



On-line Determination of a Go-NoGo State Using a Continuous Estimation of the System Response

Johanna McPhee and Rishad Irani

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

April 4, 2018

On-line Determination of a Go-NoGo State Using a Continuous Estimation of the System Response

Johanna McPhee and Rishad A. Irani
Dept. of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Canada

Johanna.McPhee3@Carleton.ca, Rishad.Irani@Carleton.ca

Abstract—The formulation, tuning and performance of a signal prediction algorithm as applied to the determination of a Go-NoGo state are discussed. Simulations were used to tune and assess the performance of the signal prediction algorithm. The paper describes the development of useful criteria, based on the mean and standard deviation of the predicted signal, used for producing the Go-NoGo state. A latching algorithm was used to improve the output of the Go-NoGo state.

Keywords—Signal prediction; Go-NoGo state; control system state estimation

I. INTRODUCTION

In marine applications, the determination of the ship motion and the relative motion between two independent bodies is a challenging task. The motion of the mothership often defines the availability for load transfer, aircraft landing opportunities and the launch and recovery of marine craft. Sophisticated and established methods do exist that determine the motion of a ship in a seaway through panel or finite element methods [1],[2]; however, these methods do not lend themselves to real-time estimation. Moreover, these methods cannot be extrapolated to determine any generalized signal or motion. Kuchler et al. [3] presented an implementation of a wave-prediction algorithm, which combats transport delays between actual ship motion and measurement of ship motion. Woodacre et al. [4] was able to build upon Kuchler et al. [3] and use a version of the prediction algorithm as a previewing function for a Model-Predictive Control scheme (MPC). The current authors are building upon these works to generalize a signal prediction algorithm (SPA) so that it can predict any sinusoidal signal and, given a set of criteria, determine if a process has a high likelihood of success.

This paper presents the formulation of the signal prediction algorithm (SPA) in the following section, followed by an investigation of the tuning parameters used for detecting the modes of the signal. Simulation results are analyzed in Section III and useful criteria for defining a “GO” scenario are developed for the purpose of producing a Go-NoGo command

signal. In Section IV, a latching algorithm is proposed to remove fluctuations in the Go-NoGo command signal. The paper ends with concluding remarks and future work. In general, the methods proposed can be used for fault detection and avoidance of unmanned systems.

II. SIGNAL PREDICTION ALGORITHM

A. Signal Prediction Formulation

The signal prediction method based on the work of Kuchler et al [3] and Woodacre et al. [4],[5] is composed of three distinct parts: mode detection, estimation, and prediction. To predict the wave motion, the periodic components, or modes, must be identified. These modes are determined by decomposing the measured signal $s(t)$ into a set of N sine waves expressed as

$$s(t) = \sum_{i=1}^N A_i \sin(2\pi f_i t + \varphi_i) + v(t) \quad (1)$$

where the amplitude A , frequency f , and phase φ of each mode i are obtained by taking the Fast Fourier Transform (FFT) of past measured data over a specified time interval. Additionally, the static signal offset is denoted by $v(t)$. The specified time window for the FFT, referred to as T_{FFT} , and sampling frequency are set depending on the frequency content of the known wave spectrum. The time window T_{FFT} is chosen such that the highest frequency of the wave spectrum can be detected, while also preventing aliasing. The sampling frequency of the FFT is selected such that a desired resolution in the frequency domain is achieved.

As conditions change, the number of modes N and associated frequencies f change with time; therefore, to predict the periodic motion, a peak detection algorithm is performed at every T_{FFT} interval on the latest set of measured data. This peak detection algorithm determines the amplitude A_{FFT} and frequency f_{FFT} of each mode based on the dominant peaks in the FFT spectrum. A peak in the FFT spectrum is determined to be

dominate when it exceeds a peak detection sensitivity, η , which is set by the user. When a new set of modes is identified, it is used to initialize an observer model with a new set of parameters: A_{FFT} , f_{FFT} , ϕ_{FFT} , and N .

The observer model is used to continuously estimate the mode parameters A_{FFT} and ϕ_{FFT} such that the latest signal measurements are used to adapt A_{FFT} and ϕ_{FFT} at each time step. For the prediction algorithm, a discrete Kalman filter is implemented to estimate the system states and has the form

$$\begin{aligned} \hat{x}_{k+1} &= \Phi \hat{x}_k + L(w_k - \hat{w}_k), \quad \hat{x}_0 = x_0 \\ \hat{w}_k &= C \hat{x}_k \end{aligned} \quad (2)$$

where k is the current time step, Φ is the discrete system matrix, \hat{x} is the vector of observed states, L is the observer gains matrix, \hat{w} and w are the estimated motion and measured motion respectively, and C is the system output matrix. The discrete system matrix, observed states and system output matrix were derived following the work of Kuchler et al. [3].

For the current study, a static observer gain matrix was selected via manual tuning and has the form

$$L = [0.2 \quad 3 \quad 0.2 \quad 3 \quad \dots \quad 0.2 \quad 3 \quad 0.1] \quad (3)$$

From each detected mode, two observer states $x_{i,1}$ and $x_{i,2}$ are estimated and then rearranged to solve for the adapted observer parameters ϕ_{obs} and A_{obs} at the current time t_k , such that:

$$\begin{aligned} \phi_{obs} &= \arctan\left(\frac{2\pi f_i \hat{x}_{i,1}}{\hat{x}_{i,2}}\right) - 2\pi f_i t_k \\ A_{obs} &= \frac{\hat{x}_{i,1}}{\sin(2\pi f_i t_k + \phi_{obs})} \end{aligned} \quad (4)$$

Finally, the prediction algorithm can be used to forecast the motion sequence over a prediction horizon T_{Pred} , i.e. at the time $t_k + T_{Pred}$. Referring to Equation 1, the predicted motion at $t_k + T_{Pred}$ is

$$s_{Pred}(t) = \sum_i^N A_{obs,i,k} \sin(2\pi f_i(t_k + T_{Pred}) + \phi_{obs,k,i}) + v(t) \quad (5)$$

To test the SPA and develop the Go-NoGo discriminator, a benchmark data set was used.

B. Benchmark Data

Figure 1 shows the benchmark test case used as the input signal $s(t)$ throughout this paper. This test case originates from a set of digitized ship motion data from the Australian DSTO report [6]. The signal $s(t)$ is the resolved roll motion of the data, in degrees.

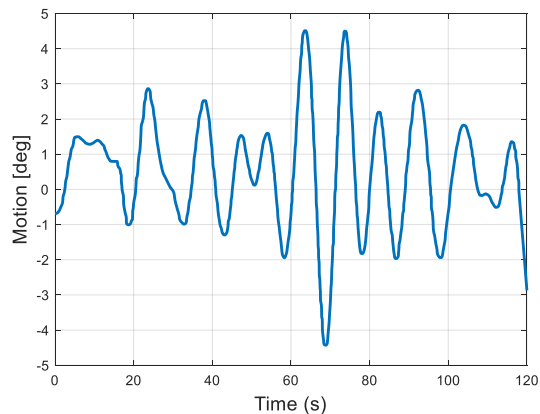


Figure 1. Benchmark test case of resolved roll motion from the Australian DSTO report [6].

C. Mode Detection Tuning

In the SPA formulation section above, it was discussed that to detect the dominant modes of the incoming signal it is necessary to select an appropriate T_{FFT} and peak detection sensitivity, η . The selection of an inappropriate T_{FFT} can result in two undesirable scenarios: the first being too short an interval that results in no modes detected, and the second being too long an interval that results in modes detected at a rate that does not keep up with changing conditions, yielding aliasing. To examine these issues, Figure 2 plots the input signal $s(t)$, as a solid blue line, and the prediction $s_{Pred}(t)$, as a dashed red line, at 0.25 s into the future. $s_{Pred}(t)$ is shifted back 0.25 seconds, i.e. 25 time steps, to place both the input signal and the prediction in the same time frame. The dotted green line between 70 s and 80 s in Figure 2 depicts the scenario where there is no predicted signal due to no modes detected during the previous T_{FFT} interval.

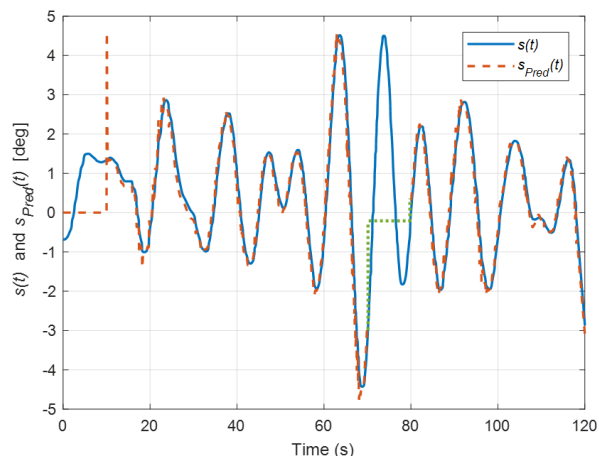


Figure 2. Measured and predicted signal together, with prediction signal time shifted to the input signal timeframe. The prediction discontinuity due to no modes detected is shown in green.

To address the issue of discontinuities in mode detection, the detection algorithm was amended such that in the case where zero modes are detected, the set of data over which the

FFT is performed is expanded every time step until at least one new mode is detected. The next mode detection is then performed after another T_{FFT} window, starting where the new set of modes was detected. In this way, the mode detection algorithm can handle slower than expected modes that may exist in the signal $s(t)$. In addition to the latter, the proposed amendments stated above also eliminate the second undesirable scenario, as the user can set a shorter T_{FFT} to avoid aliasing, without needing to consider the T_{FFT} length required to ensure continuous mode detection.

The peak sensitivity η describes the minimum peak height; a threshold value used by the peak detection algorithm to determine the dominant peaks in the FFT spectrum. Figure 3 shows the amplitude-frequency spectrum of the first 10 s of the benchmark test case in Figure 1 with $\eta = 0.01$, indicated by a dashed red line. For this η , the number of peaks detected is 6, denoted by the triangle markers. By inspection of the first 10 s of the benchmark test case in Figure 1, to capture the dominate motion at most two peaks should be detected. Any other peaks in the FFT spectrum is attributed to sensor noise. By increasing the value of η , the sensor noise can be rejected. Shown as a dashed red line in Figure 4, η has been increased by a factor of 10 ($\eta = 0.1$) and the number of peaks detected is two. The selection of peak detection sensitivity η must therefore be set based on sensor noise and is case specific. Future work could examine a recursive methodology to determine if a peak is valid or estimate how many peaks should be identified.

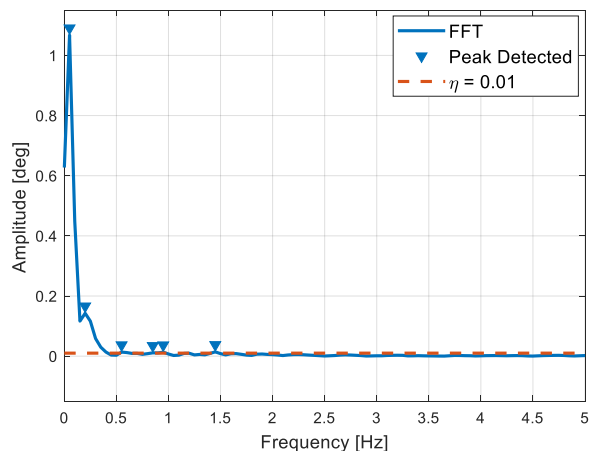


Figure 3. Dominant peaks detected from the FFT spectrum of the first 10 s of the benchmark case for $\eta = 0.01$

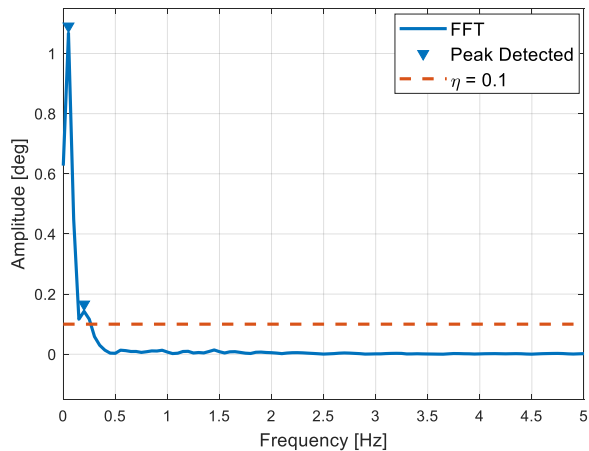


Figure 4. Dominant peaks detected from the FFT spectrum of the first 10 s of the benchmark case for $\eta = 0.1$

III. RESULTS

A. Simulation

The SPA detailed in Section II was implemented in MATLAB and simulated in the Simulink environment. Figure 5 depicts the Simulink block that calls on the SPA function. The input variables include input signal $s(t)$; the GO criteria; tuning parameters T_{FFT} and η ; and the prediction horizon T_{Pred} . The SPA block outputs the predicted signal $s(t+T_{Pred})$ and the Go/NoGo command signal. A fixed step, discrete solver was used with fixed step $dt = 0.01$ s. For the benchmark case in Figure 1, the simulation runtime is approximately 11 s for the full 118 s of the benchmark signal.



Figure 5. SPA function block implemented in the Simulink environment

B. Prediction Assessment

Figure 6 displays the predicted signal, solid red line, over a continuous prediction horizon of 10 s for the benchmark input signal, solid blue line, at time $t = 38$ s. The parameters used for the prediction were: $T_{FFT} = 10$ s and $\eta = 0.1$. In Figure 6, good accordance is observed between the input signal and predicted curve until $T_{Pred} \approx 3$ s. To investigate this further, the predicted signal is examined for 1 s, 3 s and 10 s prediction horizons.

Figure 7a depicts the prediction error for the prediction at 1 s into the future at every point in time. Similarly, Figure 7b and c respectively depict the prediction error at 3 s and 10 s into the future at every point in time. Note that the first 11 s are omitted to ignore the initialization period of the SPA. The maximum, minimum, mean and standard deviation of the error for each case are tabulated in Table I for $t = 11$ s to 110 s. From Table I, both the mean and standard deviation of the absolute error are significantly larger, by an order of magnitude, for the longest

prediction horizon of 10 s. The latter is consistent with what was qualitatively observed in Figure 6 at $t = 38$ s. The results in Table I suggest that determining the likelihood of success of a process to produce a Go-NoGo command becomes less accurate for longer prediction horizons as the mean error increases significantly over this time interval. Furthermore, defining the GO scenario based on the maximum error observed over T_{Pred} alone is not representative of true future events for longer predictions. The following section will develop useful GO criteria that considers the error that exists in the prediction.

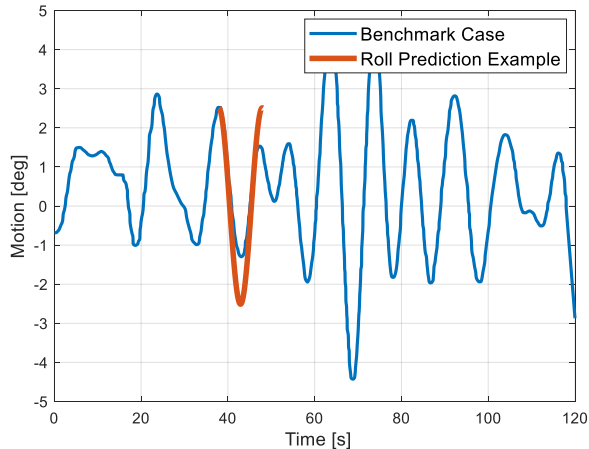


Figure 6. Benchmark data with prediction curve at $t = 38$ s plotted over a continuous time horizon of 10 s into the future.

TABLE I. ERROR METRICS FOR VARIOUS PREDICTION HORIZONS

Error [deg]	Prediction Horizon, T_{Pred}		
	$T_{Pred} = 1$ s	$T_{Pred} = 3$ s	$T_{Pred} = 10$ s
Maximum	1.74	3.72	3.88
Minimum	4.57×10^{-6}	4.41×10^{-4}	2.93×10^{-4}
Mean	0.29	0.89	1.41
Standard Deviation	0.35	0.92	1.69

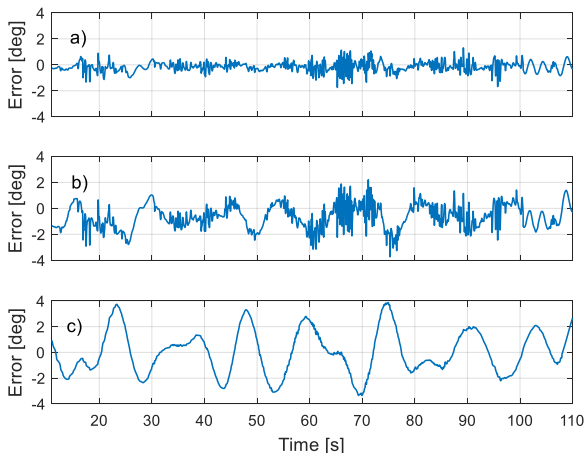


Figure 7. Error between benchmark data and prediction at a) 1 s into the future, b) 3 s into the future and c) 10 s into the future.

C. Developing the Go/NoGo Criteria

For the purposes of this paper, the SPA is used to output a Go-NoGo command based on given GO criteria. The GO criteria could be one or several parameters used to define the acceptable conditions for a GO scenario. The criteria are compared with the predicted signal over the continuous prediction horizon T_{Pred} at each time step. If over the prediction horizon, the predicted signal meets the GO criteria, a GO command is output. Conversely, if the GO criteria are not met by the predicted signal over the prediction horizon, a NoGo command is output. In this section, the development of useful definitions of the GO scenario, to be identified based on the GO criteria, are investigated.

The first definition of the GO scenario to be investigated is the simple Maximum Prediction Threshold (MPT) approach. This approach identifies a GO scenario if over the entire prediction horizon, the maximum value of the predicted signal is below the given threshold value. In other words, only one point of the predicted signal must be above the threshold to output a NoGo command. The results in Table I suggest that this simple Maximum Prediction Threshold approach would not provide a useful definition of the GO scenario for longer T_{Pred} , as there exists more error with the predicted signal as the prediction horizon increases. Two more GO criteria are suggested below that are defined such that the error that exists in the prediction is considered.

If instead, the GO scenario is defined based on the statistical spread of the predicted signal, it is possible to relax how the GO criteria identify a GO scenario. The spread of 1 Standard Deviation (1-SD) approach outputs a GO command when the prediction mean, evaluated over the prediction horizon, plus 1 standard deviation of the prediction is below a threshold value. The 1-SD approach is expressed empirically in Equation 6 as

$$\text{Go Criteria} > \overline{s_{Pred,k}} + \sigma_k \quad (6)$$

where $\overline{s_{Pred}}$ is the prediction mean over T_{Pred} at time step k , and σ is the standard deviation of the prediction over T_{Pred} at time step k . Equation 6 is equivalent to stating that for any randomly selected sample of the predicted signal over T_{Pred} , a GO command is output if 68% of the sampled points fall below the threshold.

The spread of 1.645 Standard Deviations (1.645-SD) approach defines the GO scenario to occur when the prediction mean plus 1.645 standard deviations is below the threshold value. The 1.645-SD approach is expressed empirically in Equation 7 as

$$\text{Go Criteria} > \overline{s_{Pred,k}} + 1.645 \sigma_k \quad (7)$$

Equation 7 is equivalent to stating that for any randomly selected sample of the predicted signal over T_{Pred} , a GO command is output if 90% of the sampled points fall below the threshold. Both methods described in Equations 6 and 7 are based on the same principle but differ in how conservatively they evaluate the predicted signal.

To evaluate the three definitions of the GO scenario MPT, 1-SD and 1.645-SD, the SPA was run for each case with the benchmark data and compared with the desired Go-NoGo command signal. The desired command signal was determined by setting the command as GO when the benchmark case is below the threshold value and setting the command as NoGo when the benchmark case is above the threshold value. Figure 8a shows the input signal, left axis, as a solid blue line with the GO criteria threshold, set as 2.5° , as a dashed red line. In Figure 8a, the desired Go-NoGo command signal, right axis, is also shown for $t = 50$ s to $t = 90$ s of the simulation. Here GO = 1 and NoGo = 0. Figure 8 part b, c and d depict the Go-NoGo command signal for the MPT, 1-SD and 1.645-SD approaches respectively. The three GO scenario definitions were evaluated based on the number of seconds that a GO command is output compared to the amount of GO time registered by the desired command signal. When obtaining the GO time, the first 11 s of data are omitted to ignore the SPA initialization period. In Table II, the GO time for the benchmark case from $t = 11$ s to $t = 118$ s is tabulated for each approach.

TABLE II. EVALUATION OF GO SCENARIO DEFINITIONS

"GO" metric	"GO" scenario definition [11 s – 118 s]			
	Desired	MPT	1-SD	1.645-SD
"GO" time [s]	98.98	87.71	94.78	77.94
Error [s]		11.27	4.20	21.04

Due to its close agreement with the desired command signal, the best definition of the GO scenario, based on a given threshold value, is the 1-SD method. Moreover, with the least amount of GO time, the 1.645-SD method was found to be the most conservative method for defining the GO scenario.

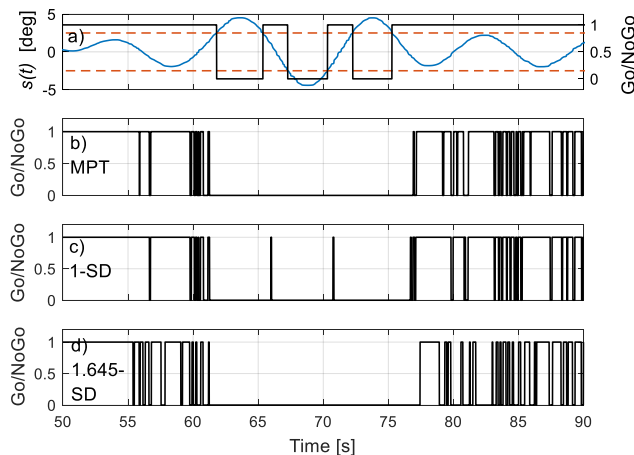


Figure 8. Go-NoGo command signals with desired command signal in a), and resulting command outputs for approaches b) MPT, c) 1-SD and d) 1.645-SD. For display purposes, the results are shown for $t = 50$ s to $t = 90$ s.

D. Go-NoGo Command Latching

Observing the Go-NoGo signals in Figure 8b, c and d at 60 s, 80 s and 90 s, there is rapid fluctuation in the command signal.

For a physical implementation of the SPA, it is desired that the SPA latches on to a GO or NoGo command to meet the responsiveness of the physical system components. A method for Go-NoGo command latching is proposed below to limit this undesirable fluctuation.

Figure 9 shows the logic flowchart for the latching algorithm. To start, the latching algorithm receives the Go-NoGo state determined from the predicted signal. Next, the algorithm looks to see if a target state has been set. If no target state has been set, the current Go-NoGo state is set as the target state. The algorithm must then determine whether it will commit to this target by watching the incoming Go-NoGo state over an evaluation period T_{eval} . During this evaluation period, if at any point the incoming Go-NoGo state differs from the target state, the current Go-NoGo state is set as a new target state and T_{eval} is reset. However, if all Go-NoGo states received during the evaluation period match the target state, the algorithm will "latch" onto the target state and output this command over the runtime interval T_{run} . After T_{run} is complete, a new evaluation period begins.

Figure 10 shows the results of the latching algorithm where the Go-NoGo command signal without latching is shown in 10a, and the Go-NoGo command signal with latching is shown in 10b. For illustrative purposes, T_{eval} was set as 0.1 s and T_{run} was set as 2 s, i.e. if a GO condition is sustained for 0.1 s the system will latch in a GO condition for at least 2 seconds. The command signals are output for the benchmark test case for $t = 50$ s to $t = 90$ s. It is observed that for the selected T_{eval} and T_{run} , all fluctuation was removed from the original command signal in Figure 10 a. T_{eval} and T_{run} are case specific and should be based on the physical systems involved.

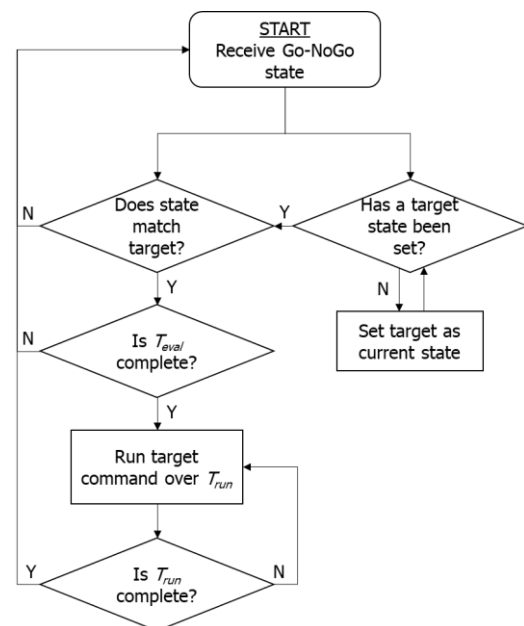


Figure 9. Flow chart of the logic used in the command signal latching algorithm.

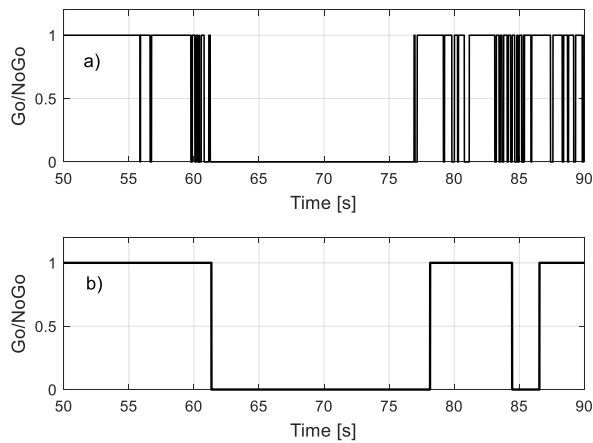


Figure 10. Go-NoGo command signals for the benchmark test case without latching in a) and with command latching in b). For display purposes, the results are shown for $t = 50$ s to $t = 90$ s.

IV. CONCLUSIONS

This paper describes the generalized formulation and tuning of a signal prediction algorithm as it applies to the determination of a Go-NoGo state. A key modification made to the SPA was the implementation of continuous mode detection that enables the algorithm to dynamically handle changing conditions of the input signal. Tuning the SPA also included setting a peak height sensitivity parameter that is used for determining the dominant peaks in the FFT spectrum during mode detection. By investigating the FFT spectrum of one identification period of the input signal, it was determined that the peak height sensitivity should be set such that it ignores sensor noise in the measured signal and is therefore case specific.

The paper also described the development of useful GO criteria used for producing the Go-NoGo state. Because prediction error increases over the prediction horizon, it was determined that a GO scenario identified by the maximum point of the prediction should not be used, as this approach can lead to erroneous conclusions regarding the Go-NoGo state. Instead, a GO scenario defined by the mean and one standard deviation of the predicted signal over the prediction horizon was found to

produce a Go-NoGo signal that agreed the most with the desired Go-NoGo signal.

Performance of the SPA was further improved with the implementation of a latching algorithm that successfully removed undesirable fluctuation in the Go-NoGo signal.

It is anticipated that several practical industrial applications can benefit from the proposed methods of determining a Go-NoGo state. In general, the methods proposed can be used for fault detection and avoidance of unmanned systems. Specifically, with UAV ship landing operations the SPA can be employed to estimate the ‘quiescent period’, where the roll and pitch angles of the ship are below critical threshold values. From the estimated quiescent period, a Go-NoGo state is inferred that can be used to increase the effectiveness of autonomous vertical landings on transient platforms.

ACKNOWLEDGEMENT

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), Kraken Robotic Systems Inc. and Carleton University for their support of this research.

REFERENCES

- [1] Kevin McTaggart, “ShipMo3D version 3.0 user manual for creating ship models,” Tech. Rep. TM 2011-308, Defence Research and Development Canada, 2012.
- [2] Weymouth, Gabriel David, Robert Vance Wilson, and Frederick Stern. "RANS computational fluid dynamics predictions of pitch and heave ship motions in head seas." *Journal of Ship Research* 49, no. 2 (2005): 80-97.
- [3] S. Kuchler, T. Mahl, J. Neupert, K. Schneider, and O. Sawodny, “Active control for an offshore crane using prediction of the vessels motion,” *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 2, pp. 297–309, 2011.
- [4] J.K. Woodacre, R.J. Bauer, R.A. Irani, Hydraulic valve-based active-heave compensation using a model-predictive controller with non-linear valve compensations, *Ocean Engineering*, Volume 152, 2018, Pages 47-56.
- [5] J. Woodacre, W. Wahidi, R. Bauer and R. Irani, "Coupling a Standard Hydraulic Valve and Advanced Control to Achieve a Motion Compensation System," in *ASNE Launch and Recovery Symposium*, 2016.
- [6] Arney, A., (1994, October). FFG-7 ship motion and airwake trial. part II: removal of ship motion effects from measured airwake data. Air Operations Division Aeronautical and Maritime Research Laboratory, Melbourne Victoria, Australia.