

Otonom Sürüş Sistemleri: Temel Kavramlar, Kontrol Mimarileri ve Algoritmik Temeller

1 Otonom Sürüş Mimarileri ve Altyapısı

1.1 Tarihsel Gelişim, Motivasyon ve Etik Zorluklar

Otonom sürüş sistemlerinin geliştirilmesi, otomobilin icadından hemen sonra, 19. yüzyılın sonlarında hayalini kurulmaya başlanan bir hedeftir.¹ İlk denemeler arasında 1925'te uzak-tan kumandalı olarak trafiğe çıkarılan "Phantom Car"¹ ve 1960'larda yol altyapısına gömülü manyetik kablolarla yönlendirilen sistemler (RCA Labs' Wire Controlled Car) bulunmaktadır.¹ Ancak, bugünkü vizyon tabanlı otonominin temelleri, 1980'lerde Carnegie Mellon Üniversitesi'nin Navlab projesi ve yol görüntülerini doğrudan araç komutlarına dönüştüren yapay sinir ağı tabanlı ALVINN (1988) gibi öncü çalışmalarla atılmıştır.¹

1.1.1 Motivasyon ve Faydalar

Otonom sürüş teknolojisine yapılan yoğun yatırımın temel motivasyonu, sadece sürüş kolaylığını artırmak değil, aynı zamanda küresel ölçekte yol güvenliğini ve çevresel verimliliği iyileştirmektir.¹ Dünya genelinde trafik kazalarının büyük çoğunluğunun insan hatasından (hız, dikkat dağınıklığı, alkol) kaynaklandığı göz önüne alındığında, otonom araçlar kazaları yüzde 87'ye varan oranlarda azaltma potansiyeli taşımaktadır.¹ Ayrıca, daha verimli sürüş profilleri sayesinde yakıt tüketimi ve egzoz emisyonları yüzde 66'ya kadar düşürülebilir, bu da çevresel açıdan önemli bir kazanç sağlar.¹ Otonom sistemler ayrıca engelliler için mobilitayı artırır ve araç paylaşım modellerini optimize ederek park alanı kullanımını yüzde 48'e kadar azaltabilir.¹

1.1.2 Otonomi Seviyeleri (SAE J3016) ve Regülasyon

Otonom sistemlerin kabiliyetleri, 2014 yılında SAE (Society of Automotive Engineers) tarafından yayımlanan J3016 standardıyla 0'dan 5'e kadar seviyelendirilmiştir.¹ Level 2 (Kısmi Otomasyon) seviyesinde araç, hem boylamsal hem de yanal kontrolü sağlasa da, sürücünün her an tetikte olması beklenir. Level 3 (Şartlı Otomasyon) ise kritik bir eşiktir; bu seviyede sistem, yetersiz kaldığı durumları erken teşhis edip sürücüyü devralmaya çağırır. En yüksek seviye olan Level 5 (Tam Otomasyon), sistemin tüm koşullarda insan müdahalesi olmadan çalışabileceği anlamına gelir.¹

Bu sistemlerin yaygınlaşması, Fonksiyonel Güvenlik (Functional Safety) standartlarını zorunlu kılmaktadır. ISO 26262 standardı, otomotiv yazılım ve donanımının geliştirilmesinden hizmetten çıkarılmasına kadar olan tüm yaşam döngüsünde güvenlik gereksinimlerini detaylandırır.² Bu standarda uygunluk, otonom sürüş sistemlerinin ticari olarak sertifikalandırılması için hayati öneme sahiptir.

1.1.3 Etik Zorluklar (The Trolley Problem)

Otonom araçların tasarımı, teknik zorlukların ötesinde, etik ikilemleri de beraberinde getirir. En bilinen örnek olan “Trolley Problemi”¹, aracın kaçınılmaz bir kaza anında, yayaları mı yoksa araç içindeki yolcuları mı koruyacağı gibi programlanmış kararları gerektirir.¹ Bu tür durumlar, felsefi birer düşünce deneyinden, aracın davranışını belirleyen algoritmik bir gereksinime dönüşmüştür. Araştırmalar, Moral Machine gibi platformlar aracılığıyla, bu etik kararların toplumsal tercihlere göre nasıl şekillenebileceğini incelemektedir.¹

1.2 Otonom Sürüş Mimarileri: Hiyerarşik Yaklaşımlar

Otonom sürüş yığınının (stack) tasarlanmasında, sistemin karmaşıklığını yönetmek ve güvenilirliği sağlamak amacıyla farklı mimariler geliştirilmiştir. Bu mimariler temelde Modüler, Uçtan Uca ve Doğrudan Algılama yaklaşımları olarak sınıflandırılabilir.

1.2.1 Modüler Boru Hattı Mimarisi (Modular Pipeline)

Modüler mimari, otonom sürüş görevini Algılama (Perception), Konumlandırma (Localization), Planlama (Planning) ve Kontrol (Control) gibi sıralı ve ayrık bileşenlere ayırır.¹ Bu hiyerarşik yapı, DARPA Challenge’in ilk kazananları ve günümüz endüstri liderleri (Waymo, Uber, Zoox) tarafından yaygın olarak kullanılmaktadır.¹

Bileşenlerin Roller ve Çıktıları: Algılama (Perception): Çevreyi algılar (nesne tespiti, sınıflandırma, takip). Çıktısı, tanımlanmış nesnelerin listesi, serbest sürüş alanı ve trafik sinyali durumları gibi semantik ve geometrik temsillerdir.¹

Konumlandırma (Localization): Aracın global veya lokal haritadaki anlık, yüksek hassasiyetli pozisyonunu ve yönelimini belirler. Çıktı tipi, santimetre düzeyinde hassasiyete sahip Ego-Araç durumu (x, y, ψ) , hız ve ivme bilgileridir.¹

Planlama (Planning): Algılama ve Konumlandırma çıktılarını kullanarak güvenli, etik ve verimli bir hareket stratejisi oluşturur. Çıktısı, takip edilecek geometrik yörünge noktaları (waypoints) ve hız profilini içeren yüksek seviyeli komutlardır.¹

Kontrol (Control): Planlanan yörüngeyi fiziksel olarak takip etmek için gereken düşük seviyeli komutları hesaplar. Çıktısı, aktüatörlere gönderilen sinyallerdir (direksiyon açısı δ , gaz/fren basıncı).¹

1.2.2 Uçtan Uca Öğrenme (End-to-End – E2E)

E2E öğrenme, ham sensör girdilerini (örneğin RGB görüntüler) doğrudan kontrol eylemlerine (direksiyon, gaz, fren) eşleyen tek bir büyük sinir ağı kullanır.¹ Ara temsillerin öğrenme aşamasında gizli kalması nedeniyle bu yaklaşıma “kara kutu” denir.¹ E2E sistemlerin başlıca avantajı, modelin doğrudan sürüş görevini optimize edecek şekilde eğitilebilmesi ve veri etiketleme maliyetlerinin (yalnızca aktüatör ve sensör verisi gerektirdiği için) düşük olmasıdır.¹ Ancak, yorumlanabilirlik ve genelleme yetenekleri düşüktür.¹

1.2.3 Doğrudan Algılama (Direct Perception)

Doğrudan Algılama, Modüler ve E2E yaklaşımları arasında bir melezdir. Bu yöntemde, ham sensör girdisinden, sürüş için anlamlı olan ve psikolojik bir terim olan “Affordances” (olanaklar) adı verilen yorumlanabilir, düşük boyutlu ara temsiller öğrenilir.¹ Bu temsiller (örneğin, şerit işaretlerine uzaklık, önleyici araca olan mesafe) daha sonra klasik PID

kontrolcöleri veya sonlu durum makineleri (FSM) gibi kontrol algoritmalarına girdi olarak verilir.¹ Ara temsillerin kompakt ve yorumlanabilir olması, modelin genelleme yeteneđini ve denetlenebilirliđini artırır.¹

1.3 Örnek Çözüm Entegrasyonu: Mimarilerin Kapsamlı Karşılaştırması

Otonom sürüş mimarilerinin seçimi, geliştirme maliyetleri, güvenlik sertifikasyonu ve operasyonel zorluklar üzerinde doğrudan etkilidir. Aşağıdaki kavramsal tablo, Modüler Boru Hattı ve Uçtan Uca yaklaşımlar arasındaki temel farklılıkları ortaya koymaktadır.¹

Otonom Sürüş Mimarileri Karşılaştırması

Kriter	Modüler Boru Hattı	Uçtan Uca (E2E)
Şeffaflık / Yorumlanabilirlik	Yüksek (ayrık modül çıktıları denetlenebilir)	Düşük (kara kutu modelinde karar süreci kapalıdır) ¹
Hata Ayıklama / Teşhis	Kolay (hata kaynağı modül bazında izole edilebilir)	Zor (hata teşhisi güçleşir) ¹
Veri Gereksinimi	Orta (her modül için özel etiketlenmiş veri setleri gerekir)	Çok yüksek (tüm davranış kapsayacak çeşitli veri dağılımı gereklidir) ¹
HD Harita Bağımlılığı	Yüksek (özellikle konumlandırma ve planlama)	Düşük/azaltılmış ¹
Regölasyon Uyumu	Kolay (ISO 26262 V&V süreçleri desteklenir)	Zor (güvenlik sertifikasyonu için doğrulama güçlüğü)

Mimari Seçiminin Analizi:

Modüler sistemlerin sektör liderleri tarafından tercih edilmesinin temel nedenlerinden biri, şeffaflık ve denetlenebilirliktir. Modüler mimaride, Algılama ve Planlama gibi kritik aşamalar için santimetre düzeyinde hassasiyet gerektiren Yüksek Tanımlı (HD) haritalara bağımlılık yüksektir.¹ Bu haritaların taranması, oluşturulması ve sürekli güncellenmesi (yol yapısındaki her değişiklikte) yüksek üretim ve operasyonel maliyetler yaratır.¹

Buna karşın, E2E sistemlerin “kara kutu” doğası, kontrol komutlarının neden verildiğini (şeffaflık) ve hataların kökenini (teşhis) anlamayı zorlaştırır. Bu, özellikle ISO 26262 gibi standartlara uyum ve dolayısıyla güvenlik sertifikasyonu süreçlerinde büyük bir engel teşkil eder. Bu durum, modüler mimarinin yüksek maliyetine rağmen güvenlik odaklı geliştirme yapan kuruluşlar için neden daha cazip olduğunu açıklamaktadır.

2 Araç Kinematığı, Dinamik Modeller ve Kontrol Temelleri

2.1 Araç Modellemesi ve Kinematik Temeller

2.1.1 Holonomik ve Non-Holonomik Kısıtlamalar

Mühendislik sistemlerinin hareketini incelerken, sistemin konfigürasyon uzayındaki kısıtlamalar önemlidir. Bir kısıtlama, eğer sistemin koordinatları cinsinden

$$f(x_1, \dots, x_N) = 0$$

formunda ifade edilebiliyorsa, holonomik olarak adlandırılır. Holonomik sistemlerde, kontrol edilebilir serbestlik derecesi (DOF) toplam serbestlik derecesine eşittir ve sistem konfigürasyon uzayında serbestçe hareket edebilir.¹

Non-holonomik kısıtlamalar ise hız uzayını sınırlar ve

$$f(x_1, \dots, x_N, \dot{x}_1, \dots, \dot{x}_N) = 0$$

formunda olabilir, ancak koordinatlar cinsinden entegre edilemez. Bir otomobil, tekerleklerin yana kaymadan yuvarlanması (non-slip condition) nedeniyle, ileri hızı (v) ve direksiyon açısı (δ) ile hareket etmek zorunda olduğu için non-holonomik bir sisteme örnektir.¹ Bu kısıtlama, aracın anlık olarak her yöne serbestçe kayarak hareket edemeyeceği anlamına gelir.

2.1.2 Kinematik Bisiklet Modeli (Kinematic Bicycle Model)

Otomobil modellemesindeki en basit ve yaygın kullanılan model, Kinematik Bisiklet Modelidir.¹ Bu model, araç dinamiğini basitleştirmek için iki ana varsayım yapar:

- Dört tekerleği, aksların merkezinde bulunan iki hayali tekerlek ile temsil eder.
- En kritik varsayım olarak, tekerlek kayması (wheel slip) olmadığını kabul eder.¹

Bu, tekerlek yöneliminin her zaman tekerlek hız vektörü yönünde olduğu anlamına gelir. Bu varsayım nedeniyle model, ancak düşük hızlarda güvenilir sonuçlar verir.

Aracın durum vektörü genellikle

$$\mathbf{x} = [X, Y, \psi]^\top$$

olarak tanımlanır; burada X ve Y global konum, ψ ise araç yönelim açısıdır.

Kinematik Bisiklet Modeli Durum Değişim Denklemleri (Basitleştirilmiş, $\beta \approx 0$):

$$\dot{X} = v \cos(\psi) \quad | \quad \dot{Y} = v \sin(\psi) \quad | \quad \dot{\psi} = \frac{v}{L} \tan(\delta)$$

Burada v ileri hız, L aks mesafesi ve δ direksiyon açısıdır.¹

2.1.3 Dinamik Bisiklet Modeli ve Lastik Kuvvetleri

Gerçek sürüş koşullarında, özellikle yüksek hızlarda ve keskin manevralarda, tekerlek kayması daima mevcuttur.¹ Lastik ile zemin arasındaki sürtünme kuvvetleri, aracın yörüngesini doğrudan etkiler. Dinamik Bisiklet Modeli, bu kuvvetleri ve aracın atalet momentini (I_z) hesaba katarak kinetik prensiplere dayanır.¹

Lastik Kayması ve Sürtünme: Lastikler, kayma miktarı (slippage) belirli bir eşiği aşmadıkça statik sürtünme bölgesinde çalışır. Bu bölgede yanal kuvvet, kayma açısıyla lineer olarak artar. Ancak, kayma başladığında, sürtünme, statik sürtünmeden daha küçük olan kinetik (sliding) sürtünmeye düşer, bu da aracın kontrolünü zorlaştırır.¹ Kontrol sistemlerinin temel amacı, daima statik sürtünme limitleri içerisinde kalmaktır.

2.2 Uygulama: Koordinat Çerçevesi Dönüşümü ve Euler İlerlemesi

Otonom kontrol döngüsünün ilk adımlarından biri, global olarak tanımlanan hedefleri aracın kendi lokal hareket çerçevesine dönüştürmektir.

2.2.1 Küreselden Araç Çerçevesine Dönüşümün Temelleri

Kontrol algoritmaları, direksiyon komutlarını aracın ileri eksenine göre hesaplamak zorundadır. Bu, global koordinattaki hedefin (p_{hedef}) önce öteleme (translation), ardından döndürme (rotation) ile aracın lokal çerçevesine (p_H) getirilmesini gerektirir.¹

Öteleme (Translation): Aracın anlık global konumuna (p_k) göre hedefin göreceli konumu ($\Delta X, \Delta Y$) hesaplanır.

Döndürme (Rotation): Global göreceli vektör, aracın yönelim açısının negatifi ($-\psi_k$) kadar döndürülür. Bu, vektörü aracın X_H (ileri) ve Y_H (sol) eksenlerine hizalar.¹

Dönüşüm matrisi $R(-\psi_k)$ ile çarpım gerçekleştirilir:

$$\begin{pmatrix} X_H \\ Y_H \end{pmatrix} = \begin{pmatrix} \cos(\psi_k) & \sin(\psi_k) \\ -\sin(\psi_k) & \cos(\psi_k) \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix}$$

Örnek Çözüm Entegrasyonu 1 (Dönüşüm): Araç Koordinatlarında Hedef Konum Hesabı¹ Veri/Sabitler

- Anlık durum p_k
- Hedef p_{hedef}
- Yönelim ψ_k

Değerler: $(0, 0)$, $(5.0, 2.0)$ m, $\psi_k = \pi/4 \approx 0.7854$ rad.

$\cos(\pi/4) = \sin(\pi/4) \approx 0.7071$ kullanılarak $\Delta X = 5.0$ ve $\Delta Y = 2.0$ bulunur:

$$X_H = 5.0(0.7071) + 2.0(0.7071) = 7.0 \times 0.7071 \approx 4.9497 \text{ m}$$

$$Y_H = -5.0(0.7071) + 2.0(0.7071) = -3.0 \times 0.7071 \approx -2.1213 \text{ m}$$

Sonuç: Hedef konum araç çerçevesinde $p_H \approx (4.95, -2.12)$ m'dir. Negatif Y_H değeri, hedefin aracın sağında olduğunu gösterir.¹

2.2.2 Örnek Çözüm Entegrasyonu 2: Euler İlerlemesi ile Konum Tahmini

Kontrol komutu uygulandıktan sonraki araç durumunu tahmin etmek için kinematik denklemlerin zamanla entegrasyonu gerekir. Basit bir ayrıklaştırma yöntemi olan Euler İlerlemesi kullanılır. $\Delta t = 1.0$ s için, Pure Pursuit (PP) algoritması ile hesaplanan $\delta \approx -0.3736$ radyan direksiyon açısı kullanılır.¹

Türevlerin Hesabı ($v = 8.0$ m/s, $L = 2.5$ m, $\psi_k \approx 0.7854$ rad):

$$\dot{x}_k = 8.0 \times \cos(0.7854) \approx 5.6568 \text{ m/s}$$

$$\dot{y}_k = 8.0 \times \sin(0.7854) \approx 5.6568 \text{ m/s}$$

$$\dot{\psi}_k = \frac{8.0}{2.5} \times \tan(-0.3736) \approx 3.2 \times (-0.3939) \approx -1.2605 \text{ rad/s}$$

Euler Entegrasyonu ($p_{k+1} = p_k + \dot{p}_k \Delta t$):

$$x_{k+1} = 0.0 + 5.6568 \times 1.0 = 5.6568 \text{ m}$$

$$y_{k+1} = 0.0 + 5.6568 \times 1.0 = 5.6568 \text{ m}$$

$$\psi_{k+1} = 0.7854 + (-1.2605) \times 1.0 = -0.4751 \text{ rad}$$

Analiz: Euler İlerlemesinin Sınırlılıkları Bu hesaplama, $v = 8.0$ m/s gibi yüksek bir hızda ve $\Delta t = 1.0$ s gibi büyük bir zaman adımında kinematik modelin kullanıldığı durumlar için kritik sonuçlar doğurur. Kinematik modelin tekerlek kaymasını yok sayması, yüksek hızdaki keskin manevralarda ($\dot{\psi}_k \approx -1.2605$ rad/s) fiziksel kısıtlamaları ihmal eder. Daha da önemlisi, Euler entegrasyonunda, aracın yönelimi (ψ) bu 1 saniyelik zaman adımı boyunca sabit kabul edilir. Gerçekte, ψ sürekli ve hızlı bir şekilde değişirken, bu lineer kabul, tahmini konumda ciddi sayısal sapmalara yol açar ve birikimli hatayı artırır.

Bu durum, kontrol stratejilerinin (MPC gibi) daha küçük zaman adımları ve daha yüksek frekanslarda çalışması veya daha hassas Runge–Kutta gibi entegrasyon yöntemlerini kullanması gerektiğini gösterir. Bu sapma, Model Tahmine Dayalı Kontrolün (MPC) neden sadece anlık değil, geleceğe dönük bir optimizasyon çerçevesi gerektirdiğinin temel nedenidir.

3 Yörünge Takip Kontrol Algoritmaları

3.1 Geometrik Kontrol Yöntemleri

Geometrik kontrol yöntemleri, aracın kinematik modelini kullanarak, basit geometri kuralları ile istenen yörüngeyi takip etmek için gerekli olan direksiyon açısını hesaplar. Bu yöntemler, hesaplama açısından hızlı ve düşük yüküldür.

3.1.1 Stanley Kontrolcü

Stanley Kontrolcü, referans noktası olarak aracın ön aksını alır ve amacı, yörüngeyi en yakın noktası ile ön aks arasındaki çapraz izleme hatasını (e) ve aracın yörüngeye göre yönelim hatasını (ψ) minimize etmektir.¹ Stanley yönteminin, direksiyon açısını

$$\delta = \psi - \arctan\left(\frac{ke}{v}\right)$$

formunda tanımlayan kontrol yasası sayesinde, çapraz izleme hatasının hızdan bağımsız olarak üstel olarak sıfıra yakınsadığı kanıtlanmıştır. Bu özellik, düz yollarda yüksek performans ve küresel kararlılık sağlar, ancak virajlarda Pure Pursuit’e kıyasla daha düşük performans gösterebilir.¹

3.1.2 Pure Pursuit (PP) Algoritması

Pure Pursuit, direksiyon açısını belirlemek için dinamik olarak belirlenen bir “Bakış Mesafesi” (L_d) kadar ilerideki hedef noktasını (Target Point – TP) kullanır.¹ Algoritma, arka aks merkezinden başlayarak TP’den geçecek dairesel bir yörüngeyi tanımlar ve bu yörüngeyi takip etmek için gereken direksiyon açısını (δ) hesaplar.¹ PP, özellikle dönüşlerde iyi performans gösterir.³

3.2 Uygulama: Pure Pursuit Direksiyon Açısı (δ) Türetilmesi

PP algoritmasının çekirdeği, direksiyon açısının aracın aks mesafesi (L), bakış mesafesi (L_d) ve hedefin araç çerçevesindeki açısı (α) arasındaki geometrik ilişkiden türetilmesidir.¹

3.2.1 Geometrik Temeller ve Eğrilik Formülü

Türetme, arka aks merkezi, Anlık Dönme Merkezi (ICR) ve TP'nin oluşturduğu ikizkenar üçgenin analizine dayanır.⁴

Açısal İlişki: ICR'den TP'ye ve arka aksa olan uzaklıklar dairenin yarıçapına (R) eşittir. Bu, oluşan üçgenin ikizkenar olduğunu gösterir. Bu geometrik ilişki kullanılarak, hedefin araç çerçevesindeki açısı (α) ile ICR'deki açı (γ_1) arasında

$$\gamma_1 = 2\alpha$$

ilişkisi kurulur. Ayrıca,

$$\gamma_2 = 90^\circ - \alpha$$

olduğu bulunur.⁴

Sinüs Teoreminin Uygulanması: Üçgene Sinüs Teoremi uygulanır:

$$\frac{L_d}{\sin(2\alpha)} = \frac{R}{\sin(90^\circ - \alpha)}.$$

Yarıçap Formülü (R): Trigonometrik kimlikler ($\sin(90^\circ - \alpha) = \cos(\alpha)$ ve $\sin(2\alpha) = 2\sin(\alpha)\cos(\alpha)$) yerine konulur ve R izole edilir:

$$R = \frac{L_d}{2\sin(\alpha)}.$$

Bu, dairesel yörüngenin eğrilik (κ) değerini

$$\kappa = \frac{1}{R} = \frac{2\sin(\alpha)}{L_d}$$

olarak verir.

Direksiyon Açısı Formülü (δ): Kinematik Bisiklet Modeli, arka aks merkezi etrafında dönen bir araç için direksiyon açısını

$$\tan(\delta) = \frac{L}{R}$$

olarak tanımlar. R ifadesi yerine konulduğunda, nihai Pure Pursuit direksiyon açısı formülü türetilir:¹

$$\delta = \arctan\left(\frac{2L\sin(\alpha)}{L_d}\right).$$

3.2.2 Örnek Çözüm Entegrasyonu 3: Gerekli Yönlendirme Açısı (δ) Hesabı

Daha önce Bölüm 2.2.1'de hesaplanan araç çerçevesindeki hedef konumuna ($p_H \approx (4.95, -2.12)$ m) karşılık gelen $\sin(\alpha) \approx -0.3939$ değeri kullanılarak δ hesaplanır.¹

Veri/Sabitler:

- Aks mesafesi: $L = 2.5$ m
- Bakış mesafesi: $L_d = 5.01$ m
- Bakış açısı: $\sin(\alpha) \approx -0.3939$

$$\delta = \arctan\left(\frac{2 \times 2.5 \times (-0.3939)}{5.01}\right) = \arctan\left(\frac{-1.9695}{5.01}\right)$$
$$\delta = \arctan(-0.3931) \approx -0.3736 \text{ radyan.}$$

3.3 Optimal Kontrol: Model Tahmine Dayalı Kontrol (MPC)

Model Tahmine Dayalı Kontrol (MPC), otonom sürüşte kullanılan en gelişmiş kontrol stratejilerinden biridir.⁵ PP gibi anlık geometrik gereksinime odaklanan geri beslemeli kontrolcülerin aksine, MPC, gelecekteki yörüngeyi bir zaman ufku (prediction horizon) boyunca tahmin eden ve kontrol sinyallerini bir maliyet fonksiyonunu (cost function) minimize edecek şekilde optimize eden öngörülü bir çerçevedir.¹

3.3.1 MPC'nin Avantajları ve Kısıt Yönetimi

MPC'nin geleneksel kontrolcülere göre temel avantajları, kısıt yönetimi ve ileriye dönük optimizasyon yeteneğidir:

- **Kısıt Optimizasyonu:** MPC, sürüş güvenliği ve araç dinamiği ile ilgili kısıtlamaları (örneğin maksimum direksiyon açısı, yana kayma limitleri, hızlanma limitleri) doğrudan matematiksel kısıtlar ($\underline{\delta} \leq \delta_t \leq \bar{\delta}$) olarak optimizasyon formülasyonuna dahil eder.¹ Bu, kontrol eylemlerinin her zaman fiziksel olarak uygulanabilir ve güvenli olmasını garanti eder. PP'de bu kısıtlamalar kontrol sonrası kırpma (clipping) ile uygulanır ki bu optimal değildir.¹
- **İleriye Dönük Optimizasyon:** MPC, gelecekteki N adımdaki durumu tahmin eder ve kontrol sinyallerini, yörünge hatasını, enerji tüketimini veya sarsıntıyı (jerk) azaltacak şekilde seçer.¹ Bu, PP'ye kıyasla çok daha yumuşak ve küresel olarak daha iyi bir yörünge takibi ve kontrol sağlar.¹
- **Yüksek Performans:** MPC, çok değişkenli, kısıtlı sistemlerin denetiminde çok güçlü bir stratejidir ve geleneksel PID kontrolcülere kıyasla yüksek kapalı çevrim performansı gösterir.⁵

Kontrol Stratejileri Karşılaştırması

Kriter	Pure Pursuit (PP)	Model Tahmine Dayalı Kontrol (MPC)
Temel Mekanizma	Anlık geometrik geri besleme	Geleceği tahmin eden, optimizasyon tabanlı kontrol ¹
Kısıt Yönetimi	Kontrol sonrası kırpma (optimal değil)	Kısıtları doğrudan formülasyona dahil eder (optimal) ¹
Hesaplama Yüğü	Düşük (analitik çözüm)	Yüksek (sürekli non-lineer optimizasyon) ¹
Avantajı	Basitlik, hız	Kısıt optimizasyonu, yumuşak ve güvenli yörünge ¹

MPC'nin Fiziksel Kısıtlamaları Yönetme Yeteneği:

MPC'nin temel gücü, otonom aracın kinematik model varsayımlarının (örneğin kayma yok) veya Euler entegrasyonunun getirdiği sayısal kararsızlıkların ötesine geçebilmesinde yatar. Dinamik kısıtlamaları (lastik sürtünme limitleri) doğrudan matematiksel kısıtlar olarak formülasyona dahil ederek, MPC, aracın fiziksel sınırlarını ihlal etmeyecek şekilde kontrol sinyalini optimize eder. Bu, MPC'nin sadece performansı artırması değil, aynı zamanda fiziksel güvenliği (safety envelope) garanti altına alması için kritik öneme sahiptir. Bu karmaşık optimizasyon, gerçek zamanlı uygulamalar için FPGA veya yüksek hızlı mikroişlemciler gerektirir.⁵

4 Otonom Sürüş için Makine Öğrenimi Paradigması

4.1 Taklitçi Öğrenme (Imitation Learning – IL)

Taklitçi Öğrenme, bir uzmanın sergilediği davranışları kopyalayarak bir politika öğrenmeyi amaçlar. Bu, genellikle durumları (s_i) optimal eylemlere (a_i^*) eşleyen denetimli bir öğrenme problemi olarak formüle edilir.¹

4.1.1 Davranış Klonlama ve Kovaryat Kayması

Davranış Klonlama (Behavioral Cloning – BC), uzman demonstrasyonlarından toplanan verilerle bir politikayı eğitir. Ancak, bu yaklaşım denetimli öğrenmenin temel varsayımı olan IID (Independent and Identically Distributed – Bağımsız ve Aynı Dağılımlı) varsayımını ihlal eder.¹

Kovaryat Kayması (Covariate Shift) Problemi: Modelin test sırasında karşılaştığı durum dağılımı, eğitim sırasında gördüğü (uzmanın sürüşü) dağılımdan sapma eğilimi gösterir.¹ Ajan, uzman yörüngesinden küçük bir hata ile ayrıldığında (örneğin yoldan hafifçe sapma), bu durum eğitim setinde az rastlanan veya hiç görülmeyen “patolojik” bir yeni gözlemdir. Model bu yeni duruma uygun bir eylem üretmez, hata birikir (kaskat hatası) ve aracı yoldan çıkarır. Bu performans kaybı, öğrenilen otonom sürüş algoritmasının, eğitimde kullanılan verilerin dağılımına ait olmayan durumları gözlemlemesinden kaynaklanır.⁶

4.1.2 DAgger (Dataset Aggregation) Algoritması

DAgger, Kovaryat Kayması sorununu çözmek için tasarlanmıştır. Temel fikir, öğrenilen politikayı geliştirmek amacıyla uzmandan aktif olarak daha fazla örnek toplamaktır.¹

DAgger’ın Adım Adım Çalışma Prensipleri:¹

1. **Rollout:** Ajan, mevcut (eğitilmiş) politikası (π_i) ile ortamda (örneğin bir simülasyon) etkileşime girer ve uzman yörüngesinden sapmış yeni durumları içeren yörüngeleri toplar. Bu yörüngeler, ajanın hatalı davranışını içerir.
2. **Uzman Sorgulama:** Toplanan bu hatalı/yeni durumlara karşılık gelen optimal kurtarma eylemleri (π_{uzman}), uzman tarafından etiketlenir (sorgulanır).
3. **Agregasyon:** Bu yeni etiketlenmiş veri seti (\mathcal{D}_{yeni}), orijinal uzman demonstrasyonlarına (\mathcal{D}) eklenir ($\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{yeni}$).
4. **Yeniden Eğitim:** Politika, genişletilmiş ve hatalı durumları da içeren veri seti üzerinde yeniden eğitilir.

Bu döngü, politikayı, kendi hatalarından nasıl kurtulacağını öğrenmeye zorlar ve eğitim dağılımını, ajanın test sırasında karşılaşması muhtemel dağılımlara sürekli olarak yaklaştırır. Bu, modelin genelleme yeteneğini ve sürüş dayanıklılığını temelden artırır.¹

4.2 Evrişimli Sinir Ağları (CNN) ve Parametre Verimliliği

Evrişimli Sinir Ağları (ConvNet'ler), görüntü gibi uzamsal verileri işlemek için tasarlanmış ve geleneksel Tam Bağlantılı (FC) ağlara kıyasla devrim niteliğinde performans ve verimlilik sağlayan yapılardır.⁷

Ağırlık Paylaşımı ve Çeviri Değişmezliği: CNN'lerin temel yapı taşı Evrişimsel Katman'dır. Burada ağırlık paylaşımı (weight sharing) mekanizması kullanılır: Aynı filtre kümesi (çekirdek), giriş görüntüsünün farklı uzamsal konumlarında kullanılır.⁷ Bu, öğrenilen yerel örüntülerin (örneğin kenarlar) görüntünün neresinde olursa olsun tutarlı bir şekilde tespit edilmesini sağlar.⁷ Bu özellik, Çeviri Değişmezliği (Translation Equivariance) sağlar, yani bir nesne görüntünün farklı bir yerinde belirdiğinde ağ, onu yine de aynı nesne olarak tanımlayabilir.⁷

4.2.1 Örnek Çözüm Entegrasyonu 4: CNN Katmanının Parametre Verimliliği

Ağırlık paylaşımının en somut sonucu, öğrenilecek parametre sayısının dramatik bir şekilde azalmasıdır. Bu, görsel görevlerde aşırı öğrenmeyi önler ve hesaplama verimliliği sağlar.¹

Verilen parametreler:

- Giriş boyutu: $W \times H = 30 \times 32$
- Giriş derinliği: $C_{\text{in}} = 1$
- Çıkış derinliği (filtre sayısı): $C_{\text{out}} = 45$
- Çekirdek boyutu: $K = 3 \times 3$

Evrişim Katmanının Toplam Ağırlık Sayısı (P_{CNN}): Bias terimi dahil:

$$P_{\text{CNN}} = C_{\text{out}} \times (K \times K \times C_{\text{in}} + 1)$$
$$P_{\text{CNN}} = 45 \times (3 \times 3 \times 1 + 1) = 45 \times 10 = 450 \text{ parametre.}$$

Eşdeğer Tam Bağlantılı Katmanın Toplam Ağırlık Sayısı (P_{FC}): $N_{\text{in}} = W \times H \times C_{\text{in}}$ ve $N_{\text{out}} = C_{\text{out}}$ kullanılır:

$$N_{\text{in}} = 30 \times 32 \times 1 = 960$$
$$P_{\text{FC}} = (N_{\text{in}} \times N_{\text{out}}) + N_{\text{out}} = (960 \times 45) + 45 = 43200 + 45 = 43245 \text{ parametre.}$$

Parametre Verimliliği Analizi:

Katman Tipi	Toplam Parametre Sayısı	Verimlilik Mekanizması
Evrişimli (CNN)	450 ¹	Ağırlık paylaşımı ve yerel bağlantılar ⁷
Tam Bağlantılı (FC)	43245 ¹	Her nöronun tam bağlantısı

Verim Oranı: $\approx 1/96$.

Düşük aşırı öğrenme riski ve yüksek hesaplama hızı sağlar.¹ CNN'in 96 kat daha az parametre kullanması, modelin kapasitesini kontrol altında tutar. Yüksek boyutlu girişler için 43.245 parametreye sahip bir FC katmanı kullanmak, modelin eğitim verisine aşırı uyum sağlamasına (overfitting) yol açarak genelleme yeteneğini düşürür. CNN'in uzamsal öncelikleri matematiksel olarak kodlaması, bu verimliliği sağlayarak otonom sistemlerde gerçek zamanlı ve güvenilir çıkarımı mümkün kılar.

4.3 Pekiştirmeli Öğrenme (Reinforcement Learning – RL)

Pekiştirmeli Öğrenme, bir ajanın bir ortamla etkileşim kurarak, gelecekteki kümülatif ödülü maksimize edecek eylemleri seçmeyi öğrendiği bir paradigmaya dayanır.¹

4.3.1 Markov Karar Süreçleri ve Bellman Optimalite

RL problemleri, Durum (S), Eylem (A), Ödül (\mathcal{R}), Geçiş Olasılığı (P) ve İskonto Faktörü (γ) bileşenlerinden oluşan Markov Karar Süreçleri (MDP) ile resmleştirilir.¹ Kritik bir kavram olan Eylem-Değer Fonksiyonu $Q^\pi(s_t, a_t)$, s_t durumunda a_t eylemini aldıktan sonra politikanın takip edilmesiyle beklenen indirgenmiş (discounted) kümülatif ödülü temsil eder.¹

Optimal politikayı (π^*) tanımlayan Bellman Optimalite Denklemi (BOE), optimal Q^* fonksiyonunu rekürsif olarak ifade eder:

$$Q^*(s_t, a_t) = \mathbb{E} \left[r_t + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t, a_t \right].$$

4.3.2 Q-Learning ve Off-Policy Mekanizması

Q-Learning, bu Q^* fonksiyonunu iteratif olarak tahmin eden bir algoritmadır. Güncelleme, Temporal Difference (TD) hatasını minimize etmeye çalışır:

$$Q_{i+1}(s_t, a_t) = Q_i(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q_i(s_{t+1}, a') - Q_i(s_t, a_t) \right].^1$$

Örnek Çözüm Entegrasyonu 5: Q-Learning’i Off-Policy Yapan Mekanizma¹
Q-Learning’i politika dışı (Off-Policy) yapan temel mekanizma, maksimum değer kullanıldığı güncelleme hedefinin hesaplanma biçimidir: $\gamma \max_{a'} Q_i(s_{t+1}, a')$.¹

Politika dışı öğrenmede, eylemlerin seçilmesi için kullanılan davranış politikası (behavior policy), öğrenilen hedef politikadan (target policy) farklıdır. Q-Learning’de ajan, keşif amaçlı (örneğin ϵ -greedy) herhangi bir eylemi (a_t) seçmiş olsa da, Q -fonksiyonu güncellemesi daima s_{t+1} durumunda gelecekteki en yüksek beklenen getiriye (optimal eylem a' seçilerek) hesaplar.¹ Başka bir deyişle, algoritma, keşif için rastgele eylemlerden veri toplasa bile, bu verileri kullanarak optimal Q^* değerine doğru ilerler. Bu “decoupling”, algoritmanın veri verimliliğini (data efficiency) artırır, zira eski deneyimler tekrar tekrar kullanılabilir.⁴

4.3.3 Derin Deterministik Politika Gradyanları (DDPG)

Geleneksel Q-Learning ve DQN (Derin Q-Ağları), ayrık (discrete) eylem alanları için tasarlanmıştır. Otonom sürüşte direksiyon açısı veya gaz torku gibi sürekli (continuous) eylem alanlarını ele almak için DDPG (Deep Deterministic Policy Gradients) kullanılır.¹

Aktör-Eleştirmen Etkileşimi: DDPG, bir Aktör ve bir Eleştirmen ağı kullanır.¹

- **Eleştirmen (Critic – Q):** Durum-eylem değeri fonksiyonunu $Q(s, a)$ tahmin eder. Aktörün ürettiği eylemleri değerlendirerek MSBE (Mean-Squared Bellman Error) kaybını minimize eder.¹
- **Aktör (Actor – μ):** Deterministik politikayı öğrenir ve doğrudan en iyi eylemi $a = \mu_{\theta^\mu}(s)$ çıktılar. Aktör, Eleştirmenin gradyanını kullanarak güncellenir. Aktör, $Q(s, a)$

değerini maksimize edecek yönde (gradyan tırmanışı) Politika Gradyan Teoremi doğrultusunda ilerler.¹

Off-Policy Özelliği: DDPG, tıpkı DQN gibi Deneyim Tekrar Tamponu (Replay Buffer) kullanır. Bu mekanizma, eski verilerin tekrar kullanılmasına izin vererek algoritmayı Off-Policy (Politika Dışı) yapar ve veri verimliliği sağlar. On-Policy algoritmaların (örneğin PPO) aksine, her güncellemeden sonra veri setini atmak zorunda kalmaz.¹

5 Konumlandırma (Localization) ve Sensör Füzyonu

5.1 Konumlandırma Teknolojileri ve Kısıtlamaları

Konumlandırma, aracın çevresel ve harita referans çerçevesindeki (x, y, ψ) pozisyonunu yüksek doğrulukla belirleme yeteneğidir.¹

5.1.1 GNSS (GPS) ve IMU

NAVSTAR GPS/GNSS: Aracın mutlak konumunu (enlem, boylam) sağlar.¹ Ancak kentsel kanyonlarda, tünellerde veya yoğun ağaçlık alanlarda sinyal kaybı veya çoklu yol yayılımı (multipath) nedeniyle konum hassasiyeti ve güvenilirliği bozulabilir.¹

IMU (Atalet Ölçüm Birimi): İvmeölçer ve jiroskoplardan gelen verilerle anlık lineer ivmeyi ve açısal hızı ölçer.¹ GPS sinyali olmadan kısa süreli konum takibi (dead reckoning) için esastır.¹ Ancak, ölçülen gürültülü ivmelerin zamanla iki kez entegrasyonu nedeniyle hatası sürekli birikir (drift), bu da uzun vadede IMU'yu tek başına güvenilirmez kılar.¹

5.1.2 HD Haritalar ve SLAM

HD Haritalar: Santimetre hassasiyetinde geometrik ve anlamsal veri katmanlarıdır (şeritler, yol kenarları). LiDAR nokta bulutunu haritadaki statik özelliklerle eşleştirerek (map matching) yüksek doğruluklu, lokal konum bilgisi sağlar.¹

SLAM (Simultaneous Localization and Mapping): Haritalama ve konumlandırmayı eşzamanlı çözer.¹ SLAM'ın kritik bileşeni olan Döngü Kapatma (Loop Closure), ajanın daha önce ziyaret ettiği bir yeri tanımasını sağlayarak, birikmiş odometrik hataları global olarak düzelten bir kısıt olarak işlev görür.¹

5.2 Durum Tahmini ve Sensör Füzyonu Mekanizması

Güvenilir ve sürekli bir konum tahmini için GNSS, IMU ve HD Harita verilerinin birleştirilmesi (sensör füzyonu) gerekir. Bu süreç genellikle Genişletilmiş Kalman Filtresi (EKF) veya İşaretsiz Kalman Filtresi (UKF) gibi Bayes filtreleri ile uygulanır.¹

UKF/EKF Füzyon Mekanizması: Bu filtreler, IMU'nun yüksek frekanslı ancak sürüklenmeye eğilimli kısa vadeli hassasiyetini, GPS/HD Haritaların düşük frekanslı ancak mutlak doğruluğu ile birleştirir.¹

5.2.1 Füzyonun İki Temel Aşaması

- **Tahmin (Prediction) Adımı:** Kısa zaman aralıklarında (yüksek frekansta), IMU'dan gelen veriler kullanılarak aracın durumu (pozisyon, hız, ivme) tahmin edilir. Bu adım, GPS sinyali olmasa bile konumu korur, ancak modelin belirsizliği (kovaryans) artar.¹

- **Düzeltilme (Correction) Adımı:** GPS veya HD Haritalardan (LiDAR eşleşmesi) gelen daha az gürültülü mutlak konum ölçümleri, filtreye girdi olarak verilir. Filtre, bu ölçümler ile Tahmin adımı arasındaki farkı (inovasyon) kullanarak, IMU'dan kaynaklanan birikmiş hatayı (drifti) düzeltir ve belirsizliği azaltır.¹ Bu döngü, IMU'nun sürekli verisiyle GPS'in kesintilerini doldururken, IMU hatasını mutlak referanslarla periyodik olarak sıfırlar.¹

Konumlandırma Sensör Füzyonu Mekanizması (EKF/UKF)

Sensör	Bilgi Türü / Frekans	Füzyon Rolü
IMU	İvme, açısal hız / yüksek frekans, hata birikimi (drift) ¹	Tahmin adımı (prediction) ¹
GNSS/HD Harita	Mutlak konum (x, y) / düşük frekans, mutlak doğruluk ¹	Düzeltilme adımı (correction), drifti sıfırlama ¹

UKF/EKF algoritmaları durum tahmini için dinamik modeller (örneğin CTRV, CTRA) kullanır ve doğrusal olmayan dinamik ile ölçüm modellerini yönetebilir.¹

6 Algılama (Perception) ve Çevre Modellemesi

6.1 Algılama Modülünün Rolü ve Sensörler

Algılama modülü, otonom sistemin çevreyi anlamasını sağlayan ilk ve en kritik bileşendir. Ham sensör verilerini işleyerek çevresel durumun semantik ve geometrik bir temsili oluşturur.¹

Algılama Modülünün Temel İşlevleri:

- Nesne tespiti ve takibi: Diğer araçları, yayaları, bisikletlileri ve diğer engelleri belirlemek ve 3D konumlarını izlemek.
- Şerit ve yol tespiti: Aracın kendi şeridini ve genel sürüş alanını (freespace) belirlemek.¹
- Çevre modelleme: Yol işaretlerinin, trafik ışıklarının ve çevresel kısıtlamaların semantik içeriğini çıkarmak.¹

Temel Sensörlerin Sağladığı Bilgiler:¹

Sensör	Algılama Yöntemi	Sağladığı Temel Bilgi	Avantajı/Kısıtlaması
Kamera	Pasif (RGB)	Semantik içerik (sınıf, renk, işaret anlamı), zengin doku	Ucuz, yüksek çözünürlüklü; aydınlatmaya hassas.
LiDAR	Aktif (lazer)	Yüksek doğruluklu, yoğun 3D geometrik nokta bulutu	Doğrudan mesafe ölçümü; yüksek maliyet.
Radar	Aktif (radyo)	Yüksek doğruluklu radyal hız (Doppler etkisi) ve menzil	Kötü hava koşullarına (yağmur, sis) dirençli; düşük çözünürlük.

Sensör füzyonu, farklı sensörlerin tamamlayıcı bilgilerini birleştirerek (örneğin LiDAR'ın geometrik doğruluğu ile Kameranin semantik zenginliğini) güvenilirliği artırmayı amaçlar.

6.2 Planlama ve Kontrol Alt Sistemlerinin Ayrımı

Modüler mimaride, Planlama ve Kontrol modülleri, çıktılarının soyutluk seviyesi ve çalışma frekansı açısından belirgin şekilde ayrılmıştır.¹

Planlama ve Kontrol Alt Sistemlerinin Farkları

Sistem	Rol/Seviye	Çıktı Türü / Zaman Ölçeği (Frekans)
Planlama (Planning)	Taktiksel, yüksek seviye	Geometrik yörünge noktaları (waypoints), hız profili; orta-uzun menzil (10–50 Hz) ¹
Kontrol (Control)	Operasyonel, düşük seviye	Fiziksel aktüatör komutları (direksiyon açısı, tork/basınç); kısa menzil (100 Hz ve üzeri) ¹

Planlama modülü, çevresel durumu analiz ederek ne yapılacağına (şerit değiştirme, hızlanma) karar verir. Bu kararlar, geometrik yörüngeler olarak sunulur. Kontrol modülü ise, bu yörüngenin nasıl uygulanacağını belirler. Kontrol, aracın dinamik performansını ve stabilitesini garanti altına almak için milisaniyeler mertebesinde (100 Hz ve üzeri) yüksek frekansta çalışmak zorundadır.¹ Bu yüksek frekans gereksinimi, MPC gibi karmaşık kontrolcülere yönelik yüksek hesaplama gücü ihtiyacını pekiştirmektedir.

6.3 Yol ve Şerit Tespiti

Yol ve şerit tespiti, aracın bir haritaya ihtiyaç duymadan lokal olarak kendini konumlandırması ve navigasyon yapması için esastır.¹

Temsiller ve Zorluklar: Şerit işaretleri tespiti, genellikle görüntü gradyanları kullanılarak yapılır. Tespit edilen şerit işaretlerini bir şerit merkez çizgisine dönüştürmek, genellikle Ters Perspektif Haritalama (IPM) gerektirir. IPM, düz yol yüzeyi varsayımı altında, 2D görüntü piksellerini kuş bakışı (BEV) 3D koordinatlara dönüştürür.¹

Şerit Takibi: Tek kare tabanlı tespitlerin gürültüsünü gidermek ve süreklilik sağlamak için, şerit parametreleri (genişlik, eğrilik) Kalman Filtresi (KF) gibi durum tahmin algoritmaları kullanılarak zaman içinde izlenir. Kısa eğrilik değişimlerini (düz yoldan viraja geçiş) modellemek için dairesel veya Klotoid modelleri kullanılır.¹

6.4 3D Rekonstrüksiyon ve Hareket Analizi

Çevrenin geometrisini ve dinamiklerini anlamak, otonom sürüş için zorunludur.

Stereo Eşleştirme ve Derinlik: Stereo kamera çiftleri, insan gözüne benzer şekilde, iki görüntü arasındaki disparity (farklılık) miktarını ölçerek derinlik haritası oluşturur. Derinlik, disparity ile ters orantılıdır.¹ Uzak noktalarda (yüksek derinlik), ölçümdeki hata karesel olarak artar, bu da stereo vizyonun uzak menzildeki doğruluğunu sınırlar.¹

Optik ve Sahne Akışı:

- **Optik Akış (Optical Flow):** İki ardışık görüntü arasındaki piksel hareketinin 2D vektör alanını tahmin eder.¹

- **Sahne Akışı (Scene Flow):** Her bir 3D yüzey noktasının anlık 3D hız vektörünü tahmin eder.¹

Optik akışın aksine, Sahne Akışı, diğer araçların, yayaların ve engellerin 3D uzaydaki gerçek hareketini tahmin ederek Planlama modülüne hayati bilgi sağlar. Genellikle ardışık stereo görüntü çiftlerinden (toplam 4 görüntü) hesaplanır.

Nesne Tespiti (Object Detection): Görüntü (2D) veya LiDAR nokta bulutları (3D) üzerinde nesnelerin türünü ve konumunu belirler. 3D nesne tespiti, aracın sadece konumunu değil, 3D boyutunu, yönelimini ve uzaklığını da tahmin eder. LiDAR, uzak mesafelerde bile yüksek 3D doğruluk sağlarken, Kamera semantik zenginlik sunar; bu nedenle her iki modaliteyi birleştiren multi-modal füzyon yöntemleri en iyi sonuçları vermektedir.¹

7 Sonuç

Bu kapsamlı çalışma, otonom sürüş sistemlerinin Mimari (Modüler vs. E2E), Kontrol Teorisi (PP vs. MPC) ve Makine Öğrenimi (IL vs. RL, CNN) temelini oluşturan kritik kavramları derinlemesine incelemiştir. Ortaya çıkan bulgular ve uygulamalı örnek çözümler, sistem tasarımında karşılaşılan temel ödünleşimleri netleştirmektedir.

Temel Çıkarımlar

Güvenlik ve Mimarinin İlişkisi: Modüler mimari, yüksek maliyetine ve HD harita bağımlılığına rağmen,¹ şeffaf ara temsilleri sayesinde güvenlik açısından kritik doğrulama ve onaylama (V&V) süreçlerini desteklediği için endüstriyel dağıtım için hâlâ baskın tercihtir. E2E sistemlerin “kara kutu” doğası, regülasyonel uyumu zorlaştırmaktadır.

Kontrolün Fiziksel Sınırları Yönetmesi: Klasik kinematik modellerin (örneğin, $v = 8.0$ m/s hızında 1 saniyelik Euler ilerlemesi)¹ yüksek hızda ve büyük zaman adımlarında sayısal kararsızlık göstermesi, MPC gibi ileriye dönük optimizasyon algoritmalarının önemi artırmaktadır. MPC, araç dinamiği kısıtlamalarını doğrudan formülasyona dahil ederek, aracın fiziksel güvenlik sınırları içerisinde kalmasını (safety envelope) garanti eder.¹

Makine Öğreniminde Dağılım Tutarlılığı: Taklitçi öğrenmedeki Kovaryat Kayması problemi, modelin eğitim dağılımında görmediği “patolojik” durumlarla başa çıkamamasına neden olur. DAgger algoritması, ajanın hatalı yörüngelerindeki kurtarma eylemlerini uzman sorgulama yoluyla toplayarak bu soruna çözüm sunar ve modelin genelleme yeteneğini radikal bir şekilde artırır.¹

Verimlilik ve Hesaplama Mimarisi: CNN’lerin ağırlık paylaşımı sayesinde, görsel görevlerde eşdeğer FC katmanlarına göre yaklaşık 96 kat daha az parametre kullanması,¹ otonom sistemlerde model kapasitesini kontrol altında tutarak aşırı öğrenmeyi önlerken, gerçek zamanlı çıkarım için gereken hesaplama verimliliğini sağlamaktadır.

Otonom sürüş alanında devam eden ilerlemeler, Algılama ve Kontrol sistemlerinin birleştirilmesinde (DDPG gibi sürekli kontrol sağlayan RL yöntemleri ve Direct Perception gibi melez mimarilerle) yatmaktadır. Bu ilerlemeler, güvenilirliği artırmak ve güvenlik sertifikasyonunu kolaylaştırmak amacıyla şeffaflık, verimlilik ve fiziksel kısıt optimizasyonunu bir arada sağlamayı hedeflemektedir.