# System Identification of a Non-Holonomic Wheeled Mobile Robot

Karam, Carlo B.  Khairallah, Nadim F.

## 1. PROJECT DESCRIPTION

### 1.1. The Platform

The Kobuki mobile base is a low-cost research platform designed to advance control theory related research, particularly nonlinear control. It is a wheeled mobile robot equipped with two differentially driven wheels, and two castor wheels for support. With continuous, fully integrated operation in mind, the platform provides external power output for a main computer and other possible actuators, and sports a fully embedded system equipped with a coded motherboard and factory calibrated odometry [1].

As with most commercially available mobile robots, the Kobuki platform does not allow direct motor voltage control, but is instead equipped with a low level PID which tracks input reference velocities. This allows for a higher-level control through ROS (Robot Operating System). However, simulated models of the Kobuki base are available, allowing direct PWM control of the motors. As students of the System Identification course, we will be taking advantage of such models in order to generate our input-output data and perform parametric system identification on the dynamics of the mobile robot.

### 1.2. Objective

As part of a Final Year Project (FYP), Carlo Karam and his team are developing control algorithms on the aforementioned Kobuki platform. More specifically, the FYP objective is to develop a domestic service robot capable of serving drinks to household residents. In order to achieve this purpose, the mobile robot will have to carry cups from a kitchen to a given seating area while following a person at walking speed, and then autonomously deliver the drinks to the waiting guests via trajectory planning and tracking. Thus, a valid controller needs to be developed to allow the successful completion of these tasks.

System identification methods will play an important role in the controller design process through two different contributions. First, despite the nonlinear dynamics and non-holonomic constraints of the robot, we will attempt to approximate the nonlinear I/O relationship between the actuators input (motor PWM) and the platform output (positions, velocities and orientation) by a linear relationship, which would ultimately allow us to conduct the controller design process on a linear system. This step would replace deriving the actual model from the actuators and WMR dynamics. Second, although we have nominal values for parameters in the platform datasheet, parametric uncertainty remains a problem as some of these parameters are generally taken from ideal CAD models (i.e. inertia and geometrical parameters), especially since this particular application entails a varying mass parameter during operation (the progression from fully loaded with cups to empty, each cup weighing around 600 grams).

As such, in order to achieve the "Smart Coffee Delivery" goal, we have opted for an adaptive controller that would be able to handle the variation in the model parameters during operation. In the implementation of the adaptive controller, we will have to estimate some parameters in real time (i.e. mass), through an online recursive least-squares algorithm.

# 2. DATA PREPROCESSING

## 2.1. Input Data

The input into the system will be a Pseudo-Random Binary Signal (PBRS), so that it is sufficiently informative and rich to excite all the system dynamics and lead to a good identification. A PBRS is a deterministic signal whose frequency properties mimic white noise. Although a PRBS is inherently periodic (period length of $2^n - 1$), it is possible to generate a nonperiodic input signal by generating a signal of period length N but only taking part of the generated samples as the input.

For the purposes of the experiment, we have generated two different 1000-point PRBS and fed them into the system as PWM signals for the left and right motors. Both inputs were chosen to vary in a positive range so as not to cause the robot to constantly change directions as can be seen in figure 1.

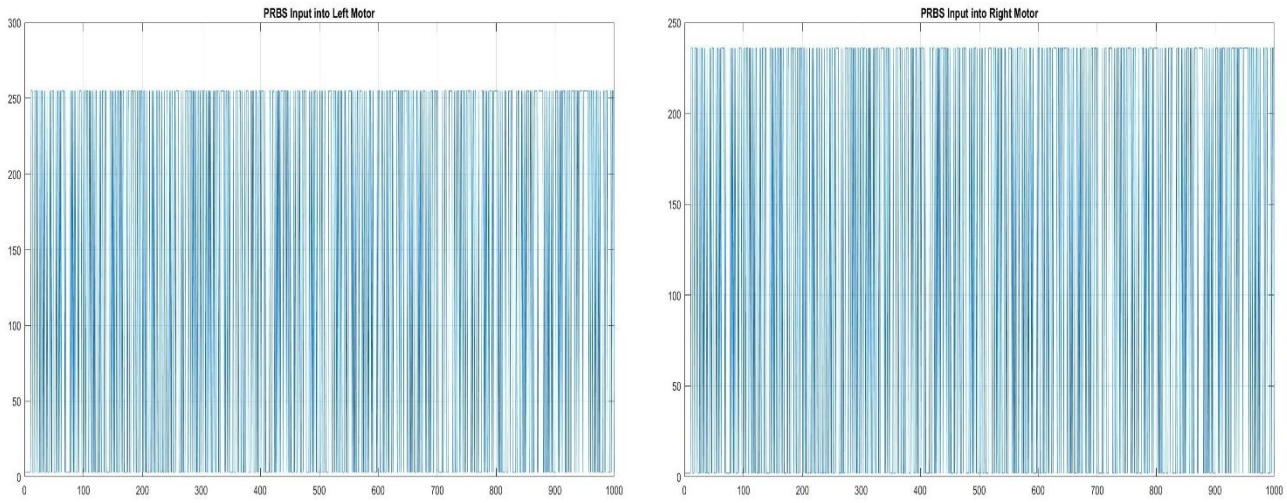


*Figure 1. PWM Input into Left Motor (left) and Right Motor (right)*

These inputs clearly have nonzero mean, meaning that we should take that into account before proceeding with the data analysis. This is as straightforward as subtracting the sample means from the data as such:

$$u_1[k] = u_1^m[k] - \overline{u_1} = u_1^m[k] - \frac{1}{N}\sum_{k=1}^{N} u_1^m[k]$$

$$u_2[k] = u_2^m[k] - \overline{u_2} = u_2^m[k] - \frac{1}{N}\sum_{k=1}^{N} u_2^m[k]$$

Computations reveal that $\overline{u_1} = 128.7480$ and $\overline{u_2} = 118.7660$. With the mean now taken care of, the two PRBS inputs become very similar to gaussian white noise (zero mean).

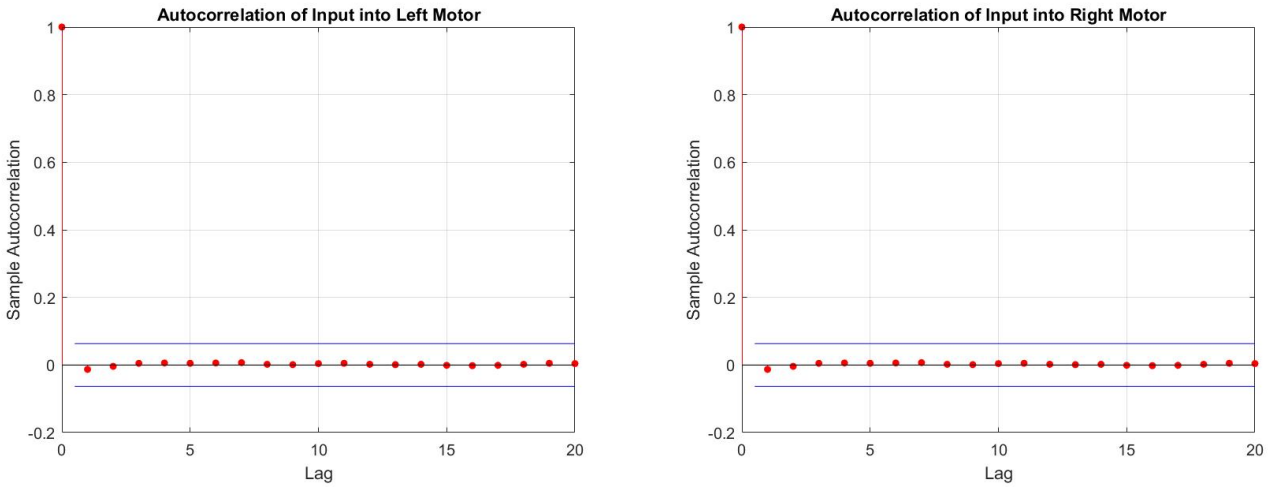


*Figure 2. Sample Autocorrelation of Input into Left Motor (left) and Right Motor (right)*

Figure 2 shows that the obtained signal auto-correlations are impulses at the origin, thus confirming that the PRBS does in fact mimic white-noise properties and is persistently exciting.

### 2.2. Output Data

The output data in our experiment consists of the robot's linear and angular velocities. While both of them exhibit low sample means, it remains necessary for us to account for them by also subtracting them from the data as we did for the inputs. The computations indicate that our linear velocity mean $\overline{y_1} = 0.2851$ and our angular velocity mean $\overline{y_2} = 0.0542$.

With both our input and output data now preprocessed, we can move on to the preliminary analysis phase, which will determine whether or not we can model our input-output relationship as linear, and hint at the lags needed to build our models.

## 3. PRELIMINARY DATA ANALYSIS

To analyze if the I/O relation can be modelled using a linear system, we calculated the correlation and coherence using the input and output auto-spectrum/cross-spectrum for a wide frequency range.

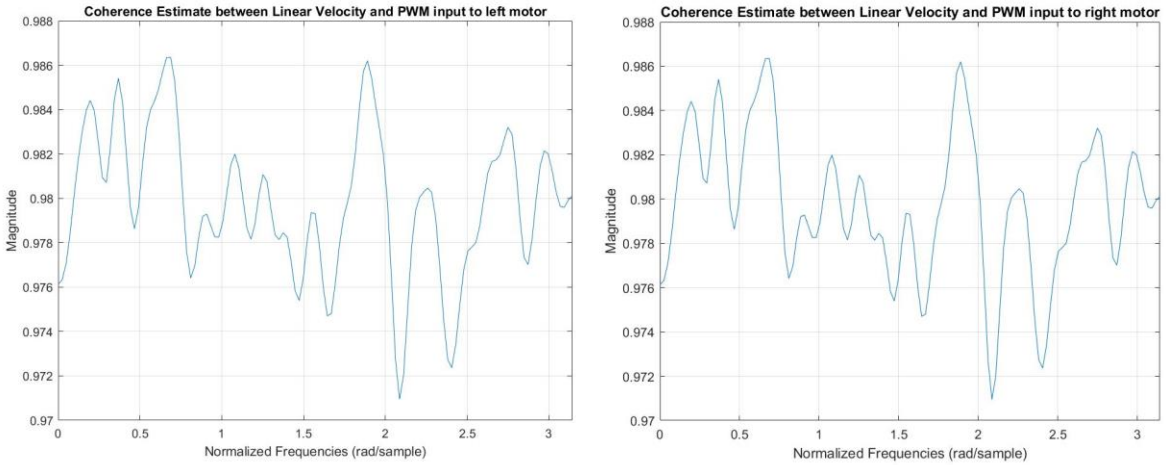


*Figure 3. Coherence Estimates between Linear Velocity and Inputs into Left Motor (left) and Right Motor (right)*

Figure 3 clearly shows a high magnitude coherence between the inputs and outputs, thus indicating an I/O relationships which can be represented by a linear system. On the other hand, figure 4 below shows a somehwat weak linear relationship between the PWM inputs and the angular velocity outputs of the robot.

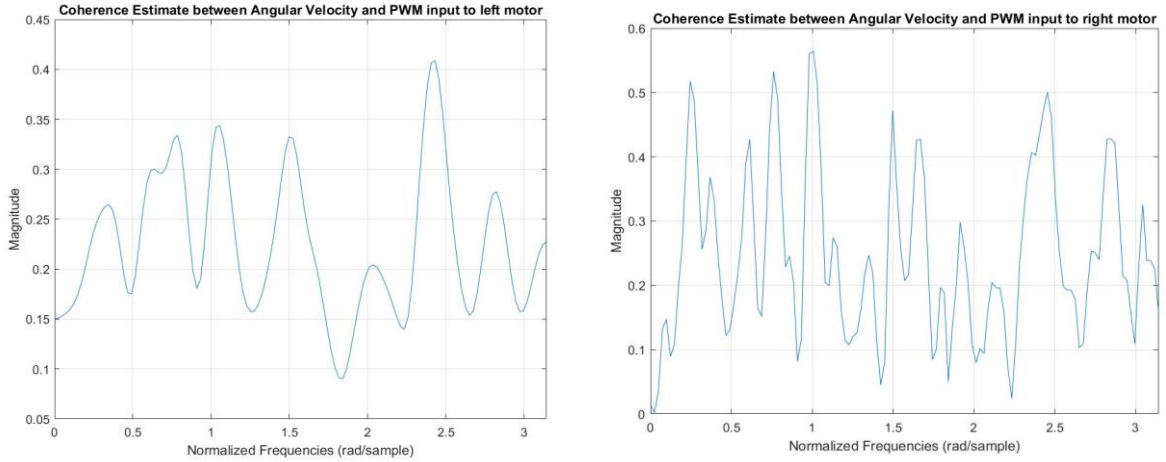


*Figure 4. Coherence Estimates between Angular Velocity and Inputs into Left Motor (left) and Right Motor (right)*

The frequency range of our system does in fact extend over the interval $[0,2\pi]$ as shown by figure 5 below.

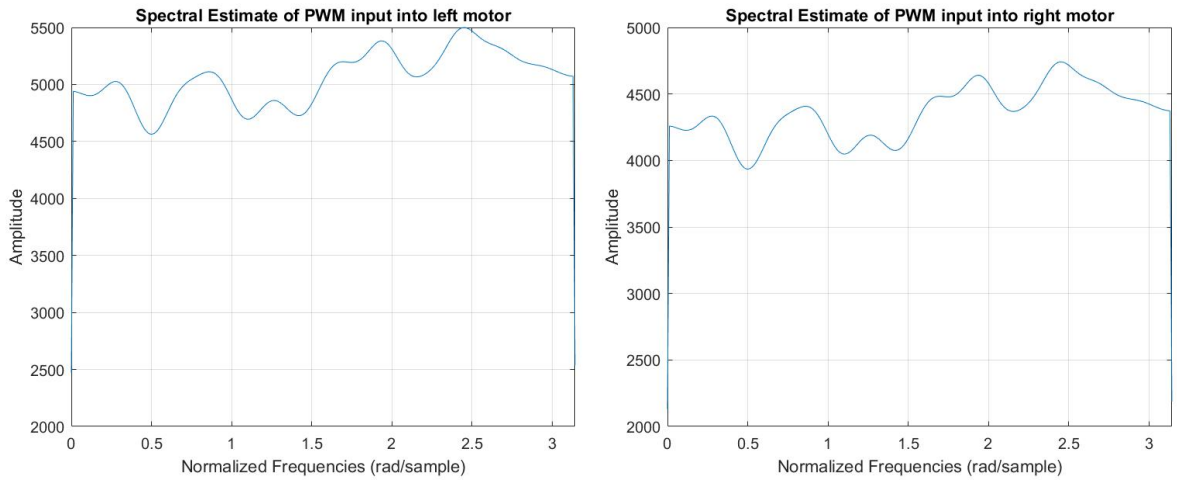


*Figure 5. Spectral Estimates of PWM inputs into Left and Right motors*

The spectral estimates shown in figure 4 were generating using Welch's method with a window size of 32. These results indicate that the system operates within the entire frequency range of $[0,2\pi]$.

# 4. PARAMETRIC IDENTIFICATION

## 4.1. Model Fitting

The outputs were designed to be the WMR's linear and angular velocities based on two main criteria: controller design and cross-correlation plots.

First, the dynamic Model-Reference Adaptive Controller (full state MRAC feedback) designed for the Adaptive Control course project ensures that the system states, selected to be the linear and angular velocities of the WMR, tracks the reference model's velocities. These reference velocities are computed - by passing through the reference model state space representation - the specified velocities, produced by a kinematic controller to ensure that the WMR tracks the desired path. So, the adaptive controller compensates the varying dynamics of the WMR (mass changing when delivering cups) to guarantee that the WMR's velocities track the velocities calculated by the kinematic controller to result in perfect path following. The controller architecture is shown in figure 6 and the circular trajectory tracking result is seen in figure 7. Note that, in figure 7, the blue circle (robot) is over the red circle (specified circular trajectory), indicating that the robot is following the desired trajectory. Also, the blue circle has the same shape as the green circle proving that the robot velocities are following the reference model velocities (without perfect model tracking, blue and green circles wouldn't have the same shape) but the blue and green circle are not superposed because of the initial difference in position between the robot and the reference model created in the transient phase (i.e. when velocities weren't the same).

Additionally, the WMR's pose can be easily calculated from the linear and angular velocities by applying the kinematics equation of the unicycle-like model.

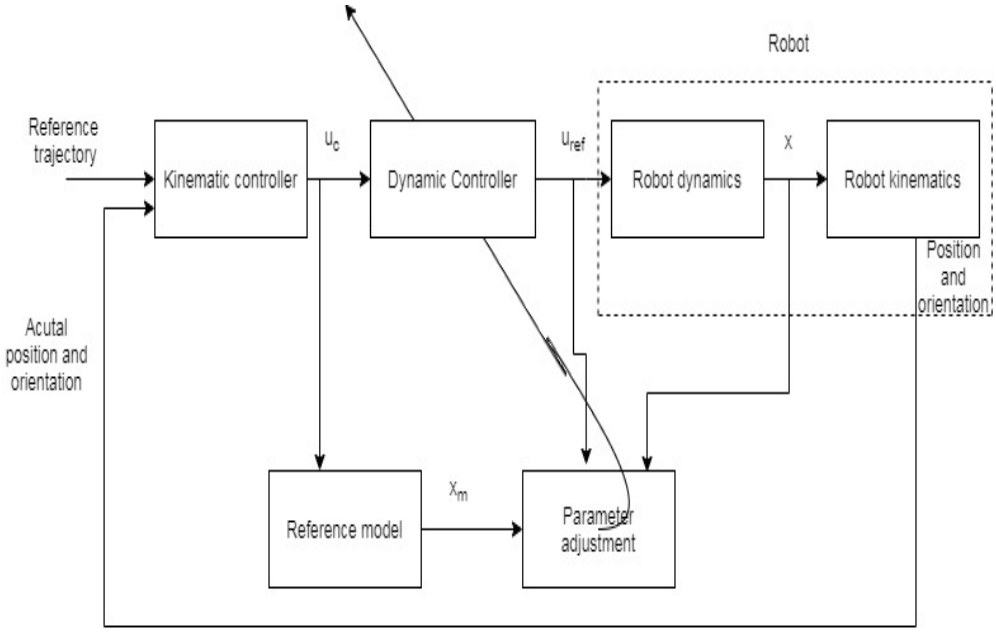


*Figure 6. Block Diagram Representation of Adaptive Controller*

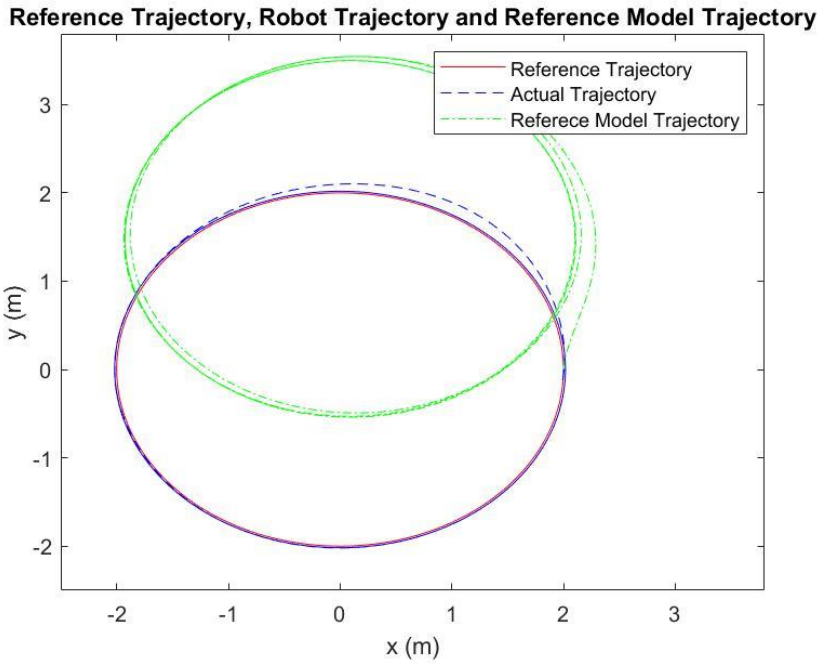


*Figure 7. Trajectory Tracking Performance of Adaptive Controller*

Second, this selection was based on a careful inspection of the cross-correlation plots of WMR's positions and motor PWMs on one hand and WMR's velocities and motor PWMs on the other hand. Indeed, the plots between positions and PWMs revealed very little cross-correlation and suggested that position outputs could not be well estimated from motor PWMs. WMR's velocities plots, however, clearly showed a large cross-correlation at a lag of two, suggesting that the velocities at the current time step are affected by the motor PWMs two time steps before (figures 8 and 9).

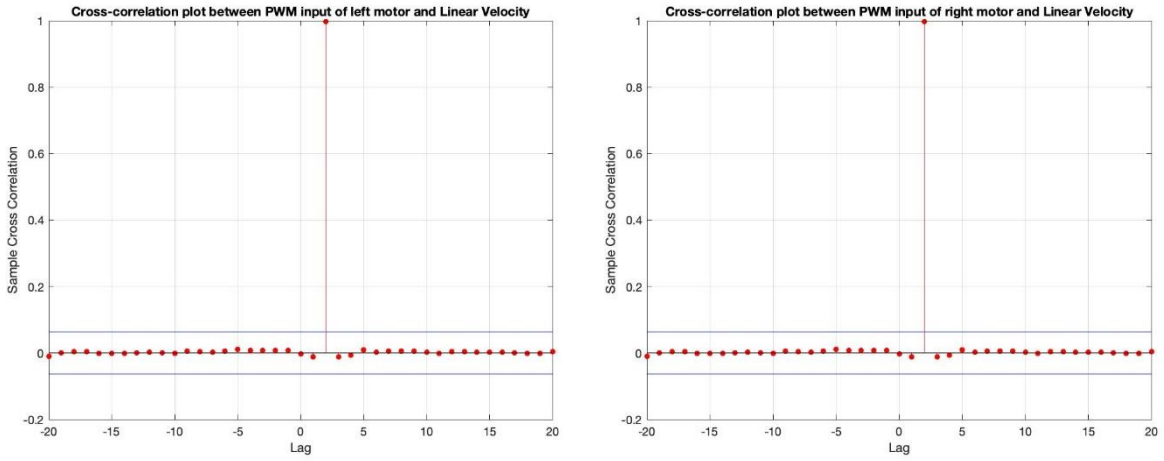


*Figure 8. Cross-Correlation between Linear Velocity and PWM inputs*

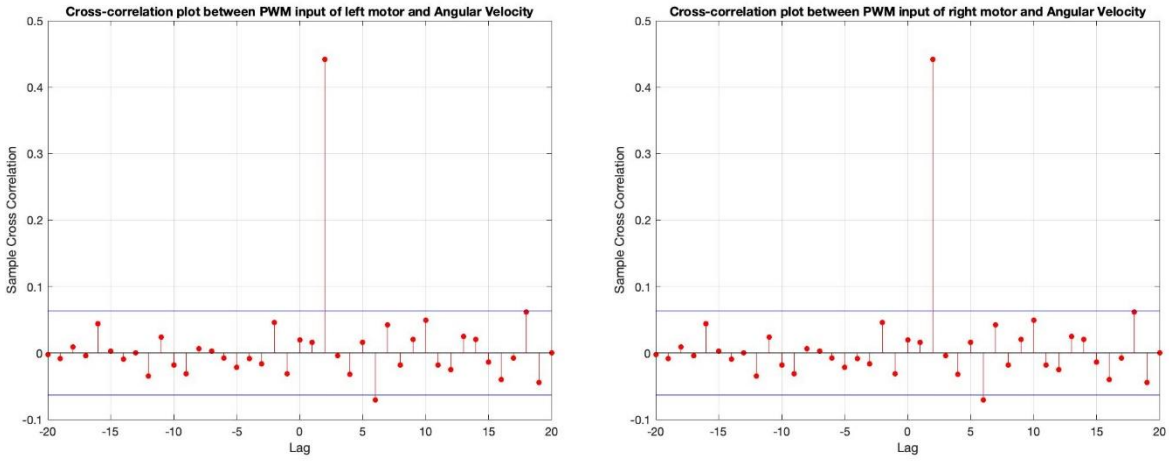


*Figure 9. Cross-Correlation between Angular Velocity and PWM inputs*

## 4.2. Model Structure

In the modelling process, a general linear difference equation that represents an ARX (Auto-Regressive with eXogenous input) model is chosen for both outputs (linear and angular velocities). We get two linear difference equations, one for each output. Also, in each of these linear difference equations, both inputs appear as both motor PWMs affect each output:

Model structure for linear velocity ($y_1 = v$):

$$A_1(q^{-1})y_1[k] = B_1(q^{-1})u_1[k] + B_2(q^{-1})u_2[k] + \epsilon[k]$$

Where:

$$A_1(q^{-1}) = 1 + {a_1}^1 q^{-1} + {a_2}^1 q^{-2} + \cdots + {a_{na}}^1 q^{-na}$$

$$B_1(q^{-1}) = {b_1}^1 q^{-1} + {b_2}^1 q^{-2} + \cdots + {b_{nb}}^1 q^{-nb}$$

$$B_2(q^{-1}) = {b_1}^2 q^{-1} + {b_2}^2 q^{-2} + \cdots + {b_{nb}}^2 q^{-nb}$$

Model structure for angular velocity ($y_2 = \omega$):

$$A_2(q^{-1})y_2[k] = C_1(q^{-1})u_1[k] + C_2(q^{-1})u_2[k] + \epsilon[k]$$

Where:

$$A_2(q^{-1}) = 1 + {a_1}^2 q^{-1} + {a_2}^2 q^{-2} + \cdots + {a_{na}}^2 q^{-na}$$

$$C_1(q^{-1}) = {c_1}^1 q^{-1} + {c_2}^1 q^{-2} + \cdots + {c_{nc}}^1 q^{-nc}$$

$$C_2(q^{-1}) = {c_1}^2 q^{-1} + {c_2}^2 q^{-2} + \cdots + {c_{nc}}^2 q^{-nc}$$

with $u_1 = PWM_{left}$ and $u_2 = PWM_{right}$

With order of $A_{1,2}$ is na, order of $B_{1,2}$ is nb and order of $C_{1,2}$ is nc. Also take nb = nc to have same order of both inputs in both linear difference equations. Moreover, notice that $B_{1,2}$ and $C_{1,2}$ do not have feedthrough terms since the input takes some time (lags) to affect the output.

To determine the correct model structure, a careful observation of the autocorrelation plots of the linear and angular velocities as well as the crosscorrelation plots of between both velocities and both motor PWMs is conducted:

By examining the autocorrelation plots of the linear and angular velocities in figure 10, we remarked that these resemble that of white noise (i.e. an impulse at the origin). This observation demonstrates that the velocities do not have a defined time structure as velocities do not correlate with any delayed version of themselves. In other words, we cannot predict current velocities from past velocities data. As a consequence of this white noise-like autocorrelation plots, our system does not have an autoregression on velocities which implies that the order of A is 0 (na=0).

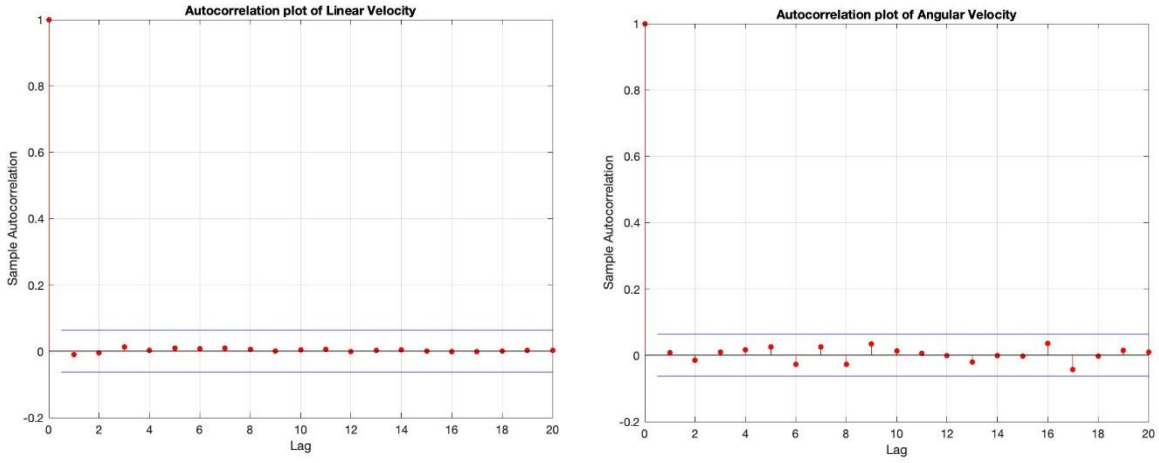


*Figure 10. Autocorrelation of Linear and Angular Velocities*

As a result, our model structure becomes a FIR model for both outputs:

$$y_1[k] = B_1(q^{-1})u_1[k] + B_2(q^{-1})u_2[k] + \epsilon[k]$$

$$y_2[k] = C_1(q^{-1})u_1[k] + C_2(q^{-1})u_2[k] + \epsilon[k]$$

Furthermore, the observation of crosscorrelation plots in figures and 7 and 8 shows a clear relation between each output (linear velocity on one hand and angular velocity on the other hand) at a given time step and both inputs (left and right motors PWMs) two time steps earlier. Also, the crosscorrelation at lags different than two were minimal and negligible compared to that at lag two.

We therefore selected nb = nc =2 and took ${b_1}^1 = {b_1}^2 = {c_1}^1 = {c_1}^2 = 0$ since outputs are not correlated with inputs from the previous time step.

Consequently, our ideal FIR models for estimating linear and angular velocities from left and right motors PWMs become:

$$y_1[k] = {b_2}^1 u_1[k-2] + {b_2}^2 u_2[k-2] + \epsilon[k]$$

$$y_2[k] = {c_2}^1 u_1[k-2] + {c_2}^2 u_2[k-2] + \epsilon[k]$$

Refer to the above models as the ideal FIR models based on autocorrelation of outputs and crosscorrelations between inputs and outputs plots. Each of these models has two unknown parameters defined as:

$$\theta_1 = \begin{bmatrix} {b_2}^1 \\ {b_2}^2 \end{bmatrix} \; and \; \theta_2 = \begin{bmatrix} {c_2}^1 \\ {c_2}^2 \end{bmatrix}$$

As a result, the above ideal FIR models can be cast as linear regression models:

$$y_1[k] = \phi^T \theta_1 + \epsilon[k]$$

$$y_2[k] = \phi^T \theta_2 + \epsilon[k]$$

Where

$$\phi^T = [u_1[k-2] \quad u_2[k-2]]$$

Moreover, we fitted the I/O data to increasingly complex FIR and ARX models (both for linear and angular velocities as outputs) with orders ranging from 2 to 8 for FIR and 3 to 12 for ARX.

We assumed that na = nb and na = nc in the ARX models.

## 4.3. Model Selection

We used the Least Squares Estimation (LSE) method to find the parameters $\theta$ of the ideal FIR, and the increasingly complex FIR and ARX models. Furthermore, we found, for each model, the value of the cost function $V(\theta)$ (half mean square of residual), the estimated noise variance $\sigma_e$, the covariance matrix of the estimates $cov(\theta)$.

$$V(\theta) = 0.5 \sum_k \epsilon[k]^2$$

$$\sigma_e = \frac{2V(\theta)}{N-p}$$

$$cov(\theta) = \sigma_e (\Phi^T \Phi)^{-1}$$

We then compared all these models in two scenarios. The first is with the simulated data we obtained that was just preprocessed to remove the mean from the signals. Since our data is generated from a simulated model, it does not contain noise. The second scenario is closer to the actual system: in addition to removing the DC offset in the data, we corrupted the velocities outputs with white noise. The variance of this white noise is calculated in the code to give a desired signal-to-noise ratio (SNR) that the user can specify. In the scenario with noise, we chose a SNR of 5 to corrupt the data.

The results of our simulations and estimations for both outputs and both scenarios are presented below:

Scenario 1: no noise (take simulated data as is without corrupting it)

| | | Linear Velocity y1 | | Angular Velocity y2 | |
|---|---|---|---|---|---|
| | | $V(\theta)$ | $\sigma_e$ | $V(\theta)$ | $\sigma_e$ |
| Ideal FIR model | | 0.1114 | 2.23e-4 | 6.1639 | 0.0124 |
| FIR | Order = 2 | 39.268 | 0.0787 | 7.7563 | 0.0155 |
| | Order = 4 | 0.2485 | 4.99e-4 | 6.1665 | 0.0124 |
| | Order = 6 | 0.3305 | 6.65e-4 | 6.1644 | 0.0124 |
| | Order = 8 | 0.1303 | 2.63e-4 | 6.1451 | 0.0124 |
| ARX | Order = 3 | 39.2647 | 0.0788 | 7.7559 | 0.0156 |
| | Order = 6 | 0.1972 | 3.97e-4 | 6.1674 | 0.0124 |
| | Order = 9 | 0.0809 | 1.63e-4 | 6.1815 | 0.0125 |
| | Order = 12 | 0.1111 | 2.25e-4 | 6.14 | 0.0124 |

| | | Linear Velocity y1 | Angular Velocity y2 |
|---|---|---|---|
| | | $cov(\theta)$ | $cov(\theta)$ |
| Ideal FIR model | | $\begin{bmatrix} 0.00000043 & -0.00000046 \\ -0.00000046 & 0.0000005 \end{bmatrix}$ | $\begin{bmatrix} 0.000035 & -0.000038 \\ -0.000038 & 0.000041 \end{bmatrix}$ |
| FIR | Order = 2 | $\begin{bmatrix} 0.000224 & -0.000242 \\ -0.000242 & 0.000261 \end{bmatrix}$ | $\begin{bmatrix} 0.000044 & -0.000048 \\ -0.000048 & 0.000052 \end{bmatrix}$ |
| ARX | Order = 3 | $\begin{bmatrix} 0.0010 & 0.00000023 & -0.00000025 \\ 0.00000023 & 0.000224 & -0.000242 \\ -0.00000025 & -0.000242 & 0.000261 \end{bmatrix}$ | $\begin{bmatrix} 0.0010 & -0.0000026 & 0.0000028 \\ -0.0000026 & 0.000042 & -0.000048 \\ 0.0000028 & -0.000048 & 0.000052 \end{bmatrix}$ |

Scenario 2: with noise (take simulated data and corrupting it with white noise at a specified SNR)

| | | Linear Velocity y1 | | Angular Velocity y2 | |
|---|---|---|---|---|---|
| | | $V(\theta)$ | $\sigma_e$ | $V(\theta)$ | $\sigma_e$ |
| Ideal FIR model | | 7.9836 | 0.0160 | 7.7155 | 0.0155 |
| FIR | Order = 2 | 50.8597 | 0.1019 | 11.6929 | 0.0234 |
| | Order = 4 | 15.8032 | 0.0317 | 10.1232 | 0.0203 |
| | Order = 6 | 15.8652 | 0.0319 | 10.1254 | 0.0204 |
| | Order = 8 | 15.6661 | 0.0316 | 10.1075 | 0.0204 |
| ARX | Order = 3 | 43.8399 | 0.0879 | 9.6197 | 0.0193 |
| | Order = 6 | 7.2483 | 0.0146 | 8.0792 | 0.0163 |
| | Order = 9 | 7.4067 | 0.0149 | 8.0642 | 0.0163 |
| | Order = 12 | 7.2951 | 0.0148 | 8.0501 | 0.0163 |

Noting that we have, $\sigma_{e,actual} = 0.0157$ for $y_1$ and $\sigma_{e,actual} = 0.0077$ for $y_2$.

| | | Linear Velocity y1 | Angular Velocity y2 |
|---|---|---|---|
| | | $cov(\theta)$ | $cov(\theta)$ |
| Ideal FIR model | | $\begin{bmatrix} 0.000226 & -0.000244 \\ -0.000244 & 0.000264 \end{bmatrix}$ | $\begin{bmatrix} 0.000102 & -0.00011 \\ -0.00011 & 0.00012 \end{bmatrix}$ |
| FIR | Order = 2 | $\begin{bmatrix} 0.00035 & -0.00037 \\ -0.00037 & 0.00040 \end{bmatrix}$ | $\begin{bmatrix} 0.000094 & -0.000102 \\ -0.000102 & 0.00011 \end{bmatrix}$ |
| ARX | Order = 3 | $\begin{bmatrix} 0.0010 & 0.0000085 & -0.0000092 \\ 0.0000085 & 0.00029 & -0.00032 \\ -0.0000092 & -0.00032 & 0.00034 \end{bmatrix}$ | $\begin{bmatrix} 0.0010 & -0.0000031 & 0.0000033 \\ -0.0000031 & 0.000078 & -0.000084 \\ 0.0000033 & -0.000084 & 0.000091 \end{bmatrix}$ |

The above results are all in accordance with our expectations. As we can see, the loss function value decreases as the model order increases, and increases as we pass from noisless to noisy scenarios. Furthermore, the covariances increase with the addition of noise to our outputs. For the sake of brevity and avoiding encumbring the document, only the covariances of the low-order models were displayed.

By studying these results, it is clear that the best model to proceed with is the ideal FIR model which offers the best compromise between accuracy and simplicity, thus confirming our initial intuition. We do not have to resort to further tests and criteria as the choice is quite straightforward.

## 5. MODEL VALIDATION

Following the ideal FIR model, we proceeded to produce a comparison between the measured and estimated outputs using our estimated parameters. As explained above, we performed this comparison under two different scenarios, one with unaltered outputs and one with corrupted outputs with a SNR of 5. For the first scenario (no noise), figure 11 shows that the linear velocity

estimate is extremely accurate, whereas that of the angular velocity is a bit lacking. This can be explained by the coherence plots shown previously, where the relationship between angular velocity and PWM inputs was shown to be poorly modeled by a linear I/O relation.

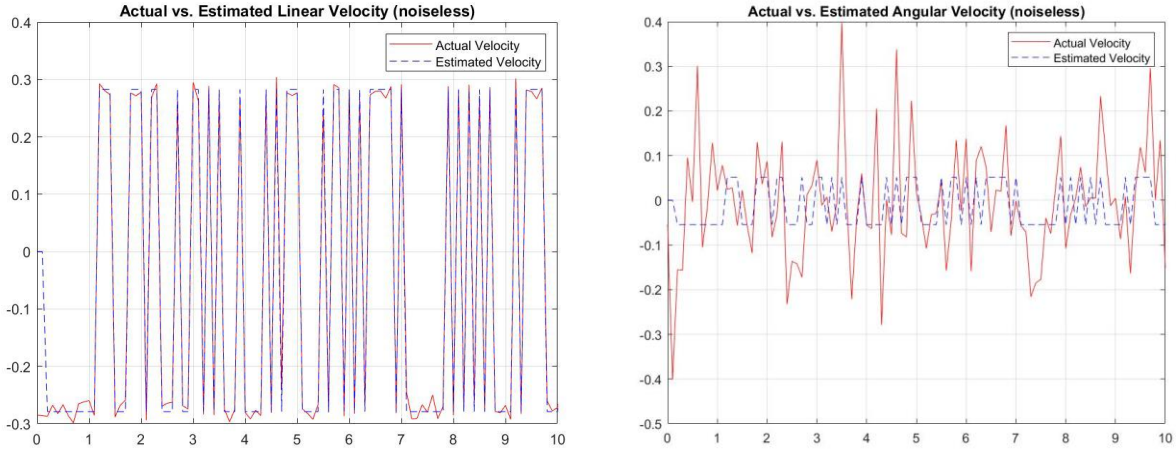


*Figure 11. Actual vs Estimated Linear and Angular Velcoties (uncorrputed output)*

For the case with added noise, we obtain very similar results as well, as shown by figure 12, albeit with a naturally poorer performance.

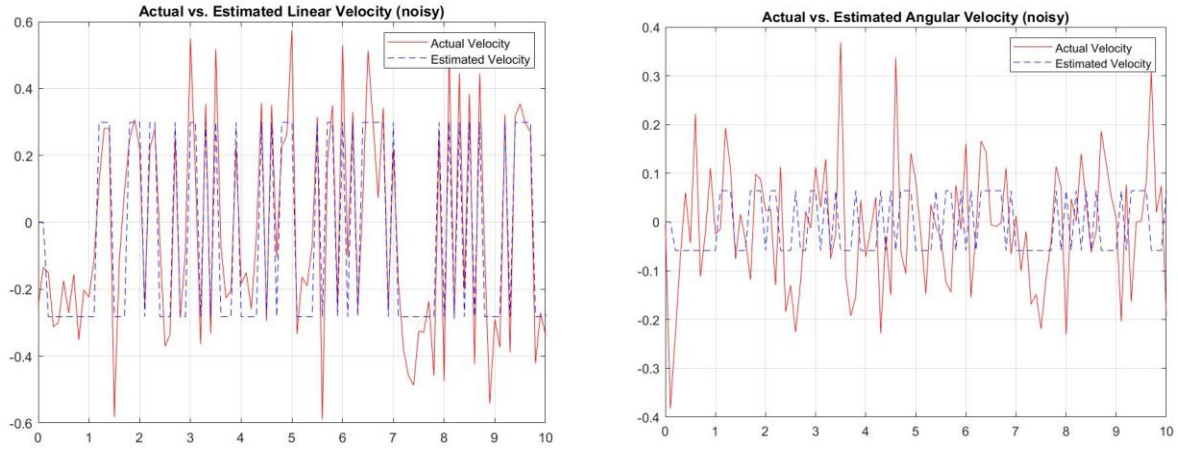


*Figure 12. Actual vs Estimated Linear and Angular Velcoties (corrupted output)*

To further validate our model, we verified that it does in fact satisfy the LSE assumptions: the residual is white-noise like and the regressor and residual are uncorrelated. We obtain similar results in both the noisy and noiseless cases, with the former being shown in figures 13 and 14.

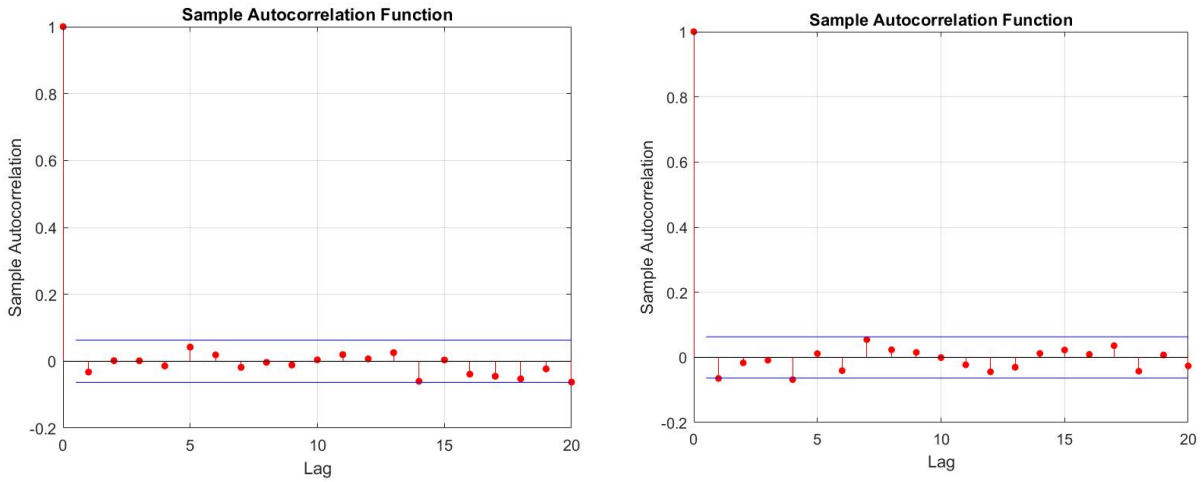


*Figure 14. Autocorrelation Function of the Residual*

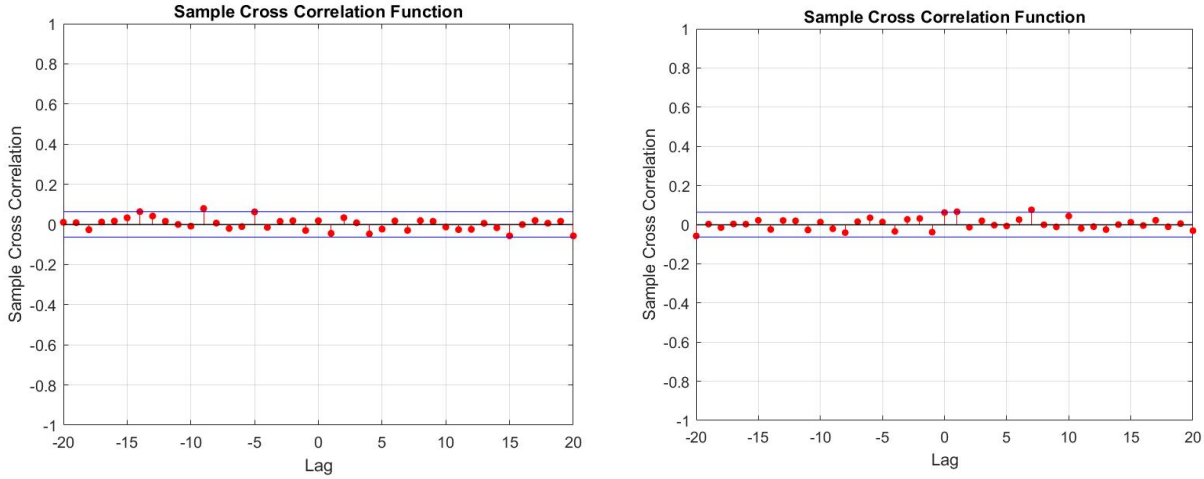


*Figure 13. Cross Correlation between Regressor and Residual (linear and angular outputs)*

As we can see, the residual autocorrelation (for both linear and angular velocities) strongly resembles an impulse at the origin, indicating that it's white-noise like, while the corss correlation between the regressor and the residuals is very close to zero for lags greater or equal to 0.

We can therefore conclude that our model is in fact valid and somewhat accurately estimates the actual dynamic model of the robot.