Investigating the Use of Inductive Transfer Learning and RNN to Quantify Extreme Event Statistics of Ship Motions

By

Jarod Kramer

B.S. Mechanical Engineering
University of Missouri, 2013

Submitted to the Department of Mechanical Engineering
and
and the System Design and Management Program
in partial fulfillment of the requirements for the degrees of

NAVAL ENGINEER
and
MASTER OF SCIENCE IN ENGINEERING MANAGEMENT

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Jarod Kramer
Department of Mechanical Engineering
May 12, 2023

Certified by: Themistoklis Sapsis
Professor of Mechanical Engineering, Thesis Supervisor

Accepted by: Joan S. Rubin
Executive Director, System Design and Management

Accepted by: Nicolas Hadjiconstantinou
Department of Mechanical Engineering
Chairman, Committee for Graduate Students

Investigating the Use of Inductive Transfer Learning and RNN to Quantify Extreme Event Statistics of Ship Motions

by

Jarod Kramer

Submitted to the Department of Mechanical Engineering

and

and the System Design and Management Program
on May 12, 2023, in partial fulfillment of the
requirements for the degrees of

Naval Engineer
and
Master of Science in Engineering and Management

Abstract

Ship motion software has been a critical tool for designers to study the extreme responses of ships in irregular waves. These studies and simulations often take thousands of hours to predict and analyze the ship's motion. Simulation results are often imperative to ensure the development of accurate operational guidance, typically in the form of plots, advising the crew on safe course and speed combinations to avoid dangerous roll and pitch motions. Two programs in use by the Navy to fill this need are the fast, lower-fidelity SimpleCode program and the slower, higher-fidelity Large Amplitude Motion Program (LAMP). Previous efforts have developed a framework to leverage machine learning through a Long Short-Term Memory (LSTM) network architecture to augment the SimpleCode program by mapping its ship motion output to the more accurate LAMP output without adding significant computational overhead. This process of using an LSTM neural network to improve the SimpleCode output provides the opportunity to supply predictions and guidance to the crew in real-time. However, the limits of this mapping across various sea domains still need to be discovered. By investigating these limits, a more generalized LSTM can be realized through inductive transfer learning and a model agnostic meta-learning approach, one that leverages the training of previous networks to augment SimpleCode across a broader range of seas or produce more accurate results on a narrow set of sea conditions after very few training samples.

Thesis Supervisor: Themistoklis Sapsis
Title: Professor of Mechanical Engineering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The safe operation of a ship in heavy weather or on high seas has long been a significant concern for the maritime industry and navies worldwide. These operating conditions can lead to excessive ship motions resulting in a range of hazardous conditions such as loss of stability or surf-riding. Therefore, the ability to predict a ship's behavior in these conditions is of great importance.

One facet of ensuring the safe operation of a ship in these conditions is through the development of ship-specific operational guidance regarding the ship's heading and speed for a given set of wave conditions that will minimize the risk of the ship experiencing a catastrophic event. A developed product of this guidance is often provided in the form of a polar plot, which is a two-dimensional plot of the ship's heading and speed and depicts the points at which ship motion becomes unsafe for a given speed and heading combination in the present seas. However, the development of these tools dictates the need for a dependable analysis of the hull's seakeeping, behavior, and performance qualities. This is often achieved through the numerical modeling of the ship's motion in waves [9].

One tool developed by the U.S. Navy Defense Advanced Research Projects Agency (DARPA) in 1988 and still in use today for the numerical modeling of ship motions is the Large Amplitude Motion Program (LAMP). LAMP employs a time-stepping approach using a three-dimensional potential flow panel method to compute all forces acting on the ship at each time step [11]. Furthermore, LAMP comes in four ascending solution "levels" of fidelity ranging from LAMP-1, the body linear solution, to LAMP-4, the body nonlinear solution. In essence, LAMP can produce high-fidelity results of ship motion but at a high cost computationally [8]. However, the high fidelity and high computation cost result in long computation times, hindering operational guidance development. It is infeasible to provide updated guidance in real-time, nor is it reasonable to provide guidance for every possible set of conditions [9].

Similarly, SimpleCode provides a computationally cheaper option than LAMP but with reduced fidelity. SimpleCode simplifies the local variations of wave pressure, thereby allowing the surface integral in hydrostatic and Froud-Krylov forces equations to be treated as a volume integral, allowing the forces acting on the submerged volume to be evaluated instantaneously and only at the geometric center of the volume while added mass and damping terms are handled via coefficients. Ultimately, the SimpleCode approach has been found to be a "suitable candidate" for reproducing the most important nonlinear ship motions [9].

In the case of extreme ship motions in random waves, it has been shown that the nonlinear ship dynamics that drive large amplitude motions require modeling via Monte Carlo simulation when in the time domain. The rarity of these extreme events dictates that a large quantity of data is needed to study these extreme events, often making a direct Monte Carlo simulation impractical due to the sheer quantity of simulation data [2]. This led to the development of computationally low-cost models such as SimpleCode, and SimpleCode is approximately two orders of magnitude faster than LAMP [8]. In fact, this reduced computation time utilizing SimpleCode is sufficient to employ in real-time scenarios. It would take SimpleCode approximately five minutes to generate a polar plot from real-time satellite weather data. In contrast, LAMP would take approximately ten hours to generate the same, albeit a more accurate plot, in the same scenario [8]. However, this reduced computation time comes at the cost of reduced accuracy, thereby reducing the usability of SimpleCode. This created an opportunity to leverage machine learning to improve the accuracy of SimpleCode while maintaining the reduced computation time by mapping SimpleCode ship motion output to LAMP ship motion output.

This opportunity has been explored in previous work and serves as the foundation for this thesis. Specifically, it has been demonstrated that a long-short term memory (LSTM) neural network can produce low error mappings between SimpleCode output and LAMP outputs across a range of conditions with little additional computational overhead [8]. These previous efforts are discussed further in chapter 2. While the previous work demonstrated multiple successful applications using LSTM in conjunction with SimpleCode and LAMP, it was primarily limited to mapping unimodal seas and small-scale experiments [8]. This thesis expands upon the previous efforts by focusing on the potential of transfer learning techniques to expand both the training domain and the testing domain covered by a single LSTM through replication and alteration on an ad hoc basis.

Chapter 2

Background and Previous Work

Previous efforts utilized LSTM neural networks to map the fast, low-fidelity SimpleCode output and the slower, high-fidelity LAMP output for ship motion prediction across various conditions. His work demonstrated the initial capabilities and challenges surrounding using an LSTM model to supplement the SimpleCode model. Specifically, and in relation to this thesis, previous work has demonstrated that [8]:

1. The LSTM can produce a low error map between SimpleCode and LAMP at a very low computational overhead.

2. The LSTM approach is successful in multiple applications, including ship motion and polar plot generation

3. Improving performance across large domains can be accomplished through the proper selection of training domain points as opposed to simply increasing the number of training points.

4. The limits of a single LSTM's ability to cover a large domain were not encountered.

5. Hyperparameters selected were robust across the many problems and scenarios.

Ultimately, this thesis endeavors to use the same LSTM networks and structures to find the limit of an LSTM's ability to cover a larger domain and uses inductive transfer learning to push that limit further. However, a brief overview of the previous work serves as the foundation for this thesis.

2.1 Neural Network Structure

2.1.1 Full Connected Neural Network

The foundation of a neural network is the "neuron" or "node", and it is a non-linear activation function that receives an input vector, $x \in \mathbb{R}$, parameterized by a weight vector, $(w_1, \dots, w_m) \in \mathbb{R}$ and offset, w_0, and produces a single output vector [4]. The non-linear activation functions are necessary to change the representational capacity of the neuron. Any bounded piecewise of a continuous and non-constant function can serve as an activation function. However, common activation functions are step function, rectified linear unit (ReLU) function, sigmoid function, hyperbolic tangent function, or softmax function [5]. Figure 2-1 shows a schematic of a single node in a neural network.

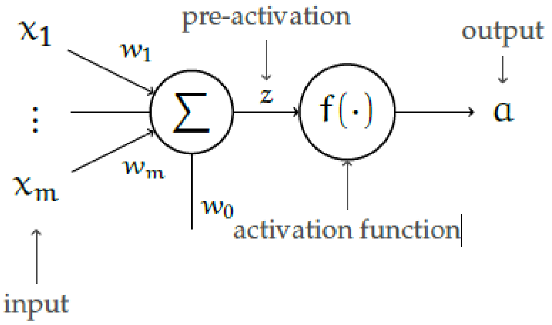

Figure 2-1: Schematic of a single node in a neural network.

Mathematically, the output of a node is given by:

$$a = f(z) = f((\sum_{j=1}^{m} x_j w_j) + w_0) = f(w^T x + w_0) \tag{2.1}$$

We can then organize multiple neurons into hidden layers and a neural network. A layer consists of multiple neurons that are not connected to each other. Each layer consists of, and is considered fully connected, neurons that all receive the same input from a vector, either initial or from a previous layer [4]. These layers are then connected to form a neural network consisting of an input layer, hidden layers, and a final output layer where the previous layer's outputs are fed as inputs into the next layer. Figure 2-2 shows the schematic of a neural network with a single hidden layer. In figure 2-2, A^l is the activation vector, thus A^0 is the input vector, Z^l is the pre-activation layer, f^l is the activation function, W^l is the weight vector, and W_0^l is the bias of layer l. Mathematically, the pre-activation and activation layers are defined by 2.2 and 2.3.

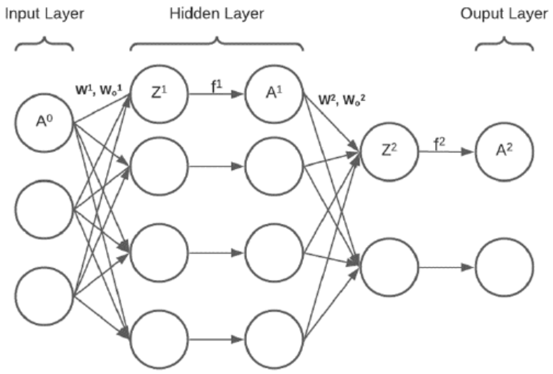

Figure 2-2: Schematic of a neural network.

$$Z^l = W^l A^{l-1} + W^l \tag{2.2}$$

$$A^l = f^l(Z^l) \tag{2.3}$$

Ultimately, the objective of any given network when given input is to find the right combination of weights and biases needed to produce the correct output or as close as possible to it. Finding these right combinations of weights and biases is often accomplished through a process used during network training called error back-propagation.

First, weights and biases are initialized as a random number, and the outputs are calculated for a given set of inputs (often referred to as a "forward pass"). The loss, broadly speaking, is the error associated with the difference between the neural network output and the correct output is determined. Finally, error back-propagation is performed by calculating the gradient of the loss function with respect to the weights and biases of the network at the output and then updating the weights and biases back towards the offsets. This process is repeated until the loss is minimized. In summary, weights and biases are randomly initialized, an output is generated given some input, the loss is computed, the gradient of the loss with respect to the weights and biases is calculated, and the weights and biases are updated [4]. Figure 2-3 shows an overview of this process.

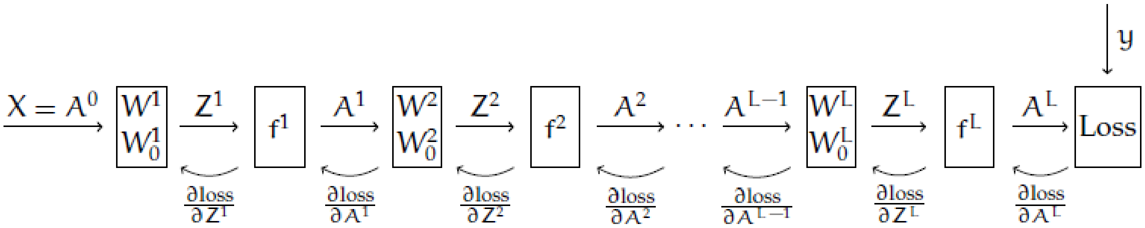
y
$X = A^0$
W^1
W_0^1
Z^1
f^1
A^1
W^2
W_0^2
Z^2
f^2
A^2
$\cdots$
A^{L-1}
W^L
W_0^L
Z^L
f^L
A^L
Loss
$\frac{\partial loss}{\partial Z^1}$
$\frac{\partial loss}{\partial A^1}$
$\frac{\partial loss}{\partial Z^2}$
$\frac{\partial loss}{\partial A^2}$
$\frac{\partial loss}{\partial A^{L-1}}$
$\frac{\partial loss}{\partial Z^L}$
$\frac{\partial loss}{\partial A^L}$

Figure 2-3: Overview of the error back-propagation process.

This cycle of forward and backward passes to find the proper weights and biases is repeated on a data set designated as the "training data". The training data is a set of input-output pairs used to teach the network what combination of weights and biases produce the correct output. However, care must be taken to avoid overfitting the network to the training data; that is, the network must be able to generalize to new data.

Overfitting, in essence, is when the network effectively memorizes the training set, making it ineffective at predicting the correct outcomes with new datasets. To avoid overfitting, the training data is split into training and validation sets. The training set is used to train the network, as previously discussed, and the validation set is used to evaluate the network's performance. This network performance evaluation occurs after each training epoch, but no error back-propagation occurs. Training can continue so long as the validation errors continue to trend down. If the validation errors begin to increase or stop changing, then training is halted to avoid overfitting the training data. The network is then tested on a set of data not used in training or validation, the test set, to determine the network's performance on new data. The results from the test set serve as a good measure of the generality of the network since they are derived separately from the training and validation sets [4].

2.1.2 Long-Short Term Memory

One shortcoming of a fully connected neural network identified in previous work for our use case is that it relies on instantaneously supplied input data. However, our extreme events tend to take the form of a rare transition occurring many standard deviations away from the mean [18] [8]. A fully connected neural network is not well suited to capture the causal relationships from our time-series input data. Thus, a unique recurrent neural network (RNN) architecture referred to as long-short term memory (LSTM) is introduced. An RNN differs in that it includes connections to nodes in previous time steps, but this makes error back-propagation difficult because the gradients tend to vanish, making it very difficult to train [4]. LSTM architecture addresses this through its ability to learn long-term dependencies and alter the error through three gating networks: the input gate, the forget gate, and the output gate [7]. The general architecture of an LSTM network is shown in 2-4 and governed by equations 2.4 through 2.9.

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{2.4}$$

$$i_t = \sigma(W_l x_t + U_l h_{t-1} + b_l) \tag{2.5}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.6}$$

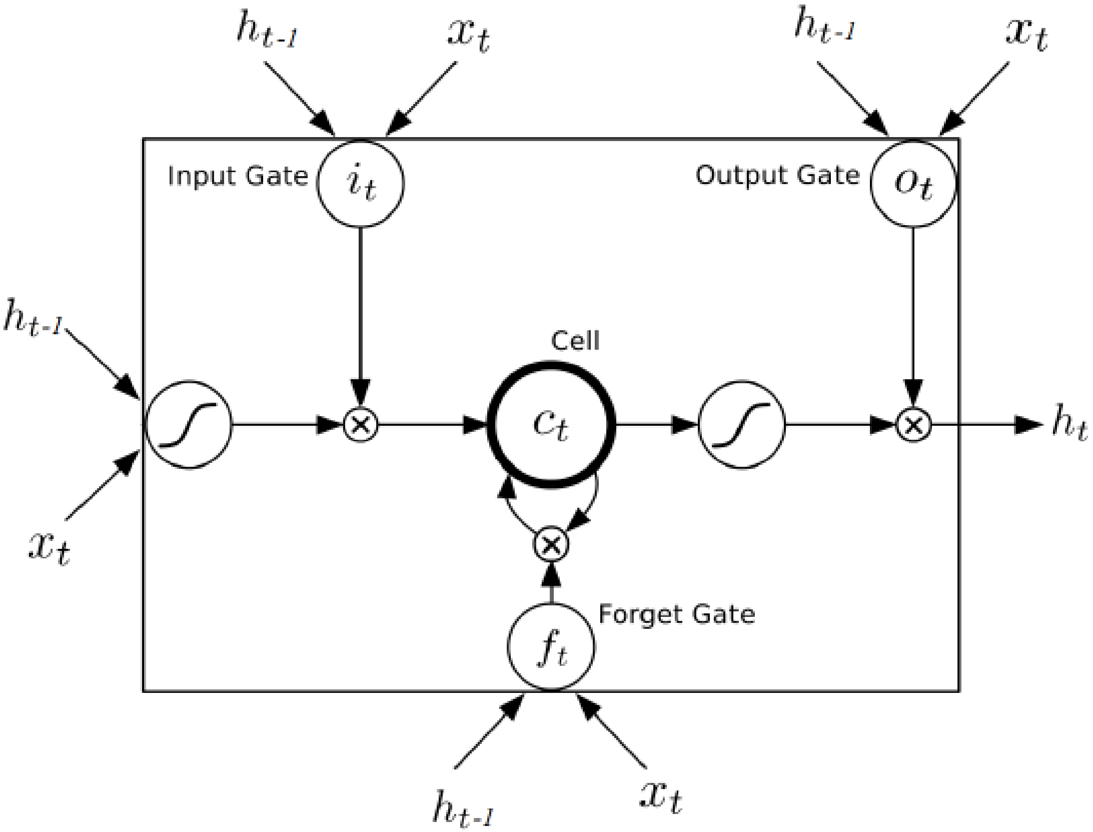

Figure 2-4: Schematic of a LSTM network.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.8}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.9}$$

Where the elements above are summarized as follows [7]:

- x is the input vector.
- i is the input gate.

- W and U are the weight matrices.
- b is the bias vector.
- g is the cell gate.
- c is the cell state.
- σ is the sigmoid function.
- $\odot$ is hadamard product.

All networks used in the previous work this thesis is based on are LSTMs, and they consist of an input layer, two LSTM hidden layers consisting of 30 neurons each, a linear layer, and an output layer [8]. The input layer accepts inputs of wave height and SimpleCode predictions for heave, roll, pitch, and vertical bending moments (VBM). This is provided to the first LSTM layer, which then generates and outputs a hidden state vector, h_t^1, that will serve as the input to the second LSTM hidden layer. The output of the second LSTM hidden layer, h_t^2, is then passed to the linear layer, which generates a vector of predictions, y_t, containing predictions for heave, roll, pitch, and VBM. This process is repeated at each time step, t, and a schematic of this process is shown in Figure 2-5 [8].

2.2 Input and Output Data

SimpleCode predictions for heave, roll, pitch, and VBM at each time step, along with wave surface elevation and the ship's center of gravity, served as the input vector for the LSTM network in all experiments. The wave elevation was calculated based on the Longuet-Higgins model, which was used in SimpleCode and LAMP calculations [9]. The output vector consisted of predictions for heave, roll, pitch, and VBM. The outputs are compared to a LAMP record of the same conditions as the SimpleCode input.

Normalizing all input and output data is necessary due to the sheer scale of the data.

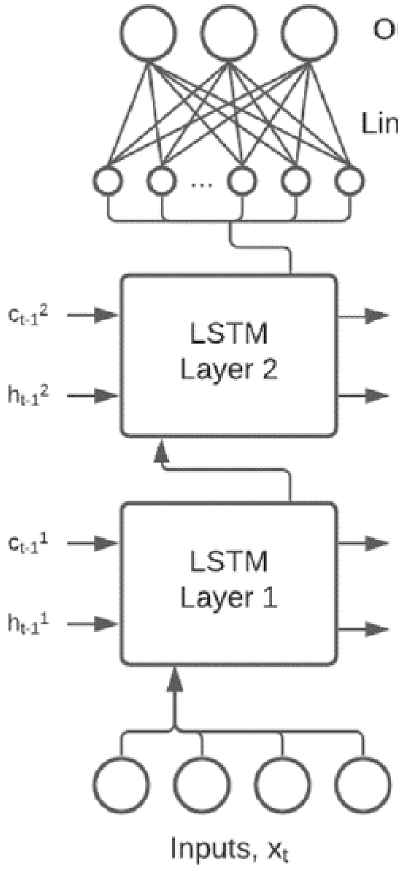

Figure 2-5: LSTM architecture used in this thesis and previous work [8].

Data with such an inherently large magnitude may have a dominating effect over smaller magnitude inputs, thereby hampering the neural network's performance and convergence. The input and output data were normalized using equation 2.10 [8].

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{2.10}$$

Where x_{norm} is the normalized data variable (e.g., SimpleCode heave), x is the data, μ is the mean of the data, and σ is the standard deviation of the data. Afterward, the standardized output data is reverted to the original scale using equation 2.11 [8].

$$x = x_{norm}\sigma + \mu \tag{2.11}$$

2.3 Ship Description

For all simulations of this thesis and those in previous work, the ship used was the flared variant of the Office of Naval Research (ONR) Topside series [3]. The Topsides series was developed to research the effects of geometry on ship motions and stabilities, and the flared variant has a similar topside flare to that of modern, large surface combatants [15]. The hull is shown in Figure 2-6 and its characteristics are listed in Table 2.1 [8].

Parameter	**Value**
Length	157 m
Beam	22 m
Draft	5.5 m
Displacement	8,730 t

Table 2.1: Hull characteristics of the ONR Topside flared variant [8].

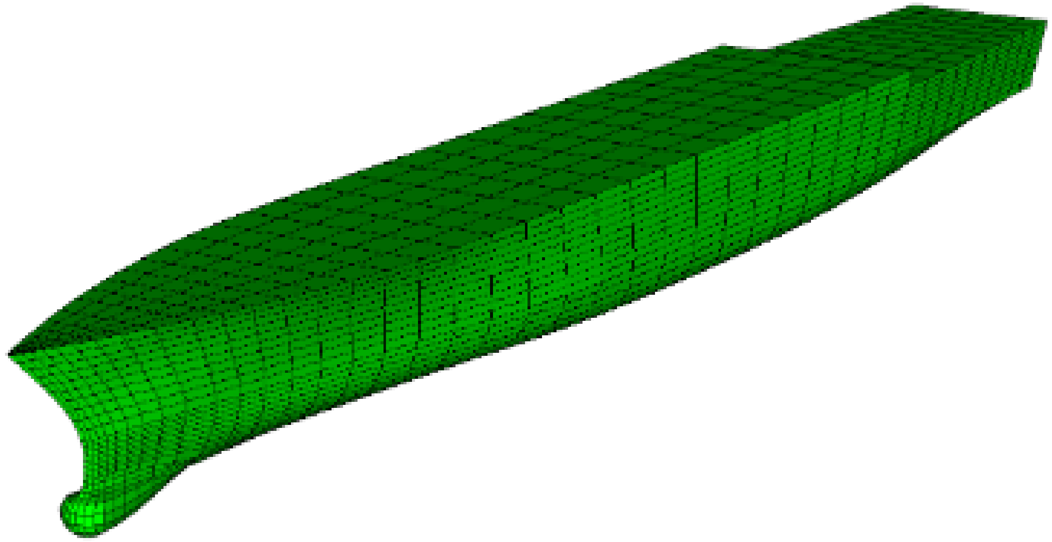

Figure 2-6: Hull geometry of the ONR Topside flared variant [15].

2.4 Hyperparameters

Hyperparameters consist of many arbitrary model parameters that are determined by the user when constructing a network and impact model performance. Some examples of hyperparameters available are the learning rate or the number of training epochs, and they are used to tune the LSTM networks. This thesis used previous work for all hyperparameter tuning except for briefly exploring additional hidden layers and a larger layer size. Expressly, the following hyperparameters were set based on previous work in reference [8]:

1. **Time Resolution** of the data was set at 0.1-second intervals.

2. **Training Data Sequence Length** was set to 1,800 seconds resulting in 18,000 points when sampled at 0.1-second intervals.

3. **Number of Layers and Hidden State Size** were set at 2 and 30 respectively.

4. **Learning Rate** was set at 0.01.

Due to the infeasibility of tuning hyperparameters for every test case, only two simple cases were used to tune the hyperparameters and used throughout this thesis: a test case of head-on seas and a test case of beam seas. In summary, the hyperparameter values were initialized at random values, used to train a model and record the models' performance on the test case. This process was repeated for 1,000 iterations, and the hyperparameters that resulted in the best performance were used for the rest of the experiments and throughout this thesis. A full description of hyperparameter performance can be found in reference [8].

2.5 Objective Functions

The objective function is used to determine the performance of the LSTM network and is a measure of the difference between the predicted and actual outputs of the LSTM network. In essence, one aims to minimize the objective function, thereby minimizing loss. Three separate objective functions were used in this thesis and previous work to determine how effective an LSTM network is at predicting ship motions: mean squared error (MSE) was particularly useful for training, amplitude magnified MSE (AMMSE), which emphasizes points with greater magnitude, and peak MSE which only points of local peaks are summed [8].

$$\textbf{MSE:} \quad L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.12}$$

$$\textbf{AMMSE:} \quad L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \cdot (\varepsilon_1 + \varepsilon_2\hat{y}_i^2) \tag{2.13}$$

$$\textbf{Peak MSE:} \quad L = \frac{1}{N}\sum_{i=peak}^{N}(y_i - \hat{y}_i)^2 \tag{2.14}$$

where y_i is the actual output, $\hat{y}_i$ is the predicted output, N is the number of points, and ε_1 and ε_2 are constants. Ultimately, MSE was used as the objective function in most experiments. A full description of the objective functions can be found in reference [8].

2.6 Model Performance

Apart from the objective function mentioned in the previous section, the Single Significant Amplitude (SSA) was used to determine the usefulness and performance of the LSTM network. The SSA was used heavily in previous work and this thesis to compare the performance between different networks, or an LSTM network and LAMP. It serves as an alternative to the absolute maximum value observed and is an estimate of the "largest one-third of observed amplitudes of motion " [9]. The SSA is calculated using the following equation:

$$\text{SSA} = 2\sqrt{\hat{V}_x} \tag{2.15}$$

Where $\hat{V}_x$ is a variance estimate of the amplitude observed calculated from the collection of peaks and troughs of the motion under investigation [9]. More information for the metrics for model performance can be found in reference [8].

Chapter 3

Unimodal LSTM's in Bimodal Seas

Previous work by Howard briefly explored the effects and merit of expanding the training domain range by a small amount. This was accomplished by creating three separate LSTM networks denoted “Narrow”, “Medium”, and “Wide” LSTM, where each name describes the relative range of the training domain compared to the others. All three LSTMs were trained exclusively on unimodal seas using evenly spaced training domain points centered around a midpoint for each dimension of the unimodal seas, four dimensions: significant wave height, modal period, sea heading angle (relative to the ship), and ship speed. Note that the sea heading angle is always relative to ship heading, which is established at 0°. The significant wave height and modal period are of the Bretschneider wave spectrum. The values for the training data points are listed in Table 3.1. The Medium and Wide training data sets were constructed of all possible combinations of their respective training points, 81 simulations in total. The Narrow training set was constructed of 81 simulations focused on the domain midpoint values [8].

The three LSTMs mentioned above were tested against a set of unimodal seas to determine the impact of increasing the training domain range across the four domain parameters. The unimodal test set consisted of varying two parameters from a mini-

Parameter	**Narrow**	**Medium**	**Wide**
Significant Wave Height (m)	7.5	7.0, 7.5, 8.0	6.5, 7.0, 8.5
Modal Period (sec)	15	14, 15, 16	13, 15, 17
Sea Heading Angle (°)	135	125, 135, 145	115, 135, 155
Ship Speed (knots)	8	6, 8, 10	4, 8, 12

Table 3.1: Training points for Narrow, Medium, and Wide LSTM's [8].

mum to a maximum value at a set increment size. The other two domain parameters were fixed at the midpoints. Table 3.2 describes the unimodal test set for the Narrow, Medium, and Wide LSTMs. 2-dimensional SSA error heatmaps of pitch and roll were created for each pair of variables and each LSTM. The specific results for all three LSTMs in the test case can be found in reference [8].

Parameter	**Min**	**Max**	**Midpoint**	**Increment**
Significant Wave Height (m)	5.5	9.5	7.5	0.5
Modal Period (sec)	11	19	15	1
Sea Heading Angle (°)	95	175	135	10
Ship Speed (knots)	0	16	8	2

Table 3.2: Unimodal test set for Narrow, Medium, and Wide LSTM's [8].

The results demonstrated that increasing the range of the training domain, even slightly, can significantly impact an LSTM network's ability to extrapolate to regions outside the training domain [8]. The Medium LSTM performed much better than the Narrow LSTM across the test set, and the Wide LSTM performed better than the Medium LSTM in the test set. Although, the most considerable performance improvements were demonstrated in the former comparison.

Thus, new test sets were created to explore the performance of the LSTMs on bimodal seas test sets. Bimodal sea states are sea states with both primary and secondary seas present. Thus, the parameter domain was expanded from four variables to seven by including secondary significant wave height, secondary wave modal period, and secondary wave heading angle. The new test set of bimodal seas fixed primary seas at the midpoints mentioned in Table 3.2 and fixed secondary significant wave height and secondary modal period at 3 meters and 20 seconds, respectively. The secondary wave heading angle was varied from 0 to 360 degrees in increments of 10 degrees. Results of the three unimodal LSTMs testing on the bimodal test showed that the LSTMs could provide valuable corrections in the presence of secondary seas, but the results were inconsistent [8]. Thus, bimodal training may be warranted.

For comparison, a fourth LSTM was created, one trained on bimodal seas. The primary seas of the Medium LSTM training set were duplicated and converted to bimodal seas by randomly adding secondary seas to 72 of the 81 training records, each with a secondary significant wave height of 3 meters and secondary modal period of 20 seconds, while the remaining nine records remained unimodal. This "Bimodal LSTM" was then tested on both test sets. The average roll and pitch SSA error for SimpleCode and all LSTMs across both test sets are shown in Table 3.3 [8].

Model	**Roll**		**Pitch**	
	Unimodal	Bimodal	Unimodal	Bimodal
Simple Code	2.49	3.18	0.88	0.88
Narrow	1.77	0.36	0.40	0.23
Medium	0.71	0.64	0.10	0.03
Wide	0.45	0.72	0.07	0.05
Bimodal	0.71	0.26	0.05	0.04

Table 3.3: Average SSA errors in unimodal and bimodal test sets [8].

These tests serve as the bridge point to the work conducted in this thesis. The tests conducted by Howard [8] demonstrate that the LSTMs can provide valuable corrections to the SimpleCode model in the presence of secondary seas, especially with bimodal training. Additionally, the LSTMs can extrapolate to regions outside the training domain. The efforts of this thesis will build upon the work of Howard et al. [8] by exploring the LSTM's performance on a greater test domain and investigating the merits of leveraging transfer learning to improve the performance of a unimodal trained LSTM versus training exclusively on bimodal seas.

3.1 Testing Domain Expansion

The first step was to expand the test domain to include a greater range of potential secondary seas. This was accomplished in a manner similar to the process outlined previously and referenced in Table 3.2. The test domain was expanded to seven parameters: primary and secondary wave system significant wave height, modal period, sea heading angle, and ship speed. Two of the seven parameters varied from a minimum to a maximum value at a set increment size. The other five parameters were fixed at a specific point based on LSTM model training or the observed midpoint of the parameter range. For all but primary and secondary sea headings, the fixed point also corresponds with the midpoint of the parameter when varied. A fixed secondary sea heading angle far from the primary sea heading angle used for training was chosen as previous work indicated it was the poorest performing area. Table 3.4 describes the test set.

	Parameter	**Min**	**Max**	**Fixed Point**	**Increment**
Primary System	Significant Wave Height (m)	5.5	9.5	7.5	0.5
	Modal Period (sec)	11	19	15	1
	Sea Heading Angle (°)	0	360	90	15
	Ship Speed (knots)	0	16	8	2
Secondary System	Significant Wave Height (m)	1.5	5.5	3.5	0.5
	Modal Period (sec)	16	24	20	1
	Sea Heading Angle (°)	0	360	270	15

Table 3.4: Bimodal test set for Medium and Wide LSTMs.

3.2 Measuring Performance

Only the Medium LSTM and Wide LSTM were tested due to their superior performance compared to the Narrow LSTM. The training was identical to that described above and in Table 3.1. Two-dimensional heatmaps of model error were generated for each possible pair of variables and each LSTM. This results in 21 pairs and 84 heatmaps of LSTM models, one for pitch and one for roll. While heave error was also explored, it was always on the magnitude of centimeters, whereas roll and pitch errors exhibited significantly more error. Thus, the focus remained on the roll and pitch errors. Model error was defined as the absolute value of the difference between the model SSA and the LAMP SSA as seen in Equation 3.1. This established a metric for comparing different models.

$$\text{Model Error} = |SSA_{LSTM} - SSA_{LAMP}| \tag{3.1}$$

Additionally, these heatmaps were displayed with a similar heatmap depicting the model error of SimpleCode for the same test set. Finally, a third heatmap was generated, dubbed the "delta heatmap", based on the difference between the SimpleCode model error and the respective LSTM model error. In other words, the third heatmap is the difference in model error between two separate models. This is shown mathematically by Equation 3.2 and allows us to quickly, visually compare the relative performance of the two models directly.

$$\Delta_{SSA} = (\text{Model Error})_{SimpleCode} - (\text{Model Error})_{LSTM} \tag{3.2}$$

This was performed for both roll error and pitch error, bringing the total to 210 heatmaps used to compare the performance of the Medium and Wide LSTM in bimodal seas over that of SimpleCode. The delta heatmaps were color-coded, and a new scale was established to improve their readability. These practices were reused throughout this thesis. In the delta heatmaps, green indicates better performance by the LSTM, while red indicates a better performance by SimpleCode. The color scale is centered on zero, such that a white color indicates a zero difference between the two models.

For the sake of brevity, only heatmaps exploring combinations of primary and secondary sea heading angles and ship speed will be displayed and discussed here. The reason for choosing these parameters is that they are the only parameters in the domain that change in real time aboard a ship. All other parameters can be pretrained by a network based on location and the time of year due to the availability of large historical datasets and current weather reports. Wave heading angles and ship speed are the only parameters changed frequently and rapidly by the ship. In other words, these are the only three parameters of the domain the ship has control over and are rapidly changing. This is discussed further in Section 4.1.

Additionally, both the Medium and Wide LSTMs performed nearly identically. Thus, only Medium LSTM heatmaps will be included in the written report moving forward. All heatmaps will be stored in the repository listed in Appendix C. All heatmaps in the written thesis, in larger print, can be found in Appendix A and Appendix **??**.

3.3 Results: Medium LSTM

These results on the expanded bimodal domain served as a baseline measure of performance for later work. For the reasons mentioned earlier, only heatmaps exploring combinations of primary and secondary sea heading angles and ship speed will be displayed and discussed here. In general, variation of the primary seas had a much more significant impact on the performance of the LSTM compared to SimpleCode than the variation of secondary seas parameters. Specifically for the parameters discussed in this section, the expansion of the primary seas heading domain coupled with the addition of bimodal seas increased the roll error dramatically in some areas, often to the point of performing worse than SimpleCode. At the same time, the pitch stayed relatively small but still significant. However, varying secondary wave heading angles had a much less profound effect than varying primary wave headings. The unimodal LSTMs show some capacity to extrapolate to seas beyond their training, but additional bimodal training is warranted. A summary of the absolute maximum SSA model error, the location of the maximum, and the mean model error for each experiment in both roll error and pitch error are shown in Table 3.5 and Table 3.6. Here, it is easy to see that the maximum error in both roll and pitch is often equal to or much greater than that of SimpleCode, while the mean remains comparable in cases of expanded primary wave headings. This demonstrates how the LSTM struggles to extrapolate in this much larger domain so far from its training point. Meanwhile, the secondary seas heading angle seems to have little impact on the performance of the LSTM.

Varied Parameters	**SimpleCode**		**LSTM**	
	Max	Mean	Max	Mean
Primary Heading vs Speed	6.67	2.60	9.78	3.12
Secondary Heading vs Speed	6.82	4.18	2.25	0.49
Primary vs Secondary Heading	6.67	2.62	11.07	3.05

Table 3.5: Maximum and mean SSA roll error in bimodal seas.

Varied Parameters	**SimpleCode**		**LSTM**	
	Max	Mean	Max	Mean
Primary Heading vs Speed	1.67	0.71	1.98	0.52
Secondary Heading vs Speed	1.70	1.23	0.51	0.17
Primary vs Secondary Heading	1.37	0.65	1.35	0.38

Table 3.6: Maximum and mean SSA pitch error in bimodal seas.

3.3.1 Primary Wave Heading Angle vs Ship Speed

Consistent with previous results on the smaller bimodal set, the Medium LSTM performed better than SimpleCode within and near the realm of its training domain (i.e., the LSTM performed will near the speeds and heading it was trained on) and performed worse when wave angles were near the beam. In the case of roll error, the LSTM performed best near its training headings and the mirror of those headings. Performance dropped the farther the heading was from the training domain, especially with following seas but also in bow seas at particularly low speeds. There is also a degradation in performance when seas are near the beams for both roll and pitch.

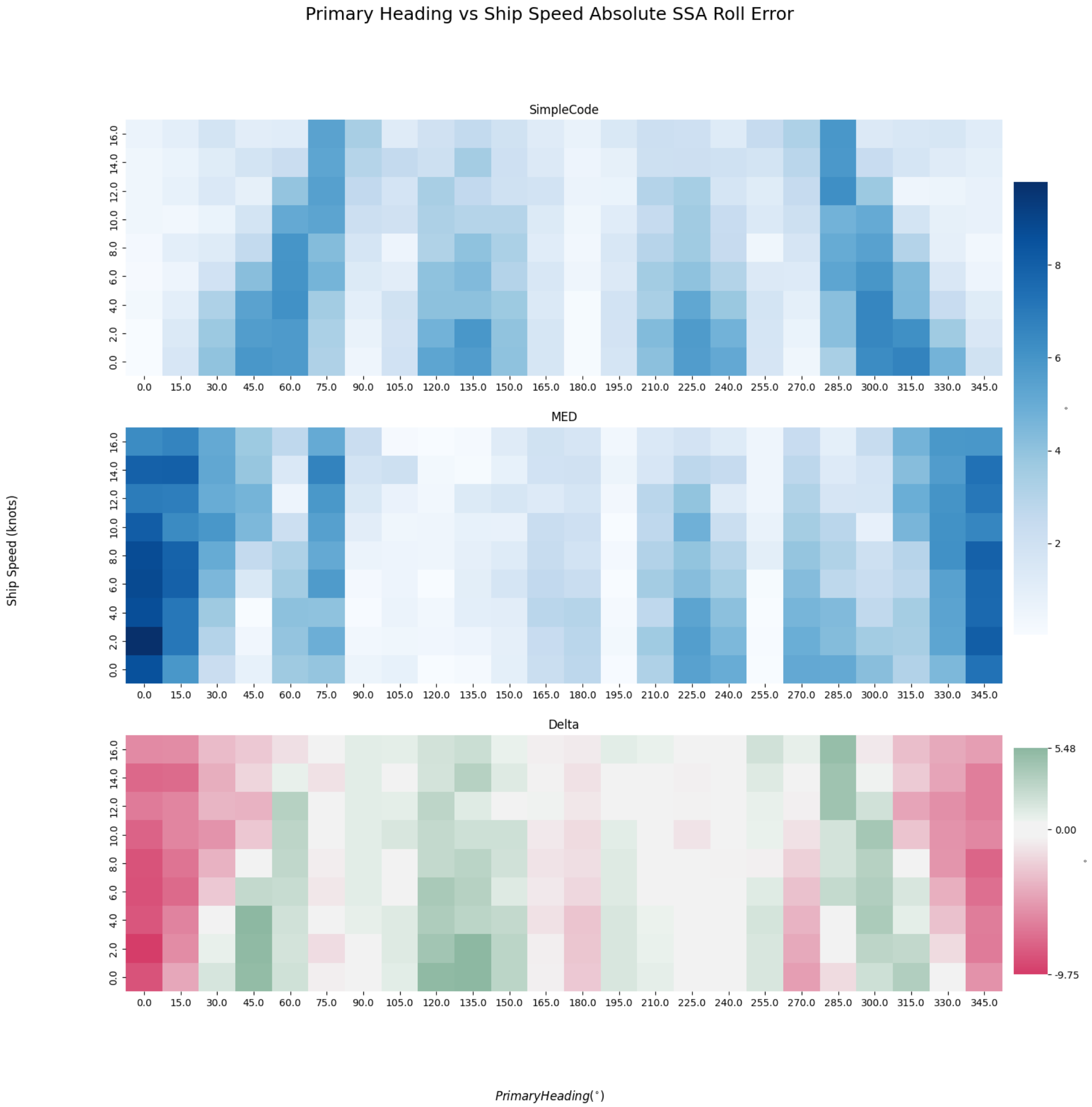

Figure 3-1: Primary wave heading angle vs ship speed roll error.

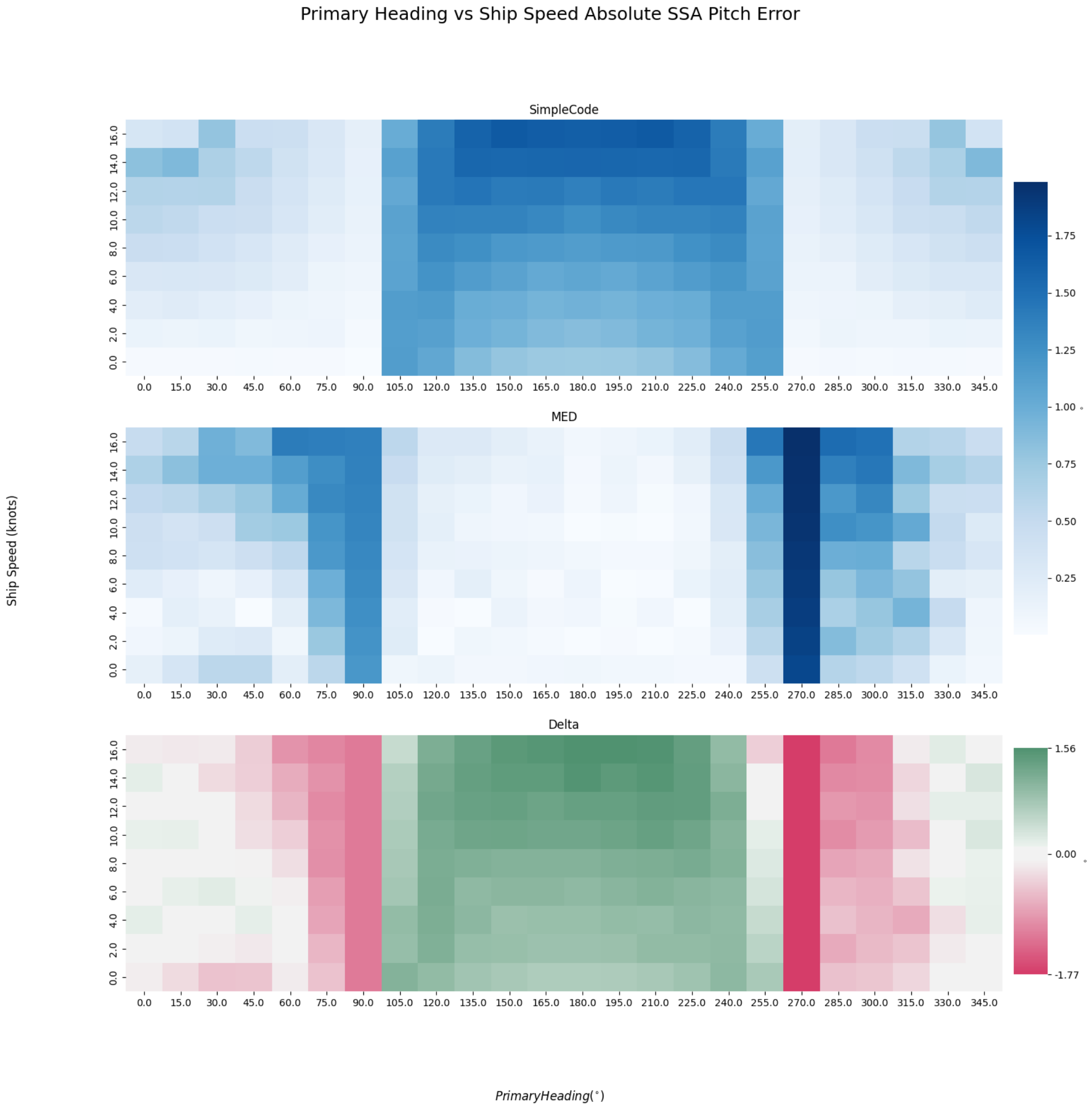

Figure 3-2: Primary wave heading vs ship speed pitch error.

3.3.2 Secondary Wave Heading Angle vs Ship Speed

Figure 3-3 and Figure 3-4 show the performance of the Medium LSTM when the secondary wave heading angle is varied. The results demonstrate that the variation of the secondary wave heading angle and speed has a much smaller impact on the performance of the LSTM compared to SimpleCode. Furthermore, the addition of speed variation to secondary wave heading variation had very little influence over

LSTM performance as these results for both roll and pitch are very similar to previous work varying secondary wave heading alone [8].

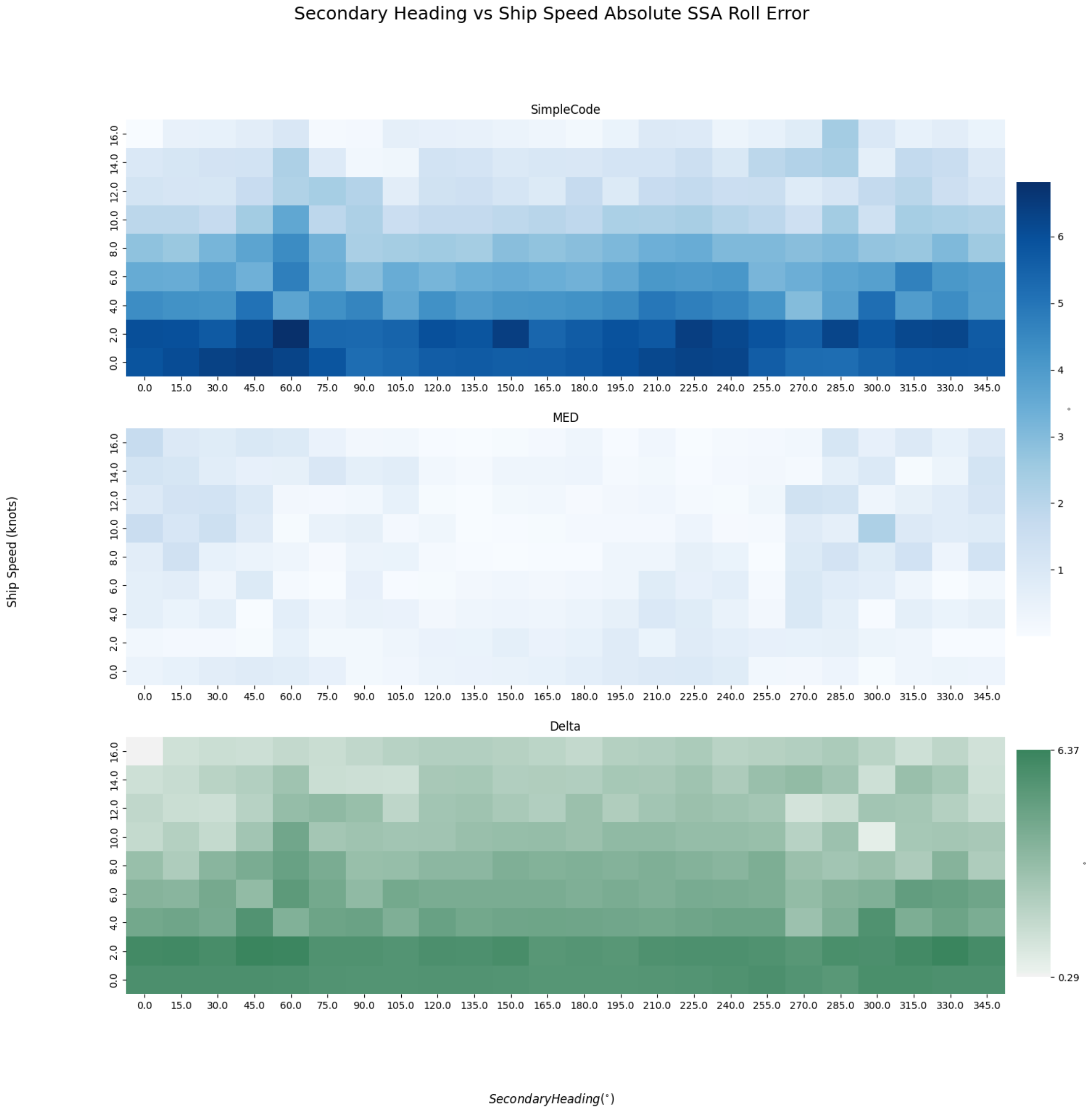

Figure 3-3: Secondary wave heading vs ship speed roll error.

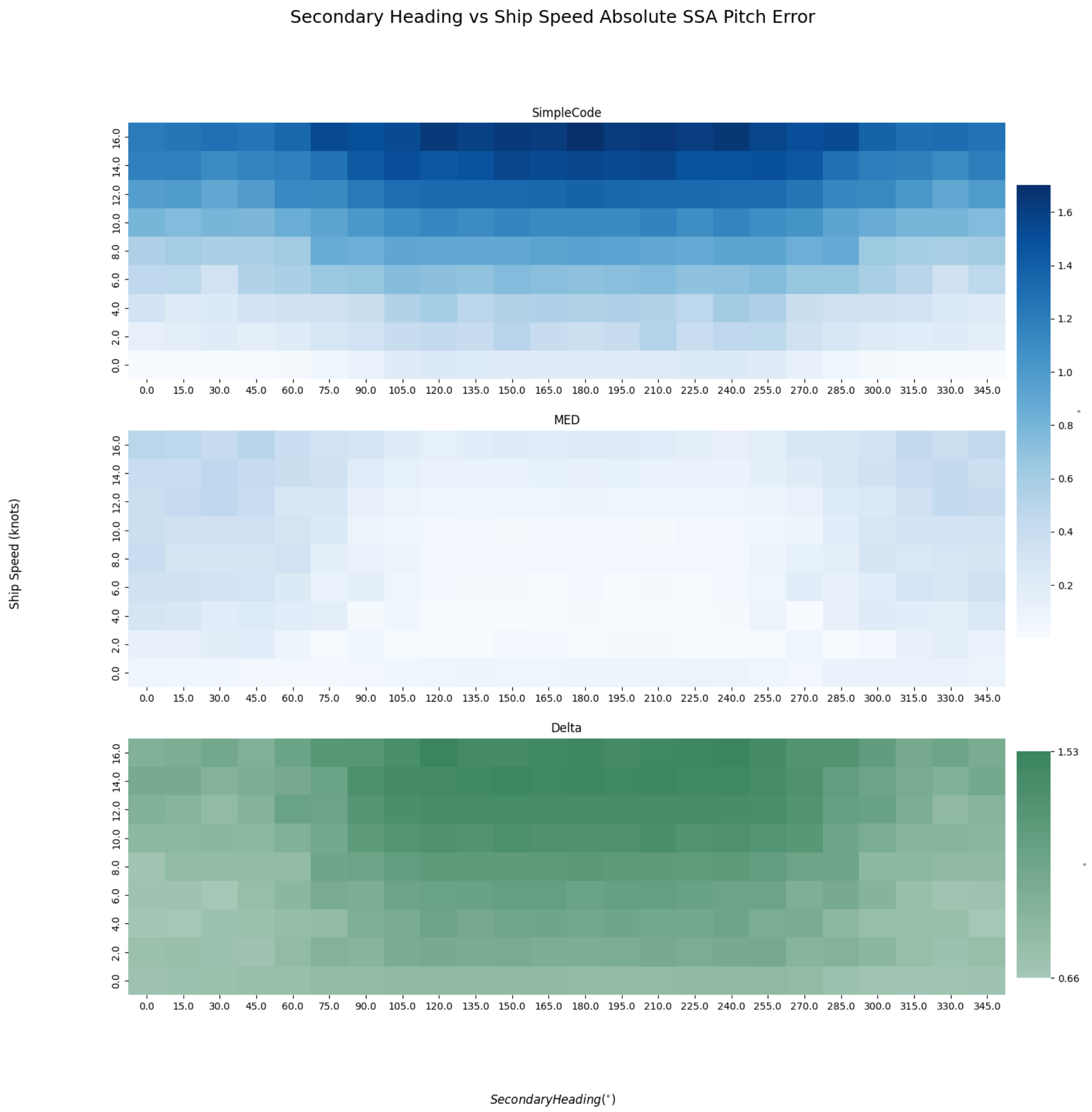

Figure 3-4: Secondary wave heading vs ship speed roll error.

3.3.3 Primary vs Secondary Wave Heading Angle

Figure 3-5 and Figure 3-6 show the performance of the Medium LSTM when both the primary and the secondary wave heading angles are varied. However, the primary wave heading angle is limited to 180° to increase the legibility of the graph. While it is not true that the LSTM performance is symmetric about 180°, the results are similar enough to warrant the simplification. Again, the LSTM performs the poorest with following seas or beam seas.

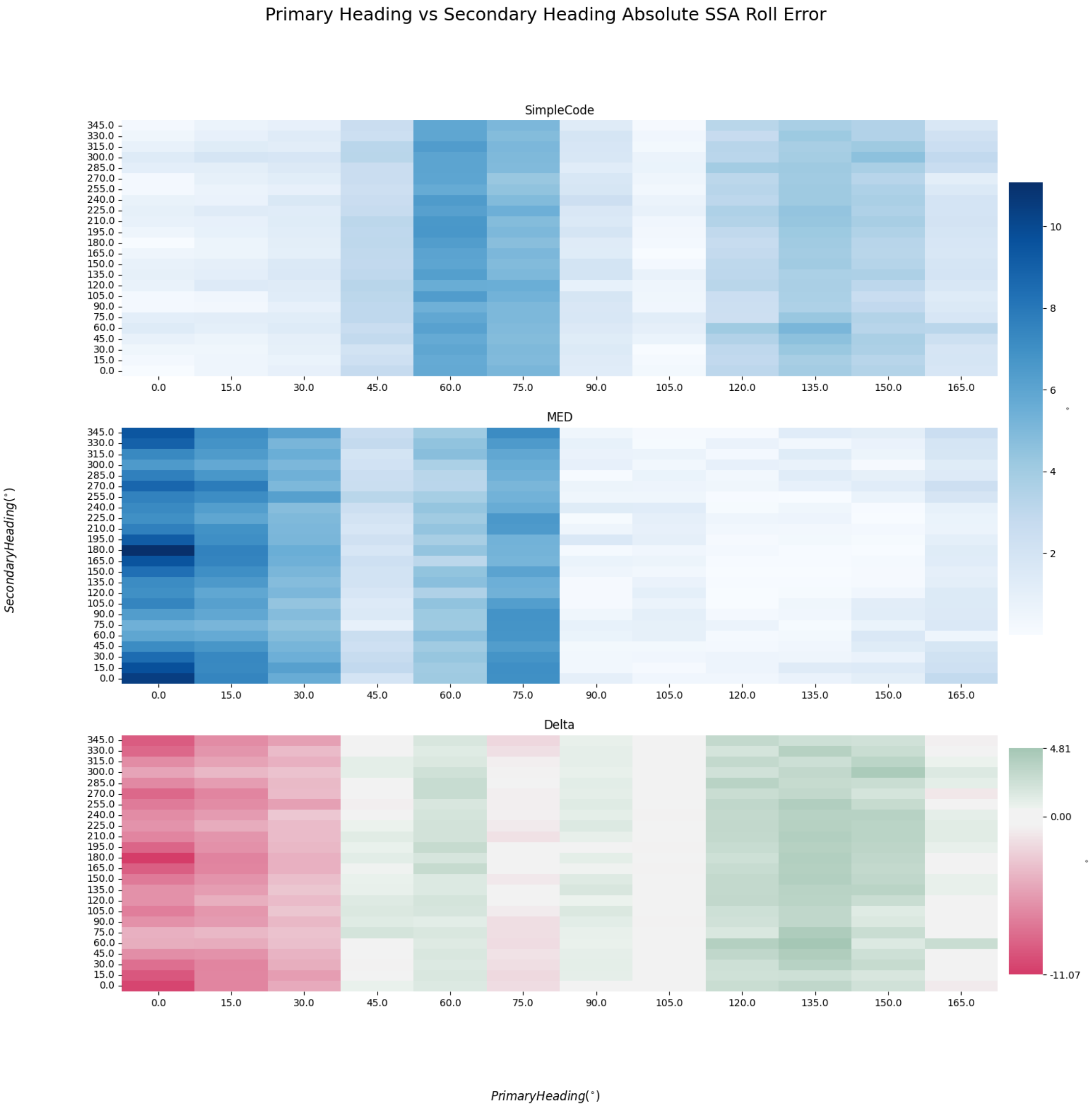

Figure 3-5: Primary wave heading vs secondary wave heading roll error.

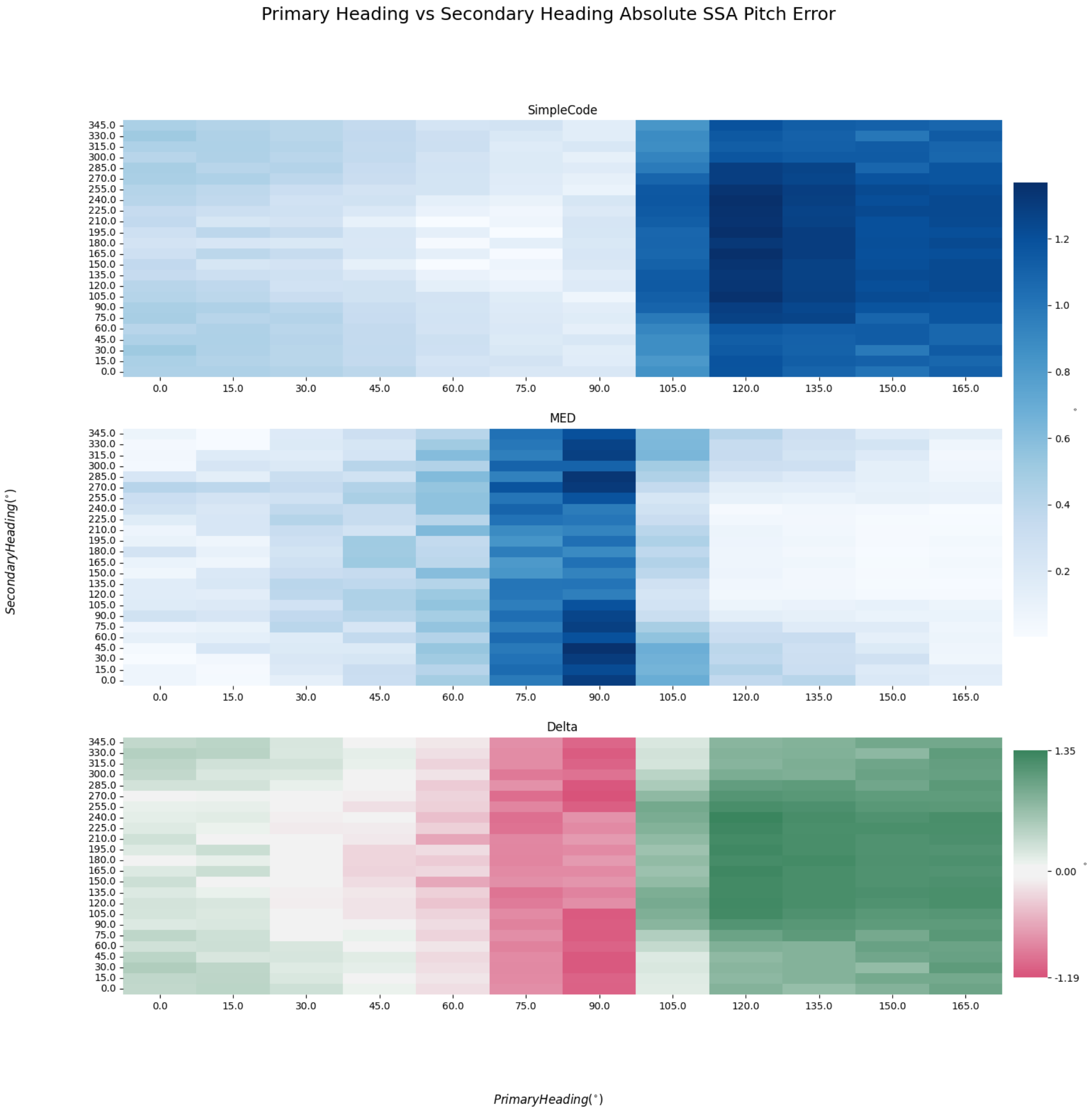

Figure 3-6: Primary wave heading vs secondary wave heading roll error.

Chapter 4

Improving LSTM performance in the Bimodal Domain

With a baseline performance in an expanded, varying bimodal domain established, the next step is to investigate opportunities to improve the performance of the LSTM in the bimodal domain. The evident approach is the development of bimodal-trained LSTMs, and this approach was briefly investigated in previous work and discussed in Section 3. The conclusion was that the bimodal-trained LSTM generally outperformed the unimodal-trained LSTM in bimodal cases and performed comparably to unimodal LSTMs in unimodal seas, albeit inconsistently [8]. The less obvious approach is to investigate the potential to use elements of transfer learning to improve the performance of a unimodal trained LSTM in the bimodal domain.

4.1 Transfer Learning

Broadly speaking, transfer learning is a set of techniques that aim to extract knowledge from a source domain and leverage that knowledge to improve the learning of a model on a target domain [12]. Transfer learning is particularly useful when the problems are different but very similar, as is this use case [14]. While training a neural network for every possible set of seas is impossible, using a robust unimodal LSTM and training it from real-time weather data is feasible. The secondary wave systems may be described as a small swell from a distant storm; thus, a ship may use a unimodal LSTM most of the time and leverage transfer learning from the available data, perhaps even onboard sensors, to update the LSTM as needed.

To give a more formal definition of transfer learning from Pan et al., "Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function f_T in $\mathcal{D}_T$ by using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{T}_S \neq \mathcal{T}_T$. [12]"

Regarding the use case investigated in this thesis, the source domain would be considered the unimodal seas training set, while the target domain is the bimodal seas training set. Similarly, the source and target learning tasks are the unimodal seas test sets and bimodal seas test sets corrections, respectively. To help demonstrate this, we can think of the source domain data in the form of label data pairs, $\mathcal{D}_S = \{(x_i^S, y_i^S), \ldots, (x_n^S, y_n^S)\}$ [16]. Here, x_i^S is the instance of the input vector from SimpleCode discussed in Section 2.1.1 with specific unimodal waves, and y_i^S is the instance of the desired output from the LSTM, corresponding with the ship motion determined by LAMP using the same unimodal wave system inputs as the SimpleCode record. Likewise, the target domain data can also be written as data pairs, $\mathcal{D}_T = \{(x_i^T, y_i^T), \ldots, (x_n^T, y_n^T)\}$ [16] where x_i^T is again the instance of the input vector from SimpleCode, but in from a bimodal wave system training set. Again, y_i^T is the instance of the desired output from the LSTM, corresponding with the ship motion determined by LAMP using the same wave system inputs as the SimpleCode record, bimodal in this case. We are applying knowledge from the unimodal seas to the LSTM model in the bimodal seas by initializing the model with weights and biases already learned from the unimodal training set.

This specific case can be further described as inductive transfer learning. Inductive transfer learning is a subset of transfer learning in which the target task differs from the source task while the source and target domains may or may not be the same. The labeled data pairs in the target domain are required to “induce” the correct predictive model function [12].

Transfer learning has become a standard for neural networks in image processing, but it is still relatively young in applications involving time series. Some reasons for this are due to data scarcity driven by the high cost of labeling time-series data and the lack of transfer learning portability because data format changes between target domains [10]. However, neither of these challenges is present in this use case. The source and target domains are similar, with nearly identical data formats. However, for real-world applications, we want fast model adaptation. Thus, the aim was to investigate the limits of the transfer learning process so that as few samples as practical are taken when investigating the new domain and task.

4.2 Model Agnostic Meta-Learning

Model agnostic meta-learning (MAML) provides a framework for inductive transfer learning problems that will help adapt a model to a new problem domain with very few data samples [14]. Put plainly, the objective of MAML is to learn a set of suitable initialization parameters for a network such that the network can be quickly adapted to a new task after only a few task-specific samples to learn, often referred to as few-shot problems [1]. In other words, we want to develop a base model that can be easily fine-tuned and adapted to perform well on a new domain and task. The complete MAML algorithm and its development is detailed in [6].

To fully describe the MAML, some new but similar terms must be introduced. First, in this meta-learning scenario, we have a collection of learning tasks, τ_i, drawn from a distribution over tasks $p(\tau)$. Using met-training, the model learns a new task τ_i drawn from $p(\tau)$ by using K labeled samples and updated via feedback from loss, $\mathcal{L}_{\tau_i}$. Finally, the model is tested on new samples from τ_i, and the model is updated [6]. Each learning task consists of a training dataset to update the model and a testing dataset, sometimes called query and support datasets, respectively, to evaluate the performance. Each dataset is again comprised of (x_i, y_i) data pairs as before. It works via an inner loop that initializes a base network with parameters θ and updates the parameters using the gradient descent of each task for each step. This results in parameters adapted to each task that then update θ based on the performance with the test samples. This is repeated until convergence is reached [13].

In the problem outlined in this thesis, the initial base model is the Medium unimodal LSTM discussed previously. This starting point was repeated for every pair of varied parameters. It was assumed that this gives a starting point focused on the most influential input parameters in determining the ship's motion. The task distribution is all the combinations of varied parameters within a bimodal test set. The learning tasks are each of the individual combinations of the varied parameters, while the others are held stagnant. The objective was to establish a baseline LSTM model well suited to handle changing seas with only a few training samples.

4.3 Unimodal and Transfer Learning Results

4.3.1 Random Samples

As discussed earlier, only a few samples were taken in each iteration. Initially, five random samples were taken from a given learning task. Unsurprisingly, this leads to inconsistent results. Heatmaps were plotted to compare the performance of the base unimodal LSTM without additional bimodal training and the LSTM after training on five random samples of bimodal seas. A third heatmap was created as before, using Equation 3.2, except the SimpleCode model error is replaced with the base LSTM model error. Thus, a positive, green number indicates that the LSTM model that underwent the transfer learning process performed better. In contrast, a red, negative number indicates the base model performs better in those conditions. The five random samples selected for additional training are highlighted in yellow boxes.

Figure 4-1 through Figure 4-12 display the heatmap results of roll and pitch SSA error across the primary and secondary wave heading domains. Generally, pitch prediction was improved through this method. Pitch either retained a similar value to the base model or improved by a large amount. This was true for both primary and secondary wave heading angles. Roll, on the other hand, shows improvement near new training samples and worsening predictions in other areas. Specifically, areas of the domain that were not covered by the new bimodal training nor the original unimodal training, at least in the case of varying primary seas heading angle. Although, this was observed in the case of a varying secondary wave heading angle too.

This may occur due to the unprincipled sampling method. In other words, by leaving the new domain samples up to chance, we may be hurting the process instead of helping it. This is explored in the next section by controlling the sampling process on a very rudimentary level.

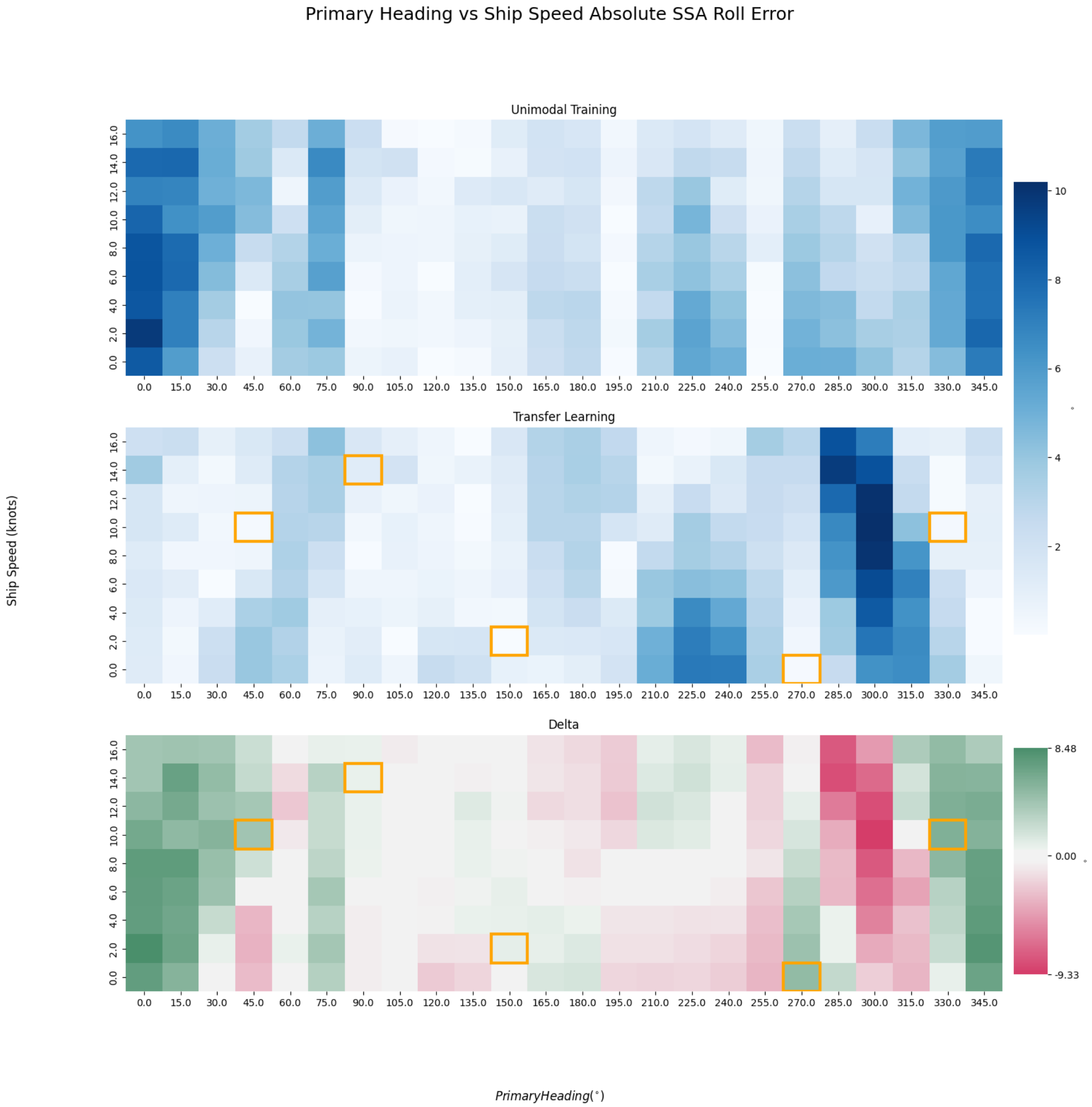

Figure 4-1: Primary wave heading angle vs ship speed roll error: Unimodal Training and Random Transfer Learning.

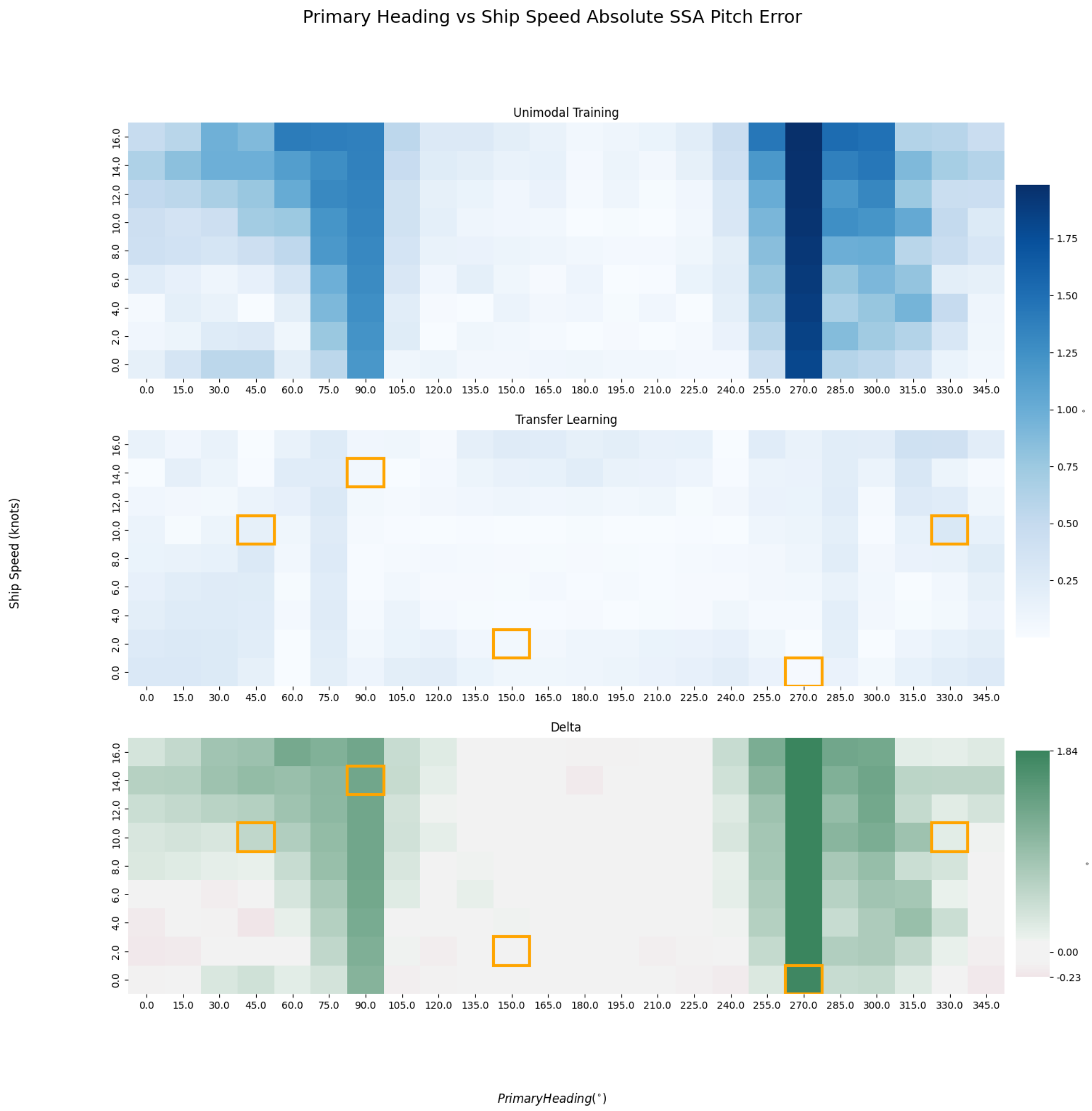

Figure 4-2: Primary wave heading angle vs ship speed pitch error: Unimodal Training and Random Transfer Learning.

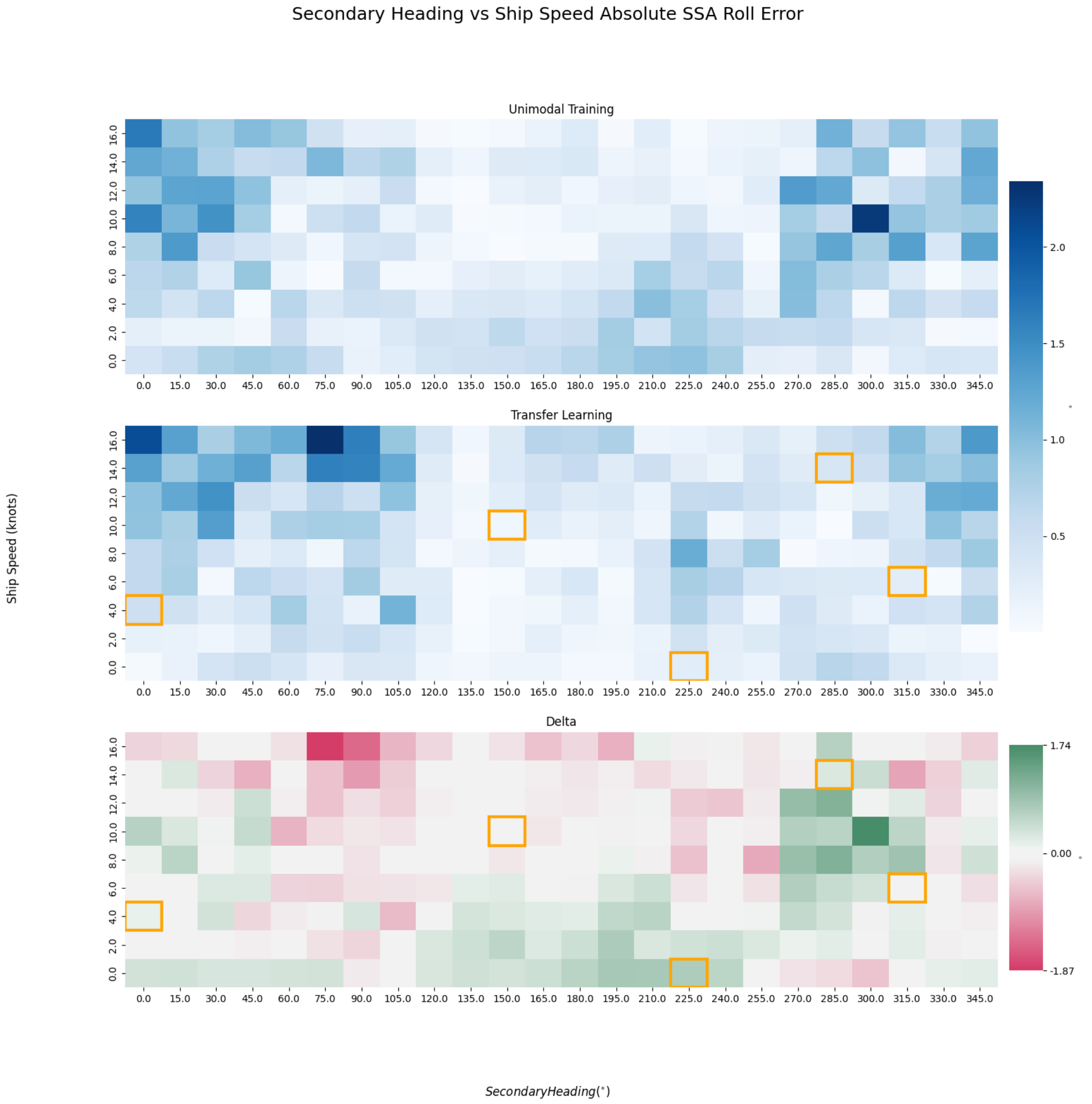

Figure 4-3: Secondary wave heading angle vs ship speed roll error: Unimodal Training and Random Transfer Learning.

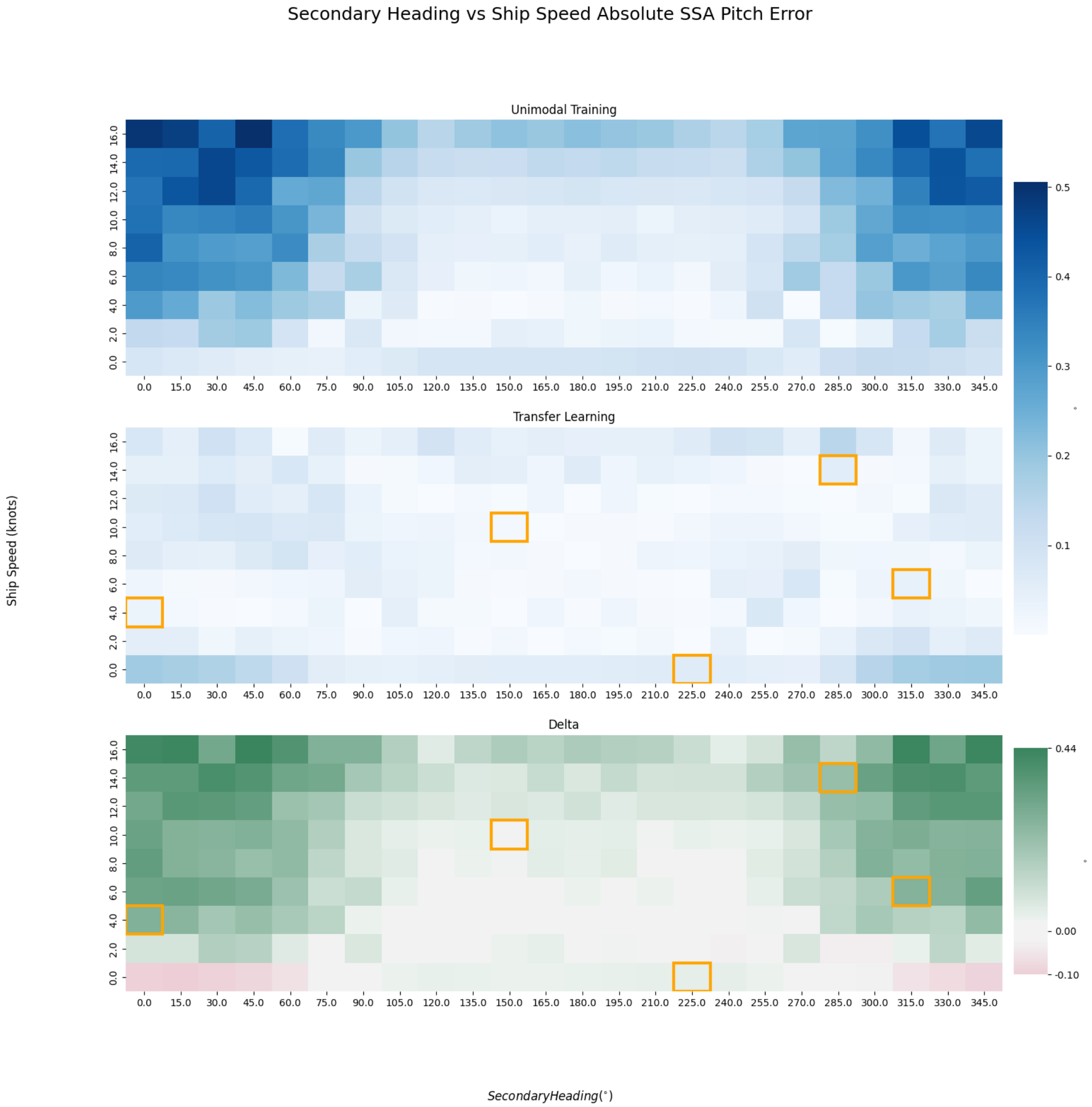

Figure 4-4: Secondary wave heading angle vs ship speed pitch error: Unimodal Training and Random Transfer Learning.

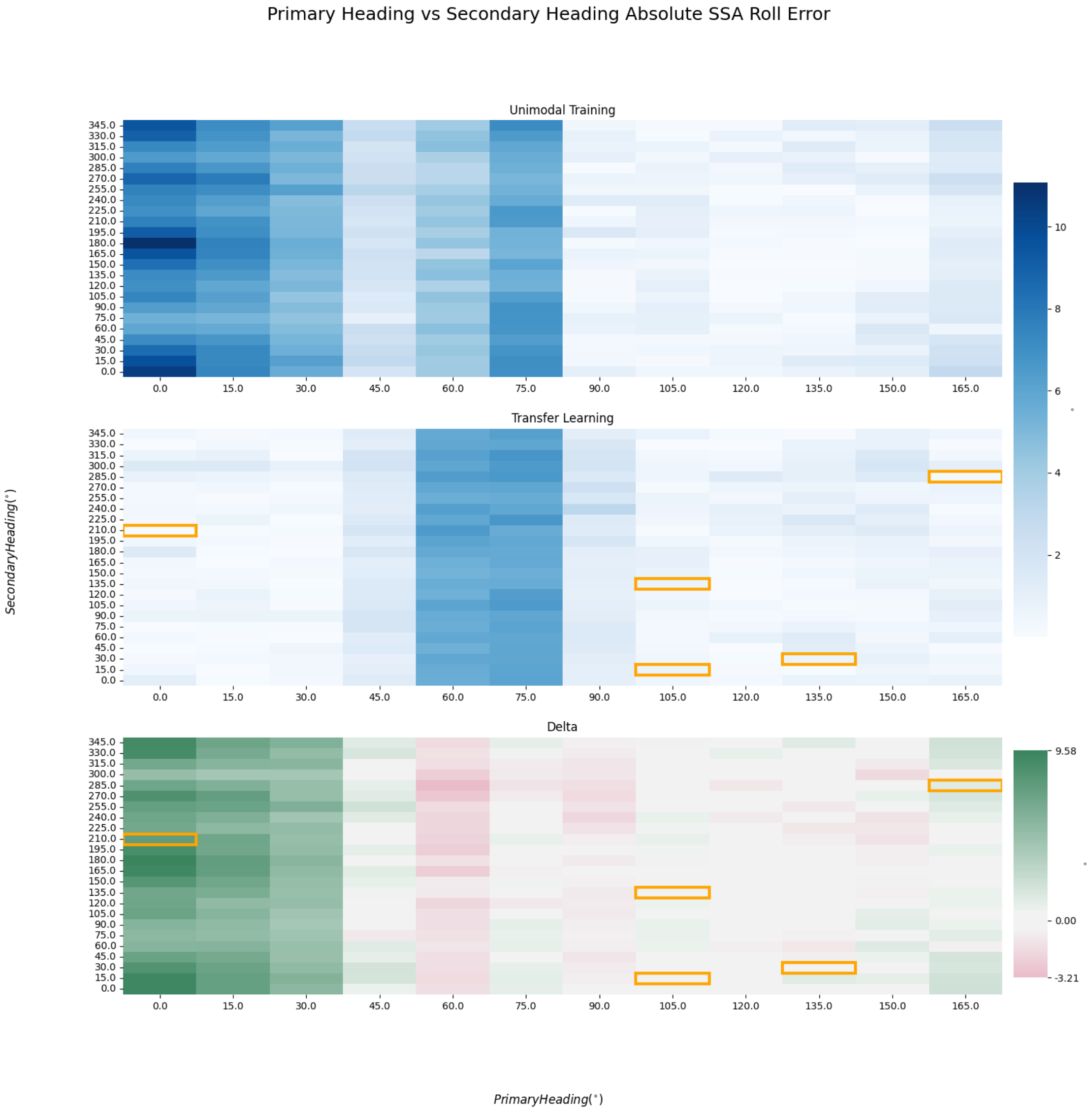

Figure 4-5: Primary vs Secondary wave heading angle roll error: Unimodal Training and Random Transfer Learning.

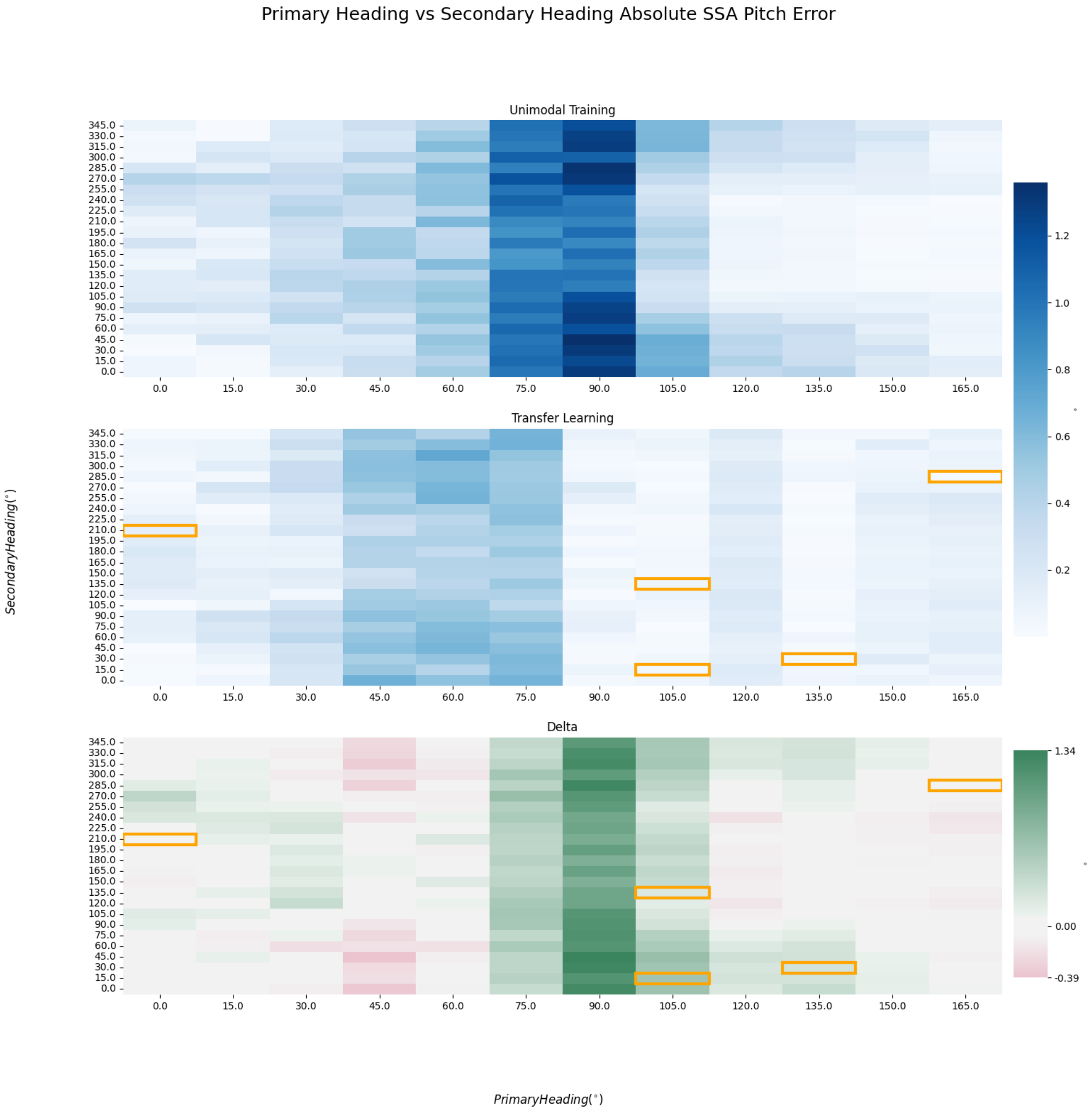

Figure 4-6: Primary vs Secondary wave heading angle pitch error: Unimodal Training and Random Transfer Learning.

4.3.2 Controlled Samples

To achieve better performance on the meta-learning model after transfer learning, the sampling process was controlled by ensuring that the samples were drawn from an even spacing throughout the domain. Furthermore, the sample size was increased to eight to sample the entire domain better. The eight samples are highlighted in yellow boxes. The samples were spaced every 90° at 12 and 14 knots to cover the whole domain. Figure 4-7 through Figure 4-12 display the results.

Pitch error experienced nearly identical improvements to random sampling. However, roll SSA error was improved slightly in the case of a primary wave heading angle variation and worsened in secondary wave heading angle variation. In the case of a primary wave heading angle variation, the roll error improved throughout the domain, and in slightly greater magnitude, except for the area where the original unimodal LSTM trained and its approximate mirror, roughly centered around 135°, 10 knots and 260° and 10 knots. Additionally, roll error generally worsens over varying secondary heading angles and speeds.

Both of these results may be indicative of “negative transfer” where we are failing to adequately answer the question of “what to transfer” [12]. In other words, we transfer information that harms and hinders the process rather than improve it. There may be several solutions to this issue, such as further training on more records of the chosen sample point, smarter sampling, better source domain selection, or a combination of all three [17]. A full investigation will be left to future work, but the source domain selection will be explored in the next section.

While there were some improvements, particularly with the primary wave heading angle, the results are still inconsistent. There is also the remaining question of how well this approach performs compared to an LSTM model trained on bimodal seas from its inception.

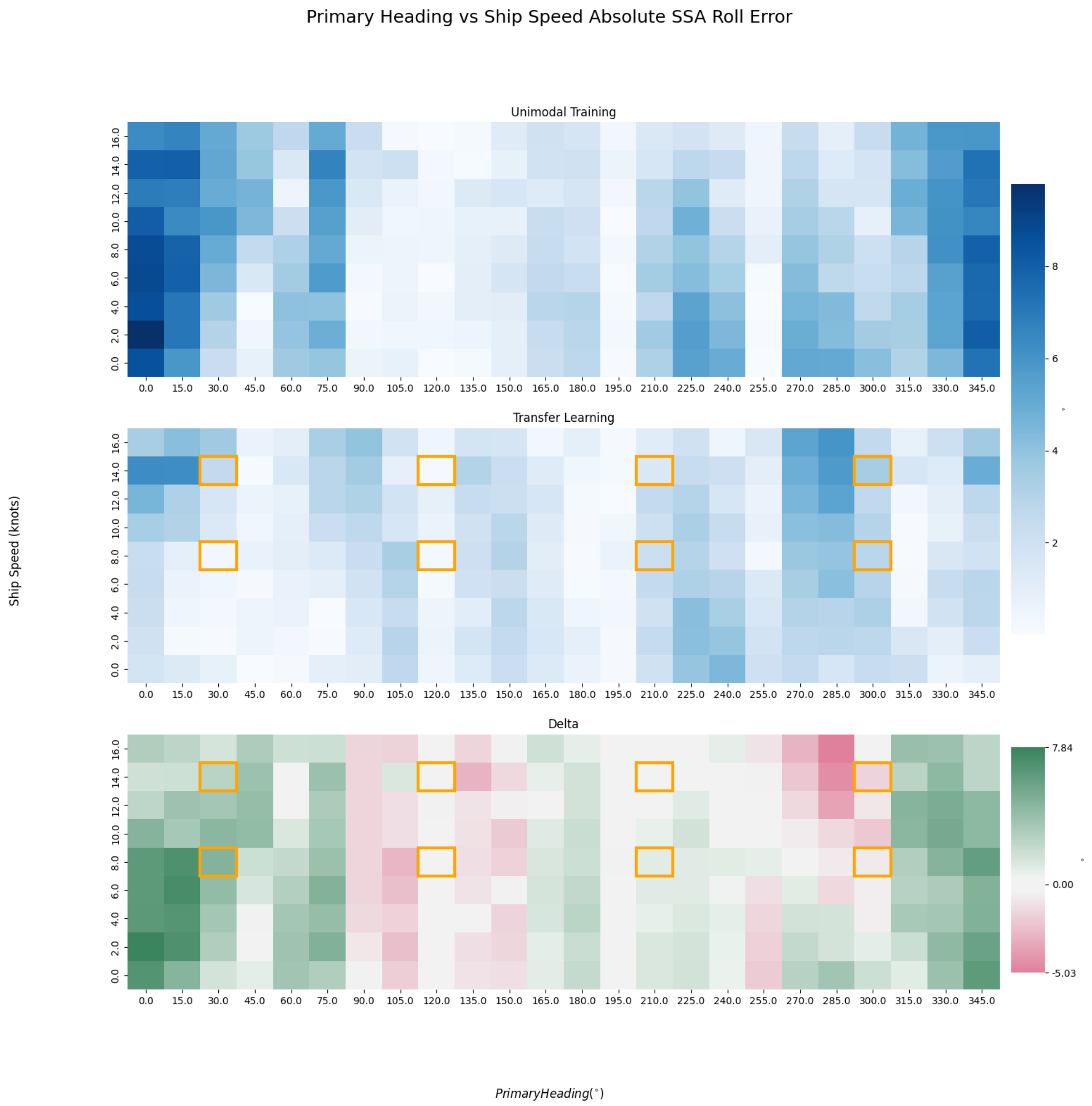

Figure 4-7: Primary wave heading angle vs ship speed roll error: Unimodal Training and Controlled Transfer Learning.

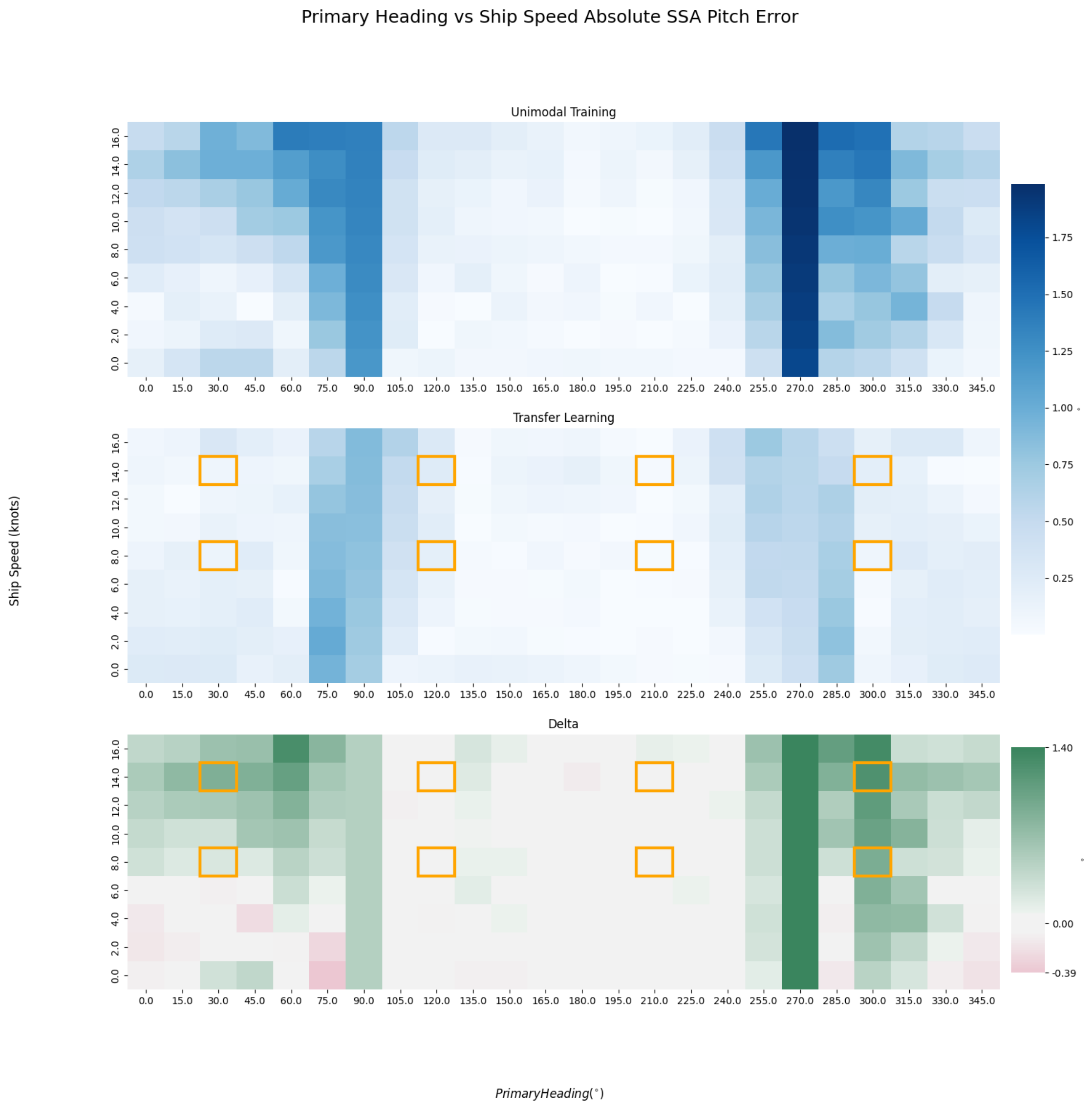

Figure 4-8: Primary wave heading angle vs ship speed pitch error: Unimodal Training and Controlled Transfer Learning.

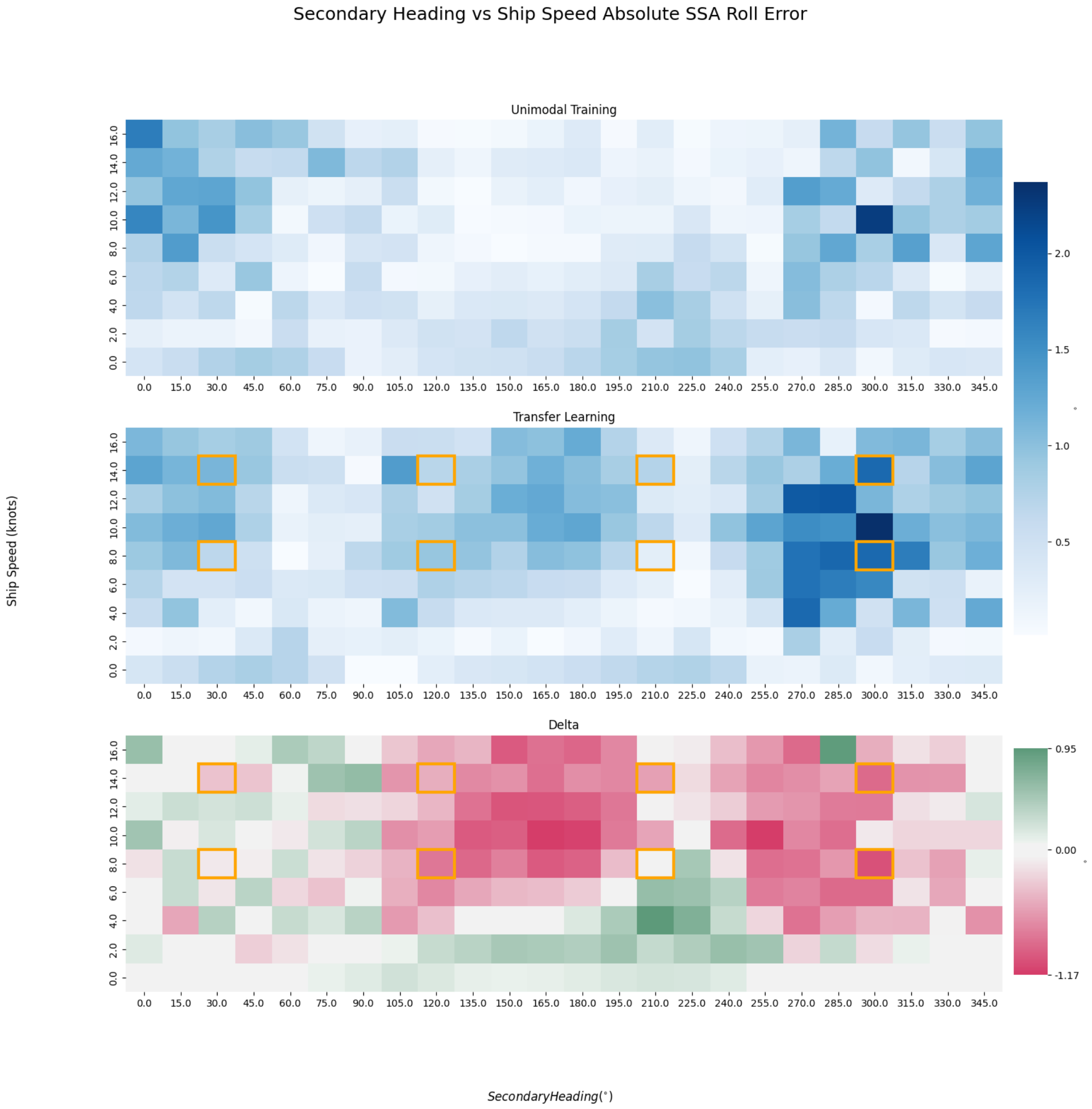

Figure 4-9: Secondary wave heading angle vs ship speed roll error: Unimodal Training and Controlled Transfer Learning.

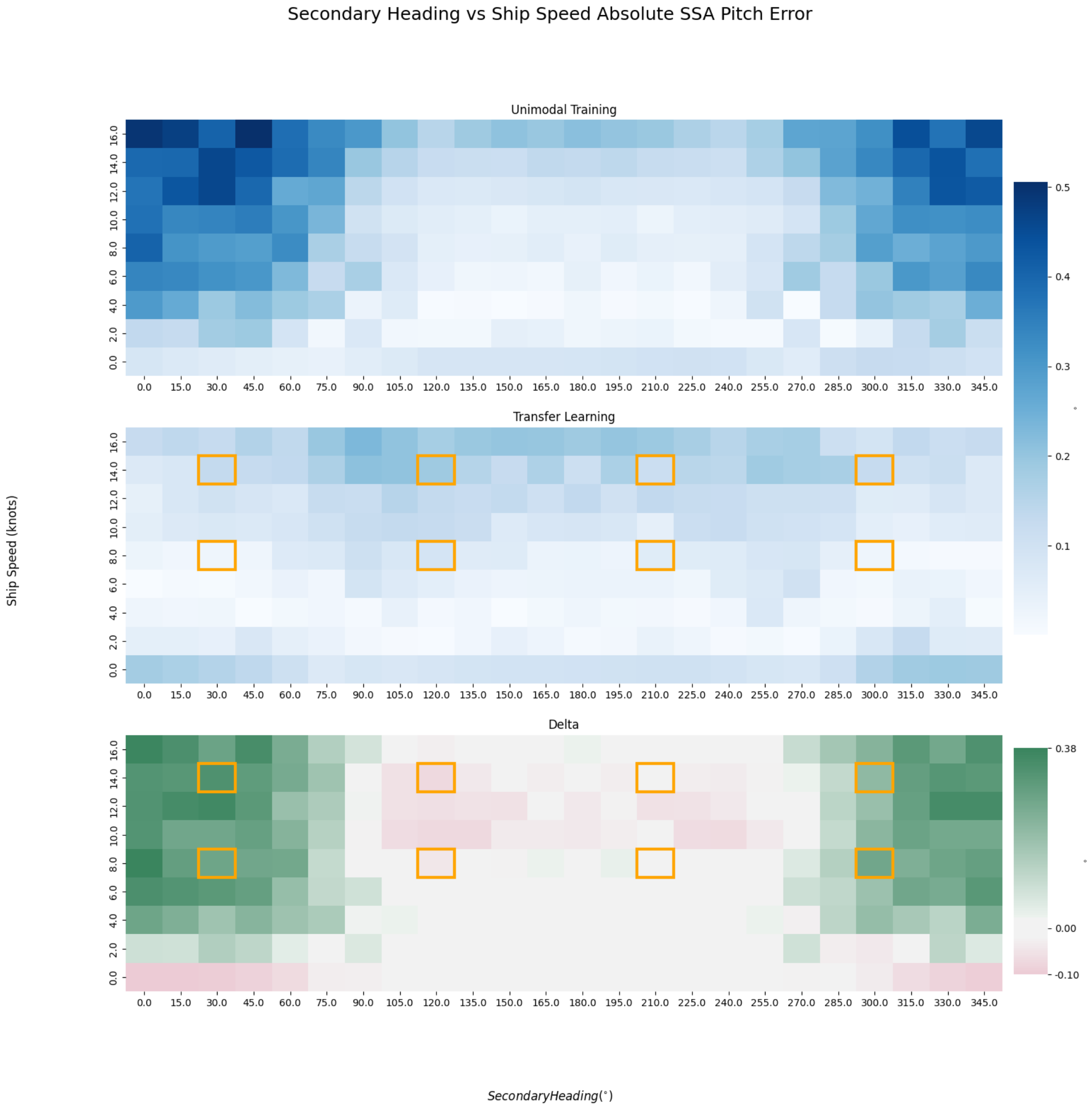

Figure 4-10: Secondary wave heading angle vs ship speed pitch error: Unimodal Training and Controlled Transfer Learning.

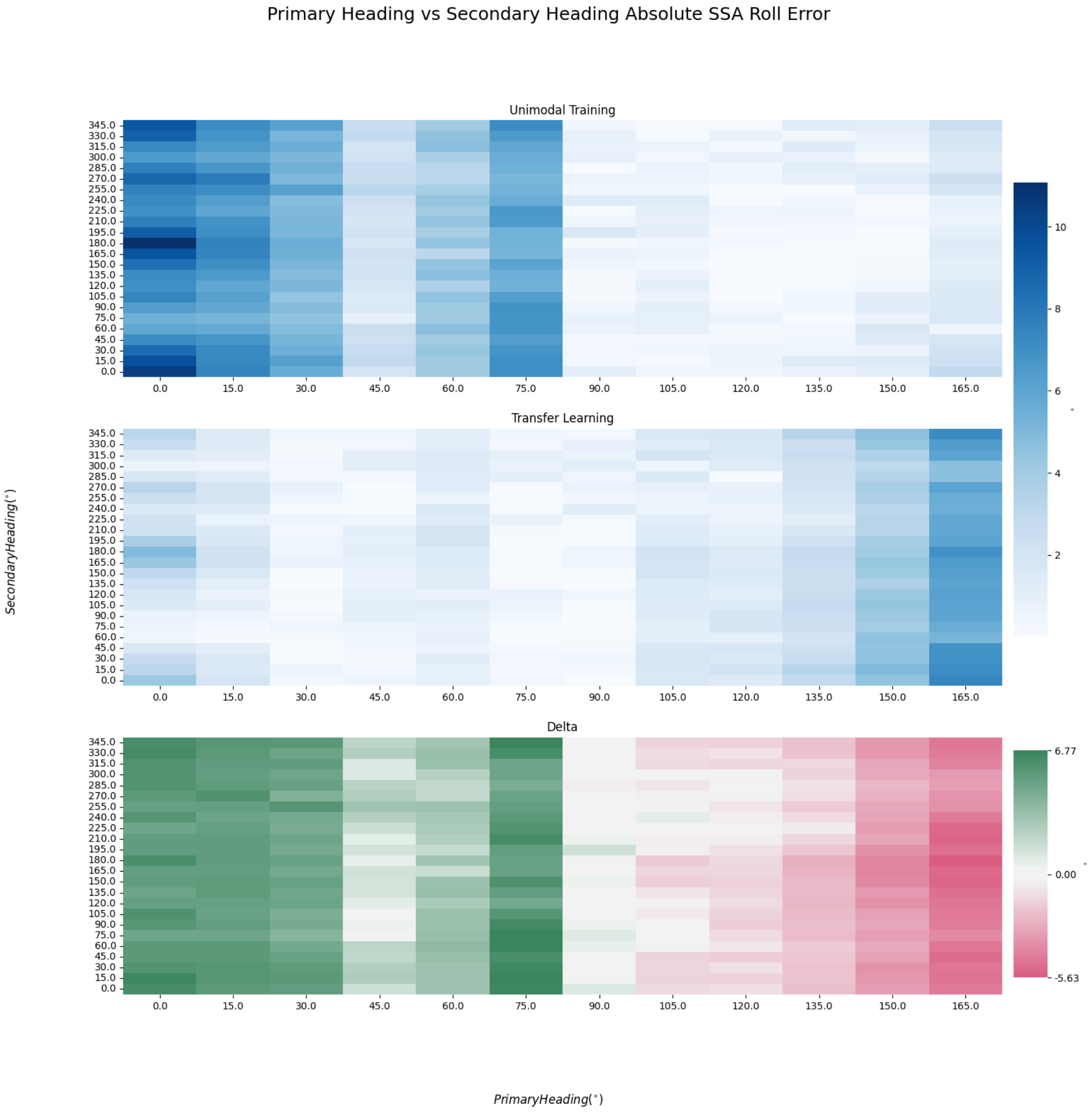

Figure 4-11: Primary vs Secondary wave heading angle roll error: Unimodal Training and Controlled Transfer Learning.

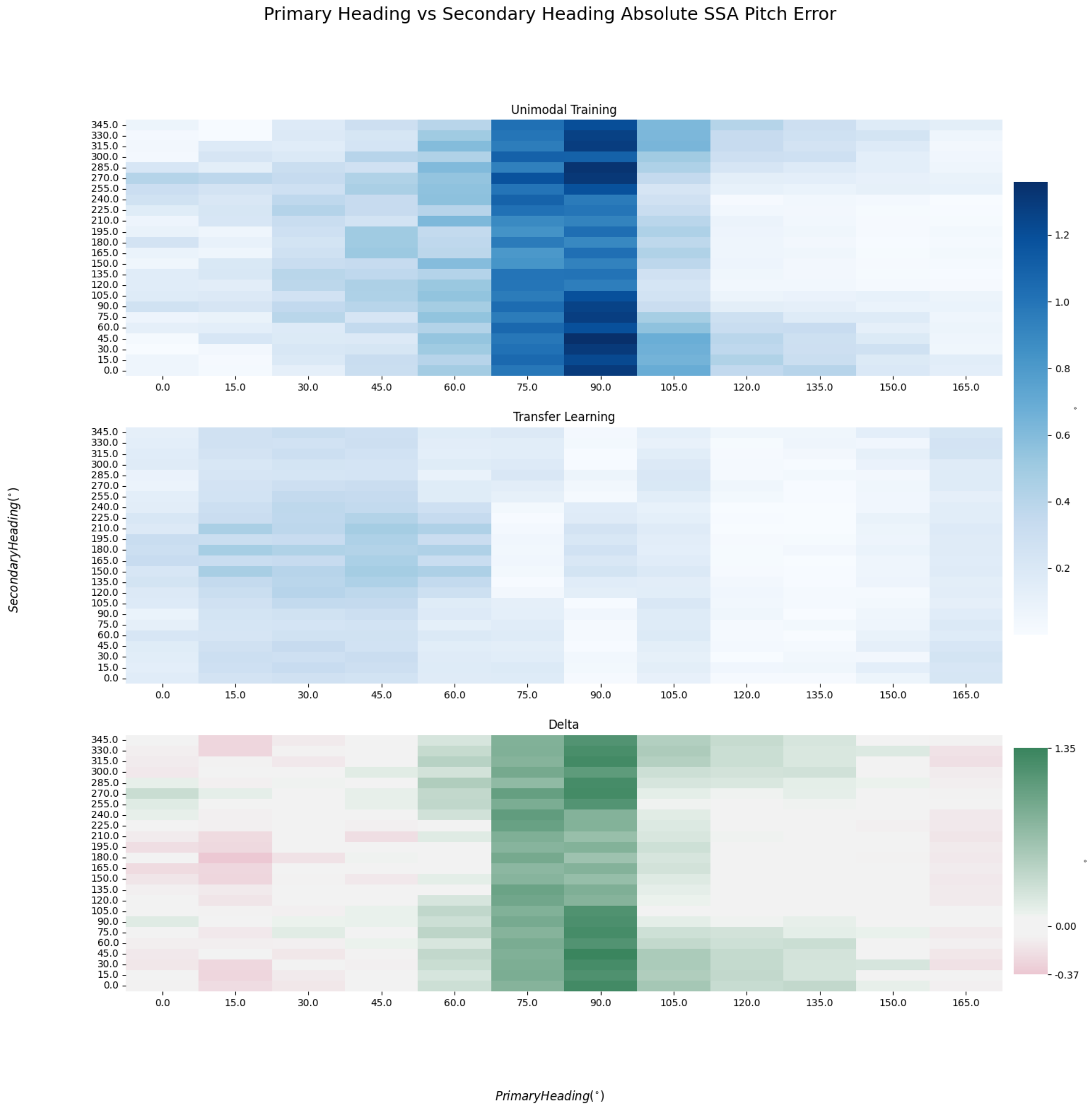

Figure 4-12: Primary vs Secondary wave heading angle pitch error: Unimodal Training and Controlled Transfer Learning.

4.4 Bimodal LSTM Comparison

In this chapter, the source domain serving as the initial base network is revisited and compared to an LSTM trained exclusively on bimodal wave sets. Random sampling was briefly explored but consistently showed little difference in performance between a bimodal LSTM and a unimodal LSTM that experienced additional bimodal training.

Therefore, the results of this chapter will focus on the controlled sampling approach in all comparisons with eight samples.

4.4.1 Initial Training Domain Expansion

In an attempt to further investigate the driving factors behind the results shown previously, the Medium LSTM training domain at its creation was revisited. To recap the relevant portions of Section 3, the Medium LSTM was initially trained on 81 records of unimodal seas. The 81 records stem from enumerating all possible combination pairs of variable values listed in Table 3.1 [8]. This ultimately represents a tiny portion of the primary seas experienced in both the expanded unimodal and bimodal domains described in Table 3.4 and may have contributed to the inconsistent results. I.e., the assumption that the unimodal LSTM would be a quality starting point for the base network is invalid when the initial training domain is so small compared to the testing domain. The primary wave heading angle vs. ship speed domain alone comprised 216 different record sets of input parameters.

Thus, the testing domain for primary and secondary wave heading angles was reduced to accommodate a larger training domain. This facilitated the recreation of a baseline Medium LSTM familiar with the entire unimodal domain before any transfer learning occurs. The testing domain removed all ship speeds deemed unrealistic for the scenario: anything less than six knots. Simultaneously, the training domain was expanded to include all possible combinations of the primary wave heading angles and ship speed.

4.4.2 Bimodal LSTM Comparison

A traditional LSTM initialized with random parameters and trained on the identical eight records was created to serve as the new and final baseline comparison. Bimodal LSTMs consistently outperformed the Narrow, Medium, and Wide LSTMs in bimodal seas as detailed by Howard [8] and in Section 3.

4.4.3 Results

The expansion of the initial training domain had a noticeable effect on the results. The most significant effect is seen in the roll error, particularly in varying primary wave headings, which also corresponds with the largest errors seen previously. The pitch error for varying primary wave heading also largely improved. Neither of these results is surprising as so much additional unimodal training was provided to the base LSTM. Figure 4-13 and Figure 4-14 show the results of the primary roll and pitch error.

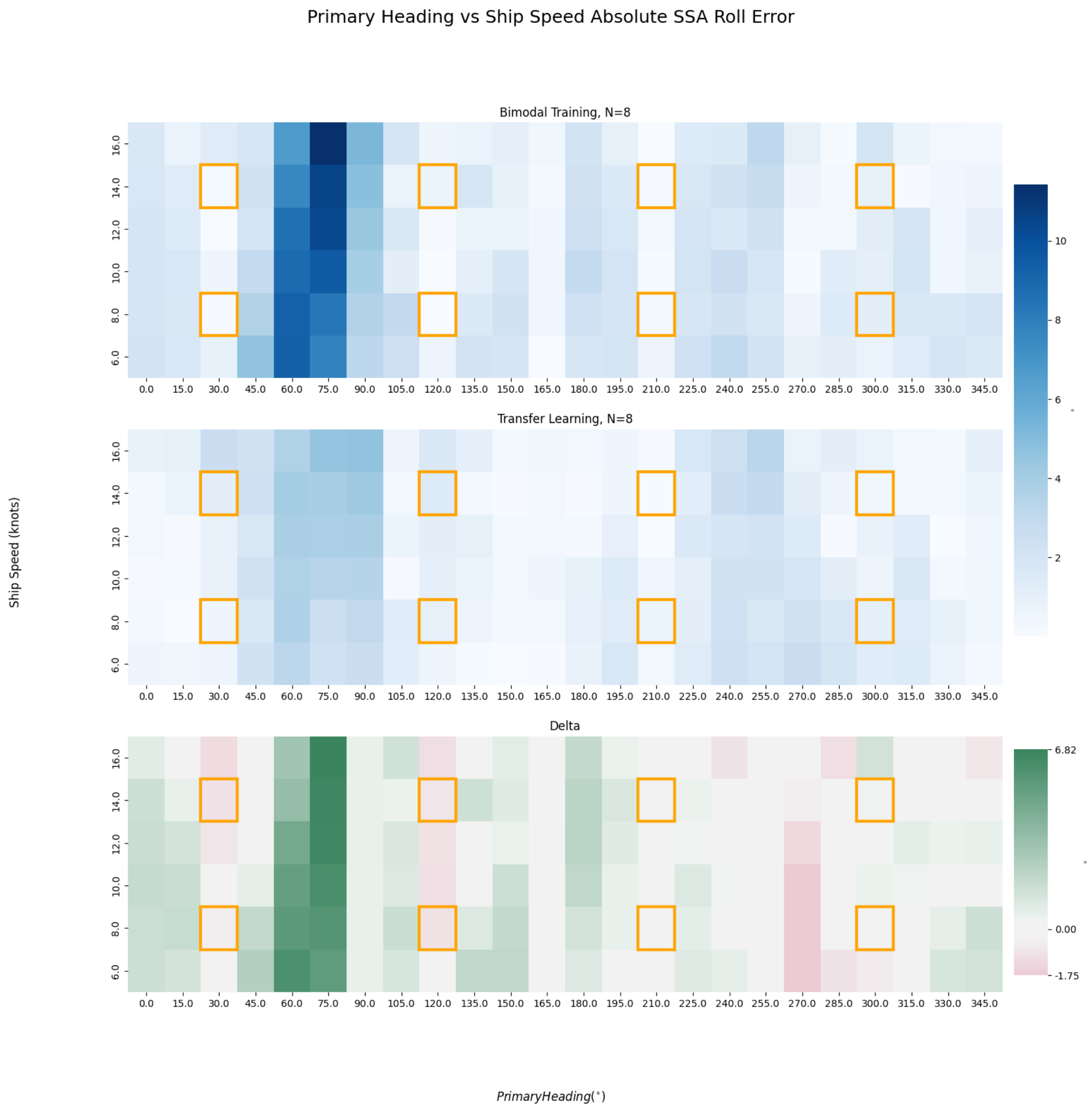

Figure 4-13: Primary wave heading angel vs ship speed roll error: Bimodal Training and Controlled Transfer Learning.

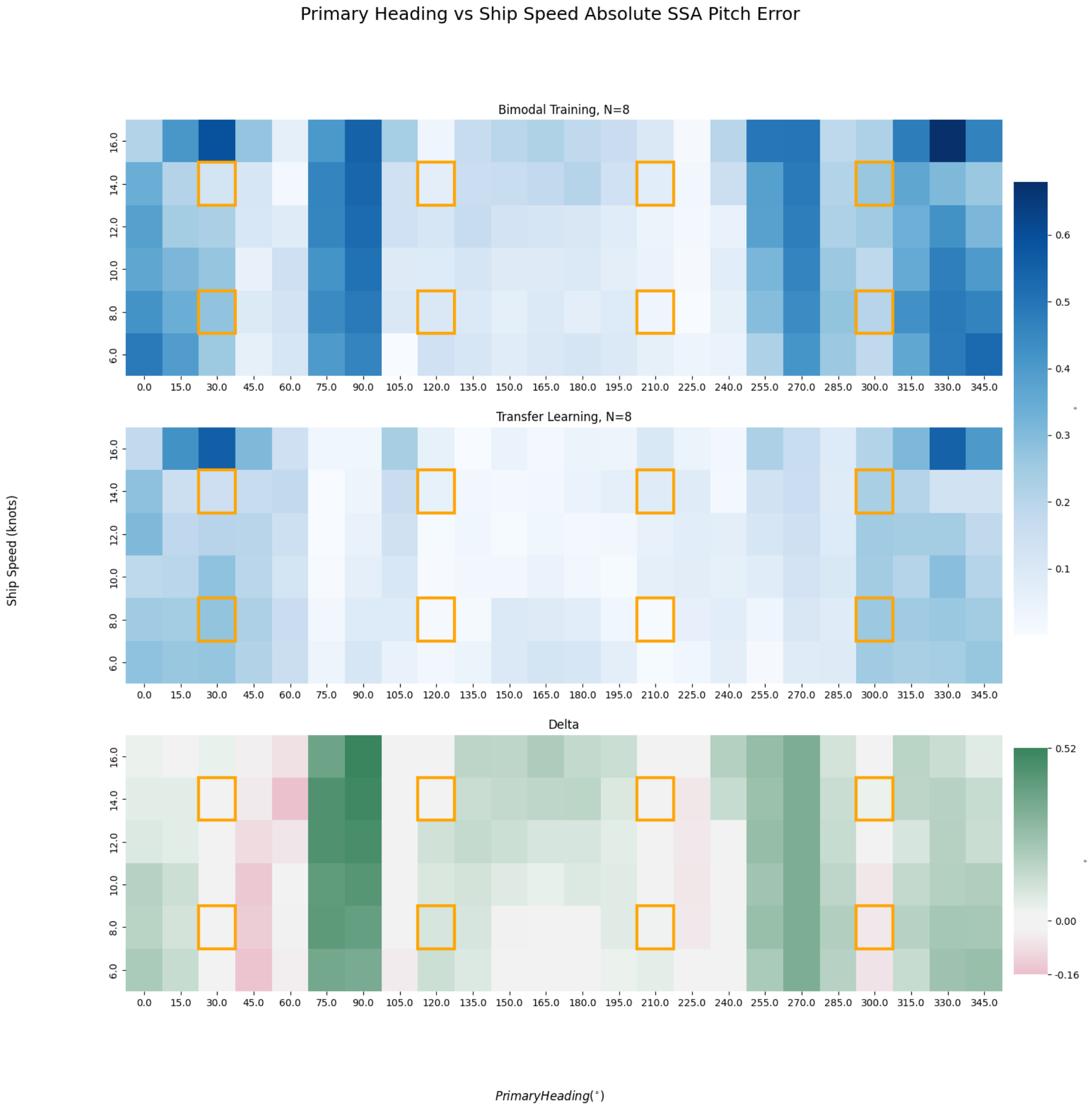

Figure 4-14: Primary wave heading angel vs ship speed pitch error: Bimodal Training and Controlled Transfer Learning.

While the primary wave headings test case errors were consistent and predictable, the secondary ones were far more interesting. The results were inconsistent, with the bimodal LSTM outperforming the unimodal LSTM with transfer learning in some records and equally performing or underperforming in others. Through multiple reperformances of the tests, the results were consistently varying. One explanation may be that the results were ultimately driven by the random initialization of the bimodal LSTM, meaning the unimodal LSTM and transfer learning process would not be a great replacement for a quality bimodal network without further investigation. Figure 4-15 and Figure 4-16 show results corresponding to the same test as Figures 4-13 and 4-14. Additional figures are included in Appendix B and demonstrate a test where the bimodal LSTM underperformed compared to the unimodal LSTM with transfer learning on the secondary wave heading test case.

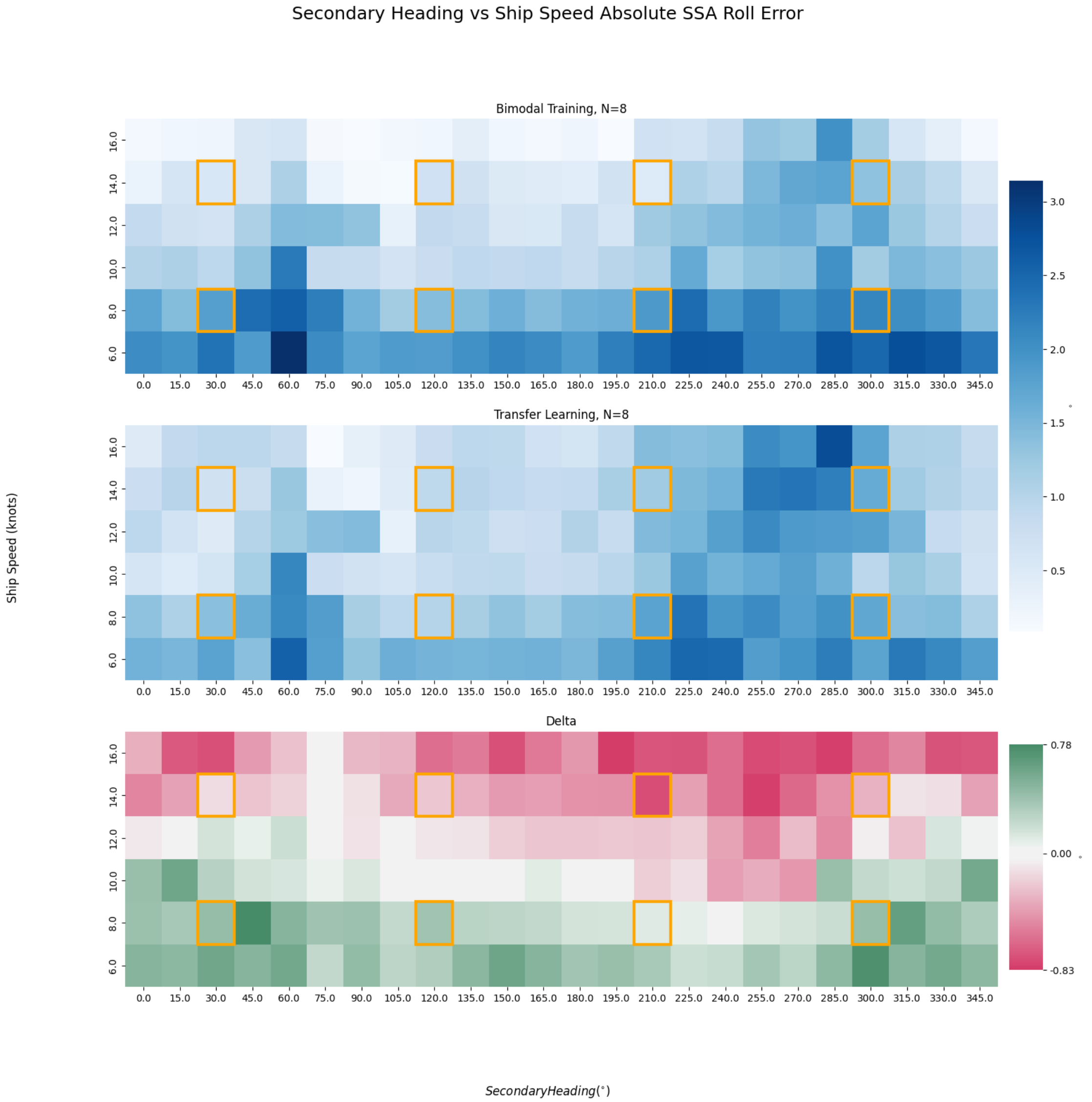

Figure 4-15: Secondary wave heading angle vs ship speed roll error: Bimodal Training and Controlled Transfer Learning Run 1.

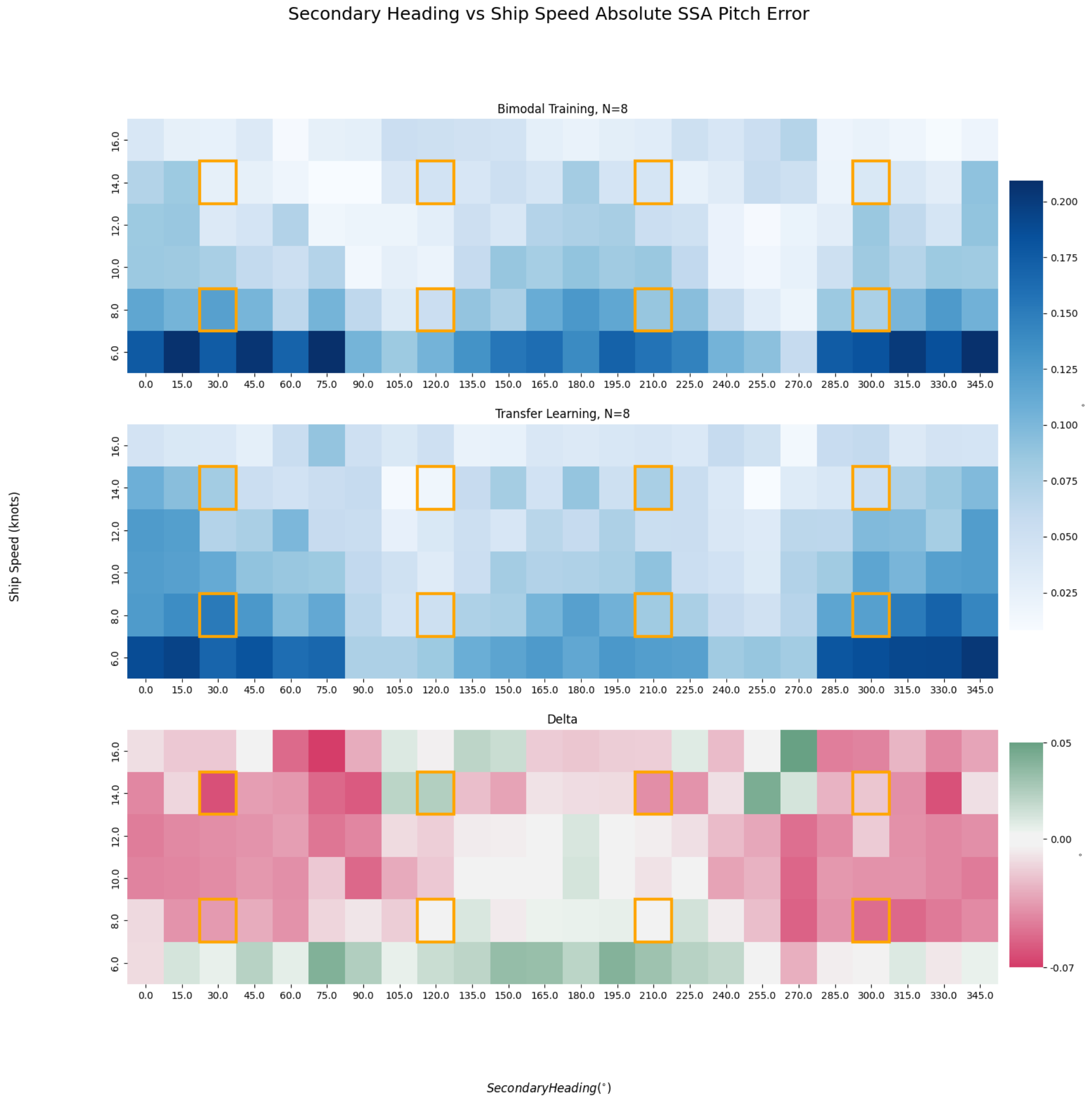

Figure 4-16: Secondary wave heading angle vs ship speed pitch error: Bimodal Training and Controlled Transfer Learning Run 1.

4.4.4 Shifted Sampling

A final effort was made to "expand" the bimodal training domain by shifting the bimodal and transfer learning sampling points to cover the full spread of the domain better. While this was not investigated in detail, it did provide promising results. One set of speeds was chosen to shift by approximately 45° rather than 90° to cover better the entire domain of primary wave headings, including the beams seas the LSTM often performs poorly on. This resulted in a noticeable improvement in both the bimodal LSTM and the Medium LSTM. Figures 4-17 and 4-18 show the results of the primary wave heading test case roll and pitch error.

While the magnitude of the error correction difference between the two decreased, the overall performance of the Medium LSTM improved substantially. Figure 4-13 shows the highest SSA roll error occurring around the beams with an error of approximately 6°. Figure 4-17 a reduction in the error in the beam seas case to approximately 2.5°, likely due to the bimodal training the LSTM received on the beam. Similarly, one sample near the beam drastically improved some of the bimodal LSTM's worst-performing conditions. This demonstrates that smarter sampling may significantly improve the performance of both LSTMs.

In this case, the pitch error for the Medium LSTM performed very comparably to or slightly worse than the bimodal LSTM. However, it again greatly outperformed the bimodal LSTM in the set of conditions where the bimodal LSTM performs the worst without having any set of conditions where the Medium LSTM performed extremely poorly compared to the bimodal LSTM. In other words, the Medium LSTM with transfer learning did not experience the worst pitch errors shown by the bimodal LSTM while having little to no poorly performing areas.

Varying secondary wave headings again had a more mixed result. While the Medium LSTM with transfer learning outperformed the bimodal LSTM in the case of pitch error, the same cannot be said for roll error. This may still indicate too much bias of the Medium model toward the primary wave conditions. Figures 4-19 and 4-20 show the results of the primary wave heading test case roll and pitch error.

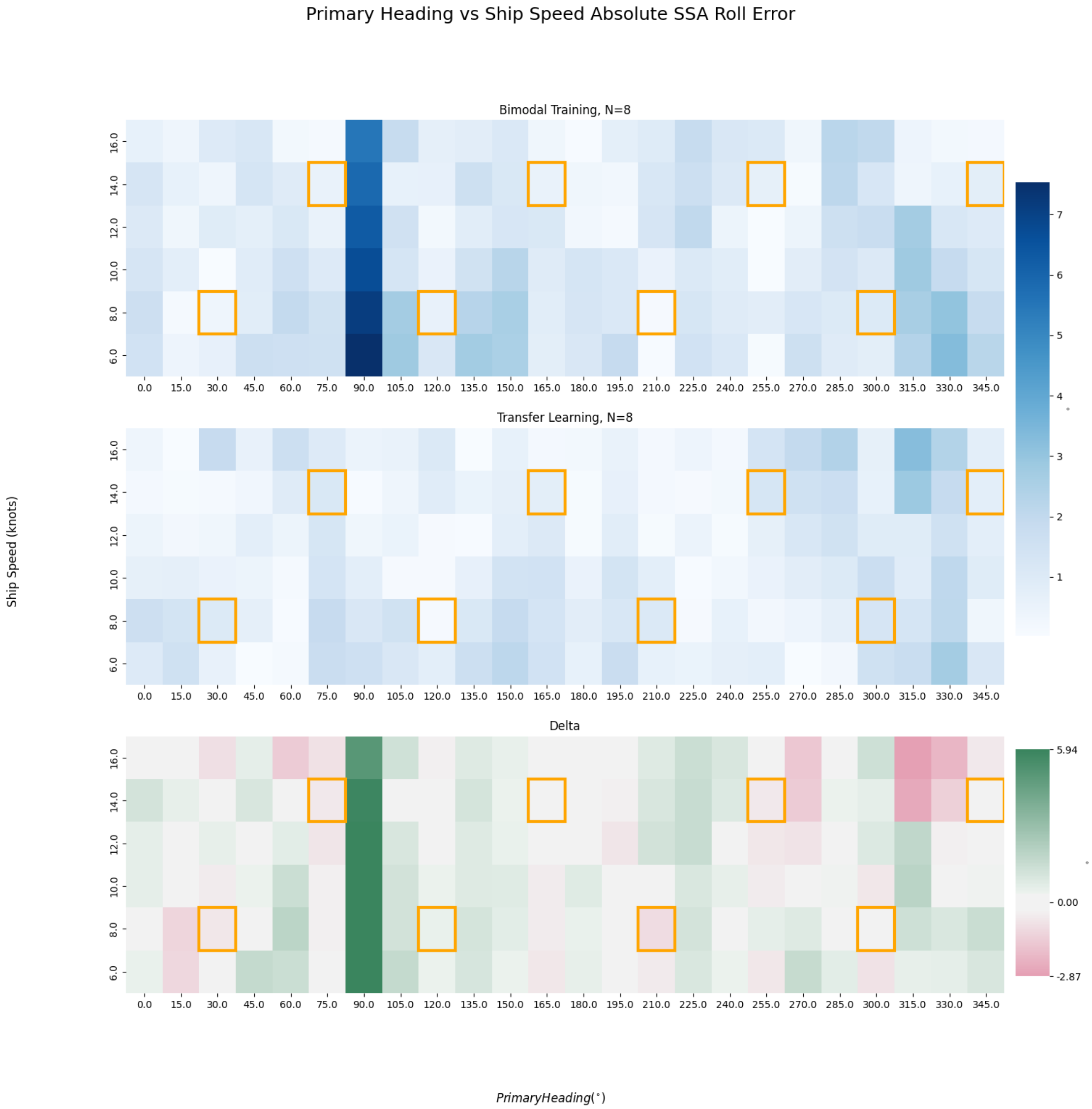

Figure 4-17: Primary wave heading angel vs ship speed roll error: Shifted sample.

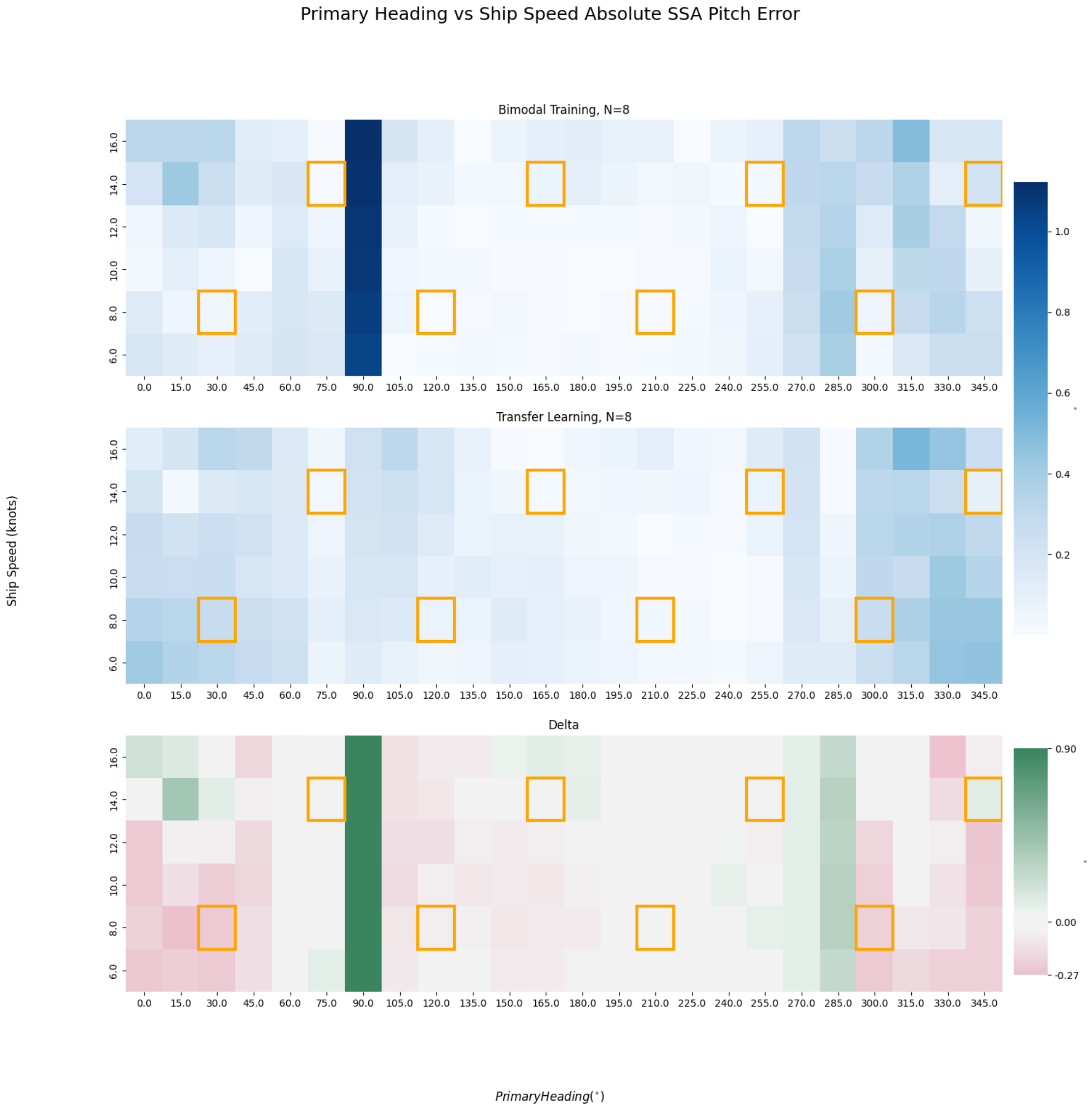

Figure 4-18: Primary wave heading angel vs ship speed pitch error: Shifted sample.

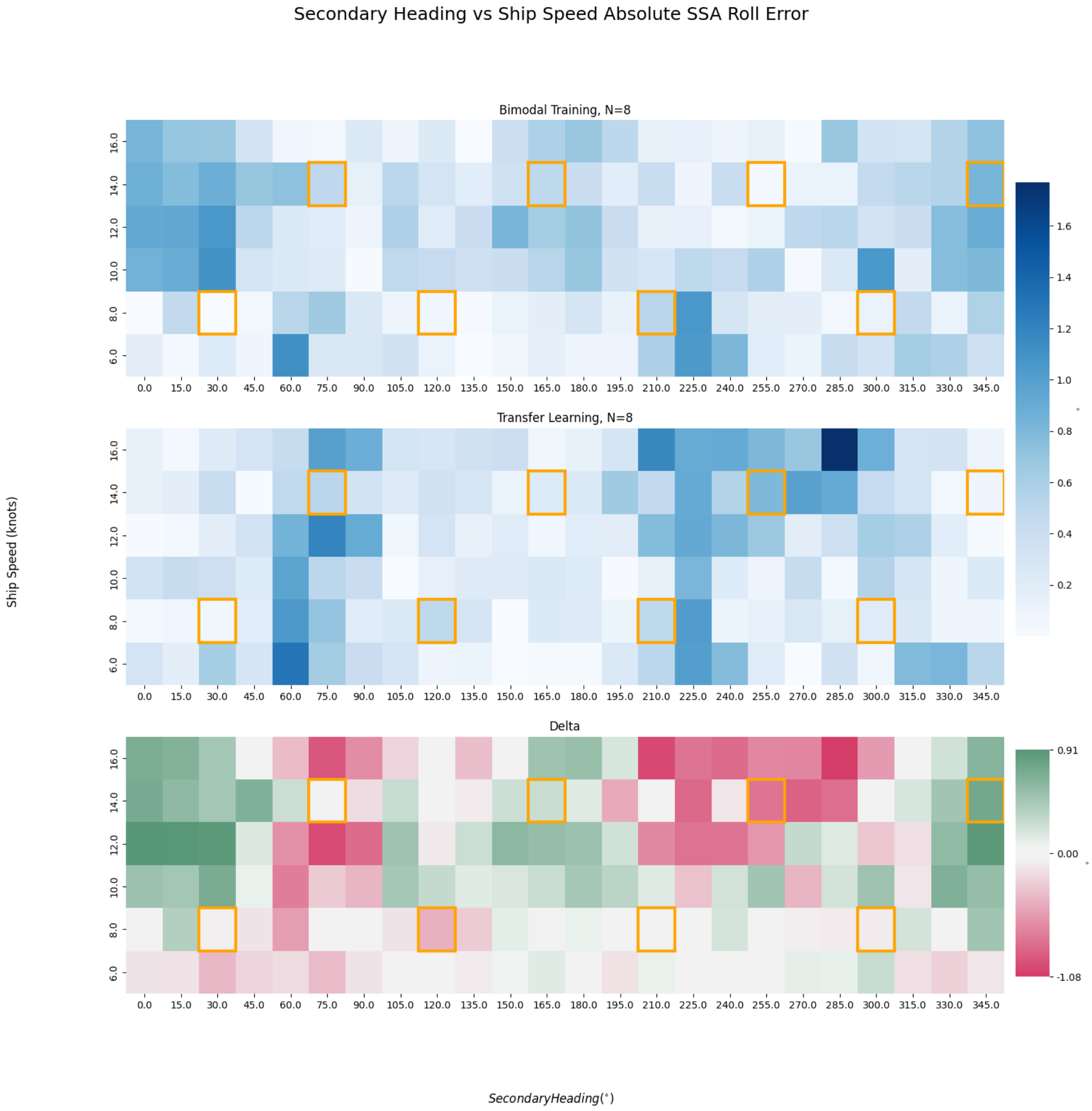

Figure 4-19: Secondary wave heading angle vs ship speed roll error: Shifted sample.

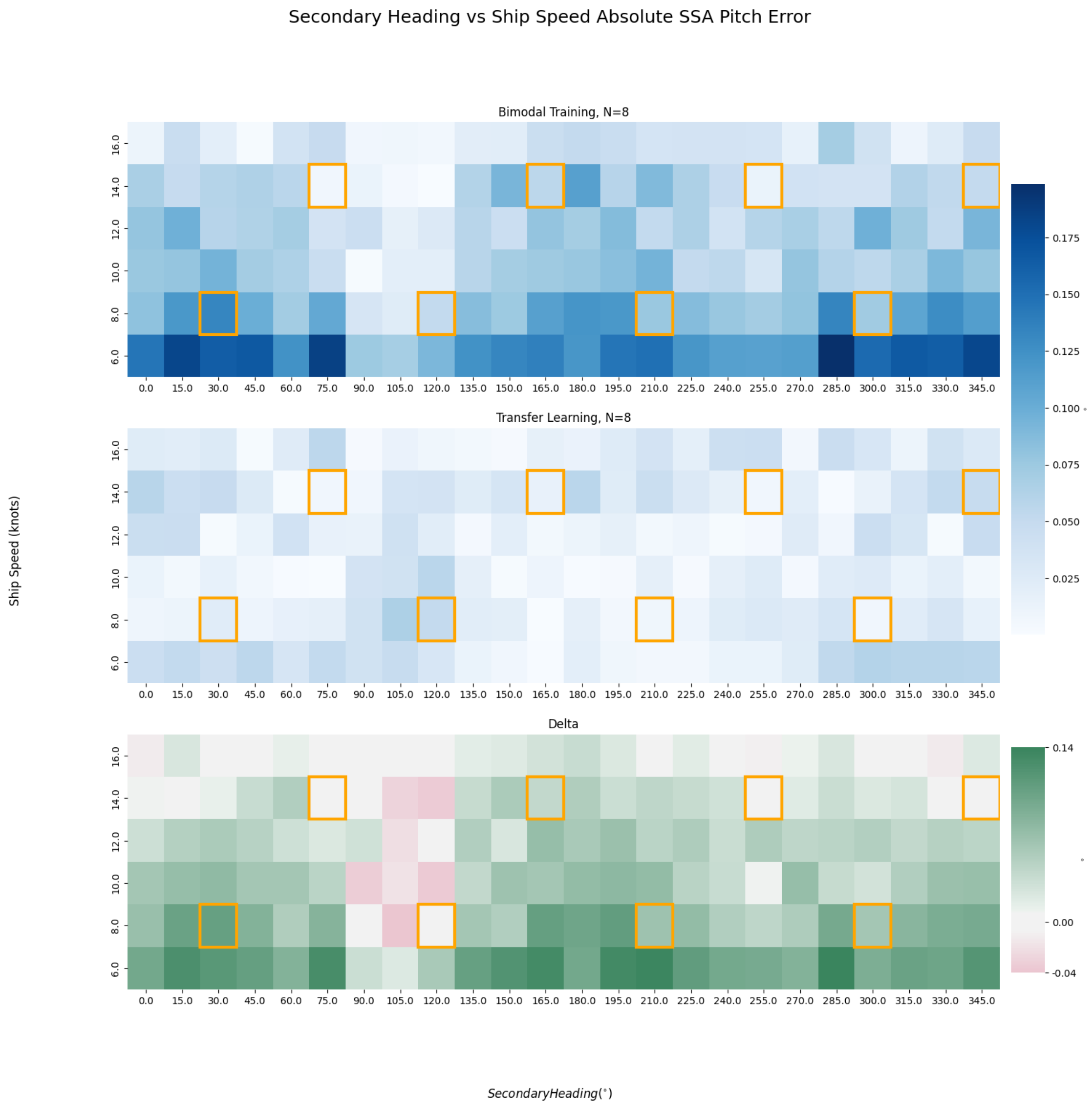

Figure 4-20: Secondary wave heading angle vs ship speed pitch error: Shifted sample.

4.4.5 Training Time

The computation training time for each experiment was also tracked, both bimodal and Medium LSTMs, for the eight controlled samples. Figure 4-21 shows these results, and Table 4.1 maps the results to the varied parameter pairs. The training time required to fine-tune the Medium LSTM was always less than that of training the bimodal LSTM, and it was often half or one-third as long. This shows the potential speed with which the transfer learning approach can leverage while still performing on par or better than the bimodal LSTM.

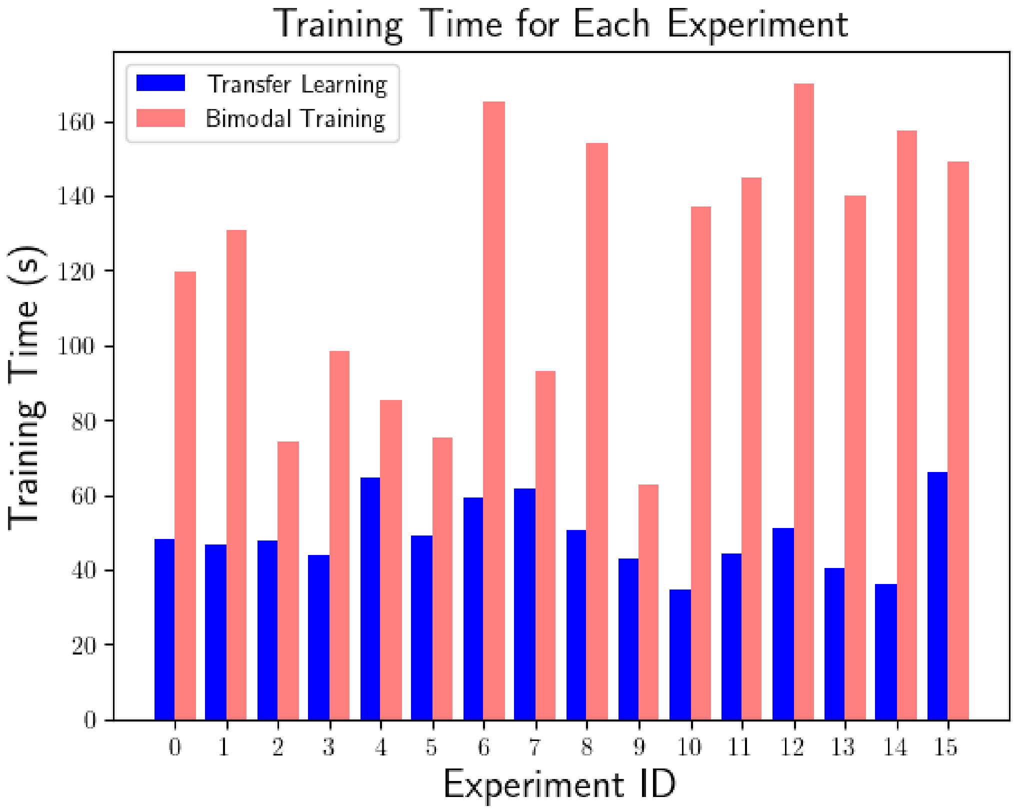

Figure 4-21: Training Time for Bimodal LSTM and Medium LSTM

Experiment ID	**Parameters Varied**
0	Sec Heading and Ship Speed
1	Pri Heading and Sec Heading
2	Pri Heading and Sec Height
3	Pri Heading and Sec Period
4	Pri Heading and Ship Speed
5	Sec Height and Sec Heading
6	Pri Height and Sec Period
7	Sec Height and Ship Speed
8	Pri Height and Pri Heading
9	Pri Height and Sec Heading
10	Pri Height and Sec Height
11	Pri Height and Pri Period
12	Pri Height and Sec Period
13	Pri Height and Ship Speed
14	Sec Period and Sec Heading
15	Sec Period and Ship Speed

Table 4.1: Experiement ID to parameter pairs mapping.

Chapter 5

Conclusions and Recommendations

This thesis expanded upon previous work using an LSTM model to predict ship motion statistics with a focus on bimodal seas and transfer learning. The conclusion of this thesis is:

- The LSTM model is a viable method for producing a low error map between SimpleCode and LAMP ship motion statistics in a range of bimodal seas.
- Inductive transfer learning through a model agnostic meta-learning framework using LSTMs trained on unimodal seas as base initialization models can be used to cover a wider test domain but still requires investigation to produce consistent results.
- Only a few samples are needed to alter the base model's predictions to cover a specific set of wave parameters.
- The training time to produce and fine-tune a base model to a specific set of conditions is much less than the time required to initialize and train a new model from scratch.
- There is great value in expanding the initial training domain of the base model, thereby improving the model's ability to generalize to new wave conditions.

- Training record selection for fine-tuning the base model has a noticeable effect on model performance.

There are many avenues to explore for future work. The following discusses potential routes in no particular order of importance:

- While the transfer learning process described generally improved a given model's performance against a baseline case, the results were not always consistent. Determination of the driving factors behind this inconsistency could open the opportunity to improve the model's trustworthiness.

- Improve the MAML framework used in this thesis to allow for better generalization of the base model. This may be done using new techniques, such as Reference [1] or [13], or by exploring the effects of changing the framework itself by redefining the base model initialization, task distribution, and learning tasks.

- Explore other models, such as convolutional neural networks.

- Improve the wave input, which is treated as a single point centered on the hull for this thesis. The framework for a wave grid is in place and may have promising results for generality, but it was not explored to maintain a focus on comparison to past results.

- Conduct a hyperparameter search for the LSTM model and transfer learning process. This thesis used fixed hyperparameters from previous work. The networks were expanded briefly to 3 layers of 50 units and showed promising results, but this was not explored in depth to stay consistent with the parameters of this and previous work.

- Improve the sampling method. The methods used in this thesis were straightforward. Even a modicum of thought put into the sampling process improved results greatly.

Appendix A

Bimodal Seas Expanded Domain Resultant Heatmaps

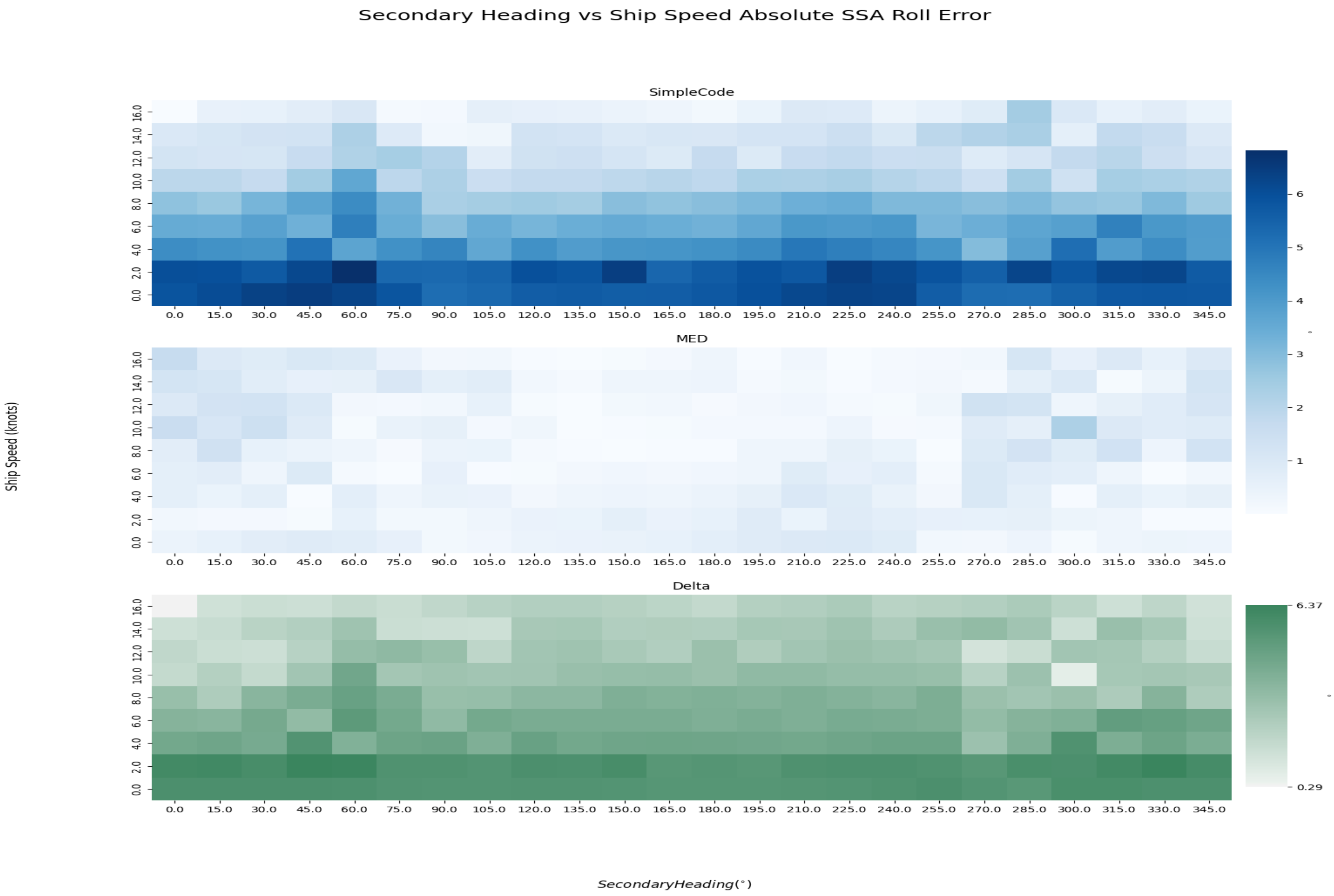

Figure A-1: Secondary wave heading vs Ship Speed absolute SSA roll error.

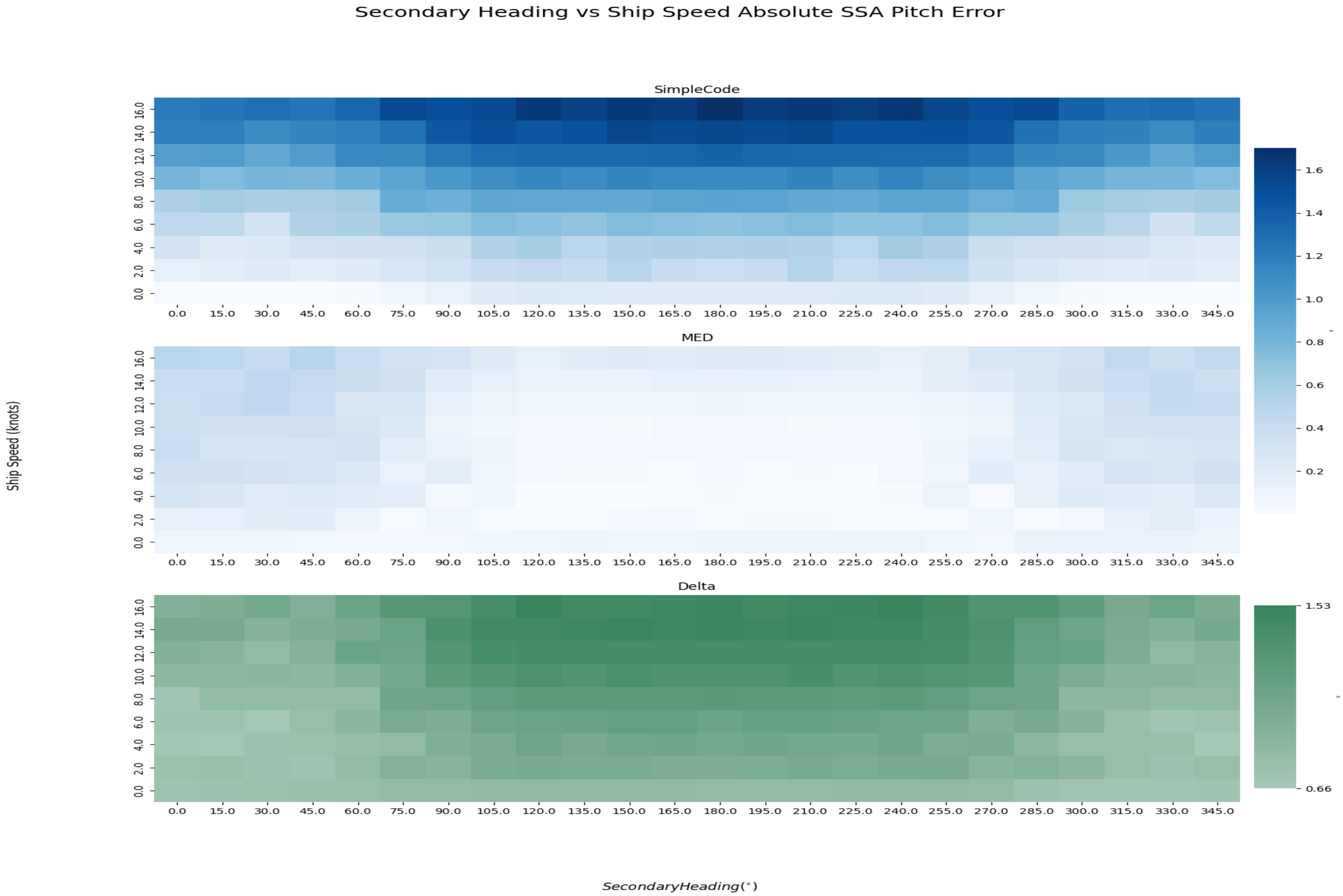

Figure A-2: Secondary wave heading vs ship speed absolute SSA pitch error.

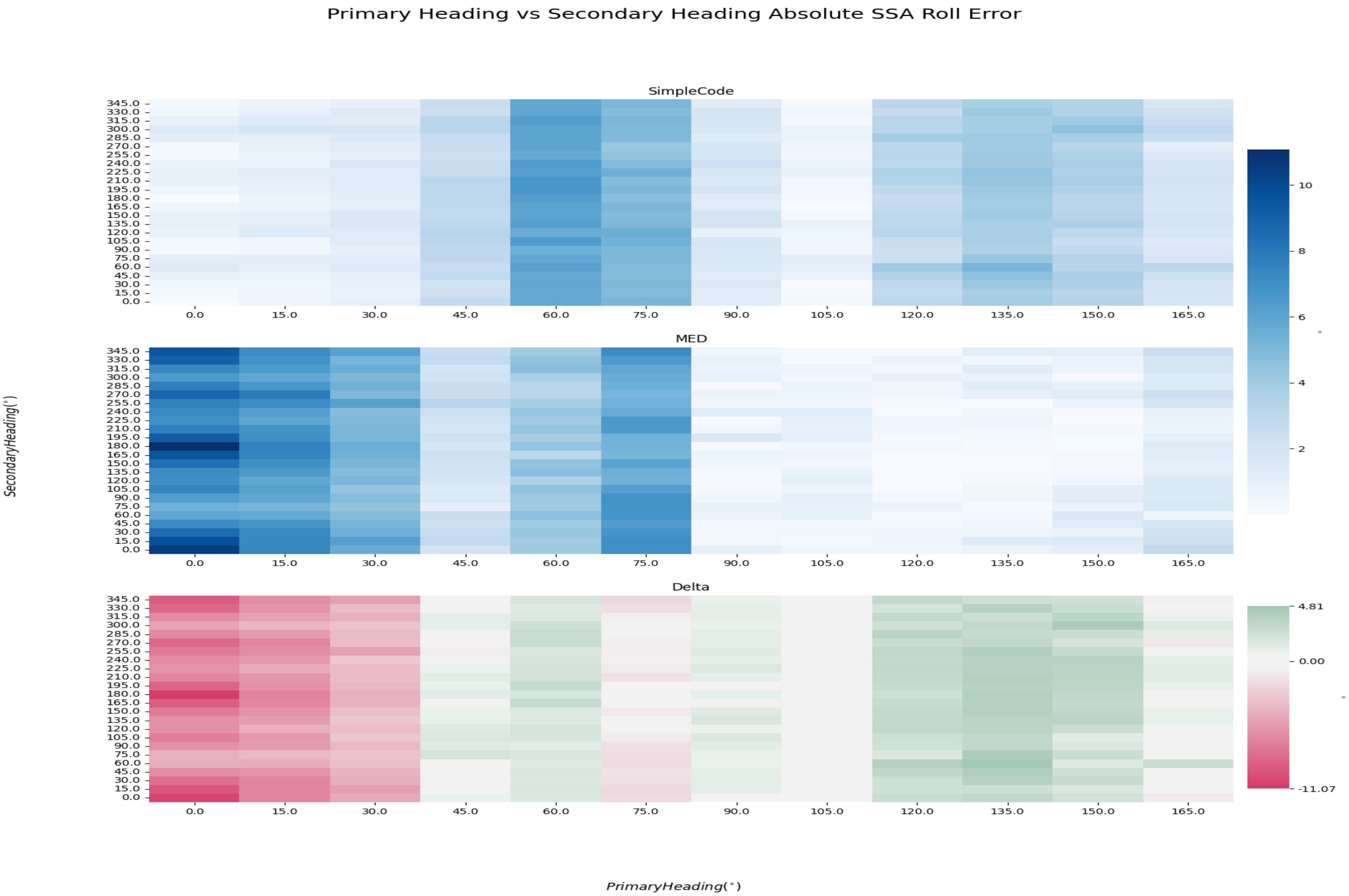

Figure A-3: Primary wave heading vs secondary wave heading absolute SSA roll error.

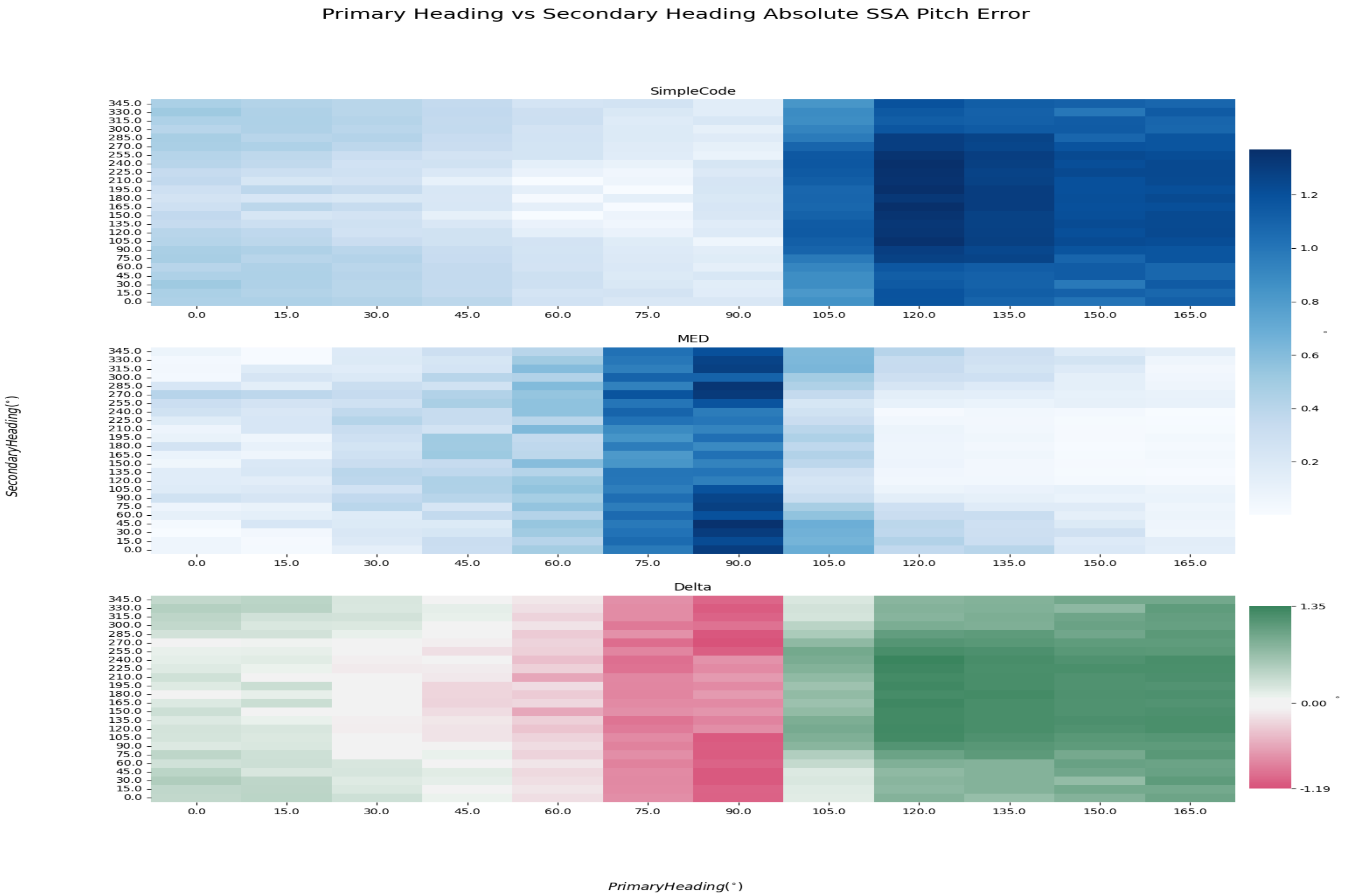

Figure A-4: Primary wave heading vs secondary wave heading absolute SSA pitch error.

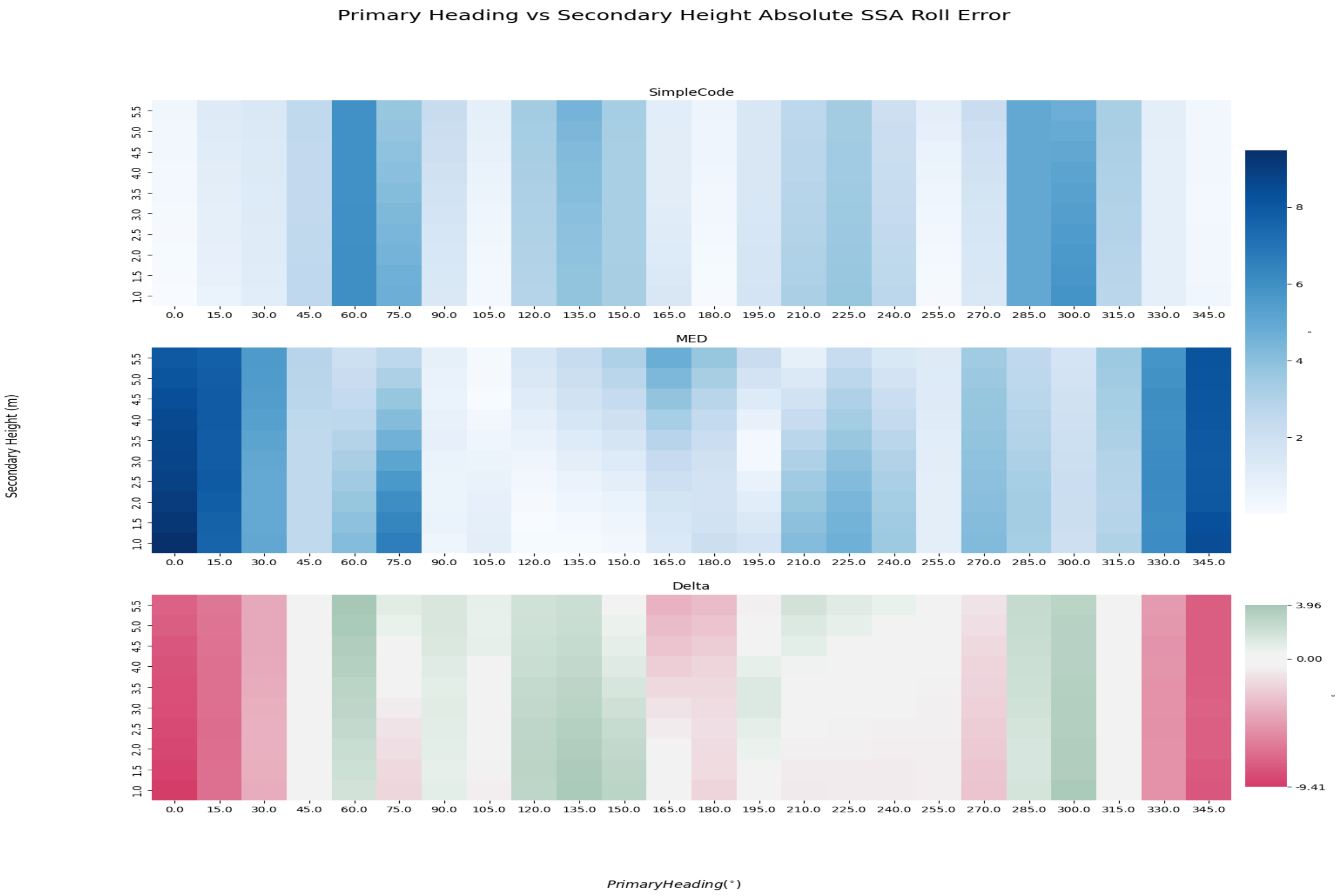

Figure A-5: Primary wave heading vs secondary wave height absolute SSA roll error.

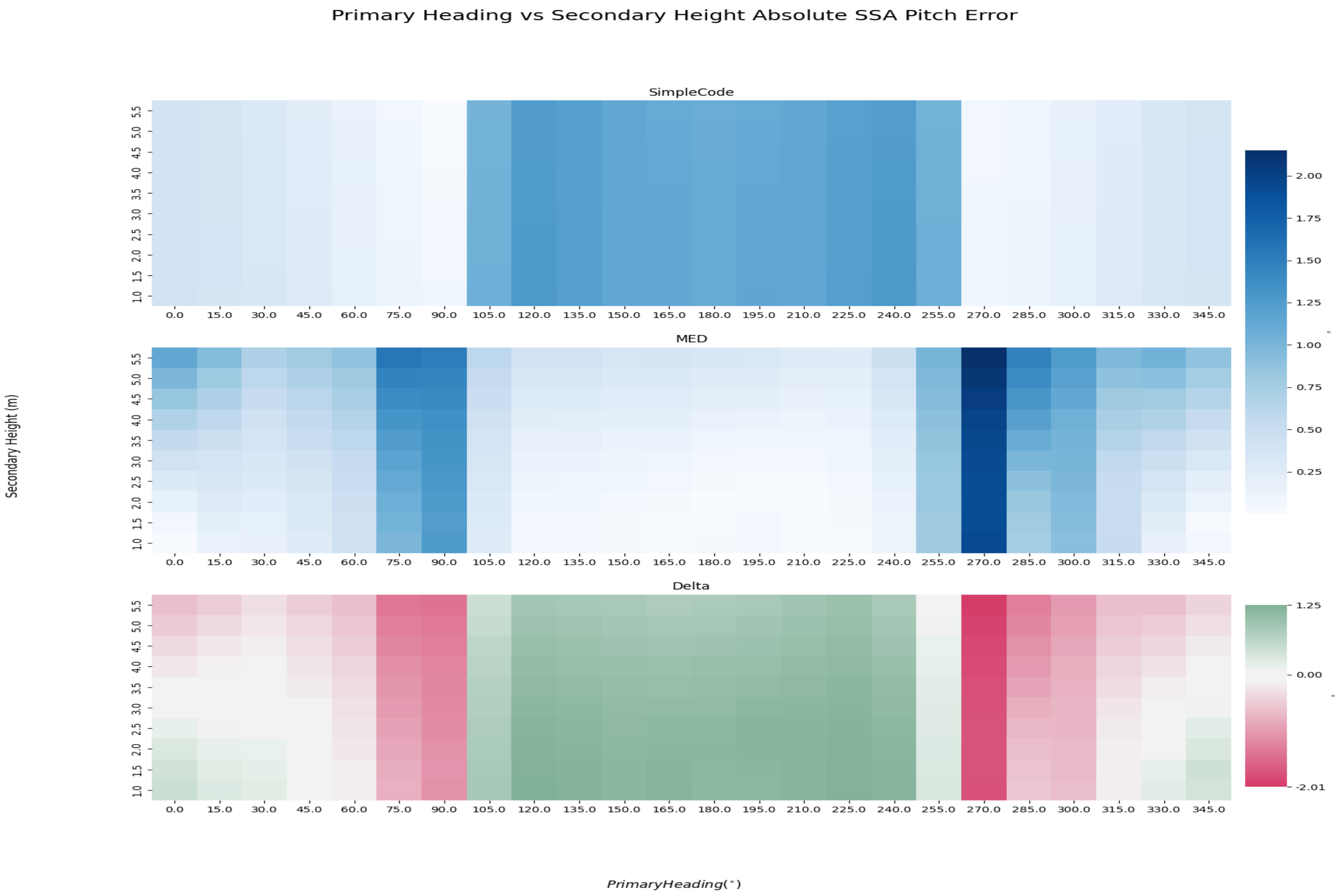

Figure A-6: Primary wave heading vs secondary wave height absolute SSA pitch error.

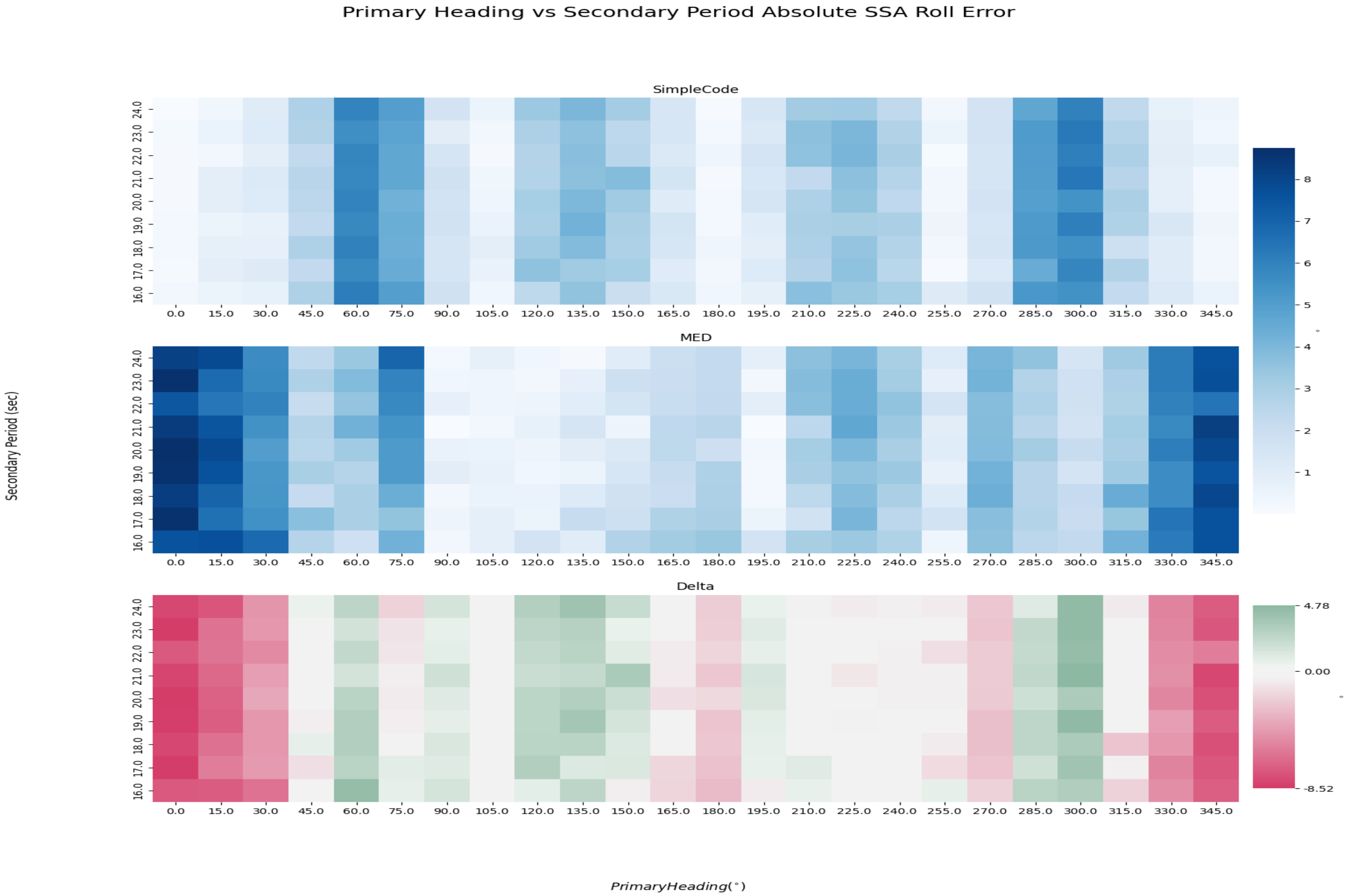

Figure A-7: Primary wave heading vs secondary wave period absolute SSA roll error.

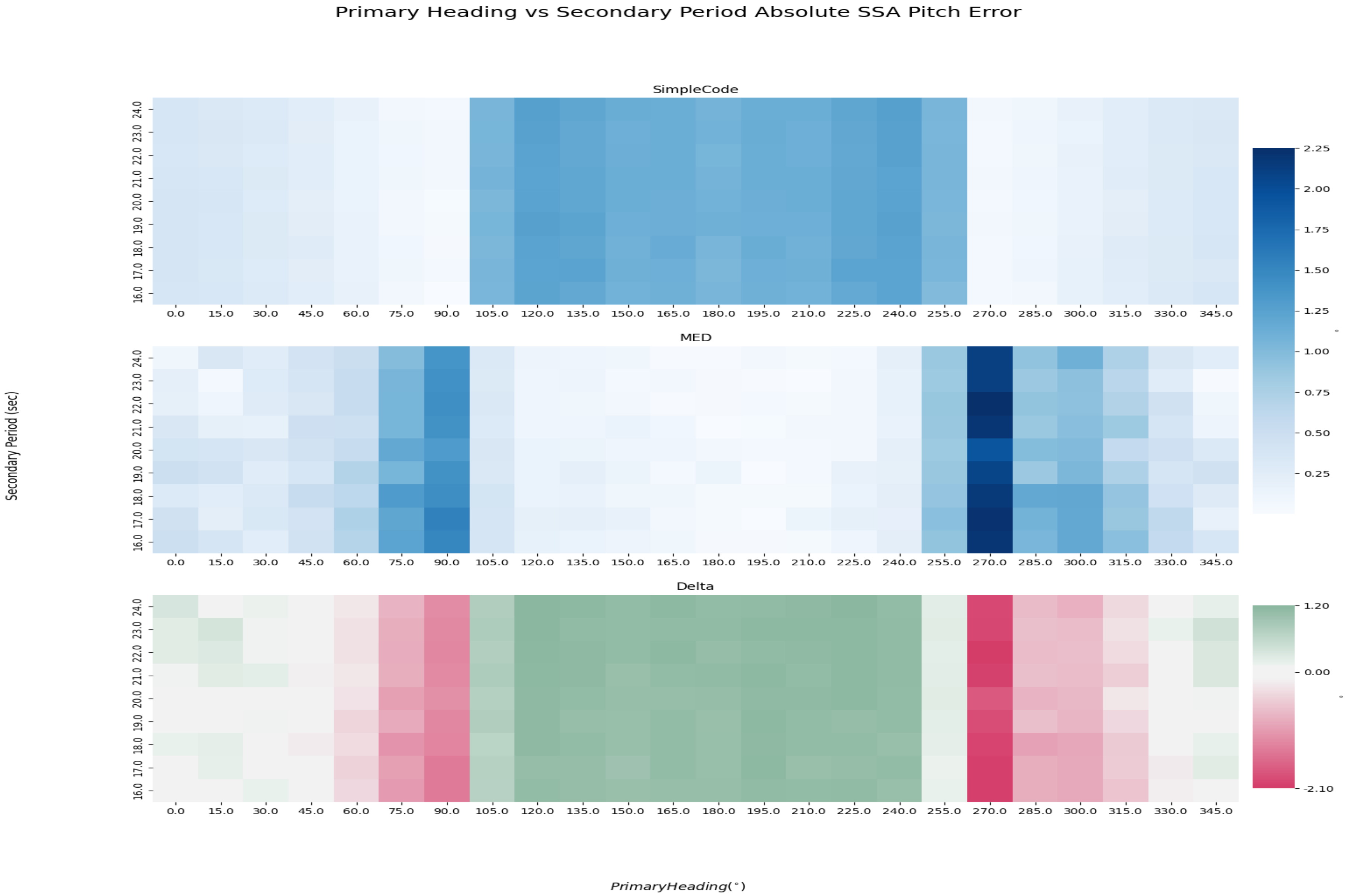

Figure A-8: Primary wave heading vs secondary wave period absolute SSA pitch error.

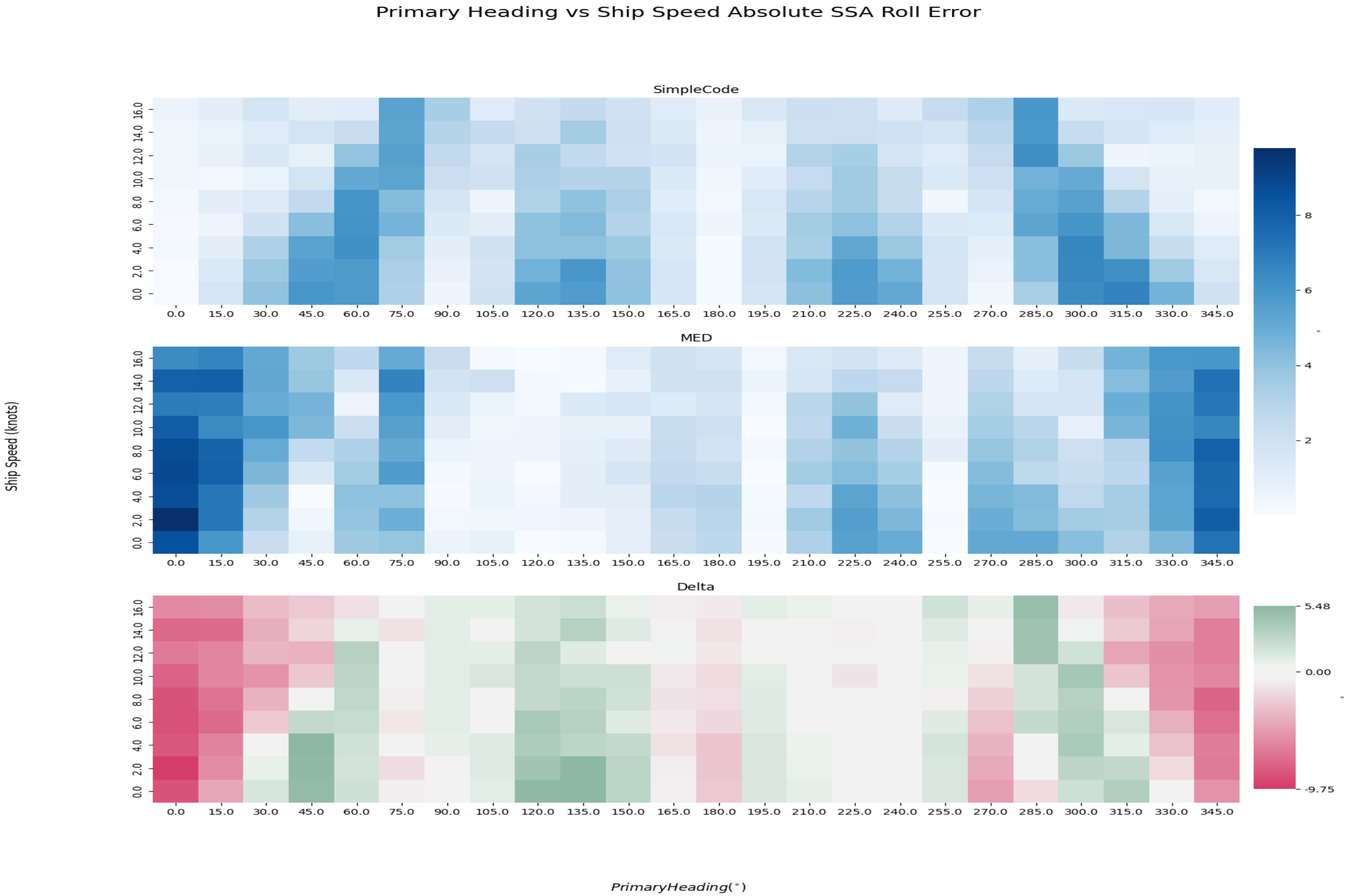

Figure A-9: Primary wave heading vs ship speed absolute SSA roll error.

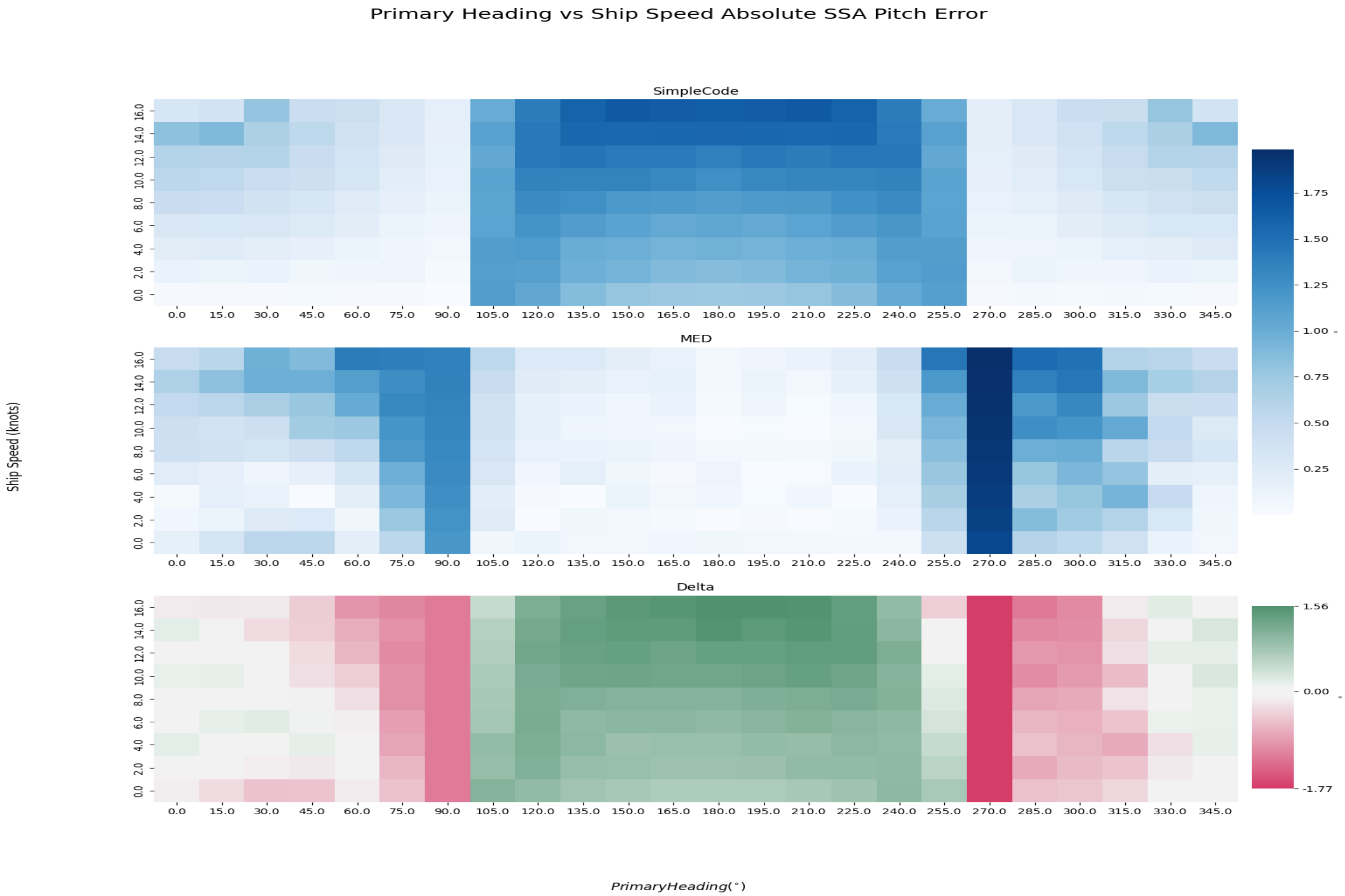

Figure A-10: Primary wave heading vs ship speed absolute SSA pitch error.

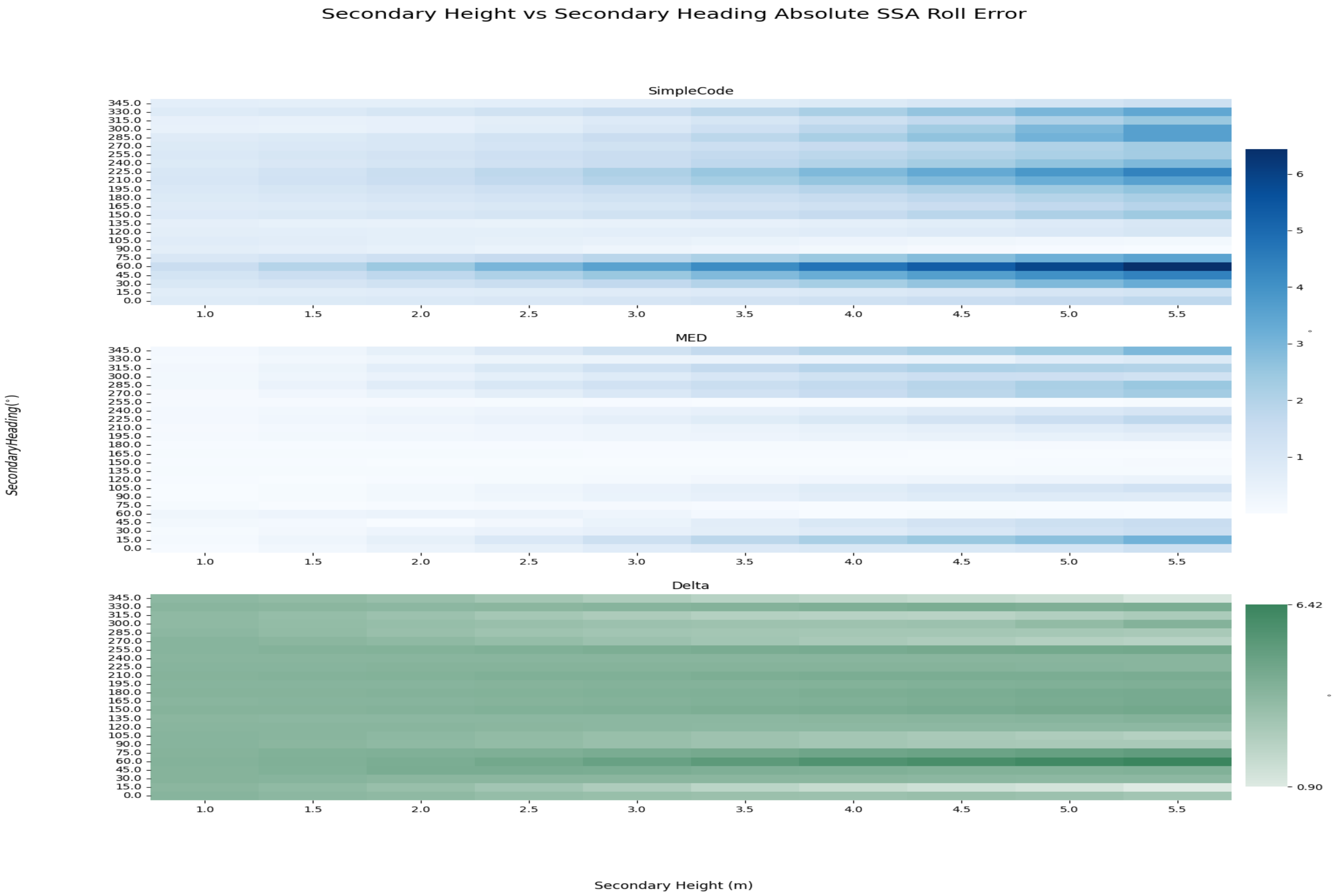

Figure A-11: Secondary wave height vs secondary wave heading absolute SSA roll error.

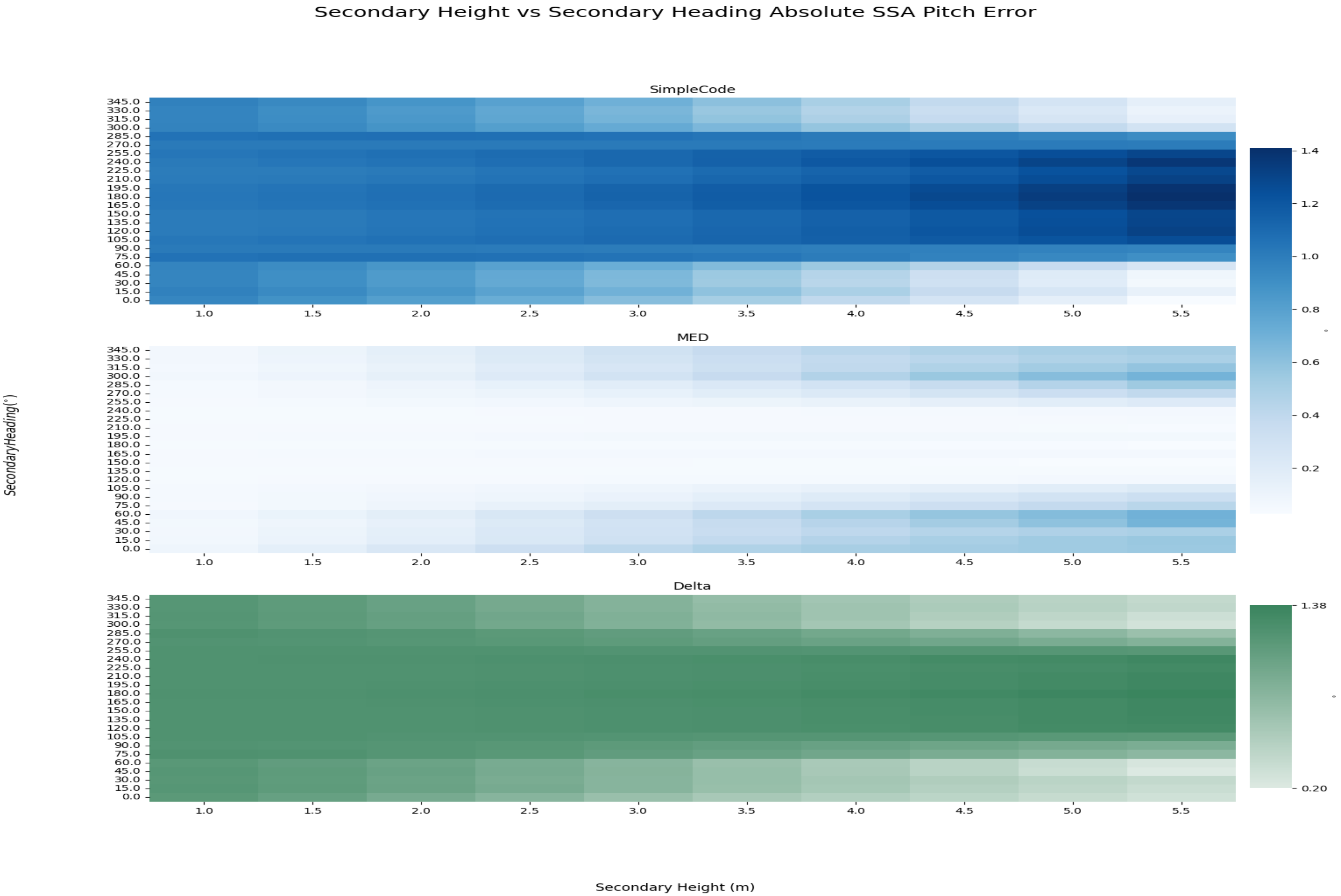

Figure A-12: Secondary wave height vs secondary wave heading absolute SSA pitch error.

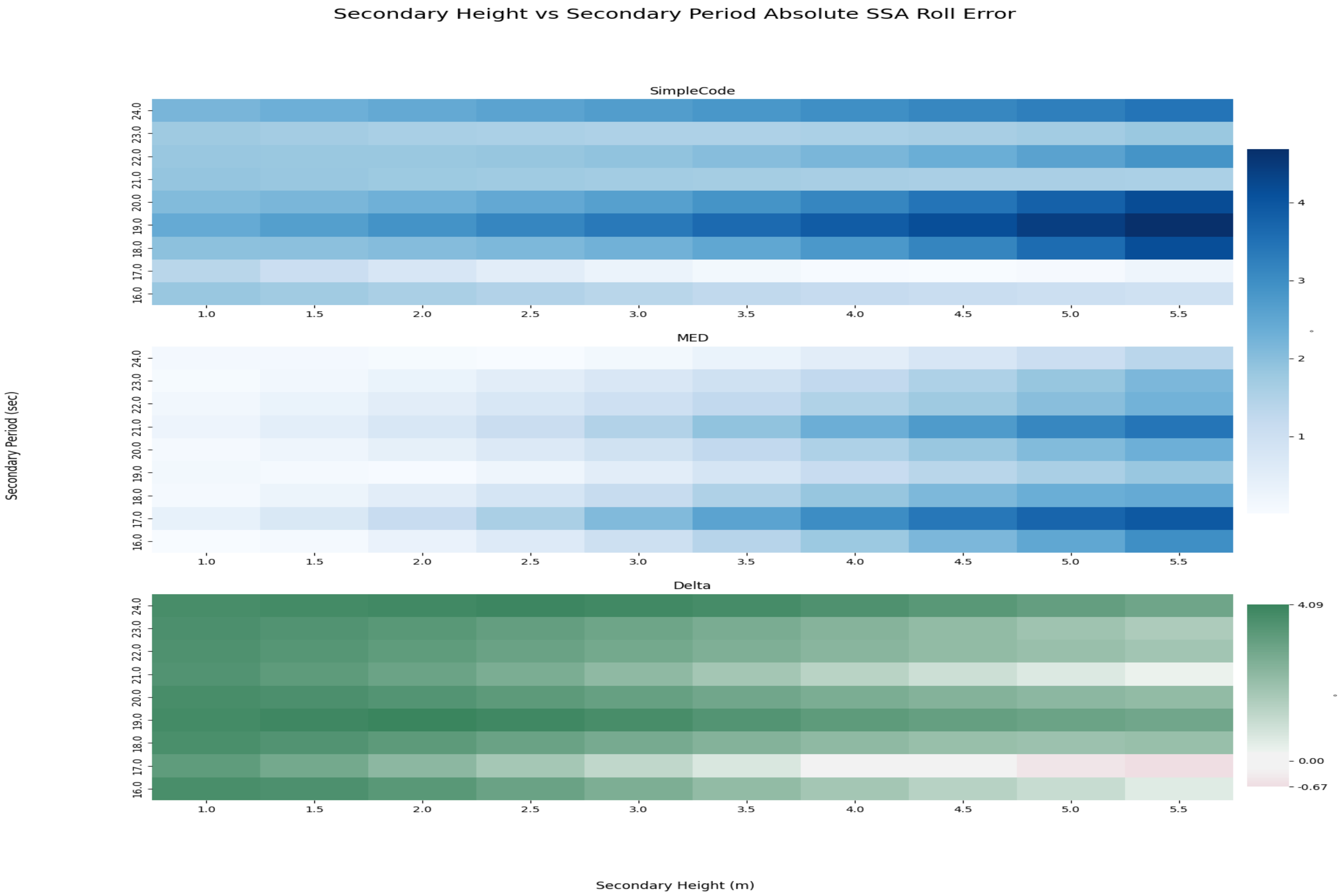

Figure A-13: Secondary wave height vs secondary wave period absolute SSA roll error.

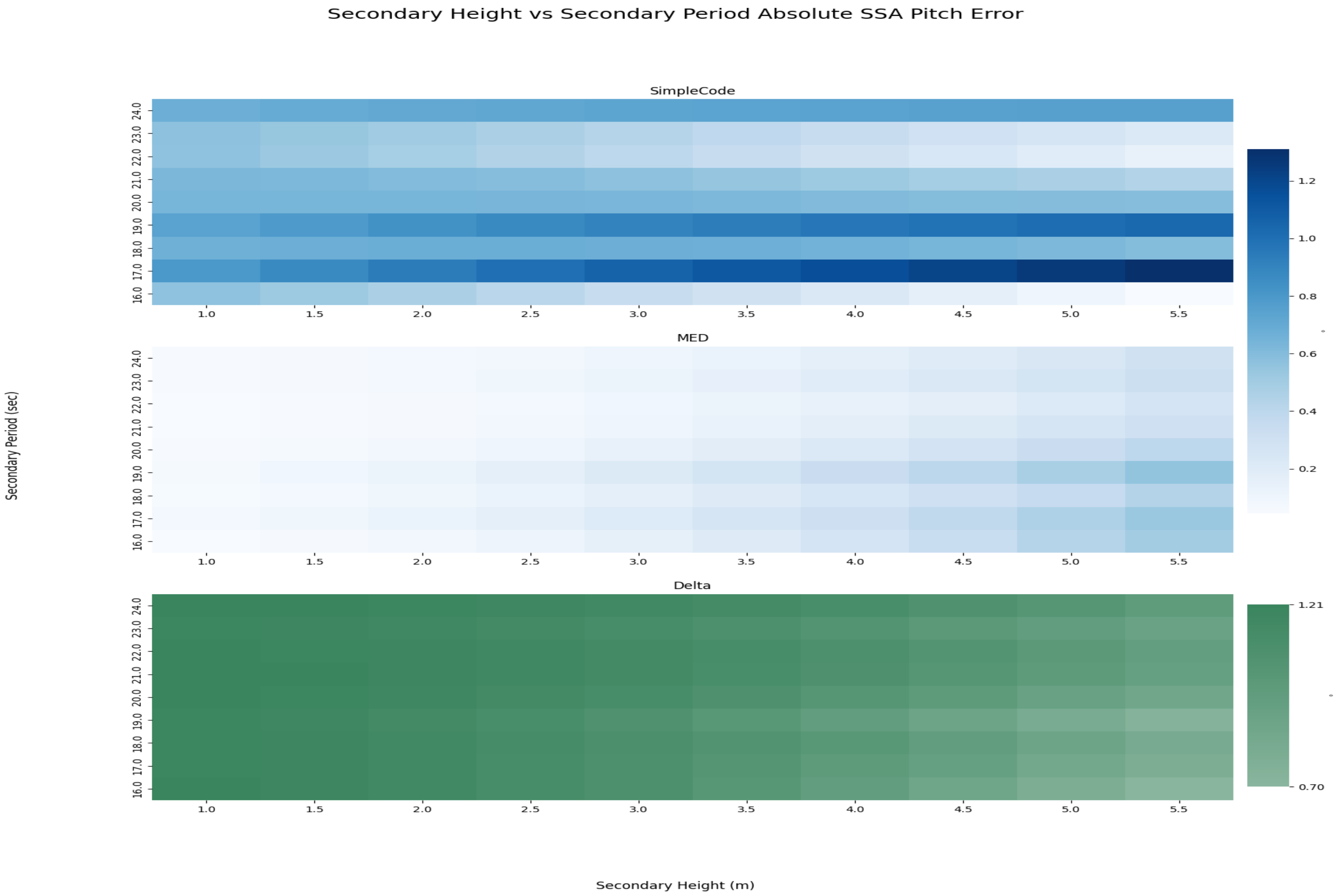

Figure A-14: Secondary wave height vs secondary wave period absolute SSA pitch error.

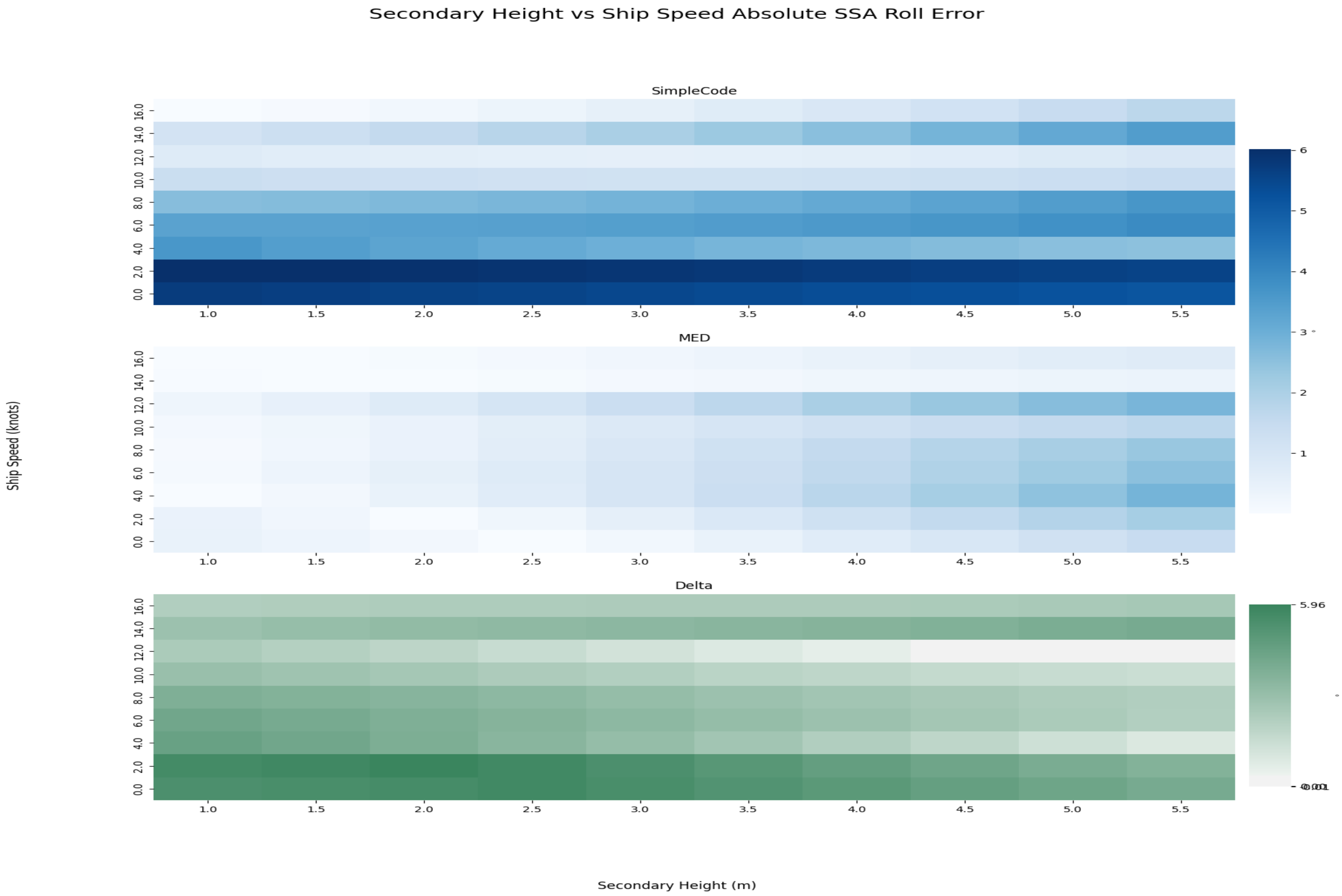

Figure A-15: Secondary wave height vs ship speed absolute SSA roll error.

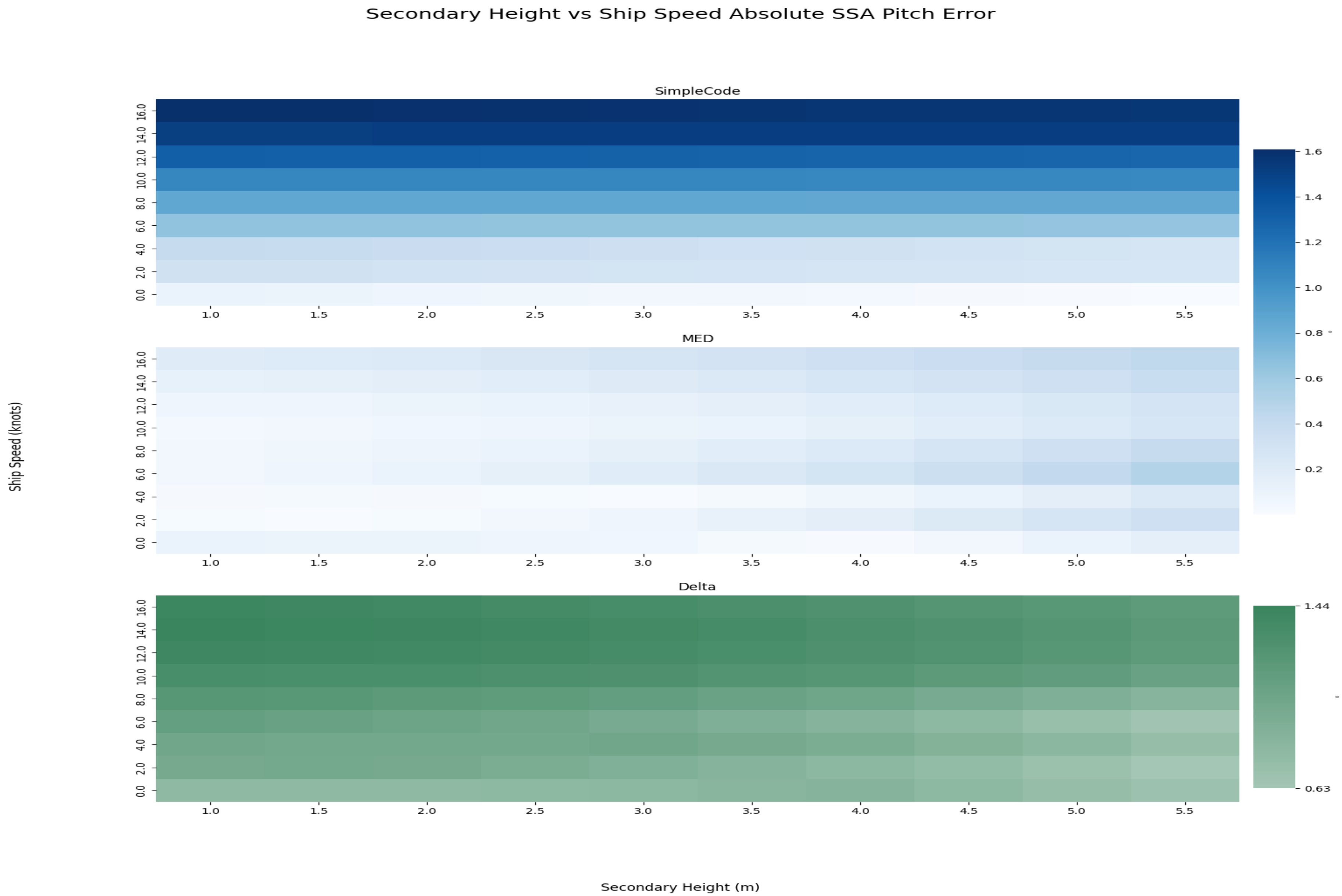

Figure A-16: Secondary wave height vs ship speed absolute SSA pitch error.

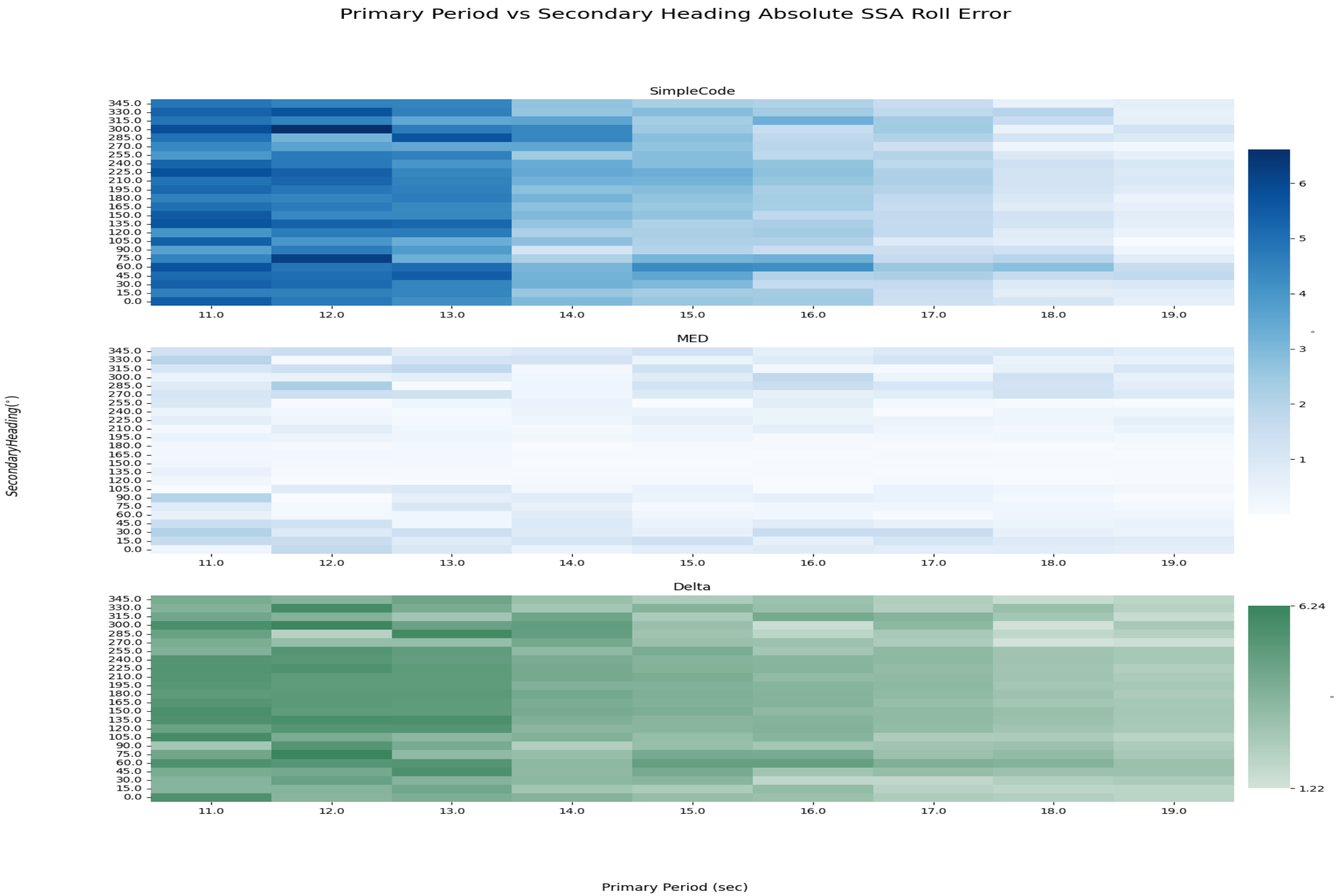

Figure A-17: Primary wave period vs secondary wave heading absolute SSA roll error.

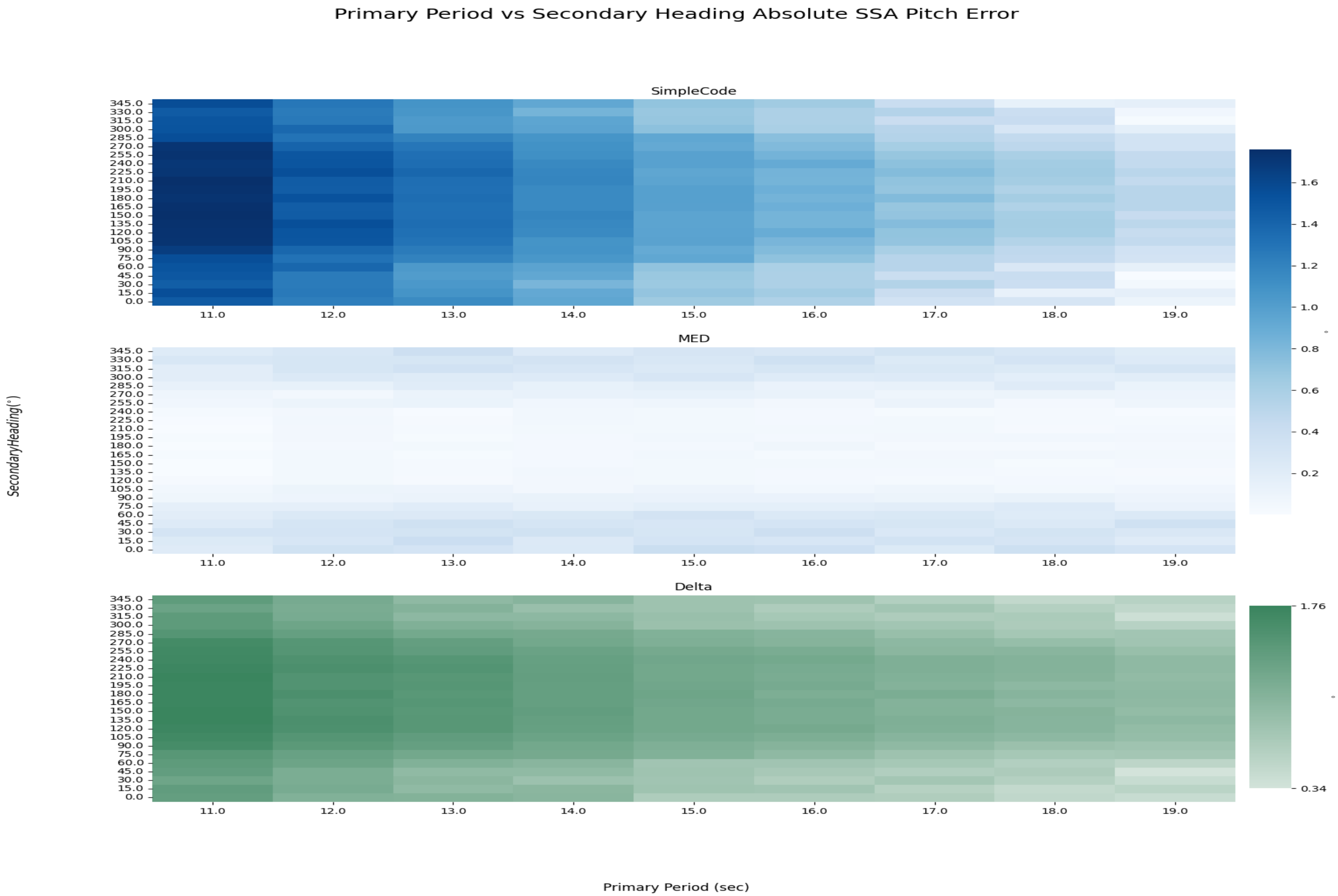

Figure A-18: Primary wave period vs secondary wave heading absolute SSA pitch error.

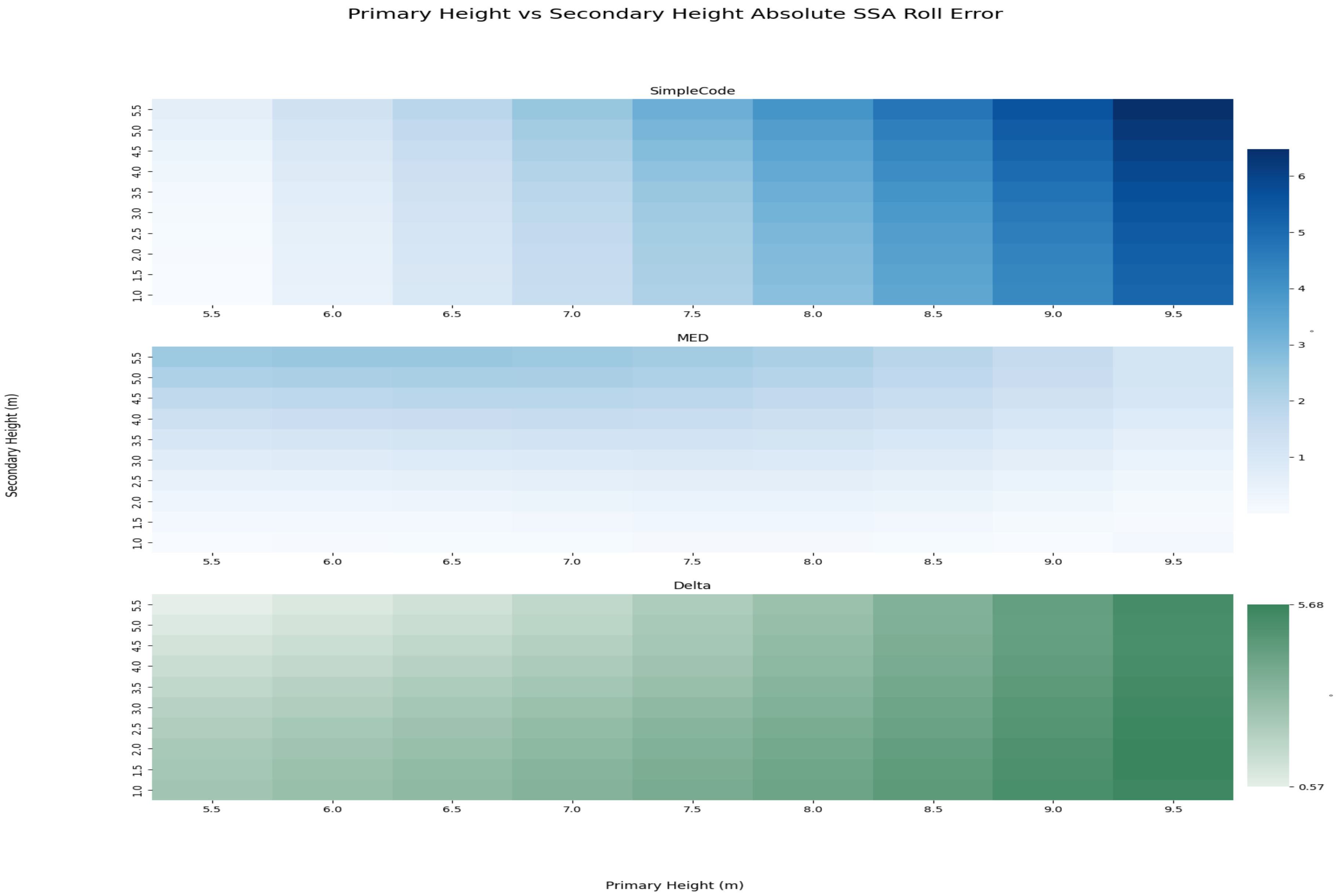

Figure A-19: Primary wave height vs secondary wave height absolute SSA roll error.

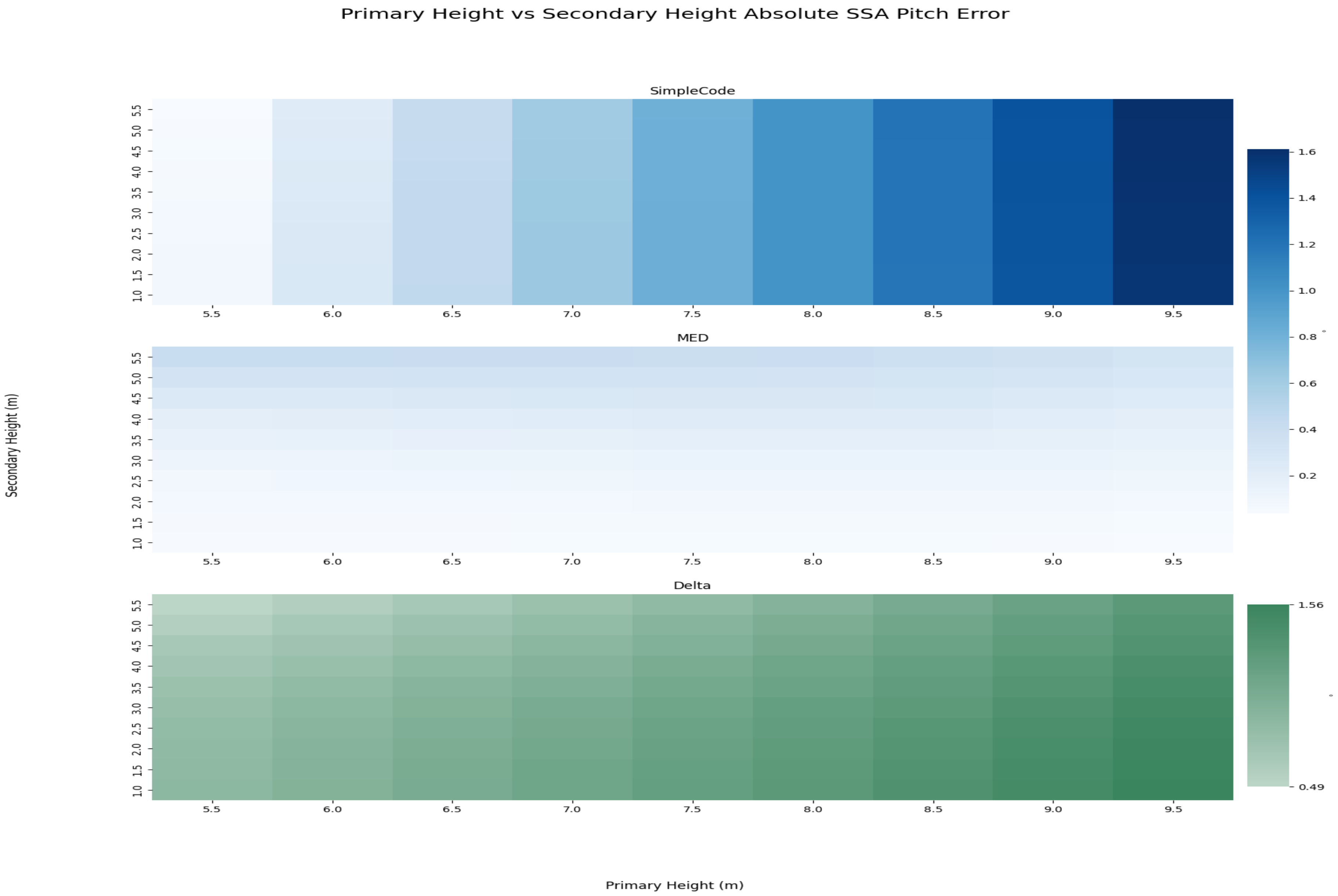

Figure A-20: Primary wave height vs secondary wave height absolute SSA pitch error.

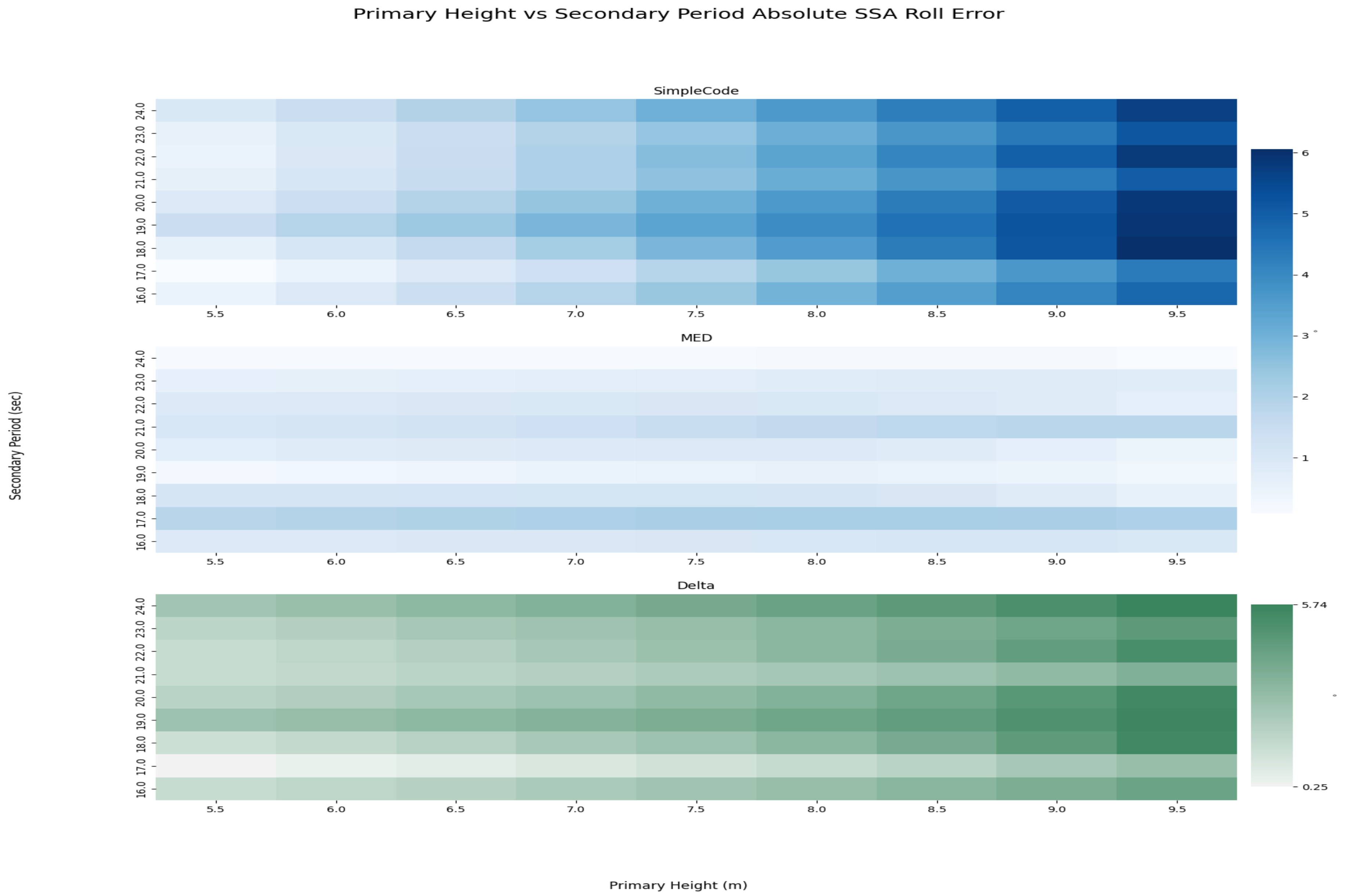

Figure A-21: Primary wave height vs secondary wave height absolute SSA roll error.

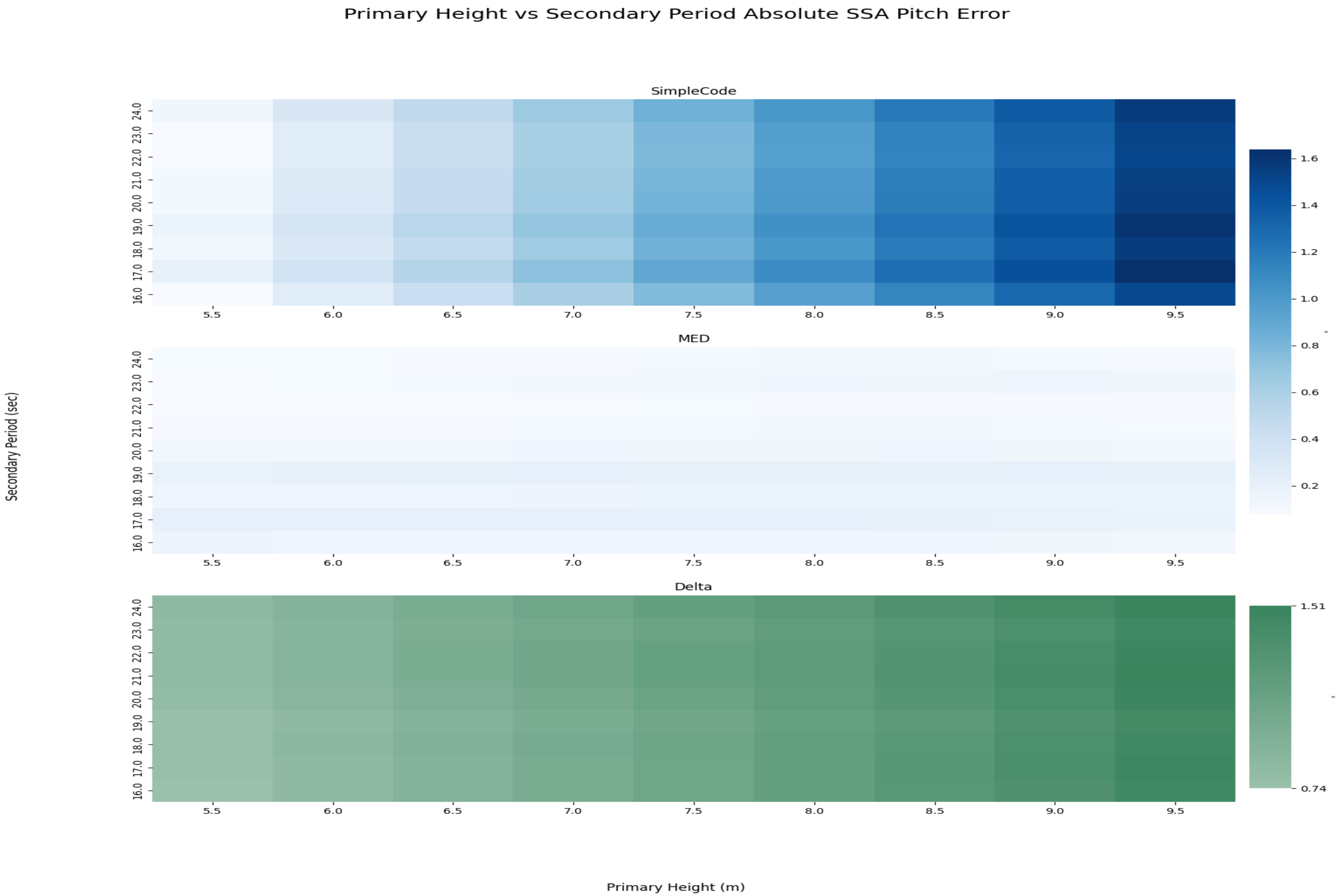

Figure A-22: Primary wave height vs secondary wave period absolute SSA pitch error.

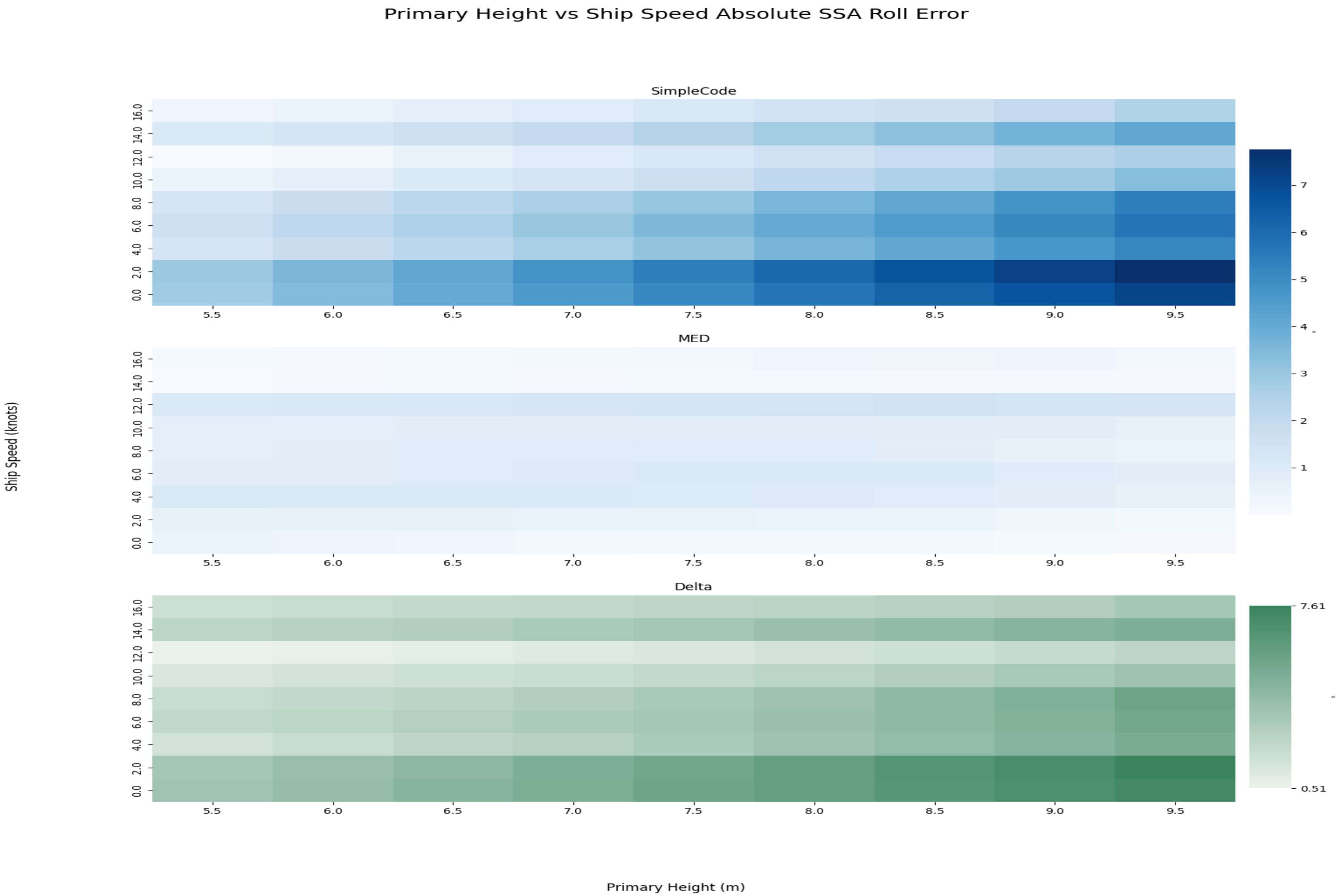

Figure A-23: Primary wave height vs ship speed absolute SSA roll error.

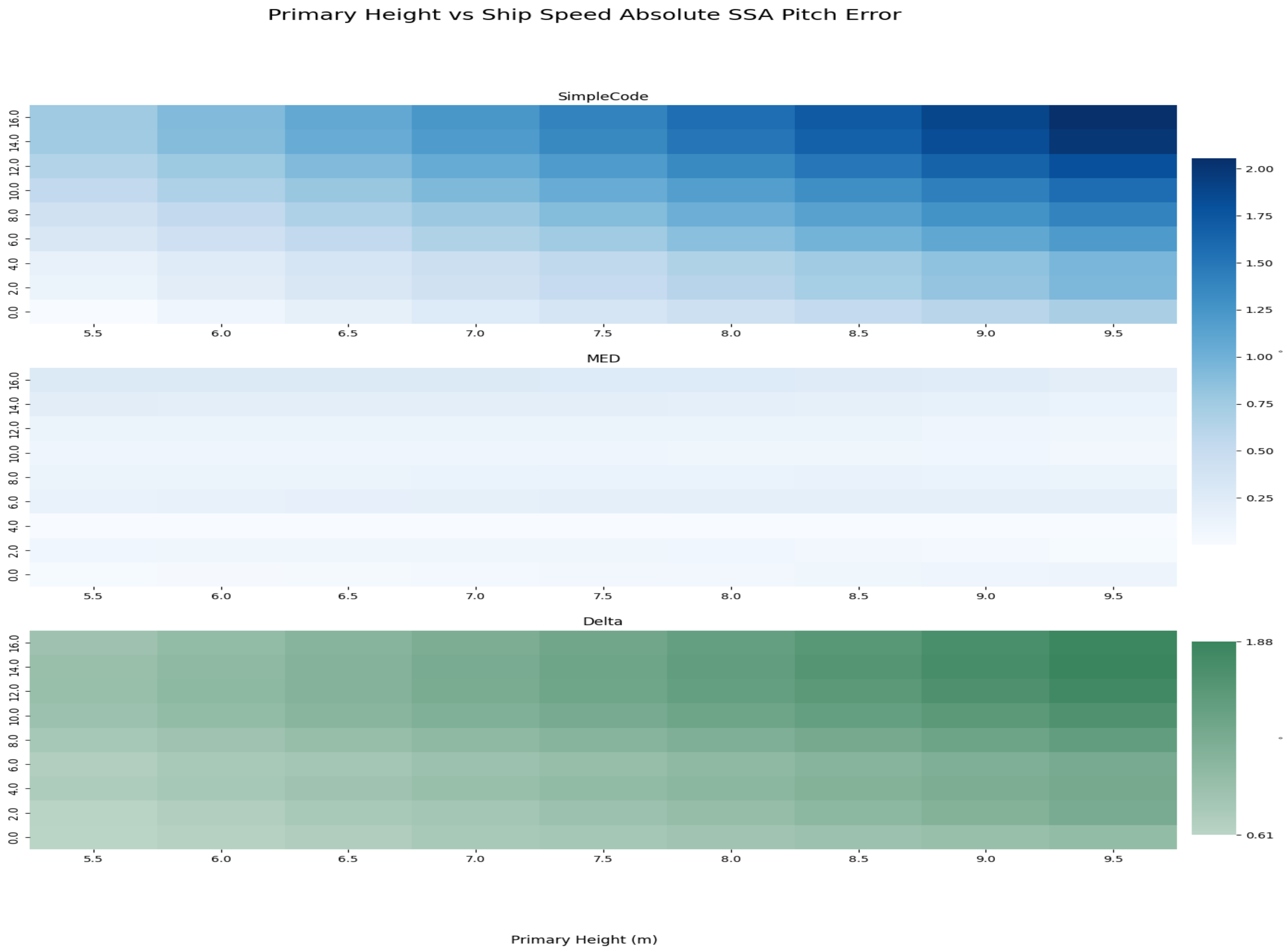

Figure A-24: Primary wave height vs ship speed absolute SSA pitch error.

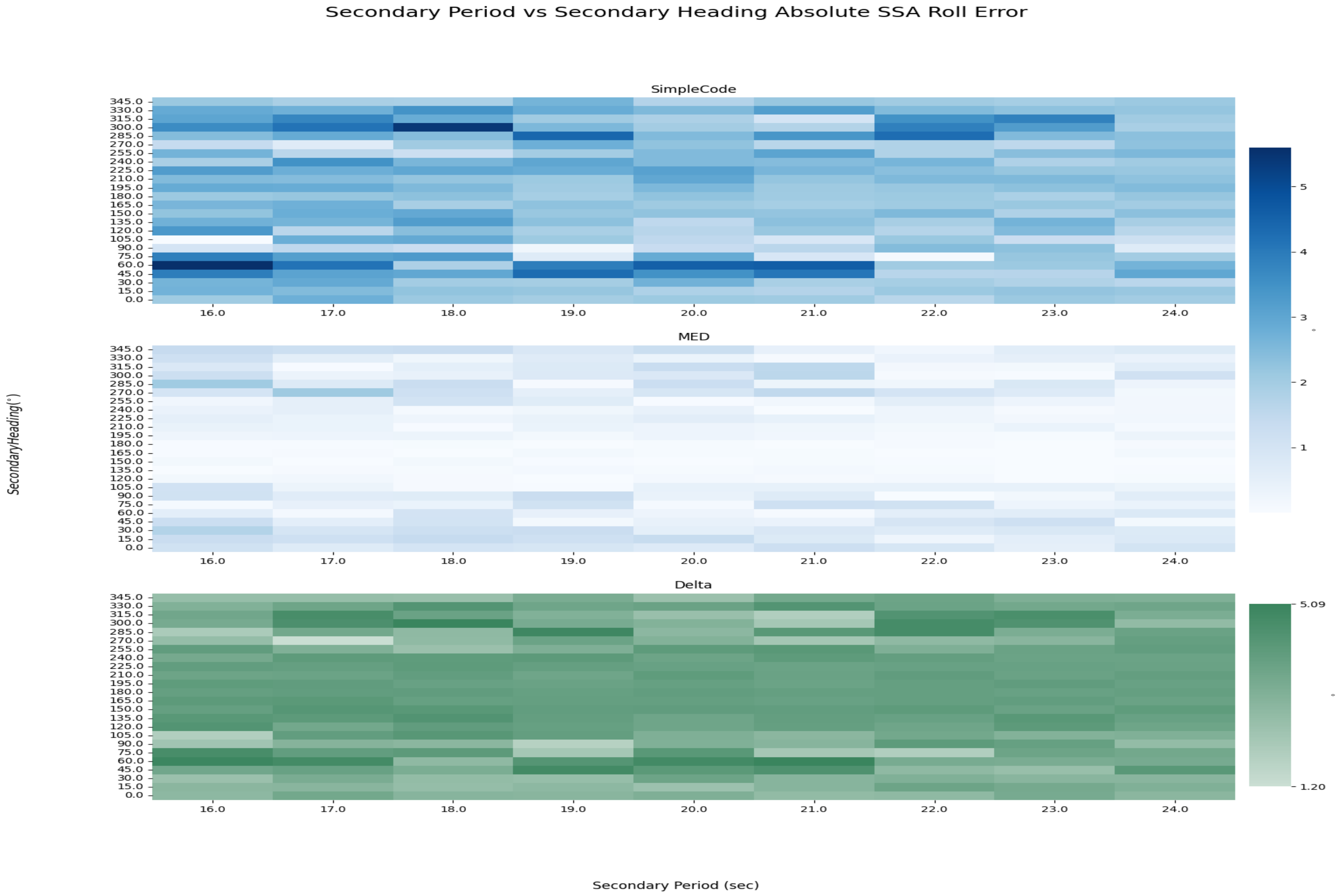

Figure A-25: Secondary wave period wave height vs secondary wave heading absolute SSA roll error.

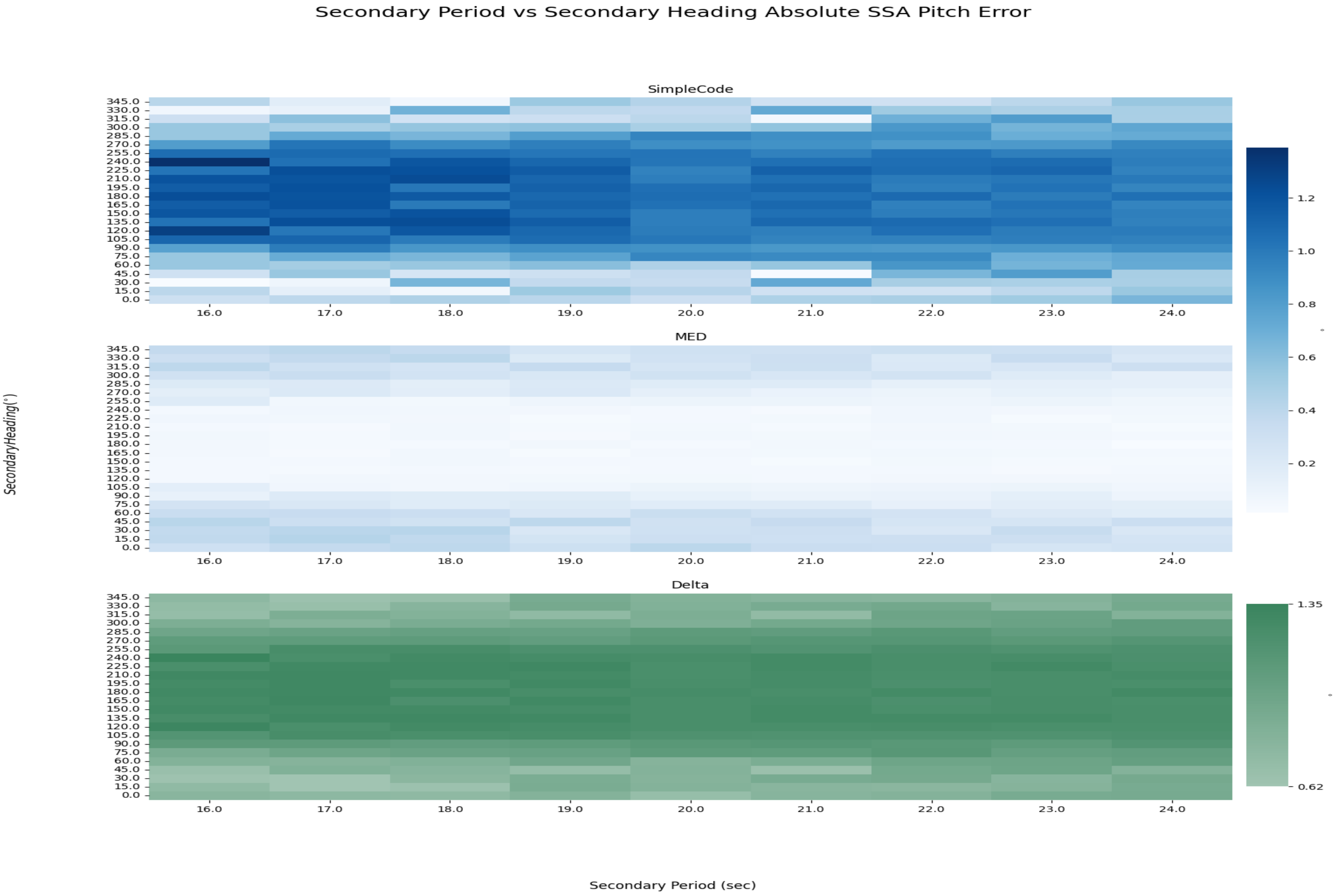

Figure A-26: Secondary wave period wave height vs secondary wave heading absolute SSA pitch error.

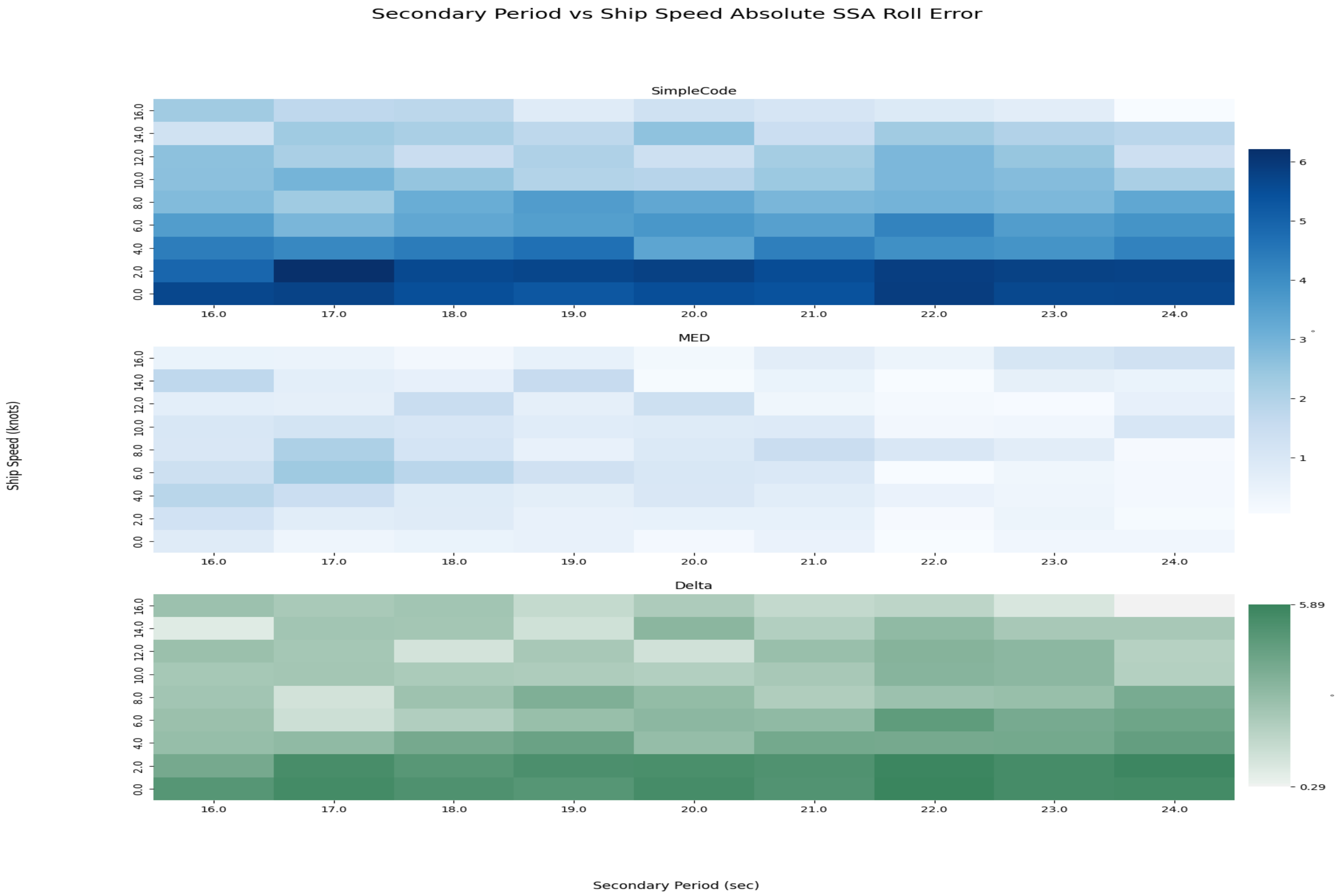

Figure A-27: Secondary wave period wave height vs ship speed absolute SSA roll error.

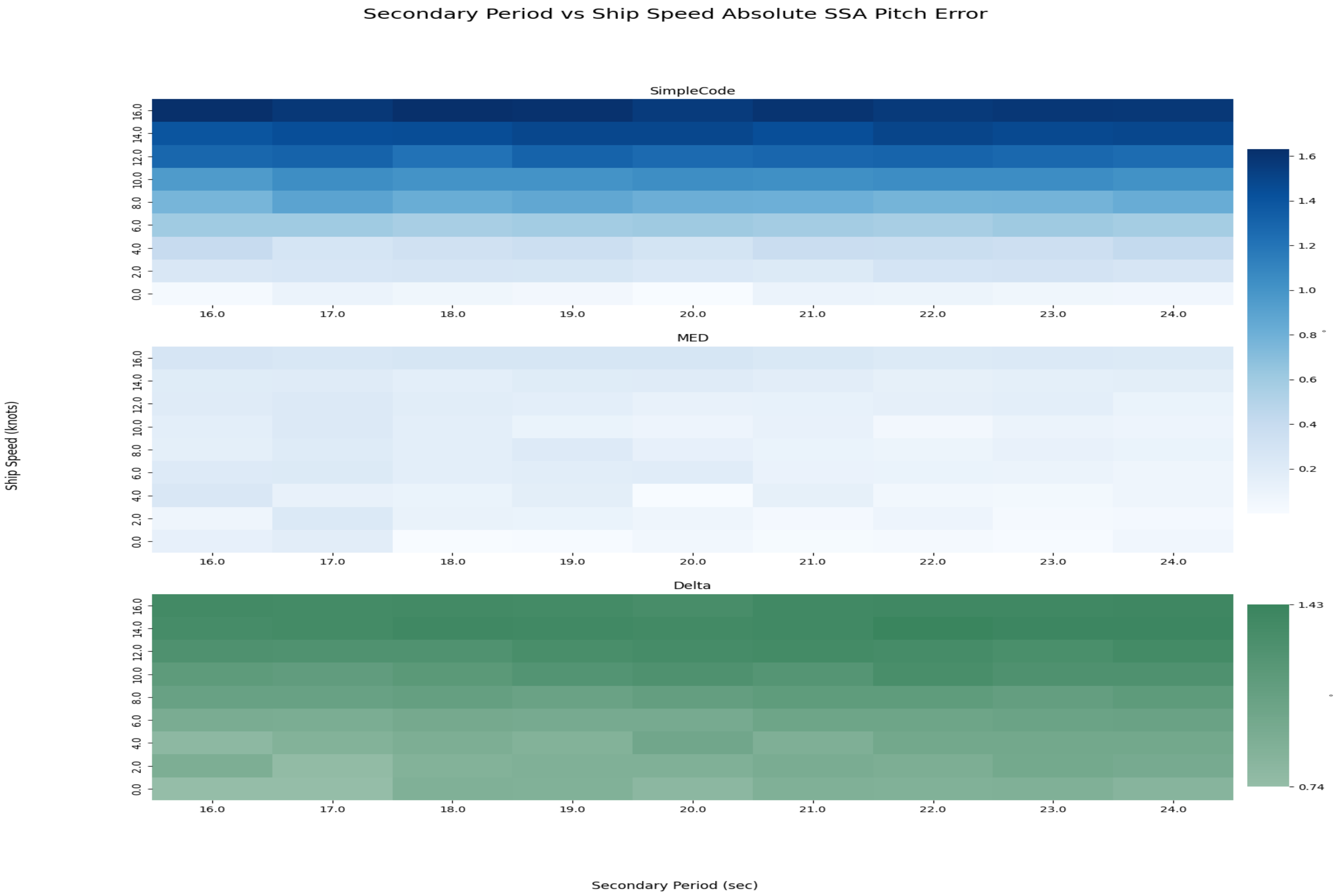

Figure A-28: Secondary wave period wave height vs ship speed absolute SSA pitch error.

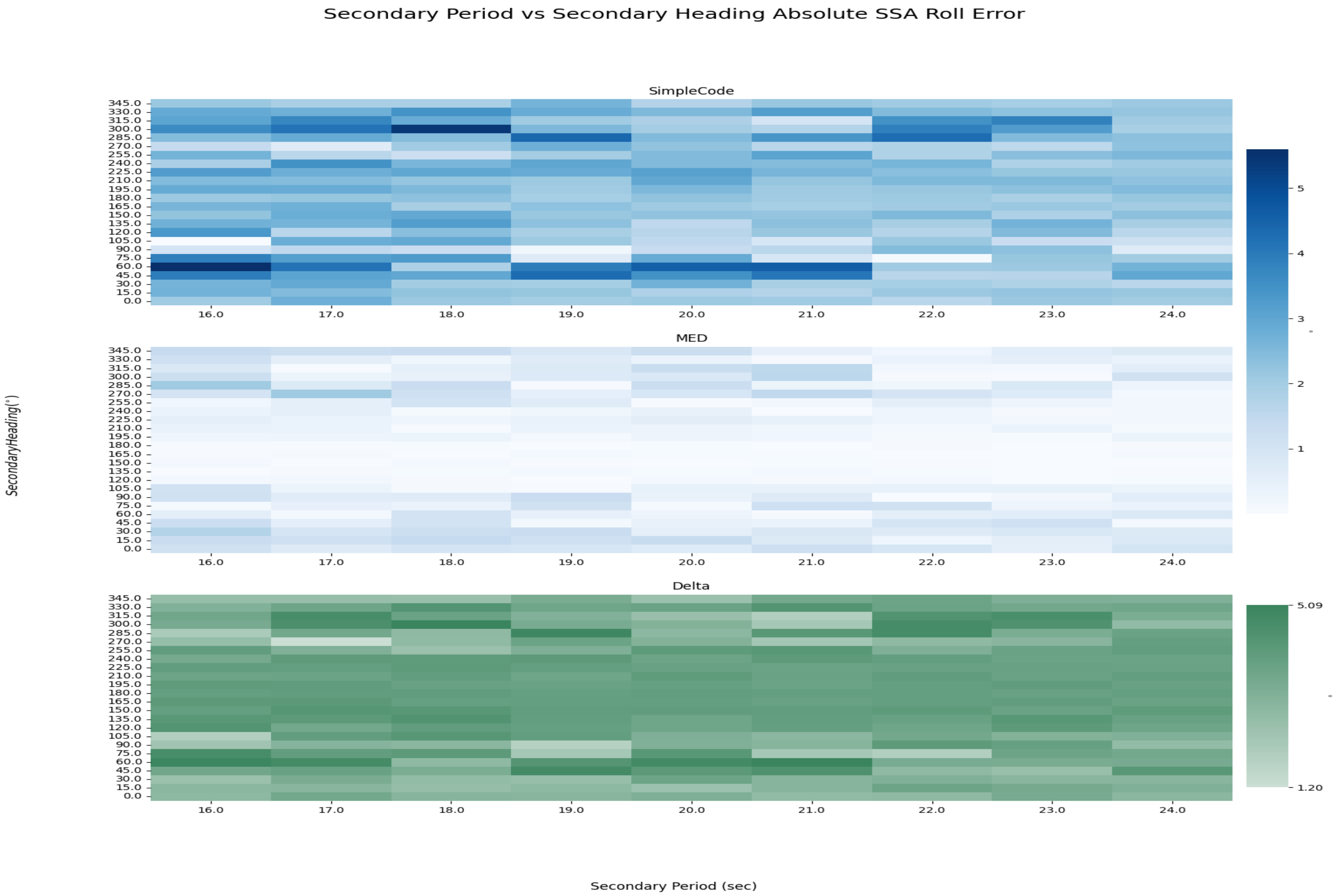

Figure A-29: Secondary wave period wave height vs secondary wave heading absolute SSA roll error.

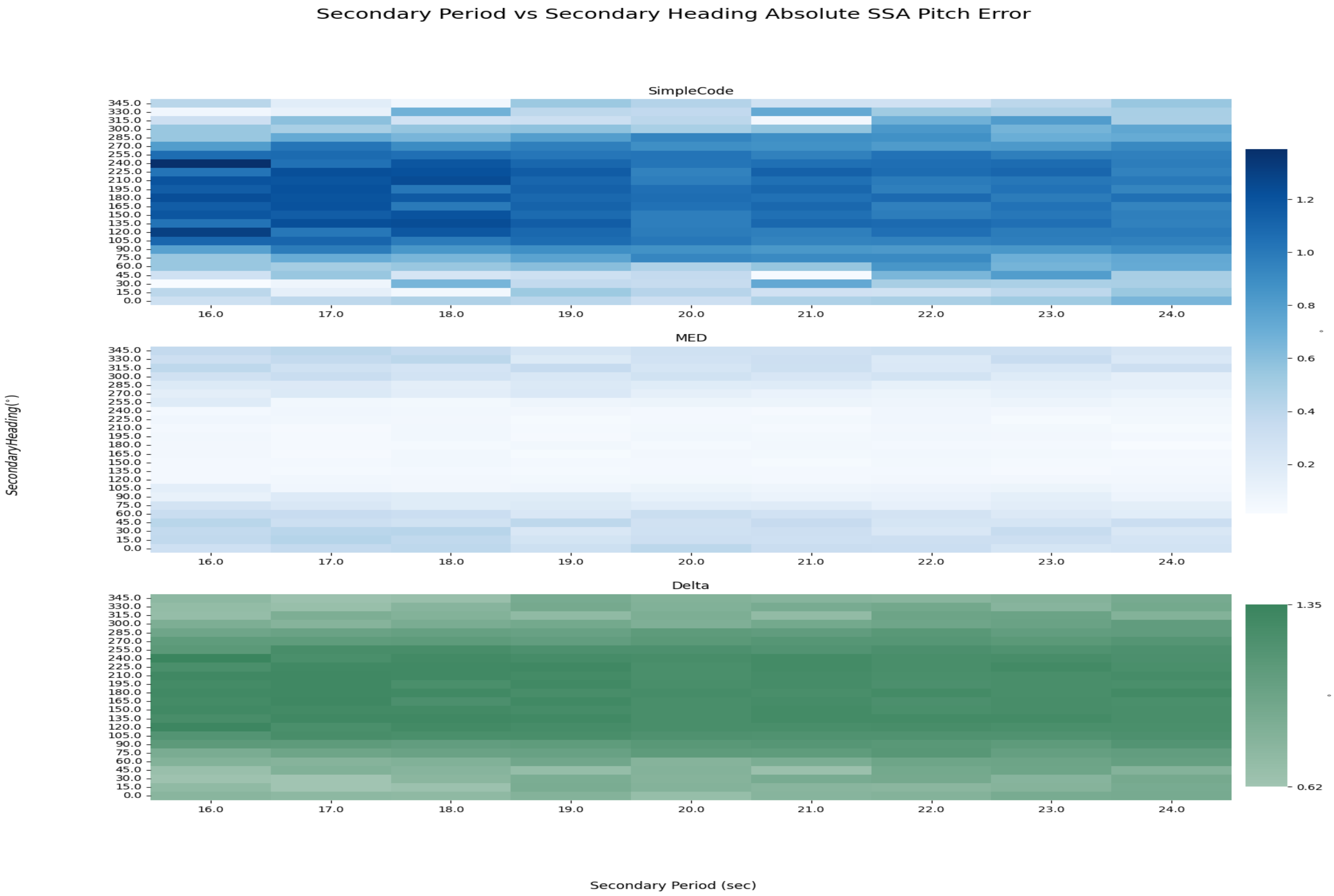

Figure A-30: Secondary wave period wave height vs secondary wave heading absolute SSA pitch error.

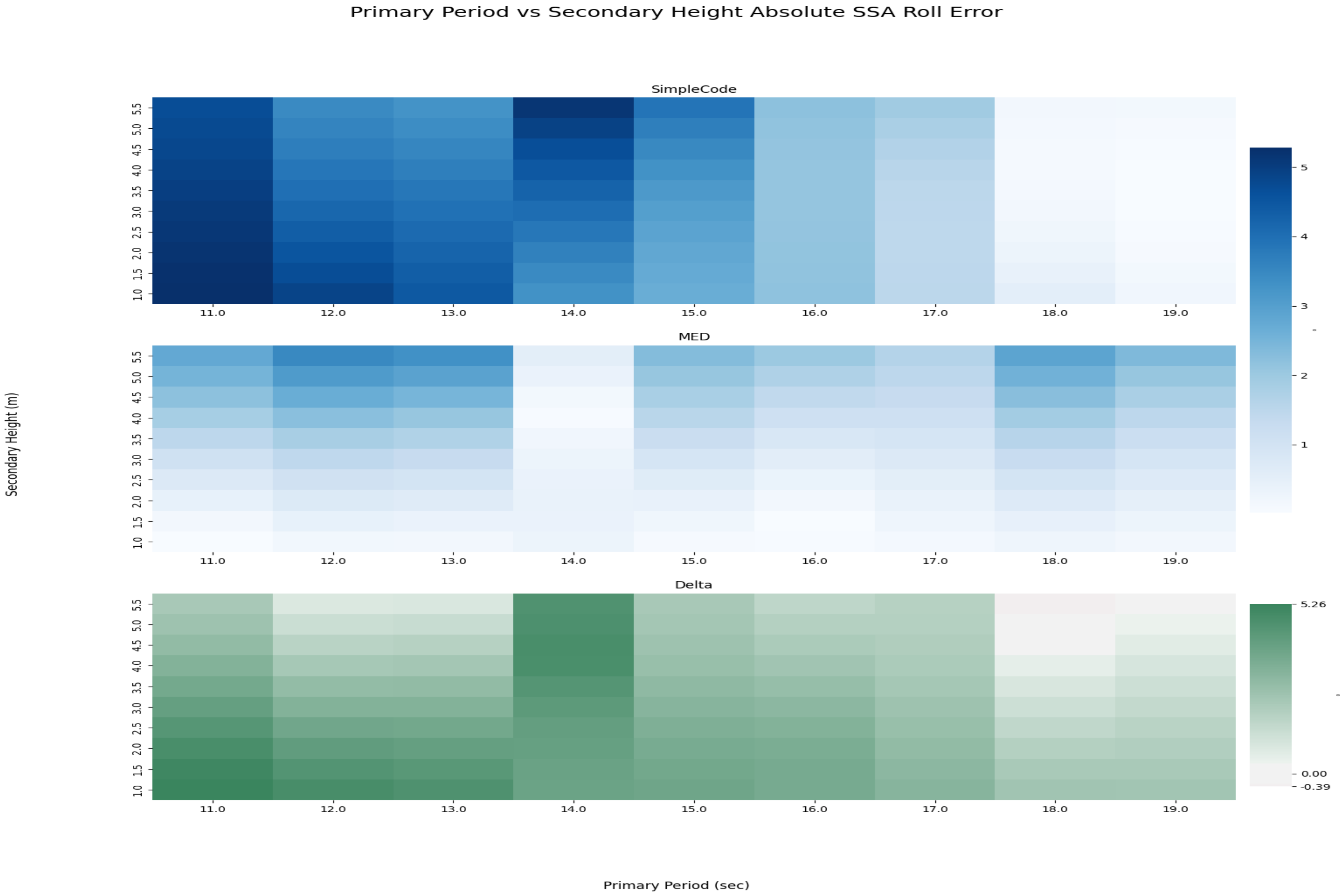

Figure A-31: Primary wave period wave height vs secondary wave height absolute SSA roll error.

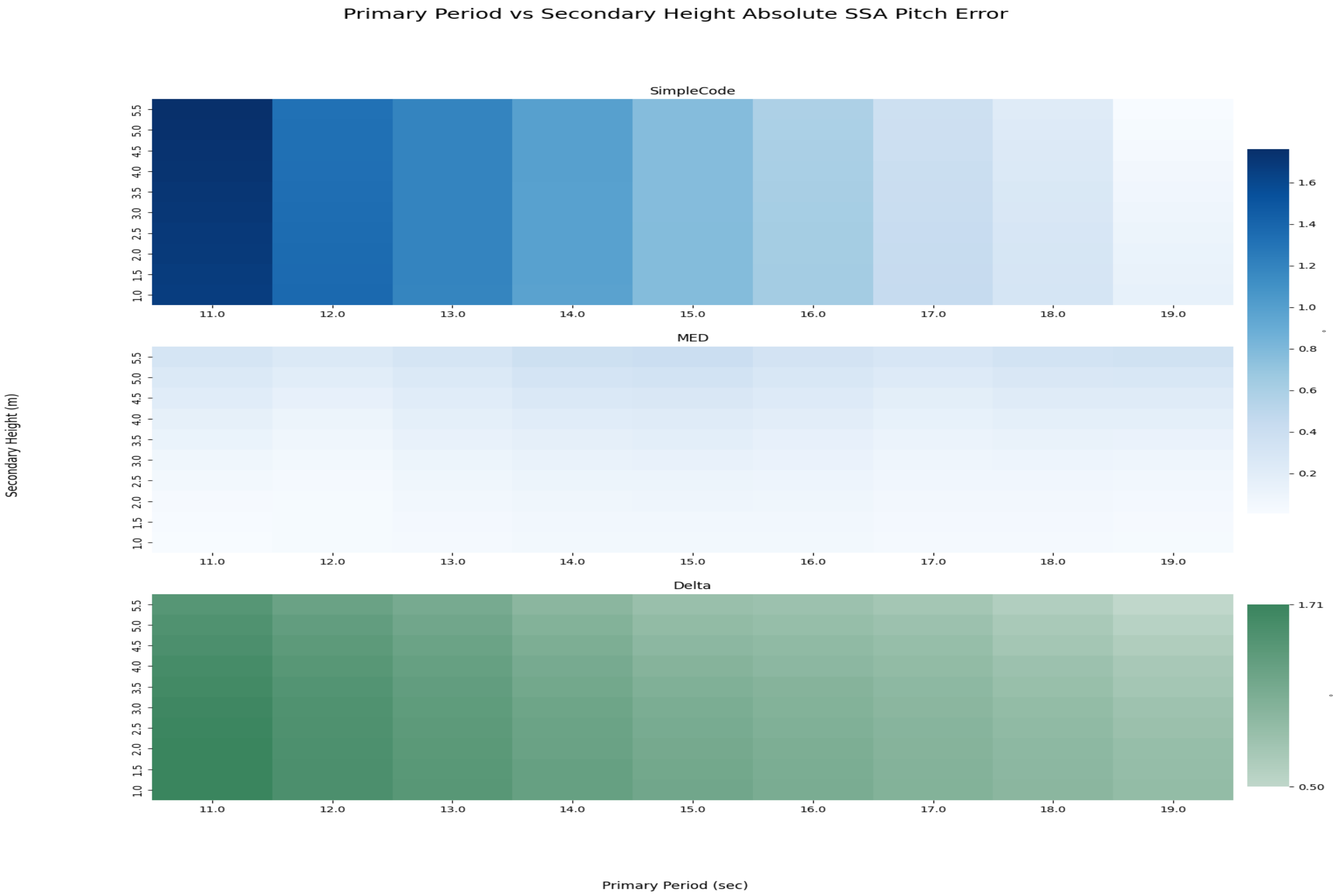

Figure A-32: Primary wave period wave height vs secondary wave height absolute SSA pitch error.

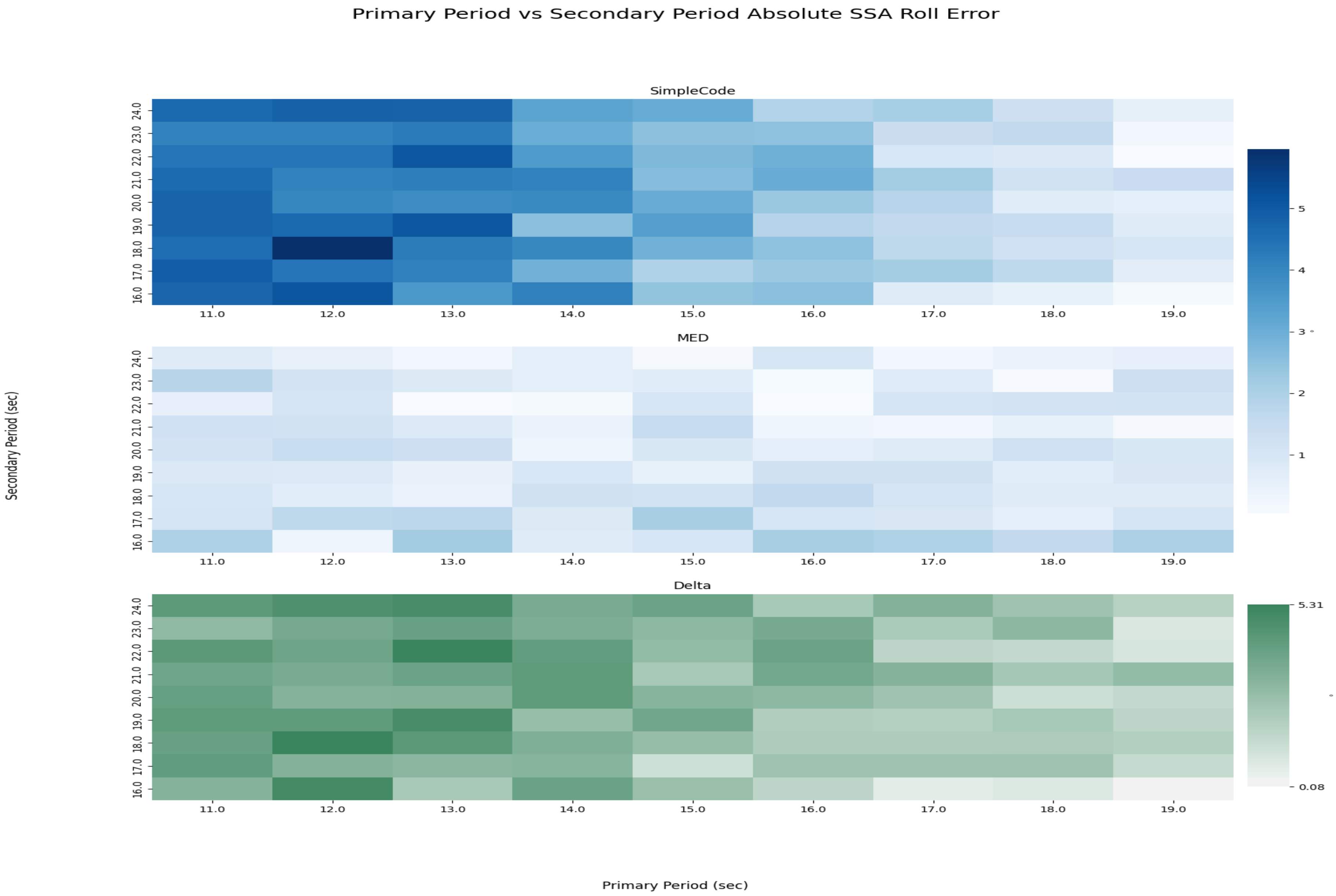

Figure A-33: Primary wave period wave height vs secondary wave period absolute SSA roll error.

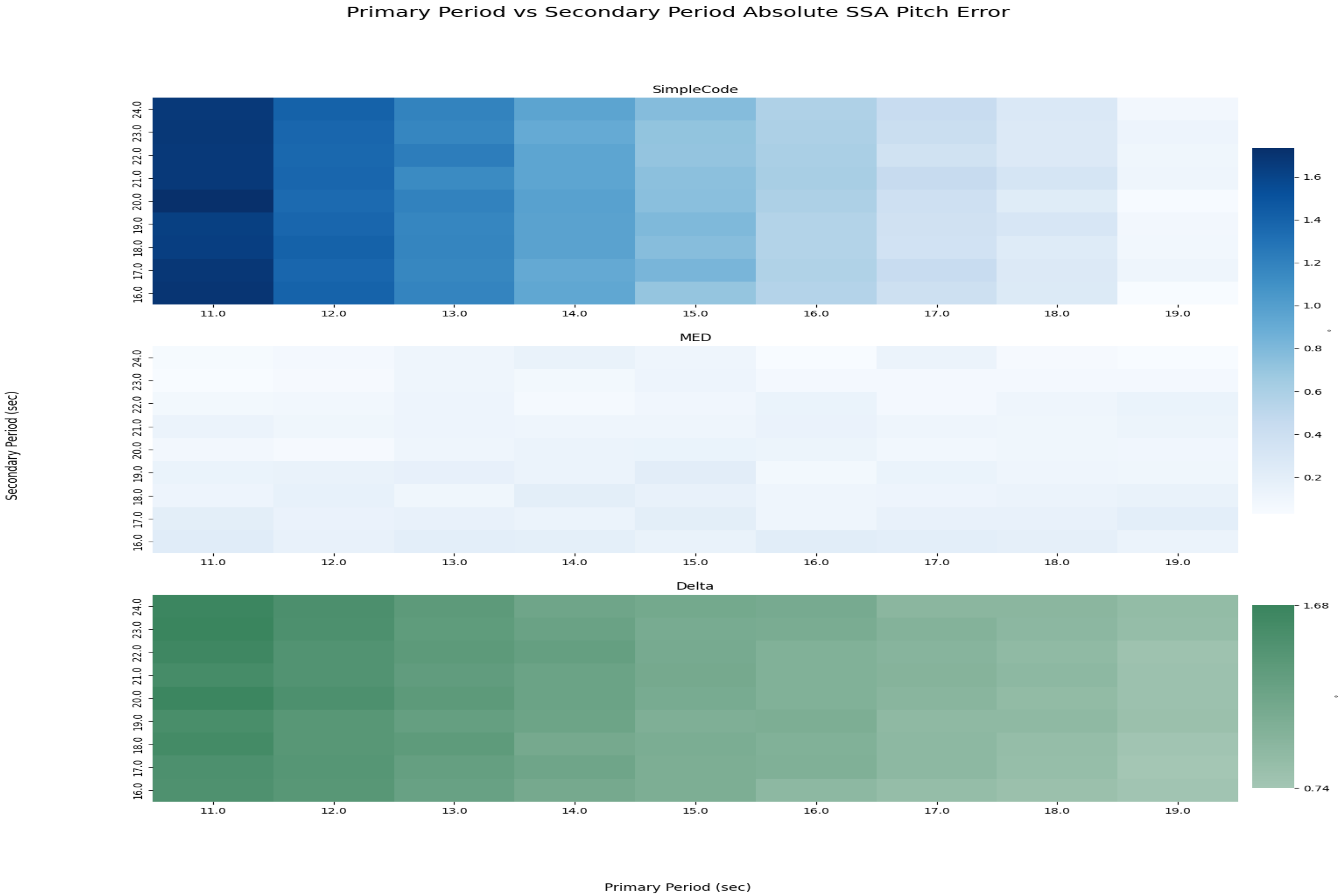

Figure A-34: Primary wave period wave height vs secondary wave period absolute SSA pitch error.

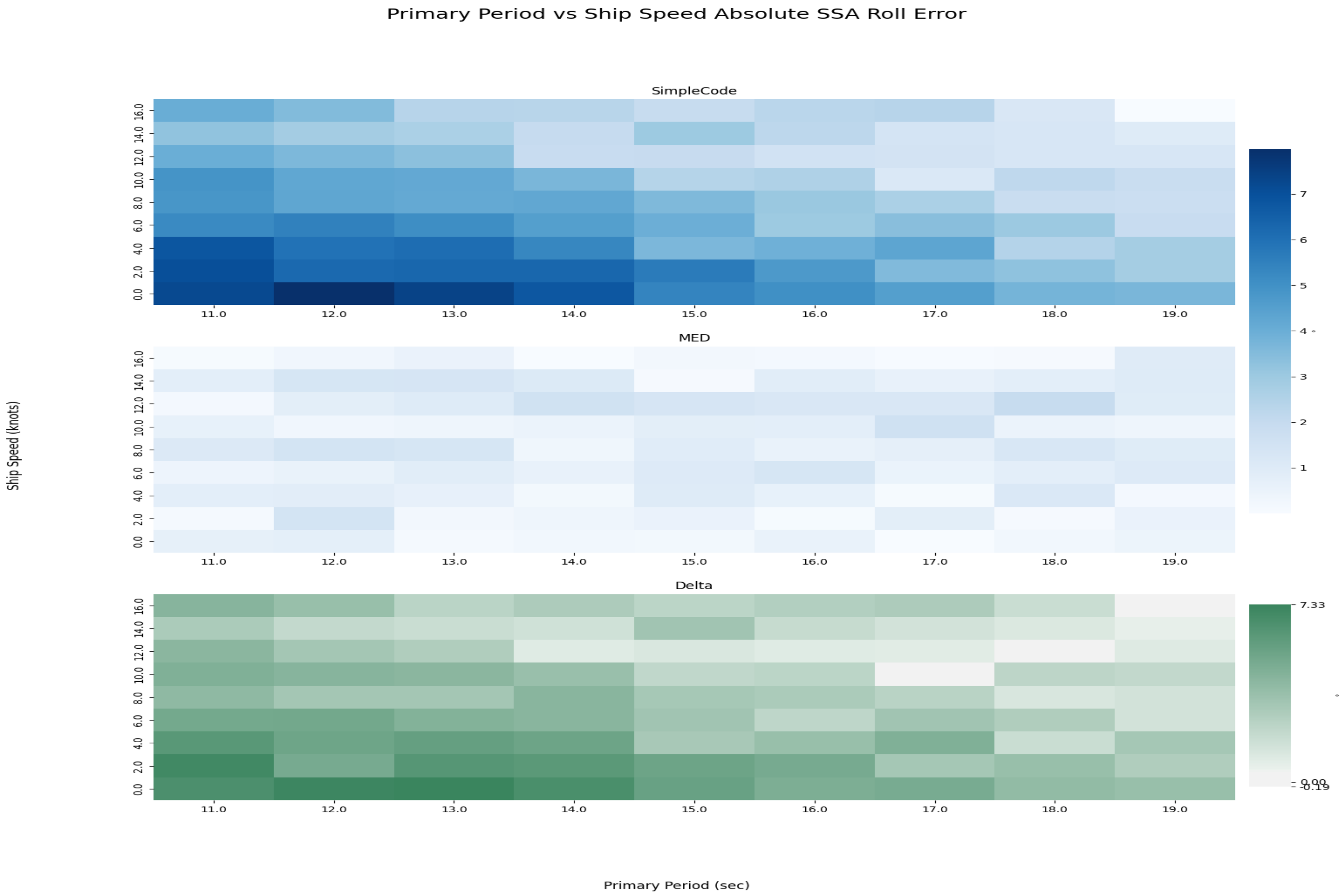

Figure A-35: Primary wave period wave height vs ship speed absolute SSA roll error.

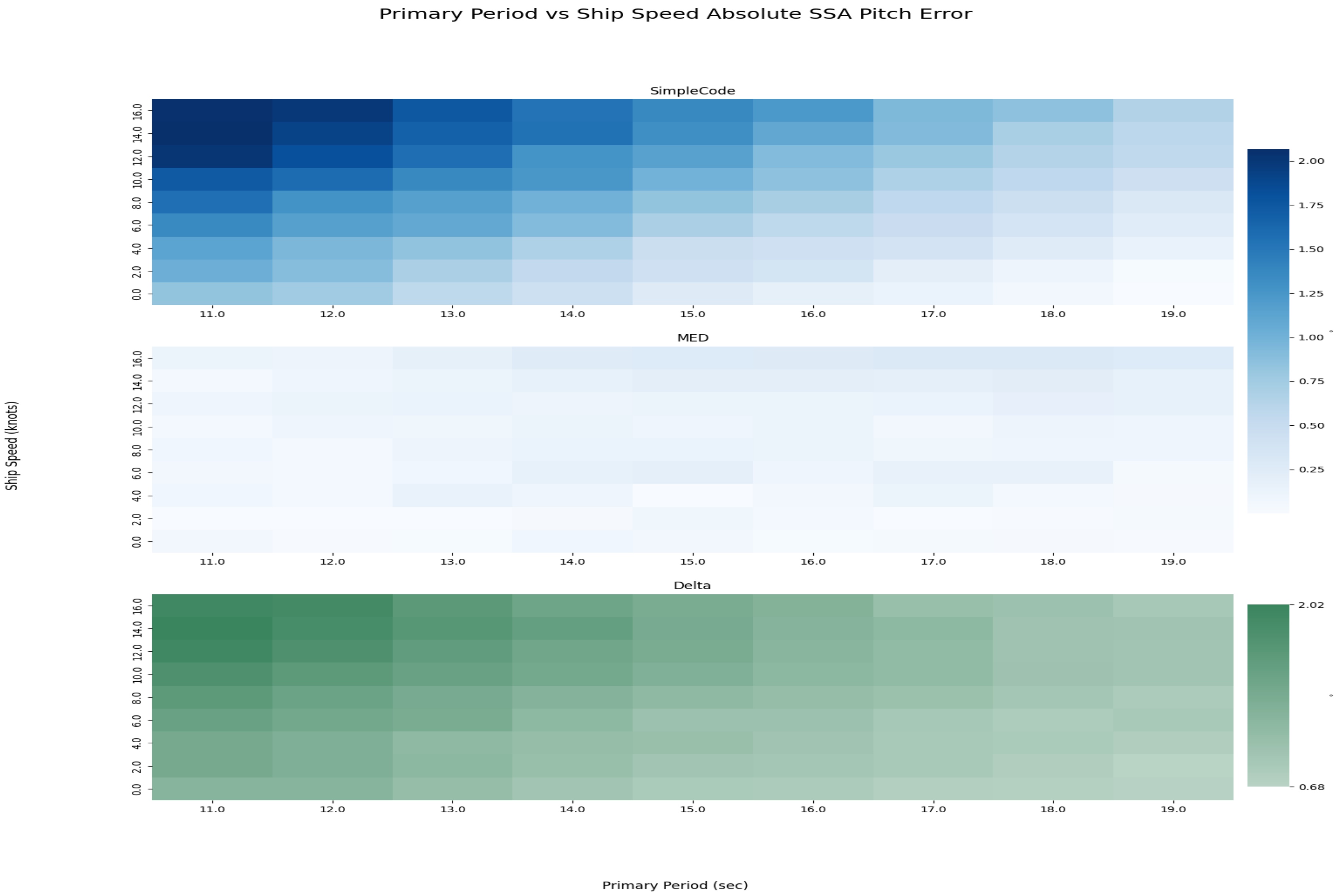

Figure A-36: Primary wave period wave height vs ship speed absolute SSA pitch error.

Appendix B

Bimodal Training vs Transfer Learning Heatmaps

B.1 Additional Secondary Heading Testing Examples

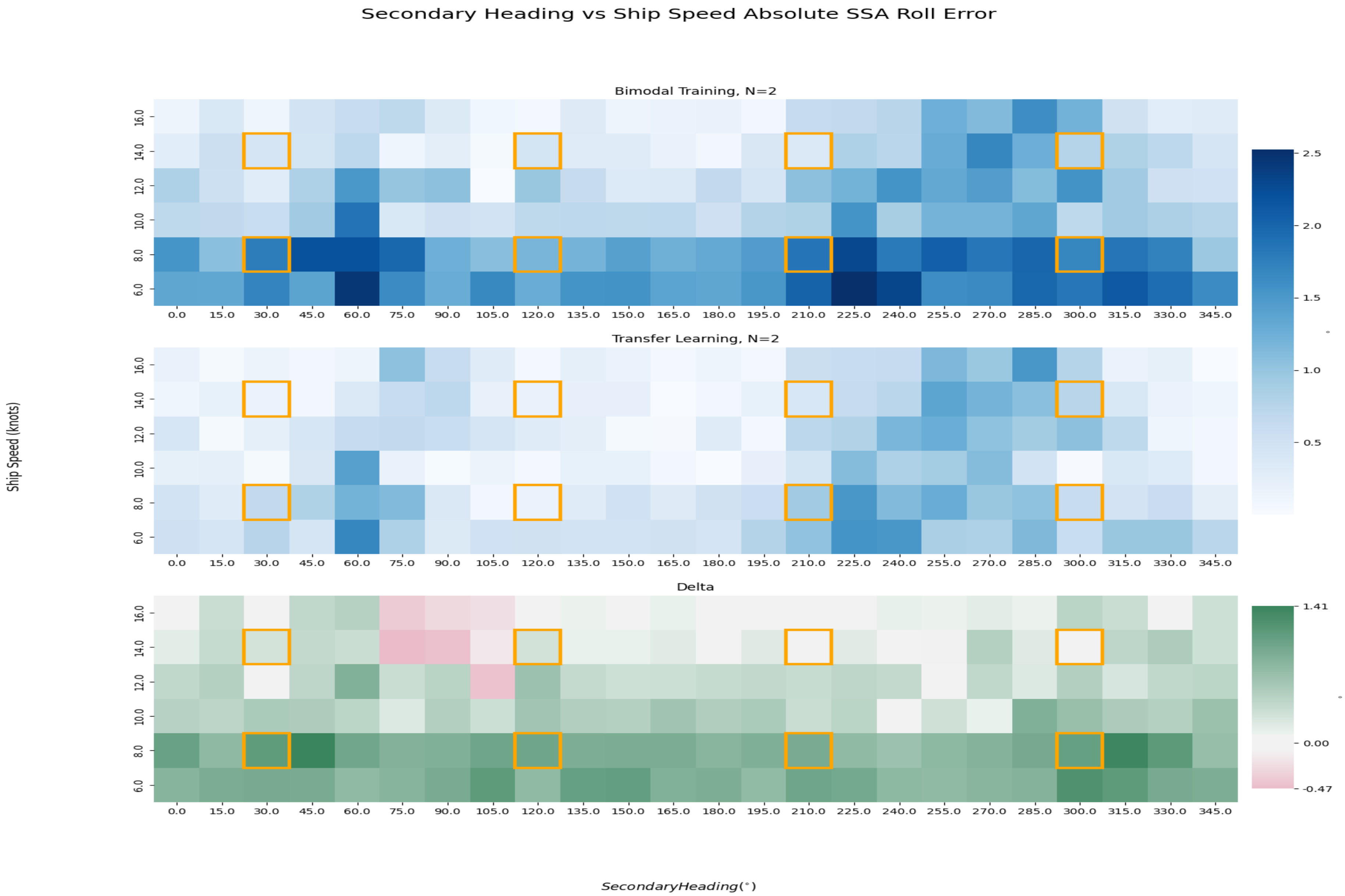

Figure B-1: Secondary wave heading vs Ship Speed absolute SSA roll error with a controlled sample.

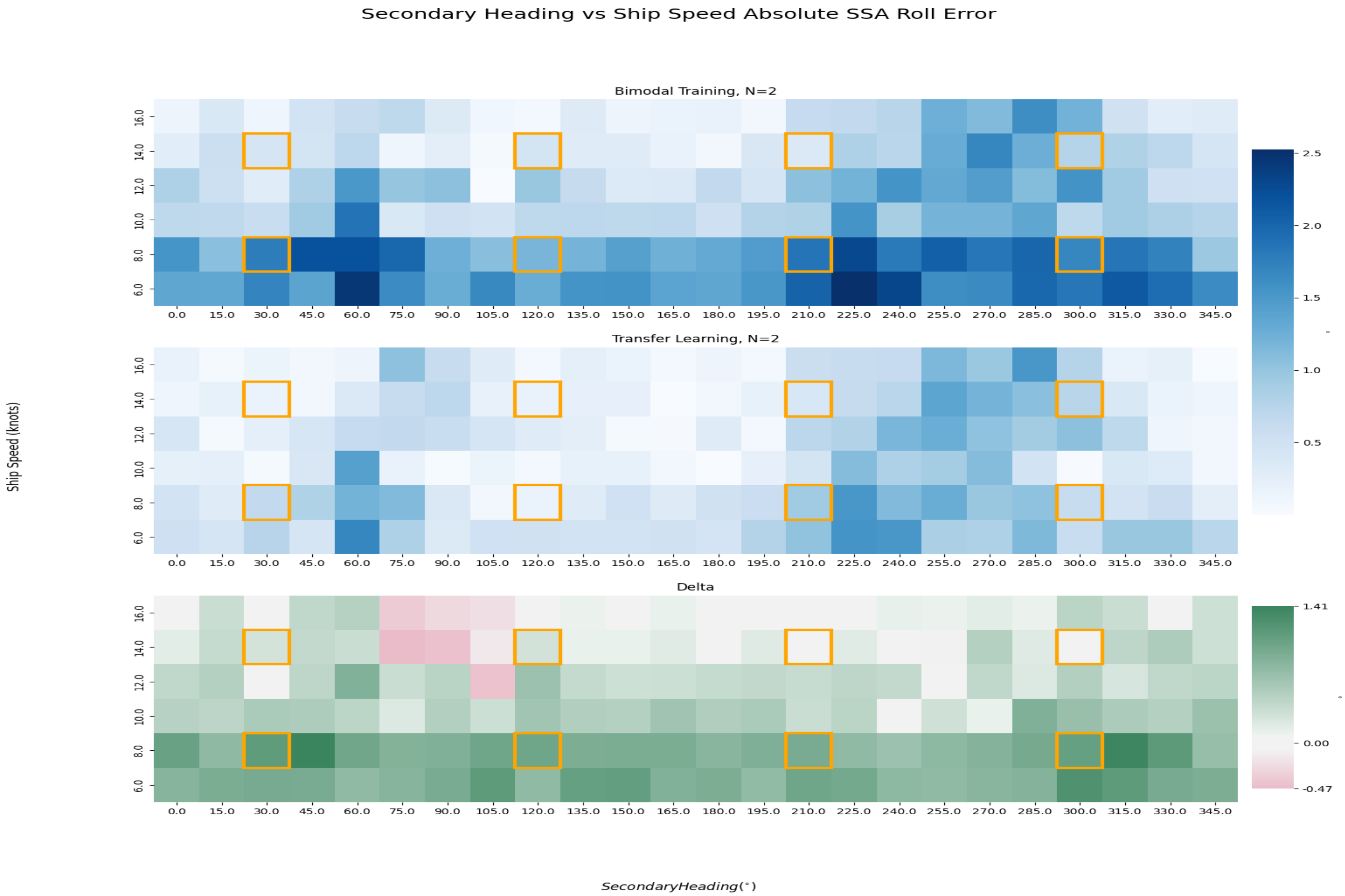

Figure B-2: Secondary wave heading vs ship speed absolute SSA pitch error with a controlled sample.

Appendix C

Repository

The code used to produce the results in this thesis is available on GitHub: `https://github.com/jrkbvb/SimpleCode_to_LAMP_LSTM`.

Bibliography

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML, March 2019. arXiv:1810.09502 [cs, stat].

[2] Vadim Belenky, Kenneth M. Weems, Christopher C. Bassler, Martin J. Dipper, Bradley L. Campbell, and Kostas J. Spyrou. Approaches to rare events in stochastic dynamics of ships. *Probabilistic Engineering Mechanics*, 28:30–38, April 2012.

[3] Richard C Bishop, William Belknap, Charles Turner, Beverly Simon, and Joseph H Kim. GM, Roll Damping, and Above-Water Form on the Roll Response of Model 5613.

[4] I Dori. Chapter 8 Lecture Notes from Introduction to Machine Learning., 2021.

[5] Ömer Faruk Ertuğrul. A novel type of activation function in artificial neural networks: Trained activation function. *Neural Networks*, 99:148–157, March 2018.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, July 2017. arXiv:1703.03400 [cs].

[7] S Hochreiter and J Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

[8] Dayne M Howard. Quantifying Extreme Event Statistics for Ship Motions and Loads Using Low-Fidelity Models and Recurrent Neural Networks.

[9] Michael D Levine, Vadim Belenky, and Kenneth M Weems. Method for Automated Safe Seakeeping Guidance. 2021.

[10] Frédéric Li, Kimiaki Shirahama, Muhammad Adeel Nisar, Xinyu Huang, and Marcin Grzegorzek. Deep Transfer Learning for Time Series Data Based on Sensor Modality Classification. *Sensors*, 20(15):4271, July 2020.

[11] Woei-Min Lin, Matthew Collette, David Lavis, Stuart Jessup, and John Kuhn. Recent Hydrodynamic Tool Development and Validation for Motions and Slam Loads on Ocean-Going High-Speed Vessels.

[12] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[13] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-Learning with Implicit Gradients, September 2019. arXiv:1909.04630 [cs, math, stat].

[14] Marc Rußwurm, Sherrie Wang, Marco Körner, and David Lobell. Meta-Learning for Few-Shot Land Cover Classification, April 2020. arXiv:2004.13390 [cs, stat].

[15] Themis Sapsis, Vadim Belenky, Kenneth Weems, and Vladas Pipiras. Extreme properties of impact-induced vertical bending moments.

[16] Junyou Shi, Qingjie He, and Zili Wang. A Transfer Learning LSTM Network-Based Severity Evaluation for Intermittent Faults of an Electrical Connector. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(1):71–82, January 2021.

[17] Felipe Leno Da Silva and Anna Helena Reali Costa. A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. *Journal of Artificial Intelligence Research*, 64:645–703, March 2019.

[18] Zhong Yi Wan, Pantelis Vlachas, Petros Koumoutsakos, and Themistoklis Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLOS ONE*, 13(5):e0197704, May 2018.