

Ozan Özışık, Robot Takımı ile Eş Zamanlı Konum Belirleme ve Haritalama, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 2010.

Sebastian Thrun, Wolfram Burgard, Dieter Fox, Probabilistic Robotics, The MIT Press, 2005.

Genişletilmiş Kalman Filtresi Genel Algoritması

GKF_Algoritması($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, N_{t-1}$)

1: $N_t = N_{t-1}$

2: $F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 \end{pmatrix}$

3: $\bar{\mu}_t = \mu_{t-1} + F_x^T Mtr_{kinematik}$

4: $G_t = I + F_x^T g_t F_x$

5: $R_t = V_t M_t V_t^T$

6: $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t F_x$

7: $\bar{\mu}_t$ ve $\bar{\Sigma}_t$ matrislerinin boyutunu eklenecek yeni yer işaretçisi için büyüt

8: Döngü_1: gözlenen tüm z_t^i yer işaretçileri için yap

8.1: yer işaretçisinin mutlak (küresel koordinat düzlemine göre) koordinatını bul.

8.2: yer işaretçisini $\bar{\mu}_t$ matrisine ekle, $\bar{\Sigma}_t$ matrisinde de bu yer işaretçisi için ilklendirme yap.

8.3: Döngü_2: $k=1$ 'den $(N_t + 1)$ 'e $\bar{\mu}_t$ matrisindeki tüm yer işaretçileri için yap

8.3.1: robotun k . yer işaretçisini görüyor olsa nasıl bir ölçüm değeri elde edeceğine dair hesaplanan değerlerden oluşan beklenen ölçüm matrisini hesapla (\hat{z}_t^k).

$$8.3.2: F_{x,k} = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 1 & 0 \dots 0 \end{pmatrix}$$

$$8.3.3: H_t^k = h_t F_{x,k}$$

$$8.3.4: \psi_k = H_t^k \bar{\Sigma}_t [H_t^k]^T + Q_t$$

$$8.3.5: \pi_k = (z_t^i - \hat{z}_t^k)^T \psi_k^{-1} (z_t^i - \hat{z}_t^k)$$

$$8.4: \pi_{N_t+1} = \alpha$$

$$8.5: j(i) = \arg \min_k \pi_k$$

8.6: Koşullu Kontrol:

8.6.1: Eğer $j(i) > N_t$ ise

$$8.6.1.1: N_t = N_t + 1$$

8.6.1.2: $\bar{\mu}_t$ ve $\bar{\Sigma}_t$ matrislerinin boyutunu eklenecek yeni yer işaretçisi için

büyüt

8.6.2: Eğer $j(i) > N_t$ değilse

$$8.6.2.1: K_t^i = \bar{\Sigma}_t [H_t^{j(i)}]^T \psi_{j(i)}^{-1}$$

$$8.6.2.2: \bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^{j(i)})$$

$$8.6.2.3: \bar{\Sigma}_t = (I - K_t^i H_t^{j(i)}) \bar{\Sigma}_t$$

$$9: \mu_t = \bar{\mu}_t$$

$$10: \Sigma_t = \bar{\Sigma}_t$$

11: μ_t ile Σ_t 'yi geri döndür.

Açıklamalar:

Genişletilmiş Kalman Filtresi (GKF), Kalman Filtresi'nin doğrusal olmayan fonksiyonlarda da kullanılmasını sağlayan bir metottur, robotun hareket ve algı fonksiyonları doğrusal olmadığından robotik problemlerinde de kullanılır.

GKF'de doğrusal olmayan fonksiyonları doğrusallaştırmak için birinci dereceden Taylor açılımı kullanılır. Aşağıda Taylor açılımının genel hali görülmektedir:

$$f(x+a) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots$$

Burada kullanılan a , komşuluğu dahilinde f fonksiyonunun sonsuz sayıda türevinin alınabildiği reel ya da karmaşık bir sayıdır. GKF'de fonksiyon parametresi olarak en olası değer olan μ_{t-1} değeri kullanılır.

Robotun hareket fonksiyonu olan $g(u_t, x_{t-1})$ fonksiyonunu ele alırsak:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1})$$

$$g(u_t, x_{t-1}) = g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1})$$

olur. Aynı işlemi algı fonksiyonu için uygularsak:

$$h(x_t) \approx h(\bar{\mu}_t) + h'(\bar{\mu}_t)(x_t - \bar{\mu}_t)$$

$$h(x_t) = h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t)$$

olur.

Burada yer verilen Genişletilmiş Kalman Filtresi algoritmasında adımlar mümkün olduğunca matematiksel olarak ifade edilmeye çalışılmıştır fakat bazı hesaplar tamamen kullanılan algılayıcı, efektör vs. (ve bunların nasıl modellendiği) ile ilgili olduğu için bu hesaplar sözel olarak ifade edilmiştir.

Algoritmada giriş olarak μ_{t-1} , Σ_{t-1} , u_t , z_t ve N_{t-1} değerleri alınmaktadır. μ_{t-1} , robotların ve yer işaretçilerinin konum bilgilerini ifade eden Gauss olasılık dağılımlarının ortalama değerini tutan sütun matristir. Üst kısımda robotların (x, y, θ) bilgileri tutulur, daha sonra ise yer işaretçilerinin bilgileri gelir. Σ_{t-1} , robotların ve yer işaretçilerinin konum bilgilerini ifade eden Gauss olasılık dağılımlarının kovaryanslarının tutulduğu matristir. u_t , aslında robota

gönderilen kontrol işaretidir fakat pek çok uygulamada robota gönderilen komutlar değil, robottan dönen odometri verisi kullanılır. z_t , robotun yaptığı gözlemlerdir. N_{t-1} ise bir önceki adımda elde tutulan yer işaretçisi sayısıdır.

Bu algorithma 1. adım ile 6. adım arası hareket güncellemesidir. Burada ortalama değerler matrisi (μ_t) ile kovaryans matrisi (Σ_t) üzerinde kinematik denklemler ve hareket belirsizlikleri kullanılarak güncellemeler yapılmaktadır. Bu matrislerin hareket güncellemesi yapılmış halleri üzerleri çizili olarak ifade edilmektedir ($\bar{\mu}_t, \bar{\Sigma}_t$). 7. ve 8. adımlarda algı güncellemesi yapılmaktadır. Bu adımlarda z_t gözlemleri ile daha önceden görülmüş ve durum uzayına eklenmiş yer işaretçileri karşılaştırılmakta, eşleşme bulunduğu yer işaretçisinin görülmesi beklenen yer ile görüldüğü yer arasındaki fark kullanılarak durum güncellenmekte, eşleşme bulunamazsa bu yer işaretçisi yeni bir yer işaretçisi olarak mevcut duruma eklenmektedir.

Algorithma her adımda yapılan işlemler aşağıda anlatılmaktadır:

- 1: Bir önceki adımdaki yer işaretçisi sayısı şimdiki yer işaretçisi sayısı olarak alınmaktadır.
- 2: F_x matrisi tanımlanmaktadır. Bu matris, $Mtr_{kinematik}$ matrisini μ_{t-1} matrisi ile toplanabilecek şekilde uygun boyutlu hale getiren ve çarpma işlemi sonucunda μ_{t-1} 'de yalnızca ilgili yerlerin (ele alınan robotun konum bilgilerinin) değişmesini sağlayan matristir. Boyutları $3 \times (3 \times RobotSayisi + 2 \times N_t)$ şeklindedir. Burada $Mtr_{kinematik}$ matrisi, ilgili robotun konum parametrelerinin her birinde hareket sonucu gerçekleştiği kinematik denklemler kullanılarak hesaplanan değişimdir.
- 3: μ_{t-1} matrisine hareket sonucu robotun konumunda gerçekleşen değişimler eklenmekte ve $\bar{\mu}_t$ matrisi elde edilmektedir.
- 4: Robotun hareketi sonucu kovaryans matrisinde (Σ_{t-1}) gerçekleşen dönüşümü (*transformasyon*) hesaplamak için kullanılan G_t matrisi hesaplanmaktadır. Burada G_t , daha önce de ifade edildiği gibi $g(u_t, \mu_{t-1})$ fonksiyonunun türevidir.

$$g(u_t, \mu_{t-1}) = \mu_{t-1} + Mtr_{kinematik}$$

olduğu için bunun μ_{t-1} 'e (yani x , y ve θ parametrelerine) göre türevi

$$g'(u_t, \mu_{t-1}) = I + Mtr'_{kinematik}$$

şeklinde olmaktadır. $Mtr'_{kinematik}$ de burada g_t ile ifade edilmektedir. F_x yine matrislerin tüm durum matrisi boyutuna uygun olması için kullanılmaktadır. I ise boyutu kullanıldığı yerdeki diğer matrislerle uyumlu olacak şekilde ayarlanan birim matristir.

5: Hareket sonucu durum uzayında oluşan belirsizliği ifade eden R_t matrisi hesaplanmaktadır. Burada M_t , kontrol uzayındaki hareket gürültüsünü ifade eden kovaryans matrisidir. Bu gürültünün durum uzayına aktarımı doğrusal bir yaklaşımla yapılır. Bu yaklaşım için gereken jacobian matrisi olan V_t , hareket fonksiyonunun u_t ve μ_{t-1} 'de hareket parametrelerine göre türevidir.

6: Duruma ait kovaryans matrisinde hareket güncellemesi yapılmaktadır. Önceki adımdan gelen kovaryans matrisinde dönüşüm uygulanmakta, buna hareket gürültüsü eklenmektedir.

7: Bu algorithmada gözlemler elde tutulan yer işaretçileri ile kıyaslanırken öncelikle mutlaka yeni bir yer işaretçisi gibi durum matrisine eklenmektedir, gözlem ile (aslında zaten gözlemin kendisi olan) yer işaretçisi arasındaki mesafe olarak ise belli bir eşik değeri atanmaktadır. Eğer gözleme daha yakın bir yer işaretçisi bulunursa gözlemle bu yer işaretçisi eşleşmiş olur, güncellemeler yapılır, son eklenen yer işaretçisini dışarıda bırakacak şekilde matris boyutları küçültülür. Eğer gözleme en yakın yer işaretçisi son eklenen yer işaretçisi olursa yeni bir yer işaretçisi görülmüş demektir, bu yer işaretçisi korunur. Bu adımda gerçekleştirilen $\bar{\mu}_t$ ve $\bar{\Sigma}_t$ matrislerinin büyütülmesi işte bu yer işaretçisinin yerleştirilmesi içindir.

8: Döngü başlamaktadır, tüm gözlemler için kıyaslama yapılacaktır.

8.1: Robot tarafından yapılan gözlemler robotun yerel koordinat düzlemine göre dir. Bu adımda gözlenen yer işaretçisinin küresel koordinat düzlemine göre koordinatı bulunmaktadır.

8.2: Yer işaretçisi $\bar{\mu}_t$ matrisine eklenir. $\bar{\Sigma}_t$ kovaryans matrisinde de bu yer işaretçisi için belirsizlik ilklendirilir.

8.3: Döngü başlamaktadır, elde tutulan tüm yer işaretçilerine bakılacaktır.

8.3.1: Robotun, k. yer işaretçisini görüyor olsa nasıl bir ölçüm değeri elde edeceğine dair hesaplanan değerlerden oluşan beklenen ölçüm matrisini hesapla (\hat{z}_t^k).

8.3.2: Matris çarpımında boyutların tutmasını ve yalnızca ilgili robot konumu ve yer işaretçisinde değişiklik yapılmasını sağlayan $F_{x,k}$ matrisi tanımlanmaktadır.

8.3.3: Durum uzayındaki belirsizliğin ölçüm uzayındaki karşılığını hesaplamada kullanılan H_t^k matrisi hesaplanmaktadır. H_t^k matrisi, yer işaretçilerinin mutlak ve göreceli konumları arasında dönüşüm yapan fonksiyonun tüm durum vektörüne göre kısmi türevidir. Bu fonksiyon yalnızca ilgili robot ile yer işaretçisine bağlı olduğundan türev, h_t jacobian'ı ile h_t jacobian'ının durum uzayı boyutlarında matrise aktarılmasını sağlayan $F_{x,k}$ matrisinin çarpımına dönüşür. Burada h_t , yer işaretçilerinin mutlak ve göreceli konumları arasında dönüşüm yapan fonksiyonun robot konumu (x, y, θ) ve mutlak yer işaretçisi konumu parametrelerine (r, θ) göre kısmi türevidir.

8.3.4: Gözlem ile elde tutulan yer işaretçisi karşılaştırılırken kullanılacak toplam belirsizlik hesaplanmaktadır. Burada Q_t ölçüm gürültüsüdür.

8.3.5: z_t^i gözlemi ile \hat{z}_t^k beklenen ölçümü arasındaki Mahalanobis mesafesi hesaplanmaktadır.

8.4: Gözlem ile aslında zaten gözlemin kendisi olan yeni eklenen yer işaretçisi arasındaki mesafe olarak bir eşik değer kabul ediliyor.

8.5: Gözleme en yakın yer işaretçisi bulunuyor.

8.6: Koşullu kontrol

8.6.1: Eğer en yakın yer işaretçisi yeni eklenen yer işaretçisi ise

8.6.1.1: Yer işaretçisi sayısı bir artırılıyor.

8.6.1.2: $\bar{\mu}_t$ ve $\bar{\Sigma}_t$ matrislerinin boyutları bundan sonra eklenecek yer işaretçisi için büyütülür.

8.6.2: Eğer en yakın yer işaretçisi daha önce eklenen yer işaretçilerinden biri ise

8.6.2.1: Yeni gözlemden elde edilen bilginin durum üzerinde ne kadar etkili olacağını belirleyen Kalman kazanımı hesaplanır.

8.6.2.2: Kalman kazanımı ve yer işaretçisinin beklenen yeri ile gözlenen yeri arasındaki fark kullanılarak durum ortalama değer matrisinde güncelleme yapılır.

8.6.2.3: Kalman kazanımı kullanılarak durum kovaryans matrisinde güncelleme yapılır, belirsizlikler azalır.

9: $\bar{\mu}_t$ matrisi μ_t matrisine atanır.

10: $\bar{\Sigma}_t$ matrisi Σ_t matrisine atanır.

11: μ_t ve Σ_t matrisleri geri döndürülür.

Ozan Özışık, Robot Takımı ile Eş Zamanlı Konum Belirleme ve Haritalama, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 2010.

Genişletilmiş Kalman Filtresi Algoritması Uygulaması

Burada GKF'nin bu çalışmadaki uygulama detayları verilmiştir.

1. Giriş:

Uygulama ortalama değer matrisinin r robot sayısı olmak üzere $(r \times 3) \times 1$ şeklinde boyutlandırılması ve robotların ilk konumları ile ilklendirilmesi ile başlamaktadır. Kovaryans matrisi de $(r \times 3) \times (r \times 3)$ şeklinde boyutlandırılmaktadır ve robotların ilk konumlarını kesin olarak bildiğimiz için 0 ile ilklendirilmektedir.

$$\mu_0 = (x_1 \quad y_1 \quad \theta_1 \quad x_2 \quad y_2 \quad \theta_2 \quad \dots \quad x_r \quad y_r \quad \theta_r)^T$$

$$\Sigma_0 = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

Algılayıcı konumları da robot konumlarına göre her adımda yeniden hesaplanıp tutulmaktadır.

Bu çalışmada merkezi kontrol mekanizması kullanılmıştır, tüm robotlar 6 kızılötesi algılayıcılarından aldıkları verileri ve iki tekerlerinden ayrı ayrı aldıkları odometri verilerini merkez bilgisayara göndermektedirler.

u_t kontrol işareti olarak robotların gönderdiği odometri verisi kullanılmaktadır.

GKF'ye algılayıcı verileri doğrudan gönderilmektedir, bu verilerden doğrusal özelliklerin çıkarımı, yani z_t gözlemlerinin elde edilmesi GKF içinden çağrılan bir metot ile halledilmektedir. GKF'nin bir gözlem olsun ya da olmasın her durumda çağrılmasının nedeni hareket güncelleme işleminin de GKF içinde yer almasıdır. Ayrıca Hough Dönüşümü uygulanacak noktasal verilerin her adımda elde edilmesi, kümeye eklenmesi gerekmektedir. GKF her robot için ayrı ayrı çalıştırılır.

2. Hareket Güncellemesi

Hareket güncellemesinde ortalama değer matrisi odometri verileri ve kinematik denklemler kullanılarak güncelleştirilmekte, kovaryans matrisi de dönüşüm matrisi ve hareket gürültüsü ile güncellenmektedir.

2.1 $Mtr_{kinematik}$ matrisi:

Uygulamada kullanılan kinematik denklemler robotun düz ya da eğimli gitmesine göre değişmektedir. Robotun sağ ve sol tekerlerden aldığı odometri verileri eşitse yani robot düz gidiyorsa $Mtr_{kinematik}$ matrisi şu değerleri almaktadır:

$$Mtr_{kinematik} = \begin{pmatrix} odometri[sol] \cos(\theta_{t-1}) \\ odometri[sol] \sin(\theta_{t-1}) \\ 0 \end{pmatrix}$$

Robotun eğimli gittiği durumda $Mtr_{kinematik}$ matrisi şöyle hesaplanmaktadır:

$$alfa = \frac{odometri[sag] - odometri[sol]}{aksUzunlugu}$$

$$donusYaricapi = \frac{odometri[sag]}{alfa}$$

$$dd = 2 \times \left(donusYaricapi - \frac{aksUzunlugu}{2} \right) \times \sin\left(\frac{alfa}{2}\right)$$

$$Mtr_{kinematik} = \begin{pmatrix} dd \times \cos(\theta_{t-1} + alfa / 2) \\ dd \times \sin(\theta_{t-1} + alfa / 2) \\ alfa \end{pmatrix}$$

Burada şu belirtilmelidir, tüm uygulama boyunca açılarla ilgili yapılan her işlemde sonra açılar $0 - 2\pi$ aralığında kalması için gerekli işlem yapılmaktadır. Burada da alfa değeri hesaplandıktan sonra bu değer $0 - 2\pi$ aralığında olup olmadığı kontrol edilmiş, değilse bu aralığa çekilmiştir.

2.2 g_t matrisi:

$Mtr_{kinematik}$ matrisinin (x, y, θ) robot konumu parametrelerine göre kısmi türevi olan g_t matrisi robot düz gidiyorsa

$$g_t = \begin{pmatrix} 0 & 0 & odometri[sol] \sin(\theta_{t-1}) \\ 0 & 0 & -odometri[sol] \cos(\theta_{t-1}) \\ 0 & 0 & 0 \end{pmatrix},$$

eğimli gidiyorsa

$$g_t = \begin{pmatrix} 0 & 0 & dd \times \sin(\theta_{t-1} + alfa / 2) \\ 0 & 0 & -dd \times \cos(\theta_{t-1} + alfa / 2) \\ 0 & 0 & 0 \end{pmatrix}$$

değerlerini almaktadır. Bu matrislerde sütunlar $Mtr_{kinematik}$ matrisinin ilgili satırının sırasıyla x 'e, y 'ye ve θ 'ya göre kısmi türevini tutmaktadır.

2.3 V_t matrisi:

Hareket uzayındaki gürültünün durum uzayına taşınmasını sağlayan ve $Mtr_{kinematik}$ matrisinin hareket parametrelerine yani sağ ve sol teker için odometri değerlerine göre kısmi türevi olan V_t matrisinin değerleri de robotun düz ya da eğimli gitmesine göre kinematik denklemler değiştiğinden her iki durumda farklı şekillerde hesaplanmaktadır.

Robot düz gidiyorsa

$$V_t = \begin{pmatrix} \cos(\theta_{t-1}) & \cos(\theta_{t-1}) \\ \sin(\theta_{t-1}) & \sin(\theta_{t-1}) \\ 0 & 0 \end{pmatrix}$$

olmaktadır. Robotun eğimli gittiği durumda kinematik denklemler sağ ve sol odometri değerlerine bağlı pek çok değişken içerdiğinden bu hesaplar bira daha karmaşıktır, eşitlikler parçalara ayrılarak türev alınmıştır, türev alma adımları ayrıntılı şekilde aşağıda verilmiştir.

$$Mtr_{kinematik}(0,0) = dd \times \cos\left(\theta_{t-1} + \frac{alfa}{2}\right)$$

$$Mtr_{kinematik}(1,0) = dd \times \sin\left(\theta_{t-1} + \frac{alfa}{2}\right)$$

$$Mtr_{kinematik}(2,0) = alfa$$

$$f = \frac{aksUzunlugu \times (odometri[sag] + odometri[sol])}{odometri[sag] - odometri[sol]}$$

$$g = \sin\left(\frac{odometri[sag] - odometri[sol]}{2 \times aksUzunlugu}\right)$$

$$h = \cos\left(\theta_{t-1} + \frac{odometri[sag] - odometri[sol]}{2 \times aksUzunlugu}\right)$$

$$j = \sin\left(\theta_{t-1} + \frac{odometri[sag] - odometri[sol]}{2 \times aksUzunlugu}\right)$$

$$r = \frac{odometri[sag] - odometri[sol]}{aksUzunlugu}$$

olmak üzere

$$Mtr_{kinematik}(0,0) = f \times g \times h$$

$$Mtr_{kinematik}(1,0) = f \times g \times j$$

$$Mtr_{kinematik}(2,0) = r$$

olur.

Eğer

$$e = odometri[sag] - odometri[sol]$$

$$k = odometri[sag] + odometri[sol]$$

denirse f , g , h , j ve r değerleri de şöyle ifade edilebilir:

$$f = \frac{aksUzunlugu \times k}{e}$$

$$g = \sin\left(\frac{e}{2 \times aksUzunlugu}\right)$$

$$h = \cos\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$j = \sin\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$r = \frac{e}{aksUzunlugu}$$

Tüm değerlerin kısmi türevleri aşağıdaki gibidir:

$$eTur0 = \frac{\partial e}{\partial odometri[sol]} = -1$$

$$eTur1 = \frac{\partial e}{\partial odometri[sag]} = 1$$

$$kTur0 = \frac{\partial k}{\partial odometri[sol]} = 1$$

$$kTur1 = \frac{\partial k}{\partial odometri[sag]} = 1$$

$$fTur0 = \frac{\partial f}{\partial odometri[sol]} = \frac{aksUzunlugu \times kTur0 \times e - aksUzunlugu \times k \times eTur0}{e^2}$$

$$fTur1 = \frac{\partial f}{\partial odometri[sag]} = \frac{aksUzunlugu \times kTur1 \times e - aksUzunlugu \times k \times eTur1}{e^2}$$

$$gTur0 = \frac{\partial g}{\partial odometri[sol]} = \frac{eTur0}{2 \times aksUzunlugu} \times \cos\left(\frac{e}{2 \times aksUzunlugu}\right)$$

$$gTur1 = \frac{\partial g}{\partial odometri[sag]} = \frac{eTur1}{2 \times aksUzunlugu} \times \cos\left(\frac{e}{2 \times aksUzunlugu}\right)$$

$$hTur0 = \frac{\partial h}{\partial \text{odometri[sol]}} = -\frac{eTur0}{2 \times aksUzunlugu} \times \sin\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$hTur1 = \frac{\partial h}{\partial \text{odometri[sag]}} = -\frac{eTur1}{2 \times aksUzunlugu} \times \sin\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$jTur0 = \frac{\partial j}{\partial \text{odometri[sol]}} = \frac{eTur0}{2 \times aksUzunlugu} \times \cos\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$jTur1 = \frac{\partial j}{\partial \text{odometri[sag]}} = \frac{eTur1}{2 \times aksUzunlugu} \times \cos\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$rTur0 = \frac{\partial r}{\partial \text{odometri[sol]}} = \frac{eTur0}{aksUzunlugu}$$

$$rTur1 = \frac{\partial r}{\partial \text{odometri[sag]}} = \frac{eTur1}{aksUzunlugu}$$

$$dx0 = \frac{\partial Mtr_{kinematik}(0,0)}{\partial \text{odometriDegeri[sol]}} = fTur0 \times g \times h + f \times gTur0 \times h + f \times g \times hTur0$$

$$dx1 = \frac{\partial Mtr_{kinematik}(0,0)}{\partial \text{odometriDgeri[sag]}} = fTur1 \times g \times h + f \times gTur1 \times h + f \times g \times hTur1$$

$$dy0 = \frac{\partial Mtr_{kinematik}(1,0)}{\partial \text{odometriDegeri[sol]}} = fTur0 \times g \times j + f \times gTur0 \times j + f \times g \times jTur0$$

$$dy1 = \frac{\partial Mtr_{kinematik}(1,0)}{\partial \text{odometriDgeri[sag]}} = fTur1 \times g \times j + f \times gTur1 \times j + f \times g \times jTur1$$

$$dth0 = \frac{\partial Mtr_{kinematik}(2,0)}{\partial \text{odometriDegeri[sol]}} = rTur0$$

$$dth1 = \frac{\partial Mtr_{kinematik}(2,0)}{\partial odometriDegeri[sag]} = rTur1$$

$$V_t = \begin{pmatrix} dx0 & dx1 \\ dy0 & dy1 \\ dth0 & dth1 \end{pmatrix}$$

3. Algı Güncellemesi

Algı güncellemesinde öncelikle robotun hareket güncellemesi sonucu elde edilen konumu ve algılayıcı ölçümleri bir doğru çıkarım fonksiyonuna gönderilir. Bu fonksiyonda kullanıcı seçimine göre İki Noktadan Doğru Çıkarımı ile ya da Hough Dönüşümü ile noktalardan doğru elde edilir. Hough Dönüşümü uygulanıyorsa tabi ki önceden belirlenmiş bir sayıda nokta toplanana kadar doğru uydurma işlemi yapılmaz ve kümeye nokta eklemeye devam edilir. Doğru çıkarım fonksiyonunun döndürdüğü gözlemler algı güncellemesi için kullanılır, eğer bir gözlem elde edilemezse algı güncellemesi yapılmaz.

Genel algoritmadan farklı olarak bu uygulamada gözlemler robotun yerel koordinat düzlemine göre değil, küresel koordinat düzlemine göre elde edilmektedir. Çünkü robot hareket halindeyken noktasal verileri toplamaktadır ve bu nokta kümesinden doğru çıkarılmaktadır, bu işlem küresel koordinat düzleminde yapılabilmektedir. Dolayısı ile genel algoritmadakinden farklı olarak gözlemin mutlak koordinatının yani küresel koordinat düzlemindeki konumunun elde edilmesine gerek yoktur fakat göreceli koordinatının yani yerel koordinat düzlemine göre konumunun elde edilmesi gerekmektedir. Gözlemlerin göreceli koordinatlarını ifade etmek için de bu gözlemi yapan algılayıcıdan

3.1 h_t matrisi

Tüm gözlemler için elde edilen yer işaretçileri ile karşılaştırma yapılarak bunların daha önceden görülen bir yer işaretçisi mi yoksa yeni bir yer işaretçisi mi olduğunun bulunması gerekmektedir. Karşılaştırma yapılırken yer işaretçilerinin robotun yerel koordinat düzlemindeki konumları kullanılır. Yer işaretçilerinin belirsizliklerinin de yine ölçüm uzayında ifade edilmesi gerektiğinden burada h_t jacobian'ı kullanılarak bir dönüşüm yapılır.

h_t jacobian'ı, yer işaretçilerinin mutlak ve göreceli konumları arasında dönüşüm yapan

fonksiyonun (x, y, θ) robot konumu ve (r, θ) mutlak yer işaretçisi konumu parametrelerine göre kısmi türevidir.

$$h_t = \begin{pmatrix} \frac{\partial r_{goreceli}}{\partial x} & \frac{\partial r_{goreceli}}{\partial y} & \frac{\partial r_{goreceli}}{\partial \theta_{robot}} & \frac{\partial r_{goreceli}}{\partial r_{mutlak}} & \frac{\partial r_{goreceli}}{\partial \theta_{mutlak}} \\ \frac{\partial \theta_{goreceli}}{\partial x} & \frac{\partial \theta_{goreceli}}{\partial y} & \frac{\partial \theta_{goreceli}}{\partial \theta_{robot}} & \frac{\partial \theta_{goreceli}}{\partial r_{mutlak}} & \frac{\partial \theta_{goreceli}}{\partial \theta_{mutlak}} \end{pmatrix}$$

Bir gözlemin göreceli konumu bulunurken aşağıdaki eşitlik kullanılmaktadır. Bu eşitlik, noktanın doğruya uzaklığı formülünden çıkarılmıştır. Burada x_0 ve y_0 , İki Noktadan Doğru Çıkarımı kullanılıyorsa aynı taraftaki iki algılayıcının orta noktası, Hough Dönüşümü kullanılıyorsa doğru uydurulan nokta kümesini oluşturan algılayıcının konumudur. θ_{mutlak} , doğrunun küresel koordinat düzlemine göre, Hough uzayı parametreleri ile ifade edilmiş açısı, $\theta_{goreceli}$ ise robota göre açısıdır. Noktanın doğruya uzaklığı formülü aradaki dik uzaklığı vermektedir fakat bizim bulmak istediğimiz uzaklık algılayıcının ışını göndereceği açıyla ilgilidir, bu nedenle mesafe $\cos \theta_{goreceli}$ 'ye bölünmektedir.

$$r_{goreceli} = \frac{\left| \frac{\cos \theta_{mutlak}}{\sin \theta_{mutlak}} x_0 + y_0 - \frac{r_{mutlak}}{\sin \theta_{mutlak}} \right|}{\cos \theta_{goreceli} \sqrt{\cot^2 \theta_{mutlak} + 1}}$$

$$f = \left| \frac{\cos \theta_{mutlak}}{\sin \theta_{mutlak}} x_0 + y_0 - \frac{r_{mutlak}}{\sin \theta_{mutlak}} \right|$$

$$g = \cos \theta_{goreceli} \sqrt{\cot^2 \theta_{mutlak} + 1}$$

$$i = \frac{\cos \theta_{mutlak}}{\sin \theta_{mutlak}} x_0 + y_0 - \frac{r_{mutlak}}{\sin \theta_{mutlak}}$$

$$m = \frac{\cos \theta_{mutlak}}{\sin \theta_{mutlak}} x_0$$

$$n = \frac{\cos \theta_{mutlak}}{\sin \theta_{mutlak}}$$

$$p = \frac{r_{mutlak}}{\sin \theta_{mutlak}}$$

$$j = \sqrt{\cot^2 \theta_{mutlak} + 1}$$

$$k = \cos \theta_{goreceli}$$

$$i = n \times x_0 + y_0 - p$$

$$f = |i|$$

$$g = j \times k$$

$$r_{goreceli} = \frac{f}{g}$$

$$r'_{goreceli} = \frac{f'g - fg'}{g^2}$$

$$f' = \text{sgn}(i) \times i'$$

$$i' = m' + y'_0 - p'$$

$$m' = n'x_0 + nx'_0$$

$$n' = \frac{\cos' \theta_{mutlak} \sin \theta_{mutlak} - \cos \theta_{mutlak} \sin' \theta_{mutlak}}{\sin^2 \theta_{mutlak}}$$

$$p' = \frac{r'_{mutlak} \sin \theta_{mutlak} - r_{mutlak} \sin' \theta_{mutlak}}{\sin^2 \theta_{mutlak}}$$

$$g' = j' \times k + j \times k'$$

$$j' = \left((\cot^2 \theta_{mutlak} + 1)^{\frac{1}{2}} \right)' = \frac{1}{2} (\cot^2 \theta_{mutlak} + 1)^{-\frac{1}{2}} \times (\cot^2 \theta_{mutlak} + 1)'$$

$$j' = \frac{1}{2} (\cot^2 \theta_{mutlak} + 1)^{-\frac{1}{2}} \times 2 \times \cot \theta_{mutlak} \times \cot' \theta_{mutlak} = (\cot^2 \theta_{mutlak} + 1)^{-\frac{1}{2}} \times \cot \theta_{mutlak} \times \cot' \theta_{mutlak}$$

$$k' = \cos' \theta_{goreceli}$$

$$x_0 = x + a \cos \theta_{robot} + b \sin \theta_{robot}$$

$$y_0 = y + c \cos \theta_{robot} + d \sin \theta_{robot}$$

Burada x ve y robot konumu parametreleridir. a , b , c ve d parametreleri ise algılayıcının robot üzerindeki konumu ile ilgili sabit mesafe bilgileridir.

$$x'_0 = x' + a \cos' \theta_{robot} + b \sin' \theta_{robot}$$

$$y'_0 = y' + c \cos' \theta_{robot} + d \sin' \theta_{robot}$$

Bu noktada eşitliklerdeki tüm değişkenler temel bileşenlerine ayrılmış durumda. Bu bileşenlerin x , y , θ_{robot} , r_{mutlak} ve θ_{mutlak} parametrelerine göre kısmi türevleri aşağıda sıralanmıştır. Bu değerlerin yerlerine konulmasıyla h_i matrisi elde edilmektedir.

$$\frac{\partial \cos \theta_{goreceli}}{\partial x} = 0 \quad \frac{\partial \cot \theta_{mutlak}}{\partial x} = 0 \quad \frac{\partial \cos \theta_{robot}}{\partial x} = 0 \quad \frac{\partial \sin \theta_{robot}}{\partial x} = 0 \quad \frac{\partial \cos \theta_{mutlak}}{\partial x} = 0$$

$$\frac{\partial \sin \theta_{mutlak}}{\partial x} = 0 \quad \frac{\partial r_{mutlak}}{\partial x} = 0 \quad \frac{\partial x}{\partial x} = 1 \quad \frac{\partial y}{\partial x} = 0$$

$$\frac{\partial \cos \theta_{goreceli}}{\partial y} = 0 \quad \frac{\partial \cot \theta_{mutlak}}{\partial y} = 0 \quad \frac{\partial \cos \theta_{robot}}{\partial y} = 0 \quad \frac{\partial \sin \theta_{robot}}{\partial y} = 0 \quad \frac{\partial \cos \theta_{mutlak}}{\partial y} = 0$$

$$\frac{\partial \sin \theta_{mutlak}}{\partial y} = 0 \quad \frac{\partial r_{mutlak}}{\partial y} = 0 \quad \frac{\partial x}{\partial y} = 0 \quad \frac{\partial y}{\partial y} = 1$$

Duvar robotun solunda görüldüyse $\frac{\partial \cos \theta_{goreceli}}{\partial \theta_{robot}} = -\sin \theta_{goreceli}$

Duvar robotun önünde görüldüyse $\frac{\partial \cos \theta_{goreceli}}{\partial \theta_{robot}} = -\sin \theta_{goreceli} \times \text{sgn}(\theta_{robot} - \theta_{mutlak})$

Duvar robotun sağında görüldüyse $\frac{\partial \cos \theta_{goreceli}}{\partial \theta_{robot}} = \sin \theta_{goreceli}$

$$\frac{\partial \cot \theta_{mutlak}}{\partial \theta_{robot}} = 0 \quad \frac{\partial \cos \theta_{robot}}{\partial \theta_{robot}} = -\sin \theta_{robot} \quad \frac{\partial \sin \theta_{robot}}{\partial \theta_{robot}} = \cos \theta_{robot}$$

$$\frac{\partial \cos \theta_{mutlak}}{\partial \theta_{robot}} = 0 \quad \frac{\partial \sin \theta_{mutlak}}{\partial \theta_{robot}} = 0 \quad \frac{\partial r_{mutlak}}{\partial \theta_{robot}} = 0 \quad \frac{\partial x}{\partial \theta_{robot}} = 0 \quad \frac{\partial y}{\partial \theta_{robot}} = 0$$

$$\frac{\partial \cos \theta_{goreceli}}{\partial r_{mutlak}} = 0 \quad \frac{\partial \cot \theta_{mutlak}}{\partial r_{mutlak}} = 0 \quad \frac{\partial \cos \theta_{robot}}{\partial r_{mutlak}} = 0 \quad \frac{\partial \sin \theta_{robot}}{\partial r_{mutlak}} = 0 \quad \frac{\partial \cos \theta_{mutlak}}{\partial r_{mutlak}} = 0$$

$$\frac{\partial \sin \theta_{mutlak}}{\partial r_{mutlak}} = 0 \quad \frac{\partial r_{mutlak}}{\partial r_{mutlak}} = 1 \quad \frac{\partial x}{\partial r_{mutlak}} = 0 \quad \frac{\partial y}{\partial r_{mutlak}} = 0$$

Duvar robotun solunda görüldüyse $\frac{\partial \cos \theta_{goreceli}}{\partial \theta_{mutlak}} = \sin \theta_{goreceli}$

Duvar robotun önünde görüldüyse $\frac{\partial \cos \theta_{goreceli}}{\partial \theta_{mutlak}} = \sin \theta_{goreceli} \times \text{sgn}(\theta_{robot} - \theta_{mutlak})$

Duvar robotun sağında görüldüyse $\frac{\partial \cos \theta_{goreceli}}{\partial \theta_{mutlak}} = -\sin \theta_{goreceli}$

$$\frac{\partial \cot \theta_{mutlak}}{\partial \theta_{mutlak}} = -\left(\frac{1}{\sin^2 \theta_{mutlak}}\right) \quad \frac{\partial \cos \theta_{robot}}{\partial \theta_{mutlak}} = 0 \quad \frac{\partial \sin \theta_{robot}}{\partial \theta_{mutlak}} = 0 \quad \frac{\partial \cos \theta_{mutlak}}{\partial \theta_{mutlak}} = -\sin \theta_{mutlak}$$

$$\frac{\partial \sin \theta_{mutlak}}{\partial \theta_{mutlak}} = \cos \theta_{mutlak} \quad \frac{\partial r_{mutlak}}{\partial \theta_{mutlak}} = 0 \quad \frac{\partial x}{\partial \theta_{mutlak}} = 0 \quad \frac{\partial y}{\partial \theta_{mutlak}} = 0$$

$$\frac{\partial r_{goreceli}}{\partial x} = \frac{\text{sgn}(i) \times \frac{\cos \theta_{mutlak}}{\sin \theta_{mutlak}}}{\sqrt{\cot^2 \theta_{mutlak} + 1} \times \cos \theta_{goreceli}}$$

$$\frac{\partial r_{goreceli}}{\partial y} = \frac{\text{sgn}(i)}{\cos \theta_{goreceli} \sqrt{\cot^2 \theta_{mutlak} + 1}}$$

$$\frac{\partial r_{goreceli}}{\partial \theta_{robot}} = \frac{(\text{sgn}(i) \times ((nb + d) \cos \theta_{robot} - (na + c) \sin \theta_{robot}))g - \left| nx_0 + y_0 - \frac{r_{mutlak}}{\sin \theta_{mutlak}} \right| (j \times \cos' \theta_{goreceli})}{g^2}$$

$$\frac{\partial r_{goreceli}}{\partial r_{mutlak}} = \frac{\left(\frac{\text{sgn}(i)}{\sin \theta_{mutlak}} \right)}{\cos \theta_{goreceli} \sqrt{\cot^2 \theta_{mutlak} + 1}}$$

$$\frac{\partial r_{goreceli}}{\partial \theta_{mutlak}} = \frac{\text{sgn}(i) \times \left(-x_0 - \frac{n(\cos \theta_{mutlak} x_0 - r_{mutlak})}{\sin \theta_{mutlak}} \right) \times g - f \left(\frac{-\cot \theta_{mutlak} \times \cos \theta_{goreceli}}{\sin^2 \theta_{mutlak} \times j} + j \cos' \theta_{goreceli} \right)}{g^2}$$

$\theta_{goreceli}$, θ_{mutlak} ile algılayıcının duruş açısının farkı alınarak bulunur. Algılayıcının duruş açısı

θ_{robot} kullanılarak hesaplanır. $\theta_{goreceli}$ hesabı ile ilgili kısmi türevler aşağıda verilmiştir.

$$\frac{\partial \theta_{goreceli}}{\partial x} = 0 \quad \frac{\partial \theta_{goreceli}}{\partial y} = 0 \quad \frac{\partial \theta_{goreceli}}{\partial \theta_{robot}} = -1 \quad \frac{\partial \theta_{goreceli}}{\partial r_{mutlak}} = 0 \quad \frac{\partial \theta_{goreceli}}{\partial \theta_{mutlak}} = 1$$

$$h_t = \begin{pmatrix} \frac{\partial r_{goreceli}}{\partial x} & \frac{\partial r_{goreceli}}{\partial y} & \frac{\partial r_{goreceli}}{\partial \theta_{robot}} & \frac{\partial r_{goreceli}}{\partial r_{mutlak}} & \frac{\partial r_{goreceli}}{\partial \theta_{mutlak}} \\ \frac{\partial \theta_{goreceli}}{\partial x} & \frac{\partial \theta_{goreceli}}{\partial y} & \frac{\partial \theta_{goreceli}}{\partial \theta_{robot}} & \frac{\partial \theta_{goreceli}}{\partial r_{mutlak}} & \frac{\partial \theta_{goreceli}}{\partial \theta_{mutlak}} \end{pmatrix}$$

h_t matrisi hesabı ile ilgili son olarak şu belirtmelidir, θ_{mutlak} sıfır derece olduğunda

$$r_{goreceli} = \frac{\left| x_0 - \frac{r_{mutlak}}{\cos \theta_{mutlak}} \right|}{\cos \theta_{goreceli}}$$

haline gelmektedir, dolayısı ile h_t matrisi de değişmektedir. Burada bu eşitliğin türevine yer verilmemiştir.

3.2 Gzlem – Yer İřaretisi Karřılařtırması

Bu uygulamada gzlemlerle yer iřaretileri karřılařtırılırken ncelikle grldkleri konumlara bakılmaktadır, gzlenen dođru parasının u noktaları yer iřaretisinin u noktalarından uzaktaysa bu direkt yeni bir yer iřareti kabul edilmektedir.

Karřılařtırmada kullanılan bir yer iřaretisinin robota gre aısı olan $\theta_{goreceli}$ hesaplandıđında 90 dereceden byk ıkarsa robotun bu yer iřaretisini o konumda grmesi zaten imkansız olduđundan dođrudan gzlemin bu yer iřaretisi olmadıđı kabul edilmektedir.

Bu bahsedilen durumlar ortaya ıkmadıđında gzlem ile yer iřaretisi arasındaki Mahalanobis mesafesine bakılmaktadır.