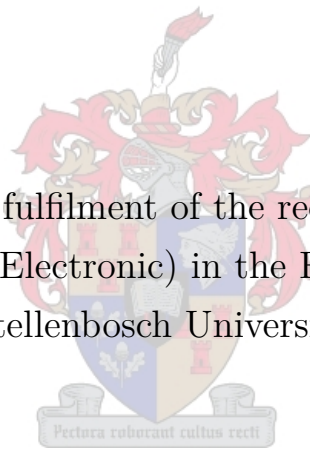# Data-Driven System Identification etc

Luke Skywalker

99652154

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University.

Supervisor: Dr O. W. Kenobi

Department of Electrical and Electronic Engineering

October 2099

Acknowledgements

I would like to thank my dog, Muffin. I also would like to thank the inventor of the incubator; without him/her, I would not be here. Finally, I would like to thank Dr Herman Kamper for this amazing report template.

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.

   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.

   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.

   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.

   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *Student number*	Handtekening / *Signature*
Voorletters en van / *Initials and surname*	Datum / *Date*

Abstract

English

The English abstract.

Afrikaans

Die Afrikaanse uittreksel.

Contents

List of Figures

List of Tables

Nomenclature

Variables and functions

$p(x)$	Probability density function with respect to variable x.
$P(A)$	Probability of event A occurring.
ε	The Bayes error.
ε_u	The Bhattacharyya bound.
B	The Bhattacharyya distance.
s	An HMM state. A subscript is used to refer to a particular state, e.g. s_i refers to the ith state of an HMM.
$\mathbf{S}$	A set of HMM states.
$\mathbf{F}$	A set of frames.
$\mathbf{o}_f$	Observation (feature) vector associated with frame f.
$\gamma_s(\mathbf{o}_f)$	A posteriori probability of the observation vector $\mathbf{o}_f$ being generated by HMM state s.
μ	Statistical mean vector.
Σ	Statistical covariance matrix.
$L(\mathbf{S})$	Log likelihood of the set of HMM states $\mathbf{S}$ generating the training set observation vectors assigned to the states in that set.
$\mathcal{N}(\mathbf{x}\|\mu, \Sigma)$	Multivariate Gaussian PDF with mean μ and covariance matrix Σ.
a_{ij}	The probability of a transition from HMM state s_i to state s_j.
N	Total number of frames or number of tokens, depending on the context.
D	Number of deletion errors.
I	Number of insertion errors.
S	Number of substitution errors.

Acronyms and abbreviations

AE	Afrikaans English
AID	accent identification
ASR	automatic speech recognition
AST	African Speech Technology
CE	Cape Flats English
DCD	dialect-context-dependent
DNN	deep neural network
G2P	grapheme-to-phoneme
GMM	Gaussian mixture model
HMM	hidden Markov model
HTK	Hidden Markov Model Toolkit
IE	Indian South African English
IPA	International Phonetic Alphabet
LM	language model
LMS	language model scaling factor
MFCC	Mel-frequency cepstral coefficient
MLLR	maximum likelihood linear regression
OOV	out-of-vocabulary
PD	pronunciation dictionary
PDF	probability density function
SAE	South African English
SAMPA	Speech Assessment Methods Phonetic Alphabet

Chapter 1

Introduction

The last few years have seen great advances in speech recognition. Much of this progress is due to the resurgence of neural networks; most speech systems now rely on deep neural networks (DNNs) with millions of parameters [1, 2]. However, as the complexity of these models has grown, so has their reliance on labelled training data. Currently, system development requires large corpora of transcribed speech audio data, texts for language modelling, and pronunciation dictionaries. Despite speech applications becoming available in more languages, it is hard to imagine that resource collection at the required scale would be possible for all 7000 languages spoken in the world today.

I really like apples.

1.1. Section heading

Chapter 2

Modelling

This chapter discusses the mathematical modelling of a quadrotor with a suspended payload which is based on a practical quadrotor UAV named Honeybee. The model is first derived as a 2D model. The system identification and control system techniques in later chapters will then be explained based on the 2D model to avoid unnecessary complexity. Finally, it will be described how this model and the techniques in later chapters are extended to the 3D case. This 3D mathematical model will be used in a nonlinear simulation of a quadrotor and suspended payload.

2.1. Coordinate frames

2.2. States

2.3. Forces and moments

2.4. Lagrangian mechanics

2.5. Linearised model

2.6. Discretised model

2.7. Model verification

Chapter 3

System identification

System identification is the process of creating mathematical models of a dynamical system by using input and output measurements of the system. Two major approaches are used to represent the dynamics of such a system, resulting in white-box or black-box models.

3.1. Parameter estimation

3.1.1. White-box models

In white-box models the physics of a model are understood by the user. These models are therefore determined from first principles. This can be done by modelling physical processes with techniques like Lagrangian mechanics or Newton equations. In this case, the mathematical relations between physical properties are predefined in the modelling phase. System identification is then reduced to parameter estimation to determine values for the parameters used in die model.

This is the approach used by [?] and [?] for swing damping control of a quadrotor with an unknown suspended payload. The system was modelled as two rigid bodies connected by a link and the following assumptions were made regarding the suspended payload:

- The payload is a point mass.

- The link is massless.

- The link is rigid.

- The link is attached to the CoM of the quadrotor.

The only unknown parameters in the quadrotor and payload model is the payload mass and link length. These parameters are first estimated and then inserted into the predefined, linearised model. This model is used by a LQR controller to damp swing angles while also controlling the vehicle.

The approach works well for systems with predictable dynamics, but it is not very adaptable. The payload considered by [**?**] and [**?**] is limited to a small rigid mass suspended from the quadrotor by a non-stretching cable. In this use case it was shown that a LQR controller successfully controls a quadrotor while minimising payload swing angles. However, if a payload or cable is used that violates one of the modelling assumptions, the predefined model no longer accurately represent the system. Since the controller is dependent on this model, the mismatch between the model and actual dynamics may result in undesirable controller behaviour.

Parameter estimation

3.1.2. Payload mass estimation

RLS

3.1.3. Cable length estimation

The cable length is estimated from the measurement of natural frequency of the swinging payload. As described by [**?**], the natural frequency is given by:

$$\omega_n = \sqrt{\frac{g}{l} \cdot \frac{m_q + m_p}{m_q}} \tag{3.1}$$

The natural frequency is measured by performing a FFT on the payload swing angle response after a position step by the quadrotor. The dominant frequency identified by the FFT during free swing is the natural frequency of the payload.

?? shows the payload swing angle after the system is stimulated by a position step setpoint. As shown in **??** the first few seconds of the step response are not used in the FFT. This is to minimise the effect of the quadrotor controllers on the swing angle frequency by excluding the transient response in the FFT.

?? shows the resulting amplitude spectrum of the payload swing angle response. The dominant frequency is clearly identified as ??. Since m_q and g is known, and m_p and ω_n has been estimated, l can now be determined from 3.1. In this case the estimated length is ??, compared to the actual length of ??.

Frequency resolution ?? error for different lengths??

3.2. Data-driven system identification

3.2.1. Black-box models

In contrast to white-box models, black-box models do not require predefined mathematical relations between system parameters. No prior knowledge of the physics of the system

are considered and no modelling assumptions are made. Black-box techniques determine the mathematical relationship between inputs and outputs of a system based only on measurement data. This is referred to as data-driven system identification.

Black-box models can be categorised as either non-linear or linear models. Non-linear models are often more accurate than linear models because complex, real-world dynamics are better approximated by non-linear systems. The dynamics of a quadrotor and suspended payload are also non-linear. Examples of black box models with quadrotors and payloads in literature ???

However, non-linear models are inherently more complex than linear models. Controllers that use non-linear models are usually more computationally complex than those with linear models. Control archictetures for quadrotors used in practical applications are mostly implemented on onboard hardware. Therefore there is value in low-complexity, linear models since these may be simple enough to execute on low cost hardware. trade-off between accuracy and complexity. Non-linear models may require control implementations that are too computationally expensive and may not be practically realisable on the available hardware on a quadrotor.

DMDc and HAVOK are the two data-driven system identification methods investigated in this paper. These are linear regression techniques that produce a linear model to approximate non-linear dynamics. Non-linear data-driven techniques like Neural Networks and SINDy [**?**] may produce models that are more accurate, however they have greater computational complexity. DMDc and HAVOK are less computationally complex and their models are suitable for linear MPC, which is significantly faster than non-linear MPC. This is desirable for a RUAV use case, where onboard computational power is limited.

3.2.2. System Identification Procedure

?? do correct numbering 1. Takeoff and hover 2. Command a series of velocity step inputs with random step sizes and time intervals 3. Measure and save input and output data 4. Apply algorithm to data and generate model

3.2.3. Dynamic Mode Decomposition with Control

DMD is a linear regression technique that is used here to approximate a non-linear dynamical system [**?**]. It only uses temporal measurements to reconstruct the system dynamics, with no prior modelling assumptions. DMDc is an adaptation of DMD that also accounts for control [**?**].

This section provides an overview of the specific implementation of DMDc used in this paper. Note that this implementation is a slight adaptation of DMDc, and includes time-delay-embedding of multiple observables. [?] and [?] use time-delay-embedding in their DMD adaptions in similar ways.

Only the non-velocity observables, $\boldsymbol{y} = \begin{bmatrix} x_Q & z_Q & \theta & \beta \end{bmatrix}^T$, of the 2D RUAV and payload system are used in this implementation. The algorithm produces a linear, discrete state-space model. Therefore discrete instances, $\boldsymbol{y}_k$, of the continuous time observable, $\boldsymbol{y}(t)$, are used, where $\boldsymbol{y}_k = \boldsymbol{y}(kT_s)$, and T_s is the sampling time of the model. Delay-coordinates (i.e. $\boldsymbol{y}_{k-1}, \boldsymbol{y}_{k-2}$, etc.) are also included in the state-space model. Hence, we define an extended state vector as:

$$\boldsymbol{a}_k = \begin{bmatrix} \boldsymbol{y}_{k-(q-1)} & \cdots & \boldsymbol{y}_{k-1} & \boldsymbol{y}_k \end{bmatrix}^T, \tag{3.2}$$

where $\boldsymbol{a}_k \in \mathbb{R}^{(n_y)(q)}$, and the subscript of $\boldsymbol{a}$ denotes the highest subscript of $\boldsymbol{y}$ in the vector. The state-space model is therefore defined as:

$$\boldsymbol{a}_{k+1} = \boldsymbol{A}\boldsymbol{a}_k + \boldsymbol{B}\boldsymbol{u}_k, \tag{3.3}$$

where $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ is the input vector at the current time-step, $\boldsymbol{A} \in \mathbb{R}^{(q \cdot n_y) \times (q \cdot n_y)}$ is the system matrix and $\boldsymbol{B} \in \mathbb{R}^{(q \cdot n_y) \times n_u}$ is the input matrix. The training data consists of discrete measurements, $\boldsymbol{y}_k$, and corresponding inputs, $\boldsymbol{u}_k$, taken at regular intervals of $\Delta t = T_s$, during a simulated flight. The training data is collected into the following matrices:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \cdots & \boldsymbol{y}_w \\ \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \cdots & \boldsymbol{y}_{w+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_q & \boldsymbol{y}_{q+1} & \boldsymbol{y}_{q+2} & \cdots & \boldsymbol{y}_{w+q-1} \end{bmatrix},$$

$$= \begin{bmatrix} \boldsymbol{a}_q & \boldsymbol{a}_{q+1} & \boldsymbol{a}_{q+2} & \cdots & \boldsymbol{a}_{w+q-1} \end{bmatrix},$$

$$\boldsymbol{X}' = \begin{bmatrix} \boldsymbol{a}_{q+1} & \boldsymbol{a}_{q+2} & \boldsymbol{a}_{q+3} & \cdots & \boldsymbol{a}_{w+q} \end{bmatrix},$$

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{u}_q & \boldsymbol{u}_{q+1} & \boldsymbol{u}_{q+2} & \cdots & \boldsymbol{u}_{w+q-1} \end{bmatrix}, \tag{3.4}$$

where w is the number of columns in the matrices, $\boldsymbol{X}'$ is the matrix $\boldsymbol{X}$ shifted forward by one time-step, and $\boldsymbol{\Upsilon}$ is a matrix of inputs. Equation (3.3) can be combined with the matrices in Equation (3.4) to produce:

$$\boldsymbol{X}' = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}\boldsymbol{\Upsilon}. \tag{3.5}$$

Note that the primary objective of DMDc is to determine the best fit $\boldsymbol{A}$ and $\boldsymbol{B}$ that approximates the observed process, given the data in $\boldsymbol{X}$, $\boldsymbol{X}'$ and $\boldsymbol{\Upsilon}$ [?]. In order to group

the unknowns into a single matrix, Equation (3.3) is manipulated into the form,

$$X' = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X \\ \Upsilon \end{bmatrix} = G\Omega, \tag{3.6}$$

where Ω contains the state and control data, and G represents the system and input matrices. A SVD is preformed on Ω resulting in: $\Omega = U\Sigma V^T$. Often, only the first p columns of U and V are required for a good approximation [?], hence the SVD is used in the truncated form:

$$\Omega \approx \tilde{U}\tilde{\Sigma}\tilde{V}^T, \tag{3.7}$$

where ˜ represents rank-p truncation. In many cases, the truncated form results in better approximations from noisy measurements than the exact form, since the effect of measurement noise is mostly captured by the truncated columns of U and V. By truncating these columns, the influence of noise in the regression problem is reduced.

Now, by combining (3.7) with the over-constrained equality in (3.6), the least-squared solution, G, can be found with:

$$G \approx X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}. \tag{3.8}$$

A and B is obtained by breaking up G into two separate matrices: $G = \begin{bmatrix} A & B \end{bmatrix}$.

Since the state vector, a, includes delay-coordinates, some matrix entries are known a priori and are independent of the dynamics. For example, the values of y_k should be mapped from their position in a_k to specific indices in a_{k+1}. Due to the least-squares fitting and coordinate transformation, DMDc will not produce these exact values in A and B. By forcing each of these matrix entries to 1 or 0, the state-prediction performance of the model is improved.

3.2.4. Hankel Alternative View Of Koopman

HAVOK is a data-driven, regression technique that provides a connection between DMD and Koopman operator theory [?, ?]. We have adapted the standard HAVOK algorithm slightly to account for the effect of control and to extract a discrete, linear model that approximates the behaviour of a controlled dynamical system. In this section, a brief overview is provided for this implementation and expansion of HAVOK.

The extracted discrete state-space model is defined as:

$$a_{k+1} = \tilde{A}a_k + \tilde{B}u_k, \tag{3.9}$$

where $\boldsymbol{a}_k$ is the state vector previously defined in Section 3.2.3, $\tilde{\boldsymbol{A}} \in \mathbb{R}^{(q \cdot n_y) \times (q \cdot n_y)}$ is the system matrix, and $\tilde{\boldsymbol{B}} \in \mathbb{R}^{(q \cdot n_y) \times n_u}$ is the input matrix. Here, $\tilde{}$ is used to differentiate these matrices from $\boldsymbol{A}$ and $\boldsymbol{B}$ used in DMDc.

The original HAVOK algorithm, developed by [?], constructs a Hankel matrix from output variables only. In order to incorporate the effect of control, an extended Hankel matrix, $\boldsymbol{\Pi}$, is created by appending a matrix of inputs to a Hankel matrix of measurements:

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{a}_q & \boldsymbol{a}_{q+1} & \boldsymbol{a}_{q+2} & \cdots & \boldsymbol{a}_{w+q-1} \\ \boldsymbol{u}_q & \boldsymbol{u}_{q+1} & \boldsymbol{u}_{q+2} & \cdots & \boldsymbol{u}_{w+q-1} \end{bmatrix}, \tag{3.10}$$

where w is the number of columns in $\boldsymbol{\Pi}$. A truncated SVD of this Hankel matrix results in following approximation:

$$\boldsymbol{\Pi} \approx \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}^T, \tag{3.11}$$

where $\tilde{}$ represents rank-p truncation. It is important to note that the model extracted by HAVOK depends on the choice of hyperparameters, p and q. The number of samples in the training data, $N_{train} = w + q - 1$, also influences the accuracy of the model.

The columns of $\tilde{\boldsymbol{V}}$ are the most significant principal components of the system dynamics [?]. This matrix, $\tilde{\boldsymbol{V}}$, can be considered to contain a time-series of the pseudo-state, $\boldsymbol{v}$, such that $\tilde{\boldsymbol{V}}^T = \begin{bmatrix} \boldsymbol{v}_q & \boldsymbol{v}_{q+1} & \cdots & \boldsymbol{v}_w \end{bmatrix}$, characterises the evolution of the actual dynamics in an eigen-time-delay coordinate system [?]. Consider the following discrete, state-space formulation:

$$\boldsymbol{v}_{k+1} = \boldsymbol{\Lambda}\boldsymbol{v}_k. \tag{3.12}$$

Recall that DMDc finds a best fit linear operator that directly maps $\boldsymbol{a}_k$ to $\boldsymbol{a}_{k+1}$. Similarly, HAVOK determines the best fit linear operator $\boldsymbol{\Lambda}$ that maps the pseudo-state $\boldsymbol{v}_k$ to $\boldsymbol{v}_{k+1}$. So, in order to setup an over-determined equality for (3.12), $\tilde{\boldsymbol{V}}^T$ is divided into two matrices:

$$\begin{aligned} \boldsymbol{V}_1 &= \begin{bmatrix} \boldsymbol{v}_q & \boldsymbol{v}_{q+1} & ... & \boldsymbol{v}_{w-1} \end{bmatrix}, \\ \boldsymbol{V}_2 &= \begin{bmatrix} \boldsymbol{v}_{q+1} & \boldsymbol{v}_{q+2} & ... & \boldsymbol{v}_w \end{bmatrix}, \end{aligned} \tag{3.13}$$

where $\boldsymbol{V}_2$ is $\boldsymbol{V}_1$ advanced a single step forward in time. The matrices from Equation (3.13) are now combined with Equation (3.12) and the best fit $\boldsymbol{\Lambda}$ is determined with the Moore-Penrose pseudoinverse:

$$\boldsymbol{V}_2 = \boldsymbol{\Lambda}\boldsymbol{V}_1 \qquad \Rightarrow \qquad \boldsymbol{\Lambda} \approx \boldsymbol{V}_1\boldsymbol{V}_1^{\dagger} \tag{3.14}$$

It can be shown from Equation (3.11) that Equation (3.12) is transformed from the eigen-time-delay coordinate system to the original coordinate system as the following:

$$
\begin{bmatrix} \boldsymbol{a}_{k+1} \\ \boldsymbol{u}_{k+1} \end{bmatrix} = (\tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}})\boldsymbol{\Lambda}(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}})^{\dagger} \begin{bmatrix} \boldsymbol{a}_k \\ \boldsymbol{u}_k \end{bmatrix}. \tag{3.15}
$$

This form is used to extract $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ from the matrix, $(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}})\boldsymbol{\Lambda}(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}})^{\dagger}$, in the following way:

$$
\begin{bmatrix} \boldsymbol{a}_{k+1} \\ \boldsymbol{u}_{k+1} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}} \\ \textit{(discarded)} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_k \\ \boldsymbol{u}_k \end{bmatrix}. \tag{3.16}
$$

Note that the matrix entries in Equation (3.16) that map $\boldsymbol{u}_k$ to $\boldsymbol{u}_{k+1}$ are meaningless for our purposes and are discarded. Similarly to DMDc, some matrix entries in $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ are known a priori due to the relative positions of delay coordinates. These are forced to 1 or 0 to improve the prediction performance of the model.

Chapter 4

Summary and Conclusion

Bibliography

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, 2012.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

Appendix A

Project Planning Schedule

This is an appendix.

Appendix B

Outcomes Compliance

This is another appendix.