

# Modele Dayalı Öngörülü Ağ Bağlantılı Kontrol Sistemi

A. Teoman Naskalı<sup>1</sup>, Ahmet Onat<sup>2</sup>, Ozan Mutluer<sup>3</sup>

Sabancı Üniversitesi Mekatronik  
MDBF Orhanlı Tuzla 34956, İstanbul

<sup>1</sup>teoman@su.sabanciuniv.edu

<sup>2</sup>onat@sabanciuniv.edu

<sup>3</sup>ozanmutluer@su.sabanciuniv.edu

## Özetçe

Ağ bağlantılı kontrol sistemlerinin endüstriyel alandaki ihtiyaçları karşılayan çeşitli avantajları vardır. Uygulamalar karmaşıklaştıkça ağ bağlantılı kontrol sistemlerinin kullanımının kaçınılmaz hale gelmesi beklenmektedir. Ancak haberleşme ağının neden olduğu belirsiz gecikmeler ve veri kayıpları, çevrim dinamiklerini olumsuz etkilemekte ve kararsızlıklara sebep olabilmektedir.

Bu çalışmada veri gecikmesi ve kaybı ile algılayıcı güdültüsü gibi ideal olmayan durumlarda da çalışabilen bir ağ bağlantılı kontrol sistem mimarisi tanıtılacaktır. Yapı olarak, kontrol edilen sistemin bir modelinin, kontrolörün de içinde bulundurulması sayesinde haberleşme ağının neden olduğu kayıplar ve gecikmelerin kompanze edilmesi sağlanmaktadır. Model sayesinde, öngörülmuş kontrol çıktıları hesaplanıp haberleşme gerçekleşmediği durumlarda sistemin bunlarla kontrolü sayesinde yüksek derecede veri kayıplarında bile kararsızlığın önlenmesi amaçlanmaktadır. Önerilen yöntemde kontrol edilen sistemin durumu ile kontrolör içindeki modelin durumunun eşleştirilmesi önemli bir problem haline gelmektedir. Bu yapı bilgisayar, kontrol ve haberleşme dallarının özelliklerini kullanarak her birinin eksikliğini tamamlamaya yönelik olup, çeşitli kontrol metodlarıyla kullanılmaya açıktır.

Önerilen Modele Dayalı Öngörülü Ağ Bağlantılı Kontrol Sistemi (MODOAKOS) benzetim yolu ile bir doğru akım motorunun kontrolüne uygulanmıştır. Normal ağ bağlantılı kontrol sistemin kararlılığını bozucu gecikme ve kayıplar varken bile önerilen sistem altında kontrol uygulandığında kararlı çalışma bozulmamış ancak referans girişine olan cevabın geciktiği gözlenmiştir.

## 1. Giriş

Son yıllarda sayısal kontrol sistemlerinin önemi artmış ve sıradan uygulamalarda bile kullanılmaya başlanmıştır. Sayısal yaklaşım sayesinde kontrolörün karar verme yetisi artıp sistemde değişiklikler yapılması da kolayca mümkün olur. Aynı zamanda standart parçaların değişik işlere uygulanması sayesinde maliyetin düşmesi ve uygulamaların daha kolay yapılabilmesi mümkün olur.

Ağ bağlantılı kontrol sistemlerinde (ABKS) kontrol çevrimi bir haberleşme ağı üzerinden kapatılır. Kontrolör ve kontrol edilen sistem fiziksel olarak birbirinden ayrıdır. Bu sistemlerde

algılayıcı ve eyleyiciler de aynı ağa bağlı ve bir miktar hesap kapasitesi olan birer bilgisayar düğümleridir.

Bir ABKSde algılayıcı bilgisayar düğümleri kontrol edilen sistemin çıkışlarını sabit aralıklarla ölçüp ağ üzerinden kontrolöre göndermekle yükümlüdürler. Eyleyici düğümler de kontrolörden alınan kontrol sinyallerini uygun eyleyicilerle kontrol edilen sisteme uygularlar. Kontrolör düğümleri ise algılayıcı düğümlerden aldıkları bilgiye ve bir kontrol algoritmasına dayanarak hesapladıkları kontrol çıktısını, kontrol edilen sisteme uygulanmak üzere eyleyici düğüme gönderirler.

Ancak tasarımın karmaşıklığı ve haberleşmedeki gecikmeler ABKSnin dezavantajlarıdır. Belirsiz gecikme ve kayıplara neden olan haberleşme ağının kontrol çevrimine dahil edilmesiyle ABKSnin analiz ve tasarım karmaşıklığı artar; bunların kontrol üzerindeki etkilerinin bulunması gereklidir.

Bir ABKSde, haberleşme protokolüne bağlı olan ve genelde sabit ve hatta sınırlı olmayan üç temel gecikme düşünülebilir: Algılayıcı ve kontrolör düğümleri arasında örnekleme anı  $t_k$  sırasında olan haberleşme gecikmesi:  $\tau_{sc}(t_k)$ , aynı örnekleme anı sırasında kontrolör düğümündeki hesaplama gecikmesi:  $\tau_c(t_k)$  ve de kontrolör ve eyleyici birimleri arasında oluşan haberleşme gecikmesi:  $\tau_{ca}(t_k)$ . Bu gecikmeler ağdaki haberleşme yükü, mesajların öncelikleri ve elektriksel güdültü gibi etkilere bağlıdır.

Algılayıcı ve eyleyicinin içindeki işlemlerden dolayı olan gecikmeler de  $\tau_s(t_k)$  ve  $\tau_a(t_k)$  şeklinde gösterilebilir, ancak bunlar  $\tau_{sc}(t_k)$  ve  $\tau_{ca}(t_k)$  içine katılmış olarak düşünülebilir.

Algılamadan eyleyici çıkışına kadarki gecikmeler de yukarıdaki gecikmelerin toplamı olarak gösterilebilir:

$$\tau(t_k) = \tau_{sc}(t_k) + \tau_c(t_k) + \tau_{ca}(t_k) \quad (1)$$

## 2. Genel Bilgi

Ağ bağlantılı kontrol sistemleri konusunda yapılan çalışmaların bir kısmı kararlılığı iyileştirmeye yöneliktir [1, 2, 3]. Dead-band alanında yapılan çalışmalar [4], tekrarlanan bilgilerin gönderilmesini sınırlayarak yapılması gereken haberleşmenin miktarını azaltmaya yöneliktir [5]. Ancak bu çalışmalarda haberleşmenin kayıpsız olduğu varsayılmaktadır. Ohnishi tarafından önerilen kazanç adaptasyonu [6] ve ağ izleyicileri [7] yöntemleri haberleşme ağının durumunu takip edip gecikmenin kontrol algoritması üzerindeki etkisini karşılayacak şekilde kazanç değerini değiştirecek şekilde çalışmaktadır.

Ancak ağdaki gecikmelerin simetrik ve değişimlerinin yavaş olduğunu varsaymaktadır. Gecikme hakkında bir ön bilgi olduğu varsayılmaktadır.

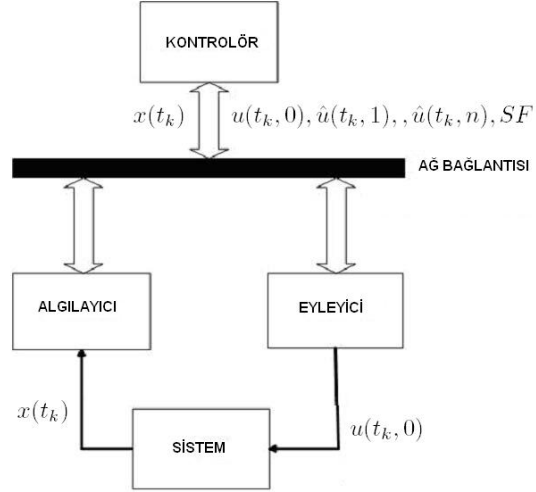
Benzer durumlarda model öngörülü kontrol (model predictive control, MPC) da kullanılabilir [8, 9] ama bunlar kontrolör ve eyleyici arasında doğrudan bağlantı olduğunu varsaydıklarından tam anlamıyla ABKS oldukları söylenemez. Bunun nedeni algılayıcıdan kontrolöre olan bağlantıda bir aksaklık olursa, öngörülere temel olacak bilginin ortadan kalkmasıdır. Aynı zamanda referans sinyali için bir ön bilgi olduğu da varsayılmaktadır.

Bu problemleri çözebilmek için bu çalışmada basit bir ABKS'nin performansını değişen gecikmeler ve kayıplar altında iyileştiren bir yöntem önerilmektedir. Mimarinin temel ABKS mimarisi olduğu ve doğrudan bağlantıların bulunmadığı varsayıldığından bu yöntem var olan ABKSLere de uygulanabilir. Gerçek uygulamalarda mümkün olmayacağından, referans sinyalinin önceden bilindiği varsayımı burada da yapılmamıştır.

### 3. Modele Dayalı Öngörülü Ağ Bağlantılı Kontrol Sistemi

Bu çalışmada ağ bağlantılı kontrol sisteminin veri kayıpları altında kararlılığının bozulmasını geciktirmek için yeni bir mimari önerilmiştir. Bunu başarabilmek için kontrol edilen sistemin bir modeli kontrolörün içinde bulundurulmakta ve sisteme olan kontrol çıkışının şimdiki ve gelecekteki birkaç örnekleme zamanı için öngörülen değerleri hesaplanmaktadır. Daha sonra tüm bu değerler eyleyici düğümüne her örnekleme anında bir kere olmak üzere yollanır ve bunların ilki sisteme uygulanır. Kontrolör düğümünden eyleyici düğümüne bilgi ulaşmadığı takdirde, daha önceden yollanan tahmini değerler eyleyici düğüm tarafından kontrol edilen sisteme uygulanır. Sistemin adı da buradan gelmektedir: Modele Dayalı Öngörülü Ağ Bağlantılı Kontrol Sistemi (MODOAKOS). Genel mimarisi bAKOS ile aynı olan MODOAKOS Şekil 1'de görülmektedir.

Önerilen kontrol sistemi beş kısımdan oluşmaktadır: *Haberleşme ağı*: Bu kısımda paket kaybı ve gecikme tamamen rastlantısal oluşmaktadır; ağlardaki kayıpların korelasyonlu olmalarına karşın benzetim için tamamen rastlantısal bir davranış benimsenmiştir. *Algılayıcı düğüm*: Bu kısmın temel amacı algılama ve sistemin  $x(t_k)$  durumlarını kontrolöre yollamaktır. *Kontrolör düğümü*: Bu kısım kontrol çıkışları olan  $u(t_k, i)$  veya  $\hat{u}(t_k, i)$ yi içeren kontrol paketi  $Pt(t_k)$ yi oluşturup eyleyiciye gönderir. Burada  $u(t_k, i)$  algılayıcıdan gelen veri üzerine hesaplanan kontrol çıkışını,  $\hat{u}(t_k, i)$  ise algılayıcıdan veri gelmemesi durumunda veya ileriye yönelik öngörülen kontrol çıkışını gösterir;  $t_k$  paketin oluşturulduğu örnekleme zamanı ve  $i$ ,  $t_k$  anında ileriye yönelik olarak öngörülen kontrol çıkışının sıra numarasıdır;  $u(t_k, i)$ ,  $t_k$  anından sonraki  $i$ nci örnekleme anında uygulanması öngörülen çıkıştır. *Kontrol edilen sistemin modeli*: Bu model algılayıcı düğümünden kontrolör düğümüne kontrol edilen sistemin durum değişkenleri  $x(t_k)$  ulaştığı zaman ilklendirilir. Bunun nedeni model durumları ile gerçek sistem arasında eşlik sağlamaktır. Haberleşme geçkeleşmediğinde ise, var olan durumdan bir sonraki örnekleme zamanına iterasyon yapılarak yeni durum bulunur. *Eyleyici düğüm*: Bu düğüm kontrolör düğümü



Şekil 1: Modele dayalı öngörülü ağ bağlantılı kontrol sistem yapısı

tarafından üretilip gönderilmiş olan kontrol çıkışı  $u$  veya öngörülen kontrol çıkışını  $\hat{u}$  her örnekleme zamanında sisteme uygular. Her bilgisayar düğümünde periyodik işler çalışır. Geç paketler kayıp varsayılır.

Algılayıcı ve eyleyici arasındaki paket kaybı kontrolör düğümündeki öngörü ile telafi edilirken, kontrolör düğümü ve eyleyici düğümü arasındaki paket kaybı da eyleyici düğümünde bir seçme algoritması ve öngörülüş kontrol çıkışlarının uygulanması ile telafi edilir.

Kontrol algoritmasının sistem durumunun  $x(t_k)$  türev değerlerini ilgilendiren kısımları algılayıcı düğümünde hesaplanır çünkü ağdaki paket kaybı nedeniyle türev değerlerinin kontrolör düğümündeki sürekliliği garantilenemez. Daha sonra sistem durumu ve türev değerleri bir veri paketi içine yerleştirilir ve kontrolör düğümüne yollanır.

Kontrolör düğümü, algılayıcı düğümünden bu verileri alıp modelin başlangıç şartları olarak kullanır. Eğer bir veri kaybı oluşmuşsa kontrolör düğümü sistem modeli  $\hat{P}$  ve  $x(t_{k-1})$  terimlerini kullanarak  $t_k$  zamanı için öngörülüş sistem durumu  $\hat{x}_{t_k-1}(t_k)$  terimini oluşturur. Bu öngörüler yapılırken kontrolör düğümünden eyleyici düğümüne olan bağlantının bir önceki zaman aralığında kırılmadığı varsayılır. Kırılmış ise, ölçüm değerleri yerine öngörü değerleri kullanılır.

Kontrol çıkışlarının hesaplanmasında ölçülen durum mu yoksa öngörülen durum mu kullanıldığı önemlidir. Çünkü eğer öngörülen durum kullanılıyorsa kontrolör eyleyici arasında veri kaybı olması ihtimali nedeniyle kontrol edilen sisteme uygulanmış olan son kontrol çıktısının ne olduğu belirsizdir. Örneğin eğer bir kontrol paketi kaybolursa, kontrolör düğümü  $u(t_k, 0)$ in sisteme uygulandığını varsayar ve hesaplamalarını buna göre yapabilir, ama aslında  $\hat{u}(t_{k-1}, 1)$  uygulanmış olabilir. Bu, sistem ve modeli arasındaki uyumluluğun kaybolmasına yol açar. Bu nedenle kontrol çıkışının hesabında hangi verinin kullanıldığı algılayıcı bayrağı  $SF$  adı altında kontrol paketine eklenir. Kontrol çıktılarını algılayıcıdan gelen sistem du-

rumu kullanıldıysa bire tersi durumda  $SF$  sıfıra çekilir.

Kontrolörde  $x(t_k)$  veya  $\hat{x}_{t_{k-1}(t_k)}$ ye bir kontrol yöntemi uygulanır ve kontrol çıkışı  $u(t_k, 0)$  elde edilir. Takiben  $u(t_k, 0)$ ,  $\hat{P}$  ve  $x(t_k)$ a uygulanır ve  $t_{k+1}$  için öngörülen durumlar  $\hat{x}_{t_k}(t_{k+1})$  bulunur ve bununla tahmin edilen bir sonraki kontrol çıkışı  $\hat{u}(t_k, 1)$  hesaplanır. Bu işlem  $n$  defa  $\hat{u}(t_k, n)$  elde edilene dek tekrarlanır.  $Pt(t_k)$  paketi  $t_k$  zamanında hazırlanır ve  $n + 1$  kontrol çıkışı ile bir algılayıcı bayrağından oluşur:  $u(t_k, 0), \hat{u}(t_k, 1), \dots, \hat{u}(t_k, n), SF$ . Paket ağ bağlantısı üzerinden eyleyici düğüme yollar. Şekil 1 ağ paketlerinin içeriğini ve AKOS içindeki ilişkilerini anlatır. Tahmin sayısı  $n$ , şu özelliklere göre seçilir: Modelin doğruluğu, paket boyutunun ağ bağlantısını fazla yüklememesi ve eldeki işlem gücü.

Eyleyici düğüm kontrol düğümünden gelen sinyalleri sisteme uygular. Ama verilen bir örnekleme zamanı için elde birden fazla kontrol çıkışı olduğundan bir seçim yapılması gerekmektedir ve bu seçim sistem ve modelin durumları arasındaki uyumluluk baz alınarak yapılır. Bu uyumluluk eyleyici tarafından paketin zamanında ulaşması ve paketin içindeki  $SF$  bayrağı kullanılarak algılanır. Eyleyici düğümünün iki durumu vardır, uyumlu kip ve kesilmiş kip.

**Uyumlu Kip:** Uyumlu kip, model durumunun sistem durumu ile eş olduğunun varsayıldığını belirtir. Eyleyici düğümü uyumlu kipteyken kontrolör düğümünden  $t_k$  zamanında bir kontrol paketi alırsa bu paketten gelen kontrol çıkışı sisteme uygulanır. Algılayıcı bayrağı bir ise bu değer  $u(t_k, 0)$ , sıfır ise  $\hat{u}(t_k, 0)$ dir. Algılayıcı bayrağının sıfır seviyesi uyumlu kipteyken dikkate alınmaz. Ancak uyumlu kipteyken paket kaybı olmuşsa sisteme  $\hat{u}(t_{k-1}, 1)$  uygulanır ve eyleyicinin durumu kesilmiş kipe çevrilir. Eyleyici senkronize moda sadece bir seviyesinde  $SF$  içeren paket ulaştığı zaman döner.

**Kesilmiş kip:** Eğer eyleyici kesilmiş kipte iken sıfır seviyesinde  $SF$  içeren bir kontrol paketi gelirse, paket değerlendirilmez; eyleyici kesilmiş kipe girmeden önce gelen en son paketteki öngörülmiş kontrol çıkışları birbiri ardına kullanılır. Paketlerin reddedilmesinin nedeni ise kontrol çıkışları üretilirken kontrolör eyleyici bağlantısında paket kaybı olmadığına varsayılmasıdır. Ama bir önceki paket kesinlikle kaybolduğundan en son ulaşan sıfır  $SF$ li paketteki kontrol çıkışları doğru olmayabilir. Bir  $SF$ li paket gelince uyumlu kipe geçilir.

## 4. Sonuçlar

Teklif edilen yöntem, ağ bağlantılı gerçek zamanlı sistemleri test etmek için hazırlanmış bir Matlab eki olan TrueTime [10, 11] kullanılarak bilgisayar benzetimleri ile sınanmıştır. TrueTime, bilgisayar sistemlerinin benzetimini komut seviyesinde, haberleşme ağlarının benzetimini ise veri aktarım seviyesinde yapar. Bu nedenle benzetimlerin gerçek hayata yakın olduğu söylenebilir.

Kontrol edilen sistem olarak, aşağıdaki transfer fonksiyonu ile tanımlanan DC motor kullanılmıştır.

$$G(s) = \frac{1000}{s(s+1)} \quad (2)$$

Aşağıdaki denklemler kullanılarak bir PD kontrolör yapılmıştır:

$$KP(t_k) = K(r(t_k) - y(t_k)) \quad (3)$$

$$KD(t_k) = \alpha_d KD(t_k - 1) + \beta_d (y(t_k - 1) - y(t_k)) \quad (4)$$

$$\alpha_d = \frac{T_d}{N_h + T_d} \quad (5)$$

$$\beta_d = \frac{NKT_d}{N_h + T_d} \quad (6)$$

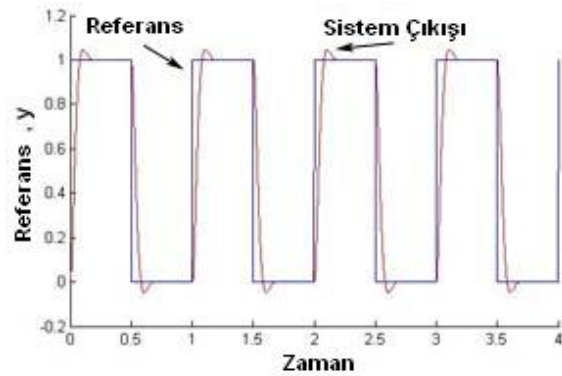
$$u(t_k) = KP(t_k) + KD(t_k) \quad (7)$$

Burada,  $r(t_k)$ ,  $y(t_k)$ ,  $u(t_k)$  sırasıyla referans, sistem çıkışı ve kontrol çıkışı;  $KP(t_k)$ ,  $KD(t_k)$  ise kontrolün orantısal ve türev bileşenleri,  $K$  orantısal kazanç ve  $t_k$ , örnekleme zamanıdır. Ayrıca  $N$ ,  $N_h$ ,  $\alpha_d$  ve  $\beta_d$  sabit katsayılarıdır. Sistem çıkışı  $y(t_k)$ ın değeri  $Hx(t_k)$  ile bulunur. Burada  $H$  sistem çıkış matrisi ve  $x(t_k)$  sistemin  $t_k$  durumudur.

Önerilen kontrol sistemi bir basit Ağ Bağlantılı Kontrol Sistemi(bAKOS) ile karşılaştırılmıştır. Basit AKOSda sadece algılayıcı düğüm periyodik olarak girişi örnekler. Kontrolör ve eyleyici düğümleri ise olay güdümlü çalışır ve sadece ağdan kontrol çıkışını hesaplamak ve bunu sisteme uygulama amaçlı bir mesaj geldiği zaman işlerler. Performans ölçütü olarak referans ve sistem çıkışı arasındaki ortalama karesel hata (RMS; OKH)kullanılır.

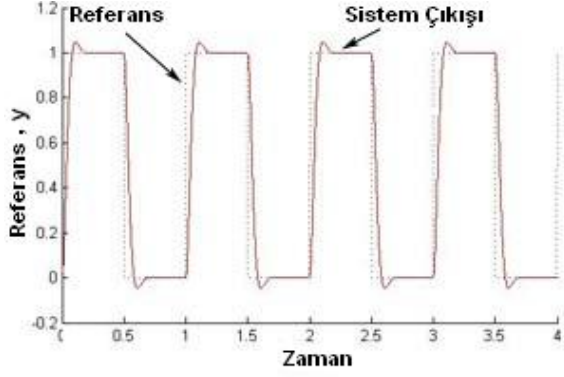
### 4.1. Modele Dayalı Öngörülü Ağ Bağlantılı Kontrol Uygulaması

Denklem 2deki sürekli zaman sistem modelinin durum uzayı betimlemesi 0.01 saniyelik bir örnekleme zamanı ile ayrıştırılır. Ağ bağlantısının düğümler arasında veri paketlerini göndermek için yeterli zamanının olması için kontrolör düğümü algılayıcı düğümünden 0.001 saniye sonra ve eyleyici düğümü kontrolör düğümünden 0.001 saniye sonra işlemeye başlar. Tezin doğruluğunu görebilmemiz için benzetimler mükemmel bir model ile yapılmıştır. Daha ileride mükemmel olmayan modellerin etkisi de incelenecektir. İdeal koşullar ve paket kaybı olmadığı durumlarda MODOAKOS ve bAKOS yaklaşık aynı sonuçları vermektedirler: Şekiller 2 ve 3.

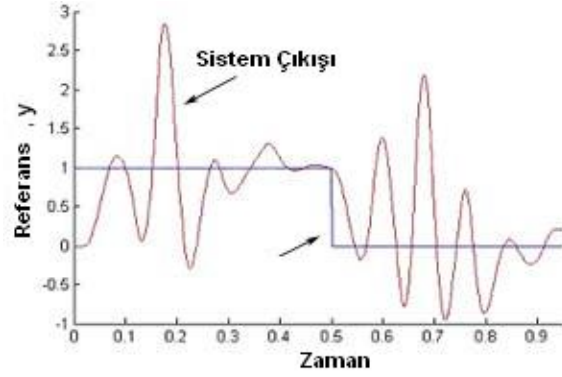


Şekil 2: Basit AKOS, ideal koşullar OKH: 0.2324

Paket kaybı arttıkça performanstaki bozulmanın arttığı da gözlemlenmektedir. Ancak kontrol kalitesindeki düşmenin nedeni iki sistem için farklıdır. Basit AKOSdaki OKH artışının sebebi Şekil 6da görüldüğü gibi kararlılık kaybıdır ve nedeni paket gecikmelerinin yarattığı çevrim gecikmesidir. Diğer taraftan MODOAKOSdaki OKH artışının nedeni paketler

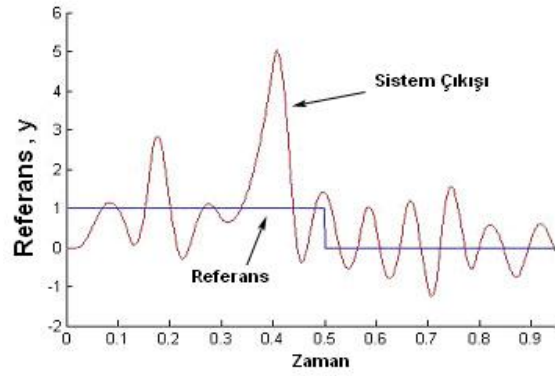


Şekil 3: MODOAKOS, ideal koşullar OKH: 0.23252

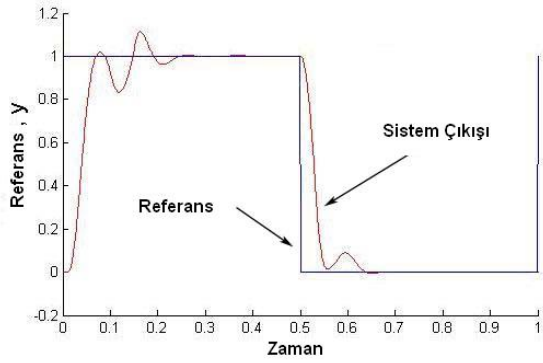


Şekil 5: Basit AKOS %20 paket kaybı OKH:0.66509

ulaşmasa bile önceden hesaplanmış bir kontrol çıkışının sisteme uygulanmasıdır. Bu durumda referans değeri değişse bile sistem eski bir referansa doğru kontrol edilebilmektedir. Referanstaki bu bozulma Şekil 9da açıkça görülebilir. Bu Şekilde referans sinyal ve sistem çıkışı ile kontrol paketinden kullanılan öngörülen kontrol çıkış değeri sıra numarası  $i$  birlikte gösterilmiştir. Daha iyi görülmesi için öngörü değeri -4 kadar kaydırılmıştır. Eyleyici senkronize moda tekrar girdiğinde sistemin referansı yakaladığı ve eyleyici düğümünün referansa gitmek için senkronize moda kalmasına gerek olmadığı görülebilir. Sistemimizin benzer araştırmalarda olduğu gibi [9] önceden bilinen bir referans sinyaline sahip olmadığına dikkat edilmelidir. Basit AKOSun performansı MODOAKOSdan daha hızlı bir şekilde bozulur. Aşırı paket kaybında bile bAKOSda kararlılık bozulurken MODOAKOSda kararlılığın bozulmadığı görülmektedir(Şekil 9). Performanstaki düşüş şu şekillerde görülebilir: 4- 6.



Şekil 6: Basit AKOS %30 paket kaybı OKH:1.023



Şekil 4: Basit AKOS %10 paket kaybı OKH: 0.23876

Sistemin performansını gürültü altında incelemek için, eyleyici düğümün sisteme uyguladığı kontrol çıkışına sınırlı bant genişlikli beyaz gürültü eklenir. Bunun sistemde kararlılığı bozucu başka bir etkisi daha vardır: Sisteme koyulan gerçek kontrol sinyali hesaplanandan farklı olduğu için, sistem ve model durumlarının uyumluluğu tehlikeye girer. Sinyale eklenen gürültünün gücü sinyalin dinamik erimi ile orantılıdır. Gürültünün gücü ve sinyal arasında bir bağlantı sağlamak için, gürültünün gücü ideal ağ bağlantısı ile üretilen kontrol çıkışının

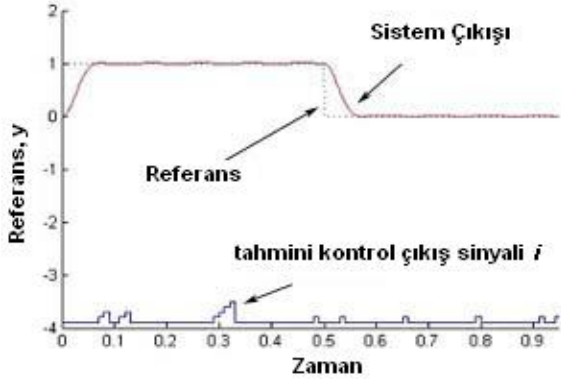
OKH değerinin bir oranı olarak belirtilmiştir.

$$u_n(t_k) = u(t_k) + n(t_k) \quad (8)$$

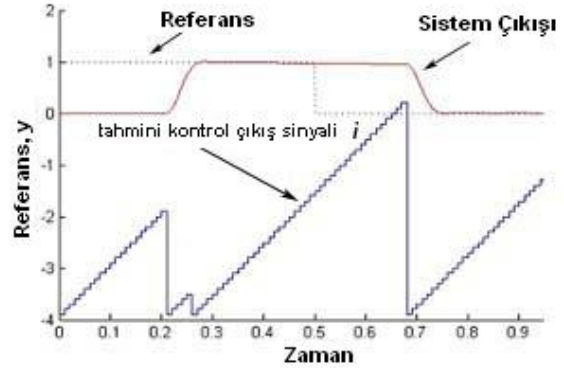
Burada  $u(t_k)$  kontrol çıkışı,  $n(t_k)$  gürültü terimi,  $u_n(t_k)$  gürültü eklenmiş kontrol çıkışı terimidir. Gürültü fonksiyonu  $n(t_k) = f(u_{RMS} * C_n)$  sınırlı bant genişlikli beyaz gürültü üretir. Gürültünün gücü bir parametredir. Sabit katsayı  $C_n$  gürültüyü gösterir ve  $u_{RMS}$  ise istatistiksel olarak belirlenen  $u$  nun OKH değeridir. Bu gürültü yarı rastlantısal olsa da değişik sistemlerin aynı koşullarda benzetim yapılabilmesini sağlamak için bu çalışmada aynı başlangıç değeri kullanılmıştır. Şekil 10da gürültü, hesaplanmış kontrol çıkışı ve kontrol çıkışı görülebilir.

Kayıpsız ideal koşullarda bAKOSun performansı MODOAKOS ile neredeyse aynıdır. Ancak paket kaybı yüzünden oluşan hatanın gürültü yüzünden oluşan hatayı aştığı ve sistemin kararlılığını yitirdiği görülür.

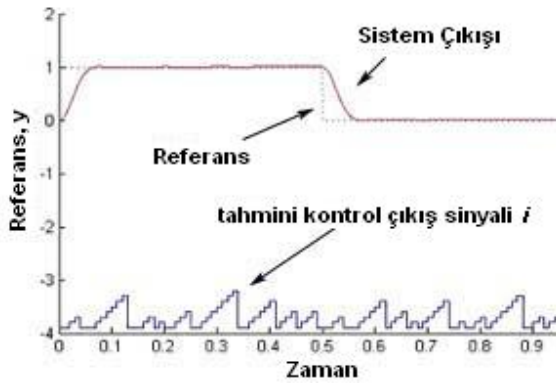
Düşük seviyeli gürültünün %60 paket kaybına kadar kararlılığı bozmadığı görülür. Şekil 11de MODOAKOSun paket kaybı olmayan bir ağdaki performansı ve Şekil 12da % 60 paket kaybı altındaki performansı karşılaştırılabilir. Yüzde 70 paket kaybından sonra gürültünün kontrol kalitesi üzerindeki etkisi görülmeye başlar. Geri besleme gürültünün etkisini azaltsa da, yüksek oranda paket kaybı geri besleme çevriminin kapatılmasını engeller. Bu nedenle paket kaybının artışı sis-



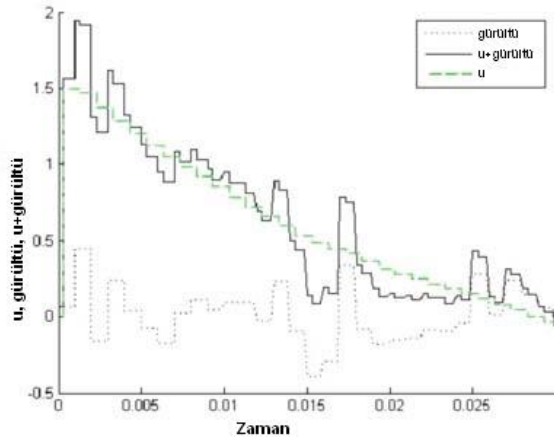
Şekil 7: MODOAKOS %10 paket kaybı OKH: 0.22644



Şekil 9: MODOAKOS %90 paket kaybı OKH: 0.65153



Şekil 8: MODOAKOS %50 paket kaybı OKH: 0.22644



Şekil 10: Gürültü ve  $u$  üzerindeki etkisi

temin gürültüye karşı koyma yetisini azaltır.

Şekil 13de MODOAKOSın 0.001 gürültü ve %70 paket kaybı altındaki performansı ile birlikte en son ulaşan paketin sinyal numarası görülmektedir. Burada kontrol kalitesinin en kötü olduğu zamanda(benzetimin 0,8nci saniyesinde) kontrolör düğümünden eyleyici düğümüne olan uyum 0,06 saniye için kırılr. Bu nedenle sistem 0,06 saniye boyunca açık sistem gibi çalışır. Bunun anlamı eyleyici düğümü kontrolör düğümünden 60 paket karşılamadığıdır. Sistem gürültüye rağmen kararlılığını yitirmemiştir. Bu nedenle tanıttığımız metodun gürültülü ortamda veri kaybına karşı iyileşme sağladığını söyleyebiliriz. Basit AKOS burada böyle bir iyileşme gösteremediği için gösterilmemiştir.

Seğirmenin etkisi ise [12]de incelenmiştir.

## 5. Vargı

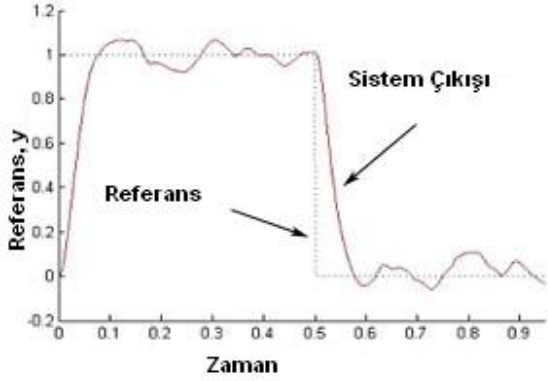
Bu çalışmada yeni bir ağ Bağlantılı kontrol sistemi olan Modele Dayalı Öngörülü Ağ Bağlantılı Kontrol Sistemi (MODOAKOS) tanıtılmıştır. Modern bilgisayarların hızlı işlem yeteneklerini kullanarak gerçek zamanlı olmayan haberleşme ağları üzerinden kontrolü mümkün kılmıştır. Metot algılayıcı ve eyleyici düğümler üzerindeki hesap kapasitesinden yararlanarak kontrol edilen sistem ile kontrolör üzerindeki modelinin durum değişkenlerinin eş tutulabilmesini sağlamaktadır. Kontrolör durumlarının bir kısmı algılayıcı üzerinde işlenerek

veri kaybının ardından normal bağlantıya geçiş sırasında da eşleşmenin sorunsuz olması sağlanmaktadır.

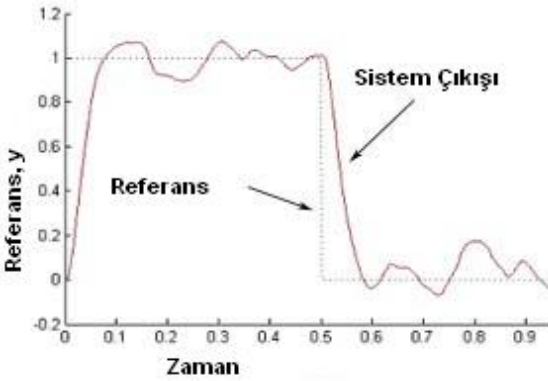
MODOAKOS metodu kontrol edilen sisteme uygulanacak çıktıların öngörülmesinde bir modelden yararlanmakta ve ileriye yönelik belirli sayıda bir dizi kontrol sinyali oluşturulmaktadır. Bu sinyaller eyleyici düğümüne gönderilerek orada koşmakta olan bir durum makinesi sayesinde (kontrol edilen sistem ve modelinin durumları arasındaki eşlik bozulmayacak şekilde) uygulanacak kontrol sinyali seçilir. Kontrolör ve eyleyici düğümleri arasındaki haberleşme bağlantısı koparsa eyleyici düğüm en son ulaşan veri paketindeki öngörölmüş kontrol çıktıları kullanarak paket kaybının etkileri azaltılır. Bu koşullarda eyleyici düğümüne veri ulaşmadığından referanstaki değişiklikler de takip edilemez.

Önerilen metodun çeşitli özellikleri bir DC motor üzerinde yapılan benzeşimlerle incelenmiştir. Veri gecikmesi ve kaybının kararlılık üzerindeki etkilerinin bAKOSa göre daha az olduğu ve referansı takipte yaşanan bir gecikme olarak ortaya çıktığı görülmüştür. Her ne kadar uygulanan adımsal referans sinyalinde önemsiz gibi görünse de sürekli değişen referans sinyallerinde bu etki önemli hale gelebilir. Gürültünün etkisi ise paket kayıpları ve gecikmenin da az olduğu durumlarda düşük olmakta ancak kayıplar arttığında geri besleme çevrimi kesildiğinden, daha etkili hale gelmektedir.

Burada yapılan çalışmada kullanılan sistemden farklı



Şekil 11: MODOAKOS 0.001 gürültü altında %0 paket kaybı OKH: 0.22726



Şekil 12: MODOAKOS 0.001 gürültü altında %60 paket kaybı OKH:0.2421

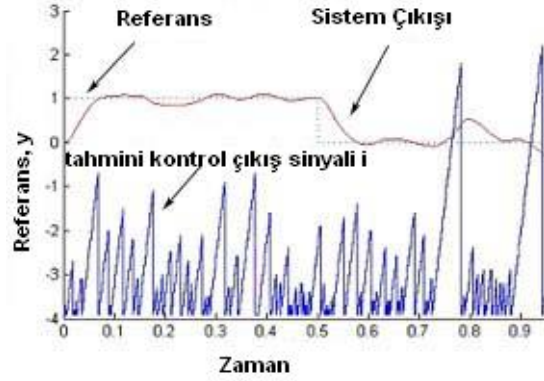
olarak açık çevrim kararsız sistemlerin de incelenmesi planlanmaktadır.

Hali hazırda MODOAKOSun endüstriyel bilgisayarlar ve gerçek zamanlı Linux kullanılarak gerçek bir DC motorun kontrolü için uygulanması çalışmaları sürmektedir.

Bu araştırma, TÜBİTAK tarafından 106E155 numaralı proje kapsamında desteklenmiştir.

## 6. Kaynakça

- [1] M.S. Branicky, S.M. Phillips, and Wei Zhang. Scheduling and feedback co-design for networked control systems. *IEEE Conference On Decision and Control*, 2(41):1211–1217, December 2002.
- [2] M. S. Branicky, V. Liberatore, and S. M. Phillips. Networked control system co-simulation for co-design. *American Control Conference*, 4:3341–3346, June 2003.
- [3] Q Ling and M. Lemmon. Robust performance of soft real-time networked control systems with data dropouts. *IEEE Conference On Decision and Control*, December 2002.
- [4] P Otanez, J. Moyne, and D. Tilbury. Using deadbands to reduce communication in networked control systems. *American Control Conference*, 2000.



Şekil 13: MODOAKOS 0.001 gürültü altında %70 paket kaybı OKH:0.27542

- [5] J. Yook, D. Tilbury, and N. Soparkar. Performance evaluation of distributed control systems with reduced communications. *IEEE Control Systems Magazine*, 21(1):84–99, 2001.
- [6] C. Mo-Yuen and Y. Tipsuwan. Gain adaptation of networked dc motor controllers based on qos variations. *IEEE Transactions on Industrial Electronics*, 50(5):936–943, October 2003.
- [7] K. Natori and K. Onishi. Time delay compensation by communication disturbance observer in bilateral teleoperation systems. *Advanced Motion Control 2006*, pages 218–223, March 2006.
- [8] J.B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20(3):38–52, June 2000.
- [9] G. P. Liu, J. X. Mu, and D. Rees D. Networked predictive control of systems with random communication delay. *UKACC International Conference on Control*, September 2004.
- [10] D. Henriksson, A. Cervin, and K. Årzén. Truetime: Real-time control system simulation with matlab/simulink. *Proceedings of the Nordic MATLAB Conference*, October 2003.
- [11] D. Henriksson, A.Cervin, and K. Arzen. Simulation of control loops under shared computer resources. *Proceedings of the 15th IFAC World Congress on Automatic Control*, July 2002.
- [12] A. Teoman Naskali and A. Onat. Jitter analysis of model based predictive networked control system. *6th WSEAS International Conference on APPLIED COMPUTER SCIENCE*, December 2006.
- [13] J. K. Yook, D. M. Tilbury, H. S. Wong, and N. R. Soparkar. Trading computation for bandwidth: State estimators for reduced communication in distributed control systems. *Proceedings of 2000JUSFA 2000 Japan-USA Symposium on Flexible Automation*, July 2000.
- [14] J. Yook, D. Tilbury, K. Chervela, and N. Soparkar. Decentralized, modular real-time control for machining applications. *American Control Conference*, pages 844–849, 1998.