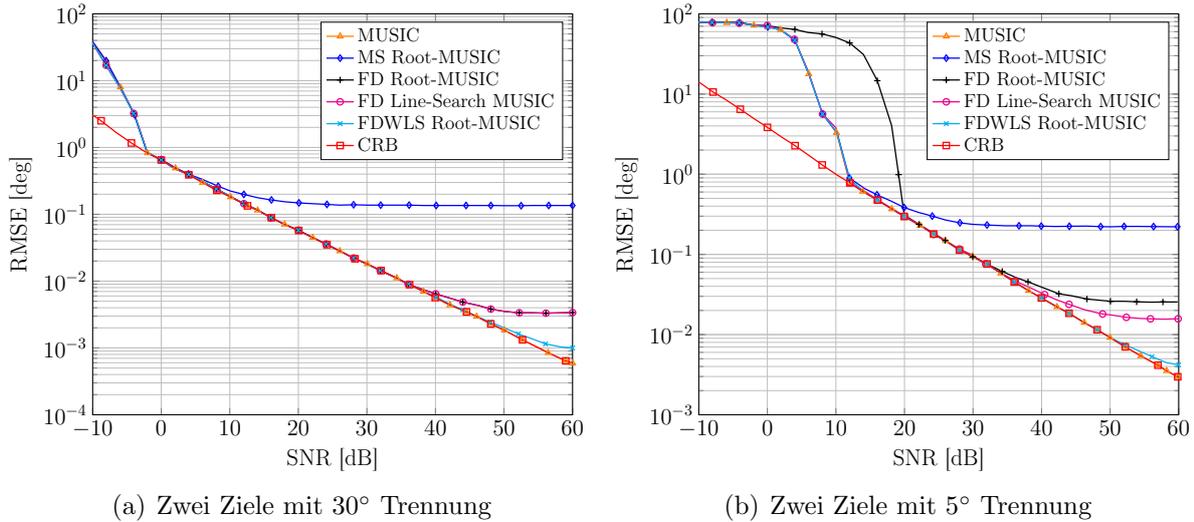


Abbildung 3: RMSE der Einfallswinkelschätzung für zwei benachbarte Ziele.

des Fluggerätes in der Luft sowie die Positionierung der Antennen an Bord basierend auf Manifold-Matrizen berücksichtigt. In Abb. 5 ist die Positionierung der zunächst als planar angenommenen Antennen, sowie deren Normalenvektoren, aus Sicht des Modells dargestellt.

Auf Basis einer solchen Manifold-Matrix wurde eine omnidirektionale Abstrahlcharakteristik unter Verwendung von mehreren planaren Multimoden-Antennen optimiert und für ein einzelnes Antennenelement in einer Antennenmesskammer messtechnisch verifiziert [A]. Die erreichte Performance der omnidirektionalen Charakteristik aus Simulation und Messung kann Abb. 6 entnommen werden. Es wurde gezeigt, dass für einen Winkelbereich von insgesamt 90° eine gute Performance erreicht werden kann.

Das Setup wurde ebenfalls verwendet, um den Antennengewinn für einzelne Winkel basierend auf einem konvexen Problem zu optimieren. Beispielhaft ist das entsprechende Beamforming-Ergebnis in Abb. 7 für einen Winkel von 45° dargestellt.

Die Möglichkeit der Strahlenformung mittels einer Multimoden-Antenne kann auch auf die im Rahmen des Projektes neu entwickelten Antennentypen übertragen werden. Dabei zeigt sich, dass im Vergleich zur Verwendung einzelner Ports ein zusätzlicher Gewinn von etwa 3 dB erzielt werden kann. Der erreichbare Gewinn als Funktion des Winkels ist in Abb. 8 für die für die Realisierung der Antenne gewählten Ports (1,4,5) dargestellt. Eine beispielhafte Charakteristik ist für den Winkel von 45° abgebildet.

Diese Form erlaubt das sektorweise Abfragen anderer Interrogatoren und bietet somit Raum für zukünftige Verbesserungen des Abfrage- und Lokalisierungsalgorithmus, da so Mehrdeutigkeiten bei der Lokalisierung schon bei der Abfrage reduziert werden können.

1.2.2 AP6.3: Implementierung und Test Advanced A2A Link (AP CAU-2 (E6.2.CAU), AP CAU-3 (E6.3.CAU))

In diesem Arbeitspaket wurde bezüglich der untersuchten Algorithmen aus AP1.1.1 zur Einfallswinkelschätzung ein vielversprechender Algorithmus ausgewählt. Anschließend wurde er programmiert und simuliert um das theoretische Konzept zu validieren. Ferner wurde

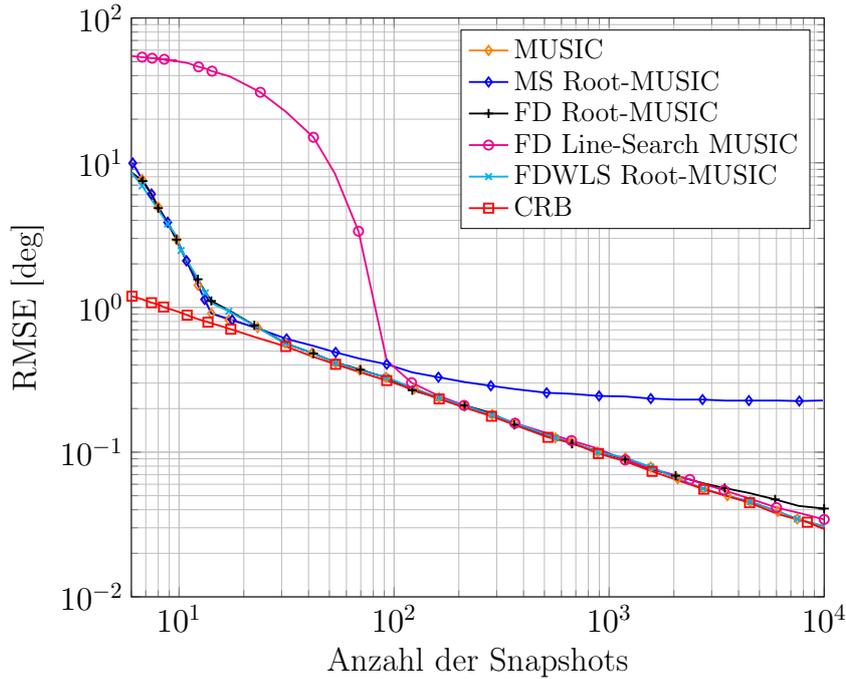


Abbildung 4: RMSE der Einfallswinkelschätzung als Funktion der Anzahl der Snapshots.

ein Algorithmus zur Laufzeitmessung zum Zweck der Entfernungsschätzung vorgeschlagen und ebenfalls programmiert und simuliert. Schließlich wurden die simulierten Algorithmen hinsichtlich einer Implementation optimiert und als ausführbarer Code bereitgestellt.

Als erster Schritt wurden wichtige Parameter für die praktische Anwendung in enger Zusammenarbeit mit den Partnern FAV und LUH diskutiert und erarbeitet. Ein Einbaukonzept von zwei identischen quaderförmigen Multimoden-Antennen, jeweils auf der Ober- und Unterseite des Fluggeräts angebracht, wurde vom Partner LUH vorgeschlagen, siehe Abb. 9. Damit dient der Körper des Fluggeräts als ideale Massefläche für die Multimoden-Antennen, somit lässt sich eine vertikale Polarisation in 360° Azimut erreichen.

Jede der angebrachten Multimoden-Antennen besitzt drei Ports, mit denen sie drei orthogonale elektromagnetische Fernfelder ausstrahlen kann. Das ermöglicht eine getrennte Betrachtung der oberen und unteren Halbräume. Das heißt, jeder Halbraum wird von einer Multimoden-Antenne mit drei orthogonalen Felder bestrahlt, was eine eindeutige Einfallswinkelschätzung ermöglicht. Abb. 10 zeigt die Gewinne der vertikal polarisierten Abstrahlcharakteristik der beschriebenen Multimoden-Antenne im Fernfeld. Nach Aussage des Partners FAV hält sich der Aufwand aus Hardware-Sicht zur Realisierung der notwendigen HF-Ketten zwischen den Ports und der Signalverarbeitungseinheit für das vorgeschlagene Konzept in angemessenem Rahmen.

Im nächsten Schritt bekam die CAU vom Partner LUH Simulationsdaten des Prototypen der oben beschriebenen quaderförmigen Multimoden-Antenne. Die Simulationsdaten beinhalteten die elektromagnetischen Fernfelder der vertikalen und horizontalen Komponenten im oberen Halbraum, siehe Abb. 10. Für die Simulation der Fernfelder wurde der obere Halbraum in 5° Schrittweite in Elevation (θ) und Azimut (ϕ) quantisiert. Für jede quantisierte Richtung (θ, ϕ) wurden die Fernfelder anschließend simuliert und gespei-

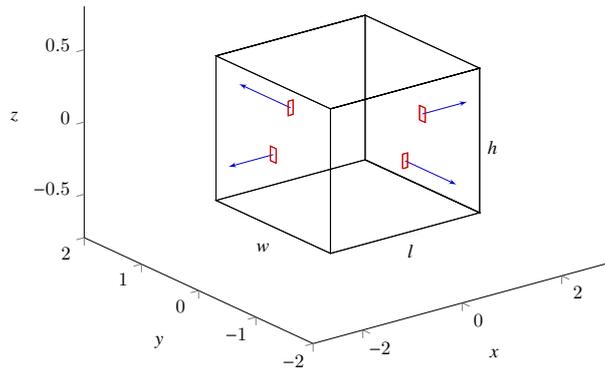
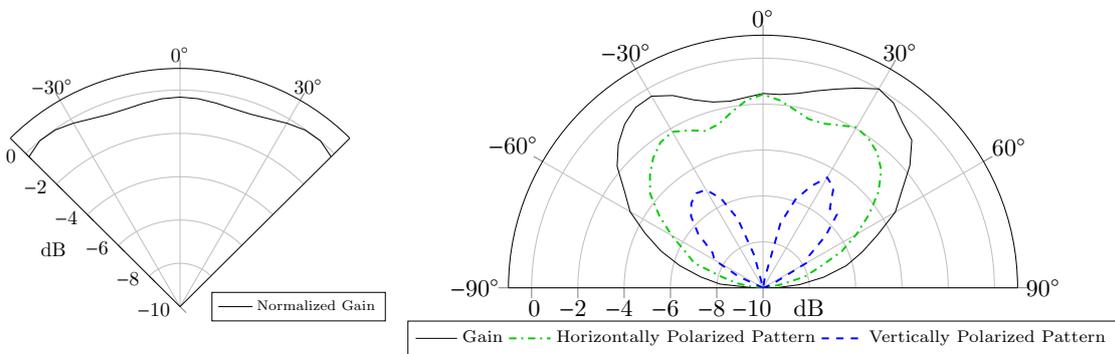


Abbildung 5: Antennenpositionen auf einem Quader als Modell eines Fluggerätes, aus [A].



(a) Theoretisches Ergebnis.

(b) Gemessenes Ergebnis.

Abbildung 6: Normierte Abstrahlungscharakteristik optimiert für omnidirektionale Abstrahlung, aus [A].

chert. Deshalb wurden die Simulationsdaten für die Analyse der Einfallswinkelschätzung mithilfe der “Wavefield Modellierungs”-Methode (WM) interpoliert [26]. Ziel dieser Interpolation ist die Prüfung der erreichbaren Genauigkeit mit dem entworfenen Prototyp und die Berechnung der unteren Schranke (CRB) des angewandten Algorithmus.

Um den geeignetsten Algorithmus für die Simulations- und Implementierungs-Software zwecks der Einfallswinkelschätzung auszuwählen, wurden die Ergebnisse aus AP1.1.1 analysiert. Ziel des Auswahlprozesses war es, einen Kompromiss zu finden zwischen Berechnungskomplexität und Schätzgenauigkeit des Algorithmus. Wie in AP1.1.1 beschrieben, hängt die Berechnungskomplexität im betrachteten Fall hauptsächlich von der Anzahl M der Transformationskoeffizienten und der Anzahl J der Suchpunkte im Winkelraster ab, siehe Abb. 2. Die erreichbare Schätzgenauigkeit, in Abb. 3 für $M = 25$ dargestellt, zeigt, dass für eine gute Schätzgenauigkeit über großem SNR-Bereich eine höhere Anzahl von M nötig ist. Allerdings zeigt Abb. 2(a), dass ein großer Wert des Parameters M die Berechnungskomplexität für transformationsbasierte oder approximationsbasierte Algorithmen extrem erhöhen würde. Darüber hinaus zeigt Abb. 2(a) die Unabhängigkeit des MUSIC-Algorithmus vom Parameter M . Außerdem weist MUSIC eine präzise Schätzgenauigkeit über einen großen SNR-Bereich auf, siehe Abb. 3. Nachteil

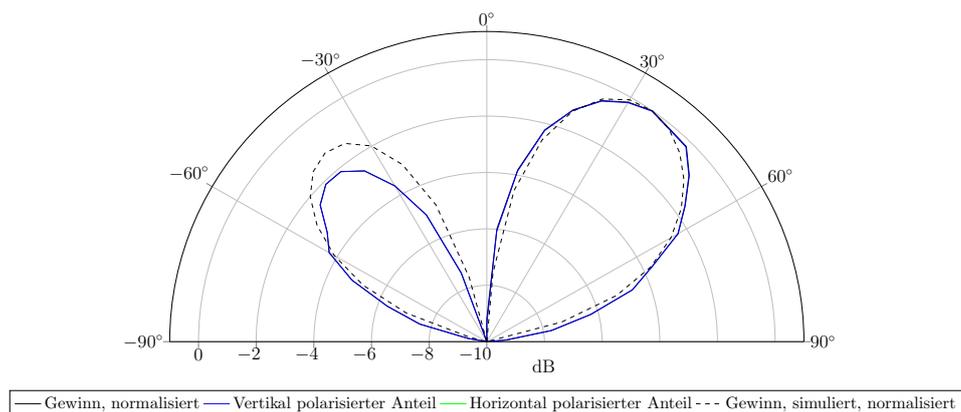


Abbildung 7: Abstrahlungscharakteristik optimiert für einen Winkel von 45° , aus [C]. Es zeigt sich eine gute Übereinstimmung zwischen den Ergebnissen von Simulation und Messung.

bei MUSIC ist die Abhängigkeit vom Parameter J . Entsprechend der Simulationsdaten des Partners LUH, ergibt die 5° -Quantisierung des oberen Halbraums ein Winkelraster mit $J = 1297$ Richtungen (θ, ϕ) . Laut Abb. 2(b) liegt die Berechnungskomplexität bei diesem Wert noch im vergleichbaren Raum mit den suchfreien Algorithmen in dem betrachteten Fall. Darauf basierend wurde MUSIC als ein guter Kompromiss zwischen Berechnungskomplexität und Schätzgenauigkeit ausgewählt, und dementsprechend bei der Simulations- und Implementierungs-Software angewandt.

Die Simulations-Software wurde in MATLAB-Code programmiert und getestet. Die Ergebnisse der Einfallswinkelschätzung sind in Abb. 11 dargestellt.

Für die Simulation wurden sog. Mode S Antwortsignale [29] angenommen. Die Signale werden mithilfe der vertikal polarisierten elektromagnetischen Fernfelder der quaderförmigen Multimoden-Antenne empfangen. Für jeden SNR-Wert wird der Signalempfang von jeder Einfallrichtung (θ, ϕ) im Winkelraster simuliert. Für jede Einfallrichtung (θ, ϕ) werden der Elevations- und Azimutwinkel $(\hat{\theta}, \hat{\phi})$ gemeinsam geschätzt. Im Anschluss werden für jeden SNR-Wert die Schätzfehler von $\hat{\theta}$ und $\hat{\phi}$ im oberen Halbraum gemittelt. Die RMSE-Werte und die entsprechenden CRB-Werte von $\hat{\theta}$ sind in Abb. 11(a) aufgetragen und entsprechend von $\hat{\phi}$ in Abb. 11(b).

Nach Schätzung des Einfallswinkels wird die Entfernung des Ziels mithilfe einer Laufzeitmessung geschätzt. Die Laufzeitmessung wird durch Berechnung der Umlaufzeit (Round-Trip Delay, RTD) ermöglicht. Dabei handelt es sich um die Zeitdifferenz zwischen der Aussendung einer Abfrage und dem Erhalten der Antwort. Unter der Annahme, dass die Bearbeitungszeit der Abfrage bei dem abgefragten Luftverkehrsteilnehmer bekannt ist, kann die Umlaufzeit zwischen Sender und Empfänger gemessen werden. Die Distanz ergibt sich dann durch Teilung der Umlaufzeit durch zwei und Multiplikation des Ergebnisses mit der Lichtgeschwindigkeit. Der entworfene Algorithmus zur Entfernungsschätzung geht von einem Mode S Antwortsignal aus. Das bedeutet, die Struktur der Antwort ist bekannt und besteht hauptsächlich aus zwei Teilen: der Präambel und dem Datenblock. Die Präambel hat ein einheitliches Impulsmuster der Länge $8 \mu\text{s}$ [29]. Daher kann die Umlaufzeit durch eine Kreuzkorrelation zwischen der empfangenen Antwort und der bekannten Präambel bestimmt werden. Im Unterschied zur Einfallswinkelschätzung reicht es für die

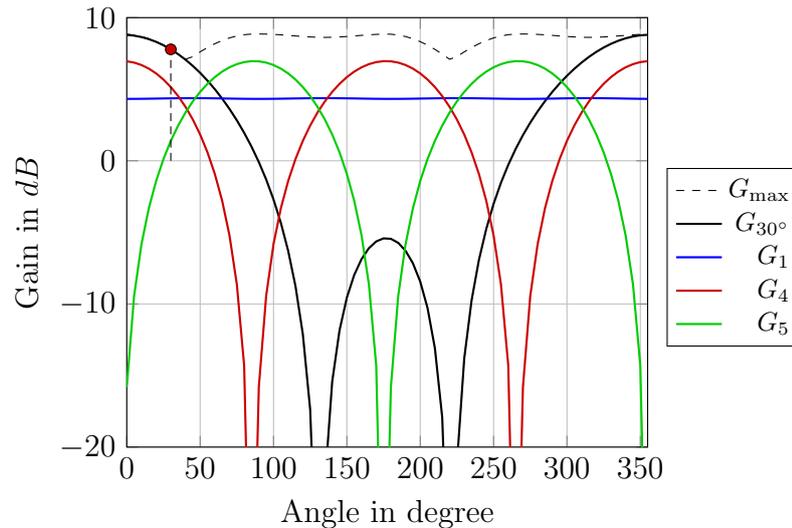


Abbildung 8: Gewinne der Ports 1, 4 und 5 sowie der Kombination aus diesen. Es kann ein zusätzlicher Gewinn durch geschickte Kombination und Gewichtung der Ports erreicht werden. Eine Beispielcharakteristik ist für den Winkel von 30° abgebildet.

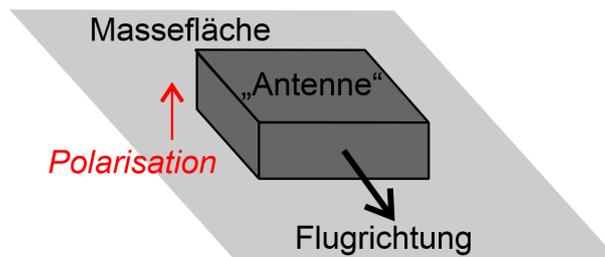


Abbildung 9: Einbaukonzept der quaderförmigen Multimoden-Antenne auf der Ober- und Unterseite des Fluggeräts.

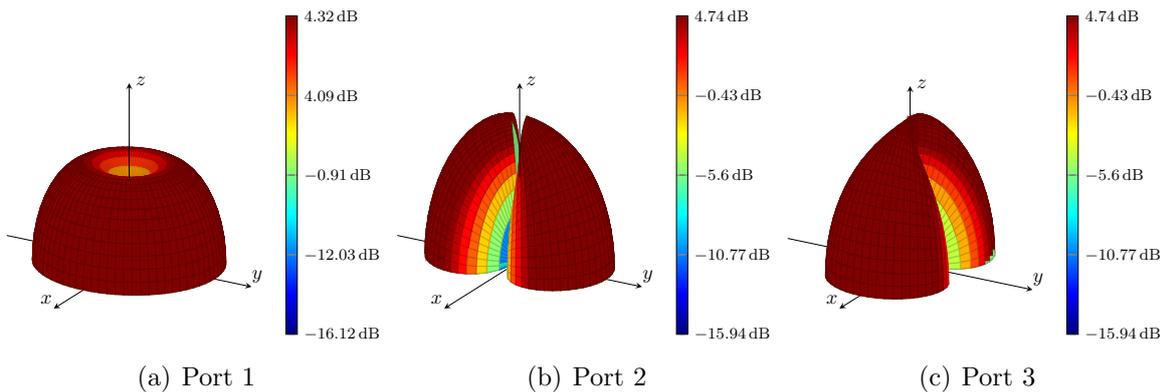
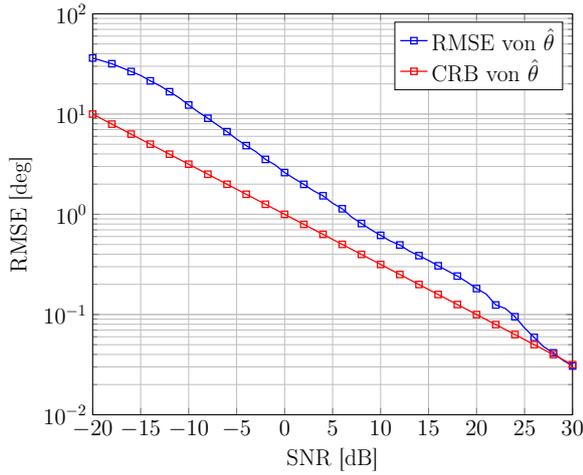
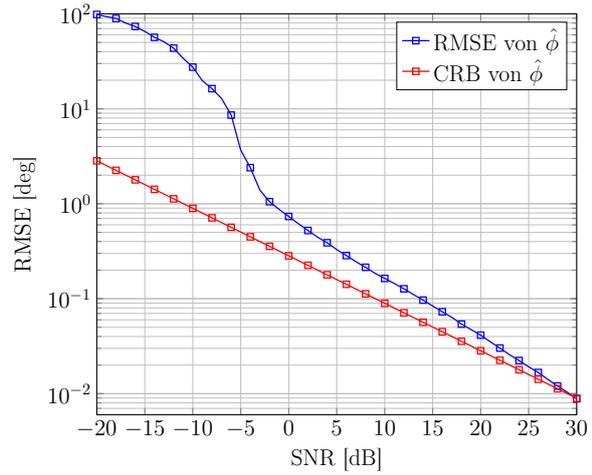


Abbildung 10: Vertikal polarisierte Abstrahlcharakteristik des Antennengewinns der betrachteten Multimoden-Antenne.

Entfernungsschätzung, die empfangene Antwort von einem der drei verfügbaren Ports auszulesen. Deshalb entscheidet sich der Algorithmus mithilfe der Einfallswinkelschätzung für den Port mit dem höchsten Gewinn in der geschätzten Richtung. In Nachfolgearbeiten



(a) RMSE und CRB des Elevationswinkels θ



(b) RMSE und CRB des Azimutwinkels ϕ

Abbildung 11: RMSE und CRB der Einfallswinkelschätzung im 3D-Raum.

wäre auch eine Anwendung des für die Kommunikation gezeigten Beamformings denkbar.

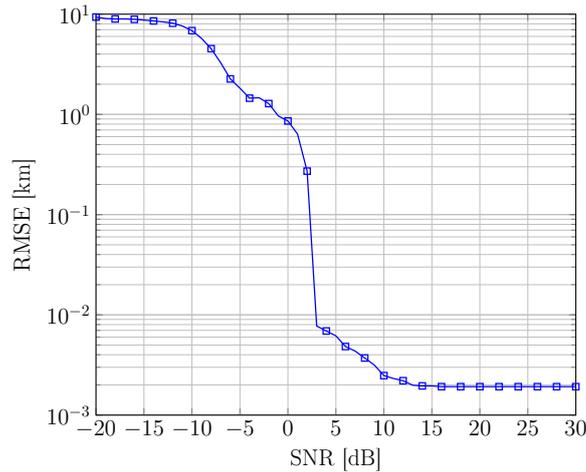


Abbildung 12: RMSE der Entfernungsschätzung.

Bei der Simulation wurde die Entfernung nach jeder Einfallswinkelschätzung ($\hat{\theta}$, $\hat{\phi}$) mit der besten Empfangsversion der Antwort geschätzt und der Fehler wurde ermittelt. Anschließend wurden die Schätzfehler für jeden SNR-Wert gemittelt und als RMSE als Funktion des SNR in Abb. 12 dargestellt.

Optimierung und Bereitstellung des entworfenen Algorithmus (AP CAU-4 (E6.3.3))

Nachdem die entworfenen Algorithmen zur Einfallswinkel- und Entfernungsschätzung in einer MATLAB-Umgebung programmiert und simuliert wurden, erfolgte eine Optimierung der Software für die praktische Implementierung. Bei den theoretischen Simulationen wurde der Empfang aus jeder Richtung (θ, ϕ) im Winkelraster simuliert. Das Winkelraster

beinhaltet, wie oben beschrieben, die quantisierten Richtungen mit einer 5° Schrittweite. Das ergibt $J = 1297$ Richtungen, in denen das MUSIC-Nullspektrum evaluiert wird. Laut Abb. 2 hat der Algorithmus damit eine relativ kleine Berechnungskomplexität. Allerdings kann die Berechnungskomplexität steigen wenn eine Auflösung von geringer als 5° gewünscht ist. Beispielsweise wenn die gemessenen Fernfelder für die echte Multimoden-Antenne in 1° Auflösung verfügbar sind. Das ergibt ein Winkelraster mit $J = 32041$ Richtungen in denen das MUSIC-Nullspektrum evaluiert werden muss. Das würde zu einem erheblichen Anstieg der Berechnungskomplexität führen. Um in dem Fall die Berechnungskomplexität gering zu halten, wurde der Algorithmus optimiert. Nach der ersten Schätzung der Richtung im groben Winkelraster mit 5° Auflösung, definiert der Algorithmus ein kleineres Winkelraster mit einer feineren Auflösung (in diesem Beispiel 1°) im umgebenden Raum der geschätzten Richtung. Danach wird die Richtung ausschließlich im neu definierten Raum mit der feineren Auflösung erneut geschätzt. Somit wird ein aufwendiger Such- und Evaluierungsprozess über dem gesamten Winkelraster $J = 32041$ vermieden und die Berechnungskomplexität bleibt nahezu unverändert, denn der Such- und Evaluierungsprozess über dem neu definierten Winkelraster hat eine vernachlässigbare Berechnungskomplexität.

Die optimierte Simulations-Software, siehe Anhang B, wurde für die praktische Implementierung angepasst und als ausführbarer MATLAB-Code, siehe Anhang C, dem Projektpartner FAV bereitgestellt. Zusätzlich wurde die MATLAB-basierte Software auf Wunsch des Partners FAV in C-Code übersetzt und diesem ebenfalls zur Verfügung gestellt.

2 Wichtigste Positionen des zahlenmäßigen Nachweises

Der wesentliche Teil der Kosten dieses Teilprojekts entstand durch Personalausgaben. Während der gesamten Laufzeit wurde ein Wissenschaftlicher Mitarbeiter aus dem Projekt finanziert. Ferner fielen Kosten für turnusmäßige Treffen mit Projektpartnern an. Details finden sich in den Ausgabenerklärungen unter der Erteilungsnummer 20D19051 für die Jahre 2020, 2021 und 2022.

3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die durchgeführten Forschungsarbeiten innerhalb des Verbundprojekts sowie die dafür aufgewandten Ressourcen waren notwendig und angemessen, da sie der im Projektantrag formulierten Planung entsprachen und alle wesentlichen im Arbeitsplan formulierten Aufgaben erfolgreich bearbeitet wurden. Darüber hinaus waren keine zusätzlichen Ressourcen für das Projekt notwendig bzw. verfügbar.

4 Verwertbarkeit der Ergebnisse

Die geleisteten Arbeiten flossen, neben der oben beschriebenen Ausarbeitung der Anforderungen an die Systemkomponenten sowie der Konzepterstellung, im Wesentlichen in die Entwicklung eines MATLAB-basierten Softwarepakets. Mit Hilfe dieses Softwarepakets wurde mittels Simulation die Funktionsfähigkeit der entwickelten Algorithmen verifiziert. Auf Wunsch des Projektpartners FAV wurde das Softwarepaket zusätzlich in C-Code übersetzt, um eine praktische Implementierung auf Seiten des Industriepartners zu ermöglichen. Beide Softwarepakete wurden der FAV GmbH zur Verfügung gestellt.

Es wurde ein ausführliches Journalpaper in der angesehenen Fachzeitschrift IEEE Transactions on Antennas and Propagation veröffentlicht, das sowohl die Kanalmodellierung als auch eine Optimierung und Messung von Beamforming für Multimoden-Antennen enthält [A]. Ferner wurde ein Artikel zur gemeinsamen Kommunikation, Sensorik und Lokalisierung für luftgestützte Anwendungen in arXiv veröffentlicht [B]. Darüber hinaus wurde ein IEEE Konferenzbeitrag über Konzepte zum Signal von Luftfahrzeugtranspondern publiziert [C].

5 Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen

Im Projektzeitraum sind uns keine Kenntnisse über Ergebnisse Dritter bekannt geworden, welche für die Durchführung des Vorhabens relevant waren. Neben dem Studium einschlägiger Fachzeitschriften wurde hierfür die Datenbank IEEEExplore regelmäßig durchsucht.

6 Erfolgte und geplante Veröffentlichungen der Ergebnisse

Literatur

- [A] N. L. Johannsen, S. A. Almasri, and P. A. Hoeher, “Geometry-based UAV MIMO channel modeling for multi-mode antennas,” accepted for publication in IEEE Transactions on Antennas and Propagation, 2022, doi: 10.1109/TAP.2022.3191418.
- [B] N. L. Johannsen, M. Schurwanz, L. Grundmann, J. Mietzner, D. Manteuffel, and P. A. Hoeher, “Joint communication, sensing and localization for airborne applications,” arXiv:2209.10991, Sep. 2022, doi:10.48550/arXiv.2209.10991.
- [C] N. L. Johannsen, L. Grundmann, D. Manteuffel, and P. A. Hoeher, “Aircraft transponder signaling concepts using multi-mode multi-port antenna,” in Proc. 2022 International Conference on Information and Communication Technology Convergence (ICTC), Oct. 2022.

Anhang

A Literaturliste

- [1] T. E. Tuncer and F. Friedlander, *Classical and Modern Direction-of-Arrival Estimation*. Academic Press, 2009.
- [2] S. Aditya, A. F. Molisch, and H. M. Behairy, “A survey on the impact of multipath on wideband time-of-arrival based localization,” *Proceedings of the IEEE*, vol. 106, no. 7, pp. 1183–1203, 2018.
- [3] Y. Chen and C. F. Wang, *Characteristic Modes: Theory and Applications in Antenna Engineering*. Wiley, 2015.
- [4] R. Martens and D. Manteuffel, “Systematic design method of a mobile multiple antenna system using the theory of characteristic modes,” *IET Microwaves, Antennas and Propagation*, vol. 8, no. 12, pp. 887–893, Sep. 2014.
- [5] D. Wen, Y. Hao, H. Wang, and H. Zhou, “Design of a wideband antenna with stable omnidirectional radiation pattern using the theory of characteristic modes,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 5, pp. 2671–2676, 2017.
- [6] F. A. Dicandia, S. Genovesi, and A. Monorchio, “Advantageous exploitation of characteristic modes analysis for the design of 3-D null-scanning antennas,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 8, pp. 3924–3934, 2017.
- [7] D.-W. Kim and S. Nam, “Systematic design of a multiport MIMO antenna with bilateral symmetry based on characteristic mode analysis,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 3, pp. 1076–1085, 2018.
- [8] D. Manteuffel and R. Martens, “Compact multimode multielement antenna for indoor UWB massive MIMO,” *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 7, pp. 2689–2697, 2016.
- [9] P. Hoeher and N. Doose, “A massive MIMO terminal concept based on small-size multi-mode antennas,” *Transactions on Emerging Telecommunications Technologies*, vol. 28, Mar. 2015.
- [10] P. A. Hoeher, D. Manteuffel, N. Doose, and N. Peitzmeier, “Ultra-wideband massive MIMO communications using multi-mode antennas,” *Frequenz*, vol. 71, no. 9-10, pp. 439–448, 2017. [Online]. Available: <https://doi.org/10.1515/freq-2017-0148>
- [11] N. L. Johannsen, N. Doose, and P. A. Hoeher, “Highly-efficient hybrid beamforming by port selection in massive MIMO multi-mode antenna systems,” in *Proc. 12th International ITG Conference on Systems, Communications and Coding (SCC 2019)*, 2019, pp. 1–6.
- [12] N. L. Johannsen, N. Peitzmeier, P. A. Hoeher, and D. Manteuffel, “On the feasibility of multi-mode antennas in UWB and IoT applications below 10 GHz,” *IEEE Communications Magazine*, vol. 58, no. 3, pp. 69–75, Mar. 2020.

- [13] N. L. Johannsen and P. A. Hoeher, “Single-element beamforming using multi-mode antenna patterns,” *IEEE Wireless Communications Letters*, vol. 9, no. 7, pp. 1120–1123, Jul. 2020.
- [14] N. L. Johannsen, N. Peitzmeier, P. A. Hoeher, and D. Manteuffel, *M4 – Multi-Mode Massive MIMO*, Book Chapter 6 in *Wireless 100 Gbps and Beyond*. IHP – Innovations for High Performance Microelectronics, 2020.
- [15] S. Alkubti Almasri, N. Doose, and P. A. Hoeher, “Parametric direction-of-arrival estimation for multi-mode antennas,” in *Proc. 14th Workshop on Positioning, Navigation and Communications (WPNC)*, 2017, pp. 1–5.
- [16] S. Alkubti Almasri, R. Pöhlmann, N. Doose, P. A. Hoeher, and A. Dammann, “Modeling aspects of planar multi-mode antennas for direction-of-arrival estimation,” *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4585–4597, 2019.
- [17] R. Pöhlmann, S. A. Almasri, S. Zhang, T. Jost, A. Dammann, and P. A. Hoeher, “On the potential of multi-mode antennas for direction-of-arrival estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 5, pp. 3374–3386, 2019.
- [18] S. Alkubti Almasri and P. A. Hoeher, “Impact of phase center displacement on direction-of-arrival estimation using multi-mode antennas,” in *Proc. 24th International ITG Workshop on Smart Antennas (WSA 2020)*, 2020, pp. 1–4.
- [19] R. Francis, R. Vincent, J.-M. Noël, P. Tremblay, D. Desjardins, A. Cushley, and M. Wallace, “The flying laboratory for the observation of ADS-B signals,” *International Journal of Navigation and Observation*, vol. 2011, 2011.
- [20] DO-260B, “Minimum operational performance standards for 1090 MHz extended squitter automatic dependent surveillance—broadcast (ADS-B) and traffic information services—broadcast (TIS-B),” *Tech. Rep.*, 2009.
- [21] The European Organisation for Civil Aviation Equipment, “Minimum operational performance specification for secondary surveillance radar mode S transponders,” *ED-73C*, May 2008.
- [22] C. A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed. John Wiley & Sons, Inc., 2016.
- [23] EUROCAE, “Minimum operational performance standards for airborne collision avoidance system Xu (ACAS Xu),” *ED 275*, vol. 1, Dec. 2020.
- [24] G. Manfredi and Y. Jestin, “An introduction to ACAS Xu and the challenges ahead,” in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sep. 2016, pp. 1–9.
- [25] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [26] M. Doron and E. Doron, “Wavefield modeling and array processing. I. spatial sampling,” *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2549–2559, 1994.

- [27] M. Rubsamen and A. B. Gershman, “Direction-of-arrival estimation for nonuniform sensor arrays: From manifold separation to Fourier domain MUSIC methods,” *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 588–599, 2009.
- [28] S. M. Kay, *Fundamentals of Statistical Signal Processing — Estimation Theory*. Prentice Hall, 1993.
- [29] C. Wolff. (1998) Mode S Antworttelegramm. [Online]. Available: <https://www.radartutorial.eu/13.ssr/sr24.de.html>

B MATLAB-Software für Simulationen

B.1 simDoaDistEstim.m

```
% The script simDoaDistEstim simulates the direction of arrival (elevation
% (theta) and azimuth (phi)) and distance estimation of the signal IQrx
% provided by FAV impinging on the cubid multi-mode antenna designed by LUH
% over a range of SNR values.
%
% The signal IQrx is assumed to be received by the simulated multi-mode
% antenna (MMA) pattern. These patterns are simulated in a quantized manner
% over grid of points in the upper hemisphere. The grid has a step size of
% 5 degrees in both azimuth and elevation directions.
%
% Author: saaa@tf.uni-kiel.de

Demo_main % Script provided by FAV for generating the signal IQrx
% get MMA Patterns and array response matrix
gndPlane = 'xy'; % ground plane
% Two possible mode combinations ([1 2 3] or [1 4 5])
modeComb1 = [1 2 3];
% modeComb2 = [1 4 5];
chosenComb = modeComb1; % Chosen combination after dicussion with FAV
% and LUH
nModes = length(chosenComb); % Number of modes
pol = 'theta'; % Vertical Polarization
[~,~,~,AMM,~,phiMMDeg,~,thetaMMDeg,~] = getPattern3D(gndPlane,pol); %
% Obtain the array response matrix AMM (manifold matrix)

% Interpolate MMA patterns using Wavefield modelling:
% The objective of wavefield modelling is to obtain the radiation
% pattern at any arbitrary direction, not only at the grid points of 5
% degrees step size, at which the MMA was simulated by LUH.
angResFine = 1; % step size of the interpolated fine angular grid [deg]
NPhi = 5; % Phi coefficients
NTheta = 5; % theta coefficients, phi and theta coefficients are
% necessary for the wavefiled modelling approach
[AInterp,~,G,~,phiFineGridDeg,~,thetaFineGridDeg] = wm3D(angResFine, ...
    pol,NPhi,NTheta); % Interpolate the array response matrix for the
% fine grid search

% Simulation of DoA Estimation
% The signals are received via the array response matrix, calculated
% from the simulated readiation patterns provided by LUH.
% Since the radiation patterns are provided in 5 degree step size in
% both elevation and azimuth directions, theta needs to be a multiple of
% 5 between 0 and 90 and phi needs to be a multiple of 5 between 0 and
% 355
Nruns = 10; % Number of simulation runs
SNRdB_vec = -20:30; % SNR range in dB
thetaMSEvsAngSnr = zeros(length(thetaMMDeg),length(SNRdB_vec)); % Theta
% MSE for each angle vs SNR
phiMSEvsAngSnr = zeros(length(thetaMMDeg),length(SNRdB_vec)); % Phi MSE
% for each angle vs SNR
distMSEvsSNR = zeros(length(thetaMMDeg),length(SNRdB_vec)); % Distance
% MSE for each angle vs SNR
```

```

SNRlin_vec = 10 .^(SNRdB_vec./10); % Linear SNR
N0_vec = sqrt(1./(2*SNRlin_vec)); % Noise standard deviaton
r = zeros(nModes,length(IQrx)); % Initialize received signal vector
threshold = 0.8; % Threshold for peak detection in the cross correla-
% tion function
trueDist = config.sim_dist; % True distance defined in Demo_main output
% structure "config"
% Loop over SNR
for SNR_ind = 1:length(SNRdB_vec)
    tic
    SNR = SNRdB_vec(SNR_ind);
    N0 = N0_vec(SNR_ind);
    disp(['SNR=' num2str(SNR) 'dB'])
    thetaMSE = zeros(length(thetaMMDeg),1); % Theta MSE of each angle
    phiMSE = zeros(length(thetaMMDeg),1); % Phi MSE of each angle
    distMSE = zeros(length(thetaMMDeg),1); % Distance MSE of each angle
    % loop over angular points
    parfor angIdx = 1:length(thetaMMDeg)
        theta = thetaMMDeg(angIdx); % Elevation of the angular point
        phi = phiMMDeg(angIdx); % Azimuth of the angular point

        % Simulation runs
        thetaErrorDeg = zeros(Nruns,1); % Theta error over simulation
        % runs
        phiErrorDeg = zeros(Nruns,1); % Phi error over simulation
        % runs
        distError = zeros(Nruns,1); % Distance error over simulation
        % runs
        for nRun=1:Nruns % Loop over simulation runs

            % Received signal
            r = AMM(angIdx,chosenComb).' * IQrx + N0 * randn(nModes, ...
                length(IQrx)); % Received IQrx with the MMA plus receiver
            % noise

            %% DoA estimation
            R = (1/size(r,2)) * (r*r'); % Covariance matrix of the
            % received signal
            [V,D]=eig(R); % Eigen decomposition of the covariance
            % matrix
            [~,I] = max(D,[],'all'); % Find the largest eigen value
            [~,colMax]=ind2sub(size(D),I); % Determine the signal
            % eigenvector(signal subspace)
            V(:,colMax) = []; % Extract the noise subspace
            NN = V; % Assign the noise subspace to NN

            % MUSIC spectrum
            PMusic = sim_music_cost3D(phiFineGridDeg,thetaFineGridDeg, ...
                NN, G(:,chosenComb), NPhi, NTheta); % Calculate the
            % MUSIC null-spectrum at the points of the interpolated
            % fine grid
            [~, minIdx] = min(PMusic); % Find the minimum of the music
            % spectrum
            thetaEstimCoarse = thetaFineGridDeg(minIdx); % Coarse esti-
            % mate of theta
            phiEstimCoarse = phiFineGridDeg(minIdx); % Coarse estimate
            % of phi

            %% Optimization

```

```

% Define a smaller search intrerval around the coarse esti-
% mates calculate the MUSIC spectrum over it with finer steps,
% then look for the minimum of the null-spectrum
% Define the finer ranges:
fineSearchinterval = 1;
lowerBndTheta = thetaEstimCoarse-fineSearchinterval;
lowerBndTheta(lowerBndTheta < 0) = 0;
upperBndTheta = thetaEstimCoarse+fineSearchinterval;
upperBndTheta(upperBndTheta > 90) = 90;
lowerBndPhi = phiEstimCoarse-fineSearchinterval;
lowerBndPhi(lowerBndPhi < 0) = 0;
upperBndPhi = phiEstimCoarse+fineSearchinterval;
upperBndPhi(upperBndPhi > (360-fineSearchinterval)) = ...
360-fineSearchinterval;

% Optimize the results by searching over the finer grid
costFuncWrapper = @(args) sim_music_cost3D(args(1), ...
    args(2), NN, G(:,chosenComb), NPhi, NTheta); % Define a
% function handle for the cost function
options = optimoptions(@fmincon,'Algorithm','sqp', ...
    'StepTolerance', 1e-6, 'Display', 'off'); % Optimization
% options
[argMin,~] = fmincon(costFuncWrapper, [phiEstimCoarse ...
    thetaEstimCoarse], [], [], [], [], [lowerBndPhi ...
    lowerBndTheta], [upperBndPhi upperBndTheta], [], ...
    options); % Find the minima of the fine null-spectrum

phiHat = argMin(1); % Fine estimate of theta
thetaHat = argMin(2); % Fine estimate of phi

% Calculate errors
thetaErrorDeg(nRun) = abs(theta-(thetaHat)); % Theta error
phiErrorDeg(nRun) = abs(phi-(phiHat)); % Phi error

%% Distance estimation
gainDoa = abs(AInterp(minIdx,chosenComb)); % Gain based on
% the estimated DoA
 [~,bestPort] = max(gainDoa); % Find the port with the maxi-
% mum gain
[c, lags] = xcorr(abs(r(bestPort,:)),abs(preamble)); % Cross
% correlation between the received signal over the port with
% maximum gain and the preamble
[m,~] = max(abs(c)); % peak point of the cross correlation
% function
corrNorm = c/m; % Normalize the cross correlation function

 [~,indNorm] = find(abs(corrNorm)>threshold);
delaySamples = lags(indNorm(1)); % delay in samples
delaySeconds = delaySamples/config.fs; % delay in seconds
distance = delaySeconds * physconst('LightSpeed'); % distance
% in meters

% Distance estimation error
distError(nRun) = abs(distance - trueDist);
end
thetaMSE(angIdx) = sum(thetaErrorDeg.^2)/Nruns; % Theta MSE
phiMSE(angIdx) = sum(phiErrorDeg.^2)/Nruns; % Phi MSE
distMSE(angIdx) = sum(distError.^2)/Nruns; % Distance MSE
end

```

```

thetaMSEvsAngSnr(:,SNR_ind) = thetaMSE; % Theta MSE vs SNR
phiMSEvsAngSnr(:,SNR_ind) = phiMSE; % Phi MSE vs SNR
distMSEvsSNR(:,SNR_ind) = distMSE; % Distance MSE vs SNR
etime = toc;
disp(['time= ' num2str(etime) ' sec'])
end

% Calculate CRB
[CRLBThetavsSNR, CRLBPhivvsSNR] = calcCRLB3D_det(phiMMDeg, thetaMMDeg, ...
    SNRdB_vec, length(IQrx), G(:,chosenComb), NPhi, NTheta); % Calculate CRB
% for theta and phi

thetaMSEvsSNR_mean = mean(thetaMSEvsAngSnr(thetaMMDeg<85,:), 1); % Mean of
% theta MSEs in the upper Hemisphere
phiMSEvsSNR_mean = mean(phiMSEvsAngSnr(phiMMDeg~=0,:), 1); % Mean of phi MSEs
% in the upper Hemisphere
CRLB_theta_mean = mean(CRLBThetavsSNR(thetaMMDeg<85,:), 1); % Mean of theta
% CRBs in the upper Hemisphere
CRLB_phi_mean = mean(CRLBPhivvsSNR(phiMMDeg~=0,:), 1); % Mean of phi CRBs in
% the upper Hemisphere
distMSEvsSNRMean = mean(distMSEvsSNR,1); % Mean of distance MSEs in the
% upper Hemisphere

% plot the results
figure;
semilogy(SNRdB_vec, sqrt(thetaMSEvsSNR_mean)) % Plot theta RMSE
grid on, hold on
semilogy(SNRdB_vec, sqrt(CRLB_theta_mean), '--') % Plot theta CRB
xlabel('SNR [dB]'), ylabel('RMSE(\theta) [deg]')
title('Estimation of theta')
legend('RMSE \theta', 'CRB \theta')

figure
semilogy(SNRdB_vec, sqrt(phiMSEvsSNR_mean)) % Plot phi RMSE
grid on, hold on
semilogy(SNRdB_vec, sqrt(CRLB_phi_mean), '--') % Plot phi CRB
xlabel('SNR [dB]'), ylabel('RMSE(\phi) [deg]')
title('Estimation of phi')
legend('RMSE \phi', 'CRB \phi')

figure
semilogy(SNRdB_vec, sqrt(distMSEvsSNRMean)) % Plot distance RMSE
grid on, hold on
title('Distance Estimation')
xlabel('SNR [dB]'), ylabel('Distance RMSE [m]')

```

B.2 Demo_main.m

Dieses MATLAB-Skript und die darin enthaltenen Funktionen wurden der CAU vom Partner FAV zur Generation des Testsignals bereitgestellt. Um das Skript für die darauf folgende Simulation vorzubereiten, hat die CAU kleine Anpassungen und Modifikationen vorgenommen. Deshalb wird das Skript und die darin enthaltenen Funktionen an dieser Stelle hinzugefügt.

```
% Main file to demonstate the MASTER 360 data aquisition interface
% 2021-10 Andre.Burkard@funkeavionics.de
% Testes with octave 5.2.
%
% see help requestRxData for the interface description
%

% clc
% clear all
% close all

try
    % octave compatibility, try to load signal processing packet
    pkg load signal
catch
end

% configure for data simulation
config.src      = 'SIM';
config.sim_w    = complex(randn(1,6), randn(1,6));
%config.sim_w   = ones(1,6);
config.sim_dist = 10e3;
config.sim_df   = 0;
config.fs       = 20e6;

% requests receiver data
[IQrx, preamble, info] = requestRxData(config);

%% plot receiver data

N_rx = size(IQrx,1);
t = (0:(size(IQrx,2)-1)) /info.fs;

figure
%for each receiver, plot baseband data as phase/magnitude
for k_rx = 1:N_rx

    % Magnitude in left column
    subplot(N_rx,2,2*k_rx-1)
    mag = abs(IQrx(k_rx,:)); % suppress plotting with low magitude
    plot(t*1e6, mag);

    title(sprintf('Rx %d', k_rx))
    xlabel('time [us]')
    ylabel('Magnitude')

    % Phase in right column
    subplot(N_rx,2,2*k_rx)
```

```

phi = angle(IQrx(k_rx,:))/pi*180;
phi(mag<0.1) = NaN;
plot(t*1e6, phi);

title(sprintf('Rx %d', k_rx))
xlabel('time [us]')
ylabel('Angle [deg]')
ylim([-180 180])
end

%for each receiver, plot baseband real/imag
figure
for k_rx = 1:N_rx
    subplot(N_rx,1,k_rx)

    % plot I/Q
    plot(t*1e6, real(IQrx(k_rx,:)));
    hold all
    plot(t*1e6, imag(IQrx(k_rx,:)));

    % label plot
    title(sprintf('Rx %d', k_rx))
    xlabel('time [us]')
    ylabel('real/imag')
end

```

B.3 requestRxData.m

```
function [IQrx, preamble,info] = requestRxData(config);
% REQUESTRXDATA Requests one captured transponder telegram.
% Calling this function will arm the aquisition trigger, this function
% blocks until one telegram is aquired
%
% Interface version: 0.01, 2021-10-12
% Author: Andre.Burkard@funkeavionics.de
%
% [IQ, info] = REQUESTRXDATA(config);
%
% *Inputs*
% CONFIG is a struct to configure the data aquisition
%
% config.src 'SIM' to get simulation data or URL
% (or IP) of the MASTER360 receiver
% config.trigger_ext Enable external trigger input: 0 | 1
% config.trigger_MODES Enable ModeAC reception trigger: 0 | 1
% config.trigger_ICAO ICAO 24-bit address trigger filter
% as integer, 0 = All
% config.trigger_DF MODE-S Message type filter as
% integer, 0 = ALL
%
% config.sim_w Complex antenna weight vector for
% simulation
% config.sim_df Delta frequency between Tx and Rx [Hz]
% config.sim_dist Simulated radio! distance [m]
%
% *OUTPUTS*
% IQ is N,M matrix, with complex receiver samples
% N is the number of receiver channels, default is 6
% M number of samples
%
% INFO is a struct with informations about the aquired data telegram
% info.fs Sampling rate in Hz
% info.triggerSrc Trigger: 'SIM' | 'EXT' | 'MODES'

if config.src == 'SIM'
% simulate data
% addpath('sim')
[IQrx, preamble, info] = simRxData(config);
else
error('Not implemented')
% request data from actual device, implemented as MEX-C-Code
end
```

B.4 simRxData.m

```
function [IQrx, preamble, info] = simRxData(config)

% config
fs = 20e6;
t_end = 350e-6;
payload1 = 'a000149aa0fa092d7ff8003d258e';
N0 = 0; % TBD

c0 = 299792458; % [m/s]

% fill not specified configuration fields
if ~isfield(config, 'sim_df')
    config.sim_df = 500e3;
end
if ~isfield(config, 'sim_dist')
    config.sim_dist = 0;
end

% simulation timebase
t_sim = 0:(1/fs):t_end;

% build mode-S telegramm
[t_sym symbols] = ModeS_Modulate(payload1);

% randomize puls phases - Transponders are not required to have coherent
% pulse phases
pulse_phases = ones(size(symbols));
curr_phase = rand(1,1);
curr_sym = 0;
for k = 1:length(symbols)
    if curr_sym ~=symbols(k)
        curr_sym =symbols(k);
        curr_phase = rand(1,1);
    end
    pulse_phases(k) = curr_phase;
end
symbols = symbols .* exp(2i*pi*pulse_phases);

% upsample data, shift by radio distance
t_dist1 = config.sim_dist/c0;

IQtx = interp1(t_sym, symbols, t_sim, 'nearest', 'extrap');
IQrx = interp1(t_sym+t_dist1, symbols, t_sim, 'nearest', 'extrap');

preamble = IQtx(1:(config.fs*8e-6)+1);

% Receiver noise
IQrx = IQrx + N0*randn(size(IQrx));

% Baseband filter
N=4;
w_cutoff = 2*pi*8e6; % not 3dB BW!
```

```

[b,a] = besself(N,w_cutoff) ;
I = lsim( tf(b,a), real(IQrx), t_sim)';
Q = lsim( tf(b,a), imag(IQrx), t_sim)';
IQrx = I + 1i*Q;

% TX-RX frequency derivation
IQrx = IQrx .* exp(2i*pi*config.sim_df*t_sim);

% simulate antennas
%   IQ = config.sim_w' * IQ;

% format return values
info.fs = fs;
info.triggerSrc = 'SIM';

```

B.5 ModeS_Modulate.m

```

function [t_sim symbols] = ModeS_Modulate(payload)
% [https://www.radartutorial.eu/13.ssr/sr24.de.html]

% preamble 0 . 1 . 2 . 3 . 4 . 5 . 6 . 7 . 8
symbols = [1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0];

% Convert Payload
for i = 1:2:length(payload)
    byte = hex2dec(payload(i:(i+1)));
    % for each byte in payload:
    for nBit = 7:-1:0 % Bitorder MSB first?
        if bitand(byte, int16(2)^nBit);
            sym = [1 0]; % 1
        else
            sym = [0 1]; % 0
        end
        symbols = [symbols, sym];
    end
end

end

% append 0s at both ends
symbols = [0,symbols, 0];

% time vector for symbols, each symbol is 0.5us long
t_sim = (0:(length(symbols)-1))*0.5e-6;

```

B.6 getPattern3D.m

```
function [E,EPhi,ETheta,A,phi,phiDeg,theta,thetaDeg,EPhase] = ...
getPattern3D(gndPlane,pol)
% getPattern3D obtains the radiation pattern samples of the upper
% hemisphere
%
% Author: saaa@tf.uni-kiel.de
%
% [E,EPhi,ETheta,A,phi,phiDeg,theta,thetaDeg,EPhase] =
% getPattern3D(gndPlane,pol)
%
% Inputs:
%     gndPlane    Ground plane of the MMA antenna: 'xy' or 'xz'
%     pol         Ploarization of the radiation patterns:
%                 'theta': Vertical polarization
%                 'phi': Horizontal polarization
%                 'RHCP': Right-hand circular polarization
%                 'LHCP': Left-hand circular polarization
%                 'mm': Sum of the magnitude square of the field
%                 components
%
% Outputs:
%     E           Total field depending on the polarization
%     EPhi        Azimuth field component
%     ETheta      Elevation field component
%     A           Array response matrix
%     phi         Azimuth angle grid in radian
%     phiDeg      Azimuth angle grid in degree
%     theta       Elevation angle grid in radian
%     thetaDeg    Elevation angle grid in degree
%     EPhase      Phase of the total Field

% parameters
angRes = 5; % Step size of the simulated MMA patterns, valid for phi and
% theta
nModes = 8; % No. of MMA modes

% 360 azim * 180 elev - 180 azim * "poles" + "poles"
ERaw=complex(zeros(360/angRes * 180/angRes - 180/angRes * 2 + 2,nModes*2));
E = zeros(size(ERaw,1),nModes);
A = zeros(size(ERaw,1),nModes);
EPhi = zeros(size(ERaw,1),nModes);
ETheta = zeros(size(ERaw,1),nModes);
EPhase = zeros(size(ERaw,1),nModes);
phi = zeros(size(ERaw,1),nModes);
phiDeg = zeros(size(ERaw,1),nModes);
theta = zeros(size(ERaw,1),nModes);
thetaDeg = zeros(size(ERaw,1),nModes);

antennaData = coder.load('ModalFarfieldsCuboid.mat');
if isequal(gndPlane, 'xy') % Choose the ground plane
    fields_GNDPlane = antennaData.fields_GNDxy;
elseif isequal(gndPlane, 'xz')
    fields_GNDPlane = antennaData.fields_GNDxz;
else
    disp('Enter either xy or xz')
    return
end
```

```

% Extract EPhi, ETheta and the phi and theta grids
for m=1:nModes
    ERaw(:,2*m-1)=fields_GNDPlane.E_phi(:,m);
    ERaw(:,2*m)=fields_GNDPlane.E_theta(:,m);
end
Eangle = fields_GNDPlane.FF_Angle;

% Calculate the field components for different polarizations
EAngleRad = deg2rad(Eangle);
EPhiRaw = complex(zeros(size(Eangle,1),nModes));
EThetaRaw = complex(zeros(size(Eangle,1),nModes));
EThetaCom = complex(zeros(size(Eangle,1),nModes));
EPhiCom = complex(zeros(size(Eangle,1),nModes));
Atheta = complex(zeros(size(Eangle,1),nModes));
Aphi = complex(zeros(size(Eangle,1),nModes));
ERHCP = complex(zeros(size(Eangle,1),nModes));
ELHCP = complex(zeros(size(Eangle,1),nModes));
ARHCP = complex(zeros(size(Eangle,1),nModes));
ALHCP = complex(zeros(size(Eangle,1),nModes));
EMM = complex(zeros(size(Eangle,1),nModes));
AMM = complex(zeros(size(Eangle,1),nModes));
for m=1:nModes
    EPhiRaw(:,m) = ERaw(:,2*m-1);
    EThetaRaw(:,m) = ERaw(:,2*m);

    % combine E_phi and E_theta according to the polarization and the array
    % response matrix:
    % According to Balanis and Ioannides the array response matrix is
    % defined:
    % a(phi,theta) = sqrt(D(phi,theta)) .* exp(j*angle(E(phi,theta)))
    % D:directivity, E: E-field

    % Phase rotation compensation of vertical and horizontal polarizations
    EThetaCom(:,m) = ERaw(:,2*m) .* exp(-1j*EAngleRad(:,2));
    EPhiCom(:,m) = ERaw(:,2*m-1) .* exp(1j * EAngleRad(:,2));
    % calculate the directivity in theta and phi directions
    [Dtheta,Dphi] = EtoD(EThetaCom(:,m), EPhiCom(:,m), Eangle(:,1), angRes);
    % array response matrix
    Atheta(:,m) = sqrt(Dtheta) .* exp(1j * angle(EThetaCom(:,m)));
    Aphi(:,m) = sqrt(Dphi) .* exp(1j * angle(EPhiCom(:,m)));

    % Right handed circular polarization
    ERHCP(:,m) = (ERaw(:,2*m) - 1j*ERaw(:,2*m-1)) / sqrt(2);
    ERHCP(:,m) = ERHCP(:,m) .* exp(-1j * EAngleRad(:,2));

    % Left handed circular polarization
    ELHCP(:,m) = (ERaw(:,2*m) + 1j*ERaw(:,2*m-1)) / sqrt(2);
    ELHCP(:,m) = ELHCP(:,m) .* exp(1j * EAngleRad(:,2));

    % calculate the directivity for RHCP and LHCP
    [DRHCP,DLHCP] = EtoD(ERHCP(:,m), ELHCP(:,m), Eangle(:,1), angRes);
    % array response matrix
    ARHCP(:,m) = sqrt(DRHCP) .* exp(1j * angle(ERHCP(:,m)));
    ALHCP(:,m) = sqrt(DLHCP) .* exp(1j * angle(ELHCP(:,m)));

    % Magnitude of the field E = e_rhoi*EPh + e_theta * ETh taking the raw
    % phi and theta compopnents i.e. the original polarization of MM antenna
    EMM(:,m) = sqrt(abs(ERaw(:,2*m)).^2 + abs(ERaw(:,2*m-1)).^2);

```

```

    phaseArg = ERaw(:,2*m)./ERaw(:,2*m-1);
    % calculate the directivity in theta and phi
    [DTh,DPH] = EtoD(ERaw(:,2*m), ERaw(:,2*m-1), Eangle(:,1), angRes);
    D0 = DTh + DPH; % According to balanis (17-21)
    AMM(:,m) = sqrt(D0) .* exp(1j * atan2(imag(phaseArg), real(phaseArg)));
    % Array response vector
end

if isequal(pol,'RHCP')
    ETemp = ERHCP;
    ATemp = ARHCP;
elseif isequal(pol,'LHCP')
    ETemp = ELHCP;
    ATemp = ALHCP;
elseif isequal(pol,'phi')
    ETemp = EPhiCom;
    ATemp = APhi;
elseif isequal(pol,'theta')
    ETemp = EThetaCom;
    ATemp = ATheta;
elseif isequal(pol,'mm')
    ETemp = EMM;
    ATemp = AMM;
else
    disp('Polarization should be: "Phi", "theta", "RHCP", "LHCP", "mm"')
    return
end

% Take the upper Hemisphere patterns
az = Eangle(:,2);
ev = Eangle(:,1);
uHemiSphere = ev <= 90;
thetaDeg = ev(uHemiSphere);
phiDeg = az(uHemiSphere);
theta = deg2rad(thetaDeg);
phi = deg2rad(phiDeg);

E = complex(zeros(length(theta),nModes));
A = complex(zeros(length(theta),nModes));
EPhi = complex(zeros(length(theta),nModes));
ETheta = complex(zeros(length(theta),nModes));
EPhase = zeros(length(theta),nModes);

for m = 1:nModes
    E(:,m) = ETemp(uHemiSphere,m);
    A(:,m) = ATemp(uHemiSphere,m);
    EPhi(:,m) = EPhiRaw(uHemiSphere,m);
    ETheta(:,m) = EThetaRaw(uHemiSphere,m);
    EPhase(:,m) = angle(E(:,m));
end
end

```

B.7 EtoD.m

```
function [Dtheta, Dphi] = EtoD(Etheta, Ephi, thetaDeg, angRes)
% EtoD calculates the partial directivities in a given direction (thetaDeg,phiDeg)
% from the azimuth and elevation electric field components Ephi and Etheta
%
% [Dtheta, Dphi] = EtoD(Etheta, Ephi, thetaDeg, angRes)
%
% Inputs:
%   Etheta:    complex E-field in theta direction (far field)
%   Ephi:      complex E-field in phi direction (far field)
%   thetaDeg:  sampling points in theta domain [deg]
%   angRes:    angular resolution [deg]
%
% Outputs:
%   Dtheta:    partial directivity in theta direction
%   Dphi:      partial directivity in phi direction
%
% Calculation is according to:
% [1] C. A. Balanis, Antenna Theory: Analysis and Design. John Wiley &
% Sons, 2015, pp. 37-42.

% intrinsic impedance in free space
% eta = 120*pi;
eta = 376.730313;

% total and partial radiation intensities = power radiated from an antenna
% per unit solid angle
U = (1/(2*eta))*(abs(Etheta).^2 + abs(Ephi).^2);
Utheta = (1/(2*eta))*(abs(Etheta).^2);
Uphi = (1/(2*eta))*(abs(Ephi).^2);

% total radiated power = surface integral over radiation intensity
Prad = sum(U.*sin(repmat(thetaDeg,1,size(U,2))*pi/180))*(angRes*pi/180)* ...
(angRes*pi/180);

% partial directivities
Dtheta = 4*pi*Utheta./repmat(Prad,size(Utheta,1),1);
Dphi = 4*pi*Uphi./repmat(Prad,size(Uphi,1),1);

end
```

B.8 wm3D.m

```
function [AInterp,A,G,phiFineGrid,phiFineGridDeg,thetaFineGrid, ...
    thetaFineGridDeg]=wm3D(angResFine,pol,NPhi,NTheta) %#codegen
% wm3D interpolates the field (or the array response matrix) samples based
% on a wave modelling approach to obtain the radiation patterns in an
% angular grid with any desired resolution
%
% Author: saaa@tf.uni-kiel.de
%
% [AInterp,A,G,phiFineGrid,phiFineGridDeg,thetaFineGrid,thetaFineGridDeg]= ...
% wm3D(angResFine,pol,NPhi,NTheta)
%
% Inputs:
%   angResFine   Desired fine angular resolution
%   pol          Ploarization of the radiation patterns:
%               'theta': Vertical polarization
%               'phi': Horizontal polarization
%               'RHCP': Right-hand circular polarization
%               'LHCP': Left-hand circular polarization
%               'mm': Sum of the magnitude square of the field components
%   NPhi        Number of coefficients in azimuth direction
%   NTheta      Number of coefficients in elevation direction
%               The value of 5 for the two previous coefficients
%               delivers precise results
%
% Outputs:
%   AInterp     Interpolated array response matrix
%   A          Original array response matrix
%   G          Interpolation coefficient matrix
%   phiFineGrid   Fine azimuth angle grid in radian
%   phiFineGridDeg   Fine azimuth angle grid in degree
%   thetaFineGrid   Fine elevation angle grid in radian
%   thetaFineGridDeg   Fine elevation angle grid in degree

gndPlane = 'xy'; % Ground plane of the MMA antenna: 'xy' or 'xz'
Ncoeff = (2*NPhi+1)*(2*NTheta+1); % Fourier series coefficients
[~,~,~,A,phi,~,theta,~,~] = getPattern3D(gndPlane,pol); % obtain the array
% response matrix
nModes = size(A,2);
G = complex(zeros(Ncoeff,nModes));
for mode = 1:nModes
[G(:,mode),~] = coeffMatrix3D(phi, theta, A(:,mode), NPhi, NTheta); % cal-
% culate the interpolation coefficient matrix
end

% Define grids of phi and theta in fine resolution
phiFine = deg2rad(0:angResFine:360-angResFine);
thetaFine = deg2rad(angResFine:angResFine:90);
phiFineGrid = repmat(phiFine,length(thetaFine),1);
phiFineGrid = phiFineGrid(:);
thetaFineGrid = repmat(thetaFine,1,length(phiFine))';
phiFineGrid = [phiFineGrid;0];
thetaFineGrid = [thetaFineGrid;0];
phiFineGridDeg = round(rad2deg(phiFineGrid));
thetaFineGridDeg = round(rad2deg(thetaFineGrid));

% Calculate the interpolated array response matrix
Mx = repmat(-NPhi:NPhi,1,2*NTheta+1);
```

```
My = repmat(-NTheta:NTheta,2*NPhi+1,1);  
My = My(:).';  
B = exp(1j * (phiFineGrid*Mx) + (thetaFineGrid*My)); % Fourier series  
AInterp = B * G;  
end
```

B.9 coeffMatrix3D.m

```
function [G,B] = coeffMatrix3D(phi, theta, E, NPhi, NTheta)
% coeffMatrix3D calculates the interpolation coefficient matrix by taking
% the least squares solution
%
% Author: saaa@tf.uni-kiel.de
%
% Inputs:
%   phi      Azimuth angle of the angular samples
%   theta    Elevation angle of the angular samples
%   E        Radiation pattern samples at the given angles
%   NPhi     Number of coefficients in azimuth direction
%   NTheta   Number of coefficients in elevation direction
%           The value of 5 for the two previous coefficients
%           delivers precise results
%
% Outputs:
%   G        Interpolation coefficient matrix
%   B        Fourier series

Mx = repmat(-NPhi:NPhi,1,2*NTheta+1);
My = repmat(-NTheta:NTheta,2*NPhi+1,1);
Mz = My(:).';

% Fourier series
B = exp(1j * (phi*Mx) + (theta*My));
% Least squares solution
G = B\E;
end
```

B.10 simMusicCost3D.m

```
function [P_music] = simMusicCost3D(phiTilde,thetaTilde, NN, G, NPhi, NTheta)
% simMusicCost3D Calculates the MUSIC spectrum of the given noise space NN
% over the hypothesis grid of angles
% phiTilde and thetaTilde
%
% Author: saaa@tf.uni-kiel.de
%
% Inputs:
%   phiTilde      Azimuth angle of the hypothesis grid of angles
%   thetaTilde    Elevation angle of the hypothesis grid of angles
%   NN            Noise space of the received signal
%   G             Interpolation coefficient matrix produced by the w
%                avefield modelling method
%   NPhi          Number of coefficients in azimuth direction
%   NTheta        Number of coefficients in elevation direction
%                The value of 5 for the two previous coefficients
%                delivers precise results
%
% Outputs:
%   P_music       MUSIC spectrum of the hypothesis grid of angles

P_music = zeros(length(phiTilde),1);
PP = complex(zeros(length(phiTilde),1));

    for ind = 1:length(phiTilde)
        [A,~] = WMInterpPattern3D(phiTilde(ind),thetaTilde(ind),G, ...
            NPhi, NTheta);
        A = A.';
        PP(ind)=A'*(NN*NN')*A;
        P_music(ind) = abs(PP(ind));
    end

end
```

B.11 WMInterpPattern3D.m

```
function [A,B] = WMInterpPattern3D(phiDeg,thetaDeg,G,NPhi,NTheta)
% WMInterpPattern3D calculates the radiation pattern at any arbitrary
% direction in space determined by phiDeg and thetaDeg
%
% Author: saaa@tf.uni-kiel.de
%
% Inputs:
%   phiDeg    azimuth of the aimed direction
%   thetaDeg  elevation of the aimed direction
%   G         Interpolation coefficient matrix
%   NPhi      Number of coefficients in azimuth direction
%   NTheta    Number of coefficients in elevation direction
%             The value of 5 for the two previous coefficients
%             delivers precise results
%
% Outputs:
%   A         Radiation pattern at the aimed direction
%   B         Fourier series

phi = deg2rad(phiDeg);
theta = deg2rad(thetaDeg);

Mx = repmat(-NPhi:NPhi,1,2*NTheta+1);
My = repmat(-NTheta:NTheta,2*NPhi+1,1);
My = My(:).';
B = exp(1j * (phi*Mx) + (theta*My)); % Fourier series

A = B * G; % radiation pattern

end
```

B.12 CRLB3D.m

```
function [CRLB_theta, CRLB_phi] = CRLB3D(phiDegVec, thetaDegVec, snrVec, K, ...
    G, NPhi, NTheta)
% CRLB3D calculates the Cramer-Rao lower bound (CRLB) of variance for joint
% elevation and azimuth estimation
%
% Author: saaa@tf.uni-kiel.de
%
% Inputs:
%     phiDegVec    azimuth angle
%     thetaDegVec  elevation angle
%     snrVec       SNR values in dB
%     K           Number of snapshots
%     G           Interpolation coefficient matrix
%     NPhi        Number of coefficients in azimuth direction
%     NTheta      Number of coefficients in elevation direction
%               The value of 5 for the two previous coefficients
%               delivers precise results
%
% Outputs:
%     CRLB_theta   CRLB of theta
%     CRLB_phi    CRLB of phi

CRLB_theta = zeros(length(thetaDegVec), length(snrVec));
CRLB_phi = zeros(length(thetaDegVec), length(snrVec));

thetaVec = thetaDegVec*pi/180;
phiVec = phiDegVec*pi/180;

% maximum order N
Nx = NPhi;
Ny = NTheta;

% dadTheta
n = repmat(-Nx:Nx,1,2*Ny+1);
m = repmat(-Ny:Ny,2*Nx+1,1);
m = m(:)';
dFcp = ones(length(thetaVec),1) * li*m .* exp(li*(thetaVec*m+phiVec*n));
dadThetaDeg = (dFcp*G)*pi/180; % in deg^-1

% dadPhi
dFcp = ones(length(thetaVec),1) * li*n .* exp(li*(thetaVec*m+phiVec*n));
dadPhiDeg = (dFcp*G)*pi/180; % in deg^-1

%     a = idft2(thetaVec, phiVec, a_N, Nphi, Ntheta);
[a,~] = WMInterpPattern3D(phiDegVec,thetaDegVec,G,NPhi, NTheta);

% calc CRLB
for idx = 1:length(thetaDegVec)

    for snrIdx = 1:length(snrVec)
        snrLin = 10^(snrVec(snrIdx)/10);

        % friedlander
        A = (a(idx,:)).';
        D = [(dadThetaDeg(idx,:)).', (dadPhiDeg(idx,:)).'];
        Pi_A = A * pinv(A);
        Pi_A_o = eye(size(G,2)) - Pi_A;
    end
end
```

```
CRBM = (1/(2*K*snrLin)) * inv(real(D'*Pi_A_o*D));  
CRBM(isinf(CRBM)) = NaN;  
CRLB_theta(idx,snrIdx) = CRBM(1,1);  
CRLB_phi(idx,snrIdx) = CRBM(2,2);  
end  
end
```

C MATLAB-Software für praktische Implementierungen

Ähnlich der für Simulationen verwendeten Software wurden kleine Anpassungen beim Skript Demo_main vorgenommen. Weiterhin wurde das Skript simDoaDistEstim für die praktische Implementierung umgeschrieben und im Skript Demo_main eingebaut. Die darunterliegenden Funktionen mit denselben Namen wie im Anhang B können unverändert genutzt werden.

C.1 Demo_main.m

```
% Main file to demonstate the MASTER 360 data aquisition interface
% 2021-10 Andre.Burkard@funkeavionics.de
% Testes with octave 5.2.
%
% see help requestRxData for the interface description
%

% clc
% clear all
% close all

try
    % octave compatibility, try to load signal processing packet
    pkg load signal
catch
end

% configure for data simulation
config.src      = 'SIM';
config.sim_w    = complex(randn(1,6), randn(1,6));
%config.sim_w   = ones(1,6);
config.sim_bearing = 0;
config.sim_elevation = 25;
config.sim_dist = 10e3;
config.count = 1;
config.sim_SNRdB = 0;
config.sim_df = 0;
config.fs = 20e6;

% requests receiver data
[IQrx, preamble, info] = requestRxData(config);

% Doa and distance
[r, ac, count] = DoaDistEstim(IQrx,preamble,config);

% Output the results
[AC,Count] = pushRxData(ac,count);

%% plot receiver data
N_rx = size(r,1);
t = (0:(size(r,2)-1)) /info.fs;
```

```

figure
%for each receiver, plot baseband data as phase/magnitude
for k_rx = 1:N_rx

    % Magnitude in left column
    subplot(N_rx,2,2*k_rx-1)
    mag = abs(r(k_rx,:)); % suppress plotting with low magnitude
    plot(t*1e6, mag);

    title(sprintf('Rx %d', k_rx))
    xlabel('time [us]')
    ylabel('Magnitude')

    % Phase in right column
    subplot(N_rx,2,2*k_rx)
    phi = angle(r(k_rx,:))/pi*180;
    phi(mag<0.1) = NaN;
    plot(t*1e6, phi);

    title(sprintf('Rx %d', k_rx))
    xlabel('time [us]')
    ylabel('Angle [deg]')
    ylim([-180 180])
end

%for each receiver, plot baseband real/imag
figure
for k_rx = 1:N_rx
    subplot(N_rx,1,k_rx)

    % plot I/Q
    plot(t*1e6, real(r(k_rx,:)));
    hold all
    plot(t*1e6, imag(r(k_rx,:)));

    % label plot
    title(sprintf('Rx %d', k_rx))
    xlabel('time [us]')
    ylabel('real/imag')
end

```

C.2 DoaDistEstim.m

```
function [r, ac, count] = DoaDistEstim(IQrx,preamble,config) %#codegen
% simDoaDistEstim calculates the direction of arrival and the distance of
% the received signal IQrx arriving from the azimuth angle config.sim_bearing
% and the elevation angle config.sim_elevation at the distance config.sim_dist
% given a signal to noise ratio config.sim_SNRdB at the input of the signal
% processing unit.
% The signal IQrx is assumed to be received by the simulated multi-mode
% antenna (MMA) pattern provided by LUH. These patterns are simulated in a
% grid of points in the upper hemisphere. The grid has a step size of 5
% degrees in both azimuth and elevation
% directions.
%
% Author: saaa@tf.uni-kiel.de
%
% [r, ac, count] = simDoaDistEstim(IQrx,preamble,config)
%
% Inputs:
%     IQtx:                Transmitted signal
%     preamble:            Upsampled preamble of a typical Mode S signal
%     config.sim_bearing   True azimuth angle of arrival in degrees,
%                           i.e. it belongs to [0:5:355]
%     config.sim_elevation True elevation angle of arrival in degrees
%                           , i.e. it belongs to [0:5:90]
%     config.sim_dist      True radial distance to the targeted
%                           object in meters
%     config.sim_SNRdB     Signal to noise ratio at the input of the
%                           signal processing unit in dB
%     config.fs            Sampling frequency
%
% Outputs:
%     r:                   Received signal at the output of the MM receiver
%     ac.Bearing:          Estimated bearing of the target [0 to 360] deg
%     ac.Elevation:        Estimated elevation of the target [0:90] deg
%     ac.Distance:         Estimated distance of the target in meters
%     count:               Number of targets

% Assure that the given true angles have valid values
if (mod(config.sim_bearing,5) ~= 0) && (mod(config.sim_elevation,5) ~= 0)
    disp(['Phi and theta should be multiples of 5 within [0:5:355] and ' ...
        '[0:5:90] respectively'])
    ac.Bearing = 0;
    ac.Elevation = 0;
    ac.Distance = 0;
    count = 0;
    return
else
    % get MMA Patterns and array response matrix
    gndPlane = 'xy'; % ground plane
    % Two possible mode combinations (agreed after discussion with project
    % partners to be a combination of ports 1,2 and 3
    modeComb1 = [1 2 3];
    % modeComb2 = [1 4 5];
    chosenComb = modeComb1; % Chosen combination
    nModes = length(chosenComb);
    pol = 'theta'; % Vertical Polarization
    % Obtain the array response matrix AMM
```

```

[~,~,~,AMM,~,phiMMDeg,~,thetaMMDeg,~] = getPattern3D(gndPlane,pol);

% Interpolate MMA patterns using Wavefield modelling:
% The objective of wavefield modelling is to obtain the radiation
% pattern at any
% arbitrary direction, not only at the grid points of 5 degrees step
% size, at which the MMA was simulated.
angResFine = 1; % step size of the interpolated fine angular grid [deg]
NPhi = 5; % Phi coefficients
NTheta = 5; % theta coefficients, phi and theta coefficients are
% necessary for the wavefiled modelling approach
[AInterp,~,G,~,phiFineGridDeg,~,thetaFineGridDeg] = wm3D(angResFine,pol, ...
    NPhi,NTheta); % Interpolate the array response matrix for the fine grid search

%% Simulation of DoA Estimation
% The signals are received via the array response matrix, that was calculated
% from the simulated radiation patterns provided by LUH.
% Since the radiation patterns are provided in 5 degree step size in both
% elevation and azimuth directions, theta needs to be a multiple of 5
% between 0 and 90 and phi needs to be a multiple of 5 between 0 and 355

% Index of the true DoA angles
% For theta = 0 the signal is arriving at the antenna from the
% postive direction of the z axis, hence there is only one responce of
% the antenna for any value of phi.
if config.sim.elevation == 0 % Extract the values of E_the and E_phi
    % for a signal arriving from theta = 0
    angIdxMM = (thetaMMDeg==0) & (phiMMDeg==0);
else % otherwise
    angIdxMM = (thetaMMDeg==config.sim.elevation) & (phiMMDeg==config.sim.bearing);
end

% Receiver Noise
SNRlin = 10 .^(config.sim.SNRdB./10);
N0 = sqrt(1./(2*SNRlin));

% Received signal
% Receive IQrx with the MMA and add noise
r = AMM(angIdxMM,chosenComb).' * IQrx + N0 * randn(nModes,length(IQrx));

%% DoA estimation
R = (1/size(r,2)) * (r*r'); % Covariance matrix of the received signal
[V,D]=eig(R); % Eigen decomposition of the covariance matrix
 [~,I] = max(D,[],'all'); % Find the largest eigen value
 [~,colMax]=ind2sub(size(D),I); % Determine the signal eigenvector(signal
% subspace)
V(:,colMax) = []; % Extract the noise subspace
NN = V; % Assign the noise subspace to NN

% MUSIC spectrum
PMusic = simMusicCost3D(phiFineGridDeg,thetaFineGridDeg, NN, ...
    G(:,chosenComb), NPhi, NTheta); % Calculate the MUSIC spectrum at
% the points of the interpolated fine grid
[~, minIdx] = min(PMusic); % Find the minimum of the music spectrum
thetaEstimCoarse = thetaFineGridDeg(minIdx); % Coarse estimate of theta
phiEstimCoarse = phiFineGridDeg(minIdx); % Coarse estimate of phi

%% Optimization
% Define a smaller search intrerval around the coarse estimates calculate

```

```

% the MUSIC spectrum over it with finer steps, then look for the minimum
% of the spectrum
% Define the finer ranges:
fineSearchinterval = 1;
lowerBndTheta = thetaEstimCoarse-fineSearchinterval;
lowerBndTheta(lowerBndTheta < 0) = 0;
upperBndTheta = thetaEstimCoarse+fineSearchinterval;
upperBndTheta(upperBndTheta > 90) = 90;
lowerBndPhi = phiEstimCoarse-fineSearchinterval;
lowerBndPhi(lowerBndPhi < 0) = 0;
upperBndPhi = phiEstimCoarse+fineSearchinterval;
upperBndPhi(upperBndPhi > (360-fineSearchinterval)) = 360-fineSearchinterval;

% Optimize the results by searching over the finer grid
costFuncWrapper = @(args) simMusicCost3D(args(1), args(2), NN, ...
    G(:,chosenComb), NPhi, NTheta);
options = optimoptions(@fmincon,'Algorithm','sqp','StepTolerance', ...
    1e-6, 'Display','off');
[argMin,~] = fmincon(costFuncWrapper, [phiEstimCoarse thetaEstimCoarse], ...
    [], [], [], [lowerBndPhi lowerBndTheta], [upperBndPhi upperBndTheta], ...
    [], options);

ac.Bearing = argMin(1); % Fine estimate of theta
ac.Elevation = argMin(2); % Fine estimate of phi

%% Distance estimation
% The following solution compares the received reply with the
% stadarized preamble of Mode S signals, i.e. it calculates the creoss
% correlation between the received signal and the standard preamble.

% Since the received signal from only one port is required for distance
% estimation, the port
% with the highest gain in the direction of arrival is chosen for
% distance estimation
gainDoa = abs(AInterp(minIdx,chosenComb)); % Gain based on the estimated DoA
[~,bestPort] = max(gainDoa); % Find the port with the maximum gain
[c, lags] = xcorr(abs(r(bestPort,:)),abs(preamble)); % Cross correlation
% between the preamble and the received signal over the port with maximum gain
[m,~] = max(abs(c)); % peak point of the cross correlation function
corrNorm = c/m; % Normalize the cross correlation function
threshold = 0.8; % Define a threshold for the correlation
[~,indNorm] = find(abs(corrNorm)>threshold); % Find correlation incidents
% above the threshold
delaySamples = lags(indNorm(1)); % delay in samples of the first incident
delaySeconds = delaySamples/config.fs; % delay in seconds
ac.Distance = delaySeconds * physconst('LightSpeed'); % distance in meters

% Pass the number of the targets to the output
count = config.count;
end
end

```

C.3 pushOutput.m

```
% Main file to demonstate the MASTER 360 data output interface
% 2022-02 wolfgang.schaefer@funkeavionics.de
% Testes with octave 5.2.
%
% see help pushRxData for the interface description
%

clc
clear all
close all

% configure for data simulation
ac.ICAO      = "3BCDEF";
ac.Bearing   = randn(1);
ac.Elevation = randn(1);
ac.Distance  = 50340;
count = 1;

% requests receiver data
pushRxData(ac, count);
```

C.4 pushRxData.m

```
function [AC, Count] = pushRxData(ac, count);
%
%
%
%   Interface version:  0.01, 2022-02-07
%   Author:             wolfgang.schaefer@funkeavionics.de
%
%   [AC, COUNT] = PUSH_RXDATA();
%
% *Inputs*
%   ac.ICAO is the target ID
%   ac.Bearing is the bearing of the target [tbd! deg/rad]
%   [-180 to 180 or 0 to 360 deg] from north
%   ac.Elevation is the elevation of the target [tbd! deg/rad]
%   ac.Distance is the distance of the target [tbd! m/nm]
%
%   count how many targets are possible
%
% *OUTPUTS*
%   TBD! interface via TCP/IP
%
%
%   printf( "ICAO: %s\n\r", ac.ICAO );
%   fprintf( 'Bearing: %f\n\r', ac.Bearing );
%   fprintf( 'Elevation: %f\n\r', ac.Elevation );
%   fprintf( 'Distance: %f\n\r', ac.Distance );
```

```
fprintf( 'Num: %d\n\r', count );  
  
% open TCP/IP socket and push AC  
AC = ac;  
Count = count;  
  
end
```