

**Destek Vektör Makineleriyle Gerçek-Zamanlı Genelleştirilmiş Öngörölü Kontrol**

**Proje No: 106E125**

Doç. Dr. Serdar İPLİKÇİ

ŞUBAT 2010

DENİZLİ

## Önsöz

Doğrusal denetim sistemleri kuramsal olarak belli bir olgunluğa ulaşmış olup endüstride de çok yaygın olarak kullanılmaktadırlar. Ancak, endüstride doğrusal olmayan süreçler de karşımıza çıkmaktadır. Doğrusal olmayan süreçlerin denetiminde her ne kadar doğrusal yaklaşımlar belli bir başarımlar sağlasalar da yetersiz kaldıkları durumlar da ortaya çıkmaktadır ki bu da doğrusal olmayan denetim kuramı fikrini doğurmuştur. Doğrusal olmayan denetim kuramı büyük oranda sistemin matematiksel modelini kullanan analitik yöntemlere dayanır. Ancak, sistemin matematiksel modelinin bulunamadığı veya eksik bulunduğu durumlar da mevcuttur. Bu durumlarda, sistemin giriş-çıkış büyüklüklerinin gözlemlenmesinden elde edilecek verilerle oluşturulacak yeni bir sistem modeli kullanılır. Literatürde, yapay sinir ağları ve bulanık mantık sistemlerinin pek çok türü hatta bunların bileşkeleri gibi esnek işlem (soft computing) araçları bu amaçla kullanılmıştır.

Bu projede ise esnek işlem araçlarına alternatif olarak literatürde yerini alan Destek Vektör Makineleri (Support Vector Machines - SVMs) yaklaşımıyla matematiksel modeli bilinmeyen ve doğrusal olmayan sistemlerin denetimi konusunda bir çalışma yapılmıştır. Bu çalışma sonucunda geliştirilen yöntem gerçek bir sisteme (üç-tanklı sıvı sistemi) başarılı bir şekilde uygulanmıştır.

Bu çalışma, "Destek Vektör Makineleriyle Gerçek-Zamanlı Genelleştirilmiş Öngörülü Kontrol" başlıklı ve 106E125 nolu kariyer projesi kapsamında TÜBİTAK tarafından desteklenmiştir.

## İçindekiler

Tablolar Listesi	4
Şekiller Listesi	5
Özet	7
Abstract	9
1. GİRİŞ	11
2. PROJEDE KULLANILAN SİSTEM	13
2.1 Üç-Tanklı Sıvı Sistemi	13
3. YÖNTEMLER	16
3.1 Çevrim-dışı SISO Denetim	16
3.1.1 Yöntem	16
3.1.2 Deney	24
3.1.2.1 Modelleme Verileri	24
3.1.2.2 SVM Modelinin Elde Edilmesi	28
3.1.2.3 Deneysel Sonuçlar	29
3.1.2.4 Denetim Parametrelerinin Etkileri	31
3.2 Çevrim-içi SISO Denetim	40
3.2.1 Yöntem	40
3.2.2 Deney	51
3.2.2.1 Modelleme Verileri	51
3.2.2.2 SVM Modelinin Elde Edilmesi	51
3.2.2.3 Deneysel Sonuçlar	51
3.2.2.4 Denetim Parametrelerinin Etkileri	55
3.3 Çevrim-dışı MIMO Denetim	61
3.3.1 Yöntem	61
3.3.2 Deney	68
3.3.2.1 Modelleme Verileri	68
3.3.2.2 SVM Modelinin Elde Edilmesi	70
3.3.2.3 Deneysel Sonuçlar	72
3.3.2.4 Denetim Parametrelerinin Etkileri	75
4. SONUÇLAR	77
REFERANSLAR	82

**Tablolar Listesi**

Tablo 2.1.1 sisteme ilişkin deęişkenler ve parametreler.

14

## Şekiller Listesi

Şekil 2.1.1	Üç-tanklı sıvı denetim sistemi.	13
Şekil 3.1.1.	SISO sistem.	16
Şekil 3.1.2.	SVM-Tabanlı GPC şeması.	19
Şekil 3.1.3.	a) $\tau_{\max} = 1$ sn; b) $\tau_{\max} = 100$ sn.	25
Şekil 3.1.4.	Modelleme için toplanan veriler.	26
Şekil 3.1.5.	Alçak Geçiren Filtrenin frekans cevabı.	27
Şekil 3.1.6.	Ölçülen çıkış ve filtrelenmiş çıkış işaretleri.	27
Şekil 3.1.7.	Üç-tanklı sistem için SISO SVM modeli.	28
Şekil 3.1.8.	En iyi $(\sigma, \varepsilon)$ çifti için grid araması.	28
Şekil 3.1.9.	Basamak biçimli referans için deneysel sonuçlar.	30
Şekil 3.1.10.	Sinüs biçimli referans için deneysel sonuçlar.	31
Şekil 3.1.11.	Denetleyicinin farklı parametre değerleri için başarımları.	33
Şekil 3.1.12.	$N_y$ 'nin sistem cevabına olan etkisi.	35
Şekil 3.1.13.	$N_u$ 'nun sistem cevabına olan etkisi.	37
Şekil 3.1.14.	$\lambda$ 'nın sistem cevabına olan etkisi.	39
Şekil 3.2.1.	Çevrim-içi SVM-Tabanlı GPC şeması.	40
Şekil 3.2.2.	Çevrim-içi SVM-Tabanlı GPC algoritması	50
Şekil 3.2.3.	Basamak referans girişi için deney sonuçları.	52
Şekil 3.2.4.	Sinüs referans girişi için deney sonuçları.	53
Şekil 3.2.5.	Sabit referans girişi ve değişken musluk parametreleri için deney sonuçları	55
Şekil 3.2.6.	Sabit sistem parametreleri durumu için basamak referans işareti.	57
Şekil 3.2.7.	Sistem parametreleri sabit iken modelleme parametrelerinin denetim başarımları üzerindeki etkileri.	57
Şekil 3.2.8.	Referans işareti sabit ancak bir sistem parametresi değişiyor iken modelleme parametrelerinin denetim başarımları üzerindeki etkileri.	59
Şekil 3.3.1.	MIMO sistem.	61
Şekil 3.3.2.	MIMO SVM-Tabanlı GPC şeması.	63
Şekil 3.3.3.	a) $\tau_{\max} = 1$ sn; b) $\tau_{\max} = 100$ sn.	69
Şekil 3.3.4.	Modelleme için toplanan veriler.	70
Şekil 3.3.5.	Üç-tanklı sistem için MIMO SVM modeli.	70
Şekil 3.3.6.	En iyi $(\sigma, \varepsilon)$ çifti için grid araması.	71

Şekil 3.3.7.	Basamak biçimli referanslar için deneysel sonuçlar.	73
Şekil 3.3.8.	Sinüs biçimli referanslar için deneysel sonuçlar.	74
Şekil 3.3.9.	Denetleyicinin farklı parametre değerleri için başarımları.	76

## Özet

Bu projede, Destek Vektör Makineleri (Support Vector Machines - SVMs) yaklaşımıyla matematiksel modeli bilinmeyen ve doğrusal olmayan sistemlerin denetimi konusunda bir çalışma yapılmıştır. Denetlenecek sistemin SVM modeli, Genelleştirilmiş Öngörülü Denetim (Generalized Predictive Control - GPC) çerçevesinde kullanılmıştır. Bu çalışma sonucunda geliştirilen yöntemler gerçek bir sisteme (üç-tanklı sıvı sistemi) başarılı bir şekilde uygulanmıştır.

Projede geliştirilen yöntemler çevrim-dışı ve çevrim-içi olmak üzere iki ana gruptan oluşmaktadır: 1) Çevrim-dışı yöntemlerde önce SVM modeli oluşturulmakta daha sonra da bu model denetimde kullanılmaktadır. Bu esnada sistemin parametrelerinin değişmediği varsayılmaktadır. Çevrim-dışı yöntemler bir-girişli bir-çıkışlı (SISO) ve çok-girişli çok-çıkışlı (MIMO) olmak üzere iki ayrı kategoride ele alınmıştır. 2) Çevrim-içi denetimde ise önceden hiçbir veriye ihtiyaç duymadan boş bir SVM modeliyle denetime başlanmakta, denetim ve modelleme işleri senkron bir şekilde yürütülmektedir. bu denetimdeki denetleyici sistemdeki parametre değişimlerine uyarlanabilmektedir.

Çevrim-dışı yöntemlerinde ilk olarak, denetlenecek sistemden belli bir süre boyunca veriler toplanır. Bu aşamada sisteme ilişkin parametrelerin ve veri toplama parametrelerinin uygun olmasına dikkat edilir. Ardından, toplanan bu verilerle bir eğitim ve bir test seti oluşturulur. Eğitim seti SVM modelinin geliştirilmesinde test seti ise sınanmasında kullanılacaktır. Daha sonra, bu setlerle, sisteminin bir SVM modeli oluşturulur ve bu model Genelleştirilmiş Öngörülü Denetim döngüsünde kullanılır. GPC yönteminde, önce bir aday denetim vektörü oluşturulur, ardından denetlenecek sistemin bu vektöre olan ileri yönelik cevabı tahmin edilir, sonra da bu tahmin hatasını azaltacak şekilde aday denetim vektörü en iyilenir. En iyilenmiş denetim vektörünün ilk elemanı sisteme giriş olarak uygulanır. Bu işlem, her örnekleme döngüsü içerisinde tekrarlanır. Benzer işlemler çevrim-içi denetim için de geçerlidir. Tek fark, çevrim-içi denetimde önceden bir SVM modeli elde edilmeyip boş bir SVM modeliyle başlanması ve modelin denetim esnasında kendini geliştirmesidir.

Bu çerçevede SVM modelinin iki farklı rolü vardır. Birincisi, SVM modeli ileri yönelik tahminde kullanılır. İkincisi, aday denetim vektörünün en iyilenmesinde gerekli olan türev bilgisi SVM modelinden elde edilir. Bu bağlamda, sistemin SVM modeli çok kritik bir öneme sahiptir, zaten GPC yaklaşımlarında denetimin başarımı büyük oranda modelin başarımına bağlıdır.

Üç-tanklı sıvı sistemi, bu yöntemlerin denenmesi için ideal bir sistemdir, çünkü musluk parametrelerini değiştirerek sistem SISO veya MIMO olacak şekilde ayarlanabilmektedir. Proje sürecinde üç-tanklı sıvı sisteminin denetimi, sırasıyla SISO çevrim-dışı, SISO çevrim-içi ve MIMO çevrim-dışı yöntemleriyle başarılı bir şekilde gerçekleştirilmiştir.

**Anahtar Kelimeler:** Genelleştirilmiş Öngörülü Denetim, Destek Vektör Makineleri, MIMO Modelleme ve Denetim.



## Abstract

In this project, we have studied on controlling unknown nonlinear systems via Support Vector Machines (SVMs) approach. The SVM model of the plant to be controlled is employed in the Generalized Predictive Control (GPC) framework. The developed methods have been successfully applied on a real system – a three-tank liquid system.

The developed methods in the project are composed of two groups namely off-line and on-line: 1) in the off-line methods, first, we construct a SVM model of the plant and then use it in the control. Meanwhile, it is assumed that the system parameters are not changing. The off-line methods are further split into two categories as single-input single-output (SISO) and multiple-input multiple-output (MIMO). 2) As far as the on-line method is concerned, we start the control task with an empty SVM model and then the control and modeling tasks are handled synchronously. The controller can adapt the variations in the system parameters.

In the off-line methods, first, the system is run for a certain period. At this point, it should be noted that both system and data collection parameters are chosen properly. Next, training and testing sets are formed by using the collected data. The training set is used for obtaining the SVM model, while the testing set is spared for model validation. Afterwards, a proper SVM model of the system is constructed and then used in the GPC loop. In the GPC loop, first, a candidate control vector is determined and then the future trajectory of the system in response to the control vector is predicted and then the candidate control vector is optimized in order to minimize the prediction error. Finally, the first element of the optimized control vector is applied to the system as the input. This process is repeated during every sampling periods. Similar process is applied for on-line control method. The only difference is that in the on-line method an *a priori* SVM model is not required; rather the initially empty SVM model is developed during the control task.

In these methods, the role of the SVM model is twofold. First, it is used for future predictions. Second, the gradient information required for optimization of the candidate control vector is extracted from the SVM model. In this respect, the SVM model plays a crucial role, for the control performance in the GPC methods are dominantly depend on the performance of the models.

The three-tank system is an ideal system for application of these methods, for the system can be set as a SISO or a MIMO system by changing the valve parameters. During the

project, the off-line SISO, the off-line MIMO and the on-line SISO methods have been successfully applied on the three-tank system, respectively.

**Keywords:** Generalized Predictive Control, Support Vector Machines, MIMO Modeling and Control.

## 1. GİRİŞ

Genelleştirilmiş Öngörülü Denetim (Generalized Predictive Control - GPC) (CLARKE, 1987a,b) yöntemi, Model Tabanlı Öngörülü Denetim (Model-Based Predictive Control - MPC) teknikleri sınıfına dahildir. Bu teknikler, 30 yıldan fazla bir zamandır açık-çevrimi kararsız sistemlerin, parametreleri ve/veya ölü-zamanları zamanla-değişen sistemlerin (CLARKE ve MOHTADI, 1989; CAMACHO, 1993), endüstriyel süreçlerin (RICHALET, 1993; CLARKE, 1988) yanı sıra çok değişik alanlarda denetim amaçlı kullanılmışlardır (Qin ve Badgwell, 2003). Literatürde önerilen ilk MPC tekniğini (RICHALET ve ark., 1978) takiben geliştirilen pek çok MPC yönteminin (De KEYSER ve Van CAUWENBERGHE, 1985; SOETERBOEK, 1992) arasında en yaygın olanı GPC yöntemidir. Yinede, tüm MPC tekniklerinin ardında yatan fikir aynıdır: denetimi yapılacak sistemin modelini kullanarak elde edilen ileri yönelik tahmine dayanarak, her bir örnekleme periyodunda, sonlu-ufuklu açık-çevrimli bir en iyileme problemi çözümlenerek bir dizi denetim işareti elde edilir ve dizinin ilk elemanı sisteme uygulanır. Model tabanlı denetim yöntemlerinde denetimi yapılacak sistemin modeli önemli bir rol oynar. Matematiksel modelin mevcut olduğu durum için literatürde pek çok çalışma yapılmıştır (CAMACHO E.F. ve BORDONS, 2003; ROSSITER, 2003). Bu projede ise denetimi yapılacak sistemin matematiksel modelinin mevcut olmadığı varsayılacaktır. Bu durumda, sistemin giriş-çıkış büyüklüklerinin ölçülmesiyle elde edilen modeller üzerinden denetim gerçekleştirilir.

Model tabanlı tekniklerde denetimi yapılacak sistemin modelinin doğruluğu çok önemli bir rol oynadığından, matematiksel modeli bilinmeyen sistemlerin denetimi için pek çok doğrusal ve doğrusal-olmayan modelleme yöntemleri uzunca zamandır kullanılmaktadır. Son yıllarda, işlemsel zeka alanındaki gelişmelerle birlikte, GPC döngüsünde kullanılacak modelin elde edilmesinde Yapay Sinir Ağları (SOLOWAY ve HALEY, 1996; HAGAN ve ark., 2002), Bulanık Sistemler (HUANG ve ark., 2000; LI ve ark., 2004), karma sistemler (HU ve ROSE, 1999; SARIMVEIS ve BAFAS, 2003) ve Genetik Algoritmalar (MARTINEZ ve ark., 1998) gibi esnek işlem araçlarından yararlanılmıştır.

Ancak bu yaklaşımların pek çoğunda elde edilen modeller sistemin dinamiğindeki ve/veya ortamdaki değişimlere uyum sağlayamamaktadır ki bu değişimler göz ardı edilemez olduğunda bu durum bir dezavantaja dönüşebilmektedir. GPC mekanizmalarının bu tür değişimlere uyarlanabilir olması için kullanılan modelin de uyarlanabilir olması gerekir. Bu amaçla literatürde, esnek modeller kullanan GPC yapıları önerilmiştir. Örnek olarak,

başlangıçta sistemi temsil eden boş bir kural tabanı çevrim-içi bir öğrenme yöntemiyle yavaş yavaş geliştirilerek uyarlanır bir denetim sağlanmıştır (HU ve ROSE, 1999). Diğer benzer çalışmalarda ise (TAN ve ark., 2001; MAHFOUF, 2003), modeller yinelemeli en küçük kareler yöntemiyle güncellenmiştir. Diğer taraftan, yapay sinir ağı tabanlı çalışmalarda (WANG ve ark., 2000; PARLOS ve ark., 2001; ALEXANDRIDIS ve SARIMVEIS, 2005), çevrim-içi olarak güncellenen yapay sinir ağları kullanılmıştır.

Matematiksel modeli bilinmeyen sistemlerin GPC döngüsü içinde denetimi için yukarıda bahsi geçen yöntemler çoğunlukla bir-girişli bir-çıkışlı (SISO) durumlar için gerçekleştirilmiştir. Ancak, endüstride karşılaşılan gerçek sistemler çoğu zaman çok-girişli çok-çıkışlı (MIMO) olarak karşımıza çıkmaktadır. Yine de bir SISO sistemin denetiminde belli bir başarıma sahip bir yöntem bazı değişikliklerle MIMO duruma uyarlanabilir. Örnek olarak, (YU ve GOMM, 2003) ve (CHEN ve YEA, 2002)'deki çalışmalarda çoklu MISO NARX yapay sinir ağı modelleri kullanılarak doğrusal olmayan sistemlerin GPC döngüsü ile denetimi gerçekleştirilmiştir. Diğer taraftan, yeni bir çalışmada (LAWRYNCZUK, 2009), NARX modeli kullanmak yerine yapılandırılmış bir yapay sinir ağı mimarisi ile de model tabanlı denetim gerçekleştirilmiştir. Yapay sinir ağlarıyla denetim konusunda detaylı bir çalışma (LAWRYNCZUK, 2007)'de bulunabilir. Benzer şekilde, literatürde, MIMO sistemlerin GPC döngüsünde denetimi için Takagi-Sugeno bulanık modellerinin kullanıldığı çalışmalar bulunabilir (ROUBOS ve ark., 1999; LI ve ark., 2004).

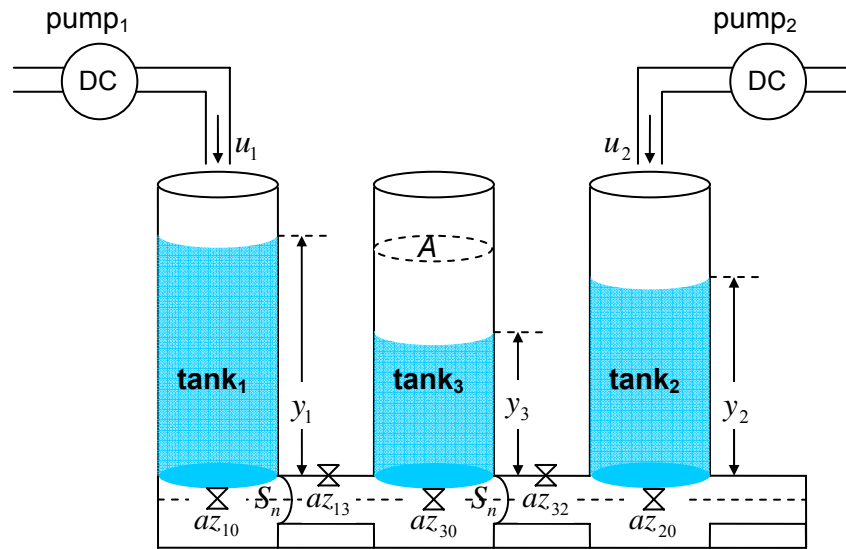
Bu projede ise, Destek Vektör Makineleri (Support Vector Machines - SVMs) yaklaşımlarıyla Genelleştirilmiş Öngörülü Denetim (Generalized Predictive Control - GPC) yaklaşımını bir arada kullanılarak matematiksel modeli bilinmeyen ve doğrusal olmayan sistemler için çevrim-içi ve çevrim-dışı denetim yöntemleri geliştirilmiştir. SVM bağlanım algoritmaları verilen bir veri kümesi için elde edilen bağlanım problemini çözerek doğrusal-olmayan sistemlerin modellenmesinde kullanılabilen araçlardır (VAPNIK, 1995; VAPNIK, 1998a,b). İstatistiksel Öğrenme Kuramı (Statistical Learning Theory) ve Yapısal Riski En Aza İndirme İlkesi'ne (Structural Risk Minimization Principle) dayanan SVM algoritmaları herhangi bir sınıflandırma veya bağlanım problemini global minimumu garanti ederek çözebilir. Global minimumun garanti edilmesi, bağlanım probleminin bir Karesel Programlama (Quadratic Programming - QP) problemine dönüştürülerek çözülmesiyle sağlanır. Bu sayede, son yıllarda, SVM tabanlı algoritmalar çok hızlı bir şekilde gelişmiş ve pek çok alana uygulanmıştır (CRISTIANINI ve TAYLOR, 2000; SCHÖLKOPF ve ark., 1999). Sonraki bölümde, projede geliştirilen yöntemler anlatılacaktır. Bu projeden elde edilen sonuçlar da Bölüm 3'te verilecektir.

## 2. PROJEDE KULLANILAN SİSTEM

Projede geliştirilen yöntemlerde en temel varsayım, denetimi yapılacak sistemin matematiksel modelinin bilinmediği varsayımdır. Bu yöntemlerde denetim, sistemin belli bir süre çalıştırılmasıyla toplanacak giriş-çıkış verilerinden elde edilecek modeller üzerinden yapılmaktadır. Dolayısıyla, denetimi yapılacak sistemin, bu şekildeki bir veri toplama işlemine müsait olması gerekliliği ortaya çıkmaktadır. Bu gereklilik de, projede geliştirilen yöntemlerde başka bir varsayımı da beraberinde getirmiştir. Bu da, denetimi yapılacak olan sistemin kararlı bir sistem olduğu varsayımdır. Aksi halde veri toplanması neredeyse imkansız hale gelir. Projede ele alınan denetim problemi, kararsız bir sistemi kararlı hale getirmekten çok, matematiksel modeli bilinmeyen kararlı bir sistemin belli kısıtlar altında istenilen referans işaretini mümkün olan en küçük hatalarla takip etmesini sağlamaktır. Dolayısıyla, projede geliştirilen yöntemlerin başarımı da bu çerçevede değerlendirilecektir. Bir sonraki alt-bölümde ayrıntılı bir şekilde anlatılan üç-tanklı sıvı sistemi, projede geliştirilen yöntemlerin denenmesi için ideal bir sistemdir.

### 2.1 Üç-Tanklı Sıvı Sistemi

Projede geliştirilen denetim yöntemlerinin testleri, AMIRA firmasının ürettiği DTS200 model, üç tank ve iki pompadan oluşan bir sıvı denetim sistemi üzerinde gerçekleştirilmiştir. Sistemin bir şeması aşağıdaki şekilde görülmektedir.



Şekil 2.1.1 Üç-tanklı sıvı denetim sistemi.

Üç-boyutlu, doğrusal-olmayan ve etkileşimli durumlara sahip olan bu sistem bazı özellikleri nedeniyle literatürde çokça kullanılmaktadır. Bu özelliklerden en önemlisi, sistemin SISO veya MIMO olarak ayarlanabilmesidir. Başka bir özelliği ise sistemin parametrelerinin kolaylıkla hatta denetim esnasında bile değiştirilebilmesidir. Vurgulanması gereken son bir özellik ise tanklar arası musluk parametrelerinin değiştirilmesiyle sistemin zorluk derecesinin değiştirilebilmesidir. Bu özellikler, bu sistemin, çeşitli denetim algoritmalarının sınanması için ideal bir test ortamı olmasına imkan vermiştir. Bu projede her ne kadar sistemin denklemlerinin bilinmediği varsayımıyla denetim gerçekleştiriliyor olsa da bu sistemin matematiksel modelinin verilmesinde fayda vardır. Sistemin matematiksel modeli aşağıdaki gibi bir diferansiyel denklem setiyle verilmektedir:

$$\begin{aligned} \dot{y}_1(t) &= \frac{1}{A}[u_1(t) - Q_{13}(t)], \\ \dot{y}_2(t) &= \frac{1}{A}[u_2(t) + Q_{32}(t) - Q_{20}(t)], \\ \dot{y}_3(t) &= \frac{1}{A}[Q_{13}(t) - Q_{32}(t)] \end{aligned} \quad (2.1.1)$$

burada

$$\begin{aligned} Q_{13}(t) &= az_{13}S_n \operatorname{sgn}(y_1(t) - y_3(t))\sqrt{2g|y_1(t) - y_3(t)|} \\ Q_{32}(t) &= az_{32}S_n \operatorname{sgn}(y_3(t) - y_2(t))\sqrt{2g|y_3(t) - y_2(t)|} \\ Q_{20}(t) &= az_{20}S_n\sqrt{2gy_2(t)} \end{aligned} \quad (2.2)$$

şeklinde. Bu denklemlerdeki büyüklüklerin açıklamaları ve aldıkları değerler Tablo 2.1.1'de verilmiştir (AMIRA, 2000).

Tablo 2.1.1 sisteme ilişkin değişkenler ve parametreler.

Parametre/Değişken	Değer/Açıklama
$y_i(t)$ : tank <sub>i</sub> 'deki sıvı seviyesi	$i^{\text{inci}}$ çıkış [m]
$u_i(t)$ : pompa <sub>i</sub> 'nin sıvı debisi	$i^{\text{inci}}$ giriş [m <sup>3</sup> /sn]
$az_{13}$ : tank <sub>1</sub> ve tank <sub>3</sub> arasındaki musluk katsayısı	<i>Bu katsayıların değerleri kullanılan yöntemle göre değişmektedir. Aldıkları değerler yöntemlerin açıklanması kısmında verilmiştir.</i>
$az_{32}$ : tank <sub>3</sub> ve tank <sub>2</sub> arasındaki musluk katsayısı	
$az_{10}$ : tank <sub>1</sub> ve rezervuar arasındaki musluk katsayısı	
$az_{20}$ : tank <sub>2</sub> ve rezervuar arasındaki musluk katsayısı	
$az_{30}$ : tank <sub>3</sub> ve rezervuar arasındaki musluk katsayısı	
A: silindirlerin kesiti	0.0154 [m <sup>2</sup> ]
$S_n$ : $n^{\text{inci}}$ bağlantı borusunun kesiti	$5 \times 10^{-5}$ [m <sup>2</sup> ]
g: yerçekimi katsayısı	9.81 [m/sn <sup>2</sup> ]

Bu sistemin pompalarının sınır deęerleri  $u_{1\min} = u_{2\min} = 0 \text{ m}^3/\text{sn}$  ve  $u_{1\max} = u_{2\max} = 0.0001 \text{ m}^3/\text{sn}$  olarak verilmiřtir. Bu sistemde ama, pompa<sub>1</sub> (MIMO durumda ayrıca pompa<sub>2</sub>) debisini deęiřtirerek tank<sub>1</sub> (MIMO durumda ayrıca tank<sub>2</sub>) sıvı seviyesinin denetlenmesidir.

### 3. YÖNTEMLER

Bu bölümde, projede geliştirilen yöntemlere yer verilmiştir. Proje üç kısımdan oluşmaktadır: 1) Çevrim-dışı SISO denetim. 2) Çevrim-içi SISO denetim. 3) Çevrim-dışı MIMO denetim. Bu denetimlerin hepsinde de üç-tanklı sıvı sistemi kullanılmıştır. Sistemin musluk parametreleri denetimin durumuna göre (SISO veya MIMO) uygun değerlere getirilmiştir. Okuma kolaylığı açısından yöntemler ayrı ayrı altbölümler halinde verilmiştir.

#### 3.1 Çevrim-dışı SISO Denetim

##### 3.1.1 Yöntem



Şekil 3.1.1. SISO sistem.

Şekil 3.1.1.'de görüldüğü gibi doğrusal olmayan sürekli zamanlı bir-girişli bir-çıkışlı bir sistemi ele alalım. Sistemin dinamiklerinin aşağıdaki gibi bir **Nonlinear AutoRegressive with eXogenous inputs (NARX)** modeliyle temsil edilebildiğini varsayıyoruz.

$$y(n) = f(u(n), \dots, u(n - n_1), y(n - 1), \dots, y(n - m_1)), \quad (3.1.1)$$

burada  $u(n)$   $n^{\text{inci}}$  zaman indeksinde sisteme uygulanan denetim işareti,  $y(n)$  ise sistemin çıkışıdır,  $n_1$  ve  $m_1$  de modelde yer alan geçmiş giriş ve çıkış gecikmelerini göstermektedir. Ayrıca, bu modeldeki  $f$  fonksiyonunun da bilinmediği varsayılmıştır. Yani sistemin matematiksel modeli elimizde mevcut değildir. Bunun yerine elimizde aşağıdaki gibi bir eğitim veri seti mevcuttur.

$$\begin{aligned} T &= \{u(k), \dots, u(k - n_1), y(k - 1), \dots, y(k - m_1); y(k)\}_{k=1}^{k=N} \\ &= \{\mathbf{x}_k; y(k)\}_{k=1}^{k=N} \end{aligned} \quad (3.1.2)$$



burada  $\mathbf{x}_k \in X \subseteq \mathfrak{R}^{n_1+m_1+1}$  büyüklüğü  $k^{\text{nci}}$  giriş veri noktasını ve  $y(k) \in Y \subseteq \mathfrak{R}$  da buna karşı düşen çıkış değerini göstermektedir. Bu verileri kullanarak sistemin giriş-çıkış ilişkisini temsil edebilecek bir model oluşturmak istiyoruz. Bu amaçla, veri seti  $T$ 'yi kullanarak sistemin modelini elde edecek SVM algoritması şu şekildedir. SVM bağlanım modelinin birincil formu (3.1.3) denkleminde görüldüğü gibidir; bu model öznitelik uzayı olan  $F$  uzayında doğrusaldır.

$$\hat{y}(i) = \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b \quad (3.1.3)$$

burada  $\mathbf{w}$  vektörü  $F$  öznitelik uzayında bir vektördür,  $\Phi(\cdot)$  fonksiyonu giriş uzayından öznitelik uzayına bir dönüşümdür,  $b$  büyüklüğü bir yanlılık (bias) terimidir ve  $\langle \cdot \rangle$  operatörü de  $F$  uzayında bir iç-çarpımı göstermektedir. SVM bağlanım algoritmaları genel olarak bir bağlanım problemini ikincil uzayda bir en iyileme problemi olarak görürler. SVM modelinin ikincil uzaydaki formu şu şekildedir:

$$\hat{y}(i) = \sum_{j=1}^N \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b, \quad (3.1.4)$$

burada  $\alpha_j$ 'ler her bir eğitim verisinin katsayılarıdır ve  $K(\mathbf{x}_i, \mathbf{x}_j)$  da şu şekilde verilen bir çekirdek fonksiyonudur:  $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = K_{ij}$ . Çekirdek fonksiyonu, öznitelik uzayında iç-çarpım işlemi gerçekleştirdiğinden dolayı  $\Phi(\mathbf{x})$  fonksiyonunun açık formunun bilinmesine gerek kalmaz. Denklem (3.1.4) ile verilen modelde, sıfırdan farklı bir  $\alpha_j$  değerine karşı düşen  $\mathbf{x}_j$  gibi bir eğitim noktasına *destek vektörü* denmektedir. Bu çalışmada, aşağıdaki gibi verilen Vapnik's  $\varepsilon$ -insensitive kayıp fonksiyonunu ( $L(\varepsilon, y, \hat{y})$ ) kullanan  $\varepsilon$ -SVR algoritması benimsenmiştir:

$$L(\varepsilon, y, \hat{y}) = \begin{cases} 0 & y - \hat{y} \leq \varepsilon \\ y - \hat{y} & y - \hat{y} > \varepsilon \end{cases} \quad (3.1.5)$$

$\varepsilon$ -SVR algoritması, bağlanım probleminin birincil formunu şu şekilde formülize eder:

$$\min_{\mathbf{w}, b, \xi, \xi^*} P_\varepsilon = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (3.1.6)$$

kısıtlar:

$$\begin{aligned} y(i) - \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - b &\leq \varepsilon + \xi_i, \quad i = 1, \dots, N \\ \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b - y(i) &\leq \varepsilon + \xi_i^*, \quad i = 1, \dots, N \\ \xi_i, \xi_i^* &\geq 0, \quad i = 1, \dots, N \end{aligned} \quad (3.1.7)$$

burada  $\varepsilon$  değeri çıkış için tolare edilebilen en büyük hata değeridir,  $\xi_i$ 'ler ve  $\xi_i^*$ 'ler gevşek değişkenlerdir,  $\|\cdot\|$  operatörü Öklit normudur ve  $C$  de modelin karmaşıklığı ile  $\varepsilon$ 'den daha büyük hatalara gösterilen tolerans derecesi arasında denge kurmayı sağlayan bir parametredir. Böylece, en iyileme probleminin ikincil formu aşağıdaki gibi bir karesel programlama (Quadratic Programming - QP) problemine dönüşür:

$$\min_{\beta, \beta^*} D_\varepsilon = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K_{ij} (\beta_i - \beta_i^*) (\beta_j - \beta_j^*) + \varepsilon \sum_{i=1}^N (\beta_i + \beta_i^*) - \sum_{i=1}^N y(i) (\beta_i - \beta_i^*) \quad (3.1.8)$$

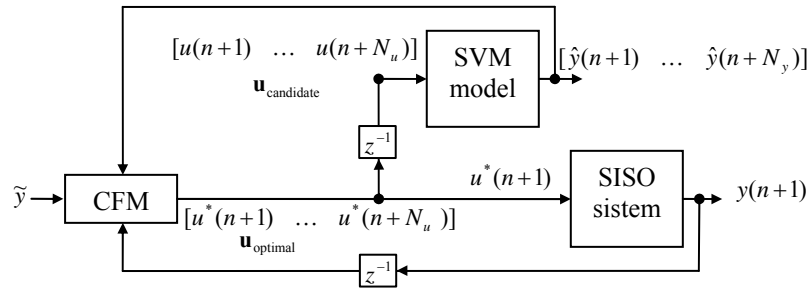
kısıtlar:

$$0 \leq \beta_i, \beta_i^* \leq C, \sum_{i=1}^N (\beta_i - \beta_i^*) = 0, i = 1, \dots, N \quad (3.1.9)$$

Denklem (3.1.8 ve 3.1.9) ile verilen QP probleminin çözümü  $\beta_i$ 'ler ve  $\beta_i^*$ 'lerin optimum değerlerini verir. Modeldeki  $b$ 'ler,  $0 \leq \beta_i - \beta_i^* \leq C$  koşulunun sağlandığı her bir  $\mathbf{x}_i$  destek vektörü için  $y(i) - \hat{y}(i) = \varepsilon$  koşulu sağlanacak şekilde bulunurlar. Eğer  $j = 1, \dots, N$  olmak üzere  $\mathbf{x}_j$  destek vektörü için  $\alpha_j$  gibi yeni bir katsayı  $\alpha_j = \beta_j - \beta_j^*$  şeklinde tanımlanırsa, o zaman Denklem (3.1.4)'teki gibi bir SVM modeli elde edilir. Hatta sadece destek vektörleri dikkate alındığında model şu hale gelir:

$$\hat{y}(i) = \sum_{\substack{j=1 \\ j \in SV}}^{\#SV} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \quad (3.1.10)$$

burada  $\#SV$  modeldeki destek vektör sayısıdır (SMOLA ve SCHÖLKOPF, 1998; CRISTIANINI ve TAYLOR, 2000; SCHÖLKOPF ve ark., 1999). Denklem (3.1.10) ile verilen SVM modeli, tüm eğitim verisinin sadece destek vektörleriyle temsil edilmesi bakımından seyrek bir modeldir.  $\varepsilon$ -SVR algoritmasının parametreleri, çıkıştaki hata toleransı  $\varepsilon$ , ayar parametresi  $C$ , eğitim verilerinin sayısı  $N$  ve modelde kullanılan çekirdek fonksiyonunun parametreleridir.  $\varepsilon$ -SVR algoritmasının en büyük avantajı uygun  $\varepsilon$  değerleri seçilerek toplam eğitim hatasının en büyük değerinin önceden bilinebilmesidir. Diğer taraftan, bu algoritmada elde edilen modeldeki destek vektör sayısı kontrol edilemez ve bazı durumlarda seyreklik ortadan kalkabilir.



Şekil 3.1.2. SVM-Tabanlı GPC şeması.

Denetimi yapılacak sistemin SVM modeli elde edildikten sonra artık bu model SVM-Tabanlı GPC içerisinde kullanılabilir. Şekil 3.1.2'de, projede geliştirilen SVM-Tabanlı GPC yapısı görülmektedir. Burada,  $\tilde{y}$  referans işareti ve  $\hat{y}$  da tahmindir. Bu yöntemde, SVM modeli sistemin  $\mathbf{u}_{\text{candidate}}$  aday denetim vektörüne karşılık gelecekteki yörüngesini tahmin eder ve aynı zamanda bedel fonksiyonunu en aza indirme (Cost Function Minimization - CFM) bloğunda gerekli türev bilgisini sağlar. CFM bloğu referans işareti, ölçülen çıkışı ve SVM modelinin tahminlerini alır ve en iyilenmiş bir denetim vektörü oluşturur. Bu en iyilenmiş vektörün ilk elemanı sisteme denetim girdileri olarak uygulanır. Bu algoritmayı biraz daha açmak gerekirse, örnekleme periyotlarının her birinde ilk olarak, aşağıdaki gibi bir aday denetim vektörü oluşturulur:

$$\mathbf{u}_{\text{candidate}} = [u(n+1) \quad \dots \quad u(n+N_u)] \quad (3.1.11)$$

burada  $N_u$  denetim girdisine ilişkin denetim ufkudur. Bu denetim vektörü sisteme doğrudan uygulanmaz çünkü henüz en iyilenmiş değildir. Bu en iyileme, SVM modelinin aday denetim vektörüne karşı ürettiği tahminlerle yapılmaktadır. Bura en iyilemede amaç,  $\mathbf{u}_{\text{candidate}}$  vektörünü uygun bir şekilde değiştirerek tahmin hatalarını en aza indirirken denetim işaretinde de çok fazla değişime yol açmamaktır. Bunu aşağıdaki amaç fonksiyonu ile göstermek daha kolay olacaktır:

$$F = \sum_{k=1}^{N_y} (\tilde{y}(n+k) - \hat{y}(n+k))^2 + \sum_{k=1}^{N_u} \lambda (\Delta u(n+k))^2 \quad (3.1.12)$$

burada  $N_y$  kestirim ufkudur ve  $\lambda$  da denetim girdisi değişimleri için ceza terimleridir. Bu en iyilemeyi gerçekleştirmek için aşağıdaki gibi genel bir güncelleme kuralı kullanılır:

$$\mathbf{u}_{\text{optimal}} = \mathbf{u}_{\text{candidate}} + s\mathbf{p}, \quad (3.1.13)$$

burada  $\mathbf{p}$  arama yönü ve  $s$  de adım-uzunluğu olarak adlandırılır. Ardından, en iyilenmiş denetim vektöründe  $(n+1)$  zaman indeksine karşı düşen eleman, yani  $u^*(n+1)$  elemanı,

sisteme giriş olarak uygulanır. En iyileme literatüründe,  $\mathbf{p}$  vektörünün bulunması için birinci- ve ikinci-dereceden türev tabanlı yöntemler mevcuttur (NOCEDAL ve WRIGHT, 1999; VENKATARAMAN, 2002). Birinci-dereceden metotlar gradyant vektörünü kullanırken, ikinci-dereceden metotlar buna ilave olarak Hessian matrisini kullanırlar. Bu çalışmada, *Değiştirilmiş Newton (Modified Newton)* adı verilen yöntem kullanılmıştır. Bu yöntemde, arama yönü, pozitif tanımlı Hessian matrisinin tersi ile birlikte en büyük artım yönünü gösteren  $\mathbf{g}$  gradyant vektörüne zıt yönde seçilir. Başka bir deyişle,

$$\mathbf{p} = -\mathbf{H}^{-1} \mathbf{g}, \quad (3.1.14)$$

burada,

$$\mathbf{g} = \frac{\partial F}{\partial \mathbf{u}_{\text{candidate}}} = \left[ \frac{\partial F}{\partial u(n+1)} \quad \cdots \quad \frac{\partial F}{\partial u(n+N_u)} \right]^T. \quad (3.1.15)$$

ve

$$\mathbf{H} = \frac{\partial^2 F}{\partial \mathbf{u}_{\text{candidate}}^2} = \begin{bmatrix} \frac{\partial^2 F}{\partial u(n+1)\partial u(n+1)} & \cdots & \frac{\partial^2 F}{\partial u(n+1)\partial u(n+N_u)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial u(n+N_u)\partial u(n+1)} & \cdots & \frac{\partial^2 F}{\partial u(n+N_u)\partial u(n+N_u)} \end{bmatrix}^T. \quad (3.1.16)$$

şeklindedir. Gradyant vektörü  $\mathbf{g}$ 'nin hesabında gerekli olan kısmi türevler aşağıdaki şekilde bulunur:

$$\begin{aligned} \frac{\partial F}{\partial u(n+h)} &= -2 \sum_{k=1}^{N_y} (\tilde{y}(n+k) - \hat{y}(n+k)) \frac{\partial \hat{y}(n+k)}{\partial u(n+h)} \\ &\quad + 2\lambda \sum_{k=1}^{N_u} (\Delta u(n+k)) (\delta_{h,k} - \delta_{h,k-1}) \end{aligned} \quad (3.1.17)$$

burada  $\delta_{h,k}$  Kronecker Delta fonksiyonudur. Benzer şekilde, Hessian matrisi  $\mathbf{H}$ 'nin hesabında gerekli olan kısmi türevler de aşağıdaki şekilde bulunur:

$$\begin{aligned} \frac{\partial^2 F}{\partial u(n+h)\partial u(n+m)} &= 2 \sum_{k=1}^{N_y} \frac{\partial \hat{y}(n+k)}{\partial u(n+h)} \frac{\partial \hat{y}(n+k)}{\partial u(n+m)} \\ &\quad - 2 \sum_{k=1}^{N_y} \frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h)\partial u(n+m)} (\tilde{y}(n+k) - \hat{y}(n+k)) \\ &\quad + 2\lambda \sum_{k=1}^{N_u} (\delta_{h,k} - \delta_{h,k-1}) (\delta_{m,k} - \delta_{m,k-1}) \end{aligned} \quad (3.1.18)$$

İlerleyen kısımlarda, SVM modeli kullanılarak,  $\hat{y}(n+k)$  tahminleri ile  $\frac{\partial \hat{y}(n+k)}{\partial u(n+h)}$  ve

$\frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h) \partial u(n+m)}$  kısmi türevlerinin nasıl bulunacağı gösterilmiştir.

Eğer (IPLIKCI, 2006a)'da SISO sistemler için önerilen SVM-Tabanlı GPC yöntemini izleyecek olursak, ilk olarak şu anki durum vektörünü aşağıdaki şekilde oluşturulmalıdır:

$$\mathbf{c}(n) = [u(n), \dots, u(n-n_1), y(n-1), \dots, y(n-m_1)]^T, \quad (3.1.19)$$

Böylelikle, SVM modelinin çıkışı şu hale gelir,

$$\hat{y}(n) = \sum_{j=1}^{\#SV} \alpha_j K(\mathbf{c}(n), \mathbf{x}_j) + b. \quad (3.1.20)$$

Bu çalışmada, aşağıdaki gibi verilen radyal tabanlı fonksiyon (Radial Basis Function - RBF) çekirdek fonksiyonu olarak seçilmiştir:

$$K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2}\right),$$

burada  $\sigma$  genişlik parametresidir. Eğer  $d_{j,n}$  uzaklığı,  $j^{\text{inci}}$  destek vektörü  $\mathbf{x}_j$  ile şu anki durum vektörü  $\mathbf{c}(n)$  arasındaki Öklit uzaklığı olarak aşağıdaki gibi tanımlanırsa,

$$\begin{aligned} d_{j,n} &= (\mathbf{c}(n) - \mathbf{x}_j)^T (\mathbf{c}(n) - \mathbf{x}_j) \\ &= \sum_{i=0}^{n_1} (x_{j,i+1} - u(n-i))^2 + \sum_{i=1}^{m_1} (x_{j,i+1+n_1} - y(n-i))^2, \end{aligned} \quad (3.1.21)$$

o zaman, çekirdek fonksiyonu şu şekilde yazılabilir,

$$K(\mathbf{c}(n), \mathbf{x}_j) = \exp\left(-\frac{d_{j,n}}{2\sigma^2}\right), \quad (3.1.22)$$

ve bunun sonucunda, SVM modelinin çıkışı şu hale gelir,

$$\hat{y}(n) = \sum_{j=1}^{\#SV} \alpha_j \exp\left(-\frac{d_{j,n}}{2\sigma^2}\right) + b. \quad (3.1.23)$$

Böylelikle, ileri yönelik tahminler aşağıdaki formülle yapılabilir:

$$\hat{y}(n+k) = \sum_{j=1}^{\#SV} \alpha_j \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right) + b, \quad k = 1, \dots, N_y, \quad (3.1.24)$$

burada,

$$d_{j,n+k} = \sum_{i=0}^{n_1} \begin{cases} (x_{j,i+1} - u(n+k-i))^2 & k - N_u < i \\ (x_{j,i+1} - u(n+N_u))^2 & k - N_u \geq i \end{cases} + \sum_{i=1}^{\min(k,m_1)} (x_{j,i+1+n_1} - \hat{y}(n+k-i))^2 + \sum_{i=k+1}^{m_1} (x_{j,i+1+n_1} - y(n+k-i))^2 \quad (3.1.25)$$

İleri yönelik tahminler bulunduktan sonra,  $\frac{\partial \hat{y}(n+k)}{\partial u(n+h)}$  kısmi türevleri artık şu şekilde bulunabilir:

$$\frac{\partial \hat{y}(n+k)}{\partial u(n+h)} = \sum_{j=1}^{\#SV} \alpha_j \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h)}, \quad (3.1.26)$$

burada,

$$\begin{aligned} \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h)} &= \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial d_{j,n+k}} \frac{\partial d_{j,n+k}}{\partial u(n+h)} \\ &= -\frac{1}{2\sigma^2} \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right) \frac{\partial d_{j,n+k}}{\partial u(n+h)} \end{aligned} \quad (3.1.27)$$

ve

$$\begin{aligned} \frac{\partial d_{j,n+k}}{\partial u(n+h)} &= -2 \sum_{i=0}^{n_1} \begin{cases} (x_{j,i+1} - u(n+k-i)) \delta_{k-i,h} & k - N_u < i \\ (x_{j,i+1} - u(n+N_u)) \delta_{N_u,h} & k - N_u \geq i \end{cases} \\ &\quad - 2 \sum_{i=1}^{\min(k,m_1)} (x_{j,i+1+n_1} - \hat{y}(n+k-i)) \frac{\partial \hat{y}(n+k-i)}{\partial u(n+h)} \Gamma_{k-i-h}, \end{aligned} \quad (3.1.28)$$

$$k = 1, \dots, N_y \quad h = 1, \dots, N_u.$$

Benzer şekilde,  $\frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h) \partial u(n+m)}$  kısmi türevleri şu şekilde bulunur:

$$\frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h) \partial u(n+m)} = \sum_{j=1}^{\#SV} \alpha_j \frac{\partial^2 \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h) \partial u(n+m)}, \quad (3.1.29)$$

burada,

$$\begin{aligned}
\frac{\partial^2 \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h)\partial u(n+m)} &= \frac{\partial^2 \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial d_{j,n+k}^2} \frac{\partial d_{j,n+k}}{\partial u(n+h)} \frac{\partial d_{j,n+k}}{\partial u(n+m)} \\
&+ \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial d_{j,n+k}} \frac{\partial^2 d_{j,n+k}}{\partial u(n+h)\partial u(n+m)} \\
&= \frac{1}{4\sigma_y^4} \exp\left(-\frac{d_{j,n+k}}{2\sigma_y^2}\right) \frac{\partial d_{j,n+k}}{\partial u_{n+h}} \frac{\partial d_{j,n+k}}{\partial u_{n+m}} \\
&- \frac{1}{2\sigma_y^2} \exp\left(-\frac{d_{j,n+k}}{2\sigma_y^2}\right) \frac{\partial^2 d_{j,n+k}}{\partial u_{n+h}\partial u_{n+m}}
\end{aligned} \tag{3.1.30}$$

ve

$$\begin{aligned}
\frac{\partial^2 d_{j,n+k}}{\partial u(n+h)\partial u(n+m)} &= 2 \sum_{i=0}^{n_1} \begin{cases} \delta_{k-i,h,m} & k - N_u < i \\ \delta_{N_u,h,m} & k - N_u \geq i \end{cases} \\
&- 2 \sum_{i=1}^{\min(k,m_1)} (x_{j,i+1+m_1} - \hat{y}(n+k-i)) \frac{\partial^2 \hat{y}(n+k-i)}{\partial u(n+h)\partial u(n+m)} \Gamma_{k-i-\max(h,m)} \\
&+ 2 \sum_{i=1}^{\min(k,m_1)} \frac{\partial \hat{y}(n+k-i)}{\partial u(n+h)} \frac{\partial \hat{y}(n+k-i)}{\partial u(n+m)} \Gamma_{k-i-\max(h,m)} \\
&k = 1, \dots, N_y \quad h = 1, \dots, N_u,
\end{aligned} \tag{3.1.31}$$

burada  $\Gamma_x$  aşağıdaki gibi bir birim basamak fonksiyonudur,

$$\Gamma_x = \begin{cases} 0 & x < 0 \\ 1 & 0 \leq x \end{cases} \tag{3.1.32}$$

Böylece, arama yönü olan  $\mathbf{p}$ 'nin hesabında gerekli olan ileri yönelik tahminler ve kısmi türevler sistemin SVM modeli kullanılarak elde edilebilir. Ayrıca, not edilmelidir ki RBF çekirdek fonksiyonu için verilen bu formüller diğer tipteki çekirdek fonksiyonlarına da kolaylıkla uyarlanabilir. Arama yönü  $\mathbf{p}$  bulunduktan sonra, adım-uzunluğu  $s$ 'nin bulunması artık bir-boyutlu en iyileme problemine dönüşür ki bu çalışmada bu problem *Altın Oran* (*Golden-Section*) yöntemiyle çözülmüştür. Sonuç olarak, SVM-Tabanlı GPC prosedürü aşağıdaki gibi maddeler halinde açıklanabilir:

**Adım1.** Sistemin dinamiklerini modellemek için gerekli verileri toplamak için ilk olarak, giriş işaretinin alt ve üst sınırları, yani  $u_{\min}$  ve  $u_{\max}$  değerleri, belirlenir. Ardından, giriş işaretinin en kısa ve en uzun zaman aralıkları, yani  $\tau_{\min}$  ve  $\tau_{\max}$  değerleri, uygun bir şekilde seçilir. Sonra, sistem belli bir periyot boyunca koşuturulur. Bu esnada, sisteme,

genlikleri  $[u_{\min}, u_{\max}]$  aralığında rastgele ve süreleri  $[\tau_{\min}, \tau_{\max}]$  aralığında rastgele değişen bir dizi giriş işareti uygulanır.

**Adım2.** Denklem (3.1.1) ile verilen NARX modelinde görülen  $n_1$  ve  $m_1$  belirlenir ve bir eğitim verileri seti oluşturulur. Ardından, bu setteki tüm giriş ve çıkış değişkenleri  $[0,1]$  aralığına normalize edilir.

**Adım3.** Sistemin SVM modeli elde edilirken  $N$  adet veri eğitim için geri kalanı da test için kullanılır.

**Adım4.** Her iterasyonda, ya da örnekleme periyodu süresince, SVM-Tabanlı GPC algoritması şu şekilde uygulanır:

Sistemin çıkışından ölçümleri al, aday denetim vektörü  $\mathbf{u}_{\text{candidate}}$ 'yu oluştur ve bu vektöre karşı düşen ileri yönelik tahminleri SVM modelini kullanarak elde et. Yine SVM modelini kullanarak, formüller yardımıyla gradyant vektörü  $\mathbf{g}$ 'yi bul ve arama yönü olarak  $\mathbf{p} = -\mathbf{H}^{-1}\mathbf{g}$  al. Ardından, en iyi adım-uzunluğunu bul ve  $\mathbf{u}_{\text{optimal}} = \mathbf{u}_{\text{candidate}} + s\mathbf{p}$  kuralı ile güncellemeyi yap. Son olarak,  $\mathbf{u}_{\text{optimal}}$  vektöründe  $(n+1)$  zaman indeksine karşı düşen ilk elemanı sisteme uygula.

### 3.1.2 Deney

DTS200 sistemindeki musluk katsayıları uygun şekilde ayarlanarak ve sadece 1. tanka sıvı pompalanarak bu tankın sıvı seviyesi kontrol edilmeye çalışılmış ve böylece sistem tek-girişli-tek-çıkışlı bir sisteme dönüştürülmüştür.

#### 3.1.2.1 Modelleme Verileri

Sistemin SVM modelinin olabildiğince doğru bir şekilde elde edilebilmesi için, uygun modelleme verisinin oluşturulmasına özel bir önem verilmiştir. Kabul edilebilir bir MIMO denetimi için, SVM modeli sistemi tüm çalışma bölgesinde doğru bir şekilde temsil etmelidir. Modelleme verisinin tüm çalışma bölgesini kaplaması için, veri toplama aşamasında bazı parametreler üzerinde çalışılmış ve onların uygun değerlerinin bulunması sağlanmıştır. Bu parametreler şunlardır:

- *Musluk katsayıları* ( $az_{13}$ ,  $az_{32}$ ,  $az_{10}$ ,  $az_{20}$  ve  $az_{30}$ ):

Sistemin beş adet musluk katsayısı vardır:  $az_{13}$ ,  $az_{32}$ ,  $az_{10}$ ,  $az_{20}$  ve  $az_{30}$ . Bu parametrelerin sistem davranışı üzerinde oldukça önemli etkileri vardır. Örneğin,  $az_{13}$  ve  $az_{32}$  parametreleri tanklar arasındaki etkileşim derecesini belirlerken,  $az_{10}$ ,  $az_{20}$  ve  $az_{30}$  parametreleri de tanklardan rezervuara akan sıvı debilerini belirler. Bu parametre

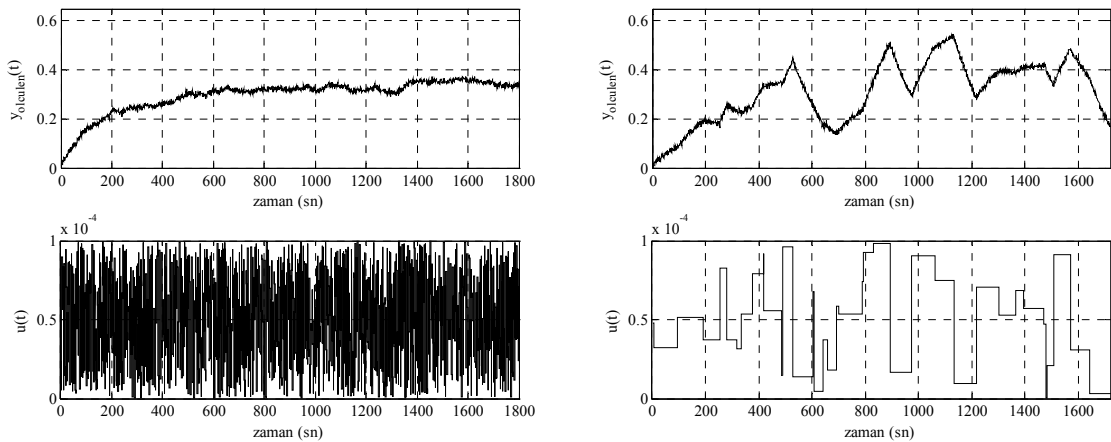


değerleri 0 ile 1 arasında değişir: 0 pozisyonunda musluktan hiçbir akış olmazken, 1 pozisyonu musluğun tamamen açık olduğu anlamına gelir. Bu çalışmada, uygun bir modelleme verisi elde etmek için, muslukların pozisyonları deneme-yanılma ile bulunmuştur ve sonunda parametre değerleri yaklaşık olarak aşağıdaki şekilde seçilmiştir:

$$az_{13} = 0.42, az_{32} = 1.00, az_{10} = 0.00, az_{20} = 1.00, az_{30} = 0.00.$$

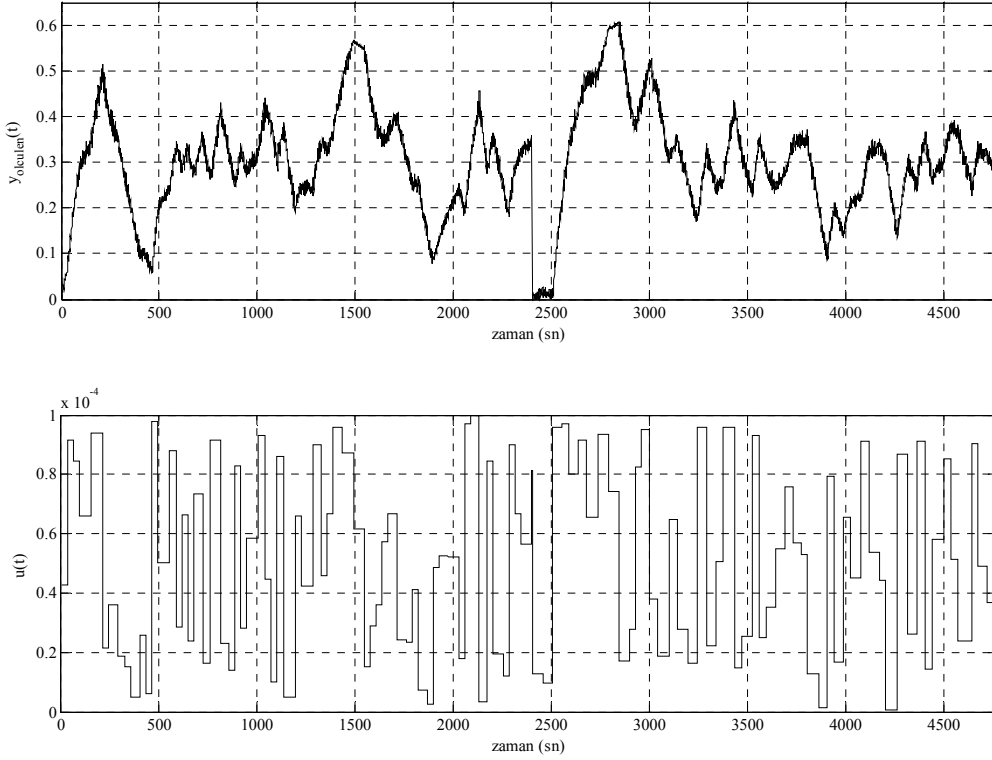
- Giriş işaretlerinin en uzun süresi ( $\tau_{\max}$ ):

Veri toplama aşamasında  $u_1(t)$  ve  $u_2(t)$  giriş işaretlerinin genlikleri sırasıyla  $[u_{1\min}, u_{2\min}]$  ve  $[u_{1\max}, u_{2\max}]$  aralıklarında rastgele değiştirilirken, süreleri de  $\tau_{\min}$  ve  $\tau_{\max}$  arasında rastgele değişmektedir. Bu çalışmada,  $\tau_{\min}$  süresi, örnekleme periyodu  $T_s$ 'ye eşit seçilmiştir, yani  $\tau_{\min} = T_s = 1$  sn. Musluk katsayılarında olduğu gibi,  $\tau_{\max}$  süresinin de modelleme verileri üzerinde büyük etkisi vardır. Daha ayrıntı vermek gerekirse,  $\tau_{\max}$  süresi kısaldıkça, giriş işaretleri sistemin tüm modlarını uyarmada yetersiz kalırlar ve bunun sonucunda modelleme verileri tüm çalışma bölgesine oranla daha dar bir bantta yer alırlar. Bu durum Şekil 3.1.3a'da görülmektedir. Burada,  $\tau_{\max} = 1$  sn alınarak elde edilen sistem çıkışlarının tüm çalışma bölgesine kıyasla çok dar bir bantta seyrettiğini görmekteyiz. Diğer taraftan, eğer  $\tau_{\max}$  süresi uzadıkça, verilerin taradığı bant daha da genişler ancak  $\tau_{\max}$  iyice uzarsa bu sefere de giriş işaretleri sistemin tüm modlarını uyarmada yine yetersiz kalır. Bu durum Şekil 3.1.3b'de görülmektedir ki burada veriler  $\tau_{\max} = 100$  sn alınarak toplanmıştır. Bu yüzden bu iki marjinal durum arasında en iyi değer bulunması gerekmektedir ki bu çalışmamızda bu değer  $\tau_{\max} = 60$  sn olarak bulunmuştur.



Şekil 3.1.3. a)  $\tau_{\max} = 1$  sn; b)  $\tau_{\max} = 100$  sn.

Sonuç olarak bu çalışmada, musluk parametreleri  $az_{13} = 0.42$ ,  $az_{32} = 1.00$ ,  $az_{10} = 0.00$ ,  $az_{20} = 1.00$ ,  $az_{30} = 0.00$  şeklinde ve  $\tau_{\max}$  da  $\tau_{\max} = 60$  sn olarak seçilmiştir. Bu parametre değerleriyle toplanan veriler Şekil 3.1.4'te görülmektedir.

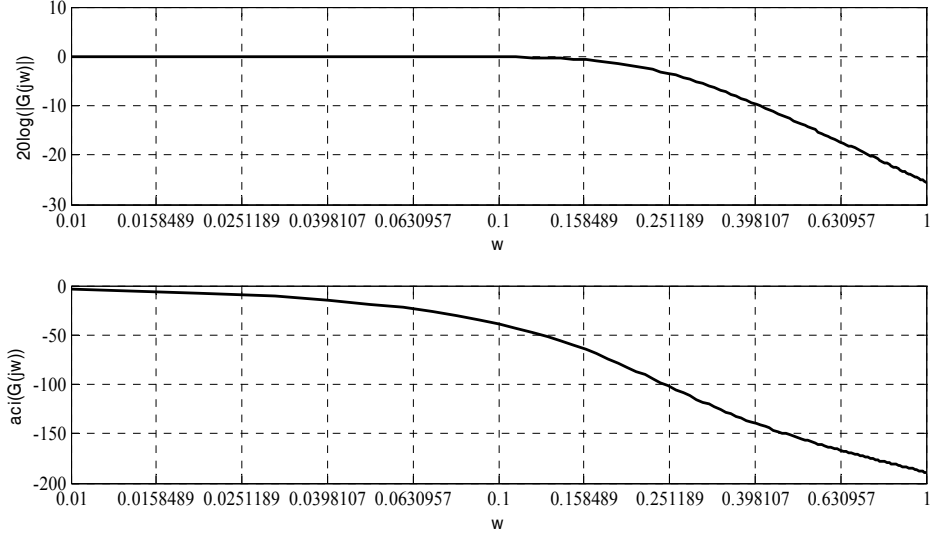


Şekil 3.1.4. Modelleme için toplanan veriler.

Bu arada şu hususu belirlemek gerekiyor. Her ne kadar sistemin ölçülmesi (kalibrasyonu) üretici firma tarafından yeniden yapılmış olsa da hala sıvı seviye algılayıcısından gelen seviye bilgisi gerçek değere eşit olmamakta ve küçük de olsa bir miktar sapma gözlenmektedir. Ayrıca, DTS200 sisteminin bulunduğu ortamın iyi yalıtılmamış olmasından kaynaklanan birtakım bozucu gürültüler de sıvı seviye algılayıcısından gelen seviye bilgisine eklenmekte ve toplanan verileri kirletmektedir. Bu durum yukarıdaki şekildeki sıvı seviyesi grafiğindeki dalgalanmalardan da açıkça görülmektedir. Ölçüm değerlerinin üzerine binen gürültünün etkisini azaltmak veya tamamen yok etmek için ölçüm değerlerinden oluşan işaret aşağıdaki gibi bir ayrık-zamanlı alçak-geçiren-filtreden geçirilmiştir:

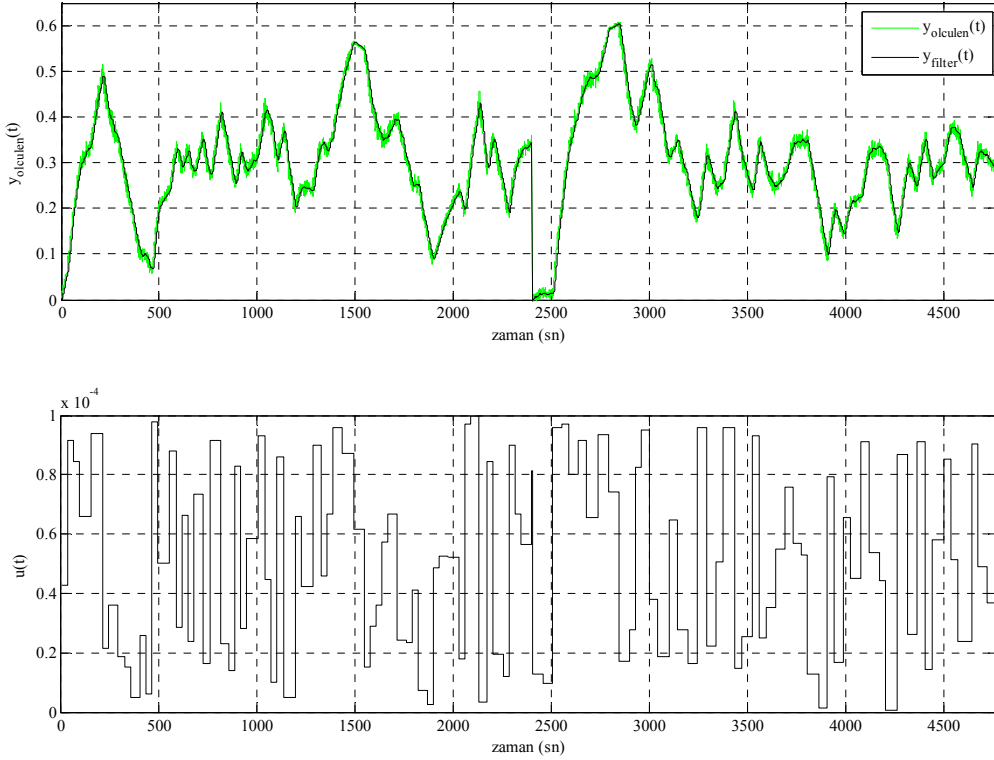
$$H(z) = \frac{0.02483z^{-1} + 0.02224z^{-2}}{1 - 1.672z^{-1} + 0.719z^{-2}} \quad (3.1.28)$$

Bu filtrenin aşağıdaki Bode diyagramından da görüleceği gibi yüksek frekanslı gürültü bileşenleri filtreden geçememektedir.



Şekil 3.1.5. Alçak Geçiren Filtrenin frekans cevabı.

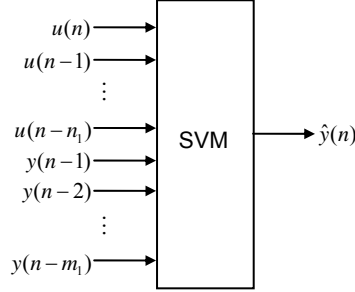
Aşağıdaki şekilde de algılayıcıdan gelen işaret ve onun filtrelenmiş hali görülmektedir.



Şekil 3.1.6. Ölçülen çıkış ve filtrelenmiş çıkış işaretleri.

### 3.1.2.2 SVM Modelinin Elde Edilmesi

Toplanan veriler kullanılarak sistemin bir SISO modelinin elde edilmesi Şekil 3.1.7'de görüldüğü gibi gerçekleştirilmiştir.

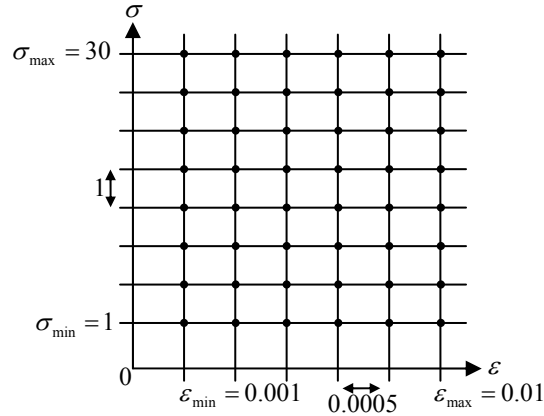


Şekil 3.1.7. Üç-tanklı sistem için SISO SVM modeli.

Bu noktada sistemin giriş-çıkış ilişkisinin aşağıdaki gibi olduğu varsayılmıştır:

$$y(n) = f(u(n), \dots, u(n - n_1), y(n - 1), \dots, y(n - m_1)), \quad (3.1.29)$$

burada giriş ve çıkış gecikmeleri  $n_1 = m_1 = 5$  olarak alınmıştır. Her ne kadar üç-tanklı sıvı sisteminin üçüncü dereceden bir sistem olması nedeniyle giriş ve çıkış gecikmelerinin  $n_1 = m_1 = 3$  olarak alınmasının yeterli olacağı düşünülse de, daha güvenilir bir model elde etmek adına bu gecikme sayılarının biraz daha büyük seçilmesinde hiçbir sakınca yoktur.



Şekil 3.1.8. En iyi  $(\sigma, \epsilon)$  çifti için grid araması.

Modelleme aşamasında kullanılan SVM parametrelerinin belirlenmesine gelince; bu çalışmada ayar parametresi etkisinin görece olarak çok az olması sebebiyle  $C = 1000$  olarak sabit tutulmuştur. Bunun yerine, modelleme başarımında çok daha etkili parametreler olan  $\sigma$  ve  $\epsilon$  parametrelerinin en uygun değerlerinin bulunmasına çalışılmıştır. Bu amaçla, Şekil

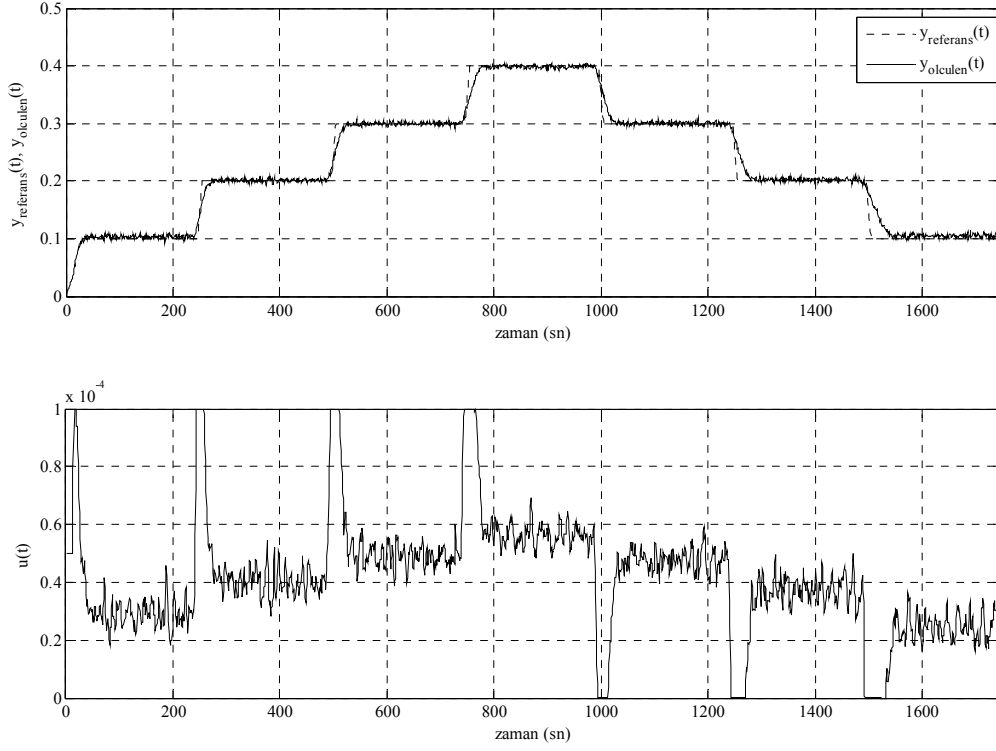
3.1.8'de görüldüğü gibi,  $\sigma\varepsilon$ -düzleminde  $[\sigma_{\min}, \sigma_{\max}]$  ve  $[\varepsilon_{\min}, \varepsilon_{\max}]$  aralıkları arasında bir grid araması yapılmıştır.

Bu aramada, grid üzerindeki her bir  $(\sigma, \varepsilon)$  noktası için verilerin yarısı kullanılarak bir SVM modeli oluşturulmuş diğer yarısıyla da bu model test edilmiştir. Böylece, en küçük test hatasını veren  $(\sigma, \varepsilon)$  çifti belirlenmiştir. Bunun sonucunda, SVM için en iyi  $(\sigma, \varepsilon)$  çifti (25.0, 0.003) olarak bulunmuş ve bu değerlerle 13 destek vektöründen oluşan bir SVM modeli elde edilmiştir.

### 3.1.2.3 Deneysel Sonuçlar

Denetim aşamasında, denetim parametreleri şu şekilde seçilmiştir:  $N_y = 10$ ,  $N_u = 2$  ve  $\lambda = 2.0$ . Bu kısımda, en iyiye yakın bu denetim parametreleriyle bulunan deneysel sonuçlar yer almaktadır. Bir sonraki kısımda ise bu denetim parametrelerinin denetim başarımı üzerindeki etkileri incelenmiştir. Deneysel sonuçlar, SVM-Tabanlı GPC yöntemi için iki farklı referans işareti için denetlenmiştir. Şekil 3.1.9, basamak şeklindeki referans işareti için elde edilen deneysel sonuçları göstermektedir. Şekilden görüldüğü gibi DTS200 sisteminin çıkışı olan 1. tanktaki sıvı istenilen seviyelerde istenildiği kadar tutulabilmiştir. Ancak, şekle bakıldığında, sıvı seviyesinin sabit kalmayıp belli bir bantta dalgalandığı düşünülebilir. Bu, önceden de belirtildiği gibi, tamamen sıvı seviye algılayıcısının bozucu faktörlerden etkilenmesi sonucu oluşan ölçüm hatalarından kaynaklanmaktadır. Gerçekte deney boyunca sıvı seviyesinde böyle bir dalgalanma gözlenmemiştir.

Sonuç olarak, tek-girişli tek-çıkışlı bir sistem olarak parametreleri ayarlanan DTS200 sisteminin 1. tankının sıvı seviyesi, sistemin denklemlerini gerektirmeden, sadece sistemden toplanan verilerle oluşturulan SVM modelini kullanan SVM-Tabanlı GPC yöntemiyle oldukça başarılı bir şekilde kontrol edilmiştir.

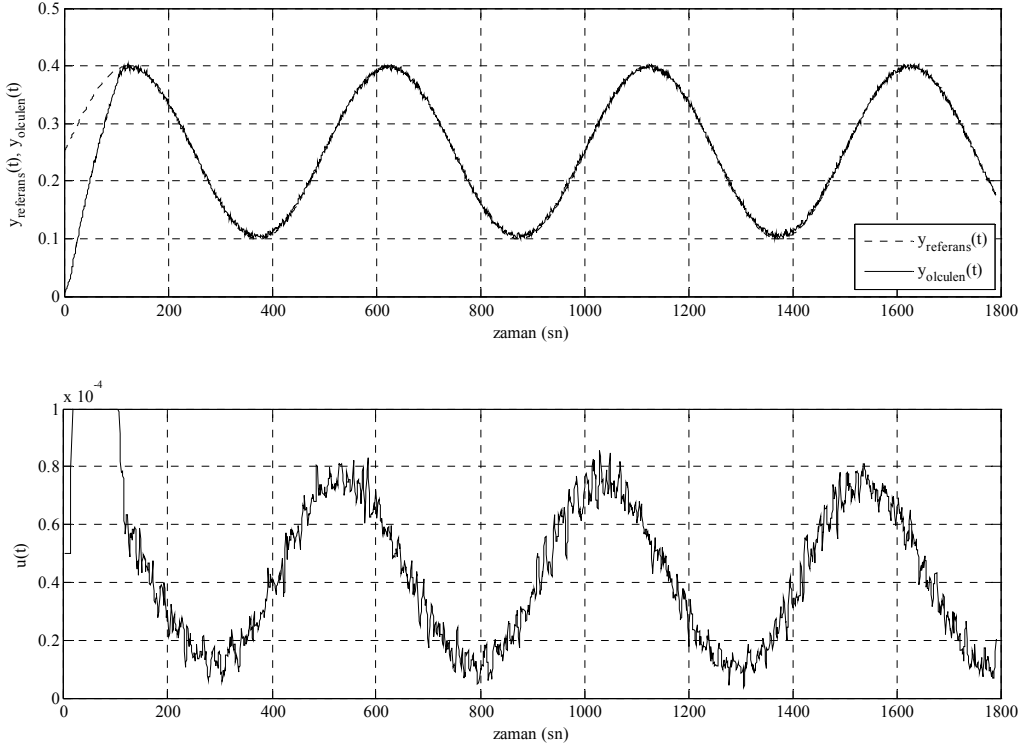


Şekil 3.1.9. Basamak biçimli referans için deneysel sonuçlar.

Şekil 3.1.10'da ise referans işareti,

$$y_{ref}(t) = 0.25 + 0.15 \sin 0.004\pi t$$

şeklinde değişen bir sinüs işaretidir. Burada amaç, sürekli değişen bir referans işaretine karşı SVM-Tabanlı GPC denetim sisteminin başarımını değerlendirmektir. Şekilden de görüldüğü gibi, SVM-Tabanlı GPC denetim ile 1. tankın sıvı seviyesi zamanla değişen bir referans girişini takip edebilmektedir.



Şekil 3.1.10. Sinüs biçimli referans için deneysel sonuçlar.

Bu deneysel sonuçlar en iyiye yakın bir denetim parametreleri setiyle alınmıştır. Bir sonraki kısımda ise, denetim parametrelerinin  $(N_y, N_u, \lambda)$  denetim başarımı üzerindeki etkileri incelenmiştir.

#### 3.1.2.4 Denetim Parametrelerinin Etkileri

Denetim parametreleri, kestirim ufku  $N_y$ , denetim ufku  $N_u$ , ve ceza parametresi  $\lambda$  şeklindedir. Her bir parametrenin etkisi belirlenirken, söz konusu parametre yavaş yavaş artırılırken, diğerleri sabit tutuldu. Her bir durumda, denetleyicinin başarımı aşağıdaki başarımla gösterildiği ölçüldü:

$$F = \sum_{n=1}^{1400} (\tilde{y}(n) - \hat{y}(n))^2 + \sum_{n=1}^{1400} \lambda (\Delta u(n))^2, \quad (3.1.30)$$

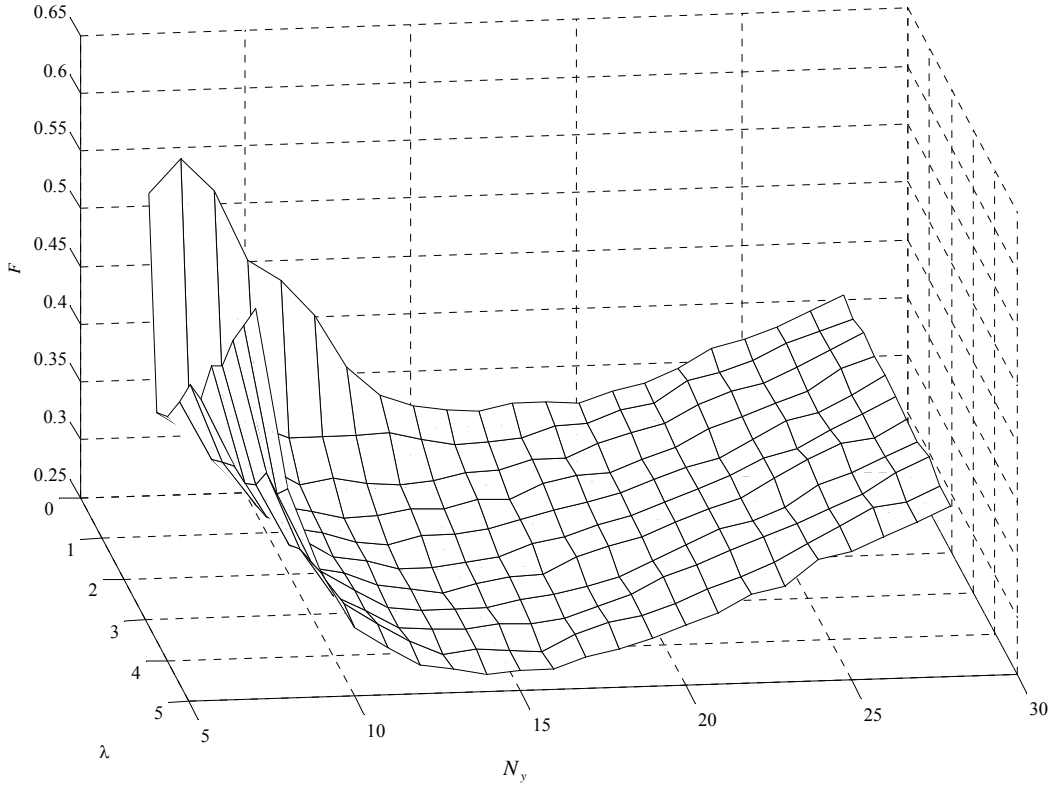
buradaki referans işareti önceki kısımda verilen basamak işaretinin daha kısa bir versiyonudur. Deneysel sonuçlarda,  $N_u$  parametresinin 2 değerinden itibaren önemli bir başarımla iyileştirmesi sağlamadığı gözlemlendiğinden değeri  $N_u = 2$  olarak sabit tutulmuştur. Şekil 3.1.11'de, farklı denetim parametreleri için denetleyicinin başarımı görülmektedir. Bu

şekilden, en iyi denetim parametreleri çiftinin  $N_y=15$  ve  $\lambda=4.5$  şeklinde olduğu görülmektedir. En iyi kestirim ufku parametresinin  $N_y=15$  olması beklenen bir sonuç iken en iyi ceza parametresinin  $\lambda=4.5$  olarak bulunması şaşırtıcı olabilir.

Bu aslında sistemdeki gürültüden kaynaklanan akla yatkın bir sonuçtur. Bir an için sistemde hiç gürültünün olmadığını ya da çok düşük genlikte olduğunu varsayalım. Bu durumda, en iyi  $\lambda$  parametresi sıfıra yakın bir değer olacaktır, çünkü denetim işaretleri arasındaki farkların çok düşük seviyelere inmesinden dolayı sadece referans işaret ile sistem çıkışı arasındaki farklar önemli hale gelecektir. Oysa sistemde belli bir gürültü işareti vardır. Sistemin çıkışındaki ölçümlerin gürültülü olması nedeniyle doğal olarak bu gürültüyü tolare etme eğiliminde olan denetim işaretinin de salınımlı olacağı beklenmektedir. Ancak, hatırlanacağı gibi,  $\lambda$  parametresi de ardışık denetim işaretleri arasındaki farkı cezalandırmaktadır. Sıfıra yakın bir  $\lambda$  parametresi denetim işaretindeki salınımları yeterince iyi cezalandıramadığından denklem (3.1.30) ile verilen başarımların sayısal değeri artar. Diğer taraftan, daha büyük  $\lambda$  parametresi değerleri, denetim işaretlerindeki salınımları daha çok bastırarak ve denetim başarımlarını iyileştirecektir. Ancak aşırı büyük  $\lambda$  değerlerine sahip bir denetleyici özellikle basamak şeklindeki referans işaretinin bir değerden diğerine geçişleri esnasında sistem çıkışının bu hızlı değişime ayak uydurmasını güçleştirecek ve dolayısıyla denklem (3.1.30) ile verilen başarımların sayısal değeri artacaktır. Dolayısıyla, en iyi  $\lambda$  parametresi değerinin ara bir değer çıkması tamamen gürültü ile ilgili olup akla yatkındır.

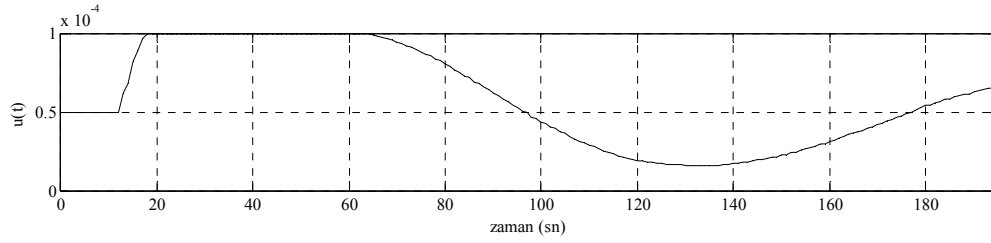
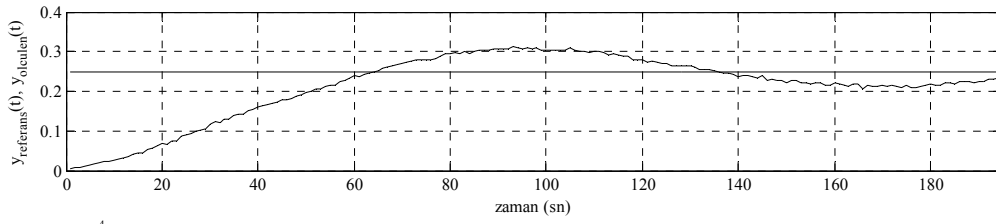
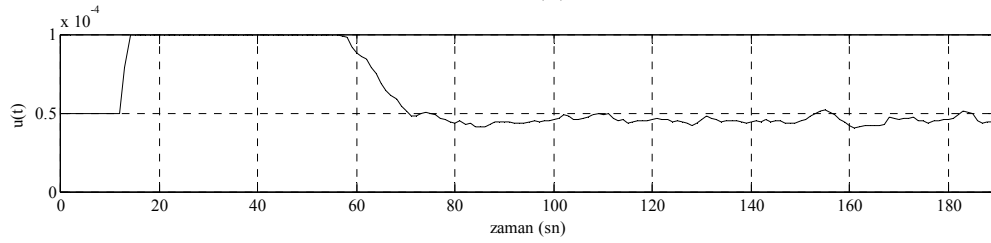
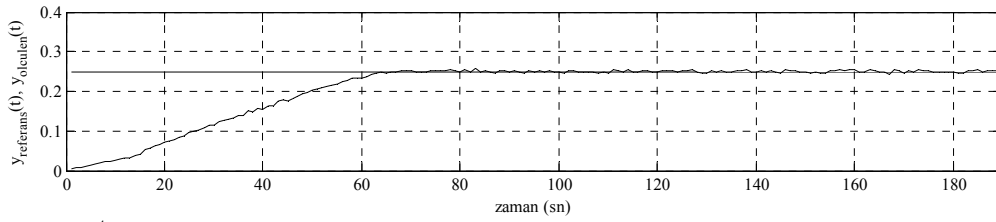
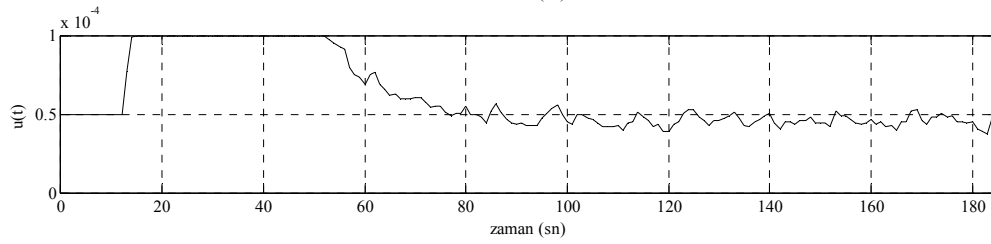
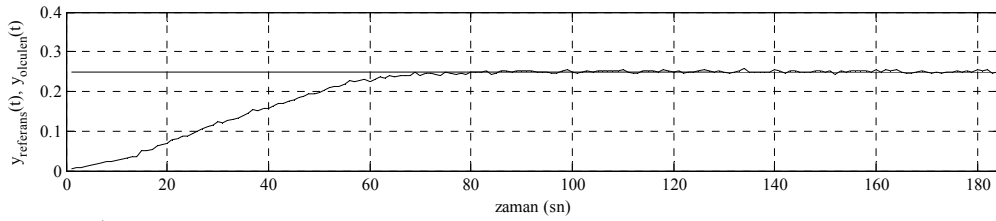
$N_y$  parametresinin etkilerine gelince, ki bu da oldukça etkin bir parametredir,  $N_y$  sadece başarımları değil aynı zamanda da kararlılığı ve işlemsel karmaşıklığı da etkilediğini göz önüne almak gerekir. Küçük  $N_y$  değerleri kararsızlığa, daha doğrusu çıkışta salınımlara yol açabiliyorken, çok büyük  $N_y$  değerleri de işlemsel karmaşıklığı ciddi boyutlara getirebilir. Bu yüzden bu parametre dikkatli seçilmelidir. Kararsızlık ve işlemsel karmaşıklık arasındaki dengeyi dikkate alarak ortalama bir  $N_y$  değeri araması yapıldığında, bu çalışmada  $N_y=15$  değerinin uygun olduğu görülmüştür.

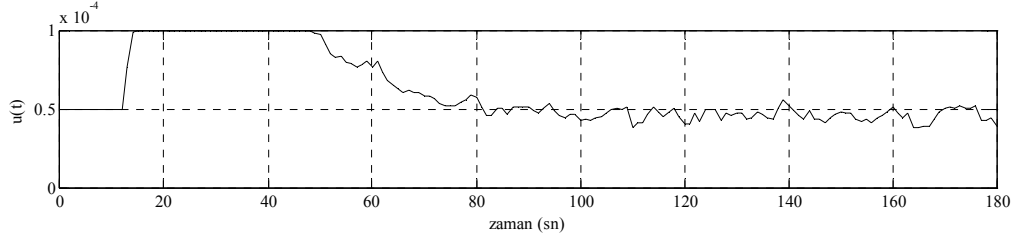
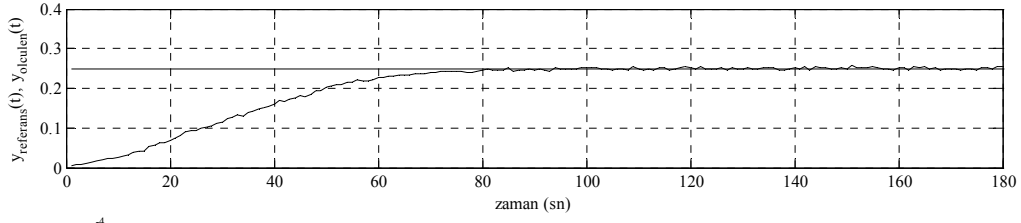
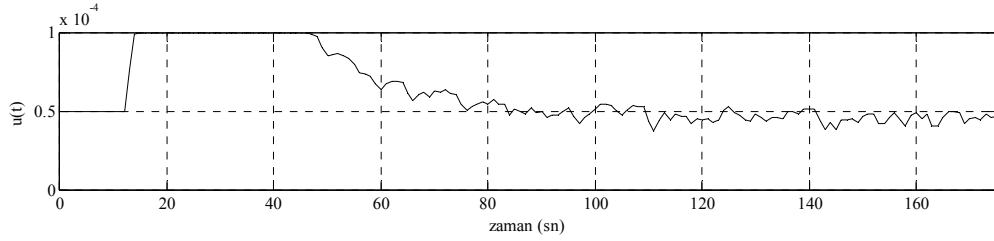
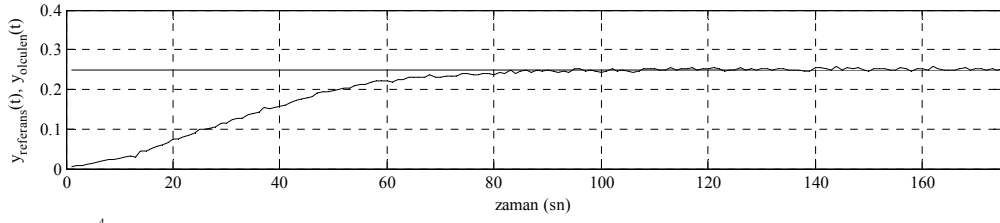
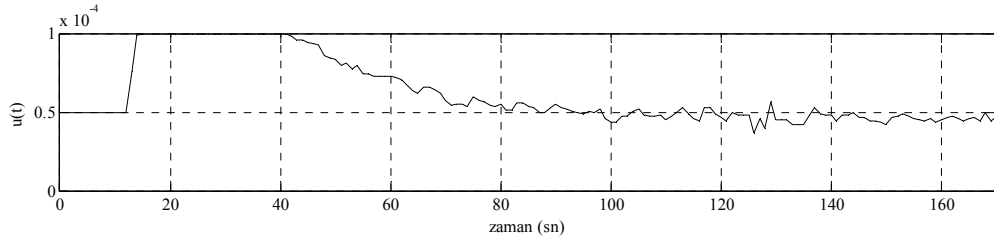
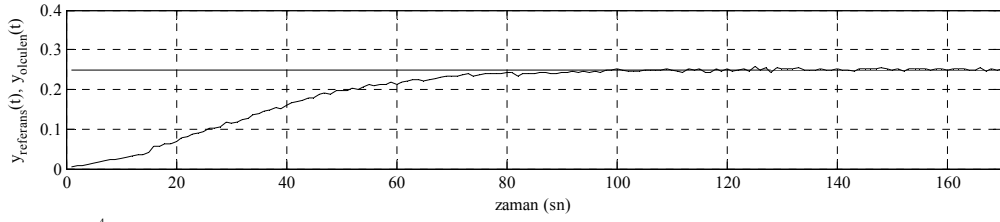


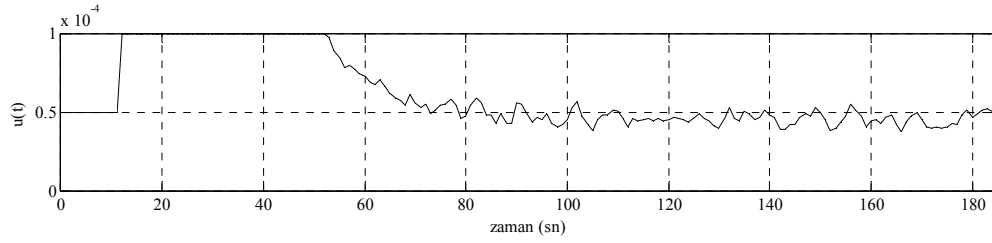
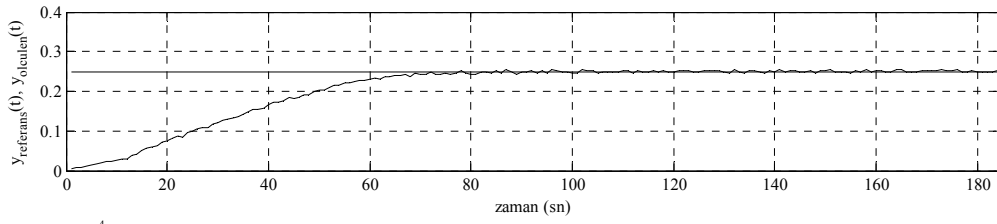
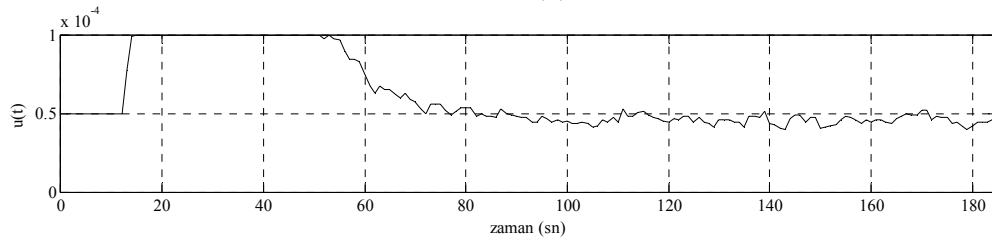
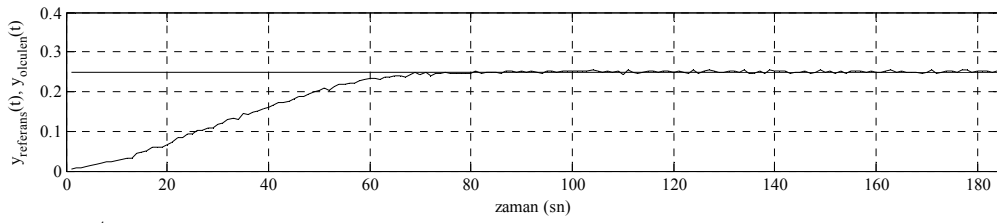
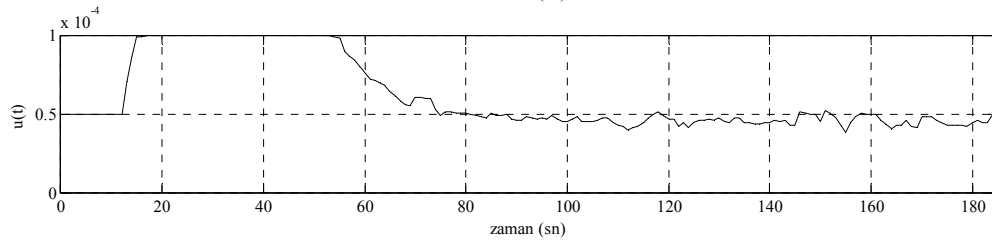
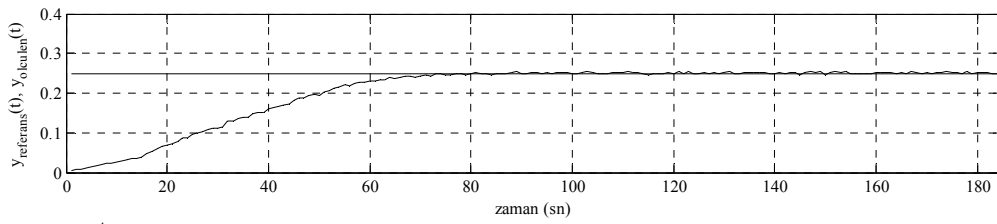


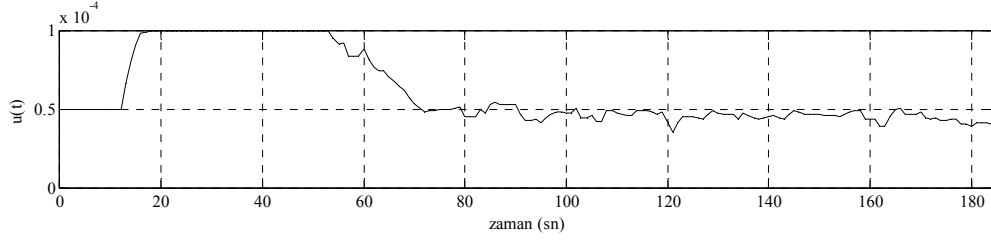
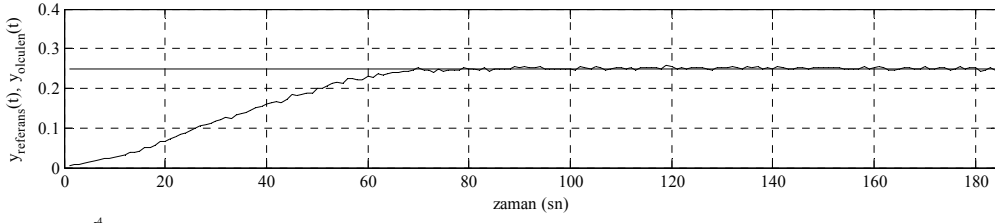
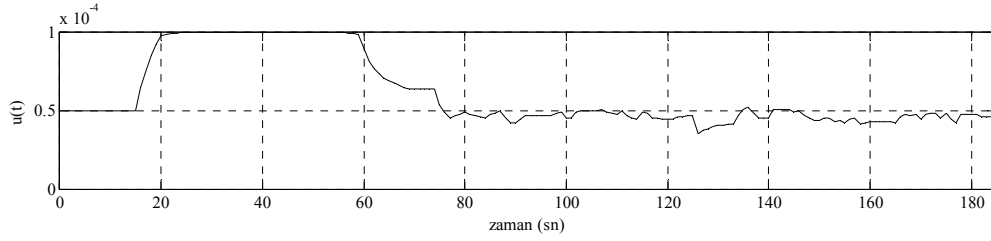
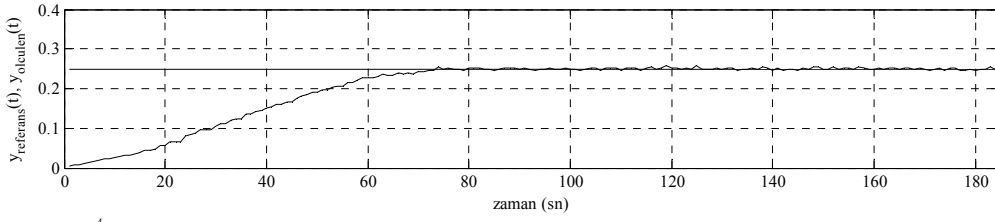
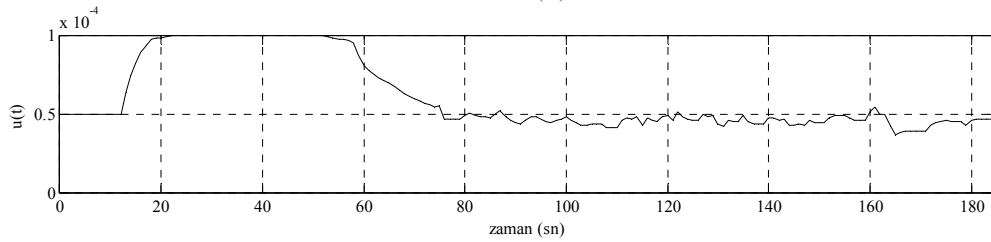
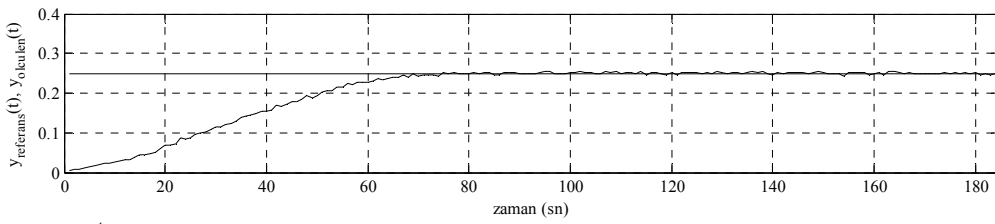
Şekil 3.1.11. Denetleyicinin farklı parametre değerleri için başarımları.

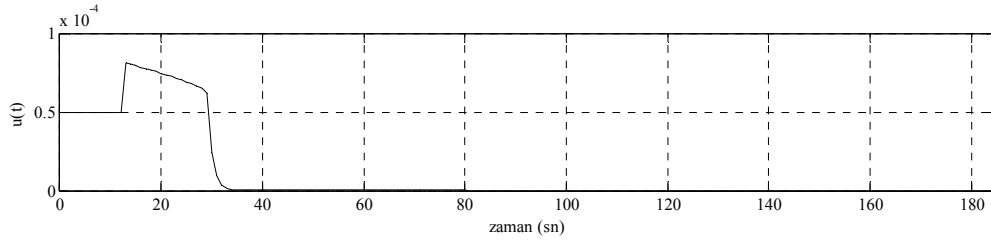
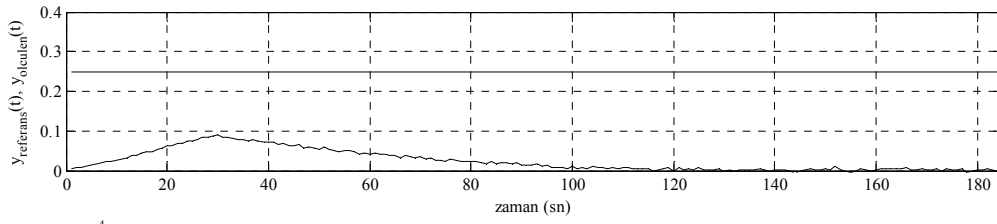
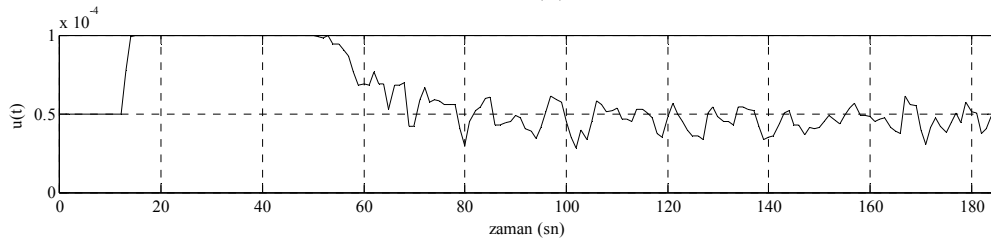
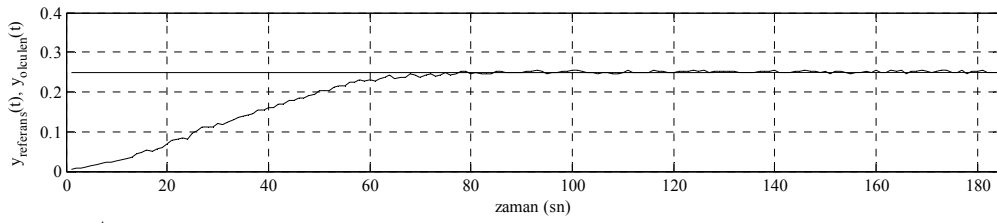
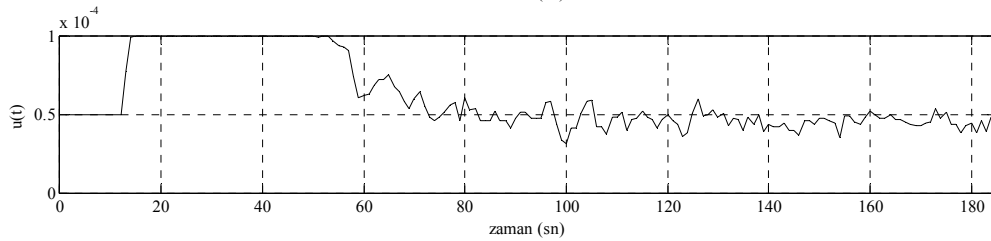
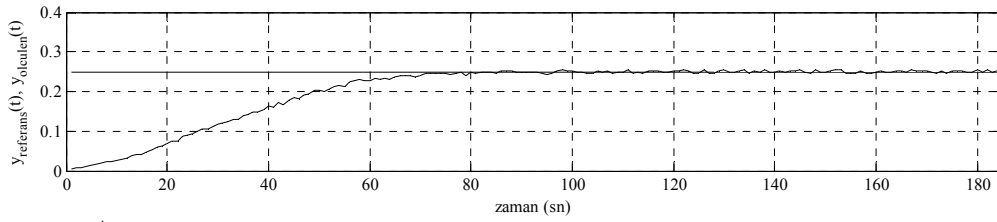
Denetim parametrelerinin sistem cevabının hızı üzerindeki etkilerini gözlemlemek için de, denetim parametrelerinden bir tanesi değiştirilirken diğerleri sabit tutulmuştur. Şekil 3.1.12'de  $N_u=2$  ve  $\lambda=5.0$  şeklinde sabit iken farklı  $N_y$  değerleri için sistemin cevabı görülmektedir. Şekilden de görüleceği gibi  $N_y$  arttıkça sistemin cevabında bir yavaşlama gözlenmektedir. Ancak,  $N_y=5$  gibi çok küçük değerleri için sistemin cevabının salınımlı olduğu gözlenmiştir. Şekil 3.1.13'de ise  $N_y=15$  ve  $\lambda=5.0$  şeklinde sabit iken farklı  $N_u$  değerleri için sistemin cevabı görülmektedir. Şekilden de görüleceği gibi, sistemin cevap hızı üzerinde  $N_u$ 'nun pek bir etkisinin olmadığı gözlenmiştir. Bütün  $N_u$  değerleri için sıvı seviyesinin yükselme zamanı aşağı yukarı aynı bulunmuştur. Son olarak, Şekil 3.1.14'de de  $N_y=15$  ve  $N_u=2$  şeklinde sabit iken farklı  $\lambda$  değerleri için sistemin cevabı görülmektedir.  $\lambda=0$  iken denetim sağlanamamıştır.  $\lambda$ 'nın değeri artırıldıkça sistemin cevabında az da olsa bir yavaşlama gözlemlendi. Ancak,  $\lambda$ 'nın etkisinin sistemin cevap hızından ziyade denetim işaretinin ani değişimleri üzerinde olduğu gözlenmiştir.

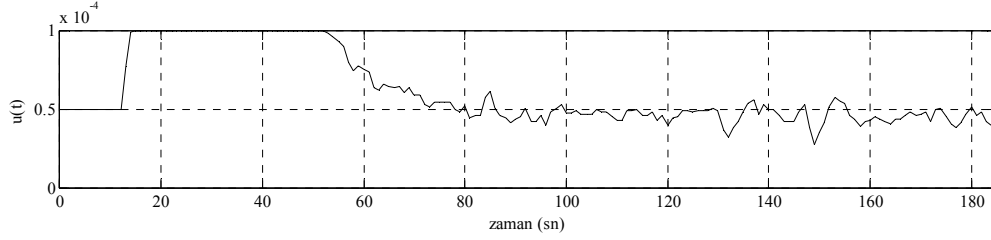
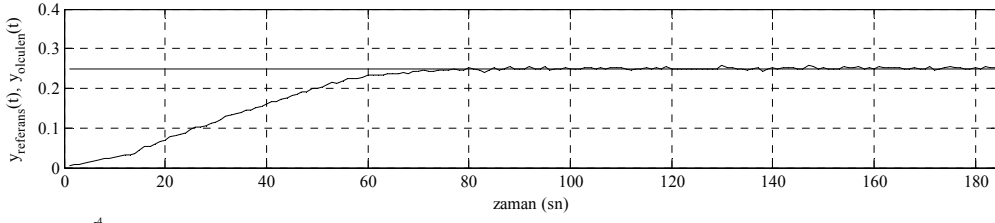
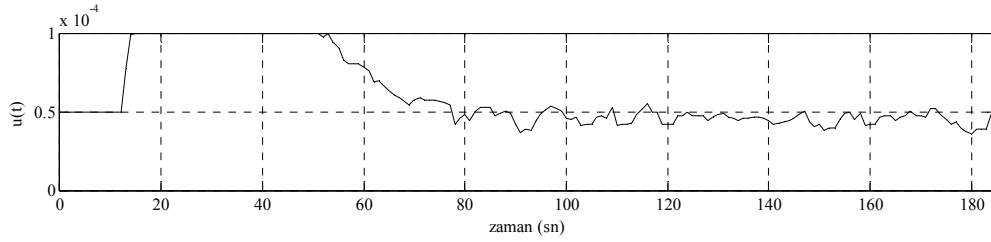
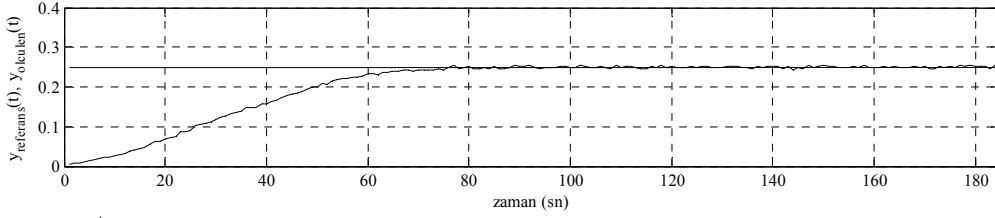
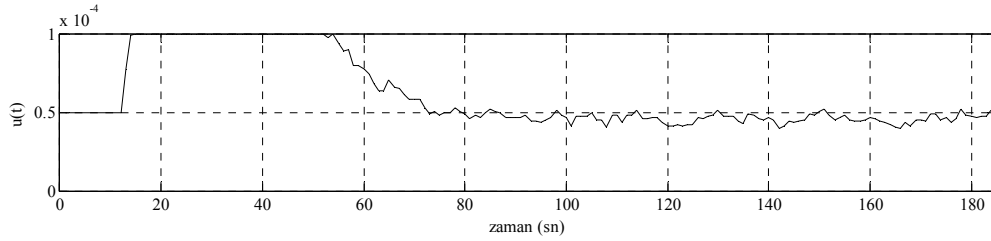
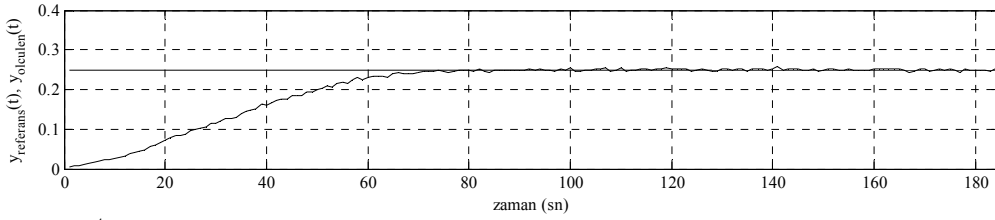
a)  $N_y = 5$ b)  $N_y = 10$ c)  $N_y = 15$

d)  $N_y = 20$ e)  $N_y = 25$ f)  $N_y = 30$ Şekil 3.1.12  $N_y$ 'nin sistem cevabına olan etkisi.

a)  $N_u = 1$ b)  $N_u = 2$ c)  $N_u = 3$

d)  $N_u = 4$ e)  $N_u = 5$ f)  $N_u = 6$ Şekil 3.1.13  $N_u$ 'nun sistem cevabına olan etkisi.

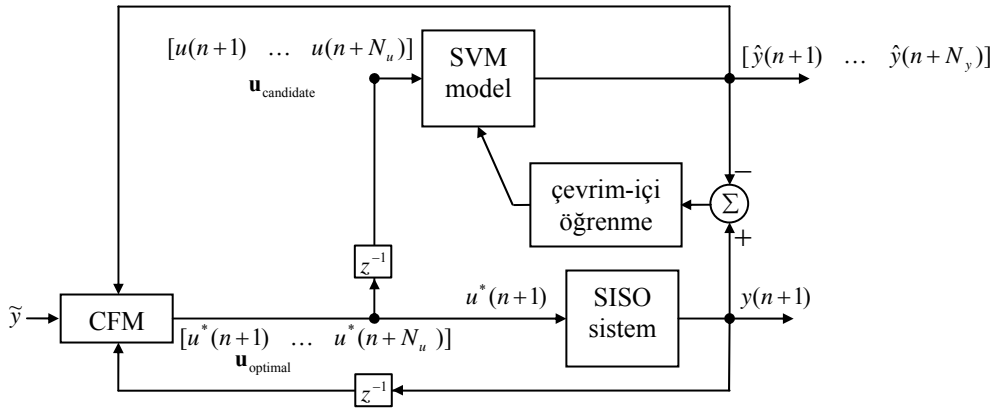
a)  $\lambda = 0$ b)  $\lambda = 1$ c)  $\lambda = 2$

d)  $\lambda = 3$ e)  $\lambda = 4$ f)  $\lambda = 5$ Şekil 3.1.14  $\lambda$ 'nın sistem cevabına olan etkisi.

### 3.2 Çevrim-içi SISO Denetim

#### 3.2.1 Yöntem

Çevrim-içi denetimin amacı genel olarak denetleyicinin değişen denetim koşullarına uyarlanır olmasıdır. Bunu yaparken de belli bir denetim başarımını sağlaması beklenir. Model tabanlı GPC yöntemlerinde denetleyicinin uyarlanır olabilmesi ancak ve ancak modelin uyarlanır tipte olması ile mümkün olabilir. Bu projede de uyarlanır bir SVM-Tabanlı GPC denetimi gerçekleştirmek için SVM modeli uyarlanır hale getirilmiştir. Bu amaçla SVM modelinin elde edilmesi için bir önceki kısımda kullanılan ve çevrim-dışı çalışan  $\varepsilon$ -SVR algoritması yerine  $\varepsilon$ -SVR algoritmasının yaptığı işi çevrim-içi olarak yapan AOSVR (Accurate On-line Support Vector Regression) algoritması (MA ve ark., 2003) kullanılmıştır. Artık önceden bir modelleme yapılmayıp denetime boş bir SVM modeliyle başlanacak, denetim ilerledikçe SVM modeli de gelişecektir. Bunun ayrıntıları ilerleyen kısımlarda verilecektir.



Şekil 3.2.1. Çevrim-içi SVM-Tabanlı GPC şeması.

Çevrim-içi SISO denetimi için (IPLIKCI, 2006b)'de önerilen yöntem, bu proje kapsamında üç-tanklı sıvı sisteminin çevrim-içi denetiminde kullanılmıştır. Bu amaçla, Şekil 3.2.1.'de görülen yapı gerçekleştirilmiştir. Burada görülen SISO sisteminin matematiksel denklemlerinin bilinmediği, sadece sistemin dinamiklerinin aşağıdaki gibi bir **Nonlinear AutoRegressive with eXogenous inputs** (NARX) modeliyle temsil edilebildiğini varsayıyoruz.

$$y(n) = f(u(n), \dots, u(n - n_1), y(n - 1), \dots, y(n - m_1)), \quad (3.2.1)$$

burada  $u(n)$   $n^{\text{inci}}$  zaman indeksinde sisteme uygulanan denetim işareti,  $y(n)$  ise sistemin çıkışıdır,  $n_1$  ve  $m_1$  de modelde yer alan geçmiş giriş ve çıkış gecikmelerini göstermektedir. Ayrıca, bu yapıda,  $\tilde{y}$  referans işareti ve  $\hat{y}$  da tahmindir. Bu yöntemde, SVM modeli sistemin  $\mathbf{u}_{\text{candidate}}$  aday denetim vektörüne karşılık gelecekteki yörüngesini tahmin eder ve aynı zamanda bedel fonksiyonunu en aza indirme (Cost Function Minimization - CFM)



bloğunda gerekli türev bilgisini sağlar. CFM bloğu referans işareti, ölçülen çıkışı ve SVM modelinin tahminlerini alır ve en iyilenmiş bir denetim vektörü oluşturur. Bu en iyilenmiş vektörün ilk elemanı sisteme denetim girdileri olarak uygulanır. Başlangıçta SVM modeli boştur ancak denetim ilerledikçe SVM modeli de gelişmeye başlar ve geliştikçe de daha iyi bir denetim başarımı sağlar. Çevrim-içi denetimde, denetim ve modelleme aynı anda yapılır. Çevrim-içi SVM-Tabanlı GPC yönteminde de SVM modelinin çevrim-içi bir tarzda elde edilmesi AOSVR algoritmasıyla mümkün olur. Yani, AOSVR algoritmasıyla, yeni bir eğitim veri noktası geldiğinde, önceden öğrendiklerini kaybetmeden yeni verinin SVM modeline çevrim-içi olarak enjekte edilmesi sağlanır. Aslında bu algoritma, çevrim-dışı çalışan  $\varepsilon$ -SVR algoritmasının yaptığı işin tıpatıp aynısını yapar, yani belli veri seti kullanarak  $\varepsilon$ -SVR algoritması ile çevrim-dışı elde edilen bir SVM modelinin aynısını aynı veri setinin çevrim-içi olarak kullanan AOSVR algoritması ile elde etmek mümkündür. AOSVR algoritması çevrim-içi çalıştığı için  $\varepsilon$ -SVR'den çok daha hızlıdır. Şimdi AOSVR algoritmasına daha yakından bakalım. AOSVR algoritması,  $\varepsilon$ -SVR algoritmasının bir devamı niteliğindedir. O yüzden ilk olarak  $\varepsilon$ -SVR algoritması ile başlayalım: Elimizde SISO bir sistemin giriş-çıkış ilişkisine ait aşağıdaki gibi bir eğitim veri seti olsun.

$$\begin{aligned} T &= \{u(k), \dots, u(k - n_1), y(k - 1), \dots, y(k - m_1); y(k)\}_{k=1}^{k=N} \\ &= \{\mathbf{x}_k; y(k)\}_{k=1}^{k=N} \end{aligned} \quad (3.2.2)$$

burada  $\mathbf{x}_k \in X \subseteq \mathfrak{R}^{n_1 + m_1 + 1}$  büyüklüğü  $k^{\text{inci}}$  giriş veri noktasını ve  $y(k) \in Y \subseteq \mathfrak{R}$  da buna karşı düşen çıkış değerini göstermektedir. Bu verileri kullanarak sistemin giriş-çıkış ilişkisini temsil edebilecek bir model oluşturmak istiyoruz. Bu amaçla, veri seti  $T$ 'yi kullanarak sistemin modelini elde edecek SVM algoritması şu şekildedir. SVM bağlanım modelinin birincil formu (3.2.3) denkleminde görüldüğü gibidir; bu model öznitelik uzayı olan  $\mathbf{F}$  uzayında doğrusaldır.

$$\hat{y}(i) = \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b \quad (3.2.3)$$

burada  $\mathbf{w}$  vektörü  $\mathbf{F}$  öznitelik uzayında bir vektördür,  $\Phi(\cdot)$  fonksiyonu giriş uzayından öznitelik uzayına bir dönüşümdür,  $b$  büyüklüğü bir yanlılık (bias) terimidir ve  $\langle \cdot \rangle$  operatörü de  $\mathbf{F}$  uzayında bir iç-çarpımı göstermektedir. SVM bağlanım algoritmaları genel olarak bir bağlanım problemini ikincil uzayda bir en iyileme problemi olarak görürler. SVM modelinin ikincil uzaydaki formu şu şekildedir:

$$\hat{y}(i) = \sum_{j=1}^N \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b, \quad (3.2.4)$$

burada  $\alpha_j$ 'ler her bir eğitim verisinin katsayılarıdır ve  $K(\mathbf{x}_i, \mathbf{x}_j)$  da şu şekilde verilen bir çekirdek fonksiyonudur:  $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = K_{ij}$ . Çekirdek fonksiyonu, öznelik uzayında iç-çarpım işlemini gerçekleştirdiğinden dolayı  $\Phi(\mathbf{x})$  fonksiyonunun açık formunun bilinmesine gerek kalmaz. Denklem (3.2.4) ile verilen modelde, sıfırdan farklı bir  $\alpha_j$  değerine karşı düşen  $\mathbf{x}_j$  gibi bir eğitim noktasına *destek vektörü* denmektedir. Çevrim-içi SVM-Tabanlı GPC yönteminde kullanılan AOSVR algoritmasına öncülük eden  $\varepsilon$ -SVR algoritması aşağıdaki gibi verilen Vapnik's  $\varepsilon$ -insensitive kayıp fonksiyonunu ( $L(\varepsilon, y, \hat{y})$ ) kullanmaktadır:

$$L(\varepsilon, y, \hat{y}) = \begin{cases} 0 & y - \hat{y} \leq \varepsilon \\ y - \hat{y} & y - \hat{y} > \varepsilon \end{cases} \quad (3.2.5)$$

$\varepsilon$ -SVR algoritması, bağlanım probleminin birincil formunu şu şekilde formülize eder:

$$\min_{\mathbf{w}, b, \xi, \xi^*} P_\varepsilon = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (3.2.6)$$

kısıtlar:

$$\begin{aligned} y(i) - \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - b &\leq \varepsilon + \xi_i, \quad i = 1, \dots, N \\ \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b - y(i) &\leq \varepsilon + \xi_i^*, \quad i = 1, \dots, N \\ \xi_i, \xi_i^* &\geq 0, \quad i = 1, \dots, N \end{aligned} \quad (3.2.7)$$

burada  $\varepsilon$  değeri çıkış için tolare edilebilen en büyük hata değeridir,  $\xi_i$ 'ler ve  $\xi_i^*$ 'ler gevşek değişkenlerdir,  $\|\cdot\|$  operatörü Öklit normudur ve  $C$  de modelin karmaşıklığı ile  $\varepsilon$ 'den daha büyük hatalara gösterilen tolerans derecesi arasında denge kurmayı sağlayan bir parametredir. Böylece, en iyileme probleminin ikincil formu aşağıdaki gibi bir karesel programlama (Quadratic Programming - QP) problemine dönüşür:

$$\min_{\beta, \beta^*} D_\varepsilon = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K_{ij} (\beta_i - \beta_i^*) (\beta_j - \beta_j^*) + \varepsilon \sum_{i=1}^N (\beta_i + \beta_i^*) - \sum_{i=1}^N y(i) (\beta_i - \beta_i^*) \quad (3.2.8)$$

kısıtlar:

$$0 \leq \beta_i, \beta_i^* \leq C, \quad \sum_{i=1}^N (\beta_i - \beta_i^*) = 0, \quad i = 1, \dots, N \quad (3.2.9)$$

Denklem (3.2.8 ve 3.2.9) ile verilen QP probleminin çözümü  $\beta_i$ 'ler ve  $\beta_i^*$ 'lerin optimum değerlerini verir. Modeldeki  $b$ 'ler,  $0 \leq \beta_i - \beta_i^* \leq C$  koşulunun sağlandığı her bir  $\mathbf{x}_i$  destek vektörü için  $y(i) - \hat{y}(i) = \varepsilon$  koşulu sağlanacak şekilde bulunurlar. Eğer  $j = 1, \dots, N$  olmak

üzere  $\mathbf{x}_j$  destek vektörü için  $\alpha_j$  gibi yeni bir katsayı  $\alpha_j = \beta_j - \beta_j^*$  şeklinde tanımlanırsa, o zaman Denklem (3.2.4)'teki gibi bir SVM modeli elde edilir. Hatta, sadece destek vektörleri dikkate alındığında model şu hale gelir:

$$\hat{y}(i) = \sum_{\substack{j=1 \\ j \in SV}}^{\#SV} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \quad (3.2.10)$$

burada #SV modeldeki destek vektör sayısıdır.

AOSVR algoritması bu noktadan itibaren başlamaktadır. AOSVR algoritması, artımlı (incremental learning) ve azalımlı (decremental unlearning) olmak üzere iki farklı güncellemeye yapabilir. Artımlı algoritma, mevcut bir SVM modeline yeni bir veri noktası eklenmek istendiğinde önceki öğrenilenleri kaybetmeden o veri noktası ile SVM modelinin güncellenmesini sağlar. Azalımlı algoritma ise bunun tam tersinin yapar, yani mevcut bir SVM modelinden bir veri noktasının çıkarılması işini önceden öğrenilen veri noktalarını kaybetmeden gerçekleştirir. Bu özellik diğer regresyon yöntemlerinde olmayan üstün bir özelliktir. Çünkü veriyi bir sisteme öğretmek mümkündür ama bir veriyi diğer verilerden öğrendiklerinin kaybetmeden unutturmak kolay değildir. Şimdi ilk olarak AOSVR ile artımlı algoritmayı görelim: Denklem (3.2.8) ile verilen dual biçimin Lagrangian formülasyonu şu şekildedir:

$$\begin{aligned} L_D = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K_{ij} (\beta_i - \beta_i^*) (\beta_j - \beta_j^*) + \varepsilon \sum_{i=1}^N (\beta_i + \beta_i^*) - \sum_{i=1}^N y(i) (\beta_i - \beta_i^*) \\ & - \sum_{i=1}^N (\delta_i \beta_i + \delta_i^* \beta_i^*) + \zeta \sum_{i=1}^N (\beta_i - \beta_i^*) + \sum_{i=1}^N (\kappa_i (\beta_i - C) + \kappa_i^* (\beta_i^* - C)) \end{aligned} \quad (3.2.11)$$

burada  $\delta_i, \delta_i^*, \kappa_i, \kappa_i^*$  ve  $\zeta$  büyüklükleri Lagrange çarpanlarıdır.  $L_D$ 'nin optimizasyonu aşağıdaki Karush-Kuhn-Tucker (KKT) koşullarını verir:

$$\frac{\partial L_D}{\partial \beta_i} = 0 \rightarrow \sum_{i=1}^N K_{ij} (\beta_i - \beta_i^*) + \varepsilon - y_i + \zeta - \delta_i + \kappa_i = 0, \quad (3.2.12)$$

$$\frac{\partial L_D}{\partial \beta_i^*} = 0 \rightarrow \sum_{i=1}^N K_{ij} (\beta_i - \beta_i^*) - \varepsilon - y_i + \zeta + \delta_i^* - \kappa_i^* = 0, \quad (3.2.13)$$

$$\delta_i, \delta_i^* \geq 0 \quad \delta_i \beta_i = 0 \quad \delta_i^* \beta_i^* = 0 \quad (3.2.14)$$

$$\kappa_i, \kappa_i^* \geq 0 \quad \kappa_i (\beta_i - C) = 0 \quad \kappa_i^* (\beta_i^* - C) = 0 \quad (3.2.15)$$

Eğer  $h(\mathbf{x}_i)$  gibi bir destek vektörü için hata fonksiyonu şu şekilde tanımlanırsa,

$$h(\mathbf{x}_i) \equiv \hat{y}(\mathbf{x}_i) - y_i = \sum_{j=1}^N \alpha_j K_{ij} + b - y_i, \quad (3.2.16)$$

ki burada  $\alpha_j = \beta_j - \beta_j^*$  büyüklüğü  $j = 1, 2, \dots, \#SV$  için  $\mathbf{x}_j$ 'e karşı düşen yeni katsayıdır, o zaman KKT koşulları aşağıdaki biçimde yeniden yazılabilir:

$$h(\mathbf{x}_i) \geq \varepsilon, \quad \alpha_i = -C, \quad (3.2.17)$$

$$h(\mathbf{x}_i) = \varepsilon, \quad -C < \alpha_i < 0, \quad (3.2.18)$$

$$-\varepsilon \leq h(\mathbf{x}_i) \leq \varepsilon, \quad \alpha_i = 0, \quad (3.2.19)$$

$$h(\mathbf{x}_i) = -\varepsilon, \quad 0 < \alpha_i < C, \quad (3.2.20)$$

$$h(\mathbf{x}_i) \leq -\varepsilon, \quad \alpha_i = C, \quad i = 1, 2, \dots, N \quad (3.2.21)$$

Eğitim kümesindeki herhangi bir eğitim verisi  $\mathbf{x}_i$ , sahip olduğu  $\alpha_i$  değerine göre aşağıdaki üç kümeden birine aittir:

$$\mathbf{x}_i \in \mathbf{E}, \text{ eğer } |\alpha_i| = C \text{ ise}$$

$$\mathbf{x}_i \in \mathbf{S}, \text{ eğer } 0 < |\alpha_i| < C \text{ ise}$$

$$\mathbf{x}_i \in \mathbf{R}, \text{ eğer } |\alpha_i| = 0 \text{ ise.}$$

Artımlı algoritmada,  $\mathbf{x}_c$  gibi yeni bir eğitim verisi geldiğinde, ona karşı düşen  $\alpha_c$  değeri, daha sonradan  $\Delta\alpha_c$  değerine ayarlanmak üzere, başlangıçta sıfır yapılır. Ardından, sistem KKT koşullarına göre dengede kalacak şekilde mümkün olan en büyük değişim olan  $\Delta\alpha_c$  hesaplanır. Bu işlem esnasında, bazı veri noktaları bir kümeden diğerine geçerken bazıları da yerinde kalır. Yeni gelen veri noktası da dahil olmak üzere tüm veriler KKT koşullarını sağlayana kadar bu işlem tekrarlanır. Önceden eğitimde kullanılmış  $\mathbf{x}_c$  gibi bir eğitim verisi eğitim kümesinden çıkarılmak istendiğinde ise artımlı algoritmanın tamamen tersi olan azalımlı algoritma benzer şekilde kullanılabilir. Daha fazla ayrıntı için (MA ve ark., 2003)'e başvurulabilir.

Şimdi çevrim-içi SVM-Tabanlı GPC yöntemine geri dönelim. Bu yöntemde, örnekleme periyotlarının her birinde ilk olarak, aşağıdaki gibi bir aday denetim vektörü oluşturulur:

$$\mathbf{u}_{\text{candidate}} = [u(n+1) \quad \dots \quad u(n+N_u)] \quad (3.2.22)$$

burada  $N_u$  denetim girdisine ilişkin denetim ufkudur. Bu denetim vektörü sisteme doğrudan uygulanmaz çünkü henüz en iyilenmiş değildir. Bu en iyileme, SVM modelinin aday denetim vektörüne karşı ürettiği tahminlerle yapılmaktadır. Bura en iyilemede amaç,  $\mathbf{u}_{\text{candidate}}$  vektörünü uygun bir şekilde değiştirerek tahmin hatalarını en aza indirirken denetim

işaretinde de çok fazla değişime yol açmamaktır. Bunu aşağıdaki amaç fonksiyonu ile göstermek daha kolay olacaktır:

$$F = \sum_{k=1}^{N_y} (\tilde{y}(n+k) - \hat{y}(n+k))^2 + \sum_{k=1}^{N_u} \lambda (\Delta u(n+k))^2 \quad (3.2.23)$$

burada  $N_y$  kestirim ufkudur ve  $\lambda$  da denetim girdisi değişimleri için ceza terimleridir. Bu en iyilemeyi gerçekleştirmek için aşağıdaki gibi genel bir güncelleme kuralı kullanılır:

$$\mathbf{u}_{\text{optimal}} = \mathbf{u}_{\text{candidate}} + s\mathbf{p}, \quad (3.2.24)$$

burada  $\mathbf{p}$  arama yönü ve  $s$  de adım-uzunluğu olarak adlandırılır. Ardından, en iyilenmiş denetim vektöründe  $(n+1)$  zaman indeksine karşı düşen eleman, yani  $u^*(n+1)$  elemanı, sisteme giriş olarak uygulanır. En iyileme literatüründe,  $\mathbf{p}$  vektörünün bulunması için birinci- ve ikinci-dereceden türev tabanlı yöntemler mevcuttur (NOCEDAL ve WRIGHT, 1999; VENKATARAMAN, 2002). Birinci-dereceden metotlar gradyant vektörünü kullanırken, ikinci-dereceden metotlar buna ilave olarak Hessian matrisini kullanırlar. Bu çalışmada, *Değiştirilmiş Newton (Modified Newton)* adı verilen yöntem kullanılmıştır. Bu yöntemde, arama yönü, pozitif tanımlı Hessian matrisinin tersi ile birlikte en büyük artım yönünü gösteren  $\mathbf{g}$  gradyant vektörüne zıt yönde seçilir. Başka bir deyişle,

$$\mathbf{p} = -\mathbf{H}^{-1} \mathbf{g}, \quad (3.2.25)$$

burada,

$$\mathbf{g} = \frac{\partial F}{\partial \mathbf{u}_{\text{candidate}}} = \left[ \frac{\partial F}{\partial u(n+1)} \quad \cdots \quad \frac{\partial F}{\partial u(n+N_u)} \right]^T. \quad (3.2.26)$$

ve

$$\mathbf{H} = \frac{\partial^2 F}{\partial \mathbf{u}_{\text{candidate}}^2} = \begin{bmatrix} \frac{\partial^2 F}{\partial u(n+1)\partial u(n+1)} & \cdots & \frac{\partial^2 F}{\partial u(n+1)\partial u(n+N_u)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial u(n+N_u)\partial u(n+1)} & \cdots & \frac{\partial^2 F}{\partial u(n+N_u)\partial u(n+N_u)} \end{bmatrix}. \quad (3.2.27)$$

şeklindedir. Gradyant vektörü  $\mathbf{g}$ 'nin hesabında gerekli olan kısmi türevler aşağıdaki şekilde bulunur:

$$\begin{aligned} \frac{\partial F}{\partial u(n+h)} &= -2 \sum_{k=1}^{N_y} (\tilde{y}(n+k) - \hat{y}(n+k)) \frac{\partial \hat{y}(n+k)}{\partial u(n+h)} \\ &\quad + 2\lambda \sum_{k=1}^{N_u} (\Delta u(n+k)) (\delta_{h,k} - \delta_{h,k-1}) \end{aligned} \quad (3.2.28)$$

burada  $\delta_{h,k}$  Kronecker Delta fonksiyonudur. Benzer şekilde, Hessian matrisi  $\mathbf{H}$ 'nin hesabında gerekli olan kısmi türevler de aşağıdaki şekilde bulunur:

$$\begin{aligned} \frac{\partial^2 F}{\partial u(n+h)\partial u(n+m)} &= 2 \sum_{k=1}^{N_y} \frac{\partial \hat{y}(n+k)}{\partial u(n+h)} \frac{\partial \hat{y}(n+k)}{\partial u(n+m)} \\ &\quad - 2 \sum_{k=1}^{N_y} \frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h)\partial u(n+m)} (\tilde{y}(n+k) - \hat{y}(n+k)) \\ &\quad + 2\lambda \sum_{k=1}^{N_u} (\delta_{h,k} - \delta_{h,k-1})(\delta_{m,k} - \delta_{m,k-1}) \end{aligned} \quad (3.2.29)$$

İlerleyen kısımlarda, SVM modeli kullanılarak,  $\hat{y}(n+k)$  tahminleri ile  $\frac{\partial \hat{y}(n+k)}{\partial u(n+h)}$  ve

$\frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h)\partial u(n+m)}$  kısmi türevlerinin nasıl bulunacağı gösterilmiştir.

Eğer (IPLIKCI, 2006b)'de SISO sistemler için önerilen çevrim-içi SVM-Tabanlı GPC yöntemini izleyecek olursak, ilk olarak şu anki durum vektörünü aşağıdaki şekilde oluşturulmalıdır:

$$\mathbf{c}(n) = [u(n), \dots, u(n-n_1), y(n-1), \dots, y(n-m_1)]^T, \quad (3.2.30)$$

Böylelikle, SVM modelinin çıkışı şu hale gelir,

$$\hat{y}(n) = \sum_{j=1}^{\#SV} \alpha_j K(\mathbf{c}(n), \mathbf{x}_j) + b. \quad (3.2.31)$$

Bu çalışmada, aşağıdaki gibi verilen radyal tabanlı fonksiyon (Radial Basis Function - RBF) çekirdek fonksiyonu olarak seçilmiştir:

$$K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2}\right),$$

burada  $\sigma$  genişlik parametresidir. Eğer  $d_{j,n}$  uzaklığı,  $j^{\text{inci}}$  destek vektörü  $\mathbf{x}_j$  ile şu anki durum vektörü  $\mathbf{c}(n)$  arasındaki Öklit uzaklığı olarak aşağıdaki gibi tanımlanırsa,

$$\begin{aligned} d_{j,n} &= (\mathbf{c}(n) - \mathbf{x}_j)^T (\mathbf{c}(n) - \mathbf{x}_j) \\ &= \sum_{i=0}^{n_1} (x_{j,i+1} - u(n-i))^2 + \sum_{i=1}^{m_1} (x_{j,i+1+n_1} - y(n-i))^2, \end{aligned} \quad (3.2.32)$$

o zaman, çekirdek fonksiyonu şu şekilde yazılabilir,

$$K(\mathbf{c}(n), \mathbf{x}_j) = \exp\left(-\frac{d_{j,n}}{2\sigma^2}\right), \quad (3.2.33)$$

ve, bunun sonucunda, SVM modelinin çıkışı şu hale gelir,

$$\hat{y}(n) = \sum_{j=1}^{\#SV} \alpha_j \exp\left(-\frac{d_{j,n}}{2\sigma^2}\right) + b. \quad (3.2.34)$$

Böylelikle, ileri yönelik tahminler aşağıdaki formülle yapılabilir:

$$\hat{y}(n+k) = \sum_{j=1}^{\#SV} \alpha_j \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right) + b, \quad k=1, \dots, N_y, \quad (3.2.35)$$

burada,

$$d_{j,n+k} = \sum_{i=0}^{n_1} \begin{cases} (x_{j,i+1} - u(n+k-i))^2 & k - N_u < i \\ (x_{j,i+1} - u(n+N_u))^2 & k - N_u \geq i \end{cases} + \sum_{i=1}^{\min(k, m_1)} (x_{j,i+1+n_1} - \hat{y}(n+k-i))^2 + \sum_{i=k+1}^{m_1} (x_{j,i+1+n_1} - y(n+k-i))^2 \quad (3.2.36)$$

İleri yönelik tahminler bulunduktan sonra,  $\frac{\partial \hat{y}(n+k)}{\partial u(n+h)}$  kısmi türevleri artık şu şekilde bulunabilir:

$$\frac{\partial \hat{y}(n+k)}{\partial u(n+h)} = \sum_{j=1}^{\#SV} \alpha_j \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h)}, \quad (3.2.37)$$

burada,

$$\begin{aligned} \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h)} &= \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial d_{j,n+k}} \frac{\partial d_{j,n+k}}{\partial u(n+h)} \\ &= -\frac{1}{2\sigma^2} \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right) \frac{\partial d_{j,n+k}}{\partial u(n+h)} \end{aligned} \quad (3.2.38)$$

ve

$$\begin{aligned} \frac{\partial d_{j,n+k}}{\partial u(n+h)} &= -2 \sum_{i=0}^{n_1} \begin{cases} (x_{j,i+1} - u(n+k-i)) \delta_{k-i,h} & k - N_u < i \\ (x_{j,i+1} - u(n+N_u)) \delta_{N_u,h} & k - N_u \geq i \end{cases} \\ &\quad - 2 \sum_{i=1}^{\min(k, m_1)} (x_{j,i+1+n_1} - \hat{y}(n+k-i)) \frac{\partial \hat{y}(n+k-i)}{\partial u(n+h)} \Gamma_{k-i-h} \end{aligned}, \quad (3.2.39)$$

$$k=1, \dots, N_y \quad h=1, \dots, N_u.$$

Benzer şekilde,  $\frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h) \partial u(n+m)}$  kısmi türevleri şu şekilde bulunur:

$$\frac{\partial^2 \hat{y}(n+k)}{\partial u(n+h)\partial u(n+m)} = \sum_{j=1}^{\#SV} \alpha_j \frac{\partial^2 \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h)\partial u(n+m)}, \quad (3.2.40)$$

burada,

$$\begin{aligned} \frac{\partial^2 \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u(n+h)\partial u(n+m)} &= \frac{\partial^2 \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial d_{j,n+k}^2} \frac{\partial d_{j,n+k}}{\partial u(n+h)} \frac{\partial d_{j,n+k}}{\partial u(n+m)} \\ &+ \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial d_{j,n+k}} \frac{\partial^2 d_{j,n+k}}{\partial u(n+h)\partial u(n+m)} \\ &= \frac{1}{4\sigma_y^4} \exp\left(-\frac{d_{j,n+k}}{2\sigma_y^2}\right) \frac{\partial d_{j,n+k}}{\partial u_{n+h}} \frac{\partial d_{j,n+k}}{\partial u_{n+m}} \\ &- \frac{1}{2\sigma_y^2} \exp\left(-\frac{d_{j,n+k}}{2\sigma_y^2}\right) \frac{\partial^2 d_{j,n+k}}{\partial u_{n+h}\partial u_{n+m}} \end{aligned} \quad (3.2.41)$$

ve

$$\begin{aligned} \frac{\partial^2 d_{j,n+k}}{\partial u(n+h)\partial u(n+m)} &= 2 \sum_{i=0}^{n_1} \begin{cases} \delta_{k-i,h,m} & k - N_u < i \\ \delta_{N_u,h,m} & k - N_u \geq i \end{cases} \\ &- 2 \sum_{i=1}^{\min(k,m_1)} (x_{j,i+1+n_1} - \hat{y}(n+k-i)) \frac{\partial^2 \hat{y}(n+k-i)}{\partial u(n+h)\partial u(n+m)} \Gamma_{k-i-\max(h,m)} \\ &+ 2 \sum_{i=1}^{\min(k,m_1)} \frac{\partial \hat{y}(n+k-i)}{\partial u(n+h)} \frac{\partial \hat{y}(n+k-i)}{\partial u(n+m)} \Gamma_{k-i-\max(h,m)} \\ &k = 1, \dots, N_y \quad h = 1, \dots, N_u, \end{aligned} \quad (3.2.42)$$

burada  $\Gamma_x$  aşağıdaki gibi bir birim basamak fonksiyonudur,

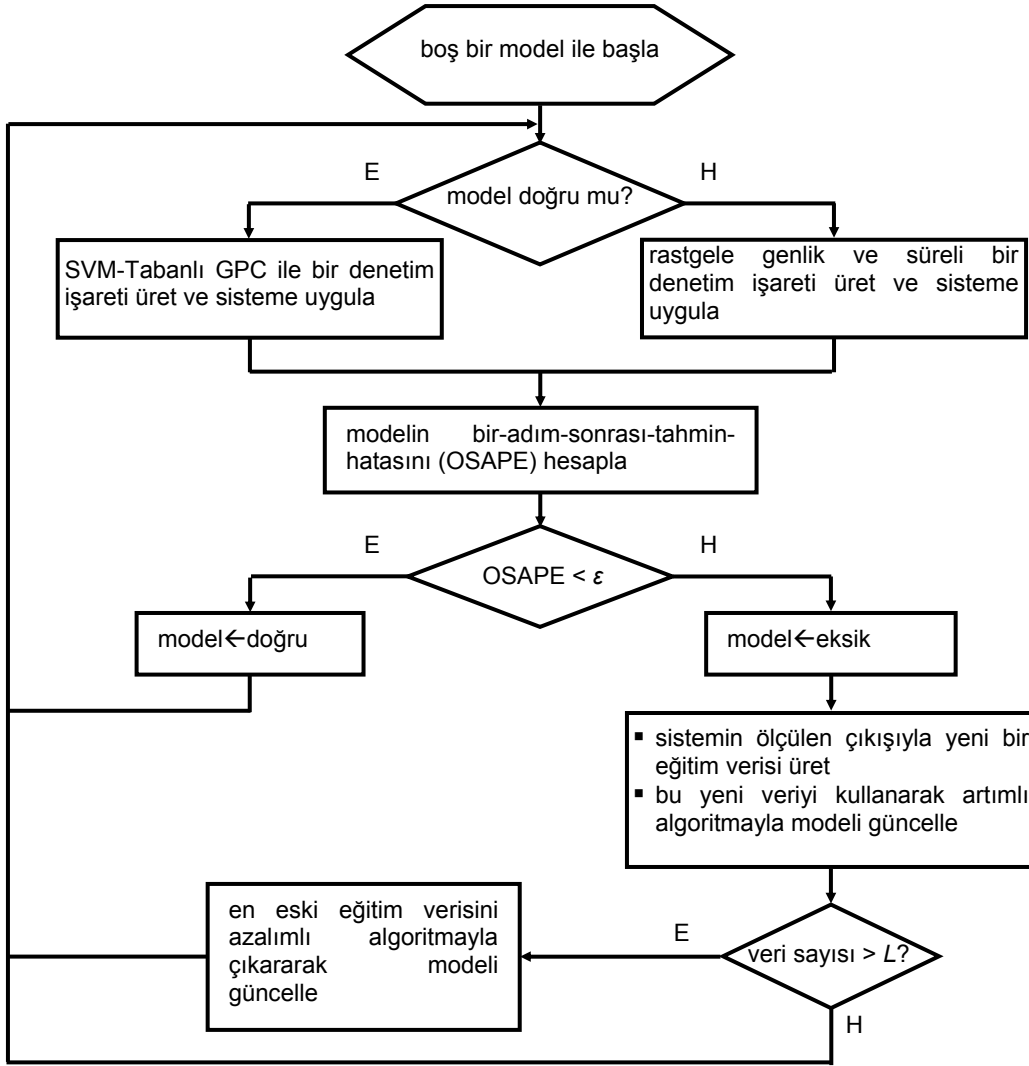
$$\Gamma_x = \begin{cases} 0 & x < 0 \\ 1 & 0 \leq x \end{cases}. \quad (3.2.43)$$

Böylece, arama yönü olan  $\mathbf{p}$ 'nin hesabında gerekli olan ileri yönelik tahminler ve kısmi türevler sistemin SVM modeli kullanılarak elde edilebilir. Ayrıca, not edilmelidir ki RBF çekirdek fonksiyonu için verilen bu formüller diğer tipteki çekirdek fonksiyonlarına da kolaylıkla uyarlanabilir. Arama yönü  $\mathbf{p}$  bulunduktan sonra, adım-uzunluğu  $s$ 'nin bulunması artık bir-boyutlu en iyileme problemine dönüşür ki bu çalışmada bu problem *Altın Oran (Golden-Section)* yöntemiyle çözülmüştür.



Çevrim-içi denetimde boş bir SVM modeliyle başlandığı ve modelin denetim esnasında AOSVR algoritmasıyla gelişeceği daha önceden de belirtilmişti. Ama bu noktada akla şöyle bir soru gelebilir: GPC döngüsünde SVM modelinin kullanıldığı durumlarda denetleyicinin denetim işaretinin nasıl üreteceği yukarıda anlatıldı ancak SVM modeli boş iken veya henüz sistemi tam temsil edemiyorken denetim işareti nasıl üretilecek ve SVM modelinin sistemi tam olarak temsil edebildiğine nasıl karar verilecek? bunun için Şekil 3.2.2’de görüldüğü gibi bir algoritma kullanılmıştır.

Bu algoritmaya biraz daha yakından bakalım. Boş bir SVM modeliyle başlanır. Modelin doğru olup olmadığı, modelin o anda ürettiği OSAPE (one-step ahead prediction error) değeri ile belirlenir. OSAPE değeri  $\varepsilon$ ’dan küçükse model *doğru*, aksi halde *eksik* olarak nitelendirilir. Model doğru olduğu sürece denetim işareti standart SVM-Tabanlı GPC tarafında üretilir. Ancak model eksik ise denetim işareti, genliği  $[u_{\min}, u_{\max}]$  aralığında ve süresi de  $[\tau_{\min}, \tau_{\max}]$  aralığında rastgele değişen bir uyarı işareti olarak sisteme uygulanır. bu uyarı işaretleri sayesinde SVM modelinin gelişmesi için gerekli veri noktaları elde edilir. Yeni bir veri noktası geldiğinde, bu veri SVM modeline AOSVR algoritmasıyla enjekte edilir. OSAPE’nin eşik değeri olarak  $\varepsilon$  seçilmesi anlamlıdır. Çünkü, SVM modelinin her bir veri noktası için  $\varepsilon$ ’dan daha az hata üretmesi istenir. SVM modeli yeterince doğruyken ve denetim işareti SVM modeli üzerinden elde ediliyorken sistemin herhangi bir parametresinde veya referans işaretinde bir değişiklik meydana geldiğinde SVM modeli yine eksik duruma düşebilir ki bu durumda da yine yeni veriler üretilerek SVM modelinin yeni değişimlere uyarlanır olması sağlanır. SVM modeli eksik olduğu sürece her örnekleme periyodunda yeni bir veri noktası gelecek ve AOSVR algoritmasından bu verinin modele enjekte edilmesi istenecektir. Bu durum zamanla SVM modelinin öğrenmesi gereken veri sayısında bir şişmeye yol açabilir. Çünkü, sistemin parametreleri zamanla değişebilir veya referans işareti ve dolayısıyla çalışma noktası sürekli değişebilir. Veri sayısındaki şişme durumlarında, SVM modelinden sadece görel olarak eski verileri unutulması istenebilir. Mesela, önceden çalışılmış ve artık çalışılmayacak bir çalışma noktasına ilişkin verilerin unutulması istenebilir, ya da sistemin parametreleri zamanla değişmiştir ve artık SVM modelinin eski sistem parametreleriyle öğrendiği dinamik davranışı unutulması istenebilir. Bu durumda AOSVR’nin azalimli algoritması devreye girer ve modeldeki diğer verilere dokunmadan istenilen herhangi bir verinin unutulmasını sağlar. Şekil 3.2.2’den de görüleceği gibi, çevrimi-içi SVM-Tabanlı GPC algoritmasında bu işlem belli bir  $L$  sayısına bağlanmıştır. Öğrenilecek veri sayısı bu rakamı geçtiğinde azalimli algoritma ile en eski veri SVM modelinden çıkartılır. Bütün bu işlemler her bir örnekleme periyodunda tekrarlanır. bütün bunlar göz önüne alındığında,  $L$  sayısının belirlenmesinin tamamen denetim problemine bağlı olduğu görülmektedir.



Şekil 3.2.2. Çevrim-içi SVM-Tabanlı GPC algoritması

Sonuç olarak, çevrimi-içi SVM-Tabanlı GPC prosedürü aşağıdaki gibi maddeler halinde açıklanabilir:

**Adım1.** Denetim esnasında sistemin dinamiklerini modellemede kullanılacak verileri toplamak için denetim işareti süreleri, yani  $\tau_{\min}$  ve  $\tau_{\max}$ , ve NARX parametreleri, yani  $n_1$  ve  $m_1$ , belirlenir. Ayrıca, normalizasyon için gerekli olan, giriş işaretinin alt ve üst sınırları, yani  $u_{\min}$  ve  $u_{\max}$  değerleri, çıkış işaretinin alt ve üst sınırları, yani  $y_{\min}$  ve  $y_{\max}$  değerleri belirlenir.

**Adım2.** Her iterasyonda, ya da örnekleme periyodu süresince, Şekil 3.2.2.'de görülen algoritma uygulanır. Bu algoritmada, artımlı ve azalimli algoritmalar AOSVR algoritmalarıdır ve OSAPE değeri de SVM modelinin bir adım sonrası için ürettiği

tahmin hatasıdır. Yine bu algorithmada, modelin doğru olduğu durumlarda, SVM-Tabanlı GPC şu şekilde uygulanır:

Sistemin çıkışından ölçümleri al, aday denetim vektörü  $\mathbf{u}_{\text{candidate}}$ 'yu oluştur ve bu vektöre karşı düşen ileri yönelik tahminleri SVM modelini kullanarak elde et. Yine SVM modelini kullanarak, formüller yardımıyla gradyant vektörü  $\mathbf{g}$ 'yi bul ve arama yönü olarak  $\mathbf{p} = -\mathbf{H}^{-1}\mathbf{g}$  al. Ardından, en iyi adım-uzunluğunu bul ve  $\mathbf{u}_{\text{optimal}} = \mathbf{u}_{\text{candidate}} + s\mathbf{p}$  kuralı ile güncellemeyi yap. Son olarak,  $\mathbf{u}_{\text{optimal}}$  vektöründe  $(n+1)$  zaman indeksine karşı düşen ilk elemanı sisteme uygula.

### 3.2.2 Deney

#### 3.2.2.1 Modelleme Verileri

Denetim yöntemi çevrim-içi çalıştığı için önceden herhangi bir modelleme verisi toplanmamaktadır. Onun yerine boş bir SVM modeliyle başlanmakta ve SVM modeli denetim esnasında geliştirilmektedir.

#### 3.2.2.2 SVM Modelinin Elde Edilmesi

Denetime boş bir SVM modeliyle başlanmakta ve SVM modeli denetim esnasında Şekil 3.2.2'deki algoritma ile geliştirilmektedir.

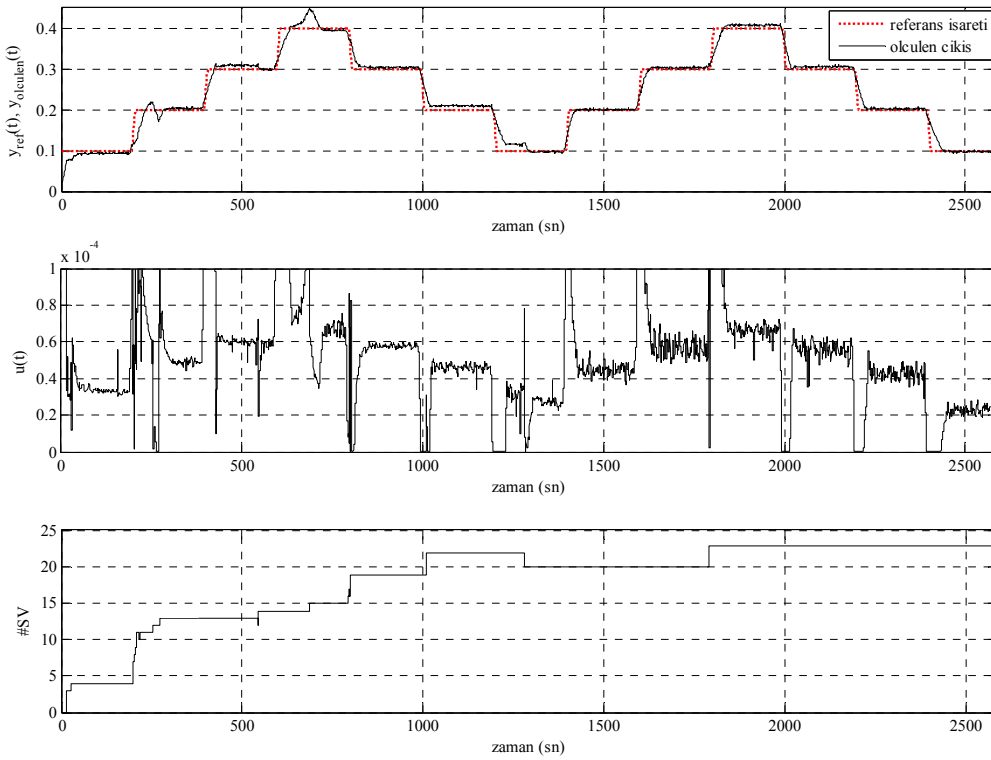
#### 3.2.2.3 Deneysel Sonuçlar

Çevrimiçi-SVM-Tabanlı GPC yöntemi ile deneysel üç-tanklı sistemin denetimi aşamasında aşağıdaki parametre değerleriyle çalışılmıştır:

örnekleme zamanı (sn)	1.000
$u_{\min}$ ( $\text{m}^3/\text{sn}$ )	0.0000
$u_{\max}$ ( $\text{m}^3/\text{sn}$ )	0.0001
$n_1$	5
$m_1$	5
$N_u$	2
$N_y$	20
$\lambda$	2.00
$\sigma$	10.0
$\varepsilon$	0.010

Her ne kadar üç-tanklı sıvı sisteminin üçüncü dereceden bir sistem olması nedeniyle giriş ve çıkış gecikmelerinin  $n_1 = m_1 = 3$  olarak alınmasının yeterli olacağı düşünülse de, daha güvenilir bir model elde etmek adına bu gecikme sayılarının biraz daha büyük seçilmesinde hiçbir sakınca yoktur.

DTS200 sistemindeki musluk katsayıları yaklaşık olarak  $az_{13}=0.42$ ;  $az_{32}=1.0$ ;  $az_{10}=0.0$ ;  $az_{20}=1.0$  ve  $az_{30}=0.0$  şeklinde ayarlanarak ve sadece 1. tanka sıvı pompalanarak bu tankın sıvı seviyesi kontrol edilmeye çalışılmış ve böylece sistem tek-girişli-tek-çıkışlı bir sisteme dönüştürülmüştür.

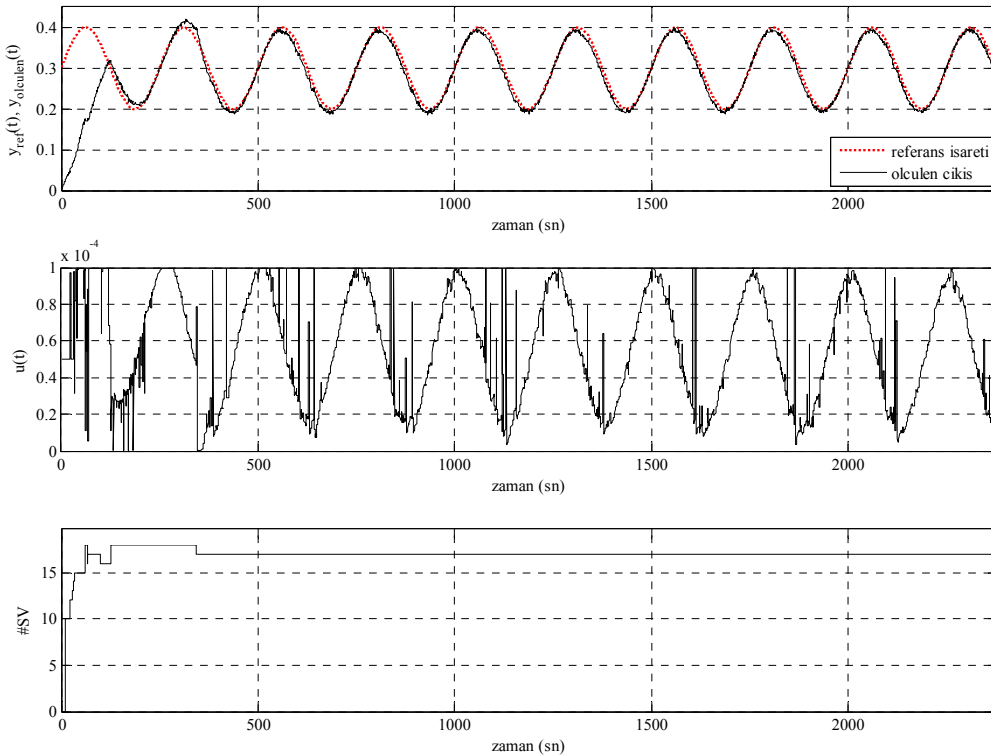


Şekil 3.2.3. Basamak referans girişi için deney sonuçları.

Basamak şeklindeki referans girişi için elde edilen sonuçlar Şekil 3.2.3'te görülmektedir. En üstteki şekilde referans işareti ve deneyde ölçülen sıvı seviyesi görülmektedir. Ortamdaki gürültüden dolayı ölçülen değerler gerçek değerden bir miktar sapmaktadır. Bu yüzden de, ölçülen sıvı seviyesi istenilen referans değerden farklı gibi gözükse de gerçekte böyle bir sapma gözlenmemiştir. Ortadaki şekilde ise uygulanan denetim işareti görülmektedir. Denetleyici ölçüm hatalarını ve modelleme hatalarını telafi edebilmek için zaman zaman aşırı

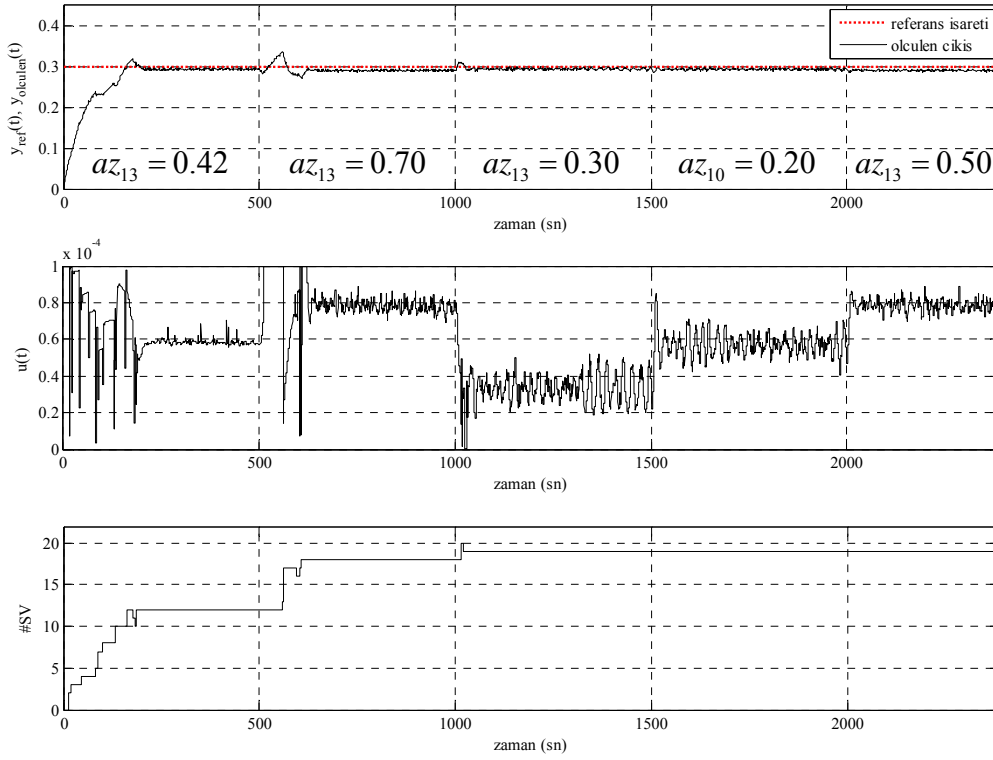
dalgalanmalar sergilemektedir. En alttaki şekilde ise destek vektör sayısı görülmektedir. Şekilden de görüleceği gibi, boş bir SVM modeliyle işe başlanmaktadır. Bu yüzden de ilk zamanlardaki denetim başarımı oldukça düşüktür. Ancak zaman ilerledikçe SVM modeli gelişmektedir, yani SVM modelindeki destek vektör sayısı artmaktadır, böylece çevrim-içi SVM tabanlı GPC denetleyicisi daha iyi bir başarımla sergilemektedir. Belli bir zaman sonra destek vektör sayısı değişmemektedir. Bu da, SVM modelinin artık yeterince iyi temsil ettiğini göstermektedir. Zaten bunun böyle olduğu denetim başarımından da gözlenmektedir. Örneğin başlangıçta 0.20 m referans değeri için çok iyi olmayan bir denetim başarımı varken, en sonda yine 0.20 m referans değeri için olan denetim başarımı çok yüksektir, çünkü artık SVM modeli gelişmiştir ve sistemi çok daha iyi temsil edebilmektedir.

Benzer gözlemleri sinüs şeklindeki referans işareti için de söyleyebiliriz. Şekil 3.2.4'te sinüzoidal referans işareti için elde edilen deney sonuçları görülmektedir. Yine, başlangıçta çok iyi olmayan denetim başarımı, SVM modelinin gelişmesiyle birlikte daha da iyileşmiş ve en son periyotta en yüksek denetim başarımı elde edilmiştir.



Şekil 3.2.4. Sinüs referans girişi için deney sonuçları.

Şekil 3.2.5'te ise referans işareti sabit iken sistem parametrelerinden birinin değiştirilmesi sonucunda denetleyicinin bu değişime kendini nasıl uyarladığı görülmektedir. Bu amaçla, 1. ve 3. tankı birbirine bağlayan  $az_{13}$  musluğunun değeri ilk 500 saniyede nominal değeri olan 0.42'de tutularak denetleyicinin belli bir aşamaya gelmesi sağlanmıştır. Nitekim bunu en alttaki destek vektörü sayısından (#SV) görmekteyiz. SVM modelindeki destek vektörü sayısı yaklaşık 200 saniye boyunca sürekli değişmiş, SVM modeli sistemin bu çalışma noktasındaki dinamiklerini yakalayınca artı model daha fazla değişmemiştir. Daha sonra 500. saniyede  $az_{13}$  musluğu biraz açılarak yaklaşık 0.70 değerine getirilmiştir. Bu nedenle de 1. tankın sıvı seviyesi kısa bir süre azalma eğilimi göstermiştir ancak denetleyici buna karşı bir tepki vererek sıvı seviyesini referans değerde tutmaya çalışmış ancak sistemin parametresi değiştiğinden bunu ilk aşamada başaramamıştır. Daha sonra yeni gelen verilerle SVM modeli değişime kendini uyarlamış (bunu 600 sn civarındaki #SV'nin artışından anlıyoruz) ve bu andan itibaren sıvı seviyesini tekrar referans değerde tutmayı başarmıştır. Benzer şekilde, 1000. saniyede bu kez musluk yaklaşık 0.30 değerine doğru kapatılmış, yani iki tank arasındaki sıvı geçiş katsayısı azaltılmıştır. bunun sonucunda 1. tanktaki sıvı seviyesi hafifçe artma eğilimi göstermiş, denetleyici bunu karşılamak için bir pompanın debisini düşürmüştür ancak sistemin parametresi değiştiğinden dolayı sıvı seviyesini referans değerde tutması bir miktar gecikmiştir. Bu esnada SVM modeli parametre değişikliğine kendini uyarlamıştır (bunu 1050 sn civarındaki #SV'nin artışından anlıyoruz). Dikkat edilirse 500. saniyedeki parametre değişikliği için SVM modelindeki uyarlama süresi ve miktarı, 1000. saniyedeki üstelik daha büyük değişiklikte daha az görülmektedir. Bunun sebebi artık SVM modelinin parametre değişikliklerine daha gürbüz hale gelmesidir. Nitekim bunu 1500. ve 2000. saniyelerdeki parametre değişimlerinde daha iyi gözlemliyoruz. 1500. saniyede bu defa  $az_{10}$  tamamen kapalı durumdan çıkarılarak yaklaşık 0.20 değerine getirilmiştir. Böylece, 1. tanktan rezervuara sızıntı olması sağlanmıştır. Ama şekilden de görüldüğü gibi SVM modelinde herhangi bir değişiklik olmamasına rağmen denetim başarılı bir şekilde devam etmiştir. Denetim işaretinin genliğine bakıldığında, bu parametre değişimini karşılamak için pompadan geçen sıvı debisinin bir miktar arttığı görülebilir. Benzer şekilde 2000. saniyede  $az_{13}$  musluğu bu defa yaklaşık 0.50 değerine getirilmiş olmasına rağmen SVM modelinde herhangi bir uyarlama olmamış ama denetim başarılı bir şekilde devam etmiştir. Bu da gösteriyor ki, denetleyici belli bir aşamadan sonra küçük parametre değişimlerine gürbüz olabilmektedir.



Şekil 3.2.5. Sabit referans girişi ve değişken musluk parametreleri için deney sonuçları

Bu deneyde, dinamik denklemleri bilinmeyen ve doğrusal olmayan bir sistem, önceden herhangi bir şekilde veri toplanmadan doğrudan boş bir SVM modeliyle denetlenmeye başlanmış, model boş olduğundan doğal olarak ilk aşamalardaki denetim başarımı düşük olmuştur. Ancak zaman ilerledikçe SVM modeli uygun bir şekilde gelişerek sistemi daha iyi temsil etmeye başlamış ve her geçen örnekleme anında başarım daha da iyileşmiştir. Böylece, çevrim-içi SVM tabanlı GPC yöntemiyle aynı anda hem modelleme hem de denetleme başarıyla gerçekleştirilmiştir.

#### 3.2.2.4 Denetim Parametrelerinin Etkileri

Çevrim-içi SVM tabanlı GPC denetim yönteminin denetim parametrelerinin etkileri incelenirken, bu projede kullanılan çevrim-dışı yöntemlerden farklı olarak modelleme parametreleri de ele alınacaktır. Çevrim-dışı yöntemlerde modelleme parametreleri ( $\sigma, \varepsilon$ ), daha doğru bir modelin daha iyi bir denetim başarımı sağlayacağı düşüncesiyle sadece modelleme başarımı açısından incelenmiş, denetim parametrelerinin ( $N_y, \lambda$ ) etkileri ise model elde edildikten sonra incelenmiştir. Parametrelerin çevrim-içi denetim başarımı

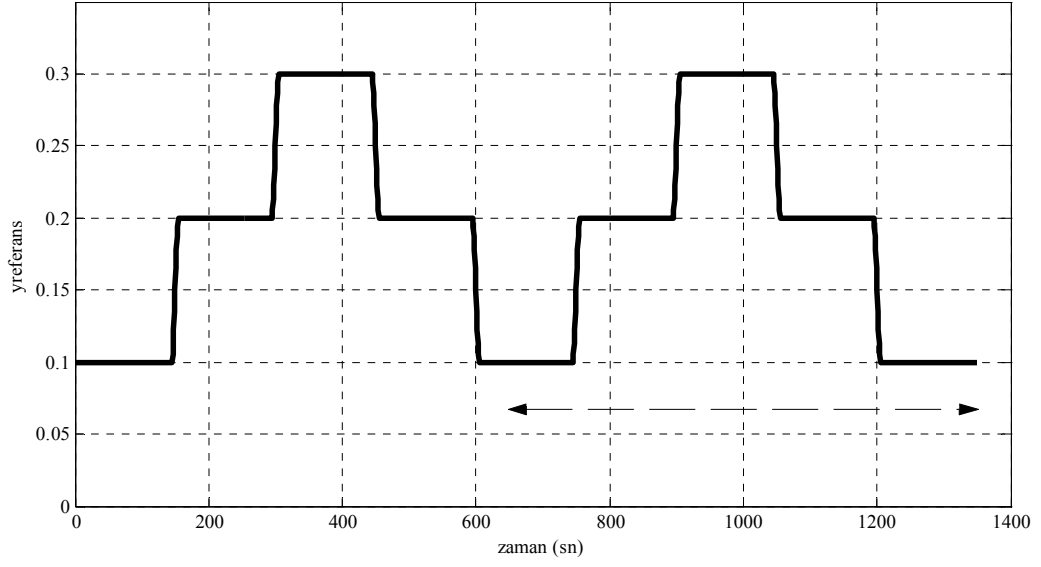
üzerindeki etkilerini incelerken kolaylık olsun diye önceki tecrübelerimizi kullanarak tersinden gideceğiz. Yani, denetim parametreleri  $(N_y, \lambda)$ , en iyi değerlerinde sabit tutularak sadece modelleme parametrelerinin  $(\sigma, \varepsilon)$  etkileri ele alınacaktır. Böylece, denetim parametreleri  $N_y=15$  ve  $\lambda=4.5$  olarak sabit tutulmuştur. Modelleme parametrelerinin etkileri incelenirken de AOSVR algoritmasının bazı hassasiyetleri göz önünde bulundurulmuştur: Bazı durumlarda AOSVR algoritması yığılma yapabilmektedir. Yığılma esnasında algoritma yeni gelen veriyi KKT koşullarını sağlayacak şekilde SVM modeline enjekte edememekte ve bunun sonucunda algoritma kilitlenmektedir. Yığılma, çoğu zaman çok küçük  $\sigma$  ve  $\varepsilon$  değerlerinde gürültü içeren bir veri geldiğinde ortaya çıkmaktadır. Bu nedenle parametrelerin etkileri incelenirken çok küçük  $\sigma$  ve  $\varepsilon$  değerlerinden kaçınılacaktır. Parametrelerin etkileri iki farklı durum için incelenmiştir: 1) Sistem parametrelerinin sabit ancak referans işaretinin basamak şeklinde değiştiği durum, 2) Referans işareti sabit ancak sistem bir sistem parametresinin belli aralıklarla değiştiği durum.

Durum 1) Sistem parametrelerinin sabit ancak referans işaretinin basamak şeklinde değiştiği durum için parametre etkilerinin incelemesini yaparken farklı  $\sigma$  ve  $\varepsilon$  değerleri için aşağıdaki başarımların göstergesi kullanılmıştır:

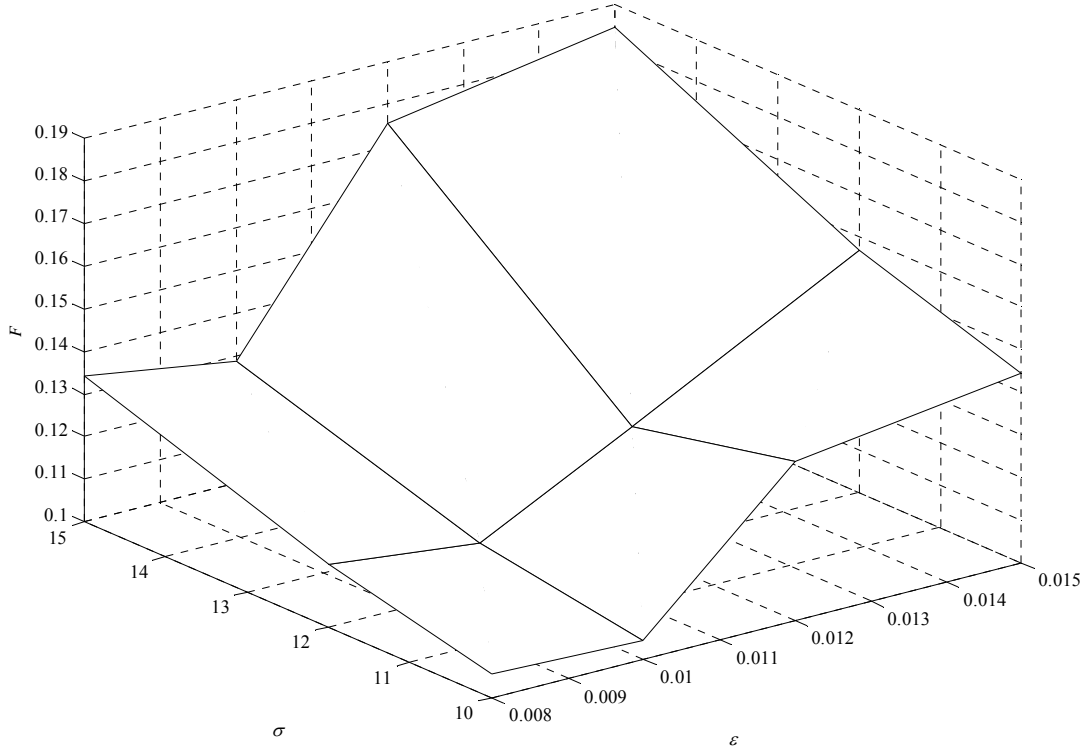
$$F = \sum_{n=635}^{1335} (\tilde{y}(n) - \hat{y}(n))^2 + \sum_{n=635}^{1335} \lambda (\Delta u(n))^2, \quad (3.2.44)$$

burada  $\tilde{y}$ , Şekil 3.2.6'da görüldüğü gibi daha basit ve kısa bir basamak referans işaretidir. Başarım göstergesindeki zaman indeksi, Şekil 3.2.6'da da görüldüğü gibi, 635. sn ile 1335. sn arasındadır. Bunun sebebi, referans işaretinin ilk çevriminin sonuna kadar (yani 635. sn'ye kadar) denetleyici sistemin dinamiklerini öğrenmektedir. Bu aşamada, rastgele sisteme giriş işaretleri de verilebilmektedir. Ama ikinci çevrimde artık denetleyicinin öğrenmesi gereken bir durum kalmamıştır ve esas başarımın bu çevrimde olduğu varsayımıyla Denklem (3.2.44)'teki gibi bir başarımların göstergesi tanımlanmıştır. Şekil 3.2.7'de sistem parametreleri sabit iken modelleme parametrelerinin denetim başarımı üzerindeki etkileri görülmektedir. Buna göre,  $\sigma$  ve  $\varepsilon$  parametreleri azaldıkça başarımın iyileştiği gözlenmiştir. Ancak, yukarıda da belirtildiği gibi, bu parametrelerin daha küçük değerleri AOSVR algoritmasında yığılmaya sebep olabilmektedir.





Şekil 3.2.6. Sabit sistem parametreleri durumu için basamak referans işareti.



Şekil 3.2.7. Sistem parametreleri sabit iken modelleme parametrelerinin denetim başarıımı üzerindeki etkileri.

Sistemin parametreleri sabitken referans işaretin değişmesi durumundaki denetleyici başarımı daha çok denetleyicinin referans işaretteki değişime ne kadar gürbüz olması ile ilgilidir. Bu gürbüzlük, modelleme açısından bakıldığında, sistemin dinamiklerinin daha geniş bir çalışma bölgesinde temsil edilmesiyle mümkün olabilir.  $\sigma$  parametresi, SVM modelinin genelleme yeteneğini belirler. Sigma arttıkça genelleme yeteneği de artar. Dolayısıyla bu problemde daha büyük  $\sigma$ 'ların daha iyi bir başarımlı sağlaması beklenebilir. diğer taraftan  $\varepsilon$  değeri de modelin doğruluğu ile karmaşıklık arasında bir ayar sağlar. Küçük seçildikçe doğruluk ve karmaşıklık artar. Tabi, karmaşıkların artması AOSVR algoritmasını olumsuz etkiler, algoritmanın tıkanmasına yol açabilir. O yüzden  $\varepsilon$  değeri çok küçük olmamalıdır.

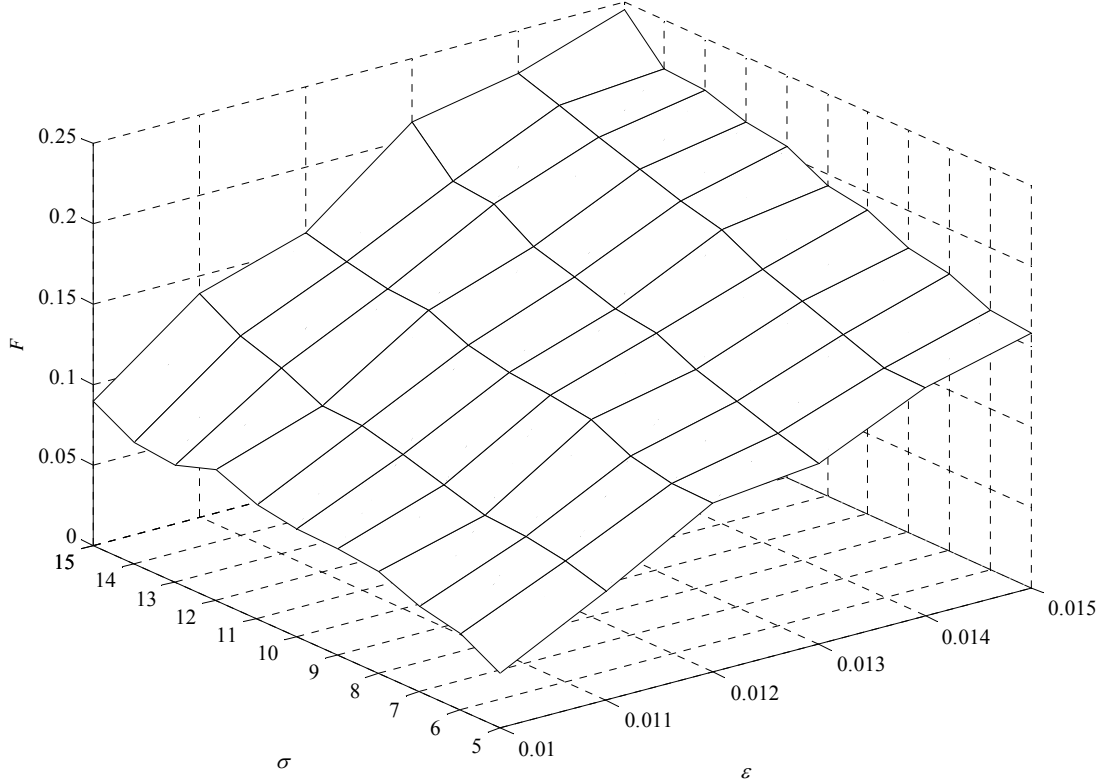
Deneyleerde, çok küçük  $\sigma$  ve  $\varepsilon$  değerlerin de destek vektör sayısının çok arttığı gözlemlendi. O yüzden daha büyük değerlerle çalışıldı.

Durum II) Referans işaret  $\tilde{y}=0.2$  m'de 600 saniye boyunca sabit fakat sistemin  $a_{z_{13}}$  parametresinin 300. saniyede  $a_{z_{13}}=1.00$  yapıldığı durum için parametre etkilerinin incelemesini yaparken farklı  $\sigma$  ve  $\varepsilon$  değerleri için aşağıdaki başarımlı göstergesi kullanılmıştır:

$$F = \sum_{n=100}^{600} \sum_{j=1}^{N_y} (\tilde{y}(n+j) - \hat{y}(n+j))^2 + \sum_{n=100}^{600} \sum_{j=1}^{N_u} \lambda (\Delta u(n+j))^2 ,$$

burada  $\tilde{y}=0.2$  m'de sabit referans işaretidir. Deneye, sistemin nominal parametreleri ile başlanmış, 300. saniyede  $a_{z_{13}}$  parametresi nominal değeri olan 0.42'den bir miktar artırılarak 1.00 değerine getirilmiştir, yani musluk tam açılmıştır. Başarımlı ölçütündeki başlangıç zamanının 100 saniye alınmasının sebebi, ortalama ilk 100 saniye içinde denetleyicinin sistemin dinamiklerini öğrenme aşamasında olması ve bu aşamada sisteme rastgele giriş işaretlerinin de veriliyor olmasıdır. Parametre değişimine olan uyarlanma aslında 300. saniyeden sonra ortaya çıkmaktadır. Ancak başlangıç zamanını 300. saniye olarak aldığımızda da şöyle bir sorunla karşılaşıldı:  $\sigma$  ve  $\varepsilon$  değerlerinin büyük olduğu durumlarda sistemin çıkışı referansı yeterince iyi takip edememektedir (belli bir ofset oluşmakta) ve 300. saniyede sistem parametresi değiştiğinde denetleyicinin buna uyarlanamamasına rağmen bu parametre değişimi sonucu ofset ortadan kalkmaktadır (tabi bunun tersi de olabilirdi). Bu durumda da başarımlı göstergesi oldukça iyi çıkmakta ve yanıltıcı olmaktadır. O nedenle başlangıç zamanı 100. saniye olarak

seçilmiştir. Şekil 3.2.8'de farklı  $\sigma$  ve  $\varepsilon$  değerleri için denetleyicinin başarımları görülmektedir.



Şekil 3.2.8. Referans işareti sabit ancak bir sistem parametresi değişiyor iken modelleme parametrelerinin denetim başarımları üzerindeki etkileri.

Deneylerde,  $\sigma$  değeri arttıkça denetleyicinin sistemdeki ani parametre değişimlerine olan uyarlanması daha da yavaş olduğu hatta büyük  $\sigma$  değerleri için uyarlanma olmadığı gözlemlendi. Bunun nedeni  $\sigma$  değerinin artmasıyla SVM modelinin genelleme yeteneğinin artması ve sistemdeki parametre değişimlerine daha duyarsız hale gelmesidir. Deneylerde, büyük  $\sigma$  değerleri için, ani parametre değişiminin olduğu anlarda SVM modelinin değişmediği gözlemlendi. Bunun sonucunda sistemin çıkışı ile referans işaret arasında bir sürekli-hal hatası oluştu.  $\sigma$  değeri azaldıkça, SVM modelinin genelleme yeteneği azalır ancak sistemdeki parametre değişimlerine daha hızlı tepki verir. Diğer taraftan,  $\sigma$  değerinin çok küçülmesi SVM modelindeki destek vektör sayısının artmasına neden olduğu gibi AOSVR algoritmasının yeni verileri modele enjekte etmesinde zorluklar yaşamasına yol açabilir. Küçük  $\sigma$  değerleri daha lokalize SVM modeli verirken, büyük  $\sigma$  değerleri daha global SVM modeli vermektedir. Referans işaretinin sabit olduğu durumlarda, sistemin çalışma bölgesi görece olarak daha küçüktür, dolayısıyla SVM modeli daha lokalize bir

bölgede çalışacağından daha küçük  $\sigma$  değerlerinin daha iyi denetim başarımı vermesi beklenen bir sonuçtur. Diğer taraftan, referans işaretinin basamak şeklinde olduğu durumda, çalışma bölgesi daha büyüktür ve ayrıca denetleyici referans işaretindeki değişimleri tolare etmek durumundadır. Bu durumda daha büyük  $\sigma$  değerlerinin daha iyi denetim başarımı vermesi beklenir.  $\sigma$  değeri bunlar göz önüne alınarak seçilmelidir.

Diğer taraftan,  $\varepsilon$  değeri ise bir veri noktası için modelin yapmasına izin verilen en büyük hata değeri olduğu için modelin doğruluğunu, genelleme kapasitesini ve modeldeki destek vektör sayısını çok etkiler.  $\varepsilon$ 'un büyük seçilmesi durumunda destek modeldeki destek vektör sayısı düşüktür ancak SVM modelinin sistemi belli bir çalışma noktasında iyi temsil edememesi sonucunda bir sürekli-hal hatası oluşur. Ayrıca,  $\varepsilon$  büyüdükçe sistemdeki ani parametre değişimlerine SVM modelinin uyarlanması daha da güçleşir. ve denetleyici bu değişimlere daha duyarsız hale gelir.  $\varepsilon$  küçüldükçe sistemdeki ani parametre değişimlerine SVM modelinin uyarlanması daha çabuk olur ancak SVM modelindeki destek vektör sayısı hızla artar.

### 3.3 Çevrim-dışı MIMO Denetim

#### 3.3.1 Yöntem



Şekil 3.3.1. MIMO sistem.

Şekil 3.3.1.'de görüldüğü gibi doğrusal olmayan sürekli zamanlı  $R$  adet girişi ve  $Q$  adet çıkışı olan bir sistemi ele alalım. Sistemin dinamiklerinin aşağıdaki gibi bir **Nonlinear AutoRegressive with eXogenous inputs (NARX)** modeliyle temsil edilebildiğini varsayıyoruz.

$$y_q(n) = f_q(u_1(n), \dots, u_1(n - n_1), \dots, u_R(n), \dots, u_R(n - n_R), y_1(n - 1), \dots, y_1(n - m_1), \dots, y_Q(n - 1), \dots, y_Q(n - m_Q))', \quad (3.3.1)$$

burada  $q = 1, \dots, Q$  ve  $u_r(n)$   $n^{\text{inci}}$  zaman indeksinde sisteme uygulanan  $r^{\text{inci}}$  denetim işareti,  $y_q(n)$  ise sistemin  $q^{\text{uncu}}$  çıkışıdır,  $n_i$ 'ler ve  $m_i$ 'ler de modelde yer alan geçmiş giriş ve çıkış gecikmelerini göstermektedir. Ayrıca, bu modeldeki  $f_q$  fonksiyonlarının da bilinmedikleri varsayılmıştır. Yani sistemin matematiksel modeli elimizde mevcut değildir. Bunun yerine elimizde aşağıdaki gibi bir eğitim veri seti mevcuttur.

$$\begin{aligned} T_q &= \{u_1(k), \dots, u_1(k - n_1), \dots, u_R(k), \dots, u_R(k - n_R), \\ & y_1(k - 1), \dots, y_1(k - m_1), \dots, y_Q(k - 1), \dots, y_Q(k - m_Q); y_q(k)\}_{k=1}^{k=N} \\ &= \{\mathbf{x}_k; y_q(k)\}_{k=1}^{k=N} \end{aligned} \quad (3.3.2)$$

burada  $\mathbf{x}_k \in X \subseteq \mathfrak{R}^{\sum n_i + \sum m_i + R}$  büyüklüğü  $k^{\text{inci}}$  giriş veri noktasını ve  $y_q(k) \in Y \subseteq \mathfrak{R}$  da buna karşı düşen çıkış değerini göstermektedir. Bu verileri kullanarak sistemin giriş-çıkış ilişkisini temsil edebilecek bir model oluşturmak istiyoruz. Bu amaçla, veri seti  $T_q$ 'yu kullanarak sistemin modelini elde edecek SVM algoritması şu şekildedir. SVM bağlanım modelinin birincil formu (3.3.3) denkleminde görüldüğü gibidir; bu model öznelik uzayı olan  $F$  uzayında doğrusaldır.

$$\hat{y}_q(i) = \langle \mathbf{w}_q, \Phi(\mathbf{x}_i) \rangle + b_q \quad (3.3.3)$$

burada  $\mathbf{w}_q$  vektörü  $F$  öznitelik uzayında bir vektördür,  $\Phi(\cdot)$  fonksiyonu giriş uzayından öznitelik uzayına bir dönüşümdür,  $b_q$  büyüklüğü bir yanlılık (bias) terimidir ve  $\langle \cdot \rangle$  operatörü de  $F$  uzayında bir iç-çarpımı göstermektedir. SVM bağlanım algoritmaları genel olarak bir bağlanım problemini ikincil uzayda bir en iyileme problemi olarak görürler. SVM modelinin ikincil uzaydaki formu şu şekildedir:

$$\hat{y}_q(i) = \sum_{j=1}^N \alpha_{qj} K(\mathbf{x}_i, \mathbf{x}_j) + b_q, \quad (3.3.4)$$

burada  $\alpha_{qj}$ 'lar her bir eğitim verisinin katsayılarıdır ve  $K(\mathbf{x}_i, \mathbf{x}_j)$  da şu şekilde verilen bir çekirdek fonksiyonudur:  $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = K_{ij}$ . Çekirdek fonksiyonu, öznitelik uzayında iç-çarpım işlemini gerçekleştirdiğinden dolayı  $\Phi(\mathbf{x})$  fonksiyonunun açık formunun bilinmesine gerek kalmaz. Denklem (3.3.4) ile verilen modelde, sıfırdan farklı bir  $\alpha_{qj}$  değerine karşı düşen  $\mathbf{x}_j$  gibi bir eğitim noktasına *destek vektörü* denmektedir. Bu çalışmada, aşağıdaki gibi verilen Vapnik's  $\varepsilon$ -insensitive kayıp fonksiyonunu ( $L(\varepsilon, y, \hat{y})$ ) kullanan  $\varepsilon$ -SVR algoritması benimsenmiştir:

$$L(\varepsilon, y, \hat{y}) = \begin{cases} 0 & y - \hat{y} \leq \varepsilon \\ y - \hat{y} & y - \hat{y} > \varepsilon \end{cases} \quad (3.3.5)$$

$\varepsilon$ -SVR algoritması, bağlanım probleminin birincil formunu şu şekilde formülize eder:

$$\min_{\mathbf{w}_q, b_q, \xi, \xi^*} P_{\varepsilon_q} = \frac{1}{2} \|\mathbf{w}_q\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (3.3.6)$$

kısıtlar:

$$\begin{aligned} y_q(i) - \langle \mathbf{w}_q, \Phi(\mathbf{x}_i) \rangle - b_q &\leq \varepsilon_q + \xi_i, \quad i = 1, \dots, N \\ \langle \mathbf{w}_q, \Phi(\mathbf{x}_i) \rangle + b_q - y_q(i) &\leq \varepsilon_q + \xi_i^*, \quad i = 1, \dots, N \\ \xi_i, \xi_i^* &\geq 0, \quad i = 1, \dots, N \end{aligned} \quad (3.3.7)$$

burada  $\varepsilon_q$  değeri  $q^{\text{uncu}}$  çıkış için tolare edilebilen en büyük hata değeridir,  $\xi_i$ 'ler ve  $\xi_i^*$ 'ler gevşek değişkenlerdir,  $\|\cdot\|$  operatörü Öklit normudur ve  $C$  de modelin karmaşıklığı ile  $\varepsilon_q$ 'den daha büyük hatalara gösterilen tolerans derecesi arasında denge kurmayı sağlayan bir parametredir. Böylece, en iyileme probleminin ikincil formu aşağıdaki gibi bir karesel programlama (Quadratic Programming - QP) problemine dönüşür:

$$\min_{\beta_{qj}, \beta_{qj}^*} D_{\varepsilon_q} = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K_{ij} (\beta_{qi} - \beta_{qi}^*) (\beta_{qj} - \beta_{qj}^*) + \varepsilon_q \sum_{i=1}^N (\beta_{qi} + \beta_{qi}^*) - \sum_{i=1}^N y_q(i) (\beta_{qi} - \beta_{qi}^*) \quad (3.3.8)$$

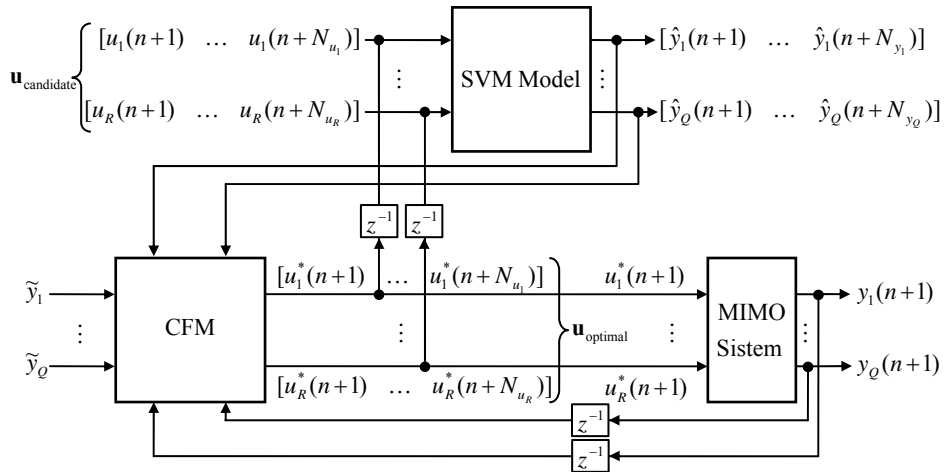
kısıtlar:

$$0 \leq \beta_{qi}, \beta_{qi}^* \leq C, \sum_{i=1}^N (\beta_{qi} - \beta_{qi}^*) = 0, i = 1, \dots, N \quad (3.3.9)$$

Denklem (3.3.8 ve 3.3.9) ile verilen QP probleminin çözümü  $\beta_{qi}$ 'ler ve  $\beta_{qi}^*$ 'lerin optimum değerlerini verir. Modeldeki  $b_q$ 'lar,  $0 \leq \beta_{qi} - \beta_{qi}^* \leq C$  koşulunun sağlandığı her bir  $\mathbf{x}_i$  destek vektörü için  $y_q(i) - \hat{y}_q(i) = \varepsilon_q$  koşulu sağlanacak şekilde bulunurlar. Eğer  $j = 1, \dots, N$  olmak üzere  $\mathbf{x}_j$  destek vektörü için  $\alpha_{qj}$  gibi yeni bir katsayı  $\alpha_{qj} = \beta_{qi} - \beta_{qj}^*$  şeklinde tanımlanırsa, o zaman Denklem (3.3.4)'teki gibi bir SVM modeli elde edilir. Hatta, sadece destek vektörleri dikkate alındığında model şu hale gelir:

$$\hat{y}_q(i) = \sum_{\substack{j=1 \\ j \in SV}}^{\#SV} \alpha_{qj} K(\mathbf{x}_i, \mathbf{x}_j) + b_q \quad (3.3.10)$$

burada #SV modeldeki destek vektör sayısıdır (SMOLA ve SCHÖLKOPF, 1998; CRISTIANINI ve TAYLOR, 2000; SCHÖLKOPF ve ark., 1999). Denklem (3.3.10) ile verilen SVM modeli, tüm eğitim verisinin sadece destek vektörleriyle temsil edilmesi bakımından seyrek bir modeldir.  $\varepsilon$ -SVR algoritmasının parametreleri, çıkıştaki hata toleransı  $\varepsilon_q$ , ayar parametresi C, eğitim verilerinin sayısı N ve modelde kullanılan çekirdek fonksiyonunun parametreleridir.  $\varepsilon$ -SVR algoritmasının en büyük avantajı uygun  $\varepsilon_q$  değerleri seçilerek toplam eğitim hatasının en büyük değerinin önceden bilinebilmesidir. Diğer taraftan, bu algorithmada elde edilen modeldeki destek vektör sayısı kontrol edilemez ve bazı durumlarda seyreklik ortadan kalkabilir.



Şekil 3.3.2. MIMO SVM-Tabanlı GPC şeması.

Denetimi yapılacak sistemin SVM modeli elde edildikten sonra artık bu model MIMO SVM-Tabanlı GPC içerisinde kullanılabilir. Şekil 3.3.2'de, projede geliştirilen MIMO SVM-Tabanlı GPC yapısı görülmektedir. Burada,  $\tilde{y}_q$ 'ler referans işaretler ve  $\hat{y}_q$ 'lar da tahminlerdir. Bu yöntemde, SVM modeli MIMO sisteminin  $\mathbf{u}_{\text{candidate}}$  aday denetim vektörüne karşılık gelecekteki yörüngesini tahmin eder ve aynı zamanda bedel fonksiyonunu en aza indirme (Cost Function Minimization - CFM) bloğunda gerekli türev bilgisini sağlar. CFM bloğu referans işaretleri, ölçülen çıkışları ve SVM modelinin tahminlerini alır ve en iyilenmiş bir denetim vektörü oluşturur. Bu en iyilenmiş vektörün ilk elemanları sisteme denetim girdileri olarak uygulanır. Bu algoritmayı biraz daha açmak gerekirse, örnekleme periyotlarının her birinde ilk olarak, aşağıdaki gibi bir aday denetim vektörü oluşturulur:

$$\mathbf{u}_{\text{candidate}} = [u_1(n+1) \quad \dots \quad u_1(n+N_{u_1}) \quad \dots \quad u_R(n+1) \quad \dots \quad u_R(n+N_{u_R})] \quad (3.3.11)$$

burada  $N_{u_r}$ 'ler her bir denetim girdisine ilişkin denetim ufuklarıdır. Bu denetim vektörü sisteme doğrudan uygulanmaz çünkü henüz en iyilenmiş değildir. Bu en iyileme, SVM modelinin aday denetim vektörüne karşı ürettiği tahminlerle yapılmaktadır. Bura en iyilemede amaç,  $\mathbf{u}_{\text{candidate}}$  vektörünü uygun bir şekilde değiştirerek tahmin hatalarını en aza indirirken denetim işaretlerinde de çok fazla değişime yol açmamaktır. Bunu aşağıdaki amaç fonksiyonu ile göstermek daha kolay olacaktır:

$$F = \sum_{q=1}^Q \sum_{k=1}^{N_{y_q}} (\tilde{y}_q(n+k) - \hat{y}_q(n+k))^2 + \sum_{r=1}^R \sum_{k=1}^{N_{u_r}} \lambda_r (\Delta u_r(n+k))^2 \quad (3.3.12)$$

burada  $N_{y_q}$ 'lar kestirim ufuklarıdır ve  $\lambda_r$ 'lar da denetim girdileri değişimleri için ceza terimleridir. Bu en iyilemeyi gerçekleştirmek için aşağıdaki gibi genel bir güncelleme kuralı kullanılır:

$$\mathbf{u}_{\text{optimal}} = \mathbf{u}_{\text{candidate}} + s\mathbf{p}, \quad (3.3.13)$$

burada  $\mathbf{p}$  arama yönü ve  $s$  de adım-uzunluğu olarak adlandırılır. Ardından, en iyilenmiş denetim vektöründe  $(n+1)$  zaman indeksine karşı düşen elemanlar, yani  $u_1^*(n+1), \dots, u_R^*(n+1)$  elemanları, sisteme giriş olarak uygulanır. En iyileme literatüründe,  $\mathbf{p}$  vektörünün bulunması için birinci- ve ikinci-dereceden türev tabanlı yöntemler mevcuttur (NOCEDAL ve WRIGHT, 1999; VENKATARAMAN, 2002). Birinci-dereceden metotlar gradyant vektörünü kullanırken, ikinci-dereceden metotlar buna ilave olarak Hessian matrisini kullanırlar. İşlemsel kolaylık açısından, bu çalışmada, *Dik İniş* (Steepest-Descent – SD) adı verilen yöntem



kullanılmıştır. Bu yöntemde, arama yönü, en büyük artım yönünü gösteren  $\mathbf{g}$  gradyant vektörüne zıt yönde seçilir. Başka bir deyişle,

$$\mathbf{p} = -\mathbf{g}, \quad (3.3.14)$$

burada,

$$\mathbf{g} = \frac{\partial F}{\partial \mathbf{u}_{\text{candidate}}} = \left[ \frac{\partial F}{\partial u_1(n+1)} \quad \cdots \quad \frac{\partial F}{\partial u_1(n+N_{u_1})} \quad \cdots \quad \frac{\partial F}{\partial u_R(n+1)} \quad \cdots \quad \frac{\partial F}{\partial u_R(n+N_{u_R})} \right]^T. \quad (3.3.15)$$

Gradyant vektörü  $\mathbf{g}$ 'nin hesabında gerekli olan kısmi türevler  $r = 1, \dots, R$ ;  $h = 1, \dots, N_{u_r}$  olmak üzere aşağıdaki şekilde bulunur:

$$\begin{aligned} \frac{\partial F}{\partial u_r(n+h)} = & -2 \sum_{q=1}^Q \sum_{k=1}^{N_{y_q}} (\tilde{y}_q(n+k) - \hat{y}_q(n+k)) \frac{\partial \hat{y}_q(n+k)}{\partial u_r(n+h)} \\ & + 2 \lambda_r \sum_{k=1}^{N_{u_r}} (\Delta u_r(n+k)) (\delta_{h,k} - \delta_{h,k-1}) \end{aligned} \quad (3.3.16)$$

burada  $\delta_{h,k}$  Kronecker Delta fonksiyonudur. İlerleyen kısımlarda, SVM modeli kullanılarak,

$\hat{y}_q(n+k)$  tahminlerinin ve  $\frac{\partial \hat{y}_q(n+k)}{\partial u_r(n+h)}$  kısmi türevlerinin nasıl bulunacağı gösterilmiştir.

Eğer (IPLIKCI, 2006)'da SISO sistemler için önerilen SVM-Tabanlı GPC yöntemi MIMO sistemlere uyarlanacaksa, ilk olarak şu anki durum vektörünü aşağıdaki şekilde oluşturulmalıdır:

$$\mathbf{c}(n) = [u_1(n), \dots, u_1(n-n_1), \dots, u_R(n), \dots, u_R(n-n_R), y_1(n-1), \dots, y_1(n-m_1), \dots, y_Q(n-1), \dots, y_Q(n-m_Q)]^T, \quad (3.3.17)$$

Böylelikle, MIMO SVM modelinin  $q^{\text{uncu}}$  çıkışı şu hale gelir,

$$\hat{y}_q(n) = \sum_{j=1}^{\#SV} \alpha_{qj} K(\mathbf{c}(n), \mathbf{x}_j) + b_q, \quad q = 1, \dots, Q. \quad (3.3.18)$$

Bu çalışmada, aşağıdaki gibi verilen radyal tabanlı fonksiyon (Radial Basis Function - RBF) çekirdek fonksiyonu olarak seçilmiştir:

$$K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2}\right),$$

burada  $\sigma$  genişlik parametresidir. Eğer  $d_{j,n}$  uzaklığı,  $j^{\text{inci}}$  destek vektörü  $\mathbf{x}_j$  ile şu anki durum vektörü  $\mathbf{c}(n)$  arasındaki Öklit uzaklığı olarak aşağıdaki gibi tanımlanırsa,

$$\begin{aligned}
d_{j,n} &= (\mathbf{c}(n) - \mathbf{x}_j)^T (\mathbf{c}(n) - \mathbf{x}_j) \\
&= \sum_{r=1}^R \sum_{i=0}^{n_r} (x_{j,i+r+\Gamma_{r-2}} - u_r(n-i))^2 + \sum_{q=1}^Q \sum_{i=1}^{m_q} (x_{j,i+R+\sum_{z=1}^R n_z + \Gamma_{q-2}} - y_q(n-i))^2, \quad (3.3.19)
\end{aligned}$$

burada  $\Gamma_x$  aşağıdaki gibi bir birim basamak fonksiyonudur,

$$\Gamma_x = \begin{cases} 0 & x < 0 \\ 1 & 0 \leq x \end{cases}, \quad (3.3.20)$$

o zaman, çekirdek fonksiyonu şu şekilde yazılabilir,

$$K(\mathbf{c}(n), \mathbf{x}_j) = \exp\left(-\frac{d_{j,n}}{2\sigma^2}\right), \quad (3.3.21)$$

ve, bunun sonucunda, MIMO SVM modelinin  $q^{\text{uncu}}$  çıkışı şu hale gelir,

$$\hat{y}_q(n) = \sum_{j=1}^{\#SV} \alpha_{qj} \exp\left(-\frac{d_{j,n}}{2\sigma^2}\right) + b_q, \quad q = 1, \dots, Q. \quad (3.3.22)$$

Böylelikle, ileri yönelik tahminler aşağıdaki formülle yapılabilir:

$$\hat{y}_q(n+k) = \sum_{j=1}^{\#SV} \alpha_{qj} \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right) + b_q, \quad q = 1, \dots, Q; \quad k = 1, \dots, N_{y_q}, \quad (3.3.23)$$

burada

$$\begin{aligned}
d_{j,n+k} &= \sum_{r=1}^R \sum_{i=0}^{n_r} \begin{cases} (x_{j,i+r+\Gamma_{r-2}} - u_r(n+k-i))^2 & k - N_{u_r} < i \\ (x_{j,i+r+\Gamma_{r-2}} - u_r(n+N_{u_r}))^2 & k - N_{u_r} \geq i \end{cases} \\
&+ \sum_{q=1}^Q \sum_{i=1}^{\min(k, m_q)} (x_{j,i+R+\sum_{z=1}^R n_z + \Gamma_{q-2}} - \hat{y}_q(n+k-i))^2 \\
&+ \sum_{q=1}^Q \sum_{i=k+1}^{m_q} (x_{j,i+R+\sum_{z=1}^R n_z + \Gamma_{q-2}} - y_q(n+k-i))^2. \quad (3.3.24)
\end{aligned}$$

İleri yönelik tahminler bulunduktan sonra,  $\frac{\partial \hat{y}_q(n+k)}{\partial u_r(n+h)}$  kısmi türevleri artık şu şekilde

bulunabilir:

$$\frac{\partial \hat{y}_q(n+k)}{\partial u_r(n+h)} = \sum_{j=1}^{\#SV} \alpha_{qj} \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma^2}\right)}{\partial u_r(n+h)}, \quad (3.3.25)$$

burada

$$\begin{aligned} \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma_q^2}\right)}{\partial u_r(n+h)} &= \frac{\partial \exp\left(-\frac{d_{j,n+k}}{2\sigma_q^2}\right)}{\partial d_{j,n+k}} \frac{\partial d_{j,n+k}}{\partial u_r(n+h)} \\ &= -\frac{1}{2\sigma_q^2} \exp\left(-\frac{d_{j,n+k}}{2\sigma_q^2}\right) \frac{\partial d_{j,n+k}}{\partial u_r(n+h)} \end{aligned} \quad (3.3.26)$$

ve

$$\begin{aligned} \frac{\partial d_{j,n+k}}{\partial u_r(n+h)} &= -2 \sum_{i=0}^{n_r} \begin{cases} (x_{j,i+r+\Gamma_{r-2}} \sum_{z=1}^{r-1} n_z - u_r(n+k-i)) \delta_{k-i,h} & k - N_{u_r} < i \\ (x_{j,i+r+\Gamma_{r-2}} \sum_{z=1}^{r-1} n_z - u_r(n+N_{u_r})) \delta_{N_{u_r},h} & k - N_{u_r} \geq i \end{cases} \\ &\quad - 2 \sum_{q=1}^Q \sum_{i=1}^{\min(k,m_q)} (x_{j,i+R+\sum_{z=1}^R n_z + \Gamma_{q-2}} \sum_{z=1}^{q-1} m_z - \hat{y}_q(n+k-i)) \frac{\partial \hat{y}_q(n+k-i)}{\partial u_r(n+h)} \Gamma_{k-i-h} \\ &\quad q = 1, \dots, Q; k = 1, \dots, N_{y_q}; r = 1, \dots, R; h = 1, \dots, N_{u_r} \end{aligned} \quad (3.3.27)$$

Böylece, arama yönü olan  $\mathbf{p}$ 'nin hesabında gerekli olan ileri yönelik tahminler ve kısmi türevler sistemin MIMO SVM modeli kullanılarak elde edilebilir. Ayrıca, not edilmelidir ki RBF çekirdek fonksiyonu için verilen bu formüller diğer tipteki çekirdek fonksiyonlarına da kolaylıkla uyarlanabilir. Arama yönü  $\mathbf{p}$  bulunduktan sonra, adım-uzunluğu  $s$ 'nin bulunması artık bir-boyutlu en iyileme problemine dönüşür ki bu çalışmada bu problem *Altın Oran (Golden-Section)* yöntemiyle çözülmüştür. Sonuç olarak, MIMO SVM-Tabanlı GPC prosedürü aşağıdaki gibi maddeler halinde açıklanabilir:

**Adım1.** Sistemin dinamiklerini modellemek için gerekli verileri toplamak için ilk olarak, giriş işaretlerinin alt ve üst sınırları, yani  $r = 1, \dots, R$  için  $u_{r\min}$  ve  $u_{r\max}$  değerleri, belirlenir.

Ardından, giriş işaretlerinin en kısa ve en uzun zaman aralıkları, yani  $\tau_{\min}$  ve  $\tau_{\max}$  değerleri, uygun bir şekilde seçilir. Sonra, sistem belli bir periyot boyunca koşturulur. Bu esnada, sisteme, genlikleri  $[u_{r\min}, u_{r\max}]$  aralığında rastgele ve süreleri  $[\tau_{\min}, \tau_{\max}]$  aralığında rastgele değişen bir dizi giriş işareti uygulanır.

**Adım2.** Denklem (3.3.1) ile verilen NARX modelinde görülen  $n_i$ 'ler ve  $m_i$ 'ler belirlenir, ve bir eğitim verileri seti oluşturulur. Ardından, bu setteki tüm giriş ve çıkış değişkenleri  $[0,1]$  aralığına normalize edilir.

**Adım3.** Sistemin MIMO SVM modeli, her bir çıkış için ayrı ayrı oluşturulmuş MISO SVM modellerin birleştirilmesiyle elde edilir. Bir çıkışa ilişkin MISO SVM modeli elde edilirken  $N$  adet veri eğitim için geri kalanı da test için kullanılır.

**Adım4.** Her iterasyonda, ya da örnekleme periyodu süresince, MIMO SVM-Tabanlı GPC algoritması şu şekilde uygulanır:

Sistemin çıkışlarından ölçümleri al, aday denetim vektörü  $\mathbf{u}_{\text{candidate}}$  'yu oluştur ve bu vektöre karşı düşen ileri yönelik tahminleri MIMO SVM modelini kullanarak elde et. Yine MIMO SVM modelini kullanarak, formüller yardımıyla gradyant vektörü  $\mathbf{g}$  'yi bul ve arama yönü olarak  $\mathbf{p} = -\mathbf{g}$  al. Ardından, en iyi adım-uzunluğunu bul ve  $\mathbf{u}_{\text{optimal}} = \mathbf{u}_{\text{candidate}} + s\mathbf{p}$  kuralı ile güncellemeyi yap. Son olarak,  $\mathbf{u}_{\text{optimal}}$  vektöründe  $(n+1)$  zaman indeksine karşı düşen ilk elemanları sisteme uygula.

### 3.3.2 Deney

#### 3.3.2.1 Modelleme Verileri

Sistemin SVM modelinin olabildiğince doğru bir şekilde elde edilebilmesi için, uygun modelleme verisinin oluşturulmasına özel bir önem verilmiştir. Kabul edilebilir bir MIMO denetimi için, SVM modeli sistemi tüm çalışma bölgesinde doğru bir şekilde temsil etmelidir. Modelleme verisinin tüm çalışma bölgesini kaplaması için, veri toplama aşamasında bazı parametreler üzerinde çalışılmış ve onların uygun değerlerinin bulunması sağlanmıştır. Bu parametreler şunlardır:

- *Musluk katsayıları* ( $az_{13}$ ,  $az_{32}$ ,  $az_{10}$ ,  $az_{20}$  ve  $az_{30}$ ):

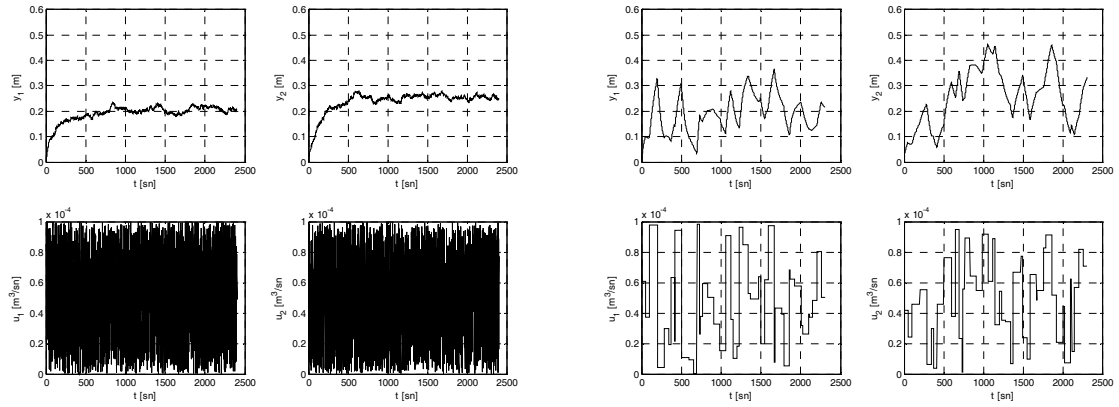
Sistemin beş adet musluk katsayısı vardır:  $az_{13}$ ,  $az_{32}$ ,  $az_{10}$ ,  $az_{20}$  ve  $az_{30}$ . Bu parametrelerin sistem davranışı üzerinde oldukça önemli etkileri vardır. Örneğin,  $az_{13}$  ve  $az_{32}$  parametreleri tanklar arasındaki etkileşim derecesini belirlerken,  $az_{10}$ ,  $az_{20}$  ve  $az_{30}$  parametreleri de tanklardan rezervuara akan sıvı debilerini belirler. Bu parametre değerleri 0 ile 1 arasında değişir: 0 pozisyonunda musluktan hiçbir akış olmazken, 1 pozisyonu musluğun tamamen açık olduğu anlamına gelir. Bu çalışmada, uygun bir modelleme verisi elde etmek için, muslukların pozisyonları deneme-yanılma ile bulunmuştur, ve sonunda parametre değerleri aşağıdaki şekilde seçilmiştir:

$$az_{13} = 0.22, az_{32} = 0.28, az_{10} = 0.40, az_{20} = 0.27, az_{30} = 0.27.$$

- *Giriş işaretlerinin en uzun süresi* ( $\tau_{\text{max}}$ ):

Veri toplama aşamasında,  $u_1(t)$  ve  $u_2(t)$  giriş işaretlerinin genlikleri sırasıyla  $[u_{1\text{min}}, u_{2\text{min}}]$  ve  $[u_{1\text{max}}, u_{2\text{max}}]$  aralıklarında rastgele değiştirilirken, süreleri de  $\tau_{\text{min}}$  ve  $\tau_{\text{max}}$  arasında rastgele değişmektedir. Bu çalışmada,  $\tau_{\text{min}}$  süresi, örnekleme periyodu  $T_s$  'ye eşit seçilmiştir, yani  $\tau_{\text{min}} = T_s = 1$  sn. Musluk katsayılarında olduğu gibi,  $\tau_{\text{max}}$  süresinin de modelleme verileri üzerinde büyük etkisi vardır. Daha ayrıntı vermek

gerekirse,  $\tau_{\max}$  süresi kısaltıldıkça, giriş işaretleri sistemin tüm modlarını uyarmada yetersiz kalırlar ve bunun sonucunda modelleme verileri tüm çalışma bölgesine oranla daha dar bir bantta yer alırlar. Bu durum Şekil 3.3.3a'da görülmektedir. Burada,  $\tau_{\max} = 1$  sn alınarak elde edilen sitem çıkışlarının tüm çalışma bölgesine kıyasla çok dar bir bantta seyrettiğini görmekteyiz. Diğer taraftan, eğer  $\tau_{\max}$  süresi uzadıkça, verilerin taradığı bant daha da genişler ancak  $\tau_{\max}$  iyice uzarsa bu sefere de giriş işaretleri sistemin tüm modlarını uyarmada yine yetersiz kalır. Bu durum Şekil 3.3.3b'de görülmektedir ki burada veriler  $\tau_{\max} = 100$  sn alınarak toplanmıştır. Bu yüzden bu iki marjinal durum arasında en iyi değer bulunması gerekmektedir ki bu çalışmamızda bu değer  $\tau_{\max} = 40$  sn olarak bulunmuştur.

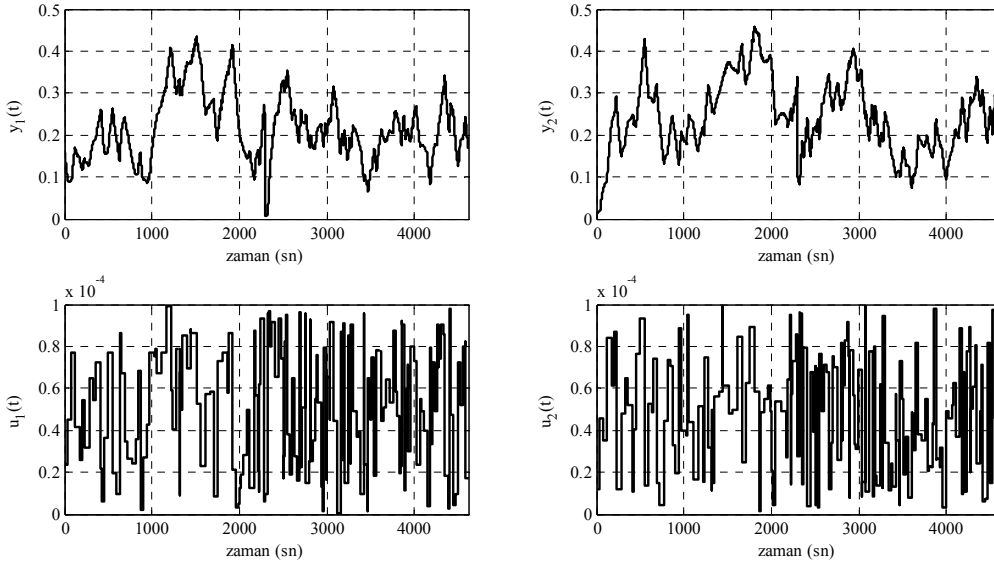


Şekil 3.3.3. a)  $\tau_{\max} = 1$  sn; b)  $\tau_{\max} = 100$  sn.

Sonuç olarak bu çalışmada, musluk parametreleri  $az_{13} = 0.22$ ,  $az_{32} = 0.28$ ,  $az_{10} = 0.40$ ,  $az_{20} = 0.27$ ,  $az_{30} = 0.27$  şeklinde ve  $\tau_{\max}$  da  $\tau_{\max} = 40$  sn olarak seçilmiştir. Bu parametre değerleriyle toplanan veriler Şekil 3.3.4'te görülmektedir.

Ayrıca, sistemde oluşan ölçme hatalarının etkilerini en aza indirmek için hem veri toplama hem de denetim aşamalarında sistemden ölçülen büyüklükler aşağıdaki gibi bir alçak-geçiren filtreden geçirilerek ölçülen büyüklüklerdeki yüksek frekanslı gürültü bileşenleri süzölmüştür:

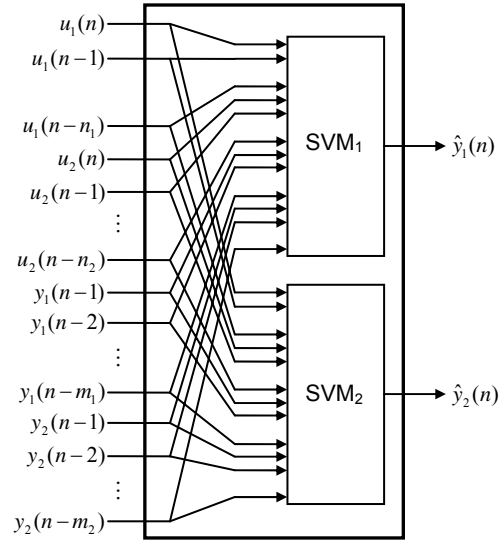
$$H(z) = \frac{0.02483z^{-1} + 0.02224z^{-2}}{1 - 1.672z^{-1} + 0.719z^{-2}} \quad (3.3.28)$$



Şekil 3.3.4. Modelleme için toplanan veriler.

### 3.3.2.2 SVM Modelinin Elde Edilmesi

Toplanan veriler kullanılarak sistemin bir MIMO modelinin elde edilmesi, Şekil 3.3.5'te görüldüğü gibi, her bir çıkış için ayrı ayrı SVM modellerinin birleştirilmesiyle gerçekleştirilmiştir.



Şekil 3.3.5. Üç-tanklı sistem için MIMO SVM modeli.

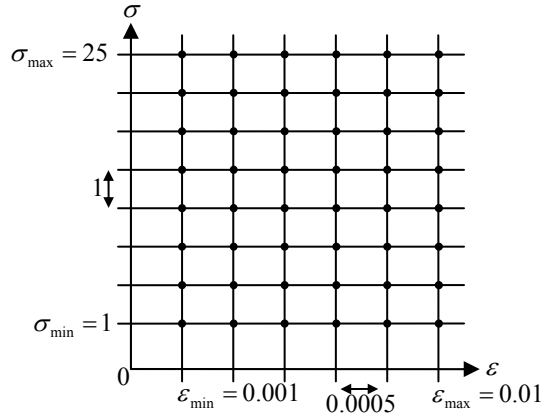
Bu noktada sistemin giriş-çıkış ilişkisinin aşağıdaki gibi olduğu varsayılmıştır:

$$y_1(n) = f_1(u_1(n), \dots, u_1(n-n_1), u_2(n), \dots, u_2(n-n_2), y_1(n-1), \dots, y_1(n-m_1), y_2(n-1), \dots, y_2(n-m_2)) \quad (3.3.29)$$

$$y_2(n) = f_2(u_1(n), \dots, u_1(n - n_1), u_2(n), \dots, u_2(n - n_2), y_1(n - 1), \dots, y_1(n - m_1), y_2(n - 1), \dots, y_2(n - m_2))$$

burada giriş ve çıkış gecikmeleri  $n_1 = n_2 = m_1 = m_2 = 5$  olarak alınmıştır. Her ne kadar üç-tanklı sıvı sisteminin üçüncü dereceden bir sistem olması nedeniyle giriş ve çıkış gecikmelerinin  $n_1 = m_1 = 3$  olarak alınmasının yeterli olacağı düşünülse de, daha güvenilir bir model elde etmek adına bu gecikme sayılarının biraz daha büyük seçilmesinde hiçbir sakınca yoktur.

Modelleme aşamasında kullanılan SVM parametrelerinin belirlenmesine gelince; bu çalışmada ayar parametresi etkisinin görece olarak çok az olması sebebiyle  $C = 1000$  olarak sabit tutulmuştur. Bunun yerine, modelleme başarımında çok daha etkili parametreler olan  $\sigma$  ve  $\varepsilon$  parametrelerinin en uygun değerlerinin bulunmasına çalışılmıştır. Bu amaçla, Şekil 3.3.6'da görüldüğü gibi,  $\sigma\varepsilon$ -düzleminde  $[\sigma_{\min}, \sigma_{\max}]$  ve  $[\varepsilon_{\min}, \varepsilon_{\max}]$  aralıkları arasında bir grid araması yapılmıştır.



Şekil 3.3.6. En iyi  $(\sigma, \varepsilon)$  çifti için grid araması.

Bu aramada, grid üzerindeki her bir  $(\sigma, \varepsilon)$  noktası için verilerin yarısı kullanılarak bir SVM modeli oluşturulmuş diğer yarısıyla da bu model test edilmiştir. Böylece, en küçük test hatasını veren  $(\sigma, \varepsilon)$  çifti belirlenmiştir. Bu işlem, sistemin iki ayrı çıkışına ilişkin iki ayrı SVM modeli için de yapılmıştır. Bunun sonucunda, SVM<sub>1</sub> için en iyi  $(\sigma, \varepsilon)$  çifti  $(8.0, 0.0025)$  olarak bulunmuş ve bu değerlerle 43 destek vektöründen oluşan bir SVM modeli elde edilmiştir. Benzer şekilde, SVM<sub>2</sub> için en iyi  $(\sigma, \varepsilon)$  çifti  $(20.0, 0.0025)$  olarak bulunmuş ve bu değerlerle 11 destek vektöründen oluşan bir SVM modeli elde edilmiştir. Sistemin simetrik olması ve musluk değerlerinin birbirinden çok uzak olmamasına rağmen SVM<sub>1</sub> ve SVM<sub>2</sub> modellerinin neden birbirinden çok farklı  $(\sigma, \varepsilon)$  çiftleriyle elde edilmiş olduğu sorusu akla gelebilir. bunun sebebi tamamen eğitim verileriyle ilgilidir. Şekil 3.3.4'te görülen modelleme için toplanan

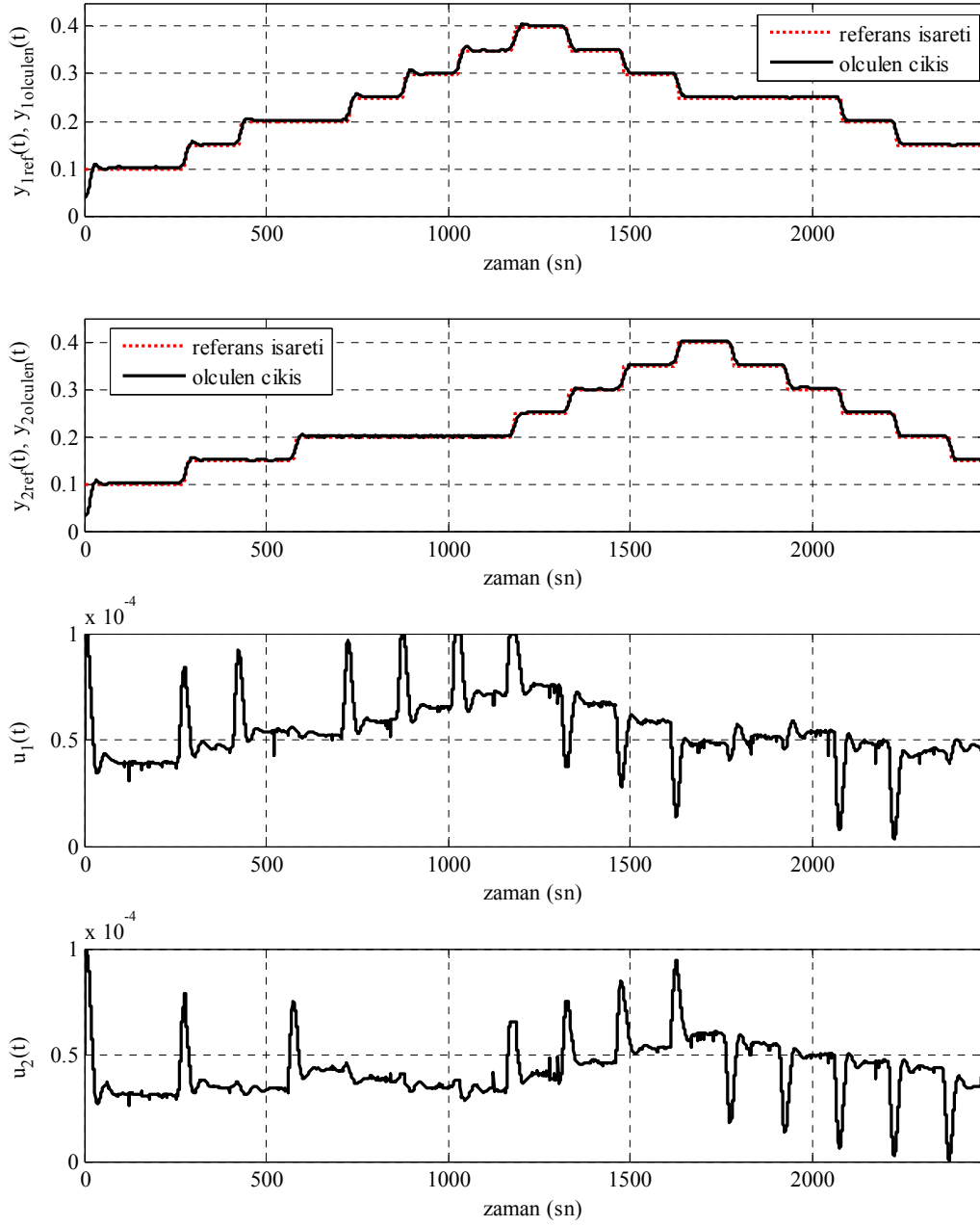
verilere bakıldığında veri toplama süresince her iki pompanın ürettikleri debilerin birbirinden tamamen farklı olduğu görülecektir. Bunun sonucunda tanklardaki sıvı seviyeleri de her örnekleme anında birbirinden farklı olacaktır. Dolayısıyla en iyi SVM<sub>1</sub> ve SVM<sub>2</sub> modellerini veren  $(\sigma, \varepsilon)$  çiftlerinin birbirinden çok farklı yerlerde olması normaldir.

### 3.3.2.3 Deneysel Sonuçlar

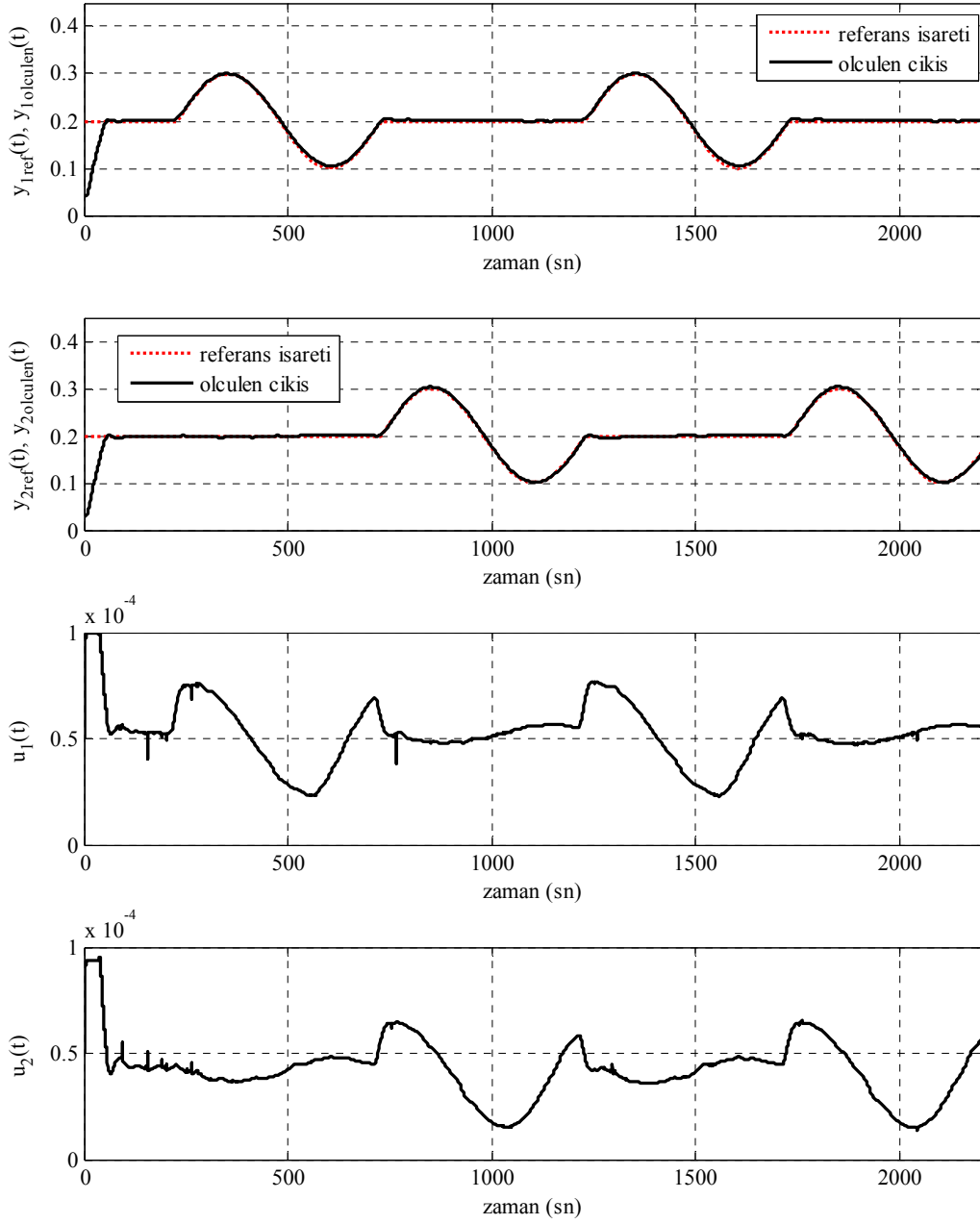
Denetim aşamasında, denetim parametreleri şu şekilde seçilmiştir:  $N_{y_1} = N_{y_2} = 30$ ,  $N_{u_1} = N_{u_2} = 2$  ve  $\lambda_1 = \lambda_2 = 0.5$ . Bu kısımda, en iyiye yakın bu denetim parametreleriyle bulunan deneysel sonuçlar yer almaktadır. Bir sonraki kısımda ise bu denetim parametrelerinin denetim başarımı üzerindeki etkileri incelenmiştir. Deneysel sonuçlar, MIMO SVM-Tabanlı GPC yöntemi için iki farklı referans işareti için denetlenmiştir. Şekil 3.3.7, basamak şeklindeki referans işareti için elde edilen deneysel sonuçları göstermektedir. Burada, bazı anlarda, referans işaretleri aynı anda değişmekteyken, bazı anlarda ise biri sabit kalıp diğeri değişmektedir. Referans işaretlerinin farklı anlarda değiştirilmesinin sebebi, tanklar arası etkileşimin denetleyici tarafından nasıl tolare edileceğinin gözlenmesidir. Buna göre, şekilden de görüleceği gibi, tank<sub>1</sub> için referans işareti değiştirilirken tank<sub>2</sub> için referans 0.20 m'de sabit tutulmuştur ve bunun sonucunda tank<sub>2</sub> için referans sabit olsa bile denetleyici pompa<sub>2</sub>'ye gönderdiği  $u_2(t)$  işaretlerinin uygun şekilde manipüle ederek tank<sub>2</sub>'deki sıvı seviyesini 0.20 m'de sabit tutabilmiştir. Aksi halde, tank<sub>2</sub> için referansın sabit olması nedeniyle pompa<sub>2</sub>'nin debisi de sabit tutulsaydı tank<sub>2</sub>'deki sıvı seviyesi tanklar arası etkileşimden dolayı 0.20 m'de sabit kalmayacaktı. Benzer gözlemler tank<sub>1</sub>'in 0.25 m'de sabit tutulurken tank<sub>2</sub>'deki sıvı seviyesinin değiştiği periyot için de gözlenebilir.

Tanklar arası etkileşimi denetleyicinin tolare edebildiğini daha iyi görmek için deneyler sinüs biçimli referans işareti için de tekrarlanmıştır. Şekil 3.3.8'den de görüleceği gibi, bir tank sinüs biçimli referans işaretini takip ederken, diğer tanka ilişkin referans işareti sabit bile olsa bu tanka sıvı pompalayan pompa sabit bir debiyle değil aksine aradaki etkileşimi karşılayarak tanktaki sıvı seviyesini sabit tutmak için uygun bir debiyle sıvı pompalamaktadır. Sonuç olarak, deneysel sonuçlardan da görüleceği gibi, MIMO SVM-Tabanlı GPC yöntemi oldukça yüksek bir denetim başarımına sahiptir ve tanklar arası etkileşimi başarılı bir şekilde tolare etmiştir.





Şekil 3.3.7. Basamak biçimli referanslar için deneysel sonuçlar.



Şekil 3.3.8. Sinüs biçimli referanslar için deneysel sonuçlar.

Bu deneysel sonuçlar en iyiye yakın bir denetim parametreleri setiyle alınmıştır. Bir sonraki kısımda ise, denetim parametrelerinin  $(N_{y_1}, N_{y_2}, N_{u_1}, N_{u_2}, \lambda_1, \lambda_2)$  denetim başarımı üzerindeki etkileri incelenmiştir.

#### 3.3.2.4 Denetim Parametrelerinin Etkileri

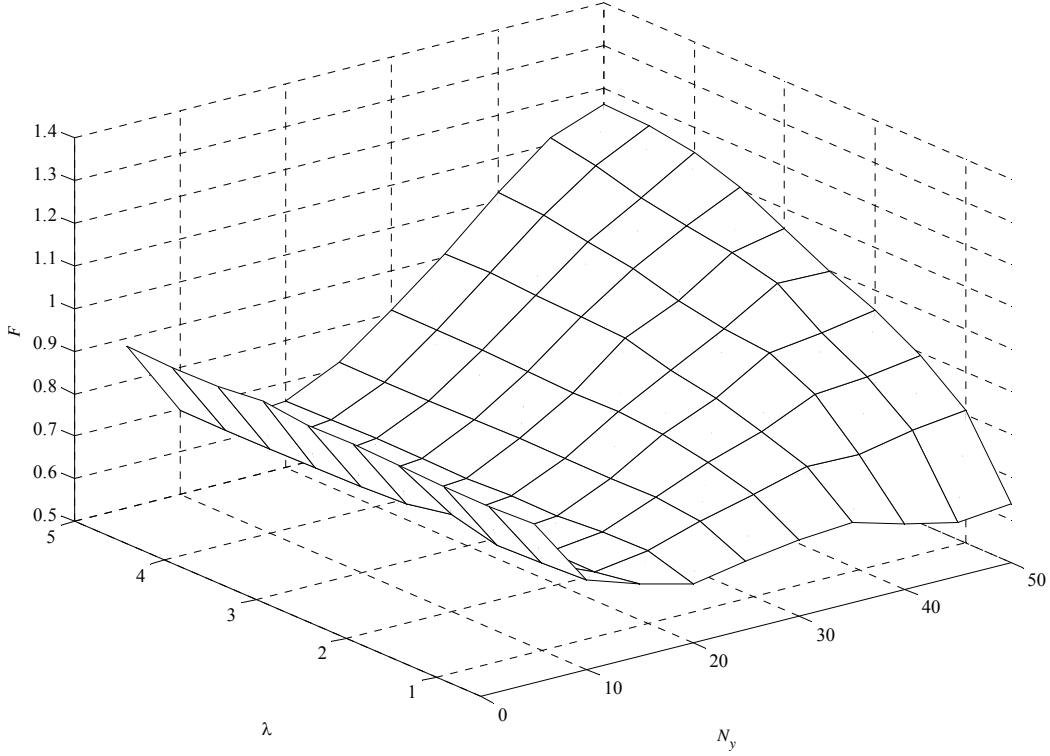
Denetim parametrelerinin etkileri incelenirken, kolaylık olması bakımından, kestirim ufku parametreleri  $N_{y_1} = N_{y_2} = N_y$ , şeklinde ve denetim ufku parametreleri  $N_{u_1} = N_{u_2} = N_u$  şeklinde, ve ceza parametreleri de  $\lambda_1 = \lambda_2 = \lambda$  şeklinde birbirine eşitlendi. Her bir parametrenin etkisi belirlenirken, söz konusu parametre yavaş yavaş artırılırken, diğerleri sabit tutuldu. Her bir durumda, denetleyicinin başarımı aşağıdaki başarımla göstergesi ile ölçüldü:

$$F = \sum_{n=1}^{1500} \sum_{q=1}^2 (\tilde{y}_q(n) - \hat{y}_q(n))^2 + \sum_{n=1}^{1500} \sum_{r=1}^2 \lambda (\Delta u_r(n))^2,$$

buradaki referans işareti önceki kısımda verilen basamak işaretinin daha kısa bir versiyonudur. Deneysel sonuçlar, denetim ufku  $N_u$ 'nun 2 değerinden itibaren önemli bir başarımla iyileştirmesi sağlamadığı görüldüğünden değeri  $N_u = 2$  olarak sabitlenmiştir. Şekil 3.3.9'da farklı denetim parametreleri için denetleyicinin başarımları görülmektedir. Diğer parametreler sabit tutulup  $\lambda$  azaltıldıkça  $F$ 'nin de azaldığı görülmektedir. Bu yüzden de, ceza parametresi  $\lambda = 1.0$  değerine ayarlanmıştır. Son olarak,  $N_y$ 'nin seçimi söz konusu olduğunda, ki bu da etkin bir parametredir, bu parametrenin sadece başarımı değil aynı zamanda da kararlılığı ve işlemsel karmaşıklığı da etkilediğini göz önüne almak gerekir. Küçük  $N_y$  değerleri kararsızlığa, daha doğrusu çıkışta salınımlara yol açabiliyorken, çok büyük  $N_y$  değerleri de işlemsel karmaşıklığı ciddi boyutlara getirebilir. Bu yüzden bu parametre dikkatli seçilmelidir. Kararsızlık ve işlemsel karmaşıklık arasındaki dengeyi dikkate alarak ortalama bir  $N_y$  değeri araması yapıldığında, bu çalışmada  $N_y = 30$  değerinin uygun olduğu görülmüştür.

En iyi  $\lambda$  değeri SISO durumda daha büyük bulunmuştu ve bu durum ortamdaki gürültü seviyesi ile açıklanmıştı. MIMO durumda deney şartlarının değişmemesine rağmen en iyi  $\lambda$  değerinin çok daha düşük çıkması şu şekilde açıklanabilir: Birincisi, SISO başarımla göstergesi ile MIMO başarımla göstergesi birbirinden farklıdır. İlkinde  $\lambda$  ile sadece tek bir denetim işareti ( $u_1$ ) cezalandırılırken ikincisinde tek bir  $\lambda$  ile her iki denetim işareti ( $u_1$  ve  $u_2$ ) birden

cezalandırılmaktadır. dolayısıyla her türlü deney koşulu aynı olsa bile SISO ve MIMO için bulunan en iyi  $\lambda$  değerlerinin tıpatıp aynı çıkması beklenemez. İkincisi, her ne kadar deney koşullarının değişmediği varsayılsa da, sistemin bulunduğu ortamdaki diğer cihazların çalışma durumlarına göre ve hatta deneyin yapıldığı gün ve saate göre ortamdaki gürültü seviyesi değişebilmektedir. Ayrıca, sistemin uzun süreli çalıştırmalar nedeniyle gürültüye duyarlılığının arttığı da göz önüne alınırsa bu sonucun çok şaşırtıcı olmayacaktır.



Şekil 3.3.9. Denetleyicinin farklı parametre değerleri için başarımları.

#### 4. SONUÇLAR

Doğrusal olmayan sistemlerin modele dayalı öngörülü denetimi söz konusu olduğunda, aklı gelen ilk yaklaşım, bu sistemleri belirli çalışma noktalarında doğrusallaştırıp denetimi doğrusal modeller üzerinden yapmaktır. Bu yaklaşımda, denetimi yapılacak sistemlerin matematiksel modelleri bilinmese bile, belirli çalışma noktaları civarında doğrusal modelleri elde etmek mümkündür. Bu yaklaşım her ne kadar teoride ve pratikte oldukça ilerlemiş olsa da bu yaklaşıma alternatif yöntemlerin de geliştirilmesi gerekmektedir. Bu projede, matematiksel modeli bilinmeyen ve doğrusal olmayan sistemlerin modele dayalı öngörülü denetimi için bir yöntem geliştirilmiştir. Modele dayalı öngörülü denetimde en önemli eleman, denetimi yapılacak sistemin modelidir. Çünkü, ileri yönelik öngörüler ve en iyileme gibi birtakım hesaplamalar tamamen model üzerinden yapılmaktadır. Dolayısıyla, modele dayalı öngörülü denetimin başarımı çok büyük oranda modelin başarımına bağlıdır. Modele dayalı öngörülü denetimde kullanılacak bir modelin başarımı belirli kriterlere bağlıdır. İyi bir model mümkün olduğunca geniş bir çalışma bölgesinde, sistem zamanla değişse bile sistemin olası tüm davranış modlarını mümkün olduğunca yüksek bir doğrulukla temsil edebilmelidir. Doğrusal olmayan bir sistem için böylesine bir model geliştirebilmek için araştırmacılar çoğunlukla bulanık mantık ve yapay sinir ağları gibi Esnek İşlem (Soft Computing) araçlarına başvurmuşlardır. Bu araçlar esasen doğrusal olmayan parametrik matematiksel yapılardır. Bu yapılardaki parametreler yapıların içinde çoğunlukla doğrusal olmayan bir biçimde bulunsa da bir kısmı doğrusal da olabilir. Bir sisteme ilişkin bir giriş-çıkış verisini kullanarak bu araçlarla sistemi modellemek için bu yapılardaki parametreler uygun şekilde güncellenir. Doğrusal parametreler yine doğrusal yöntemlerle (örneğin, en küçük kareler) güncellenebilmektedir. Ancak, doğrusal olmayan parametrelerin güncellenmeleri biraz daha zahmetlidir. Bu güncellemeler genellikle türev-tabanlı en iyileme yöntemleriyle yapılmakla beraber zaman zaman türev kullanmayan yöntemler de kullanılmaktadır. Türev-tabanlı yöntemler belli bir başarımı yakalayacak kadar iyi sonuç verse de ortak bir sorunu aşmak zorundadırlar: yerel minimuma takılma. Bu yöntemler belli bir çalışma noktası civarında Taylor açılımına dayanmaktadırlar ve bu açılımdaki yüksek dereceden terimlerin atılması sonucunda yaklaşık bir doğruluğa sahiptirler. Açılım sadece belli bir nokta civarında olduğundan bundan elde edilen bilgi global olmayıp bu çalışma noktası civarında yerel bir bilgidir. Bunun sonucunda da türev-tabanlı yöntem, başlangıç noktasına bağlı olarak kendine en yakın yerel minimuma yakınsar ve müdahale edilmezse orada kalır. Bu yerel minimumun tatmin edici bir model vermesi garanti değildir. Bu tamamen eldeki verilerin karmaşıklığına, yapının karmaşıklığına ve başlangıç noktasına bağlıdır. Yinede, bulanık mantık ve yapay

sinir ağırları gibi Esnek İşlem araçlarıyla elde edilmiş modellere dayalı öngörülü denetimin başarılı uygulamalarını literatürde bulmak mümkündür. Bu çalışmada ise, Esnek İşlem araçlarına alternatif olarak modellemede kullanılabilecek bir yapı ile modele dayalı öngörülü denetimin deneysel uygulamaları yapılmıştır. Bu çalışmada kullanılan yapılara, Destek Vektör Makineleri (Support Vector Machines - SVMs) adı verilmektedir. Bunlar da yapay sinir ağırları gibi doğrusal olmayan parametrik matematiksel yapılardır. Bu yapıların tercih edilmesinin en önemli nedeni global minimumu garanti etmeleridir. SVM yapıları, verilen bir modelleme verisindeki her bir veri için yapılmasına izin verilen en büyük hata değerinden daha küçük hata üretecek şekilde elde edilebilir. Böylece SVM yapısının her bir veri için ürettiği hata belli bir değerin altında kalır. Tabii burada da aşırı öğrenme gibi problemler ortaya çıkabilmektedir.

Bu projede kullanılan yöntemler çevrimiçi ve çevrimdışı olmak üzere iki farklı gruba ayrılabilir. Çevrimdışı yöntemlerde sistemin SVM modeli önceden (çevrimdışı) elde edilip denetim daha sonra yapılıyorken çevrimiçi yöntemlerde SVM modellemesi ve denetim aynı anda (çevrimiçi) yapılmaktadır. Projede uygulaması yapılan çevrimdışı yöntemler hem SISO hem de MIMO olarak kullanılmışlardır. Çevrimiçi yöntemin ise sadece SISO uygulaması gerçekleştirilmiştir. Bütün bu yöntemlerin testleri bir laboratuvar ortamında kurulu, üç-tanklı sıvı sistemi üzerinde yapılmıştır. Bu sistem bu yöntemlerin uygulanabilmesi için çok ideal bir sistemdir. Hem SISO hem de MIMO olarak kullanılabilir. Ayrıca musluk parametreleri değiştirilerek sistemin karmaşıklık ve karşılıklı etkileşim dereceleri değiştirilebilmektedir. Sistemin tek dezavantajı sadece C++ komutlarıyla çalışabiliyor olmasıdır. Bir yöntem denenmeden önce bu sistemin uygun musluk değerleri deneme-yanılmalarla belirlenmiştir.

Projede kullanılan yöntemlerin başarımı oldukça yüksektir. Ancak, bu başarımı etkileyen pek çok faktörün bulunduğu proje süresi boyunca gözlenmiştir. Bu faktörlerin bir kısmı yöntemlerin kendisindedir ancak bir kısmı da dışsal etkilerdir. Yöntemlerin kendisinden gelen faktörler kaçınılmazdır. Çünkü bu yöntemlerin temel varsayımı, denetimi yapılacak sistemin denklemlerinin bilinmediğidir ki bu durumda modelleme yapmak gerekir ve bunun sonucunda da modelleme parametreleri etkili olmaya başlar. Sistemden veri toplarken kullanılan  $\tau_{max}$  parametresi toplanan verilerin sistemin tüm çalışma bölgesine dağılımı ve sistemin tüm modlarını uyarması bakımından çok etkilidir. Bu yüzden, bu projede bu parametrenin uygun bir şekilde belirlenmesine özen gösterilmiştir. Toplanan veri sayısı da modelleme başarımında etkilidir ancak sistem yeterli sayıda veri toplanmasına elverişli olduğundan bu konuda bir sorun yaşanmamıştır. Modelleme verileri toplandıktan sonra, NARX parametreleri belirlenip veriler normalize edilerek modelleme için hazır hale getirilir.

NARX parametreleri, modelde yer alan geçmiş giriş ve çıkış sayılarıdır. Bu parametrelerin belirlenmesi için literatürde önerilmiş yöntemler kullanılmaktadır. Yeterince büyük seçilmesi koşuluyla bu parametrelerin denetim başarımı üzerinde çok büyük etkileri yoktur. Modelleme verileri hazır hale geldiğinde SVM modelinin elde edilmesi için öncelikle bir çekirdek fonksiyonunun seçilmesi gerekir. Çekirdek fonksiyonunun seçimi modelleme başarımında önemlidir. Ancak bu çalışmada, çok yaygın olarak kullanıldıkları için sadece Gaussian tipli çekirdek fonksiyonu kullanılmıştır. SVM modeli elde edilirken karşımıza modelleme başarımını etkileyen iki önemli parametre çıkmaktadır. Bunlardan ilki çekirdek fonksiyonunun genişlik parametresi olan  $\sigma$  parametresi, diğeri de her bir veri için modelin yapmasına izin verilen en büyük hata değeri olan  $\epsilon$  parametresidir. SVM modelinin genelleme yeteneği  $\sigma$  parametresi ile doğru orantılıdır. Bu parametre küçüldükçe model daha lokalize olurken büyüdüğü de daha global hale gelir. Bu yüzden bu iki uç durum arasında bir denge kurmak gerekir. Diğer taraftan, SVM modelinin toplam öğrenme hatası da  $\epsilon$  parametresi ile doğru orantılıdır. Bu parametre küçüldükçe öğrenme hatası da küçülür en uç durumda aşırı öğrenme veya ezberleme denilen durum oluşabilir.  $\epsilon$  parametresi büyüdüğü de öğrenme hatası büyür uç durumda da az öğrenme gibi bir durum oluşur. Burada da bir dengenin bulunması gerekmektedir. Bu iki parametre modelleme başarımını dolayısıyla da denetim başarımını çok etkilemektedir. Her ikisi için de uygun değerlerin bulunması kolay değildir. Uygun değerlerin bulunmasını sağlamak amacıyla toplanan verilerin bir kısmı test verileri olarak ayrılmış, bu iki parametrenin birlikte oluşturdukları iki-boyutlu parametre uzayında oluşturulan bir grid üzerindeki her bir parametre çifti için SVM modeli elde edilmiş ve her bir modelin test hatası bulunmuştur. En küçük test hatasını veren SVM modeline ilişkin parametre çifti uygun parametre çifti olarak seçilmiştir. Bu oldukça zaman alıcı bir işlemdir ancak bir kez yapıldığında, sistemin parametreleri değişmediği sürece, aynı SVM modeli sürekli kullanılabilir. SVM modeli elde edildikten sonra artık bu modele dayalı öngörülü denetim yapılabilir. Bu denetim esnasında da bazı denetim parametreleri karşımıza çıkmaktadır. Bunlardan en etkili olanları denetim ufku  $N_y$  ve ceza parametresi  $\lambda$  parametreleridir.  $N_y$  parametresi denetim ufkunun uzunluğunu belirler. Değeri azaldıkça, denetleyicinin refleksleri artar, yani küçük bir değişime ani tepkiler vermeye başlar. Çok küçük seçilmesi durumunda sistemin çıkışında osilasyonlara bile yol açabilir. Denetim ufkunun değeri arttıkça denetleyici daha temkinli ve daha az tepkisel bir moda doğru yaklaşır. Çok büyük  $N_y$  değerlerinde sistemin çıkışı referans işaretindeki değişimleri takip edememeye başlar ve ayrıca işlemsel yük çok fazla miktarda artar. Bu parametre için de bir dengenin bulunması gerekir. Son olarak, ceza parametresi olan  $\lambda$  parametresi de denetim işaretindeki ani değişimleri önlemesi bakımından denetim başarımında etkilidir. Esasında,

SVM modelinin sistemi çok iyi temsil ettiği ve ortamda herhangi bir ölçüm hatasına yol açacak bir gürültü bileşeninin olmadığı ideal durumda ceza parametresinin çok küçük hatta sıfır seçilmesinde bir sakınca yoktur. Ancak, her ne kadar SVM yaklaşımı sistemi çok iyi modellese bile ortada az da olsa bir modelleme gürültüsü olacaktır. Bunun yanı sıra, deneylerin yapıldığı ortamdan kaynaklanan gürültü bileşenleri de mevcuttur. Bu durumda ceza parametresinin çok küçük seçilmesi, denetleyici işaretinde çok ani değişimlere izin verir ki bu pratik açıdan istenmeyen bir durumdur. Diğer taraftan ceza parametresinin çok büyük seçilmesi durumunda da sistemin çıkışı referans işareti takip edememeye başlar. O yüzden bu parametrenin seçiminde de bir dengenin bulunması gerekmektedir.

Benzer şekilde çevrimiçi denetim başarımında da parametrelerin etkileri ön plana çıkmaktadır. Modelleme ve denetim eşzamanlı olarak yapıldığından, modelleme parametreleri denetim başarımında çok daha etkili hale gelir. Çevrimiçi denetimde sistem parametreleri sabitken değişken referans işaretinin uygun bir şekilde takip edilebilmesi için  $\sigma$  parametresi yeterince büyük seçilmelidir. Aksi halde SVM modeli daha lokalize olur ve referans işaretin takibi güçleşir. Çok büyük seçilmesi durumunda da çevrimiçi öğrenme algoritmasında (AOSVR) sorunlara yol açar. Benzer şekilde  $\epsilon$  parametresi de yeterince büyük seçilmelidir. Çok küçük seçilmesi aşırı öğrenme ile birlikte AOSVR’de sorunlara, çok büyük seçilmesi de yetersiz öğrenme ile birlikte SVM modelinin gelişmemesine yol açar. Diğer taraftan, çevrimiçi denetimden esas olarak referans işareti sabitken sistemin bir ya da birkaç parametresinde değişimler oluşması durumunda denetleyicinin bu değişimlere uyarlanması istenmektedir. Bu da ancak SVM modelinin bu değişimlere ayak uydurmasıyla olur.  $\sigma$  parametresinin büyük seçilmesi, SVM modelinin küçük parametre değişimlerine duyarlı kalmasına yol açar, yani model kendini yenilemez ve bunun sonucunda belirli bir sürekli hal hatası oluşur.  $\sigma$  parametresi küçüldükçe SVM modelinin parametre değişimlerine olan refleksi artar ancak çok küçük değerlerde modelin lokalize hale gelmesinin yanı sıra modeldeki destek vektör sayısı artar ve AOSVR algoritmasında sorunlar yaşanabilir. Aynı gözlemler  $\epsilon$  parametresi için de geçerlidir. Çevrimdışı denetleyici için denetim ufku  $N_y$  ve ceza parametresi  $\lambda$ ’nın denetim başarımı üzerindeki etkileri çevrimiçi denetleyici için de benzer şekildedir.

Özetlemek gerekirse, karşı karşıya olduğumuz denetim problemleri için ortaya konulan yöntemler kaçınılmaz olarak beraberinde parametre seçimi problemini getirmektedir. Veri toplama, modelleme ve denetim aşamalarında seçilmesi gereken bu parametrelerin etkileri bir önceki bölümde grafiklerle sayısal olarak verilmiş, yukarıda da kısaca özetlenmiştir. Yine de, bu projede kullanılan yöntemler için bu parametrelerin en iyiye yakın değerlerinin birbirine



yakın olduğu gözlenmiştir. Sayısal değerler vermek gerekirse, düzgelenmiş (normalize) verilerle çalışılması durumunda, üç-tanklı sıvı sisteminin ilgili musluk değerleri için en uygun  $\tau_{max}$  parametresi örnekleme periyodunun 40 katıdır. SVM modeline ilişkin parametrelerden  $\sigma$  parametresinin en iyi değeri 10 civarındayken,  $\varepsilon$  parametresinin en iyi değeri de 0.01 civarındadır. Denetim parametrelerinden  $N_y$ 'nin 20 civarlarında seçilmesi uygunken,  $\lambda$ 'nin değerinin [1, 5] aralığında seçilmesi uygundur.  $\lambda$ 'nin en iyi değeri aslında ortamdaki o anda kaynaklanan gürültü bileşenlerine bağlıdır. Her ne kadar üç-tanklı sıvı sistemi için verilen bu değerlerin farklı bir sistem için en uygun değerler olmama olasılığı bulursa da bu değerler, uygun değerlerin bulunmasına yönelik araştırmada fikir vermektedir.

Sonuç olarak bu proje dahilinde, matematiksel modelleri bilinmeyen ve doğrusal olmayan sistemlerin çevrimiçi ve çevrimdışı denetimleri için SVM-Tabanlı GPC yöntemleri geliştirilerek üç-tanklı sıvı sistemine başarılı bir şekilde uygulanmıştır. Projede, öngörülen tüm hedeflere ulaşılmıştır. Geliştirilen yöntemlerin kaçınılmaz olarak beraberinde getirdiği bir problem olan "parametre seçimi" problemi üzerinde oldukça çok çalışılmış ve kullanılan yöntemler için birbirine çok yakın en iyi parametre değerleri bulunmuştur.

Her ne kadar projede geliştirilen bu yöntemlerin daha farklı gerçek-zamanlı denetim uygulamalarında kullanılabilecek durumda oldukları söylenebilse de hala bu yöntemlerin iyileştirme gerektiren yanları bulunmaktadır: En önemlisi kuşkusuz parametre seçimi konusundadır. Parametre seçiminin yanı sıra çekirdek fonksiyonunun seçimi de aslında bir araştırma konusudur. Çünkü, modellenecek sistemin dinamik davranışlarına ilişkin önceden sahip olunan bilgilerle daha iyi bir çekirdek fonksiyonu seçimi yapılabilir. Ayrıca, çevrimiçi AOSVR algoritmasına kıyasla daha gürbüz çevrimiçi SVM modelleme algoritmalarının geliştirilmesi, çevrimiçi SVM-Tabanlı GPC yönteminin daha güvenilir olmasını sağlayacaktır. Projede geliştirilen yöntemlerden kaynaklanmayan ancak proje süresi boyunca çok sık karşı karşıya kalınan "ölçüm gürültüsü" problemi de çözülmeyi bekleyen konular arasındadır. Gürültülü bir ortamdaki bir ölçümün hiçbir zaman hatasız olması beklenemez ancak gürültü ile gerçek-zamanlı sistemden alınan veriyi birbirinden en az kayıpla ayırtıracak filtreler tasarlanarak SVM tabanlı denetim yöntemlerinin başarımları artırılabilir.

**REFERANSLAR**

- ALEXANDRIDIS A. ve SARIMVEIS H., "Nonlinear Adaptive Model Predictive Control Based on Self-Correcting Neural Network Models," *AIChE Journal*, Cilt. 51, s. 2495–2506, 2005.
- AMIRA DTS200 - Laboratory Setup Three Tank System, Amira GmbH, Duisburg, 2000.
- CAMACHO E.F., "Constrained Generalized Predictive Control," *Automatica*, Cilt. 38, s. 327–332, 1993.
- CAMACHO E.F. ve BORDONS C., "Model Predictive Control," Springer, Berlin, 2003.
- CHEN J. ve YEA Y., "Neural Network-based Predictive Control for Multivariable Processes," *Chem. Eng. Comm.*, Cilt. 189, s. 865–894, 2002.
- CLARKE D.W., MOHTADI C. ve TUFFS P.C., "Generalized Predictive Control – part 1: The Basic Algorithm," *Automatica*, Cilt. 23, s. 137–148, 1987.
- CLARKE D.W., MOHTADI C. ve TUFFS P.C., "Generalized Predictive Control – part 2: Extensions and Interpretations," *Automatica*, Cilt. 23, s. 149–163, 1987.
- CLARKE D.W. ve MOHTADI C., "Properties of Generalized Predictive Control," *Automatica*, Cilt. 25, s. 859–875, 1989.
- CLARKE D.W., "Applications of Generalized Predictive Control to Industrial Processes," *IEEE Control Systems Magazine*, Cilt.122, s.49–55, 1988.
- CRISTIANINI N. ve TAYLOR J.S., "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods," Cambridge University Press, New York, 2000.
- HAGAN M.T., DEMUTH H.B. ve De JESUS O., "An introduction to the Use of Neural Networks in Control Systems," *International Journal of Robust and Nonlinear Control*, Cilt. 12, s. 959–985, 2002.

- HU J.Q. ve ROSE E., "Generalized Predictive Control Using a Neuro-Fuzzy Model," *International Journal of Systems Science*, Cilt. 30, s. 117–122, 1999.
- HUANG Y.L., LOU H.H., GONG J.P.ve EDGAR T.F., "Fuzzy Model Predictive Control," *IEEE Transactions on Fuzzy Systems*, Cilt. 8, s. 665–678, 2000.
- IPLIKCI S., "Support Vector Machines-based Generalized Predictive Control," *International Journal of Robust and Nonlinear Control*, Cilt. 16, s. 843–862, 2006.
- IPLIKCI S., "Online Trained Support Vector Machines-based Generalized Predictive Control of Non-linear Systems," *International Journal of Adaptive Control and Signal Processing*, Cilt. 20, s. 599–621, 2006.
- De KEYSER R.M.C. ve Van CAUWENBERGHE A.R., "Extended Prediction Self Adaptive Control," *Proceedings of the 7th IFAC Symposium on Identification and System Parameter Estimation*, 1985, s.1255–1260.
- LAWRYNCZUK M., "A Family of Model Predictive Control Algorithms with Artificial Neural Networks," *Int. J. Appl. Math. Comput. Sci.*, Cilt. 17, s. 217–232, 2007.
- LAWRYNCZUK M., "Efficient Nonlinear Predictive Control Based on Structured Neural Models," *Int. J. Appl. Math. Comput. Sci.*, Cilt. 19, s. 233–246, 2009.
- LI N., LI S.Y.ve XI Y.G., "Multi-model Predictive Control Based on the Takagi-Sugeno Fuzzy Models: a Case Study," *Information Sciences*, Cilt. 165, s. 247–263, 2004.
- MA J., THEILER J. ve PERKINS S., "Accurate On-line Support Vector Regression," *Neural Computation*, Cilt. 15, s. 2683–2703, 2003.
- MAHFOUF M., ABBOD M.F. ve LINKENS D.A., "Online Elicitation of Mamdani-type Fuzzy Rules via TSK-based Generalized Predictive Control," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Cilt. 33, s. 465–475, 2003.
- MARTINEZ M., SENENT J.S. ve BLASCO X., "Generalized Predictive Control Using Genetic Algorithms (GAGPC)," *Engineering Applications of Artificial Intelligence*, Cilt. 11, s. 355–367, 1998.

- NOCEDAL J. ve WRIGHT S.J., "Numerical Optimization," Springer Series in Op. Res., Springer-Verlag, NY, 1999.
- QIN S.J. ve BADGWELL T.A., "A Survey of Industrial Model Predictive Control Technology," Control Engineering Practice, Cilt.11, s.733–764, 2003.
- PARLOS A.G., PARTHASARATHY S. ve ATIYA A.F., "Neuro-predictive Process Control Using Online Controller Adaptation," IEEE Transactions on Control Systems Technology, Cilt.9, s. 741–754, 2001.
- RICHALET J., "Industrial applications of Model-Based Predictive Control," Automatica, Cilt. 29, s. 1251–1274, 1993.
- RICHALET J.A., RAULT A., TESTUD J.L. ve PAPON J., "Model Predictive Heuristic Control: Applications to an Industrial Process," Automatica, Cilt.14, s.413–428,1978.
- ROSSITER J.A., "Model-Based Predictive Control: A Practical Approach," CRC Press, Florida, 2003.
- ROUBOS J.A., MOLLOV S., BĂBUSKA R. ve VERBRUGGEN H.B., "Fuzzy Model-based Predictive Control Using Takagi-Sugeno Models," International Journal of Approximate Reasoning, Cilt. 22, s. 3–30, 1999.
- SARIMVEIS H. ve BAFAS G., "Fuzzy Model Predictive Control of Non-linear Processes Using Genetic Algorithms," Fuzzy Sets and Systems, Cilt. 139, s. 59–80, 2003.
- SCHÖLKOPF B., BURGESS C.J.C. ve SMOLA A.J., "Advances in Kernel Methods: Support Vector Learning," The MIT Press, Cambridge MA, 1999.
- SMOLA A.J. ve SCHÖLKOPF B., "A Tutorial on Support Vector Regression," NeuroCOLT Tech. Rep. No. NC-TR-98-030, Royal Holloway College, University of London, 1998.
- SOETERBOEK R., "Predictive Control: A Unified Approach," Prentice Hall, Englewood Cliffs, NewYork, 1992.

SOLOWAY D. ve HALEY P.J., "Neural Generalized Predictive Control: a Newton-Raphson algorithm," Proceedings of the IEEE International Symposium on Intelligent Control, 1996, s: 277–282.

TAN K.K., LEE T.H., HUANG S.N. ve LEU F.M., "Adaptive-Predictive Control of a Class of SISO Nonlinear Systems," Dynamics and Control, Cilt. 11, s. 151–174, 2001.

VAPNIK V., "The Nature of Statistical Learning Theory," Springer-Verlag, London, 1995.

VAPNIK V., "Statistical Learning Theory," John Wiley, New York, 1998.

VAPNIK V., "The Support Vector Method of Function Estimation," Nonlinear Modeling Advanced Black Box Techniques, Kluwer Academic Publishers, Boston, 1998.

VENKATARAMAN P., "Applied Optimization with MATLAB Programming," John Wiley and Sons, New York, 2002.

WANG D.H. ve SOH C.B., "Adaptive Neural Model-based Decentralized Predictive Control," International Journal of Systems Science, Cilt. 31, s. 119–130, 2000.

YU D.L. ve GOMM J.B., "Implementation of Neural Network Predictive Control to a Multivariable Chemical Reactor," Control Engineering Practice, Cilt. 11, s. 1315–1323, 2003.

**TÜBİTAK**  
**PROJE ÖZET BİLGİ FORMU**

<b>Proje No:</b> 106E125
<b>Proje Başlığı:</b> Destek Vektör Makineleriyle Gerçek-Zamanlı Genelleştirilmiş Öngörülü Kontrol
<b>Proje Yürütücüsü ve Araştırmacılar:</b> Doç. Dr. Serdar İPLİKÇİ (yürütücü) Yrd. Doç. Dr. Sezai TOKAT (araştırmacı)
<b>Projenin Yürütüldüğü Kuruluş ve Adresi:</b> Pamukkale Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği, Kınıklı Kampüsü, 20040 Denizli
<b>Destekleyen Kuruluş(ların) Adı ve Adresi:</b>
<b>Projenin Başlangıç ve Bitiş Tarihleri:</b> <b>Başlangıç:</b> 1 Şubat 2007 <b>Bitiş:</b> 1 Şubat 2010
<b>Öz (en çok 70 kelime)</b> Bu projede, Destek Vektör Makineleri (Support Vector Machines - SVMs) yaklaşımıyla matematiksel modeli bilinmeyen ve doğrusal olmayan sistemlerin denetimi konusunda bir çalışma yapılmıştır. Denetlenecek sistemin SVM modeli, Genelleştirilmiş Öngörülü Denetim (Generalized Predictive Control - GPC) çerçevesinde kullanılmıştır. Bu çalışma sonucunda geliştirilen yöntemler gerçek bir sisteme (üç-tanklı sıvı sistemi) başarılı bir şekilde uygulanmıştır.
<b>Anahtar Kelimeler:</b> Genelleştirilmiş Öngörülü Denetim, Destek Vektör Makineleri, MIMO Modelleme ve Denetim.
<b><a href="#">Fikri Ürün Bildirim Formu</a> Sunuldu mu?</b> Evet <input type="checkbox"/> Gerekli Değil <input checked="" type="checkbox"/>

Fikri Ürün Bildirim Formu'nun tesliminden sonra 3 ay içerisinde patent başvurusu yapılmalıdır.

**Projeden Yapılan Yayınlar:**

- 1) IPLIKCI S., "A Support Vector Machines Based Control Application to the Experimental Three-Tank System," ISA Transactions dergisine "minor revision" ile kabul edildi.
- 2) IPLIKCI S., "Controlling the Experimental Three-Tank System via Support Vector Machines," 9th International Conference on Adaptive and Natural Computing Algorithms, Kuopio, Finland, 2009, Lecture Notes in Computer Sciences, Vol. 5495, 391-400, Springer-Verlag, Berlin Heidelberg, 2009.
- 3) IPLIKCI S., "Destek Vektör Makineleriyle Deneysel Sıvı Tankının Denetimi", Türkiye Otomatik Kontrol (TOK) Toplantısı 2008 Bildiriler Kitabı, 748-753, İstanbul, 2008.
- 4) IPLIKCI S., "Destek Vektör Makineleriyle Çevrimiçi Denetim", Türkiye Otomatik Kontrol (TOK) Toplantısı 2007 Bildiriler Kitabı, 271-276, İstanbul, 2007.