

Modeling and Event-Triggered Control of Multiple 3D Tower Cranes over WSNs

FAISAL ALTAF



KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden October 2010

XR-EE-RT 2010:021



Modeling and Event-Triggered Control of Multiple 3D Tower Cranes over WSNs

FAISAL ALTAF

Stockholm October 2010

Automatic Control Laboratory
School of Electrical Engineering
Kungliga Tekniska Högskolan

XR-EE-RT 2010:021

Abstract

Wireless control of 3D tower cranes is a topic of great industrial significance as it offers greater flexibility and lower maintenance cost. But the reduction in shared network resource utilization and the desired level of control performance against packet losses and delays are main concerns.

Our first aim is to develop a mathematical model of the 3D tower crane for the control design purpose. The second aim is to design a model-based event-triggered controller for wireless control of multiple 3D tower cranes to asymptotically track step reference inputs. The controller and cranes are connected in a star topology over simulated wireless sensor network with IEEE 802.15.4 MAC protocol for channel access. The third aim is to numerically evaluate and compare the performance of the event-triggered controller with that of a periodic controller under network induced delays, packet dropouts and with respect to scalability and the effect of MAC protocol on the stability.

The dynamic model for the 3D tower crane has been identified through physical modeling followed by lumped parametric estimation. The proposed event-triggered control policy for tracking problem uses the integral control structure for which theoretical results have been derived using Lyapunov input-to-state stability theorem.

Under nominal network settings, the event-triggered controller reduces the network resource utilization by 4 times but achieves almost the same control performance as compared to periodic controller executions. The network cost under event-triggered control policy not only depends on the point network induced delay enters into the system, but also increases almost monotonically with it. The event-triggered control policy utilizes the lower communication cost as compared to periodic case only up to certain amount of delay in a loop and hence must only be preferred over periodic case after considering the level and location of delay in the system. Under these conditions, up to 13 crane tasks are schedulable over network under event-triggered control policy using CSMA/CA random access. All the simulations have been done in MATLAB® Simulink using TrueTime toolbox.

Acknowledgements

I am highly grateful to my Allah Almighty for blessing me with great health, wisdom, consistent motivation and resilience to complete this research project successfully. I do not have any words to show my gratitude for my respected parents who have always been supportive beyond my expectations throughout my life. My parent's hardwork to move us from village to city and then facing very hard time to meet all the expenses and sacrificing all leisures and luxuries of their life just for our better education has always been a source of great motivation for me to improve my knowledge and do something better to make them feel proud. I would also like to thank my all siblings for their encouragement, motivation and prayers for me.

I am highly grateful to Automatic Control Laboratory (ACL) at KTH for providing me an excellent environment and resources that helped me a lot to produce better research. I would like to thank my thesis advisor and examiner, Assistant Professor Henrik Sandberg, for suggesting interesting topic which provided me with ample opportunity to learn many new things. He was always available for useful discussions. I would like to thank my co-supervisor Jose Araujo (PhD Candidate, ACL, KTH) for his cooperation, useful discussions, reviewing my thesis and willingness to help anytime.

I would like to thank Dr. Maben Rabi (Cambridge) for giving me useful tips on event-triggered control during first few days of my project. I would also like to thank Professor Paulo Tabuada (UCLA) and Dr. Manuel Mazo Jr. (UCLA) for very fruitful discussion and giving interesting seminar about event-triggered control which gave quite good direction to my thesis. I would also like to thank Christian Larsson and Per Hägg (System Identification Group, ACL, KTH) for discussion on system identification of tower crane. I would like to thank Assistant Professor Patric Jensfelt, Program Director and Teacher, for offering such a great master program which greatly enhanced my engineering knowledge and open many new horizons. I am highly thankful to National University of Sciences and Technology, Pakistan, for supporting my master studies in Sweden. I would also like to thank Professor Mojeb-bin-Ehsaan (NUST, Pakistan) and his research group for giving me ample opportunity to enhance my practical engineering design skills that helped me a lot during my graduate studies.

I would also like to thank all my sincere friends who have always been a source of great motivation and encouragement for me.

Last but not least, I feel blessed to have a loving wife, Mehmoona, who was patient, kind and cooperative during last two years despite of my irregular and heavily loaded routines. She has always been a source of great motivation to generate positive energy inside me whenever I was feeling down and depressed.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Statement and Goals	2
1.3	Major Contributions	3
1.4	Outline	4
2	System Identification of Tower Crane	5
2.1	System Description and Main Components	6
2.2	Crane Model Structure Selection	9
2.3	Physical Modeling of DC Servo-Subsystem	12
2.3.1	KLT Model for Servo-Subsystem of Trolley-Subsystem . .	13
2.3.2	KLT Model for Servo-Subsystem of Arm-Subsystem . .	15
2.4	Estimation of Lumped-KLT Parameters of Servo-Subsystems . .	16
2.4.1	Step Response Method	17
2.4.2	Spectral Analysis using Chirp	18
2.4.3	Parametric Estimation of KLT using PEM	20
2.5	Model Validation of Servo-Subsystems	28
2.5.1	Test 1- Running Model and System in Parallel:	28
2.5.2	Test 2- Residual Stochastic Analysis:	28
2.5.3	Test 3- Residual Bode Analysis:	31
2.5.4	Test 4- Model-Based Control Design:	32
2.5.5	Section Summary	33
2.6	Constraints-Driven Estimation of all Sub-Parameters Contained in Lumped-KLT	34
2.6.1	Identification of Sub-Parameters for Trolley Subsystem . .	34
2.6.2	Identification of Sub-Parameters for Arm Subsystem . .	36
2.7	Complete Crane Model	38
2.8	Complete Crane Model Validation	41
2.9	Model Deficiencies and Improvement	42
3	Controller Design: Continuous-Time Approach	45
3.1	Pre-Control Study: State-Space Analysis	46
3.1.1	Conversion to State-Space Form	46
3.1.2	Controllability and Observability:	48
3.1.3	Section Summary	49
3.2	Control Design: Integral Full-State Feedback Controller	50
3.2.1	Control System Performance Specifications	50
3.2.2	Robust Tracking and Disturbance Rejection	50

3.2.3	Gain-Scheduling Controller: Pole Placement using Controllable Canonical Form	52
3.2.4	Robust Fixed-Gain Controller using Linearization	60
3.2.5	Section Summary	62
3.3	Simulation Results and Analysis	65
3.3.1	Model M_2	65
3.3.2	Model M_1	65
3.4	Chapter Summary	67
4	Controller Design: Event-Triggered Sampling	75
4.1	Event-Triggered Control Theory	76
4.1.1	Lyapunov Input-to-State Stability	76
4.1.2	Regulation Control Problem	78
4.1.3	Tracking Control Problem	79
4.1.4	Computation of Minimum Inter-execution Time	81
4.1.5	Event-Triggered Control Research Problems	82
4.1.6	Section Summary	83
4.2	Feasibility Analysis of Event-Triggered Control For the 3DOF Crane	83
4.2.1	Linearization of the Crane System	83
4.2.2	Computation of σ , $\dot{\sigma}$ and τ_{min} :	85
4.2.3	Section Summary	85
4.3	Implementation Details, Issues and Improvements	85
4.3.1	Implementation Issues	85
4.3.2	Deficiencies and Improvement of Event-Generation Rule .	88
4.3.3	Parametric Tuning of Event-Generation Rule	89
4.3.4	Event Detection Circuit	90
4.3.5	Control of Single Crane over WSAN	91
4.3.6	Control of Multiple Cranes over WSAN	92
4.3.7	Section Summary	92
4.4	True-Time Simulation Results and Analysis	94
4.4.1	Simulation of Control of a Single Crane over IEEE 802.15.4	95
4.4.2	Simulation of Control of Multiple Cranes over IEEE 802.15.4	95
4.5	Chapter Summary	96
5	Performance Evaluation and Comparative Analysis	103
5.1	Derivation of Performance Criterion for Comparison	103
5.1.1	Running Cost Function:	103
5.1.2	Control Action Cost Function:	104
5.1.3	Network Cost:	104
5.1.4	Complete Network Control Cost:	105
5.1.5	Main Objective of Event-Triggered Controller:	105
5.2	Single Crane Case: Performance Comparison and Analysis	105
5.2.1	Performance Analysis under Zero Delays	106
5.2.2	Performance Analysis under Network Induced Delays . .	106
5.2.3	Section Summary	108
5.3	Multiple Crane Case: Performance Analysis	109
5.3.1	Performance Evaluation without MAC Protocol	109
5.3.2	Scalability Properties of ETC and MAC Protocol	110
5.3.3	Section Summary	111

CONTENTS	ix
5.4 Chapter Summary	111
6 Summary and Conclusion	123
7 Future Work	127
A DC Servo-Subsystem	129
A.1 Physical Modeling of DC Servo-Subsystem	129
A.1.1 Mechanical Load Dynamic Equation	130
A.1.2 Electrical Circuit Dynamic Equation	131
A.1.3 Transfer Function and Simplification of DC-Servo System	132
References	135

List of Tables

2.1	Parameters Of DC Servo Subsystem	13
2.2	Identified System Parameters	40
2.3	Comparison between States in Model M2 and Laboratory System	41
4.1	IEEE 802.15.4 Network parameter values and delays used in the simulation.	94
5.1	IEEE 802.15.4 Network parameter values and delays used in the simulation.	106

List of Figures

1.1	Problem Statement	3
2.1	Steps in System Identification Approach.	6
2.2	Tower Crane Setup and System Description. Y-axis is out of plane	7
2.3	Definitions of variables adopted for tower crane system.	8
2.4	Block Diagram of 3D Tower Crane with the layout and interconnection of two subsystems inside. The A , B and C will be described in chapter 3.	11
2.5	Step Response for Trolley Servo-Subsystem.	18
2.6	Step Response for Arm Servo-Subsystem. Blue shows response before filtering and red after filtering.	19
2.7	Spectral Estimation of frequency function Trolley Servo according to SPA procedure. Top: Magnitude Response Bottom: Phase Response.	19
2.8	Spectral Estimation of frequency function of Arm Servo according to SPA procedure. Top: Magnitude Response Bottom: Phase Response.	20
2.9	Estimation Data for Identification of Trolley Servo. Top: Trolley Output Speed. Bottom: Identification Input. Blue shows original data and red shows filtered data	27
2.10	Estimation Data for Identification of Arm Servo. Top: Trolley Output Speed. Bottom: Identification Input. Blue shows original data and red shows filtered data	27
2.11	Validation Test 1 for Trolley Servo. Model Validation by simulating the identified model for validation input data and comparing with validation data output from real system. Note that Trolley servo subsystem has very small backlash in gears, that's why here we see almost no oscillations except some quantization noise.	29
2.12	Validation Test 1 for Arm Servo. Model Validation by simulating the identified model for validation input data and comparing with validation data output from real system. Real Arm servo has some oscillatory behavior as well which our identified model hasn't captured. The reason for oscillation in arm servo is relatively large backlash in gears which we didn't consider during our identification process.	29

2.13 Validation Test 2 for Trolley Servo:Residual Analysis - Auto-correlation of residuals and Cross-correlation bet. residuals and past inputs for trolley servo subsystem. The above plot shows the acf of residues $\varepsilon = y - \hat{y}$ for trolley servo position $y = X_w$. Ideally residues should have white noise properties but here we see that correlation exists between different residue samples. It just shows that the output could have been predicted from last input but here the main reason for this correlation is up-sampling of original sampled data that we did to increase the number of data samples during rise time. The lower plot shows the cross-correlation between input and residues. This plot says that input has excited all the modes of the trolley servo system well and almost nothing has gone into residues. 30

2.14 Validation Test 2 for Trolley Servo:Residual Analysis - Auto-correlation of residuals and Cross-correlation bet. residuals and past inputs for arm servo subsystem. The above plot shows the acf of residues $\varepsilon = y - \hat{y}$ for arm servo position $y = \theta$. Ideally residues should have white noise properties but here we see that correlation exists between different residue samples. Note that this acf is a lot better than that for trolley due to no up-sampling for arm servo as it was not required. The lower plot shows the cross-correlation between input and residues. This plot says that input has excited all the modes of the arm servo system well and almost nothing has gone into residues. 31

2.15 Validation Test 3 for Trolley Servo: Residual Dynamics or Model-Error Model. Amplitude response of residue model is quite close to zero and it shows that almost nothing has been missed by our identified in terms of gains. The phase response of residual model has significant phase response for input frequency $f > 0.7 \text{ Hz}$. It shows that our estimated model for trolley has missed some phase properties of the real system and for fast changing input $f > 0.7 \text{ Hz}$ it may not predict the correct phase information of trolley position X_w . So we can say our model is good for input frequencies $f_{in} \in [0, 0.7] \text{ Hz}$ 32

2.16 Validation Test 3 for Arm Servo: Validation Test 3 for Arm Servo: Residual Dynamics or Model-Error Model. Amplitude response of residue model is quite close to zero and it shows that almost nothing has been missed by our identified model in terms of gains. The phase response of residual model is also acceptable for input frequencies $f \leq 1 \text{ Hz}$ but for $f > 1 \text{ Hz}$ there is a drastic increase in phase response of residual model which shows that our estimated model for arm has missed phase properties of the real arm servo system for fast changing input $f > 1 \text{ Hz}$ so it may not predict the correct phase information of arm position θ . So we can say our model is good for input frequencies $f_{in} \in [0, 1] \text{ Hz}$ 33

2.17 Model Validation for the customized signal. Top: Trolley Position, X_w . Bottom: Arm Position, θ . Blue: Input Signal, Green: Actual System Response. Red: Model Response. 42

2.18 Model Validation for the customized signal. Top: Payload In-Plane Angle, $-\alpha$. Bottom: Payload Out-of-Plane Angle, β . Blue: Actual System Response, Red: Model Response.	43
3.1 Block Diagram of 3D Tower Crane with the layout and interconnection of two subsystems inside. Note that C_{11} is the 1 st row of C_1	48
3.2 3D Plot of Actual K_{21} function and its Estimate \hat{K}_{21} computed using Surface Fitting Tool.	56
3.3 3D Plot of Actual K_{22} function and its Estimate \hat{K}_{22} computed using Surface Fitting.	57
3.4 3D Plot of Actual K_{23} function and its Estimate \hat{K}_{23} computed using Surface Fitting Tool. Root mean square error for this fit is high and hence the improvement is required.	58
3.5 3D Plot of Actual K_{24} function and its Estimate \hat{K}_{24} computed using Surface Fitting. Root mean square error for this fit is high and hence the improvement is required.	59
3.6 3D Plot of Actual K_{25} function and its Estimate \hat{K}_{25} computed using Surface Fitting Tool	60
3.7 Plot of coefficients of the characteristic polynomial of the closed-loop trolley subsystem. Blue- For Fixed-Gain Controller, Red- For Gain-Scheduling Controller. Plot shows that all coefficients are greater than 0 and also except for α_{cs12} all other coefficients are quite closely matched in the region $L \in [0.1, 0.8]$. Thus according to the plots the closed-loop system under fixed-gain controller is stable $\forall L \in [0, 0.8]$ and would achieve the performance quite similar to the gain-scheduling controller at least for $L \in [0.1, 0.8]$	63
3.8 Plot of coefficients of the characteristic polynomial of the closed-loop Arm subsystem using fixed gain. The plot for the coefficients of characteristic equation for gain-scheduling controller are not shown as the coefficients computation is very messy which could not be done even with the help of Symbolic Math toolbox. So the above plots only show that for the fixed-gain controller (3.2.43) the closed loop system is stable for $L \in [0.1, 0.8]$ and $X_w \in [0, 0.5]$	64
3.9 Block Diagram of Integral Control Scheme of 3D Tower Crane. The integral anti-windup scheme has also been used. Everything inside the dashed-dotted box constitutes the Integral Controller and inside the dotted box are the arm and trolley subsystems.	66
3.10 Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.6\text{m}$. Both Inputs excited Simultaneously. The fixed-gain controller has been used which achieves all the control requirements mentioned in section 3.2.1. See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1.	69
3.11 Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.1\text{m}$. Both Input excited Simultaneously. Fixed-Gain controller works quite well for sling lengths as short as 0.1m.	70

3.12 Payload angles response of crane's model \mathbf{M}_2 for various sling length $L \in [0.1, 0.8]$. Reference Input is Step. We can see that for $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ the control system behave nicely by killing the payload oscillations so the proposed fixed-gain controller works quite well at least in the region $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ and brings the payload oscillation to the ball \mathcal{B}_{r_1} of radius $r_1 = 0.005\text{m}$ at maximum within two periods of oscillations and also never leaves the ball \mathcal{B}_{r_2} of radius $r_2 = 0.05\text{m}$ during the whole move.	70
3.13 Triangular Input Response of Crane's model \mathbf{M}_2 for sling length $L = 0.6\text{m}$. Both Inputs are excited Simultaneously. The simulation has been done to evaluate the performance of the closed-loop control system in comparison to open-loop response. We are especially interested to see the efficiency of controller to kill the payload oscillation shown in the subfigure 3.13(b).	71
3.14 Step Response for sling length $L = 0.6\text{m}$. Both Input excited Simultaneously. The above plot shows that there is almost no difference between control system performance of both models so the controller designed on the basis of identified model M_2 also works perfectly for the complete nonlinear model \mathbf{M}_1 and since the \mathbf{M}_1 is a quite close description of the real system so it is quite likely that the same controller should also work quite well for the real system.	72
3.15 Payload angles response for various sling length $L \in [0.1, 0.8]$. Reference Input is Step. We can see that for $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ the control system behave nicely by killing the payload oscillations so the designed controller works quite well at least in the region $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ and drives the payload to the ball \mathcal{B}_{r_1} of radius $r_1 = 0.005$ at maximum within two periods of oscillations.	73
4.1 Evolution of ϕ for $\Delta \neq 0$. The annotation shows the values computed for $\dot{\sigma}_{min}$ and $\dot{\sigma}$	86
4.2 Block Diagram of Event Detector Circuit. There are two modules inside this circuit namely the Integrator Module (\mathcal{IM}) and Event Generation Module (\mathcal{EGM}). The \mathcal{IM} implements the dynamical part of the controller that is the integral part for our controller. If there is no dynamics inside the controller then this circuit can be bypassed. The \mathcal{EGM} implements the event triggering rule given by (4.3.8) It generates the trigger whenever rule is violated. Note that this event detection circuit can be implemented using dedicated analog and/or digital hardware. The full practical details for implementation of this circuit are not obvious yet and beyond the scope of this thesis.	91
4.3 Block Diagram of Event-Triggered Control Scheme.	92
4.4 Block Diagram of Event-Triggered Control Scheme for Two cranes. One Controller is controlling two cranes. Communication is half duplex. Only one crane can communicate with controller at a time. Controller updates only one plant actuator at a time.	93

4.5 IEEE 802.15.4 unslotted CSMA/CA algorithm with non-beacon mode.	95
4.6 TrueTime Implementation of Event-Triggered Control System for Single Crane. The TrueTime network block has been used to simulate the wireless channel that is being accessed using IEEE802.15.4 protocol.	97
4.7 Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.6\text{m}$. Both Inputs excited Simultaneously. The fixed-gain event-triggered integral controller has been used which achieves all the control requirements mentioned in section 3.2.1. See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1.	98
4.8 Schedules for Network and Controller Tasks. Note that schedules are heavy initially but then they become relaxed. Also note that how in just 60 controller updates the control system has achieved almost same performance which is achieved through 250 controller updates in case of periodic sampling. So the Event-triggered control completely outperforms the periodic time-triggered controller owing to less network usage.	99
4.9 The evolution of event generation rule. The plot shows how the error norm \mathbf{e} evolves and when it just crosses $\sigma \ \mathbf{W}(\mathbf{X} - \mathbf{X}^*)\ + \delta$ the event is generated.	99
4.10 TrueTime Implementation of Event-Triggered Control System for multiple cranes. Three TrueTime kernels have been used for simulating sensor nodes and controller nodes. One TrueTime network block has been used to simulate the IEEE802.15.4 shared wireless network. The network settings used are shown in the table 4.1.	100
4.11 Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.6\text{m}$. Both Inputs excited Simultaneously. The one event-triggered controller is controlling two crane processes. Controller can communicate only with one crane at a time. We can see that all the control requirements mentioned in section 3.2.1 have been achieved. See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1.	101
4.12 Schedules for Network and Controller Tasks. Note that schedules are heavy initially but then become relaxed. Also note that how in just 59 controller updates the control system has achieved almost same performance which was achieved through 250 controller updates in case of periodic sampling. So the Event-triggered control completely outperforms the periodic time-triggered controller owing to less network usage.	102
5.1 Block Diagram illustrating different kind of delays in the system.	107
5.2 Step Response of crane under event-triggered and periodic control policy. Both meet all the control requirements mentioned in section 3.2.1 with almost similar performance but ETC achieves this in almost 4 times less controller updates. Both controllers have kept the payload inside the ball \mathcal{B}_{r_1} throughout the whole move and also driven it to the ball \mathcal{B}_{r_2} at steady-state. See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1.	112

- 5.3 Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of controller-to-actuator delay Δ_{ca} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. We can see that up to $\Delta_{ca} \approx 0.170$, ETC dominates TTC but then ETC degrades and lost its benefit of less network usage. 113
- 5.4 Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of sensor-to-controller delay Δ_{sc} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. P_r is almost the same as for Δ_{ca} but now P_n degrades below that of periodic at higher level of delay. 114
- 5.5 Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of delay inside event-detector circuit Δ_{eds} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. The running performance under Δ_{eds} is almost the same as for Δ_{ca} and Δ_{sc} but now the network performance under ETC degrades below that of periodic at much higher level of Δ_{eds} 115
- 5.6 Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of delay Δ_{ped} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. Both Network and control performance under ETC and TTC degrades with delay but now the rate of degradation is lower. The network control system can tolerate higher level of Δ_{ped} delay as compared to Δ_{sc} and Δ_{ca} . . 116
- 5.7 Multiple Crane Case with Packet losses: Event-Triggered Controller Performance for 3 Cranes under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode without back-offs. We can see that all cranes have satisfactory performance under ETC even when no packet retransmission is enabled. The main reason for small number of collision is that ETC itself introduces some randomness in packet transmission which reduces probability of collision. 117
- 5.8 Packet Loss Probability in the absence of MAC protocol: Performance of network under Event-Triggered Control policy as a function of number of cranes with IEEE 802.15.4 wireless network settings using unslotted CSMA/CA MAC protocol in non-beaconed mode without back-offs (network parameter macMaxCSMABackoffs = 0). The graphs show that when the cranes are not allowed to retransmit the packet then the network control system breaks down for 7 number of cranes in the network. The dash-dot red line is the maximum allowed packet loss probability above which control performance is not acceptable. 118

- 5.9 Step Response of 7 Cranes under Event-triggered control policy using IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode without back-offs (network parameter $\text{macMaxCSMABackoffs} = 0$). Although packet loss probability of crane-1 > crane-6 > crane-5 but still crane-5 has worst performance among all. The reason is the higher number of packet losses during transient phase of crane-5. 119
- 5.10 Scalability Properties: Performance of Event-Triggered Controller (ETC) w.r.t. number of cranes under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in non-beaconed mode with (network parameter $\text{macMaxCSMABackoffs} \neq 0$) and without (network parameter $\text{macMaxCSMABackoffs} = 0$) back-offs. The graph shows that Event-triggered controller has better scalability properties when transmission back-off mechanism is employed. NCS under event-triggered controller with zero back-off breaks down for 13 cranes but when back-off is enabled the NCS performance is restored. 120
- 5.11 Packet Loss Probability under MAC protocol: Performance of Event-Triggered Controller (ETC) w.r.t. number of cranes under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs (network parameter $\text{macMaxCSMABackoffs} \neq 0$). The graph shows that MAC protocol has significantly increased the network performance. The dash-dot red line is the maximum allowed packet loss probability above which control performance is not acceptable. 121

Chapter 1

Introduction

The 3D Tower Cranes have widespread use in construction industry. They have three degrees of freedom which gives it capability to independently translate, rotate and elevate the heavy building blocks during the construction of, especially, tall buildings. Any tower crane is basically built of three main parts:

Base is bolted to large heavy concrete slab to support the whole crane.

Tower gives height to the crane and normally has to grow with the building.

Slewing Unit is attached to the top of tower, which consists of motor and gear assembly and enable the crane to rotate at the tower's axis.

The slewing unit consists of four more parts which are:

Arm or Jib is a long beam like structure which extends out horizontally from crane's center.

Trolley runs along the arm and translates the load. The whole arm along with trolley can rotate around tower's axis.

Machinery Arm is a short horizontal beam, which extends out opposite to the long arm, contains the counter weight to balance the bending moment on crane due to load. The machinery arm also contains motor, gear assembly and drive electronics.

Operator Cabin It is attached to the side of slewing unit and is the place where the operator has all the manual controls.

1.1 Motivation

The tower cranes are mostly manually operated to move very heavy loads which requires quite smooth operation to ensure the safety on site. So this is a great motivation for automatic control of tower cranes to ensure greater safety, lower wear and tear of parts, fast, reliable and repeatable operation. Moreover, the tower cranes has quite messy cable harness going up with them and since the height of crane has to grow with the building as well so managing this cable harness becomes a big mess which also pose high maintenance cost and down

time due to greater risk of cable break downs. Moreover rewiring the whole device each time for different height requirements makes the installation process very tedious. So greater flexibility and ease of installation is a great motivation for complete wireless control of crane in a way that we do both wireless sensing and actuation. The wireless control provides greater flexibility with very low installation time and easy modifications. It also greatly reduces the wire harness mess which means less breakdowns, lower maintenance time and hence reduces the maintenance and operational cost. But employing wireless control introduces some new issues that relates to safety and robustness due to reduced data integrity on wireless channel. Problems like packet losses and time-varying delays makes the system prone to instability and hence the robustness may be compromised which raise many safety concerns. So to evaluate the effect of packet losses and delays we are doing control of multiple cranes which compete for the shared network resource. Another problem is related to power utilization. For true wireless operation, stand-alone battery at the point of load will be required. This means for long operation, power conservation and greater efficiency is quite critical. Other than the power required for crane itself the wireless motes will pose some communication cost as well. We cannot do much about the power used by the crane but to save the communication energy we aim to devise a control policy which reduces the communication between the remote controller and a crane. This is the motivation for the event-triggered control which is quite an exciting and hot area especially for applications involving hardware having low computational capabilities and limited communication bandwidth. In such applications classical controller implementation techniques based on periodic sampling of a process are not optimal as they are normally very expensive due to unnecessary periodic execution of control even when it is not required. Also, the sampling frequency in periodic case is something which is not easy to decide as there is no hard and fast rule on how fast to sample. Normally people resort to some rule of thumbs but there is no sound theoretical reasoning behind them. So to optimally use the limited computational resource or the bandwidth of communication medium the efficient way is to sample only when something significant (event) happen in the process. Normally the event is triggered when the error in states crosses some certain threshold which depends upon the system dynamics and the controller structure. This technique has been especially explored over last few years [2, 9, 10, 11] for the control applications involving sensor and actuator networks. The applications using Wireless Sensors and Actuators Networks (WSAN) are even more interesting to deal with using the event-triggered techniques due to limited bandwidth and energy conservation requirements for WSAN. To the best of our knowledge, the event-triggered control over wireless sensor networks for controlling higher order nonlinear systems under packet losses and delays is being studied for the first time in this thesis.

1.2 Problem Statement and Goals

The control problem solved in this thesis is shown in Figure 1.1. The first focus of the thesis is to identify the dynamic model for the tower crane. The second major focus is the networked control of two or more cranes using single event-triggered controller. The controller can only measure and actuate one crane at

a time which is the usual case in a wireless network. Each crane has the sensor/actuator node which sends the state information to remote controller node whenever state-based event occurs. Controller gets the state information and update the controller output and transmit it over the IEEE 802.15.4 network to the corresponding actuator node. So there is a network between sensor-to-controller and controller-to-actuator. We need to suggest an Event-Triggered Control (ETC) scheme which outperforms the Periodic Time-Triggered Control (TTC) in terms of overall network control performance. We would also investigate how the event-triggered control scheme scales with the increase in number of systems to be controlled and would compare it with that of time-triggered case. So we are aiming to achieve following goals in our thesis:

1. Develop dynamic Model of Tower Crane
2. Design model-based Event-Triggered Controller (ETC) to reduce utilization of shared network resource, asymptotically track step reference and damp payload oscillation.
3. Performance Evaluation of ETC under packet dropouts, delays and with respect to scalability properties.
4. Performance comparison with periodic time-triggered controller (TTC)
5. Evaluation of MAC Protocol for stability under event-triggered control prolicity.

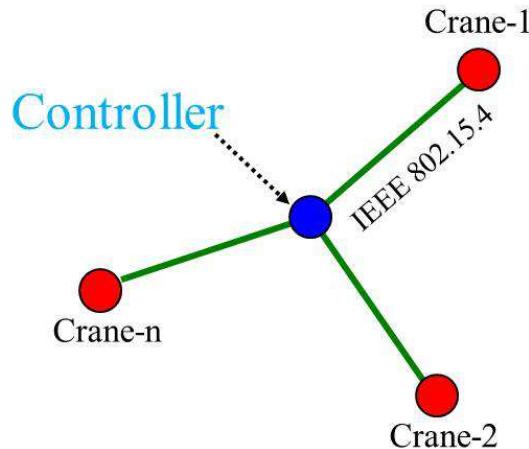


Figure 1.1: Problem Statement

1.3 Major Contributions

Major contributions of our work includes:

1. Developed the general scheme for identifying the dynamic model of 3DOF tower cranes. The controller designed on the basis of identified model works quite well for the actual system also.

2. Designed the Gain-Scheduled continuous-time Integral controller using LPV (Linear Parameter Varying) system description for 3DOF tower crane.
3. Designed the Fixed-Gain CT controller that achieves the semi-global asymptotic tracking.
4. Designed the Event-Triggered Controller for both regulation and tracking problem using Integral control structure.
5. Addressed some implementation issues related to dynamic controller and tracking under event-triggered sampling.
6. Suggested some improvements to tune the event-generation rule to address the conservativeness.
7. Developed the complete Simulink® model using TrueTime toolbox for networked control of multiple tower cranes.
8. Numerical performance analysis of event-triggered control under network induced delays and thorough comparison with periodic controller.
9. Numerical performance evaluation of event-triggered controller under packet dropouts for multiple crane case
10. Evaluated the effect of MAC protocol on the stability under event-triggered control policy

1.4 Outline

The work in this thesis can be roughly divided into following five phases:

1. System Identification of Crane (Chapter 2)
2. System analysis and design of continuous-time controllers (Chapter 3)
3. Theory and design of Event-Triggered controller (Chapter 4)
4. True-time implementation of event-triggered controller for multiple cranes over WSAN (Chapter 4)
5. Performance Evaluation and comparative analysis (Chapter 5)

Chapter 2

System Identification of Tower Crane

The laboratory model of 3D tower crane shown in Figure 2.2 is a 8th¹ order multi-input and multiple-output (MIMO) system with highly nonlinear dynamics and strong coupling between various inputs and outputs. The laboratory crane system under consideration lacks the dynamic model so the first goal is to come up with simple but reasonable mathematical model which is also necessary to conduct any theoretical study for event-triggered control. The dynamic model of the tower crane has been obtained after processing through five major steps as shown in Figure 2.1. A general parametric dynamic model structure proposed by Omar Hanafy in his PhD thesis [5] for similar kind of crane systems has been selected. But before estimation of model parameters, we improved the model for actuators. Omar has modeled actuators as single integrators with gain whereas we have used KLT² model instead. After this improvement, we split the system into several subsystems and then decoupling them, we identified each subsystem separately. Gray-box system identification techniques along with physical modeling have been used to identify the parameters of the improved model. The main trick used to identify all important system parameters like inertia, masses and motor constant etc is to capture all of them into a set of two parameters for each axis by reflecting corresponding system parameters onto the respective actuator shafts and then modeling actuator as first order transfer functions (KLT) during physical modeling phase. System identification phase mainly involves the parametric estimation through **prediction-error-method** (PEM) using **tailor-made model** but it is also preceded by **Transient-Analysis** and **Spectral-Analysis**. Although spectral analysis doesn't reveal much about nonlinear systems but it can be applied on the parts of system which are considered linear. Here the spectral analysis has been done to figure out the bandwidth of system's actuators which was then used to design input for parametric estimation phase. So using the combination of least-squares, transient-analysis and spectral analysis two parameters

¹Here we are not considering the z-axis actuator dynamics that is the model for load lift-line motor is not included and the lift-line length L will be considered constant during the move.

²First order transfer function with Gain (K), time constant (T) and delay term (L) is called KLT model

in each transfer function have been estimated. Each of the two parameters is actually a **lumped-parameter** consisting set of many other **sub-parameters**. Each sub-parameter inside each lumped-parameter has been identified using set of equations defining all the actuator constraints for DC motor and then the constrained system of equations was solved to find the value for each sub-parameter and then these parameters have been used inside the selected crane model. The model has been verified using residual covariance, Bode analysis and model-based control design and these tests revealed quite good results.

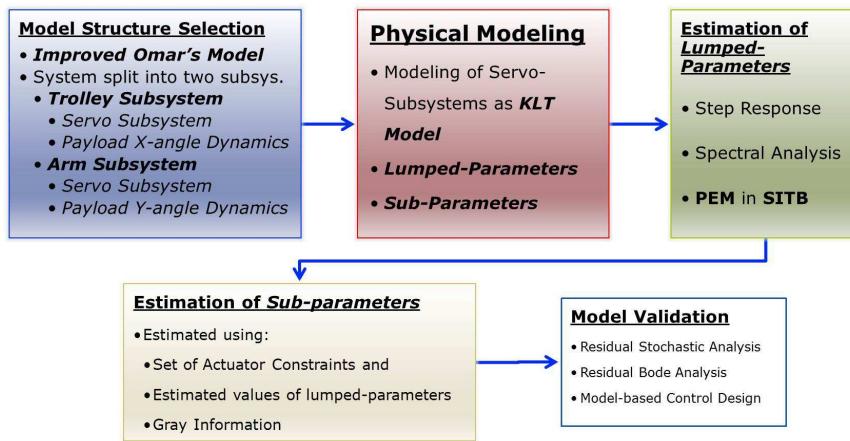


Figure 2.1: Steps in System Identification Approach.

2.1 System Description and Main Components

The tower crane is a nonlinear and multi-variable system with oscillatory behavior. The laboratory model of our 3D Tower Crane has following main parts:

1. Trolley
2. Trolley Motor or X-Drive
3. Arm or Jib
4. Arm Motor or T-Drive
5. Payload
6. Payload Motor or Z-axis Drive
7. Counter Weight

The setup of our laboratory tower crane system is shown in the Figure 2.2 and it consists of a payload hanging on a pendulum like lift-line wound by a DC motor mounted on a trolley. The payload is lifted and lowered in the z-direction. Both the arm and trolley are capable of horizontal motion i.e. the trolley in

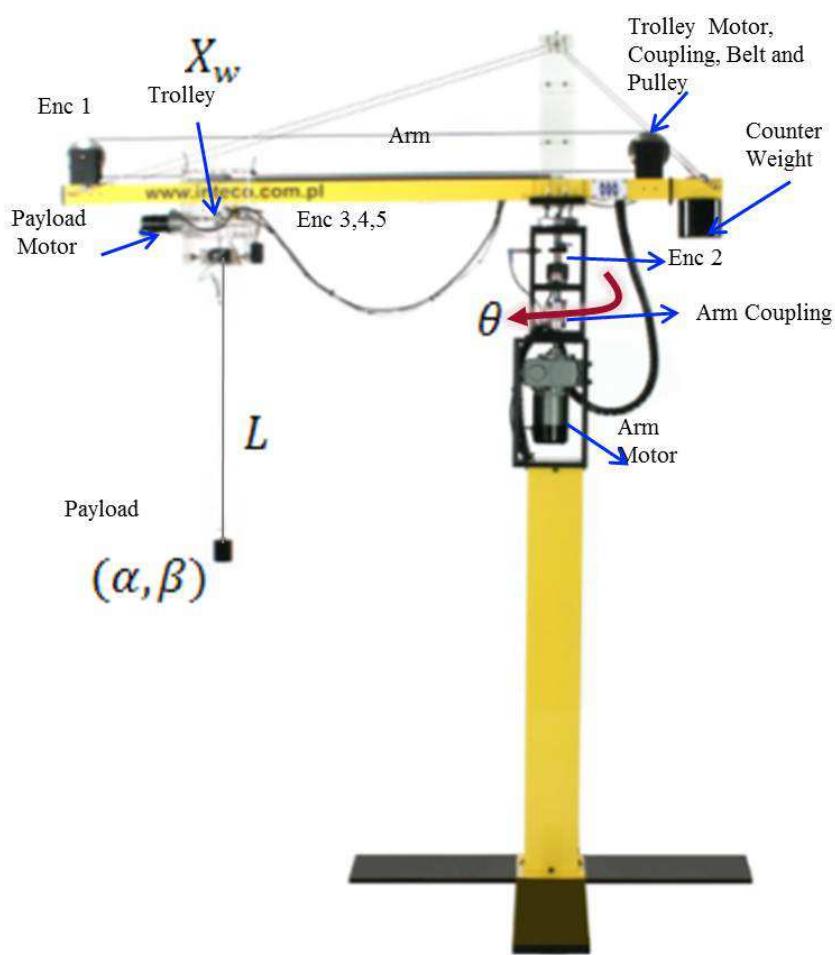


Figure 2.2: Tower Crane Setup and System Description. Y-axis is out of plane

the radial X direction along the arm and the arm in the rotary direction. The payload attached to the end of the lift-line can move freely in three dimensions [1]. The tower crane is driven by three DC-motors and outputs are measured by five incremental optical encoders. It has three control inputs with DC motors as actuators:

1. Input Voltage to translate trolley on arm
2. Input Voltage to rotate the arm
3. Input Voltage to lift the payload

It has five outputs being measured by quadrature incremental optical encoders. These output variables ³ are shown in the Figure 2.3 and are defined as follows:

1. The angle of arm θ
2. Position of trolley on Arm, X_w
3. length of Payload string, L
4. Swing angle of load in XZ-plane, α . It is the angle between the projection of lift-line onto YZ-plane and the lift-line.
5. Swing angle of load in YZ-plane, β . It is the angle between the projection of lift-line on to YZ-plane and the z-axis.

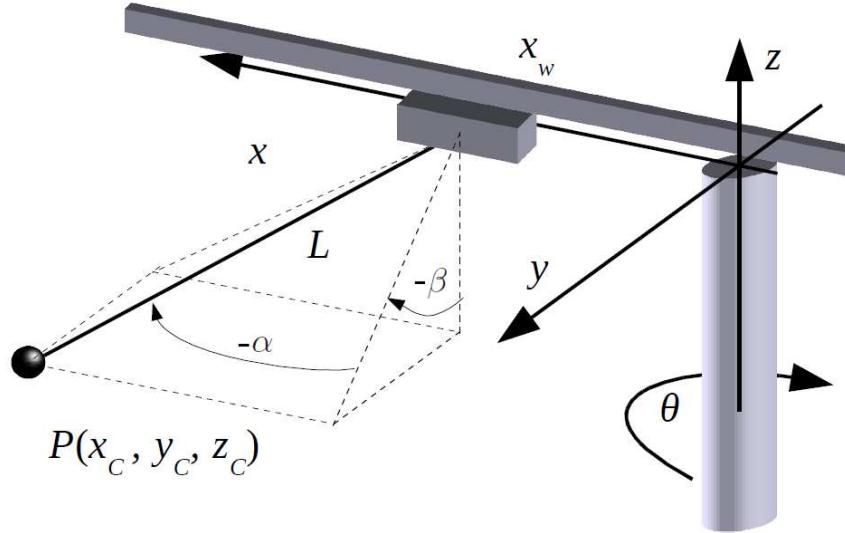


Figure 2.3: Definitions of variables adopted for tower crane system.

³Please note that all state variables defined here are equivalent to those in [1] except β and θ . The angle β here is equal to $-\beta$ in [1] and similarly θ is equal to $-\theta$ in [1]

2.2 Crane Model Structure Selection

There are many readily available model structures proposed by various people for 3D tower cranes but we are interested in parameterized dynamic model. Although in the user manual of the crane [1] a mathematical model has been proposed but it has following drawbacks:

1. It is the kinematic model derived using geometry and trigonometric rules and that is why it is far more complex owing to a lot of sine and cosine terms and not suitable for control design purpose.
2. Another problem with that model is that it doesn't consider the mass properties of the system which can, of course, vary from system to system.
3. Moreover the dynamics of actuators has not been considered and hence acceleration has been considered as input which actually is not the case as on lowest control level armature voltage is the only manipulating variable.
4. Although in the simulink the model has been modified to include the effect of actuators, it is still not good enough as the motors have been modeled as constant gains and dominant pole dynamics has been mimicked by using rate limiters which is problematic itself as it doesn't allow the step inputs or any other fast rising inputs to pass through.

So the aim is to select a dynamic model which not only contains the system physical properties like mass, inertia, radius etc but also captures the actuator dynamics at least as first order transfer function. So as per the above stated requirements and intent of use for control design purpose the simplified parametric model of tower crane proposed by Omar Hanafy in his PhD thesis [5] is given by⁴:

$$\ddot{X}_w + m_t g \alpha = K_{mx} v_{ax} \quad (2.2.1)$$

$$L\ddot{\alpha} + g\alpha - \ddot{X}_w = 0 \quad (2.2.2)$$

$$(1 + M_r X_w^2) \ddot{\theta} - m_r g X_w \beta = K_{my} v_{a\theta} \quad (2.2.3)$$

$$L\ddot{\beta} + g\beta + x\ddot{\theta} = 0 \quad (2.2.4)$$

Now onwards the model (2.2.1)-(2.2.4) will be referred as **Omar's Model** [5]. But here it is important to emphasize that in the above model dc motors have been modeled as single integrator with constant gains between voltage input and output speed. Note trolley servo-subsystem (2.2.1) in the absence of payload becomes double integrator and similarly arm servo-subsystem (2.2.3) in the absence of payload and trolley also reduces to double integrator. The acceleration components like $m_t g \alpha$, $M_r X_w^2 \ddot{\theta}$ and $m_r g X_w \beta$ that come due to payload and trolley can be considered as *disturbance or coupling terms* which have been added to double integrators to give full dynamics as shown in (2.2.1) and (2.2.3). Now instead of double integrator we aim to give better model of trolley and arm servo-subsystems. So our first goal of modeling is to improve

⁴Please note that all state variables defined in the model (2.2.1)-(2.2.4) are equivalent to those in [1] except β and θ . The angle β here is equal to $-\beta$ in [1] and similarly θ is equal to $-\theta$ in [1]

the model of motors in the **Omar's model** and hence using **KLT** model from voltage input to output speed, we propose⁵ following improved model structure:

$$\ddot{X}_w + \left(\frac{1}{\tau_x}\right)\dot{X}_w(t) + m_t g \alpha = K_{mx} v_{ax}(t - \tau_{d_x}) \quad (2.2.5)$$

$$L\ddot{\alpha} + g\alpha - \ddot{X}_w = 0 \quad (2.2.6)$$

$$(1 + M_r X_w^2) \ddot{\theta}(t) + \left(\frac{1}{\tau_y}\right) \dot{\theta}(t) - m_r g X_w \beta = K_{my} v_{a\theta}(t - \tau_{d_y}) \quad (2.2.7)$$

$$L\ddot{\beta} + g\beta + X_w \ddot{\theta} = 0 \quad (2.2.8)$$

where:

- m = Mass of payload = 0.32 kg (provided by manufacturer)
- M_t = Mass of trolley = 0.7 kg (provided by manufacturer)
- M_{eq} = Equivalent mass of trolley sub-system seen at the output shaft of gearhead of motor M_1
- $m_t = \frac{m}{M_{eq}}$
- J_{leg} = Moment of inertia of the tower and arm (excluding trolley) about the z-axis seen at the output shaft of gearhead of motor M_2
- $m_r = \frac{m}{J_{leg}}$
- $M_r = \frac{M_t}{J_{leg}}$
- K_{mx} = Acceleration Gain for Trolley Servo
- K_{my} = Acceleration Gain for Arm Servo
- v_{ax} = Voltage input to dc motor for moving trolley
- $v_{a\theta}$ = Voltage input to dc motor for rotating arm
- τ_x = Time Constant of Trolley Servo Subsystem
- τ_y = Time Constant of Arm Servo Subsystem

According to the above description we need to find values of parameters M_{eq} , J_{leg} , K_{mx} , K_{my} , τ_x , and τ_y to fully characterize our tower crane system. Now for ease of identification, the crane can be divided mainly into two sub-systems each of which contains two more subsystems as shown in the Figure 2.4:

1. **Trolley Sub-system** described by equations (2.2.5)-(2.2.6)

Trolley Servo-Subsystem described by (2.2.5)

Payload X-angle Dynamics described by (2.2.6)

2. **Arm Sub-system** described by (2.2.7)-2.2.8

Arm Servo-Subsystem described by (2.2.7)

Payload Y-angle Dynamics described by (2.2.8)

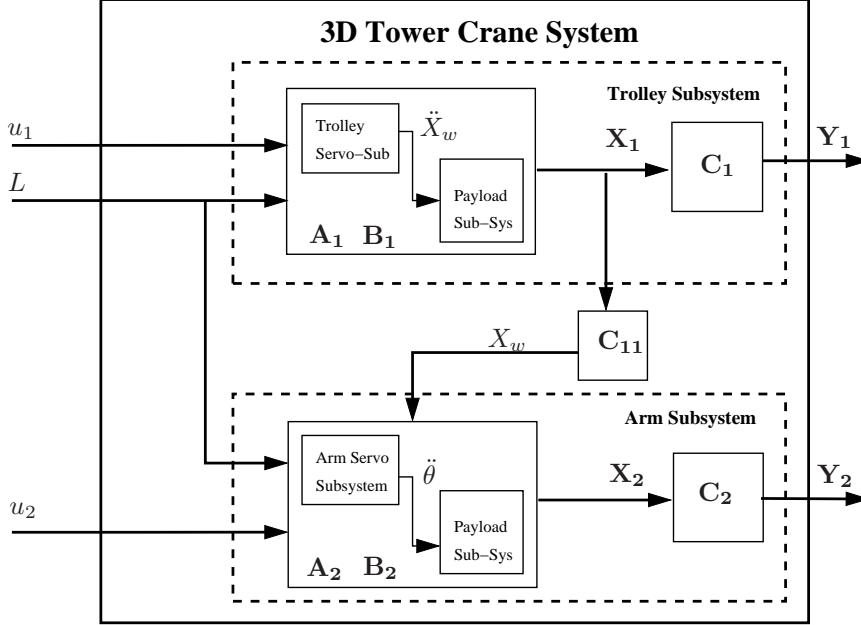


Figure 2.4: Block Diagram of 3D Tower Crane with the layout and interconnection of two subsystems inside. The **A**, **B** and **C** will be described in chapter 3.

Now if we drop the coupling terms like $m_t g\alpha$, $M_r X_w^2 \ddot{\theta}$ and $m_r g X_w \beta$ then the model (2.2.5)-(2.2.8) of tower crane is reduced to:

$$\ddot{X}_w + \left(\frac{1}{\tau_x}\right) \dot{X}_w(t) + m_t g\alpha = K_{mx} v_{a_x}(t - \tau_{d_x}) \quad (2.2.9)$$

$$(1 + M_r X_w^2) \ddot{\theta}(t) + \left(\frac{1}{\tau_y}\right) \dot{\theta}(t) - m_r g X_w \beta = K_{my} v_{a_\theta}(t - \tau_{d_y}) \quad (2.2.10)$$

This simplification through decoupling and splitting system into simpler subsystems has greatly reduced the complexity of identification process of the MIMO nonlinear system like tower crane which otherwise would have been quite complicated. Physically speaking, the decoupling as shown in the model (2.2.9)-(2.2.10) can be achieved if we adopt the following method to log the data during system identification.

1. Remove the payload from crane to make the reaction forces, $m_t g\alpha$ and $m_r g X_w \beta$ due to payload, zero.
2. Excite one input at a time which means the identification data for trolley and arm will be recorded separately in two different experiments.
3. During excitation of arm servo subsystem, freeze the trolley at origin and this will ensure that inertial force, $M_r X_w^2 \ddot{\theta}$, due to trolley movement along arm remains zero.

⁵The motivation for this improved model structure is covered in Section 2.7 where we developed this improved model in a reverse way by incorporating already known coupling terms to the improved KLT model for DC servo subsystems.

2.3 Physical Modeling of DC Servo-Subsystem

In the absence of *coupling terms*, the general but simplified model derivation for any DC Servo system containing dc motor, gearbox, inertial load etc has been described in detail in Appendix A. Here we will restate the derived model (A.1.29) as follows:

$$\frac{\omega_l(s)}{V_a(s)} = \frac{K}{\tau s + 1} \cdot e^{-s\tau_d} \quad (2.3.1)$$

The model (2.3.1) is usually called **KLT Model** with following parameters defined as (See Appendix A):

DC Gain of Complete Servo System:

$$K = \frac{K_g K_T}{R_a b_{l_{eq}} + K_g^2 K_T K_b} \quad (2.3.2)$$

Mechanical Time Constant of Complete Servo System:

$$\tau = \frac{R_a J_{l_{eq}}}{R_a b_{l_{eq}} + K_g^2 K_T K_b} \quad (2.3.3)$$

Where:

$$K_g = \frac{\theta_m}{\theta_l} \quad (2.3.4)$$

$$J_{l_{eq}} = K_g^2 J_m + J_l \quad (2.3.5)$$

$$b_{l_{eq}} = K_g^2 b_m + b_l \quad (2.3.6)$$

The parameters contained in (2.3.2) and (2.3.3) are described in Table 2.1. The model (2.3.1) is called **KLT Model** because it contains three parameters: K-Gain, T-time constant and L-delay. So to approximate the DC-Servo system behavior we need to identify these three parameters: K , τ and τ_d . As we can see from (2.3.2) and (2.3.3) that all system parameters required for the crane model are encoded into these two parameters K and τ . So these two parameters will be called as **lumped-parameters** which contain the information for all other important parameters designated as **sub-parameters** of the system like mass properties. Hence the procedure for our identification is as follows:

- We will first derive the transfer function for trolley servo-subsystem and arm servo-subsystem on the basis of (2.3.1).
- Then we will estimate the lumped parameters K , τ and τ_d of the **KLT Model** for both Trolley and Arm Servo Subsystems.
- Then we will try to use some gray information to reduce the number of unknown parameters in each lumped parameter
- Finally, we will define a set of constraints coming from actuator dynamics and the mechanical components to calculate each parameter inside lumped parameters.

Table 2.1: Parameters Of DC Servo Subsystem

Parameter Description	Notation
Gear Ratio	K_g
Armature Resistance	R_a
Motor Back-EMF Constant ($\frac{V_s}{rad}$)	K_b
Torque Constant ($\frac{N\cdot m}{A}$)	K_T
Time Constant of DC Servo	τ
System Delay	τ_d
Velocity Gain of DC Servo	K
Total Inertia seen by Gearbox Output Shaft (S_2)	$J_{l_{eq}}$
Total damping seen by Gearbox Output shaft (S_2)	$b_{l_{eq}}$
Load inertia seen at S_2	J_l
Inertia of Motor Rotor	J_m
Angular Position of motor	θ_m
Angular Position of load	θ_l
Damping coefficient of load as seen at S_2	b_l
Damping of Motor Rotor	b_m
Armature Voltage/Motor Input Voltage	v_a

2.3.1 KLT Model for Servo-Subsystem of Trolley-Subsystem

Servo-Subsystem for trolley subsystem consists of following components as shown in Figure 2.2:

1. DC Motor
2. Gearbox
3. Couplings
4. Pulleys and Belts
5. Trolley
6. Geared motor for payload sling along with related mechanical construction

The model (2.3.1) describes any DC servo rotational system but the Trolley-Subsystem of the crane has translational motion along the arm so few terms need to be modified for capturing all relevant parameters in this context. So here we would introduce slight modifications. Since trolley translate along the arm so its speed can be written as:

$$v_t(t) = \dot{X}_w = r_x \omega_l \quad (2.3.7)$$

Where:

- v_t = Translational speed of trolley along arm
- ω_l = Rotational speed of gearbox output shaft (S_2) driving trolley
- r_x = radius of pulley attached to S_2 and together with belt it transforms the rotational motion into translational motion
- X_w = Position of trolley on the arm w.r.t the origin.

Determination of relation for Inertia and Damping

The friction due to sliding of trolley along the guide on arm will be ignored in our computations. We will only consider the damping and inertia quantities. The friction will not be analytically analyzed but the dead-zone region will be determined later to know about the accumulative effect of stiction friction on the servo system. So stiction friction will not be the part of our system identification process. Force balance equation for trolley leaving out load due to motor + gearbox is given by:

$$\begin{aligned} F_l &= M_t \dot{V}_t + b_x v_t \\ F_l &= M_t r_x \dot{\omega}_l + b_x r_x \omega_l \\ \therefore T_l &= r_x \times F_l \\ \Rightarrow T_l &= r_x^2 M_t \dot{\omega}_l + r_x^2 b_x \omega_l \end{aligned} \quad (2.3.8)$$

Comparing trolley load equation (2.3.8) with $T_l = J_l \dot{\omega}_l + b_l \omega_l$ we get:
Equivalent rotational inertia of load for Trolley Subsystem:

$$J_l = r_x^2 M_t \quad (2.3.9)$$

Equivalent rotational damping coefficient of load for Trolley Subsystem:

$$b_l = r_x^2 b_x \quad (2.3.10)$$

Now using (2.3.9) and (2.3.10) in (2.3.5) and (2.3.6) the relation for total inertia and damping (motor + gearbox + mechanical-load) can be written as:

$$J_{l_{eq_1}} = K_{g_1}^2 J_{m_1} + r_x^2 M_t \quad (2.3.11)$$

$$b_{l_{eq_1}} = K_{g_1}^2 b_{m_1} + r_x^2 b_x \quad (2.3.12)$$

Transfer Function of DC Servo for Trolley Subsystem

Here the armature voltage to the motor is given by $V_{a_1} = G_{a_1} V_{ax}$ where G_{a_1} is the amplifier voltage gain given by motor power driver and V_{ax} is the pulse width modulation (PWM) duty cycle $\in [-1, 1]$ value fed to the crane from MATLAB® RWT toolbox. Now the transfer function similar to (2.3.1) for Trolley Subsystem from PWM input V_{ax} to trolley speed V_t can be written using (2.3.7), (2.3.11) and (2.3.12) in (2.3.2) and (2.3.3):

$$\boxed{\frac{V_t(s)}{V_{ax}(s)} = \frac{K_x}{\tau_x s + 1} \cdot e^{-s\tau_{dx}}} \quad (2.3.13)$$

Where:

- DC Gain of Complete Trolley Servo-Subsystem ($\frac{m}{s V}$):

$$\boxed{K_x = \frac{K_{g_1} K_{T_1} G_{a_1} r_x}{R_{a_1} b_{l_{eq_1}} + K_{g_1}^2 K_{T_1} K_{b_1}}} \quad (2.3.14)$$

- Mechanical Time Constant of Trolley Servo-Subsystem (sec):

$$\boxed{\tau_x = \frac{R_{a_1} J_{l_{eq_1}}}{R_{a_1} b_{l_{eq_1}} + K_{g_1}^2 K_{T_1} K_{b_1}}} \quad (2.3.15)$$

- V_{ax} = PWM Duty Cycle Input for Trolley Motor
- $J_{l_{eq_1}}$ and $b_{l_{eq_1}}$ are given by (2.3.11) and (2.3.12) respectively.

2.3.2 KLT Model for Servo-Subsystem of Arm-Subsystem

Servo system for Arm subsystem consists of following components as shown in Figure 2.2:

1. DC Motor
2. Gearbox
3. Parts of tower attached to gearbox shaft
4. Couplings
5. Counter Weight
6. Arm or Jib

Here the armature voltage to the motor is given by $V_{a_2} = G_{a_2} V_{a\theta}$ where G_{a_2} is the amplifier voltage gain given by motor power driver and $V_{a\theta}$ is the PWM duty cycle value $\in [-1, 1]$ fed to the crane from MATLAB RWT toolbox. So the transfer function of complete Servo system of Arm Subsystem from PWM input $V_{a\theta}$ to arm speed ω_a can be written as:

$$\boxed{\frac{\omega_a(s)}{V_{a\theta}(s)} = \frac{K_y}{\tau_y s + 1} \cdot e^{-s\tau_{d_y}}} \quad (2.3.16)$$

Where:

- DC Gain of Complete Servo of Arm Subsystem $\frac{rad}{sV}$:

$$\boxed{K_y = \frac{K_{g_2} K_{T_2} G_{a_2}}{R_{a_2} b_{l_{eq_2}} + K_{g_2}^2 K_{T_2} K_{b_2}}} \quad (2.3.17)$$

- Mechanical Time Constant of Complete Servo of Arm Subsystem [sec]:

$$\boxed{\tau_y = \frac{R_{a_2} J_{l_{eq_2}}}{R_{a_2} b_{l_{eq_2}} + K_{g_2}^2 K_{T_2} K_{b_2}}} \quad (2.3.18)$$

- $V_{a\theta}$ = PWM Duty Cycle Input for Trolley Motor
- $J_{l_{eq_2}}$ and $b_{l_{eq_2}}$ are given by:

$$J_{l_{eq_2}} = K_{g_2}^2 J_{m_2} + J_{l_y} \quad (2.3.19)$$

$$b_{l_{eq}} = K_g^2 b_m + b_{l_y} \quad (2.3.20)$$

2.4 Estimation of Lumped-KLT Parameters of Servo-Subsystems

So we need to find K_x , K_y , τ_x , τ_y , τ_{dx} and τ_{dy} to identify the transfer functions (2.3.13) and (2.3.16). We will employ following methods turn by turn to identify these parameters:

1. Step Response Method: It will be used to get the rough estimates of parameters to feed as initial guess to the Prediction Error Method (PEM) inside System Identification ToolboxTM (SITB) [13, 6, 7].
2. Spectral Analysis using Chirp: Used to determine the bandwidth of actuators that would help to design the input for identification using prediction error method.
3. Parametric Identification using Prediction Error Method (PEM) in SITB: To find more refined approximates of parameters on the basis of linear least squares.

During all system identification experiments the following considerations were observed:

- All identification experiments have been conducted without payload **coupling term** by removing the payload. As mentioned earlier the effects due to payload will be catered for separately and it will be described later.
- For the Arm Servo-Subsystem the trolley moves along the arm and hence has varying contribution for moment of inertia so the moment of inertia due to trolley (**coupling term**) will be neglected during this identification process by fixing it at the origin. The effects due to trolley position will be catered for separately that will be explained later.
- The data logging from crane has been done through MATLAB® RWT and the minimum possible sampling time is 10ms so for identification process we may need to upsample as the time constant is on the order of this sampling time.
- The speed output is not available so in all experiments wherever required we have used the direct differentiation of position output for speed estimation. Due to limitation of sampling interval the quantization noise is expected which can be removed by using zero-phase filtering⁶ [13, 6, 7] for both input and output if required.
- In all experiments duty cycle ratio, D , has been used as input and since the amplifier rail voltage is known to be 12V so the actual voltage can always be determined.

⁶“Pre-filtering the input and output data through the same filter does not change the input-output phase relationship for a linear system. So it will not change the phase response of the identified model and that’s why the term “zero-phase filtering” is used”

2.4.1 Step Response Method

As both trolley and arm servo-subsystems has been modeled as 1st order transfer functions and moreover the open loop response is stable as well as monotone so the step response method is the first natural choice for crude approximation of parameters K_x , K_y , τ_x , τ_y , τ_{d_x} and τ_{d_y} . The approximate values returned by the step response method will be used as initial guess for search based on PEM. Lets take the step Input:

$$V_a(s) = \frac{A}{s} \quad (2.4.1)$$

and the speed response is given by:

$$Y(s) = \left(\frac{K}{\tau s + 1}\right) \cdot \frac{A}{s} \quad (2.4.2)$$

taking inverse laplace transform

$$y(t) = AK(1 - e^{-t/\tau}) \quad (2.4.3)$$

so steady state response is given by:

$$y_{ss}(t) = AK \quad (2.4.4)$$

So if we know the step size and speed value at steady-state then the DC gain K can be easily found whereas system time constant τ_x can be read from the plot of step response. Step response would give the rough idea of the parameters which can be further refined by using the method of least squares. The value of time constant τ and delay τ_d , especially, are not easy to read off the plot so the reading may be erroneous which needs to be refined by advanced methods that will be explored later.

Trolley Subsystem:

The step response for trolley speed is shown in the Figure 2.5 . The steady-state response is given by $v_{tss}(t) = AK_x$. Note that the speed output has quantization noise at steady-state so the mean value has been considered for $v_{tss}(t)$. Now choosing the step of size $A = 0.7$ the step response for trolley reveals following estimates for parameters of the trolley servo model 2.3.13:

$$\tau_x = 0.020s \quad (2.4.5)$$

$$K_x = 0.14286 \frac{m}{sV} \quad (2.4.6)$$

$$\tau_{d_x} = 0.062s \quad (2.4.7)$$

Arm Subsystem:

The step response for arm speed is shown in the Figure 2.6. Due to quantization noise it is not easy to read the parameter values so smoothing is done by filtering

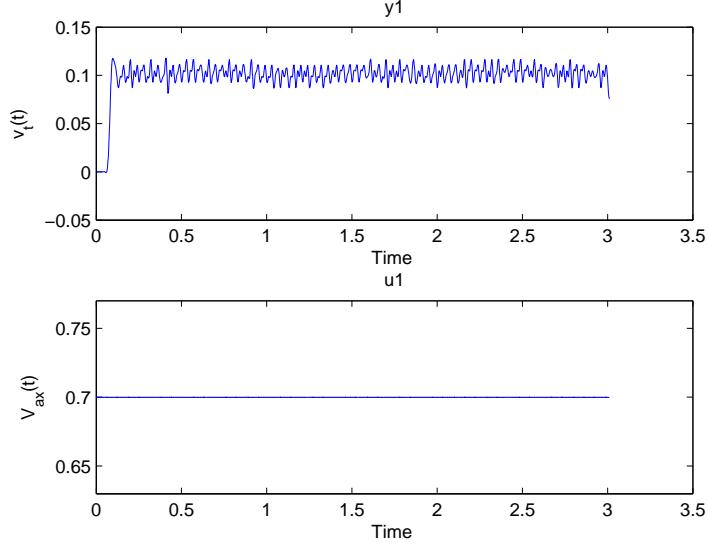


Figure 2.5: Step Response for Trolley Servo-Subsystem.

both input and output to keep the same input-output phase relationship. The steady-state response is given by $\omega_{ass}(t) = AK_y$. Note that even after filtering the speed output has some quantization noise at steady-state so the mean value has been used for $\omega_{ass}(t)$. Now choosing the step of size $A = 0.7$ the step response for arm reveals following estimates for parameters of the arm servo model 2.3.16 :

$$\tau_y = 0.080s \quad (2.4.8)$$

$$K_y = 0.66371 \frac{rad}{sV} \quad (2.4.9)$$

$$\tau_{dy} = 0.150s \quad (2.4.10)$$

2.4.2 Spectral Analysis using Chirp

Before parametric estimation it is beneficial to know the bandwidth of the actuators and for that frequency response function is required. Frequency response function describes the steady-state response of a system to sinusoidal inputs. Since until now we only have the input and output data so only estimate of frequency function is possible using periodogram method. Using chirp signal as input and logging output speed data the spectral analysis has been done to estimate the power spectral density $\Phi_u(\omega)$ of input and cross spectral density $\Phi_{yu}(\omega)$ on the basis of periodogram using input-output data which gives following estimate of frequency function [6]:

$$\hat{G}_N(i\omega) = \frac{\hat{\Phi}_{yu}(\omega)}{\hat{\Phi}_u(\omega)} \quad (2.4.11)$$

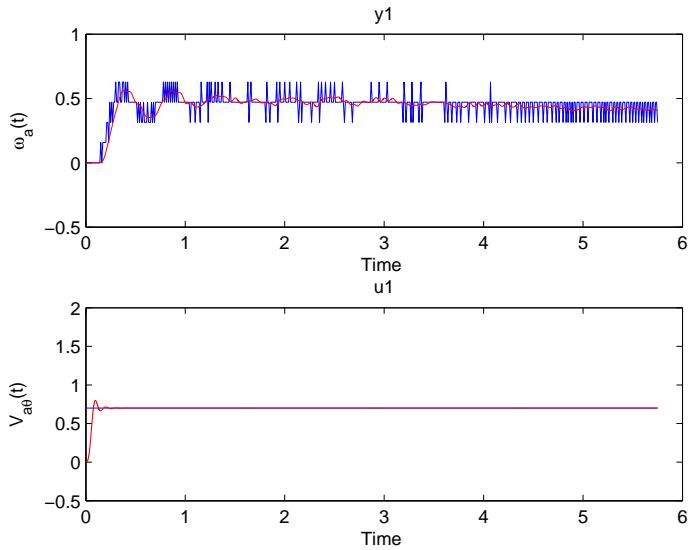


Figure 2.6: Step Response for Arm Servo-Subsystem. Blue shows response before filtering and red after filtering.

Frequency Function for Trolley Servo Subsystem:

Bode plot for trolley servo system is shown in the Figure 2.7. The plot shows that $-3dB$ bandwidth $\approx 10Hz$.

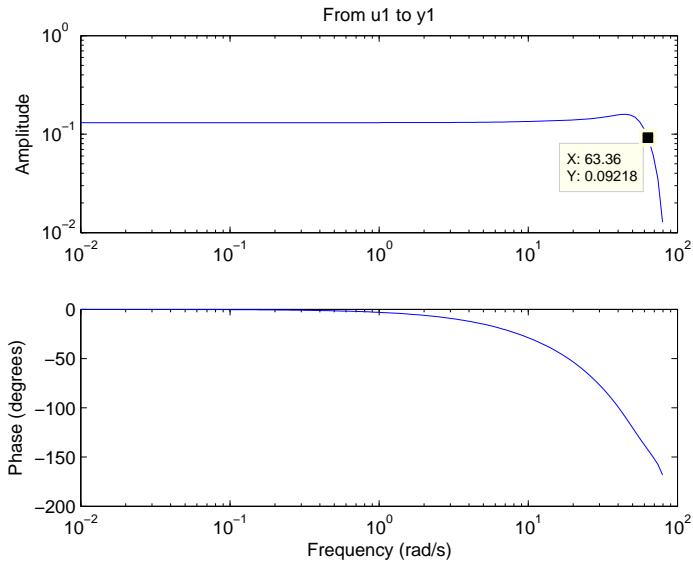


Figure 2.7: Spectral Estimation of frequency function Trolley Servo according to SPA procedure. Top: Magnitude Response Bottom: Phase Response.

Frequency Function for Arm Servo-Subsystem:

Bode plot for arm servo system is shown in the Figure 2.8. The plot shows that $-3dB$ bandwidth $\approx 2.593Hz$.

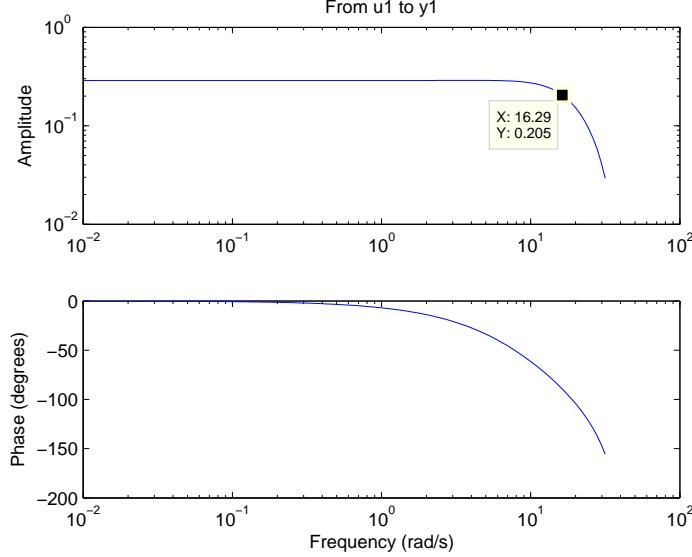


Figure 2.8: Spectral Estimation of frequency function of Arm Servo according to SPA procedure. Top: Magnitude Response Bottom: Phase Response.

2.4.3 Parametric Estimation of KLT using PEM

In this section our previous estimates would be complemented using prediction error method. Since in this case tailor-made model is linear so method of linear least squares would be used to get the best possible fit of the KLT model to the noisy data collected from crane. Before we proceed we need to discretize models (2.3.13) and (2.3.16). Since in our case the delay τ_{d_x} and τ_{d_y} are greater than sampling time h so according to [4] for first order systems with delay:

$$\dot{y}(t) = \left(\frac{-1}{\tau}\right)y(t) + \left(\frac{K}{\tau}\right)u(t - \tau_d) \quad (2.4.12)$$

Where:

$$\alpha = \frac{-1}{\tau}, \quad \beta = \frac{K}{\tau} \quad (2.4.13)$$

According to [4] for the case where $\tau_d > h$ the discrete version of (2.4.12) for ZOH can be written at time step ' k ' as:

$$y(kh) = ay((k-1)h) + b_1u((k-d)h) + b_2u((k-d-1)h) \quad (2.4.14)$$

Where h = Sampling Interval and if we select τ'_d such that:
 $\tau_d = (d - 1)h + \tau'_d$ where $\tau'_d < h$ and $d \in \mathbb{N}$ then:

$$a = e^{\alpha h} = e^{\frac{-h}{\tau}} \quad (2.4.15)$$

$$\begin{aligned} b_1 &= \int_0^{(h-\tau'_d)} e^{\alpha s} ds \beta = \int_0^{(h-\tau'_d)} e^{(\frac{-1}{\tau})s} ds (\frac{K}{\tau}) \\ &= -K(e^{\frac{(\tau'_d-h)}{\tau}}) - 1 \end{aligned} \quad (2.4.16)$$

$$\begin{aligned} b_2 &= e^{\alpha(h-\tau)} \int_0^{\tau'_d} e^{\alpha s} ds \beta = e^{(\frac{-1}{\tau})(h-\tau'_d)} \int_0^{\tau'_d} e^{(\frac{-1}{\tau})s} ds (\frac{K}{\tau}) \\ &= e^{\frac{(\tau'_d-h)}{\tau}} (-K(e^{\frac{-\tau'_d}{\tau}}) - 1) \end{aligned} \quad (2.4.17)$$

Now looking at (2.4.14) the best predictor for output of the system at step ' k' is given by:

$$\hat{y}(k|\theta) = \phi^T(k)\theta \quad (2.4.18)$$

Where:

θ = Column vector that contains the unknown parameters (2.4.15)-(2.4.17)

$\phi(k)$ = a column vector called regressor vector that contains old inputs and outputs called regressors

$$\theta = \begin{pmatrix} a \\ b_1 \\ b_2 \end{pmatrix}, \phi(k) = \begin{pmatrix} y((k-1)h) \\ u((k-d)h) \\ u((k-d-1)h) \end{pmatrix} \quad (2.4.19)$$

Due to measurement noise the prediction error can be written as:

$$\varepsilon(k, \theta) = y(kh) - \hat{y}(k|\theta) \quad (2.4.20)$$

Note that $y(kh)$ is the actual measured output which is noisy and $\hat{y}(k|\theta)$ is the predicted output. For $k = 1 \dots N$ and using (2.4.18) and (2.4.19) in (2.4.20) we get:

$$\boxed{\underbrace{\begin{pmatrix} \varepsilon(1, \theta) \\ \varepsilon(2, \theta) \\ \vdots \\ \varepsilon(N, \theta) \end{pmatrix}}_{=\epsilon} = \underbrace{\begin{pmatrix} y(1h) \\ y(2h) \\ \vdots \\ y(Nh) \end{pmatrix}}_{=Y} - \underbrace{\begin{pmatrix} y(0) & u((1-d)h) & u((1-d-1)h) \\ y(1) & u((2-d)h) & u((2-d-1)h) \\ \vdots & \vdots & \vdots \\ y(N-1) & u((N-d)h) & u((N-d-1)h) \end{pmatrix}}_{=X} \underbrace{\begin{pmatrix} a \\ b_1 \\ b_2 \end{pmatrix}}_{=\theta}} \quad (2.4.21)$$

$$\Rightarrow \epsilon = Y - X\theta \quad (2.4.22)$$

Where ϵ is the prediction error vector, X is the regressor matrix which contains the input-output data at last N sampling instants and Y is the vector containing the actual measured output data at N consecutive sampling instants up to current time step.

Now lets choose the quadratic cost function:

$$V_N(\theta) = \epsilon^T \epsilon \quad (2.4.23)$$

as a measure of how well the observation data fits to the model. It is natural to choose value of θ which minimizes (2.4.23)

$$\hat{\theta} = \arg \min_{\theta} V_N(\theta) \quad (2.4.24)$$

Using $\frac{dV_N}{d\theta}|_{\theta=\hat{\theta}} = 0$ the optimal value of parameter vector can be found as:

$$\hat{\theta} = (X^T X)^{-1} X^T Y \quad (2.4.25)$$

(2.4.25) gives the linear least square estimate of parameters (2.4.15)-(2.4.17) of the model (2.4.14) for observation data $y(k)$ and $u(k)$. Now we will write linear least square estimator for both trolley and arm servo-subsystems.

Trolley Servo System:

So according to (2.4.12) the trolley servo model (2.3.13) can be written as:

$$\dot{v}_t(t) = \left(\frac{-1}{\tau_x} \right) v_t(t) + \left(\frac{K_x}{\tau_x} \right) v_{a_x}(t - \tau_{d_x}) \quad (2.4.26)$$

Where:

$$\alpha_x = \frac{-1}{\tau_x}, \quad \beta_x = \frac{K_x}{\tau_x} \quad (2.4.27)$$

Discrete Version of (2.4.26) for ZOH can be written as:

$$v_t(kh) = a_x v_t((k-1)h) + b_{1_x} v_{a_x}((k-d_x)h) + b_{2_x} v_{a_x}((k-d_x-1)h) \quad (2.4.28)$$

Where h = Sampling Interval and if we select τ'_{d_x} such that:
 $\tau_{d_x} = (d_x - 1)h + \tau'_{d_x}$ where $\tau'_{d_x} < h$ and $d_x \in \mathbb{N}$ then using (2.4.15), (2.4.16) and (2.4.17):

$$a_x = e^{-\frac{h}{\tau_x}} \quad (2.4.29)$$

$$b_{1_x} = -K_x \left(e^{\left(\frac{\tau'_{d_x}-h}{\tau_x} \right)} - 1 \right) \quad (2.4.30)$$

$$b_{2_x} = e^{\left(\frac{\tau'_{d_x}-h}{\tau_x} \right)} \left(-K_x \left(e^{\left(\frac{-\tau'_{d_x}}{\tau_x} \right)} - 1 \right) \right) \quad (2.4.31)$$

Here $y(k) = v_t(k)$ and $u(k) = v_{a_x}$ so according to (2.4.28) and (2.4.18) the best predictor for speed of the trolley at discrete time step ' k ' is given by:

$$\hat{v}_t(k|\theta_x) = \phi_x^T(k)\theta_x \quad (2.4.32)$$

Where parameter vector and regressor vector is given by:

$$\theta_x = \begin{pmatrix} a_x \\ b_{1_x} \\ b_{2_x} \end{pmatrix}, \phi(k) = \begin{pmatrix} v_t((k-1)h) \\ v_{a_x}((k-d_x)h) \\ v_{a_x}((k-d_x-1)h) \end{pmatrix} \quad (2.4.33)$$

Now according to (2.4.25) the least square estimate of parameters (2.4.29)-(2.4.31) is given by:

$$\hat{\theta}_x = (X_1^T X_1)^{-1} X_1^T Y_1 \quad (2.4.34)$$

which minimizes the quadratic cost function:

$$V_N(\theta_x) = \epsilon_1^T \epsilon_1 \quad (2.4.35)$$

Where: ϵ_1, X_1 and Y_1 are given by substituting $y(k) = v_t(k)$ and $u(k) = v_{a_x}$ in (2.4.21).

Now using the crude estimates (2.4.6)-(2.4.7) from step response analysis, the identification problem in compact form can be stated as following minimization problem:

$$\min_{\theta_x} \epsilon_1^T \epsilon_1 \quad \text{s.t.} \quad \begin{cases} \hat{v}_t(k|\theta_x) = \phi_x^T(k)\theta_x \\ \epsilon_1 = Y_1 - X_2\theta_x \\ K_x \in [0.13, 0.15] \quad \tau_x \in [0.0150, 0.040] \quad \tau_{d_x} \in [0.025, 0.070] \end{cases} \quad (2.4.36)$$

The solution to this problem is given by (2.4.34) which can be solved through iterative search by varying parameter values in the range given by (2.4.36).

Arm Servo System:

Similarly for Arm-servo system (2.3.16) can be written as:

$$\dot{\omega}_a(t) = \left(\frac{-1}{\tau_y} \right) \omega_a(t) + \left(\frac{K_y}{\tau_y} \right) v_{a_\theta}(t - \tau_{d_y}) \quad (2.4.37)$$

Where:

$$\alpha_y = \frac{-1}{\tau_y}, \quad \beta_y = \frac{K_y}{\tau_y} \quad (2.4.38)$$

Discrete version of (2.4.37) for ZOH can be written as:

$$\omega_a(kh) = a_y \omega_a((k-1)h) + b_{1_y} v_{a_\theta}((k-d_y)h) + b_{2_y} v_{a_\theta}((k-d_y-1)h) \quad (2.4.39)$$

Where h = Sampling Interval and if we select τ'_{d_y} such that:
 $\tau'_{d_y} = (d_y - 1)h + \tau_{d_y}$ where $\tau'_{d_y} < h$ and $d_y \in \mathbb{N}$ then using (2.4.15), (2.4.16) and (2.4.17):

$$a_y = e^{\frac{-h}{\tau_y}} \quad (2.4.40)$$

$$b_{1_y} = -K_y(e^{(\frac{\tau'_{d_y}-h}{\tau_y})} - 1) \quad (2.4.41)$$

$$b_{2_y} = e^{(\frac{\tau'_{d_y}-h}{\tau_y})}(-K_y(e^{(\frac{-\tau'_{d_y}}{\tau_y})} - 1)) \quad (2.4.42)$$

Here $y(k) = \omega_a(k)$ and $u(k) = v_{a_\theta}$ so according to (2.4.39) and (2.4.18) the best predictor for speed of the arm at discrete time step ' k ' is given by:

$$\hat{\omega}_a(k|\theta_y) = \phi_y^T(k)\theta_y \quad (2.4.43)$$

Where parameter vector and regressor vector is given by:

$$\theta_y = \begin{pmatrix} a_y \\ b_{1_y} \\ b_{2_y} \end{pmatrix}, \phi_y(k) = \begin{pmatrix} \omega_a((k-1)h) \\ v_{a_\theta}((k-d_y)h) \\ v_{a_\theta}((k-d_y-1)h) \end{pmatrix} \quad (2.4.44)$$

Now according to (2.4.25) the least square estimate of parameters (2.4.40)-(2.4.42) is given by:

$$\hat{\theta}_y = (X_2^T X_2)^{-1} X_2^T Y_2 \quad (2.4.45)$$

which minimizes the quadratic cost function:

$$V_N(\theta_y) = \epsilon_2^T \epsilon_2 \quad (2.4.46)$$

Where: ϵ_2 , X_2 and Y_2 are given by substituting $y(k) = \omega_a(k)$ and $u(k) = v_{a_\theta}$ in (2.4.21).

Now using the crude estimates (2.4.9)-(2.4.10), the identification problem in compact form can be stated as following minimization problem:

$$\min_{\theta_y} \epsilon_2^T \epsilon_2 \quad \text{s.t.} \begin{cases} \hat{\omega}_a(k|\theta_y) = \phi_y^T(k)\theta_y \\ \epsilon_2 = Y_2 - X_2\theta_y \\ K_y \in [0.5, 0.8] \quad \tau_y \in [0.030, 0.1] \quad \tau_{d_y} \in [0.075, 0.175] \end{cases} \quad (2.4.47)$$

The solution to this problem is given by (2.4.45) which can be solved through iterative search by varying parameter values in the range given by (2.4.47).

Discussion on Identification Process:

The following procedure will be adopted for solving identification problems (2.4.36) and (2.4.47):

Design of Input: The input will be designed by taking into consideration the estimated bandwidth of actuators. As from spectral analysis we know that the desired bandwidth lies in the range 2Hz for arm motor and 10Hz for trolley motor so this information will be used to design the input with the appropriate frequency content in its spectrum. The spectral properties of input should be well according to the bandwidth of the system being identified. Moreover the type of input is also the choice to be made from deterministic binary signals like PRBS (pseudo-random binary signal) or square wave, telegraph signal, chirp etc. Since servo subsystem of crane is almost linear except static non-linearities like backlash and dead-zone. The dead-zone has been estimated separately using manual methods by slowly increasing input signal in small steps. So in the presence of dead-zone deterministic binary signals with levels high above dead-zone range is the natural choice. In our case owing to simplicity the ac square wave with ± 0.7 amplitude and frequency 0.2Hz has been selected. Moreover speed estimation due to direct differentiation is noisy due to quantization effects so digressing from the rule we have used comparatively small frequency 0.2Hz for the input which would help the predictor to average the steady-state value to get more realistic estimate. But this frequency selection is also justifiable since according to estimated frequency function plot the amplitude gain of the system is almost flat up to breaking point so there is no peaking in the system response and no fast modes are expected in the system except in nonlinear ranges which are not desired. The input and output data is shown in the Figure 2.9 for trolley and Figure 2.10 for arm servo subsystem.

Filtering and Smoothing: Due to quantization noise the data recorded from system has been smoothed by filtering both input and output to keep the same input-output phase relationship. The butterworth bandpass filter of order 5 has been used. This filtering is advantageous especially for steady-state response where after filtering the signals acquire more smooth and steady values. This filtering doesn't change the phase response of the system because the same filter is applied to both input and output which introduce same delay to both and hence their relative phase remains intact. So this filtering doesn't change the phase properties of the system. Normally in identification the pass-band is selected on the basis of desired bandwidth in which we want to fit data to model. This pass-band can be approximated on the basis of breakpoints in the approximate bode plot of the system. So according to our spectral analysis $[f_{p_1}, f_{p_2}] = [0.01, 25]\text{Hz}$ has been used for trolley servo and $[f_{p_1}, f_{p_2}] = [0.01, 8]\text{Hz}$ for arm servo subsystem. The input and output data before and after filtering is shown in the Figure 2.9 for trolley and Figure 2.10 for arm servo subsystem. The plot in blue shows the original recorded data and the one in red shows the filtered data.

Up-Sampling Using Interpolation Filter: Since $h = 0.01[\text{sec}]$ is the smallest possible sampling interval due to hardware limitation to record the system input-output data and according to the step response τ_x roughly lies in the range of $0.02 - 0.04[\text{sec}]$ but for the accurate identification the sampling interval should be at least one tenth of rise time so the identification data will be first upsampled by factor of 10 using linear interpolation filter and then new identification data will be used for identification process. After up-sampling, the new sampling interval $h = 0.001[\text{sec}]$. The plots shown in the Figures 2.9 and 2.10

are that of up-sampled signals.

Outliers: Outliers in the identification data set will be removed. These outliers may exist due to backlash especially in the gears of arm servo-subsystem at reversal or at startup. The data segments which are doubtful, if any, may be removed manually by looking carefully at instants of reversal or start-up.

PEM-Based Search in SITB: Since the delays τ_{d_x} and τ_{d_y} are not known exactly so it's not straight forward to form the regressor matrices X_1 and X_2 to solve equations (2.4.34) and (2.4.45). Instead we have to iteratively search for the parameters by increasing d_x and d_y and for each the cost will be calculated. As from step response we know the range in which delay lies so the upper and lower bound on d_x and d_y are known which reduce the possibilities. After calculating cost for each d_x and d_y we will choose the minimum among them. This iterative search can be automated in system identification toolbox (SITB) using '**PEM**' method by specifying process model with single pole and delay term. Before starting search in SITB using '**PEM**' the values for parameters obtained from step response will be fed as initial guess and also the minimum and maximum range of parameters will be specified. The following initial guesses were fed to SITB:

Initial Guess on the basis of Step Response For Trolley:

$$K_x \in [0.13, 0.15] \quad \tau_x \in [0.0150, 0.040] \quad \tau_{d_x} \in [0.025, 0.070] \quad (2.4.48)$$

Initial Guess on the basis of Step Response For Arm:

$$K_y \in [0.5, 0.8] \quad \tau_y \in [0.030, 0.1] \quad \tau_{d_y} \in [0.075, 0.175] \quad (2.4.49)$$

Estimated Servo Models: On the basis of (2.4.35), (2.4.34),(2.4.32) and (2.4.48) the SITB solves the identification problem (2.4.36) which returns the following estimate for *Trolley Servo-Subsystem*:

$$\boxed{\frac{V_t(s)}{V_{ax}(s)} = \frac{0.14114}{0.024s + 1} \cdot e^{-0.030936s}} \quad (2.4.50)$$

$$\Rightarrow K_x = 0.14114 \quad \tau_x = 0.024 \quad \tau_{d_x} = 0.030936 \quad (2.4.51)$$

The above model has been estimated using '**PEM**' using '**Auto**' Search-Method and with the value of cost function $V_N(\theta_x) = 4.53721 \times 10^{-5}$ and $FPE = 4.53842 \times 10^{-5}$.

On the basis of (2.4.46), (2.4.45), (2.4.43) and (2.4.49) the SITB solves the identification problem (2.4.47) returns the following estimate for *Arm Servo-Subsystem*:

$$\boxed{\frac{\omega_a(s)}{V_{a\theta}(s)} = \frac{0.68552}{0.057549s + 1} \cdot e^{-0.097298s}} \quad (2.4.52)$$

$$\Rightarrow K_y = 0.68552 \quad \tau_y = 0.057549 \quad \tau_{d_y} = 0.097298 \quad (2.4.53)$$

The above model has been estimated using '**PEM**' using '**Auto**' Search-Method and with the value of cost function $V_N(\theta_y) = 3.5325 \times 10^{-3}$ and $FPE = 3.5467 \times 10^{-3}$.

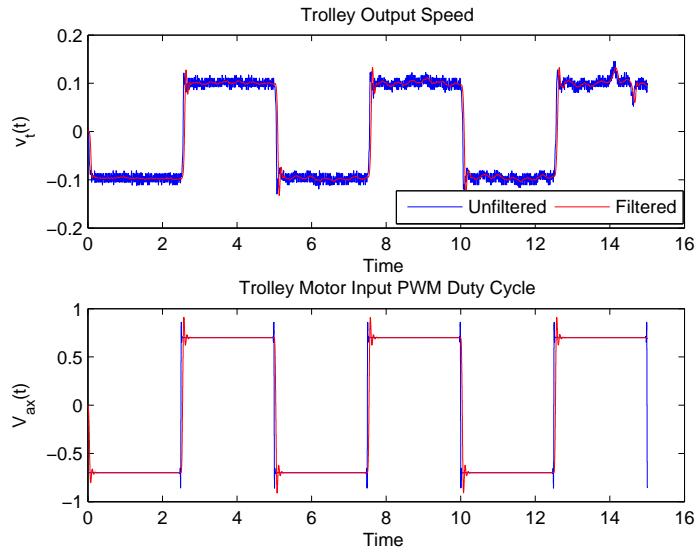


Figure 2.9: Estimation Data for Identification of Trolley Servo. Top: Trolley Output Speed. Bottom: Identification Input. Blue shows original data and red shows filtered data

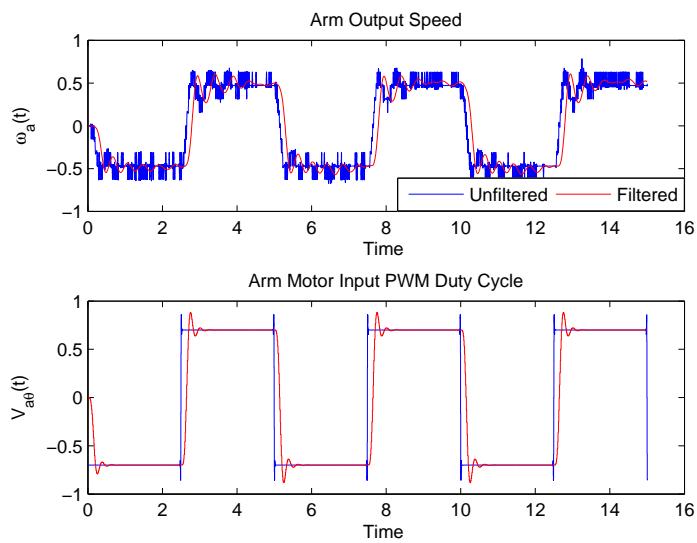


Figure 2.10: Estimation Data for Identification of Arm Servo. Top: Trolley Output Speed. Bottom: Identification Input. Blue shows original data and red shows filtered data

2.5 Model Validation of Servo-Subsystems

The estimated models (2.4.50) and (2.4.52) will be validated through four methods:

1. **Test 1- Running Model and System in Parallel:** Model output and actual system output will be compared for same input but different from what that was used for identification.
2. **Test 2- Residual Stochastic Analysis:** Validation of the model done on the basis of prediction error stochastic properties.
3. **Test 3- Residual Bode Analysis:** Validation done on the basis of Bode plots of model-error model (MEM).
4. **Test 4- Model-Based Control Design:** If the controller designed on the basis of the identified model works for real system then nothing else is required to ascertain the validity of the model.

2.5.1 Test 1- Running Model and System in Parallel:

The large input-output data is recorded and half of it is used for estimation and other half for validation process. This is the straightforward test in which both model and real system are fed with the same input and then their outputs are compared by computing the prediction error $\hat{R}_\varepsilon(k)$ and it should be as small as possible. The plots for both actual and model output for validation input are shown in Figures 2.11 and 2.12 for both trolley and arm respectively for 5-step ahead prediction. The fit is 91.7% and 87.43% for trolley servo-subsystem and Arm servo-subsystem respectively which is quite good and hence the Test-1 validates the estimated models.

2.5.2 Test 2- Residual Stochastic Analysis:

The residual analysis is done on the basis of properties of prediction error (2.4.20) to see if the identified model has missed any dynamics which is reflected in prediction error or residues. Since the prediction error comes due to measurement noise which is white in nature so the residues should have more and less the characteristics of white noise which means the auto-correlation function (acf):

$$\hat{R}_\varepsilon(k) = \frac{1}{N} \sum_{n=1}^N \varepsilon(n)\varepsilon(n+k) \quad (2.5.1)$$

ideally should have single peak at zero lag but in our case due to up-sampling using interpolation the correlation is natural between prediction errors at different time steps k . So here at least we expect to have decaying acf plot. The acf plots for both trolley and arm servo subsystems are shown in Figure 2.13 and 2.14 respectively. The acf plots doesn't depict the properties of white noise but they are decaying. Similarly another property of prediction errors is that they should be ideally independent of the input. This is the natural requirement because if there is some correlation between input $u(k)$ and residues $\varepsilon(k)$ then it means that some components of the residue have originated from the input

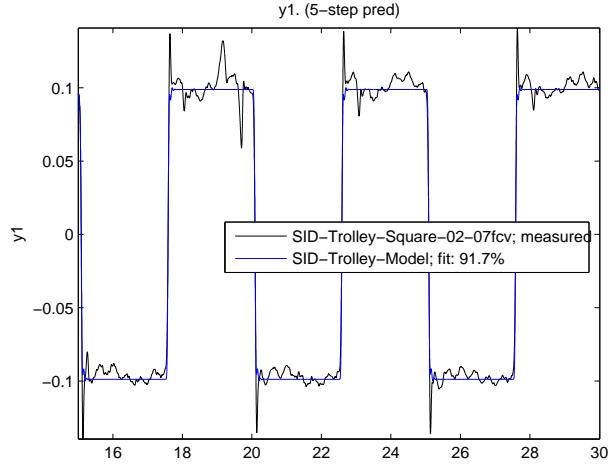


Figure 2.11: Validation Test 1 for Trolley Servo. Model Validation by simulating the identified model for validation input data and comparing with validation data output from real system. Note that Trolley servo subsystem has very small backlash in gears, that's why here we see almost no oscillations except some quantization noise.

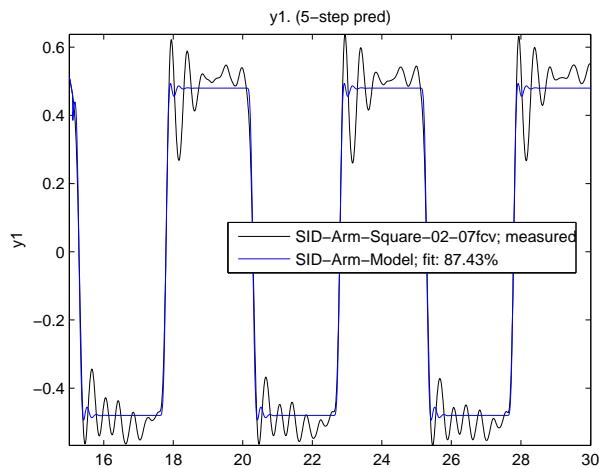


Figure 2.12: Validation Test 1 for Arm Servo. Model Validation by simulating the identified model for validation input data and comparing with validation data output from real system. Real Arm servo has some oscillatory behavior which our identified model hasn't captured. The reason for oscillation in arm servo is relatively large backlash in gears which we didn't consider during our identification process.

which in other terms means the input $u(k)$ has not excited all modes of the system and the model $\hat{y}(k|\hat{\theta}_N)$ has missed some dynamics. This can be evaluated by computing the cross-correlation between residue $\varepsilon(k)$ and the input $u(k)$

$$\hat{R}_{\varepsilon u}(k) = \frac{1}{N} \sum_{n=1}^N \varepsilon(n+k)u(n) \quad (2.5.2)$$

and if $\varepsilon(k)$ and $u(k)$ are really independent then (2.5.2) should be normally distributed with following variance:

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N \hat{R}_\varepsilon(k)\hat{R}_u(k) \quad (2.5.3)$$

So result should be acceptable if $\hat{R}_{\varepsilon u}(k)$ stays within $\pm 3\sigma$ limit. Yellow region in bottom subplot in Figures 2.13 and 2.14 shows $\pm 3\sigma$ region. we can see that for both trolley and arm $\hat{R}_{\varepsilon u}(k)$ stays within yellow region for almost all lags and hence Test-2 also validates the estimated models.

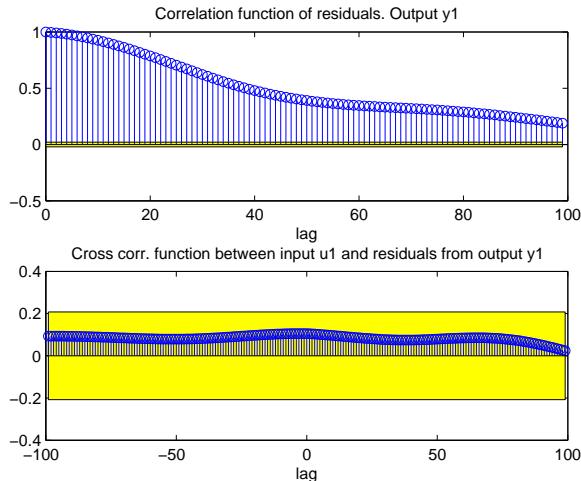


Figure 2.13: Validation Test 2 for Trolley Servo:Residual Analysis - Auto-correlation of residuals and Cross-correlation bet. residuals and past inputs for trolley servo subsystem. The above plot shows the acf of residues $\varepsilon = y - \hat{y}$ for trolley servo position $y = X_w$. Ideally residues should have white noise properties but here we see that correlation exists between different residue samples. It just shows that the output could have been predicted from last input but here the main reason for this correlation is up-sampling of original sampled data that we did to increase the number of data samples during rise time. The lower plot shows the cross-correlation between input and residues. This plot says that input has excited all the modes of the trolley servo system well and almost nothing has gone into residues.

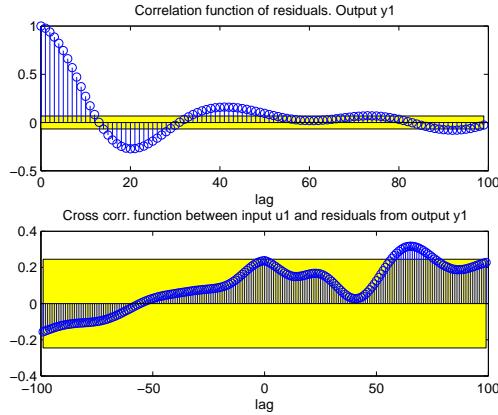


Figure 2.14: Validation Test 2 for Trolley Servo:Residual Analysis - Auto-correlation of residuals and Cross-correlation bet. residuals and past inputs for arm servo subsystem. The above plot shows the acf of residues $\varepsilon = y - \hat{y}$ for arm servo position $y = \theta$. Ideally residues should have white noise properties but here we see that correlation exists between different residue samples. Note that this acf is a lot better than that for trolley due to no up-sampling for arm servo as it was not required. The lower plot shows the cross-correlation between input and residues. This plot says that input has excited all the modes of the arm servo system well and almost nothing has gone into residues.

2.5.3 Test 3- Residual Bode Analysis:

According to [7] if the estimated model is intended for controller design then the more effective validation test is to display the frequency function of the estimate of residual dynamics along with estimated confidence regions. This gives a picture of the frequency ranges that the estimated model has not captured in the input-output behavior. Depending on the intended use, the model could then be accepted even when Validation Test 2 is violated, provided that the errors occur in the undesired or harmless frequency ranges. As in our case the major intent of the model is for controller design so this test is of particular significance. For this test a **model error model** (MEM), that is, a model from the identification input to the residuals is computed to see if any unmodeled dynamics are left in the desired frequency ranges. The **MEM** has been built by first using '**resid**' command in SITB to compute model residues and then '**arx**' command to estimate the error-model from input to model-residue as 10th order ARX model. The dynamics of this '**MEM**' should be ideally negligible which means both amplitude and phase plots in the bode diagram of **MEM** should be close to zero with narrow possible confidence region. The bode plots for both trolley and arm servo-subsystem residual dynamics is shown in the Figures 2.15 and 2.16 respectively. The desired frequency range of operation is up to 3Hz and 2Hz for trolley and arm servo-subsystems respectively. Although the trolley servo subsystem has flat amplitude response up to 10Hz but such high frequency inputs are not desired because experimental study has revealed that inputs above 3Hz to trolley motor and above 2Hz to arm motor may induce vibration into the crane system which may cause safety and durability

issues due to mechanical wear out. The plots show that residual amplitude response is very low for both trolley and arm which means not much has been missed in terms of amplitude by the estimated models. The phase response of residual model for trolley has significant phase response for input frequency $f > 0.7 \text{ Hz}$. It shows that our estimated model for trolley has missed some phase properties of the real system and for fast changing input $f > 0.7 \text{ Hz}$ it may not predict the correct phase information of trolley position X_w . So we can say our estimated model for trolley servo is good for input frequencies $f_{in} \in [0, 0.7] \text{ Hz}$. Similarly, the phase response of residual model for arm is also acceptable for input frequencies $f \leq 1 \text{ Hz}$ but for $f > 1 \text{ Hz}$ there is a drastic increase in phase response of residual model which shows that our estimated model for arm has missed phase properties of the real arm servo system for fast changing input $f > 1 \text{ Hz}$ so it may not predict the correct phase information of arm position θ . So we can say our estimated model for arm is good for input frequencies $f_{in} \in [0, 1] \text{ Hz}$.

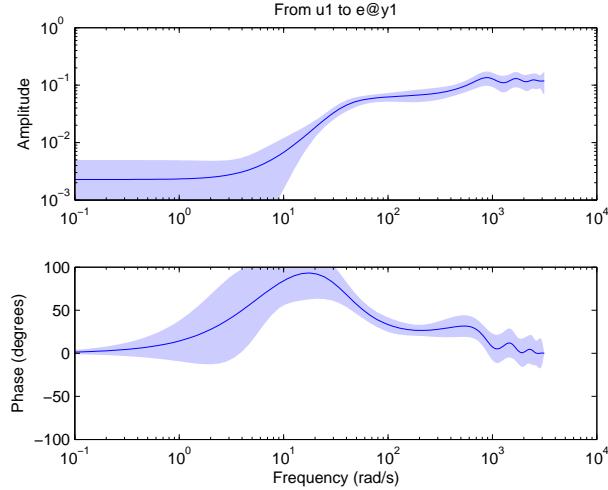


Figure 2.15: Validation Test 3 for Trolley Servo: Residual Dynamics or Model-Error Model. Amplitude response of residue model is quite close to zero and it shows that almost nothing has been missed by our identified in terms of gains. The phase response of residual model has significant phase response for input frequency $f > 0.7 \text{ Hz}$. It shows that our estimated model for trolley has missed some phase properties of the real system and for fast changing input $f > 0.7 \text{ Hz}$ it may not predict the correct phase information of trolley position X_w . So we can say our model is good for input frequencies $f_{in} \in [0, 0.7] \text{ Hz}$.

2.5.4 Test 4- Model-Based Control Design:

This is the most important validation test. If the controller design on the basis of the model works for the real system as well then it completely validates the estimated model. Sometime little fine tuning is needed for the controller to work for the real system and if, for example, we have to adjust them only by $\pm 10 - 15\%$ even then model will be considered valid. The model has also passed

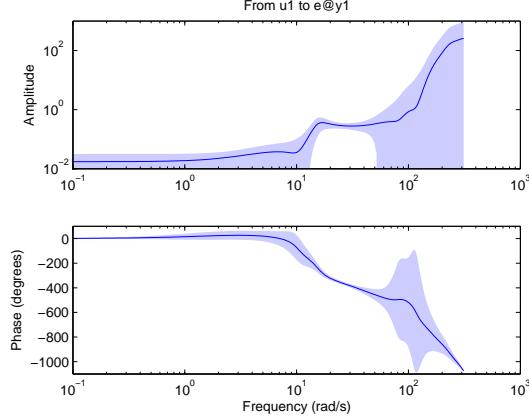


Figure 2.16: Validation Test 3 for Arm Servo: Validation Test 3 for Arm Servo: Residual Dynamics or Model-Error Model. Amplitude response of residue model is quite close to zero and it shows that almost nothing has been missed by our identified model in terms of gains. The phase response of residual model is also acceptable for input frequencies $f \leq 1 \text{ Hz}$ but for $f > 1 \text{ Hz}$ there is a drastic increase in phase response of residual model which shows that our estimated model for arm has missed phase properties of the real arm servo system for fast changing input $f > 1 \text{ Hz}$ so it may not predict the correct phase information of arm position θ . So we can say our model is good for input frequencies $f_{in} \in [0, 1] \text{ Hz}$.

this test since the controller designed on the basis of model also worked quite well for the simulink model provided by manufacturer. See Figure 3.14

2.5.5 Section Summary

In this section we have validated the identified model for servo subsystems of our crane. We applied four different validation tests to validate different aspects of our model. According to test-3 the models does not simulate well the phase properties of the system for higher frequency ranges. So according to test-3 our estimated model for trolley servo is good for input frequencies in the range $f_{in} \in [0, 0.7] \text{ Hz}$ and estimated model for arm is good for input frequencies in the range $f_{in} \in [0, 1] \text{ Hz}$ although desired ranges were $f_{in} \in [0, 3] \text{ Hz}$ for trolley and $f_{in} \in [0, 2] \text{ Hz}$ for arm servo. The reason for this disparity is the backlash in the gears which we did not consider in our modeling. Also according to validation test-1 our model didn't capture the oscillation in the arm position θ . The reason for this oscillation is also the backlash. The most important test is test-4 according to which, the controller designed on the basis of identified model also works for the simulink model provided by manufacturer and hence our identified model served our model identification purpose.

2.6 Constraints-Driven Estimation of all Sub-Parameters Contained in Lumped-KLT

In the last section we obtained the estimates of lumped-KLT parameters given by (2.4.51) and (2.4.53). But as shown by (2.3.14), (2.3.15), (2.3.17) and (2.3.18) the lumped parameters K_x , τ_x , K_y and τ_y contains many other physical parameters which we call here as sub-parameters. We are interested to find all these sub-parameters.

2.6.1 Identification of Sub-Parameters for Trolley Subsystem

Graying-out the Model to reduce unknown Parameters

According to datasheet of trolley motor:

$$\text{Gear Ratio:} \quad K_{g_1} = 76.84 \quad (2.6.1)$$

$$\text{Maximum Armature Voltage:} \quad v_{a1max} = 24V \quad (2.6.2)$$

$$\text{Maximum Torque @ 24 V:} \quad T_{1max} = 0.00651Nm \quad (2.6.3)$$

$$\text{Maximum Angular Speed@ } T_{1max}: \quad \omega_{1Tmax} = 442.5666 rad/s \quad (2.6.4)$$

$$\text{Maximum Armature Current @ } T_{1max}: \quad i_{a1max} = 0.34A \quad (2.6.5)$$

Since in such low speed applications the damping is normally very low so looking at (2.3.14) and (2.3.15) we can see that $K_{g_1}^2 K_{T_1} K_{b_1} \gg R_{a_1} b_{l_{eq_1}}$ so we can safely assume that:

$$\text{Equivalent Damping:} \quad b_{l_{eq_1}} \approx 0 \quad (2.6.6)$$

Also the following crane parameters have been provided by manufacturer:

$$\text{Mass of Trolley:} \quad M_t = 0.7kg \quad (2.6.7)$$

$$\text{Mass of Payload:} \quad m = 0.32kg \quad (2.6.8)$$

Following parameters have been measured experimentally using instruments like DMM and Oscilloscope:

$$\text{Armature Resistance:} \quad R_{a_1} = 32.575\Omega \quad (2.6.9)$$

$$\text{Motor Driver Amplifier Gain:} \quad G_{a_1} = 12 \quad (2.6.10)$$

Determination of Pulley Radius According to [1] the scale coefficient between encoder count, N , and trolley position X_w is given by $K_p = 5.81573 \times 10^{-5} \frac{m}{count}$. So the relation between encoder count and distance traveled by the trolley can be written as $X_w = K_p \times N$ where K_p = Scale Coefficient and N = Total Number of encoder counts recorded during trolley move. Since the resolution of the trolley encoder is 4096 CPR(counts per revolution) which means for 1 revolution of gearbox shaft $N = 4096$ counts and trolley covers $X_w = 2\pi r_x$ [m] so:

$$\text{Pulley Radius:} \quad r_x = \frac{K_p N}{2\pi} = 0.0379127m \quad (2.6.11)$$

So the load inertia can be computed by inserting (2.6.7) and (2.6.11) into (2.3.9):

$$\text{Load Inertia seen at Shaft } S_2: \quad J_l = 0.0010062 \text{ kgm}^2 \quad (2.6.12)$$

Also according to (2.4.27) $\beta_x = \frac{K_x}{\tau_x}$ and using the new notation $K_{mx} = \beta_x$ and value of damping from (2.6.6) we get:

Acceleration-Gain for PWM input $\in [-1, 1]$:

$$K_{mx} = \frac{K_{g1} K_{T1} G_{a1} r_x}{R_{a1} J_{l_{eq}}} \quad (2.6.13)$$

and Time Constant:

$$\tau_x = \frac{R_{a1} J_{l_{eq}}}{K_{g1}^2 K_{T1} K_{b1}} \quad (2.6.14)$$

Since in SI units $K_{T1} = K_{b1}$ so:

$$\tau_x = \frac{R_{a1} J_{l_{eq}}}{K_{g1}^2 K_{T1}^2} \quad (2.6.15)$$

And using (2.4.51) and (2.6.6) into (2.3.14) and (2.3.15):

$$K_{mx} = 5.8808 \frac{m}{s^2 V} \quad (2.6.16)$$

Physical Constraints Set to Extract rest of Parameters

Now the only unknown sub-parameters left are $J_{l_{eq}}$, K_{T1} , K_{b1} and J_{m1} . These can be computed by drawing set of constraints as shown below:

Constraint-1:

From (2.6.13) we can write:

$$K_{mx} R_{a1} J_{l_{eq}} - K_{g1} K_{T1} G_{a1} r_x = 0 \quad (2.6.17)$$

Constraint-2:

From (2.6.15) we can write:

$$R_{a1} J_{l_{eq}} - \tau_x K_{g1}^2 K_{T1}^2 = 0 \quad (2.6.18)$$

Constraint-3: Electrical Circuit Constraint

From A.1.12 we can write:

$$\omega_{1Tmax} K_{b1} = v_{a1max} - R_{a1} i_{a1max} \quad (2.6.19)$$

Constraint-4: Constraint on Motor Constants

In SI units both torque constant, K_{T1} , and back-emf constant K_{b1} must be equal for DC motors so:

$$K_{T1} - K_{b1} = 0 \quad (2.6.20)$$

Constraint-5: Mechanical Constraint

According to (2.3.11) and (2.3.9) we can write another constraint:

$$J_{l_{eq}} - K_{g1}^2 J_{m1} = J_l \quad (2.6.21)$$

Writing constraints (2.6.17) to (2.6.21) in matrix form:

$$\underbrace{\begin{pmatrix} R_{a_1} K_{mx} & -K_{g_1} G_{a_1} r_x & 0 & 0 & 0 \\ R_{a_1} & 0 & -K_{g_1}^2 \tau_x & 0 & 0 \\ 0 & 0 & 0 & \omega_{1T_{max}} & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -K_{g_1}^2 \end{pmatrix}}_{=A} \underbrace{\begin{pmatrix} J_{l_{eq1}} \\ K_{T_1} \\ K_{T_1}^2 \\ K_{b_1} \\ J_{m_1} \end{pmatrix}}_{=X} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ v_{a_{1max}} - R_{a_1} i_{a_{1max}} \\ 0 \\ J_{l_1} \end{pmatrix}}_{=B} \quad (2.6.22)$$

Now using (2.6.1) - (2.6.12) and (2.6.16) in the above constraint matrix (2.6.22) the vector X of unknown sub-parameters can be found as:

$$X = A^{-1}B \quad (2.6.23)$$

$$\Rightarrow X = \begin{pmatrix} 0.0053275 \\ 0.0292035 \\ 0.0012246 \\ 0.0292035 \\ 7.32 \times 10^{-7} \end{pmatrix} \quad (2.6.24)$$

Note that the constraint $K_{T_1} K_{b_1} = K_{T_1}^2$ is not fully satisfied as $K_{T_1}^2 = 0.0012246 \neq K_{T_1} K_{b_1} = 0.00085264$. This may be the result of aging in the system where speed constant and torque constant are no more equal or it may also arise due to the approximation error as well as estimation error. Since the error for this constraint is quite small so we will accept the parameter values.

So we get the required parameter values:

$$\text{Equivalent Mass Moment of Inertia: } J_{l_{eq1}} = 0.0053275 \text{ kgm}^2 \quad (2.6.25)$$

$$\text{Torque Constant of Motor: } K_{T_1} = 0.0292035 \frac{\text{Nm}}{\text{A}} \quad (2.6.26)$$

$$\text{Back EMF Constant of Motor: } K_{b_1} = 0.0292035 \frac{\text{Vs}}{\text{rad}} \quad (2.6.27)$$

$$\text{Mass Moment of Inertia of Motor Rotor } J_{m_1} = 7.32 \times 10^{-7} \text{ kgm}^2 \quad (2.6.28)$$

2.6.2 Identification of Sub-Parameters for Arm Subsystem

Graying-out the Model to reduce unknown Parameters

According to datasheet of arm motor:

$$\text{Gear Ratio: } K_{g_1} = 362.5 \quad (2.6.29)$$

$$\text{Maximum Armature Voltage: } v_{a_{2max}} = 12V \quad (2.6.30)$$

$$\text{Maximum Torque @ 12 V: } T_{2max} = 0.14 \text{ Nm} \quad (2.6.31)$$

$$\text{No Load Angular Speed: } \omega_{2NL} = 417.570 \text{ rad/sec} \quad (2.6.32)$$

$$\text{No Load Armature Current: } i_{a_{2NL}} = 0.9A \quad (2.6.33)$$

$$\text{Armature Resistance: } R_{a_2} = 0.20\Omega \quad (2.6.34)$$

$$\text{Motor Rotor Inertia: } J_{m_2} = 9.44 \times 10^{-5} \text{ kgm}^2 \quad (2.6.35)$$

Looking at (2.3.14) and (2.3.15) we can see that $K_{g2}^2 K_{T2} K_{b2} \gg R_{a2} b_{l_{eq2}}$ so we can safely assume that:

$$\text{Equivalent Damping:} \quad b_{l_{eq2}} \approx 0 \quad (2.6.36)$$

Following parameters have been measured experimentally using Oscilloscope:

$$\text{Motor Driver Amplifier Gain:} \quad G_{a2} = 12 \quad (2.6.37)$$

Also according to (2.4.38) $\beta_y = \frac{K_y}{\tau_y}$ and using the new notation $K_{my} = \beta_y$ and value of damping from (2.6.36) we get:

Acceleration-Gain for PWM input $\in [-1, 1]$:

$$K_{my} = \frac{K_{g2} K_{T2} G_{a2}}{R_{a2} J_{l_{eq2}}} \quad (2.6.38)$$

and Time Constant:

$$\tau_x = \frac{R_{a2} J_{l_{eq2}}}{K_{g2}^2 K_{T2} K_{b2}} \quad (2.6.39)$$

Since in SI units $K_{T2} = K_{b2}$ so:

$$\tau_x = \frac{R_{a2} J_{l_{eq2}}}{K_{g2}^2 K_{T2}^2} \quad (2.6.40)$$

And using (2.4.53) and (2.6.36) into (2.3.17) and (2.3.18):

$$K_{my} = 11.912 \frac{\text{rad}}{\text{s}^2 \text{V}} \quad (2.6.41)$$

Physical Constraints Set to Extract rest of Parameters

Now the only remaining unknown sub-parameters are $J_{l_{eq2}}$, K_{T2} , K_{b2} and J_{l2} . These can be computed by drawing set of constraints as shown below:

Constraint-1:

From (2.6.38) we can write:

$$K_{my} R_{a2} J_{l_{eq2}} - K_{g2} K_{T2} G_{a2} = 0 \quad (2.6.42)$$

Constraint-2:

From (2.6.40) we can write:

$$R_{a2} J_{l_{eq2}} - \tau_x K_{g2}^2 K_{T2}^2 = 0 \quad (2.6.43)$$

Constraint-3: Electrical Circuit Constraint

From (A.1.12) we can write:

$$\omega_{2NL} K_{b2} = v_{a2max} - R_{a2} i_{a1NL} \quad (2.6.44)$$

Constraint-4: Constraint on Motor Constants

In SI units both torque constant, K_{T_2} , and back-emf constant K_{b_2} must be equal for DC motors so:

$$K_{T_2} - K_{b_2} = 0 \quad (2.6.45)$$

Constraint-5: Mechanical Constraint

According to (2.3.19) we can write another constraint:

$$J_{l_{eq2}} - J_{l_2} = K_{g_2}^2 J_{m_2} \quad (2.6.46)$$

Writing constraints (2.6.42) to (2.6.46) in matrix form:

$$\underbrace{\begin{pmatrix} R_{a_2} K_{my} & -K_{g_2} G_{a_2} & 0 & 0 & 0 \\ R_{a_2} & 0 & -K_{g_2}^2 \tau_y & 0 & 0 \\ 0 & 0 & 0 & \omega_{2NL} & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \end{pmatrix}}_{=A} \underbrace{\begin{pmatrix} J_{l_{eq2}} \\ K_{T_2} \\ K_{T_2}^2 \\ K_{b_2} \\ J_{l_2} \end{pmatrix}}_{=X} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ v_{a2max} - R_{a_2} i_{a2NL} \\ 0 \\ K_{g_2}^2 J_{m_2} \end{pmatrix}}_{=B} \quad (2.6.47)$$

Now using (2.6.29) - (2.6.35) and (2.6.41) in the above constraint matrix (2.6.47) the vector X of unknown parameters can be found as:

$$X = A^{-1}B \quad (2.6.48)$$

$$\Rightarrow X = \begin{pmatrix} 50.8107 \\ 0.0278 \\ 0.0013 \\ 0.0278 \\ 38.4059 \end{pmatrix} \quad (2.6.49)$$

Note that the constraint $K_{T_2} K_{b_2} = K_{T_2}^2$ is not fully satisfied as $K_{T_2}^2 = 0.0013 \neq K_{T_2} K_{b_2} = 0.00077284$. This may be the result of aging in the system where speed constant and torque constant are no more equal or it may also arise due to the approximation error as well as estimation error. Since the error for this constraint is small so we will accept the parameter values.

So we get the required parameter values:

$$\text{Equivalent Mass Moment of Inertia: } J_{l_{eq2}} = 50.8107 \text{ kgm}^2 \quad (2.6.50)$$

$$\text{Torque Constant of Motor: } K_{T_2} = 0.0278 \frac{\text{Nm}}{\text{A}} \quad (2.6.51)$$

$$\text{Back EMF Constant of Motor: } K_{b_2} = 0.0278 \frac{\text{Vs}}{\text{rad}} \quad (2.6.52)$$

$$\text{Mass Moment of Inertia of load } J_{l_2} = 38.4059 \text{ kgm}^2 \quad (2.6.53)$$

2.7 Complete Crane Model

In this section for the sake of completion, we will first motivate for our suggested improvements in **Omar's Model** structure that we stated in (2.2.5)-(2.2.6) in Section 2.2. Since in the **Omar's Model** given by (2.2.1)-(2.2.4) DC

motors have been modeled as single integrator with constant gains between voltage input and output speed so we proposed improvement to use **KLT Model** structure instead. So during the identification process, we used **KLT Model** structure and the effects of payload dynamics and trolley movement along the arm were dropped through decoupling. So for the complete description of trolley and arm dynamics we will now add these effects into our improved model for servo-subsystems (2.4.26) and (2.4.37):

Trolley Subsystem: In the absence of payload the dynamics of the trolley is given by (2.4.26) which can be written as:

$$\ddot{X}_w(t) + \left(\frac{1}{\tau_x}\right)\dot{X}_w(t) = K_{mx}v_{a_x}(t - \tau_{d_x}) \quad (2.7.1)$$

Now in the following we will update the trolley servo model (2.7.1) to include the effects of payload motion. The payload motion acts as disturbance or coupling force on trolley dynamics but the magnitude and sign of this force depends on the position of payload w.r.t. z-axis. If during the trolley motion the payload is lagging behind the trolley then its sign will be negative which means it would try to slow down the trolley and if the payload is ahead of trolley then its sign will be positive which means it would tend to increase the trolley acceleration. According to [5] as shown in (2.2.1) this disturbance force has been modeled as follows:

$$F_{dist} = m_t g \alpha \quad (2.7.2)$$

According to the angle conventions used the model for trolley servo thus can be appended to accommodate the effects due to payload dynamics. So in the presence of payload attached to the string the dynamics of trolley is given by:

$$\ddot{X}_w + \left(\frac{1}{\tau_x}\right)\dot{X}_w(t) = K_{mx}v_{a_x}(t - \tau_{d_x}) - F_{dist} \quad (2.7.3)$$

$$\ddot{X}_w + \left(\frac{1}{\tau_x}\right)\dot{X}_w(t) + m_t g \alpha = K_{mx}v_{a_x}(t - \tau_{d_x}) \quad (2.7.4)$$

Arm Subsystem: In the absence of payload and trolley frozen at origin the dynamics of the arm is given by (2.4.37) which can be written as:

$$\ddot{\theta}(t) + \left(\frac{1}{\tau_y}\right)\dot{\theta}(t) = K_{my}v_{a_\theta}(t - \tau_{d_y}) \quad (2.7.5)$$

Now in the following we will update the arm servo model (2.7.5) to include the coupling torques due to payload and trolley motion.

Coupling Torque Due to Trolley Movement along Arm: The trolley motion along the arm acts as disturbance torque for arm dynamics. This disturbance torque component is inertial in nature and its magnitude depends on the trolley position X_w on the arm and increases with increase in X_w . According to [5] as shown in (2.2.3) this disturbance torque term can be modeled as:

$$T_{dist1} = M_r X_w^2 \ddot{\theta}(t) \quad (2.7.6)$$

Coupling Torque Due to Payload Dynamics: The payload motion also acts as disturbance torque on arm dynamics but the magnitude and sign of this torque

depends on the position of payload w.r.t. z-axis and trolley position. If during the arm motion the payload is lagging behind the arm then its sign will be negative which means it would try to slow down the arm and if the payload is ahead of arm then its sign will be positive which means it would tend to increase the arm acceleration. The magnitude of this disturbance torque also heavily depends on the trolley position, larger the X_w larger will be the magnitude. According to [5] this disturbance torque has been modeled as follows:

$$T_{dist2} = m_r g X_w \beta \quad (2.7.7)$$

According to the angle conventions used the model for arm thus can be appended to accommodate the effects due to trolley motion and payload dynamics.

$$\ddot{\theta}(t) + \left(\frac{1}{\tau_y}\right)\dot{\theta}(t) = K_{my}v_{a_\theta}(t - \tau_{d_y}) - T_{dist1} + T_{dist2} \quad (2.7.8)$$

$$(1 + M_r X_w^2)\ddot{\theta}(t) + \left(\frac{1}{\tau_y}\right)\dot{\theta}(t) - m_r g X_w \beta = K_{my}v_{a_\theta}(t - \tau_{d_y}) \quad (2.7.9)$$

Now using (2.7.4), (2.2.2),(2.7.9) and (2.2.4) the complete improved crane model with actuator dynamics is given as follows:

$$\ddot{X}_w + \left(\frac{1}{\tau_x}\right)\dot{X}_w(t) + m_t g \alpha = K_{mx}v_{a_x}(t - \tau_{d_x}) \quad (2.7.10)$$

$$L\ddot{\alpha} + g\alpha - \ddot{X}_w = 0 \quad (2.7.11)$$

$$(1 + M_r X_w^2)\ddot{\theta}(t) + \left(\frac{1}{\tau_y}\right)\dot{\theta}(t) - m_r g X_w \beta = K_{my}v_{a_\theta}(t - \tau_{d_y}) \quad (2.7.12)$$

$$L\ddot{\beta} + g\beta + X_w \ddot{\theta} = 0 \quad (2.7.13)$$

Table 2.2: Identified System Parameters

Parameter Description	Expression	Value
Total Load on Trolley Motor	$M_{eq} = \frac{J_{l_{eq1}}}{r_x^2}$	3.7089[kg]
Total Arm Inertia	$J_{l_{eq}} = J_{l_{eq2}}$	50.8107[kgm ²]
Mass of Payload	m	0.32[kg]
Mass of Trolley	M_t	0.7[kg]
Time Constant of Trolley Servo	τ_x	0.024[sec]
Time Constant of Arm Servo	τ_y	0.0575[sec]
Velocity Gain of Trolley Servo	K_x	$0.14114[\frac{m-s^{-1}}{V}]$
Velocity Gain of Arm Servo	K_y	$0.68552[\frac{rad-s^{-1}}{V}]$
Acceleration Gain of Trolley Servo	$K_{mx} = \frac{K_x}{\tau_x}$	$5.8808[\frac{rad-s^{-2}}{V}]$
Acceleration Gain of Arm Servo	$K_{my} = \frac{K_y}{\tau_y}$	$11.912[\frac{rad-s^{-2}}{V}]$
Payload Mass Relative to M_{eq}	$m_t = \frac{m}{M_{eq}}$	0.0863
Payload Mass Relative to $J_{l_{eq}}$	$m_r = \frac{m}{J_{l_{eq}}}$	0.0063
Trolley Mass Relative to $J_{l_{eq}}$	$M_r = \frac{M_t}{J_{l_{eq}}}$	0.0138

Now using the identified parameter values as given in Table 2.2 in the model (2.7.10)-(2.7.13) we get:

$$\boxed{\begin{aligned} \ddot{X}_w + 41.6667\dot{X}_w(t) + 0.8464\alpha &= 5.8808v_{a_x}(t - 0.03094) \\ L\ddot{\alpha} + 9.81\alpha - \ddot{X}_w &= 0 \\ (1 + 0.0138X_w^2)\ddot{\theta} + 17.3765\dot{\theta} - 0.0618X_w\beta &= 11.912v_{a_\theta}(t - 0.0973) \\ L\ddot{\beta} + 9.81\beta + X_w\ddot{\theta} &= 0 \end{aligned}} \quad (2.7.14)$$

The model given by (2.7.14) will be referred as **M₂**⁷ here onwards. Also note that all state variables in **M₂** are equivalent to those in [1] except β and θ . The angle β in **M₂** is equal to $-\beta$ in [1] and similarly θ in **M₂** is equal to $-\theta$ in [1]. Also, the comparison of the model with the actual Laboratory system is given in the table 2.3. The table shows how variables in model **M₂** correspond to actual signals being acquired from real laboratory crane system using MATLAB RTW.

Table 2.3: Comparison between States in Model M2 and Laboratory System

States in Model M ₂	Measured States in MATLAB
X_w	X_w
θ	T
α	$-X$
β	Y
L	L

When applying controller designed on the basis of **M₂** to the real system then both the X -angle and its derivative must be inverted before feeding them back to the controller.

2.8 Complete Crane Model Validation

According to [5] the model **M₂**⁸ is the simplified form of the highly nonlinear model and this simplification has been done by using small angle approximation and dropping few nonlinear terms. For simulation purpose, one can easily use the complete nonlinear version of model **M₂** by just plugging in the same estimated parameters values that are given in Table 2.2 . Since the main purpose is the control design on the basis of the simplified model ,**M₂**, so here we are interested to evaluate how well this simplified model captures the actual system's dynamics. That's why for complete model validation we run ,**M₂**, and real system in parallel. The special input has been designed to test the model by running it in parallel to the real system and the result is shown in the figures

⁷Note that we ourselves named the model given by (2.7.14) as **M₂** for the easy referral. Also, we ourselves named the simulink model provided by the manufacturer [1] as **M₁** for the easy referral

⁸Recall that our model **M₂** given by (2.7.14) is bit different from [5] in terms of actuator model. we modeled actuators as KLT whereas in [5] they are modeled as single integrator with gain between voltage and speed.

2.17 and 2.18. Figures show that even after linear approximations model \mathbf{M}_2 follows the system almost perfectly for servo-subsystems and also acceptably well for payload dynamics except some difference mainly in amplitude of payload angles as shown in Figure 2.18 and hence we can hope to come up with model-based controller that would work for the actual system as well to ensure at least semi-global asymptotic stability with acceptable control performance after some fine tuning.

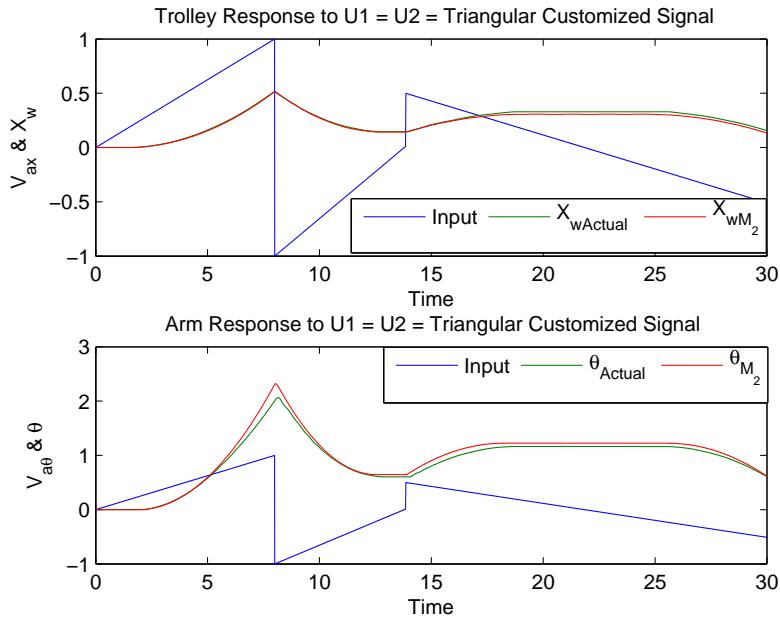


Figure 2.17: Model Validation for the customized signal. Top: Trolley Position, X_w . Bottom: Arm Position, θ . Blue: Input Signal, Green: Actual System Response, Red: Model Response.

2.9 Model Deficiencies and Improvement

The model \mathbf{M}_2 has some deficiencies as described below:

1. Model may not describe the true system well especially in terms of actuator phase response for input frequencies higher than 1Hz.
2. The \mathbf{M}_2 model structure used from [5] is the simplified version suitable for control design purpose but for simulation purpose we suggest to use its more elaborated and highly nonlinear model structure also given in [5]. The same parameter values used for \mathbf{M}_2 and given in the table 2.2 will be used for that highly nonlinear model as well. We suggest to implement that highly nonlinear model using s-function block in Simulink.
3. Since the model \mathbf{M}_2 has been simplified by neglecting the coupling between input $V_{a\theta}$ (arm motor) and payload α angle so it would not capture

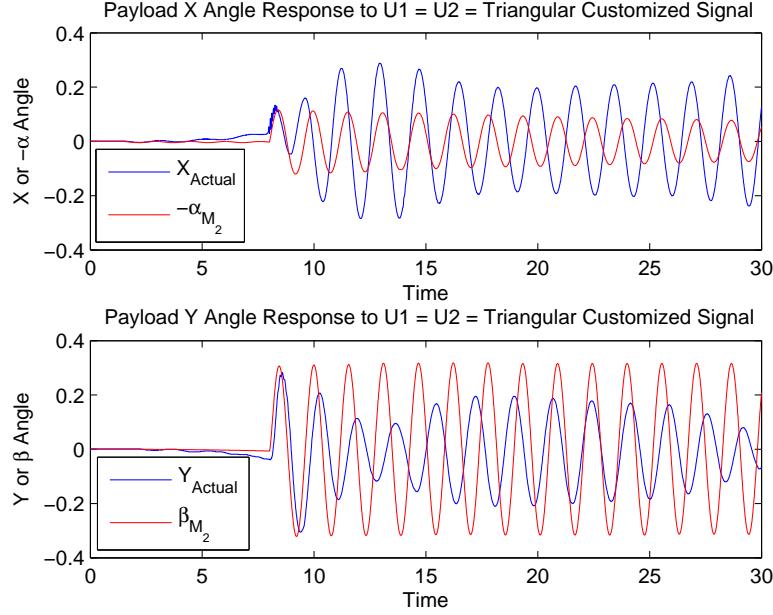


Figure 2.18: Model Validation for the customized signal. Top: Payload In-Plane Angle, $-\alpha$. Bottom: Payload Out-of-Plane Angle, β . Blue: Actual System Response, Red: Model Response.

any dynamics of α for excitation of arm servo whereas in the actual system the experiments have revealed that there is a small coupling between $V_{a\theta}$ and α which becomes considerable when arm excited with input frequency close to the natural frequency of oscillation of payload. This coupling is negligible unless we do not excite this resonant mode.

4. The model \mathbf{M}_2 does not predict the payload dynamics well when the trolley is very close to the origin. In this scenario the actual system shows some payload oscillation for input to any axis but the model doesn't predict this behavior well.
5. We also suggest to use better input to excite the system for identification. The pseudo-random binary signal (PRBS) or telegraph signal may be considered to evaluate if there is any improvement in estimates of time constant τ and system gain K .
6. Although during identification process we have outlined the procedure to take the backlash into consideration but not much attention was given to carefully select the backlash free data segments. Due to this reason the modeled delay τ_d might be little-bit erroneous so here we suggest to carefully select the data segments from identification data by treating the data in the small neighborhood of the instant of reversals as outliers. This may improve the estimate of system delay and time constant.
7. From application point of view we also suggest to estimate the coulomb

friction coefficients of the crane using '**PEM**'.

8. We also suggest to remove the trolley from the arm if possible to nullify the effects of trolley on estimation of arm inertia. We have tried to keep the trolley at origin during experiments but since trolley is not the point load so some contribution from its distributed mass may appear in the estimate.

Chapter 3

Controller Design: Continuous-Time Approach

In this chapter the controller design will be carried out for the real crane system on the basis of the model (2.7.14) identified in Chapter 2. In Section 3.1 the model will be transformed into state-space form and then to reduce the complexity, the complete state-space model will be split into two subsystems each of which is linear in term of its own states but parameterized in terms of L and a coupling term X_w . This split will make the controller design and analysis a lot simpler. These two subsystems are interconnected through only one state X_w . The controllability and observability properties of each subsystem will be studied separately as well. In the section 3.2 the controller design will be carried out for each subsystem independently. The full state feedback controller is not enough for the system because it is not robust to operation in the whole state-space especially in the presence of friction and would require gain scheduling. Thus for the robust tracking of step reference input and disturbance rejection, the full state feedback integral control structure will be employed by augmenting the system with integrators. Since each subsystem is linear, the pole placement technique on the basis of controllable canonical form is used to design the controller to achieve the desired response. Since the controller gains obtained are functions of L and X_w , the Gain-Scheduling Controller is achieved. Owing to the computational overhead related to the Gain-Scheduling Controller the Fixed-Gain Controller will also be proposed where constant gains are obtained by carefully selecting the value of operating point (X_w, L) and evaluating the Gain-Scheduling Controller at that point. The equilibrium point is selected by discretizing the subset of state space. In Section 3.3 the Fixed-Gain Integral Controller is tested on fully nonlinear simulation model provided by the manufacturer [1] and the step tracking performance is evaluated for various sling lengths L showing very good results. Furthermore, in simulations the dead-zone nonlinearity due to stiction friction in the gearbox is also included and the controller performance is evaluated achieving very good results as well.

3.1 Pre-Control Study: State-Space Analysis

In this section the model (2.7.14) will be first converted into state-space form and then the controllability and observability properties will be tested.

3.1.1 Conversion to State-Space Form

Selecting the following state variables:

$$\begin{aligned} x_1 &= X_w & x_2 &= \dot{X}_w & x_3 &= \alpha & x_4 &= \dot{\alpha} \\ x_5 &= \theta & x_6 &= \dot{\theta} & x_7 &= \beta & x_8 &= \dot{\beta} \\ u_1 &= V_{ax} & u_2 &= V_{a\theta} \end{aligned}$$

and designating four measured outputs:

$$y_1 = x_1 \quad y_2 = x_3 \quad y_3 = x_5 \quad y_4 = x_7$$

we get following state-space system representation of model **M₂**

$$\underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{pmatrix}}_x = \begin{pmatrix} x_2 \\ -41.667x_2 - 0.8464x_3 + 5.8808u_1 \\ -\frac{41.667}{L}x_2 - \frac{10.6564}{L}x_3 + \frac{5.8808}{L}u_1 \\ x_5 \\ \frac{1}{1+0.0138x_1^2}(-17.3765x_6 + 0.0618x_1x_7 + 11.912u_2) \\ x_8 \\ \frac{1}{L}(-9.81x_7 - \frac{x_1}{1+0.0138x_1^2}(-17.3765x_6 + 0.0618x_1x_7 + 11.912u_2)) \end{pmatrix}_{f(x,u)} \quad (3.1.1)$$

The system (3.1.1) is quite complex and it is not easy at all to design the controller for the system of this high complexity. So instead of designing one centralized controller for the whole system we propose to split the system into two simpler subsystems and then design the controller for each independently. Recall that in the last chapter as per our physical nomenclature for the crane system we mainly classified it into two subsystems physically that are: **Trolley-Subsystem** and **Arm-Subsystem**. Now, note that the state-space system (3.1.1) similarly can be split into two subsystems where state equation (3.1.1)-(3.1.1.4) describes the **Trolley Subsystem** and (3.1.1.5)-(3.1.1.8) describes the **Arm Subsystem**. Also note that these two subsystems are coupled only by X_w . The block diagram describing the connection of two subsystems is shown in the figure 3.1. Furthermore, note that the controller for each subsystem can be designed separately and also it can use only the local state information. Using L and X_w as parameters, we can transform (3.1.1) into two linear parameter varying (LPV) subsystems as given below:

Trolley Subsystem:

$$\begin{aligned}
 \underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix}}_{\dot{X}_1} &= \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -41.667 & -0.8464 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{41.667}{L} & -\frac{10.6564}{L} & 0 \end{pmatrix}}_{A_1} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}}_{X_1} + \underbrace{\begin{pmatrix} 0 \\ 5.8808 \\ 0 \\ \frac{5.8808}{L} \end{pmatrix}}_{B_1} u_1 \\
 \underbrace{\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}}_{Y_1} &= \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{C_1} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \\
 \dot{X}_1 &= A_1 X_1 + B_1 u_1 \\
 Y_1 &= C_1 X_1
 \end{aligned} \tag{3.1.2}$$

Note that the **Trolley Subsystem** (3.1.2) is linear but parameterized in terms of L .

Arm Subsystem:

Since the only coupling term between the arm and trolley subsystems is $X_w = x_1$ so treating the trolley position X_w as the varying parameter inside arm subsystem and then defining a new varying parameter given by:

$$\Phi(X_w) = \frac{1}{1 + 0.0138X_w^2} \tag{3.1.3}$$

we can write the arm subsystem as follows:

$$\begin{aligned}
 \underbrace{\begin{pmatrix} \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{pmatrix}}_{\dot{X}_2} &= \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -17.3765\Phi(X_w) & 0.0618X_w\Phi(X_w) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{17.3765X_w\Phi(X_w)}{L} & -\frac{(9.81+0.0618X_w^2\Phi(X_w))}{L} & 0 \end{pmatrix}}_{A_2} \underbrace{\begin{pmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix}}_{X_2} \\
 &\quad + \underbrace{\begin{pmatrix} 11.912\Phi(X_w) \\ 0 \\ -\frac{11.912X_w\Phi(X_w)}{L} \end{pmatrix}}_{B_2} u_2 \\
 \underbrace{\begin{pmatrix} y_3 \\ y_4 \end{pmatrix}}_{Y_2} &= \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{C_2} \begin{pmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} \\
 \dot{X}_2 &= A_2 X_2 + B_2 u_2 \\
 Y_2 &= C_2 X_2
 \end{aligned} \tag{3.1.4}$$

Note that input and system matrices of the **Arm Subsystem** (3.1.4) are parameterized in terms of L and X_w . Also note that (3.1.2) and (3.1.4) are LPV description of original highly nonlinear system (3.1.1). Now we will be able to use linear controller design methods for our nonlinear system which would have been quite complicated otherwise. For general details of LPV design methods see [14, 15].

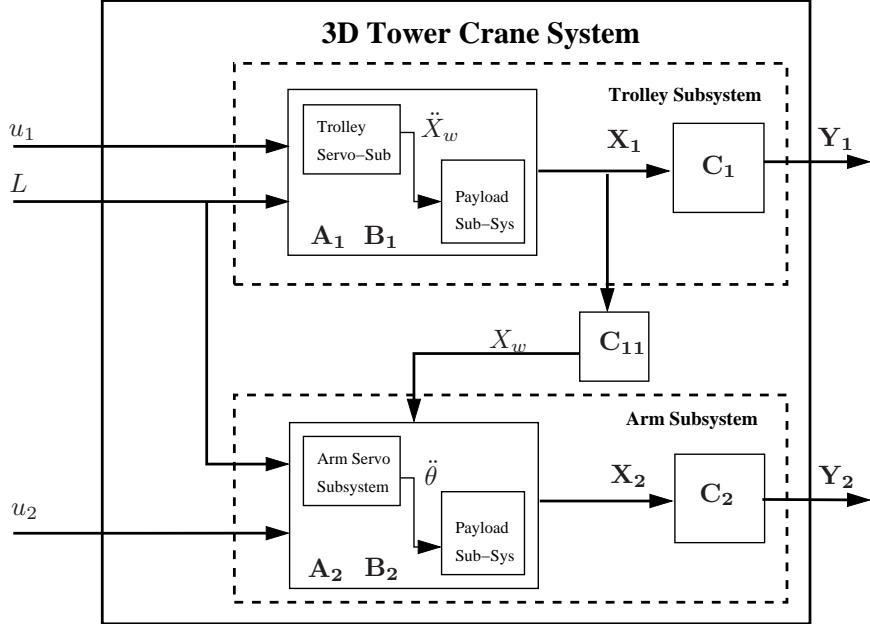


Figure 3.1: Block Diagram of 3D Tower Crane with the layout and interconnection of two subsystems inside. Note that C_{11} is the 1st row of C_1

3.1.2 Controllability and Observability:

Before designing controllers for subsystems (3.1.2) and (3.1.4) we have to first verify controllability (or at least stabilizability) and observability (or at least detectability) properties. We can do this for each separately:

Trolley Subsystem

Controllability:

$$\mathcal{C}_1 = \begin{pmatrix} 0 & 5.8808 & -245.0352936 & 10209.88558 - \frac{4.97750912}{L} \\ 5.8808 & -245.0352936 & 10209.88558 - \frac{4.97750912}{L} & \frac{414.795745}{L} - 425415.3024 \\ 0 & \frac{5.8808}{L} & -\frac{245.0352936}{L} & \frac{10209.88558}{L} - \frac{62.66815712}{L^2} \\ \frac{5.8808}{L} & -\frac{245.0352936}{L} & \frac{10209.88558}{L} - \frac{62.66815712}{L^2} & \frac{2818.591975}{L^2} - \frac{425415.3024}{L} \end{pmatrix} \quad (3.1.5)$$

The controllability matrix \mathcal{C}_1 has full row rank = 4 except for $L = 0$ where it is undefined so the 4-dimensional pair (A_1, B_1) is controllable for $L \in (0, \infty)$.

Observability:

$$\mathcal{O}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -41.667 & -0.8464 & 0 \\ 0 & -\frac{41.667}{L} & -\frac{10.6564}{L} & 0 \\ 0 & 1736.138889 & 35.2669488 & -0.8464 \\ 0 & \frac{1736.138889}{L} & \frac{35.2669488}{L} & -\frac{10.6564}{L} \end{pmatrix} \quad (3.1.6)$$

The observability matrix \mathcal{O}_1 has full column rank = 4 except for $L = 0$ where it is undefined so the 4-dimensional pair $(\mathbf{A}_1, \mathbf{C}_1)$ is observable for $L \in (0, \infty)$.

Arm Subsystem

Controllability: The controllability matrix \mathcal{C}_2 is too large to be presented but the result shows that 4-dimensional pair $(\mathbf{A}_2, \mathbf{B}_2)$ is controllable for $L \in (0, \infty)$ and $X_w \in (0, \infty)$.

Observability:

$$\mathcal{O}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & -17.3765\Phi & 0.0618X_w\Phi & 0 \\ 0 & \frac{17.3765X_w\Phi}{L} & -\frac{1.0(0.061803X_w^2\Phi+9.81)}{L} & 0 \\ 0 & 301.9427522\Phi^2 & -1.0738677X_w\Phi^2 & 0.0618X_w\Phi \\ 0 & -\frac{301.9427522X_w\Phi^2}{L} & \frac{1.0738677X_w^2\Phi^2}{L} & -\frac{1.0(0.061803X_w^2\Phi+9.81)}{L} \end{pmatrix} \quad (3.1.7)$$

The observability matrix \mathcal{O}_2 has full column rank = 4 except for $L = 0$ where it is undefined so the 4-dimensional pair $(\mathbf{A}_1, \mathbf{C}_1)$ is observable for $L \in (0, \infty)$ and $X_w \in [0, \infty)$.

3.1.3 Section Summary

In this section we have split the system into two subsystems **Arm Subsystem** and **Trolley Subsystem** and proved that each subsystem is both controllable and observable in the desired range of operation. Each subsystem is linear in its local states. The **Trolley Subsystem** is parameterized in terms of L and the **Arm Subsystem** is parameterized in terms of L and X_w . It is clear that by treating the system as the interconnection of two subsystems we have greatly reduced the complexity of the control design problem. Now in the next section the controller design will be carried out for each subsystem (3.1.2) and (3.1.4) independently.

3.2 Control Design: Integral Full-State Feedback Controller

In this section two controllers \mathbf{K}_1 and \mathbf{K}_2 will be designed separately on the basis of (3.1.2) and (3.1.4) for **Trolley Subsystem** and **Arm Subsystem** respectively. First of all the control performance specification will be outlined in Section 3.2.1. The robustness issue will be raised in Section 3.2.2 and solution will be presented accordingly. The controller design procedure will be shown in Section 3.2.3. Since the subsystems are parameterized and have high order so the pole placement is not easy owing to computational complexity. So here each subsystem will be transformed into controllable canonical form to do the pole placement using MATLAB® Symbolic Math Toolbox. Both the Gain-Scheduling and Fixed-Gain controllers will be proposed. The original gain functions obtained by the pole placement for the Gain-Scheduling Controller of the **Arm Subsystem** involves very high degree of polynomials and a large number of terms. Thus to make the gain scheduling functions usable they will be simplified by employing the surface fitting technique to do their close approximations with low order and simpler polynomials.

3.2.1 Control System Performance Specifications

The requirement for control system's performance are as follows:

1. The main purpose of crane control is to kill payload oscillation but we also want to have smooth trolley and arm response.
2. $\pm 5\%$ steady-state error is acceptable in both X_w and θ .
3. Payload Performance at steady-state: As far as payload oscillation is concerned the angles are not good measures for control performance specification since the magnitude of payload deviation from 0 depends on angles as well as the sling length. So specification in terms of angles will be different for different sling lengths hence we require the payload to remain within the ball \mathcal{B}_{r_1} of radius 0.005m at steady-state which means $\|L(\alpha \ \beta)\|^2 \leq 0.005$ m and also the oscillation must converge to the ball within at maximum two periods of oscillation.
4. Payload Performance during Transient: We also require the specification for the maximum admissible swing during the transient and it should be such that the payload remains inside the ball \mathcal{B}_{r_2} with $r = 0.05$ m. It means both trolley and arm should move such that throughout the move $\|L(\alpha \ \beta)\|^2 \leq 0.05$ m

3.2.2 Robust Tracking and Disturbance Rejection

In this thesis we will be looking at the tracking problem for step reference input. Full state feedback controller requires feed-forward gain for step-reference tracking. This feed-forward gain is fixed on the basis of type of reference input but in the presence of modeling uncertainty or disturbance signal or friction the output will not track asymptotically any step reference input and hence steady-state error will never converge to zero if that feed-forward gain is not adjusted

accordingly so for robust tracking and disturbance rejection a more advanced controller structure will be employed which achieves this gain adjustment using feedback in automatic fashion. According to [8] this can be achieved by employing two feedback loops- the internal loop is the same full-state feedback loop and the external loop is the unity feedback loop where the output which we want to track is fed back to compute the error. Also an integrator is introduced in the external loop of the system which integrates the error and after giving some gain the output is added to the internal loop. The closed-loop system configuration is shown in the Figure 3.9. The integrator output gives a new augmented state. Now we derive the system equations for each augmented subsystem.

Augmented Trolley Subsystem: The integrator output can be written as:

$$x_{a_1} = \int_0^t (r_1 - y_1) dt = \int_0^t (r_1 - C_{11}X_1) dt$$

Where C_{11} is the 1st row of C_1

$$\dot{x}_{a_1} = r_1 - C_{11}X_1 \quad (3.2.1)$$

Now augmenting (3.1.2) with (3.2.1) we get the new augmented trolley subsystem:

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{X}_1 \\ \dot{x}_{a_1} \end{pmatrix}}_{\dot{\mathbf{X}}_{a_1}} &= \underbrace{\begin{pmatrix} A_1 & 0 \\ -C_{11} & 0 \end{pmatrix}}_{\bar{\mathbf{A}}_1} \underbrace{\begin{pmatrix} X_1 \\ x_{a_1} \end{pmatrix}}_{\mathbf{X}_{a_1}} + \underbrace{\begin{pmatrix} \mathbf{0}_{4 \times 1} \\ 1 \end{pmatrix}}_{\bar{\mathbf{G}}_1} r_1 + \underbrace{\begin{pmatrix} \mathbf{B}_1 \\ 0 \end{pmatrix}}_{\bar{\mathbf{B}}_1} u_1 \\ \dot{\mathbf{X}}_{a_1} &= \bar{\mathbf{A}}_1 \mathbf{X}_{a_1} + \bar{\mathbf{G}}_1 r_1 + \bar{\mathbf{B}}_1 u_1 \end{aligned} \quad (3.2.2)$$

Augmented Arm Subsystem: The integrator output can be written as:

$$x_{a_2} = \int_0^t (r_2 - y_3) dt = \int_0^t (r_2 - C_{21}X_2) dt$$

Where C_{21} is the 1st row of C_2

$$\dot{x}_{a_2} = r_2 - C_{21}X_2 \quad (3.2.3)$$

Now augmenting (3.1.4) with (3.2.3) we get the new augmented arm subsystem:

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{X}_2 \\ \dot{x}_{a_2} \end{pmatrix}}_{\dot{\mathbf{X}}_{a_2}} &= \underbrace{\begin{pmatrix} A_2 & 0 \\ -C_{21} & 0 \end{pmatrix}}_{\bar{\mathbf{A}}_2} \underbrace{\begin{pmatrix} X_2 \\ x_{a_2} \end{pmatrix}}_{\mathbf{X}_{a_2}} + \underbrace{\begin{pmatrix} \mathbf{0}_{4 \times 1} \\ 1 \end{pmatrix}}_{\bar{\mathbf{G}}_2} r_2 + \underbrace{\begin{pmatrix} \mathbf{B}_2 \\ 0 \end{pmatrix}}_{\bar{\mathbf{B}}_2} u_2 \\ \dot{\mathbf{X}}_{a_2} &= \bar{\mathbf{A}}_2 \mathbf{X}_{a_2} + \bar{\mathbf{G}}_2 r_2 + \bar{\mathbf{B}}_2 u_2 \end{aligned} \quad (3.2.4)$$

Now we require to do the pole placement for the augmented subsystems but it can not be achieved in MATLAB® using `place` command because our system

and input matrices involve parameters L and X_w which cannot be handled by `place` command because it can only take numerical values. Moreover we also want to critically damp our servo subsystem which means multiple poles are required at the same place and this requirement cannot be met by `place` command either so in the next section we will first outline the method of pole placement for general linear system and then we will apply it to our subsystems (3.2.2) and (3.2.4).

3.2.3 Gain-Scheduling Controller: Pole Placement using Controllable Canonical Form

For any general linear system the pole placement can be done by transforming the system to controllable canonical form using similarity transformation. Let's say for a general 5th order linear system with state vector $\mathbf{x} \in \mathbf{R}^5$, u and $r \in \mathbf{R}$:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{G}r + \mathbf{B}u \quad (3.2.5)$$

Let's suppose the characteristic equation of 3.2.5 is given by:

$$\Delta(s) = s^5 + \alpha_1 s^4 + \alpha_2 s^3 + \alpha_3 s^2 + \alpha_4 s + \alpha_5 \quad (3.2.6)$$

Now according to [8] if (3.2.5) is controllable then using $\bar{\mathbf{X}} = \mathbf{P}\mathbf{X}$ it can be transformed into controllable canonical form:

$$\begin{aligned} \dot{\bar{\mathbf{X}}} &= \underbrace{\begin{pmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & -\alpha_4 & -\alpha_5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}}_{\bar{\mathbf{A}}} \bar{\mathbf{X}} + \underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{\bar{\mathbf{B}}} u + \bar{\mathbf{G}}r \quad (3.2.7) \\ \dot{\bar{\mathbf{X}}} &= \bar{\mathbf{A}}\bar{\mathbf{X}} + \bar{\mathbf{B}}u + \bar{\mathbf{G}}r \end{aligned}$$

Where $\bar{\mathbf{A}} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}$, $\bar{\mathbf{B}} = \mathbf{P}\mathbf{B}$ and $\bar{\mathbf{G}} = \mathbf{P}\mathbf{G}$ and if the controllability matrix of (3.2.5) is \mathcal{C} then that of (3.2.7) is given by:

$$\bar{\mathcal{C}} = \mathbf{P}\mathcal{C} \Rightarrow \mathbf{P}^{-1} = \mathcal{C}\bar{\mathcal{C}}^{-1} \quad (3.2.8)$$

Where:

$$\bar{\mathcal{C}}^{-1} = \begin{pmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ 0 & 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 0 & 1 & \alpha_1 & \alpha_2 \\ 0 & 0 & 0 & 1 & \alpha_1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.2.9)$$

Now using full state feedback:

$$\begin{aligned} u &= \mathbf{K}\mathbf{x} = \underbrace{\mathbf{K}\mathbf{P}^{-1}}_{\bar{\mathbf{K}}}\bar{\mathbf{X}} \\ u &= \bar{\mathbf{K}}\bar{\mathbf{X}} \end{aligned} \quad (3.2.10)$$

where $\mathbf{K} = \bar{\mathbf{K}}\mathbf{P}$ and closed-loop system is given by:

$$\dot{\bar{\mathbf{X}}} = (\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{\mathbf{K}})\bar{\mathbf{X}} + \mathbf{G}r \quad (3.2.11)$$

Note that the closed-loop systems $(\mathbf{A} + \mathbf{B}\mathbf{K})$ and $(\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{\mathbf{K}})$ have same eigenvalues. Now lets say desired pole location is known so the desired characteristic polynomial is given by:

$$\Delta_d(s) = s^5 + \bar{\alpha}_1 s^4 + \bar{\alpha}_2 s^3 + \bar{\alpha}_3 s^2 + \bar{\alpha}_4 s + \bar{\alpha}_5 \quad (3.2.12)$$

Now if we select $\bar{\mathbf{K}}$ as:

$$\bar{\mathbf{K}} = (\alpha_1 - \bar{\alpha}_1 \quad \alpha_2 - \bar{\alpha}_2 \quad \alpha_3 - \bar{\alpha}_3 \quad \alpha_4 - \bar{\alpha}_4 \quad \alpha_5 - \bar{\alpha}_5) \quad (3.2.13)$$

then using (3.2.13) in (3.2.11) we get:

$$\dot{\bar{\mathbf{X}}} = \underbrace{\begin{pmatrix} -\bar{\alpha}_1 & -\bar{\alpha}_2 & -\bar{\alpha}_3 & -\bar{\alpha}_4 & -\bar{\alpha}_5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}}_{\bar{\mathbf{A}}_{cs}} \bar{\mathbf{X}} + \bar{\mathbf{G}}r \quad (3.2.14)$$

Note that since $\bar{\mathbf{A}}_{cs}$ in (3.2.14) is in controllable canonical form so its characteristic polynomial is equal to (3.2.12) and since $(\mathbf{A} + \mathbf{B}\mathbf{K})$ and $(\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{\mathbf{K}})$ are equivalent so it implies the desired pole placement is achieved. Now using (3.2.9), (3.2.8) and (3.2.13) the controller gain for original system (3.2.5) is given by:

$$\boxed{\mathbf{K} = \bar{\mathbf{K}}\mathbf{P} = \bar{\mathbf{K}}\bar{\mathcal{C}}\mathcal{C}^{-1}} \quad (3.2.15)$$

So the procedure for eigenvalue assignment is as follows:

1. Write the characteristic polynomial of the original system (3.2.5) as given by (3.2.6)
2. Compute the Controllability matrix \mathcal{C} of the original system.
3. Compute the Controllability matrix $\bar{\mathcal{C}}$ using (3.2.9).
4. Compute the desired characteristic polynomial on the basis of desired pole locations as given by (3.2.12).
5. Compute controller gains $\bar{\mathbf{K}}$ for transformed system as given in (3.2.13).
6. Compute controller gains \mathbf{K} for original system using (3.2.15)

Pole Placement for Augmented Trolley Subsystem:

According to [5] the payload oscillation can be damped within minimum one period of oscillation. This can be achieved if the response of the Trolley and Arm is critically damped. Under this condition the payload completes one cycle exactly in settling time T_s of the trolley or arm which means the period of

payload oscillation T should be equal to T_s . Using this criterion the location of poles for each subsystem can be determined as follows:

To make the closed-loop servo subsystem critically damped two repeated poles are chosen at $-a$ and since for a 2nd order critically damped systems the settling time $T_s = \frac{4}{a}$ and the period of payload oscillation $T = 2\pi\sqrt{\frac{L}{g}}$ and hence for $T_s = T$ we get the desired pole location:

$$a = \frac{2}{\pi} \sqrt{\frac{g}{L}} \quad (3.2.16)$$

Since the behavior of payload is oscillatory so the closed-loop poles for payload should be at $-\zeta\omega_n \pm \omega_n\sqrt{1-\zeta^2}i$. The closed-loop pole for integrator will be placed at $-b$ whose value will be chosen such that it is greater than $-a$. Now using the best damping ratio $\zeta = 0.707$ and $\omega_n = \sqrt{\frac{g}{L}}$ the closed loop characteristic polynomial for each subsystem is given by:

$$\boxed{(s+a)^2(s^2 + 2\zeta\omega_n s + \omega_n^2)(s+b) = 0 \\ \left(s + \frac{2}{\pi}\sqrt{\frac{g}{L}}\right)^2 \left(s^2 + 1.414\sqrt{\frac{g}{L}}s + \frac{g}{L}\right)(s+b) = 0} \quad (3.2.17)$$

Now using the procedure outlined in the previous subsection, the pole placement will be carried out for **Trolley Subsystem** to achieve the desired closed-loop characteristic similar to (3.2.3).

The open-loop characteristic equation of the augmented trolley subsystem (3.2.2) is given by:

$$s^5 + \underbrace{41.667}_{\alpha_1} s^4 + \underbrace{\left(\frac{10.6564}{L}\right)}_{\alpha_2} s^3 + \underbrace{\left(\frac{408.75327}{L}\right)}_{\alpha_3} s^2 \quad (3.2.18)$$

Using $b = 0.6003$ the set of desired eigenvalues according to (3.2.3) is given as follows:

$$\boxed{\left\{ \frac{-1.993952}{\sqrt{L}}, \frac{-1.993952}{\sqrt{L}}, \frac{-2.2147 + 2.21474i}{\sqrt{L}}, \frac{-2.2147 - 2.21474i}{\sqrt{L}}, -0.6003 \right\}} \quad (3.2.19)$$

and the desired closed-loop characteristic equation is given as follows:

$$\begin{aligned} & s^5 + \frac{\left(0.6003 L^2 + 8.417303 L^{\frac{3}{2}}\right)}{L^2} s^4 + \frac{\left(31.449832 L + 5.052907 L^{\frac{3}{2}}\right)}{L^2} s^3 \\ & + \frac{\left(18.879334 L + 56.731810 \sqrt{L}\right)}{L^2} s^2 + \frac{\left(34.056105 \sqrt{L} + 39.002900\right)}{L^2} s \\ & + \frac{(23.413441)}{L^2} \end{aligned} \quad (3.2.20)$$

and using (3.2.15) Controller K_1 is given by:

$$\boxed{\mathbf{K}_1(L)^T = \begin{pmatrix} -\frac{0.5903228072\sqrt{L}+0.676069723}{L} \\ 6.799379812 - \frac{0.9833796556}{\sqrt{L}} \\ -0.2688982494\sqrt{L} - 2.859747108 \\ 0.1838027486L - 0.4479397792\sqrt{L} \\ \frac{0.4058446547}{L} \end{pmatrix}} \quad (3.2.21)$$

Note that $\mathbf{K}_1(L)$ is a gain vector whose each element is a function of sling length L . Hence (3.2.21) is the gain-scheduling controller for the **Trolley Subsystem**.

Pole Placement for Augmented Arm Subsystem:

Similarly characteristic equation of (3.2.4) is given as follows:

$$\Delta_2(s) = s^5 + \left(\frac{17.3765}{(0.0138 X_w^2 + 1)} \right) s^4 + \left(\frac{\left(1.0 \left(\frac{0.061803 X_w^2}{(0.0138 X_w^2 + 1)} + 9.81 \right) \right)}{L} \right) s^3 + \left(\frac{17.3765 \left(\frac{0.061803 X_w^2}{(0.0138 X_w^2 + 1)} + 9.81 \right)}{L (0.0138 X_w^2 + 1)} - \frac{1.0738677 X_w^2}{L (0.0138 X_w^2 + 1)^2} \right) s^2 \quad (3.2.22)$$

and using $b = 0.5430$ in (3.2.3) the set of desired eigenvalues is given as follows:

$$\boxed{\left\{ \frac{-1.993952}{\sqrt{L}}, \frac{-1.993952}{\sqrt{L}}, \frac{-2.2147 + 2.21474i}{\sqrt{L}}, \frac{-2.2147 - 2.21474i}{\sqrt{L}}, -0.5430 \right\}} \quad (3.2.23)$$

and the desired closed-loop characteristic equation is given as follows:

$$\begin{aligned} s^5 &+ \frac{\left(0.543 L^2 + 8.417303 L^{\frac{3}{2}} \right)}{L^2} s^4 + \frac{\left(31.449832 L + 4.570596 L^{\frac{3}{2}} \right)}{L^2} s^3 \\ &+ \frac{\left(17.077258 L + 56.731810 \sqrt{L} \right)}{L^2} s^2 + \frac{\left(30.805372 \sqrt{L} + 39.002900 \right)}{L^2} s \\ &+ \frac{(21.178575)}{L^2} \end{aligned} \quad (3.2.24)$$

$$\mathbf{K}_2(X_w, L) = (K_{21}(X_w, L) \ K_{22}(X_w, L) \ K_{23}(X_w, L) \ K_{24}(X_w, L) \ K_{25}(X_w, L)) \quad (3.2.25)$$

The computations involved are quite complicated and thus the MATLAB® *Sym-*
bolic Math Toolbox has been employed to compute the controller using (3.2.15). The analytical expression for Controller gain \mathbf{K}_2 is too big to display here so only the plot of each element function $\in \mathbf{K}_2$ will be shown. Note that each element of $\mathbf{K}_2(X_w, L)$ is a function of X_w and L . The 3D plots of each gain function are shown in Figures 3.2(a), 3.3(a), 3.4(a), 3.5(a) and 3.6(a). To get the simplified analytical expressions for gain functions, the actual data generated from computed gain functions $\in \mathbf{K}_2$ has been used to estimate the gain functions $\in \hat{\mathbf{K}}_2$ using low order polynomials having far less complexity. The

estimation has been done in MATLAB® using the *Surface Fitting Tool*. In the following we will show the results obtained from surface fitting for the estimate of each element of \mathbf{K}_2 .

Surface Fitting of Arm Gain Function- K_{21} : The surface fitted to the data generated from K_{21} is shown in Figure 3.2(b) and the actual is shown in Figure 3.2(a). In the following we will present the estimated model for the fitted surface along with its coefficients and goodness of fit.

General model:

$$\hat{K}_{21} = p_{00} + p_{10}X_w + p_{01}L + p_{11}X_wL + p_{02}L^2 + p_{12}X_wL^2 + p_{03}L^3 + p_{13}X_wL^3 + p_{04}L^4 + p_{14}X_wL^4 + p_{05}L^5 + ce^{-(wL)^2} \quad (3.2.26)$$

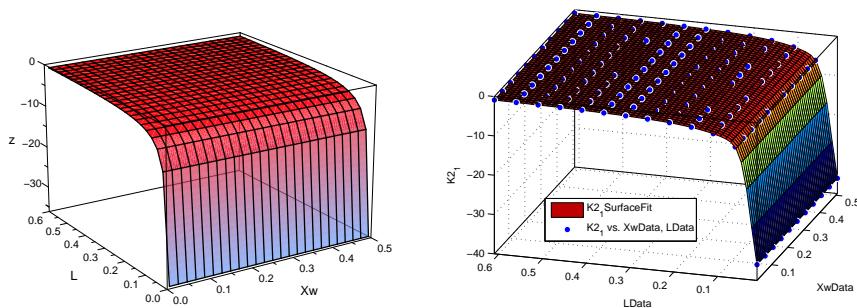
Coefficients:

$$\begin{aligned} c &= -30.13 & p_{00} &= -9.374 & p_{01} &= 81.32 & p_{02} &= -363.7 \\ p_{03} &= 862.2 & p_{04} &= -1032 & p_{05} &= 489.6 & p_{10} &= -0.2667 \\ p_{11} &= 4 & p_{12} &= -20.2 & p_{13} &= 40.56 & p_{14} &= -28.01 \\ w &= 30.79 \end{aligned} \quad (3.2.27)$$

Goodness of Fit:

$$R - Square = 1.0000 \quad AdjR - Square = 1.0000 \quad RMSE = 0.0099 \quad (3.2.28)$$

So according to above goodness measures, the fit is acceptable.



(a) Actual K_{21} -Gain Schedule as a function of X_w and L (b) Estimated \hat{K}_{21} -Gain Schedule as a function of X_w and L .

Figure 3.2: 3D Plot of Actual K_{21} function and its Estimate \hat{K}_{21} computed using Surface Fitting Tool.

Surface Fitting of Arm Gain Function- K_{22} :The surface fitted to the data generated from K_{22} is shown in Figure 3.3(b) and the actual is shown in Figure 3.3(a). In the following we will present the estimated model for the fitted

surface along with its coefficients and goodness of fit.

General model:

$$\hat{K}_{22} = p_{00} + p_{10} X_w + p_{01} L + p_{02} L^2 + p_{03} L^3 + p_{04} L^4 + p_{05} L^5 + p_{06} L^6 + p_{07} L^7 + p_{08} L^{10} \quad (3.2.29)$$

Coefficients:

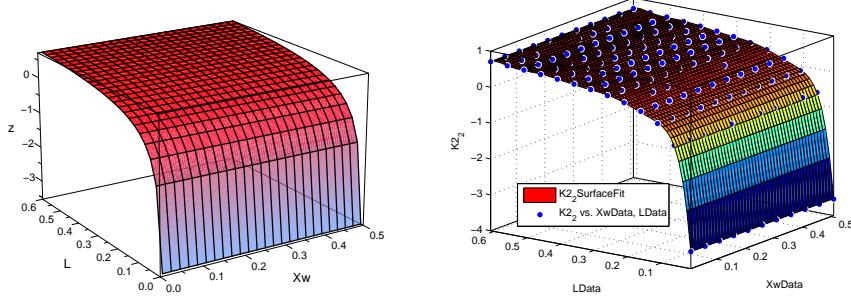
$p_{00} = -4.778$	$p_{01} = 142.5$	$p_{02} = -1860$
$p_{03} = 1.33 \times 10^{+4}$	$p_{04} = -5.492 \times 10^{+4}$	$p_{05} = 1.319 \times 10^{+5}$
$p_{06} = -1.738 \times 10^{+5}$	$p_{07} = 9.984 \times 10^{+4}$	$p_{08} = -1.808 \times 10^{+4}$
$p_{10} = -0.0101$		

(3.2.30)

Goodness of Fit:

$$R - Square = 0.9998 \quad AdjR - Square = 0.9998 \quad RMSE = 0.0151 \quad (3.2.31)$$

So according to above goodness measures, the fit is acceptable.



(a) Actual K_{22} -Gain Schedule as a function of X_w and L (b) Estimated \hat{K}_{22} -Gain Schedule as a function of X_w and L .

Figure 3.3: 3D Plot of Actual K_{22} function and its Estimate \hat{K}_{22} computed using Surface Fitting.

Surface Fitting of Arm Gain Function- K_{23} :The surface fitted to the data generated from K_{23} is shown in Figure 3.4(b) and the actual is shown in Figure 3.4(a). In the following we will present the estimated model for the fitted surface along with its coefficients and goodness of fit.

General model:

$$K_{23} = p_{00} + p_{10} X_w + p_{01} L + p_{20} X_w^2 + p_{30} X_w^3 + p_{40} X_w^4 + p_{50} X_w^5 + p_{60} X_w^6 + p_{70} X_w^7 + p_{80} X_w^8 \quad (3.2.32)$$

Coefficients:

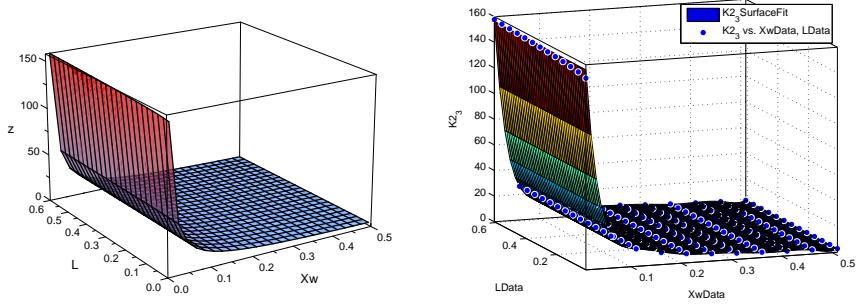
$$\begin{aligned}
 p_{00} &= 223.4 & p_{01} &= 0.7278 & p_{10} &= -8102 \\
 p_{20} &= 1.319 \times 10^5 & p_{30} &= -1.167 \times 10^6 & p_{40} &= 6.09 \times 10^6 \\
 p_{50} &= -1.924 \times 10^7 & p_{60} &= 3.614 \times 10^7 & p_{70} &= -3.711 \times 10^7 \\
 p_{80} &= 1.602 \times 10^7
 \end{aligned}$$

(3.2.33)

Goodness of Fit:

$$R-Square = 0.9998 \quad AdjR-Square = 0.9998 \quad RMSE = 0.5661 \quad (3.2.34)$$

The higher value for root mean square error (RMSE) shows that fit is not good everywhere and hence for some X_w values \hat{K}_{23} may not give a good estimate. For future work, we recommend to find the better estimate to minimize the error at least $\forall X_w \in [0, 0.5]$. We will not improve it further in our study.



(a) Actual K_{23} -Gain Schedule as a function of X_w and L (b) Estimated \hat{K}_{23} -Gain Schedule as a function of X_w and L . Fit is not good especially for very small values of X_w

Figure 3.4: 3D Plot of Actual K_{23} function and its Estimate \hat{K}_{23} computed using Surface Fitting Tool. Root mean square error for this fit is high and hence the improvement is required.

Surface Fitting of Arm Gain Function- K_{24} :The surface fitted to the data generated from K_{24} is shown in Figure 3.5(b) and the actual is shown in Figure 3.5(a). In the following we will present the estimated model for the fitted surface along with its coefficients and goodness of fit.

General model:

$$\begin{aligned}
 \hat{K}_{24} = & p_{00} + p_{10}X_w + p_{01}L + p_{20}X_w^2 + p_{11}X_wL + p_{02}L^2 + p_{30}X_w^3 + p_{21}X_w^2L \\
 & + p_{12}X_wL^2 + p_{03}X_w^3 + p_{40}X_w^4 + p_{31}X_w^3L + p_{22}X_w^2L^2 + p_{13}X_wL^3 \\
 & + p_{50}X_w^5 + p_{41}X_w^4L + p_{32}X_w^3L^2 + p_{23}X_w^2L^3 + p_{60}X_w^6 + p_{70}X_w^7
 \end{aligned}$$

(3.2.35)

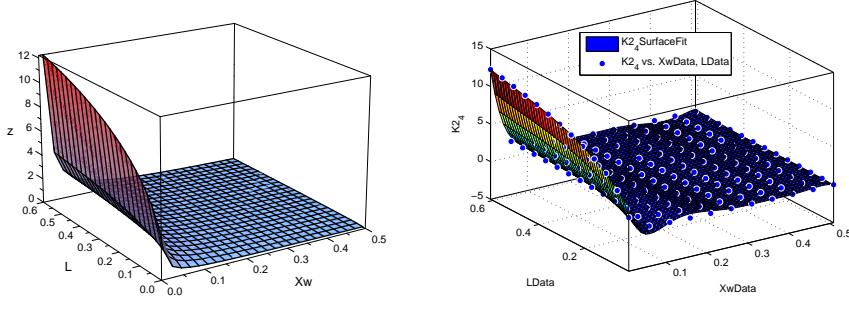
Coefficients:

$$\begin{aligned}
 p_{00} &= 6.017 & p_{01} &= 37.5 & p_{02} &= -54.12 \\
 p_{03} &= 34.03 & p_{13} &= -249.6 & p_{23} &= 387 \\
 p_{10} &= -313.7 & p_{11} &= -480.4 & p_{12} &= 497.3 \\
 p_{20} &= 5233 & p_{21} &= 2142 & p_{22} &= -1289 \\
 p_{30} &= -4.031 \times 10^{+4} & p_{31} &= -4028 & p_{32} &= 978.7 \\
 p_{40} &= 1.658 \times 10^{+5} & p_{41} &= 2755 & p_{50} &= -3.777 \times 10^{+5} \\
 p_{60} &= 4.502 \times 10^{+5} & p_{70} &= -2.194 \times 10^{+5}
 \end{aligned}
 \tag{3.2.36}$$

Goodness of Fit:

$$R - Square = 0.9950 \quad AdjR - Square = 0.9942 \quad RMSE = 0.2010 \tag{3.2.37}$$

The higher value for root mean square error (RMSE) shows that the fit is not good everywhere and hence for some L and X_w values \hat{K}_{24} may not give a good estimate. For future work, we recommend to find the better estimate to minimize the error at least $\forall L \in [0.1, 0.8]$ and $\forall X_w \in [0, 0.5]$. We will not improve it further in our study.



(a) Actual K_{24} -Gain Schedule as a function of X_w and L
(b) Estimated \hat{K}_{24} -Gain Schedule as a function of X_w and L . Fit is not good especially for very small values of X_w .

Figure 3.5: 3D Plot of Actual K_{24} function and its Estimate \hat{K}_{24} computed using Surface Fitting. Root mean square error for this fit is high and hence the improvement is required.

Surface Fitting of Arm Gain Function- K_{25} :The surface fitted to the data generated from K_{25} is shown in the figure 3.6(b) and the actual is shown in the figure 3.6(a). In the following we will present the estimated model for the fitted surface along with its coefficients and goodness of fit.

General model:

$$\begin{aligned}
 \hat{K}_{25} = & p_{00} + p_{10} X_w + p_{01} L + p_{02} L^2 + p_{03} L^3 + p_{04} L^4 + p_{05} L^5 \\
 & + p_{06} L^6 + p_{07} L^7 + p_{08} L^8
 \end{aligned}
 \tag{3.2.38}$$

Coefficients:

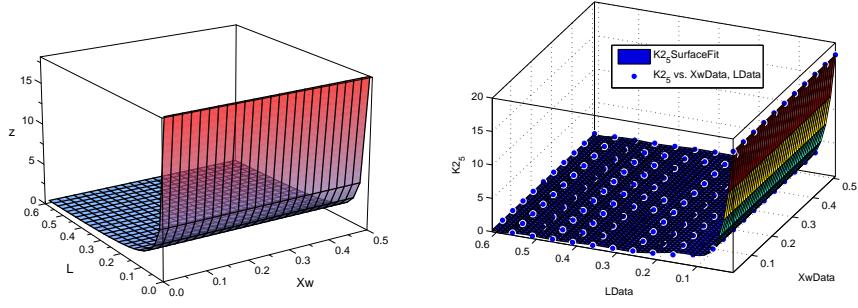
$p_{00} = 25.61$	$p_{01} = -865$	$p_{02} = 1.269 \times 10^{+4}$
$p_{03} = -9.841 \times 10^{+4}$	$p_{04} = 4.412 \times 10^{+5}$	$p_{05} = -1.183 \times 10^{+6}$
$p_{06} = 1.87 \times 10^{+6}$	$p_{07} = -1.608 \times 10^{+6}$	$p_{08} = 5.793 \times 10^{+5}$
$p_{10} = 0.02005$		

(3.2.39)

Goodness of Fit:

$$R - Square = 0.9998 \quad AdjR - Square = 0.9998 \quad RMSE = 0.0732 \quad (3.2.40)$$

So according to above goodness measures, the fit is acceptable.



(a) Actual K_{25} -Gain Schedule as a function of X_w and L (b) Estimated \hat{K}_{25} -Gain Schedule as a function of X_w and L .

Figure 3.6: 3D Plot of Actual K_{25} function and its Estimate \hat{K}_{25} computed using Surface Fitting Tool

Now using the estimated gain functions instead of (3.2.25), the Gain-Scheduling Controller for **Arm Subsystem** is given by:

$$\hat{K}_2 = (\hat{K}_{21} \quad \hat{K}_{22} \quad \hat{K}_{23} \quad \hat{K}_{24} \quad \hat{K}_{25}) \quad (3.2.41)$$

Where the estimates $\hat{K}_{21}, \hat{K}_{22}, \hat{K}_{23}, \hat{K}_{24}$ and \hat{K}_{25} are given by (3.2.26), (3.2.29), (3.2.32), (3.2.35) and (3.2.38) respectively.

So in this section we have tried to simplify the actual gain functions using surface fitting. According to results, the estimates for K_{21}, K_{22} and K_{25} are good but that for K_{23} and K_{24} are not good for all values of L and X_w . Hence we recommend to improve them further.

3.2.4 Robust Fixed-Gain Controller using Linearization

The gain-scheduling controllers pose a high computation overhead due to a lot of floating point multiplications so the implementation of such controller requires some considerable processing power. Hence to save the processing power the

fixed-gain controller has been devised. The fixed gains have been obtained by carefully selecting the equilibrium point (X_w, L) and then evaluating the gain-scheduling expressions at the selected point. The natural question arises about how to select the equilibrium point (X_w, L) so to answer the question we will describe the process of its selection. Since we know that $L \in [0, 0.8]$ and $X_w \in [0, 0.5]$ so the $L - X_w$ -Space is discretized in small steps of 0.01 and 2D grid is made. Then for each possible combination of L and X_w in the grid the gain-scheduling controllers (3.2.21) and (3.2.25) are evaluated to compute fixed-gain controllers for each grid point. So each grid point in the 2D grid contains one fixed-gain controller that works for the system at least in the small neighborhood of the L and X_w associated with that grid point. Our goal is to identify the best controller among 4131 fixed-gain controllers contained in 2D grid. Since we know the desired closed-loop pole locations for each (X_w, L) pair so our target is to place the closed-loop poles as closely as possible to the desired locations using the fixed-gain controller and hence the criterion is the degree of deviation from the desired closed-loop pole locations (3.2.19) and (3.2.23) for each grid point (X_w, L) . Selecting one fixed-gain controller at a time, the closed-loop pole residues are computed for all the grid points and then the norm of all residues is calculated. This process is repeated for all fixed-gain controllers. So the controller which gives the minimum residue norm has been selected. So according to the criterion, the fixed-gain controller computed at $(X_w, L) = (0.44, 0.16)$ gives the best performance in the region $L \in [0, 0.8]$ and $X_w \in [0, 0.5]$ among all possible fixed-gain controllers.

Trolley Subsystem:

So evaluating (3.2.21) at $L = 0.16$ we get following fixed-gain controller for trolley subsystem:

$$K_1 = (-5.7015 \quad 4.3408 \quad -2.9673 \quad -0.1498 \quad 2.5317) \quad (3.2.42)$$

Arm Subsystem:

Similarly evaluating the actual expression for gain-scheduling controller K_2 at $L = 0.16$ and $X_w = 0.44$ we get the following fixed-gain controller for arm subsystem:

$$K_2 = (-2.7525 \quad 0.1138 \quad 3.5477 \quad 0.1716 \quad 1.1358) \quad (3.2.43)$$

Note that the fixed-gain controller has been computed by evaluating the actual gain-scheduling functions since their analytical expression is available to us in symbolic toolbox. The estimated gain functions can also be used but there will be some small error that's why we have used the actual function to compute the value of controller gains.

Stability Analysis

The graphical stability analysis has been done for fixed-gain controllers (3.2.42) and (3.2.43). Using the fixed-gain controllers the closed-loop characteristic equations for each augmented subsystem has been computed and then its coefficients

are plotted vs L and X_w . For stability all the coefficients must be positive. The plots for coefficients of closed-loop characteristic equation for gain-scheduling controller has also been drawn and compared to the corresponding plot for fixed-gain. The characteristic equation for augmented trolley subsystem using the fixed-gain controller is given by:

$$\begin{aligned} \dot{\Delta}_{cs_1} = s^5 + & \underbrace{\left(\frac{0.8807525488}{L} + 16.1388549 \right)}_{\dot{\alpha}_{cs_{11}}} s^4 + \underbrace{\left(\frac{28.10653552}{L} + 33.52786858 \right)}_{\dot{\alpha}_{cs_{12}}} s^3 \\ & + \underbrace{\left(\frac{158.3221666}{L} + 14.91682028 \right)}_{\dot{\alpha}_{cs_{13}}} s^2 + \underbrace{\left(\frac{328.9083908}{L} \right)}_{\dot{\alpha}_{cs_{14}}} s \\ & + \underbrace{\frac{(146.334007 L + 0.0000000007855589703)}{L^2} - 0.0000005041356892}_{\dot{\alpha}_{cs_{15}}} \end{aligned} \quad (3.2.44)$$

To compare the characteristic of the closed-loop control system with Fixed-Gains to that with Gain-Schedules we will also introduce the characteristic equation for augmented trolley subsystem with gain scheduling controller as follows. Note that for brevity we will not show the numerical expression as it is too large:

$$\Delta_{cs_2} = s^5 + \alpha_{cs_{11}} s^4 + \alpha_{cs_{12}} s^3 + \alpha_{cs_{13}} s^2 + \alpha_{cs_{14}} s + \alpha_{cs_{15}} \quad (3.2.45)$$

The closed-loop characteristic equation for the augmented arm subsystem is too large to show here so we will show only the plot for its coefficients.

$$\dot{\Delta}_{cs_2} = s^5 + \dot{\alpha}_{cs_{21}} s^4 + \dot{\alpha}_{cs_{22}} s^3 + \dot{\alpha}_{cs_{23}} s^2 + \dot{\alpha}_{cs_{24}} s + \dot{\alpha}_{cs_{25}} \quad (3.2.46)$$

Here the comparison with the coefficients of closed-loop system with gain-scheduling controller is not possible since the computation of the closed-loop characteristic equation is too large and even the MATLAB® Symbolic Toolbox did not return the answer. The plots for the coefficients of the augmented trolley subsystem and augmented arm subsystem are shown in Figure 3.7 and 3.8. The plot shows that the coefficients are positive for all L and X_w and hence both closed-loop systems are stable. Moreover it can also be seen that for the trolley subsystem for $L \in [0.1, 0.8]$ the coefficients of the closed-loop characteristic equations for two controllers are closely matched. For the arm subsystem no graphical comparison can be given due to reasons mentioned above.

3.2.5 Section Summary

In this section both fixed-gain and gain-scheduling integral controllers have been designed independently for each subsystem. Both controllers are full state feedback but for the robust step reference tracking, integrators have been introduced into the system. The pole placement has been done using controllable canonical form for controller design. In the gain-scheduling controller the gains are function of L and X_w . The gain function for the arm subsystem is too messy and hence it has been simplified using surface fitting to estimate it using less complex polynomials. To save the processing power the robust fixed-gain controller

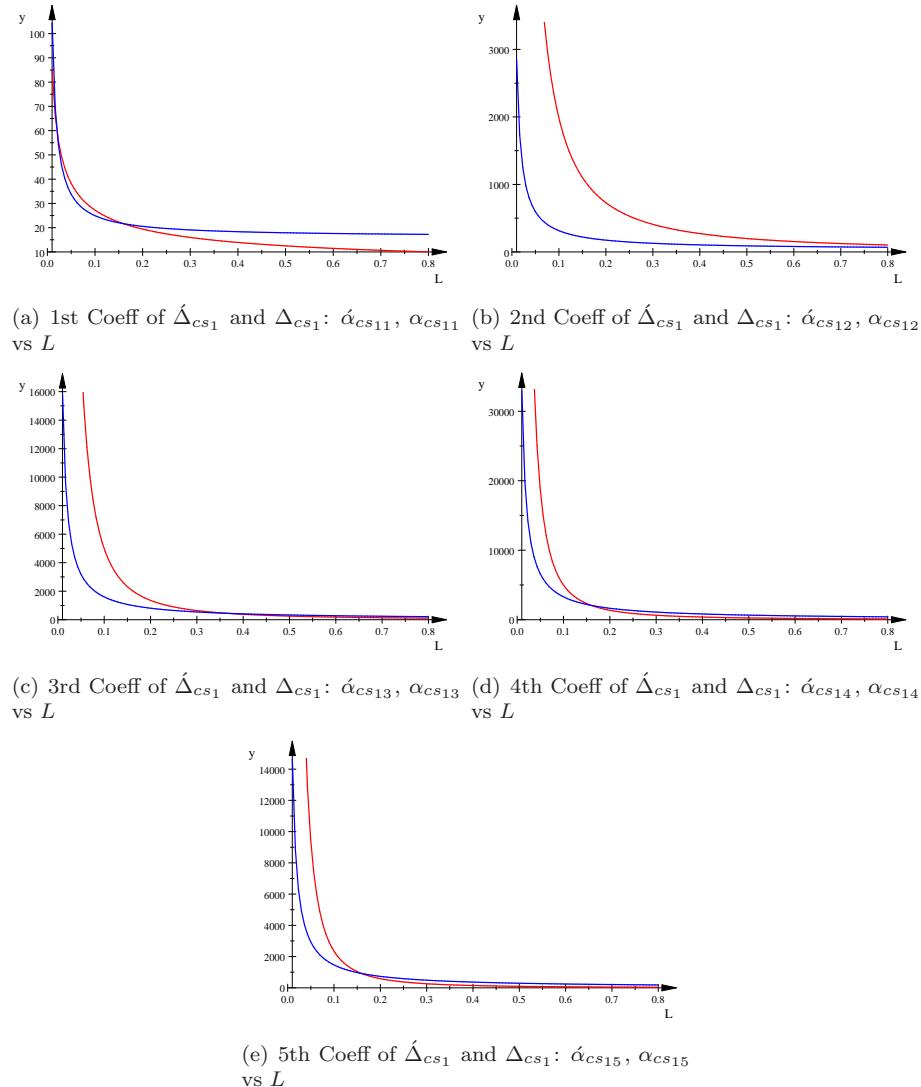


Figure 3.7: Plot of coefficients of the characteristic polynomial of the closed-loop trolley subsystem. Blue- For Fixed-Gain Controller, Red- For Gain-Scheduling Controller. Plot shows that all coefficients are greater than 0 and also except for $\dot{\alpha}_{cs_{12}}$ all other coefficients are quite closely matched in the region $L \in [0.1, 0.8]$. Thus according to the plots the closed-loop system under fixed-gain controller is stable $\forall L \in [0, 0.8]$ and would achieve the performance quite similar to the gain-scheduling controller at least for $L \in [0.1, 0.8]$

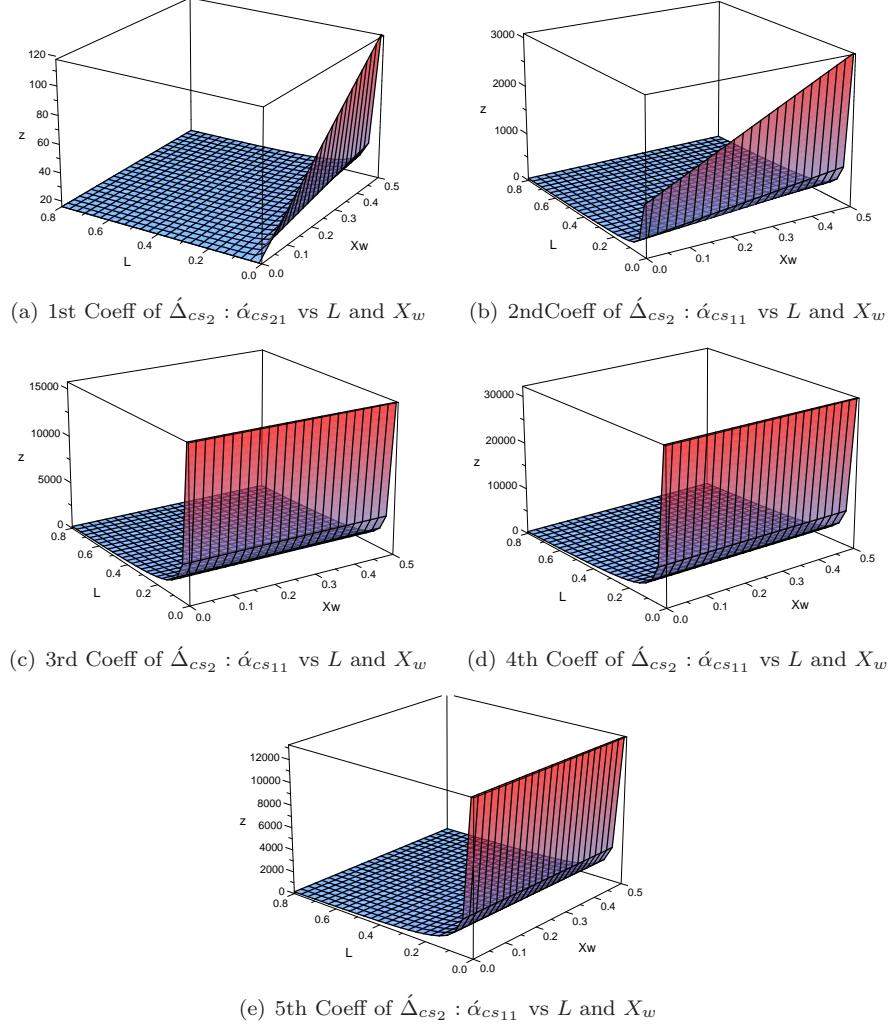


Figure 3.8: Plot of coefficients of the characteristic polynomial of the closed-loop Arm subsystem using fixed gain. The plot for the coefficients of characteristic equation for gain-scheduling controller are not shown as the coefficients computation is very messy which could not be done even with the help of Symbolic Math toolbox. So the above plots only show that for the fixed-gain controller (3.2.43) the closed loop system is stable for $L \in [0.1, 0.8]$ and $X_w \in [0, 0.5]$.

has been computed by evaluating the gain scheduling controller or linearizing the system at carefully selected operating point (X_w, L) . The graphical stability analysis has also been conducted and results shows that fixed-gain controller behaves well atleast in the region $L \in [0.1, 0.8]$ and $X_w \in [0.01, 0.5]$.

3.3 Simulation Results and Analysis

In this section the performance of the fixed-gain controllers (3.2.42) and (3.2.43) will be evaluated for the estimated model \mathbf{M}_2 ¹ given by (3.1.1) and for the simulation model \mathbf{M}_1 ² provided as a simulink block by manufacturer [1]. The block diagram of a complete 3D tower crane control system given in the Figure 3.9 has been used for simulation. The integral anti-windup scheme has also been incorporated to cater for locked rotor or any kind of actuator saturation. In the block diagram T_{t_1} and T_{t_2} are tracking time constants and both have been assigned the value of 1.25. The control system has been simulated with fixed-gains \mathbf{K}_1 and \mathbf{K}_2 given by (3.2.42) and (3.2.43) respectively. Note that to make our simulations close to reality we have also used the deadzone nonlinearity to capture the effects of stiction friction in the gearbox. According to our experimental testing the deadzone for trolley is given by $V_{dz_x} \approx 0.16667$ and for the arm it is given by $V_{dz_y} \approx 0.2283$

3.3.1 Model \mathbf{M}_2

First we will evaluate the performance of our fixed-gain controllers (3.2.42) and (3.2.43) for the model \mathbf{M}_2 given by (3.1.1). The step response of the system for $L = 0.6m$ are shown in Figure 3.10 and 3.11, respectively. The step response shows a good reference tracking and the payload oscillations are damped quite efficiently as well. The system has been simulated for various $L \in [0.1, 0.8]$ for step reference and the simulation result in Figure 3.12 shows that payload oscillations converge to \mathcal{B}_{r_1} within at maximum two periods of oscillation. Moreover, over the whole move the oscillation remains within \mathcal{B}_{r_2} . The response of both trolley and arm is smooth as well. To compare the performance of controller to damp payload oscillations, the closed-loop system has been compared with the open-loop system for the same triangular reference input used in Chapter 2 for model validation (Figure 2.17). The Figure 3.13(b) shows the plots of payload oscillation for both closed-loop and open-loop systems. The plot shows that the controller is performing quite well by damping the oscillation within almost 2 – 3 seconds whereas in the absence of controller the payload keeps on oscillating with almost the same large amplitude for whole simulation time. Moreover it is very interesting to note that the controller is not even allowing the payload to go out of the ball \mathcal{B}_{r_2} throughout the move.

3.3.2 Model \mathbf{M}_1

Now we will apply the designed controller (3.2.42)-(3.2.43) to the complete non-linear simulink model \mathbf{M}_1 provided by the crane manufacturer [1] for simulation purpose. This model is quite a good simulation of the real crane system. So if

¹Recall that we ourselves named the model given by (3.1.1) as \mathbf{M}_2 for the easy referral

²Recall that we ourselves named the model given in [1] as \mathbf{M}_1 for the easy referral

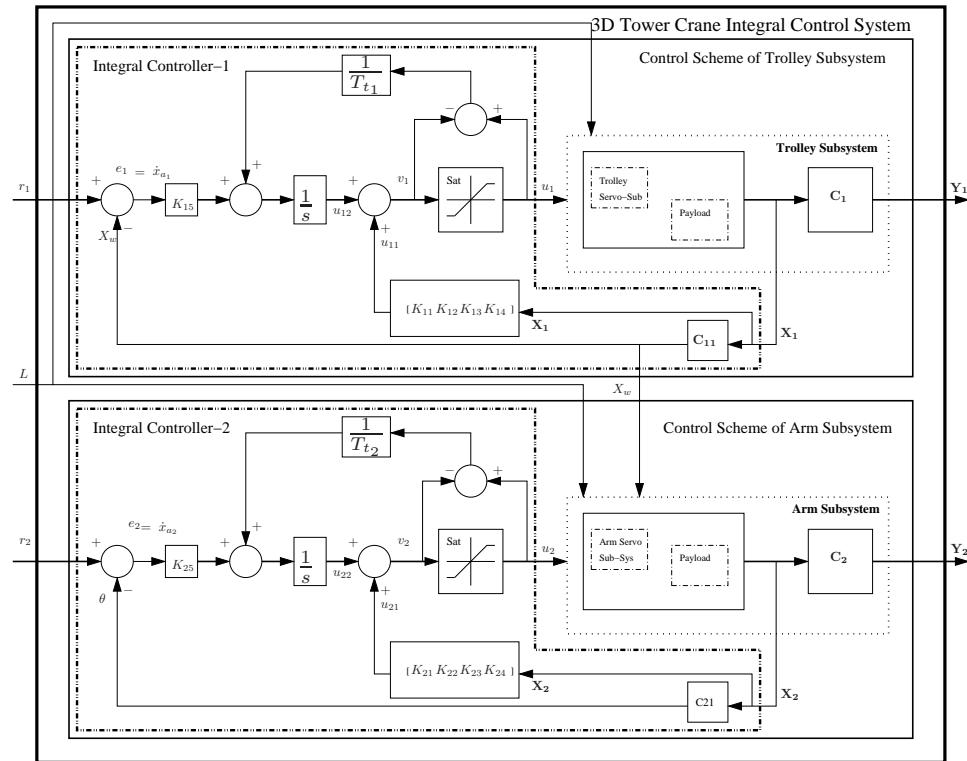


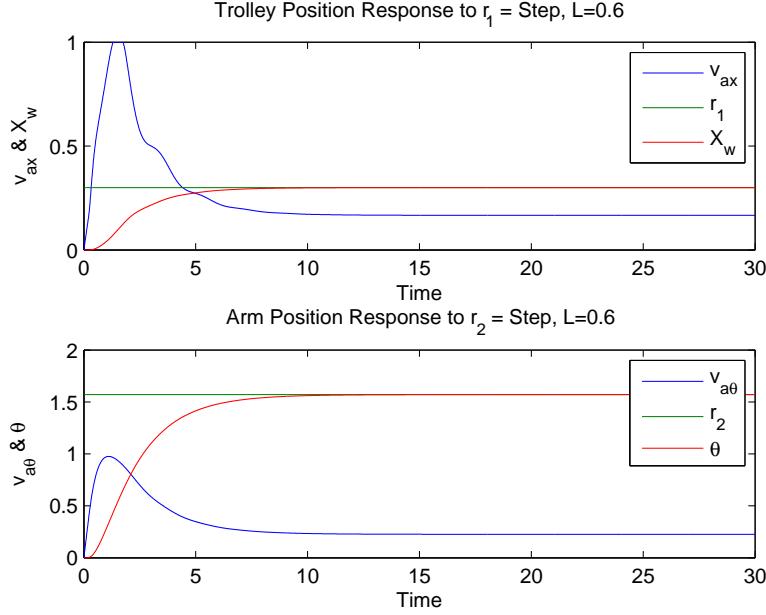
Figure 3.9: Block Diagram of Integral Control Scheme of 3D Tower Crane. The integral anti-windup scheme has also been used. Everything inside the dashed-dotted box constitutes the Integral Controller and inside the dotted box are the arm and trolley subsystems.

the controller works for \mathbf{M}_1 then it is highly likely it should also work for the real crane system. The success of the controller would also validate our identification process and hence the model (3.1.1) or (2.7.14). The Figure 3.14 shows the response of both \mathbf{M}_1 and \mathbf{M}_2 together in the same plot for the fixed-gain controllers (3.2.42) and (3.2.43). The response shows that there is almost no difference in the performance. The response of trolley and arm servo subsystems are almost exactly the same however the payload response differ slightly. Although the response profile is almost the same but the one for \mathbf{M}_1 has a bit larger amplitude as compared to that of \mathbf{M}_2 . Nevertheless the controller damps the payload oscillations for both systems efficiently and the oscillation converges to the ball \mathcal{B}_{r_1} within at maximum two periods of oscillation. Not only at steady-state but also throughout the move the payload remains within ball \mathcal{B}_{r_2} . Figure 3.15 shows the response of the system \mathbf{M}_1 for various sling lengths. According to the simulations the system \mathbf{M}_1 exhibits the nice behavior for all $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ so all control performance specification are met for both \mathbf{M}_1 and \mathbf{M}_2 . The simulation results of the model \mathbf{M}_2 -based controller for \mathbf{M}_1 also validated the identification phase and hence the estimated model \mathbf{M}_2 has also passed the Model validation Test-4 as mentioned in the previous chapter.

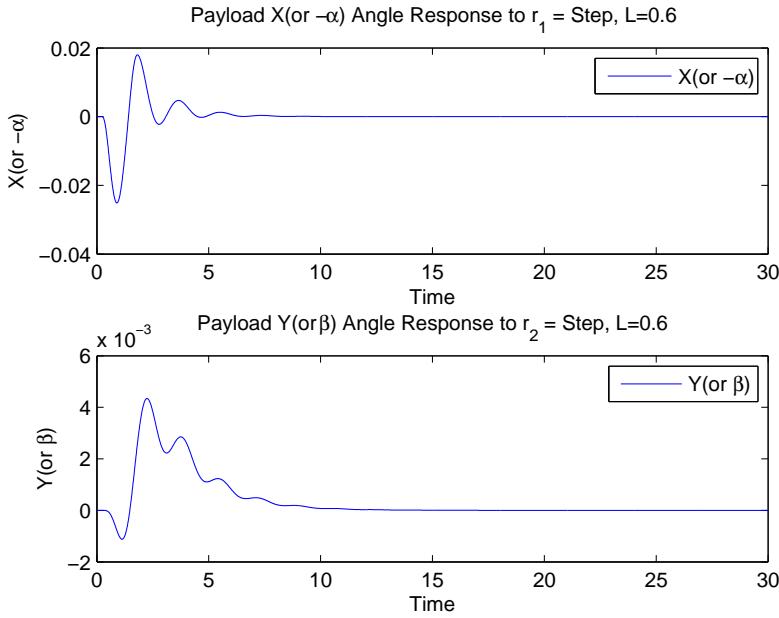
3.4 Chapter Summary

In this chapter, for simplifying the analysis, the crane model (3.1.1) has been divided into two linear parameter varying(LPV) subsystems namely the **Trolley Subsystem** given by (3.1.2) and **Arm Subsystem** given by (3.1.4). The **Trolley Subsystem** is parameterized in terms of sling length L and the **Arm Subsystem** is parameterized in terms of L and X_w . To achieve robust tracking for step reference input the Integral full state feedback Controller has been designed using pole placement in parameterized form using LPV forms. The pole placement has been done by transforming the system into controllable canonical form which gives two controllers namely \mathbf{K}_1 for **Trolley Subsystem** and \mathbf{K}_2 for **Arm Subsystem**. These controller gains are parameterized in terms of L and X_w . The actual parameterized form for \mathbf{K}_1 is given by (3.2.21). Although the actual closed form expression for \mathbf{K}_2 has also been computed but it is not feasible to show since it is too large. The controller \mathbf{K}_2 has been simplified using surface fitting and hence closely approximated using much simpler polynomials. The estimated expression for $\hat{\mathbf{K}}_2$ is given by (3.2.41) where its gain function elements are given by (3.2.27), (3.2.30), (3.2.33), (3.2.36) and (3.2.39). The estimates for $K_{21} K_{22}$ and K_{25} are good but that for K_{23} and K_{24} are not good for all values of L and X_w and hence requires further improvement. The gain-scheduling controllers pose a high computation overhead due to a lot of floating point multiplications so the implementation of such controller will require some considerable processing. Hence to save the processing power the fixed-gain controller has been devised. The fixed gains are obtained by carefully selecting the operating equilibrium point (L, X_w) and then evaluating the gain-scheduling expressions at the selected point. In the end the performance of the fixed-gain controller has been evaluated through simulations of the closed loop system for step and triangular reference inputs. The simulation results show that the fixed-gain controller works quite well not only for the model \mathbf{M}_2 but

also for \mathbf{M}_1 . Although the fixed-gain controller was designed on the basis of the identified model \mathbf{M}_2 , there is almost no difference between its performances for both \mathbf{M}_2 and \mathbf{M}_1 . Hence the model-based control design has also validated the identification process adopted in Chapter 2.

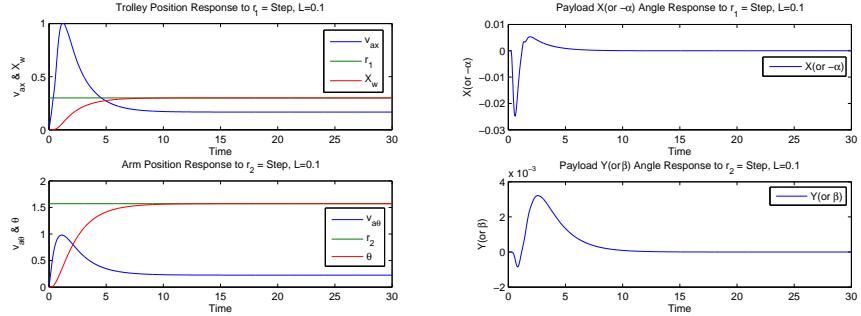


(a) Response of X_w and θ for $L = 0.6$. Red-Response, Blue-Control Action and Green-Reference Input. The step reference tracking is achieved with almost zero steady-state error. Note that control action also stays within limits and goes into saturation for very small time and recovers from the saturation quickly. The response is quite smooth. Note that due to deadzone (stiction friction) in gearbox, control signal settles to value > 0 that is equal to PWM input corresponding to stiction friction.



(b) Response of $-\alpha$ and β for $L = 0.6$. Oscillations have been killed quite efficiently and converges to B_{r_1} with $r = 0.005$ in less than 5 seconds.

Figure 3.10: Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.6\text{m}$. Both Inputs excited Simultaneously. The fixed-gain controller has been used which achieves all the control requirements mentioned in section 3.2.1. See the definitions of the balls B_{r_1} and B_{r_2} in the section 3.2.1.



(a) Response of X_w and θ for $L = 0.1$. Arm and trolley asymptotically tracks the step with load oscillations are killed and converges to the almost zero steady-state error.
(b) Response of $-\alpha$ and β for $L = 0.1$. Payload X and Y angle response to $r_1 = \text{Step}$, $L = 0.1$. Pay-load oscillations are killed and converges to the ball \mathcal{B}_{r_1} ($r = 0.005$) within 2 seconds.

Figure 3.11: Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.1\text{m}$. Both Input excited Simultaneously. Fixed-Gain controller works quite well for sling lengths as short as 0.1m .

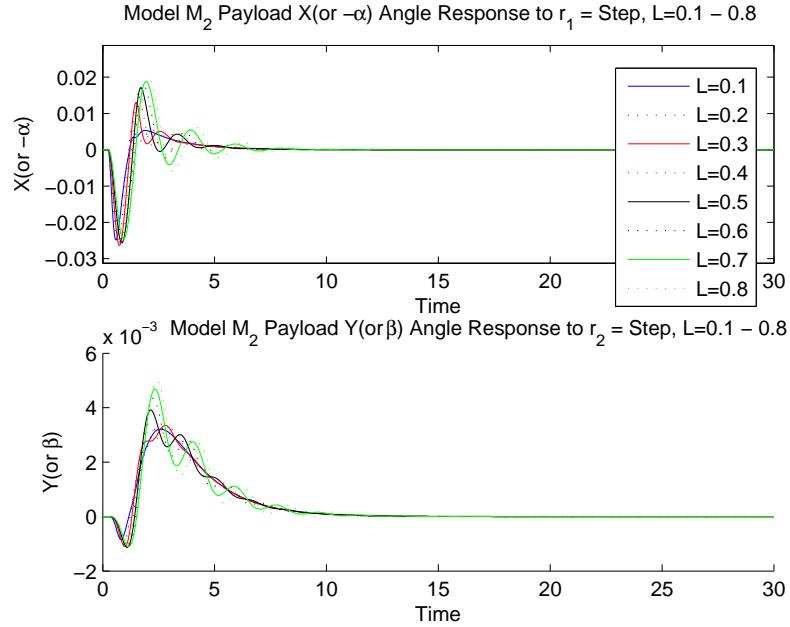
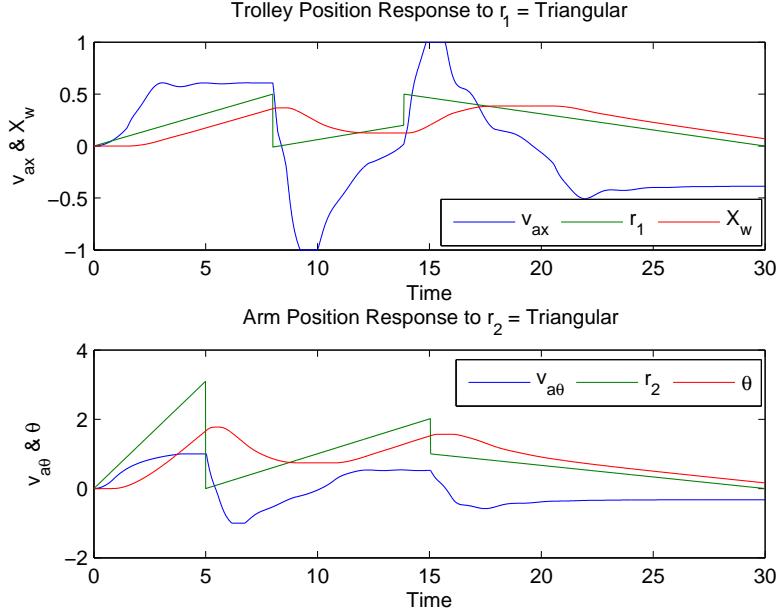
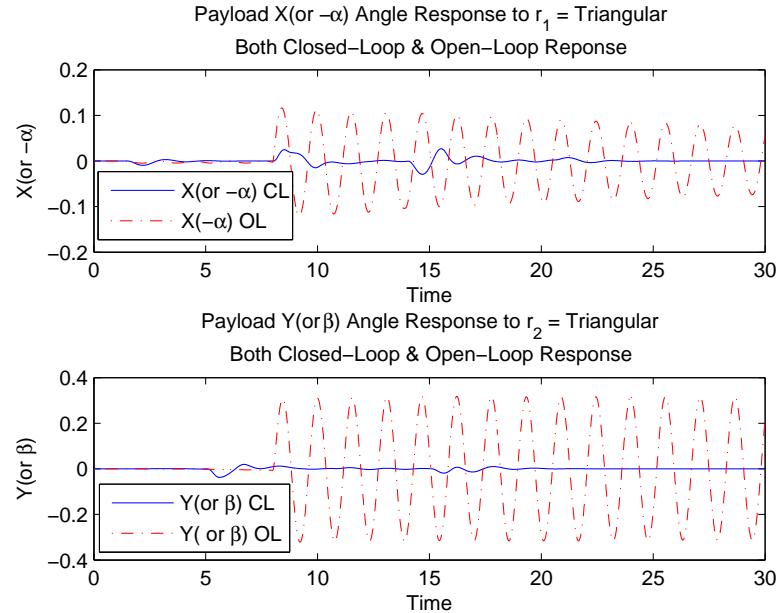


Figure 3.12: Payload angles response of crane's model \mathbf{M}_2 for various sling length $L \in [0.1, 0.8]$. Reference Input is Step. We can see that for $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ the control system behave nicely by killing the payload oscillations so the proposed fixed-gain controller works quite well at least in the region $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ and brings the payload oscillation to the ball \mathcal{B}_{r_1} of radius $r_1 = 0.005\text{m}$ at maximum within two periods of oscillations and also never leaves the ball \mathcal{B}_{r_2} of radius $r_2 = 0.05\text{m}$ during the whole move.

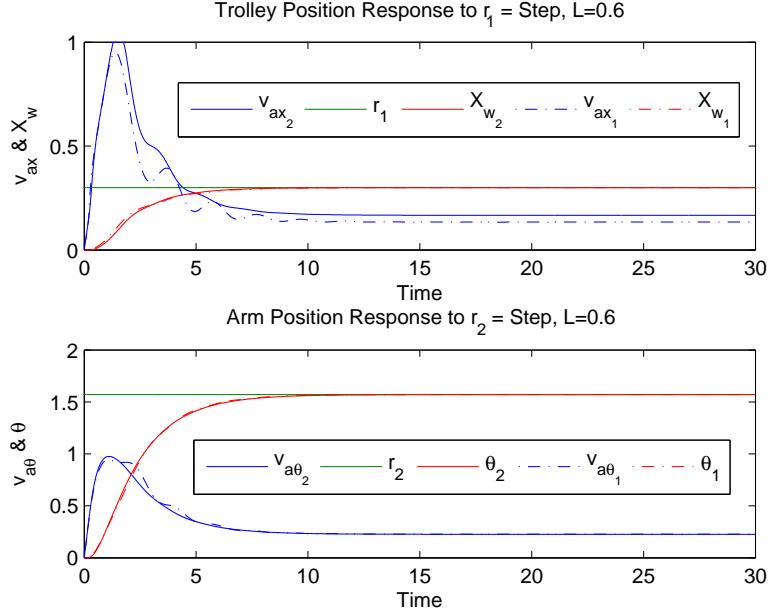


(a) Response of X_w and θ for $L = 0.6$. Although the controller has been designed for the robust tracking of step reference inputs but as shown above it also tracks the triangular type signal but with some tracking error.

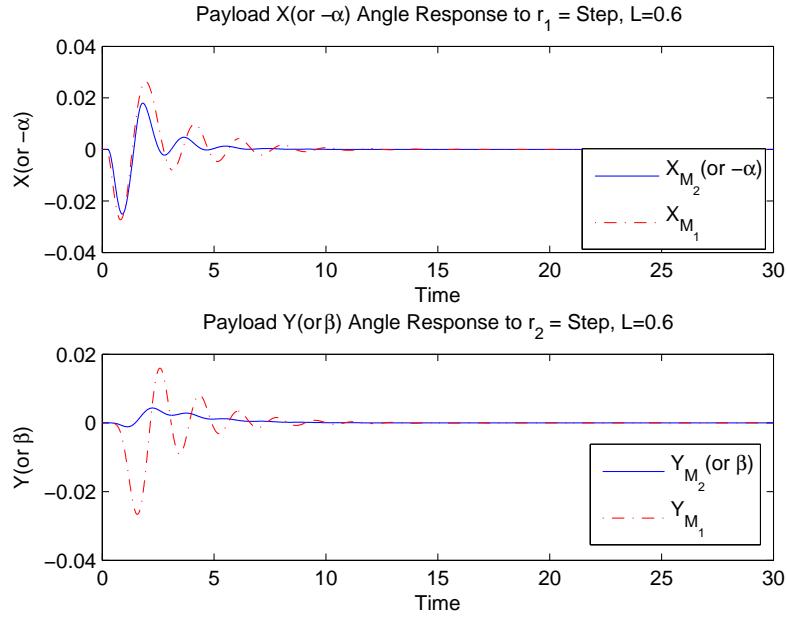


(b) Response of $-\alpha$ and β for $L = 0.6$. Blue Solid line shows closed-loop (CL) response and dashed-dotted line shows open-loop response. It can be seen that the controller has nicely killed the oscillations throughout the move.

Figure 3.13: Triangular Input Response of Crane's model **M₂** for sling length $L = 0.6\text{m}$. Both Inputs are excited Simultaneously. The simulation has been done to evaluate the performance of the closed-loop control system in comparison to open-loop response. We are especially interested to see the efficiency of controller to kill the payload oscillation shown in the subfigure 3.13(b).



(a) Response of X_w and $θ$ for $L = 0.6$. The solid lines show the response for the model \mathbf{M}_2 and the dashed-dotted line shows the response for the complete nonlinear simulation model \mathbf{M}_1 provided by manufacturer. For both models tracking of step is achieved with almost zero steady-state error.



(b) Response of $-α$ and $β$ for $L = 0.6$. The solid lines show the response for the model \mathbf{M}_2 and the dashed-dotted line shows the response for the complete nonlinear simulation model \mathbf{M}_1 provided by manufacturer. Oscillations have been killed quite efficiently and converges to the ball \mathcal{B}_{r_1} of radius $r_1 = 0.005$ within 5 seconds.

Figure 3.14: Step Response for sling length $L = 0.6\text{m}$. Both Input excited Simultaneously. The above plot shows that there is almost no difference between control system performance of both models so the controller designed on the basis of identified model M_2 also works perfectly for the complete nonlinear model M_1 and since the M_1 is a quite close description of the real system so it is quite likely that the same controller should also work quite well for the real system.

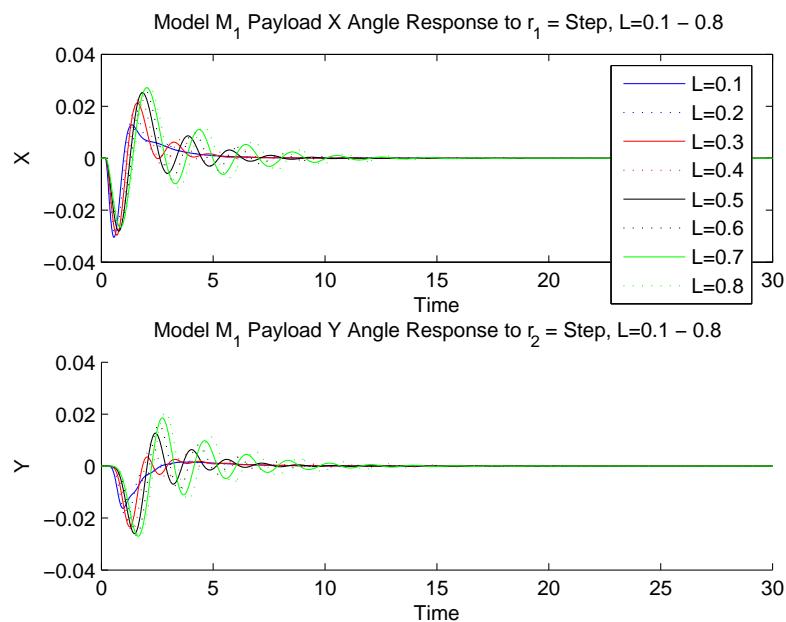


Figure 3.15: Payload angles response for various sling length $L \in [0.1, 0.8]$. Reference Input is Step. We can see that for $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ the control system behave nicely by killing the payload oscillations so the designed controller works quite well at least in the region $L \in [0.1, 0.8]$ and $X_w \in [0, 0.3]$ and drives the payload to the ball \mathcal{B}_{r_1} of radius $r_1 = 0.005$ at maximum within two periods of oscillations.

Chapter 4

Controller Design: Event-Triggered Sampling

The time-triggered controller using periodic sampling is not optimal for network control application and especially it becomes very expensive in terms of communication cost with increase in number of nodes or processes in network. Thus the periodic controller may not be good for control over wireless networks in terms of communication energy. The alternative is the control on the basis of event-triggered data samples. The first comparison between reimann(periodic) and lebesgue (event-triggered)sampling was given in [12] under stochastic settings where it has been proved that for low order systems, lebesgue sampling outperforms the reimann sampling in terms of low variance in state. The Event-Triggered control is quite an exciting and hot area especially for applications involving hardware having low computational capabilities and limited communication bandwidth. In such applications classical controller implementation techniques based on periodic sampling of a process are not optimal as they are normally very expensive due to unnecessary periodic execution of control even when it is not required. The sampling frequency in the periodic case is something which is not easy to decide as there is no hard and fast rule on how fast to sample. Normally people resort to some rule of thumbs but there is no sound theoretical reasoning behind them. So to optimally use the limited computational resource or the bandwidth of communication medium the efficient way is to sample only when something significant (event) happens in the process. Normally the event is triggered when the error in states crosses some certain threshold which depends upon the system dynamics and the controller structure. This technique has been especially explored over last few years [2, 9, 10, 11] for the applications involving sensor and actuator networks for embedded systems. The applications using Wireless Sensors and Actuators Networks (WSAN) are even more interesting to deal with using the event-triggered techniques due to limited bandwidth and energy conservation requirements for WSAN and for the first time, studied in detail in this chapter. In section 4.1 we will present theoretical analysis based on Lyapunov input-to-state stability theorem for deriving the sufficient conditions for asymptotic stability under event-triggered control policy. We will treat both regulation and tracking problems. In Section 4.2 we will do the feasibility analysis of event-triggered control for crane system on

the basis of the results developed in Section 4.1. In Section 4.3 we will discuss some implementation issues of event-triggered control and would also suggest some improvements and give some implementation details. In Section 4.4 the wireless closed-loop system under event-triggered control will be simulated using True-Time for both single crane and multiple crane case.

4.1 Event-Triggered Control Theory

In this section we would present theoretical results for event-triggered control. The main mathematical tool used in this development is the Lyapunov Input-to-State Stability(ISS) theorem. Our approach is primarily based on the results obtained in [2], [9], [10] and [11].

4.1.1 Lyapunov Input-to-State Stability

Let us consider the general linear system:

$$\dot{\mathbf{X}} = \mathbf{A} \mathbf{X} + \mathbf{B} \mathbf{u} + \mathbf{G} \mathbf{r} \quad (4.1.1)$$

$$\mathbf{Y} = \mathbf{C} \mathbf{X} \quad (4.1.2)$$

Note that $\mathbf{X} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{Y} \in \mathbb{R}^p$ and $\mathbf{r} \in \mathbb{R}^m$. Now suppose the asymptotically stabilizing controller $\mathbf{u} = \mathbf{K} \mathbf{X}(t)$ has already been designed in the continuous domain for the system (4.1.1). Now let's say the designed controller is operating on the sampled data $\mathbf{X}(t_k)$ instead of continuous $\mathbf{X}(t)$. So in this case the computed control signal will be given by:

$$\mathbf{u} = \mathbf{K} \mathbf{X}(t_k) \quad (4.1.3)$$

Where \mathbf{K} is $m \times n$ controller gain matrix. Now let's say the state $\mathbf{X}(t_k)$ is measured and sent to controller at time instant t_k then for the time $t \in [t_k, t_{k+1})$ between two measurements, the sensor measurement error $\mathbf{e} \in \mathbb{R}^n$ can be introduced as:

$$\mathbf{e}(t) = \mathbf{X}(t_k) - \mathbf{X}(t) \quad (4.1.4)$$

Now in the presence of this measurement error our controller will not be computing the actual control action. Instead it would be applying the control signal given by:

$$\mathbf{u} = \mathbf{K} (\mathbf{X}(t) + \mathbf{e}(t)) \quad (4.1.5)$$

Now using (4.1.5) in the system (4.1.1) we get the following closed-loop system under sample and hold implementation of the controller:

$$\dot{\mathbf{X}} = (\mathbf{A} + \mathbf{B}\mathbf{K}) \mathbf{X} + \mathbf{B}\mathbf{K}\mathbf{e} + \mathbf{G} \mathbf{r} \quad (4.1.6)$$

Now differentiating (4.1.4) we get the following measurement error dynamics:

$$\dot{\mathbf{e}} = -(\mathbf{A} + \mathbf{B}\mathbf{K}) \mathbf{X} - \mathbf{B}\mathbf{K}\mathbf{e} - \mathbf{G} \mathbf{r} \quad (4.1.7)$$

Now using (4.1.6) and (4.1.7) and taking \mathbf{e} as new state variable we write the following new closed-loop system:

$$\begin{pmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{e}} \end{pmatrix} = \begin{pmatrix} (\mathbf{A} + \mathbf{BK}) & \mathbf{BK} \\ -(\mathbf{A} + \mathbf{BK}) & -\mathbf{BK} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{e} \end{pmatrix} + \begin{pmatrix} \mathbf{G} \\ -\mathbf{G} \end{pmatrix} \mathbf{r} \quad (4.1.8)$$

Now we want to determine under which constraints on \mathbf{e} the system (4.1.8) will remain stable. For this illustration we will make use of the theorem from [3] for input to state stability (ISS). Here the measurement error \mathbf{e} is the input so in this context the system (4.1.6) can be defined as ISS under the following conditions (similar definition is also given in [2]):

Definition 4.1.1 Let there exists a \mathbb{C}^1 function $V(\mathbf{X}) : \mathbb{R}^n \rightarrow \mathbb{R}$ such that:

$$\alpha_1(\|\mathbf{X}\|) \leq V(\mathbf{X}) \leq \alpha_2(\|\mathbf{X}\|) \quad (4.1.9)$$

$$\frac{\partial V}{\partial \mathbf{X}} \dot{\mathbf{X}} \leq -W_3(\mathbf{X}), \quad \|\mathbf{X}\| \geq \rho(\|\mathbf{e}\|) > 0 \quad (4.1.10)$$

Where α_1, α_2 are class \mathcal{K}_∞ functions, ρ is a class \mathcal{K} function and $-W_3(\mathbf{X})$ is a continuous positive definite function on \mathbb{R}^n . If we can find such a function $V(\mathbf{X})$ that satisfies both (4.1.9) and (4.1.10) then the system (4.1.6) will be ISS with respect to measurement error $\mathbf{e} \in \mathbb{R}^n$ and the function $V(\mathbf{X})$ will be called an ISS Lyapunov function.

Now let us define the closed-loop system in the absence of \mathbf{e} as: $\mathbf{A}_{cs} = \mathbf{A} + \mathbf{BK}$. Now using $V(\mathbf{X}) = \mathbf{X}^T \mathbf{P} \mathbf{X}$ as a ISS Lyapunov function candidate where \mathbf{P} is obtained by solving Lyapunov equation:

$$\mathbf{A}_{cs}^T \mathbf{P} + \mathbf{P} \mathbf{A}_{cs} = -\mathbf{Q} \quad (4.1.11)$$

The first condition (4.1.9) is satisfied with:

$$\alpha_1(\|\mathbf{X}\|) = \lambda_{min}(\mathbf{P}) \|\mathbf{X}\|^2 \leq \mathbf{X}^T \mathbf{P} \mathbf{X} \leq \lambda_{max}(\mathbf{P}) \|\mathbf{X}\|^2 = \alpha_2(\|\mathbf{X}\|) \quad (4.1.12)$$

To establish the second condition (4.1.10), we proceed as follows:

$$\begin{aligned} \dot{V}(\mathbf{X}) &= \frac{\partial V}{\partial \mathbf{X}} \dot{\mathbf{X}} = \dot{\mathbf{X}}^T \mathbf{P} \mathbf{X} + \mathbf{X}^T \mathbf{P} \dot{\mathbf{X}} \\ \dot{V}(\mathbf{X}) &= \mathbf{X}^T \mathbf{A}_{cs}^T \mathbf{P} \mathbf{X} + \mathbf{e}^T \mathbf{K}^T \mathbf{B}^T \mathbf{P} \mathbf{X} + \mathbf{r}^T \mathbf{G}^T \mathbf{P} \mathbf{X} + \mathbf{X}^T \mathbf{P} \mathbf{A}_{cs} \mathbf{X} \\ &\quad + \mathbf{X}^T \mathbf{P} \mathbf{B} \mathbf{K} \mathbf{e} + \mathbf{X}^T \mathbf{P} \mathbf{G} \mathbf{r} \end{aligned}$$

Since both $\mathbf{X}^T \mathbf{P} \mathbf{B} \mathbf{K} \mathbf{e}$ and $\mathbf{X}^T \mathbf{P} \mathbf{G} \mathbf{r}$ are scalars so $\mathbf{X}^T \mathbf{P} \mathbf{B} \mathbf{K} \mathbf{e} = \mathbf{e}^T \mathbf{K}^T \mathbf{B}^T \mathbf{P} \mathbf{X}$ and $\mathbf{X}^T \mathbf{P} \mathbf{G} \mathbf{r} = \mathbf{r}^T \mathbf{G}^T \mathbf{P} \mathbf{X}$. Hence:

$$\dot{V}(\mathbf{X}) = \mathbf{X}^T (\mathbf{A}_{cs}^T \mathbf{P} + \mathbf{P} \mathbf{A}_{cs}) \mathbf{X} + 2\mathbf{X}^T \mathbf{P} \mathbf{B} \mathbf{K} \mathbf{e} + 2\mathbf{X}^T \mathbf{P} \mathbf{G} \mathbf{r}$$

Since \mathbf{A}_{cs} is stable under CT Controller \mathbf{K} so the Lyapunov equation (4.1.11) gives a positive definite solution where $\mathbf{Q} > 0$ and $\mathbf{Q} = \mathbf{Q}^T$. Hence:

$$\dot{V}(\mathbf{X}) = -\mathbf{X}^T \mathbf{Q} \mathbf{X} + 2\mathbf{X}^T \mathbf{P} \mathbf{B} \mathbf{K} \mathbf{e} + 2\mathbf{X}^T \mathbf{P} \mathbf{G} \mathbf{r} \quad (4.1.13)$$

Note that the 1st term is negative definite whereas 2nd and 3rd terms are sign indefinite and hence to establish their sign definiteness we will introduce some inequalities as follows. Recall that a positive definite quadratic function $\mathbf{X}^T \mathbf{Q} \mathbf{X}$ satisfies the following inequality:

$$\lambda_{min}(\mathbf{Q}) \|\mathbf{X}\|^2 \leq \mathbf{X}^T \mathbf{Q} \mathbf{X} \leq \lambda_{max}(\mathbf{Q}) \|\mathbf{X}\|^2$$

So using the following inequalities:

$$\lambda_{min}(\mathbf{Q}) \|\mathbf{X}\|^2 \leq \mathbf{X}^T \mathbf{Q} \mathbf{X} \quad (4.1.14)$$

$$\mathbf{X}^T \mathbf{P} \mathbf{B} \mathbf{K} \mathbf{e} \leq \|\mathbf{P} \mathbf{B} \mathbf{K}\| \|\mathbf{X}\| \|\mathbf{e}\| \quad (4.1.15)$$

$$\mathbf{X}^T \mathbf{P} \mathbf{G} \mathbf{r} \leq \|\mathbf{P} \mathbf{G}\| \|\mathbf{X}\| \|\mathbf{r}\| \quad (4.1.16)$$

we can ensure that RHS of (4.1.13) becomes greater than LHS and hence we get following inequality:

$$\dot{V}(\mathbf{X}) \leq -a \|\mathbf{X}\|^2 + b \|\mathbf{X}\| \|\mathbf{e}\| + c \|\mathbf{X}\| \|\mathbf{r}\| \quad (4.1.17)$$

Where:

$$a = \lambda_{min}(\mathbf{Q}), \quad b = 2 \|\mathbf{P} \mathbf{B} \mathbf{K}\|, \quad c = 2 \|\mathbf{P} \mathbf{G}\| \quad (4.1.18)$$

4.1.2 Regulation Control Problem

For the regulation problem $\mathbf{r} = 0$ the closed-loop system is given by:

$$\dot{\mathbf{X}} = (\mathbf{A} + \mathbf{B} \mathbf{K}) \mathbf{X} + \mathbf{B} \mathbf{K} \mathbf{e} \quad (4.1.19)$$

So for the above system, the inequality (4.1.17) becomes:

$$\dot{V}(\mathbf{X}) \leq -a \|\mathbf{X}\|^2 + b \|\mathbf{X}\| \|\mathbf{e}\| \quad (4.1.20)$$

Now we will use a part of $-a \|\mathbf{X}\|^2$ to dominate $b \|\mathbf{X}\| \|\mathbf{e}\|$:

$$\begin{aligned} \dot{V}(\mathbf{X}) &\leq -a(1 - \gamma) \|\mathbf{X}\|^2 - a\gamma \|\mathbf{X}\|^2 + b \|\mathbf{X}\| \|\mathbf{e}\| \\ \dot{V}(\mathbf{X}) &\leq -a(1 - \gamma) \|\mathbf{X}\|^2, \quad \forall \|\mathbf{X}\| \geq \frac{b}{a\gamma} \|\mathbf{e}\| \end{aligned} \quad (4.1.21)$$

Where $0 < \gamma < 1$. So the condition (4.1.10) is also satisfied with $W_3(\mathbf{X}) = -a(1 - \gamma) \|\mathbf{X}\|^2$ and $\rho(\|\mathbf{X}\|) = \frac{b}{a\gamma} \|\mathbf{e}\|$. Hence the control system (4.1.19) will be ISS with respect to measurement error \mathbf{e} if:

$$\|\mathbf{e}\| \leq \frac{a\gamma}{b} \|\mathbf{X}\| \quad (4.1.22)$$

It means the controller needs to be executed only when the measurement error goes out of the ball \mathbf{B}_μ with $\mu = \frac{a\gamma}{b} \|\mathbf{X}\|$. Now using $\sigma = \frac{a\gamma}{b}$ with $\gamma < 1$ the standard **control-execution rule** or **event-generation rule** can be stated as follows:

$$\|\mathbf{e}\| \geq \sigma \|\mathbf{X}\|$$

$$\sigma = \frac{a\gamma}{b}$$

(4.1.23)

Note that the above rule will be called as *standard event-generation rule*.

4.1.3 Tracking Control Problem

The tracking problem $\mathbf{r} \neq 0$ is not easy to handle. It will be dealt using a slightly different approach where we employ the special controller structure for tracking. Looking at (4.1.23) we can see that the event-generation rule involves only the states of plant and it will execute only either for the non-zero initial condition or for the disturbance at plant input otherwise it can never even start. So the standard event-generation rule (4.1.23) suffers from the so-called *self-start problem* and hence it cannot work for reference tracking. The solution is to incorporate the reference signal $\mathbf{r}(t)$ into the event-generation rule (4.1.23). But we cannot just simply add it to the rule since a reset mechanism is also required to make $\mathbf{e}(t)$ zero again after an event generation. The simple idea is to make $\mathbf{r}(t)$ part of the state by augmenting the system with integrator state as follows:

$$x_a = \int_0^t (\mathbf{r} - \dot{\mathbf{C}}\mathbf{X}) dt \quad (4.1.24)$$

Note that $x_a \in \mathbb{R}^{\dot{p}}$ where \dot{p} is the number of outputs to be tracked whereas the matrix $\dot{\mathbf{C}}$ has appropriate order i.e. $(\dot{p} \times n)$. Also note that we cannot track a higher number of reference inputs than that of control inputs so, essentially, the dimension of \mathbf{r} and x_a should be equal to that of \mathbf{u} . Now the system (4.1.1) can be rewritten as follows:

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{\mathbf{X}} \\ \dot{x}_a \end{pmatrix}}_{\dot{\mathbf{X}}_a} &= \underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{0}_{n \times \dot{p}} \\ -\dot{\mathbf{C}} & \mathbf{0}_{\dot{p} \times \dot{p}} \end{pmatrix}}_{\bar{\mathbf{A}}} \underbrace{\begin{pmatrix} \mathbf{X} \\ x_a \end{pmatrix}}_{\mathbf{X}_a} + \underbrace{\begin{pmatrix} \mathbf{B} \\ \mathbf{0}_{\dot{p} \times m} \end{pmatrix}}_{\bar{\mathbf{B}}} \mathbf{u} + \underbrace{\begin{pmatrix} \mathbf{0}_{n \times m} \\ \mathbf{I}_{\dot{p} \times m} \end{pmatrix}}_{\bar{\mathbf{G}}} \mathbf{r} \\ \dot{\mathbf{X}}_a &= \bar{\mathbf{A}}\mathbf{X}_a + \bar{\mathbf{B}}\mathbf{u} + \bar{\mathbf{G}}\mathbf{r} \end{aligned} \quad (4.1.25)$$

Note that $\mathbf{X}_a \in \mathbb{R}^{n+\dot{p}}$ and $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$ and $\bar{\mathbf{G}}$ are $(n+\dot{p}) \times (n+\dot{p})$, $(n+\dot{p}) \times m$ and $(n+\dot{p}) \times \dot{p}$ matrices respectively. Since we are aiming to asymptotically track the step reference so this problem can also be formulated as stabilization/regulation problem to non-zero origin. So here we would first shift the origin of the system to the desired equilibrium point using change of variables. Now lets introduce the desired state vector for equilibrium point:

$$\mathbf{X}_a^* = (r_1 \ 0 \ 0 \ 0 \ x_{a_1}^* \ r_2 \ 0 \ 0 \ 0 \ x_{a_2}^*)^T \quad (4.1.26)$$

The control required to keep the crane at \mathbf{X}_a^* is computed by putting $\dot{\mathbf{X}}_a = 0$:

$$0 = \bar{\mathbf{A}}\mathbf{X}_a^* + \bar{\mathbf{B}}\mathbf{u}^* + \bar{\mathbf{G}}\mathbf{r} \Rightarrow \mathbf{u}^* = \mathbf{0} \quad (4.1.27)$$

Now using change of variables:

$$\xi = \mathbf{X}_a - \mathbf{X}_a^* \quad (4.1.28)$$

$$\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}^* = \mathbf{u} \quad (4.1.29)$$

Using values of \mathbf{X}_a^* we get:

$$\xi = \begin{pmatrix} x_1 - r_1 \\ x_2 \\ x_3 \\ x_4 \\ x_{a_1} - x_{a_1}^* \\ x_5 - r_2 \\ x_6 \\ x_7 \\ x_8 \\ x_{a_2} - x_{a_2}^* \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_{a_1} \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_{a_2} \end{pmatrix} = \begin{pmatrix} \xi_1 - r_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 + x_{a_1}^* \\ \xi_6 - r_2 \\ \xi_7 \\ \xi_8 \\ \xi_9 \\ \xi_{10} + x_{a_2}^* \end{pmatrix} \quad (4.1.30)$$

Since \mathbf{X}_a^* is fixed so the dynamics of new state ξ is given by $\dot{\xi} = \dot{\mathbf{X}}_a$. Now using values of $x_1, \dots, x_4, x_{a_1}, x_5, \dots, x_8, x_{a_2}$ from (4.1.30) and $\tilde{\mathbf{u}} = \mathbf{u}$, the system (4.1.25) can be easily transformed into the the following new system with origin shifted to \mathbf{X}_a^* :

$$\dot{\xi} = \bar{\mathbf{A}}\xi + \bar{\mathbf{B}}\mathbf{u} \quad (4.1.31)$$

So by using change of variables we have transformed our problem of asymptotically tracking step into regulation problem but with non-zero origin. Now introducing the measurement error as:

$$\mathbf{e}(t) = \xi(t_k) - \xi(t) \quad (4.1.32)$$

Since reference signal is step in our case and hence it is fixed so there will be no measurement error in the reference signal for our current problem so the measurement error (4.1.32) can also be written as:

$$\mathbf{e}(t) = \mathbf{X}_a(t_k) - \mathbf{X}_a(t) \quad (4.1.33)$$

Now the closed-loop system under measurement error can be written as:

$$\dot{\xi} = (\bar{\mathbf{A}} + \bar{\mathbf{B}}\mathbf{K})\xi + \bar{\mathbf{B}}\mathbf{K}\mathbf{e} \quad (4.1.34)$$

Note that for our system, the system matrix $\bar{\mathbf{A}}$ and input matrix $\bar{\mathbf{B}}$ remains the same. Following the same procedure shown in the previous section the standard **control-execution rule** or **event-generation rule** for asymptotically tracking step signal can be stated as follows:

$$\begin{aligned} & \|\mathbf{e}\| \geq \sigma \|\xi\| \\ & \|\mathbf{e}\| \geq \sigma \|\mathbf{X}_a - \mathbf{X}_a^*\| \\ & \sigma = \frac{a\gamma}{b}, \gamma < 1, \mathbf{e}(t) = \mathbf{X}_a(t_k) - \mathbf{X}_a(t) \end{aligned} \quad (4.1.35)$$

It means the controller needs to be executed only when the measurement error goes out of the ball \mathbf{B}_μ with $\mu = \frac{a\gamma}{b} \|\mathbf{X}_a - \mathbf{X}_a^*\|$.

4.1.4 Computation of Minimum Inter-execution Time

According to [2] it has been guaranteed that no Zeno behavior would exist if we use the above event-generation rule. So there would exist a finite minimum inter-execution time and hence for any linear system it can be guaranteed that for stabilization, a finite number of controller execution would be required. According to the Corollary IV.I given in [2] the minimum inter-execution time τ_{min} can be computed as follows for two cases:

Case-I: Loop Delay $\Delta \neq 0$

In this case we have to execute the event generation rule bit earlier to cater for the delay. So instead of using $\|\mathbf{e}\| \geq \sigma \|\mathbf{X}\|$ we will use $\|\mathbf{e}\| \geq \dot{\sigma} \|\mathbf{X}\|$ where $\dot{\sigma} \leq \sigma$. But now a question arises on how to calculate $\dot{\sigma}$. We will use Corollary IV.I given in [2]. According to the corollary the $\dot{\sigma}$ is bounded by:

$$\sigma'_{min} \leq \dot{\sigma} \leq \sigma'_{max} \quad (4.1.36)$$

Where:

$$\sigma'_{min} = \frac{\Delta \|[\mathbf{A} + \mathbf{BK}|\mathbf{BK}] (\sigma + 1)}{1 - \Delta \|[\mathbf{A} + \mathbf{BK}|\mathbf{BK}] (\sigma + 1)} \quad (4.1.37)$$

$$\sigma'_{max} = \phi(-\Delta, \sigma) \quad (4.1.38)$$

Where $[\mathbf{A} + \mathbf{BK}|\mathbf{BK}]$ is a matrix formed by $[\mathbf{A} + \mathbf{BK}]$ followed by the columns of $[\mathbf{BK}]$ and $\phi(t, \phi_0)$ is the solution of the following differential equation:

$$\dot{\phi} = \|\mathbf{A} + \mathbf{BK}\| + (\|\mathbf{A} + \mathbf{BK}\| + \|\mathbf{BK}\|)\phi + \|\mathbf{BK}\| \phi^2 \quad (4.1.39)$$

with the initial condition $\phi(0, \phi_0) = \phi_0$. Note that we can choose any value for $\dot{\sigma}$ between σ'_{min} and σ'_{max} . Now the minimum inter-execution time can be computed as follows:

$$\phi(\tau, \sigma'_{min}) = \dot{\sigma} \quad (4.1.40)$$

and the minimum inter-execution time is given by:

$$\tau_{min} = \Delta + \tau \quad (4.1.41)$$

The values for τ_{min} can be computed by solving the differential equation (4.1.39) and then using the computed value of σ'_{min} in (4.1.40), we can solve for τ_{min} .

Case-II: Loop-Delay $\Delta = 0$

In this case note that when (4.1.22) will be violated at time instant $t = t_k$ then the event-generation rule (4.1.23) will be executed immediately and if there is no network delay Δ in the control loop then the measurement error will become zero again, $\mathbf{e} = \mathbf{X}(t_k) - \mathbf{X}(t_k) = 0$. So in this way the inequality (4.1.22) will become valid again and $\|\mathbf{e}\|$ will start growing again. As the system will start converging to the origin then the evolution of $\|\mathbf{e}\|$ will slow down and hence there will be less and less executions of the event-generation rule. The minimum inter-execution time will be given by solving the following for computed σ :

$$\phi(\tau_{min}, 0) = \sigma \quad (4.1.42)$$

The values for τ_{min} can be computed by solving the differential equation (4.1.39) and then using the computed value of σ in (4.1.42), we can solve for τ_{min} .

4.1.5 Event-Triggered Control Research Problems

There are a few open problems in Event-triggered control. Few of them are discussed below:

Tracking Systems

Tracking is not easy to implement. Tracking involves a time-varying reference signal $r(t)$. Two problems arise namely *self-start* and *reference-update problem*.

Self-Start Problem: Looking at (4.1.23) we can see that the event generation rule involves only the states of the plant and it will execute only for the non-zero initial condition and for the disturbance at plant input otherwise it can never even start. So the standard event-triggered rule suffers from the so-called *self-start problem* and hence it cannot work for reference tracking. The solution we proposed to handle it requires the dynamic controller but it brings other topological and implementation issues as well. So according to our proposed solution, for example, the static controller with feedforward gain cannot work for tracking under event-triggered sampling. So the current scheme can only utilize the dynamic controller structure for tracking. Further study is required in this context to make it possible to utilize static controller with feedforward gain to asymptotically track step reference. The problem is even more tricky for reference signals with higher order polynomials of time t . In this case normally the higher order derivatives of reference signal are also included in state-space which may pose further computational difficulties for generating the event.

Reference Update Problem: The normal topology in network control consists of a remote controller node close to the operator or command generator and sensor/actuator nodes near the plant. Since the reference signal is normally given at the controller node so passing the reference signal to the sensor node which is required for evaluating event-generation rule increases the communication cost and hence kills the purpose of event-triggered controller. This problem can be handled for the cases where the reference trajectory is known in advance. The cases where the reference is arbitrarily changing can not be easily handled with current settings as it would make the communication very expensive. The one possibility is to send the new reference signal value to the node whenever the controller update is triggered but it is not clear how the system will perform since in the event-generation rule it is assumed that continuous measurement of reference signal is available. Therefore, it is still an open research problem to devise a better method for tracking reference signals with higher order t -polynomials.

Dynamic Controller Issues

The proposed solution for the tracking under event-triggered sampling requires the use of dynamic controller structure that raised some other implementation difficulties as well. For example, to implement the numerical integration on the controller side using euler's approximation will no more be valid since the sampling interval h is no longer constant and hence, under varying sampling, the integration inside the controller becomes a big challenge and is still an open

research problem. We propose a simple solution to this problem where we treat extra state x_a as the state of the plant rather than the state inside controller. This way we can devise some mechanism to compute the integration using conventional methods. However, this would require some extra hardware. Another method is to calculate $\tau = t_{k+1} - t_k$ for each controller update and then use this variable time step for numerical integration instead of fixed ' h '. Hence, there are some possibilities which needs to be analyzed. Thus the implementation of dynamic controllers is an open research problem which needs further probing.

4.1.6 Section Summary

In this section we have derived the theoretical results for both regulation and tracking problems under event-triggered sampling. The Lyapunov ISS theorem has been used for derivation. The main idea illustrated in this section has been derived from research literature given in [2]. The main results obtained are given by (4.1.23) and (4.1.35).

4.2 Feasibility Analysis of Event-Triggered Control For the 3DOF Crane

4.2.1 Linearization of the Crane System

Now we will apply the methodology developed in the previous section to our 3D tower crane. For this purpose we will linearize our two augmented subsystems (3.2.2) and (3.2.4) around $(X_w, L) = (0.44, 0.16)$ and then we will combine them to get the complete state-space representation as follows:

$$\bar{\mathbf{A}} = \begin{pmatrix} \bar{\mathbf{A}}_1 & \mathbf{0}_{5 \times 5} \\ \mathbf{0}_{5 \times 5} & \bar{\mathbf{A}}_2 \end{pmatrix}$$

The linearization gives the following system matrix:

$$\bar{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -41.6667 & -0.8464 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -260.4167 & -66.6025 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -17.3303 & 0.0271 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 47.6583 & -61.3871 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.2.1)$$

The input matrix for the complete augmented crane system is given as follows:

$$\bar{\mathbf{B}} = \begin{pmatrix} \bar{\mathbf{B}}_1 & \mathbf{0}_5 \\ \mathbf{0}_5 & \bar{\mathbf{B}}_2 \end{pmatrix}$$

$$\bar{\mathbf{B}} = \begin{pmatrix} 0 & 0 \\ 0.4901 & 0 \\ 0 & 0 \\ 3.0629 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.9900 \\ 0 & 0 \\ 0 & -2.7225 \\ 0 & 0 \end{pmatrix} \quad (4.2.2)$$

The reference matrix can be written as:

$$\bar{\mathbf{G}} = \begin{pmatrix} \bar{\mathbf{G}}_1 & \mathbf{0}_5 \\ \mathbf{0}_5 & \bar{\mathbf{G}}_2 \end{pmatrix}$$

Note that both in $\bar{\mathbf{B}}$ and $\bar{\mathbf{G}}$, $\mathbf{0}_5$ is a column vector of order 5.

Reference vector is given by:

$$\mathbf{r}^T = (r_1 \ r_2) \quad (4.2.3)$$

Input vector is given by:

$$\mathbf{u}^T = (u_1 \ u_2) \quad (4.2.4)$$

State vector is given by:

$$\mathbf{X}^T = (x_1 \ x_2 \ x_3 \ x_4 \ x_{a_1} \ x_5 \ x_6 \ x_7 \ x_8 \ x_{a_2}) \quad (4.2.5)$$

The gain \mathbf{K} is a 2×10 matrix given by:

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_1 & \mathbf{0}_5 \\ \mathbf{0}_5 & \mathbf{K}_2 \end{pmatrix} \quad (4.2.6)$$

Where \mathbf{K}_1 and \mathbf{K}_2 are given by (3.2.42) and (3.2.43) and $\mathbf{0}_5$ is a zero row vector of order 5. The linearized system can now be written as:

$$\dot{\mathbf{X}} = \bar{\mathbf{A}} \mathbf{X} + \bar{\mathbf{B}} \mathbf{u} + \bar{\mathbf{G}} \mathbf{r} \quad (4.2.7)$$

The controller is given by:

$$\mathbf{u} = \mathbf{K} \mathbf{X} \quad (4.2.8)$$

Where \mathbf{K} and \mathbf{X} are given by (4.2.6) and (4.2.5) respectively. Now the closed-loop system can be written as:

$$\dot{\mathbf{X}} = \underbrace{(\bar{\mathbf{A}} + \bar{\mathbf{B}} \mathbf{K})}_{\mathbf{A}_{cs}} \mathbf{X} + \bar{\mathbf{G}} \mathbf{r} \quad (4.2.9)$$

4.2.2 Computation of σ , $\dot{\sigma}$ and τ_{min} :

Using above data for crane we have first calculated \mathbf{P} (4.2.9) using $\mathbf{Q} = \mathbf{I}$ in the lyapunov equation (4.1.11). Then using the computed \mathbf{P} , \mathbf{Q} , \mathbf{B} from (4.2.2) and \mathbf{K} from (4.2.6), the event-generation rule have been characterized for crane as follows for $\gamma = 1$:

Case-I Loop Delay $\Delta = 0$:

Value of σ : Using (4.1.35) and (4.1.18) we get:

$$\boxed{\sigma = 0.0080} \quad (4.2.10)$$

Value of τ_{min} : Using (4.2.10) in (4.2.11) we get:

$$\boxed{\tau_{min} = 26\mu s} \quad (4.2.11)$$

Case-II Loop Delay $\Delta \neq 0$: Now for $\Delta = 1\mu s$, the following values of $\dot{\sigma}$ and τ_{min} have been calculated using (4.1.37) and (4.1.38):

Value of $\dot{\sigma}$:

$$\dot{\sigma}_{min} = 0.000437 \quad (4.2.12)$$

$$\dot{\sigma}_{max} = 0.0077 \quad (4.2.13)$$

and we choose following value for $\dot{\sigma}$:

$$\boxed{\dot{\sigma} = 0.0077} \quad (4.2.14)$$

Value of τ_{min} : Using (4.1.40) we get:

$$\boxed{\tau_{min} = 25\mu s} \quad (4.2.15)$$

The evolution of $\phi(t, \dot{\sigma}_{min})$ is shown in Figure 4.1.

4.2.3 Section Summary

In this section we have explored the theoretical feasibility of controlling crane under event-triggered sampling and results shows that it is feasible.

4.3 Implementation Details, Issues and Improvements

This phase involves the real-time implementation and simulation of event-triggered control in Simulink environment using TrueTime kernel.

4.3.1 Implementation Issues

There are some issues with implementation of the event-generation rule for our current application and we will solve them as follows:

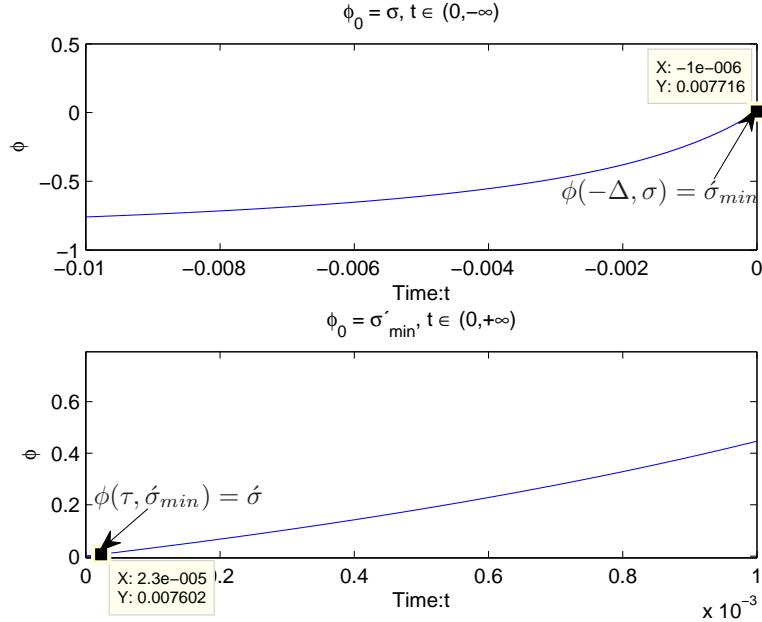


Figure 4.1: Evolution of ϕ for $\Delta \neq 0$. The annotation shows the values computed for $\dot{\sigma}_{min}$ and $\dot{\sigma}$.

Tracking Issues under ETC

As discussed earlier, the tracking problem under event-triggered sampling is not easy to handle and requires a dynamic controller. Since we have already proposed the integral controller in the last chapter, the requirement of a dynamic controller is already met. The event-generation rule depends on all the states of the system and since for our augmented system (4.2.7) the states $x_{a_1} = \int (r_1 - x_1)dt$ and $x_{a_2} = \int (r_2 - x_5)dt$ depend on the reference signal, it is not easy to compute the norm at the sensor node due to a reference update problem. since for our networked control application the controller will be at the remote place along with the reference command signal whereas the circuit implementing the event-generation rule will be at the plant node so the big question arise as how to pass the reference signal to the event generation circuit from controller node. If we send the reference signal to the sensor node from the controller node then it will increase the network cost and hence will kill the purpose of event-triggered control. So sending the reference signal over the network frequently is completely ruled out.Instead we propose the following scheme for the case where reference trajectory is known a priori:

- Pre-compute the whole reference trajectory and send it to the sensor node at startup altogether, which means each sensor node has full access to the reference signal at each time step. In other words we need to have a time function of reference trajectory inside each sensor node. Each sensor node need to have a timer which would be reset and started when a new move command will be available. During a move each sensor node has full access to the synchronized clock time ' t ' and $r(t)$, and so $r(t)$ can be

evaluated at each time step. We also assume that the reference command signal will not be modified too frequently. In our current application we are looking at step reference signals so we assume that the amplitude of step is known to each sensor node in advance for each move.

As derived in the last section, the event-generation rule for the tracking problem is given by (4.1.35) which is stated here again:

$$\boxed{\begin{aligned} \|e(t)\| &\geq \sigma \underbrace{\|\mathbf{X}_a(t) - \mathbf{X}_a^*(t)\|}_{\xi} \\ \|e(t)\| &\geq \sigma \|\xi\| \end{aligned}} \quad (4.3.1)$$

Where $\mathbf{X}_a^*(t)$ is the desired state vector given as follows:

$$\mathbf{X}_a^{*T} = (r_1 \ 0 \ 0 \ 0 \ x_{a_1}^* \ r_2 \ 0 \ 0 \ 0 \ x_{a_2}^*) \quad (4.3.2)$$

To compute the desired state vector we also need to calculate the final desired values $x_{a_1}^*$ and $x_{a_2}^*$ of the integrator states. They will be computed as follows: Looking at the control system block-diagram of Figure 3.9 it can be easily seen that for the trolley-subsystem

$$v_1 = u_{11} + u_{12}$$

Since at equilibrium $v_1 = 0$:

$$\Rightarrow u_{12}^* = -u_{11}^*$$

Since $u_{11}^* = K_1 \mathbf{X}_1^*$ and we also know that: $\mathbf{X}_1^{*T} = (r_1 \ 0 \ 0 \ 0)$ hence:

$$\Rightarrow u_{11}^* = K_{11} r_1 \Rightarrow u_{12}^* = -K_{11} r_1$$

Since in closed-loop control system configuration shown in the block-diagram 3.9, $x_{a_1}^* = u_{12}^*$ and $x_{a_2}^* = u_{22}^*$

$$\Rightarrow x_{a_1}^* = -K_{11} r_1 \quad (4.3.3)$$

and similarly for the arm subsystem:

$$x_{a_2}^* = -K_{21} r_2 \quad (4.3.4)$$

Now the desired state vector is given by:

$$\boxed{\mathbf{X}_a^{*T} = (r_1 \ 0 \ 0 \ 0 \ -K_{11} r_1 \ r_2 \ 0 \ 0 \ 0 \ -K_{21} r_2)} \quad (4.3.5)$$

Dynamic Controller Implementation Issues under ETC

As discussed before, dynamic controllers are not easy to implement under event-triggered sampling. The problem arises due to non-uniform sampling. Since the sampling interval is not constant, the digital implementation of the dynamic controller becomes a big challenge under varying sampling interval. We propose following solution to this problem:

- Implement the dynamic part of the controller inside the event-generation circuit¹. It means the circuit or the sensor node which is measuring the x_1 should also compute $x_{a1} = \int (r_1 - x_1)dt$ and the one measuring x_2 should also compute $x_{a2} = \int (r_2 - x_2)dt$. The computation for the integral will be done either in continuous domain using an analog integrator circuit or in the digital domain using periodic sampling. The integrator states x_{a1} and x_{a2} will also be multiplied with the gains K_{15} and K_{25} respectively inside event-generation circuit and then we will pass the augmented state vector \mathbf{X} to the controller node where the rest of control action will be computed. Note that we can also compute the time interval between current and past event and use it inside controller node to implement the integration. Both are possible but it has been observed that the former implementation gives better performance.

4.3.2 Deficiencies and Improvement of Event-Generation Rule

The event-generation rule (4.3.1) has three deficiencies:

Degree of Freedom is limited

In the standard event-generation rule there is only one degree of freedom given by parameter σ to tune the performance of the control system. Sometimes it is required to give more weight to some states relative to others. In such situation the standard event-generation rule will not allow to achieve this purpose since the only available parameter to tune is σ and if we change that it would affect the response of all states. Sometimes we do not care much about some of the states so to reduce their share in the event-generation rule we need to add extra degree of freedom in the system which we can do by introducing the diagonal weighting matrix \mathbf{W} as follows.

$$\|e(t)\| \leq \sigma \|\mathbf{W} \xi\| \quad (4.3.6)$$

where the matrix \mathbf{W} is given by:

$$\mathbf{W} = \text{diag}(w_1, w_2, \dots, w_{10}) \quad (4.3.7)$$

The weights $w_1, w_2 \dots w_{10}$ should be greater than 1 and they can be varied to fine tune the performance.

Quantization Effect at Equilibrium

When the system will converge to the desired state then the numerical values of both $e(t)$ and $\varepsilon(t)$ will be ≈ 0 and hence due to quantization effects the false or undesired triggering may occur. To address this issue the small offset δ will be added to the RHS of (4.3.2) to avoid this problem.

¹The circuit used to generate the event or trigger when the event-generation rule is violated. It will be covered in next section.

Minimum Sampling Time or Greediness

In the standard event-generation rule there is no control on the actual minimum inter-execution time or minimum sampling interval denoted by τ_{min} . In case something goes wrong with one of the system in the network then due to greediness of the event-triggered control policy the whole network may become congested and hence can destabilize all the plants which were normally operating otherwise. Thus, to save the network from congestion we would introduce another degree of freedom in the original event-generation rule by incorporating the parameter τ_{mina} which would ensure that certain system doesn't use more than certain amount of resources. The value is selected by the user on the basis of a rough idea about system's bandwidth. The time between every two consecutive triggering instants is measured and if it is greater than τ_{mina} only then controller is executed otherwise no update is required. The algorithm is given by Algorithm 1.

Algorithm 1 To remove Greediness

```

if  $\tau = t_k - t_{k-1} > \tau_{mina}$  then
    Update the Controller
else
    Do not Update
end if

```

Now incorporating all the changes an *Improved Event-Generation Rule* is given by:

$$\|e(t)\| \geq \sigma \|\mathbf{W} \xi\| + \delta \quad (4.3.8)$$

Note that the minimum inter-execution time check is not the part of the rule (4.3.8) explicitly but it will be implemented separately as mentioned in Algorithm 1.

4.3.3 Parametric Tuning of Event-Generation Rule

Now instead of one we have four parameters σ , δ , τ_{mina} and \mathbf{W} in the improved event-generation rule (EGR). This gives a greater degree of freedom to tune the performance according to the given conditions. Tuning of these parameters depends on the trade-off between network performance and control performance. The tuning of EGR should be done in the following sequence:

Tuning using σ :

The value of σ directly affects the control performance. One should start with the value obtained on the basis of Lyapunov ISS theorem (See Section 5.2). Keeping all other parameters constant, $\mathbf{W} = \mathbf{I}$, $\delta = 0$ and $\tau_{mina} = 0$, we should increase σ at first place to find some good compromise between control and network performance. For large values of σ we get less controller executions and hence the better network performance but it also reduces the control performance.

Tuning using offset δ :

If we cannot find any suitable compromise between network and control performance using σ alone then we can use δ to fine tune the performance. We can reduce the sampling interval by increasing δ but there is some upper limit on it because if we would increase it beyond that then it would badly affect the transient response due to delayed controller execution and hence it would induce more errors at steady-state as well as more control executions will be the result eventually. So there is a compromise in choosing δ . The event-generation policy can be tuned by starting with $\delta = 0$ and then increasing it in small steps and analyzing the overall performance both in terms of control and network.

Tuning using τ_{mina} :

If the network control performance is still not good then we can introduce τ_{mina} parameter to reduce the network cost. So if we have priori information about the appropriate sampling interval for the system then we can use that information to decide upon τ_{mina} . This would ensure that the minimum interval between two events is never less than some user-controlled value. We can reduce the network cost by increasing τ_{mina} but it would degrade the control performance. This gives us the extra degree of freedom to have the tighter control on the system's performance which would enable us to bring in good compromise between network and control performance.

Tuning using Weighting Matrix \mathbf{W} :

This parameter acts as the large tuning knob which should be used as a last resort to enhance the network performance. By tuning $\{w_1, w_2 \dots w_{10}\}$ we can control the effect of each state on the event-generation. Note that for $w_1 = w_2 = \dots = w_{10} = 1$ we get the standard event-generation rule. The values greater than 1 means that we are allowing larger error in the state before the event is triggered and hence the effect would be to reduce the sampling interval. For values less than 1 we are allowing smaller errors in state before the event is triggered and hence the smaller sampling interval is the result but values less than 1 are not used because our derived event-generation rule is already conservative.

4.3.4 Event Detection Circuit

One of the main difficulties of the event-triggered control is the detection of the event and generation of the trigger for the controller update. This requires dedicated hardware. The task becomes even more difficult in case of dynamic controllers. As we mentioned above that integral part of the dynamic controller is not easy to implement digitally inside the controller node due to the varying sampling time. So according to our proposed solution the integral action will also be implemented inside the dedicated hardware named as Event-Detection circuit which may be either analog or digital. The event detection circuit is shown in Figure 4.2. This circuit achieves two functions, namely, event-generation and integral part of the controller. It has two modules, the Integrator Module \mathcal{IM} and the Event Generation Module \mathcal{EGM} . The module \mathcal{IM} implements the integral part of the controller. The module \mathcal{EGM} implements the event generation

rule (4.3.8). This circuit can be materialized in both digital and analog. Since our goal is to minimize the network usage and not the processor time, a solution around dedicated digital hardware like MCU (micro-controller unit) can be proposed. In case of a digital implementation, we can use periodic sampling of the plant and hence approximate the integral action as usual and evaluate the event-generation rule as well. In case of analog we can employ the operational amplifier with dynamic feedback circuit to implement the integration but implementation of \mathcal{EGM} using analog hardware will not be easy.

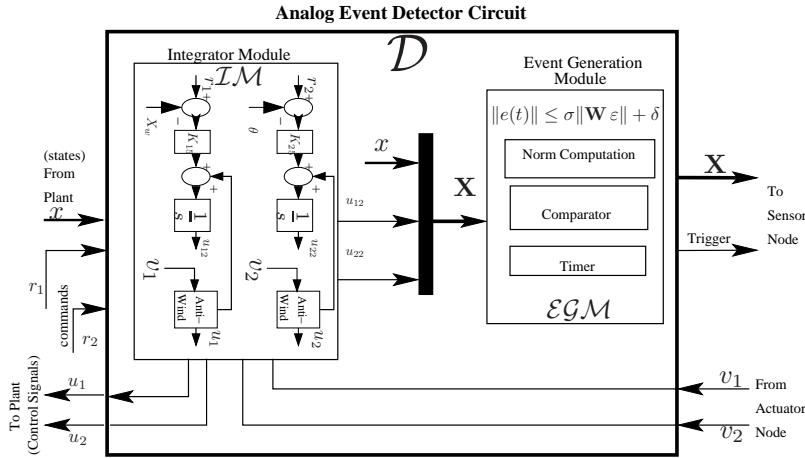


Figure 4.2: Block Diagram of Event Detector Circuit. There are two modules inside this circuit namely the Integrator Module (\mathcal{IM}) and Event Generation Module (\mathcal{EGM}). The \mathcal{IM} implements the dynamical part of the controller that is the integral part for our controller. If there is no dynamics inside the controller then this circuit can be bypassed. The \mathcal{EGM} implements the event triggering rule given by (4.3.8) It generates the trigger whenever rule is violated. Note that this event detection circuit can be implemented using dedicated analog and/or digital hardware. The full practical details for implementation of this circuit are not obvious yet and beyond the scope of this thesis.

4.3.5 Control of Single Crane over WSAN

The block diagram of event-triggered control of a single crane is shown in Figure (4.3). The event detector circuit measures the plant states x and \mathcal{IM} computes the augmented integrator states x_{a1} and x_{a2} . Then, \mathcal{EGR} evaluates the event-generation rule (4.3.8) and generates the trigger and places the state data on the output bus \mathbf{X} . The trigger has been interfaced with the hardware interrupt of the sensor node. After some setup time the sensor node reads the plant states \mathbf{X} from the event detector circuit output bus and then pass it to the controller node over a IEEE 802.15.4 wireless network. The controller computes the control signal and then pass it back to the actuator node which is collocated with sensor node. The actuator node sends out the computed control signal \mathbf{v} on the D/A lines to the event detection circuit where the \mathcal{IM} module apply the anti-windup scheme and then send the control signal \mathbf{u} to plant actuators.

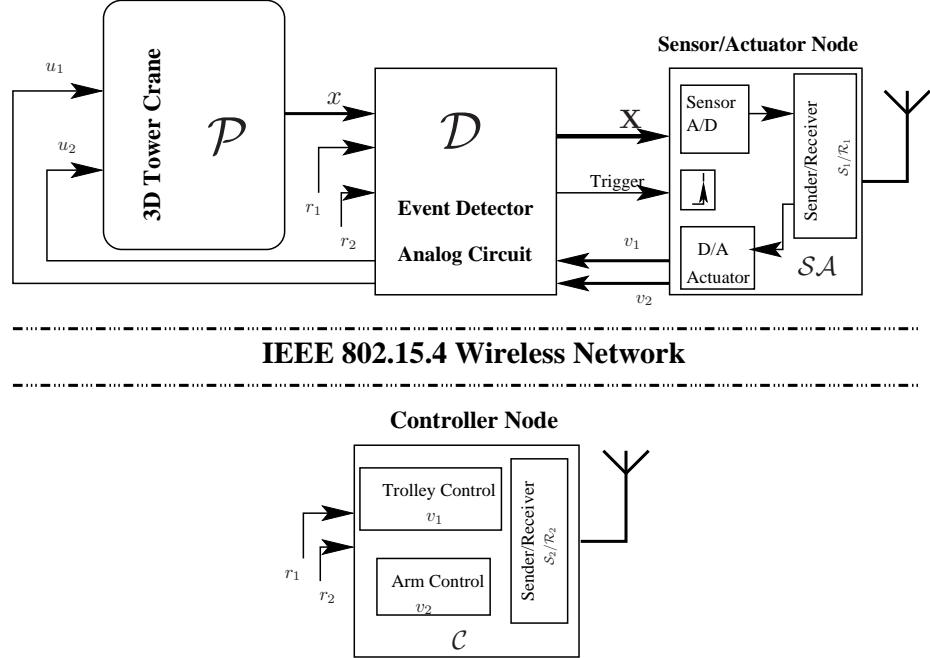


Figure 4.3: Block Diagram of Event-Triggered Control Scheme.

4.3.6 Control of Multiple Cranes over WSAN

This task involves controlling two or more cranes using a single event-triggered controller over a wireless network where the controller node is at a remote location. The scenario is similar to one operator controlling two cranes simultaneously. In case of a dynamic controller it is necessary to keep the state of the controller in memory for each crane. For example for our case of an integral controller the integrator state must be retained in memory separately for each crane. This is similar to the scenario where the operator keeps the record of history of control action applied to each crane. In our current implementation this requirement is being naturally met since we have implemented the integrator inside the event detection circuit where the states are being kept in memory. The block diagram for the closed-loop system is shown in Figure 4.4. It is important to note that only one crane can communicate with the controller at a time. This scenario may introduce time varying delay and packet losses due to message collision which makes the task quite challenging. The network topology includes one controller node, \mathcal{C} , and two sensor/actuator nodes, \mathcal{SA}_1 and \mathcal{SA}_2 .

4.3.7 Section Summary

In this section we have proposed the solution for issues related with tracking of a 3DOF tower crane under event-triggered control. It has been identified that a dynamic controller is not easy to implement under event-triggered sampling as the numerical integration becomes a challenge. We have proposed a solution for this where the controller integrator state has been treated as an augmented

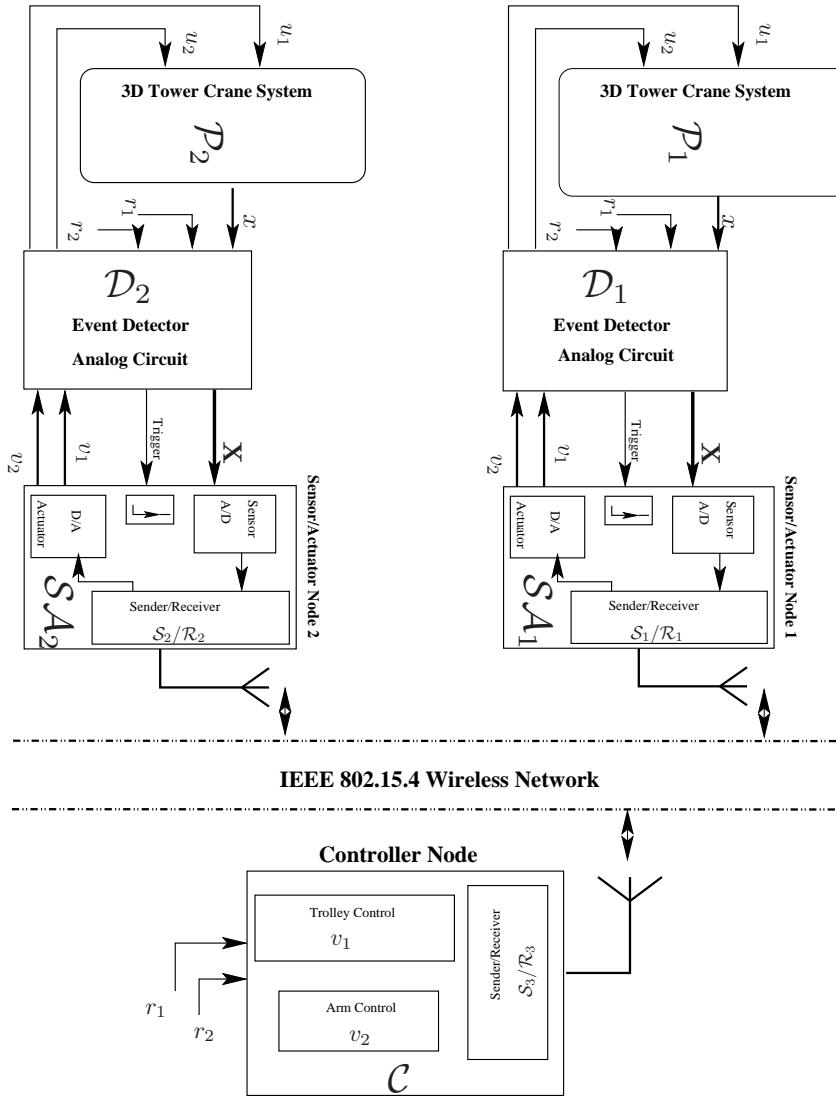


Figure 4.4: Block Diagram of Event-Triggered Control Scheme for Two cranes. One Controller is controlling two cranes. Communication is half duplex. Only one crane can communicate with controller at a time. Controller updates only one plant actuator at a time.

plant state and hence implemented on the plant side using dedicated hardware. We have also identified some issues with the standard event-generation rule and suggested some improvements which would give more freedom to the control designer for tuning event-triggered control system. We have also outlined the rough sketch of event-detection circuit and identified the main modules that would be used for its implementation. In the end the conceptual block-diagram of the wireless event-triggered control system setup has been shown for both single and multiple crane case. These block diagrams illustrates the main components and layout of control system.

4.4 True-Time Simulation Results and Analysis

Now following the above manual tuning guidelines we got the following parameter values for the crane system which asymptotically track the step reference:

$$\boxed{\begin{aligned} \sigma &= 0.075, \quad \delta = 0.01, \quad \tau_{mina} = 0.04, \quad \mathbf{W} = \mathbf{I}_{10} \\ \|e(t)\| &\geq 0.075\|\mathbf{I}_{10}\xi\| + 0.01 \end{aligned}} \quad (4.4.1)$$

Where \mathbf{I}_{10} is 10×10 identity matrix. Note that the actual value for σ that asymptotically stabilized the system is almost 10 times higher than the computed value given in (4.2.10). Hence it shows that theoretical results for event-generation rule (4.1.23) or (4.1.35) are quite conservative and it gives only sufficient but not necessary conditions. The event-triggered control systems (4.3) and (4.4) have been simulated, using TrueTime, in Simulink®. We will now present the results for both single and multiple crane control scenarios. The settings of the parameters for IEEE 802.15.4 network and control loop delays used in simulation are given in the Table 4.1. The TrueTime network block has been used for simulation of IEEE 802.15.4 protocol with non-beacon based modality which uses an un-slotted CSMA/CA algorithm shown in Figure 4.5.

Table 4.1: IEEE 802.15.4 Network parameter values and delays used in the simulation.

Parameters	Values
Data Rate [kbps]	250
Retry Limits	2
Path Loss Function	$1/d^{3.5}$
Minimum Frame Size [bits]	272
macMinBE	3
aMaxBE	5
macMaxCSMABackoffs	4
Sensor-Controller Delay, Δ_{sc} [s]	0.0005
Controller-Actuator Delay, Δ_{ca} [s]	0.005
Actuator-Plant Delay, Δ_{ap} [s]	0.0005

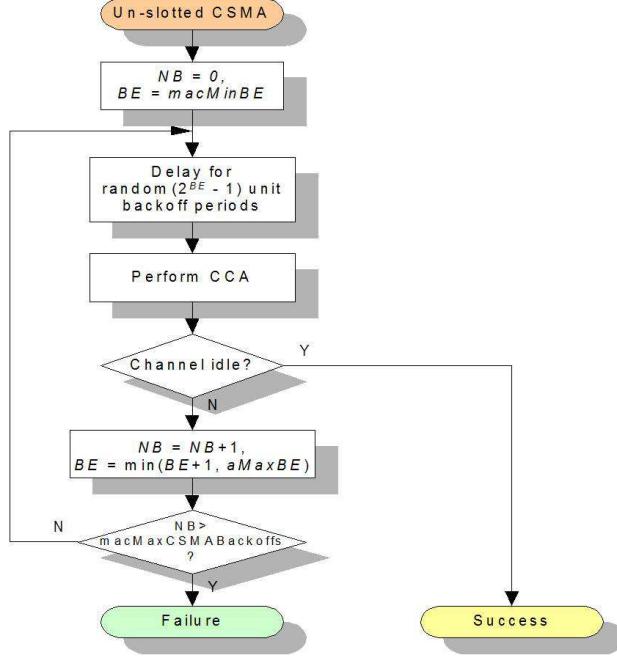


Figure 4.5: IEEE 802.15.4 unslotted CSMA/CA algorithm with non-beacon mode.

4.4.1 Simulation of Control of a Single Crane over IEEE 802.15.4

The control system shown in Figure 4.3 has been implemented in TrueTime and its simulation model is shown in Figure 4.6. Figure 4.7 shows the step response and Figure 4.8 shows the schedules of network and controller tasks. We can see that the controller inter-execution time increases as the system converges to the desired state. During the transient response the event-triggered controller executes more frequently to keep the system on the right path from very start and hence ensuring the smooth response. We have achieved almost similar tracking performance as with periodic implementation but with more than 4 times less controller executions and network usage. So the event-triggered controller is performing quite well for this wireless networked control application. The detailed analysis will be given in next chapter where we would give more analytical comparisons.

4.4.2 Simulation of Control of Multiple Cranes over IEEE 802.15.4

The control system shown in Figure 4.4 has been implemented in TrueTime and its simulation model is shown in Figure 4.10. Figure 4.11 shows the step response and Figure 4.12 shows the schedules of network and controller tasks. We can see that the controller inter-execution time increases as the system converges to the desired state. We have achieved almost similar (See Section 5.2.1) tracking performance as with periodic implementation but with more than

4 times less controller executions and network usage. Both systems exhibits the nice performance and that too with very low communication cost. So the event-triggered controller is performing quite well for this wireless networked control application. The detailed analysis will be given in next chapter where we would give more analytical comparison.

4.5 Chapter Summary

In this chapter we have presented theoretical results, practical details and simulation results for event-triggered control for both single and double crane case. The event-generation rule has been derived on the basis of the Lyapunov ISS theorem and the ideas given in [2]. The theoretical result obtained for event-generation rule suffers from some practical issues which have been addressed with some improvements. Theoretical results are also quite conservative. For the regulation problem with static controller, the implementation is not hard but the main issues arise when we deal with the tracking problem using a dynamic controller structure. A Dynamic controller is not easy to implement under event-triggered sampling due to varying sampling interval and may require some extra hardware according to our proposed solution. According to simulation results, the event-triggered control performs well and will achieve almost the same control performance under nominal delays as with periodic controller but with far less communication cost (See Section 5.2.1). The detailed performance analysis will be given in next chapter.

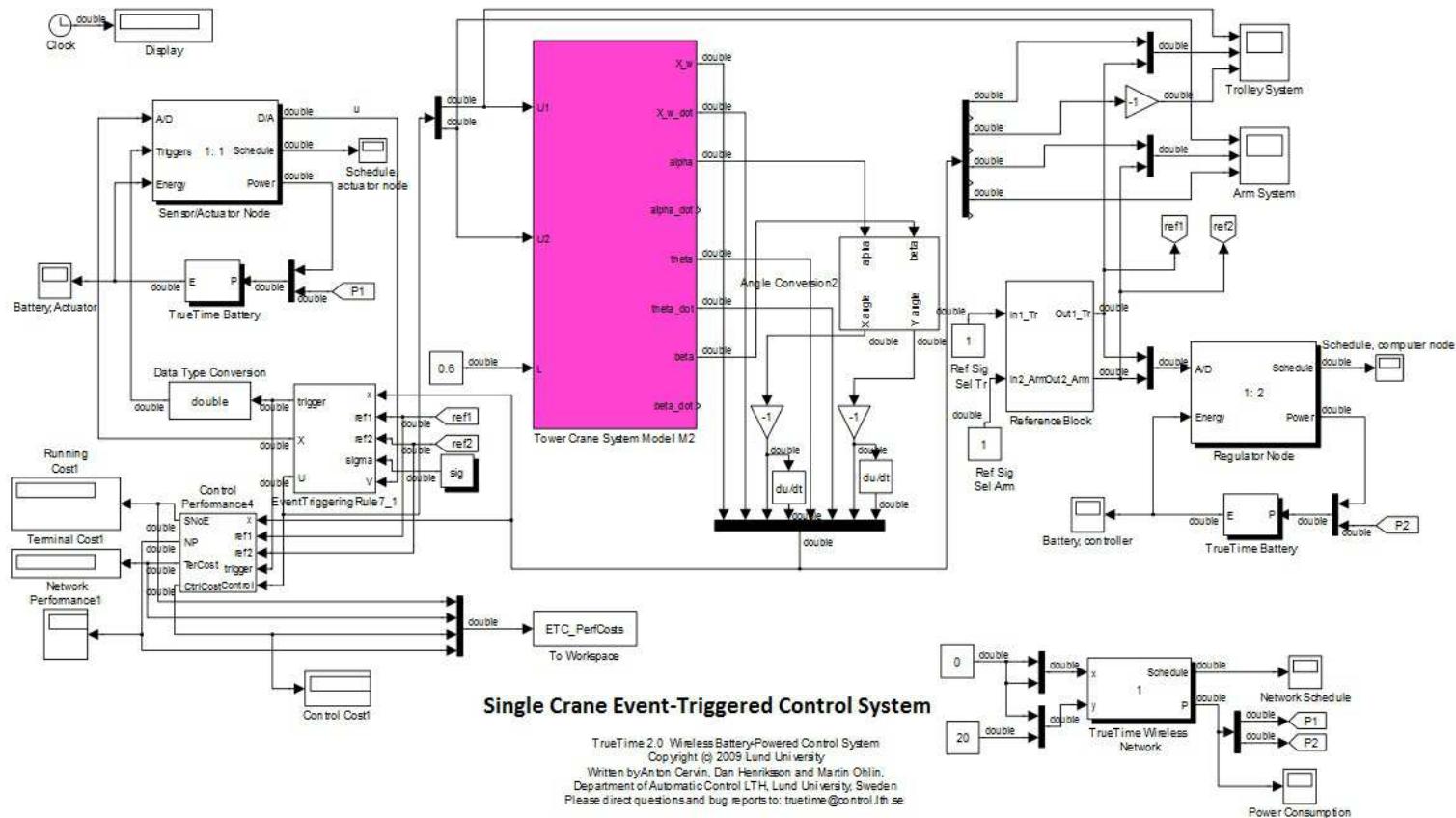
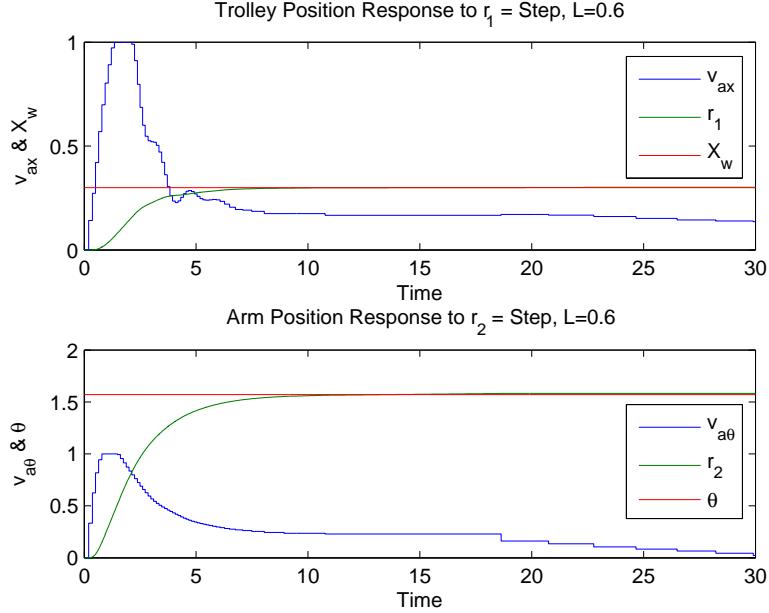
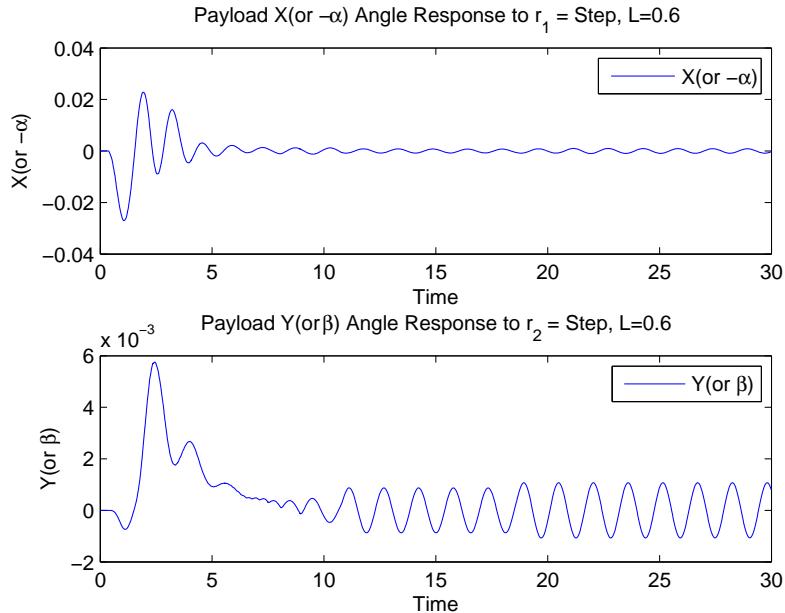


Figure 4.6: TrueTime Implementation of Event-Triggered Control System for Single Crane. The TrueTime network block has been used to simulate the wireless channel that is being accessed using IEEE802.15.4 protocol.



(a) Response of X_w and θ for $L = 0.6$. Red-Response, Blue-Control Action and Green-Reference Input. The step reference tracking is achieved with almost zero steady-state error. Note that control action also stays within limits and goes into saturation for very small time and recovers from the saturation quickly. The response is quite smooth. Note that due to deadzone (stiction friction) in gearbox, control signal settles to value > 0 that is equal to PWM input corresponding to stiction friction.



(b) Response of $-\alpha$ and β for $L = 0.6$. Oscillations have been killed quite efficiently and converges to \mathcal{B}_{r_1} with $r = 0.005$ in less than 5 seconds.

Figure 4.7: Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.6\text{m}$. Both Inputs excited Simultaneously. The fixed-gain event-triggered integral controller has been used which achieves all the control requirements mentioned in section 3.2.1. See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1.

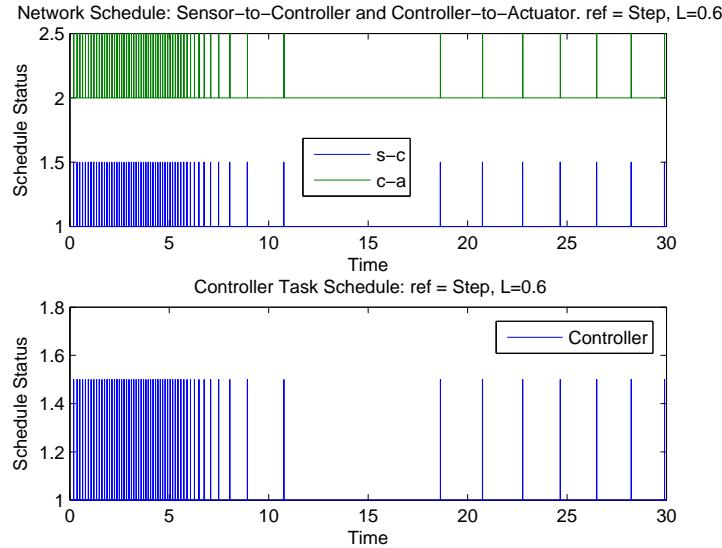


Figure 4.8: Schedules for Network and Controller Tasks. Note that schedules are heavy initially but then they become relaxed. Also note that how in just 60 controller updates the control system has achieved almost same performance which is achieved through 250 controller updates in case of periodic sampling. So the Event-triggered control completely outperforms the periodic time-triggered controller owing to less network usage.

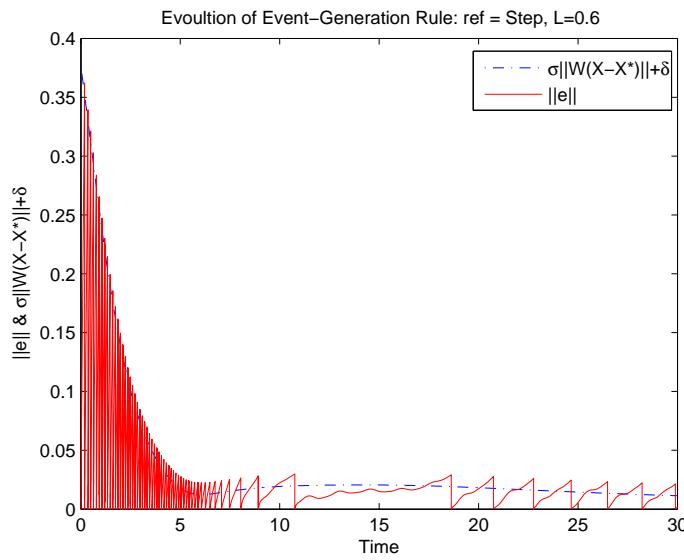


Figure 4.9: The evolution of event generation rule. The plot shows how the error norm e evolves and when it just crosses $\sigma\|W(\mathbf{X} - \mathbf{X}^*)\| + \delta$ the event is generated.

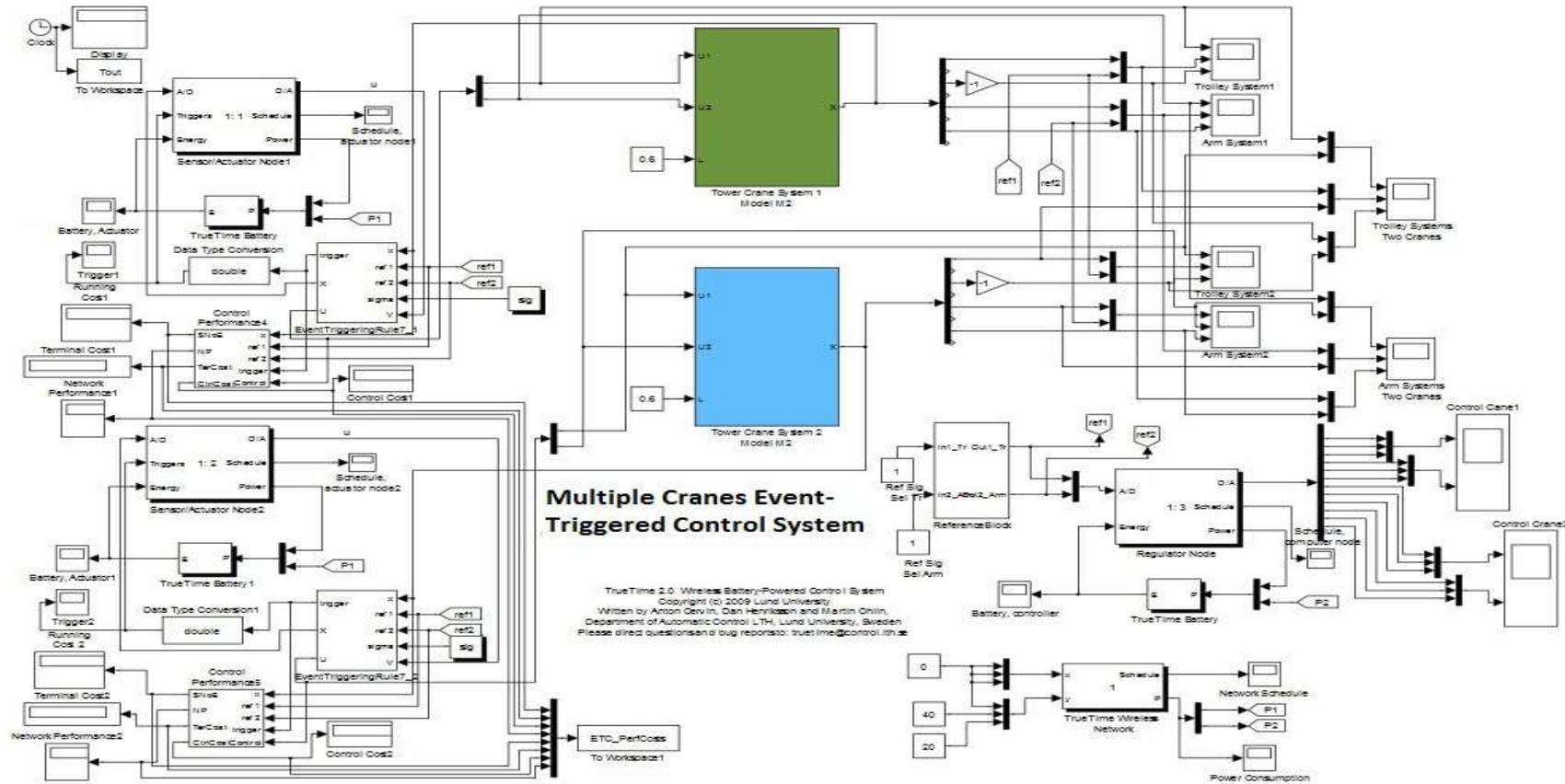
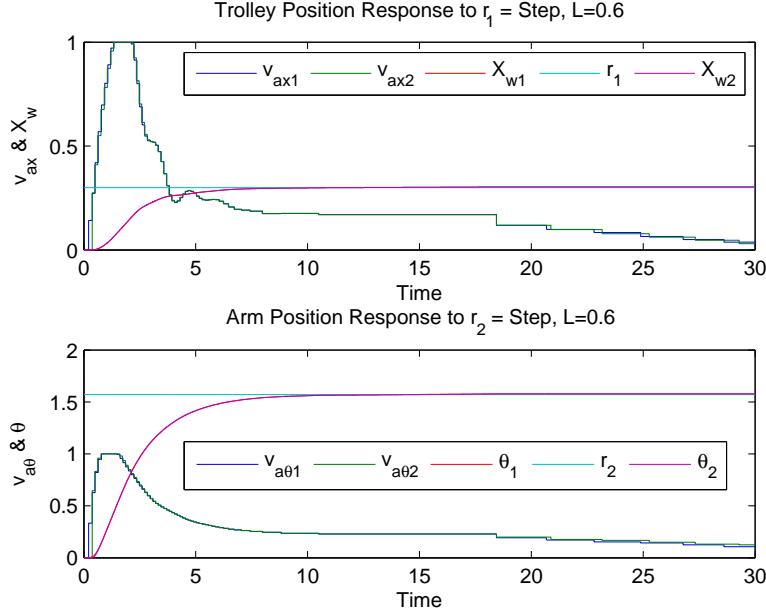
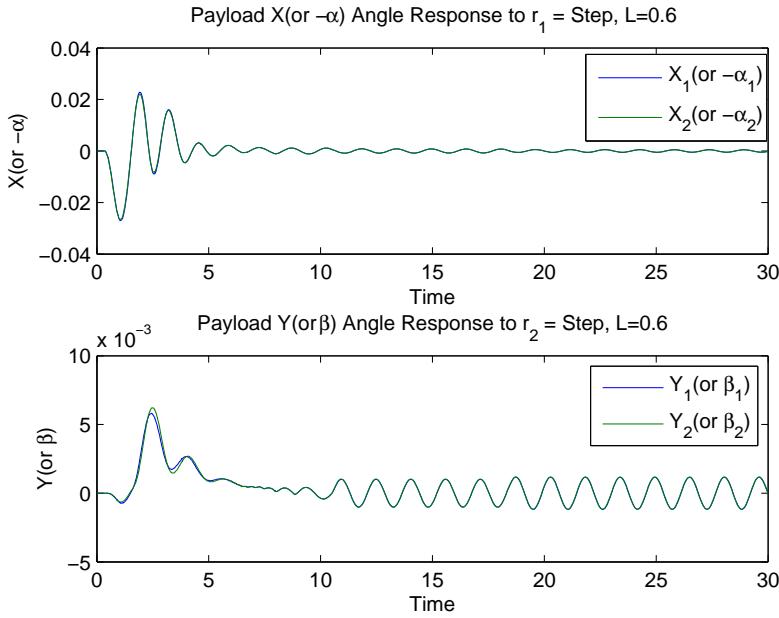


Figure 4.10: TrueTime Implementation of Event-Triggered Control System for multiple cranes. Three TrueTime kernels have been used for simulating sensor nodes and controller nodes. One TrueTime network block has been used to simulate the IEEE802.15.4 shared wireless network. The network settings used are shown in the table 4.1.



(a) Response of X_w and θ for $L = 0.6$. Red and Magenta-Response, Blue and Green-Control Action and Cyan-Reference Input. For both cranes the step reference tracking is achieved with almost zero steady-state error. Note that control action also stays within limits and goes into saturation for very small time and recovers from the saturation quickly. The response is quite smooth. Note that due to deadzone (stiction friction) in gearbox, control signal settles to value > 0 .



(b) Response of $-\alpha$ and β for $L = 0.6$. Oscillations have been killed quite efficiently and converges to \mathcal{B}_{r_1} with $r = 0.005$ in less than 5 seconds.

Figure 4.11: Step Response of crane's model \mathbf{M}_2 for sling length $L = 0.6\text{m}$. Both Inputs excited Simultaneously. The one event-triggered controller is controlling two crane processes. Controller can communicate only with one crane at a time. We can see that all the control requirements mentioned in section 3.2.1 have been achieved. See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1.

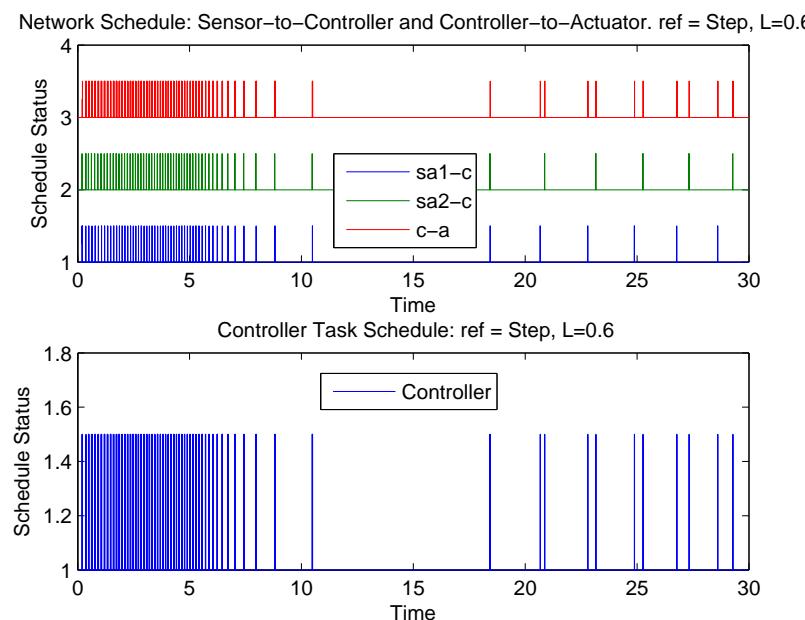


Figure 4.12: Schedules for Network and Controller Tasks. Note that schedules are heavy initially but then become relaxed. Also note that in just 59 controller updates the control system has achieved almost same performance which was achieved through 250 controller updates in case of periodic sampling. So the Event-triggered control completely outperforms the periodic time-triggered controller owing to less network usage.

Chapter 5

Performance Evaluation and Comparative Analysis

The relative performance of closed-loop network control system (NCS) under event-triggered control (ETC) will be evaluated under packets drop-outs and network-induced delays. The overall performance criterion has been setup which captures running cost, control cost and communication cost and it will be compared with the reference cost derived from the system operating under periodic time-triggered controller (TTC).

5.1 Derivation of Performance Criterion for Comparison

First of all for the performance evaluation and analytical comparison we need to setup up some cost function which captures the overall performance of the closed-loop control system under different operating conditions. We will use the signal norms in \mathcal{L}_2 space to setup the following cost functions:

5.1.1 Running Cost Function:

It measures the performance of control system throughout the operation. A system may have good terminal characteristic but may have bad transient response so for a high performance control system it is very important that not only the system converge to the desired state but it achieves that nicely and smoothly.

$$J_r = \sqrt{\int_0^t \|\mathbf{X} - \mathbf{X}^*\|^2 dt} \quad (5.1.1)$$

The reference value for the running cost is derived on the basis of control performance obtained with wired continuous-time controller and is given by:

$$J_r^* = 2.21597 \quad (5.1.2)$$

Now we can write the metric for **running performance** as follows:

$$\boxed{\begin{aligned} P_r &= \frac{1}{J_r} \\ P_r^* &= 0.45127 \end{aligned}} \quad (5.1.3)$$

Where P_r^* gives the desired value for running performance. The objective is to make the wireless crane control system, under either time-triggered controller or event-triggered controller, to stay close to the reference value P_r^* as much as possible. According to experimentation we know that the control system performance is satisfactory if $P_r \geq 0.419974$. So the minimum running performance that gives acceptable response is given by:

$$\boxed{P_{r_{min}} = 0.419974} \quad (5.1.4)$$

5.1.2 Control Action Cost Function:

This metric measures how much control energy has been spent by the controller to make the control system achieve the required performance. The controller, which spends a lot of energy to achieve the same performance which is obtained by another with less control energy, will be considered better.

$$J_c = \sqrt{\int_0^t \|\mathbf{u}\|^2 dt} \quad (5.1.5)$$

The reference value for control energy used is obtained on the basis of what has been spent by the continuous-time controller:

$$J_c^* = 2.513 \quad (5.1.6)$$

Now we can write the metric for **performance of controller** in context of energy usage as follows:

$$\boxed{\begin{aligned} P_c &= \frac{1}{J_c} \\ P_c^* &= 0.3979 \end{aligned}} \quad (5.1.7)$$

5.1.3 Network Cost:

Network cost will be captured by counting the number of times the network resource was accessed whether successful or not. So the number of controller execution will be used as a metric for measuring the network performance. Too many controller updates means bad network performance vice versa. Here we would measure the network cost against some reference to get the normalized result. As per the event-generation rule given by (4.4.1) the minimum inter-execution time we get is $\tau_{min} = 0.120$ s. So for fair comparison between event-triggered and periodic controller we have selected sampling time $h = 0.120$ s for periodic controller as well. Hence, the reference value of controller updates for finite time horizon, $t_f = 30$ s is given by $N^* = 250$. Now we can write the normalized network cost as follows:

$$J_n = \frac{N}{N^*} \quad (5.1.8)$$

Where N = Number of controller executions. The reference value of network cost is based on the time-triggered discrete controller and is given:

$$J_n^* = 1 \quad (5.1.9)$$

Now we can write the metric for **performance of network** in context of communication as follows:

$$\boxed{\begin{aligned} P_n &= \frac{1}{J_n} \\ P_n^* &= 1 \end{aligned}} \quad (5.1.10)$$

5.1.4 Complete Network Control Cost:

Now we can form the cost function for capturing the overall performance of the network control system as follows:

$$\boxed{J_{ncs} = \sqrt{\int_0^t \|\mathbf{X} - \mathbf{X}^*\|^2 dt} + \sqrt{\int_0^t \|\mathbf{u}\|^2 dt} + \frac{N}{250}} \quad (5.1.11)$$

The high value of J_{ncs} means bad network control system's performance. The reference value for J_{ncs} is given by:

$$\boxed{\begin{aligned} J_{ncs}^* &= J_r^* + J_r^* + J_n^* \\ J_{ncs}^* &= 5.72897 \end{aligned}} \quad (5.1.12)$$

Now we can write the metric for **total performance of network control system** as follows:

$$\boxed{\begin{aligned} P_{ncs} &= \frac{1}{J_{ncs}} \\ P_{ncs}^* &= 0.17455 \end{aligned}} \quad (5.1.13)$$

5.1.5 Main Objective of Event-Triggered Controller:

The main objective of event-triggered controller is to reduce the network cost such that $P_n > 1$ and $P_r \geq 0.4545$. While comparing event-triggered and time-triggered controller, we will employ the event-triggered control only if it ensures that $P_n > 1$ otherwise we will discard it and prefer the periodic time-triggered controller. So another objective of our following comparative analysis is to identify the operating conditions under which event-triggered controller is preferable over periodic time-triggered controller for wireless network control applications.

5.2 Single Crane Case: Performance Comparison and Analysis

In this section we will numerically evaluate the performance of single crane being controlled over wireless network under different operating conditions. First,

we will compare the performance of event-triggered (ETC) and periodic time-triggered controller (TTC) with zero delays in the loop and then we would give the performance comparison under various kind of delays. The IEEE 802.15.4 wireless network, with unslotted CSMA/CA MAC protocol in non-beaconed mode with back-offs, has been used in all simulations. The following network settings has been used for all simulations in this section:

Table 5.1: IEEE 802.15.4 Network parameter values and delays used in the simulation.

Parameters	Values
Data Rate [kbps]	250
Retry Limits	2
Path Loss Function	$1/d^{3.5}$
Minimum Frame Size [bits]	272
macMinBE	3
aMaxBE	5
macMaxCSMABackoffs	4
Sensor-Controller Delay, Δ_{sc} [s]	[0, 0.2]
Controller-Actuator Delay, Δ_{ca} [s]	[0, 0.2]
Event-Detector-Sensor Delay, Δ_{eds} [s]	[0, 0.2]
Actuator-Plant Delay, Δ_{ap} [s]	0

5.2.1 Performance Analysis under Zero Delays

In this section we will show the response and schedule of single crane under both event-triggered and periodic control policy with zero delays in the loop. As per the event-generation rule given by (4.4.1) the minimum inter-execution time we get is $\tau_{min} = 0.120s$. So for fair comparison between event-triggered and periodic controller we have selected sampling time $h = 0.120s$ for periodic controller as well. The time horizon for simulation is $t_f = 30s$. The Figure 5.2(a) shows the response of trolley subsystem and the figure 5.2(b) shows the response of arm subsystem. The Figure 5.2(c) shows network schedules for both periodic and event-triggered controllers. As we can see that response is almost the same under both control policies. Both controllers have kept the payload inside the ball \mathcal{B}_{r_1} throughout the whole move and also driven it to the ball \mathcal{B}_{r_2} at steady-state(See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1). The response is almost same but the network utilization under event-triggered control policy is almost 4 times less than that of periodic controller.

5.2.2 Performance Analysis under Network Induced Delays

In this section, the performance of event-triggered control and periodic time-triggered controller under network induced delays have been analyzed and compared using the performance metrics suggested in section 5.1. We are mainly interested to numerically analyze the performance of two controllers under various kind of delays as shown in the Figure 5.1 for single crane. The multiple

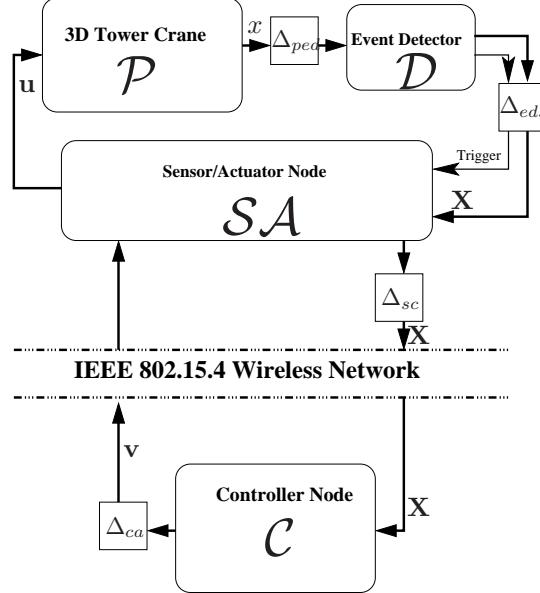


Figure 5.1: Block Diagram illustrating different kind of delays in the system.

cranes case will be dealt separately in next section. We are mainly interested to evaluate the network control performance of ETC and TTC for delays like sensor-to-controller delay Δ_{sc} , controller-to-actuator (Δ_{ca}), event-detector-to-sensor delay (Δ_{eds}) and plant-to-event-detector delay (Δ_{ped}). Keeping all other delays zero, one delay has been varied at a time and the impact on performance of both, event-triggered controller and periodic-time-triggered controller has been evaluated numerically by simulating the single wireless closed-loop crane control system shown in figure 4.3 and 4.6. The IEEE 802.15.4 wireless network, with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs, has been used in all simulations.

Controller-to-Actuator Delay

The simulation has been done for $\Delta_{ca} \in [0, 0.2]$ while changing it in steps of 0.005s for each simulation run. The performance plots are shown in the figure 5.3. The figure 5.3(a) shows how increase in Δ_{ca} affects running performance, control action and network performance. The figure 5.3(b) shows the total performance for the network control system as a function of Δ_{ca} . We can see that up to $\Delta_{ca} \approx 0.170$, ETC dominates TTC but then ETC degrades and lost its benefit of less network usage. Also, the control performance under ETC degrades bit faster with increase in delay as compared to that under periodic controller. The control performance under ETC is lost at $\Delta_{ca} \approx 0.085$ s whereas for periodic it is lost at $\Delta_{ca} \approx 0.110$.

Sensor-to-Controller Delay

The simulation has been done for $\Delta_{sc} \in [0, 0.2]$ while changing it in steps of 0.005s for each simulation run. The performance plots are shown in the figure

5.4. The figure 5.4(a) shows how increase in Δ_{sc} affects the running performance, control action and network performance. The figure 5.4(b) shows the total performance for the network control system as a function of Δ_{sc} . We can see that the running performance, P_r , is almost the same as for Δ_{ca} for both event-triggered and periodic controllers but now the network performance, P_n , under event-triggered control policy degrades below that of periodic controller at higher level of delay.

Transport Delay: Event Detector Circuit

The simulation has been done for $\Delta_{eds} \in [0, 0.2]$ while changing it in steps of 0.005s for each simulation run. The performance plots are shown in the figure 5.5. The figure 5.5(a) shows how increase in Δ_{eds} affects the running performance, control action and network performance. The figure 5.5(b) shows the total performance for the network control system as a function of Δ_{eds} . The running performance under Δ_{eds} is almost similar to that under Δ_{ca} and Δ_{sc} . But the network performance is less sensitive under the delay Δ_{eds} compared to that under Δ_{ca} and Δ_{sc} .

Transport Delay: Plant-to-Event Detector Circuit

The simulation has been done for $\Delta_{ped} \in [0, 0.2]$ while changing it in steps of 0.005s for each simulation run. The performance plots are shown in the figure 5.6. The figure 5.6(a) shows how increase in Δ_{ped} affects the running performance, control action and network performance. The figure 5.6(b) shows the total performance for the network control system as a function of Δ_{ped} . We can see that the performance of network control system is less sensitive under the delay Δ_{ped} compared to other types of delays. Although both control and network performance under ETC degrades with Δ_{ped} but now the rate of degradation is lower.

5.2.3 Section Summary

In the absence of delays, the event-triggered controller (ETC) achieves almost the same control performance but 4 times less network utilization as compared to that of periodic controller (TTC). But under delays the running performance of both event-triggered and periodic controllers degrades but the performance for event-triggered controller degrades faster. Also, with increase in delay the network utilization increases under ETC and finally goes higher than that of periodic for the delay above certain level which depends on the point at which the delay enters in the loop. The delay inside sensor, Δ_{sc} , and controller, Δ_{ca} , nodes are more damaging as compared to delays at plant output and inside event-detection circuit. The possible reason is that delay inside the mote may disturb the communication directly by blocking the incoming packets whereas the delay outside mote doesn't block the communication. The network control system is affected most by Δ_{ca} and Δ_{sc} and least by Δ_{ped} whereas Δ_{eds} has intermediate effect. Also, the simulation reveals that ETC is advantageous under certain conditions which must be ensured otherwise periodic time-triggered controller is better. If the delay in the loop is not big then event-triggered control should be the preferred choice.

5.3 Multiple Crane Case: Performance Analysis

In this section we will evaluate the performance of event-triggered controller for multiple cranes under two scenarios: 1) Cranes are not allowed to retransmit in case of collision 2) Cranes are allowed to retransmit in case of collision. We are particularly interested to see how event-triggered controller behaves with and without MAC protocol. The probability of collision and subsequent effect of packet losses will also be evaluated for event-triggered control policy. The scalability properties of event-triggered controller and the effect of MAC protocol in this regard will be compared as well.

5.3.1 Performance Evaluation without MAC Protocol

In this section we will evaluate the performance of event-triggered controller without using back-offs. First we will show how the event-triggered controller works under random network schedule even when no re-transmission is allowed in case of collision.

Event-triggered Control and Random Network Access

In this section the network control system with three crane nodes and one controller node have been simulated. The IEEE 802.15.4 wireless network with unslotted CSMA/CA MAC protocol in nonbeaconed mode without back-offs have been used. The nominal network settings given in table 4.1 have been used except macMaxCSMABackoffs which is set equal to zero here to disallow the retransmission in CSMA/CA MAC protocol. The figure 5.7 shows the response of NCS for event-triggered controller. We can see that all cranes have satisfactory performance under ETC even when no packet retransmission is enabled. Though protocol is not using back-offs and hence not using its full potential to avoid collision but still only 5 – 7 collisions occurred for each node. The main reason for small number of collision is that ETC itself introduces some randomness in packet transmission which reduces probability of collision. The lost packets have been marked by red crosses in sub-figures 5.7(c) and 5.7(d). The performance is not much affected because we are not losing high number of consecutive packets. We can see that the crane-2 undergoes larger payload oscillations comparatively due to the reason that it suffers from packet loss at startup and hence due to large error and integral action the control signal suddenly rise and gets saturated which induces bit larger oscillations but it finally converges to the ball of radius 5mm.

Packet Dropout Probability and Performance

The figure 5.8 shows the packet loss probability as a function of number of cranes in the network without back-off capability. In the figure 5.8(a) we can see that average packet loss probability drastically increase with increase in number of cranes in the network when cranes are not allowed to retransmit the packet in case of collision. The figure 5.8(b) shows the packet loss probability of each crane in the network. The control performance goes below the minimum acceptable for the network of 7 cranes. We can see that the packet loss probability of crane-1 > crane-6 > crane-5 for network of 7 cranes. The step response of seven

cranes being controlled over wireless network is shown in the figure 5.9. We can see that despite of crane-5 having lowest packet loss probability among three cranes (crane-1,6 and 5), it has worst control performance. It is because the crane-5 is losing highest number of consecutive packets during transient phase of response. The simulation results show that packet loss probability is not the only indicator of control performance but it also depends on the number of consecutive packet losses during transient phase.

5.3.2 Scalability Properties of ETC and MAC Protocol

In this section, we will see how event-triggered control policy scales with increase in number of nodes. We have simulated wireless NCS with different number of nodes starting with one crane node and going up to 13 crane nodes. The simulation has been done for event.triggered control policy with and without back-offs. To compare the performance for different number of cranes we have normalized the performance metrics by dividing the sum of performance for each crane with the number of cranes in the network. For example the total NCS performance can be written as:

$$P_{n_{cs_k}} = \frac{1}{k} \sum_{i=1}^k P_{n_{cs_i}} \quad (5.3.1)$$

where k = number of cranes in NCS. Similarly all normalized performance metrics can be computed for comparison in the same graph.

The figure 5.10 shows how the performance of network varies with increase in number of nodes. The plot compares the performance of network control system with IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode under event-triggered control policy with (network parameter $\text{macMaxCSMABackoffs} \neq 0$) and without back-offs (network parameter $\text{macMaxCSMABackoffs} = 0$) as a function of number of cranes. The graph shows that Event-triggered controller has better scalability properties when back-off mechanism is employed. The figure 5.10 shows that NCS under event-triggered controller without back-off breaks down for 7 cranes but when back-off is enabled the NCS performance is restored and is acceptable up to, at least, 13 cranes. From simulations we know that the control performance of crane is acceptable if the packet loss probability remains less than 32.31%. So the maximum threshold for packet loss probability is 32.31%. The figure 5.11 shows the packet loss probability as the function of number of cranes. The figure 5.11(a) shows the average loss probability whereas the figure 5.11(b) shows the packet loss probability for each crane in the network. We can see that MAC protocol has significantly reduced the packet loss probability of each crane way below the maximum allowed threshold and hence each crane is now exhibiting good control performance. But we can see that even after allowing back-offs the performance of NCS degrades but now the degradation process is very slow. The major reason for degradation is the delay induced by the MAC protocol and with increase in number of nodes this delay plays a major role for performance degradation. This delay directly depends on the values of MAC parameters. Smaller values induce smaller delay but it also increases the packet loss probability and hence degrades the control performance as well. It shows that MAC protocol plays significant role in network control systems.

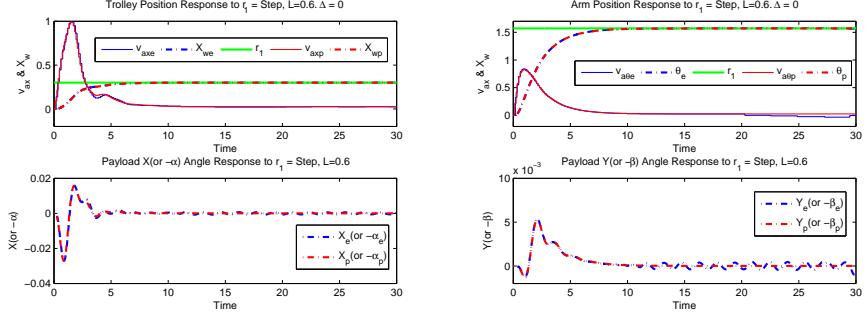
The parameters $macMinBE$, $aMaxBE$ and $macMaxCSMABackoffs$ plays significant role and they can be tuned to get the optimum performance for both network and control. This can be a separate research study to investigate a relationship between NCS performance and these parameters where the combined optimization problem can be solved which may lead to a new novel methodology of controller design and synthesis which takes into account these network protocol parameters as well.

5.3.3 Section Summary

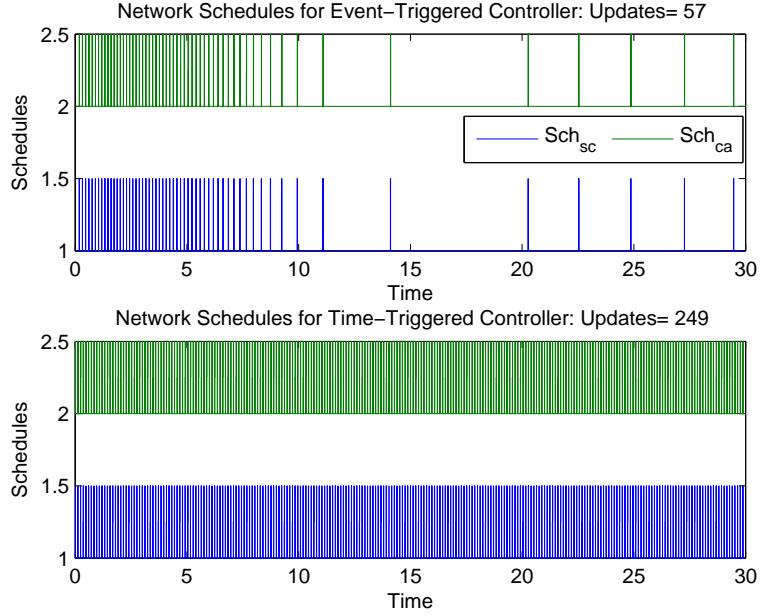
In this section we have done the network control performance evaluation for multiple cranes under packet losses and the effect of MAC protocol. The packet loss probability is not the only factor that affects the control performance but it also depends highly on the number of consecutive packet losses during transient phase of response. The MAC protocol plays a key role for achieving acceptable control performance under event-triggered control policy for large number of cranes in the network. The values of MAC parameters pose an optimization trade-off between control and network performance. The values only in certain band ensure that both the network and control performance is acceptable.

5.4 Chapter Summary

In this chapter we have done the performance evaluation of event-triggered control and also compared it with periodic controller under various operating conditions. The signal energy norms have been used to measure the network control system performance. For single crane case under zero delays, the event-triggered controller utilizes the network almost 4 times less but achieves almost the same control performance as compared to periodic controller. In the presence of network induced delays, the control performance monotonically degrades under both event-triggered and periodic controller but that for event-triggered degrades bit faster. The network utilization also monotonically increases under event-triggered controller and after certain delay, it utilizes the network even more than the periodic controller. So ETC is only advantageous under certain conditions which must be ensured otherwise periodic time-triggered controller is better. If the delay in the loop is not big then event-triggered control should be the preferred choice. For control of multiple crane under event-triggered control policy, the network is randomly accessed using CSMA/CA MAC protocol without too many collisions. According to our simulations, the packet loss probability is not the only factor that affects the control performance but it also depends highly on the number of consecutive packet losses during transient phase of response. Also, the MAC protocol plays a significant role for achieving better scalability properties under event-triggered control policy. The value of MAC parameters pose a trade-off between network and control performance, the large parameter values induce higher network delays which results in bad control performance and very small values results in too many collisions which results in bad control performance as well. So these parameters should be carefully selected for achieving better performance.

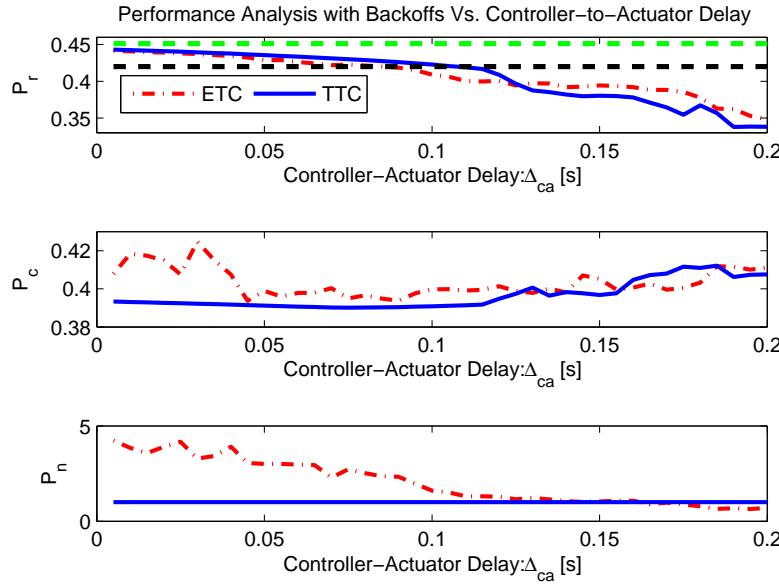


(a) Response of Trolley subsystem for both event-triggered and periodic controller. Blue: event-triggered and periodic controller. Blue: Response under Event-triggered controller, Response under Event-triggered controller, Red: Response under periodic controller and Red: Response under periodic controller and Green: Reference input. Note that the response of trolley position X_w and payload most identical but that of Y angle under ETC is almost identical for both event- triggered and periodic controller and both the control specifications given in section 3.2.1 have achieved the control requirements mentioned in section 3.2.1.

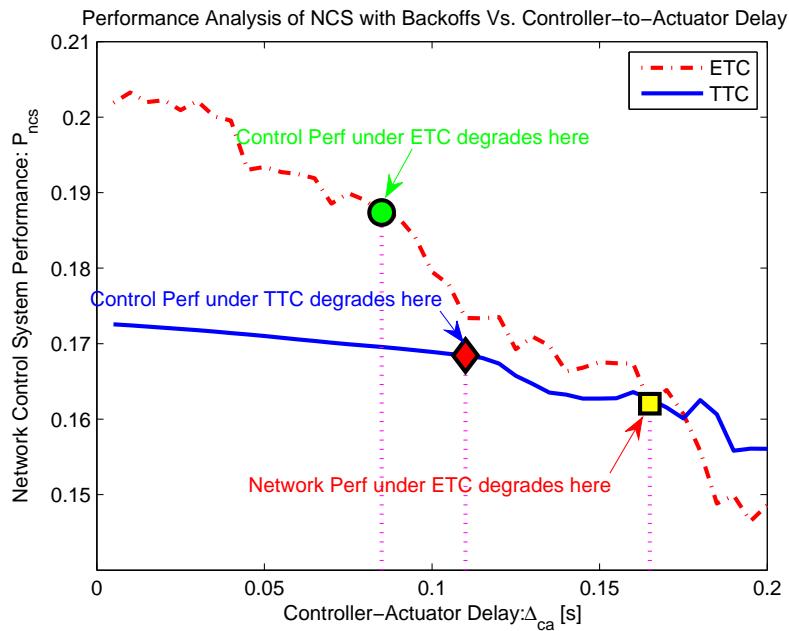


(c) Schedules for Network Tasks. Note that schedules for ETC are heavy initially but then they become relaxed. Also note that how in just 57 controller updates the control system has achieved almost same performance which is achieved through 250 controller updates in case of periodic sampling. So the Event-triggered control completely outperforms the periodic time-triggered controller owing to less network usage.

Figure 5.2: Step Response of crane under event-triggered and periodic control policy. Both meet all the control requirements mentioned in section 3.2.1 with almost similar performance but ETC achieves this in almost 4 times less controller updates. Both controllers have kept the payload inside the ball \mathcal{B}_{r_1} throughout the whole move and also driven it to the ball \mathcal{B}_{r_2} at steady-state. See the definitions of the balls \mathcal{B}_{r_1} and \mathcal{B}_{r_2} in the section 3.2.1.

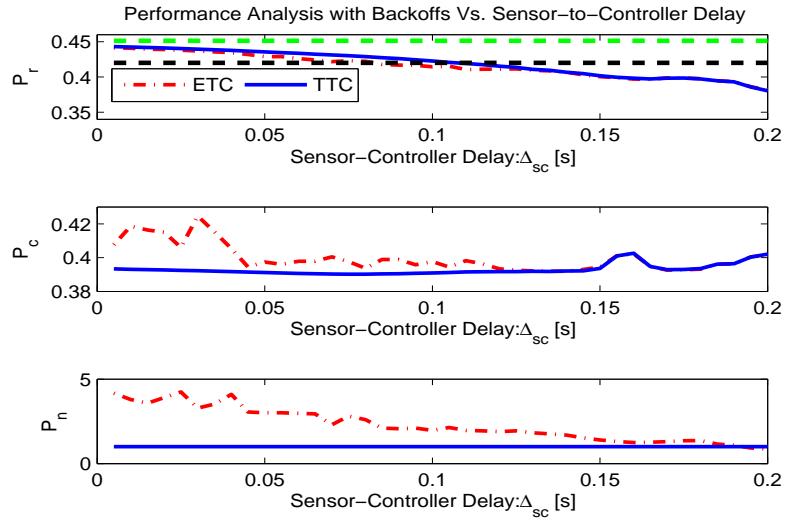


(a) CL-System's Running Performance, P_r , Controller Performance, P_c and Network Performance, P_n . Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The dashed black line is the minimum level of desired running performance. The dashed green line is the desired running performance on the basis of CT control performance.

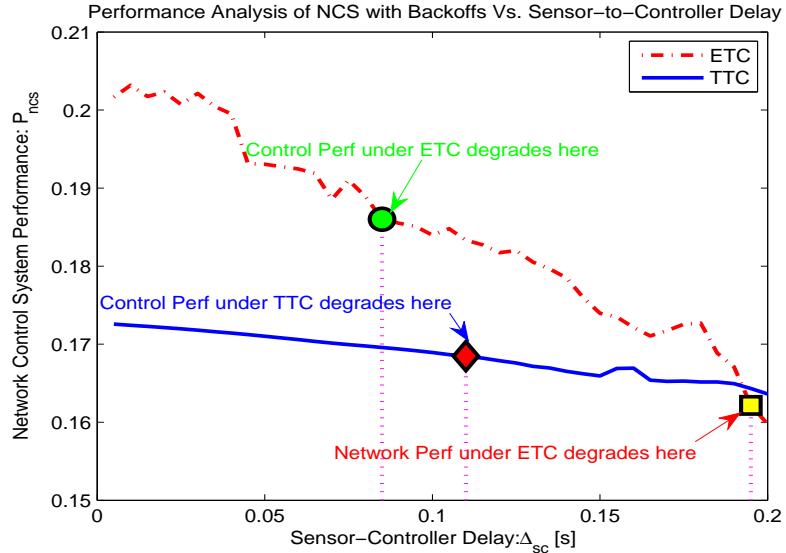


(b) Total Performance comparison of Network Control system with non-zero back-offs. Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The graph shows how the performance of NCS under two different control policies varies with controller-to-actuator delay, Δ_{ca} .

Figure 5.3: Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of controller-to-actuator delay Δ_{ca} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. We can see that up to $\Delta_{ca} \approx 0.170$, ETC dominates TTC but then ETC degrades and lost its benefit of less network usage.

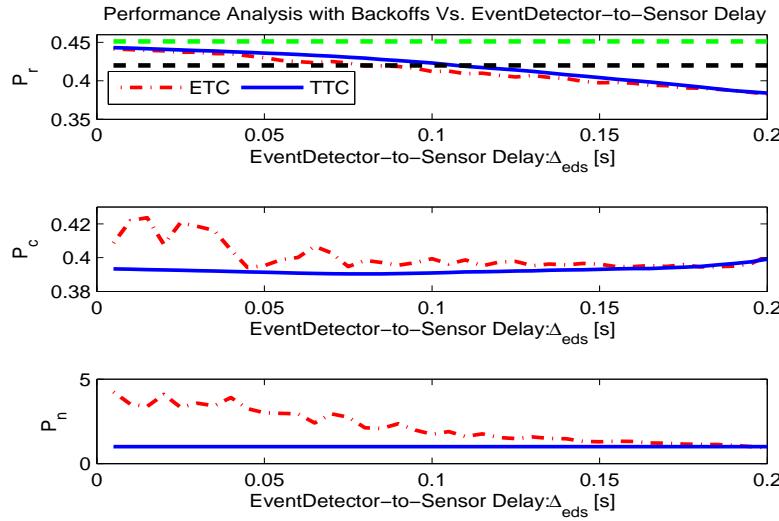


(a) CL-System's Running Performance, P_r , Controller Performance, P_c and Network Performance, P_n . Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The dashed black line is the minimum level of desired running performance. The dashed green line is the desired running performance on the basis of CT control performance.

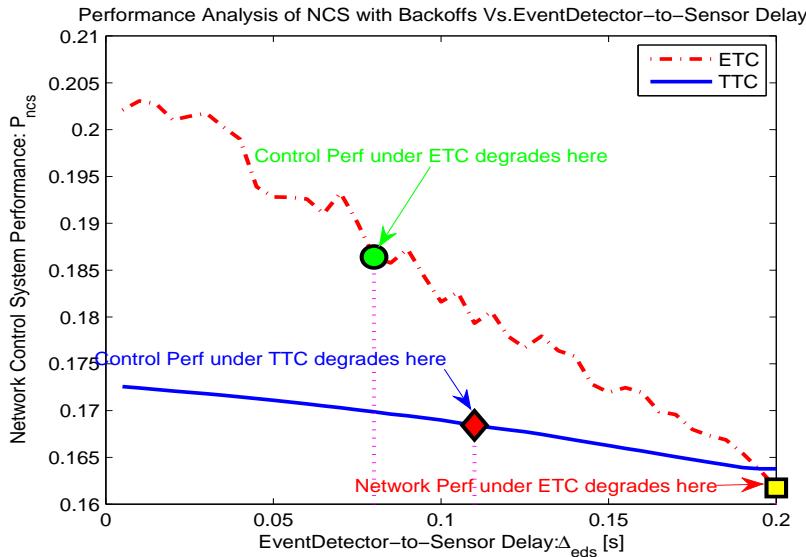


(b) Total Performance comparison of Network Control System with non-zero back-offs. Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The graph shows how the performance of NCS under two different control policies varies with sensor-to-controller delay, Δ_{sc} . As compared to Δ_{ca} the only difference is in network performance which is degrading at lower rate now.

Figure 5.4: Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of sensor-to-controller delay Δ_{sc} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. P_r is almost the same as for Δ_{ca} but now P_n degrades below that of periodic at higher level of delay.

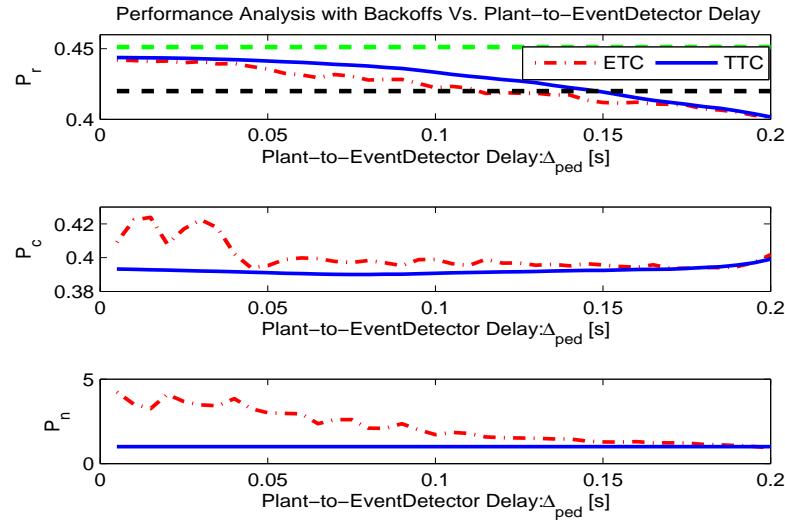


(a) CL-System's Running Performance, P_r , Controller Performance, P_c and Network Performance, P_n . Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The dashed black line is the minimum level of desired running performance. The dashed green line is the desired running performance on the basis of CT control performance.

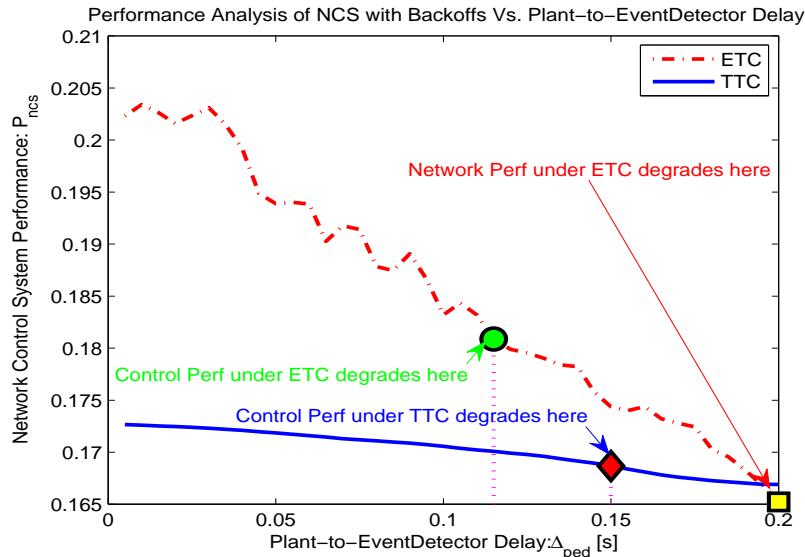


(b) Total Performance comparison of Network Control system with non-zero back-offs. Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The graph shows how the performance of NCS under two different control policies varies with event-detector-to-sensor delay, Δ_{eds} . Green ball: control performance under ETC is lost. Red Box: Control Performance under TTC is lost. Yellow Square: Network performance under ETC degrades below TTC.

Figure 5.5: Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of delay inside event-detector circuit Δ_{eds} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. The running performance under Δ_{eds} is almost the same as for Δ_{ca} and Δ_{sc} but now the network performance under ETC degrades below that of periodic at much higher level of Δ_{eds} .

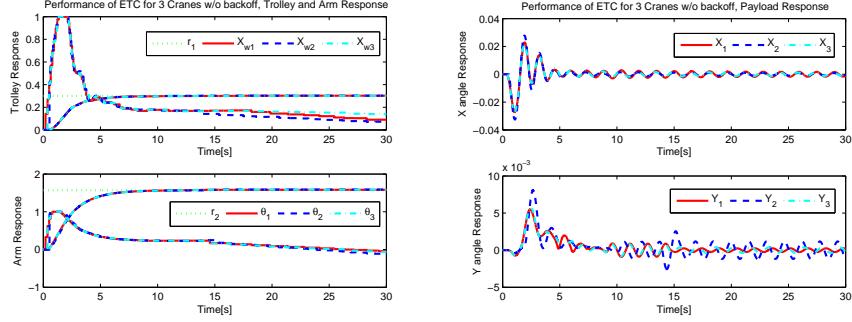


(a) CL-System's Running Performance, P_r , Controller Performance, P_c and Network Performance, P_n . Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The dashed black line is the minimum level of desired running performance. The dashed green line is the desired running performance on the basis of CT control performance.

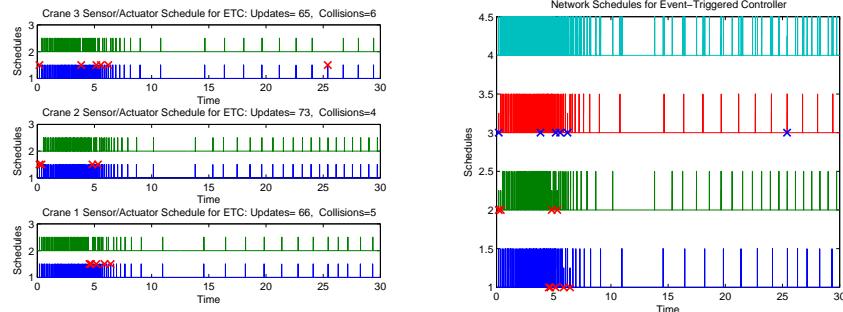


(b) Total Performance comparison of Network Control system with non-zero back-offs. Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) and the solid blue curves show that of periodic Time-Triggered Controller (TTC). The graph shows how the performance of NCS under two different control policies varies with plant-to-event-detector delay, Δ_{ped} . Green ball: control performance under ETC is lost. Red Box: Control Performance under TTC is lost. Yellow Square: Network performance under ETC degrades below TTC.

Figure 5.6: Single Crane Case: Performance of Event-Triggered Controller (ETC) and the periodic Time-Triggered Controller (TTC) in the presence of delay Δ_{ped} under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs enabled. Both Network and control performance under ETC and TTC degrades with delay but now the rate of degradation is lower. The network control system can tolerate higher level of Δ_{ped} delay as compared to Δ_{sc} and Δ_{ca} .



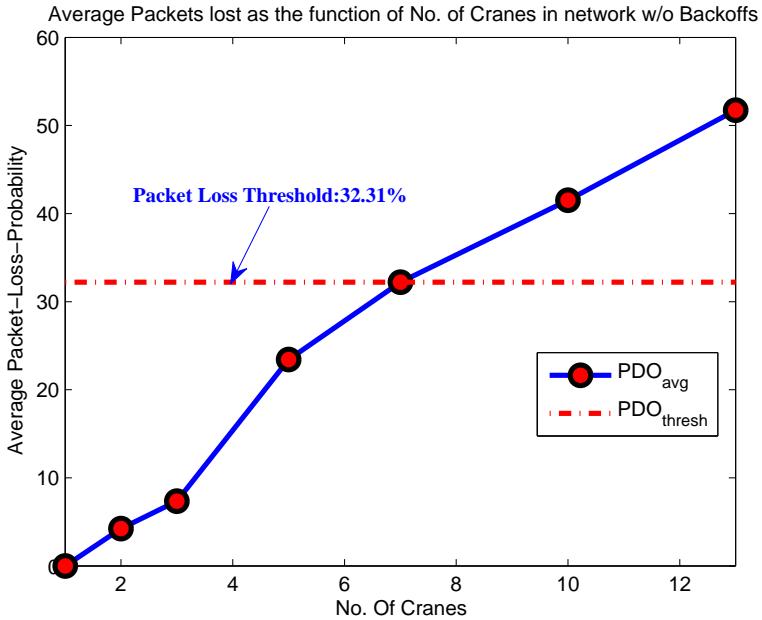
(a) The above plot shows the step response of (b) Payload Response for 3 cranes. Per-Trolley and Arm. Solid red, dotted-blue and dash-dot cyan shows the trolley response for crane 1, 2 and 3 respectively. The plots show that trolley response is still satisfactory for all cranes.



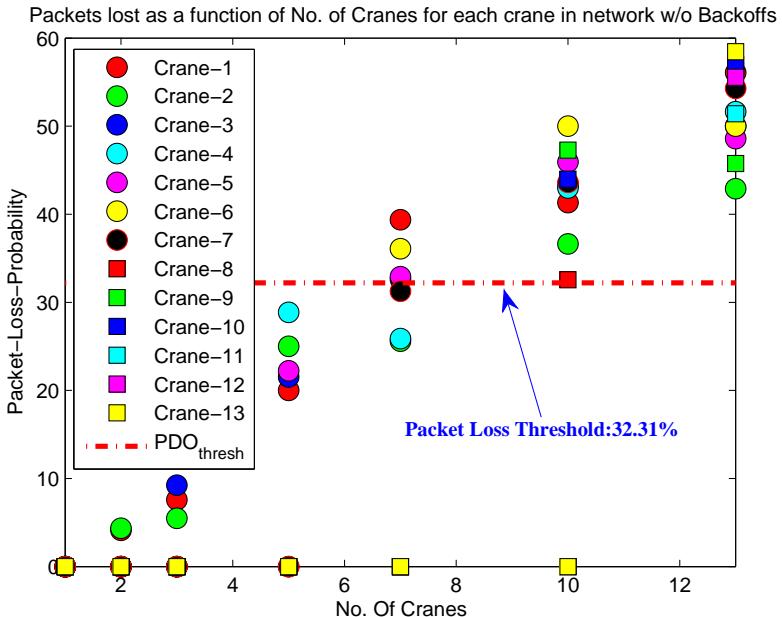
(c) Schedules for sensor and actuator for 3 cranes. In each subplot, top is for actuator and bottom is for the sensor schedule. The red cross in each subplot shows the samples which were lost due to network collisions and we can see that they never reached back to actuators and hence control signal not updated.

(d) Schedules for sensor and actuator for 3 cranes. In each subplot, top is for actuator and bottom is for the sensor schedule. The red cross in each subplot shows the samples which were lost due to network collisions and we can see that they never reached back to actuators and hence control signal not updated.

Figure 5.7: Multiple Crane Case with Packet losses: Event-Triggered Controller Performance for 3 Cranes under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode without back-offs. We can see that all cranes have satisfactory performance under ETC even when no packet retransmission is enabled. The main reason for small number of collision is that ETC itself introduces some randomness in packet transmission which reduces probability of collision.



(a) Average packet loss probability as function of number of cranes in the network. Dash-Dot red line shows the maximum allowed packet loss probability above which the control performance is unacceptable. For a network of 7 cranes, the packet loss probability touches the threshold.



(b) Packet loss probability of each crane as a function of number of cranes in the network. Notice that for seven cranes in a network, crane-1,5 and 6 have crossed the threshold.

Figure 5.8: Packet Loss Probability in the absence of MAC protocol: Performance of network under Event-Triggered Control policy as a function of number of cranes with IEEE 802.15.4 wireless network settings using unslotted CSMA/CA MAC protocol in nonbeaconed mode without back-offs (network parameter `macMaxCSMABackoffs = 0`). The graphs show that when the cranes are not allowed to retransmit the packet then the network control system breaks down for 7 number of cranes in the network. The dash-dot red line is the maximum allowed packet loss probability above which control performance is not acceptable.

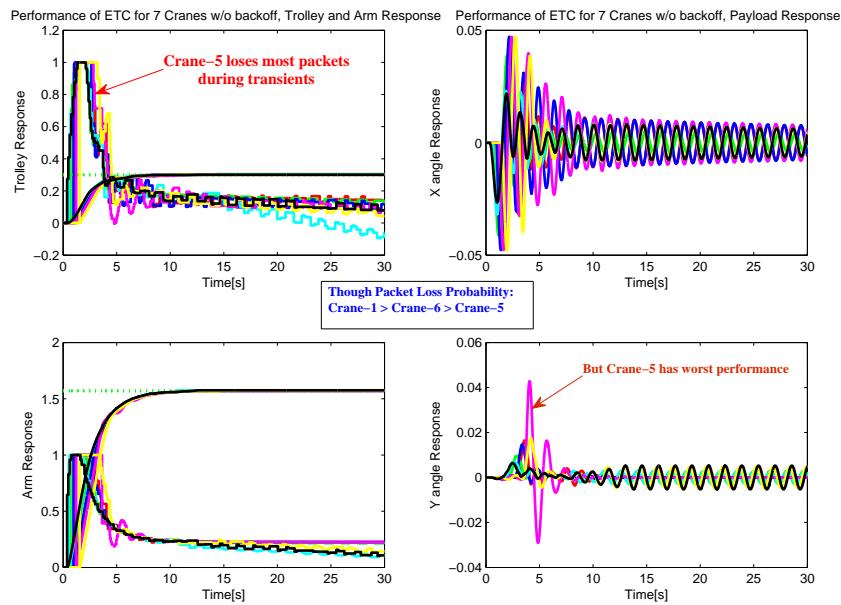
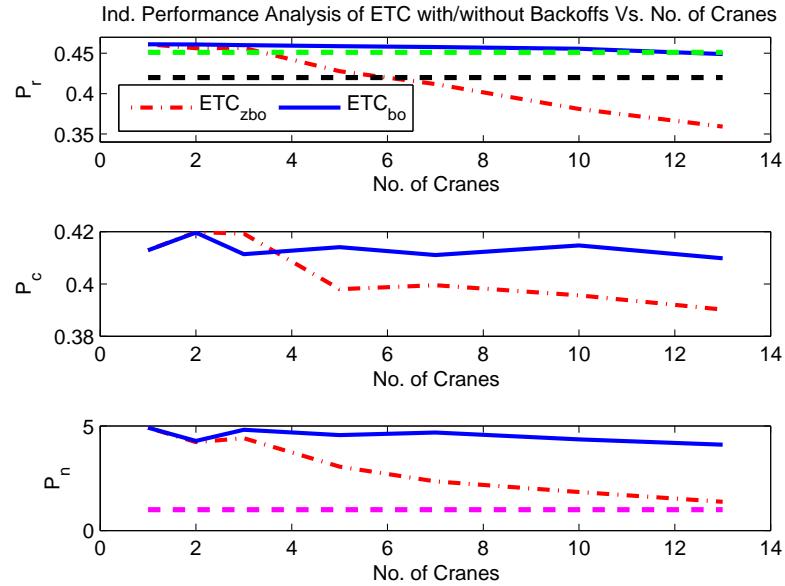
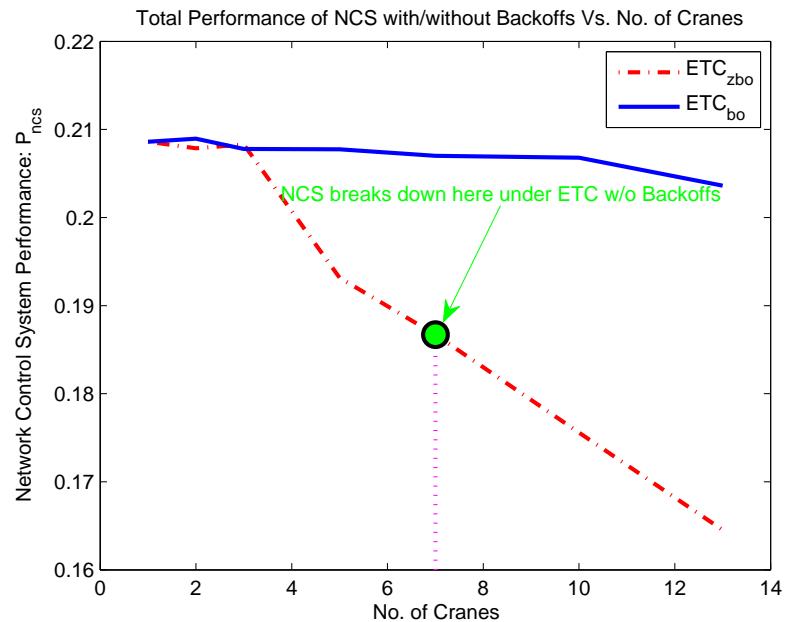


Figure 5.9: Step Response of 7 Cranes under Event-triggered control policy using IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode without back-offs (network parameter $\text{macMaxCSMABackoffs} = 0$). Although packet loss probability of crane-1 > crane-6 > crane-5 but still crane-5 has worst performance among all. The reason is the higher number of packet losses during transient phase of crane-5.

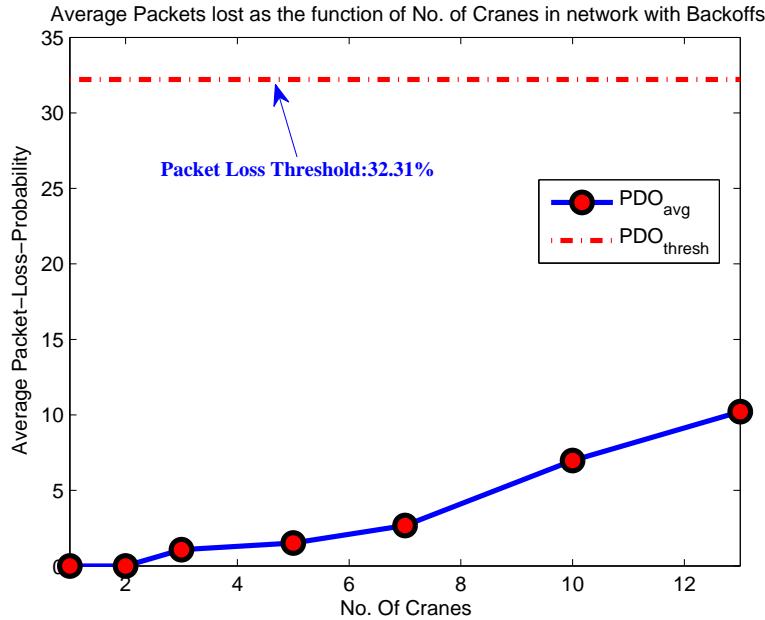


(a) CL-System's Running Performance, P_r , Controller Performance, P_c and Network Performance, P_n . Dash-Dot red graph shows the performance under Event-Triggered Controller (ETC) without back-off and the solid blue curves show that of Event-Triggered Controller (TTC) with back-off. Performance is better when back-off mechanism is enabled.

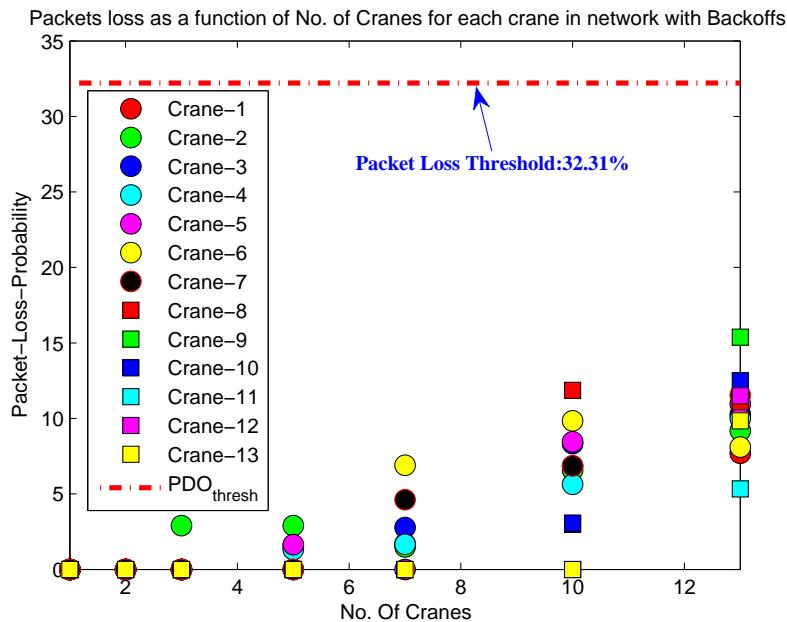


(b) Total Performance comparison of Network Control system with and without back-offs. The graph shows how the performance of NCS under two different protocol settings but same controller varies with increase in number of cranes in network. Network suffers from drastic degradation in performance when back-off is not enabled.

Figure 5.10: Scalability Properties: Performance of Event-Triggered Controller (ETC) w.r.t. number of cranes under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with (network parameter macMaxCSMABackoffs $\neq 0$) and without (network parameter macMaxCSMABackoffs = 0) back-offs. The graph shows that Event-triggered controller has better scalability properties when transmission back-off mechanism is employed. NCS under event-triggered controller with zero back-off breaks down for 13 cranes but when back-off is enabled the NCS performance is restored.



(a) Average packet loss probability as function of number of cranes in the network. MAC has significantly reduced the packet loss probability (compare to 5.8(a)).



(b) Packet loss probability of each crane as a function of number of cranes in the network. Now packet loss probaility of each crane is way below the threshold (dash-dot red line).

Figure 5.11: Packet Loss Probability under MAC protocol: Performance of Event-Triggered Controller (ETC) w.r.t. number of cranes under IEEE 802.15.4 wireless network settings with unslotted CSMA/CA MAC protocol in nonbeaconed mode with back-offs (network parameter $\text{macMaxCSMABackoffs} \neq 0$). The graph shows that MAC protocol has significantly increased the network performance. The dash-dot red line is the maximum allowed packet loss probability above which control performance is not acceptable.

Chapter 6

Summary and Conclusion

In this thesis we have successfully designed and simulated the networked control of multiple tower cranes under event-triggered control policy. We, first, identified a good model for tower crane and then proposed *Full State Feedback Integral Control* scheme for the nonlinear system by transforming the nonlinear model into two *linear parameter varying (LPV)* subsystems. The eigenvalue assignment using full state feedback has been done for both LPV subsystems, independently, by transforming them into controllable canonical form. This resulted in two *Gain-Scheduled Integral Controllers* which have been linearized at carefully selected equilibrium point by using a special criterion based on the norm of “closed-loop pole-residue”. This linearization resulted in two *Fixed-Gain Integral Controllers* which achieved the semi-global asymptotic stability for original nonlinear system. Due to ease of mathematical analysis, the *Fixed-Gain Integral Controller* has been used for the design of event-triggered control using Lyapunov input-to-state stability theorem. Due to conservativeness in results and some practical issues, a few improvements, on the basis of heuristics, have been suggested to fine tune the event-generation rule to address implementation difficulties. Some issues regarding implementation of dynamic controller under event-triggered control policy has also been addressed. The complete simulation model for networked control of multiple tower cranes using event-triggered control policy over IEEE 802.15.4 network has been developed in MATLAB® Simulink using TrueTime toolbox. The performance analysis and stability has been evaluated thoroughly using simulations under different operating conditions. The performance of network control system has been thoroughly evaluated under network-induced delays and packet losses. The effect of MAC protocol on stability and scalability properties under event-triggered control policy has also been thoroughly investigated. In the following we will summarize the main results:

Accurate System Model

The identified model worked quite well for the control design purpose. We have validated our model thoroughly using four different tests which establish the validity of our model at least for input frequencies up to 1Hz. According to our most important Test-4, the controller designed on the basis of identified model also works for the *Simulink Model* provided by crane manufacturer and hence

it proves that our identified model served our model identification purpose.

CT Controller Design using LPV Design Methods

The identified nonlinear model has been transformed into two linear parameter varying (LPV) subsystems by defining L and X_w as varying parameters. The linear control design methods like eigenvalue assignment, using controllable canonical form, has been done to achieve the *Gain-Scheduled Controllers* which achieve quite good stability and performance as well for original nonlinear system in the desired region of operation. Due to complicated gain functions and difficulty in Lyapunov analysis, the *Fixed-Gain Controllers* have been used to design event-triggered controller.

Event-Triggered Controller for Step Tracking

The *Event-Triggered Controller* (ETC) has been proposed for tracking step reference signals. The basic “trick” to handle the tracking problem is to exploit the integral control structure to make the reference signal a part of system state. Then by shifting the origin using change of variables, we transformed the tracking problem into standard regulation problem where ‘r’ no more appears in state-equation explicitly but hidden inside plant states. The Lyapunov input-to-state stability theorem has been, then, applied easily to the new transformed system and we got the event-generation rule for step tracking. Some implementation issues related to even-triggered control have also been addressed. The proposed event-triggered control scheme works quite well for both single crane and multiple crane case under zero delays.

Reduction in Communication Cost

The proposed event-triggered controller has reduced the communication cost by 4 times as compared to periodic time-triggered controller (TTC) in presence of low delays.

Good Scalability Properties and Effect of MAC Protocol

ETC scales well with the increase in number of nodes. The MAC protocol plays a key role for achieving acceptable control performance under event-triggered control policy for large number of cranes in the network. If MAC protocol is not employed then the performance of network control system degrades very fast as function of number of cranes. The values of MAC parameters pose an optimization trade-off between control and network performance. The values only in certain band ensure that both the network and control performance is acceptable.

Effect of Delays

The delay has bit greater impact on ETC as compared to TTC for both network and control performance. The impact of delay also depends on type of delay. The network control system is affected most by Δ_{ca} and Δ_{sc} and least by Δ_{ped} whereas Δ_{eds} has intermediate effect. Also, the simulation reveals that ETC is advantageous only under certain conditions which must be ensured otherwise

periodic time-triggered controller is better. If the delay in the loop is not big then event-triggered control should be the preferred choice.

Effect of Packet Losses

Extensive simulations has revealed that the packet loss probability is not the only factor that affects the control performance but it also depends highly on the number of consecutive packet losses during transient phase of response.

Tracking Performance under ETC

The tracking performance under ETC for reference signals with higher order polynomials of time ' t ', is still an open research problem. Especially, it is not clear that for tracking general reference signal, whether the ETC will still outperform TTC in terms of less controller updates or not. Moreover, the good tracking performance under ETC highly depends on good tracking controller design in continuous-time domain. According to our proposed solution for asymptotically tracking step, the dynamic controller will be required which pose some difficulties for implementation of integration due to nonuniform sampling. We solved this problem using dedicated hardware and treating *integration* as state inside plant rather than state of controller. So according to our analysis, the tracking performance highly depends upon good controller design in continuous-time and also the type of reference signal.

Event-Triggered or Periodic ?

A decision to choose between Event-Triggered and Periodic Controller highly depends on the amount and type of delay in the system. Before going for ETC, a thorough analysis must be done for all kind of delays in the control loop over network otherwise, depending upon the amount of delay, the event-triggered controller may perform even worse than periodic time-triggered controller in terms of network utilization. In the presence of less amount of delay, Event-Triggered Controller should work better than periodic time triggered controller in terms of network utilization and almost similar in terms of control performance for control over wireless sensor networks.

Chapter 7

Future Work

We recommend to explore following areas for future research:

Model Improvements

- Better identification Input signal design may improve the estimated parameters for servo subsystems.
- Friction Estimation is recommended for better control performance.
- Backlash compensation during estimation may improve the model.

Stability Proof for LPV based Design

- Thorough stability analysis is required for original nonlinear closed-loop system under fixed-gain controller designed on the basis of LPV description. It will be interesting to analyze stability issues related to “*hidden couplings*”[14, 15].
- Design \mathcal{H}_∞ or \mathcal{H}_2 robust controller using parameterized LMI(Linear Matrix Inequality) through LPV design methods[14, 15]. But other than achieving robustness we also want to meet performance requirements to damp the payload oscillations within one oscillation cycle.

Event-Triggered Control

- Evaluate for general tracking problem (higher order polynomials for $r(t)$)
- Suggest dynamic controller implementation which does not require dedicated hardware.
- Evaluate the effect of actuator dead-zone.
- Find the better Lyapunov function for original nonlinear system and propose less conservative event-generation rule which triggers at $\dot{V} = 0$.

Find Optimal Controller for ETC

$$\max_{u \in U} (t_{k+1} - t_k) \quad \text{s.t.} \quad \begin{cases} \dot{x} &= f(x, u) \\ \dot{e} &= -f(x, u) \\ e &= x(t_k) - x(t) \quad t \in [t_k, t_{k+1}] \\ u &= \gamma(t_k, x(t_k)) \in U \\ x(0) &= x_i, x(t_f) = x_f \end{cases} \quad (7.0.1)$$

Where U = set of all admissible asymptotically stabilizing controllers.

Analytical Proof for Multiple Crane

Prove closed-loop stability analytically for multiple cranes under losses.

Effect of MAC Protocol

- Evaluate the effect of changing MAC parameters on control performance.
- Measure the average delay for each node as a function of the MAC parameters.
- Integrate MAC parameters in the controller synthesis problem and find controller in unified way

Effect of Delays and Packet dropouts

Derive analytical results to prove numerical evaluation.

Appendix A

DC Servo-Subsystem

A.1 Physical Modeling of DC Servo-Subsystem

The general but simplified model for any DC Servo system containing dc motor, gearbox, inertial load etc will be derived in this appendix for sake of completion. Here we will introduce two notations S_1 and S_2 to designate motor-shaft (input of gearbox) and gearbox-shaft (output of gearbox) respectively. In the following the general procedure has been outlined for modeling of such dc servo subsystems. First of all, we will define few physical parameters in the context of modeling of geared DC-Servo system. The total load inertia and damping acting on the S_2 reflected back to S_1 is given by:

$$J_{meq} = J_m + \underbrace{\frac{J_l}{K_g^2}}_{=J_{ml}} \quad (\text{A.1.1})$$

$$b_{meq} = b_m + \underbrace{\frac{b_l}{K_g^2}}_{b_{ml}} \quad (\text{A.1.2})$$

Where:

- J_{ml} = load inertia seen at S_1
- J_l = load inertia seen at S_2
- J_m = inertia of rotor of the motor
- J_{meq} = Total Inertia seen by the motor shaft, S_1
- K_g = Gear Ratio = $\frac{\theta_m}{\theta_l}$
- θ_m = Angular Position of motor
- θ_l = Angular Position of load
- b_l = damping coefficient of load as seen at S_2
- b_{ml} = damping coefficient of load as seen at S_1

- b_m = damping of rotor of motor
- $b_{m_{eq}}$ = Total damping seen by S_1

A.1.1 Mechanical Load Dynamic Equation

Looking at S_1 , the equation governing the dynamics of load is given by

$$J_{m_{eq}} \ddot{\theta}_m + b_{m_{eq}} \dot{\theta}_m + \frac{T_{dist}}{K_g} = T_m \quad (\text{A.1.3})$$

Where:

- T_m = Total torque delivered by S_1
- T_{dist} = Disturbance Torque (gravitational) at S_2

Since in our case $T_{dist} = 0$ so:

$$J_{m_{eq}} \ddot{\theta}_m + b_{m_{eq}} \dot{\theta}_m = T_m \quad (\text{A.1.4})$$

The equation (A.1.4) is for load dynamics at S_1 . To write equation for load dynamics at the S_2 , the equation (A.1.4) will be multiplied with K_g on both sides and then inserting values of $J_{m_{eq}}$ and $b_{m_{eq}}$ we get:

$$\begin{aligned} K_g(J_m + \frac{J_l}{K_g^2})\ddot{\theta}_m + K_g(b_m + \frac{b_l}{K_g^2})\dot{\theta}_m &= K_g T_m \\ (K_g^2 J_m + J_l) \frac{\ddot{\theta}_m}{K_g} + (K_g^2 b_m + b_l) \frac{\dot{\theta}_m}{K_g} &= K_g T_m \end{aligned} \quad (\text{A.1.5})$$

Let's define:

$$\theta_l = \frac{\theta_m}{K_g} \quad (\text{A.1.6})$$

$$J_{l_{eq}} = K_g^2 J_m + J_l \quad (\text{A.1.7})$$

$$b_{l_{eq}} = K_g^2 b_m + b_l \quad (\text{A.1.8})$$

Where:

- $J_{l_{eq}}$ = Total Inertia seen by S_2
- $b_{l_{eq}}$ = Total damping seen by S_2

Using (A.1.6), (A.1.7) and (A.1.8) in (A.1.5) we get the following dynamic equation:

$$J_{l_{eq}} \ddot{\theta}_l + b_{l_{eq}} \dot{\theta}_l = K_g T_m \quad (\text{A.1.9})$$

Now take the Laplace transform of (A.1.9)

$$J_{l_{eq}} s^2 \theta_l + b_{l_{eq}} s \theta_l = K_g T_m \quad (\text{A.1.10})$$

$$\Rightarrow T_m = \frac{J_{l_{eq}} s^2 \theta_l + b_{l_{eq}} s \theta_l}{K_g} \quad (\text{A.1.11})$$

A.1.2 Electrical Circuit Dynamic Equation

The electrical subsystem of DC-servo motor can be written as:

$$v_a(t) = R_a i_a + L_a \frac{di_a}{dt} + v_b(t) \quad (\text{A.1.12})$$

Where:

- R_a = Armature Resistance
- L_a = Armature Inductance
- i_a = Armature Current
- v_b = Back EMF Voltage
- v_a = Armature Voltage/Motor Input Voltage

Introducing the gyrator equations that govern the flow of signals between mechanical and electrical subsystems of DC-Servo system:

$$v_b = K_b \dot{\theta}_m \quad (\text{A.1.13})$$

$$T_m = K_T i_a \quad (\text{A.1.14})$$

where:

- K_b = Motor Back-EMF Constant ($\frac{V \cdot s}{rad}$)
- T_m = Torque generated by motor and delivered by S_1
- K_T = Torque Constant ($\frac{N \cdot m}{A}$)

Taking laplace transform of (A.1.12),(A.1.14) and (A.1.13) we get:

$$V_a(s) = R_a I_a(s) + L_a s I_a(s) + V_b(s) \quad (\text{A.1.15})$$

$$V_b(s) = K_b s \theta_m(s)$$

$$\because \theta_m = K_g \theta_l$$

$$\Rightarrow V_b(s) = K_b K_g s \theta_l(s) \quad (\text{A.1.16})$$

$$T_m = K_T I_a(s)$$

$$\Rightarrow I_a(s) = \frac{T_m}{K_T} \quad (\text{A.1.17})$$

Using (A.1.16) and (A.1.17) in (A.1.15) we get rid of $V_b(s)$ and $I_a(s)$ and also equation now shows the electrical circuit dynamics w.r.t angular speed of the load at S_2 .

$$V_a(s) = (R_a + s L_a) \frac{T_m}{K_T} + K_b K_g s \theta_l \quad (\text{A.1.18})$$

$$\Rightarrow T_m = \frac{(V_a(s) - K_b K_g s \theta_l) K_T}{(R_a + s L_a)} \quad (\text{A.1.19})$$

A.1.3 Transfer Function and Simplification of DC-Servo System

Comparing equations (A.1.19) and (A.1.11) we get:

$$\frac{(V_a(s) - K_b K_g s \theta_l) K_T}{(R_a + s L_a)} = \frac{J_{l_{eq}} s^2 \theta_l + b_{l_{eq}} s \theta_l}{K_g} \quad (\text{A.1.20})$$

Solving equation (A.1.20) for $\frac{\theta_l(s)}{V_a(s)}$ we get transfer function from armature voltage to load angular position:

$$\frac{\theta_l(s)}{V_a(s)} = \frac{K_g K_T}{s(s^2 L_a J_{l_{eq}} + (R_a J_{l_{eq}} + L_a b_{l_{eq}})s + R_a b_{l_{eq}} + K_g^2 K_T K_b)} \quad (\text{A.1.21})$$

Similarly the transfer function from armature voltage to load angular speed can be written as:

$$\frac{\omega_l(s)}{V_a(s)} = \frac{K_g K_T}{s^2 L_a J_{l_{eq}} + (R_a J_{l_{eq}} + L_a b_{l_{eq}})s + R_a b_{l_{eq}} + K_g^2 K_T K_b} \quad (\text{A.1.22})$$

The characteristics equation of (A.1.22) is 2nd order so it can be re-written as:

$$\frac{\omega_l(s)}{V_a(s)} = \frac{K_e}{s + p_e} + \frac{K_m}{s + p_m} \quad (\text{A.1.23})$$

Where:

- p_e = pole describing electrical dynamics
- p_m = pole describing mechanical dynamics
- K_e = DC Gain for electrical subsystem of DC Servo system
- K_m = DC Gain for mechanical subsystem of DC servo system

Taking inverse Laplace transform of (A.1.23) we get the zero-state response:

$$\omega_l(t) = \int_0^t (K_e e^{-p_e(t-\xi)} + K_m e^{-p_m(t-\xi)}) V_a(\xi) d\xi \quad (\text{A.1.24})$$

In most of the cases $p_e \gg p_m$ which means natural response of the electrical dynamics would decay very fast as compared to that of mechanical subsystem so it can be safely assumed that the response of the dc servo system would be mainly governed by the pole of mechanical subsystem being a dominant pole:

$$\omega_l(t) = \int_0^t e^{-p_e(t-\xi)} V_a(\xi) d\xi \quad (\text{A.1.25})$$

By looking at equation (A.1.15) we can see that the transfer function for electrical subsystem is given by:

$$\frac{I_a(s)}{V_a(s) - V_b(s)} = \frac{1/L_a}{s + R_a/L_a} \quad (\text{A.1.26})$$

$\because p_e \gg p_m$ so equation (A.1.26) \Rightarrow we can neglect L_a i.e. $L_a = 0$ which means we are allowing current to rise instantaneously hence (A.1.22) reduces to:

$$\frac{\omega_l(s)}{V_a(s)} = \frac{K_g K_T}{R_a J_{l_{eq}} s + R_a b_{l_{eq}} + K_g^2 K_T K_b} \quad (\text{A.1.27})$$

$$\frac{\omega_l(s)}{V_a(s)} = \frac{\frac{K_g K_T}{R_a b_{l_{eq}} + K_g^2 K_T K_b}}{\frac{R_a J_{l_{eq}}}{R_a b_{l_{eq}} + K_g^2 K_T K_b} s + 1} \quad (\text{A.1.28})$$

Now for the sake of completion lets also introduce the input-output delay in the model:

$$\frac{\omega_l(s)}{V_a(s)} = \frac{\frac{K_g K_T}{R_a b_{l_{eq}} + K_g^2 K_T K_b}}{\frac{R_a J_{l_{eq}}}{R_a b_{l_{eq}} + K_g^2 K_T K_b} s + 1} \cdot e^{-s\tau_d} \quad (\text{A.1.29})$$

The model (A.1.29) is usually called **KLT Model** with following parameters defined as:

DC Gain of Complete Servo System:

$$K = \frac{K_g K_T}{R_a b_{l_{eq}} + K_g^2 K_T K_b} \quad (\text{A.1.30})$$

Mechanical Time Constant of Complete Servo System:

$$\tau = \frac{R_a J_{l_{eq}}}{R_a b_{l_{eq}} + K_g^2 K_T K_b} \quad (\text{A.1.31})$$

It is called **KLT Model** because it contains three parameters: K-Gain, T-time constant and L-delay. So to approximate the DC-Servo system behavior we need to estimate these three parameters: K , τ and τ_d .

References

- [1] Inteco 3D Tower Crane User Manual
- [2] P. Tabuada, “Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks”, IEEE Transactions on Automatic Control, vol. 52, no. 9, pp. 1680-1685, Sep. 2007.
- [3] H. Khalil, Nonlinear systems. Prentice Hall Upper Saddle River, NJ, 2002.
- [4] Åström and Wittenmark, Computer-Controlled Systems Theory and Design Third Edition
- [5] Hanafy M. Omar, Control of Gantry and Tower Cranes, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2003.
- [6] Lennart Ljung and Torkel Glad, “Modeling of Dynamic Systems”. Prentice Hall Upper Saddle River, NJ, 1994
- [7] Lennart Ljung, “System Identification: Theory for the User”. Prentice Hall Upper Saddle River, NJ, 2nd Edition, 1999 Lennart Ljung (Author)(Author), (Author)
- [8] Chi-Tsong Chen , “Linear System Theory and Design”. Oxford University Press, Inc. NY, 3rd edition, 1999
- [9] P. Tabuada and X. Wang, “Preliminary results on state-triggered scheduling of stabilizing control tasks”. Proceedings of the 45th IEEE Conference on Decision and Control, December 2006.
- [10] Manuel Mazo Jr., Adolfo Anta and Paulo Tabuada, “On Self-Triggered Control for Linear Systems: Guarantees and Complexity”. Proceedings of the European Control Conference, August 2009.
- [11] Manuel Mazo Jr. and Paulo Tabuada, “Event-triggered and Self-triggered control over sensor/actuator network”. Proceedings of the 47th IEEE Conference on Decision and Control, December 2008.
- [12] K. J. Astrom and B. Bernhardsson, “Comparison of riemann and lebesgue sampling for first order stochastic systems”, in Proc. 41st IEEE Conf. Decision and Control, Dec. 2002.
- [13] Lennart Ljung, “System Identification ToolboxTM 7 Users Guide”. The MathWorks, Inc., 2010.

- [14] W. Rugh and J. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 14011425, October 2000
- [15] J. Shamma, "Gain scheduling: Potential hazards and possible remedies," *IEEE Control System Magazine*, vol. 12, pp. 101107, 1992.

