|  |  |  |
| --- | --- | --- |
|  | |  |
| Konzeption und Realisierung eines Personenzählsystems mit Hilfe eines Radarsensors | | |
|  | | |
|  | | |
| des Studiengangs  an der Dualen Hochschule Baden-Württemberg Stuttgart | | |
| von | | |
| Abgabedatum: 10.06.2022 | | |
| Bearbeitungszeitraum | 17.01.2022 - 10.06.2022 | |
| Matrikelnummer, Kurs | 7081869, 3873765, TEL19GR5 | |
| Ausbildungsfirma | Infineon Technologies AG, Neubiberg | |
| Betreuer der Dualen Hochschule | Dr., Patrick Köberle | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Erklärung | | | | | |
| Ich versichere hiermit, dass ich meine mit dem Thema Konzeption und Realisierung eines Personenzählsystems mit Hilfe eines Radarsensors selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.  Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt. | | | | | |
|  |  |  |  |  |  |
| Ort |  | Datum |  | Unterschrift |  |

Abstract

Inhalt

[Abbildungsverzeichnis V](#_Toc105335797)

[Tabellen V](#_Toc105335798)

[Abkürzungen VI](#_Toc105335799)

[Formelgrößen und Einheiten VI](#_Toc105335800)

[4.1.1 dritte Ebene 4](#_Toc105335801)

[4.2.1 dritte Ebene 4](#_Toc105335802)

[Literaturverzeichnis 6](#_Toc105335803)

[Anhang 7](#_Toc105335804)

Abbildungsverzeichnis

**No table of figures entries found.**

Tabellen

**No table of figures entries found.**

Abkürzungen

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Formelgrößen und Einheiten

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

# Einleitung

# Aufgabenstellung

# Stand der Technik

# xxxx

## Unterüberschrift

Text

### dritte Ebene

Text

#### vierte Ebene

Text

## Unterüberschrift

Text

### dritte Ebene

Text

## Programmierung des Zählalgorithmus

Nachdem die Rohdaten verarbeitet, gefiltert und gemittelt worden sind, müssen nun die erkannten Targets aus der Klasse FTarget verfolgt und gezählt werden. Im Folgenden werden dazu das Konzept und die Realisierung des Trackings und der Zählung, Problemfälle, Schwierigkeiten sowie Lösungen des Zählalgorithmus beschrieben.

### Konzept und Realisierung der Personenverfolgung

Die Personenverfolgung ist der wahrscheinlich komplizierteste Teil des Programms. Hier könnte es sich anbieten, klassische fertige Filter zur Personenverfolgung zu verwenden, die in der Radartechnik schon bewährt sind wie ein Alpha-Beta-Tracker oder ein Kalman Filter. Mit dem vorliegenden Radar-Sensor hat sich die Geschwindigkeits-Messung mithilfe des Doppler-Effekts als nicht sehr präzise und genau in Versuchen ergeben. Aus diesem Grund wurden die zwei genannten Filter nicht verwendet, welche sonst eine gute Lösung gewesen wären. Ein anderer Vorteil einer eigenen Lösung ist, dass immer neue Personen in verschiedensten Positionen auftauchen können. Diese müssen anhand des eigenen Programms klassifiziert werden können.

In Figure 1 ist der Programmablaufplan des Personenverfolgung dargestellt, welcher im Folgenden weiter erläutert wird.

Die neuen aufbereiteten Daten werden mit jedem Durchlauf des Programms in den Vektor Targets der Klasse FTarget geschrieben, d.h. jedes Element des Vektors ist eine Person, welche duch die Klasse FTarget beschrieben wird. Dabei ist erst zur aktuellen Laufzeit bekannt, wie groß dieser Vektor ist, da nicht bekannt ist, wie viele Personen erkannt werden. Daher muss dieser Vektor bei jedem Durchlauf erst komplett gelöscht werden, bevor neue Daten hineingeschrieben werden. In der Variable position wird dazu dann die Anzahl an erkannten Personen in diesem Durchlauf gespeichert. Die For-Schleifen in diesem Abschnitt benutzen dann meist diese Variable als Endkriterium.

C:\Users\RostJosh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\AblaufplanTracking.emf

Figure 1: Programmablaufplan der Personenverfolgung

Im ersten Durchlauf wird Targets an den Vektor Targets\_aktuell, welcher auch aus Elementen der Klasse FTarget besteht, übergeben. Diese Variable wird dann gespeichert und nicht in jedem Durchlauf zurückgesetzt, um Personen zu verfolgen, die öfters erkannt werden.

Targets\_aktuell beschreibt im aktuellen Durchlauf den Abstand, Winkel und Geschwindigkeit einer Person und wird dann im Schritt der Zählung ein Herkunftsort gegeben, um schon geortete Personen zu identifizieren. In jedem weiteren Durchlauf nach dem ersten wird nun die Beträge der Differenz des Winkels und des Abstands zwischen jedem Objekt in dem Vektor Targets (nun neuere Werte) und jedem Objekt des Vektors Targets\_aktuell (hier sind theoretisch jetzt die Werte des vorherigen Durchlaufs) ermittelt. Diese Differenzen werden dann in den Differenzmatrizen Difference und Angle\_Difference gespeichert. Diese Matrizen werden dabei in jedem Durchlauf mit NaN in der Größe position initialisiert.

Dann werden diese Differenzmatrizen gefiltert. Es werden dann nur Werte, die nicht NaN und unter dem jeweiligen max. Veränderungsschwellen sind, durchgelassen. Die Veränderungsschwelle für den Abstand ist in der Variable DetectionDistance gespeichert und hat den Wert 1. Für den Winkel ist die Schwelle in AngleDetectionDistance gespeichert und hat den Wert 10. Unter der Annahme, dass der Sensor verlässliche Werte in kurzen Abständen zurückgibt, sorgen diese Schwellen dafür, dass sich Personen nur mit kleinen Differenzen fortbewegen können. Die Werte bedeuten, dass ein erkanntes Target im nächsten Durchlauf nur max. 1m und 10° bewegen kann, sonst wird es als neues Target erfasst und das alte wird gelöscht. Die gefilterten Werte werden in den Differenzmatrizen min\_Difference\_Location und min\_Difference\_Angle derselben Größe gespeichert. Auch diese Matrizen werden dabei in jedem Durchlauf mit NaN initialisiert.

Für den Fall, dass nun mehrere Personen sich in unmittelbarer Nähe aufhalten und als unterschiedliche Personen erkannt werden, müssen diese auch durchgehend erkannt werden. Aus diesem Grund werden aus der gefilterten Differenzmatrix min\_Difference\_Location reihenweise die kleinsten Werte gesucht. Diese kleinsten Werte dann in der Variable minValue gespeichert, welche jeden Durchlauf mit -1 initialisiert wird. Da nur Beträge in den Matrizen stehen können, ist -1 ein nicht gültiger Wert und kann nicht in den Matrizen stehen. Dies ist somit eine Absicherung für Fehler. minValue wird allerdings nur aus min\_Difference\_Location gesucht, da die Winkelmessung nicht sehr zuverlässig gewesen ist. Als Verbesserungspunkt für diesen Teil könnte man noch die kleinsten Winkelunterschiede suchen und diese mit den kleinsten Werten aus minValue korrelieren.

Nachdem man die kleinsten Differenzen gefunden hat, werden die Spalten gesucht, in denen die minimalen Werte gespeichert sind. Die Spalten werden dann in minColumn gespeichert. In der Figure 2 und der Figure 3 wird anhand eines Beispiels gezeigt, wie die bisher genannten Variablen in diesem Abschnitt mit echten Werten aussehen. In Targets\_aktuell sind 2 alte Targets gespeichert und in Targets 3 neue mit jeweils unterschiedlichen Distanz, Winkel und Geschwindigkeitswerten. Dem ersten Target in Targets\_aktuell ist auch ein Herkunftsort zugeordnet, dazu später mehr. In Figure 3 ist nun auch die Differenzmatrix dargestellt. Jede Zeile sind dabei die alten Distanzwerte aus Targets\_aktuell und in jeder Spalte werden davon die Distanzwerte aus Targets abgezogen. Nach der Filterung, die dann in min\_Difference\_Location zu sehen ist, werden die kleinsten Werte gesucht und dann die Spalten dieser in minColumn angezeigt. Somit kann man nun sehen, dass Targets(1) Targets\_aktuell(1) zugeordnet werden soll und Targets(3) Targets\_aktuell(3). Dabei muss natürlich beachtet werden, dass keine Ziffer doppelt in minColumn vorkommt. Falls doch muss diesem Element von Targets\_aktuell das zweitnächste Objekt der Vektors Targets zugeordnet werden. Falls dies nicht möglich ist, soll das alte Element von Targets\_aktuell gelöscht werden und das neue Element von Targets einfach zugewiesen werden. Falls in der Spalte keine Ziffer doppelt vorkommt, können existierende Elemente von Targets\_aktuell ihre neuen Werte aus Targets bekommen, indem sie die Ziffern aus minColumn zum dereferenzieren verwenden. Im Falle, dass das Elemente von Targets\_aktuell nicht existiert, muss es neu angelegt werden und dann mit den neuen Daten aus Targets zugewiesen werden.

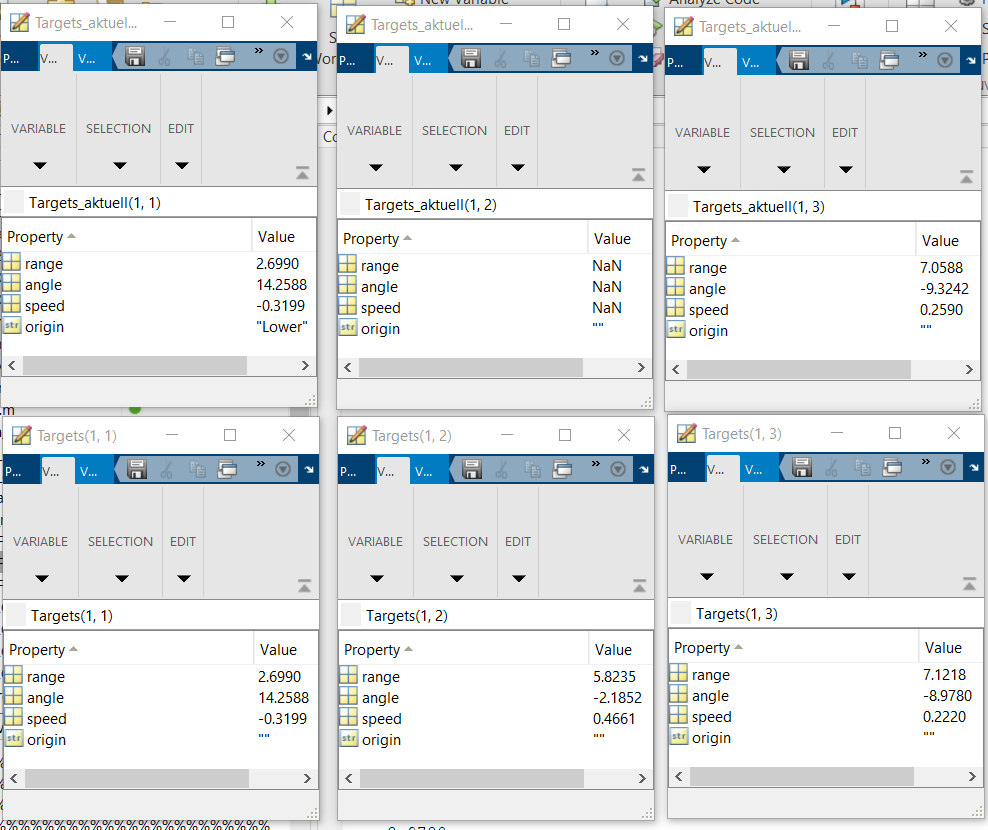


Figure 2: Darstellung eines Zeitpunkts mit 2 alten Targets und 3 neuen Targets

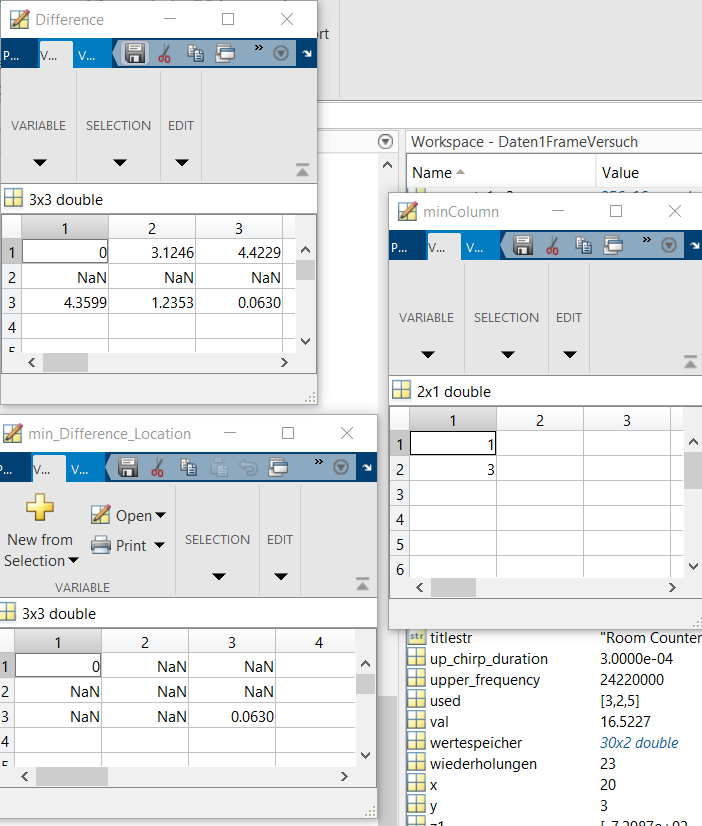


Figure 3: Zugehörige Darstellung der Differenzmatrix, gefilterten Differenzmatrix und Gefundenen Spalten

### Konzept und Realisierung der Personenzählung

In Figure 1 ist das Konzept des Versuchsaufbau dargestellt, welches das Konzept der Personenzählung direkt beeinflusst. Da sich die Präzision und Zuverlässigkeit des Abstandes in Versuchen als hoch erwiesen hat und die Zuverlässigkeit der Winkelmessung geringer, wurde dieser ungewöhnliche Versuchsaufbau gewählt, um Personen über einen längeren Zeitpunkt zu verfolgen und zu zählen.

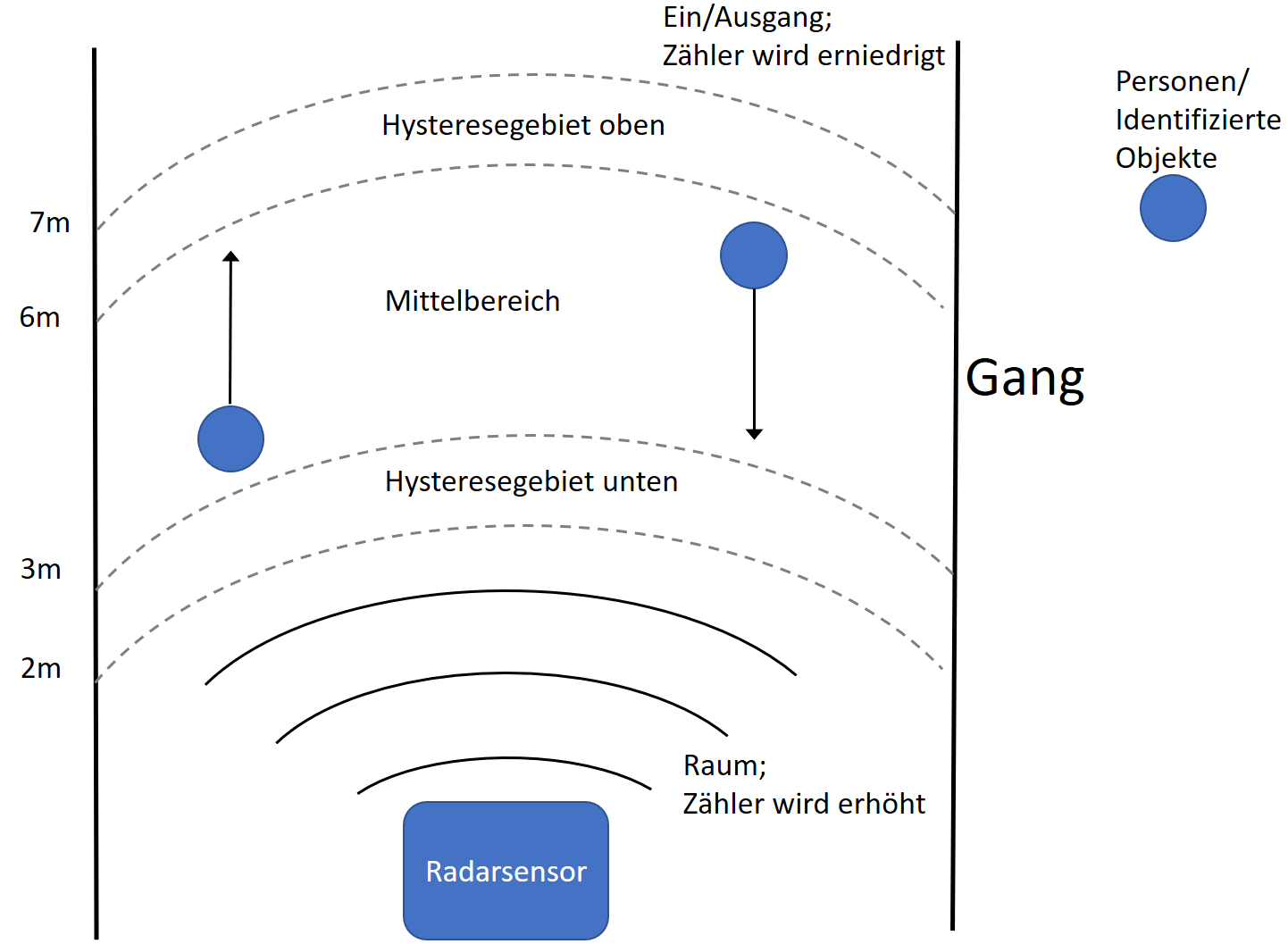


Figure 4: Konzept des Aufbaus

Der Bereich über 7 m wird als Ein- und Ausgang deklariert. Wenn eine Person erkannt wird, die sich vorher in dem nahen Hysterese-Bereich vom Radar aufgehalten und über 7m hinaus noch erkannt wird, wird der Raum Zähler erniedrigt. Durchläuft eine Person hingegen zuerst den Hysterebereich oben und ist dann unterhalb der unteren Hyteresegrenze des unteren Hysteresegebiets erkennbar (also im Raum Bereich), dann wird dies so bewertet, dass eine Person den Raum betritt und der Raum Zähler wird erhöht. Wenn eine Person sich nur im Ein/Ausgangsbereich oder im Raum Bereich befindet und dort erkannt wird, wird diese Person nicht weiter verarbeitet. Falls eine Person im Mittelbereich erkannt wird, die davor nicht in einer der beiden Hysteregebiete aufgehalten hat, wird dies als Fehler bewertet und diese Person wird gelöscht.

### Problemfälle, Schwierigkeiten und Lösungen

# Zusammenfassung

Literaturverzeichnis

**There are no sources in the current document.**

Anhang