

Entwurf und Implementierung einer modularen Softwarearchitektur für kognitive verteilte Echtzeit-MIMO-SONAR-Systeme

Frederik Kühne¹, Bastian Kaulen¹, Tim Owe Wisch¹, Gerhard Schmidt¹

¹Christian-Albrechts-Universität zu Kiel, 24118 Kiel, Deutschland, Email: {frk, bk, timw, gus}@tf.uni-kiel.de

Einleitung

Durch Fortschritte in der künstlichen Intelligenz gibt es immer bessere Ansätze von Kognition in komplexen Systemen. Auf ein SONAR-System angewandt bedeutet dies, dass eventuell eine kognitive Steuereinheit die komplette und auch sehr umfangreiche Steuerung übernimmt und damit die Ergebnisse deutlich verbessern kann. Die deutlich schnellere Reaktionszeit erlaubt so eine komplette Rekonfiguration eines verteilten MIMO-SONAR-Systems von Ping zu Ping. Ein Hafenbecken kann so in merklich verkürzter Zeit vollständig untersucht und mögliche Ziele schneller entdeckt werden. Dies erfordert allerdings ein extrem flexibles Rahmenwerk, das solche grundlegenden Code- und Speicheranpassungen ohne einen Systemneustart und ohne das „Vergessen“ der bisherigen Detektionshistorie auch zulässt. Kognitive MIMO-SONAR-Systeme bieten eine Vielzahl an Modi und Einstellungsmöglichkeiten, die teilweise zur Laufzeit verändert werden müssen. Dadurch entstehen Verarbeitungsketten, die nicht in einer einzigen, fixen Abarbeitung von Teilfunktionen resultieren. Um trotzdem optimale Effizienz zu erreichen, erfordert es eine neue Softwarearchitektur für ein SONAR-System, das flexibel ihre Struktur an die gewählten Modi anpasst. Dazu werden in diesem Beitrag variable Grundbausteine entworfen, die je nach SONAR-System optimal zusammengesetzt und parametrisiert werden können. Eine Umsetzung der Algorithmen wird in ein Echtzeitrasterwerk implementiert und die Echtzeitfähigkeit des Gesamtsystems wird gezeigt.

Grundlagen

Ein typisches aktives SONAR-System besteht im einfachsten Fall aus N_{TX} Projektoren und N_{RX} Hydrophonen. Grundlegend wird nach Reflexionen der ausgesendeten Sequenz im aufgenommenen Signal gesucht, um Rückschlüsse auf die Position und Eigenschaften eventuell vorhandener Unterwasserziele zu ziehen. Erweiterungen wie das signalangepasste Filter, das durch seine Pulskompressionseigenschaften und seinen SNR-Gewinn die zeitliche Auflösung und die Schwelle des Erkennens von Objekten signifikant erhöht, oder der Beamformer, der durch eine Phasenverschiebung der $N_{RX} > 1$ Empfangssignale oder $N_{TX} > 1$ Sendesignale dem SONAR neben der zeitlichen Dimension mit dem Winkel einen weiteren Freiheitsgrad liefert, bilden ein typisches SONAR-System. Die Entwicklung von kognitiven MIMO-SONAR-Systemen, wie sie beispielsweise in [1] und [2] beschrieben werden, treibt die Anzahl der während der Laufzeit einstellbaren Parameter durch die Zunahme von verschiedenen nötigen weiteren Berech-

nungsschritten in die Höhe. Die gleichzeitig immer weiter ansteigende Anzahl an Projektor- und Hydrophonelementen führt zu einer außergewöhnlichen Belastung der ausführenden Hardware und erschwert zunehmend eine Echtzeitdetektion der Unterwasserziele.

Verteilte Systeme

Mit der Nutzung von orthogonalen Sendesequenzen sinkt potentiell die Komplexität der Empfangshardware, da mit nur einem einzigen Empfangshydrophon nicht nur eine Entfernung, sondern mit Vorwissen über die Sendesignale auch eine Richtung aus den Empfangssignalen extrahiert werden kann. Das ermöglicht Szenarien, in denen es beispielsweise einen großen MIMO-Sender gibt und viele kleinere, räumlich verteilte Empfänger, um einen relativ verwinkelten Bereich wie beispielsweise ein Hafenbecken optimal auszuleuchten. Auch mehrere verteilte Sendeknoten sind denkbar, sowie das klassische monostatische SONAR-System. Ein rein passives Setup dagegen benötigt nicht die Sendesignalverarbeitungskette. Eine Software, die alle Fälle abdecken kann, muss also modular entworfen werden und sich flexibel an alle verschiedenen Szenarien anpassen können.

SONAR-Modi

Eine weitere Anforderung an ein modulares SONAR-System ist die Möglichkeit, verschiedene Betriebsmodi auszuwählen und einzusetzen. Die Nutzung von sende- und empfangsseitigen Schallwandlerarrays sowie Algorithmen, wie die MIMO-Signalverarbeitung, ermöglichen zahlreiche Szenarien, die unterstützt werden müssen. Auf der Sendeseite gibt es zunächst die Möglichkeit, kohärente Signale omnidirektional oder mittels Sendebeamforming gerichtet auszusenden (*single input, SI*). Dazu kommt die Möglichkeit, orthogonale Sequenzen, die entweder je Projektor omnidirektional oder auch mittels Sendebeamforming in mehrere Richtungen überlagert werden, auszusenden (*multiple input, MI*). Je nach Sendemodus besteht Empfangsseitig zunächst einmal die Möglichkeit, mit einem einzigen Hydrophon zu arbeiten (*single output, SO*) und entweder nur die Entfernung zu bestimmen (*SISO*), oder mithilfe von *MISO* auch Richtungsinformationen zu extrahieren. Dazu kann die Anzahl der Hydrophone erhöht werden, um mehrere Empfangssignale aufzunehmen (*multiple output, MO*) und einen Empfangsbeamformer zu rechnen. Zusammen mit passivem SONAR ergeben sich daraus insgesamt zehn verschiedene Modi, die unterstützt werden müssen.

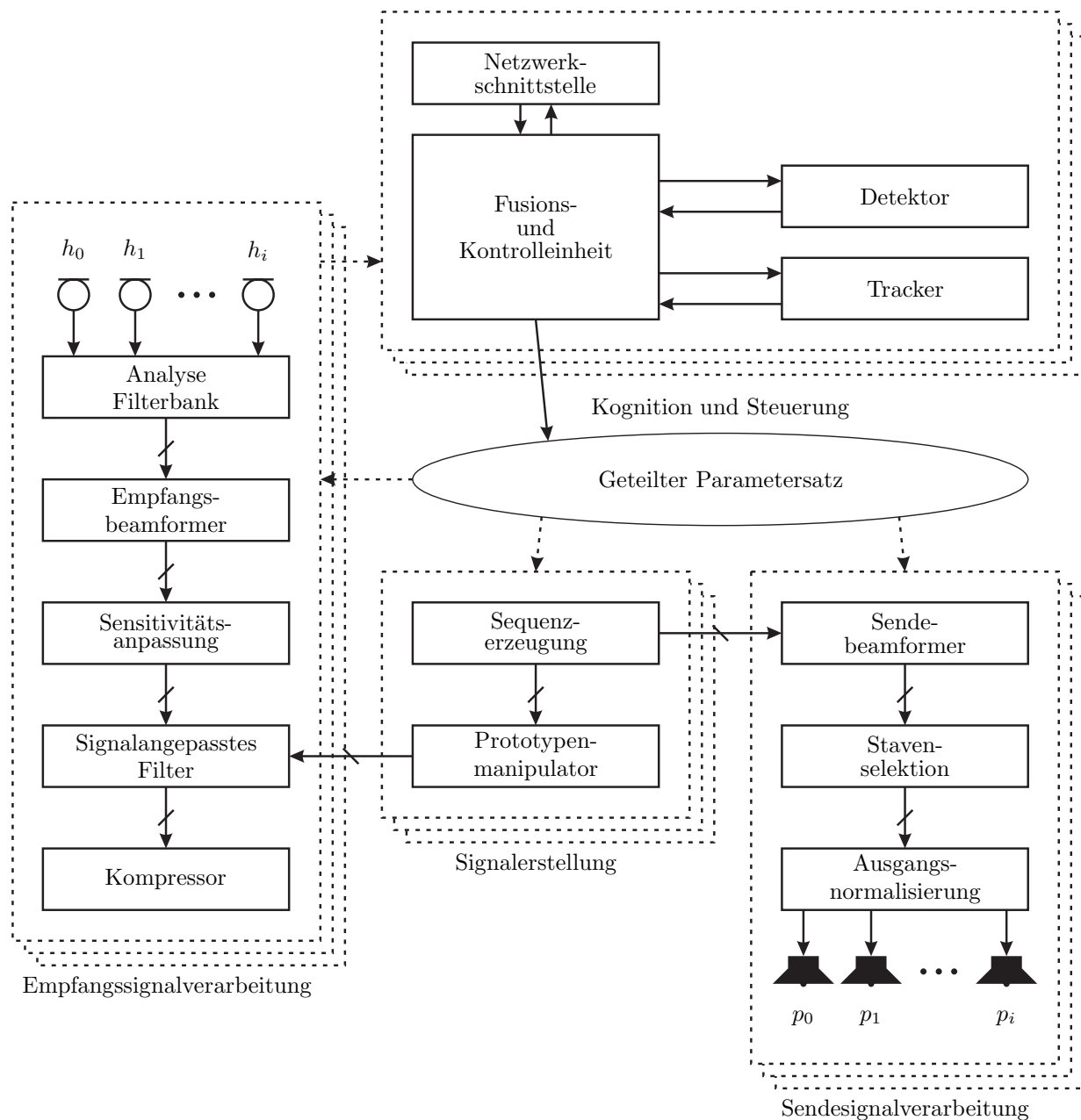


Abbildung 1: Übersicht über die Softwarearchitektur des umgesetzten modularen SONAR-Systems mit der Einteilung in vier eigenständige, auf getrennter Hardware autark einsetzbaren Modulen.

Softwarestruktur des umgesetzten SONAR-Systems

Die Softwarearchitektur des umgesetzten SONAR-Systems ist in Abbildung 1 dargestellt. Um Modularität zu gewährleisten, wurde die gesamte Struktur in vier Teile eingeteilt. Die empfangsseitige Verarbeitung bildet die grundlegenden Funktionen eines passiven SONAR-Systems ab. Für das aktive SONAR wird es um die Sendesignalverarbeitung erweitert, die alle Projektoren ansteuert. Ein eigenes Modul bildet die Signalerstellung, welche das System mit vorgegebenen Sequenzen versorgt. Ein weiteres, übergeordnetes Modul kümmert sich um die Kognition, die Vernetzung und die Steuerung verschiedener Module. Jedes Modul ist so aufgebaut, dass es sowohl einfach, mehrfach oder gar nicht initialisiert werden kann, um die gesamte Speicher- und Rechenlast so gering

wie möglich zu halten, aber so viele Modi und Systeme abbilden kann, wie nötig. Gibt es beispielsweise nur einen Computer mit einem angeschlossenen Hydrophonnarray und einem Projektorarray, so werden alle Module nur einmal auf diesem Computer initialisiert. Bei passivem SONAR kann die Signalerstellung und die Sendesignalverarbeitung eingespart werden. Verteilte Systeme können abgebildet werden, indem auf den verschiedenen Knotenpunkten nur die je nach Aufgabe und angeschlossener Hardware nötigen Module aktiviert werden. Module, die kontinuierliche Verarbeitung benötigen, also die Sende- und Empfangssignalverarbeitung, werden blockbasiert in Echtzeit ausgeführt. Im Gegensatz zur pingbasierten Verarbeitung, die eine deutlich längere Periode bedingt, kann das SONAR-System ankommende Datenpakete kleinschrittiger verarbeiten und dadurch früher

Ergebnisse liefern.

Empfangssignalverarbeitung

In dem Modul für die Empfangssignalverarbeitung werden zunächst die Signale der N_{RX} Hydrophonelemente h_i mithilfe einer Filterbank in den Frequenzbereich transformiert, da die folgenden Berechnungen dadurch effizienter durchgeführt werden können. Durch die Bandbreitenbegrenzung, die durch die verwendete Hardware, die Kanaleigenschaften und dem Anwendungsanforderungen gegeben sind, kann zusätzlich durch frequenzselektive Berechnungen die Prozessorauslastung gesenkt werden. Basierend auf der Anordnung der Hydrophonelemente im Array wird anschließend ein entsprechender Empfangs-beamformer gerechnet. Da die empfangenen Daten a priori nicht oder nur sehr ungenau bekannt sind, muss dieser Schritt in jedem Block durchgeführt werden. Danach folgt die Umrechnung der Eingangssignalamplituden entsprechend der Hydrophonsensitivitäten in Drücke mit der Einheit Pascal, um eine korrekte Skalierung der Ergebnisse zu gewährleisten. Zur qualitativen Auswertung kann dieser Block deaktiviert und somit Rechenkapazität gespart werden. Das anschließende signalangepasste Filter erzeugt daraufhin mit jedem, von dem Empfangs-beamformer berechneten Eingangsvektor, eine Zielmatrix. Die dafür benötigten Prototypen werden durch die Signalerstellung geliefert, die je nach Anforderung die MIMO-Signalverarbeitung oder auch die Dopplerverschiebung mit der Prototypenerstellung kombinieren kann [3]. Der Schritt der Korrelation ist essentiell für die Bildgebung und erfordert die meisten Rechenoperationen je Block. An dieser Stelle wird deswegen Multithreading verwendet. Dabei werden die nötigen Vektormultiplikationen über die Anzahl der Prototypen auf die verschiedenen Prozessorkerne aufgeteilt, um die Last möglichst gleich zu verteilen. Eine offline mit den Positionsdaten der verteilten Knoten initialisierte Matrix ordnet dabei den Eingangsbeams entsprechende Prototypensignale zu. Die schließlich entstandene Zielmatrix wird schlussendlich durch den Kompressor an die Bildschirmauflösung angepasst und verkleinert. Dies geschieht wahlweise durch Maximalwert-, oder Durchschnittsbildung über benachbarte Abtastpunkte.

Signalerstellung

Die Signalerstellung wird für die Empfangs- und Sendesignalverarbeitung benötigt. Sie ist ein separates Modul, weil sie zum einen sowohl in verteilten Systemen als auch in monostatischen Systemen genutzt wird und deshalb flexibel initialisiert werden muss und zum anderen, damit die Rechenlast minimiert werden kann. Sowohl die Sendesignale als auch die Prototypensignale können vor dem Systemstart bereits initialisiert und gespeichert werden, sodass die Erzeugung während der Laufzeit keine Ressourcen verbrauchen muss. Das nimmt einen großen Teil der Last aus dem periodischen Rechenprozess. Damit trotzdem ein Wechsel der Signalformen zwischen den Pingperioden ohne aufwändiger Neuberechnung stattfinden kann, können ganze Sets an Sequenzen erstellt werden. Das aktuell genutzte Set kann dann mit einfacher Zeigermanipulation während der Laufzeit ohne Last geändert werden. Wird dennoch eine Sequenzänderung

gefordert, die noch nicht geladen ist, kann diese auf einem separaten Thread mit minimaler Prozessorlast durchgeführt werden. Das Modul beinhaltet die essentielle Sequenzerzeugung, die eine definierte Anzahl an Sequenzen und Sets erstellen kann. Zusätzlich beinhaltet es die Prototypenmanipulation, die die erstellten Daten für das signalangepasste Filter vorbereitet. Das umfasst neben der Dopplerspreizung auch das empfangsseitige Sendebestimmung.

Sendesignalverarbeitung

Die Sendesignalverarbeitung nutzt die durch die Signalerstellung bereitgestellten Sequenzen und rechnet zunächst einen Sendebestimmung. Nach einer Stavenselektion und der Ausgangsnormalisierung zur Anpassung an die Sendehardware werden die Signale über die Projektoren emittiert.

Kognition und Steuerung

Das Modul für die Kognition und Steuerung stellt die Intelligenz des Systems dar. Wenn es sich um ein verteiltes Setup handelt, werden zunächst die Daten der einzelnen SONAR-Knoten ausgetauscht. Die Fusions- und Kontrolleinheit ist dann dafür verantwortlich, basierend auf Positions- und Zeitdaten eine Masterkarte zu erstellen, indem ankommende Korrelationsmatrizen der verteilten Systeme fusioniert werden. Diese Masterkarte wird dann benutzt, um Ziele zu detektieren und um die Ziele Pfaden zuzuordnen. Das ganze funktioniert auf einem separaten Thread, um die rechenintensiven und zeitkritischen Kernmodule nur minimal zu belasten. Die Fusions- und Kontrolleinheit beinhaltet außerdem eine kognitive Einheit, die auf einem Deep Reinforcement Learning Algorithmus basiert [4]. Dieses Modul extrahiert aus den zusammengetragenen Daten der Masterkarte, des Detektors und des Trackers die für den nächsten Zeitpunkt optimale Strategie, die am Effizientesten die trainierten Ziele erfüllen kann. Diese Strategie kann mithilfe eines Parametersatzes zusammengefasst werden. Um sie umzusetzen, steht dem Modul der Zugriff auf einen global geteilten Parametersatz zur Verfügung. Jedes Modul auf einem SONAR-Knoten greift während der Laufzeit auf diesen Satz an Parametern zu, um die SONAR-Daten entsprechend zu manipulieren. Externe Knoten besitzen eine eigene Kopie des Kontrolleinheitmoduls, das nach Kommunikation mit allen anderen Knoten die Steuerung übernimmt.

Geteilter Parametersatz

Der geteilte Parametersatz umfasst sämtliche globale Variablen jedes Moduls. Die von den Parametern abhängigen Module enthalten Mechanismen, die bei einer Änderung den eigenen Algorithmus entsprechend der Forderung der Kontrolleinheit adaptieren. Für den Empfangs-beamformer enthält es die Anzahl und die Richtung der zu rechnenden Beams. Das signalangepasste Filter und der Kompressor können durch die minimale und maximale zu berechnende Detektionsreichweite gesteuert werden. Auch die Positionsdaten der verteilten SONAR-Knoten adaptieren die Berechnungen des signalangepassten Filters. Für die Sequenzerzeugung gibt es eine Reihe an Parametern, die die Signalform ändern können. Zur Auswahl stehen verschiedene Signaltypen

Algorithmic core module	Running time	Average	Minimum	Maximum	CPU load	Thread
▼ Underwater signal processing module	15476.0 μ s	7661.1 μ s	1156.5 μ s	277470.1 μ s	72.5 %	Algcore processing
Signal selection (Sonar to hydro...	-	-	-	-	-	-
▼ Sonar processing	14155.0 μ s	6211.4 μ s	0.1 μ s	276051.6 μ s	64.1 %	Algcore processing
Detector	-	20.1 μ s	9.3 μ s	39.0 μ s	-	Worker thread 0
LinearKalmanFilter	-	-	-	-	-	-
▼ Merge and control unit (MCU)	-	-	-	-	-	-
Overview	-	-	-	-	-	-
Network	18.7 μ s	21.7 μ s	7.9 μ s	139.4 μ s	0.1 %	Algcore processing
▼ Receive Array	13043.9 μ s	11532.8 μ s	6286.2 μ s	275405.3 μ s	61.1 %	Algcore processing
Analysis filterbank	466.9 μ s	490.8 μ s	443.9 μ s	1165.5 μ s	2.2 %	Algcore processing
Compressor	4670.5 μ s	4524.8 μ s	2805.2 μ s	41049.1 μ s	21.9 %	Algcore processing
Correlator	7678.6 μ s	6114.9 μ s	2572.4 μ s	250794.9 μ s	36.0 %	Algcore processing
▼ Prototype factory	-	-	-	-	-	-
Prototype analysis	-	-	-	-	-	-
Prototype manipulator	-	-	-	-	-	-
▼ Prototype spreader	-	-	-	-	-	-
Sinc interpolation	-	5721.3 μ s	5325.2 μ s	8228.4 μ s	-	Worker thread 0
▼ Receive beamformer	184.1 μ s	201.9 μ s	118.5 μ s	732.7 μ s	0.9 %	Algcore processing
Receive beamformer d...	-	-	-	-	-	-
Receive sensitivity	41.1 μ s	35.2 μ s	24.7 μ s	240.2 μ s	0.2 %	Algcore processing
Receive Array Setup	-	-	-	-	-	-
Receive side transmit array se...	-	-	-	-	-	-
SequenceCreator	-	-	-	-	-	-
Tracker	-	4.1 μ s	1.2 μ s	16.0 μ s	-	Worker thread 0
Transmit Array	599.3 μ s	807.9 μ s	467.6 μ s	2401.2 μ s	2.8 %	Algcore processing

Abbildung 2: Bildschirmfoto der Auslastung der Echtzeit-SONAR-Umgebung bei einer komplexen Parametrisierung.

wie beispielsweise eine kontinuierliche Welle, ein Chirp oder bandlimitiertes Rauschen mit weiter einstellbaren Parametern wie zum Beispiel die Mittenfrequenz oder die Bandbreite. Zusätzlich kann aber auch ein Signal aus einer Datei geladen werden oder mit einem Fenster versehen werden. Auch die Signallänge ist anpassbar. Diese Änderungen können entweder während der Laufzeit berechnet werden oder durch wechseln des Sequenzsets mit a priori Wissen ohne Verzögerung genutzt werden. Der Prototypenmanipulator basiert ähnlich wie der Empfangs-beamformer für den Betrieb mit orthogonalen Sequenzen auf Parametern für die Beamanzahl und -richtung. Auch können Dopplerspreizungen für verschiedene Zielgeschwindigkeiten eingestellt werden. Für die Sendesignalverarbeitung lässt sich der Beamformer analog zur Empfangssignalverarbeitung steuern.

Ergebnisse

Die Auslastung der berechnenden Hardware wurde für eine komplexe Parametrisierung getestet. Abbildung 2 zeigt einen im Programm implementierten Komplexitätsmonitor, der die Rechenzeit für die einzelnen Module je Block anzeigt. Als Testsetup wurde ein relativ großes System mit $N_{RX} = N_{TX} = 64$ Sende- und Empfangselementen parametrisiert. Als Modus wurde omnidirektionales MIMO mit $N_{beams} = 81$ Empfangs- und empfangsseitigen Sendebeams gewählt. Zusätzlich wurden zehn verschiedene Dopplerspreizungen berechnet. Es wurde eine Blockgröße von $N_{block} = 4096$ bei einer Abtastrate von $f_s = 192$ kHz gewählt. Je Block bleibt also eine Zeit von $\frac{N_{block}}{f_s} \approx 21$ ms, um alle Berechnungen durchzuführen. Der für den Test verwendete 12-Kern-Prozessor *AMD Ryzen 9 3900x* benötigt etwa 72.5 % der verfügbaren Zeit und besteht somit den Echtzeittest. Die

größte Last liegt bei dem signalangepassten Filter und dem anschließendem Kompressor. Wie bereits im Abschnitt *Signalerstellung* beschrieben, wird die Signalerstellung offline durchgeführt und fordert somit keine Auslastung.

Ausblick

In diesem Beitrag wurde die Softwarearchitektur eines SONAR-Systems vorgestellt, das einerseits flexibel zahlreiche Modi abbilden kann und andererseits verteilte Systeme unterstützt. Erweiterungen können durch den blockweisen Aufbau der einzelnen Elemente und den geteilten Parametersatz relative unkompliziert hinzugefügt werden. Aufgrund der Komplexität muss das Programm in den nächsten Schritten gerade in realer Umgebung ausführlich getestet werden. Vor allem der Test für die Anwendung mit verteilten SONAR-Knotenpunkten muss noch durchgeführt werden.

Literatur

- [1] T. Claussen, V. D. Nguyen, U. Heute und G. Schmidt, A Real-time MIMO System for Sonar Applications, DAGA Nürnberg, 2015
- [2] T. Kaak und G. Schmidt, An Introduction to Real-time Cognitive SONAR Systems Utilizing Novel MIMO Approaches. DAGA Kiel, 2017
- [3] T. Kaak und G. Schmidt, Real-time capable multiple-input-multiple-output SONAR systems – An algorithmic approach. IET Radar, Sonar & Navigation, 2021
- [4] N. Neumann, B. Kaulen, S. Christensen und G. Schmidt, Deep Reinforcement Learning for Cognitive SONAR Systems. DAGA Wien, 2021