

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**PHYSICS-INFORMED LEARNING OF
INTERPRETABLE SURROGATES
FOR MODELING AND CONTROL**



M.Sc. THESIS

Murat Furkan MANSUR

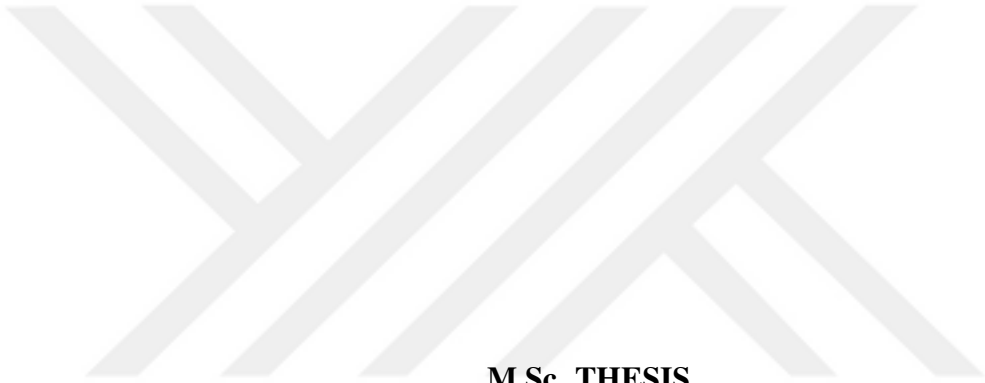
Department of Control and Automation Engineering

Control and Automation Engineering Programme

JULY 2026

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**PHYSICS-INFORMED LEARNING OF
INTERPRETABLE SURROGATES
FOR MODELING AND CONTROL**



M.Sc. THESIS

**Murat Furkan MANSUR
(504231128)**

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Tufan KUMBASAR

JULY 2026

**MODELLEME VE KONTROL İÇİN
YORUMLANABİLİR VEKİL DİNAMİKLERİN
FİZİK BİLGİLİ ÖĞRENİMİ**

YÜKSEK LİSANS TEZİ

**Murat Furkan MANSUR
(504231128)**

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Tufan KUMBASAR

TEMMUZ 2026

Murat Furkan MANSUR, a M.Sc. student of ITU Graduate School student ID 504231128 successfully defended the thesis entitled “PHYSICS-INFORMED LEARNING OF INTERPRETABLE SURROGATES FOR MODELING AND CONTROL”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

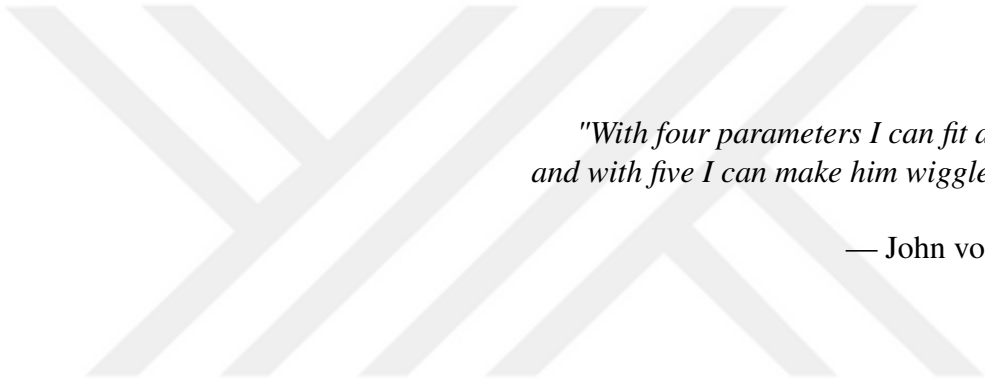
Thesis Advisor : **Prof. Dr. Tufan KUMBASAR**
Istanbul Technical University

Jury Members : **Asst. Prof. Dr. Yusuf Hüseyin ŞAHİN**
Istanbul Technical University

Asst. Prof. Dr. Bahadır ÇATALBAŞ
Yildiz Technical University

Date of Submission : **4 June 2026**
Date of Defense : **6 July 2026**





*"With four parameters I can fit an elephant,
and with five I can make him wiggle his trunk."*

— John von Neumann

To the things we can control,



FOREWORD

I would like to express my sincere gratitude to my supervisor, Prof. Dr. Tufan Kumbasar, for his invaluable guidance, continuous support, and constructive feedback throughout this work. His expertise and encouragement have been instrumental in shaping the direction of this thesis.

Throughout this journey, I have frequently found myself humbled and inspired by the brilliance of the mathematicians and scientists who came before us. Exploring the depths of dynamical systems and control, I have been fascinated by their profound insight, their relentless hard work, and the elegant mathematical foundations they built. Standing on the shoulders of these giants has been both a privilege and a great source of motivation for me.

I am profoundly grateful to my parents for their unconditional love, their endless sacrifices, and for instilling in me the value of curiosity and education from a young age. I would also like to thank my sisters for their constant encouragement and for always being a source of warmth and support in my life.

Finally, a special note of thanks must go to my two feline companions, Yulaf and Loki, who provided their own unique forms of "assistance" during the writing process. I thank Yulaf for constantly demanding to sit on my lap to ensure I took necessary breaks, and Loki for her determination to sit on my keyboard, occasionally contributing her own unintelligible lines to my code.

July 2026

Murat Furkan MANSUR
Control and Automation Engineer

TABLE OF CONTENTS

	Page
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xiii
SYMBOLS	xv
LIST OF TABLES	xvii
LIST OF FIGURES	xix
SUMMARY	xxi
ÖZET	xxiii
1. INTRODUCTION	1
1.1 Problem Definition and Motivation	1
1.2 Literature Review	3
1.3 Purpose of Thesis	5
1.4 Thesis Structure	6
2. PRELIMINARIES	9
2.1 System Identification	9
2.2 LPV and Quasi-LPV System Model	10
2.3 Physics-Informed Neural Networks	12
2.4 Inverse Physics-Informed Neural Networks	13
3. PHYSICS-INFORMED LEARNING OF INTERPRETABLE SURRO- GATES	17
3.1 The Second-Order Surrogate Hypothesis	17
3.2 Dual Network Architecture	19
3.2.1 Solution network	20
3.2.2 Parameter network	22
3.3 Curriculum Learning	23
3.4 Loss Functions and Optimization	26
4. GAIN-SCHEDULED CONTROLLER DESIGN VIA ALGEBRAIC MATCHING	29
4.1 Control over Quasi-LPV Models	29
4.2 Gain Scheduling via Polynomial Matching	30
4.3 Implementation Details	31
5. PERFORMANCE ANALYSIS	35
5.1 Experimental Setup and Datasets	35
5.2 Trajectory Reconstruction Results	38
5.3 Surrogate Model Open-Loop Prediction Results	39
5.4 Closed-Loop Control Performance	44
6. CONCLUSION AND FUTURE WORK	47
REFERENCES	51
CURRICULUM VITAE	53



ABBREVIATIONS

DL	: Deep Learning
FiLM	: Feature-wise Linear Modulation
GRU	: Gated Recurrent Unit
IPINN	: Inverse Physics-Informed Neural Network
LPV	: Linear Parameter-Varying
MIMO	: Multi-Input Multi-Output
MSE	: Mean Square Error
NODE	: Neural Ordinary Differential Equation
NRMSE	: Normalized Root Mean Square Error
NN	: Neural Network
ODE	: Ordinary Differential Equation
PINN	: Physics-Informed Neural Network
Q-LPV	: Quasi-Linear Parameter-Varying
RFF	: Random Fourier Features
RK4	: Fourth-Order Runge–Kutta
RMSE	: Root Mean Square Error
RMS	: Root Mean Square
SysID	: System Identification



SYMBOLS

t	: Continuous time variable
k	: Discrete-time index
Δt	: Sampling time / time step size
$x(t)$	: System state vector at time t
$\hat{x}(t)$	: Estimated / reconstructed state vector at time t
x_m	: Measured state / measurement vector
x_0	: Initial state of the system
$y(t)$	: System output at time t
$\hat{y}(t)$	: Estimated / reconstructed output at time t
$u(t)$	: System input / control input at time t
$v(t)$	: Velocity state component
T_m	: Measurement time set / measurement grid
T_c	: Collocation time set / collocation grid
$\theta(x)$	: State-dependent surrogate coefficient field
$\hat{\theta}(x)$	: Estimated surrogate coefficient field
θ^*	: Ridge-regression-based local coefficient hint
$k(x)$	: Effective stiffness coefficient
$d(x)$	: Effective damping coefficient
$g(x)$	: Effective input gain coefficient
$\hat{k}(x)$	: Estimated effective stiffness coefficient
$\hat{d}(x)$	: Estimated effective damping coefficient
$\hat{g}(x)$	: Estimated effective input gain coefficient
N_{sol}	: Solution Network
N_{param}	: Parameter Network
$\mathcal{L}_{\text{data}}$	: Data fitting / measurement reconstruction loss
$\mathcal{L}_{\text{phys}}$	: Physics residual loss
$\mathcal{L}_{\text{hint}}$	: Coefficient hint loss

$\mathcal{L}_{\text{TV}}$	: Total variation regularization loss
$\mathcal{L}_{\text{step}}$	: Step consistency / local smoothness loss
$\mathcal{L}$	: Total training loss
$r(t)$	: Physics residual at time t
Φ	: Regression matrix used in local ridge estimation
Y	: Regression target vector / observation vector
λ_r	: Ridge regularization coefficient
W, b	: Neural network weight matrices and bias vectors
$\sigma(\cdot)$	: Activation function
Θ	: Set of learnable neural network parameters
N	: Number of samples / collocation points in a trajectory segment
B	: Batch size
E	: Number of training epochs
η	: Learning rate
$A(\theta), B(\theta)$	: Local state-space matrices induced by surrogate coefficients
$K(x)$	: State-dependent feedback gain
$u_c(t)$	: Control law / controller output
$x_r(t)$	: Reference state trajectory
$e(t)$	: Tracking error

LIST OF TABLES

	<u>Page</u>
Table 3.1: SOLIS training procedure.	25
Table 4.1: Online closed-loop control procedure using the learned SOLIS surrogate.....	33
Table 5.1: Solution learning accuracy ($\uparrow$) on training trajectories.....	39
Table 5.2: Phase portrait similarity ($\uparrow$) measured via average cosine similarity to the ground-truth field.	42
Table 5.3: Identification rollout accuracy ($\uparrow$) on test trajectories.....	42
Table 5.4: Controller parameters used in the closed-loop simulations.	44
Table 5.5: Closed-loop control performance on the Duffing oscillator.....	46



LIST OF FIGURES

	<u>Page</u>
Figure 2.1: Schematic illustration of Physics Informed Neural Networks.	14
Figure 3.1: Overview of the proposed SOLIS dual-network architecture, showing the coupled trajectory reconstruction and local surrogate coefficient learning process.....	20
Figure 5.1: Representative trajectory reconstruction results for the benchmark systems: (a) Van der Pol oscillator, (b) Duffing oscillator, and (c) two-tank system.....	40
Figure 5.2: Phase portrait comparison results for the learned surrogate models. The top row corresponds to the Van der Pol oscillator and the bottom row corresponds to the Duffing oscillator. (a) Van der Pol surrogate phase portrait, (b) Van der Pol ground-truth phase portrait, (c) Van der Pol cosine similarity map, (d) Duffing surrogate phase portrait, (e) Duffing ground-truth phase portrait, and (f) Duffing cosine similarity map.....	41
Figure 5.3: Representative open-loop rollout results for the benchmark systems: (a) Van der Pol oscillator, (b) Duffing oscillator, and (c) two-tank system.....	43
Figure 5.4: Closed-loop tracking performance on the Duffing oscillator under the gain-scheduled controller.....	45



PHYSICS-INFORMED LEARNING OF INTERPRETABLE SURROGATES FOR MODELING AND CONTROL

SUMMARY

This thesis presents a physics-informed framework for learning interpretable surrogate models of nonlinear dynamical systems from sparse trajectory data and for using these learned models in controller synthesis. The main motivation of the study is the long-standing trade-off between flexibility and interpretability in nonlinear system identification. Classical identification methods provide structured models that are suitable for analysis and control design, but they may become restrictive when the system exhibits strong nonlinearities or state-dependent effects. In contrast, neural network based models can represent complex dynamics with high flexibility, yet they often lack a transparent structure that can be directly used for physical interpretation or controller design.

To address this issue, the thesis proposes a structured learning framework based on a second-order Quasi-LPV surrogate hypothesis. Instead of assuming a fixed global parametric model, the unknown nonlinear system is locally approximated by a second-order surrogate whose coefficients vary with the state. These coefficients correspond to effective stiffness, damping, and input gain, which preserve the semantics of classical second-order systems while allowing nonlinear behavior to be represented through state dependence. In this way, the identification problem is reformulated as the learning of a state-conditioned coefficient field rather than the estimation of a single global parameter vector.

The proposed method employs a dual-network architecture. A Solution Network is used to reconstruct smooth continuous-time trajectories from sparse measurements, while a Parameter Network predicts the local surrogate coefficients along the reconstructed trajectory. These two components are coupled through a physics-informed residual derived from the surrogate dynamics. In order to reduce the optimization difficulties commonly observed in inverse physics-informed learning, a curriculum-based training strategy is adopted. The training alternates between trajectory reconstruction and parameter identification stages. In the parameter learning stage, local ridge-regression based physics hints are used as weak supervision in order to guide the optimization and reduce convergence to trivial or physically inconsistent solutions.

The framework is evaluated on three benchmark systems with different nonlinear characteristics: the Van der Pol oscillator, the Duffing oscillator, and a real two-tank process. The performance analysis is carried out from two main identification perspectives. First, the trajectory reconstruction ability of the framework is examined in order to assess how accurately the measured system responses can be recovered

under the second-order surrogate assumption. Second, the quality of the learned surrogate model is evaluated through open-loop prediction and phase portrait analysis, in order to determine how well the local coefficient field captures the underlying nonlinear dynamics.

The results show that the proposed framework is able to reconstruct trajectories accurately and to produce coherent local surrogate models that remain physically interpretable. Compared with baseline inverse PINN formulations based on fixed global parameters, the proposed state-dependent surrogate approach provides better consistency in representing nonlinear dynamics and better open-loop predictive behavior on the considered benchmark systems. These findings support the main premise of the thesis, namely that nonlinear system identification can be formulated in a way that preserves both data-driven flexibility and control-oriented structure.

In addition to identification, the thesis also considers the use of the learned surrogate model in controller design. After the offline learning stage, the identified local second-order model is used to construct a gain-scheduled state-feedback controller through algebraic polynomial matching. By using the learned local stiffness, damping, and input gain, the controller gains are updated according to the current operating point. This provides a simple and interpretable way to connect the learned surrogate model with real-time control. The closed-loop control study is conducted on the Duffing oscillator as a case study. The results indicate that the learned surrogate can support scheduled controller design and can be used not only for analysis and prediction, but also for control-oriented applications.

Overall, the thesis proposes a unified perspective that connects nonlinear system identification, interpretable surrogate modeling, and gain-scheduled control. Its main contribution is to show that physics-informed learning can be used not only to fit trajectories, but also to recover local dynamical structure that is meaningful from a control point of view. The presented framework provides a basis for future studies on higher-dimensional systems, uncertainty-aware surrogate modeling, and more advanced control strategies built on learned local models.

MODELLEME VE KONTROL İÇİN YORUMLANABİLİR VEKİL DİNAMİKLERİN FİZİK BİLGİLİ ÖĞRENİMİ

ÖZET

Sistem tanıma, ölçülen deneysel veriler ile analiz edilebilen, simüle edilebilen ve kontrolör tasarımı için kullanılabilen matematiksel modeller arasındaki temel köprüyü oluşturmaktadır. Pek çok mühendislik uygulamasında bu görevin amacı yalnızca doğru tahminler elde etmekle sınırlı kalmayıp; tanımlanan modelin aynı zamanda fiziksel yorumlanabilirliği, kararlılık analizini ve kontrolör sentezini desteklemesi beklenmektedir. Ancak bu gereksinim, doğrusal olmayan sistemlerin modellenmesinde zorlu bir ödünleşim yaratmaktadır. Klasik sistem tanıma yöntemleri analiz ve kontrol tasarımına uygun yapısal modeller (durum uzayı, transfer fonksiyonu vb.) sunsa da, sistem güçlü doğrusal olmayan dinamikler veya duruma bağlı etkiler sergilediğinde bu yapılar fazlasıyla kısıtlayıcı hale gelebilmektedir. Buna karşın, modern yapay sinir ağı tabanlı modeller karmaşık dinamikleri yüksek bir esneklikle temsil edebilmelerine rağmen, genellikle sistemin içsel durumlarını doğrudan fiziksel analize veya kontrolör tasarımına uygun şeffaf bir yapıda sunmazlar. Bu durum, yalnızca veri uydurma yeteneği yüksek olan "kara kutu" (black-box) modeller ile kontrol mühendisliğinin ihtiyaç duyduğu analitik yapılar arasında büyük bir boşluk yaratmaktadır.

Bu tez, doğrusal olmayan dinamik sistemlerin seyrek deneysel veri kümelerinden yorumlanabilir vekil dinamik modellerini öğrenmek ve öğrenilen bu modelleri kontrolör tasarımı için doğrudan kullanmak amacıyla fizik bilgisi (physics-informed) bütünlük bir çerçeve sunmaktadır. Öncelikli hedef, bilinmeyen doğrusal olmayan dinamiklerin tam sembolik diferansiyel denklemlerini bulmak değil, deneysel verilerle tutarlı olan ve aynı zamanda klasik kontrol stratejileriyle uyumlu, fiziksel olarak yorumlanabilir bir model elde etmektir. Bu sorunu ele almak amacıyla tez, "İkinci Dereceden Yarı Doğrusal Parametresi Değişen (Q-LPV)" vekil model varsayımına dayanan SOLIS (Second-Order Local Identification of Surrogates) adında yapısal bir öğrenme çerçevesi önermektedir.

Tezin metodolojisi, doğrusal olmayan sistemlerin zaman içindeki yerel geometrisinin, duruma bağlı olarak değişen ikinci dereceden bir yapı ile yakınsanabileceği fikrine dayanmaktadır. Sabit ve global bir parametrik model varsaymak yerine, bilinmeyen sistem; katsayıları durum değişkenlerine bağlı olarak sürekli değişen ikinci dereceden bir vekil model ile ifade edilir. Öğrenilen bu katsayılar etkin direngenlik (yay katsayısı, stiffness), etkin sönüm (damping) ve etkin giriş kazancına (input gain) karşılık gelmektedir. Bu yaklaşım, klasik ikinci dereceden sistemlerin fiziksel anlamını harfiyen korurken, doğrusal olmayan karmaşık davranışların duruma bağlı katsayı alanları üzerinden ifade edilmesine olanak tanır. Böylece sistem tanıma problemi, optimizasyonu zor tek bir genel parametre vektörünün kestirimi olmaktan çıkıp, tüm

alışma uzayına yayılan, duruma baėlı bir katsayı alanının renilmesi problemi olarak yeniden formle edilmektedir.

Geleneksel Ters Fizik Bilgili Sinir Aėları (IPINN), modeli ve parametreleri tek bir kayıp fonksiyonu zerinden eřzamanlı olarak eėitmeye alıřtıėı iin oėunlukla zayıf kořullandırılmıř optimizasyon sorunlarıyla, eliřen gradyanlarla ve fiziksel olarak anlamsız yerel minimumlara yakınsama problemleriyle karřılařmaktadır. nerilen SOLIS mimarisi, bu sorunları ařmak iin ikili bir sinir aėı mimarisi ve ařamalı, mfredat tabanlı bir eėitim stratejisi kullanmaktadır. Mimari iki ana bileřenden oluřmaktadır. İlk bileřen olan zm Aėı (Solution Network), seyrek ve dzensiz lmlerden srekli zamanlı, dzgn sistem yanıtlarını geri kazanmakla grevlidir. Farklı bařlangı kořulları ve giriř sinyallerini birleřtirebilmek iin Geitli Yinelemeli Birim (Gated Recurrent Unit, GRU) tabanlı bir baėlam kodlayıcısı kullanır ve bu bilgiyi aėın i katmanlarına zellik Bazlı Doėrusal Modlasyon (Feature-wise Linear Modulation, FiLM) ile aktarır. İkinci bileřen olan Parametre Aėı (Parameter Network) ise, zm Aėı tarafından kestirilen bu durum evrimleri boyunca, sisteme ait yerel vekil model katsayılarını (direngenlik, snm, kazanç) hesaplamaktadır.

Bu iki bileřen, vekil model dinamiklerinden tretilen fiziėe dayalı bir hata terimi zerinden ortak bir yapıda birbirine entegre edilmiřtir. Ancak, diferansiyel kısıtların hesaplanması Parametre Aėının reteceėi katsayılara, katsayıların doėruluėu ise zm Aėının reteceėi yrngeye baėlıdır. Her iki aėın en bařından itibaren birlikte eėitilmesi, bu karřılıklı baėımlılık nedeniyle optimizasyon srecini ıkmaza sokabilmektedir. Mimari, bu kısır dngy kırmak amacıyla mfredat tabanlı bir ėrenme stratejisi benimser. Eėitim sreci, sistem yanıtını oluřturma ve parametre tanıma ařamaları arasında dnřml olarak ilerler. İlk ařamada, Parametre Aėı dondurularak yalnızca sistemin zaman blgesindeki yanıtı olan diferansiyel denklemin zmnn ėrenilmesine odaklanılır. Bu sayede zm Aėı, ncelikle eldeki lmlerden yola ıkarak sistemin geometrik ve zamansal yapısını doėru bir biimde oturtur. Ardından gelen parametre ėrenme ařamasında ise zm Aėı dondurulur ve elde edilen bu sistem yanıtını fiziksel olarak saėlayacak tutarlı katsayı alanı ėrenilir. Bu dnřml yapı, doėrudan yrnge uydurma ile parametre kestirimi arasındaki rekabeti ve gradyan akıřmalarını engelleyerek eėitim kararlılıėını artırır.

Parametre ėrenimi ařamasında karřılařılan bir diėer byk zorluk, fiziėe dayalı kayıp fonksiyonunun doėası gereėi dıřbkey olmaması (non-convex) ve aėın fiziksel aıdan tutarsız, ařık zmlere ok kolay yakınsayabilmesidir. rneėin, sinir aėı sadece diferansiyel hatayı sıfırlamak adına katsayıları tamamen sıfıra eřitleme eėilimi gsterebilir. Aėın optimizasyonuna rehberlik etmek ve bu riskleri en aza indirmek amacıyla algoritmaya yerel "fiziksel ipuları" entegre edilmiřtir. Bu ipuları, kestirilen sistem yanıtı zerinde hareket eden kayan pencereler kullanılarak hesaplanan Ridge regresyonu tabanlı analitik tahminlerdir. Doėrudan yrnge verisinden ekilen bu yerel regresyon tahminleri, aėın parametre uzayında kaybolmasını nleyen gvenilir bir apa grevi grr. Elde edilen bu ipuları, aėa katı bir kural olarak dayatılmak yerine "zayıf gzetim" olarak modele eklenir. Bylelikle Parametre Aėı, bir yandan verideki yerel analitik bilgiden ynlendirici bir destek alırken, diėer yandan doėrusal olmayan kayıp fonksiyonu zerinden global katsayı alanını ėrenme esnekliėini kısıtlanmadan srdrebilir.

Tezin bir diğer kritik adımı, öğrenilen bu fizik bilgili vekil modelin gerçek zamanlı kapalı çevrim kontrol uygulamalarına entegre edilmesidir. Sistem tanıma çevrimdışı olarak tamamlandıktan sonra, ağlar tarafından kestirilen yerel dinamik katsayılar, cebirsel polinom eşleştirme yoluyla kazanç programlamalı bir durum geri beslemeli kontrolör tasarlamak için kullanılır. Tasarım isterleri doğrultusunda belirlenen sönüm oranı ve doğal frekansa ile belirlenen referans karakteristik polinom, öğrenilen yerel modelin kapalı çevrim polinomuyla eşleştirilerek kontrolörün oransal ve türevsel kazançları doğrudan hesaplanır. Bu durum, modeldeki değişimlere anlık olarak uyum sağlayabilen, pratik kısıtlamaları, doyum ve değişim hızı limitleri, gözetim ve makine öğrenmesi modeliyle gerçek zamanlı kontrolü ilişkilendiren son derece şeffaf ve yorumlanabilir bir yapı sunmaktadır.

Önerilen çerçeve, farklı doğrusal olmayan karakteristiklere sahip üç referans sistem üzerinde değerlendirilmiştir: Duruma bağlı sönüm sergileyen Van der Pol osilatörü, kübik ve doğrusal olmayan direngenlik katsayısına sahip Duffing osilatörü ve ölçüm gürültüleri ile modellenmemiş karmaşık dinamikler içeren gerçek bir Çift Sıvı Tankı süreci. Performans analizleri, çerçevenin hem dinamik yanıt oluşturma yeteneği hem de öğrenilen dinamiklerin fiziksel tutarlılığı üzerinden yapılmıştır. İkinci değerlendirme aşamasında, öğrenilen yerel katsayı modelinin doğrusal olmayan yapıları ne kadar iyi yakaladığını kanıtlamak amacıyla açık çevrim öngörü testleri ve faz portresi analizleri gerçekleştirilmiş, kosinüs benzerliği haritaları çıkartılmıştır.

Elde edilen sonuçlar, SOLIS çerçevenin seyrek ve gürültülü verilerden durum eğrilerini mükemmel bir doğrulukla geri kazanabildiğini ve aynı zamanda fiziksel yorumlanabilirliğini yitirmeyen tutarlı yerel vekil dinamikler üretebildiğini göstermektedir. Sistemin tüm çalışma uzayını sabit bir parametre vektörüyle modellemeye çalışan standart Ters Fizik Bilgili Sinir Ağı formülasyonlarıyla karşılaştırıldığında, önerilen duruma bağlı vekil model yaklaşımının yapısal üstünlüğü açıkça ortaya çıkmıştır. Sabit parametrelili temel modeller, sistemin farklı çalışma bölgelerindeki doğrusal olmayan geçişleri ve yerel değişimleri yakalamakta yetersiz kalırken; önerilen SOLIS mimarisi ele alınan üç referans sistemde de faz portrelerinin aslına uygun şekilde oluşturulmasında ve altta yatan vektör alanlarının geometrisinin çıkarılmasında en yüksek tutarlılığı sergilemiştir. Ayrıca, modelin eğitim verisi dışında kalan zaman dilimlerindeki davranışını test eden ileri yönlü açık çevrim öngörü deneylerinde elde edilen yüksek doğruluk oranları, ağın yalnızca eldeki ölçümleri ezberlemediğini, aksine sistemi yöneten gerçek doğrusal olmayan fiziksel mekanizmaları başarıyla kavradığını doğrulamaktadır.

Kapalı çevrim kontrol deneyleri ise, özellikle durum değişkenlerine bağlı kübik geri çağırıcı kuvvetler barındıran ve bu yönüyle klasik doğrusal kontrolcülerini zorlayan Duffing osilatörü üzerinde bir vaka analizi olarak gerçekleştirilmiştir. Sistem tanıma aşamasında çevrimdışı olarak öğrenilen yerel katsayılar (direngenlik, sönüm ve kazanç), cebirsel polinom eşleştirme yöntemiyle gerçek zamanlı bir kontrolcüye entegre edilmiştir. Tasarlanan bu kazanç programlamalı durum geri beslemeli kontrolör; ani geçişleri ve aşımı ölçen basamak, sürekli osilasyon gerektiren sinüzoidal ve değişken frekanslı referans sinyallerini oldukça düşük izleme hataları ve optimize edilmiş bir kontrol eforu ile takip edebilmiştir. Kontrolcünün, sistemin o anki çalışma durumuna göre oransal ve türevsel kazançlarını anlık olarak güncelleyebilmesi,

doğrusal olmayan etkilere karşı güvenli bir adaptasyon sağlamıştır. Bu başarılı kapalı çevrim takip performansı, veri güdümlü sistem tanıma sürecinde sinir ağı tarafından kestirilen katsayıların yalnızca matematiksel birer eğri uydurma parametresi veya rastgele sayılar olmadığını; aksine, doğrudan modern kontrol sistemlerinin analizinde ve sentezinde kullanılabilecek kesinliğe ve fiziksel anlama sahip dinamik bileşenler olduğunu net bir biçimde kanıtlamaktadır.

Sonuç olarak tez, doğrusal olmayan karmaşık sistem tanıma, yorumlanabilir vekil modelleme ve kazanç programlamalı durum geri beslemeli kontrolü birbirine bağlayan bütünleşik bir perspektif sunmaktadır. Araştırmanın literatüre en temel katkısı, derin öğrenme temelli ve fiziğe dayalı yöntemlerin yalnızca veriye eğri uydurmak için değil, kontrol perspektifinden doğrudan faydalanılabilecek yerel dinamik yapıları geri kazanmak için de güvenle kullanılabileceğini göstermesidir. Sunulan çerçeve; çok girişli-çok çıkışlı (MIMO) sistemlere genişletilme, model öğrenimindeki belirsizliklerin Bayesçi veya aralık temelli ağlarla belirsizlik nicelendirilmesi ve öğrenilen yorumlanabilir bu vekil modellerin Model Öngörülü Kontrol (MPC) gibi daha gelişmiş algoritmalara entegre edilmesi yönündeki gelecekteki çalışmalar için son derece sağlam bir temel oluşturmaktadır.

1. INTRODUCTION

System identification provides the link between measured data and mathematical models that can be analyzed, simulated, and used for controller design [1]. In many engineering applications, this task is no longer limited to obtaining accurate predictions. The identified model is also expected to support interpretation, stability analysis, and control synthesis. This requirement creates a difficulty in nonlinear settings. Models with strong physical structure are often suitable for control design, yet they may be too restrictive to represent complex nonlinear behavior. In contrast, highly expressive learning-based models can fit complicated dynamics well, but they usually do not provide a form that is easy to interpret or use in a control-oriented setting.

This thesis studies this gap between flexibility and interpretability. It focuses on nonlinear systems for which the exact governing equations are not known in advance and only sparse trajectory data are available. The main objective is to develop a physics-informed learning framework that recovers an interpretable surrogate model from data and then uses this model for direct controller synthesis. The proposed framework, called SOLIS, learns a state-conditioned second-order surrogate representation whose coefficients retain a clear physical meaning and can be incorporated into a gain-scheduled control law.

1.1 Problem Definition and Motivation

A central difficulty in nonlinear system identification is the trade-off between model structure and model expressiveness. Classical identification methods usually employ transfer function, state-space, or grey-box representations with a fixed parametric form [1]. These approaches are valuable because they produce compact and interpretable models. Parameters such as stiffness, damping, gain, poles, and zeros can often be related to physical behavior and used directly in analysis and control design. However, this advantage comes at a cost. When the true dynamics contain strong nonlinearities,

state-dependent effects, or unmodeled phenomena, a rigid parametric structure may fail to represent the system with sufficient accuracy.

Recent advances in deep learning have offered a different direction. Neural models can approximate highly nonlinear input-output relations and dynamical behaviors with considerable flexibility [2]. Recurrent architectures and continuous-time neural models have shown strong prediction capability in system identification tasks. In particular, Neural Ordinary Differential Equations (NODEs) provide a natural way to learn continuous-time dynamics from irregularly sampled data [3]–[5]. Yet this flexibility introduces another problem. The learned dynamics are typically represented through latent states or generic neural vector fields, and such representations do not readily reveal physical quantities or control-relevant structure. As a result, there is often no direct path from accurate identification to transparent analysis or controller synthesis. Related recent efforts have sought to preserve desirable structural properties within learned models, for example through stable-by-design neural LPV state-space models [6], but the general tension between flexibility and interpretability remains open.

Physics-Informed Neural Networks (PINNs) offer a partial remedy by embedding differential equation constraints into the learning objective [7,8]. In inverse settings, PINNs can reconstruct trajectories and estimate unknown parameters from sparse measurements without relying on explicit numerical integration. This idea is attractive for system identification, especially when data are limited. However, standard inverse PINN formulations remain inadequate for many nonlinear control problems.

First, they usually assume that the mathematical form of the governing dynamics is already known and that only a small set of constant parameters must be estimated. This assumption is restrictive when the actual system is only partially known or when its effective parameters vary across the state space. Second, the simultaneous estimation of trajectories and parameters produces a difficult optimization problem. The data loss and the physics loss may induce conflicting gradients, and training can become highly sensitive to weighting choices and initialization [9]. Third, identifiability becomes problematic when sparse data are used to infer both latent

states and model parameters, particularly when the parameterization is not well aligned with the actual system behavior [10]. In addition, recent studies have highlighted the need to incorporate uncertainty awareness into learned dynamical models [11,12], which further emphasizes that accurate prediction alone is not sufficient for reliable control-oriented modeling.

These limitations motivate the main direction of this thesis. The aim is not to recover the exact nonlinear governing equation in a fully symbolic sense. Instead, the goal is to identify a surrogate model that is sufficiently expressive to capture nonlinear behavior and sufficiently structured to preserve physical interpretation. Such a model should also support direct controller synthesis. This motivates the search for a state-dependent representation that behaves locally like a second-order dynamical system and remains compatible with gain-scheduling ideas from control theory.

1.2 Literature Review

The development of system identification methods reflects a long-standing effort to balance prediction accuracy with structural transparency [2]. Classical methods remain the foundation of the field and include transfer function models, polynomial models, and state-space formulations [1]. These methods are supported by a mature theory and are tightly connected to control design. Their main limitation is that model quality depends strongly on the selected structure. If the assumed form is too simple, the identified model may not capture the essential nonlinear behavior of the system.

To address this limitation, neural network based identification methods have been studied in many forms. Feedforward networks have been used for parametric identification, while convolutional and recurrent architectures have been applied to nonlinear dynamic estimation and sequence modeling [13]–[17]. These approaches can achieve high approximation power, but they often embed the dynamics in a form that is not directly interpretable. The internal states or weights of the model do not usually correspond to physical quantities that can be used in analysis or controller design. More recently, structured neural identification approaches have also been considered, including neural LPV state-space models designed to preserve stability by construction [6].

Continuous-time neural modeling has further extended this line of work. NODE-based models learn a differential representation of the dynamics and are particularly attractive for irregularly sampled trajectories and continuous-time prediction [3]–[5,18]. This is an important step because many physical systems are naturally described in continuous time. Nevertheless, a generic learned vector field still does not automatically provide a structured representation suitable for extracting damping ratios, natural frequencies, or state-dependent gains.

PINNs were introduced as a way to combine learning from data with prior knowledge expressed through differential equations [7]. By penalizing violations of governing equations at collocation points, they incorporate physical consistency into the training objective. This principle has been extended to inverse problems, where unknown coefficients are learned together with the solution itself [8]. For identification problems with sparse measurements, this can be advantageous because the differential constraints act as a regularizing mechanism.

Despite this promise, several studies have shown that PINN training can be unstable and difficult to tune. Optimization pathologies, imbalance between competing loss terms, and spectral bias may prevent convergence to physically meaningful solutions [9]. In addition, identifiability issues arise when the available data do not sufficiently constrain the unknown parameters or when the assumed parameterization is incompatible with the true dynamics [10]. These difficulties become more pronounced in inverse settings where both the trajectory and the parameters must be learned simultaneously.

A related line of work seeks to retain physical structure while learning unknown parts of the model. Hybrid and hidden-physics approaches aim to combine differential constraints with flexible approximators for the unmodeled components. While such methods improve expressiveness, they do not necessarily provide a control-oriented representation of the learned dynamics. The resulting model may still be difficult to interpret globally or to connect with classical controller synthesis procedures. Other recent studies have also considered uncertainty-aware modeling perspectives, including interval neural networks for system identification [11] and fuzzy ODE-based

formulations for dynamic modeling and uncertainty quantification [12]. These studies show that interpretability is not the only structural requirement in modern identification problems. Reliability and uncertainty representation are also becoming increasingly important.

From the perspective of control, LPV and Quasi-LPV representations offer an important conceptual framework [19]. In these models, nonlinear behavior is described through linear dynamics whose coefficients depend on scheduling variables or system states. This preserves much of the analytical convenience of linear systems while extending the modeling capability to nonlinear regimes. However, obtaining a useful LPV or Quasi-LPV representation from first principles is often difficult, especially for systems with unknown or complicated nonlinearities. This suggests that machine learning can play a constructive role if it is used not to produce an arbitrary black-box model, but to discover a structured, state-dependent surrogate representation from data.

1.3 Purpose of Thesis

The purpose of this thesis is to develop a physics-informed identification framework for nonlinear systems that yields an interpretable surrogate model and supports direct controller design. The proposed method, SOLIS (Second-Order Local Identification of Surrogates), is built on the idea that the local temporal behavior of a smooth nonlinear system can be approximated by a second-order surrogate with state-dependent coefficients. Instead of identifying a single global parameter vector or an unconstrained neural vector field, the method learns a Quasi-LPV representation whose effective stiffness, damping, and input gain vary with the state.

This viewpoint serves two purposes. First, it allows nonlinear dynamics to be represented in a form that preserves classical second-order meaning. Second, it creates a direct connection between identification and control, since the learned local model can be used for scheduled state-feedback design.

The main contributions of the thesis can be summarized as follows.

1. A physics-informed formulation for nonlinear system identification based on a state-conditioned second-order surrogate model. This formulation interprets the

identification task as learning a Quasi-LPV coefficient field rather than estimating constant global parameters.

2. A dual-network architecture that separates trajectory reconstruction from parameter identification. A solution network learns a smooth continuous-time approximation of the measured trajectory, while a parameter network predicts local surrogate coefficients along that trajectory.
3. A curriculum-based training strategy designed to reduce the optimization difficulties of inverse PINN training. In this strategy, local ridge-regression estimates are used as weak physical hints to guide the parameter network during training and to reduce convergence to trivial solutions.
4. A gain-scheduled controller design procedure based on the learned surrogate model. By using the estimated local coefficients in a second-order polynomial matching framework, scheduled feedback gains can be computed in a form that is simple, interpretable, and suitable for real-time implementation.

The thesis therefore addresses both identification and control. Its main claim is that a properly structured physics-informed model can provide more than accurate prediction. It can also produce a representation of nonlinear dynamics that is transparent enough to support controller synthesis.

1.4 Thesis Structure

The remainder of the thesis is organized as follows.

Chapter 2 presents the theoretical background required for the study. It reviews the fundamentals of system identification, introduces LPV and Quasi-LPV representations, summarizes the main ideas behind Physics-Informed Neural Networks, and outlines the algebraic pole assignment concepts used later in the controller design stage.

Chapter 3 presents the proposed SOLIS framework. It introduces the second-order surrogate hypothesis, describes the solution and parameter networks, and explains the training strategy based on curriculum learning and local physics hints.

Chapter 4 develops the controller design method built on the learned surrogate model. The chapter shows how the local coefficients produced by the parameter network can be used to construct a scheduled state-feedback controller through algebraic polynomial matching.

Chapter 5 provides the experimental evaluation. The identification and prediction performance of the proposed approach are examined on simulated nonlinear benchmarks and on a real laboratory dataset. The chapter also presents the closed-loop results obtained with the scheduled controller.

Chapter 6 concludes the thesis by summarizing the main findings, discussing the scope and limitations of the proposed approach, and outlining possible directions for future work.



2. PRELIMINARIES

2.1 System Identification

System identification is concerned with constructing mathematical models of dynamical systems from observed input-output data [1]. Its main purpose is to obtain a representation that explains the measured behavior of a system and can be used for prediction, analysis, and control design. In engineering practice, the identified model is rarely an end in itself. Its value depends on whether it provides a reliable basis for simulation and whether it captures the dynamical properties that are relevant to the intended application.

A general continuous-time nonlinear system may be written as

$$\dot{x}(t) = f(x(t), u(t)), \quad y(t) = h(x(t), u(t)), \quad (2.1)$$

where $x(t) \in \mathbb{R}^n$ denotes the state, $u(t) \in \mathbb{R}^m$ denotes the input, and $y(t) \in \mathbb{R}^p$ denotes the measured output. The functions $f(\cdot)$ and $h(\cdot)$ are usually unknown or only partially known. The goal of system identification is to infer an approximation of these mappings, or an equivalent surrogate model, from measured trajectories.

Classical approaches to system identification typically rely on predetermined model classes such as transfer functions, autoregressive models, and state-space representations [1]. These methods are well established and closely connected to control theory. Their main strength is that the resulting models are structured and interpretable. Parameters often correspond to physically meaningful quantities, and the model form is usually suitable for stability analysis, pole placement, or observer design. However, this strength also imposes a limitation. If the chosen model structure is too restrictive, then complex nonlinear effects may be poorly represented.

In contrast, modern machine learning approaches place greater emphasis on expressive function approximation. Neural network based models can capture highly nonlinear

relationships and have shown strong predictive performance in system identification tasks [2,13,15,18]. This flexibility is attractive when the underlying physics is only partially known or when the system operates over a wide range of regimes. Nevertheless, predictive accuracy alone is not sufficient in many control-oriented applications. A black-box model may reproduce trajectories accurately while offering little insight into local stability, effective damping, or control authority.

This observation motivates an important distinction between *exact model recovery* and *useful surrogate identification*. In many practical settings, it is unrealistic to expect recovery of the true governing equations from sparse and noisy data. A more meaningful objective is to identify a surrogate model that is data-consistent, dynamically coherent, and structurally suitable for subsequent analysis and controller design. The framework developed in this thesis follows this second viewpoint. Rather than seeking a fully symbolic representation of the unknown nonlinear dynamics, it aims to learn a control-oriented surrogate that preserves interpretable second-order structure.

Another relevant distinction is between global and local modeling. A global model attempts to explain the entire operating range through a fixed representation, often with a single parameter vector. Such models are convenient when the system behaves approximately linearly or when nonlinear effects are weak. However, strongly nonlinear systems often exhibit different local behaviors in different regions of the state space. In those cases, a state-dependent local model can offer a more suitable compromise between interpretability and expressiveness. This idea leads naturally to LPV and Quasi-LPV formulations, which are discussed in the next section.

2.2 LPV and Quasi-LPV System Model

Linear Parameter-Varying (LPV) systems provide a structured framework for describing nonlinear or time-varying dynamics through linear models whose coefficients depend on a measurable scheduling variable [19]. This representation preserves much of the analytical convenience of linear systems while extending the modeling range to operating conditions where a single linear model is insufficient.

A continuous-time LPV system can be written in the state-space form

$$\dot{x}(t) = A(\theta(t))x(t) + B(\theta(t))u(t), \quad (2.2)$$

$$y(t) = C(\theta(t))x(t) + D(\theta(t))u(t), \quad (2.3)$$

where $\theta(t) \in \mathbb{R}^q$ is the scheduling variable. In standard LPV settings, $\theta(t)$ is assumed to be measurable in real time and external to the state-space description. The matrices $A(\theta)$, $B(\theta)$, $C(\theta)$, and $D(\theta)$ vary as functions of this variable.

The practical value of the LPV framework lies in the fact that many nonlinear control problems can be reformulated as control problems over a family of linear systems. This allows the use of gain scheduling and other linear design tools while still accounting for operating-point dependence. However, deriving an LPV description from first principles may be difficult, especially when the system is poorly understood or when the appropriate scheduling variable is not obvious.

A closely related concept is the Quasi-LPV representation. In Quasi-LPV systems, the scheduling variable is not an external measured signal but a function of the system state or output itself. Thus, the model retains an LPV-like form while embedding nonlinear dependence through endogenous variables. A general Quasi-LPV form may be written as

$$\dot{x}(t) = A(\theta(x(t)))x(t) + B(\theta(x(t)))u(t), \quad (2.4)$$

$$y(t) = C(\theta(x(t)))x(t) + D(\theta(x(t)))u(t). \quad (2.5)$$

This structure is especially useful when the local behavior of a nonlinear system can be approximated by a linear model whose coefficients vary smoothly across the state space.

The thesis proposed in this work is based on a second-order Quasi-LPV surrogate. For a scalar output $y(t)$ and its velocity $v(t) = \dot{y}(t)$, the local surrogate dynamics are assumed to take the form

$$\dot{y}(t) = v(t), \quad (2.6)$$

$$\dot{v}(t) = -k(x(t))y(t) - d(x(t))v(t) + g(x(t))u(t), \quad (2.7)$$

where $x(t) = [y(t), v(t)]^\top$ and the coefficient field

$$\theta(x) = [k(x) \quad d(x) \quad g(x)]^\top \quad (2.8)$$

depends on the state. In this representation, $k(x)$, $d(x)$, and $g(x)$ can be interpreted as effective stiffness, effective damping, and effective input gain, respectively. This is not assumed to be the true governing model of the system. Rather, it is a local surrogate that approximates the observed trajectory geometry in an interpretable way.

The corresponding local state-space form is

$$\dot{x}(t) = A(x(t))x(t) + B(x(t))u(t), \quad (2.9)$$

with

$$A(x) = \begin{bmatrix} 0 & 1 \\ -k(x) & -d(x) \end{bmatrix}, \quad B(x) = \begin{bmatrix} 0 \\ g(x) \end{bmatrix}. \quad (2.10)$$

This form is particularly attractive because it preserves the familiar semantics of classical second-order systems while allowing nonlinear behavior to be represented through state dependence. As a result, quantities such as local natural frequency, damping ratio, and input gain can be interpreted pointwise in the state space. This makes the representation suitable not only for identification, but also for gain-scheduled controller design.

The essential idea is therefore not to force the unknown nonlinear system into a single global linear model, nor to replace it with an unstructured black-box network. Instead, the aim is to learn a state-dependent local model whose varying coefficients capture nonlinear effects while retaining a form compatible with classical control concepts.

2.3 Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) are neural models trained by combining measurement fitting with constraints imposed by differential equations [7,8]. Their main idea is to incorporate physical knowledge directly into the learning objective. Instead of treating the problem as a purely data-driven regression task, PINNs enforce consistency with the governing dynamics at selected collocation points in the domain.

For an ordinary differential equation of the form

$$\dot{x}(t) = \hat{f}(x(t), u(t)), \quad (2.11)$$

a PINN approximates the unknown solution by a neural network

$$\hat{x}(t) = \mathcal{N}(t; W), \quad (2.12)$$

where W denotes the trainable parameters. Since $\hat{x}(t)$ is represented by a differentiable neural function, its time derivative can be computed through automatic differentiation. This makes it possible to evaluate the residual of the governing equation directly on the network output.

The training objective is typically written as a composite loss,

$$\mathcal{L}(W) = \mathcal{L}_{\text{data}}(W) + \lambda_p \mathcal{L}_{\text{phys}}(W), \quad (2.13)$$

where $\mathcal{L}_{\text{data}}$ penalizes mismatch with the available measurements and $\mathcal{L}_{\text{phys}}$ penalizes the residual of the differential equation at collocation points. In this way, the learned approximation is encouraged to satisfy both the observed data and the underlying physics.

Figure 2.1 illustrates this basic PINN framework. The top part of the figure shows the time domain together with two distinct sets of points: data points, where measurements are available, and collocation points, where the governing equation is enforced. The middle part represents the neural network surrogate that maps time to the approximate solution. The lower part shows how the total loss is formed by combining the data mismatch term with the physics residual term. This coupling between measurement information and differential consistency is the defining characteristic of PINNs.

In forward problems, the differential equation is assumed to be known and the task is to approximate its solution. This setting is especially useful when measurements are sparse, because the governing equation acts as a regularizer and restricts the solution space to functions that remain physically plausible. At the same time, PINNs avoid the need to discretize the differential equation in the traditional numerical sense, since the residual is computed directly through automatic differentiation on the neural representation.

2.4 Inverse Physics-Informed Neural Networks

Inverse Physics-Informed Neural Networks extend the PINN framework to settings where part of the physical model is unknown and must be identified from data. In this case, the objective is no longer limited to approximating the solution of a known

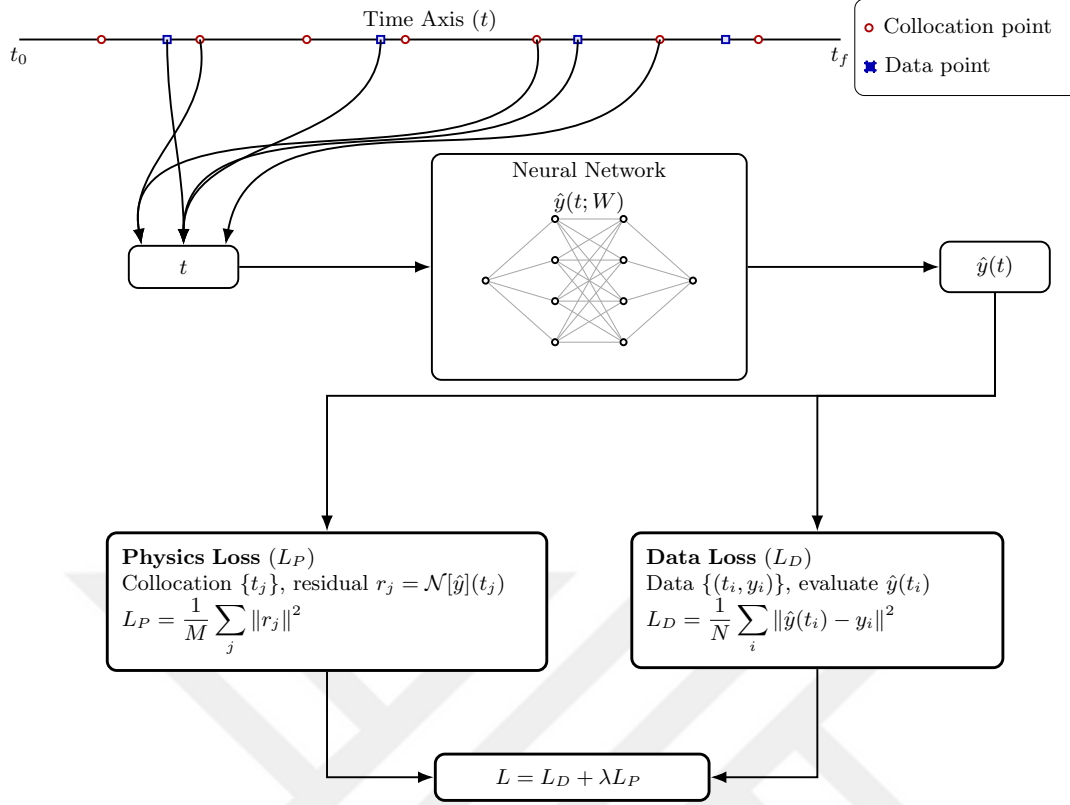


Figure 2.1: Schematic illustration of Physics Informed Neural Networks.

differential equation. Instead, the network must infer both the trajectory and the unknown parameters that govern the dynamics.

Suppose the system is described by

$$\dot{x}(t) = \hat{f}(x(t), u(t); \theta), \quad (2.14)$$

where θ denotes unknown physical parameters. An inverse PINN seeks to identify both the neural representation of the trajectory and the parameter vector θ by solving

$$\min_{W, \theta} \mathcal{L}_{\text{data}}(W) + \lambda_p \mathcal{L}_{\text{phys}}(W, \theta). \quad (2.15)$$

The data term still enforces agreement with the available measurements, while the physics term now depends on the unknown parameters as well. As a result, inverse PINNs provide a natural framework for system identification, especially when direct state measurements are incomplete or when numerical differentiation from noisy data would be unreliable.

This formulation is attractive because it allows trajectory reconstruction and parameter estimation to be handled within a single optimization problem. In principle, the network can recover latent continuous-time states and identify model coefficients simultaneously, even from sparse or irregularly sampled data. For this reason, inverse PINNs have become an appealing approach in nonlinear identification problems.

However, this joint formulation is also the source of major difficulties. The optimization problem is often ill-conditioned because the trajectory and the unknown parameters are tightly coupled. The data loss and the physics loss may induce conflicting gradients, and this can slow convergence or lead to poor local minima [9]. Moreover, inverse problems are especially sensitive to identifiability limitations. If the measured data do not sufficiently excite the dynamics, or if the assumed model structure is too restrictive, then multiple parameter values may explain the same observations with similar accuracy [10].

A further limitation is structural. Standard inverse PINNs typically assume that the mathematical form of the governing equation is already known and that only a small number of constant global parameters must be estimated. This assumption is too restrictive for many nonlinear systems encountered in practice. In such systems, the effective dynamics may vary across the state space, so representing them with a fixed parameter vector may be inadequate even if the optimization converges successfully.

This limitation is directly relevant to the present thesis. The proposed methodology does not use inverse PINNs in their standard constant-parameter form. Instead, it retains the physics-informed principle while replacing the fixed parametric model with a state-dependent second-order surrogate. In this way, the learning problem is reformulated as the identification of an interpretable coefficient field rather than the estimation of a single global parameter vector. The details of this formulation are presented in Chapter 3. At the preliminary level, the important point is that inverse PINNs provide the conceptual starting point of the proposed approach, while their structural and optimization limitations motivate the need for a more flexible and control-oriented framework.



3. PHYSICS-INFORMED LEARNING OF INTERPRETABLE SURROGATES

This chapter presents the proposed methodology for identifying interpretable surrogate models of nonlinear dynamical systems from sparse trajectory data. The central objective is not to recover the exact governing equations of the unknown system, but to construct a data-consistent and control-oriented surrogate representation that explains the observed trajectories while preserving physical meaning. To this end, the proposed framework, denoted by SOLIS, combines physics-informed learning with a state-dependent second-order surrogate structure. The resulting model is formulated as a Quasi-LPV representation whose local coefficients can later be used for controller synthesis. The methodology consists of three main components: the second-order surrogate hypothesis, a dual-network architecture for trajectory and parameter learning, and a curriculum-based training strategy.

3.1 The Second-Order Surrogate Hypothesis

Consider an unknown nonlinear dynamical system evolving according to

$$\dot{x}(t) = f(x(t), u(t)), \quad (3.1)$$

where $x(t) \in \mathbb{R}^n$ denotes the system state, $u(t)$ is a known exogenous input, and the functional form of $f(\cdot)$ is unknown. In this setting, neither the exact system order nor a global parametric structure is assumed a priori. The identification objective is therefore not exact model recovery, but the construction of a physically meaningful surrogate that is consistent with the observed trajectories and suitable for control-oriented analysis.

Suppose that measurements are available from J trajectories. For each trajectory, the available information consists of a known continuous-time input signal $u^{(j)}(t)$ over a finite horizon and a set of output measurements

$$\mathcal{D}_m^{(j)} = \left\{ \left(t_i^{(j)}, y_i^{(j)} \right) \right\}_{i=1}^{N_d^{(j)}}. \quad (3.2)$$

The complete measurement set is denoted by

$$\mathcal{D}_m = \bigcup_{j=1}^J \mathcal{D}_m^{(j)}. \quad (3.3)$$

In addition, for each trajectory, a denser set of collocation points

$$\mathcal{T}_c^{(j)} = \left\{ t_k^{(j)} \right\}_{k=1}^{N_c^{(j)}}, \quad N_c^{(j)} \gg N_d^{(j)}, \quad (3.4)$$

is used to enforce differential consistency through the physics-informed loss.

The main modeling assumption of this thesis is the following second-order surrogate hypothesis:

The local temporal geometry of smooth nonlinear system trajectories can be approximated by a second-order Quasi-LPV surrogate dynamics.

This hypothesis reflects the idea that, although the true nonlinear system may not admit a simple global representation, its local behavior along observed trajectories can be captured by a structured second-order model with state-dependent coefficients. Accordingly, the surrogate state is defined as

$$x(t) = \begin{bmatrix} y(t) \\ v(t) \end{bmatrix}, \quad v(t) = \dot{y}(t), \quad (3.5)$$

where $y(t)$ is the measured output and $v(t)$ is the associated velocity, which may be latent and therefore must be reconstructed. The surrogate dynamics are postulated in the affine form

$$\dot{y}(t) = v(t), \quad (3.6)$$

$$\dot{v}(t) = -k(x(t))y(t) - d(x(t))v(t) + g(x(t))u(t), \quad (3.7)$$

where the coefficient field

$$\theta(x) = \begin{bmatrix} k(x) \\ d(x) \\ g(x) \end{bmatrix} \quad (3.8)$$

depends on the state. In contrast to standard inverse PINN formulations, where the unknown parameters are assumed constant, the present formulation treats them as state-conditioned quantities. Thus, the identification problem becomes the learning

of a coefficient field that makes the reconstructed trajectories dynamically consistent under the surrogate model.

This representation has an important interpretability advantage. The coefficients $k(x)$, $d(x)$, and $g(x)$ admit the meanings of effective stiffness, effective damping, and effective input gain, respectively. Hence, the surrogate preserves the semantics of classical second-order systems while remaining expressive enough to represent nonlinear behavior through state dependence. The corresponding local quantities can be written as

$$\omega_n(x) = \sqrt{k(x)}, \quad \zeta(x) = \frac{d(x)}{2\sqrt{k(x)}}, \quad K(x) = \frac{g(x)}{k(x)}. \quad (3.9)$$

These state-dependent quantities provide a direct bridge from data-driven identification to control-oriented analysis.

3.2 Dual Network Architecture

The surrogate formulation above gives rise to an intrinsically coupled learning problem, since the state-dependent coefficient field $\theta(x)$ determines the differential constraints, while these constraints can only be evaluated along the system trajectory itself. At the same time, the underlying continuous-time trajectory is not directly available because the measurements are sparse and noisy. For this reason, trajectory reconstruction and parameter identification cannot be handled as separate tasks. Instead, the proposed SOLIS architecture decomposes the problem into two interacting neural components, as illustrated in Figure 3.1. The Solution Network reconstructs a smooth state trajectory from the available measurements, whereas the Parameter Network uses the reconstructed states to estimate the local surrogate coefficients along that trajectory. These two networks are coupled through the physics residual induced by the surrogate dynamics, forming a closed learning loop in which improved trajectory estimates support better coefficient prediction, and improved coefficients in turn enforce more consistent dynamics. This decomposition is intentionally asymmetric: the Solution Network is designed to capture trajectory-level temporal structure across multiple experiments, while the Parameter Network is kept local and

structurally simpler because its task is to infer pointwise surrogate coefficients rather than to represent the full temporal evolution.

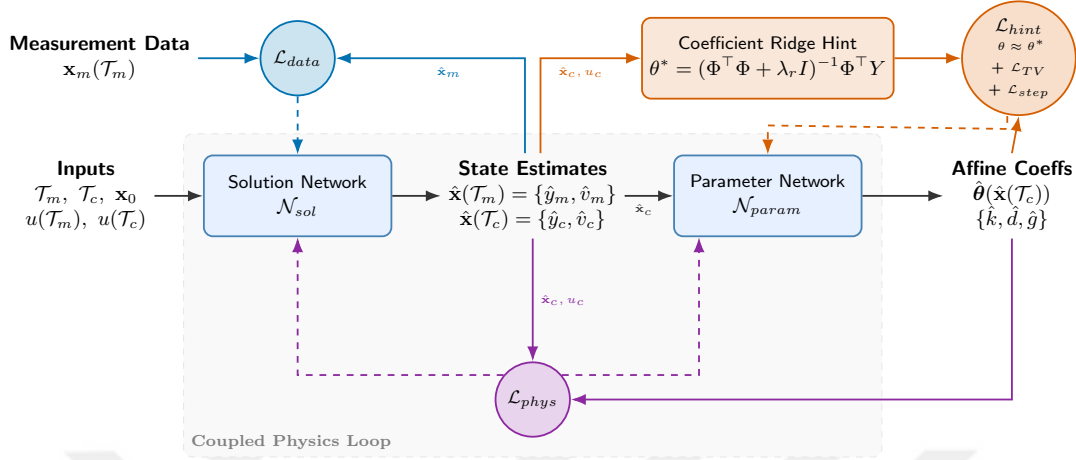


Figure 3.1: Overview of the proposed SOLIS dual-network architecture, showing the coupled trajectory reconstruction and local surrogate coefficient learning process.

3.2.1 Solution network

The solution network, denoted by $\mathcal{N}_{sol}$, approximates the continuous-time surrogate state

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} \hat{y}(t) \\ \hat{v}(t) \end{bmatrix}. \quad (3.10)$$

Its primary role is to reconstruct a smooth trajectory from sparse measurements. Unlike standard PINN settings that fit a single trajectory in isolation, the present framework aims to learn a family of trajectories corresponding to different initial conditions and input profiles using a shared model. For this reason, the solution network is conditioned on the input signal.

This architectural choice is motivated by the multi-trajectory nature of the problem. A single shared network must represent trajectories generated under different excitation profiles, initial conditions, and local operating regimes. If the network were driven only by time, it would have to explain all such variability through its weights alone, which would make generalization across trajectories more difficult. Conditioning the trajectory model on the input profile allows the temporal backbone to remain shared while still adapting its representation to the specific experiment being reconstructed.

In this sense, the Solution Network is not merely an interpolator in time, but a context-conditioned continuous-time map from experiment information and time to state.

For each trajectory, the sampled input sequence is first embedded into a compact context vector through a Gated Recurrent Unit (GRU) [20],

$$c = \text{GRU}(u_c; W_{\text{context}}), \quad (3.11)$$

where u_c denotes the input values evaluated over the collocation grid. The GRU is used only to summarize the input profile into a fixed-dimensional context representation. It is not used as a sequential state propagator, so the continuous-time character of the solution network is preserved.

The use of a GRU is motivated by the fact that the input is naturally given as a sequence over the observation horizon. The role of the GRU is not to propagate the physical state, but to compress the excitation history into a compact representation that captures ordering and temporal variation in the input. This makes it possible to summarize trajectory-specific forcing information without abandoning the continuous-time character of the surrogate state representation.

The context vector then modulates the internal features of the solution network through Feature-wise Linear Modulation (FiLM) [21]. For the l th hidden layer with pre-activation z_l , the modulated activation is written as

$$z'_l = \gamma_l(c) \odot z_l + \beta_l(c), \quad (3.12)$$

where $\gamma_l(\cdot)$ and $\beta_l(\cdot)$ are learned affine transformations of the context vector. This conditioning mechanism allows a single shared network to represent different trajectory geometries while maintaining a common temporal coordinate system.

FiLM conditioning is used because it modulates the hidden representation of the shared trajectory network in a simple and explicit manner. Compared with directly concatenating the context vector to the time input, this mechanism allows the same temporal backbone to produce different trajectory geometries through feature-wise adaptation at intermediate layers. This is particularly useful in the present setting,

where the time coordinate remains common across trajectories, but the effective shape of the trajectory depends on the excitation profile.

When the underlying dynamics contain high-frequency oscillatory components, the scalar time input may also be enriched using Random Fourier Features [22]. In that case, the encoded input to the network becomes

$$e(t) = \begin{bmatrix} \cos(2\pi Bt) \\ \sin(2\pi Bt) \end{bmatrix}, \quad (3.13)$$

where B is a fixed frequency vector sampled prior to training. This optional encoding improves the representation of rapidly varying temporal patterns without altering the overall structure of the method.

In the present framework, this encoding is included to improve representation quality when the trajectories exhibit oscillatory or higher-frequency content. Since benchmark systems such as Van der Pol and Duffing can generate responses with nontrivial temporal curvature, enriching the time representation in this way helps the network approximate such behavior more efficiently.

3.2.2 Parameter network

The parameter network, denoted by N_{param} , maps the reconstructed local state and input to the surrogate coefficients,

$$\begin{bmatrix} \hat{k}(t) \\ \hat{d}(t) \\ \hat{g}(t) \end{bmatrix} = N_{\text{param}}(\hat{y}(t), \hat{v}(t), u(t); W_{\text{param}}). \quad (3.14)$$

Its role is to identify the state-dependent coefficient field that makes the reconstructed trajectory consistent with the second-order surrogate dynamics. Since the same network is shared across all trajectories, the learned mapping represents a global state-conditioned model rather than a separate local fit for each experiment.

In contrast to the Solution Network, the Parameter Network does not need to model the full trajectory as a temporal object. Its task is local: given the reconstructed state and current input, it must infer the surrogate coefficients that best explain the local dynamics at that point. For this reason, a feed-forward structure is appropriate. It provides sufficient flexibility for nonlinear regression while keeping the coefficient

map pointwise, interpretable, and easier to regularize. This design also reflects the intended role of the learned parameters, namely as local effective quantities rather than latent dynamic states with their own temporal memory.

In the simplest implementation, N_{param} is a standard feed-forward neural network. For systems with stronger regime dependence, the architecture can be extended using a Mixture-of-Experts formulation,

$$\begin{bmatrix} \hat{k} \\ \hat{d} \\ \hat{g} \end{bmatrix} = \sum_{j=1}^M \alpha_j(x, u) E_j(x, u), \quad (3.15)$$

where $E_j(\cdot)$ denotes the j th expert and $\alpha_j(\cdot)$ are the corresponding Softmax gating weights. This extension improves local expressiveness while preserving the interpretation of the output coefficients.

To further facilitate the discovery of complex coefficient manifolds, the raw state variables may be augmented with fixed nonlinear features such as polynomial terms, absolute values, or interaction terms. This augmentation does not change the surrogate structure itself. It only provides an additional inductive bias for learning coefficient fields with more complicated geometry. In all cases, the output of the parameter network remains the same three interpretable quantities: effective stiffness, effective damping, and effective input gain.

3.3 Curriculum Learning

The surrogate formulation and the dual-network structure together create a strongly coupled optimization problem. The parameter network defines the local differential constraints, but these constraints must be evaluated along the trajectory reconstructed by the solution network. At the same time, the quality of the reconstructed trajectory depends on the current estimate of the surrogate coefficients. If both networks are trained jointly from the beginning, the optimization may become ill-conditioned and converge to trivial or physically inconsistent solutions. To reduce this difficulty, SOLIS employs a curriculum-based training strategy that separates the learning problem into stages and progressively increases the burden on the parameter network.

The training procedure alternates between two phases. In the first phase, the emphasis is on trajectory reconstruction. The parameter network is frozen and only the solution network is updated. The objective in this stage is to recover a smooth continuous-time trajectory that fits the sparse measurements while remaining compatible with the current surrogate dynamics. In this way, the solution network first learns the geometric structure of the trajectories before the parameter field is refined.

In the second phase, the solution network is frozen and the parameter network is optimized. At this stage, the reconstructed trajectory is treated as a temporary reference, and the goal becomes the identification of a coefficient field that explains the observed motion through the second-order surrogate model. This separation reduces the direct conflict between trajectory fitting and parameter estimation and makes the optimization more stable than naive joint training. The overall training procedure is summarized in Table 3.1.

A key component of the second phase is the use of *Local Physics Hints*. The main purpose of these hints is to prevent the parameter network from collapsing to uninformative coefficient fields during early optimization. Instead of relying only on the non-convex physics residual, SOLIS constructs local analytical parameter estimates from the reconstructed trajectory by means of sliding-window ridge regression. These estimates are then used as weak supervision targets for the parameter network. Since they are local and approximate, they do not replace the physics-informed objective. Rather, they provide a stabilizing anchor that guides the parameter network toward physically plausible regions of the parameter space.

To construct these hints, consider a local window centered around a collocation point t . Within this window, the surrogate dynamics are approximated in the linear regression form

$$Y \approx \Phi \theta, \quad (3.16)$$

where

$$Y = [\hat{v}(\tau)]_{\tau \in \mathcal{W}_t^{(w)}}, \quad \Phi = [-\hat{y}(\tau), -\hat{v}(\tau), u(\tau)]_{\tau \in \mathcal{W}_t^{(w)}}. \quad (3.17)$$

Table 3.1: SOLIS training procedure.

Step	Description
1	Input: Measurements $\mathcal{D}_m$, collocation points $\mathcal{D}_c$, phase lengths K_1, K_2 , hint decay γ , ridge parameter λ_r
2	Initialize: Networks $\mathcal{N}_{\text{sol}}$ and $\mathcal{N}_{\text{param}}$
3	For epoch = 1 to E do
4	Sample minibatch $\mathcal{B}$ from $\mathcal{D}_m$ and $\mathcal{D}_c$
5	Compute $\hat{\mathbf{x}}, \dot{\hat{\mathbf{x}}} \leftarrow \mathcal{N}_{\text{sol}}(\mathcal{B})$
6	Compute $\hat{\boldsymbol{\theta}} \leftarrow \mathcal{N}_{\text{param}}(\hat{\mathbf{x}}, \mathbf{u})$
7	Compute $\mathcal{L}_{\text{phys}}$
8	Set $k \leftarrow \text{epoch mod } (K_1 + K_2)$
9	If $k < K_1$ then
10	Freeze $\mathcal{N}_{\text{param}}$, unfreeze $\mathcal{N}_{\text{sol}}$
11	Compute $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{ic}}$
12	Set $\mathcal{L}_{\text{total}} \leftarrow \lambda_d \mathcal{L}_{\text{data}} + \lambda_{\text{ic}} \mathcal{L}_{\text{ic}} + \lambda_p \mathcal{L}_{\text{phys}}$
13	Update $\mathcal{N}_{\text{sol}}$
14	Else
15	Freeze $\mathcal{N}_{\text{sol}}$, unfreeze $\mathcal{N}_{\text{param}}$
16	Decay hint weight: $\lambda_h \leftarrow \lambda_h \gamma$
17	Construct local matrices Φ and Y on the minibatch
18	Compute $\boldsymbol{\theta}_{\text{ridge}}^*$ via Eq. (3.18)
19	Compute $\mathcal{L}_{\text{hint}}$ and reliability weights w_t
20	Compute $\mathcal{L}_{\text{roll}}$ and $\mathcal{L}_{\text{TV}}$
21	Set $\mathcal{L}_{\text{total}} \leftarrow \lambda_p \mathcal{L}_{\text{phys}} + \lambda_h \mathcal{L}_{\text{hint}} + \lambda_{\text{reg}} (\mathcal{L}_{\text{roll}} + \mathcal{L}_{\text{TV}})$
22	Update $\mathcal{N}_{\text{param}}$
23	End If
24	End For
25	Return: Trained networks $\mathcal{N}_{\text{sol}}$ and $\mathcal{N}_{\text{param}}$

Here, $\mathcal{W}_t^{(w)}$ denotes a local rolling window of length w . The corresponding ridge estimate is given by

$$\boldsymbol{\theta}_{\text{ridge}}^*(t) = \left(\Phi^\top \Phi + \lambda_r I \right)^{-1} \Phi^\top Y, \quad (3.18)$$

where $\lambda_r > 0$ is the ridge regularization parameter. In practice, the window length is randomized during training so that the hints do not overfit a single temporal scale. This improves robustness and encourages consistency across multiple local linearizations.

The curriculum therefore serves two related purposes. First, it decouples trajectory learning from parameter identification in the early stages of training. Second, it enriches parameter learning with analytical local guidance. Together, these

mechanisms help the model avoid the optimization failures commonly encountered in inverse physics-informed training.

3.4 Loss Functions and Optimization

The training objective of SOLIS combines data consistency, physics consistency, weak local supervision, and regularization of the learned coefficient field. Since the training proceeds in phases, not all loss terms are active at every stage. The total optimization objective is assembled from a common set of components, each serving a distinct role in the learning process.

The core coupling term is the physics residual induced by the second-order surrogate dynamics. At a collocation point t , the residual is defined as

$$r(t) = \begin{bmatrix} \frac{d\hat{y}}{dt} - \hat{v}(t) \\ \frac{d\hat{v}}{dt} + \hat{k}(t)\hat{y}(t) + \hat{d}(t)\hat{v}(t) - \hat{g}(t)u(t) \end{bmatrix}. \quad (3.19)$$

The corresponding physics loss is

$$\mathcal{L}_{\text{phys}} = \frac{1}{|\mathcal{T}_c|} \sum_{t \in \mathcal{T}_c} \|r(t)\|_2^2. \quad (3.20)$$

This term enforces consistency between the reconstructed trajectories and the learned surrogate coefficients. Since the required derivatives are obtained by automatic differentiation, the residual can be evaluated directly on the neural representation of the trajectory.

The trajectory reconstruction objective is governed by a data fidelity term defined over the available measurements,

$$\mathcal{L}_{\text{data}} = \frac{1}{|\mathcal{D}_m|} \sum_{(t_i, y_i) \in \mathcal{D}_m} \|\hat{y}(t_i) - y_i\|_2^2. \quad (3.21)$$

When the initial condition is known, an additional initial-condition penalty may be included as

$$\mathcal{L}_{\text{ic}} = \|\hat{x}(0) - x_0\|_2^2. \quad (3.22)$$

These terms are especially important in the first phase of training, where the primary goal is to recover a smooth and reliable continuous-time trajectory from sparse observations.

During the parameter-identification phase, the analytical ridge estimates introduced in the previous section are used to define the hint loss

$$\mathcal{L}_{\text{hint}} = \frac{1}{|\mathcal{T}_c|} \sum_{t \in \mathcal{T}_c} w_t \left\| \hat{\theta}(t) - \theta_{\text{ridge}}^*(t) \right\|_2^2, \quad (3.23)$$

where $w_t \in [0, 1]$ is a local reliability weight. These weights are chosen to reduce the influence of windows with weak excitation or poor conditioning. As a result, the hint loss does not force the parameter network to copy the ridge solution blindly. It only encourages agreement when the local regression estimate is informative.

To promote smooth coefficient fields and reduce abrupt variations across nearby states, a regularization term is applied to the learned parameters. This is expressed through a total variation type penalty on the coefficient field,

$$\mathcal{L}_{\text{TV}} = \frac{1}{|\mathcal{T}_c|} \sum_{t \in \mathcal{T}_c} \left\| \nabla_x \hat{\theta}(\hat{x}(t)) \right\|_1. \quad (3.24)$$

This regularization reflects the local Quasi-LPV assumption. If the surrogate is to capture the local geometry of the trajectories in a meaningful way, then the coefficient field should vary smoothly rather than fluctuate erratically from point to point.

An additional short-horizon rollout term may also be included to improve dynamical consistency over time. Let H denote the rollout horizon. Then a generic rollout loss can be written as

$$\mathcal{L}_{\text{roll}} = \frac{1}{H} \sum_{h=1}^H \left\| \hat{x}_{t+h}^{\text{roll}} - \hat{x}(t+h) \right\|_2^2, \quad (3.25)$$

where $\hat{x}_{t+h}^{\text{roll}}$ denotes the state obtained by forward integration of the local surrogate over the short horizon. This term complements the pointwise physics residual by encouraging consistent local evolution over multiple time steps.

With these components, the first phase of training focuses on the solution network and uses the objective

$$\mathcal{L}_{\text{P1}} = \lambda_d \mathcal{L}_{\text{data}} + \lambda_{\text{ic}} \mathcal{L}_{\text{ic}} + \lambda_p \mathcal{L}_{\text{phys}}, \quad (3.26)$$

where λ_d , λ_{ic} , and λ_p are nonnegative loss weights. In the second phase, the solution network is frozen and the parameter network is updated according to

$$\mathcal{L}_{\text{P2}} = \lambda_p \mathcal{L}_{\text{phys}} + \lambda_h \mathcal{L}_{\text{hint}} + \lambda_{\text{reg}} (\mathcal{L}_{\text{TV}} + \mathcal{L}_{\text{roll}}). \quad (3.27)$$

The weight of the hint term is typically decreased over training so that the optimization gradually relies less on the local analytical proxy and more on the physics-informed objective itself.

The final training procedure alternates between these two phases over the course of optimization. In each cycle, the solution network improves the continuity and fidelity of the reconstructed trajectories, and the parameter network refines the coefficient field that explains those trajectories through the surrogate dynamics. This alternating process constitutes the core of SOLIS and provides the trained surrogate model used in the experimental and control chapters.



4. GAIN-SCHEDULED CONTROLLER DESIGN VIA ALGEBRAIC MATCHING

This chapter extends the second-order Quasi-LPV surrogate introduced in Chapter 3 to a control setting. The main idea is to use the learned local model not only for analysis and prediction, but also for state-feedback controller design. Since the surrogate provides, at each state, an interpretable second-order approximation with effective stiffness, damping, and input gain, it becomes possible to compute feedback gains directly from the local model parameters. In this chapter, a simple gain-scheduled state-feedback design based on algebraic polynomial matching is presented.

4.1 Control over Quasi-LPV Models

After offline training, the learned surrogate yields, for each state $x(t)$, a local second-order approximation of the form

$$\dot{x}(t) = A(x(t))x(t) + B(x(t))u(t), \quad (4.1)$$

where

$$A(x) = \begin{bmatrix} 0 & 1 \\ -\hat{k}(x) & -\hat{d}(x) \end{bmatrix}, \quad B(x) = \begin{bmatrix} 0 \\ \hat{g}(x) \end{bmatrix}. \quad (4.2)$$

Here, $\hat{k}(x)$, $\hat{d}(x)$, and $\hat{g}(x)$ denote the state-dependent surrogate coefficients identified by the parameter network. This representation can be interpreted as a local linear second-order model whose parameters vary over the state space. As a result, it provides a natural basis for gain-scheduled control.

For regulation, or equivalently for tracking in error coordinates, a state-feedback law of the form

$$u(t) = -K(x(t))x(t) \quad (4.3)$$

is considered, where

$$K(x) = \begin{bmatrix} k_1(x) & k_2(x) \end{bmatrix}. \quad (4.4)$$

Substituting this feedback law into the local surrogate model gives the closed-loop dynamics

$$\dot{x}(t) = (A(x(t)) - B(x(t))K(x(t)))x(t). \quad (4.5)$$

Since both the model and the controller depend on the state, the resulting system is scheduled over the operating region through the learned coefficient field.

The purpose of this design is not to claim global optimality or formal robust stability guarantees. Rather, the aim is to exploit the interpretable local structure learned by the surrogate model and obtain a simple control law that adapts to local variations in the dynamics. In this sense, the controller serves as a direct demonstration that the identified surrogate is not only predictive, but also useful for control-oriented synthesis.

4.2 Gain Scheduling via Polynomial Matching

The closed-loop characteristic polynomial of the local model under the feedback law above is obtained from

$$A(x) - B(x)K(x) = \begin{bmatrix} 0 & 1 \\ -\hat{k}(x) - \hat{g}(x)k_1(x) & -\hat{d}(x) - \hat{g}(x)k_2(x) \end{bmatrix}. \quad (4.6)$$

Therefore, the local closed-loop characteristic polynomial becomes

$$s^2 + (\hat{d}(x) + \hat{g}(x)k_2(x))s + (\hat{k}(x) + \hat{g}(x)k_1(x)) = 0. \quad (4.7)$$

This expression has the same structure as a standard second-order polynomial, which makes algebraic pole matching particularly convenient.

To specify the desired local closed-loop behavior, the reference polynomial is chosen as

$$s^2 + 2\zeta_c\omega_c s + \omega_c^2 = 0, \quad (4.8)$$

where $\omega_c > 0$ is the desired natural frequency and $\zeta_c > 0$ is the desired damping ratio. Matching coefficients with the local closed-loop polynomial yields the scheduled gains

$$k_2(x) = \frac{2\zeta_c\omega_c - \hat{d}(x)}{\hat{g}_{\text{eff}}(x)}, \quad (4.9)$$

$$k_1(x) = \frac{\omega_c^2 - \hat{k}(x)}{\hat{g}_{\text{eff}}(x)}, \quad (4.10)$$

where $\hat{g}_{\text{eff}}(x)$ denotes an effective input gain used in implementation. In the ideal case, one may take $\hat{g}_{\text{eff}}(x) = \hat{g}(x)$. However, to avoid numerical difficulties and excessively large gains when the estimated input gain becomes too small, a lower bound is imposed:

$$\hat{g}_{\text{eff}}(x) = \max(\hat{g}(x), g_{\min}), \quad (4.11)$$

where $g_{\min} > 0$ is a prescribed minimum gain threshold.

This construction defines a scheduled controller because the feedback gains depend explicitly on the current local surrogate coefficients. If the identified model indicates increased local damping, the scheduled derivative gain is reduced accordingly. Likewise, if the local stiffness changes over the state space, the proportional gain is adapted to maintain the desired second-order behavior. In this way, polynomial matching provides a transparent connection between the learned surrogate parameters and the control law.

The simplicity of this design is one of its main advantages. The controller is obtained in closed form and does not require solving an optimization problem online. At the same time, it remains sensitive to local changes in the learned model, which is precisely the benefit of using a state-dependent surrogate representation instead of a single fixed linear approximation.

4.3 Implementation Details

Although the algebraic matching rule is simple, its direct use in a real-time loop requires several practical precautions. The first issue is that the scheduled gains are computed from learned surrogate coefficients, and these coefficients may contain local estimation errors or abrupt variations. For this reason, the implemented gains are bounded within predefined intervals,

$$k_1(x) \in [k_{1,\min}, k_{1,\max}], \quad k_2(x) \in [k_{2,\min}, k_{2,\max}], \quad (4.12)$$

so that unrealistic or excessively aggressive control actions are avoided. In addition, the gain updates may be rate-limited to prevent sharp changes in the controller caused by local fluctuations of the learned parameter field.

A second issue is the possibility of algebraic loops. In the proposed framework, the parameter network may depend on the control input in addition to the state. If the current control signal were used directly to compute the current surrogate coefficients, an implicit algebraic dependence could arise between coefficient evaluation and control computation. To avoid this problem, the surrogate coefficients are evaluated using the previously applied input,

$$\hat{\theta}(t) = N_{\text{param}}(x(t), u(t - \Delta t)), \quad (4.13)$$

where Δt denotes the control sampling interval. This preserves causal implementation and removes the algebraic loop in real-time execution.

A third consideration concerns robustness to surrogate mismatch. Since the learned model is only an approximate local representation of the true nonlinear dynamics, the scheduled controller should not rely on exact coefficient accuracy. One practical way to improve safety is to blend the scheduled gain with a fixed baseline stabilizing controller,

$$K_{\text{impl}}(x) = \alpha(x)K_{\text{sched}}(x) + (1 - \alpha(x))K_{\text{base}}, \quad (4.14)$$

where $K_{\text{sched}}(x)$ is the gain obtained from polynomial matching, K_{base} is a fixed stabilizing gain, and $\alpha(x) \in [0, 1]$ is a blending factor. In the simplest implementation, $\alpha(x)$ may be chosen as a constant. More refined choices may depend on confidence in the surrogate estimate.

Finally, actuator saturation must be taken into account in practical implementation. After the control law is computed, the control input is clipped to the allowable actuator range,

$$u_{\text{impl}}(t) = \text{sat}(-K_{\text{impl}}(x(t))x(t)). \quad (4.15)$$

This prevents infeasible commands and keeps the controller within safe operating limits.

The complete online implementation used in the closed-loop simulations is summarized in Table 4.1. The procedure combines surrogate-based gain scheduling with practical safety mechanisms such as gain clamping, rate limiting, and actuator saturation.

Table 4.1: Online closed-loop control procedure using the learned SOLIS surrogate.

Step	Description
1	Input: Initial state x_0 , reference functions $r(t)$, $\dot{r}(t)$, $\ddot{r}(t)$, design parameters ζ_c, ω_c , baseline gain K_0 , blending factor α , minimum effective gain $g_{\min}$, gain limits, rate limits, actuator limits, time step Δt , simulation horizon N
2	Initialize: $x \leftarrow x_0, t \leftarrow t_0, u_{\text{prev}} \leftarrow 0, k_1 \leftarrow K_0^{(1)}, k_2 \leftarrow K_0^{(2)}$
3	For $n = 1$ to N do
4	Evaluate reference signals $r(t), \dot{r}(t), \ddot{r}(t)$
5	Extract output and velocity from state: $y \leftarrow x_1, v \leftarrow x_2$
6	Compute tracking errors: $e \leftarrow y - r(t), \dot{e} \leftarrow v - \dot{r}(t)$
7	Infer local surrogate coefficients from the learned model using current state and previous input
8	$[\hat{k}, \hat{d}, \hat{g}]^\top \leftarrow \mathcal{N}_{\text{param}}(x, u_{\text{prev}})$
9	Compute effective gain: $\hat{g}_{\text{eff}} \leftarrow \max(\hat{g}, g_{\min})$
10	Compute scheduled gains by polynomial matching
11	$k_{2,\text{raw}} \leftarrow \frac{2\zeta_c\omega_c - \hat{d}}{\hat{g}_{\text{eff}}}$
12	$k_{1,\text{raw}} \leftarrow \frac{\omega_c^2 - \hat{k}}{\hat{g}_{\text{eff}}}$
13	Blend with baseline gain
14	$k_{1,\text{des}} \leftarrow (1 - \alpha)K_0^{(1)} + \alpha k_{1,\text{raw}}$
15	$k_{2,\text{des}} \leftarrow (1 - \alpha)K_0^{(2)} + \alpha k_{2,\text{raw}}$
16	Apply gain saturation: $k_{1,\text{des}} \leftarrow \text{sat}(k_{1,\text{des}}), k_{2,\text{des}} \leftarrow \text{sat}(k_{2,\text{des}})$
17	Apply gain rate limits and update gains
18	$k_1 \leftarrow k_1 + \text{sat}(k_{1,\text{des}} - k_1)$
19	$k_2 \leftarrow k_2 + \text{sat}(k_{2,\text{des}} - k_2)$
20	Compute feedback control term: $u_{\text{fb}} \leftarrow -(k_1 e + k_2 \dot{e})$
21	Compute optional feedforward term: $u_{\text{ff}} \leftarrow \frac{\ddot{r}(t)}{\hat{g}_{\text{eff}}}$
22	Add optional probing excitation Δu if enabled
23	Form raw control input: $u_{\text{raw}} \leftarrow u_{\text{ff}} + u_{\text{fb}} + \Delta u$
24	Apply actuator saturation: $u \leftarrow \text{sat}(u_{\text{raw}})$
25	Propagate the ground-truth plant by one step using u
26	Store $x, u, \hat{k}, \hat{d}, \hat{g}, k_1, k_2, r, \dot{r}, \ddot{r}$
27	Update $u_{\text{prev}} \leftarrow u, x \leftarrow x^+, t \leftarrow t + \Delta t$
28	End For

Overall, the proposed controller is intentionally simple. Its purpose is to demonstrate that the learned Quasi-LPV surrogate can be used directly for state-feedback synthesis in a transparent way. The controller does not attempt to provide a complete nonlinear control theory framework. Instead, it shows that the interpretable surrogate model

learned by SOLIS is sufficiently structured to support scheduled control design with minimal additional machinery.



5. PERFORMANCE ANALYSIS

This chapter evaluates the performance of the proposed framework from three complementary perspectives. First, the solution learning capability of the Solution Network is analyzed in order to assess how accurately the measured trajectories can be reconstructed under the second-order Q-LPV surrogate hypothesis. Second, the fidelity of the learned local surrogate model is examined by evaluating how well it captures the underlying nonlinear dynamics, particularly in terms of open-loop predictive behavior and phase portrait consistency. Third, the control-oriented value of the learned surrogate is assessed in a closed-loop setting, where the identified local model is used to synthesize scheduled state-feedback gains and the resulting tracking performance is analyzed on the nonlinear system.

The first two parts of the evaluation are carried out on all benchmark systems considered in this thesis. The closed-loop control study, however, is conducted only on the Duffing oscillator. This benchmark is used as the main case study for controller validation, since it provides a sufficiently rich nonlinear setting while still allowing the effect of the proposed gain-scheduled controller to be examined clearly.

5.1 Experimental Setup and Datasets

In this study, the proposed framework is evaluated on three benchmark systems representing different types of nonlinear behavior. These systems are selected to examine identification accuracy and control relevance under different dynamical characteristics.

The first benchmark is the Van der Pol oscillator, which is a classical nonlinear system characterized by state-dependent damping and self-sustained oscillatory behavior. This example is particularly useful for evaluating whether the proposed framework can recover nonlinear dissipation effects through the learned surrogate coefficients.

The second benchmark is the Duffing oscillator, which introduces nonlinear stiffness through a cubic restoring term. This system is used to assess whether the proposed surrogate can represent nonlinear restoring dynamics while preserving interpretable second-order structure. In addition, the Duffing oscillator is used for the closed-loop control experiments reported later in this chapter.

The third benchmark is the two-tank process, which represents a real laboratory-scale system with coupled nonlinear flow dynamics and measurement noise. In contrast to the simulated benchmarks, this example provides a more realistic setting in which the learned model must cope with imperfections in the data and unmodeled effects.

For each system, trajectory data are generated or collected under known excitation inputs. A subset of the available data is used for model training, while the remaining trajectories are reserved for evaluation. The training process follows the procedure described in Chapter 3, where the Solution Network reconstructs smooth continuous-time trajectories and the Parameter Network identifies the local surrogate coefficient field.

For the synthetic benchmarks, the measured outputs were corrupted with small additive measurement noise in order to avoid an idealized noiseless identification setting. In the experiments, noise was added to the observed output channel before training, while the underlying system dynamics remained unchanged. The noise level was kept intentionally modest so that the benchmark would remain identifiable while still reflecting realistic imperfections in measurement. For the two-tank benchmark, no artificial corruption was added; instead, the data contain intrinsic measurement noise arising from real laboratory sensing and acquisition. This distinction is important when interpreting the reconstruction and rollout results, since the proposed framework is evaluated under both synthetic and naturally occurring noisy observations.

The proposed framework is evaluated against several baseline methods representing both physics-informed neural identification and classical parametric modeling approaches. These baselines are selected to reflect both physics-informed neural identification and a classical control-oriented parametric alternative.

- IPINN: A baseline inverse physics-informed neural network based on a fixed-parameter second-order model. In this approach, the system dynamics are assumed to follow the same second-order structure used in the proposed framework, but the coefficients k , d , and g are treated as single global trainable scalars rather than state-dependent quantities. The solution is represented by a standard feedforward neural network, and the trajectory and the fixed parameters are learned jointly through the physics-informed loss.
- IPINN-M: A multi-trajectory inverse PINN baseline. This method uses the same trajectory representation mechanism as the proposed Solution Network, including input conditioning through recurrent context encoding and FiLM-based modulation. However, the governing surrogate model still assumes constant global coefficients. Therefore, this baseline isolates the contribution of the state-dependent parameter field by keeping the trajectory architecture expressive while restricting the dynamics to a fixed-parameter second-order form.
- TF: A low-order transfer function baseline estimated using MATLAB's built-in `tfest` function. This method identifies a parametric transfer function model directly from input-output data and provides a classical linear benchmark against which the proposed nonlinear surrogate can be compared.

In the present experiments, the TF baseline was obtained by fitting low-order transfer function models with MATLAB's `tfest` routine. According to the official documentation, `tfest` estimates continuous-time or discrete-time transfer function models from time-domain or frequency-domain data using a prescribed pole-zero structure, with optional delay specification and configurable estimation options [23,24]. In this thesis, `tfest` was used as a classical low-order linear identification baseline. A small search over candidate model structures was performed by varying the assumed transfer function order, and the model with the best fit under the adopted evaluation criterion was retained for comparison. This makes the TF baseline more informative than a single arbitrarily chosen transfer function structure, while still keeping it within the class of classical parametric linear models.

The performance of the proposed framework is evaluated using both reconstruction-based and dynamical metrics. For trajectory-level comparisons, the predicted outputs are compared against the reference trajectories under noisy measurements. For surrogate-model fidelity, the learned local dynamics are examined through phase portrait comparisons and open-loop rollout performance on all benchmark systems. For control experiments, closed-loop tracking behavior is analyzed only for the Duffing oscillator under the gain-scheduled state-feedback law described in Chapter 4.

For quantitative evaluation, the main performance measure reported in this thesis is the accuracy metric derived from the normalized root mean square error (NRMSE). Given the reference trajectory y_{true} and the predicted trajectory y_{pred} , the root mean square error is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{\text{true},i} - y_{\text{pred},i})^2}, \quad (5.1)$$

and the normalized root mean square error is computed as

$$\text{NRMSE} = \frac{\text{RMSE}}{\max(y_{\text{true}}) - \min(y_{\text{true}})}. \quad (5.2)$$

Using this quantity, the accuracy is defined as

$$\text{Accuracy}(\%) = \max(0, 1 - \text{NRMSE}) \times 100. \quad (5.3)$$

Accordingly, a higher accuracy value indicates better agreement between the predicted and reference trajectories. This metric is used throughout the chapter for both reconstruction and rollout evaluations.

5.2 Trajectory Reconstruction Results

This section analyzes the solution learning performance of the proposed framework. The main question here is whether the Solution Network can reconstruct smooth and accurate trajectories from sparse observations while remaining consistent with the second-order surrogate structure.

The reconstruction results are first evaluated on the observed trajectories. Since the Solution Network is optimized jointly with physics-based constraints, the objective is not only to interpolate the measured data points but also to recover a continuous-time

trajectory that remains dynamically meaningful. This is especially important when measurements are sparse or when velocity information is not directly available.

Representative trajectory reconstructions for the benchmark systems are presented in Figure 5.1. Figure 5.1(a) shows the results for the Van der Pol oscillator, Figure 5.1(b) shows the Duffing oscillator, and Figure 5.1(c) shows the two-tank system. In all three cases, the plots compare the reference trajectories with the trajectories reconstructed by the proposed framework and the selected baseline methods. The results indicate that SOLIS follows the measured dynamics more closely, particularly in regions with pronounced oscillatory behavior, nonlinear transitions, or stronger local curvature, where fixed-parameter baselines tend to accumulate mismatch.

The quantitative reconstruction results are summarized in Table 5.1. These results show that the proposed framework achieves the highest reconstruction accuracy on all benchmark systems. This supports the view that the second-order Quasi-LPV surrogate, together with the proposed training strategy, provides an effective representation of the trajectory geometry while remaining physically structured.

Table 5.1: Solution learning accuracy ($\uparrow$) on training trajectories.

System	IPINN	IPINN-M	SOLIS
Van der Pol	95.94%	98.07%	98.89%
Duffing	95.43%	97.56%	99.01%
Two-Tank	88.64%	91.25%	98.62%

Overall, these results indicate how well the proposed framework captures the solution manifold implied by the measured data. Strong reconstruction performance is important not only as an identification result in itself, but also because the reconstructed trajectories form the basis for learning the local surrogate coefficient field.

5.3 Surrogate Model Open-Loop Prediction Results

While trajectory reconstruction measures how well the observed data are fitted, it does not by itself establish whether the learned surrogate model captures the actual system dynamics in a meaningful way. For this reason, the local surrogate is evaluated in an open-loop setting through phase portrait comparisons and forward rollout experiments.

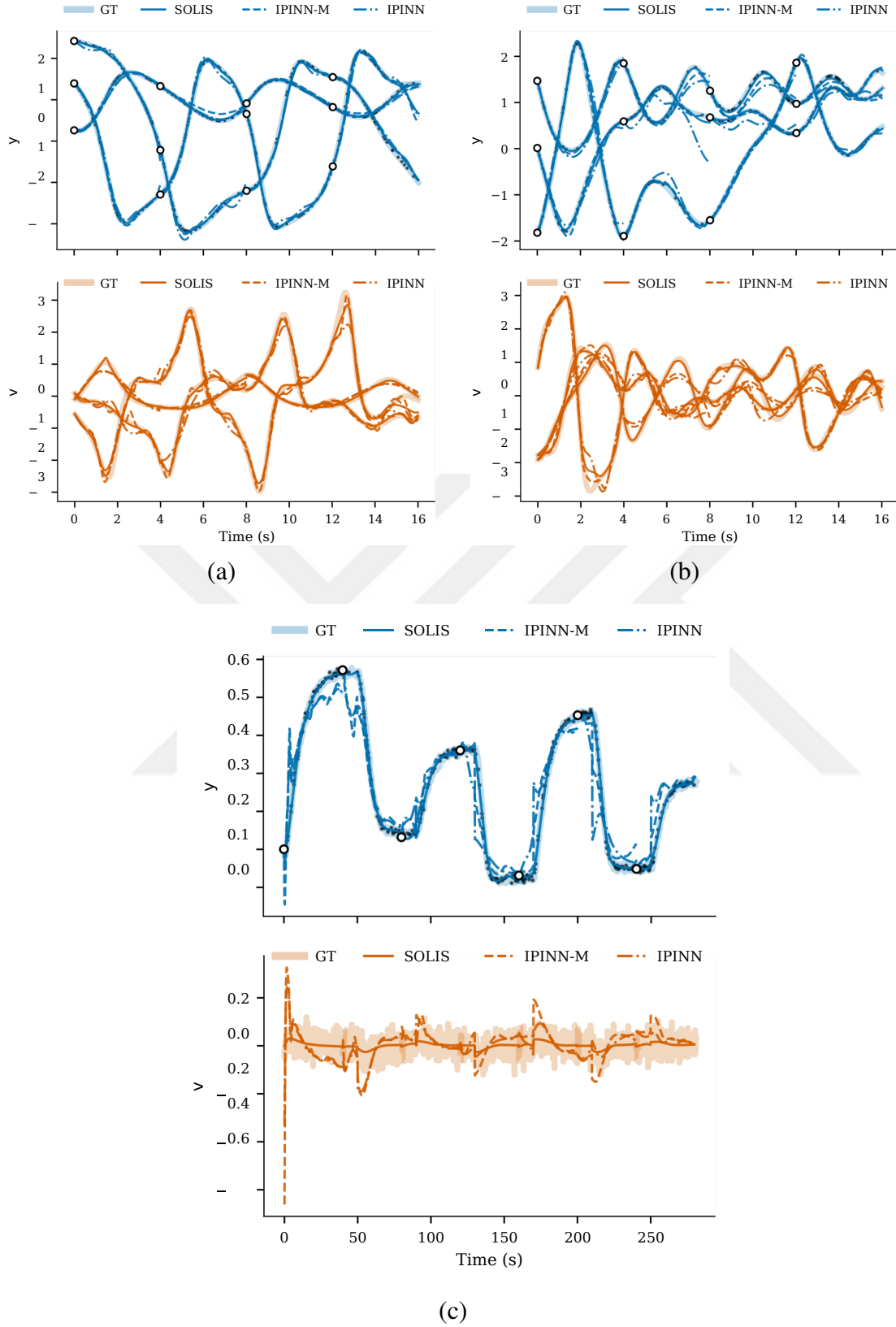


Figure 5.1: Representative trajectory reconstruction results for the benchmark systems: (a) Van der Pol oscillator, (b) Duffing oscillator, and (c) two-tank system.

A first qualitative assessment is obtained by comparing the vector field induced by the learned surrogate with the reference phase portrait of the underlying nonlinear system. Since the proposed framework identifies a state-dependent second-order local model, this comparison provides insight into whether the learned coefficient field reproduces the geometry of the true dynamics over the explored region of the state space. The phase portrait comparisons for the Van der Pol and Duffing oscillators are presented in Figure 5.2. In both cases, the learned surrogate reproduces the dominant flow structure over the regions covered by the trajectories. The cosine similarity maps provide a pointwise measure of agreement between the surrogate and the ground-truth vector field, showing that the highest consistency is obtained in the regions that are most relevant to the observed dynamics.

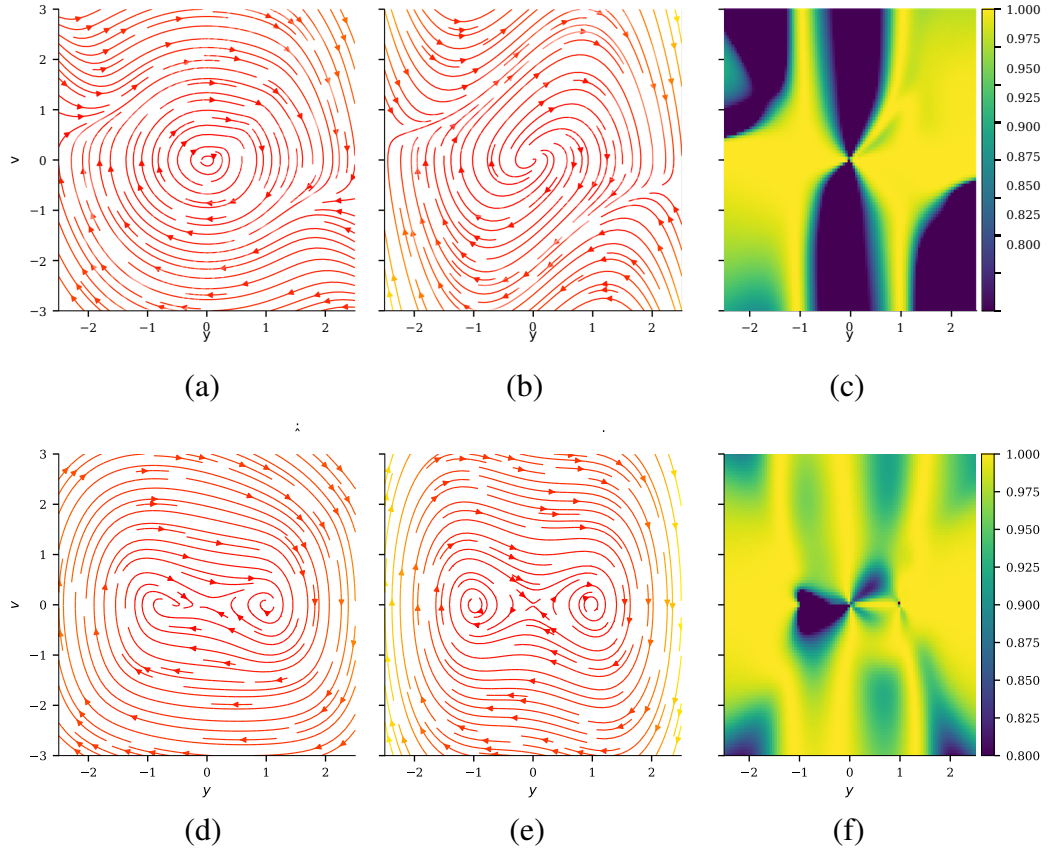


Figure 5.2: Phase portrait comparison results for the learned surrogate models. The top row corresponds to the Van der Pol oscillator and the bottom row corresponds to the Duffing oscillator. (a) Van der Pol surrogate phase portrait, (b) Van der Pol ground-truth phase portrait, (c) Van der Pol cosine similarity map, (d) Duffing surrogate phase portrait, (e) Duffing ground-truth phase portrait, and (f) Duffing cosine similarity map.

The corresponding quantitative phase portrait results are summarized in Table 5.2. The results show that SOLIS achieves the highest average cosine similarity on both benchmark systems, which indicates a more faithful recovery of the local vector field geometry than the fixed-parameter inverse PINN baselines.

Table 5.2: Phase portrait similarity ($\uparrow$) measured via average cosine similarity to the ground-truth field.

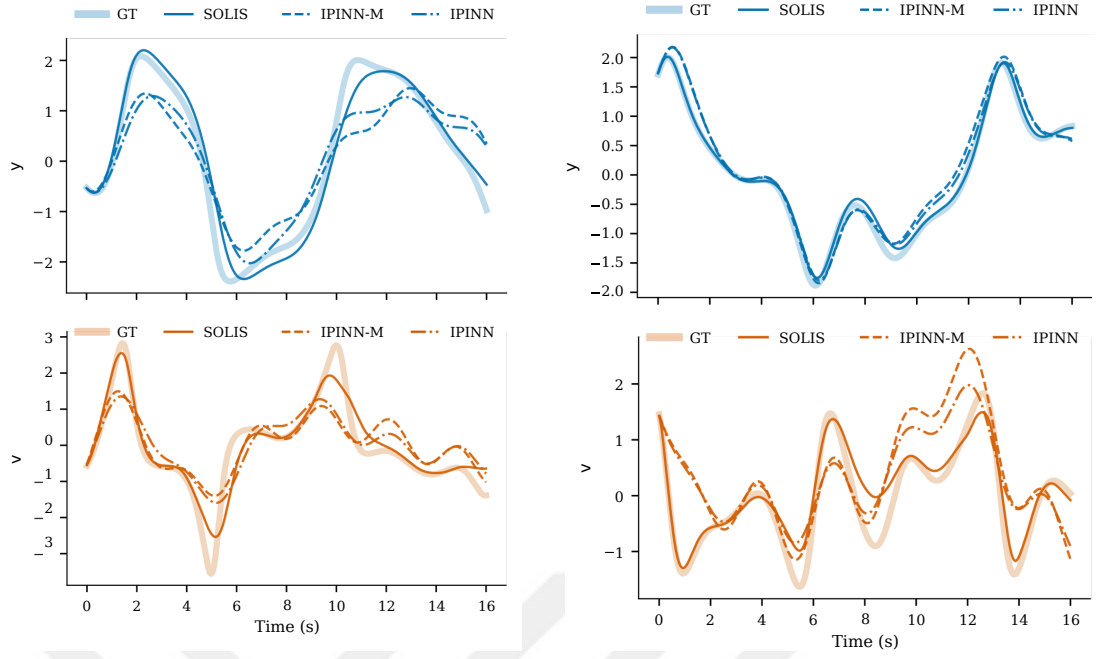
System	IPINN	IPINN-M	SOLIS
Van der Pol	0.70	0.67	0.86
Duffing	0.81	0.82	0.96

In addition to the phase portrait comparison, open-loop rollout experiments are used to assess predictive consistency beyond the training samples. Starting from a given initial condition, the learned surrogate dynamics are forward integrated and compared against the reference system trajectories. These results are illustrated in Figure 5.3, where representative rollouts are shown for the Van der Pol oscillator, the Duffing oscillator, and the two-tank system. In all three cases, the proposed method produces trajectories that remain closer to the reference response, whereas the baseline models accumulate larger deviations over time.

The quantitative rollout results are reported in Table 5.3. These results show that SOLIS attains the highest rollout accuracy on all benchmark systems. This is important because open-loop performance reflects whether the learned surrogate behaves as a coherent dynamical model rather than merely as a local interpolator. A model that fits observed samples but cannot produce consistent forward evolution would have limited value for analysis and control. In contrast, the results obtained here suggest that the learned coefficient field captures the local nonlinear mechanisms of the underlying systems in a physically meaningful way.

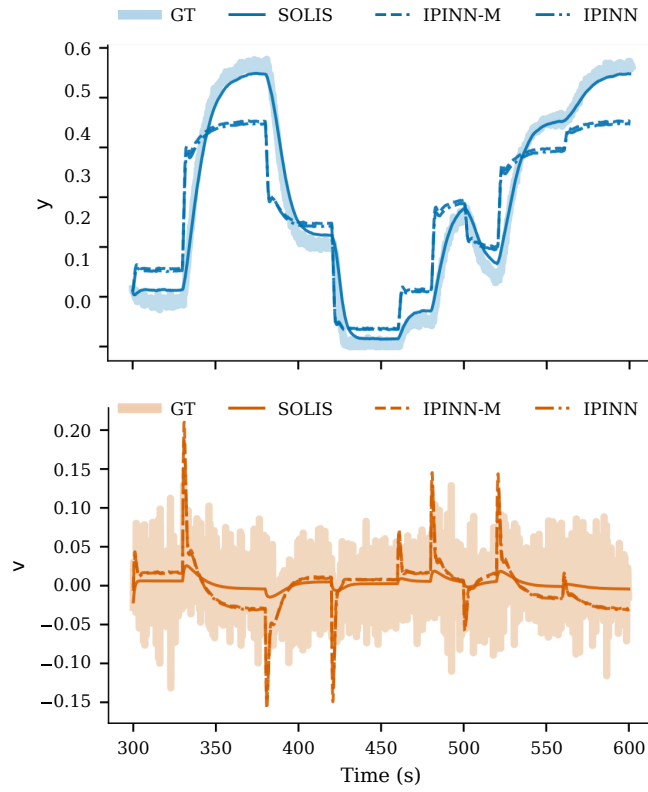
Table 5.3: Identification rollout accuracy ($\uparrow$) on test trajectories.

System	IPINN	IPINN-M	TF	SOLIS
Van der Pol	83.72%	81.84%	70.22%	90.54%
Duffing	76.60%	75.27%	74.60%	83.07%
Two-Tank	82.74%	82.90%	77.74%	84.29%



(a)

(b)



(c)

Figure 5.3: Representative open-loop rollout results for the benchmark systems: (a) Van der Pol oscillator, (b) Duffing oscillator, and (c) two-tank system.

5.4 Closed-Loop Control Performance

The final stage of the evaluation examines the control-oriented value of the learned surrogate model. After offline identification, the estimated local coefficients are used online to construct scheduled state-feedback gains according to the controller design described in Chapter 4. In the implemented simulation, the controller uses scheduled gains obtained from the learned local parameters together with blending, gain limiting, and input saturation. The simulation records the state trajectory, control input, estimated surrogate coefficients, scheduled gains, and reference signals over time.

The closed-loop experiments in this thesis are conducted on the Duffing oscillator only. This choice is made in order to examine the proposed controller on a nonlinear benchmark with state dependent restoring dynamics, while keeping the interpretation of the results clear. Accordingly, the results reported in this section should be understood as a case study demonstrating the control use of the learned surrogate, rather than as a benchmark wide validation across all datasets.

In these experiments, the controller operates by updating the feedback gains according to the current local surrogate coefficients. This creates an adaptive gain-scheduled structure that reacts to changes in the local dynamics as the state evolves. The resulting closed-loop response is compared against the desired reference behavior under several reference profiles. In the simulations used in this study, the controller parameters are chosen as shown in Table 5.4.

Table 5.4: Controller parameters used in the closed-loop simulations.

Parameter	Value
Desired damping ratio, ζ_c	0.9
Desired natural frequency, ω_c	6.0
Baseline gain, K_0	(2.0, 1.5)
Blending factor, α	0.7
Minimum effective gain, $g_{\min}$	10^{-2}
Minimum actuator input, $u_{\min}$	-5.0
Maximum actuator input, $u_{\max}$	5.0

These values define the desired second-order closed-loop behavior together with the practical safety constraints used during online implementation.

Representative closed-loop tracking results for the Duffing oscillator are presented in Figure 5.4. The figure compares the output trajectory with the desired reference under different reference signals. Qualitatively, the scheduled controller is expected to preserve good tracking performance while adapting its gains to the local surrogate dynamics. This allows the controller to respond to changes in the operating region without relying on a single fixed linear model.

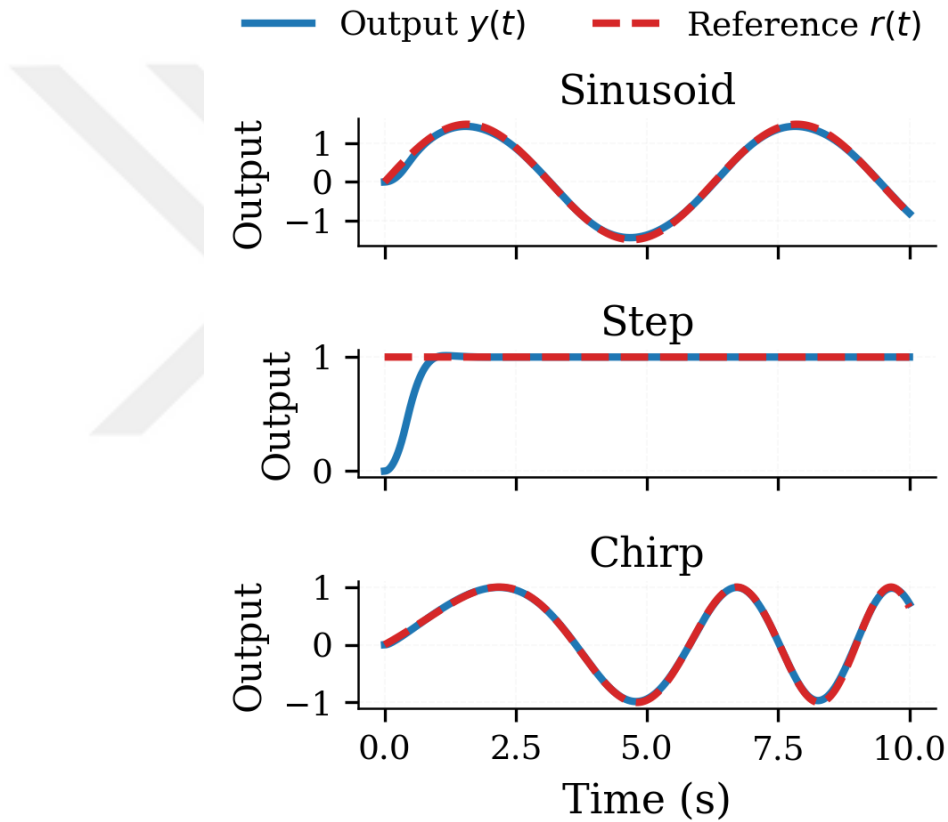


Figure 5.4: Closed-loop tracking performance on the Duffing oscillator under the gain-scheduled controller.

Quantitative performance measures are summarized in Table 5.5. In this thesis, the reported metrics are selected to reflect both tracking quality and control behavior. The tracking performance is evaluated through the trajectory accuracy and RMSE with respect to the reference signal. In addition, a control-effort measure is reported in order to quantify the input magnitude required to achieve the tracking objective. For

the step reference case, settling time and overshoot are also included to characterize the transient response. These metrics are not reported for the sinusoidal and chirp references, since they are not directly comparable to standard step-response characteristics. Since the scheduled controller is intended as a practical and interpretable control law rather than a purely numerical tracking scheme, evaluating both output performance and input usage is important.

The closed-loop analysis completes the main argument of the thesis. The contribution of the proposed framework is not limited to trajectory fitting or surrogate identification. Rather, the learned Quasi-LPV representation is shown to be directly usable for controller synthesis in the Duffing case study. Successful closed-loop performance therefore provides evidence that the identified surrogate captures control-relevant structure in the underlying nonlinear dynamics.

Table 5.5: Closed-loop control performance on the Duffing oscillator.

Metric	Sinusoidal	Step	Chirp
Accuracy (%)	98.26	81.25	99.17
RMSE	0.0523	0.1875	0.0166
Settling Time (s)	–	0.9110	–
Overshoot (%)	–	1.2140	–
Control Effort	1.2807	1.2568	2.3977

6. CONCLUSION AND FUTURE WORK

This thesis presented a physics-informed framework for learning interpretable surrogate models of nonlinear dynamical systems from sparse trajectory data and for using these learned models in controller synthesis. The main objective was to address a fundamental limitation in nonlinear system identification, namely the gap between model flexibility and control-oriented interpretability. Classical identification methods provide structured models that are suitable for analysis and controller design, but they often become restrictive in the presence of strong nonlinearities. In contrast, modern neural approaches can represent complex dynamics with high flexibility, yet they usually do not produce models that are easily interpretable or directly usable for control.

To address this issue, the thesis proposed a structured learning framework based on a second-order Quasi-LPV surrogate hypothesis. Instead of assuming a fixed global parametric model, the unknown nonlinear dynamics were represented through a state-dependent second-order surrogate with interpretable local coefficients corresponding to effective stiffness, damping, and input gain. This formulation allowed the identification problem to be recast as the learning of a coefficient field over the state space rather than the estimation of a single global parameter vector.

A dual-network architecture was introduced in order to separate trajectory reconstruction from parameter identification. Within this structure, the Solution Network was used to recover smooth continuous-time trajectories from sparse measurements, while the Parameter Network was used to identify the local surrogate coefficients along those trajectories. To reduce the optimization difficulties typically encountered in inverse physics-informed learning, a curriculum-based training strategy was adopted. This strategy alternated between trajectory learning and parameter learning, and local ridge-regression based physics hints were incorporated as weak supervision in order to stabilize the identification of the coefficient field.

The experimental results showed that the proposed framework can reconstruct system trajectories accurately, recover meaningful local surrogate dynamics, and generate coherent open-loop predictions across benchmark nonlinear systems. In particular, the framework was able to produce state-dependent surrogate models that preserved classical second-order interpretation while remaining expressive enough to capture nonlinear behavior. These results support the main premise of the thesis: it is possible to learn nonlinear surrogate models that are both data-consistent and structurally useful for control-oriented analysis.

Beyond identification, the thesis also considered the use of the learned surrogate in controller synthesis. By exploiting the local second-order structure of the identified Quasi-LPV model, a gain-scheduled state-feedback controller was constructed through algebraic polynomial matching. This controller used the estimated local stiffness, damping, and input gain to compute feedback gains online in a simple and interpretable manner. The closed-loop case study on the Duffing oscillator demonstrated that the learned surrogate can support adaptive state-feedback design and can be used not only for modeling, but also for control.

Although the results are encouraging, the present study also has several limitations. First, the proposed framework has been validated only on low-dimensional benchmark systems. While the Van der Pol oscillator, the Duffing oscillator, and the two-tank process provide useful test cases for examining the basic identification and control properties of the method, they do not yet establish how the framework behaves in higher-dimensional nonlinear systems with stronger state coupling, richer actuation structure, or partial observability. Second, the surrogate hypothesis was formulated for second-order single-output dynamics, and the closed-loop control study was limited to a single nonlinear benchmark. Third, while the learned coefficient fields were shown to be useful in practice, no formal stability guarantees were derived for the overall gain-scheduled closed-loop system. These limitations point directly to several promising directions for future research.

A first natural extension is the generalization of the proposed framework to multi-input multi-output systems and, more broadly, to higher-dimensional nonlinear dynamics.

Extending the surrogate hypothesis from scalar second-order dynamics to coupled vector-valued systems would substantially broaden the class of applications. Such an extension would require more structured parameterizations of the local dynamics and more careful treatment of coupling effects between channels. Representative future test cases may include multi-link robotic manipulators, robotic arms with flexible joints, aerial robotic systems with coupled translational and rotational dynamics, and larger process-control benchmarks. These systems would provide a more demanding setting in which the scalability and structural usefulness of the proposed surrogate formulation could be evaluated.

A second important direction is the incorporation of uncertainty quantification into the surrogate learning process. Since the learned model is obtained from finite and possibly noisy data, it would be valuable to estimate confidence in the identified coefficient field. This could be approached through Bayesian neural models, ensemble-based methods, interval methods, or other uncertainty-aware learning strategies. Uncertainty information would be especially useful for determining when the learned surrogate can be trusted for control and when more conservative action is needed.

A third direction concerns the control design itself. In this thesis, the controller was intentionally kept simple and transparent by relying on algebraic pole matching. Future work may explore more advanced control structures built on the learned surrogate, including optimal control, robust gain scheduling, model predictive control, adaptive control, or observer-based output feedback designs. These approaches could make fuller use of the learned local model while improving performance and robustness.

Another important research direction is the analysis of closed-loop stability and robustness. While the proposed controller was motivated by the local surrogate structure, a more complete theoretical study is needed in order to understand the interaction between model mismatch, scheduling variations, and closed-loop behavior. Lyapunov-based analysis, LPV stability tools, or input-to-state stability arguments may provide suitable starting points for such investigations.

Finally, the overall framework may be extended to richer classes of nonlinear systems and more realistic experimental settings. Applications to higher-dimensional systems, systems with partial observability, systems with external disturbances, and systems with stronger uncertainty would help clarify both the strengths and the limitations of the approach. Such studies would also provide a better understanding of when interpretable local surrogate learning offers a genuine advantage over purely black-box neural models.

In summary, this thesis proposed a control-oriented physics-informed learning framework that connects nonlinear system identification with interpretable surrogate modeling and scheduled feedback design. The main contribution is not only a method for fitting trajectories, but a structured perspective on how learned local dynamics can be made useful for analysis and control. It is hoped that this work provides a useful step toward data-driven modeling approaches that remain compatible with the needs of control engineering.

REFERENCES

- [1] **Ljung, L.** (1999). *System Identification: Theory for the User*, Prentice Hall PTR.
- [2] **Pillonetto, G., Aravkin, A., Gedon, D., Ljung, L., Ribeiro, A.H. and Schön, T.B.** (2025). Deep networks for system identification: a survey, *Automatica*, 171, 111907.
- [3] **Chen, R.T., Rubanova, Y., Bettencourt, J. and Duvenaud, D.K.** (2018). Neural ordinary differential equations, *Advances in neural information processing systems*, 31.
- [4] **Rahman, A., Drgoňa, J., Tuor, A. and Strube, J.** (2022). Neural ordinary differential equations for nonlinear system identification, *American Control Conference*, pp.3979–3984.
- [5] **Yang, Y. and Li, H.** (2025). Neural ordinary differential equations for robust parameter estimation in dynamic systems with physical priors, *Applied Soft Computing*, 169, 112649.
- [6] **Sertbaş, A.E. and Kumbasar, T.** (2025). Stable-by-Design Neural Network-Based LPV State-Space Models for System Identification, *Conference of Image Processing, Wavelet and Applications on Real World Problems*.
- [7] **Raissi, M., Perdikaris, P. and Karniadakis, G.E.** (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational physics*, 378, 686–707.
- [8] **Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S. and Yang, L.** (2021). Physics-informed machine learning, *Nature Reviews Physics*, 3(6), 422–440.
- [9] **Wang, S., Teng, Y. and Perdikaris, P.** (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM Journal on Scientific Computing*, 43(5), A3055–A3081.
- [10] **Kharazmi, E., Cai, M., Zheng, X., Zhang, Z., Lin, G. and Karniadakis, G.E.** (2021). Identifiability and predictability of integer-and fractional-order epidemiological models using physics-informed neural networks, *Nature Computational Science*, 1(11), 744–753.
- [11] **Ferah, M.A. and Kumbasar, T.** (2025). Introducing Interval Neural Networks for Uncertainty-Aware System Identification, *Int. Congress on Human-Computer Interaction, Optimization and Robotic Applications*.

- [12] **Güven, Y. and Kumbasar, T.** (2025). Fuzzy Logic Strikes Back: Fuzzy ODEs for Dynamic Modeling and Uncertainty Quantification, *IEEE Transactions on Artificial Intelligence*.
- [13] **Tutunji, T.A.** (2016). Parametric system identification using neural networks, *Applied Soft Computing*, 47, 251–261.
- [14] **Wu, R.T. and Jahanshahi, M.R.** (2019). Deep convolutional neural network for structural dynamic response estimation and system identification, *Journal of Engineering Mechanics*, 145(1), 04018125.
- [15] **Forgione, M., Muni, A., Piga, D. and Gallieri, M.** (2023). On the adaptation of recurrent neural networks for system identification, *Automatica*, 155, 111092.
- [16] **Hochreiter, S. and Schmidhuber, J.** (1997). Long Short-Term Memory, *Neural Comput.*, 9(8), 1735–1780.
- [17] **Wang, Y.** (2017). A new concept using LSTM Neural Networks for dynamic system identification, *2017 American control conference (ACC)*, IEEE, pp.5324–5329.
- [18] **Forgione, M. and Piga, D.** (2021). Continuous-time system identification with neural networks: Model structures and fitting criteria, *European Journal of Control*, 59, 69–81.
- [19] **Rugh, W.J. and Shamma, J.S.** (2000). Research on gain scheduling, *Automatica*, 36(10), 1401–1425.
- [20] **Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.** (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp.1724–1734.
- [21] **Perez, E., Strub, F., de Vries, H., Dumoulin, V. and Courville, A.** (2018). FiLM: Visual Reasoning with a General Conditioning Layer, *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp.3942–3950.
- [22] **Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T. and Ng, R.** (2020). Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, *Advances in Neural Information Processing Systems*, volume 33, pp.7537–7547.
- [23] **MathWorks** (2026). *tfest – Estimate transfer function model*, MATLAB System Identification Toolbox documentation, accessed 2026.
- [24] **MathWorks** (2026). *tfestOptions – Option set for tfest*, MATLAB System Identification Toolbox documentation, accessed 2026.

CURRICULUM VITAE

Name SURNAME: Murat Furkan MANSUR

EDUCATION:

- **B.Sc.:** 2023, Istanbul Technical University, Faculty of Electrical & Electronics Engineering, Control and Automation Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- Research Assistant, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Department of Control and Automation Engineering (Feb 2025–Present)
- Artificial Intelligence & Software Engineer, Cezeri Artificial Intelligence and Robotics Technologies (Baykar) (Jan 2024–Jan 2025)
- Part-Time Prototype and Data Analytics Engineer, PI Works (Oct 2023–Jan 2024)
- First Place in Istanbul and Fourth Place in Turkey, TÜBİTAK 46th High School Research Projects Competition (2015)

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Mansur, M. F., & Kumbasar, T.** (2026, June 21–26). *SOLIS: Physics-informed learning of interpretable neural surrogates for nonlinear systems* [Paper presentation]. 2026 International Joint Conference on Neural Networks (IJCNN) / IEEE World Congress on Computational Intelligence (WCCI), Maastricht, The Netherlands. (Accepted)
- **Mansur, M. F., & Kumbasar, T.** (2026, May 11–13). *SOLSTICE: Second-order local surrogate training for interpretable control extension* [Paper presentation]. 5th International Graduate Research Symposium (IGRS), Istanbul, Türkiye.