CHAPTER 3
SYSTEM IDENTIFICATION THEORY

This chapter discusses on the theoretical development of System Identification (SYSID) for coriolis mass flowrate (CMF) transfer function algorithm. The important point is to describe the mathematical theory of SYSID: the parametric structure, the statistical order, and the parameter estimation method, and to verify and validate the model behavior. The SYSID theories described and discussed here has been selected from a large number of sources but is not meant to provide a comprehensive review.

3.1 Introduction

Implementation of SYSID to develop an inferential coriolis is about using experimental data to obtain mathematical model of a coriolis dynamic system. A dynamic system for coriolis is shown in Figure 3.1. From input, $u(t)$ and output, $y(t)$ sequences obtained from the experiment, a model of how the dynamic system behaves could be figured out. However, there will always be some uncertainty due to noise on the signals and disturbances acting on the system, $v(t)$. A system is dynamic, when the output of the system at a certain time is dependent in some way on the input given at a previous time.

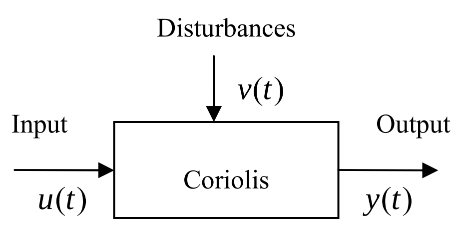

Figure 3.1: A dynamic system for coriolis [28]

A system is defined as a collection of connected components that produce observable signals which would be useless if the signals were not observable. The system would interact with environment through inputs, outputs and disturbances present in the system and the environment [28].

3.2 System Identification (SYSID)

An equivalent description of any given system could be presented in a model. The main reason for using models if compared to an actual system lies in the fact that a model is just a description of a system and not a system itself. Whereas a system might be complex, expensive or inaccessible; a corresponding model could be developed simpler using typical approximation technique that is less expensive compares to experimental work and much more mobile [28].

A model could be constructed in three ways: intuitive or verbal, graphs or tables and mathematical form. Applications solved by fuzzy logic and neural network are example of intuitive or verbal model, applications solved by bode plots and step responses are example of graphs and tables model, whilst applications solved by differential (continuous) and difference (discrete) equations are example of mathematical model. However, mathematical models are found to have exact advantages compared to others due to some reasons to be discussed in the following section [28].

If a system is unavailable, mathematical model could be used to optimize such system without requiring the presence of physical system. Different parameters and approaches could be tried on the model which makes it much more flexible than a real system. The time to scale up or down could also be changed depending on needs for time savings or time domain specification, or even to access some immeasurable quantities which might be unavailable in a real system. Furthermore, mathematical model also is safer than any hazardous system which would be possible to make training scenarios for operators under extreme conditions without taking any risks [28].

Mathematical models could be described in three forms: transfer function, state-space and block diagram which could be presented in two kinds of notations: continuous time domain and discrete time domain using Laplace transform and z-transform, respectively. These notations could be solved by two methods: physical modeling and experimental modeling which is also known as SYSID [28].

Physical modeling is a modeling which uses fundamental principles such as physical laws and relevant facts to be understood which is divided into linear and nonlinear. Whilst, SYSID is a modeling which uses experimental work to deduce system, therefore it requires prototype or real system. SYSID could be divided into nonparametric and parametric estimation methods: nonparametric is an estimation method based on step, impulse and frequency response to estimate right graphical fit of a generic model, whilst parametric is an estimation method based on user-specified models to estimate transfer functions and state-space matrices [28].

From quantitative research, there are five well-known user-specified models and one state-space approach available to identify unknown transfer function system i.e., General-Linear (GL), Autoregressive Exogeneous Input (ARX), Autoregressive Moving Average with Exogeneous Input (ARMAX), Output-Error (OE), Box-Jenkins (BJ) and state-space known as N4SID (Numerical Algorithm for Subspace State-Space). Since, numerous engineering problems have been successfully identified based on these models; the research would implement these models for designing the coriolis mass flowrate (CMF) transfer function. Notably, autoregressive in ARX and ARMAX means, previous instances of output affects current output. The classification of several reviewed literatures based on these SYSID parametric models are summarized in Table 3.1.

Table 3.1: Classification of the reviewed literature for SYSID parametric models

Classification	**References**
General Linear (GL)	Ichihashi *et al* [43], Hsieh and Rayner [44], Scott *et al.* [45], Bobet *et al.* [46], Yingli *et al.* [47], Pladdy *et al.* [48], Haifley [49], Huaien and Puthusserypady [50], Penny and Friston [51], Calhoun and Adali [52], Rong *et al.* [53], Tarnoff and Midkiff [54], Jiang *et al.* [55], Cunjun *et al.* [56], Milosavljevic *et al.* [57], Berns *et al.* [58], Manimohan and Fitzgerald [59], Rong and Herskovitz [60], Campi and Weyer [61], Xue *et al.* [62], Penney *et al.* [63], Zhang *et al.* [64], Shen *et al.* [65], Young and Jachim [66], Perttunen [67], Beckmann and Smith [68], Kect *et al* [69], Ljung [70], den Dekker *et al* [71], Soyer [72], Dodd and Haris [73], Kim [74], Gonzalves *et al* [75]
Autoregressive (ARX)	Peng *et al.* [76], Garba *et al.* [77], Kosut and Anderson [78], Tian *et al.* [79], Ohata *et al.* [80], Suzuki *et al.* [81], Monden *et al.* [82], Hashambhoy and Vidal [83], Sekizawa *et al.* [84], Frosini and Petrecca [85], Huaiyu *et al.* [86], Wei *et al.* [87], Nounou [88], Gehalot *et al.* [89], Hori *et al.* [90], Chen and Lai [91], Mosca and Zappa [92], Isaksson [93], Soderstrom *et al.* [94], Jankumas [95], Kwan and Huy [96], Rivera and Jun [97], Derbel [98], de Waele and Broersen [99], Hadjiloucas *et al* [100], Espinoza *et al* [101], Larsson *et al* [102], Elkfafi *et al* [103], Shah *et al* [104], Ling and Zhizhong [105], Kiryu *et al* [106], Suzuki and Watanabe [107], Rahiman *et al* [108], Moojun *et al* [109], Radic-Weissenfeld *et al* [110], Fukata *et al* [111], Su *et al* [112], Mossberg [113], Nasiri *et al* [114], Tanaka *et al* [115], Iwase *et al* [116], Yucai [117], van Ditzhuijzen *et al* [118], Vidal *et al* [119], Ozsoy *et al* [120]
Autoregressive Moving Average (ARMAX)	Haseyama *et al.* [121], Jinglu and Kumamaru [122], Landau and Karimi [123], Fung and Leung [124], Sakellariou and Fassois [125], Hong-Tzer *et al.* [126], Kyungno and Doo [127], Wang [128], Bore-Kuen and Bor-Sen [129], Chao-Ming *et al.* [130], Artemiadis and Kyriakopoulos [131], Song *et al.* [132], Hamerlain [133], Funaki *et al.* [134], Hong-Tzer and Chao-Ming [135], Guo and Huang [136], Haseyama and Kitajima [137], Bor-Sen *et al.* [138], Michaud *et al.* [139], Inoue *et al.* [140], Waller *et al.* [141], Jinglu *et al.* [142], Musto and Lauderbaugh [143], Timmons *et al.* [144], Grimble and Carr [145], Nassiri-Toussi and Ren [146], Irwin *et al* [147], *Wang et al* [148], Krolikowski *et al* [149], Duckgee *et al* [150], Chih-Lyang [151], Ghazy and Amin [152], Mrad *et al* [153], Bercu [154]
Output Error (OE)	Kabaila [155], Er-Wei and Yinyu [156], Kenney and Rohrs [157], Thomopoulos and Papadakis [158], Velez-Reyes and Ramos-Torres *et al.* [159], Dai and Sinha [160], Douma and Van den [161], Gustafsson and Schoukens [162], Jacobson *et al.* [163], Bhargava and Kashyap [164], Vogt *et al.* [165], Wigren and Nordsjo [166], Monin [167], Kyungno and Doo Yong [168], Mbarek *et al.* [169], Sheta and Abel-Wahab [170], Huang [171], Klauw *et al.* [172], Knyazkin *et al.* [173], Porat and Friedlander [174], Matko *et al.* [175], Doroslovacki and Fan [176], Oku *et al.* [177], Bouchard *et al.* [178], Sang Yoon and Nam Ik [179], Simon and Peceli [180], Piche [181], Roy *et al* [182], Wong [183], Ren and Kumar [184], Regalia [185], Baik and Mathews [186], Garnett *et al* [187], Duong and Landau [188]
Box Jenkins (BJ)	Chih-Chou and Chao-Ton [189], Smaoui *et al.* [190], Gersch and Brotherton [191], Tang *et al.* [192], Yu [193], Xinyao *et al.* [194], Triolo *et al.* [195], Forssell and Ljung [196], Vu *et al.* [197], Gao and Ovaska [198], Chang and Tsai [199], Yu and Chen [200], Bombois *et al.* [201], Amjady [202], Choueiki *et al.* [203], Ku-Long *et al.* [204], Matthews *et al.* [205], Chowdhury and Rahman [206], Abonyi *et al.* [207], Leski [208], Ninness and Hjalmarsson [209], Hughes [210], Vu *et al.* [211], Dimirovski and Andreeski [212], Dinda *et al* [213], Wu *et al* [214], Deacha [215], Bara [216], Yang *et al* [217], Teixeira and Zaverucha [218], Jiang *et al* [219], Gao *et al* [220], Jurado *et al* [221]
State-Space (N4SID)	Goethals *et al.* [222], Shiguo *et al.* [223], Qidwai and Bettayeb [224], Juricek *et al.* [225], Di Loreto *et al* [226], Ning Zhou *et al.* [227], Xiaorong *et al.* [228], Shi and MacGregor [229], Sima and Van Huffel [230], Fischer and Medvedev [231], Flint and Vaccaro [232], Jingbo *et al.* [233], Lieftucht *et al.* [234], Nitta [235], Lopes dos Santos *et al.* [236], Gustafsson [237], Chiuso and Picci [238], Munevar *et al.* [239], Trudnowski *et al* [240], Xianwei Zhou *et al* [241]

The articles reported in Table 3.1 have been on engineering applications such as solution for control system, communication, prediction and instrumentation applications. Generally, GL has been developed in fuzzy system, rehabilitation engineering and biomedical engineering. In brief, ARX has been used in intelligent transportation, ferroelectric control and electromagnetic compatibility, ARMAX on the other hand, has been implemented mainly in circuit design, automatic control and nuclear science. Interestingly, OE has been applied in signal processing, electrical machine and cybernetics, while BJ has been employed in artificial intelligence, neural network and adaptive process. Moreover, N4SID has been tested in nuclear science, computer aided design and nanotechnology. The SYSID algorithm adopted in this work have similarity to the methods of solving engineering problems as summarized in Table 3.1, and in fact would go further to investigate in detail the gray-box model of a coriolis flowmeter. The aim would be to implement the algorithm on a real natural gas measuring operation. Analyzing the requirement of implementing on a test rig, the algorithm would be embedded into a typical controller with an easy real-time interfacing.

3.3 SYSID parametric models

The following section describes the underlying structures about SYSID parametric models, different parametric model representations, reasons for choosing one representation over another, and how to validate the estimated models for coriolis. A coriolis system could be described using the following model [26], [27], [29], [30] and [31].

$$y(n) = q^{-k}G(q^{-1},\theta)u(n) + H(q^{-1},\theta)e(n) \tag{3.1}$$

Where $u(n)$ and $y(n)$ are the input and output of the system respectively, whilst $e(n)$ is zero-mean white noise or the disturbance to the system. White noise is a sequence of independent and identically distributed random variables of zero mean and variance, λ^2. $G(q^{-1},\theta)$ is the transfer function of the deterministic part of the system, whilst $H(q^{-1},\theta)$ is the transfer function of the stochastic part of the system, respectively [30].

The deterministic transfer function specifies the relationship between the output and the input signal, while the stochastic transfer function specifies how the output is affected by the disturbance. Some literatures refer to the deterministic and stochastic parts as system dynamics and stochastic dynamics, respectively. The term q^{-1} is the backward shift operator, which is defined by the following equation.

$$q^{-1}x(n) = x(n-1) \tag{3.2}$$

q^{-k} defines the number of delay samples between the input and the output. $G(q^{-1},\theta)$ and $H(q^{-1},\theta)$ are rational polynomials as defined by the following equations.

$$G(q^{-1},\theta) = \frac{B(q,\theta)}{A(q,\theta)F(q,\theta)} \tag{3.3}$$

$$H(q^{-1},\theta) = \frac{C(q,\theta)}{A(q,\theta)D(q,\theta)} \tag{3.4}$$

The vector θ is the set of model parameters. Equations in the following sections will not display θ to make the equations simpler and easier to read. The following equations define $A(q), B(q), C(q), D(q)$ and $F(q)$:

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + ... + a_{n_a} q^{-n_a} \tag{3.5}$$

$$B(q) = b_0 + b_1 q^{-1} + ... + b_{n_b - 1} q^{-(n_b - 1)} \tag{3.6}$$

$$C(q) = 1 + c_1 q^{-1} + c_2 q^{-2} + ... + c_{n_c} q^{-n_c} \tag{3.7}$$

$$D(q) = 1 + d_1 q^{-1} + d_2 q^{-2} + ... + d_{n_d} q^{-n_d} \tag{3.8}$$

$$F(q) = 1 + f_1 q^{-1} + f_2 q^{-2} + ... + f_{n_f} q^{-n_f} \tag{3.9}$$

Where n_a, n_b, n_c, n_d and n_f are the model orders.

3.3.1 General Linear (GL) model

Figure 3.2 depicts the signal flow of a general linear model.

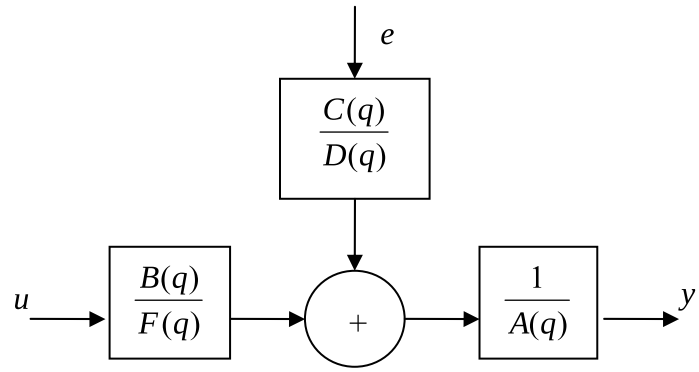

Figure 3.2: Signal Flow of GL Model [30]

A general-linear model would provide flexibility for both the system dynamics and stochastic dynamics. However, a nonlinear optimization method is required to compute the estimation of the general-linear model. The model requires intensive computation with no guarantee of global convergence. By setting one or more of $A(q), C(q), D(q)$ and $F(q)$ equal to 1, a simpler model such as ARX, ARMAX, OE and BJ model could be developed [30].

3.3.2 Autoregressive with Exogeneous Input (ARX) model

When $C(q), D(q)$ and $F(q)$ equal to 1, the general linear polynomial model transforms to an ARX model. The following equation describes an ARX model.

$$A(q)y(n) = q^{-k}B(q)u(n) + e(n) = B(q)u(n-k) + e(n) \qquad (3.10)$$

Figure 3.3 depicts the signal flow of an ARX model.

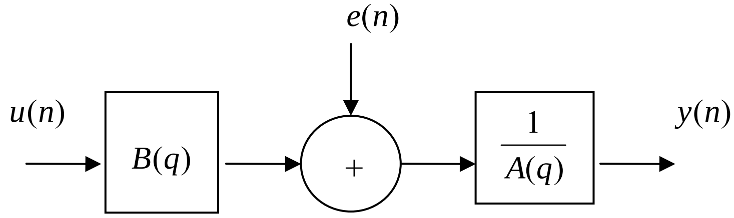

Figure 3.3: Signal Flow of ARX Model [26], [27], [28], [30]

ARX model is the simplest model that incorporates stimulus (input signal). The estimation for ARX model is the most efficient of the polynomial estimation methods because it is the result of solving linear regression equations based on analytical form. The solution estimated by ARX also is unique because the solution would always satisfy the global minimum of the loss function. The model is highly preferable, when the higher model order is needed [30].

However, the disadvantage of the ARX model is that disturbances are part of the system dynamics i.e., the transfer function of the deterministic part, $G(q^{-1},\theta)$ and the transfer function of the stochastic part, $H(q^{-1},\theta)$ have the same set of poles. The coupling would be unrealistic because the system dynamics and stochastic dynamics of the system do not share the same set of poles at all time. The disadvantage could be reduced if a signal-to-noise ratio is used. When the disturbance $e(n)$ of the system is not white noise, the coupling between the deterministic and stochastic dynamics would tend to bias the estimation of the ARX model [30].

The suitable mathematical method to identify ARX model is the least squares (LS) method i.e., a special case of the prediction error method (PEM). This is achieved by setting the model order higher than the actual model order to ensure the equation error is minimized, especially when lower signal-to-noise ratio is required. However, if the model order is increased, some dynamic characteristics of the model must be changed, such as the stability of the model [30].

3.3.3 Autoregressive Moving Average with Exogeneous Input (ARMAX) model

When $D(q)$ and $F(q)$ equal to 1, the general linear polynomial model transforms to an ARMAX model. The following equation describes an ARMAX model.

$$A(q)y(n) = q^{-k}B(q)u(n) + C(q)e(n) = B(q)u(n-k) + C(q)e(n) \qquad (3.11)$$

In the following section, Figure 3.4 depicts the signal flow of an ARMAX model.

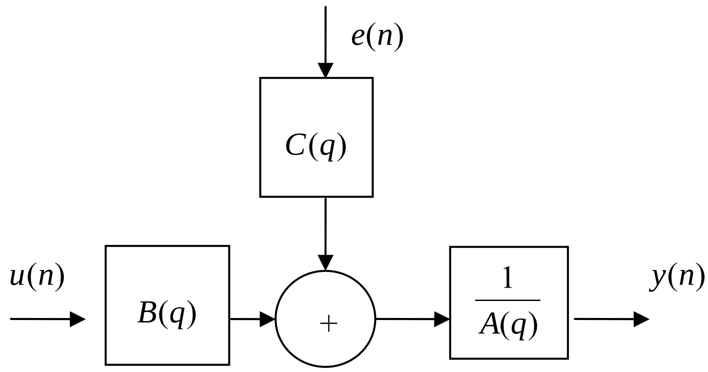

Figure 3.4: Signal Flow of ARMAX Model [26], [27], [28], [30]

Unlike the ARX model, the system structure of an ARMAX model includes disturbances dynamics. ARMAX models are useful when the dominating disturbances enter earlier in a process, such as at the input. For example, a wind gust affecting an aircraft is a dominating disturbance early in the process. The ARMAX model also has more flexibility in the handling of disturbance modeling than the ARX model. The suitable mathematical method to identify ARMAX structure is by using prediction error method, which is similar with ARX structure. However, the problem could not be solved in an analytical form; it must be solved using a computer program [30].

By developing a program, an accurate estimation could be done to search for the optimal ARMAX model based on Newton-Gauss implementation. The searching algorithm is an iterative procedure, which is sometimes inefficient and can get stuck at a local minimum, especially when the signal-to-noise ratio is low. Therefore, further validation method is needed to verify whether Newton-Gauss method could achieve required quality or estimation stuck at a local minimum. If the estimation is stuck at a local minimum, a new model structure need to be selected or new model order need to be increased [30].

3.3.4 Output Error (OE) model

When $A(q), C(q)$ and $D(q)$ equal 1, the general-linear polynomial model transforms to an output-error model. The following equation describes an output-error model.

$$y(n) = \frac{q^{-k}B(q)}{F(q)}u(n) + e(n) = \frac{B(q)}{F(q)}u(n-k) + e(n) \tag{3.12}$$

Figure 3.5 depict the signal flow of the output-error model.

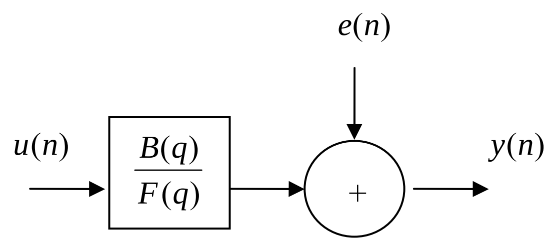

Figure 3.5: Signal Flow of OE Model [26], [27], [28], [30]

The output-error model describes the system dynamics separately and does not use any parameters for modeling the disturbance characteristics. The suitable mathematical method to identify output-error model is also the prediction error method, which is similar to ARX and ARMAX model. However, the input signal $u(n)$ must be white noise to ensure all minima are global. There is no local minimum but a local minimum could exist if the input signal is not white [30].

3.3.5 Box Jenkins (BJ) model

When $A(q)$ equals 1 the general-linear polynomial model transforms to a Box-Jenkins model. The following equation describes a Box-Jenkins model.

$$y(n) = \frac{q^{-k}B(q)}{F(q)}u(n) + \frac{C(q)}{D(q)}e(n) = \frac{B(q)}{F(q)}u(n-k) + \frac{C(q)}{D(q)}e(n) \tag{3.13}$$

In the following section, Figure 3.6 depicts the signal flow of the Box-Jenkins model.

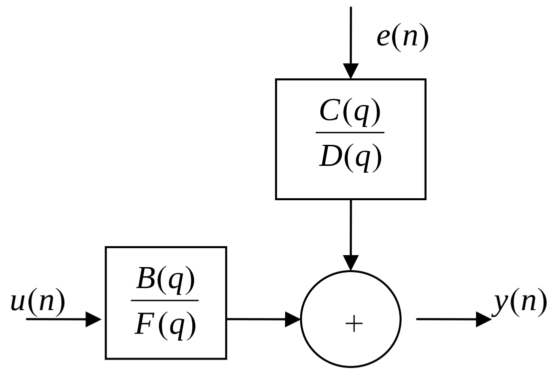

Figure 3.6: Signal Flow of BJ Model [26], [27], [28], [30]

The Box-Jenkins model provides a complete model of a system. It models disturbance properties separately from system dynamics, which is useful when disturbances enter late in the process. The suitable mathematical method to identify Box-Jenkins model is also the prediction error method i.e., similar to the ARX, ARMAX and OE models [30].

3.3.6 State-space (SS) model

In addition to parametric models, the research also determines coriolis using state-space model based on N4SID algorithm. The following describes a state-space model equation.

$$x(n+1) = Ax(n) + Bu(n) + Ke(n) \tag{3.14}$$

$$y(n) = Cx(n) + Du(n) + e(n) \tag{3.15}$$

Where $x(n)$ is the state vector, whilst A, B, C, D and K are the system matrices. The dimension of the state vector $x(n)$ is the only setting that needs to be provided for the state-space model. The state-space model describes a system based on difference equations with an auxiliary state vector, $x(n)$ which the matrices often reflect physical characteristics of a system. Hence, the state-space models are often preferable to polynomial models, especially in modern control applications. In general, the state-space model provides a more complete representation of the system than polynomial models because state-space models are similar to first principle models [30].

The previous section has described SYSID parametric models that could be used to estimate coriolis mass flowrate (CMF) transfer function. The following section discusses transformation of the models into linear difference equations with certain order.

3.4 Statistical theory of model order

The selection of model order is a step to limit the number of model orders i.e., n_a, n_b, n_c, n_d and n_f for parametric models. From prediction error standpoint, the higher the order of the model is, the better the model fits the data because the model has more degrees of freedom [30].

Higher order models also require more computation time and memory. Therefore, underestimating the system orders will result in a biased model, whilst overestimating the orders will result in high model variance. In this research, model order is chosen based on "PARSIMONY" theory.

Parsimony theory is a statistical rule that states, if there are two identifiable model structures that fit certain data, the simpler one i.e., the structure containing the smaller number of parameters will give better accuracy on average. Therefore, if the model has fitted the data well and passed verification test, the theory advocates choosing the model with the smallest degree of freedom or number of parameters. The criteria to assess the model order therefore not only must rely on prediction error but also must incorporate a penalty when the order increases [30].

To determine optimal model order, the prediction error results must be plotted as a function of model dimension, which, the minimum point from that function would determine value of optimal model order. There are three well-known criterions available for determining model order: Akaike's Information Criterion (AIC), Akaike's Final Prediction Error Criterion (FPE), and Minimum Data Length Criterion (MDL). The following section discusses each criterion respectively [26], [27], [28], [30], [31].

3.4.1 Akaike's Information Criterion (AIC)

The Akaike's Information Criterion (AIC) is a weighted estimation error based on the unexplained variation of the actual data with a penalty term when exceeding the optimal number of parameters to represent the system [30]. An optimal model is the one that minimizes the following equation

$$AIC = N \log(V_n(\hat{\theta})) + 2p \tag{3.16}$$

3.4.2 Akaike's Final Prediction Error Criterion (FPE)

Akaike's Final Prediction Error Criterion (FPE) estimates the prediction error when the model is used to predict new outputs [30]. The following equation defines the FPE criterion.

$$FPE = \left(\frac{1 + \dfrac{p}{N}}{1 - \dfrac{p}{N}} \right) MSE \tag{3.17}$$

3.4.3 Minimum Data Length Criterion (MDL)

The Minimum Data Length Criterion (MDL) is based on V_n plus a penalty for the number of terms used [30]. The following equation defines the MDL criterion.

$$MDL = V_n + \frac{p \ln N}{N} \tag{3.18}$$

For all criterions, N is the number of data points, p is the number of parameters in the model, and $V_n(\hat{\theta})$ is an index related to the prediction error or residual sum of squares

$$V_n(\hat{\theta}) = \sum_{k=1}^{N} \varepsilon^2(k) \tag{3.19}$$

where $\varepsilon(k)$ is the residual or deviation of data between actual and model output, $y(k)$ and $\hat{y}(k)$, respectively

$$\varepsilon(k) = y(k) - \hat{y}(k) \tag{3.20}$$

The previous section has discussed statistical theory that could be used to determine optimal order. The following section discusses mathematical algorithms to estimate coefficients in each SYSID parametric model.

3.5 Mathematical algorithm for coefficients of parametric models

When it comes to estimating coefficients, there are different approaches and algorithms used for GL, ARX, ARMAX, OE and BJ. In this research, the parametric models would be investigated based on three approaches i.e., non-recursive (off-line method), recursive (on-line method) and state-space [28]. However, the main attempt for all methods is to minimize the error of predicted output in relation to the actual output. The following section describes the first approach i.e., the non-recursive algorithm.

3.5.1 Non-recursive model

Non-recursive model is an estimation that identifies coriolis system based on input-output data gathered at a time prior to the current time. The following algorithm derives non-recursive model for GL model.

3.5.1.1 Non-recursive algorithm for GL model

The following algorithm is for GL single input and single output model which is described as [31].

$$A(q)y(n) = \frac{B(q)}{F(q)}u(n-k) + \frac{C(q)}{D(q)}e(n) \tag{3.21}$$

where $A(q) = 1 + \sum_{i=1}^{N_a} a_i q^{-i}$, $B(q) = \sum_{i=0}^{N_b - 1} b_i q^{-i}$, $F(q) = 1 + \sum_{i=1}^{N_f} f_i q^{-i}$, $C(q) = 1 + \sum_{i=1}^{N_c} c_i q^{-i}$,

$D(q) = 1 + \sum_{i=1}^{N_d} d_i q^{-i}$ and q is the backward shift operator, which means

$$q^{-i} y(n) = y(n-i)$$

k is the delay of the system. The purpose is to estimate the coefficients $[a_1, a_2, \ldots, a_{N_a}]$, $[b_0, b_1, \ldots, b_{N_b-1}]$, $[f_1, f_2, \ldots, f_{N_f}]$, $[c_1, c_2, \ldots, c_{N_c}]$ and $[d_1, d_2, \ldots d_{N_d}]$ based on the input-output data of a coriolis.

The multi-stage method is applied to have a coarse estimation for $A(q), B(q), F(q), C(q)$ and $D(q)$, and then the Gauss-Newton minimization method is applied to refine the results of $A(q), B(q), F(q), C(q)$ and $D(q)$. Here is the deduction based on multi-stage coarse estimation specific for GL model.

Let

$$\varepsilon(n) = F(q)\frac{C(q)}{D(q)}e(n) \tag{3.22}$$

Then equation (3.21) becomes

$$A(q)F(q)y(n) = B(q)u(n-k) - \varepsilon(n) \tag{3.23}$$

Instrumental variable (IV) method is applied to estimate $B(q)$ and $A(q)F(q)$.

Let $\varepsilon(n) = \dfrac{C(q)}{D(q)}e(n)$, then equation (3.21) becomes

$$A(q)y(n) = \frac{B(q)}{F(q)}u(n-k) + \varepsilon(n) \tag{3.24}$$

Equation (3.24) is approximated as an ARX model, whose B order is high, calculated using $(N_b + N_f + N_c + N_d)$. Then, by applying instrumental variable method, the $A(q)$ is calculated. Since $A(q)F(q)$ and $A(q)$ is known, $F(q) = (A(q)F(q)/A(q))$ could be calculated. So far, only $A(q)$, $B(q)$ and $F(q)$ is known.

By substituting them into equation (3.21), the following equation is derived

$$\hat{v}(n) = A(q)y(n) - \frac{\hat{B}(q)}{\hat{F}(q)}u(n-k) = \frac{C(q)}{D(q)}e(n) \tag{3.25}$$

Here, as elsewhere, $\hat{f}$ denotes the estimation of f. If $N_c = 0$, equation (3.24) can be rewritten as

$$D(q)\hat{v}(n) = e(n) \tag{3.26}$$

which can be treated as an AR model. With AR model estimation, then $D(q)$ is estimated. If $N_c \neq 0$, equation (3.25) is rewritten as

$$\frac{D(q)}{C(q)}\hat{v}(n) = e(n) \tag{3.27}$$

Then a high order AR model is applied to estimate $\hat{e}(n)$. Since $\hat{v}(n)$ and $\hat{e}(n)$ are known, equation (3.27) can be rewritten as

$$D(q)\hat{v}(n) = (C(q)-1)\hat{e}(n) + e(n) \tag{3.28}$$

which is a form of ARX model. Then $C(q)$ and $D(q)$ can be estimated by using ARX model estimation method with $\hat{v}(n)$ and $\hat{e}(n)$ as output and input respectively. This section has discussed non-recursive model for GL. The following section would discuss non-recursive model for ARX.

3.5.1.2 Non-recursive algorithm for ARX model

The following algorithm is for ARX single input and single output model which is described as [31].

$$A(q)y(n) = B(q)u(n-k) - e(n) \tag{3.29}$$

where $A(q) = 1 + \sum_{i=1}^{N_a} a_i q^{-i}$, $B(q) = \sum_{i=0}^{N_b - 1} b_i q^{-i}$, $u(n)$, $y(n)$ and $e(n)$ are the input, output, and disturbance of a system respectively. The purpose is to estimate the coefficients $[a_1, a_2, \ldots a_{N_a}]$ and $[b_0, b_1, \ldots, b_{N_b - 1}]$ based on the input-output data from coriolis.

Suppose the coriolis model is assumed to be higher order of difference equation models, then

$$\begin{aligned} &y(k+n) + a_{n-1}y(k+n-1) + \ldots + a_1 y(k+1) + a_0 y(k) \\ &= b_m x(k+m) + b_{m-1}x(k+m-1) + \ldots + b_1 x(k+1) + b_0 x(k) \end{aligned}$$

The equation could be rewritten by moving all terms except the most future term i.e., $y(k+n)$ of the left hand side of the equation to the right hand side. Then, by representing in matrix notations, the following equation is established.

$$y(k+n) = [-y(k+n-1), \ldots, -y(k), -x(k+m), -x(k+m-1), \ldots, -x(k)] \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_1 \\ a_0 \\ b_m \\ b_{m-1} \\ \vdots \\ b_1 \\ b_0 \end{bmatrix}$$

By writing down the equation for every $k = 1, 2, \ldots, N$, a series of equation is derived

$$\underbrace{\begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} -y(n) & -\dots & -y(1) & -x(1+m) & \dots & -x(1) \\ -y(n+1) & -\dots & -y(2) & -x(2+m) & \dots & -x(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & -\dots & -y(N-n) & -x(N) & \dots & -x(N-m) \end{bmatrix}}_{\varphi} \underbrace{\begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_1 \\ a_0 \\ b_m \\ b_{m-1} \\ \vdots \\ b_1 \\ b_0 \end{bmatrix}}_{\theta}$$

The problem is derived as $y = \varphi\theta$, which θ could be determined using regression formula [26].

$$\theta_N^{LS} = [\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^T(t)]^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)y(t) \tag{3.30}$$

where

$$\varphi(t) = [-y(t-1) \quad -y(t-2) \quad \dots \quad -y(t-N_a) \quad u(t) \quad u(t-1) \quad \dots \quad u(t-N_b+1)]^T$$

Equation (3.30) could be rewritten as the solution of the linear equations:

$$AX = Y \tag{3.31}$$

where

$$A = \begin{bmatrix} \varphi^T(p) \\ \varphi^T(p+1) \\ \vdots \\ \varphi^T(N) \end{bmatrix}, \quad X = \begin{bmatrix} a_1 \\ \vdots \\ \vdots \\ a_{N_a} \\ b_0 \\ \vdots \\ \vdots \\ b_{Nb-1} \end{bmatrix}, \quad \text{and } Y = \begin{bmatrix} y(p) \\ y(p+1) \\ \vdots \\ y(N) \end{bmatrix}$$

The previous section has discussed non-recursive model for ARX. The following section would discuss non-recursive model for ARMAX.

3.5.1.3 Non-recursive algorithm for ARMAX model

The following algorithm is for ARMAX single input and single output model which is described as [31].

$$A(q)y(n) = B(q)u(n-k) - C(q)e(n) \tag{3.32}$$

where $A(q) = 1 + \sum_{i=1}^{N_a} a_i q^{-i}$, $B(q) = \sum_{i=0}^{N_b-1} b_i q^{-i}$, $C(q) = 1 + \sum_{i=1}^{N_c} c_i q^{-i}$, $u(n)$, $y(n)$ and $e(n)$ are the input, output and disturbance of a system respectively.

The purpose is to estimate the coefficients $[a_1, a_2, \dots a_{N_a}]$, $[b_0, b_1, \dots, b_{N_b-1}]$ and $[c_1, c_2, \dots, c_{N_c}]$ based on the input-output data of coriolis system. The multi-stage method is applied to have a coarse estimation for $A(q), B(q),$ and $C(q)$, and then the Gauss-Newton minimization method is applied to refine the results of $A(q), B(q),$ and $C(q)$. Here is the deduction based on multi-stage coarse estimation specific for ARMAX model.

Let

$$v(t) = C(q)e(t) \tag{3.33}$$

Then equation (3.32) becomes

$$A(q)y(t) = B(q)u(t-k) + v(t) \tag{3.34}$$

Since $v(t)$ here is not white Gaussian noise, the instrumental variable (IV) method is applied to estimate $A(q)$ and $B(q)$. The following equation is derived

$$\hat{v}(t) = \hat{A}(q)y(t) - \hat{B}(q)u(t-k) = C(q)e(t) \tag{3.35}$$

Equation (3.35) can be rewritten as

$$\frac{1}{C(q)}\hat{v}(t) = e(t) \tag{3.36}$$

which can be treated as high order AR model. (Theoretically, it is an infinite order AR model. Practically, the dimension of the system $N_a + N_b + N_c$ is selected). With the AR model estimation, $e(t)$ can be estimated. Since the $\hat{v}(t)$ and $\hat{e}(t)$ are known, equation (3.35) can be rewritten as

$$\hat{v}(t) = (C(q)-1)\hat{e}(t) + e(t) \tag{3.37}$$

which is a form of ARX model. Then, $C(q)$ can be estimated by using ARX model estimation with $\hat{v}(t)$ and $\hat{e}(t)$ as output and input respectively. This section has discussed non-recursive model for ARMAX. The following section would discuss non-recursive model for OE.

3.5.1.4 Non-recursive algorithm for OE model

The following algorithm is for OE single input and single output model which is described as [31].

$$y(n) = \frac{B(q)}{F(q)}u(n-k) + e(n) \tag{3.38}$$

where, $B(q) = \sum_{i=0}^{N_b-1} b_i q^{-i}, F(q) = 1 + \sum_{i=1}^{N_f} f_i q^{-i}, u(n), y(n)$ and $e(n)$ are the input, output, and disturbance of a system respectively.

The purpose is to estimate the coefficients $[b_0, b_1, \ldots, b_{N_b-1}]$ and $[f_1, f_2, \ldots, f_{N_f}]$ based on the input-output data of coriolis system.

The multi-stage method is applied to have a coarse estimation for $B(q)$ and $F(q)$, and then the Gauss-Newton minimization method is applied to refine the results of $B(q)$ and $F(q)$. Here is the deduction based on multi-stage coarse estimation specific for OE model.

Let

$$v(t) = F(q)e(t) \tag{3.39}$$

Then equation (3.38) becomes

$$F(q)y(t) = B(q)u(t-k) + v(t) \tag{3.40}$$

It is obvious that equation (3.40) is in the form of ARX model. Therefore, instrumental variable method can be applied to estimate $F(q)$ and $B(q)$. This section has discussed non-recursive model for OE. The following section would discuss non-recursive model for BJ.

3.5.1.5 Non-recursive algorithm for BJ model

The following algorithm is for BJ single input and single output model which is described as [31].

$$y(n) = \frac{B(q)}{F(q)}u(n-k) + \frac{C(q)}{D(q)}e(n) \tag{3.41}$$

The purpose is to estimate the coefficients $[b_0, b_1, \ldots b_{N_b-1}], [f_1, f_2, \ldots f_{N_f}], [c_1, c_2, \ldots c_{N_c}]$ and $[d_1, d_2, \ldots, d_{N_d}]$ based on the input-output data of coriolis system. The multi-stage method is applied to have a coarse estimation for $B(q), F(q), C(q)$ and $D(q)$ and the Gauss-Newton minimization method is applied to refine the results of $B(q), F(q), C(q)$ and $D(q)$. The following section is the deduction based on multi-stage coarse estimation specific for BJ model.

Let

$$\varepsilon(n) = F(q)\frac{C(q)}{D(q)}e(n) \tag{3.42}$$

Then equation (3.41) becomes

$$F(q)y(n) = B(q)u(n-k) - \varepsilon(n) \tag{3.43}$$

Instrumental variable method is applied to estimate $B(q)$ and $F(q)$ and have

$$\hat{v}(n) = y(n) - \frac{\hat{B}(q)}{\hat{F}(q)}u(n-k) = \frac{C(q)}{D(q)}e(n) \tag{3.44}$$

Here, $\hat{f}$ denotes the estimation of f. If $N_c = 0$, equation (3.44) can be rewritten as

$$D(q)\hat{v}(n) = e(n) \tag{3.45}$$

which can be treated as an AR model. With the AR model estimation, the $D(q)$ can be estimated. If $N_c \neq 0$, equation (3.44) can be rewritten as

$$\frac{D(q)}{C(q)}\hat{v}(n) = e(n) \tag{3.46}$$

Then, a high order AR model is applied to estimate $\hat{e}(n)$. Since the $\hat{v}(n)$ and $\hat{e}(n)$ are known, equation (3.46) can be rewritten as

$$D(q)\hat{v}(n) = (C(q)-1)\hat{e}(n) + e(n) \tag{3.47}$$

which is a form of ARX model. Then $C(q)$ and $D(q)$ can be estimated by using ARX model estimation method with $\hat{v}(n)$ and $e(n)$ as output and input respectively.

The previous sections have discussed the non-recursive algorithm for coriolis model based on parametric structure i.e., GL, ARX, ARMAX, OE and BJ. The following section would discuss estimation based on AR model.

3.5.1.6 AR Estimation Method

The AR estimation method will be used to refine the estimation of GL, ARMAX and BJ model. The AR model i.e., also known as Auto-Regression model is defined as

$$A(q)y(n) = e(n) \tag{3.48}$$

Where $A(q) = 1 + \sum_{i=1}^{N_a} a_i q^{-i}$, $e(n)$ is white noise, and $y(n)$ is the signal. q is the backward shift operator, which means

$$q^{-i} y(n) = y(n-i)$$

The purpose is to estimate $A(q)$ given the signal $y(t)$ so that equation (3.48) holds.

There are five algorithms that could be used to estimate $A(q)$ i.e., Least-Square (LS), Forward-Backward (FB), Yule-Walker (YL), Principle Component Analysis (PC) and Burg method [31]. Notably, this research would only use FB algorithm for refining final estimation of GL, ARMAX and BJ model.

AR coefficients can be estimated by solving the linear equations

$$\begin{bmatrix} M_f \\ M_b \end{bmatrix} a = -\begin{bmatrix} m_f \\ m_b \end{bmatrix} \tag{3.49}$$

Where

$$M_f = \begin{bmatrix} y(n_a - 1) & y(n_a - 2) & \cdots & y(0) \\ y(n_a) & y(n_a - 1) & \cdots & y(1) \\ \vdots & \vdots & \cdots & \vdots \\ y(N-2) & y(N-3) & \cdots & y(N - n_a + 1) \end{bmatrix}$$

$$M_b = \begin{bmatrix} y(1) & y(2) & \cdots & y(n_a) \\ y(2) & y(3) & \cdots & y(n_a + 1) \\ \vdots & \vdots & \cdots & \vdots \\ y(N - n_a) & y(N - n_a + 1) & \cdots & y(N-1) \end{bmatrix}$$

$$m_f = \begin{bmatrix} y(n_a) \\ y(n_a + 1) \\ \vdots \\ y(N-1) \end{bmatrix}, \quad m_b = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N - n_a - 1) \end{bmatrix}, \quad a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_a} \end{bmatrix}$$

This section has discussed the AR estimation method using Forward-Backward algorithm for estimating parametric structure i.e., GL, ARMAX and BJ. The following section would discuss the Gauss-Newton Minimization method.

3.5.1.7 Gauss-Newton Minimization Method

The Gauss-Newton minimization will be used to refine the estimation of ARMAX, OE, BJ and GL model estimation [31].

The purpose of the polynomial model estimation i.e., ARMAX, OE, BJ, and GL model is to identify the polynomial coefficients $A(q), B(q), F(q), C(q)$ and $D(q)$ based on the input-output data of the coriolis system.

For convenience, all the coefficients to be estimated are combined together as a vector, i.e., θ. Normally, a coarse estimation is calculated for θ, using a multi-stage method.

And then, the following iteration is applied to refine θ.

$$\theta^{(i+1)} = \theta^{(i)} + \alpha f(\theta^{(i)}) \tag{3.50}$$

where α is the step size and $f(\theta^{(i)})$ is the search direction.

The purpose of the iteration is to minimize

$$V_N(\theta) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\varepsilon^2(t,\theta) \tag{3.51}$$

where $\varepsilon(t,\theta)$ is the prediction error $\hat{y}(t,\theta) - y(t,\theta)$, i.e., the difference between the measured output and predicted output of the system. So, the problem is how to select and compute search direction for different polynomial model [31]. The Gauss-Newton minimization defines the search direction $f(\theta)$.

$$f(\theta) = -[V''(\theta)]^{-1}V'(\theta) \tag{3.52}$$

where

$$V'(\theta) = -\frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta)\varepsilon(t,\theta) \tag{3.53}$$

and

$$V''(\theta) = \frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta)\psi^T(t,\theta) \tag{3.54}$$

$\psi(t,\theta)$ is the gradient vector of θ and T denotes matrix transposition. The computation of $\psi(t,\theta)$ will be discussed later. By inserting equation (3.53) and (3.54) to (3.52), it yields

$$f(\theta) = \frac{\sum_{t=1}^{N} \psi(t,\theta)\varepsilon(t,\theta)}{\sum_{t=1}^{N} \psi(t,\theta)\psi^T(t,\theta)}$$

$$= \frac{[\psi(1,\theta) \quad \psi(2,\theta) \quad \cdots \quad \psi(N,\theta)] \begin{bmatrix} \varepsilon(1,\theta) \\ \varepsilon(2,\theta) \\ \cdots \\ \varepsilon(N,\theta) \end{bmatrix}}{[\psi(1,\theta) \quad \psi(2,\theta) \quad \cdots \quad \psi(N,\theta)] \begin{bmatrix} \psi^T(1,\theta) \\ \psi^T(2,\theta) \\ \cdots \\ \psi^T(N,\theta) \end{bmatrix}}$$

$$= \begin{bmatrix} \psi^T(1,\theta) \\ \psi^T(2,\theta) \\ \cdots \\ \psi^T(N,\theta) \end{bmatrix}^{-1} \begin{bmatrix} \varepsilon(1,\theta) \\ \varepsilon(2,\theta) \\ \cdots \\ \varepsilon(N,\theta) \end{bmatrix} \tag{3.55}$$

So $f(\theta)$ can be evaluated by solving the linear equations:

$$\begin{bmatrix} \psi^T(1,\theta) \\ \psi^T(2,\theta) \\ \cdots \\ \psi^T(N,\theta) \end{bmatrix} f(\theta) = \begin{bmatrix} \varepsilon(1,\theta) \\ \varepsilon(2,\theta) \\ \cdots \\ \varepsilon(N,\theta) \end{bmatrix} \tag{3.56}$$

Gradient of ARMAX model [31]

$$\psi(t,\theta) = \frac{1}{C(q)}[-y(t-1)\ldots - y(t-N_a) \quad u(t-0)\ldots u(t-N_b+1)$$

$$\varepsilon(t-1)\ldots\varepsilon(t-N_c)]^T$$

Gradient of OE model [31]

$$F(q)\psi(t,\theta) = \varphi^T(t,\theta)$$

Gradient of BJ model [31]

$$\psi(t,\theta) = [\frac{D(q)}{C(q)F(q)}q^{-k}u(t), \frac{-D(q)}{C(q)F(q)}q^{-k}w(t),$$

$$\frac{1}{C(q)}q^{-k}\varepsilon(t), \frac{-1}{C(q)}q^{-k}v(t)], t = 1,\ldots,N$$

Gradient of GL model [31]

$$\psi(t,\theta) = [\frac{-D(q)}{C(q)}q^{-k}y(t), \frac{D(q)}{C(q)F(q)}q^{-k}u(t), \frac{-D(q)}{C(q)F(q)}q^{-k}w(t),$$

$$\frac{1}{C(q)}q^{-k}\varepsilon(t), \frac{-1}{C(q)}q^{-k}v(t)], t = 1,\ldots,N$$

This section has discussed the Gauss-Newton Minimization method used for parametric structure i.e., GL, ARMAX, OE and BJ. The following section would discuss the Instrumental Variable Method.

3.5.1.8 Instrumental-Variable Method

The Instrumental-Variable method will be used to identify ARX model in GL, ARMAX, OE, and BJ model estimation. The ARX model is given as [31].

$$A(q)y(n) = B(q)u(n-k) - e(n)$$

If the noise $e(n)$ is uncorrelated to the regression variables, $y(n)$ and $u(n)$, the coefficients of the model $A(q)$ and $B(q)$ can be estimated with least square regression method. However, if the noise is correlated to $y(n)$ and $u(n)$, then the least square regression method does not work well. Therefore, the instrumental variable method described in this section could be used to solve this kind of least square problem. The instrumental variable method suggests the solution for the ARX model as

$$\theta_N^{LS} = [\frac{1}{N}\sum_{t=1}^{N}\zeta(t)\varphi^T(t)]^{-1}\frac{1}{N}\sum_{t=1}^{N}\zeta(t)y(t) \tag{3.57}$$

Where ζ contains the instruments variables. For the open loop case, the input sequence of the system, or its filtered version, is often a good choice of instrumental variables. One of the choices of instrumentals variables is

$$\zeta = [-x(t-1) - x(t-2)\ldots - x(t-N_a)\ u(t)u(t-1)\ldots u(t-N_b+1)]^T \tag{3.58}$$

where $x(t)$ is generated from the input through a linear system

$$cN(q)x(t) = M(q)u(t) \tag{3.59}$$

where

$$N(q) = 1 + n_1 q^{-1} + n_2 q^{-2} + \ldots + n_{n_N} q^{-n_N} \tag{3.60}$$

$$M(q) = 1 + m_1 q^{-1} + m_2 q^{-2} + \ldots + m_{m_N} q^{-m_N} \tag{3.61}$$

In practical, the least square method is applied to estimate coefficients of ARX model. And then, the estimated $A(q)$ and $B(q)$ is used as $N(q)$ and $M(q)$ respectively, to have an instrumental variable estimation of the ARX model with equation (3.57).

The previous section has discussed non-recursive algorithm used for estimating coefficients of SYSID parametric models. The following section discusses the second approach i.e., recursive algorithm.

3.5.2 Recursive model

Figure 3.7 represents a general diagram for recursive system identification [30]. A recursive system identification application consists of actual coriolis system, that has an input signal i.e., stimulus signal $u(n)$ and an output signal i.e., response signal, $y(n)$.

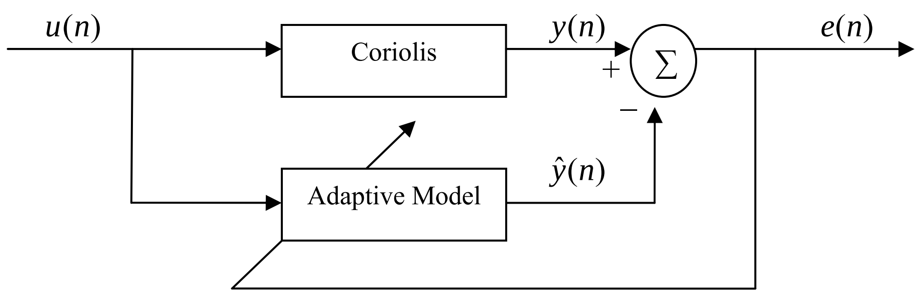

Figure 3.7: Diagram of recursive system identification [30]

The input signal $u(n)$ is the input to both the coriolis system and the adaptive model [1]. The response of the system $y(n)$ and the predicted response of the adaptive model $\hat{y}(n)$ are combined to determine the error of the system. The error of the system is defined by the following equation.

$$e(n) = y(n) - \hat{y}(n) \tag{3.62}$$

The adaptive model generates the predicted response $\hat{y}(n+1)$ based on $u(n+1)$ after adjusting the parametric vector $\vec{w}(n)$ based on the error $e(n)$. Figure 3.7 also shows how the error information $e(n)$ is sent back to the adaptive model, which adjusts the parametric vector $\vec{w}(n)$ to account for the error. The process is iterated until the magnitude of the least mean square is minimized.

Before applying the recursive model estimation, the parametric model structure i.e., GL, ARX, ARMAX, OE and BJ that determines the parametric vector $\vec{w}(n)$ need to be selected first [30]. Then, a recursive method is selected to automatically adjust the parametric vector such that the error , $e(n)$ read the minimum.

There are four types of adaptive algorithms that could be used in recursive model estimation: Least Mean Square (LMS), Normalized Least Mean Square (NLMS), Recursive Least Squares (RLS) and Kalman Filter (KF). However, only RLS algorithm is used in this research for adjusting the parametric vector $\vec{w}(n)$ to the minimum cost function.

The following equation defines the cost function, $J(n)$.

$$J(n) = E[e^2(n)] \tag{3.63}$$

When the cost function $J(n)$ is sufficiently small, the parametric vector $\vec{w}(n)$ is considered optimal for the estimation of the coriolis.

The modified cost function for RLS is given as [30]

$$J(n) = E[e^2(n)] \cong \frac{1}{N}\sum_{i=0}^{N-1} e^2(n-i) \tag{3.64}$$

which is more robust compare to previous equation (3.63), because it includes previous N error terms.

The parameter vector $\vec{w}(n)$ is initialized by using a small positive number ε as below

$$\vec{w}(0) = [\varepsilon, \quad \varepsilon, \quad \ldots, \quad \varepsilon]^T \tag{3.65}$$

Then, the data vector $\vec{\varphi}(n)$ is initialized.

$$\vec{\varphi}(0) = [0, \quad 0, \quad \ldots, \quad 0]^T \tag{3.66}$$

Next, the $n \times n$ matrix $P(0)$ is represented as below

$$P(0) = \begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \varepsilon \end{bmatrix} \tag{3.67}$$

For $n = 1$, the data vector $\vec{\varphi}(n)$ is updated based on $\vec{\varphi}(n-1)$ and the current input data $u(n)$ and output data $y(n)$. The predicted response $\hat{y}(n)$ is computed by using the following equation.

$$\hat{y}(n) = \vec{\varphi}^T(n) \cdot \vec{w}(n) \tag{3.68}$$

Then, the error $e(n)$ is computed by solving the following equation

$$e(n) = y(n) - \hat{y}(n) \tag{3.69}$$

The gain vector $\vec{K}(n)$ is updated, which is defined as equation below

$$\vec{K}(n) = \frac{P(n) \cdot \vec{\varphi}(n)}{\lambda + \vec{\varphi}^T(n) \cdot P(n) \cdot \vec{\varphi}(n)} \tag{3.70}$$

The properties of system might vary with time, so the algorithm needs to be ensured that it tracks the variation. The forgetting factor, λ method could be introduced, which is an adjustable parameter to track the variations. The smaller the forgetting factor, λ, the less previous information the algorithm would use. When a small forgetting factor is used, the adaptive model would be able to track time-varying systems that vary rapidly.

The range of forgetting factor λ is between zero and one, typically 0.98 < λ < 1. $P(n)$ is a $n \times n$ matrix whose initial value is defined by $P(0)$ in equation (3.67).

Next, the parameter vector $\vec{w}(n+1)$ is updated.

Then, the $P(n)$ matrix is updated as

$$P(n+1) = P(n) - \vec{K}(n) \cdot \vec{\varphi}^T(n) \cdot P(n) \tag{3.71}$$

The iteration is stopped if the error compared to actual coriolis is small enough, or else, n is increased by $n = n + 1$, and steps from equation (3.68) to (3.71) is repeated.

This section has discussed the recursive estimation for parametric structure i.e., GL, ARX, ARMAX, OE and BJ for coriolis using an adaptive RLS algorithm. The following section would discuss the third approach i.e., state-space model using N4SID algorithm.

3.5.3 State-space model using N4SID algorithm

The state-space model could be represented as [31].

$$\dot{x}_t = Ax_t + Bu_t + Ke_t$$
$$y_t = Cx_t + Du_t + e_t \tag{3.72}$$

The state-space of N4SID is solved in three parts: 1) estimation of matrix A, B, C, D, 2) estimation of disturbance dynamics Kalman gain, K and 3) estimation of initial states X_0, based on the input and response data sequence u_t and y_t.

The state-space model could be represented in linear regression form

$$\begin{bmatrix} \dot{x}_t \\ y_t \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} K \\ I \end{bmatrix} e_t \tag{3.73}$$

And the parameters matrix could be solved using least squares criterion approach

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \min\left(\left\| \begin{bmatrix} qX_t \\ Y_t \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_t \\ U_t \end{bmatrix} \right\| \right) = \begin{bmatrix} qX_t \\ Y_t \end{bmatrix} \left[X_t^T, U_t^T \right] \left(\begin{bmatrix} X_t \\ U_t \end{bmatrix} \left[X_t^T, U_t^T \right] \right)^{\#} \tag{3.74}$$

Where $^{\#}$ denotes the pseudo inverse [242].

From the above equation, it shows that the system matrix could be estimated if the state sequence X_t is estimated. The following section explains the first part of N4SID i.e., the estimation of system matrix $A, B, C, D,$.

3.5.3.1 Estimation of system matrix

The system is estimated using past and future Hankel data matrices as defined below [31].

$$Y_f = \begin{bmatrix} y_t & y_{t+1} & \cdots & y_{t+N-1} \\ qy_t & qy_{t+1} & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ q^{f-1}y_t & q^{f-1}y_{y+1} & \cdots & q^{f-1}y_{t+N-1} \end{bmatrix}, \quad U_f = \begin{bmatrix} u_t & u_{t+1} & \cdots & u_{t+N-1} \\ qu_t & qu_{t+1} & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ q^{f-1}u_t & q^{f-1}u_{t+1} & \cdots & q^{f-1}u_{t+N-1} \end{bmatrix}$$

$$Z_p = \begin{bmatrix} q^{-1}y_t & q^{-1}y_{t+1} & \cdots & q^{-1}y_{t+N-2} & q^{-1}y_{t+N-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ q^{-p}y_t & q^{-p}y_{t+1} & \cdots & q^{-p}y_{t+N-2} & q^{-p}y_{t+N-1} \\ q^{-1}u_t & q^{-1}u_{t+1} & \cdots & q^{-1}u_{t+N-2} & q^{-1}u_{t+N-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ q^{-p}u_t & q^{-p}u_{t+1} & \cdots & q^{-p}u_{t+N-2} & q^{-p}u_{t+N-1} \end{bmatrix} \tag{3.75}$$

Where p is the past horizon, f is the future horizon. By iterating the system equation, it is straightforward to get the extended model

$$Y_f = \Gamma_f X + H_f U_f + G_f E_f \tag{3.76}$$

Where Γ_f is the extended system observability matrix.

$$\Gamma_f = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{f-1} \end{bmatrix}, \quad H_f = \begin{bmatrix} D & 0 & 0 & 0 & 0 \\ CB & D & 0 & 0 & 0 \\ CAB & CB & D & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{f-2}B & \cdots & \cdots & CB & D \end{bmatrix}$$

$$G_f = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ CK & I & 0 & 0 & 0 \\ CAK & CK & I & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{f-2}K & \cdots & \cdots & CK & I \end{bmatrix} \tag{3.77}$$

The Kalman state X is unknown, but actually it could be calculated from past input and output data, as following

$$X = \left[L_u, L_y\right]\begin{bmatrix} U_p \\ Y_p \end{bmatrix} = L_z Z_p \tag{3.78}$$

Therefore, the equation (3.76) would be

$$Y_f = \Gamma_f L_z Z_p + H_f U_f + G_f E_f \tag{3.79}$$

U_f is projected out by multiplying $\prod_{U_f}^{\perp}$ and noise E_f is removed by multiplying Z_p^T,

$$Y_f \prod_{U_f}^{\perp} Z_p^T = \beta_z Z_z \prod_{U_f}^{\perp} Z_p^T \tag{3.80}$$

where $\beta_z = \Gamma_f L_z$.

β_z can be estimated using the QR decomposition equivalent for $[U_f^T, Z_p^T, Y_f^T]$.

$$[U_f^T, Z_p^T, Y_f^T] = [Q_1, Q_2, Q_3]\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix} \tag{3.81}$$

$$\beta_z = R_{23}^T R_{22} [R_{22}^T R_{22}]^{\uparrow} \tag{3.82}$$

After obtaining the β_z, Γ_f could be estimated by performing SVD (Singular Value Decomposition) on β_z weighted with Z_p.

$$\beta_z Z_p \overset{SVD}{=} USV^T$$

$$\hat{\Gamma}_f = U_1 \tag{3.83}$$

Since β_z and $\hat{\Gamma}_f$ are found, the state sequence X_t could be estimated by solving $\beta_z Z_p = \Gamma_f X_t$. That is

$$X_t = \left(\hat{\Gamma}_f^T \hat{\Gamma}_f\right)^{-1} \hat{\Gamma}_p^T \hat{\beta} Z_p \qquad (3.84)$$

And also, qX_t can be estimated similarly. With the Y_t, U_t, X_t and qX_t, the equation (3.74) could be solved to obtained A, B, C and D. The following section explains the second part of N4SID i.e., the estimation of Kalman gain, K.

3.5.3.2 Estimation of Kalman gain K

The estimation is given as [31].

$$Y_f \prod\nolimits_{\begin{bmatrix} Z_p \\ U_f \end{bmatrix}}^{\perp} = G_f E_f \qquad (3.85)$$

The following equation can be derived

$$R_{33}^T Q_3^T = G_f E_f \qquad (3.86)$$

Where $R_{33}^T Q_3^T$ is obtained from equation (3.81). Then, the innovation process is transformed into unit variance white sequence, e_1, i.e.,

$$e(t) = Fe_1(t) \qquad (3.87)$$

$$G_f E_f = \left(G_f \otimes F\right) E_{1f} = G_f^* E_{1f} \qquad (3.88)$$

$$G_f^* E_{1f} = R_{33}^T Q_3^T \qquad (3.89)$$

Because Q_3^T is an orthonormal matrix, $E_{1f} = Q_3^T$ is chosen, then

$$G_f^* = R_{33}^T \qquad (3.90)$$

Kalman gain K, actually KF, can be identified from the first block column of G_f^* denotes as G_{f1}^*,

$$G_{f1}^* = \begin{bmatrix} F \\ CKF \\ \vdots \\ CA^{f-2}F \end{bmatrix} = \begin{bmatrix} I_{m\times m} & 0 \\ 0 & \Gamma_{(1:m(f-1),:)} \end{bmatrix} \begin{bmatrix} F \\ KF \end{bmatrix} \tag{3.91}$$

KF and F can be estimated using least squares, which then leads to

$$\hat{K} = \left(\hat{K}F\right)\hat{F}^{-1} \tag{3.92}$$

The following algorithm explains the third part of N4SID i.e., the estimation of initial states, X_0.

3.5.3.3 Estimation of initial states X_0

When the equation $Y_f = \Gamma_f X + H_f U_f + G_f E_f$ is at $t = 0$, it could be represented as

$$Y_{f0} = \Gamma_f X_0 + H_f U_{f0} + G_f^* E_{1f0} \tag{3.93}$$

Y_{f0} is computed first for zero initial states to obtain $H_f U_{f0}$ by processing the input data through the estimated system with $X_0 = [0, \ \ldots, \ 0]$.

By removing $H_f U_{f0}$ from the double sides of the above equation, the following equation is derived

$$(G_f^*)^{-1}(Y_{f0} - H_f U_{f0}) = (G_f^*)^{-1}\Gamma_f X_0 + E_{1f0} \tag{3.94}$$

Then, X_0 is estimated using least squares method.

Sections 3.5.1 to 3.5.3 have been to describing three main algorithms to determine coefficients of SYSID parametric models: non-recursive, recursive and state-space. The following section describes mathematical theory to validate estimated model for CMF.

3.6 Validation of models

There are three approaches to validate the predicted model: 1) By using model simulation to understand the underlying dynamic relationship between the model inputs and outputs, 2) By using model prediction to test the ability of the model to predict the response of the system using past input and output data, and 3) By using model residual analysis to test the whiteness of the prediction error and the independency between the prediction error and the input signal using statistical techniques [30]. However, the first method is chosen in this research due to analyze error for estimated CMF model compares to actual coriolis. The following section discusses the first step based on approach (1) i.e., transforming linear difference equation to discrete transfer function.

3.6.1 Discrete Transfer Function

The discrete transfer function of parametric models i.e., GL, ARX, ARMAX, OE and BJ could be calculated by multiplying, q^m on the numerator and denominator part of its coefficients.

Example, if the transfer function of GL model is

$$G_i(q) = \frac{B_i(q)}{A(q)F_i(q)} = \frac{a_1 q^{-1} + \cdots + a_m q^{-m}}{1 + b_1 q^{-1} + \cdots + b_m q^{-m}} \tag{3.95}$$

The discrete transfer function could be determined by multiplying $G_i(q)$ with q^m

$$G_i(q) = \frac{a_1 q^{m-1} + \cdots + q_m}{q^m + b_1 q^{m-1} + \cdots b_m} \tag{3.96}$$

For the state-space model, the discrete transfer function could be determined by finding Z -transform of the state-space first.

The state-space model is represented as

$$x(t+1) = Ax(t) + Bu(t) + Ke(t)$$
$$y(t) = Cx(t) + Du(t) + e(t) \tag{3.97}$$

By performing the Z-transform on the state-space equations

$$\begin{aligned}(zI)X(z) &= AX(z) + BU(z) + KE(z)\\ Y(z) &= CX(z) + DU(z) + E(z)\end{aligned} \tag{3.98}$$

The following equation is derived

$$X(z) = (zI - A)^{-1}[BU(z) + KE(z)] \tag{3.99}$$

Then

$$G(z) = \frac{Y(z)}{U(z)} = C(zI - A)^{-1}B + D \tag{3.100}$$

$$H(z) = \frac{Y(z)}{U(z)} = C(zI - A)^{-1}K + I \tag{3.101}$$

Further

$$C(zI - A)^{-1}B = \frac{\det(zI - A + BC) - \det(zI - A)}{\det(zI - A)} \tag{3.102}$$

and $\det(zI - A)$ is the characteristic polynomial of matrix A. The equivalent transfer function representation for the determinant part can be presented as follow

$$G(z) = C(zI - A)^{-1}B + D = \frac{\det(zI - A + BC) - \det(zI - A)(D-1)}{\det(zI - A)} \tag{3.103}$$

Whilst the equivalent transfer function for noise part is

$$H(z) = C(zI - A)^{-1}K + I = \frac{\det(zI - A + KC) + \det(zI - A)(I-1)}{\det(zI - A)} \tag{3.104}$$

From the discrete transfer function, a series of predicted data in time-domain could be developed using power series expansion method which is discussed in the following section.

3.6.2 Power Series Expansion

The discrete transfer function

$$H(z) = \frac{b_0 z}{a_2 z^2 + a_1 z + a_0} \tag{3.105}$$

the power series expansion is

$$a_2 z^2 + a_1 z + a_0 \overset{\displaystyle E(z)}{\overline{\big)\, b_0 z}} \tag{3.106}$$

Where $E(z) = E_1 z^{-1} + E_2 z^{-2} + E_3 z^{-3} + \ldots + E_n z^{-n}$ i.e., $E(z)$ is the discrete values [251]. By determining inverse Z -transform of $E(z)$, the time-domain of predicted model is

$$e(nT) = E_1 \delta(t - T) + E_2 \delta(t - 2T) + E_3 \delta(t - 3T) + \ldots, \quad n = 0,1,2,\ldots \tag{3.107}$$

Which, T is the sampling period. If sampling period is 1 s, then

$$e(k) = E_1 \delta(t - 1) + E_2 \delta(t - 2) + E_3 \delta(t - 3) + \ldots, \quad k = 0,1,2,\ldots \tag{3.108}$$

If the discrete transfer function is

$$H(z) = \frac{b_0 + b_1 z + \ldots b_{m-1} z^{m-1} + b_m z^m}{a_0 + a_1 z + \ldots a_{n-1} z^{n-1} + a_n z^n} \tag{3.109}$$

Then, the stability could be checked from zeroes and poles location such as

$$H(z) = \frac{k(z - Z_1)(z - Z_2)\ldots(z - Z_m)}{(z - P_1)(z - P_2)\ldots(z - P_n)} \tag{3.110}$$

Which Z_m, P_n and k are zeroes, poles and gain, respectively. If zeroes and poles are inside the unit circle of discrete plane, predicted model is considered stable and optimal. The detail analysis for stability could be referred from [243], [244], [245], [246], [247] and [248].

Section 3.4 to 3.6 has shown SYSID requires different mathematical theories and algorithms to estimate parametric models. To simplify estimation process, LabVIEW software version 7.2 and LabVIEW modules such as system identification module, simulation module and control design module are combined to develop LabVIEW programs for estimating discrete CMF transfer function using non-recursive, recursive and state-space of SYSID [30], [31], [250], [251] and [252].

3.7 Summary

This chapter discusses the theoretical development of SYSID for the coriolis mass flowrate (CMF) transfer function algorithm. SYSID is solved using parametric method i.e., based on user-specified models and state-space approach, namely General-Linear (GL), Autoregressive Exogeneous Input (ARX), Autoregressive Moving Average with Exogeneous Input (ARMAX), Output-Error (OE), Box-Jenkins (BJ) and state-space approach known as N4SID (Numerical Algorithm for Subspace State-Space). The parametric structure consists of several procedures i.e., statistical order, parameter estimation and validation. Statistical order is a step to limit the number of model orders where the higher the order of the model is, the better the model fits the data since the model has more degrees of freedom. Three criterions are used for determining model order: Akaike's Information Criterion (AIC), Akaike's Final Prediction Error Criterion (FPE), and Minimum Data Length Criterion (MDL). Parameter estimations are approaches and algorithms used to estimate coefficients in GL, ARX, ARMAX, OE and BJ based on non-recursive (off-line method), recursive (on-line method) and state-space. Validation is the transformation to discrete transfer function and comparison using power series expansion, which zeroes and poles location would verify the stability.

Methods proposed in this chapter offers some promising tools for developing an inferential coriolis that is complex in nature. In the following chapter, the development of the inferential coriolis and the particular test rig required will be described and discussed.