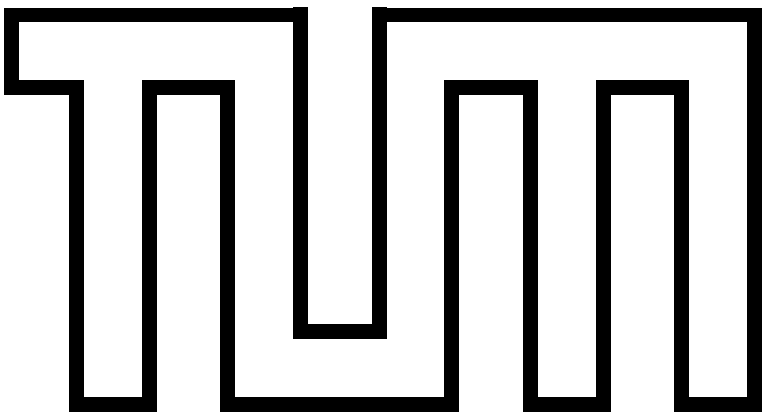


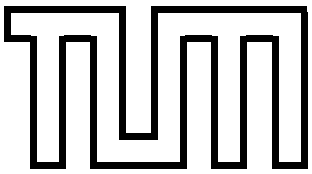
TECHNISCHE UNIVERSITÄT MÜNCHEN

FAKULTÄT FÜR INFORMATIK

Kalman-Filter und Partikel-Filter zur Selbstlokalisation von Robotern

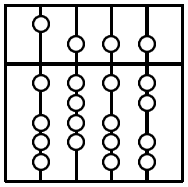
Systementwicklungsprojekt

Dirk Neumann



FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN
UNIVERSITÄT MÜNCHEN



Forschungs- und Lehreinheit IX
Bildverstehen und Wissensbasierte Systeme

Kalman-Filter und Partikel-Filter zur Selbstlokalisation von Robotern

Systementwicklungsprojekt

Dirk Neumann[1]

Betreuer : Dipl.-Inf. Thorsten Schmitt

Abgabedatum : April 2003

[1]email: dirk.neumann@informatik.tu-muenchen.de

Hiermit erkläre ich, dass ich dieses Systementwicklungsprojekt selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe.

München, April 2003

(*Dirk Neumann*)

Zusammenfassung

Ziel dieser Arbeit war die Verbesserung der Selbstlokalisation mittels statistischer Filter. Zur Lokalisation standen auf den mobilen Roboterplatformen Lokalisationsalgorithmen zur Verfügung, die jeweils auf Daten des Laser-Scanners, der omnidirektionalen Kamera oder der Odometrie aufbauen.

In einem etwas verkleinerten RoboCup-Spielfeld ([4]) wurden die Fehlerverteilungen der Lokalisationsschätzungen in Abhängigkeit von der Position des Roboters auf dem Feld gemessen. Die empirischen Fehlerverteilungen wurden durch multivariate Gauß-Funktionen approximiert. Zwischen den Messpunkten wurden die Parameter der Fehlerfunktionen durch Interpolation bestimmt. Dazu wurden die empirisch bestimmten Parameter vorab so transformiert, dass sie ebenfalls normalverteilt sind. Für das Odometrie-Fehlermodell wurden die Parameter der Fehlerverteilung anhand einer Testfahrt empirisch bestimmt.

Als statistische Filter wurde der Kalman-Filter und ein Partikel-Filter verwendet. Diese Filter wurden jeweils auf aufgenommene Daten von kritischen Situationen, wie sie im RoboCup auftreten (partielle Verdeckung, Positionsänderungen ohne Odometrie, starkes Beschleunigen, Kurvenfahren), angewendet. Die Qualität der Filterung wurde für diese Situationen, die die Voraussetzungen der Filter verletzen, ermittelt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Grundlagen

1.1 Wahrscheinlichkeit

Eine Wahrscheinlichkeitstheorie ist ein System aus Axiomen, das beschreibt, welche Eigenschaften eine Wahrscheinlichkeit besitzen muss. Zudem muss eine Wahrscheinlichkeitstheorie Regeln umfassen, um Wahrscheinlichkeiten zu konstruieren und zu interpretieren. Dabei wird neben empirischen auch logische und subjektive Theorien unterschieden ([8, S. 1]).

Im folgenden sollen die Axiome von Kolmogoroff verwendet werden, die Aussagen über Wahrscheinlichkeiten auf Aussagen über Mengen zurückführen (nach [6, S. 766]):

- Jeden zufälligen Ereignis E des Ereignisfeldes wird eine nichtnegative Zahl $P(E)$ zugeordnet, die als Wahrscheinlichkeit von E bezeichnet wird.
- Die Wahrscheinlichkeit des sicheren Ereignisses M ist $P(M) = 1$.
- Sind Ereignisse $E_1, E_2, \ldots, E_n$ paarweise unvereinbar, so ist
  $P(E_1 \cup E_2 \cup \ldots \cup E_n) = P(E_1) + P(E_2) + \ldots + P(E_n)$.

Problematisch ist hierbei, dass jedem Ereignis E eine Wahrscheinlichkeit E fest zugeordnet ist. Unsicherheiten über die Fehlerverteilung, wie sie z.B. bei Fuzzy-Mengen höherer Ordnung modelliert werden, und mögliche Modellverletzungen können nicht berücksichtigt werden. Die Forderung, dass die Wahrscheinlichkeit des sicheren Ereignisses 1 ist, wird z.B. beim Partikel-Filter erst durch "Normalisierung" erreicht und bezieht sich auch nicht mehr auf die der empirischen Ereignissen, sondern auf die der simulierten "Partikel".

1.2 Kalman-Filter

Der Kalman-Filter ([9], zit. nach [3]) stellt eine effiziente Filterung dar, wenn von normalverteilten Fehlern ausgegangen werden kann, deren Parameter a-priori bekannt ist. Wei-

terhin wird angenommen, dass die Veränderung des Zustandes linear und a-priori bekannt ist. Es wurde gezeigt, dass unter diesen Vorraussetzungen der Kalman-Filter die optimale Lösung darstellt ([11], zit. nach [5, S. 2]).

Vorraussetzungen Die Wahrscheinlichkeitsfunktion einer eindimensionalen Messung ist damit wie folgt definiert:

$$P(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\frac{-(y-\mu)^2}{2\sigma^2}}$$

Oder kürzer:

$$y \sim N\left(\mu, \sigma^2\right)$$

Im mehrdimensionalen Fall kann man analog formulieren:

$$y \sim N(\mu, \Sigma)$$

Wobei y den Messvektor, μ den Erwartungswert der Messvektoren und Σ die Kovarianz-Matrix darstellt. Für jeden Eigenvektor e_i mit dem Eigenwert λ_i^2 gilt dann:

$$P(\langle y, e_i \rangle) = \frac{1}{\sqrt{2\pi\lambda_i^2}} \cdot e^{\frac{-(\langle y,e_i\rangle - \langle \mu,e_i\rangle)^2}{2\lambda_i^2}}$$

Ferner wird gefordert, dass der Erwartungswert des Fehlers dem wahren Wert entspricht:

$$E(Y) = \mu = x$$

und dass die einzelnen Messungen voneinander unabhängig sind:

$$E\left(cov(y_i, y_j)\right) = 0, \ \forall\, i \neq j$$

Konstante Fehlerverteilung Für den speziellen Fall, dass die Kovarianz-Matrix konstant ist, reduziert sich der Kalman-Filter auf den Mittelwert:

$$\hat{x} = E(Y|t) = \frac{1}{t}\sum_{i=1}^{t} y_i$$

Die Kovarianz der Fehler ist eine Funktion der Anzahl der Messwerte t:

$$\hat{x} \sim N\left(E(Y),\ \frac{\Sigma}{t}\right) = N\left(E(X),\ \frac{\Sigma}{t}\right) = N\left(\mu,\ \frac{\Sigma}{t}\right)$$

Die Idee des Kalman-Filters ist diese Berechnung schrittweise vorzunehmen. Für eine konstante Fehlerverteilung ergibt sich dann:

$$\hat{X}|t+1 = \hat{X}|t + \frac{t}{t+1} \cdot (Y|t+1 - \mu)$$

Statt μ wird dabei der verfügbare Schätzer $\hat{X}|t$ eingesetzt:

$$\hat{X}|t+1 = \hat{X}|t + \frac{t}{t+1} \cdot \left(Y|t+1 - \hat{X}|t\right)$$

Optimale Gewichtung Sind die Parameter der Fehlerverteilung nicht konstant, so müssen zur optimalen Filterung die einzelnen Messwerte gewichtet werden. Die optimale Gewichtung ist dann:[2]

$$\hat{X}|t+1 = \hat{X}|t + \underbrace{\hat{\Sigma}_{\hat{X}}|t \cdot \left(\hat{\Sigma}_{\hat{X}}|t + \hat{\Sigma}_{Y}|t+1\right)^{-1}}_{=b} \cdot \left(Y|t+1 - \hat{X}|t\right) \tag{1}$$

[2] siehe z.B. [3]

Die Kovarianzmatrix des Schätzers $\hat{X}$ kann wie folgt geschätzt werden:

$$\hat{\Sigma}_{\hat{X}} = (1-b) \cdot \hat{\Sigma}_{\hat{X}} = \left(1 - \hat{\Sigma}_{\hat{X}}|t \left(\hat{\Sigma}_{\hat{X}}|t + \hat{\Sigma}_{Y}|t+1\right)^{-1}\right) \cdot \hat{\Sigma}_{\hat{X}} \tag{2}$$

Propagation Zustandsveränderungen müssen beim Kalman-Filter linear beschrieben werden:

$$\hat{X}|t+1 = A_{t+1 \leftarrow t}\, \hat{X}|t$$

Der Fehler des Schätzers werden dann entsprechend transformiert:

$$\hat{\Sigma}_{\hat{X}}|t+1 = A\, \hat{\Sigma}_{\hat{X}}|t\, A^T \tag{3}$$

1.3 Markov-Ketten

Bayes-Theorem Grundlegend für die *Markov-Chain-Method* und den Kalman-Filter ist das Bayes-Theorem. Es erlaubt die Berechnung der a-posteriori-Wahrscheinlichkeit $P(Y|X)$ aus der a-priori-Wahrscheinlichkeit der Messwerte $P(X|Y)$:

$$\boxed{P(X|Y) = P(Y|X) \cdot P(X) \cdot P(Y)^{-1}}$$

Beweis:

$$\begin{array}{rrcl}
 & P(X|Y) & = & P(X \cup Y) \cdot P(X) \\
\Leftrightarrow & P(X|Y) \cdot P(X)^{-1} & = & P(X \cup Y) \\
 & P(Y|X) & = & P(Y \cup X) \cdot P(Y) \\
\Leftrightarrow & P(Y|X) \cdot P(Y)^{-1} & = & P(Y \cup X) \\
\hline
\Rightarrow & P(X|Y) \cdot P(X)^{-1} & = & P(Y|X) \cdot P(Y)^{-1} \\
\Leftrightarrow & P(X|Y) & = & P(Y|X) \cdot P(X) \cdot P(Y)^{-1}
\end{array}$$

Das Bayes-Theorem kann direkt zur Integration von Messwerten verwendet werden. Dafür muss die a-priori-Verteilung der Messwerte $P(Y|X)$ bekannt sein. Da X nicht be-

kannt ist, wird die geschätzte Wahrscheinlichkeitsverteilung $P(\hat{X})$ verwendet:

$$P(\hat{X}|Y) = \int P(Y|\hat{X}) \cdot P(\hat{X})\, d\hat{X}$$

Für die sequentielle Berechnung ergibt sich folgende Formel:

$$P(\hat{X}|t+1) = \int P(y_{t+1}|\hat{X}, t) \cdot P(\hat{X}|t)\, d\hat{X}|t$$

Ist X ein diskrete Größe, dann kann das Bayes-Theorem ohne zusätzliche Annahmen angewandt werden. Ein Beispiel sind die gitterbasierte Monto-Carlo-Lokalizationsmethoden. X kann dabei die Punkte eines festgelegten Gitters annehmen.

$$P(\hat{X}|t+1) = \sum_i P(y_{t+1}|\hat{X}_i, t) \cdot P(\hat{X}_i|t) \tag{4}$$

2 Partikel-Filter

Die gitterbasierte Lokalisation hat jedoch den Nachteil, dass viele Gitterpunkte berechnet werden müssen, die eventuell eine sehr geringe Wahrscheinlichkeit besitzen und der Rechenaufwand mit der Auflösung des Gitters enorm ansteigt.

Die Partikel-Filter betrachten dagegen nicht den gesamten Zustandsraum X, sondern berechnen die Wahrscheinlichkeitsverteilung nur für eine Teilmenge $\hat{X} \subset X$. Diese Teilmenge ist selbst wieder eine Zufallsgröße. Bei den Eigenschaften der Teilmenge und in die Art und Weise wie die Teilmenge generiert wird, unterscheiden sich die Algorithmen zum Teil beträchtlich. Die Präzision der Schätzung kann erhöht werden, wenn vor allem Punkte mit hoher Wahrscheinlichkeit berücksichtigt werden und wenn sich die a-posteriori-Wahrscheinlichkeit nicht stark verändert, d.h. die Messwerte nicht stark von der Erwartung abweichen.

2.1 Sampling

Als *Sampling* bezeichnet man die Auswahl der betrachteten Zustandsmenge $\hat{X}$, den Partikeln, aus dem Zustandraum X. Diese Auswahl wird bei jeder Integration von Sensordaten oder Propagation des Zustandraumes neu berechnet und dient bei den meisten Partikel-Filter-Algorithmen gleichzeitig der Modellierung dieser Prozesse.

Sampling/importance resampling (SIR) Diese Methode geht auf Rubin ([13] und [14], zit. nach [12, S. 6]) zurück. Sie setzt voraus, dass aus der a-priori-Verteilung $P(\hat{X})$ gezogen werden kann und die Likelihood-Funktion $P(y|\hat{X})$ ausgewertet werden kann.

Die Idee ist, aus der Verteilung $P(\hat{X})$ mit einer Wahrscheinlichkeit π_i zu ziehen, wobei π_i proportional zur Likelihood des Messwertes y ist:

$$\omega_i = P(y|\hat{X}_i)$$

$$\pi_i = \frac{\omega_i}{\sum_{i=1}^{N} \omega_i} \tag{5}$$

Mit steigender Anzahl der *Samples* N konvergiert $P(\hat{X}|y)$ gegen eine statistisch abhängige Teilmenge aus $P(y)$. Denn statt eines Integrals wird zur Normalisierung die diskrete Summe $\sum_i \omega_i$ verwendet. Nach [10] (zit. nach [12, S. 7]) ist die Varianz, wenn sich die Wahrscheinlichkeitsschätzung $P(\hat{X})$ nicht stark mit $\hat{X}$ ändert, proportional zu:

$$\frac{E\left(\pi(\hat{X})^2\right)}{N} \tag{6}$$

Der SIR-Algorithmus ist also wenig robust in Bezug auf Ausreißer und stark gipflige Verteilungen.

Rejection Sampling Beim *Rejection sampling* wird aus $P(\hat{X})$ gezogen und das *Sample* mit einer Wahrscheinlichkeit von $\pi(\hat{x})$ akzeptiert wird:

$$\pi(\hat{X}) = \frac{P(y|\hat{X})}{P(y|\hat{x}_{max})}$$

$$\hat{x}_{max} = \arg\max_{\hat{x}} P(y|\hat{x})$$

Der Vorteil dieser Methode ist die Zufälligkeit der gezogenen a-posteriori Partikel. Problematisch kann sich die Berechnung von $\hat{x}_{max}$ und eine hohe Ablehnungsrate bei stark gipfligen Verteilungen erweisen.

2.2 Propagation

Die Zustandsänderung a kann als Wahrscheinlichkeitsverteilung $P(X_{t+1}|X_t, a)$ modelliert werden. Für die Propagation des Zustandraumes ergibt sich dann folgende Formel, die in Markow-Ketten verwendet wird ([2], zit. nach [5, S. 2]):

$$P(\hat{x}_{t+1}) = \int_{\hat{x}_t} P(\hat{x}_{t+1}|\hat{x}_t, a_{t+1}) \cdot P(\hat{x}_t)\, d\hat{x}_t \tag{7}$$

Um dieses statistische Modell zu simulieren, wäre es nicht sinnvoll, wie bei den Messwerte aus $P(\hat{X})$ mit einer Wahrscheinlichkeit von $P(a|\hat{X})$ neue Partikel zu ziehen. Denn wenn sich der Roboter bewegt, dann wird sich $P(\hat{X}|t+1)$ in aller Regel stark von $P(\hat{X}|t$ unterscheiden. Statt dessen ist günstiger die Wahrscheinlichkeit der Zustandänderung $P((\hat{x} + dx) - \hat{x}) = P(dx)$ zu betrachten.

$P(\hat{X}|t+1)$ erhält man dann durch ziehen aus $P(\hat{X}|t)$ und $P(dx)$, wobei dann entsprechend $x_{t+1} = x_t + dx$.

3 Fehlermodelle

Kovarianz-Berechnung Die Varianz ist definiert als der Erwartungswert der quadrierten Abweichung vom Erwartungswert einer Verteilung X:

$$\sigma^2(X) = E\left[(X - E(X))^2\right]$$

Die Kovarianz entsprechend:

$$cov(X, Y) = E\left[(X - E(X)) \cdot (Y - E(Y))\right]$$

Durch ausmultiplizieren erhält man ein einfache Berechnungsformel:

$$\begin{aligned} cov(X, Y) &= E\left[(X - E(X)) \cdot (Y - E(Y))\right] \\ &= E\left[XY\right] - E\left[X \cdot E(Y)\right] - E\left[Y \cdot E(X)\right] + E(X) \cdot E(Y) \\ &= E\left[XY\right] - 2 \cdot E(X) \cdot E(Y) + E(X) \cdot E(Y) \\ &= E\left[XY\right] - E(X) \cdot E(Y) \end{aligned}$$

Die Kovarianzmatrix einer zweier Variablen kann demnach recht einfach berechnet werden:

$$cov(X, Y) = \frac{1}{n} \sum_{t=1}^{n} x_t \cdot y_t - \frac{1}{n} \sum_{t=1}^{n} x_t \cdot \frac{1}{n} \sum_{t=1}^{n} y_t$$

Für den mehrdimensionalen Fall entsprechend:

$$\Sigma_X = \frac{1}{n} \sum_{t=1}^{n} x_t \cdot x_t^T - \left(\frac{1}{n} \sum_{t=1}^{n} x_t\right) \cdot \left(\frac{1}{n} \sum_{t=1}^{n} x_t\right)^T \tag{8}$$

Mahalanobis-Abstand Der Abstand r eines Punktes y von eine multivariaten Normalverteilung X wird meist durch den Mahalanobis-Abstand beschrieben. Dieser gibt den

Abstand in z-Werten (Vielfache der Standardabweichung) an (nach [7, S. 18]):

$$r^2 = (y - E(X))^T \Sigma^{-1} (y - E(X)) \tag{9}$$

Interpolation Gemessen wurden die Verteilung der Fehler der Positionsschätzung jeweils für die Kreuzungspunkte eines 1m-Gitters. Zwischen diesen Punkten wurde die Kovarianz-Matrix interpoliert.

Die Kovarianzen sind allerdings nicht normalverteilt, so dass eine Umformung vor der Interpolation sinnvoll wäre, um Schätzfehler nicht übermäßig zu gewichten. Für den Korrelationskoeffizienten steht dazu Fishers Z-Transformation zur Verfügung:[3]

$$Z = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right)$$

Die inverse Transformation entsprechend:

$$r = \frac{e^{2Z} - 1}{e^{2Z} + 1}$$

Der Korrelationskoeffizient steht mit der Kovarianz in folgender Beziehung:

$$r = \frac{cov(X,Y)}{\sqrt{var(X) \cdot var(Y)}}$$

Daher ist es sinnvoll die Kovarianzmatrix in die Korrelationsmatrix und die Matrix der Standardabweichungen zu zerlegen. Die Z-Transformation der Korrelationskoeffizienten und die Standardabweichungen könnten dann linear interpoliert werden.

Da dies recht aufwendig erscheint wurde aber ein einfacheres Verfahren verwendet. Es wurden Polynome 2. Grades benutzt, um zwischen den Messpunkten die Kovarianz

[3] siehe z.B. [1]

zwischen zwei Variablen zu interpolieren.

3.1 Lokalisation

Die Fehlerverteilung der Positionsschätzungen wurden für die Punkte eines 1m-Gitters bestimmt. Dazu wurde der Roboter auf die entsprechenden Koordinaten gestellt und auf 180° ausgerichtet, so dass der Laser-Scanner auf das Tor zeigt.

Anschließend wurde mit einer Sampling-Rate von 20 Hz Messwerte des Laser-Scanners, der Odometrie und der omnidirektionalen Kamera aufgenommen. Zusätzlich wurden für die Positionsschätzung mittels omnidirektionalen Kamera die Länge der erkannten Torlinie und der Öffnungswinkel protokolliert. Sie wurden jedoch nicht weiter ausgewertet.

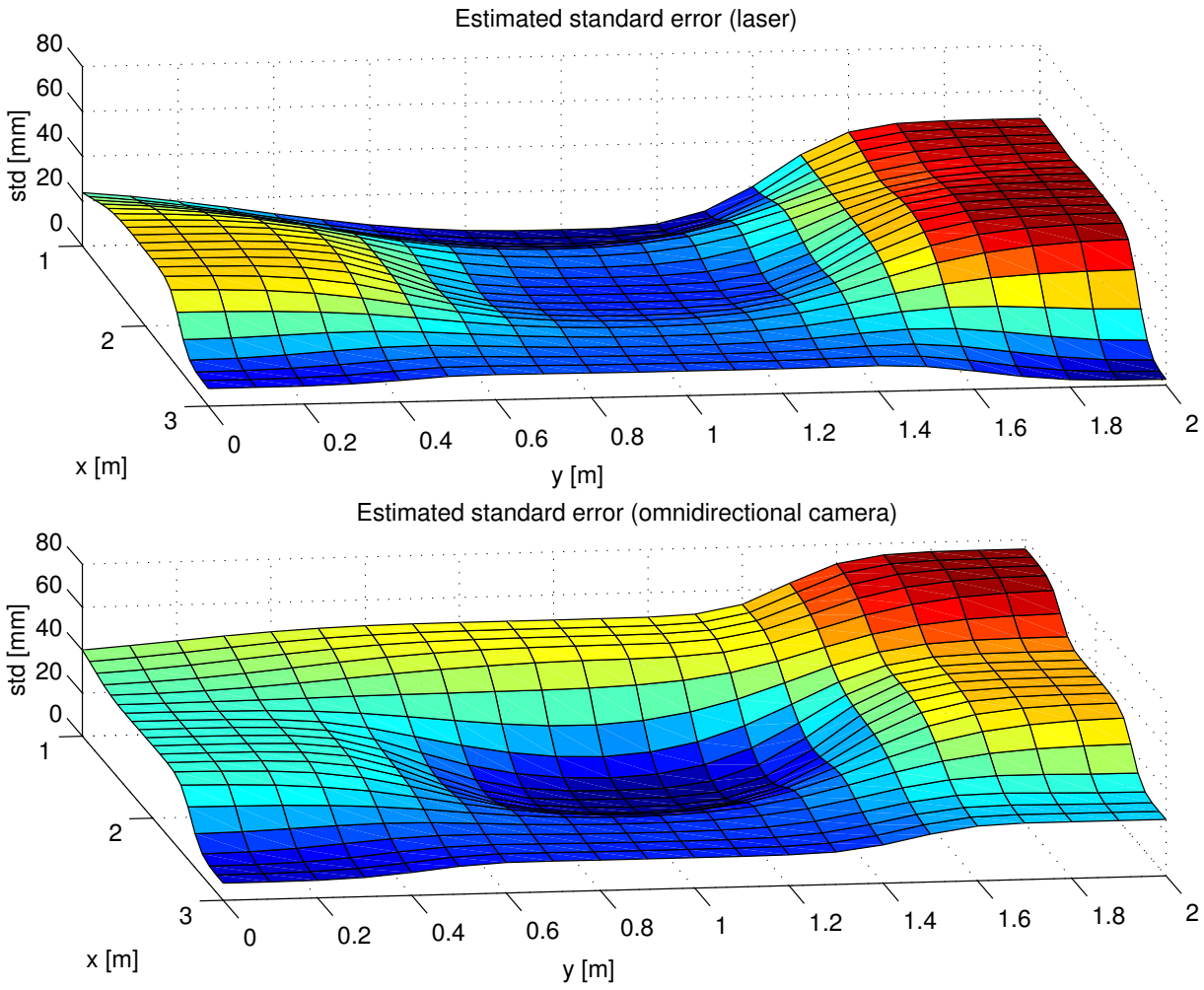
Zwischen den Messwerten wurde mittels quadratischer Polynome interpoliert. Zur Veranschaulichung wurde in der Abbildung der Mahalanobis-Abstand der interpolierten Kovarianzmatritzen verwendet:

3.2 Odometrie

Zur Bestimmung der Fehlerverteilung der Odometrie wurden die Daten einer Testfahrt verwendet. Für den Robocup ist es wichtig, ein robustes Fehlermodell zu verwenden, da Extremsituationen wie schnelles Beschleunigen (z.B. bei Kollisionen) recht häufig auftreten. Die Testfahrt wurde manuell gesteuert und enthält starke Beschleunigungen, schnelle Richtungsänderungen und eine Passage mit Rückwärtsfahren.

Von der Odometrie wurde diese Testfahrt als glatte Trajektorien gemessen (siehe Abbildung 1).

Die Positionsschätzung mittels Laser-Scanner war jedoch stark fehlerbehaftet. Deshalb wurden vor der weiteren Verwendung 10% Messwerte mit dem größten Mahalanobis-Abstand gefiltert (siehe Abbildung 2. Zusätzlich wurde die Daten zwischen der 20. und


Estimated standard error (laser)
std [mm]
80
60
40
20
0
x [m]
1
2
3
y [m]
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Estimated standard error (omnidirectional camera)
std [mm]
80
60
40
20
0
x [m]
1
2
3
y [m]
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

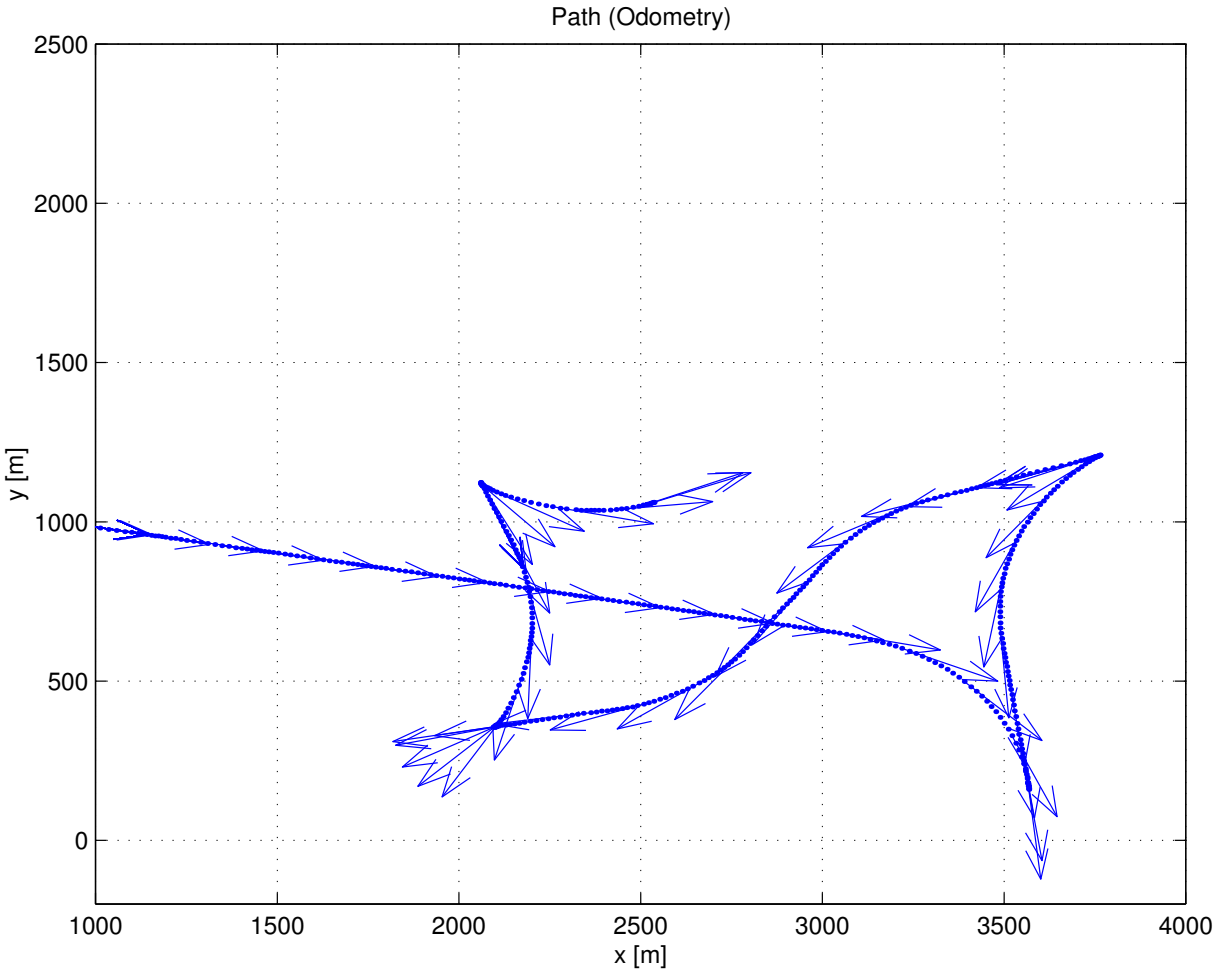


Abbildung 1: Pfad der Testfahrt (Odometrie)

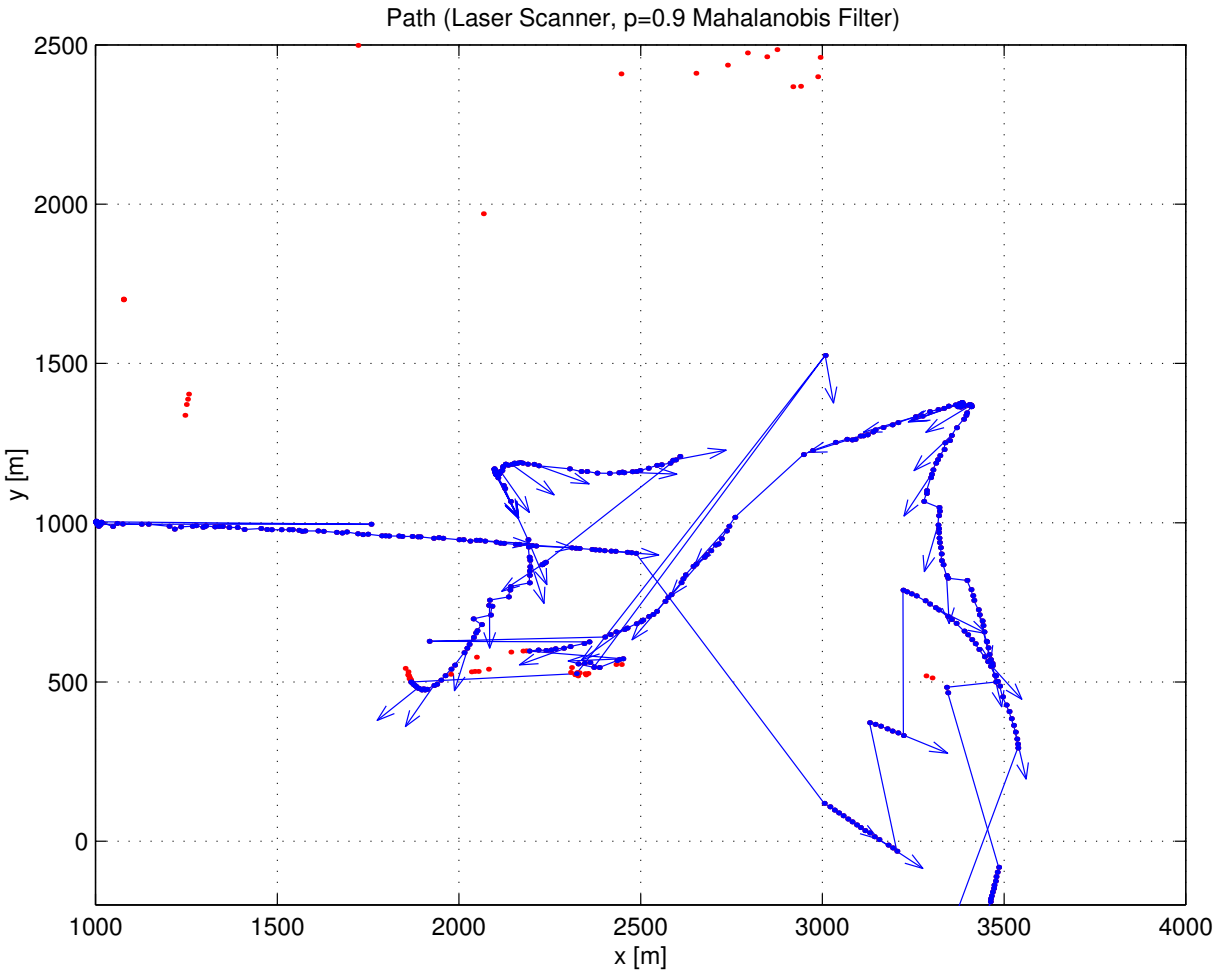


Abbildung 2: Pfad der Testfahrt (Laser-Positionsschätzung), 10% der Messwerte mit dem größten Mahalanobis-Abstand sind als einzelne Punkte dargestellt

25. Sekunde ausgeschlossen, das der Laser-Scanner in der unteren rechten Spielfeldecke z.T. falsche Daten lieferte (siehe Abbildung 3).

Für die verbleibenden Messwerte wurde jeweils die Differenz zwischen der Schätzung der Odometrie und des Laser-Scanners bestimmt:

$$\varepsilon = \varepsilon_{odo} + \varepsilon_{laser} = x_{odo} - x_{laser}$$

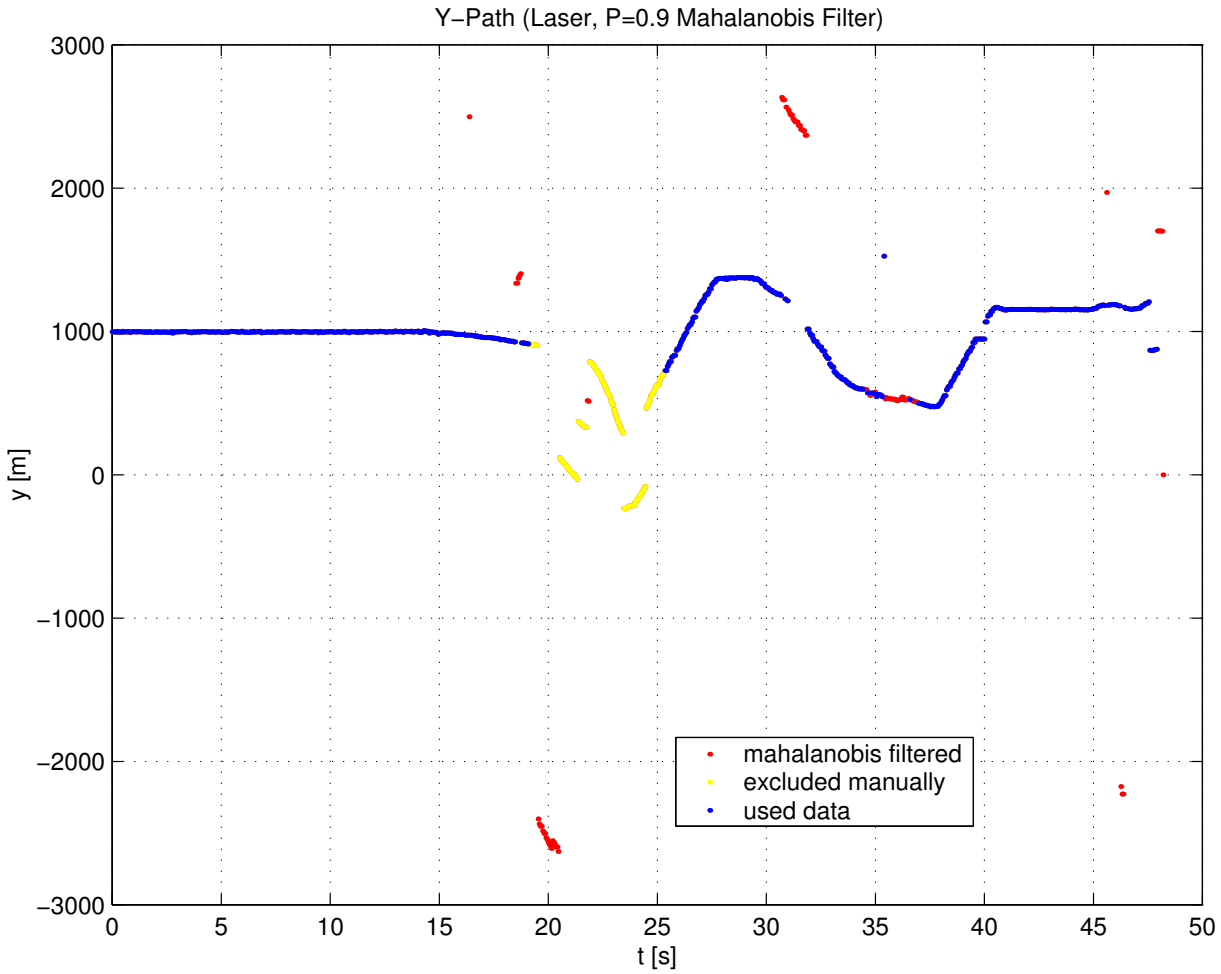


Abbildung 3: y-Koordinate der Testfahrt (Laser-Positionsschätzung), 10% der Messwerte mit dem größten Mahalanobis-Abstand sind als einzelne Punkte dargestellt

Dieser Fehler wurde nun in Abhängigkeit der Zeitdifferenz betrachtet:

$$\varepsilon|dt = x_{odo} - x_{laser}|\, t + dt$$

Da die zurückgelegt Wegstrecke dx wiederum von dt abhängt, kann somit der Fehler in Abhängigkeit von dx bestimmt werden. Für die Schätzung von $\varepsilon|dt$ wurden jeweils die 50% der Messwerte benutzt, die den kleinsten Mahalanobis-Abstand haben, d.h. die nahe dem Mittelwert sind.

Trotz des extremen Charakters der Testfahrt zeigt sich ein deutlicher linearer Zusammenhang zwischen der zurückgelegten Strecke $E(x_{laser}|t + dt - x_{laser}|t)$ und den bedingten Kovarianzen der Fehler $\Sigma|dt$ (siehe Abbildung 4).

Die Koeffizienten der linearen Funktion von dx auf Σ wurden mittels linearer Regression bestimmt.

4 Evaluation

4.1 Testszenarien

Stand Die Filter-Algorithmen wurden auf die Messreihen, die an den Gitterpunkten aufgenommen wurden, angewendet. Da die Kovarianz-Matritzen auch aus diesen Daten geschätzt wurden, überschätzen die Daten die Genauigkeit der Algorithmen.

Wahlweise seien in den Abbildungen 5-7 die Filterung der Zeitreihen für die Positionen (1,1) dargestellt. Man erkennt das beide Fitler innerhalb von 20 s (400 Iterationen) konvergieren. Da der Kalman-Filter den ersten Messwert als ersten Schätzer verwendet, der Partikel-Filter hingegen mit einer gleichförmigen Verteilung beginnt, ist die Differenz zwischen Schätzer und wahren Wert beim Kalman-Filter geringer. Allerdings stimmt die Konvergenzgeschwindigkeit, wie theoretisch zu erwarten, zwischen beiden Filtern überein.

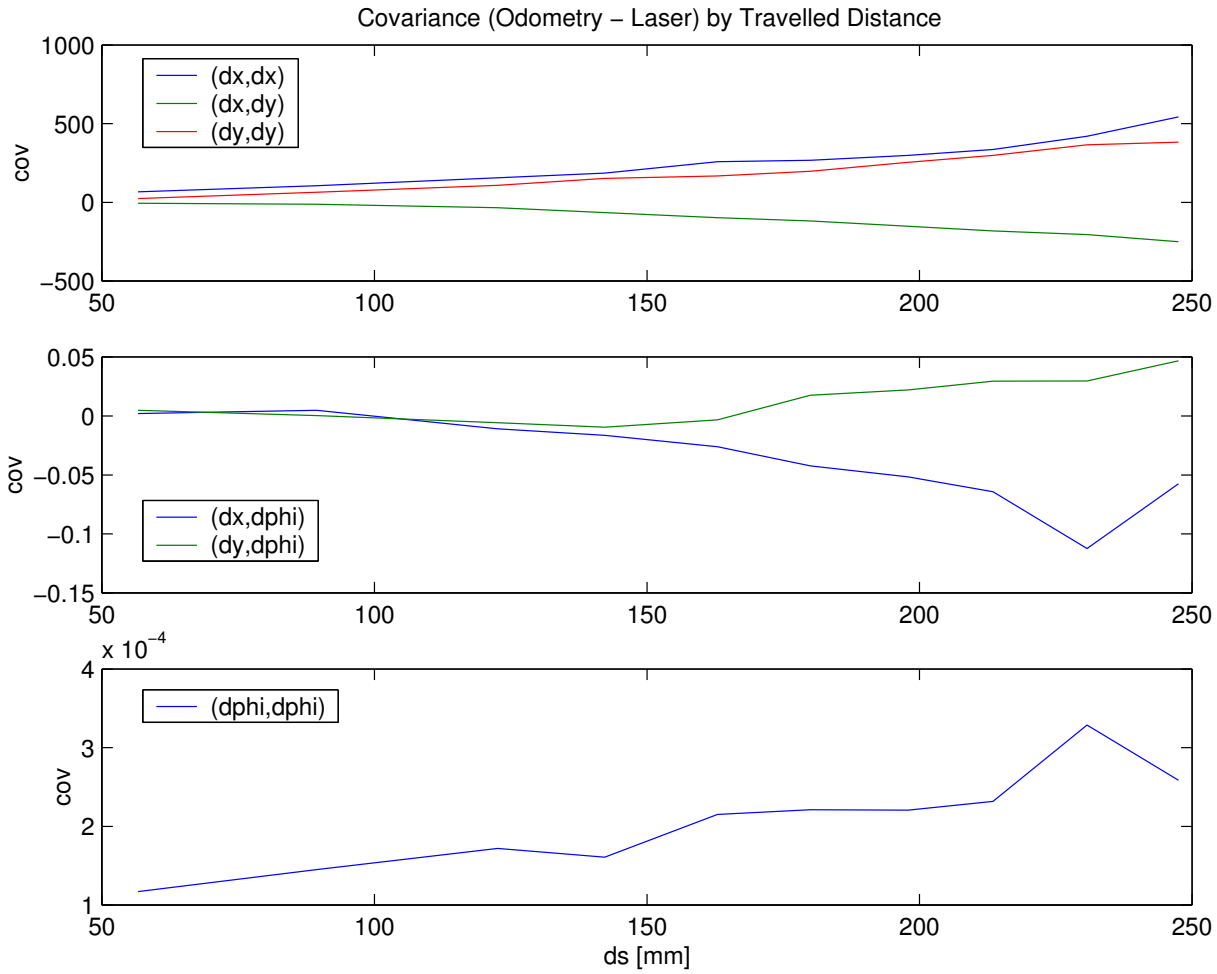


Abbildung 4: Abhängigkeit der Fehlerkovarianz von der zurückgelegten Wegstrecke

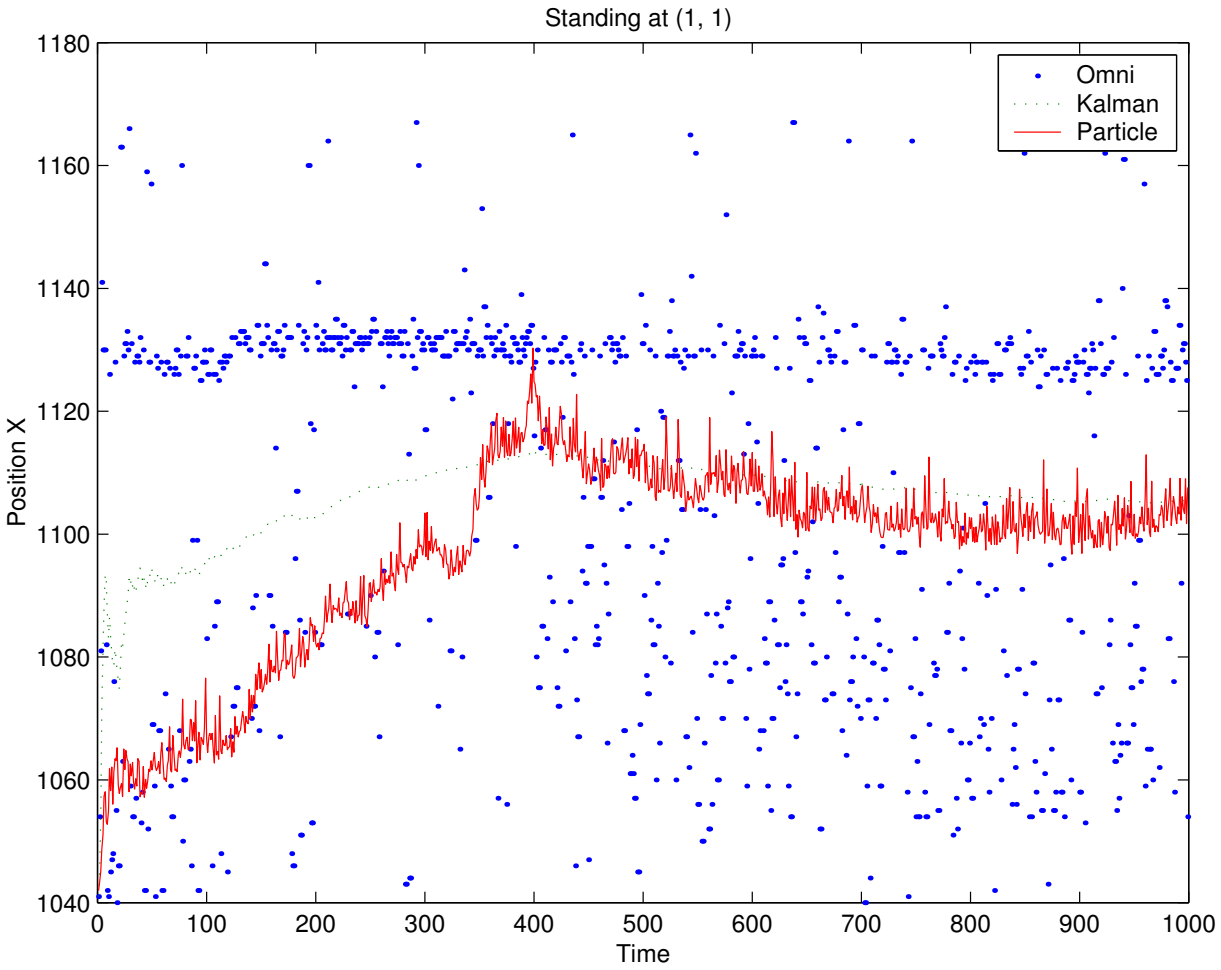


Abbildung 5: x-Positionsschätzung bei (1,1)

Deutlich sichtbar ist auch die Varianz des Partikel-Filters, die durch die begrenzte Anzahl von *Samples* bedingt ist.

Testfahrt Bei der Testfahrt zeigt sich, dass der Kalman-Filter nicht geeignet ist, große Diskrepanzen zwischen den Messwerten und der Schätzung zu integrieren (siehe Abbildung 8). Die Abweichungen betragen bis zu 2 Metern, wohingegen der größte Fehler beim Partikel-Filter ca. 50 cm beträgt.

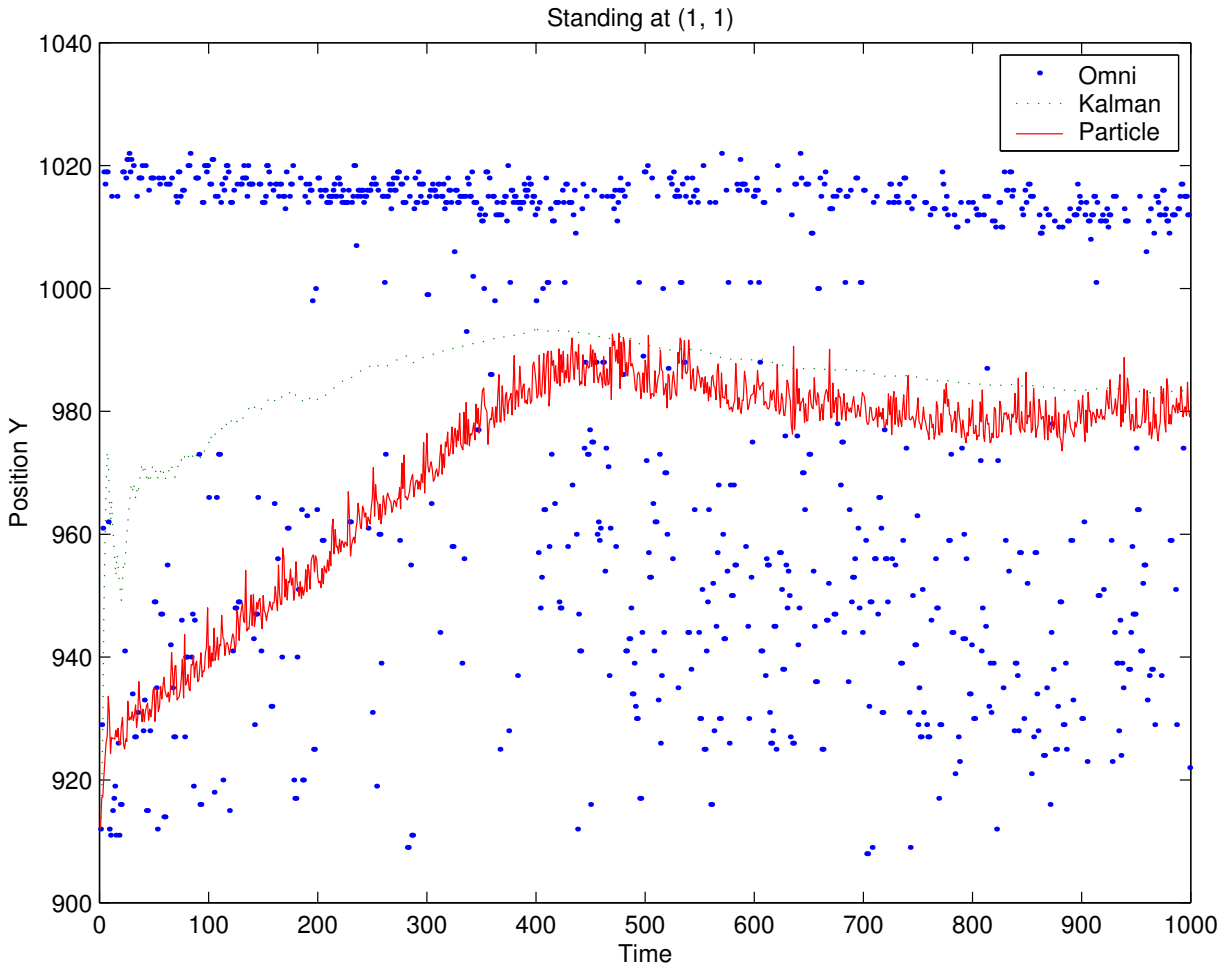


Abbildung 6: y-Positionsschätzung bei (1,1)

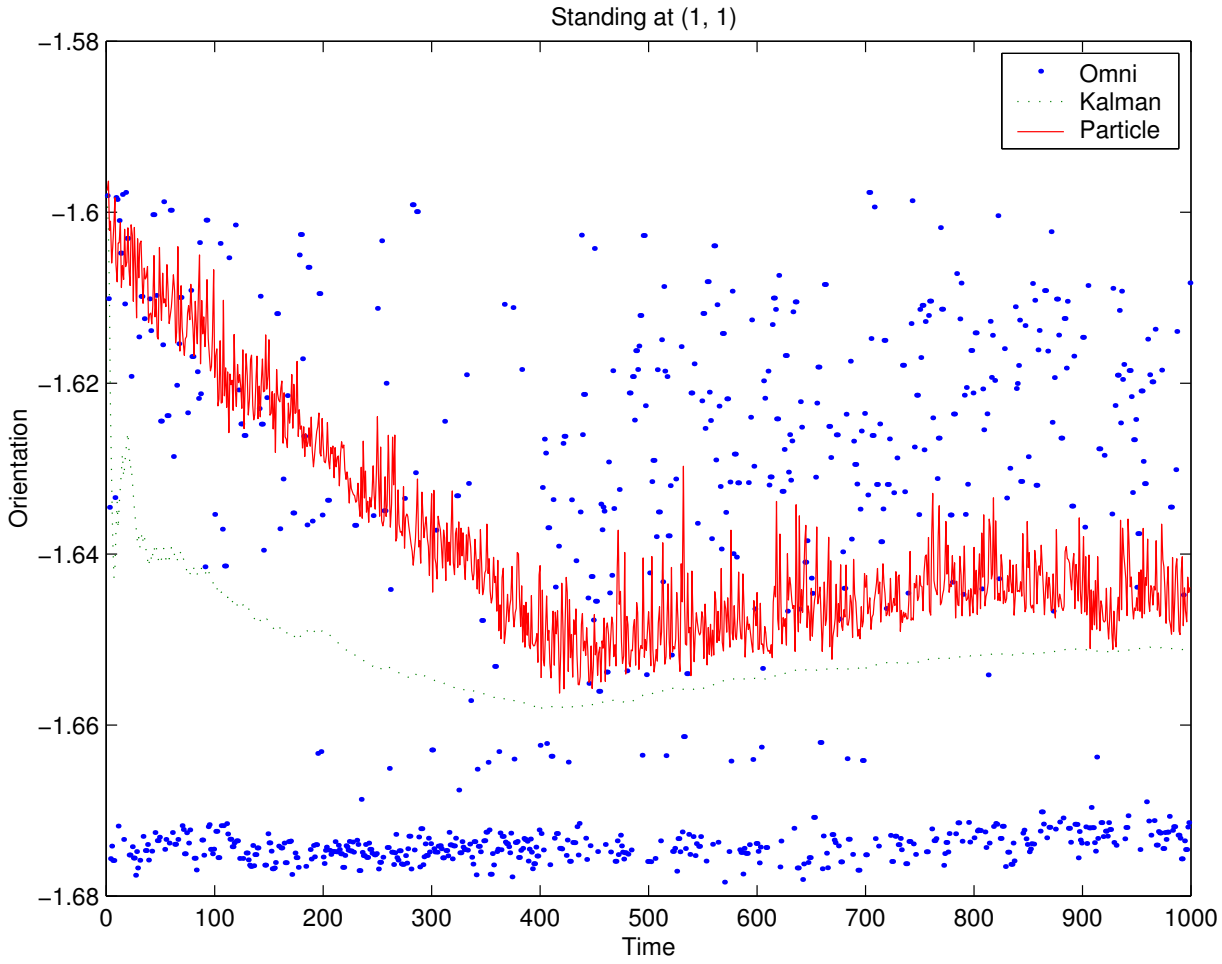


Abbildung 7: θ-Positionsschätzung bei (1,1)

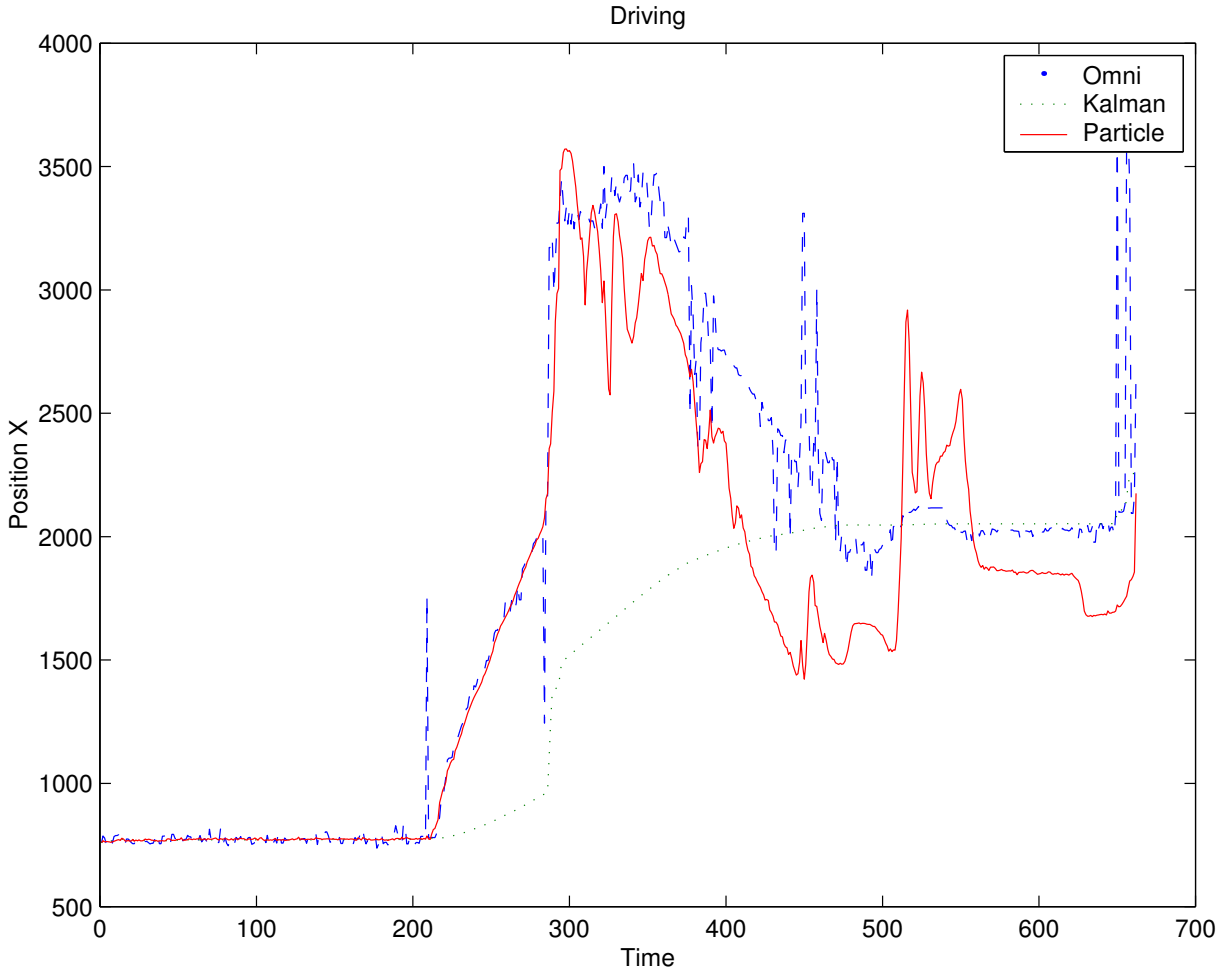


Abbildung 8: x-Positionsschätzung während der Testfahrt

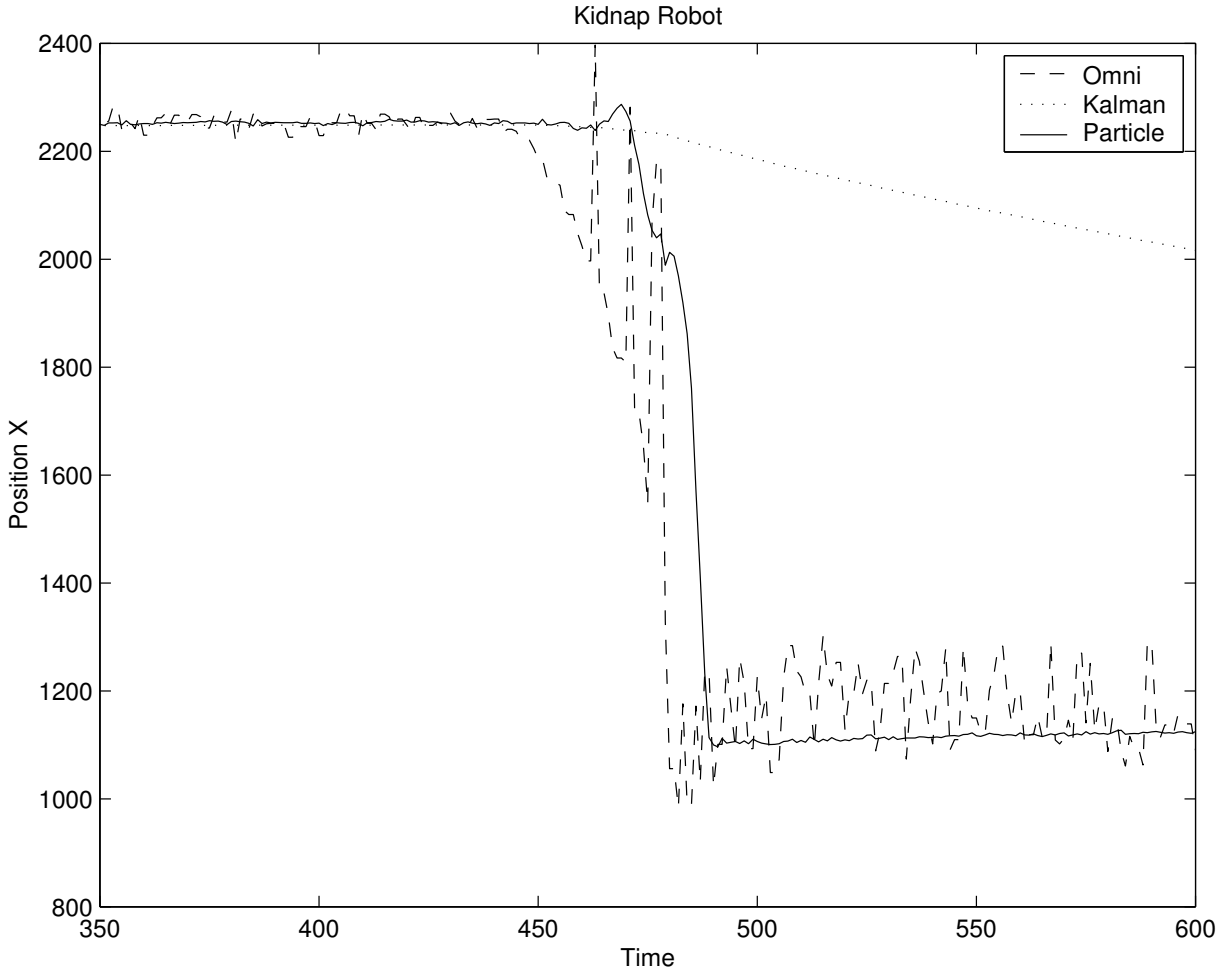


Abbildung 9: x-Positionsschätzung, Kidnap Robot

Kidnap Robot Gänzlich ungeeignet ist der Kalman-Filter, wenn die Propagation des Zustandes (z.B. durch Fehler der Odometrie-Daten) nicht möglich ist (siehe Abbildung 9).

Ein Vorteil der Partikel-Filter ist hierbei, dass die Varianz des Schätzfehlers $E[(\hat{X}-X)^2]$ proportional zur Varianz des Schätzers $E[\hat{X}^2]$ ist (6), siehe Abbildung 10. Für Handlungsplanungsalgorithmen wäre dieser Parameter sicherlich sehr von Nutzen, zumal er im vorliegenden Fall nur eine geringe Latenz zeigt.

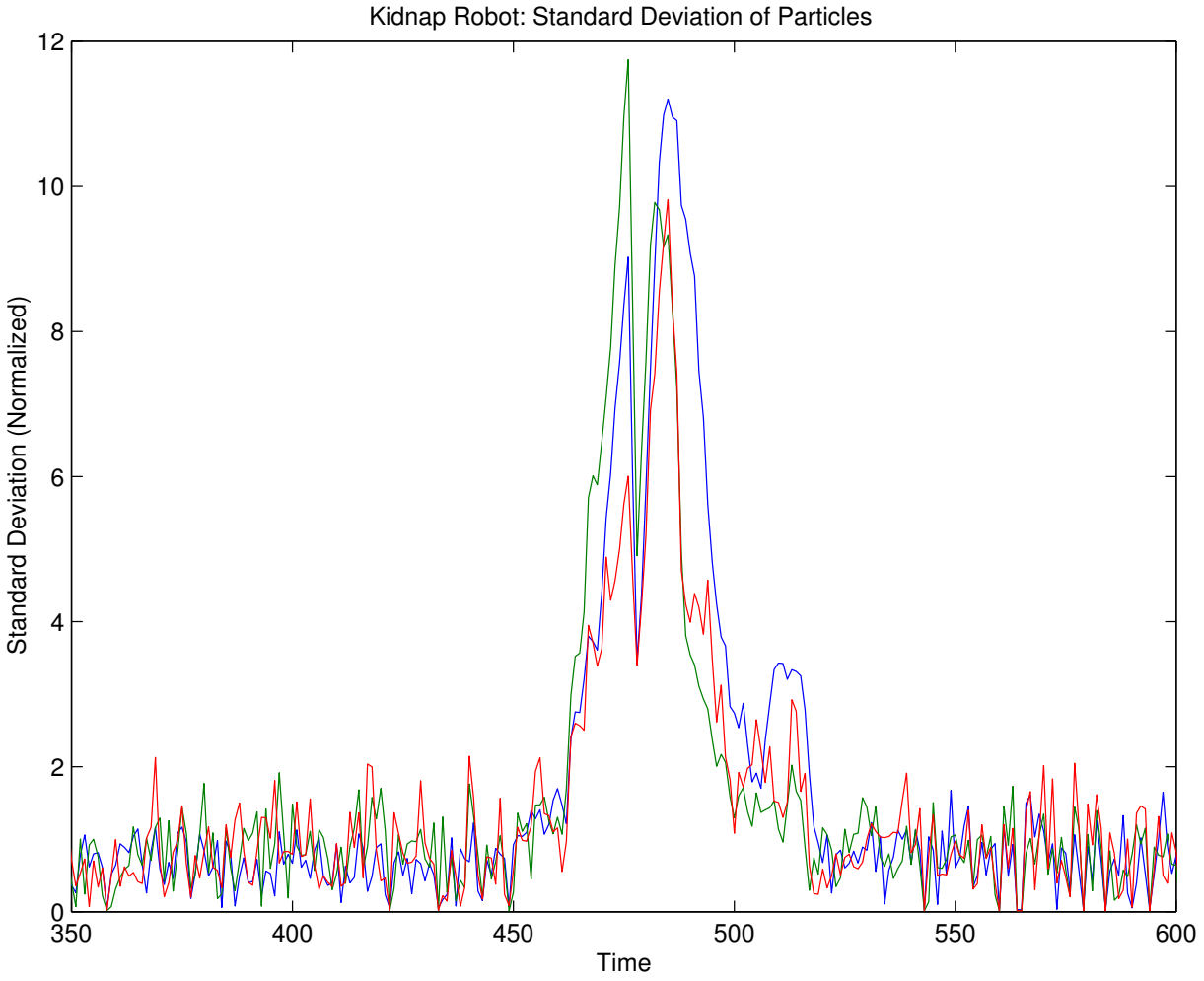


Abbildung 10: Standardabweichung für x, y, θ - Kidnap Robot

4.2 Vergleich der Methoden

Beide Verfahren, die Kalmanfilterung und die Partikelfilterung, weisen ein gutes Konvergenzverhalten unter mit dem Modell konformen Bedingungen auf. Dabei ist die Konvergenzgeschwindigkeit für beide Verfahren identisch, was aufgrund der gleichen Modellannahmen auch zu erwarten war. Allerdings tritt beim Partikelfilter ein "Eigenrauschen" auf, dass zum einen durch die Diskretisierung der Verteilung, vor allem aber durch die eingestreute uniforme Verteilung und das Resampling bedingt ist.

Deutlich Unterschiede zwischen beiden Verfahren zeigen sich dagegen bei Verletzung des Modells. Da beim Kalmanfilter die (geschätzte) Kovarianz der Pose unabhängig von den Messwerten ist, verringert sich mit jedem Messwert der Standardschätzfehler. Dies führt schon bei schnellem Fahren zu teils dramatischen Abweichungen von der realen Position um bis zu 2 Meter (Abbildung 8). Beim Partikelfilter erhöht sich dagegen schrittweise die (Ko)Varianz der Partikelverteilung, wenn abweichende Messwerte registriert werden. Allerdings nimmt damit auch die Varianz der Schätzung (Mittelwert der Partikelverteilung) zu, so dass Abweichungen bis zu einem Meter bei falscher Information (schnelle Fahrt, Abbildung 8) zu beobachten waren. Deutlich besser sind die Ergebnisse im Kidnap-Robot-Fall (Abbildung 9). Die Konvergenz des Partikelfilters kann als sehr gut beschrieben werden, wohingegen der Kalmanfilter keine brauchbare Schätzung liefert.

5 Fazit

In dieser Arbeit wurden Kalmanfilter und ein Partikelfilter für den Einsatz im RoboCup verglichen. Dabei wurde für beide Verfahren die gleichen linearen Modelle für den Fehler der Positionsschätzung mittels omnidirektionaler Kamera und für die Odometrie verwendet. Die Parameter für die Modelle wurden empirisch an verschiedenen Positionen auf dem Spielfeld bzw. anhand einer Testfahrt bestimmt. Die resultierenden Modelle wurden durch

lineare und binlineare Approximation erzeugt und spiegeln die gefundene Abhängigkeit der Fehlerverteilungen von der Position auf dem Spielfeld und von der Geschwindigkeit des Roboters wieder.

Das aufgrund der gleichen Modellannahmen zu erwartende ähnliche Konvergenzverhalten beiden Verfahren konnte bestätigt werden. Beim Kidnap-Robot-Problem (Veränderung der Position ohne Odometrie-Information) liefert das Partikelverfahren sehr verlässliche Schätzungen, während das Kalmanfilter nur unzureichend konvergiert. Für den praxisrelevanten Fall von häufig falscher Information (sehr schnelle Testfahrt) ist der Partikelfilter sehr viel besser geeignet, als dies das Kalmanverfahren ist.

Die höhere lokale Varianz des Partikel-Schätzers ("Eigenrauschen") stellt dagegen einen entscheidenden Nachteil des Verfahrens dar, da in der aktuellen Implementierung der RoboCup-Software die Verfahren für die Regelung des Roboterpfades und der Geschwindigkeit einen stetigen Verlauf der Schätzfunktion erfordern.

Literatur

[1] J. Bortz. *Statistik für Sozialwissenschaftler.* Springer, third edition, 1989.

[2] K. Chung. *Markov Chains with Stationary Transition Probabilities.* Springer, 1960.

[3] R.M. du Plessis. *Poor Man's Explanation of Kalman Filtering.* Taygeta Scientific Incorporated: Monterey CA, 1967.

[4] The RoboCup federation. Robocup-2001. middle-size robot league. rules and regulations.

[5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, Orlando, Florida, 1999.

[6] W. Gellert, H. Küstner, M. Hellwich, and H. Kästern, editors. *Kleine Enzyklopädie Mathematik.* VEB Bibliographisches Institut Leipzig, second edition, 1967.

[7] J.-S. Gutman. Vergleich von algorithmen zur selbstlokalisation eines mobilen roboters. Master's thesis, Universität Ulm, Fakultät für Informatik, 1996.

[8] J.A. Hartigan. *Bayes Theory.* Springer, first edition, 1983.

[9] R.E. Kalman. A new approach to linear filtering and prediction problems. In *Trans. ASME, Series D*, number 82 in J. Basic Eng., 1960.

[10] J. Liu. Metropolized indpendent sampling with comparision to rejection sampling importance sampling. *Statistics and Computing*, 6:113–119, 1996.

[11] P. Maybeck. *Stoachstic Models, Estimation and Control*, volume 1. Academic Press, 1979.

[12] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. Technical report, Nuffield College, University of Oxford, [http://www.nuff.ox.ac.uk/users/shephard/], 1997.

[13] D.B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the sir algorithm. discussion of tanner and wong (1987). *J. Am. Statist. Assoc.*, 23:543–546, 1987.

[14] D.B. Rubin. Using the sir algorithm to simulation posterior distributions. In J.M. Bernado, M.H. DeGroot, D.V. Lindley, and A.F.M. Smith, editors, *Bayesian Statistics 3*, pages 395–402. Oxford University Press, 1988.