Adaptive Model Predictive Control

Denis Koksal-Rivet

February 2019

FINAL HONOUR SCHOOL OF ENG

DECLARATION OF AUTHORSHIP

You should complete this certificate. It should be bound into your fourth year project report, immediately after your title page. Three copies of the report should be submitted to the Chairman of examiners for your Honour School, c/o Clerk of the Schools, examination Schools, High Street, Oxford.

Name (in capitals): ……………………………………………………………………………………………

College (in capitals): ………………………………… **Supervisor:** …………………………………………

Title of project (in capitals): ………………………………..…………………………………………………

Page count (excluding risk and COSHH assessments): …………………………………………………

Please tick to confirm the following:

I have read and understood the University's disciplinary regulations concerning conduct in examinations and, in particular, the regulations on plagiarism (*The University Student Handbook. The Proctors' and Assessors' Memorandum, Section 8.8*; available at https://www.ox.ac.uk/students/academic/student-handbook) □

I have read and understood the Education Committee's information and guidance on academic good practice and plagiarism at https://www.ox.ac.uk/students/academic/guidance/skills. □

The project report I am submitting is entirely my own work except where otherwise indicated. □

It has not been submitted, either partially or in full, for another Honour School or qualification of this University (except where the Special Regulations for the subject permit this), or for a qualification at any other institution. □

I have clearly indicated the presence of all material I have quoted from other sources, including any diagrams, charts, tables or graphs. □

I have clearly indicated the presence of all paraphrased material with appropriate references. □

I have acknowledged appropriately any assistance I have received in addition to that provided by my supervisor. □

I have not copied from the work of any other candidate. □

I have not used the services of any agency providing specimen, model or ghostwritten work in the preparation of this project report. (See also section 2.4 of Statute XI on University Discipline under which members of the University are prohibited from providing material of this nature for candidates in examinations at this University or elsewhere: http://www.admin.ox.ac.uk/statutes/352-051a.shtml.) □

The project report does not exceed 50 pages (including all diagrams, photographs, references and appendices). □

I agree to retain an electronic copy of this work until the publication of my final examination result, except where submission in hand-written format is permitted. □

I agree to make any such electronic copy available to the examiners should it be necessary to confirm my word count or to check for plagiarism. □

Candidate's signature: …………………………………………….. **Date:** ………………………..

Contents

1 Introduction

Starting from the 1960s onwards, advanced process control techniques have been of particular interest in both academia and industry. Early versions of model predictive control algorithms were first developed in the 1970s independently by Cutler and Ramaker at Shell Oil and Jacques Richalet [2]. These control methods were first developed in the process and petrochemical industries and improved controller performance compared to PID controllers that were being used at the time. Based on the success of the predictive control methods, MPC algorithms have become the most common control methodology in use in industry today.

Model predictive control is an optimal control strategy that relies on numerical optimization where future control inputs and plant responses are predicted using a system model and optimized with regards to a performance index. MPC has become a pervasive control methodology due to its ease of implementation, ability to scale to large problems, concrete stability, optimality, and robustness properties, and systematic method for dealing with input and state constraints [3].

The performance of model predictive controllers is strongly dependent upon the quality of the underlying model as inaccurate models lead to poor predictions and thus poor control of the plant. Stochastic and robust MPC approaches have commanded a lot of attention in the relevant literature as these methods deal effectively with unmodeled dynamics and quickly changing disturbances, but are inherently conservative in problems with slow changing or constant parametric uncertainty [4]. Bounds on the performance of MPC laws have led to developments that combine adaptive control with MPC which have led to decreased conservativeness compared to robust MPC and robust stability guarantees. The recent literature on adaptive MPC examines several methods for parameter identification such as comparison sets, set membership identification, recursive least squares, and neural networks. Set membership identification techniques describe a set that the true parameters of the model lie in which in turn defines a set of possible plant models, and then enforces that constraints are satisfied on all plant models within this set [1].

2 Notation

The notation used in this report will be consistent with that used by Mark Cannon and Xiaonan Lu in their paper Robust Adaptive Tube Model Predictive Control. Integers and reals will be denoted by $\mathbb{N}$ and $\mathbb{R}$ respectively, and $\mathbb{N}_{\geq 0} = \{n \in \mathbb{N} : n \geq 0\}$. The i^{th} row of a matrix and i^{th} element of a vectore will be represented as $[A]_i$ and $[a]_i$. $max_{x \in \mathbb{X}} J(x)$ will denote the maximum value of J over the set $\mathbb{X}$ while $[x]_{\geq 0} = max\{0, x\}$. The inequality $A \geq B$ will be taken to represent an element wise inequality where as $A \succeq B$ will mean $x^T Ax \geq x^T Bx$ for all vectors x and square matrices A and B. Finally, the prediction for the variable z k steps ahead, predicted at time t, will be indicated by $z_{k|t}$ [1].

3 Tube Based Model Predictive Control Algorithm

Cannon and Lu consider state space systems and implement set-based model identification and robust tube MPC. The system considered in this paper is a linear system with linear constraints on states and inputs, with an unknown additive disturbance term.

$$x_{t+1} = A(\theta)x_t + B(\theta)u_t + w_t \tag{1}$$

The system matrices $A(\Theta)$ and $B(\Theta)$ depend on an unknown parameter Θ through the relationship given in equation 2.

$$(A(\theta), B(\theta)) = (A_0, B_0) + \sum_{i=1}^{p}(A_i, B_i)[\theta]_i \tag{2}$$

The constraints on the state and inputs are expressed as

$$Fx_t + Gu_t \leq \mathbf{1} \ \ \forall t \in \mathbb{N}_{\geq 0} \tag{3}$$

and the unknown additive disturbance lies in the convex polytopic set $\mathbb{W}$ where

$$\mathbb{W} = \{w : \Pi_w w \leq \pi_w\} \tag{4}$$

with $\pi_w \geq 0$. A simplifying assumption that $\Theta_0 = \{\theta : \Pi_\theta \theta \leq \pi_0\}$ takes the form of a zonotope ($\Pi_\theta^T = [\Pi^T \quad -\Pi^T]$)is made in order to restrict the complexity of the shape of the parameter set. As in many MPC formulations the predicted control sequences have the dual form:

$$u_{k|t} = \begin{cases} Kx_{k|t} + v_{k|t}, & \forall k \in \mathbb{N}_{[0,N-1]} \\ Kx_{k|t}, & \forall k \geq N \end{cases} \tag{5}$$

The objective is thus to solve an optimal regulation problem to ensure stability of the closed-loop system while ensuring that the constraints in equation 3 are satisfied at each time step, given knowledge of $\mathbb{W}$ and Θ_0. The paper also pursues the additional objective of updating the set Θ_t using state measurements and using the updated parameter set to improve the controller performance, the details of which will be discussed later in this report.

The paper makes the assumption that there exists a polytopic set $\mathbb{X} = \{x : Vx \leq \mathbf{1}\}$ and feedback gain K so that $\mathbb{X}$ is λ-contractive for some $\lambda \in [0, 1)$:

$$V(A(\Theta) + B(\Theta)K)x \leq \lambda\mathbf{1} \quad \forall\ x \in \{x : Vx \leq \mathbf{1}\} \text{ and } \theta \in \Theta_0 \tag{6}$$

The matrices V and K are designed offline [1].

3.1 Polytopic Tubes for Constraint Satisfaction

The predicted tube for the state x at time k is defined as $\mathbb{X} = \{x : Vx \leq \alpha_k\}$. At each time step, a sequence of sets $\mathbb{X}_0, \mathbb{X}_1, ..., \mathbb{X}_{N-1}$ is constructed that satisfy

$$\Phi(\theta)x + B(\theta)v_k + w \in \mathbb{X}_{k+1} \quad \forall\ x \in \mathbb{X}_k, w \in \mathbb{W}, \theta \in \Theta \tag{7}$$

where $\Phi(\theta) = A(\theta) + B(\theta)K$. For $x_k \in \mathbb{X}\neg \text{and} u_k = Kx_k + v_k$, the constraints 3 are satisfied for the kth prediction time step if

$$H_c\alpha_k + Gv_k \leq \mathbf{1} \tag{8}$$

where H_c is computed offline by solving the linear program below.

$$[H_c]_i = \underset{H \in \mathbb{R}^{1xn_\alpha}}{\text{argmin}}\ H\mathbf{1} \ \text{ s.t. } H \geq 0, HV = [F + GK]_i \ \text{ for each } i \in \mathbb{N}_{[1,n_c]} \tag{9}$$

The proof is as follows: the constraints 3 are satisfied for $x_k \in \mathbb{X}_k$ and$u_k = Kx_k + v_k$ if,

$$\begin{aligned} Fx + G(Kx + v_k) &\leq \mathbf{1} \\ (F + GK)x + Gv_k &\leq \mathbf{1} \end{aligned} \tag{10}$$

where we know $\mathbb{X}_k = \{x : Vx \leq \alpha_k\}$, which can be expressed as $\{x : Vx \leq \alpha_k\} \subseteq \{x : (F + GK)x \leq \mathbf{1} - Gv_k\}$. Using proposition 3.31 from [5], this statement is equivalent to there being a matrix H_c such that $H_c \geq 0,\ H_cV = F + GK,\ H_c\alpha_k + Gv_k \leq \mathbf{1}$, which is equivalent to solving the linear program in equation 9. Furthermore, constraint 7 is satisfied if

$$\hat{\theta}^T \hat{H}_i\alpha_k + [V]_i B(\theta^{(j)}v_k + \bar{w}_i \leq [\alpha_{k+1}]_i \tag{11}$$

for all $j = 1, ..., m$ and $i = 1, ..., n_\alpha$ where $\bar{w}_i = \max_{w\in\mathbb{W}}[V]_i w$, $\hat{\theta}^{(j)T} = [1 \;\; \theta^{(j)T}]$, and the matrixes $\hat{H}_i$ are computed offline as the solution of the following linear program for each $i = 1, ..., n_\alpha$.

$$\begin{aligned} \hat{H}_i = \underset{H\in\mathbb{R}^{(p+1)\ x\ n_\alpha}}{\text{argmin}} \; \max_{j\in\mathbb{N}_{[1,m]}} \; \hat{\theta}^{(j)T} H\mathbf{1} \\ \text{s.t. } \hat{\theta}^{(j)T} H \geq 0, \; j \in \mathbb{N}_{1,m}, \; HV = \begin{bmatrix} [V]_i\Phi_0 \\ \vdots \\ [V]_i\Phi_p \end{bmatrix} \end{aligned} \tag{12}$$

The proof of this follows along similar lines to the previous proof: The condition of satisfaction of equation 7 is equivalent to $\{x : Vx \leq \alpha_k\} \subseteq \{x : V\Phi(\theta)x + VB(\theta)v_k + \bar{w} \leq \alpha_{k+1}\}$. Again, by proposition 3.31 in [5], this is equivalent to the following conditions being satisfied:

$$\hat{\theta}^{(j)T}\hat{H}_i \geq 0 \tag{13}$$

$$\hat{H}_i V = [[V]_i\Phi_0 \; ... \; [V]_i\Phi_p]^T \tag{14}$$

$$\hat{\theta}^{(j)T}\hat{H}_i\alpha_k + [V]_i B(\theta^{(j)})v_k + \bar{w} \leq [\alpha_{k+1}]_i \tag{15}$$

13 and 14 are computed offline by solving the LP in euqation 12 for each $i \in \mathbb{N}_{[1,n_\alpha]}$. Thus, constraint 15 is enough to ensure satisfaction of the constraint 7. The initial and final conditions on the state tubes are imposed through the conditions below [1].

$$\alpha_0 \geq Vx \tag{16}$$

$$[\alpha_N]_i \geq \hat{\theta}^{(j)T}\hat{H}_i\alpha_N + \bar{w}, \;\; \forall\, j = 1, ..., m \text{ and } i = 1, ..., n_\alpha \tag{17}$$

$$H_c\alpha_N \leq \mathbf{1} \tag{18}$$

3.2 Design of V and K

As the matrix V defines the tube cross section $\mathbb{X}_k$ at every time step, increasing the number of rows, n_α, in V will result in tighter bounds on the predicted state tube. However, there is a tradeoff that has to be considered, as increasing n_α increases the dimensions of many of the matrices invloved in the online optimization, thus greatly increasing the computation needed to compute an optimal solution at each time step.

For a given V, the λ value that satisfies the λ-contractivity condition 6 is found by minimising λ subject to $\{x : Vx \leq \mathbf{1}\} \subseteq \{x : V\Phi(\theta)x \leq \lambda\mathbf{1}$. When V is chosen independently of K this optimization is linear with respect to K and $]lambda$. V should be chosen as $V = [Q^T \;\; F_0^T]^T$ where F_0 is the rows of F where the corresponding rows of G are equal to 0. Again using proposition 3.31 from [5], the calculation of K is reduced to solving the LP problem below [1].

$$
\begin{aligned}
(\lambda, K, \hat{H}_1, ..., \hat{H}_{n_\alpha}) &= \underset{\lambda, K, \hat{H}_1, ..., \hat{H}_{n_\alpha}}{\text{argmin}} \quad \lambda \\
\text{s.t} \quad \lambda \geq \hat{\theta}^{(j)T} \hat{H}_i \mathbf{1}, \ \hat{\theta}^{(j)T} \hat{H}_i &\geq 0 \quad \forall \ j = 1, ..., m, \quad i = 1, ..., n_\alpha \\
\hat{H}_i V &= [[V]_i (A_0 + B_0 K) \ \ ... \ \ [V]_i (A_p + B_p K)]^T
\end{aligned}
\tag{19}
$$

3.3 Computing the Optimal Input

Having set out all of the constraints for the tube based MPC problem, the online optimization to compute the optimal input at each time step can now be formulated. The online optimization minimises a given cost $J(x, u)$ with respect to the constraints discussed in the previous section. More formally, the online MPC algorithm is as follows:

$$
\begin{aligned}
J^o &= \min_{\boldsymbol{v}, \boldsymbol{\alpha}} J(x, u) \\
\text{s.t} \quad H_c \alpha_k + G v_k &\leq \mathbf{1} \\
\hat{\theta}^T \hat{H}_i \alpha_k + [V]_i B(\theta^{(j)} v_k + \bar{w}_i &\leq [\alpha_{k+1}]_i \\
\hat{\theta}^{(j)T} \hat{H}_i &\geq 0 \\
\hat{H}_i V &= [[V]_i \Phi_0 \ ... \ [V]_i \Phi_p]^T \\
\hat{\theta}^{(j)T} \hat{H}_i \alpha_k + [V]_i B(\theta^{(j)}) v_k + \bar{w} &\leq [\alpha_{k+1}]_i
\end{aligned}
\tag{20}
$$

where the sequences $\boldsymbol{\alpha}$ and $\boldsymbol{v}$ are the optimization variables [1].

4 Model Identification

This section will discuss methods for set based model identification, examine a Least Mean Squares method for deriving a point estimate for the model parameter, and explore the relevant theory on the rate of parameter set convergence. Model identification is an important component of the entire MPC algorithm explored in this report as the performance of the controller discussed in the previous section improves greatly as the model becomes more accurate.

4.1 Set Based Model Identification

Cannon and Lu's discussion of set based model identification is based off of the theory explained in [6] and [7]. Observations of the system states are used online to construct polytopes that the possible model parameters could lie in. This is called the unfalsified parameter set and it is combined with the current polytopic parameter set estimate to construct the new parameter set estimate. The unfalsified parameter set is defined as

$$\begin{aligned} \Delta_t &= \{\theta : x_t - (A(\theta)x_{t-1} + B(\theta)u_{t-1}) \in \mathbb{W}\} \\ &= \{\theta : -\Pi_w D(x_{t-1}, u_{t-1})\theta \leq \pi_w + \Pi_w d_t\} \\ &= \{\theta : P_t\theta \leq Q_t\} \end{aligned} \tag{21}$$

where $P_t = -\Pi_w D(x_{t-1}, u_{t-1})$ and $Q_t = \pi_w + \Pi_w d_t$. $D(x,u)$ and d_t are defined according to the equations below.

$$D(x,u) = \begin{bmatrix} A_1x + B_1u & \dots & A_px + B_pu \end{bmatrix} \tag{22}$$

$$d_t = A_0x_{t-1} + B_0u_{t-1} - x_t \tag{23}$$

For the parameter set update algorithm, Π_θ is a matrix chosen before hand that defines the initial parameter set $\Theta_0 = \{\theta : \Pi_\theta\theta \leq \pi_0\}$, and at each subsequent time step the parameter set is defined by $\Theta_t = \{\theta : \Pi_\theta\theta \leq \pi_t\}$ where π_t is determined online by the parameter set update algorithm. The complexity of the parameter set can be controlled by choosing Π_θ appropriately, and it is common to define the parameter set to be a hypercube by defining $\Pi_\theta = [I \quad -I]^T$. The smallest set Θ_{t+1} satisifying $\Theta_{t+1} \supseteq \Theta_t \cap \Delta_t$ is obtained by solving the linear program below for $i = 1, ..., 2n$.

$$\begin{aligned} [\pi_{t+1}]_i &= \min_{\pi, H_i} \pi \\ \text{s.t.} \quad H_i \begin{bmatrix} \Pi_\theta \\ P_t \end{bmatrix} &= [\Pi_\theta]_i \\ H_i \begin{bmatrix} \pi_t \\ Q_t \end{bmatrix} &\leq \pi \\ H_i &\geq 0 \end{aligned} \tag{24}$$

If $\Pi_\theta\theta \leq \pi_0$ and π_{t+1} is found by solving the LP 24, then $\Theta_t \supseteq \Theta_{t+1} \supseteq \Theta_t \cap \Delta_t$, i.e. the parameter set is not expanding. π_{t+1} also defines the minimum volume set containing the union of Θ_t and Δ_t . The proof is as follows: The constraints in equation 24 necessarily ensure that $\Theta_{t+1} \supseteq \Theta_t \cap \Delta_t$, implying by induction that $\theta \in \Theta_t$ for all t. $H_i = [\mathbb{I} \quad 0]$ and $\pi_{t+1} = \pi_t$ provide a feasible solution for (24), so it must follow that $\pi_{t+1} \leq \pi_t$ and thus $\Theta_{t+1} \subseteq \Theta_t$ for all t. The volume of Θ_{t+1} is given by $(2_p/det(\Pi_\theta)) \prod_{i=1}^{p}([\pi_{t+1}]_i + [\pi_{t+1}]_{p+1})$ which is minimized for any Π_θ by the choice of objective function in equation 24 [1].

4.2 Point Estimate

A point estimate method for the unknown model parameters θ can be used in conjunction with the set-based model identification algorithm discussed previously to ensure a finite closed-loop gain from the additive disturbance to the system state x_k [4]. In order to provide this stability guarantee between the disturbance and state, a Least Mean Squares filter is used. Hassibi et al. have shown in [8] that the LMS filter is an H_∞ optimal map from the disturbance sequence to the sequence of prediction errors, resulting in an achievable gain of 1. The point estimate algorithm is as follows: given an estimate for the model parameters, $\hat{\theta}_k$, define the predicted state to be $\hat{x}_k = A(\hat{\theta}_k)x_k + B(\hat{\theta}_k)u_k$ and the prediction error is thus denoted as $\tilde{x}_{i|k} = A(\theta^*)x_k + B(\theta^*)u_k - \hat{x}_{1|k}$. For a given initial estimate of the parameters, $\hat{\theta}_0 \in \Theta$ and a paramter update gain μ satisfying $\frac{1}{\mu} > \sup_{(x,\mu)} ||D(x,\mu)||^2$, the estimate is defined recursively according to the equations below.

$$\begin{aligned} \tilde{\theta}_k &= \hat{\theta}_{k-1} + \mu D(x_{k-1}, u_{k-1})^T (x_k - \hat{x}_{1|k-1}) \\ \hat{\theta}_k &= \Pi_\Theta(\tilde{\theta}_k) \end{aligned} \tag{25}$$

$\Pi_\Theta(\tilde{\theta})$ represents the Euclidean projection of a point $\tilde{\theta} \in \mathbb{R}^p$ onto the set Θ and is calculated as $\Pi_\Theta(\tilde{\theta}) = \text{argmin}_{\theta \in \Theta} ||\theta - \tilde{\theta}||$. This point estimate then results in the aforementioned guarantee about the closed-loop gain from disturbance to system state. If $\sup_k ||x_k|| < \infty$, $\sup_k ||u_k|| < \infty$, then $\hat{\theta}_k$ is bounded ($\hat{\theta}_k \in \Theta$) and

$$\sup_{m \in \mathbb{N}, w_k \in \mathbb{W}, \hat{\theta}_0 \in \Theta} \frac{\sum_{k=0}^m ||\tilde{x}_{1|k}||^2}{\frac{1}{\mu}||\hat{\theta}_0 - \theta^*||^2 + \sum_{k=0}^m ||w_k||^2} \leq 1. \tag{26}$$

The boundedness of $\hat{\theta}_k$ and $\hat{\theta}_k \in \Theta$ follow trivially from equations 21 and projection. The proof for equation 26 is not as trivial; the proof is as follows:

$$\begin{aligned} &\frac{1}{\mu}||\hat{\theta}_{k+1} - \theta^*||^2 - \frac{1}{\mu}||\hat{\theta}_k - \theta^*||^2 \\ &\leq \frac{1}{\mu}||\tilde{\theta}_{k+1} - \theta^*||^2 - \frac{1}{\mu}||\hat{\theta}_k - \theta^*||^2 \\ &= \frac{1}{\mu}||\tilde{\theta}_{k+1} - \hat{\theta}_k||^2 + \frac{2}{\mu}(\tilde{\theta}_{k+1} - \hat{\theta}_k)^T(\hat{\theta}_k - \theta^*) \\ &= \frac{1}{\mu}||\mu D_k^T(\tilde{x}_{1|k} + w_k||^2 + 2(\tilde{x}_{1|k} + w_k)TD_K(\hat{\theta}_k - \theta^*) \\ &\leq (\mu||D_k||^2 - 1)||\tilde{x}_{1|k} + w_k||^2 - ||\tilde{x}_{1|k}||^2 + ||w_k||^2 \\ &\leq -||\tilde{x}_{1|k}||^2 + ||w_k||^2 \end{aligned} \tag{27}$$

where $D_k = D(x_k, u_k)$. The first inequality is due to the projection operator being non-expansive and $\theta^* \in \Theta$. Equations 25 are used in the second equality and inequality along with a completion of squares, using the identities $x_{k+1} - \hat{x}_{1|k} = \tilde{x}_{1|k} + w_k, \ \tilde{x}_{1|k} = D_k(\theta^* - \hat{\theta}_k)$. Summing 27 from $k = 0$ to m gives

$$\frac{1}{\mu}||\hat{\theta}_{m+1} - \theta^*||^2 + \sum_{k=0}^{m} ||\tilde{x}_{1|k}||^2 \leq \sum_{k=0}^{m} ||w_k||^2 + \frac{1}{\mu}||\hat{\theta}_0 - \theta^*||^2 \tag{28}$$

which proves the claim above. As a result, the prediction error converges to 0 asymptotically if $\sup_k ||x_k||^2 < \infty, \ \sup_k ||u_k||^2 < \infty$, and $\sum_{k=0}^{\infty} ||w_k||^2 < \infty$ [4].

4.3 Parameter Set Convergence

Having defined the set membership based technique for model identification, it is important to examine the properties of this membership set, chiefly whether it converges to the true parameters and how fast it converges. Bai, Cho, and Tempo's paper [9] provides a thorough examination of the bounded-error parameter estimation approach making use of some probabilistic assumptions. The choice of noise model is very important for parameter identification, and several different models are discussed in [9]. For conciseness, only the pointwise bound noise model will be discussed in this report, but the interested reader is directed to [9] for a more complete discussion surrounding different types of noise models.

A pointwise bound noise model is defined as

$$\mathbb{W} = \{w : \max_{1 \leq i \leq N} |w_i| \leq \epsilon\}. \tag{29}$$

For a system defined by $y = \Phi\theta + \nu$ where Φ is the regressor and ν is the additive noise, the membership set can be identified by the equation

$$S_N^p = \cap_{i=1}^N \{\hat{\theta} : |y_i - \phi_i^T \hat{\theta}| \leq \epsilon. \tag{30}$$

The motivation for constructing the membership set is due to the fact that given an estimate for the model parameters $\hat{\theta}$ and the presence of noise, it is generally not possible to know whether the estimate coincides with the true parameters, but it is possible to determine whether the the estimate for the parameters is consistent with observed input-output data. To evaluate the quality of parameter estimation, the diameter of the membership set is defined as

$$\text{dia}S_N^p = \sup_{\theta_1, \theta_2} ||\theta_1 - \theta_2||_2, \tag{31}$$

with the 2-norm being used to simplify the calculation. The volume of the membership set is also often used in the literature as a quantitative measure of parameter estimation, but a zero-volume membership set does not necessarily imply zero error in the parameter estimation, and thus this report will make use of the membership set diameter when evaluating parameter estimation error. A small membership set diameter infers small uncertainty in the parameter estimate, and if the diameter is zero, the membership set is a singleton at the true parameter value.

Bai et al. show that, if the noise is assumed to be a sequence of random variables, the diameter of the membership set converges. The assumption that the noise is a random variable is not necessarily straightforward. In cases where the a significant portion of the noise is due to under-modeling, the assumption will not hold, and guarantees about the convergence of the membership set diameter cannot be made. The discussion around the guarantee of diameter convergence must first begin with the definition of tightness of the noise bound. For the pointwise bound noise model, a bound ϵ on the random noise variable w_i is said to be tight if there is a non=zero probability that the value of w_i lies on the boundary at $-\epsilon$ or ϵ. More formally, for any $\rho > 0$ and each i

$$\text{Prob}\{-\epsilon \leq w_i \leq -\epsilon + \rho\} \geq p_1(\rho) > 0 \tag{32}$$

and

$$\text{Prob}\{\epsilon - \rho \leq w_i \leq \epsilon\} \geq p_1(\rho) > 0 \tag{33}$$

for some $1 > p_1(\rho) > 0$.

For the system being considered with a pointwise noise model, the diameter of the membership set converges to zero as $N \to \infty$ with probability of 1 if the noise sequence $\{w_i\}$ is a sequence of random variables and the bound ϵ is tight, and the regressor of the system is persistently exciting. The regressor ϕ_i is persistently exciting if

$$\alpha^2 I \leq \frac{1}{m} \sum_{i=i_o+1}^{i_0+m} \phi_i \phi_i^T \leq \beta^2 I \tag{34}$$

for some $\alpha, \beta, m > 0$ and all i_0. The upper bound is automatically satisfied if the system is stable and the input and noise are bounded. The proof is rather convoluted, but included below for completeness.

Proof: Let $\hat{\theta}$ be any arbitrary fixed point not equal to the true parameter value θ. To show that $\hat{\theta}$ is excluded from $S_N^p(\epsilon)$ in probability as $N \to \infty$, define

$$z_k = \max\{\frac{1}{m}\sum_{i=1}^{m}(y_i - \phi_i^T\hat{\theta})^2, \frac{1}{m}\sum_{i=m+1}^{2m}(y_i - \phi_i^T\hat{\theta})^2, ..., \frac{1}{m}\sum_{i=(k-1)m+1}^{km}(y_i - \phi_i^T\hat{\theta})^2\}. \tag{35}$$

It is enough to show that $\text{Prob}\{z_k > \epsilon^2\} \to 1$ as $k \to \infty$. Given $\tilde{\theta} = \theta - \hat{\theta}$ and ϕ_i is persistently exciting, the following equation is true for any $i_0 \geq 0$:

$$\frac{1}{m}\sum_{i=i_0+1}^{i_0+m} |\phi_i^T \tilde{\theta}|^2 \geq \alpha^2 ||\tilde{\theta}||^2 > 0 \text{ and } \sum_{i=i_0+1}^{i_0+m} |\phi_i^T \hat{\theta}| > 0 \tag{36}$$

ϕ_i is deterministic for fixed $\hat{\theta}$, and $\phi_i^T \tilde{\theta}$ is fixed and independent of w_i. Because the w_i's are independent, the bound ϵ is tight, i.e. w_i hits both the positive and negative bounds with a non-zero probability, the equation below follows for some $p > 0$ independent of i_0.

$$\text{Prob}\{\epsilon^2 - \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} w_i^2 \leq \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} |\phi_i^T \tilde{\theta}|^2 \text{ and } \sum_{i=i_0+1}^{i_0+m} \phi_i^T \tilde{\theta} w_i > 0\} \geq p > 0 \tag{37}$$

Now,

$$\frac{1}{m}\sum_{i=i_0+1}^{i_0+m} |\phi_i^T \tilde{\theta}|^2 \geq \epsilon^2 - \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} w_i^2 \text{ and } \sum_{i=i_0+1}^{i_0+m} \phi_i^T \tilde{\theta} w_i > 0 \tag{38}$$

implies

$$\begin{aligned} &\frac{1}{m}\sum_{i=i_0+1}^{i_0+m} |\phi_i^T \tilde{\theta}|^2 \geq \epsilon^2 - \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} w_i^2 \\ &> \epsilon^2 - \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} w_i^2 - \frac{2}{m}\sum_{i=i_0+1}^{i_0+m} \phi_i^T \tilde{\theta} w_i \end{aligned} \tag{39}$$

and in turn

$$\begin{aligned} &\frac{1}{m}\sum_{i=i_0+1}^{i_0+m} (y_i - \phi_i^T \hat{\theta})^2 = \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} (\phi_i^T \tilde{\theta} + w_i)^2 \\ &= \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} |\phi_i^T \tilde{\theta}|^2 + \frac{1}{m}\sum_{i=i_0+1}^{i_0+m} w_i^2 + \frac{2}{m}\sum_{i=i_0+1}^{i_0+m} \phi_i^T \tilde{\theta} w_i > \epsilon^2. \end{aligned} \tag{40}$$

Thus,

$$\text{Prob}\{\frac{1}{m}\sum_{i=i_0+1}^{i_0+m} (y_i - \phi_i^T \hat{\theta})^2 > \epsilon^2\} \geq p \tag{41}$$

or

$$\text{Prob}\{\frac{1}{m}\sum_{i=i_0+1}^{i_0+m} (y_i - \phi_i^T \hat{\theta})^2 \leq \epsilon^2\} \leq (1-p) \tag{42}$$

Therefore,

$$\begin{aligned} \text{Prob}\{z_k > \epsilon^2\} &= 1 - \text{Prob}\{z_k \leq \epsilon^2\} \\ = 1 - \text{Prob}\{\frac{1}{m}\sum_{i=1}^{m}(y_i - \phi_i^T\hat{\theta})^2 &\leq \epsilon^2\}... \\ \text{Prob}\{\frac{1}{m}\sum_{i=(k-1)m+1}^{km}(y_i - \phi_i^T\hat{\theta})^2 &\leq \epsilon^2\} \\ &\geq 1 - (1-p)^k \to 1 \text{ as } k \to \infty \end{aligned} \tag{43}$$

The equations above show that $\hat{\theta} = \theta$ is excluded from $S_N^p(\epsilon)$ in probability which means that the diameter of $S_N^p(\epsilon)$, dia $S_N^p(\epsilon) \to 0$ in probability. If this was not true, then $S_N^p(\epsilon)$ would contain points other than θ for any N with non-zero probability, contradicting the fact that $\hat{\theta} = \theta$ is excluded from $S_N^p(\epsilon)$ in probability. dia $S_N^p(\epsilon)$ converging to zero in probability means that at least one subsequence of $S_N^p(\epsilon)$ converges to zero with probability 1. Because $S_{k+1}^p(\epsilon) \subseteq S_k^p(\epsilon)$ implies that dia $S_N^p(\epsilon)$ is nonincreasing, any subsequence of dia $S_N^p(\epsilon)$ converging to zero with probability 1 implies that dia $S_N^p(\epsilon)$ converges to zero with probability 1 [9].

4.3.1 Rate of Parameter Set Convergence

It has been shown that if the conditions in the previous section are met, the membership set diameter will converge to zero with probability 1. The rate of this convergence to zero, however, is highly dependent on the shape of the distribution of the disturbance, particularly at its bounds. If the distribution for the disturbance has a lot of mass at its end points, i.e. the probability that the disturbance lies on its bound is high, then the convergence rate will be fast. Likewise if the probability that the noise will hit the boundary is low, then the convergence rate is much slower [9].

5 Persistent Excitation

The importance of a persistently exciting regressor was discussed in the previous section along with how it relates to convergence of the parameter set. This section will discuss how the persistent excitation condition is formulated within the context of the formulation discussed in the model identification section from Cannon and Lu's paper [1].

The persistent excitation condition from the previous section can be written similarly as

$$\sum_{k=t-n+1}^{t} D_k^T D_k \succeq n\beta_1^2\mathbb{I}, \tag{44}$$

for a suitable window length n, where $D_k = D(x_k, u_k)$, and $D(x, u)$ is defined in equation 22. The difficulty with this condition lies with the fact that it is generally not convex in x and u, thus a convex sufficient condition must be derived to include this PE condition into the problem formulation in a relatively straightforward way.

5.1 Convex Sufficient Condition for Persistent Excitation

The derivation for a convex sufficient condition for persistent excitation begins by assuming a control input $u^o_{0|t}$ that satisfies the state and input constraints has been obtained at time t. A perturbation δu is applied to $u^o_{0|t}$ in an attempt to make the control input persistently exciting:

$$u_t = u^o_{0|t} + \delta u \tag{45}$$

Thus, the persistent excitation condition can be rewritten in the following manner.

$$\begin{aligned}\sum_{k=t-n+1}^{t} D_k^T D_k &= \sum_{k=t-n+1}^{t-1} D_k^T D_k + D(x_t, u_t)^T D(x_t, u_t) \\ &= \sum_{k=t-n+1}^{t-1} D_k^T D_k + D(x_t, u^o_{0|t} + \delta u)^T D(x_t, u^o_{0|t} + \delta u),\end{aligned} \tag{46}$$

but,

$$\begin{aligned}D(x_t, u^o_{0|t} + \delta u) &= [A_1 x_t + B_1(u^o_{0|t} + \delta u) \ \dots \ A_p x_t + B_p(u^o_{0|t} + \delta u)] \\ &= [A_1 x_t + B_1 u^o_{0|t} \ \dots \ A_p x_t + B_p u^o_{0|t}] + [B_1 \delta u \ \dots \ B_p \delta u] \\ &= D(x_t, u^o_{0|t}) + L(\delta u) \quad \text{where } L(\delta u) = [B_1 \delta u \ \dots \ B_p \delta u].\end{aligned} \tag{47}$$

Substituting back into equation 46:

$$\begin{aligned}\sum_{k=t-n+1}^{t} D_k^T D_k &= \sum_{k=t-n+1}^{t-1} D_k^T D_k + (D(x_t, u^o_{0|t}) + L(\delta u))^T (D(x_t, u^o_{0|t}) + L(\delta u)) \\ &= \sum_{k=t-n+1}^{t-1} D_k^T D_k + D(x_t, u^o_{0|t})^T D(x_t, u^o_{0|t}) + D(x_t, u^o_{0|t})^T L(\delta u) \\ &+ L(\delta u)^T D(x_t, u^o_{0|t}) + L(\delta u)^T L(\delta u)\end{aligned} \tag{48}$$

The persistent excitation condition can thus be reformulated as

$$\sum_{k=t-n+1}^{t} D_k^T D_k = M_t + D_t^T D_t = M_t + L(\delta u)^T D(x_t, u^o_{0|t}) + D(x_t, u^o_{0|t})^T L(\delta u) + L(\delta u)^T L(\delta u) \tag{49}$$

where M_t is independent of δu,

$$M_t = \sum_{k=t-n+1}^{t-1} D_k^T D_k + D(x_t, u^o_{0|t})^T D(x_t, u^o_{0|t}) \tag{50}$$

and $L(\delta u)$ is linear in δu

$$L(\delta u) = [B_1 \delta u \ \dots \ B_p \delta u]. \tag{51}$$

The quadratic term $L(\delta u)^T L(\delta u)$ is positive semi-definite so a sufficient condition for equation 44 to hold is

$$M_t + L(\delta u)^T D(x_t, u^o_{0|t}) + D(x_t, u^o_{0|t})^T L(\delta u) \succeq n\beta_1^2 \mathbb{I}. \tag{52}$$

Equation 52 is a linear matrix inequality constraint on δu which is convex in δu and guarantees that the PE condition 44 is met with a degree of conservativeness based on δu as $O(||\delta u||^2)$.

The only decision variable in equation 52 is the disturbance on the input δu at the current time step, so there is no guarantee that there will be a feasible solution for δu that satisfies the state and input constraints as well as equation 52. However, this may not necessarily cause a problem as the PE condition does not need to hold for every time step to still guarantee the convergence of the parameter set diameter to zero.

6 Implementation

Cannon and Lu combine the theory discussed in several of the previous sections to formulate an adaptive MPC algorithm that has impressive performance properties. They make use of the algorithm described in the tube based MPC algorithm section along with using a set based model identification technique for parameter estimation, and an implementation of the persistent excitation condition. Their proposed algorithm is as follows:

Algorithm 1: Robust Adaptive MPC:

Offline, for an initial parameter set estimate Θ_0, choose V and compute the following values:

K, λ from equation 19

$H_c, \hat{H}_1, ..., \hat{H}_{n_\alpha}$ from equations 9 and 12

Online at time $t \in \mathbb{N}_{\geq 0}$ given a state measurement x_t:

1) Update Θ_t using the parameter set identification algorithm 24 and compute the vertices of Θ_t as $\theta_t^{(1)}, ..., \theta_t^{(m)}$.

2) Update λ_t by solving

$$\lambda_t = \max_{j,i} \hat{\theta}_t^{(j)T} \hat{H}_i \mathbf{1}. \tag{53}$$

3) Compute the optimal solution: $\boldsymbol{v}_t, \boldsymbol{\alpha}_t$ of the LP

$$\begin{gathered} J^o(x_t) = \min_{\boldsymbol{v},\boldsymbol{\alpha}} \sum_{k=0}^{N-1} J(x_t, u_t) \\ \text{s.t. equations 8, 11, 16, 17, 18.} \end{gathered} \tag{54}$$

4) Calculate the control law $u^o_{0|t} = Kx_t + v_{0|t}$ and check whether the PE condition 44 is satisfied with $D_t = D(x_t, u^o_{0|t})$.

(a) If condition 44 is satisfied, implement $u_t = u^o_{0|t}$

(b) Otherwise, recompute the optimization 54 with an additional optimization variable δu and with the convex PE condition 52 derived in the previous section.

(i) If the optimization is infeasible, implement $u_t = u^o_{0|t}$.

(ii) Otherwise implement $u_t = u^o_{0|t} + \delta u$.

The control input serves the dual purpose of stabilising the system and providing the requisite excitation so that the model parameters can be identified. These may at times be conflicting constraints on the control input. In cases where it is not possible to satisfy the system constraints and the PE constraints, the algorithm is designed for system constraint satisfaction to take precedence over satisfaction of the PE constraints [1].

This algorithm applied to a numerical example with a second-order discrete-time uncertain linear system highlights the advantages of the algorithm that Cannon and Lu have proposed. The specific numerical example that the algorithm is applied to is taken from a paper by Lorenzen et al. [10]. The model parameters are given by

$$\begin{aligned} A_0 &= \begin{bmatrix} 0.5 & 0.2 \\ -0.1 & 0.6 \end{bmatrix} & B_0 &= \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \\ A_1 &= \begin{bmatrix} 0.042 & 0 \\ 0.072 & 0.03 \end{bmatrix} & B_1 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ A_2 &= \begin{bmatrix} 0.015 & 0.019 \\ 0.009 & 0.035 \end{bmatrix} & B_2 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ A_3 &= \begin{bmatrix} 0 & 0 \\ -0 & 0 \end{bmatrix} & B_3 &= \begin{bmatrix} 0.0397 \\ 0.059 \end{bmatrix} \end{aligned} \tag{55}$$

and the true system parameter is $\theta^* = [0.8 \;\; 0.2 \;\; -0.5]^T$. The initial parameter set estimate is given by the hypercube centred on the origin with side length two ($\Theta_0 = \{\theta : ||\theta||_\infty \leq 1\}$) and the elements of the disturbance sequence are independently and identically distributed on $\mathbb{W} = \{w \in \mathbb{R}^2 : ||w||_\infty \leq 0.1\}$. The state and input constraints are $[x_t]_2 \geq -0.3$, and $u_t \leq 1$.

The cross-sections of the robust state tubes predicted at time $t = 0$ resulting from the robust adaptive MPC algorithm are shown below in Figure 1. The state constraint $[x_t]_2 \geq -0.3$ is represented by the dashed black line in Figure 1 and it can be seen that the predicted state tubes never violate this constraint. This means that the state constraints are satisfied for all realisations of uncertainty and disturbance. Furthermore, it can be seen from the dashed pink line in the figure that the evolution of states converges to be in a neighbourhood close to 0, implying stability of the system.

It can be seen from examining plots of the parameter set over time that the proposed algorithm also leads to the convergence of the parameter set to a singleton at the true parameter value over the course of a sufficiently long period of time. Figure 2 shows that the reduction over time of the size of the parameter set is substantial. The volume of the parameter set is nearly quartered after the first time step, and the volume after the 10th time step is merely 13 percent of the original volume. After 5000 time steps, the volume of the parameter set is 0.0089 percent of the original volume, meaning nearly all of the model uncertainty has been eliminated. At this point, the parameter set is essentially a point at the true parameter value.

It is important to highlight that the decreasing uncertainty of the true parameter values leads to a tangible improvement in performance; The decrease in size of Θ_t leads directly to a decrease in the optimal value of the MPC cost for a given value of x_t and the same set of constraints. For the numerical example cited in Cannon and Lu's paper, the optimal predicted cost given the initial

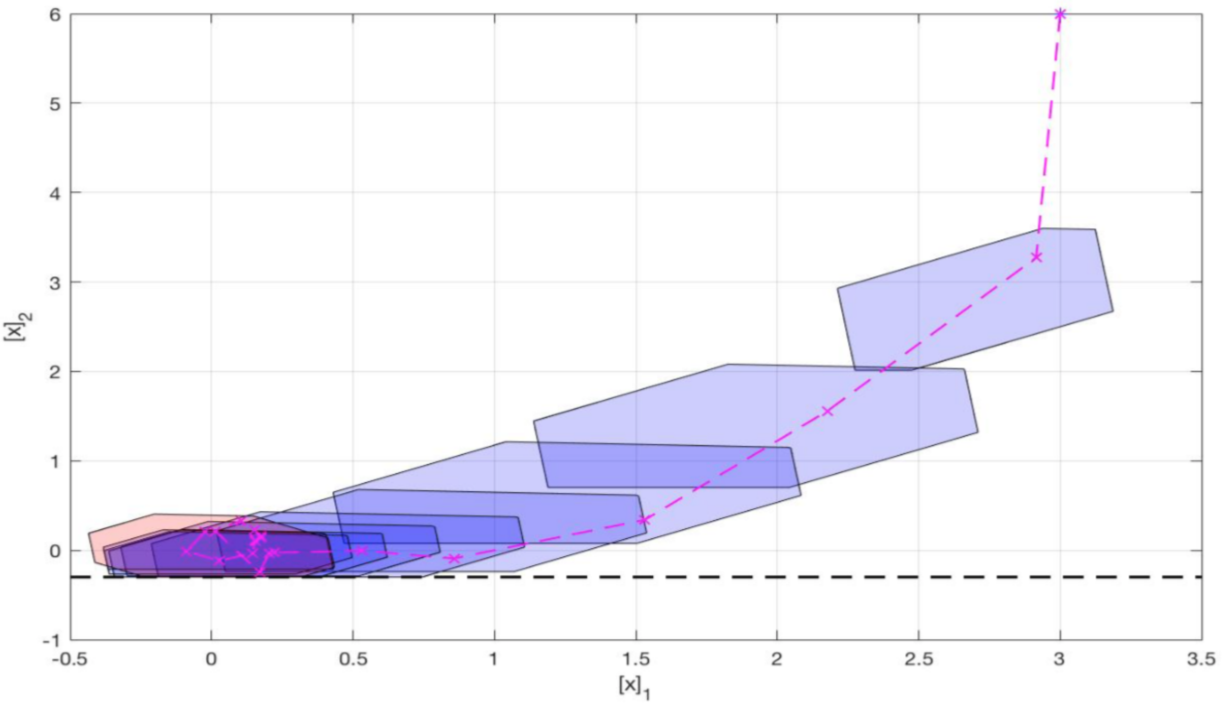

Figure 1: Closed loop state trajectory from initial condition $x_0 = (3, 6)$, and predicted state tube (polygons) $\{\mathbb{X}_0, ..., \mathbb{X}_N\}$ at time $t = 0$ [1]

parameter set estimate Θ_0 gives $J^o(x) = 62.2$ for $x = x_0 = (3, 6)$, while Θ_500 gives $J^o(x) = 57.9$, a 6.9% reduction versus the initial parameter set, and Θ_5000 gives $J^o = 53.9$, a 13.3% reduction. To compare, no parameter uncertainty gives a predicted optimal cost of $J^o(x) = 52.70$, which is a 15.3% reduction to the initial predicted optimal cost with the initial estimate for the parameter set. The original implementation of the author in this report will also consider the performance of the proposed algorithm on the same numerical example to aide the understanding of the differing performance of the algorithm outline in previous section, and the new algorithm proposed by the author in the coming sections.

6.1 H-form Implementation

The author's original implementation follows closely from the theory set out in the previous sections and the proposed algorithm from above. An important deviation between the two implementations is the choice of objective function. The author has used a nominal cost function of the form

$$J(x_k, v_k) = \boldsymbol{v}_k^T H \boldsymbol{v}_k + 2\boldsymbol{f}^T \boldsymbol{v}_k \tag{56}$$

as opposed to an expected value cost $(J(x, u) = \sum_{i=0}^{\infty} \mathbb{E}(x_{i|k}^T Q x_{i|k} + u_{i|k}^T R u_{i|k})$ or a piecewise-linear worst case cost $\max_{x_k \in \mathbb{X}, k=0,1,...} \sum_{k=0}^{\infty} (\max\{Qx_k\} + \max\{Ru_k\})$. Several factors contributed to the

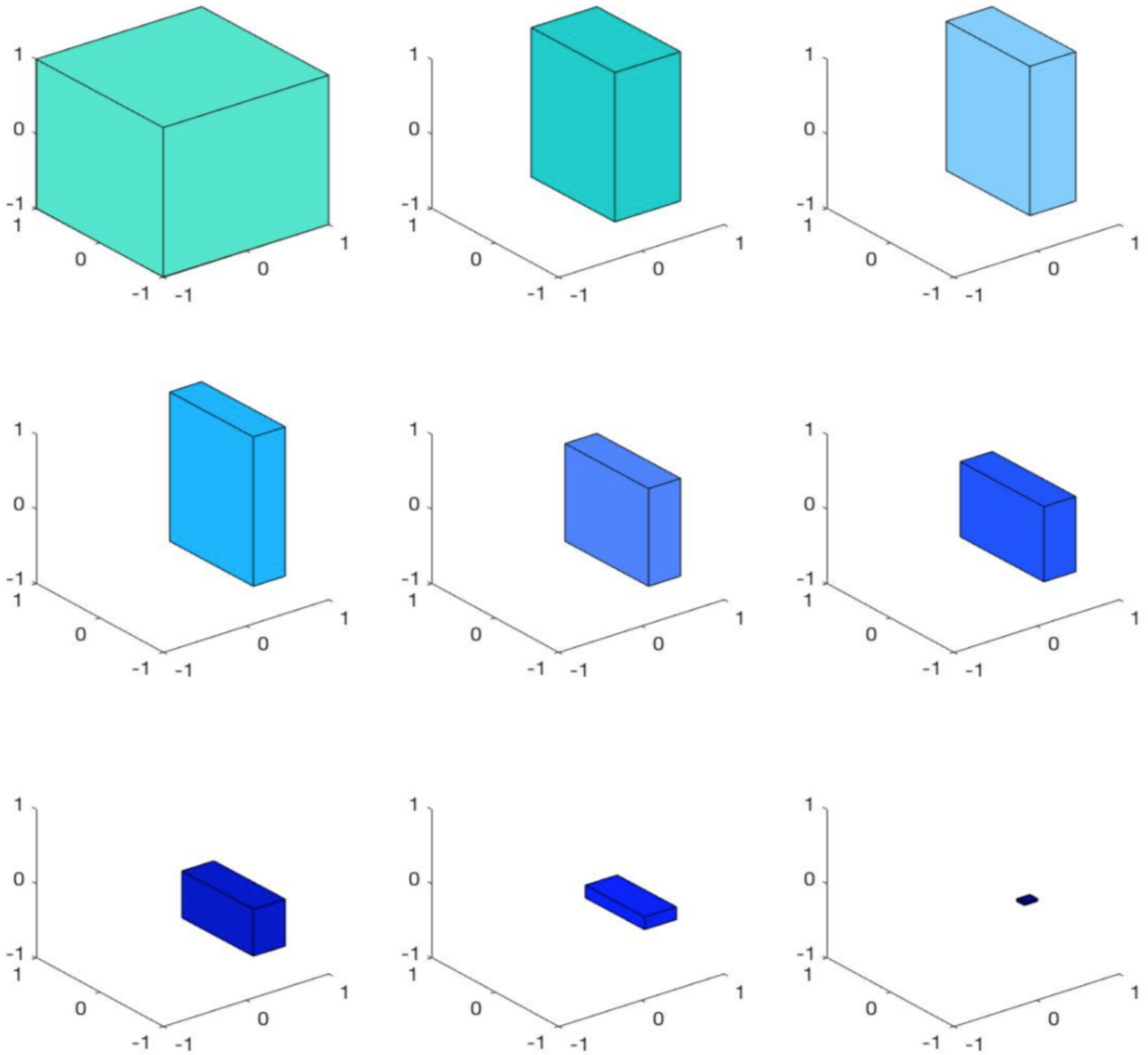

Figure 2: Parameter set Θ_t at time steps $t = 0, 1, 2$ (1st row) $t = 10, 25, 50$ (2nd row) and $t = 100, 500, 5000$ (3rd row) [1]

choice of a nominal quadratic cost as opposed to another cost function: firstly, the piecewise-linear min-max cost is very conservative when the parameter set if big (initially), where as the nominal cost based on a point estimate is only as inaccurate as the point estimate itself. Furthermore, it is not straightforward to choose the weights of the min-max cost function, whereas it is simple to choose the weights of the quadratic nominal cost based on the theory of LQR controllers. Finally, the quadratic cost decouples the constraints from the cost calculation, which allows for more freedom in the way that the constraints are defined.

In linear systems such as the numerical example that will be considered, the dependence of the state predictions $\boldsymbol{x}_k$ on the predicted inputs $\boldsymbol{u}_k$ is linear. Thus the general quadratic predicted cost $J(x_k, u_k) = \sum_{i=0}^{N}(||x_{i|k}||_Q^2 + ||u_{i|k}||_R^2)$ can be represented as a quadratic function of the input sequence

$\boldsymbol{u}_k$. The cost can thus be represented as a function of $\boldsymbol{u}$:

$$J(x_k, u_k) = \boldsymbol{u}_k^T H \boldsymbol{u}_k + 2f^T \boldsymbol{u}_k + g \tag{57}$$

H is a constant positive (semi)definite matrix, $f = f(x_k)$ is a vector depending on x_k, and $g = g(x_k)$ is a scalar depending on x_k. The online optimisation thus becomes:

$$\begin{gathered} \min_u \boldsymbol{u}^T H \boldsymbol{u} + 2f^T \boldsymbol{u} \\ \text{subject to constraints} \end{gathered} \tag{58}$$

It is easy to show that this problem is a convex optimisation problem so long as the constraints are convex due to the restriction that H is a positive (semi)definite matrix.

The general quadratic cost over an infinite horizon $J(x_k, \boldsymbol{u}_k) = \sum_{i=0}^{N-1}(||x_{i|k}||_Q^2 + ||u_{i|k}||_R^2) + ||x_{N|k}||_{\bar{Q}}^2$ can be written in the from of equation 57. To do so, begin by writing the predicted states in terms of the current state and predicted inputs:

$$\begin{aligned} x_{0|k} &= x_k \\ x_{1|k} &= Ax_k + Bu_{0|k} \\ x_{2|k} &= A^2 x_k + ABu_{0|k} + Bu(k+1|k) \\ &\vdots \\ x_{N|k} &= A^N x_k + A^{N-1}Bu_{0|k} + A^{N-2}Bu(k+1|k) + \ \dots \ + Bu(k+N-1|k) \end{aligned} \tag{59}$$

This can be written in a compact way

$$\begin{gathered} x_{i|k} = A^i x_k + C_i \boldsymbol{u}_k, \quad i = 0, ..., N \\ \text{or } \ \boldsymbol{x}_k = Mx_k + C\boldsymbol{u}_k, \ \text{ where } \ M = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}. \end{gathered} \tag{60}$$

Consequently, the C matrix is defined by

$$\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} \tag{61}$$

In the sections above, the optimal control input $u_{0|k}$ has been defined by $u_{0|k} = Kx_k + v_{0|k}$, so the above equations can be rewritten as

$$x_{i|k} = \Phi^i x_k + C_i \boldsymbol{v}_k, \quad i = 0, ..., N$$
$$\text{or} \quad \boldsymbol{x}_k = Mx_k + C\boldsymbol{v}_k, \quad \text{where} \quad M = \begin{bmatrix} \Phi \\ \Phi^2 \\ \vdots \\ \Phi^N \end{bmatrix}, \tag{62}$$

where Φ is defined as $\Phi = A + BK$, and the C matrix is defined in the same way as above, but A is replaced with Φ. Substituting for $x_{i|k}$ into equation 57 yields:

$$J_k = x_k^T M^T \bar{Q} M x_k + 2x_k^T M^T \bar{Q} C v_k + v_k^T (C^T \bar{Q} C + \bar{R}) v_k \tag{63}$$

where,

$$\bar{Q} = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & Q & 0 \\ 0 & \dots & 0 & P \end{bmatrix}$$
$$\bar{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix} \tag{64}$$

and P is the solution to the Lyapunov equation $P - (A + BK)^T P(A + BK) = Q + K^T RK$. For the numerical example considered by the author, Q was taken to be an identity matrix of size two, and $R = 1$. Equation 63 can be rewritten as $J(x_k, u_k) = x_k^T M^T \bar{Q} M x_k + 2f^T v_k + v_k^T H v_k$ where $H = C^T \bar{Q} C + \bar{R}$ and $f^T = x_k^T M^T \bar{Q} C$. The first term is a constant, and thus for the purposes of the online optimisation, the cost function can be rewritten as $J(x_k, u_k) = 2f^T v_k + v_k^T H v_k$ which is identical to the form in equation 56 that was stated in the beginning of this section [3].

The author's H-form adaptive MPC algorithm is detailed below:

Offline:

1) Set the number of time steps the simulation will run for

2) Set prediction horizon

3) Define system matrices

4) Define set that the disturbance lies in, and define initial parameter set estimate Θ_0

5) Define initial condition and true model parameter value

6) Define matrices F and G that formulate state and input constraints

7) Generate V matrix and calculate stabilising gain K from V

8) Calculate H_c and $\hat{H}_i$ from the LPs in equations 9 and 12 respectively

9) Define Q and R for use in constructing the cost function

10) Compute μ for use in calculating point estimates of the model parameters

Online:

1) Do parameter set update

2) Compute vertices of new parameter set

3) Compute radial size of the new parameter set

4) Calculate a point estimate according to equation 25

5) Optimise the cost function to compute the optimal control input for next time step

   a) Construct cost matrices

   b) Formulate cost function in terms of cost matrices

   c) Set constraints according to equation 54

   d) Optimise cost function and return optimal control input

6) Simulate the evolution of the system to its next state given the calculated control input

7) Repeat until previously specified number of time steps has been simulated

The prediction horizon was set to 10 time steps, and the system matrices are as defined above in equation 55. The initial parameter set was defined to be a hypercube centred on the origin with side length two. The complexity of the parameter set was restricted to be a hypercube so that the calculation of the parameter sets vertices would be straightforward and computationally light. The parameter set is defined as $\Theta_k = \{\theta : [I \quad -I]^T\theta \leq \pi_{\theta,k}\}$. Note that computing the vertices of a hypercube with given side lengths, centred on the origin is trivial; the method the author has used for calculating the vertices of the parameter set first translates the parameter set so that it is centred on the origin, computes the vertices (trivially), then translates these vertices to be centred on the centre of the parameter set. To do this, $\pi_{\theta,k}$ is represented as $[\bar{\pi} \quad \underline{\pi}]^T$. The centre of the parameter set θ_0 is defined as $\frac{1}{2}(\bar{\pi} - \underline{\pi})$. The parameter set, shifted to be centred on the origin, is now defined as

$$\Theta_c = \{\theta : \begin{bmatrix} I \\ -I \end{bmatrix} (\theta - \theta_0) \leq \begin{bmatrix} \bar{\pi} - \theta_0 \\ \underline{\pi} + \theta_0 \end{bmatrix}\}. \tag{65}$$

Substituting the value for θ_0 in equation 65 results in

$$\Theta_c = \{\theta : \begin{bmatrix} I \\ -I \end{bmatrix} (\theta - \theta_0) \leq \begin{bmatrix} \frac{1}{2}(\bar{\pi} + \underline{\pi}) \\ \frac{1}{2}(\bar{\pi} + \underline{\pi}) \end{bmatrix} \}, \tag{66}$$

which is of the form

$$\Theta = \{\theta : \begin{bmatrix} I \\ -I \end{bmatrix} (\theta - \theta_0) \leq \begin{bmatrix} \pi \\ \pi \end{bmatrix} \}, \tag{67}$$

meaning that the set is now centred on the origin. The vertices of a set that has the form of equation 67 is found by performing an element wise multiplication of π with each column of the matrix Z defined below

$$Z = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{68}$$

The columns of Z give all the possible positive and negative permutations of the elements of π which corresponds to the vertices of the set. Having found the vertices of the centred parameter set, the only thing left to do is to add the centre coordinates back to each vertex, resulting in the vertices of the original parameter set. Computation of the diameter of the parameter set is done in a similar way. Bai et al. define the diameter of the membership set to be

$$\text{dia } S_N(\epsilon) = \sup_{\theta_1, \theta_2 \in S_N(\epsilon)} ||\theta_1 - \theta_2||_2. \tag{69}$$

This is equivalent to the following optimisation

$$\rho(\Theta_k) = \max_i |\bar{\pi}_i - \underline{\pi}_i| \tag{70}$$

where $\bar{\pi}$ and $\underline{\pi}$ have the same meaning as in the equations above.

Implementing this algorithm in Matlab, the author was able to produce results similar to those published by Cannon and Lu in [1]. Figure 3 shows the state evolution of the system over thirty time steps using the detailed algorithm. As can be seen from the figure, the states converge to a region around the origin quickly and never violate the state constraint that is plotted in light blue.

6.2 V-form Implementation

Having recreated the results of Cannon and Lu in [1], the author has considered a tractable conversion between H-form representation and V-form representation. H-form, short for half-space representation,

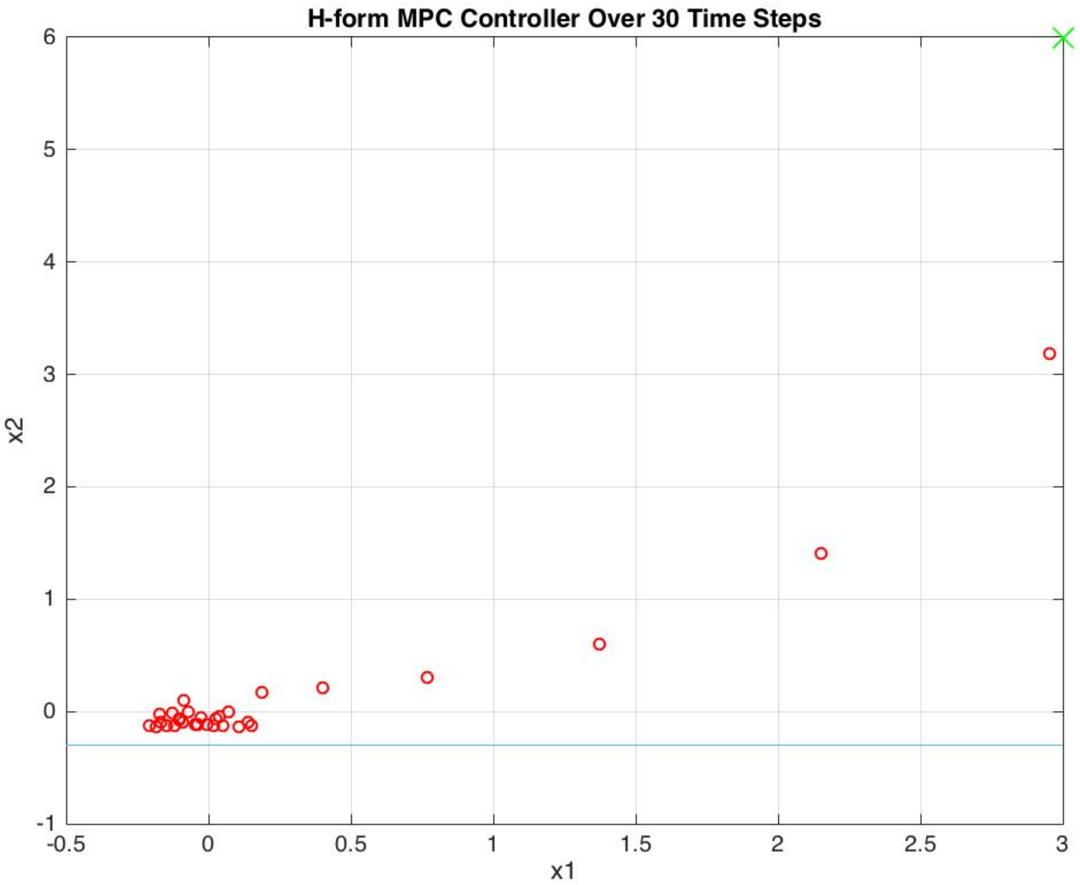

Figure 3: Evolution of system state over 30 time steps

of sets are the representation of sets in the manner that this report has considered until this point; a H-form set is defined as $S = \{s : As \leq b\}$. By Motzkin's theorem, for each H-form polytope, the convex hull of its vertices defines the same sets in the form of a V-polytope, and vice versa [11].

The H-form algorithm discussed in the previous sections lends itself well to being converted to a V-form representation. To do so, the constraints must be formulated in terms of the vertices of the predicted state tube cross sections.

If $\mathcal{X}_{i|k} \subset \mathbb{R}^n$ is used to denote a compact polytope defining the i steps ahead cross section of the predicted state tube at time k,

$$\mathcal{X}_{i|k} = \{x : Vx \leq \alpha_{i|k}\}, \tag{71}$$

then $\mathcal{X}_{i|k}$ can be represented equivalently with the vertex representation as the convex hull of the vertices of the cross section of the predicted state tube:

$$\mathcal{X}_{i|k} = \text{Co}\{x^j_{i|k},\ j = 1, ..., m\} \tag{72}$$

$x^j_{i|k}$ represents a single vertex of the cross section of the predicted state tube. The equivalency of the above representations is straightforward conceptually. In equation 71, the predicted state tube is represented as the space enclosed by the intersection of a set of half-spaces, where as equation 72 represents the same space as the convex hull of the vertices of the space. Let e_r be the rth column of the identity matrix in $\mathbb{R}^{n_\alpha x n_\alpha}$ where n_α is the number of rows of the V matrix from previous sections. For each vertex of $\mathcal{X}_{i|k}$ (for each $j \in \{1, ..., m\}$), the rth row of V, $e_r^T V$ and the rth element of $\alpha_{i|k}$, $e_r^T \alpha_{i|k}$ satisfy

$$e_r^T V x_{i|k}^j = e_r^T \alpha_{i|k} \quad \forall \ r \in \mathcal{R}_j \tag{73}$$

for some index set $\mathcal{R}_j$ with $|\mathcal{R}_j| = n$. Importantly, $\mathcal{R}_j$ can be computed offline as it is independent of $\alpha_{i|k}$. This is true because the allowable values of $\alpha_{i|k}$ are such that each hyperplane in the description of $\mathcal{X}_{i|k}$, i.e. the hyperplanes described by each row of V, has a non-empty intersection with the boundary of $\mathcal{X}_{i|k}$. Thus, for each $j \in \{1, ..., m\}$ the following relation holds.

$$x_{i|k}^j = U^j \alpha_{i|k} \tag{74}$$

U^j is a matrix of size $n \ x \ n_\alpha$ and is computed offline given knowledge of $\mathcal{R}_j$. The results stated above can be used to derive equivalent formulations of the H-form constraints in the online optimisation.

The constraint

$$Fx + G(Kx + v_{i|k}) \leq \mathbf{1} \quad \forall x \in \mathcal{X}_{i|k} \tag{75}$$

can be expressed equivalently in terms of the variables $\alpha_{i|k}$ and $v_{i|l}$ by the following expression:

$$(F + GK)U^j \alpha_{i|k} + Gv_{i|k} \leq \mathbf{1} \tag{76}$$

The constraint

$$(A(\theta) + B(\theta)K)x + B(\theta)v_{i|k} + w \in \mathcal{X}_{i+1|k} \ \ \forall w \in \mathcal{W}, \ \forall x \ in \mathcal{X}_{i|k}, \ \forall \theta \in \Theta = \{\theta : H_\theta \theta \leq h_\theta\} \tag{77}$$

can be expressed equivalently as

$$V((A(\theta) + B(\theta)K)U^j \alpha_{i|k} + B(\theta)v_{i|k}) + \bar{w} \leq \alpha_{i+1|k} \quad \forall \theta \in \Theta \tag{78}$$

where $e_l^T \bar{w} = \max_{w \in \mathcal{W}} e_l^T V w$ for $l = 1, ..., n_\alpha$ and hence as

$$VD(U^j \alpha_{i|k}, KU^j \alpha_{i|k} + v_{i|k})\theta + Vd(U^j \alpha_{i|k}, KU^j \alpha_{i|k} + v_{i|k}) + \bar{w} \leq \alpha_{i+1|k} \ \ \forall \theta \in \Theta \tag{79}$$

where $A(\theta)x + B(\theta)u = D(x,u)\theta + d(x,u)$ ($D(x,u)$ and $d(x,u)$ are both linear functions in x and u). Finally, this is implemented in the online optimisation as the set of following constraints,

$$\begin{aligned}
&\Lambda^j_{i|k} H_\theta = VD(U^j\alpha_{i|k}, KU^j\alpha_{i|k} + v_{i|k}) \\
&\Lambda^j_{i|k} \leq \alpha_{i+1|k} - Vd(U^j\alpha_{i|k}, KU^j\alpha_{i|k} + v_{i|k}) - \bar{w} \\
&\Lambda^j_{i|k} \geq 0
\end{aligned} \tag{80}$$

for $i = 1, ..., N-1$ and $j = 1, ..., m$ where N is the prediction horizon length and m is the number of vertices of the cross section of the predicted state tubes. The constraints in equation 80 apply over the length of the prediction horizon; a version of these constraints that form the terminal constraints must also be applied to the online optimisation so that it has an equivalent form to that of the H-form optimisation.

$$\begin{aligned}
&\Lambda^j_{N|k} H_\theta = VD(U^j\alpha_{N|k}, KU^j\alpha_{N|k}) \\
&\Lambda^j_{N|k} \leq \alpha_{N|k} - Vd(U^j\alpha_{N|k}, KU^j\alpha_{N|k}+) - \bar{w} \\
&\Lambda^j_{N|k} \geq 0
\end{aligned} \tag{81}$$

To implement the V-form algorithm, the constraints from the H-form algorithm in the online optimisation are replaced by the constraints in equations 76 and 80.

Having replaced the relevant H-form constraints with their V-form counterparts, the implementation of the author's V-form algorithm is very similar to the H-form algorithm. The algorithm is detailed below:

Offline:

1) Set the number of time steps the simulation will run for
2) Set prediction horizon
3) Define system matrices
4) Define set that the disturbance lies in, and define initial parameter set estimate Θ_0
5) Define initial condition and true model parameter value
6) Define matrices F and G that formulate state and input constraints
7) Generate V matrix and compute redundant rows of V and remove them
8) Compute stabilising gain K from V matrix
9) Define Q and R for use in constructing the cost function
10) Compute μ for use in calculating point estimates of the model parameters

Online:

1) Do parameter set update

2) Compute vertices of new parameter set

3) Compute radial size of the new parameter set

4) Calculate a point estimate according to equation 25

5) Optimise the cost function to compute the optimal control input for next time step

   a) Construct cost matrices

   b) Formulate cost function in terms of cost matrices

   c) Set constraints according to equations 76, 80, and 81

   d) Add additional constraint for current time step PE condition

   e) Optimise cost function and return optimal control input

6) Simulate the evolution of the system to its next state given the calculated control input

7) Repeat until previously specified number of time steps has been simulated

The ease of implementation of the persistent excitation condition was the main motivation to switch from H-form implementation to V-form implementation. In V-form, implementing the PE condition is simply a matter of including an extra constraint in the online optimisation of the form

$$M_0 + D(x_t, u^*_{0|t})^T D(x_t, u^*_{0|t}) + D(x_t, u^*_{0|t})^T \begin{bmatrix} B_1\delta u & \dots & B_v \delta u \end{bmatrix} + \begin{bmatrix} B_1 \delta u \\ \vdots \\ B_v \delta u \end{bmatrix} D(x_t, u^*_{0|t}) \geq \beta^2 I, \tag{82}$$

where

$$M_0 = \sum_{k=t-n+1}^{t-1} D(x_k, u_k)^T D(x_k, u_k) + D(x_t, u^o_{0|t})^T D(x_t, u^o_{0|t}), \tag{83}$$

and the actual control input that is used for the system is given by

$$u_{0|k} = u^*_{0|k} + \delta u. \tag{84}$$

The cost function chosen in this algorithm is the same as in the H-form algorithm plus another term $-\lambda\beta^2$, so that the total cost function is given by

$$J(x_k, u_k) = 2f^T v_k + v_k^T H v_k - \lambda \beta^2 \tag{85}$$

and the derivation is done in the same way. The weighting λ in the second term gives the user a way to denote the relative importance that the controller places on stabilising the system versus having a high degree of persistent excitation. Not surprisingly, this version of the author's adaptive MPC algorithm has performance that is comparable with the H-form algorithm, however with slightly faster parameter set convergence due to the inclusion of the persistent excitation condition in this implementations. Figure 4 shows the evolution of the states with the V-form algorithm controller. As was the case in the H-form controller, it can be seen that the system quickly converges to a small region around the origin, and that the state constraints are satisfied for all time steps. Although the control inputs are not plotted on the graph, they also do satisfy the input constraints of the system at all time steps.

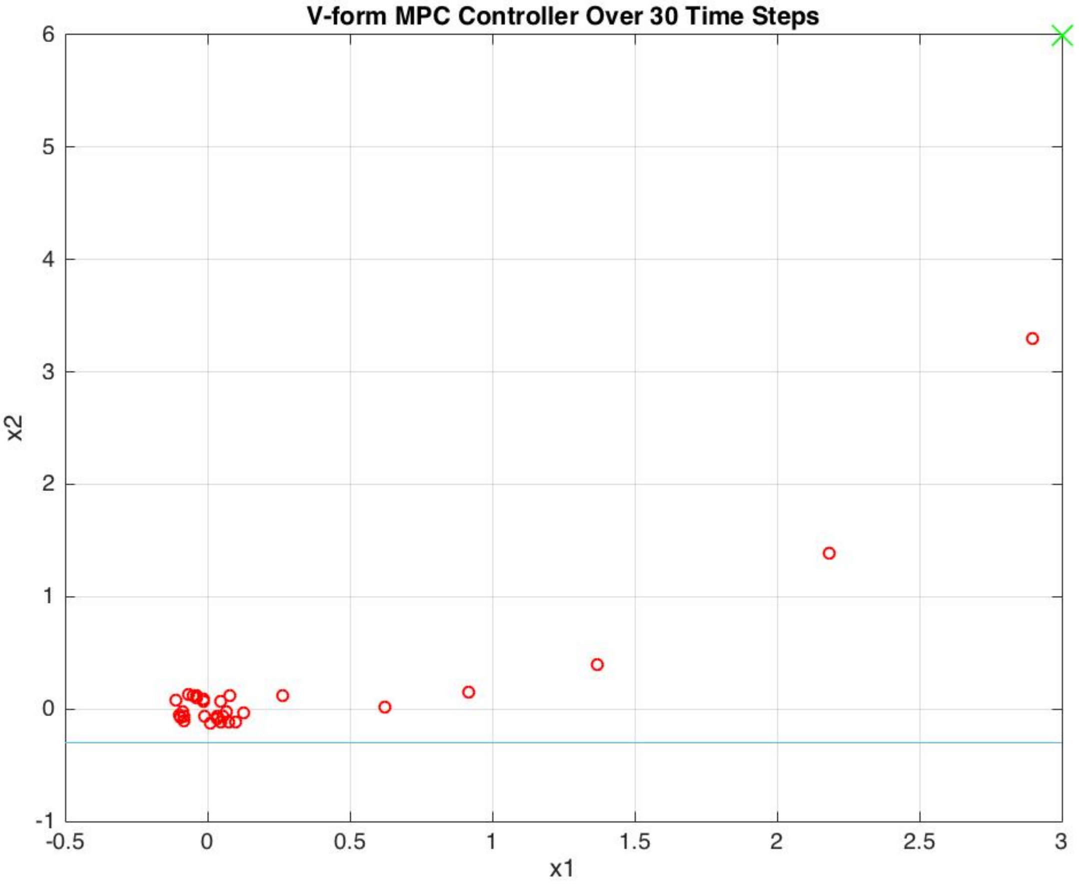

Figure 4: Evolution of system state over 30 time steps

6.2.1 Calculation of $\mathcal{R}_j$ and U^j

Allowing $\mathcal{X}(\alpha)$ to denote the subset of $\mathbb{R}^n$ defined by

$$\mathcal{X}(\alpha) = \{x : Vx \leq \alpha\} \tag{86}$$

for $\alpha \in \mathbb{R}^{n_\alpha}$ and $V \in \mathbb{R}^{n_\alpha \times n_x}$, let the vertices of $\mathcal{X}(\alpha)$ be denoted by $x^j(\alpha)$ for $j = 1, ..., m$ such that

$$\mathcal{X}(\alpha) = \text{Co}\{x^j(\alpha), j = 1, ..., m\}. \tag{87}$$

Using this notation, $\mathcal{X}_{i|k}$ will denote $\mathcal{X}(\alpha_{i|k})$ and $x^j_{i|k} = x^j(\alpha_{i|k})$.

Assume that for a given vector α (e.g vector of all ones), the polytope $\mathcal{X}(\alpha)$ has the property that for each row of V, the hyperplane described by $\{x : e_r^T V x = e_r^T \alpha\}$ is a non-redundant face of $\mathcal{X}(\alpha)$, meaning that it intersects the boundary of $\mathcal{X}(\alpha)$ at more than a single point (e_r^T represents the rth column of the identity matrix in $\mathbb{R}^{n_\alpha \times n_\alpha}$). From simple geometric arguments and linear programming duality, the conditions below provide a method of checking whether this property is satisfied, and using it to compute $\mathcal{R}_j$ and U^j.

The following conditions are equivalent:

(a). For each $r = 1, ..., n_\alpha$, the intersection of the hyperplane $H_r(\alpha) = \{x : e_r^T V x = e_r^T \alpha\}$ with $\mathcal{X}(\alpha)$ is non-empty and consists of more than a single point

(b). For each $r = 1, ..., n_\alpha$, $\mathcal{X}(\alpha) \subset \mathcal{X}(\alpha + e_r \epsilon)$ for all $\epsilon > 0$.

(c). Each vertex $x^j(\alpha)$ lies at the intersection of exactly n hyperplanes: $H_{r_1^j}(\alpha) \cap H_{r_2^j}(\alpha) \cap \ldots \cap H_{r_n^j}(\alpha) = \{x^j(\alpha)\}$ for some pairwise distinct set of indices $R_j = \{r_1^j, ..., r_n^j\} \subset \{1, ..., n_\alpha\}$.

By the equivalence of conditions (a) and (b), one can check whether condition (a) is satisfied by solving the linear programs in equation 88 for $r = 1, ..., n_\alpha$. Condition (b) holds if and only if $\bar{\alpha}_r > e_r^T \alpha$ for all $r = 1, ..., n_\alpha$. It will always be the case that the allowable values for $\alpha_{i|k}$ are such that each hyperplane in the description of $\mathcal{X}_{i|k}$ has a non-empty intersection with the boundary of $\mathcal{X}_{i|k}$ because otherwise $\alpha_{i|k}$ would be a suboptimal solution of the online MPC optimisation. In implementing the linear program from equation 88, the author found that $\mathcal{X}(\alpha)$ had fewer edges than the number of rows of V (i.e $\bar{\alpha}_r \not> e_r^T \alpha$ for all r) for the default value of α (the value of α for which $\mathcal{X}(\alpha)$ was defined to be lambda-contractive), meaning that some rows of V were redundant. The redundant rows of V were the rows r that did not satisfy $\bar{\alpha}_r > \alpha_r$, where $\bar{\alpha}$ calculated for each r by solving the linear program given in equation 88. Thus, one is able to remove the redundant rows of V without affecting $\mathcal{X}(\alpha)$, which has the benefits of satisfying the three conditions above and reducing the computation required in the online MPC optimisation without affecting the performance or feasibility properties of the MPC algorithm.

$$\begin{aligned} \bar{\alpha}_r &= \max_x e_r^T V x \\ \text{subject to } & \begin{bmatrix} e_1^T \\ \vdots \\ e_{r-1}^T \\ e_{r+1}^T \\ \vdots \\ e_{n_\alpha}^T \end{bmatrix} V x \leq \begin{bmatrix} e_1^T \\ \vdots \\ e_{r-1}^T \\ e_{r+1}^T \\ \vdots \\ e_{n_\alpha}^T \end{bmatrix} \alpha \end{aligned} \tag{88}$$

If condition (a) holds, and if the vertices $x^j(\alpha)$ are known, then $\mathcal{R}_j$ and U^j can be determined for each $j = 1, ..., m$ using condition (c). As a result, $\mathcal{R}_j$ is the index set

$$\mathcal{R}_j = \{r \in \{1, ..., m\} : e_r^T V x^j(\alpha) = e_r^T \alpha \tag{89}$$

and, having determined determined $\mathcal{R}_j = \{r_1^j, ..., r_n^j\}$, we can determine U^j from the following relations:

$$\begin{bmatrix} e_{r_1^j}^T \\ \vdots \\ e_{r_n^j}^T \end{bmatrix} V x^j(\alpha) = \begin{bmatrix} e_{r_1^j}^T \\ \vdots \\ e_{r_n^j}^T \end{bmatrix} \alpha \Rightarrow U^j = \left(\begin{bmatrix} e_{r_1^j}^T \\ \vdots \\ e_{r_n^j}^T \end{bmatrix} V \right)^{-1} \begin{bmatrix} e_{r_1^j}^T \\ \vdots \\ e_{r_n^j}^T \end{bmatrix} \tag{90}$$

It is important to note that the matrix inverse term in the expression given for U^j in equation 90 necessarily has an inverse if condition (c) holds, which necessarily holds if condition (a) or (b) holds, which the author has enforced through removing the redundant rows of the V matrix.

6.3 Comparing H-Form and V-Form Implementations

As mentioned earlier, the performance of the H-form and V-form controllers are comparable. Figures 4 and 3 show that the system converges to a region around the origin in a comparable fashion, and in both controllers the state and control inputs are satisfied for all time steps (and also all predicted time steps). The V-form controller shows slightly quicker convergence of the parameter set, but this is expected because of the implementation of the persistent excitation condition in the V-form controller but not in the H-form controller. Thus, this difference in convergence rate can be explained by the inclusion of the PE condition in the V-form controller, rather than an inherent property of the V-form implementation itself. Figures 5 and 6 show the evolution of the parameter set diameter with each time step for the H-form implementation, and V-form implementation, respectively. As can be seen

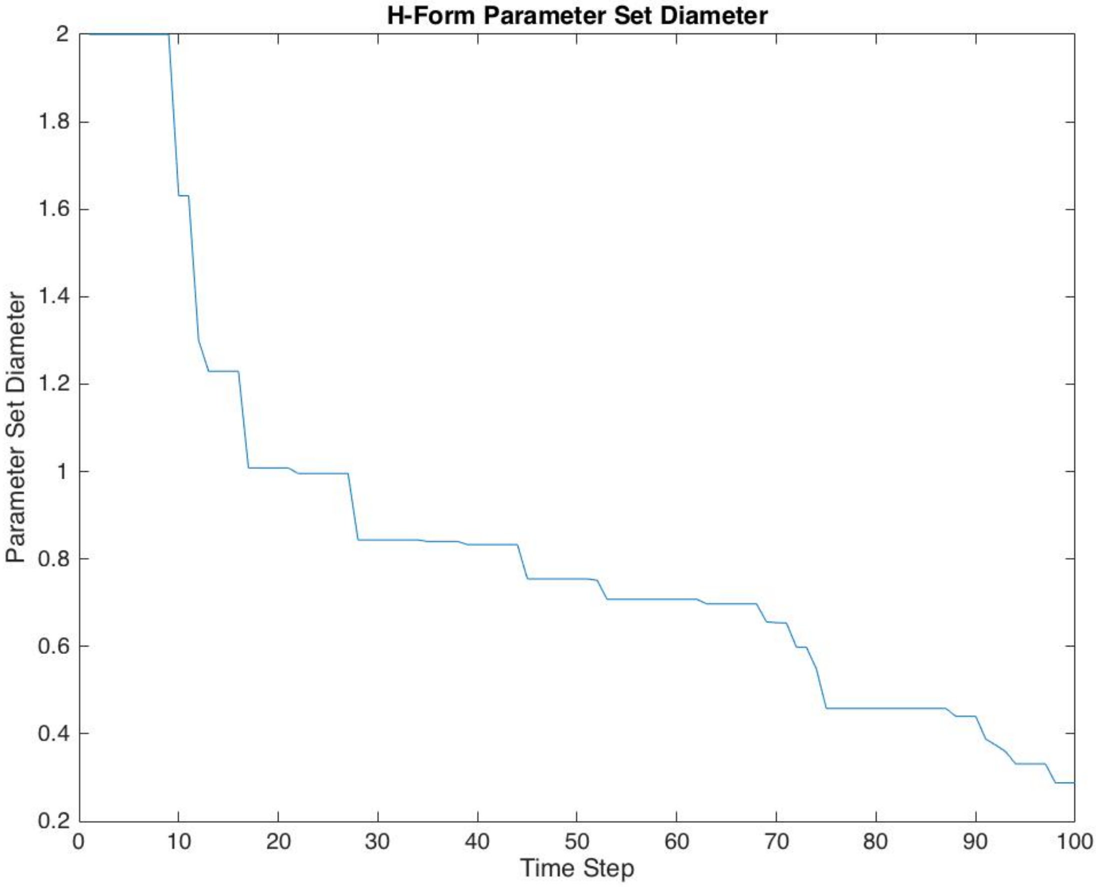

Figure 5: Evolution of parameter set diameter: H-form

from the two figures, the final parameter set diameter after 100 time steps is slightly smaller for the V-form implementation compared to the H-form (approximately 0.2 vs approximately 0.3) due to the inclusion of the persistent excitation condition. As the figures show, the parameter set diameter does not converge to zero linearly or exponentially, or in any orderly manner, making it very difficult to predict how long it will take for the parameter set estimate to converge to a singleton at the true parameter value. It should be noted that the plots of parameter set diameter should not be compared to those in Cannon and Lu's paper [1] as the distribution used for the disturbance differs. Cannon and Lu use a uniformly distributed disturbance, while the author has projected the disturbance to be on the boundary of the disturbance set at each time step in order to speed up the simulation time required to observe convergence of the parameter set. All of the author's original examples and implementations from this point on will consider disturbances that lie on the boundary of the disturbance set.

A good way to measure the performance of the two controllers is to examine the plots of the running sum of the squared states versus time. This plot gives insight into how well the controller is performing, and after an initial section of high increase, the plot will level off and become proportional to the running sum of the square of the disturbance. This follows from the fact that the closed loop system with MPC control law $u(x_k) = Kx_k + v^*_{0|k}$ is finite gain l_2 stable meaning that there exists constants $c_0, c_1, c_2 \in \mathbb{R}_{>0}$ such that for all $K \in \mathbb{N}$

$$\sum_{k=0}^{K} ||x_k||^2 \leq c_0||x_0||^2 + c_1||\hat{\theta}_0 - \theta^*||^2 + c_2 \sum_{k=0}^{K} ||w_k||^2 \qquad [4]. \tag{91}$$

After a sufficient number of time steps, the term $||\hat{\theta}_0 - \theta^*||^2$ approaches zero, meaning that the sum of the magnitude of the states is bounded by the term $c_0||x_0||^2 + c_2 \sum_{k=0}^{K} ||w_k||^2$, which is only dependent on the disturbance and initial condition. Examining the relevant plots for both the H-form and V-form implementations confirm this. As can be seen from Figure 7, the sum of squares of the states rises quickly at the beginning due to the effect of the initial condition, and the intermediate states where the system is approaching the origin. The plot then levels off considerably once the system states converge to a small region around the origin, and the slightly positive slope can then be explained by the $c_2 \sum_{k=0}^{K} ||w_k||^2$ term from equation 91. The same effect can be seen in the V-form controller in Figure 8, with the levelling off effect occurring at a slightly higher value. Because the value of the

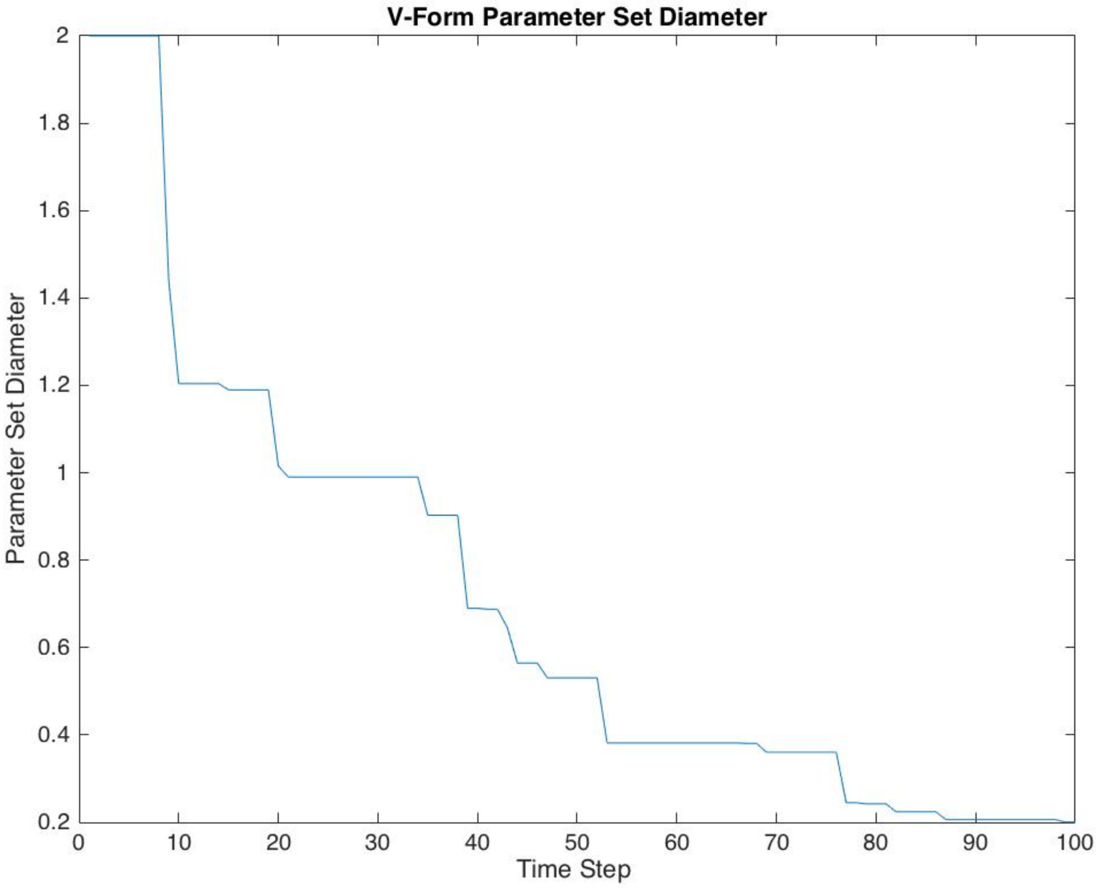

Figure 6: Evolution of parameter set diameter: V-form

sum of the states is partly dependent upon the disturbances, a random variable, the sum of states is also a random variable, thus this slightly different level between the H-form and V-form controllers is not enough to conclude that the H-form implementation has better performance, as running the simulations multiple times yields slightly different results, with the average tailing off value of each controller being roughly equal in the average.

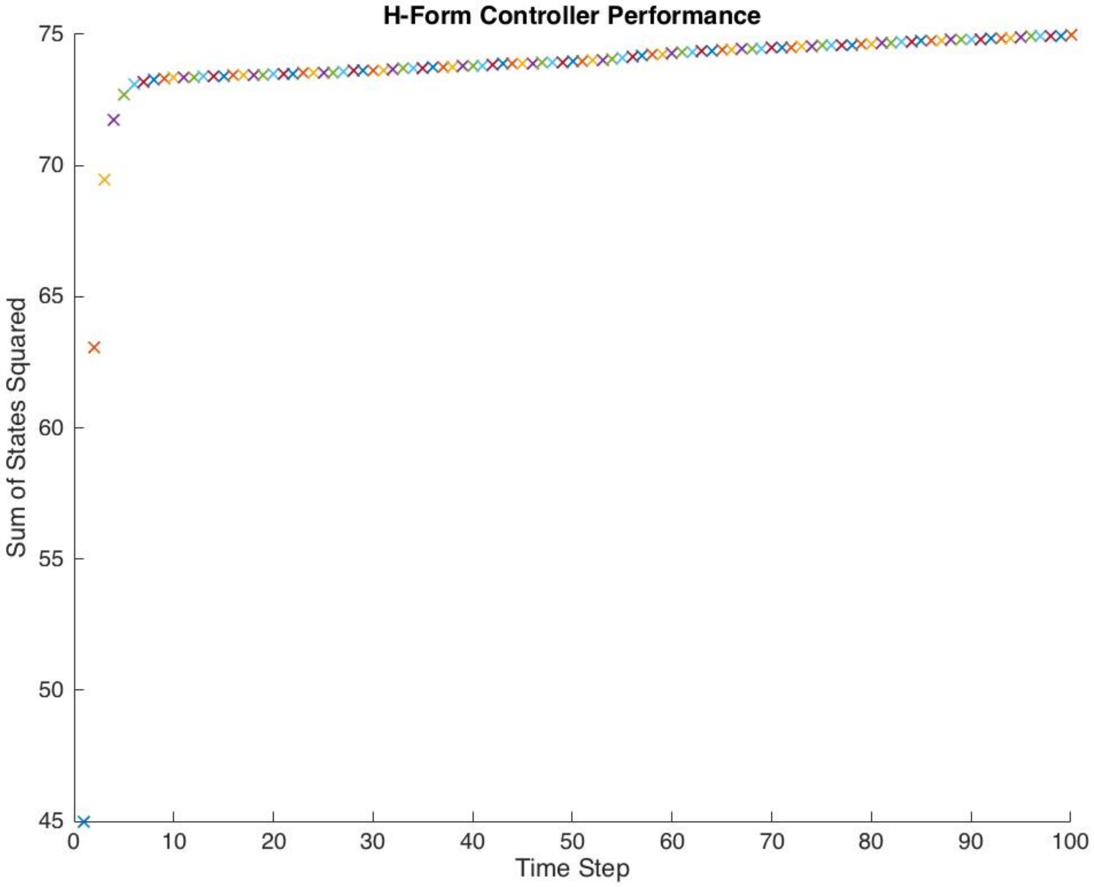

Figure 7: Sum of square of states: H-Form

Both implementations were simulated in Matlab, and both algorithms were programmed with YALMIP and Gurobi. YALMIP (yet another LMI parser) is a tool that can be used to solve optimisations problems frequently occurring in systems and control theory. YALMIP is able to automatically detect what kind of optimisation problem that the user has defined, and then chooses the suitable solver to solve the optimisation problem. If there is not a suitable solver present for the type of optimisation problem defined, YALMIP will attempt to convert the optimisation problem to a form for which an appropriate solver exists [12]. In the H-form implementation, the solver used is Gurobi which can be used for linear programming and quadratic programming applications. The V-form algorithm is slightly more complex in that the online optimisation constraints include linear matrix inequalities (LMIs), which makes the problem a semi-definite programming (SDP) problem, and thus Gurobi cannot be used for the V-form implementation. The SDP solver used in the V-form implementation is MOSEK.

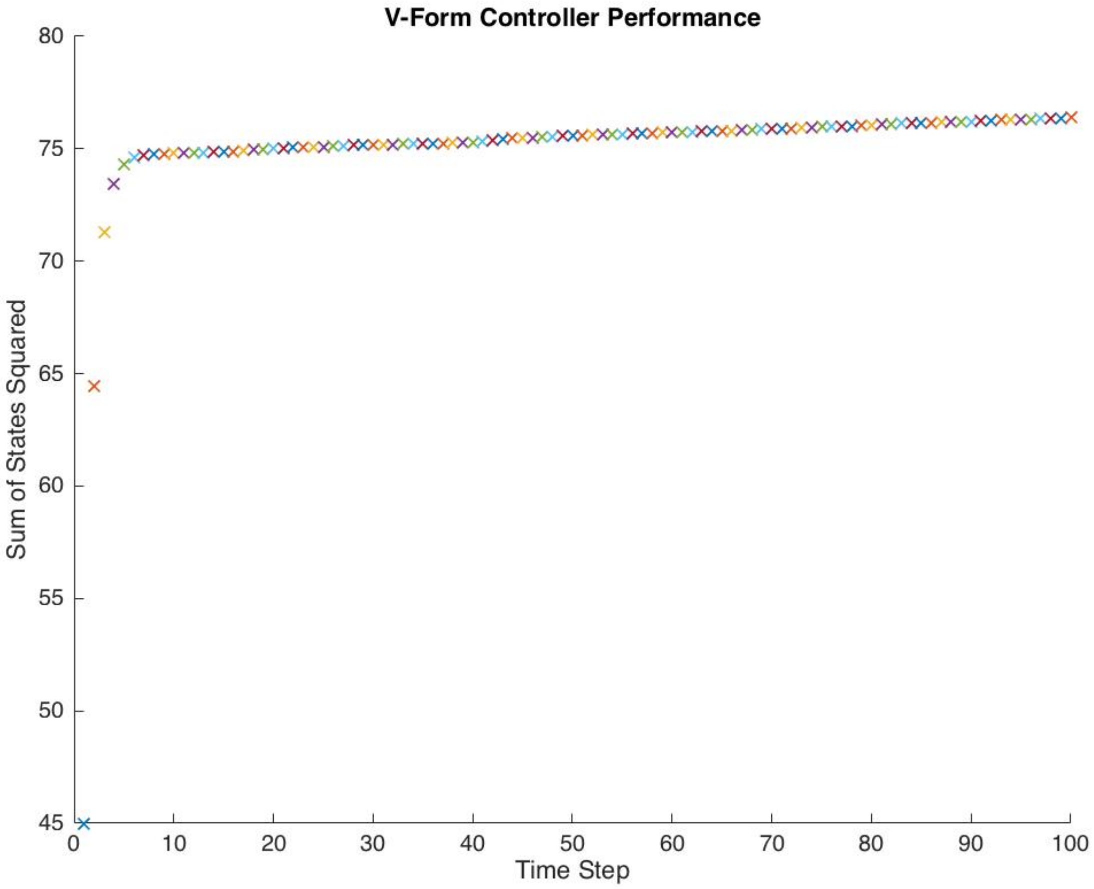

Figure 8: Sum of square of states: V-Form

6.4 Improvements on Existing Algorithms

The algorithms discussed up to this point in this report are either recreations of algorithms existing in the current literature, or recreations of these algorithms with slightly different implementations. The H-form algorithm stated in previous sections is a recreation of the work of Cannon and Lu, and while the equivalent V-form implementation of the algorithm has not been seen in literature, it is the equivalent of algorithms that have been published.

The author hypothesises that implementing a persistent excitation condition that not only accounts for previous and current time steps, but also for future time steps will increase the rate of convergence of the parameter set estimate. This technique has not been seen in the relevant literature, and the author believes it to be a novel contribution to the field. The results of implementing such a condition are discussed and examined in the next section.

Up to this point, all implementations have assumed that the current state is known perfectly, however in reality this will never be the case. Measurement errors on the state can be modelled as disturbances on the true value of the parameter. Thus in the next sections the author will consider a version of the previous algorithm where rather than the true parameter being an unknown constant, the true parameter value at a given time step lies within a bounded set containing the unknown true parameter value, $\theta_k = \theta^* + \tilde{\theta}_k$, where $\tilde{\theta}_k \in S = \{\theta : U\theta \leq h\}$.

6.4.1 Future Persistent Excitation Condition

As mentioned in previous sections, a persistent excitation condition accounting for past and present states is represented by the equation

$$\sum_{k=t-n}^{t} D(x_k, u_k)^T D(x_k, u_k) > \beta^2 I. \tag{92}$$

As mentioned earlier, this is a linear matrix inequality constraints which dictates the use of the solver MOSEK instead of Gurobi.

The idea leading to a persistent excitation condition with future time steps incorporated is a simple extension of the idea in equation 92; one must simply modify the upper limit of the summation to be $t + N$ so that the summation includes future time steps:

$$\text{Future PE Condition:} \quad \sum_{k=t-n}^{t+N-1} D(x_k, u_k)^T D(x_k, u_k) > \beta^2 I \tag{93}$$

Having extended the summation to future time steps, the obvious question that arises is the question of what values to use for x_k and u_k where k is in the future.

To implement this future PE condition, take an 'estimate' of the current state to be the actual value of the current state $\hat{x}_0 = x_t$. Estimates for future states can then be defined recursively by

$$\hat{x}_{k+1} = A(\hat{\theta})\hat{x}_k + B(\hat{\theta})\hat{u}_k, \tag{94}$$

where $x_k \in X_k = \{x : Vx \leq \alpha_k\}$ and $\hat{x}_k \in X_k$. Thus x_k and u_k can be represented as the sum of an estimate term $\hat{x}_k$ and $\hat{u}_k$ respectively, and an error term $\tilde{x}_k$ and $\tilde{u}_k$: $x_k = \hat{x}_k + \tilde{x}_k$ and $u_k = \hat{u}_k + \tilde{u}_k$. Similarly to the estimate of x, the estimate for u is defined by the equation

$$\hat{u}_k = K\hat{x}_k + \hat{v}_k, \tag{95}$$

where the estimate for v, $\hat{v}$, is the optimal predicted sequence for v computed at the last time step, shifted left by one: $\hat{v}_k = v^*_{k+1}$. Having defined the estimates of x, u, and v in this way, the values of

x_k and u_k to be used in future time steps of the PE condition are now fully defined. The reason for the future PE condition only being able to stretch $N-1$ time steps into the future as opposed to N steps into the future is due to the definition of $\hat{v}$. Because the optimal sequence for v calculated in the online optimisation is N elements long, where N is the prediction horizon length, taking the previously computed optimal sequence for v and shifting each element to the left by one step gets rid of the first element of v without replacing it with another element in the last time step, thus $\hat{v}$ is $N-1$ elements long, meaning that both $\hat{x}$ and $\hat{u}$ can also only be $N-1$ elements long. From the definitions above, it is easy to show that $\tilde{x}_k \in X_k - \{\hat{x}_k\}$ and $\tilde{u}_k \in K(X_k - \{\hat{x}_k\}) + \{v_k\} - \{\hat{v}_k\}$. Defining $D(x_k, u_k)$ to be

$$D(x_k, u_k) = \begin{bmatrix} A_1 x_k + B_1 u_k & \dots & A_P x_k + B_p u_k \end{bmatrix}, \tag{96}$$

as in previous sections, one can write the following:

$$D(x_k, u_k) = D(\hat{x}_k + \tilde{x}_k, \hat{u}_k + \tilde{u}_k) = D(\hat{x}_k, \hat{u}_k) + D(\tilde{x}_k, \tilde{u}_k) \tag{97}$$

Following on from this, one can write:

$$\begin{aligned} D^T(x_k, u_k) D(x_k, u_k) =& D^T(\hat{x}_k, \hat{u}_k) D(\hat{x}_k, \hat{u}_k) \\ &+ D^T(\hat{x}_k, \hat{u}_k) D(\tilde{x}_k, \tilde{u}_k) \\ &+ D^T(\tilde{x}_k, \tilde{u}_k) D(\hat{x}_k, \hat{u}_k) \\ &+ D^T(\tilde{x}_k, \tilde{u}_k) D(\tilde{x}_k, \tilde{u}_k) \geq \beta^2 I \end{aligned} \tag{98}$$

From similar arguments that were used in the discussion of the present time persistent excitation condition, because the last term in equation 98 is quadratic and necessarily positive definite, one can drop this term from the inequality to obtain an inequality that is sufficient for the future persistent excitation condition, but has a slight degree of conservativeness:

$$D^T(x_k, u_k) D(x_k, u_k) \geq D^T(\hat{x}_k, \hat{u}_k) D(\hat{x}_k, \hat{u}_k) + D^T(\hat{x}_k, \hat{u}_k) D(\tilde{x}_k, \tilde{u}_k) + D^T(\tilde{x}_k, \tilde{u}_k) D(\hat{x}_k, \hat{u}_k) \geq \beta^2 I \tag{99}$$

Given that the vertices of the state tubes are known, $X_k = \text{Co}\{U^j \alpha_k,\ j = 1, ..., m\}$, $D(\tilde{x}_k, \tilde{u}_k)$ can be written,

$$D(\tilde{x}_k, \tilde{u}_k) = D(\tilde{x}, K\tilde{x}_k + v_k - \hat{v}_k) \in \text{Co}\{D(U^j \alpha_k - \hat{x}_k, K(U^j \alpha_k - \hat{x}_k) + v_k - \hat{v}_k)\}. \tag{100}$$

Equation 99 then becomes

$$\begin{aligned}D^T(\hat{x}_k,\hat{u}_k)D(\hat{x}_k,\hat{u}_k)+D^T(\hat{x}_k,\hat{u}_k)D(U^j\alpha_k-\hat{x}_k,K(U^j\alpha_k-\hat{x}_k)+v_k-\hat{v}_k)+&\\ D^T(U^j\alpha_k-\hat{x}_k,K(U^j\alpha_k-\hat{x}_k)+v_k-\hat{v}_k)D(\hat{x}_k,\hat{u}_k)\geq\beta^2 I&\end{aligned}\tag{101}$$

To implement this future persistent excitation condition, the following constraints must be added to the online optimisation problem.

$$\begin{aligned}\forall\ j:\ &D_i^T(\hat{x}_i,\hat{u}_i)D_i(\hat{x}_i,\hat{u}_i)+D_i^T(\hat{x}_i,\hat{u}_i)D(U^j\alpha_k-\hat{x}_k,K(U^j\alpha_k-\hat{x}_k)+v_k-\hat{v}_k)+\\ &D_i^T(U^j\alpha_k-\hat{x}_k,K(U^j\alpha_k-\hat{x}_k)+v_k-\hat{v}_k)D_i(\hat{x}_i,\hat{u}_i)\geq M_i\\ &M_i\geq 0\\ &\sum_{i=1}^{N-1}M_i\geq\beta^2 I\end{aligned}\tag{102}$$

The above constraints are added for each $i=1,...,N-1$, so in total, imposing the future persistent excitation condition requires the inclusion of $j\times(N-1)$ additional constraints to the online optimisation.

6.4.2 Varying True Model Parameter

The previous section detailed the implementation of a persistent excitation condition that stretched into future time steps, that the author conjectures will speed up the convergence of the parameter set. This section will examine a new parameter set update algorithm that will allow the parameter set to still converge to a point θ^*, when the true parameter value used at each time step is

$$\theta_k=\theta^*+\tilde{\theta}_k\qquad\tilde{\theta}_k\in S=\{\theta:U\theta\leq h\}.\tag{103}$$

Using similar approaches to the derivation of the original parameter set update algorithm discussed in previous sections, one can begin by writing

$$x_{k+1}=A(\theta_k)x_k+B(\theta_k)u_k+w_k,\tag{104}$$

which can be expanded using the definition of $A(\theta)$ and $B(\theta)$ from equation 2 to be written as

$$x_{k+1}=\begin{bmatrix}A_0\ A_1[\theta_k]_1\ A_2[\theta_k]_2\ A_3[\theta_k]_3\end{bmatrix}x_k+\begin{bmatrix}B_0\ B_1[\theta_k]_1\ B_2[\theta_k]_2\ B_3[\theta_k]_3\end{bmatrix}u_k+w_k.\tag{105}$$

Separating the theta dependent terms from the non-theta dependent terms gives

$$\begin{aligned} x_{k+1} =& A_0 x_k + B_0 u_k + \begin{bmatrix} A_1 x_k + B_1 u_k & A_2 x_k + B_2 u_k & A_3 x_k + B_3 u_k \end{bmatrix} \theta_k + w_k \\ =& A_0 x_k + B_0 u_k + \begin{bmatrix} A_1 x_k + B_1 u_k & A_2 x_k + B_2 u_k & A_3 x_k + B_3 u_k \end{bmatrix} (\theta^* + \tilde{\theta}_k) + w_k. \end{aligned} \tag{106}$$

Using the definition of $D(x_k, u_k)$ from equation 22, this gives

$$x_{k+1} = A_0 x_k + B_0 u_k + D(x_k, u_k)\theta^* + D(x_k, u_k)\tilde{\theta}_k + w_k. \tag{107}$$

The unfalsified parameter set at time $k+1$ can thus be written

$$\Delta_{k+1} = \{\theta^* : x_{k+1} - A_0 x_k - B_0 u_k - D(x_k, u_k)\theta^* = D(x_k, u_k)\tilde{\theta}_k + w_k\}, \tag{108}$$

and the updated parameter set is given by $\Theta_{k+1} \supseteq \Theta_k \cap \Delta_{k+1}$ where $\Theta_k = \{\theta : \Pi_\theta \theta \leq \pi_{\theta,k}\}$. The vector that defines the updated parameter set is computed from the optimisation problem below:

$$\begin{aligned} [\pi_{\theta,k+1}]_i = \max_{\theta^*, \tilde{\theta}, w} & [\pi_\theta]_i \theta^* \\ & \text{s.t.} \quad \Pi_\theta \theta^* \leq \pi_{\theta,k} \\ & x_{k+1} - A_0 x_k - B_0 u_k - D(x_k, u_k)\theta^* = D(x_k, u_k)\tilde{\theta} + w \\ & U\tilde{\theta} \leq h \\ & \Pi_w w \leq \pi_w \end{aligned} \tag{109}$$

In the formulation above, we have defined $\theta^* \in \hat{\Theta}$, and $\tilde{\theta} \in S$ where $\hat{\Theta} = \{\theta : \Pi_\theta \theta \leq \pi_k\}$ and $S = \{\theta : U\theta \leq h\}$. Given these definitions for the sets that θ^* and $\tilde{\theta}$ lie in, the parameter θ_k lies in a set defined by the Minkowski sum of the sets $\hat{\Theta}$ and S: $\Theta_k = \hat{\Theta} \oplus S$, where the operator $\oplus$ denotes the Minkowski sum. The Minkowski sum of two general polytopes is not straightforward to calculate, so to ease the computational burden of the algorithm, the sets that θ^* and $\tilde{\theta}$ lie in were defined to be similar, meaning that $\theta^* \in \hat{\Theta} = \{\theta : \Pi_\theta \theta \leq \pi_\theta\}$ and $\tilde{\theta} \in S = \{\theta : \Pi_\theta \theta \leq h\}$. With the sets defined in this way, the set that θ_k lies in can trivially be defined as the Minkowski sum of $\hat{\Theta}$ and S: $\theta_k \in \Theta_k = \hat{\Theta} \oplus S = \{\theta : \Pi_\theta \theta \leq \pi_\theta + h\}$.

Because the definition of the set that θ_k lies in has been modified slightly, a slight change in the constraints to the online optimisation are also required to implement the new parameter set update algorithm. Specifically, the constraints

$$\begin{aligned} \Lambda^j_{i|k}\pi_\theta &\leq \alpha_{i+1|k} - Vd(U^j\alpha_{i|k}, KU^j\alpha_{i|k} + v_{i|k}) - \bar{w} \\ \Lambda^j_{N+1|k}\pi_\theta &\leq \alpha_{N+1|k} - Vd(U^j\alpha_{N+1|k}, KU^j\alpha_{N+1|k}) - \bar{w} \end{aligned} \tag{110}$$

must be rewritten as

$$\begin{aligned} \Lambda^j_{i|k}(\pi_\theta + h) &\leq \alpha_{i+1|k} - Vd(U^j\alpha_{i|k}, KU^j\alpha_{i|k} + v_{i|k}) - \bar{w} \\ \Lambda^j_{N+1|k}(\pi_\theta + h) &\leq \alpha_{N+1|k} - Vd(U^j\alpha_{N+1|k}, KU^j\alpha_{N+1|k}) - \bar{w}. \end{aligned} \tag{111}$$

This is a slight modification to the online optimisation proposed in previous sections, but it is necessary for the online optimisation to work correctly together with the newly proposed parameter set update algorithm.

Following along the lines of the discussion about the distribution of the disturbance and its relation to the rate of convergence of the parameter set estimate, one can make similar arguments about the value chosen at each time step for $\tilde{\theta}$. Much like the disturbance set argument, the rate of convergence of the parameter set will now also be dependent upon how often the value for $\tilde{\theta}$ lies on the boundary of the set it is contained in S. The more often that $\tilde{\theta}$ lies on the boundary of S, the quicker the parameter set estimate will converge to θ^*, and the less often $\tilde{\theta}$ lies on the boundary of S, the slower the parameter set estimate will converge.

In the simulations that the author has performed utilising the newly proposed parameter set update algorithm, both the disturbance and value for $\tilde{\theta}$ were chosen to lie on the boundaries of their respective sets to speed up parameter set convergence and lessen the simulation time required.

It is conjectured that the rate of parameter set convergence will be slower in the newly proposed algorithm with uncertainty on the true parameter value because the total uncertainty in the overall system has risen. The numerical results achieved from performing these simulations, and the simulations including the persistent excitation constraint that stretches into future time steps will be presented in the next section.

7 Results and Discussion

This section will examine the numerical results achieved from simulating controllers with two novel qualities discussed in the previous section. Both of the novel algorithms described are implemented as part of the V-form implementation. In order to implement the future PE condition, the vertices of the state tubes must be known, which necessitates the use of the V-form algorithm. While the new parameter set update algorithm can theoretically be used in conjunction with both the H-form and V-form algorithms, the numerical results displayed below were obtained from using the algorithm together with the V-form implementation.

The specific numerical example used in these simulations is the same as the numerical example from Cannon and Lu's paper, and from previous sections. The prediction horizon used was 10 time steps long ($N = 10$), and the simulation was programmed in Matlab using YALMIP and the semidefinite programming solver MOSEK.

7.1 Future PE Condition

Simulation results from the specific numerical example being examined has confirmed intuition that the inclusion of a persistent excitation condition that takes into account future time steps will increase the rate of convergence of the parameter set estimate. As Figure 9 shows that while the states do approach the origin, the region around the origin to which the states eventually converge is a much larger region than seen before without the future PE condition. It is important to note that the inclusion of the $-\lambda\beta^2$ in the cost function from equation 85 means that there are no longer any hard

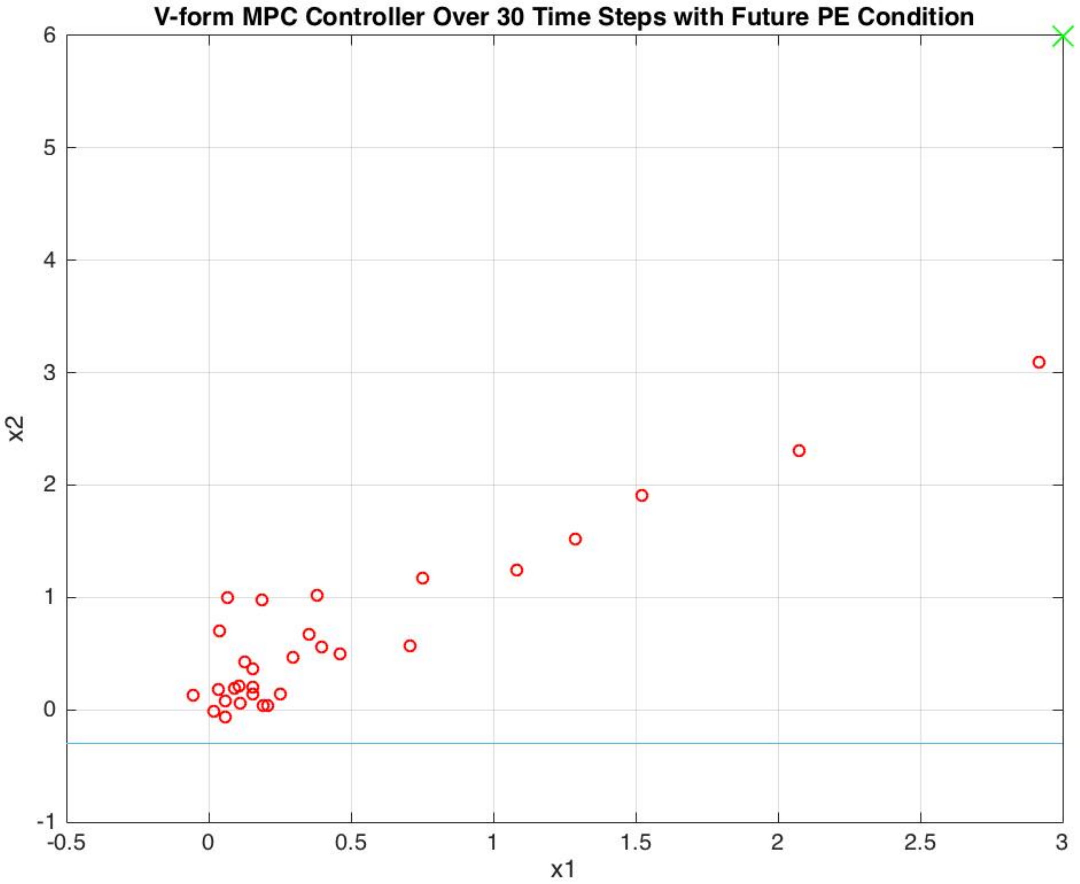

Figure 9: Evolution of states: V-form controller with future PE condition

guarantees on the stability of the controller, but in practice while the states did not converge to a tight region around the origin, they did appear to be bounded by a larger region around the origin. Figure 9 makes clear a trade-off between how closely the system tracks the reference (origin in this case), and how much new information is gained about the parameter set at each time step. In this example, because the system is allowed to 'explore' the state space more than the previous numerical example without a future PE condition, the system is able to identify the true value of the parameter quicker, and thus the parameter set estimate converges quicker than before.

Because the system does not converge to a small region around the origin, the plot of the sum of the squared states does not look as it does in previous examples. Whereas in previous examples the plot levels off at a point where it has a very shallow slope proportional to the sum of the squared disturbances thereafter, the sum of the squared states with the future PE condition imposed has clear sections where the sum rises quickly, meaning that there is not a prolonged section of shallow slope. This is evident from Figure 10. As can be seen, the sharp increase of the sum around time steps 18-23 correspond to the system states being further from the origin and 'exploring' the state space. This has the effect of increasing the information gained at each time step about the true parameter value, directly leading to quicker convergence of the parameter set. Over the long run, this will eventually lead to better overall performance as having a more accurate parameter set estimate lowers the optimal cost and thus increases performance of the controller. Thus, in the short term, the controller performs slightly worse while the system is exploring the state space and acquiring new information about the model parameters which eventually translates into better performance.

With this argument in mind, the author has proposed an algorithm that selectively implements the future persistent excitation condition at specific time steps rather than implementing it constantly at every time step. In this way, the future PE constraint can be turned off in periods where the short term performance of the controller is most important, and in periods when the tracking of the controller can be relaxed, the future persistent excitation condition can again be implemented so that the controller is learning about the true parameter at a faster rate. This is an effective way to handle the trade off between close tracking of the reference, and fast parameter set estimate convergence. The two methods that the author has considered are the case when the future persistent excitation condition is implemented on every other time step, and the case where it is implemented constantly over the first half of the simulation and then not over the second half of the simulation. Both of these proposed methods for handling the trade-off between controller performance led to quicker parameter set convergence compared to simulations when the persistent excitation condition only accounted for past and present time steps. The table below shows the final parameter set diameter after 60 time

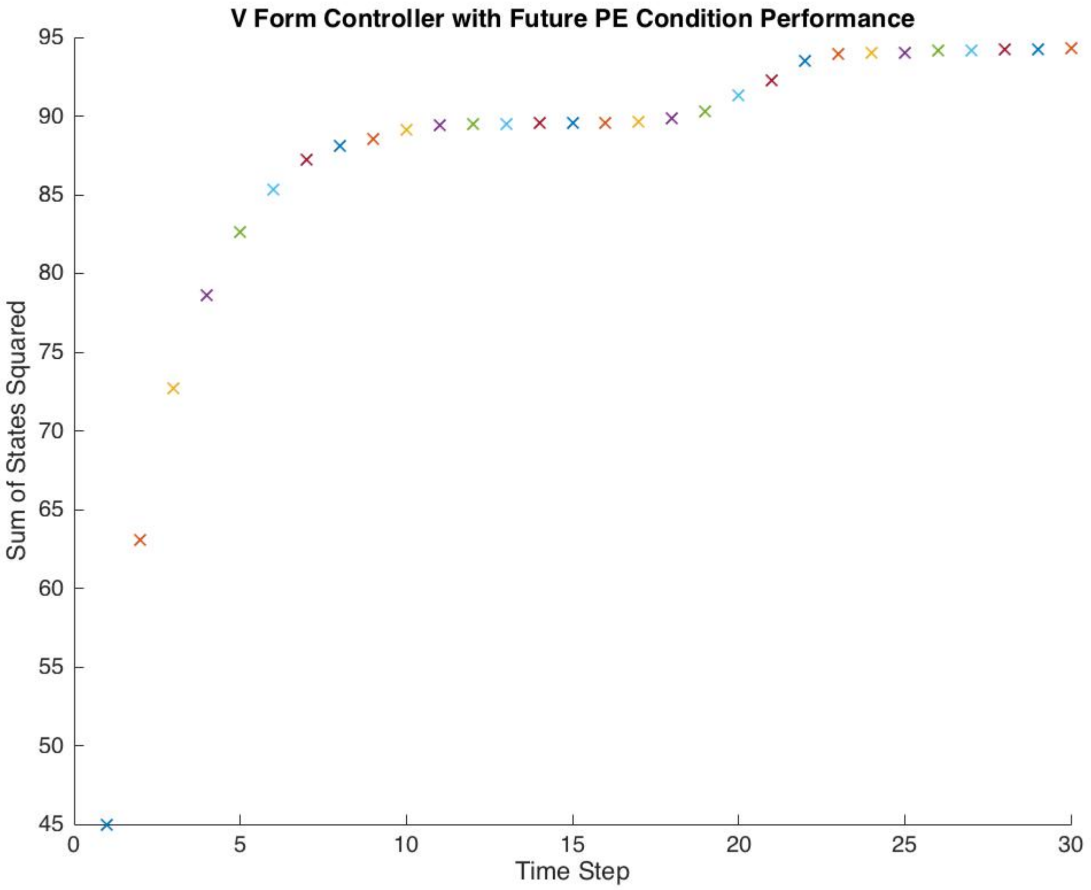

Figure 10: Sum of square of states: V-Form with Future PE condition

steps of each proposed method of implementing the persistent excitation condition over the course of ten separate simulations. The table clearly shows that both proposed methods of implementing the future persistent excitation condition result in significantly quicker parameter set convergence compared to the traditional persistent excitation condition. While implementing the future persistent excitation constantly over the first half of the simulation achieved faster parameter set convergence than implementing the condition at alternating time steps over the length of the whole simulation, the states track the origin much closer when the future PE condition is implemented at alternating time steps, which is consistent with the arguments made previously.

Figure 11 shows the evolution of the system states over 60 time steps for the implementation where the future persistent excitation condition is enforced at every other time step over the entire length of the simulation time. The points plotted in blue correspond to time steps where the future PE condition was not enforced, and the red points correspond to time steps where the future PE condition was enforced. It can be seen that the states to do not converge to a region around the origin that is as tight as the implementation that does not include a future persistent excitation condition, such as that showing in Figure 4, but it also tracks the origin much closer than a system that blindly

Alternating Future PE	Future PE First 30	Present Time PE
0.4392	0.1427	0.6999
0.1026	0.1366	0.4968
0.0370	0.6735	0.5280
0.0617	0.0971	0.2612
0.2460	0.0317	0.4197
0.5001	0.2429	0.3026
0.1368	0.2714	0.4486
0.1763	0.0165	0.0969
0.3278	0.1033	0.3519
0.2731	0.0413	0.3471
average = 0.23006	average = 0.1757	average = 0.39527

Table 1: Parameter Set Estimate Diameter After 60 Time Steps

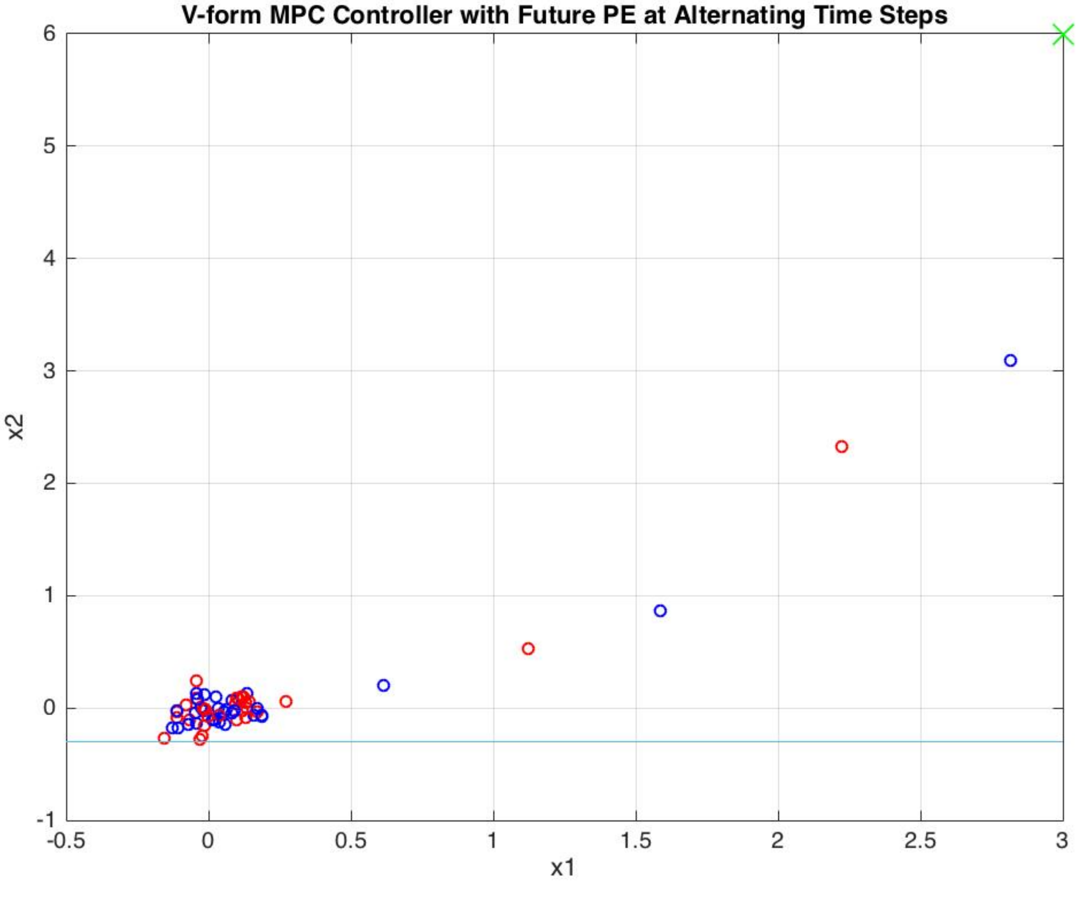

Figure 11: Evolution of States: V-form controller with Future PE condition at every other time step

implements the future persistent excitation condition at every time step.

Figure 12 shows the evolution of states of a controller that implements the future persistent excitation condition for the first thirty time steps then has no persistent excitation condition for the remaining thirty time steps. For the last thirty time steps, the future PE constraints are removed and the weight λ in the cost function is set to zero so that the objective function being optimised is

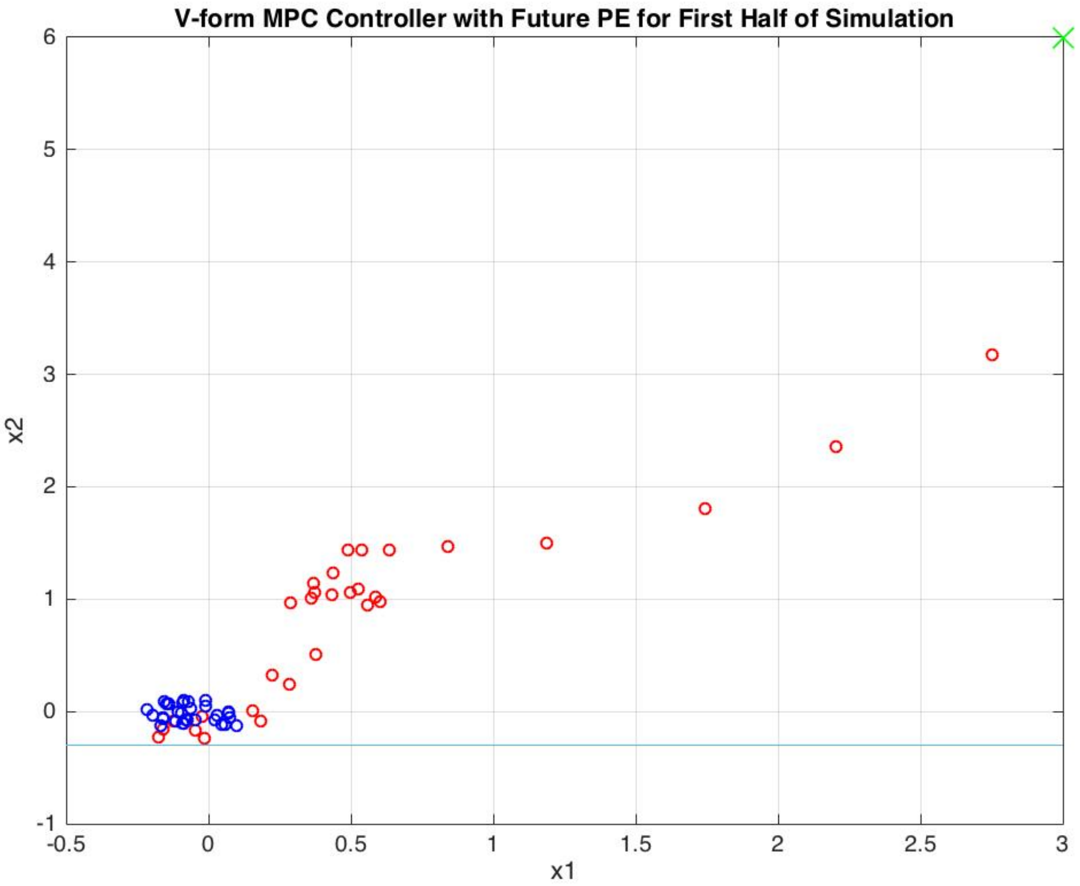

Figure 12: Evolution of States: V-form controller with Future PE condition for first 30 time steps

the original nominal cost function discussed in prior sections, meaning the stability guarantees of the controller are recovered.

The red points in the graph respond to the first thirty time steps, where the future PE condition is being implemented and the blue points correspond to the final thirty time steps with no persistent excitation condition. The red points resemble the plot in Figure 9 states as it is essentially the same simulation over the course of thirty time steps, and then the blue points resemble close the plot from Figure 4. The plot is as one would expect: in the first thirty time steps the system explores the state space more freely whilst quickly learning about the true model parameter and then the system is fully stabilised and tracks a region very tight around the origin once the future persistent excitation condition is turned off.

Plots of the sum of squared states of both simulations in Figure 13 and Figure 14 further highlight the trade-off between short term performance and speed of convergence of the parameter set estimate. It is clear to see that the short term performance of the controller that implements the future PE condition at alternating time steps is much better than the controller that implements the future PE condition for the first thirty time steps then no PE condition for the final thirty time steps. As

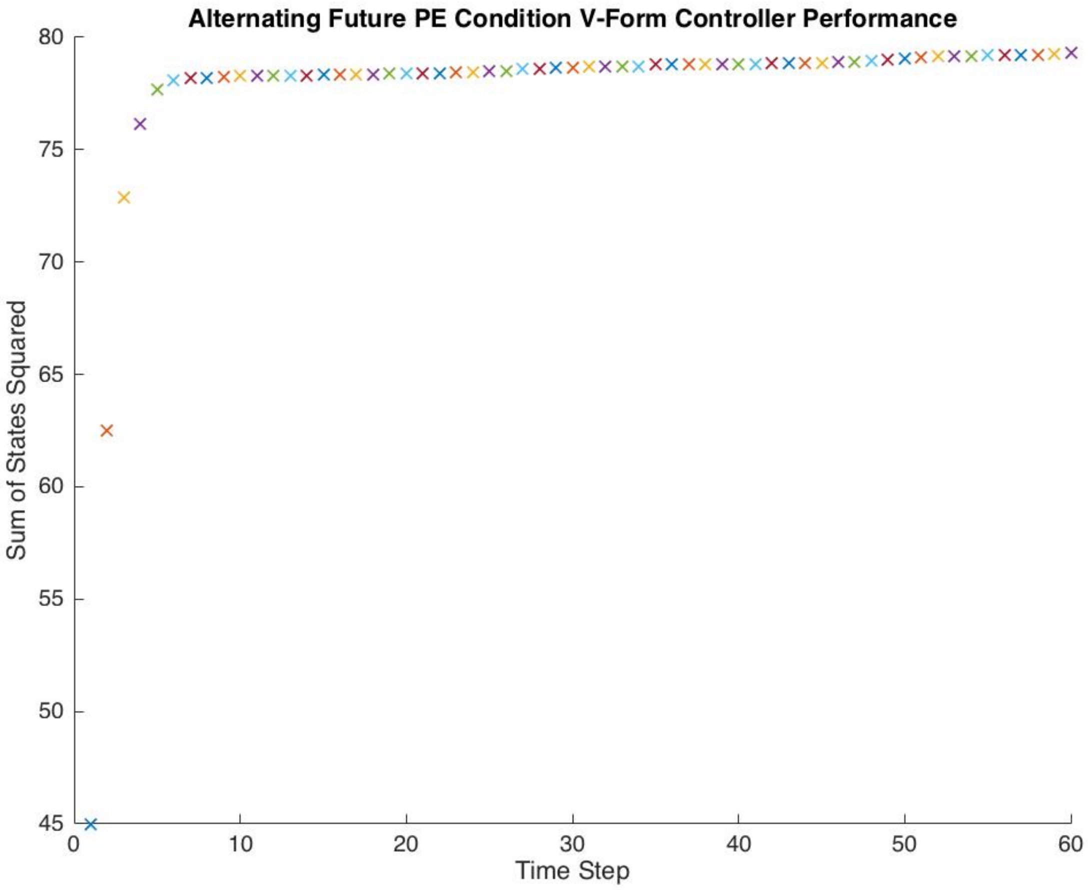

Figure 13: Performance of V-form Controller with future PE condition at every other time step

evidenced by the much lower values of sum of squared states in Figure 13 compared to those in Figure 14. This is completely in line with what would be expected as the plots of system states show that the alternating future PE system tracks the origin much more closely over the entire time range of the simulation compared to the system that implements the future PE condition for the first half of the simulation. As a result, one would expect the plot of the sum of squared states of the alternating future PE condition controller to have lower values to its counterpart. However, due to the trade-off discusses, and as the table clearly shows, the alternating future PE condition controller does not exhibit parameter set convergence that is as quick as its counterpart.

As the trade-off between parameter set convergence rate and short term performance is made clear, and obvious question that arises is how to effectively strike a balance between two competing effects. As the parameter set converges to a singleton at the true parameter value, the long term performance of the controller increases, as the optimal cost decreases. Thus the trade-off is between long term high performance and short term high performance. The author has proposed two possible methods to balance this trade-off: alternate between implementing the future persistent excitation condition and no persistent excitation condition at every other time step, and implement the future persistent

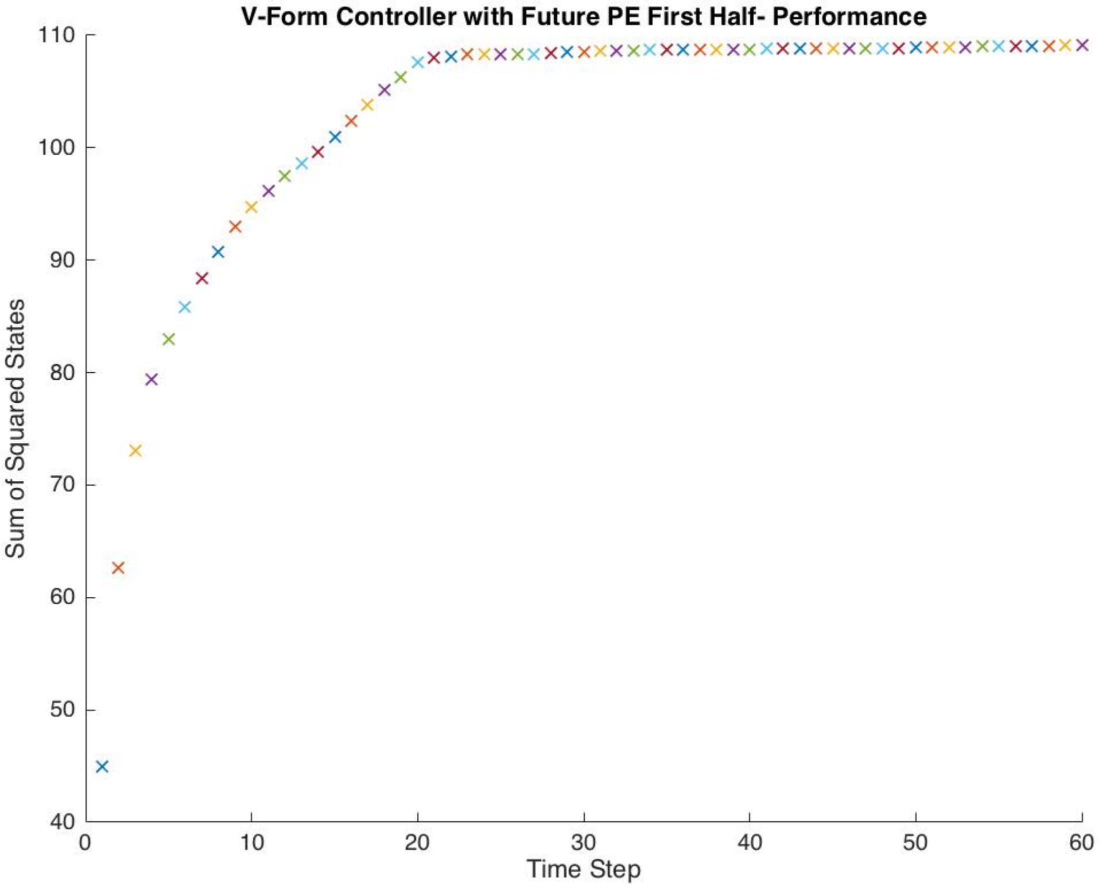

Figure 14: Performance of V-form Controller with future PE condition for first 30 time steps

excitation condition constantly for a short period of time so that the parameter set greatly decreases in size, then turn the condition off so that the system is able to track the reference well. Essentially, the user of the controller must define periods where model identification is more important than reference tracking, and periods where the opposite is true, and then implement the future persistent excitation condition within these time periods accordingly. The author believes the two proposed methods to be adequate general approaches that will work in most simple scenarios, but more work needs to be done on the topic.

Future work will consider using neural networks to intelligently choose when to apply the future persistent excitation condition, and when to not include a persistent excitation condition and prioritise reference tracking instead.

7.2 Varying True Model Parameter

It is important to still be able to recover the true parameter value in scenarios where at each time step there is a disturbance added to the true parameter value: $\theta_k = \theta^* + \tilde{\theta}_k$ where $\tilde{\theta}_k \in S = \{\theta : U\theta \leq h\}$. Due to reasons discussed in the previous section, the matrix U defining the shape of the set S is defined to be equal to the matrix defining the shape of the parameter set estimate Π_θ. This is so that the computation required to compute the Minkowski sum of the two sets is straightforward.

The objective of the newly proposed parameter set estimate update algorithm is to thus update the parameter set estimate so that it converges to a singleton containing the value of θ^*. Following on from similar arguments made in the section discussing the convergence of the original parameter set estimate, it is possible for the new parameter set estimate algorithm to converge to a singleton containing θ^* if there is a non zero probability of the value of $\tilde{\theta}$ lying on the boundary of the set S and there is also a non-zero probability of the disturbance lying on the boundary of its set W. Again, the rate of convergence of the parameter set estimate will be dependent upon the distribution for the disturbance and $\tilde{\theta}$ near the boundaries of their sets. If these values lie on the boundary of their respective sets often, convergence will be quick, and vice versa. To lower the simulation time required to observe convergence of the parameter set to θ^*, the values for both $\tilde{\theta}$ and the disturbance w_k were chosen to be on the boundaries of their respective bounding sets at each time step of the simulation.

The importance of such an algorithm to identify the true parameter value in the presence of some disturbance on the parameter is highlighted in the case where the state is being estimated, or observed with a noisy sensor. In this case, the exact value of the state will not be known, so there will be some error involved in either the estimate of the state or the observation of the state. This can be represented through the following equation,

$$\hat{x}_{k+1} = x_{k+1} - e_{k+1}, \qquad e_{k+1} \in E_{k+1} \subseteq E_0 \tag{112}$$

with the important assumption being made that the error term is non-expanding and always bounded by the initial bound on the error E_0. This error on the state can be represented as a disturbance on the system parameter value, and thus using the novel parameter set update algorithm the true parameter value of the system can still be computed. To do this, consider the unfalsified set

$$\Delta_{k+1} = \{\theta : \hat{x}_{k+1} - A_0\hat{x}_k - B_0 u_k - D(\hat{x}_k, u_k)\theta + e_{k+1} - D(e_k, 0)\theta \in W\}, \tag{113}$$

which can be rewritten in the form of equation 108 by taking

$$\tilde{\theta} = e_{k+1} - A(\theta)e_k, \qquad \theta \in \Theta_k \subset \Theta_0. \tag{114}$$

Thus the newly proposed parameter set update algorithm can be used, and the parameter set estimate will converge to the true value of the parameter.

Figure 15 below shows the diameter of the parameter set at each time step in a simulation over the course of 2000 time steps. As can be seen, the plot clearly shows that the parameter set estimate converges to a singleton, and the simulation confirms that this singleton contains the value of θ^*. This numerically confirms the analysis that showed that the parameter set would converge to a singleton containing θ^*. The reason that the parameter set converges to a single point in this simulation is because the bound on the disturbance on the parameter value is known exactly, i.e the set description $\tilde{\theta} \in S = \{\theta : \Pi_\theta \theta \leq h$ is correct. In other words, the true disturbance in the simulation is also sampled randomly from the same set S. The randomly sampled disturbance is then projected onto the boundary of the set S to speed up the convergence of the parameter set estimate, but a guarantee of convergence still holds if this is not the case because the probability of the disturbance being on the boundary of the set S is non zero.

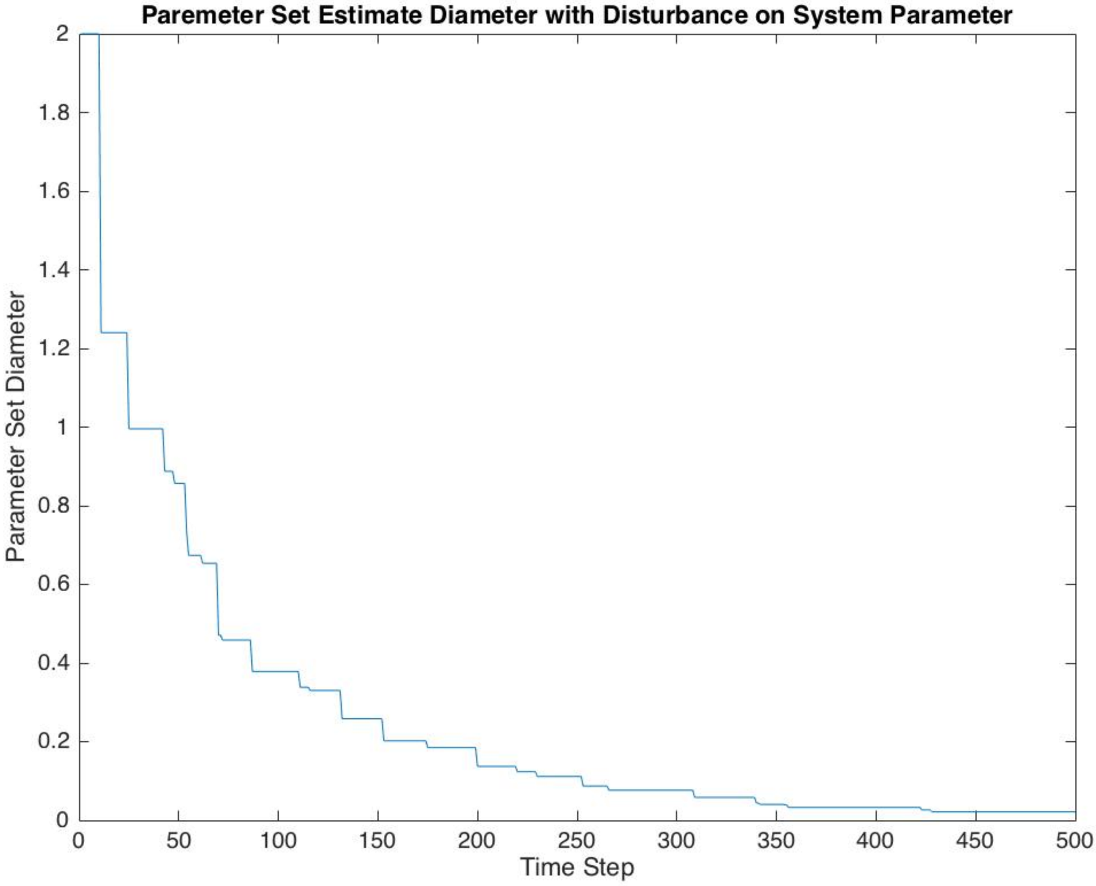

Figure 15: Diameter of the parameter set estimate when the system parameter takes the form $\theta_k = \theta^* + \tilde{\theta}$

The discussion above has considered the case where the set S that the disturbance on the system parameter lies in is known exactly. In this case, as shown above, the parameter set estimate converges to the true value θ^*. However, this case is very restricted and not likely to be satisfied in reality. The author has also examined the case where the only information known about the set S is an estimate of the set $\hat{S}$ with the condition that $\hat{S} \supset S$. It would be a useful result to be able to show that $\Theta_k \to \theta^* \oplus (\hat{S} \ominus S)$ as $k \to \infty$. The proof that this is indeed the case is shown below.

To begin the proof, it is important to show the proof of convergence of the parameter set to the true value of theta when there is no disturbance on the parameter, and the disturbance on the system lies within a bounded set ($w_k \in W$).

$$\begin{aligned} x_{k+1} &= A(\theta^*)x_k + B(\theta^*)u_k + w_k \\ &= D(x_k, u_k)\theta^* + d(x_k, u_k) + w_k \\ &= D_k\theta^* + d_k + w_k, \qquad w_k \in W \end{aligned} \tag{115}$$

The unfalsified parameter set is written as

$$\begin{aligned} \Delta_{k+1} &= \{\theta : x_{k+1} - D_k\theta - d_k \in W\} \\ &= \{\theta : D_k(\theta^* - \theta) + w_k \in W\} \\ &= \{\theta : D_k(\theta^* - \theta) \in W \oplus (-w_k)\}. \end{aligned} \tag{116}$$

Because there is an implementation of the future persistent excitation condition, $\sum_{k=t}^{t+N-1} D_k^T D_k \geq \beta I$, the following condition holds: $\sum_{k=t}^{t+N-1} ||D_k(\theta^* - \theta)||^2 \geq \beta||\theta^* - \theta||^2$. This implies that $D_k(\theta^* - \theta) \neq 0$ if $\theta^* \neq \theta$.

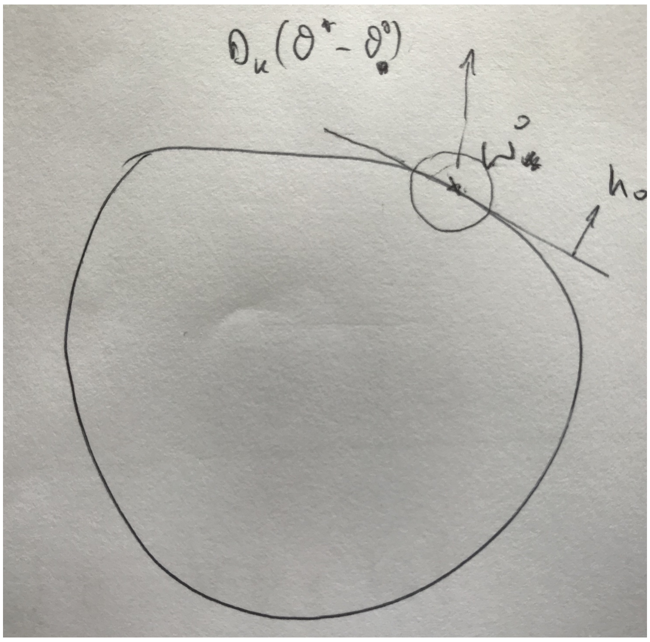

Figure 16: $h_o^T D_k(\theta^* - \theta^e) \geq 0$

The condition for a particular value of θ, θ^e, to be excluded from the unfalsified parameter set is

$h_o^T D_k(\theta^* - \theta^e) \geq 0$, where h_o is the normal vector describing the hyperplane that is tangent to the disturbance set at a point w_o. This is demonstrated graphically in Figure 16. Thus for a realisation of the disturbance w_k that is arbitrarily close to the point w_o,

$$\begin{aligned} &\forall \;\; \epsilon > 0 \quad \Pr\{||w_k - w_o|| < \epsilon\} > \delta \;\; \text{for some} \;\; \delta > 0 \\ &\text{If} \;\; ||D_k(\theta^* - \theta^e|| > \gamma \\ &\text{Then} \;\; \theta^e \notin \Delta_{k+1} \Leftarrow \begin{cases} h_o^T D_k(\theta^* - \theta^e) > 0 \\ \text{If } ||w_k - w_o|| < \epsilon \end{cases} \end{aligned} \tag{117}$$

where $\gamma = \epsilon$. As these conditions are satisfied with non zero probability, there is a non zero probability that a particular value of $\theta^e \neq \theta^*$ will be excluded from the unfalsified parameter set, meaning that over an infinite period the parameter set will converge to a singleton at the true parameter value θ^* with probability one.

For the case where the disturbance set W is not known exactly, but an estimate $\hat{W}$ that is the smallest set that bounds W is known ($\hat{W} \supset W$), the same arguments can be made to show convergence of the parameter set estimate.

$$\begin{aligned} \Delta_{k+1} &= \{\theta : D_k(\theta^* - \theta) + w_k \in \hat{W}\} \\ &= \{\theta : D_k(\theta^* - \theta) \in \hat{W} \oplus (-w_k)\} \end{aligned} \tag{118}$$

This equation is in the same form as the previous discussion, and the same arguments can be made to prove convergence of the parameter set estimate under these conditions.

For the more general case where $\hat{W}$ is a conservative estimate of the disturbance set W such that $\hat{W} = W \oplus \tilde{W}$, one can again show convergence of the parameter set estimate to the true parameter value by writing

$$\Delta_{k+1} = \{\theta : D_k(\theta^* - \theta) \in W \oplus \tilde{W} \oplus (-w_k)\}, \tag{119}$$

meaning that the condition for the exclusion of θ^e is $D_k(\theta^* - \theta^e) \oplus (-\tilde{W}) \notin W \oplus (-w_o)$. Again, invoking previous arguments, this shows convergence of the parameter set to a singleton at the true parameter value.

The final case to consider is when the system parameter contains a noisy term and only a conservative estimate of the set that the disturbance lies in is known: $\theta_k = \theta^* + \tilde{\theta}_k,\ \tilde{\theta}_k \in S,\ \hat{S} \supset S$. Again using similar arguments as before, one can write the unfalsified parameter set

$$\Delta_{k+1} = \{\theta : D_k(\theta^* - \theta) + w_k + D_k\tilde{\theta}_k \in W \oplus D_kS\}, \tag{120}$$

where $D_kS = \{z : z = D_k\tilde{\theta},\ \tilde{\theta} inS\}$. Next, let $v_k = w_k + D_k\tilde{\theta}_k,\ v_k \in W \oplus D_kS$, then

$$\Delta_{k+1} = \{\theta : D_k(\theta^* - \theta) \in W \oplus D_kS \oplus (-v_k)\}, \tag{121}$$

where $W \oplus D_kS = \{w + D_k\tilde{\theta},\ w \in W,\ \tilde{\theta} \in S\}$. If the sets $\hat{S}$ and S are of the same form, i.e $\hat{S} = \{\theta : \Pi_S\theta \leq \pi_{\hat{S}}\}$ and $S = \{\theta : \Pi_S\theta \leq \pi_S\}$, then $\hat{S} = (\hat{S} \ominus S) \oplus S$. Using this identity, the condition for inclusion of θ^e in the unfalsified set is

$$\theta^e \in \Delta_{k+1} \text{ if } D_k(\theta^* - \theta^e) \oplus \{-D_k(\hat{S} \ominus S)\} \in (W \oplus D_kS) \oplus \{-v_k^o\}, \tag{122}$$

meaning that

$$\theta^e \notin \Delta_{k+1} \Leftarrow h_o^T[D_k(\theta^* - \theta^e) - D_k\tilde{\theta}] > 0 \ \ \forall \ \ \tilde{\theta} \in \hat{S} \ominus S \tag{123}$$

proving that the parameter set estimate converges to $\theta^* \oplus (\hat{S} \ominus S)$ as $k \to \infty$.

Simulations on the numerical example discussed in previous sections have confirmed that the theoretical results obtained above do hold. To reduce convergence time, both the disturbance on the system and the disturbance on the model parameter were chosen to lie on a vertex of their respective sets. With these conditions, the parameter set estimate was found to converge to $\Theta_k = \theta^* \oplus (\hat{S} - S)$ after nearly 4000 time steps. As the disturbances were chosen to optimise the rate of convergence, one can expect the rate of convergence to be much slower than in this numerical example, however the parameter set estimate will converge to $\Theta_k = \theta^* \oplus (\hat{S} - S)$ with probability one over a sufficiently long window, as proved above.

8 Conclusions

This report has examined in detail adaptive robust MPC algorithms existing in the relevant literature and has recreated the results reported in the literature. The report discusses at great length the theory behind these adaptive robust MPC algorithms, examining both the tube based model predictive control algorithm, and the set based model identification methods used in conjunction with the MPC algorithms.

The report then moves on to discuss the persistent excitation condition required to guarantee that the set based model identification algorithm will yield a parameter set estimate that converges to a singleton at the true parameter value with probability one over the course of an infinite period of time. Conditions that affect the rate of parameter set estimate convergence are then discussed.

The report then presents a novel V-form algorithm that is a tractable equivalent of the H-form algorithm presented by Cannon and Lu [1] that has comparable performance. The motivation for developing an equivalent V-form algorithm was due to the need to know the vertices of the predicted state tubes when implementing an improved persistent excitation condition that considers a window starting at current time and extending into the future, using the predicted states and inputs. This newly presented future persistent excitation condition has not been seen in the relevant literature before, and is believed to be original work by the author. The theory needed to implement the future persistent excitation condition is spelled out in detail, and the results of the implementation are presented. This novel future persistent excitation condition condition leads to significantly faster parameter set convergence compared to implementations of traditional persistent excitation conditions at the expense of allowing the system to deviate from the reference signal to a greater extent than in the traditional persistent excitation condition implementation. Methods of implementing the future persistent excitation condition that still allow for faster parameter set convergence with tighter tracking of the reference signal are discussed, and the numerical results of these algorithms are presented, demonstrating greatly improved parameter set convergence rate compared to existing algorithms while still maintaining tight tracking of the reference signal.

Finally, the report proposes a novel parameter set estimate update algorithm in the case that the model parameter is based on a 'true' parameter in the presence of an additive disturbance on the parameter: $\theta_k = \theta^* + \tilde{\theta}_k,\ \tilde{\theta} \in S$. The algorithm is able to correctly identify the true value of the model parameter θ^* in the case where the set S is known exactly. Additionally, the author has proved that in the case where only a conservative estimate $\hat{S}$ of S is known where $\hat{S} \supset S$, the algorithm yields a parameter set estimate that converges to $\Theta_k = \theta^* \oplus (\hat{S} \ominus S)$ as $k \to \infty$. This disturbance on the model parameter allows for the consideration of estimation or observation errors in the state, and is a novel method not seen before in the relevant literature.

Future work will consider different mechanisms to devise a point estimate for the model parameter. Choosing this point estimate through the use of neural networks will be considered, and the performance of this scheme will be compared to traditional methods for picking a point estimate. Furthermore, work will consider the definition of a metric that will allow for the direct comparison of methods that implement the future persistent excitation condition at different time steps so that the trade off between parameter set convergence and how closely the reference is tracked can be quantified and compared to other systems. Finally, more intelligent methods to choose at which time steps the future persistent excitation condition is implemented on will be examined.

References

[1] M. Cannon, X. Lu, 2019, 'Robust Adaptive Tube Model Predictive Control', *American Control Conference*. Philadelphia, 10-12 July 2019

[2] D. Seborg, T. Edgar, D. Mellichamp, F. Doyle, 2011, Process Dynamics and Control. 3rd ed. New York: John Wiley & Sons.

[3] M. Cannon, 2018, *Lecture 1: Model Predictive Control*, Lecture Notes, University of Oxford, delivered 22 October 2018.

[4] M. Lorenzen, M. Cannon,F. Allgöwer, 2018, 'Robust MPC with recursive model update', *Automatica*

[5] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control.* Birkhauser Boston, 2008.

[6] L. Chisci, A. Garulli, A. Vicino, and G. Zappa, "Block recursive parallelotopic bounding in set membership identification," *Automatica*, vol. 34, no. 1, pp. 15-22, 1998

[7] S. M. Veres, H. Messaoud, and J. P. Norton, "Limited-complexity model-unfalsifying adaptive tracking-control," *International Journal of Control*, vol. 72, no. 15, pp. 1417-1426, 1999

[8] B. Hassibi, A. H. Sayed, and T. Kailath. LMS is H^{∞} optimal. In *Proceedings of the Conference on Decision and Control,* pg 74-79, vol. 1, 1993

[9] E.W. Bai, H. Cho, and R. Tempo, "Convergence Properties of the Membership Set," *Automatica*, vol. 34, no. 10, pp 1245-1249, 1998

[10] M. Lorenzen, F. Allgöwer, and M. Cannon, "Adaptive Model Predictive Control with Robust Constraint Statisfaction," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3313-3318, 2017.

[11] E. Abraham, *Modelling and Analysis of Hybrid Systems; Convex polyhedra*, Lecture Notes, RWTH Aachen University, delivered SS 2015.

[12] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB", *In Proceedings of the CACSD Conference*, 2004