



# ZUSAMMENFASSUNG MULTISENSORNAVIGATION

Fakultät für Geoinformatik  
of Hochschule für Angewandte Wissenschaften München

## **Zusammenfassung**

geschrieben bei

**Felix Strobel**

March 2019

# Inhaltsverzeichnis

<b>1</b>	<b>Stoffübersicht</b>	<b>6</b>
1.1	Homogene Matrix . . . . .	6
1.2	Pose . . . . .	6
1.3	Singulärwertzerlegung . . . . .	6
1.4	Levenberg-Marquardt . . . . .	6
1.5	Bündelblockausgleichung . . . . .	7
1.6	Trajektorie . . . . .	7
1.7	Koppelnavigation . . . . .	7
1.8	Sigma Point Kalman Filter . . . . .	7
1.9	Extended Kalman Filter . . . . .	7
1.10	ICP oder Scanmatching . . . . .	8
1.11	Partikelfilter . . . . .	8
1.12	RANSAC . . . . .	8
1.13	Loop closure . . . . .	8
1.14	SLAM . . . . .	8
<b>2</b>	<b>Einführung</b>	<b>9</b>
2.1	Lokalisieren . . . . .	9
2.2	3D Konstruieren . . . . .	9
2.3	Mobile Mapping System . . . . .	9
<b>3</b>	<b>Mathematische Grundlagen</b>	<b>10</b>
3.1	Skalarprodukt . . . . .	10
3.2	Winkel zwischen zwei Vektoren . . . . .	10
3.3	Orthogonalität . . . . .	10
3.4	Vektorprodukt . . . . .	10
3.4.1	Matrix/Vektorform . . . . .	10
3.5	Rechte Hand Regel . . . . .	11
3.6	Rotation . . . . .	11
3.6.1	Definition . . . . .	11
3.6.2	Rotation vs. Spiegelung . . . . .	11
3.6.3	Rotation . . . . .	11
3.6.4	Spiegelung . . . . .	11
3.6.5	Prüfung auf Rotation . . . . .	11
3.7	Homogene Matrix . . . . .	12
3.7.1	Translation und Rotation . . . . .	12
3.7.2	Vorteile . . . . .	12

3.7.3	Intepretation als Koordinatensystem . . . . .	12
3.7.4	Berechnung Koordinatenursprung . . . . .	12
3.7.5	Berechnung der Achsen . . . . .	12
3.8	Koordinatensysteme . . . . .	12
3.8.1	Affines Koordinatensystem . . . . .	12
3.8.2	Orthogonales Koordinatensystem . . . . .	12
3.8.3	Orientierungstreues Koordinatensystem . . . . .	12
3.9	RPY Darstellung . . . . .	13
3.9.1	Rotation als RPY . . . . .	13
3.9.2	RPY to Matrix . . . . .	13
3.9.3	Weitere Darstellungen . . . . .	13
<b>4</b>	<b>Cartesian Motion</b>	<b>14</b>
4.1	Glatter Übergang zwischen zwei Koordinatensystemen . . . . .	14
4.2	Warum nicht mit 3x3 Matrix . . . . .	14
4.3	Nachteil von RPY Darstellung . . . . .	14
<b>5</b>	<b>Trajektorie</b>	<b>15</b>
5.1	Definition . . . . .	15
5.2	Quintisches Polynom . . . . .	15
5.2.1	Interpolations Annahmen . . . . .	15
5.2.2	Pseudocode Implementierung . . . . .	15
5.2.3	Überbestimmung . . . . .	16
5.2.4	Vorteile des Quintischen Polynoms . . . . .	16
5.2.5	Vorteile von Endpunkt $T = 1$ . . . . .	16
<b>6</b>	<b>Kalmanfilter</b>	<b>17</b>
6.1	Eigenschaften . . . . .	17
6.2	Modellannahmen des Kalmanfilters . . . . .	17
6.3	Anwendung und Wirkweise . . . . .	17
6.4	Erweiterter Kalmanfilter . . . . .	17
6.5	Vorteile der Normalverteilung . . . . .	18
6.6	Extremwert/Mittelwert der Normalverteilung . . . . .	18
6.7	Kernidee der Schätzung . . . . .	18
6.8	Innovation . . . . .	18
6.9	Parameter des Update Gain . . . . .	18
6.10	Wertebereich des Update Gain . . . . .	19
6.11	Eingrenzung des Mittelwertes einer Schätzung . . . . .	19
6.12	Berechnung der Varianz . . . . .	19
<b>7</b>	<b>Kalmanfilter Gleichungen</b>	<b>20</b>
7.1	Begriffserklärungen . . . . .	20
7.2	Gleichungen . . . . .	20

# Abbildungsverzeichnis

6.1 Grundidee des Kalmanfilters . . . . .	18
---	----

# **Tabellenverzeichnis**

# 1 Stoffübersicht

## 1.1 Homogene Matrix

Matrix, die Positionen und Orientierung beinhaltet

$$M = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_1 \\ r_{31} & r_{23} & r_{33} & t_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.1)$$

Wird verwendet, damit man einfacher rechnen kann.

## 1.2 Pose

Beschreibt die Lage eines Körpers. Mit seinen Raumkoordinaten und dem Heading im Bezug auf ein Referenzkoordinatensystem.

Im  $R_2$  z.B.:  $(x, y, \alpha)$

## 1.3 Singulärwertzerlegung

Zerlegung einer Matrix in 3 Spezielle Matrizen welche miteinander Multipliziert die grundlegende Matrix ergeben. Auf der Hauptdiagonalen der mittleren Matrix stehen die Singularitäten der grundlegenden Matrix.

## 1.4 Levenberg-Marquardt

Der Levenberg-Marquardt-Algorithmus ist ein numerischer Optimierungsalgorithmus zur Lösung nichtlinearer Ausgleichs-Probleme mit Hilfe der Methode der kleinsten Quadrate. Das Verfahren kombiniert das Gauß-Newton-Verfahren mit einer Regularisierungstechnik, die absteigende Funktionswerte erzwingt. Deutlich robuster als das Gauß-Newton-Verfahren, das heißt, er konvergiert mit einer hohen Wahrscheinlichkeit auch bei schlechten Startbedingungen, allerdings ist auch hier Konvergenz nicht garantiert. Ferner ist er bei Anfangswerten, die nahe dem Minimum liegen, oft etwas langsamer.

## 1.5 Bündelblockausgleichung

Das Optimieren der SSehstrahlenbündel einer 3D-Szene, die von mehreren Kameras bzw. von einer Kamera aus mehreren Perspektiven aufgenommen wird. Bei der Bündelblockausgleichung können gleichzeitig die Positionen der Punkte im 3D-Raum, die Positionen und Orientierungen der beobachtenden Kameras sowie deren interne Kalibrierparameter derart an die Messbilder angepasst werden, dass verbleibende Fehler (z. B. Bildverzerrungen, Messfehler der Auswertung) möglichst optimal auf alle Beobachtungen verteilt werden. Speziell wird der Begriff verwendet, um nicht nur einzelne Bildpaare (je 2 überdeckende Messbilder) photogrammetrisch auszuwerten, sondern eine beliebige Anzahl von zusammenhängenden Bildern (Block) miteinander zu verknüpfen. Zur Berechnung könnte man z.B. Levenberg-Marquardt-Algorithmus nehmen.

## 1.6 Trajektorie

Der Weg eines Objektes in Abhängigkeit von der Zeit. [5]  
Stellt auch die Lösungskurve einer Differenzialgleichung dar.

$$P(t) = \begin{pmatrix} a_0 + a_1t + a_2t^2 \\ b_0 + b_1t + b_2t^2 \end{pmatrix} \quad (1.2)$$

## 1.7 Koppelnavigation

Koppelnavigation oder dead reckoning ist das aneinanderfügen vergangener Standortmessungen, welche jeweils relativ zum letzten Messzeitpunkt sind.

## 1.8 Sigma Point Kalman Filter

Kalman Filter [6] für nicht lineare Gleichungssysteme. Legt eine Normalverteilte Punktwolke um den aktuellen Punkt. Stabiler als der Kalmanfilter.

Er ist für sehr nicht lineare Zusammenhänge besser geeignet, da keine Linearisierung stattfindet.

Sowohl das Bewegungsmodell, wie auch das Messmodell können mit Sigmapoints berechnet werden. Es kann aber auch nur das Messmodell mit Sigmapoints in den neuen Zustand überführt werden.

## 1.9 Extended Kalman Filter

Der EKF ist eine nicht lineare Version des Kalman Filters welcher mittels einer Schätzung des aktuellen Mittels und der Kovarianzen linearisiert wird. Diese Linearisierung kann zu einer Ungenauigkeit des Filters führen. In extremen Fällen kann es zu einer Divergenz des Filters führen. Der Filter erreicht eine First-Order accuracy [1].

## 1.10 ICP oder Scanmatching

**Iterative Closest Point** um den kürzesten Abstand zwischen zwei Punktwolken zu bestimmen. Wird genutzt um zwei verschiedene Punktwolken aufeinander anzupassen, dazu müssen diese bereits näherungsweise angepasst sein.

## 1.11 Partikelfilter

Kann aus dem Vergleich vieler Messungen zu einer bekannten "Karte" den Ort absolut bestimmen. Damit kann er sich global positionieren. Er benötigt deutlich mehr Speicher als der Kalmanfilter und ist etwas ineffizienter, aber durch die Möglichkeit der globalen Positionierung und des fehlen einer Linearisierung ist er stabiler als ein Kalmanfilter.

## 1.12 RANSAC

Random sample consensus - Eine iterative Methode um outlier zu erkennen oder eine Gerade durch eine Punktwolke zu legen, welche viele outlier hat. Er wird u.A. im Bereich des Maschinellen Sehens verwendet um eine um Ausreißer bereinigte Datenmenge (Consensus Sets) zu erstellen. Das Consensus Sets findet in Verfahren, welche die Methode der kleinsten Quadrate verwendet, besonders wichtig, da diese mit Zunahme der Ausreißer instabiler werden.

## 1.13 Loop closure

Wenn Messungen einen geschlossenen loop bilden. Dies kann genutzt werden um Filter und Ausgleichungen zu testen oder zu verbessern.

## 1.14 SLAM

**Simultaneous localization and mapping** - Zeitgleiches Positionieren und Mapping der Messdaten. Wird in unbekannter Umgebung verwendet um eine Map zu erstellen in der sich dann positioniert werden kann.



## **2 Einführung**

### **2.1 Lokalisieren**

Die Fähigkeit sich gegenüber eines Bezugssystem zu Positionieren.

### **2.2 3D Konstruieren**

Daten werden immer vom aktuellen Standort aus aufgenommen. Diese können dann nach einer Lokalisierung ins Globale System überführt werden.

Wenn Daten von mehreren Positionen aus aufgenommen werden, so muss die Position der Sensoren zueinander bekannt sein.

### **2.3 Mobile Mapping System**

Eigenschaften eines Mobile Mapping Systems:

1. Mobile Plattform (Roboter, Flugzeug, Auto, etc.)
2. Multisensoraufbau zur Vermessung der Umgebung in zwei- oder dreidimensionaler Form
3. Berechnung des Umgebungsmodells online aber auf offline möglich.

# 3 Mathematische Grundlagen

## 3.1 Skalarprodukt

$$\vec{a} \circ \vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot \cos \varphi \quad (3.1)$$

$$\vec{a} \circ \vec{b} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \cdot \begin{pmatrix} b_x \\ b_y \end{pmatrix} = a_x b_x + a_y b_y \quad (3.2)$$

## 3.2 Winkel zwischen zwei Vektoren

$$\cos \varphi = \frac{\vec{a} \circ \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \quad (3.3)$$

## 3.3 Orthogonalität

Zwei Vektoren  $\vec{a}$  und  $\vec{b}$  sind Orthogonal zueinander wenn das Vektorprodukt 0 ergibt.

$$\vec{a} \circ \vec{b} = 0 \quad (3.4)$$

## 3.4 Vektorprodukt

$$\vec{a} \times \vec{b} = \vec{c} \quad (3.5)$$

$$|\vec{c}| = |\vec{a}| \cdot |\vec{b}| \cdot \sin \varphi \quad \text{for } (0^\circ \leq \varphi \leq 180^\circ) \quad (3.6)$$

### 3.4.1 Matrix/Vektorform

$$\vec{a} \times \vec{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - b_z a_y \\ a_z b_x - b_x a_z \\ a_x b_y - b_y a_x \end{pmatrix} \quad (3.7)$$

## 3.5 Rechte Hand Regel

Die Rechte Hand gibt die Richtung der Achsen vor, die Vektoren (Finger) bilden ein Rechtssystem deren Vektorprodukte alle 0 ergeben.

- Daumen: X
- Zeigefinger: Y
- Mittelfinger: Z

## 3.6 Rotation

### 3.6.1 Definition

1. Winkel der Drehung um jeweils eine Achse
2. Rotationsmatrix multipliziert mit der Transponierten ergibt die Einheitsmatrix  $I$
3. Spalten stehen Senkrecht aufeinander

### 3.6.2 Rotation vs. Spiegelung

### 3.6.3 Rotation

überführt ein Rechtwinkliges/Rechtshändiges Koordinatensystem in ein anderes Rechtwinkliges/Rechtshändiges Koordinatensystem.  $RxR$  ist Einheitsmatrix.

### 3.6.4 Spiegelung

Die Hand wechselt. Die Determinante wird -1

### 3.6.5 Prüfung auf Rotation

```
1 if abs(max(max((rotation * rotation') - eye(3)))) > 0.00000000000001
2 isRotation = false;
3 else
4 isRotation = true;
```

## 3.7 Homogene Matrix

### 3.7.1 Translation und Rotation

$$M = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.8)$$

### 3.7.2 Vorteile

Mehrere Bewegungen können durch Matrixmultiplikation aneinander gekettet werden bzw. leichter invertiert werden.

### 3.7.3 Interpretation als Koordinatensystem

Eine Homogene Matrix kann auch als Koordinatensystem interpretiert werden.

### 3.7.4 Berechnung Koordinatenursprung

$$T = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (3.9)$$

### 3.7.5 Berechnung der Achsen

$$x = \begin{pmatrix} r_{11} \\ r_{21} \\ r_{31} \end{pmatrix}, y = \begin{pmatrix} r_{12} \\ r_{22} \\ r_{32} \end{pmatrix}, z = \begin{pmatrix} r_{13} \\ r_{23} \\ r_{33} \end{pmatrix} \quad (3.10)$$

## 3.8 Koordinatensysteme

### 3.8.1 Affines Koordinatensystem

Lineare Koordinatensystem

### 3.8.2 Orthogonales Koordinatensystem

Rechtwinklig

### 3.8.3 Orientierungstreues Koordinatensystem

Es bleibt nach der rechten Hand definiert.

## 3.9 RPY Darstellung

Jede Rotation kann als Roll, Pitch, Yaw dargestellt werden.

Roll    X-Achse

Pitch   Y-Achse

Yaw    Z-Achse

### 3.9.1 Rotation als RPY

Dabei können Singularitäten auftreten. (Bei Pitch =  $\pi/2$  bzw.  $90^\circ$ )

**Matrix to RPY**

```

1 y = atan2(rotation(2,1), rotation(1,1));
2 p = atan2(-rotation(3,1), rotation(1,1) * cos(y) + rotation(2,1) * sin(y));
3 r = atan2(rotation(3,2)/cos(p), rotation(3,3)/cos(p));

```

### 3.9.2 RPY to Matrix

Eine Rotation, welche zuerst um die Roll, dann die Pitch und zuletzt die Yaw achse dreht

$$R_{\alpha,\beta,\gamma} = \begin{pmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{pmatrix} \quad (3.11)$$

### 3.9.3 Weitere Darstellungen

- Quaternionen [2]
- Axis/Angle Darstellung [3] -> numerisch stabil

## 4 Cartesian Motion

Cartesian Motion ist eine Bewegung auf  $n$  rechtwinklig zueinander stehenden Achsen.

### 4.1 Glatter Übergang zwischen zwei Koordinatensystemen

Um den Übergang zwischen Koordinatensystemen zu berechnen muss für die Rotation die RPY Darstellung [3.9] verwendet werden.

Anschließend wird Komponentenweise interpoliert.

1. Interpolieren von RPY

$$\begin{aligned}r^{(i)}(s) &:= (1-s) \cdot r^{(0)} + sr^{(N)} \\p^{(i)}(s) &:= (1-s) \cdot p^{(0)} + sp^{(N)} \\y^{(i)}(s) &:= (1-s) \cdot y^{(0)} + sy^{(N)}\end{aligned}\tag{4.1}$$

$$\text{mit } s := \frac{k}{N+1} \mid k \in \{1, \dots, N-1\}$$

2. Interpolieren des Ursprungs

$$\begin{aligned}T_x^{(i)}(s) &:= (1-s) \cdot T_x^{(0)} + sT_x^{(N)} \\T_y^{(i)}(s) &:= (1-s) \cdot T_y^{(0)} + sT_y^{(N)} \\T_z^{(i)}(s) &:= (1-s) \cdot T_z^{(0)} + sT_z^{(N)}\end{aligned}\tag{4.2}$$

### 4.2 Warum nicht mit 3x3 Matrix

Da die Eigenschaften der Drehungen verloren gehen würden. Daher:

1. Überführung in RPY Darstellung
2. Überführung in nächstes Koordinatensystem.
3. Zurück in Rotationsmatrix

### 4.3 Nachteil von RPY Darstellung

Dabei können Singularitäten auftreten. Bei Pitch =  $\pi/2$  bzw.  $90^\circ$ )

# 5 Trajektorie

## 5.1 Definition

Der Weg eines Objektes in Abhängigkeit von der Zeit.  
Stellt auch die Lösungskurve einer Differenzialgleichung dar.

$$P_{(t)} = \begin{pmatrix} a_0 + a_1 t + a_2 t^2 \\ b_0 + b_1 t + b_1 t^2 \end{pmatrix} \quad (5.1)$$

## 5.2 Quintisches Polynom

Ein Quintisches Polynom enthält 6 Komponenten  $\{a_0, a_1, a_2, \dots, a_5\}$  und ist, wie der Name vermuten lässt, vom 5. Grad.

### 5.2.1 Interpolations Annahmen

Es werden *Freiheitsgrad* + 1 Annahmen benötigt.  
Diese können z.B. Position, Geschwindigkeit und Beschleunigung jeweils für den Start- und End-Punkt.

Alternativ könnten auch Position, Range und Bearing jeweils für den Start und End-Punkt als Annahme dienen.

### 5.2.2 Pseudocode Implementierung

1. Erstellen der Strukturmatrix mit Normierter Zeit variable  $A(1 \times 6)$
2. Erstellen des Ergebnisvektors  $b$  ( $2 \times 1$ )
3. Lineares Gleichungssystem lösen
4. Zeitpunkt bestimmen, an dem die Position bestimmt werden soll.
5. Zeitpunkt in Gleichungssystem einsetzen

### 5.2.3 Überbestimmung

Wenn es mehr Interpolationsbedingungen als Freiheitsgrade gibt ist es nicht mehr eindeutig zu lösen. Dies führt zu einer Unsicherheit.

### 5.2.4 Vorteile des Quintischen Polynoms

- Beschreiben der Trajektorie speicherarm als Mathematische Funktion.
- Einfaches Interpolieren

### 5.2.5 Vorteile von Endpunkt $T = 1$

Das Polynom bleibt dadurch numerisch stabiler.



# 6 Kalmanfilter

## 6.1 Eigenschaften

1. Asymptotisch stabil
2. Kein anderer Schätzer liefert Schätzwerte mit kleinerer Varianz
3. Schätzwerte sind erwartungstreu (unbiased)
4. Ohne Rauschen ist er identisch zu einem rekursiven LSQ-Schätzer <sup>1</sup>
5. Er liefert den wahrscheinlichsten Schätzwert

## 6.2 Modellannahmen des Kalmanfilters

- Rauschen ist zufällig - normalverteilt und nicht systematisch
- Lineare Abbildung, damit die Normalverteilung eine Normalverteilung bleibt.
- Das Rauschen ist Zufällig.

Die Abbildung muss linear sein, damit die Varianz-Kovarianz Matrix im Anschluss immer noch eine Normalverteilung darstellt. (Die Normalverteilung muss eine Normalverteilung bleiben).

## 6.3 Anwendung und Wirkweise

Der Kalmanfilter wird verwendet um mittels einer Messung einen neuen Schätzwert zu bestimmen. Dies geschieht mit Hilfe des gewichteten Mittels.

## 6.4 Erweiterter Kalmanfilter

Der Kalmanfilter Arbeitet mit einem nicht linearen Zusammenhang. Allerdings wird dieser Zusammenhang mittels der Taylorapproximation <sup>2</sup> linearisiert, sodass die Normalverteilung bestehen bleibt.

---

<sup>1</sup>LSQ = Least Squares Quadratic

<sup>2</sup>Partielle Ableitung

## 6.5 Vorteile der Normalverteilung

- Normalverteilungen lassen sich einfach miteinander Verrechnen
- Das Messrauschen ist meist auch normalverteilt

## 6.6 Extremwert/Mittelwert der Normalverteilung

Der Mittelwert und Extremwert einer Normalverteilung sind identisch.

## 6.7 Kernidee der Schätzung

Die Kernidee hinter der Schätzung ist, dass sowohl das Messmodell wie Bewegungsmodell einen Fehler haben und die Wahrheit in der Mitte liegt. Um den Mittelwert der zwei Normalverteilungen zu berechnen werden sie zueinander auch noch gewichtet.

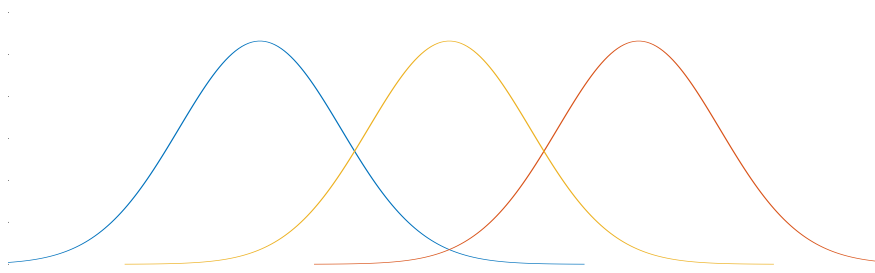


Abbildung 6.1: Grundidee des Kalmanfilters

## 6.8 Innovation

Die Innovation des Kalmanfilters ist der Vergleich zwischen Messung und Schätzung des Kalmanfilters.

## 6.9 Parameter des Update Gain

Die Standardabweichung der Messung so wie die der Schätzung.  
Das Update Gain wird wie folgt berechnet: ~

$$K = \frac{(\bar{\sigma}_k)^2}{(\bar{\sigma}_k)^2 + (\tilde{\sigma}_k)^2} \quad (6.1)$$

## 6.10 Wertebereich des Update Gain

Der Wertebereich des Gain ist normalisiert.  $[0, 1]$

## 6.11 Eingrenzung des Mittelwertes einer Schätzung

Der Mittelwert der **neuen Schätzung**  $\hat{x}_k$  wird mittels des Gain  $K$ , der **vorherigen Schätzung**  $\bar{x}_k$  und der **neuen Messung**  $\tilde{x}_k$  berechnet.

$$\hat{x}_k = \bar{x}_k + K (\tilde{x}_k - \bar{x}_k) \quad (6.2)$$

## 6.12 Berechnung der Varianz

$$\hat{\sigma}_k^2 = (1 - K) (\bar{\sigma}_k)^2 \quad (6.3)$$

# 7 Kalmanfilter Gleichungen

## 7.1 Begriffserklärungen

**Prediction Phase** Schätzen des Zustandes als Normalverteilung mit dem Bewegungsmodell  $F$  (z.B. aus Odometrie) und dem Bewegungsrauschen  $Q$

**Measurement update** Durchführung einer Messung  $z$  als Normalverteilte Zufallsvariable mit Messrauschen  $R$

**Estimation** Berechnung des Kalman Gain  $K$  aus geschätztem Zustand und neuer Messung mit Zustandsraummodellierung  $H$ . Daraus berechnet sich neuer Systemzustand  $x$  und  $P$

**a priori state estimate** Der in Schritt 1 geschätzte Systemzustand  $NV(\bar{x}_k, \bar{P}_k)$ . Dieser wird ohne einer neuen Messung geschätzt.

**a posteriori state estimate** Der in Schritt 3 geschätzte Systemzustand  $NV(\hat{x}_k, \hat{P}_k)$ . Dieser wird mit einer neuen Messung geschätzt.

## 7.2 Gleichungen

### 1. Prediction phase

$$\bar{x}_k = F \hat{x}_{k-1} \quad (7.1)$$

$F$  := Bewegungsmodell  
 $\hat{x}_{k-1}$  := Schätzung aus letztem Durchgang

$$\bar{P}_k = F \hat{P}_{k-1} F^t + Q_{k-1} \quad (7.2)$$

$\bar{P}_k$  := a priori state estimate  
 $\hat{P}_{k-1}$  := a posteriori state estimate des vorherigen Durchgangs  
 $Q_{k-1}$  := Rauschen des vorherigen Durchgangs

### 2. Measurement update

$$\tilde{z}_k = \text{Messung} + \tilde{R}_k \quad (7.3)$$

$\tilde{z}_k$  := neue Messung  
 $\tilde{R}_k$  := Rauschen der Messung

## 3. Estimation

$$K_k = \frac{\bar{P}_k H_k^t}{(H_k \bar{P}_k H_k^t + \tilde{R}_k)} \quad (7.4)$$

$K_k$  := neuer Kalman Gain

$H_k$  := Messmodell

$$\hat{x}_k = \bar{x}_k + K_k (\tilde{z}_k - H_k \bar{x}_k) \quad (7.5)$$

In dieser Gleichung werden die zwei Normalverteilungen in eine neue überführt (vgl. 6.1).

$H_k \bar{x}_k := \tilde{z}_k$  und entspricht damit der Schätzung über das Messmodell

$$\hat{P}_k = (I - K_k H_k) \bar{P}_k \quad (7.6)$$

# Literaturverzeichnis

- [1] [https://en.wikipedia.org/wiki/Order\\_of\\_accuracy](https://en.wikipedia.org/wiki/Order_of_accuracy)
- [2] <https://de.wikipedia.org/wiki/Quaternion>
- [3] [https://en.wikipedia.org/wiki/Axis%E2%80%93angle\\_representation](https://en.wikipedia.org/wiki/Axis%E2%80%93angle_representation)