ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Semester Project
Biorob Laboratory

Sensor Fusion For Speed Estimation On Quadrupedal Robots

Noémie Jaquier

Professor	Auke Ijspeert
Supervisors	Tomislav Horvat, Mostafa Ajallooeian

June 4, 2015

Project Summary

The goal of this project is to estimate the instantaneous velocity of quadrupedal robots in order to know if they get trapped, fall, are deviated or move as expected. Model-free techniques are used on data collected on two different robots: Oncilla and Pleurobot, which was simulated in Webots. The velocity estimation is done with the System Identification Toolbox of Matlab.

The velocities along x, y and z-axis of the Oncilla robot are first estimated using ARX and NLARX models with information from the robot's sensors as inputs, especially the acceleration, and the position given by a Motion Capture System as ground truth. The data from the Motion Capture System are derived to obtain the velocity in the robot's system of coordinates. Models are trained on a virtual experiment, runs during which the robot is swung manually in the air and runs during which its walks straight. An example of results is shown in Figure I.

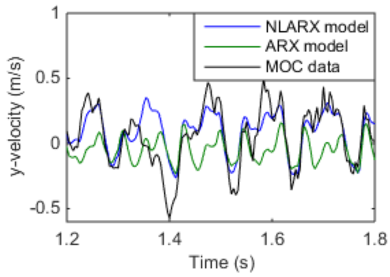


Figure I: Oncilla
Fit ARX: −10.93%. Fit NLARX: 19.96%

As obtained estimations are not reliable, the identification of model is tested with noised derivative of simple functions as input and functions themselves as ground truth. As fit values are higher than 90%, the system identification is able to find good model for noisy data.

The identification of models is then done on Pleurobot simulation trying to make the identification easier. Only the forward velocity is thus estimated, using the accelerations, the forces and the frequency reference as inputs. The identification is made with more complex models as ARMAX and BJ, which are trained on several runs of few experiments. Nonlinear models are not tried as they are very expensive in time of computation. A hand-tuned BJ model is the most accurate to give good fit values over all experiments. The same models trained on two or three experiments have a similar accuracy. An example of results is shown in Figures II.

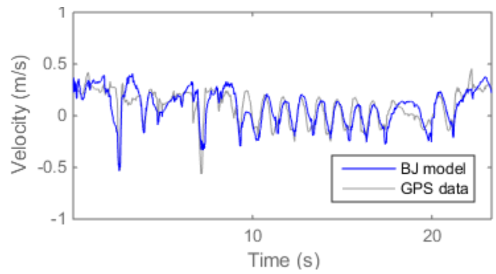


Figure II: Pleurobot, exp 4, run 4
Fit: 30.71%

The same BJ model is then tested on improved Oncilla data with the same inputs and with the lateral velocity as output. Some parameters are modified to obtain estimations which are visually similar to the measured output, even if there are some inexactitudes on amplitude causing low fit values which stay under 20%.

The report finally shows that unexpected behaviours of Pleurobot in simulations can be detected comparing the error between a multiple of the frequency reference and the estimation of velocity with a simple threshold. An example is shown in Figure III.

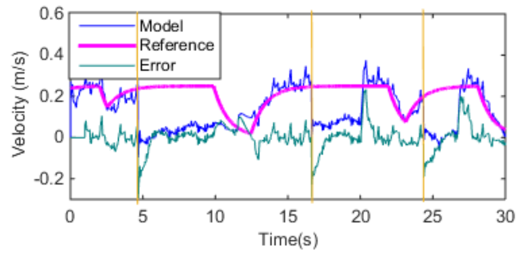


Figure III: Detection of unexpected behaviours

This method of detection should be implemented and tested in real time in Webots and adapted to Oncilla. Moreover, methods as systematic search or genetic algorithms should be tried to find optimal parameters of the models.

Contents

1 Introduction

Humans and animals have naturally a good idea of their location and of the direction and speed at which they are moving. They are able to correct their trajectories and motions as a function of their environment and unexpected changes happening around them. Robots equipped with an Inertial Measurement Unit (IMU) would be able to know how they are rotated and translated at each time if the sensory data collected where perfect. However, as there are noise and inaccuracies in measurements, the state of the robot must be estimated from a combination of sensor data.

The goal of this project is to explore model-free techniques to estimate the instantaneous velocity of quadrupedal robots during their displacements. This estimation can be useful to know if the robot is moving as expected or if it is deviated from its initial trajectory, if it gets trapped somewhere or if it has fallen.

The velocity estimation was done by using the System Identification Toolbox from Matlab. Several models were tried and optimised to fit the velocity estimated from sensory inputs to the real velocity of the robot measured by an external system. Different experiments were thus realised on Oncilla robot using its sensor readings for the estimation and a Motion Capture System as ground truth. Experiments were also realised on Pleurobot simulation on Webots.

Section 2 of this report presents the two quadrupedal robots used in experiments. Section 3 shows the system identification models used to estimate the velocity at each time step. A first set of experiments on Oncilla are then described in Section 4. Section 5 tests system identification on different noisy functions to prove that the method can be used for noisy data. Sections 6 and 7 presents final experiments respectively on Pleurobot simulation and on Oncilla. A method to detect unexpected Pleurobot's behaviours is then developed in Section 8. Section 9 finally concludes the report and makes suggestions about future work that could be done.

2 Quadrupedal Robots

2.1 Oncilla Robot

Oncilla is a compliant quadrupedal robot developed in Biorob. It is adapted for locomotion on unperceived and rough terrains. A picture of Oncilla with its velocity axis is showed in Figure 1. For this robot, the velocity along x-axis is relatively stable as it is the direction in which the robot is the longest. Moreover, the velocity along z-axis is stabilised by gravity. The velocity along y-axis, called the lateral velocity is thus less stable. For this reason and because it is important to know if the robot deviates or falls, the estimation is focused on this direction.

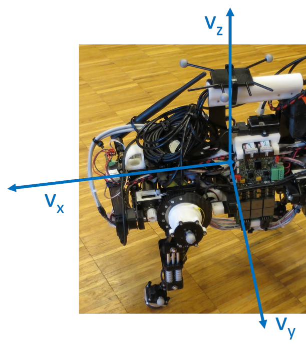


Figure 1: Oncilla Robot

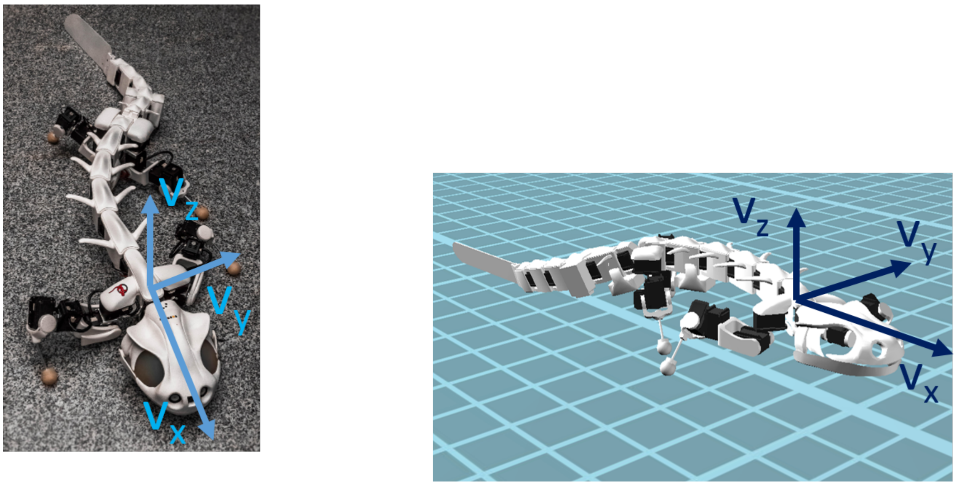
The sensors of the robot provide the different measurements given by Table 1. The part of sensor data used for system identification are described in the next sections.

Time	Sensed joint angles
Raw, pitch and yaw angles estimate	Estimated pitch steepness
Rotated acceleration measurement	Feet 3D forces
Raw acceleration	Rotation matrix
Velocity reference	Rotation reference

Table 1: Some of sensor data provided by Oncilla

2.2 Pleurobot

Pleurobot is a quadruped salamander-like robot. It is also adapted to navigate over rough terrain as one of its applications is search and rescue operations. The estimation of velocity is focused on x-axis because the robot is stable in all directions and has thus no risk of falling. However, it can get stuck against an obstacle or if its tail is blocked by some object, which makes the knowledge of the velocity along x-axis interesting.



(a) Real Pleurobot

(b) Pleurobot simulation on Webots

Figure 2: Pleurobot

The simulation in Webots is used for system identification. Available sensors measurements are given by Table 2. Used data are described in the next sections.

Time	Raw, pitch and yaw angular velocity estimate
Acceleration measurement	Feet 3D forces
Rotation matrix	Attitude
Joints angles	Frequency reference

Table 2: Sensor data provided by Pleurobot simulation

3 System Identification Models

3.1 System Identification

The goal of system identification is to derive mathematical models of dynamic systems using their inputs and outputs. Figure 3 shows the general form of such a system. To construct the model, an input is applied on the system and is measured, as is the resultant output. The model is then computed to give the output of the system in function of input data and disturbances, which are added as the system is not considered as ideal. The computation of model parameters is done using for example least-square estimation or maximum likelihood estimation.

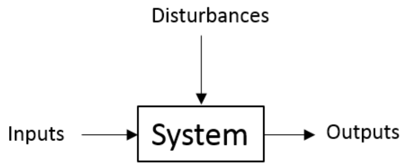


Figure 3: Model of a dynamic system

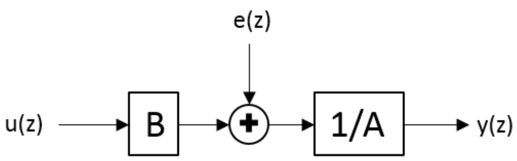
To define the model, the data measured are usually divided in training and validation sets. The model parameters are optimised so that it fits the best the training set. It is then applied on validation set to verify its accuracy. The fit value between the estimated and real outputs of the model are computed according to Equation 1.

$$fit = 100(1 - \frac{\|y - y_{estimated}\|}{\|y - mean(y)\|}) \tag{1}$$

3.2 ARX Model

The ARX (Auto Regressive model with eXternal inputs) is a simple linear auto-regressive model. The general block diagram of an ARX is shown in Figure 4 and its Z-transform is described by Equation 2. The components of the Z-transform are the following :

ny : number of system outputs
Y : matrix of current and previous outputs
nu : number of system inputs
nk_i : delay for the input i
U : matrix of current and previous inputs
E : current white noise disturbances (zero mean)
na : order of the polynomials of $A(z)$
A : matrix $ny * ny$ of polynomial coefficients
nb_i : order of the polynomials of $B_i(z) + 1$
B_i : matrix $ny * nu$ of polynomial coefficients



$$A(z)Y(z) = \sum_{i=1}^{nu} B_i(z)z^{-nk_i}U(z) + E(z) \tag{2}$$

Figure 4: ARX model

The time domain equation for the multiple-inputs and single-output ARX model is given by Equation 3. The components of the time domain equation are the following:

$y(t)$: output at time t

na: number of poles of the system

$y(t-i)$: previous outputs, with $i \in [1; na]$

nb: number of zeros plus 1 of the system

nk_i: delay of the input i

$u(t-nk_i-i)$: previous delayed inputs, with $i \in [1; nb-1]$

$e(t)$: white noise disturbance (zero mean) at time t

a_i: coefficients multiplying previous outputs

b_i: coefficients multiplying previous delayed inputs

$$y(t)+a_1y(t-1)+...+a_{na}y(t-na) = \sum_{i=1}^{nu} b_{i,1}u_i(t-nk_i) + b_{i,2}u_i(t-nk_i-1) + ... + b_{i,nb}u_i(t-nk-nb+1)+e(t) \tag{3}$$

The discrete output at time t is thus given by Equation 4.

$$y(t) = \sum_{i=1}^{nu}\sum_{j=1}^{nb} b_{i,j}u(t-nk_i-j+1) + e(t) - \sum_{i=1}^{na} a_iy(t-i) \tag{4}$$

The ARIX is an extension of the ARX that includes an integrator of noise. It is useful when the disturbance is not stationary. Its Z-transform is provided by Equation 5.

$$A(z)Y(z) = \sum_{i=1}^{nu} B_i(z)z^{-nk_i}U(z) + \frac{1}{1-z^{-1}}E(z) \tag{5}$$

The NLARX (NonLinear ARX) is an nonlinear extension of the linear model. In the linear model, the output is basically a weighted sum of previous inputs and outputs. The nonlinear model generalizes and compute the output as a function of previous inputs and outputs, as given by Equation 6. The nonlinearity estimator f is composed of linear and a nonlinear parts and can be for example a wavelet network, a sigmoid network or a neural network. The NLARX gives more flexibility to the estimation than linear models and is thus able to fit more complex functions. However, it takes a consequent longer time to be computed with the System Identification Toolbox.

$$y(t) = f(y(t-1), y(t-2), ..., u(t-1), u(t-2)...) \tag{6}$$

3.3 ARMAX Model

The ARMAX (Auto Regressive Moving Average with eXternal inputs) is an extension of the ARX model which provides more flexibility to model the noise by adding the parameter C. This model is useful when there are dominating disturbances at the inputs of the system. Figure 5 shows the general block diagram of the ARMAX model. Its Z-transform is given by Equation 7. The components changed or added compared to the ARX are the following:

E: matrix of current and previous white noise disturbances (zero mean)

nc: order of the polynomial of $C(z)+1$

C: matrix $ny * 1$ of polynomial coefficients

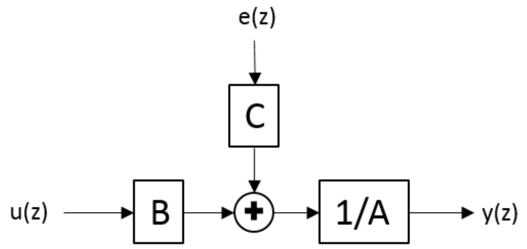


Figure 5: ARMAX model

$$A(z)Y(z) = \sum_{i=1}^{nu} B_i(z) z^{-nk_i} U(z) + C(z)E(z) \qquad (7)$$

The time domain equation for the multiple-inputs and single-output ARMAX model is given by Equation 8. The changed or added components are the following:

nc: number of zero of the disturbance part of the model

$e(t-i)$: previous white noise disturbances, with $i \in [1, nc]$

c_i: coefficients multiplying disturbances, with $i \in [0; nc]$ and c_0 equal to 1

$$y(t) + a_1 y(t-1) + ... + a_{na} y(t-na) = \sum_{i=1}^{nu} b_{i,1} u_i(t-nk_i) + ... + b_{i,nb} u_i(t-nk-nb+1) + c_0 e(t) + ... + c_{nc} e(t-nc) \qquad (8)$$

The discrete output at time t is thus given by Equation 9.

$$y(t) = \sum_{i=1}^{nu} \sum_{j=1}^{nb} b_{i,j} u(t-nk_i-j+1) + \sum_{i=0}^{i=nc} e(t-i) - \sum_{i=1}^{na} a_i y(t-i) \qquad (9)$$

The ARIMAX includes an integrator of the noise to the ARMAX. It is similar to the ARIX for the ARX structure. The Z-transform is given by Equation 10.

$$A(z)Y(z) = \sum_{i=1}^{nu} B_i(z) z^{-nk_i} U(z) + C(z) \frac{1}{1-z^{-1}} E(z) \qquad (10)$$

3.4 BJ Model

The BJ (Box-Jenkins) model provides a complete parametrization of the system as it completely separates the influence of the inputs from the influence of the noise. This model is used generally when there are measurement disturbances that enter later to the system, in the output measurements for example. The block diagram of the model is shown in Figure 6 and its Z-Transform is described by Equation 11. The changed or added components are the following:

nf: order of the polynomials of $F(z)+1$

F_i: matrix $ny * nu$ of polynomial coefficients

nd: order of the polynomial of $D(z)+1$

D: matrix $ny * 1$ of polynomial coefficients

$W(z)$: additional variable used for the transfer to time domain

$V(z)$: additional variable used for the transfer to time domain

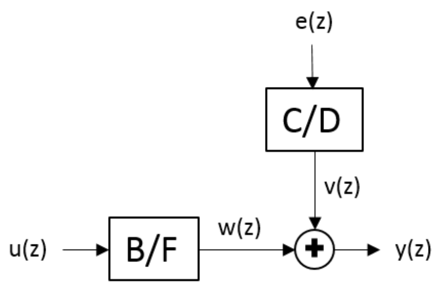


Figure 6: BJ model

$$Y(z) = \sum_{i=1}^{nu} \frac{B_i(z)}{F_i(z)} z^{-nk_i} U(z) + \frac{C(z)}{D(z)} E(z) \qquad (11)$$

The time domain equation for the multiple-inputs and single-output BJ model are given by Equations 12, 13 and 14. The changed or added components are the following:

nb: number of zeros + 1 of the input part of the model

nf: number of poles + 1 of the input part of the model

f_j: coefficients of the polynomial dividing the inputs, with $j \in [1; nf]$

$w_i(t)$: intermediate output of the input i part at time t

$w(t-j)$: previous intermediate outputs of the input i part, with $j \in [1; nf]$

nc: number of zeros + 1 of the noise part of the model

nd: number of poles + 1 of the noise part of the model

d_j: coefficients of the polynomial dividing the noise, with $j \in [1; nd]$

$v(t)$: intermediate output of the noise part at time t

$v(t-j)$: previous intermediate outputs of the noise part, with $j \in [1; nd]$

$$\begin{aligned} w_i(t) + f_{i,1}w_i(t-1) + ... + f_{i,nf}w_i(t-nf) = b_{i,1}u_i(t-nk_i) + ... + b_{i,nb}u_i(t-nk_i-nb+1) \\ v(t) + d_1v(t-1) + ... + d_{nd}v(t-nd) = c_0e(t) - c_1e(t-1) + ... + c_{nc}e(t-nc) \end{aligned} \tag{12}$$

$$\begin{aligned} w_i(t) = \textstyle\sum_{j=1}^{nb} b_{i,j}u_i(t-nk_i-nb+1) - \sum_{j=1}^{nf} f_{i,j}w_i(t-j) \\ v(t) = \textstyle\sum_{j=0}^{nc} c_je(t-j) - \sum_{j=1}^{nd} d_jv(t-j) \end{aligned} \tag{13}$$

Equation 12 gives the time domain equation for intermediate variables w and v. These variables are then determined by Equations 13 and the final discrete output is described by Equation 14.

$$y(t) = \sum_{i=1}^{nu} w_i(t) + v(t) \tag{14}$$

Integration of noise can also be added to the BJ model. The Z-transform of such a system is given by Equation 15.

$$A(z)Y(z) = \sum_{i=1}^{nu} B_i(z)z^{-nk_i}U(z) + \frac{C(z)}{D(z)}\frac{1}{1-z^{-1}}E(z) \tag{15}$$

4 System Identification on Oncilla

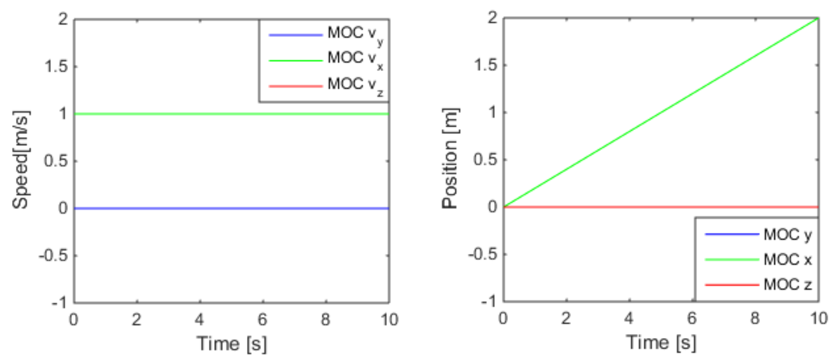
4.1 Acquisition of Data

4.1.1 Oncilla and Motion Capture System Data

The acquisition of data is done by letting the Oncilla move while its sensors record measurements described in Table 1. A Motion Capture System also records the position of three markers placed on the robot. The obtained data is then transferred to Matlab, processed and used for system identification. The Motion Capture System is considered as ground truth, so that the velocity of the robot is the derivative of the recorded positions. That data composes thus the real output of the model. Moreover, as the velocity is the integrate of the acceleration, the acceleration recorded by the sensors of the robot is the basis to estimate its speed and is given as the input to the system. Other measurements are given as the inputs to compensate the imperfections and noise of the accelerometer. The evaluation of the models obtained with system identification is done by comparing the velocity from the Motion Capture System, the real output, and the output of the model, the estimated velocity. Two different experiments are done with Oncilla to acquire data to estimate its velocity. A virtual experiment is also added to provide some very simple data to facilitate the derivation of models.

4.1.2 Zero Acceleration Experiment

A virtual experiment is created and added to the training set of system identification. It should consist of simple data describing a basic behaviour of the robot. A run where the robot walks straight at constant speed is therefore chosen, making acceleration in all directions equal to zero. Moreover, the referential of the robot is aligned to the referential of the Motion Capture System so that the rotation matrix is the identity. This is obviously not realistic as noise and perturbations of the real world do not allow a constant speed. However, it is here easy to find that the integration of acceleration is equal to the velocity and this can suggest a model containing this integration function. Figure 7 shows the speed and the position of the robot in time during the virtual experiment.



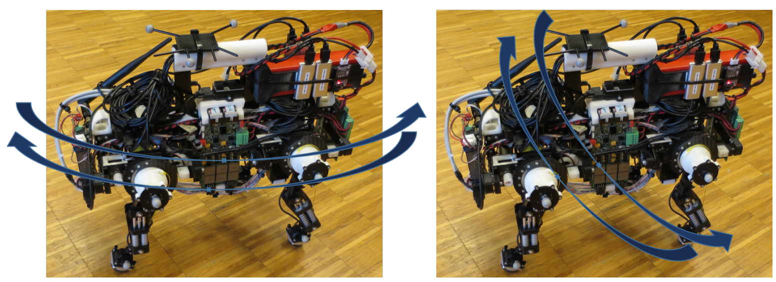
(a) Speed(t)

(b) Position(t)

Figure 7: Virtual experiment where Oncilla walks straight at constant speed

4.1.3 Oncilla Moving in Air Experiment

This second experiment is done with the real robot. It is also simple in order to make the system identification easier. During the experiment, the sensors of Oncilla are activated, but its legs does not move. The robot is carried in the air and swung several times back and forth, then several times from left to right and from right to left. The way the longitudinal respectively lateral oscillations are done is shown in Figure 8.



(a) Longitudinal oscillations

(b) Lateral oscillations

Figure 8: Air moving experiment where Oncilla is carried and swung

This experiment allows to acquire data for a simple behaviour of the robot with a reduced amount of noise as its actuators are not activated and therefore it does not move by itself.

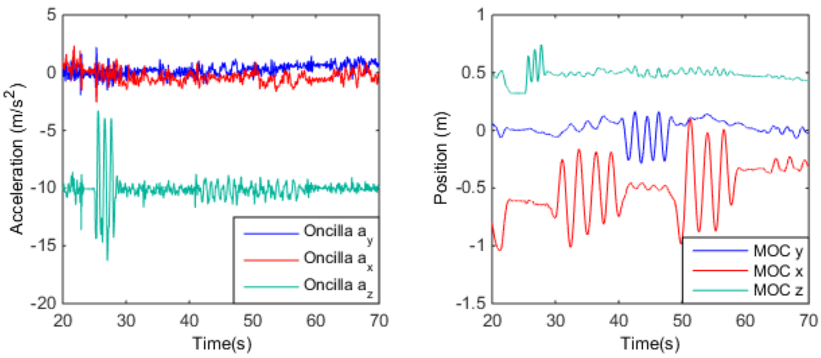
Figures 9a and 9b show respectively the acceleration of Oncilla in its referent frame and its position in the referential of the Motion Capture System. This experiment is composed of four parts: first thirty seconds, the Motion Capture System is calibrated, then longitudinal and lateral oscillations are made successively during approximately ten seconds per type of oscillations. As the interesting data is that recorded when the robot is swung, the following parts of the experiments are extracted:

Run 1: $t \in [30; 40]s$

Run 2: $t \in [41; 48]s$

Run 3: $t \in [50; 58]s$

These data sets needs then to be processed to compute the reference speed of the robot and to reduce the noise.



(a) Acceleration from sensor data (b) Position from MOC data

Figure 9: Acceleration and position of Oncilla collected during the Air Moving experiment

4.1.4 Oncilla Walking Straight Experiment

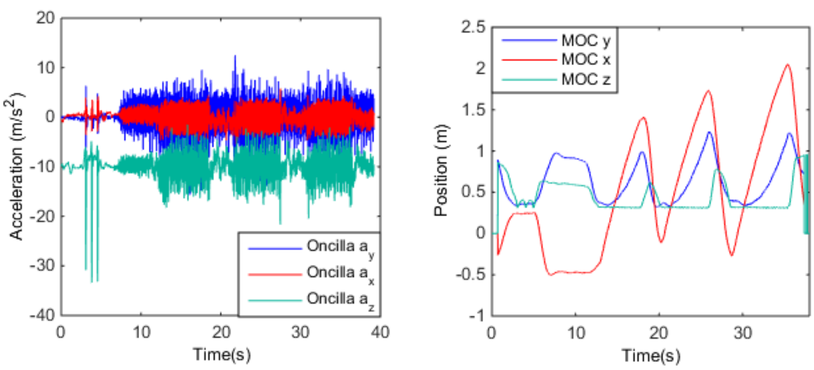
During the last experiment, Oncilla stays on the ground and goes nearly straight by itself. The commanded speed is constant, The measured acceleration and position of Oncilla respectively by its sensors and by the Motion Capture System are shown in Figures 10a and 10b. There is again a calibration phase that lasts during the first ten seconds. After that, the robot walks for approximately five seconds and is then put again to its starting position. This part is repeated three times between $t = 13s$ and $t = 30s$. As the interesting data is now recorded when the robot is walking, the following parts of the experiments are extracted:

Run 1: $t \in [13; 17]s$

Run 2: $t \in [21; 25]s$

Run 3: $t \in [29; 35]s$

An important observation is that the noise on the acceleration present in figure 10a is higher than in 9a. It is due to the interactions of the robot with its environment and to its internal noise of the sensors and actuators. It is thus even more important to reduce this noise without loosing the main information before using the data for system identification.



(a) Acceleration from sensor data (b) Position from MOC data

Figure 10: Acceleration and position of Oncilla collected during a nearly straight walk

4.2 Processing the Data

4.2.1 Computation of the Reference Speed

As said before, the data from the Motion Capture System is considered as ground truth. As the goal is to estimate the velocity of Oncilla, the provided positions in space of the three markers on the robot have to be derived to obtain the velocity. The numerical gradient of the position is computed using Equation 16.

$$\begin{aligned} &v(t) = \frac{x(t+1)-x(t-1)}{2T} \text{ for interior data points} \\ &v(t) = \frac{x(t+1)-x(t)}{T} \text{ or } v(t) = \frac{x(t)-x(t-1)}{T} \text{ for edges data points} \end{aligned} \tag{16}$$

Knowing the velocity in space of two markers allows then to compute the velocity of the robot in its own coordinate system using Figure 11 and Equations 17.

$$\begin{aligned} &\dot{x_r} = \dot{x_1}\cos\theta - \dot{y_1}\sin\theta - \dot{\theta}((x_1 - x_2)\sin\theta + (y_1 - y_2)\cos\theta) \\ &\dot{y_r} = \dot{x_1}\sin\theta + \dot{y_1}\cos\theta + \dot{\theta}((x_1 - x_2)\cos\theta - (y_1 - y_2)\sin\theta) \\ &\theta = arctan(\frac{y_1 - y_2}{x_1 - x_2}) \end{aligned} \tag{17}$$

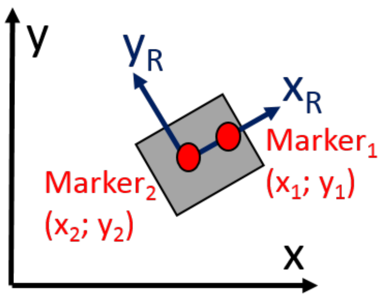
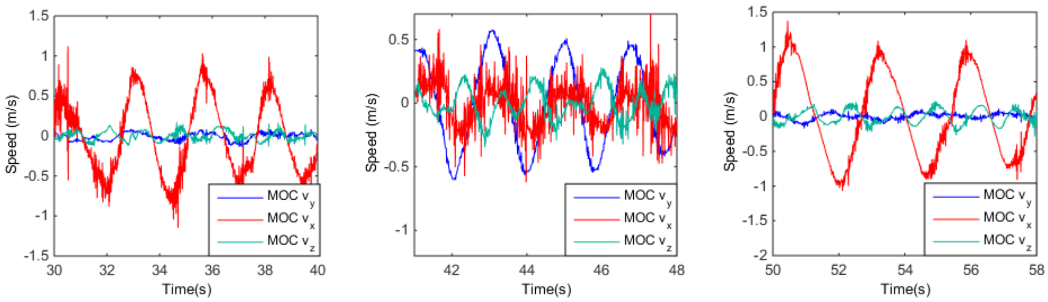


Figure 11: Schema of the robot with its own referent frame $(x_R; y_R)$ and the referential of the MOC $(x; y)$

The velocity obtained for each run of each experiment done with Oncilla are shown in Figure 12 and 13.

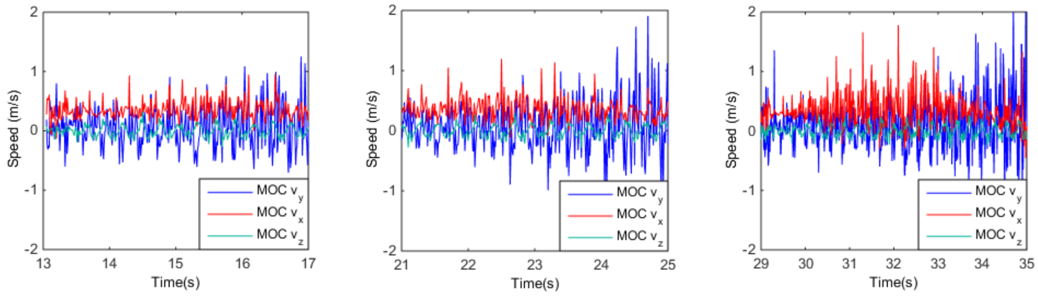


(a) Run 1 (b) Run 2 (c) Run 3

Figure 12: Oncilla velocity from MOC data for the Oncilla Moving in Air experiment: the robot is swung back and forth during runs 1 and 3 and left to right and the opposite during run 2

4.2.2 Choice of Inputs

As the velocity is the integrated acceleration, the models design of the velocity of Oncilla were done using only the information about its acceleration provided by its sensors, which are the raw acceleration and the rotation matrix. Two different inputs were tested for the system identification. The first one contains the raw acceleration in direction x, y and z and the two first lines of the rotation matrix, which gives nine input components. Six components of the rotation matrix are sufficient as the information provided by the last three



(a) Run 1 (b) Run 2 (c) Run 3

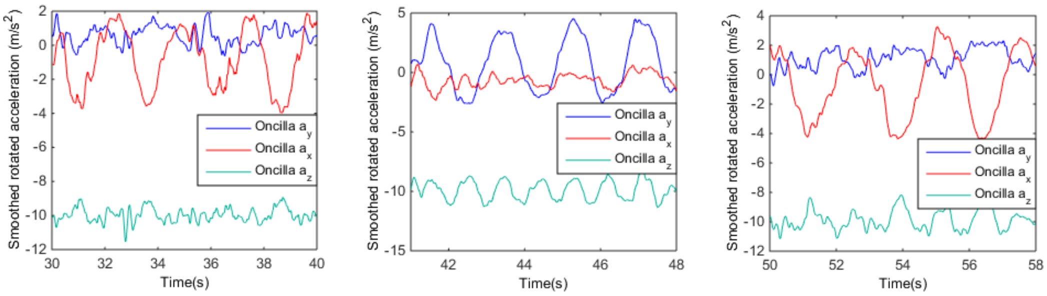
Figure 13: Oncilla velocity from MOC data for the Oncilla Walking Straight experiment

is redundant. In order to reduce the number of inputs, the second set contains the raw acceleration multiplied by the inverse rotation matrix, thus only three input components. This multiplication aligns the acceleration components with the referent frame of the robot. The identification is thus simplified, as there are less inputs and the components of the acceleration are already aligned with the speed from the Motion Capture System, giving as real output.

4.2.3 Reduction of Noise

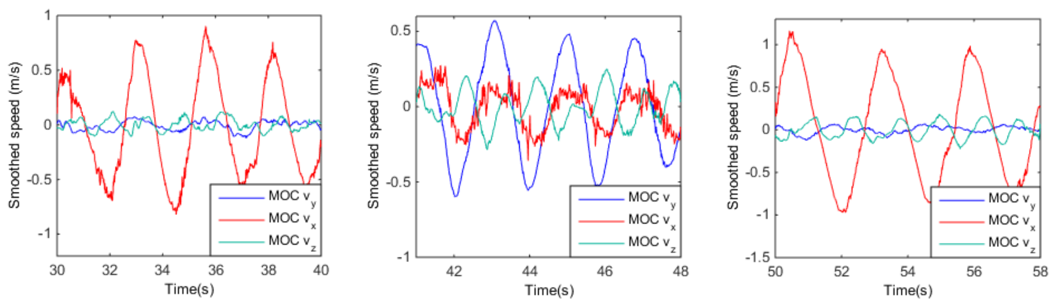
To reduce the noise for the inputs of system identification, the data is smoothed using an moving average filter. The equation of this filter is given by Equation 18, where the span is an odd integer and was chosen equal to 5 for all inputs. Smoothed inputs and outputs are shown in Figures 14, 15, 16 and 17.

$$u(t) = \sum_{i=-0.5(span-1)}^{0.5(span-1)} u(t+i) \tag{18}$$



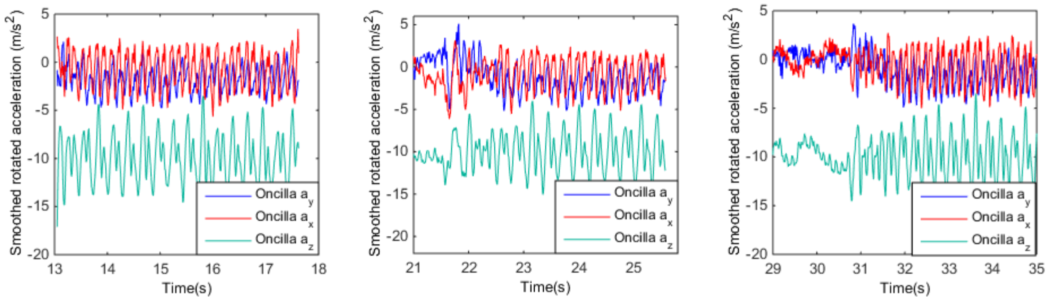
(a) Run 1 (b) Run 2 (c) Run 3

Figure 14: Smoothed and rotated Oncilla acceleration for the Moving in Air experiment



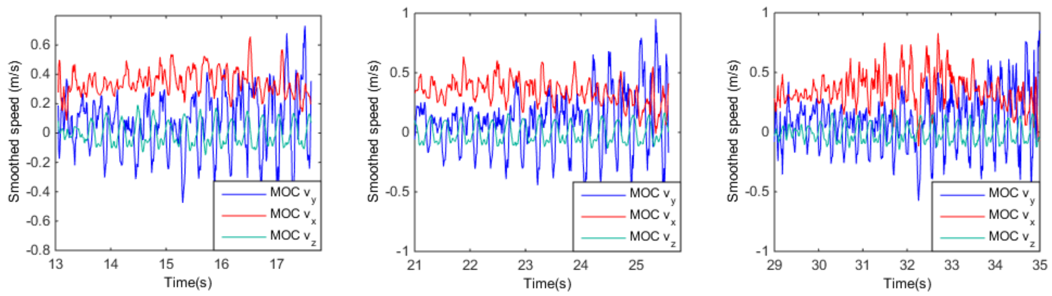
(a) Run 1 (b) Run 2 (c) Run 3

Figure 15: Smoothed Oncilla velocity from MOC data for the Oncilla Moving in Air experiment



(a) Run 1 (b) Run 2 (c) Run 3

Figure 16: Smoothed and rotated Oncilla acceleration for the Walking Straight experiment



(a) Run 1 (b) Run 2 (c) Run 3

Figure 17: Smoothed Oncilla velocity from MOC data for the Oncilla Walking Straight experiment

4.3 Velocity Estimation

4.3.1 Oncilla Moving in Air

The first estimation of the velocities in x, y and z-axis for the Oncilla Moving in Air data is done with the following parameters:

Reference name of the estimation: Moving in Air 1

Model type: ARX, focus on prediction

Model parameters: $na = nb = \begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}$, $nk = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Inputs: rotated acceleration

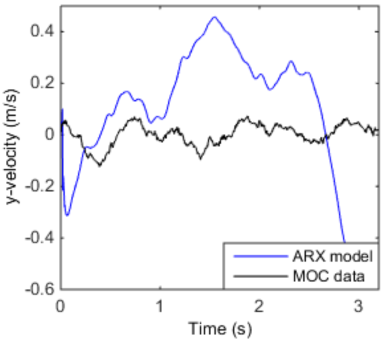
Real output: speed from the Motion Capture System

Training set: runs 1 and 2 of Oncilla in Air Moving experiment

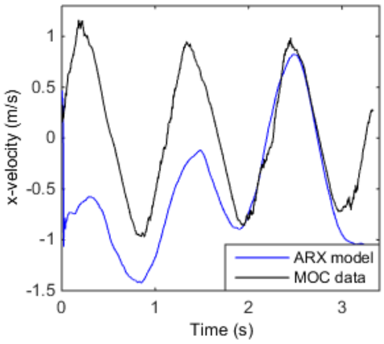
Validation set: run 3 of Oncilla in Air Moving experiment

The estimated outputs of the training set fit well the reference data for the z-axis and for the axis of oscillations, which are respectively x-axis and y-axis for run 1 and 2. The estimated velocity along the third axis has some drift and lots of shape differences from the reference.

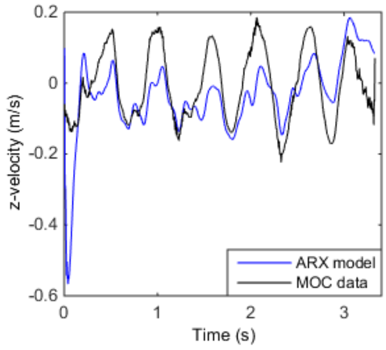
The measured and estimated outputs of the validation set are shown in Figure 18. The fit between the ARX model and the real velocity is negative for all axis. However, for x and z-axis, the model needs some time to initialise itself, then the estimated curve has a global shape similar to the reference curve. The model does not success to approximate the y-velocity as the estimation is totally different from the reference. As for the training set, the model obtained with the system identification is here able to approximate z-axis and the axis along which the robot is swung.



(a) Fit: -889.5



(b) Fit: -26.99



(c) Fit: -3.185

Figure 18: Measured and estimated model outputs for the validation set of the estimation Moving in Air 1

In order to ease the identification of a good model, the Zero Acceleration experiment is added to the training set. The parameters of this second estimation are the following:

Reference name of the estimation: Moving in Air 2

Model type: ARX, focus on prediction

Model parameters: $na = nb = \begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}$, $nk = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Inputs: rotated acceleration

Real output: speed from the Motion Capture System

Training set: Zero Acceleration experiment, runs 1 and 2 of Oncilla in Air Moving experiment

Validation set: run 3 of Oncilla in Air Moving experiment

The resulting outputs are not shown as they are exactly the same as the results of the first simulation. Moreover, the training on the zero acceleration experiment is not accurate as the model output tends to infinity for all axis.

As the identification works better on the oscillations axis, the run 2 of the Oncilla Moving in Air experiment was removed from the training set. Only the Zero Acceleration experiment and the run for which Oncilla is swung along x-axis remain. The parameters of this third estimation are the following:

Reference name of the estimation: Moving in Air 3

Model type: ARX, focus on prediction

Model parameters: $na = nb = \begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}$, $nk = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Inputs: rotated acceleration

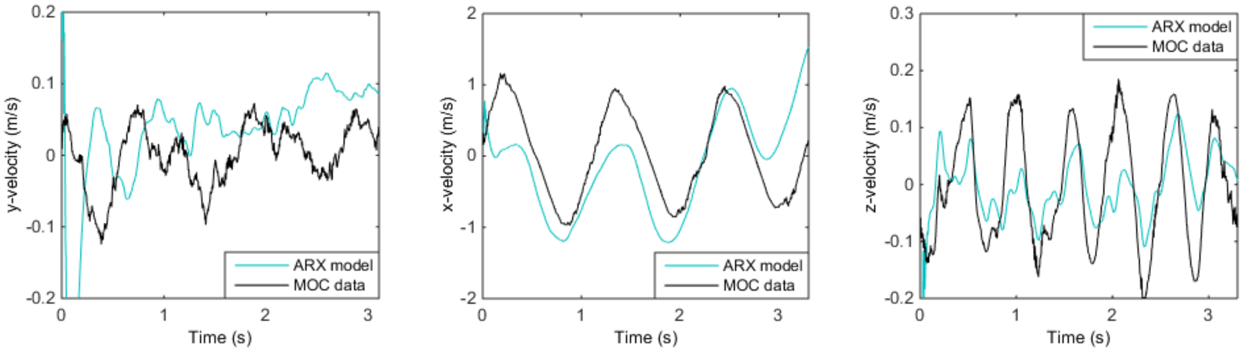
Real output: speed from the Motion Capture System

Training set: Zero Acceleration experiment, run 1 of Oncilla in Air Moving experiment

Validation set: Run 3 of Oncilla in Air Moving experiment

The training on the Zero Acceleration experiment is now more successful as the model is able to define a velocity equal to zero for y-axis, even if the values at the beginning are very high giving a fit equal to infinity for these curves. The model is nevertheless not able to determine the velocity along x and z-axis. The training on the run 1 is also improved compared to previous simulation, as all the fit have positive values and the estimations follow roughly the reference curves for all axis. The fits values stay nevertheless low as they are equal to 38.89%, 50.73% and 31.67% for y, x and z-axis.

The measured and estimated outputs of the validation set are shown in Figure 19. The fit remains low



(a) Fit: -160.9 (b) Fit: 1.014 (c) Fit: 18.79

Figure 19: Measured and estimated model outputs for the validation set of the estimation Moving in Air 3

for all axis, but the values has an considerable improvement compared to previous estimations and are now positive for all axis but y. Visually, estimated velocity along x and z-axis are similar to the reference curve after approximately one second. The estimated velocity fits poorly the reference curve during the complete run. It seems to be shifted along the time axis, as if the estimation has a delay compared to the reference.

Trying to ease the identification of a model, the fourth estimation is made with inputs with a mean equal to zero. The acceleration data are centred by removing the mean along each dimension. The gravity vector was thus removed from the z-axis. This estimation has the following parameters:

Reference name of the estimation: Moving in Air 4

Model type: ARX, focus on prediction

Model parameters: $na = nb = \begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}$, $nk = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

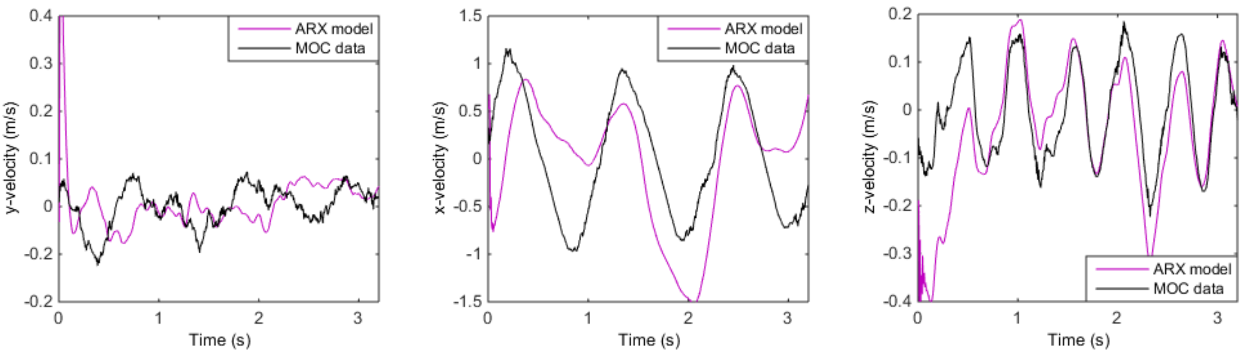
Inputs: rotated acceleration with zero mean

Real output: speed from the Motion Capture System

Training set: Zero Acceleration experiment, run 1 of Oncilla in Air Moving experiment

Validation set: run 3 of Oncilla in Air Moving experiment

The results of the training are improved for the Zero Acceleration experiment as the model is able to detect that y and z-velocity are equal to zero. It takes some time to converge to zero and the values are very high at the beginning, giving a fit equal to minus infinity. The velocity along x-axis is not approximated correctly as it diverges. The results on the run 1 are also better with centred inputs. The shape of the estimated velocity is similar to the reference curve with a fit equal to 45.62%, 74% and 63.61% for y, x and z-axis. Even if the value of the fit is less good for the y-axis than for the other ones, it is now positive and approaches 50%.



(a) Fit: -82.76 (b) Fit: -7.29 (c) Fit: -6.539

Figure 20: Measured and estimated model outputs for the validation set of the estimation Moving in Air 4

The measured and estimated outputs of the validation set are shown in Figure 20. The only improvement on the validation set is the estimation along the y-axis. It remains shifted but its shape is visually closer to the reference one. Its fit remains negative but is improved compared to previous estimation. However, the estimations for x and z-velocity were better before as their fits are again negative.

The results of training sets of all previous estimations are given in Appendix A.

4.3.2 Oncilla Walking Straight

The first estimation of the velocities for the Oncilla Walking Straight experiment is done with the following parameters:

Reference name of the estimation: Walking Straight 1

Model type: ARX, focus on prediction

Model parameters: $na = nb = \begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}, nk = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Inputs: raw acceleration, 6 components of the rotation matrix

Real output: speed from the Motion Capture System

Training set: Zero Acceleration experiment and run 1 of Oncilla Walking Straight experiment

Validation set: run 3 of Oncilla Walking Straight experiment

The main change compared to the estimations made for the Oncilla Moving in Air experiment is the inputs of the system identification : the estimation is made here with the raw acceleration and six components of the rotation matrix.

The results of the estimation on the training set show that the obtained model is able to find a constant velocity in each direction for the Zero Acceleration experiment even if the constant values do not correspond to the values expected. Moreover, the fit is equal to minus infinity for all axis as there are oscillations with high amplitude at the beginning of the simulations. For run 1, the estimated outputs have a global shape similar to the reference, but they do not follow all variations, particularly for x-velocity, which results in fit equal to 50.03%, 28.11% and 62.65% for y, x and z-axis.

The measured and estimated outputs of the validation set are shown in Figure 21. The model needs once again some time to initialise itself and the estimated outputs begin to fit the reference curve after approximately $1.2s$. Even if the best fit value is on the estimation for x-velocity, the model seem to have the same variation as the reference only for positive peaks and stays nearly constant otherwise. However, estimated y and z-velocities have shapes similar to their respective references after the initialisation time. Their bad fit values, respectively -8.531% and -12.27% are possibly due to the first half of the graphs, where the two curves are very different.

To have more data for training the model, the second run of Oncilla Walking Straight experiment is then added to the training set. Moreover, to reduce the number of input parameters, the rotated acceleration is given as input for the system identification. The parameters of the second estimation are thus the following :

Reference name of the estimation: Walking Straight 2

Model type: ARX, focus on prediction

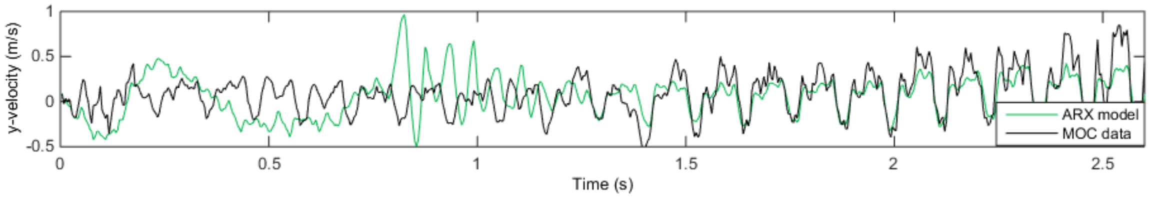
Model parameters: $na = nb = \begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}, nk = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Inputs: rotated acceleration

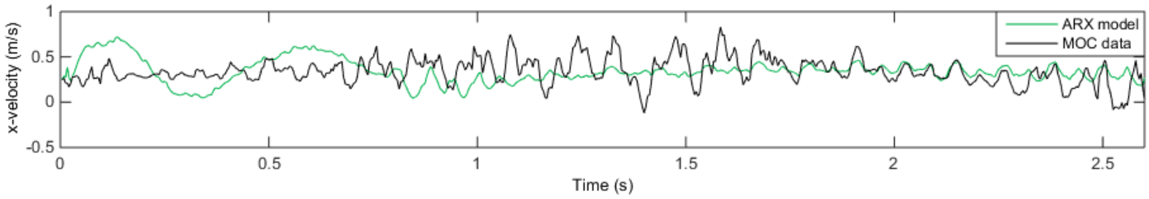
Real output: speed from the Motion Capture System

Training set: Zero Acceleration experiment, runs 1 and 2 of Oncilla Walking Straight experiment

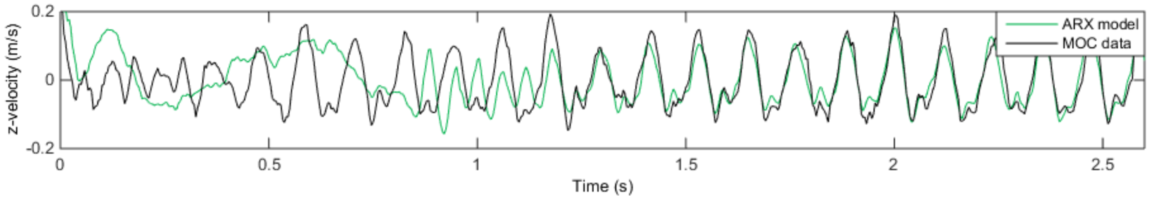
Validation set: run 3 of Oncilla Walking Straight experiment



(a) Fit: -8.531



(b) Fit: -2.812



(c) Fit: -12.27

Figure 21: Measured and estimated model outputs for the validation set of the estimation Walking Straight 1

A non-linear model is also tried to see if a more complex method gives better results. The only change compared to the second estimation is the type of the model so that the parameters for the third estimation are the following :

Reference name of the estimation: Walking Straight 3

Model type: NLARX, wavelet network, focus on prediction

Model parameters: $na = nb = \begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}, nk = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Inputs: rotated acceleration

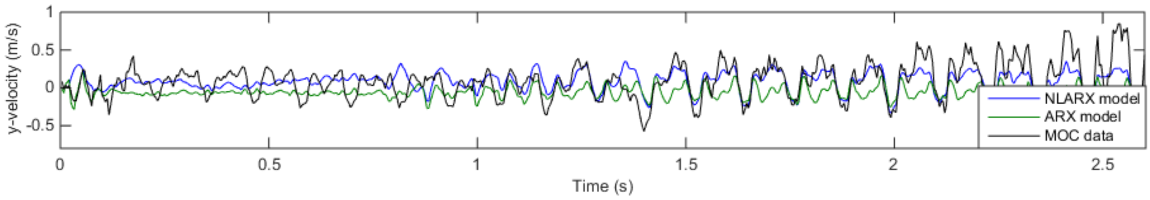
Real output: speed from the Motion Capture System

Training set: Zero Acceleration experiment, runs 1 and 2 of Oncilla Walking Straight experiment

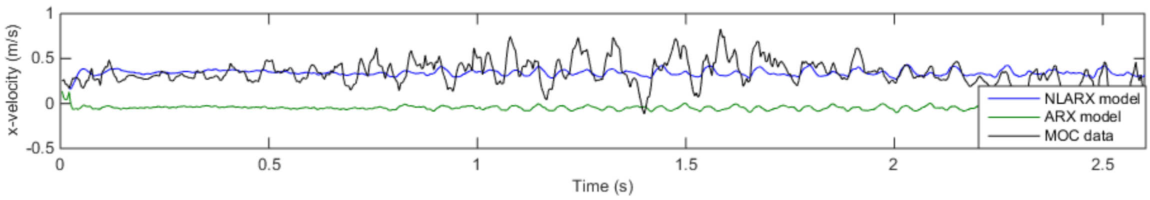
Validation set: run 2 of Oncilla Walking Straight experiment

The results of the training on Zero Acceleration experiment show a similar behaviour for ARX models of estimations Walking Straight 1 and 2. The NLARX model of the estimation Walking Straight 3 is however not able to find constant values for the velocities along the different axis. The estimations are slowly decreasing functions, which fits are also equal to minus infinity. The estimated y and z-velocities of the ARX model on runs 1 and 2 of Oncilla Walking Straight have a mean similar to the mean of the reference curves,but do not follow its peaks : the estimated velocities do not vary much compared to the reference. The estimated output of the x-axis seems to be centred around zero and are almost constant. The fit values for the ARX model on the training set are thus all negative. The results are better for the NLARX model as the fit are equal to 32.77%, 11.07% and 61.03% for y, x and z-axis of run 1 and to 26.79%, 10.57% and 35.31% for y, x and z-axis of run 2. For the two runs, the shape of the estimated y-velocities are similar to the reference, but the estimation does not reproduce all the peaks of the Motion Capture System data. For the x-axis, the estimated outputs have a mean similar to the mean of the reference, but vary only a little in comparison. The estimations of z-velocities are the most similar to the reference curves as they reproduce nearly all the peaks and variations.

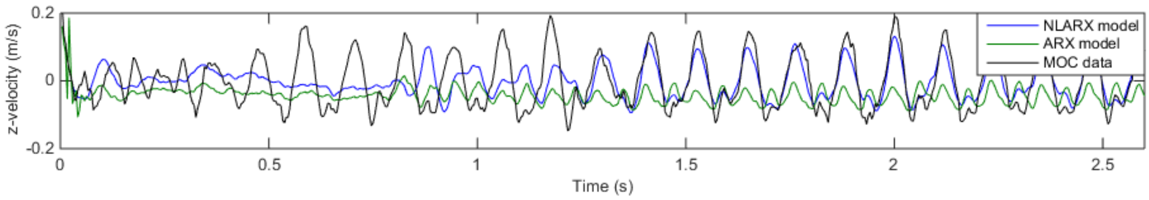
The measured and estimated outputs of the validation set of Walking Straight 2 and 3 are shown in Figure 22. The results of the ARX model on the validation set are similar to those obtained for the training set and fit values are all negative. The results of the non-linear model are more promising. As for the estimation Oncilla Walking Straight 1, the model takes some time to initialise itself. It has a shape similar to the reference for y and z-axis, even if the estimated peaks are smaller than the real ones. Along x-axis, the estimated output does not vary and fits only the smaller variations of the reference. However, fit values are very low for all dimensions.



(a) Fit ARX: -10.93
Fit NLARX: 19.96



(b) Fit ARX: -177.6
Fit NLARX: 3.434



(c) Fit ARX: -9.477
Fit NLARX: 22.56

Figure 22: Measured and estimated model outputs for the validation set of the estimations Walking Straight 2 and 3

The results of training sets of all previous estimations are given in Appendix A.

4.4 Discussion of the Results

The models found by the system identification for the experiments of this section have globally low values of fit between estimated and measured outputs. These values are often higher for velocities along z-axis than for the other directions. This was expected as the height of the robot has small periodical variations with a mean constant for the two experiments done. The estimation would be more difficult if the robot was walking on rough terrain.

For all velocities, estimations are better on training sets than on validation set, which is expected. However, in this case, the system identification is also not able to find a model approximating well the training set as the fit on the training can be expected to be at least equal to 60% for all directions.

During the training, the virtual outputs of the virtual Zero Acceleration experiment are often not estimated correctly and diverge. As the goal of this virtual experiment is to help the system identification to find a model containing an integrator, the expected estimated outputs are constant velocities. Only ARX models on

Oncilla Walking Straight experiments succeed to provide such outputs. However, adding this experiment to the training set seems generally to increase the accuracy of the obtained model. Using very simple experiments as this one seems to be a possibly efficient way to improve the identification of models.

During the Oncilla Moving in Air experiment, the movement of the robot consists of oscillations along two directions, x and z or y and z depending on the run. Training the model on such data results in good estimations in the validation set for the directions along which the robot also oscillating, but poor ones for the others, possibly because they are shifted in time.

For this section, most computed models were ARX as they are the simpler and faster to train. The only used NLARX model gives better fit values than the equivalent ARX models in this particular case. However, even if the estimation of the velocity of Walking Straight 2 is far from the measured one, the estimated outputs of Walking Straight estimations 1 and 3 fit the reference in a similar way. The fit of the ARX is probably lower because it oscillates with higher amplitude and frequency than the NLARX during the first part of the experiment. For the next sections, linear models are preferred to nonlinear ones because the difference of accuracy is not sufficient to compensate the amount of time needed to compute models such as NLARX.

Due to the noise present in data collected from Oncilla and from the Motion Capture System, smoothing the data is necessary so that the model is able to fit the data. However, this can suppress or delay some variations. The computation of the reference speed as done for Oncilla Moving in Air and Oncilla Walking Straight experiment adds singularities to the velocity in the system of coordinates of the robot. Due to the noise on the data, the distance between two markers is sometimes near to zero and cause incorrect and often high values of the angle θ and of its derivative. Moreover, a mistake was made during this computation with the data of the Motion Capture System for the estimations of this section. The velocity was rescaled instead of being rotated, which perhaps induces more singularities. The found models are correct, but they estimate the rescaled velocity instead of the velocity in Oncilla system of coordinates.

Some improvement should be made for the identification of models. First, the system identification is tried on noisy functions to verify that it is possible to find models fitting well the reference. It is then tried on simulations of Pleurobot as the data provided by simulations are less noisy than the measured ones. Moreover, more complex models such as ARMAX and BJ should be tested to see if they are more accurate and the parameters should be tuned to be optimal. Nonlinear models are not tested here as the time of computation is very important due to the complexity of the training. Furthermore, it is possible that the system identification needs a bigger amount of data to be accurate, which means collecting longer runs where the robot is moving and/or providing more kinds of input data. Longer runs should also permit to verify if models need some time to initialise themselves and, if it is true, to leave some time for the initialisation without having low values of fit, as they seem to be greatly influenced by an imprecise estimation at the beginning.

5 System Identification on Noisy Functions

5.1 Functions and Parameters Tested

In this section, the identification of models is tested on different linear and trigonometric functions. A simple linear model is used, with the following parameters:

Model type: ARX, focus on prediction/simulation

Model parameters: $na = 4$, $nb = 4$, $nk = 1$

Inputs: derivative of functions, with additional random noise $0.5 \cdot \mathcal{N}(0,1)$

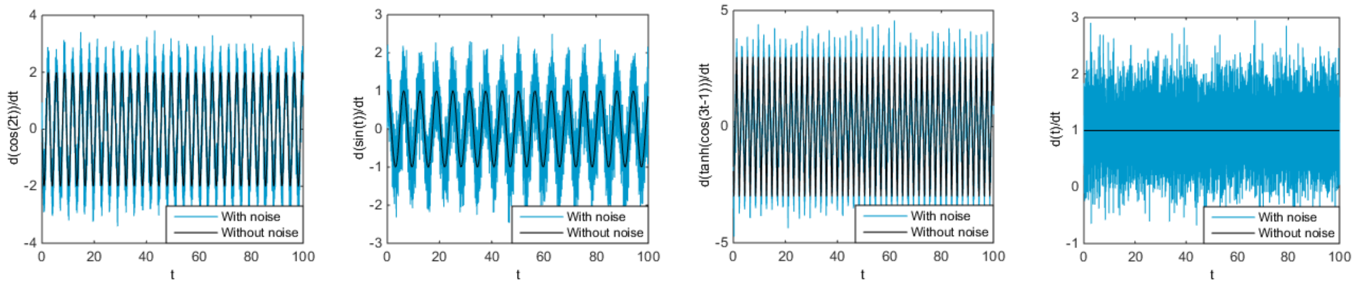
Real output: functions

Training set: $\cos(2t)$, $\sin(t)$, $\tanh(\cos(3t+1))$, t

Validation set: $2\sin(t-1)$, $t^2\sin(t)$, $\tanh(4\cos(t))$

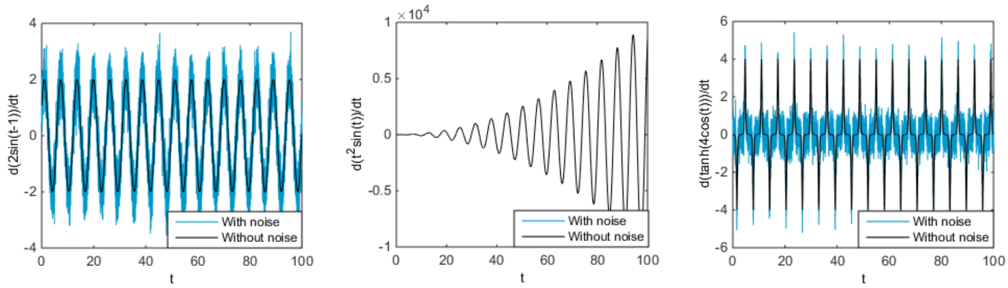
The focus of this ARX model is defined once on prediction and once on simulation to make a comparison between these two types.

The different derivative of functions used as input for training and for validation are shown respectively by Figures 23 and 24. The derivative of the functions are first computed for 10000 steps of size 0.1. A standard gaussian noise of maximum amplitude equal to 0.5 is then superposed on the derivatives.



(a) $\frac{d}{dt}\left(\cos\left(2t\right)\right)$ (b) $\frac{d}{dt}\left(\sin\left(t\right)\right)$ (c) $\frac{d}{dt}\left(\tanh\left(\cos\left(3t+1\right)\right)\right)$ (d) $\frac{d}{dt}\left(t\right)$

Figure 23: Derivative of different functions used with additional random noise $0.5 \cdot \mathcal{N}(0,1)$ for the training set



(a) $\frac{d}{dt}\left(2\sin\left(t-1\right)\right)$ (b) $\frac{d}{dt}\left(t^2\sin\left(t\right)\right)$ (c) $\frac{d}{dt}\left(\tanh 4\cos t\right)$

Figure 24: Derivative of different functions with additional random noise $0.5 \cdot \mathcal{N}(0,1)$ for the validation set

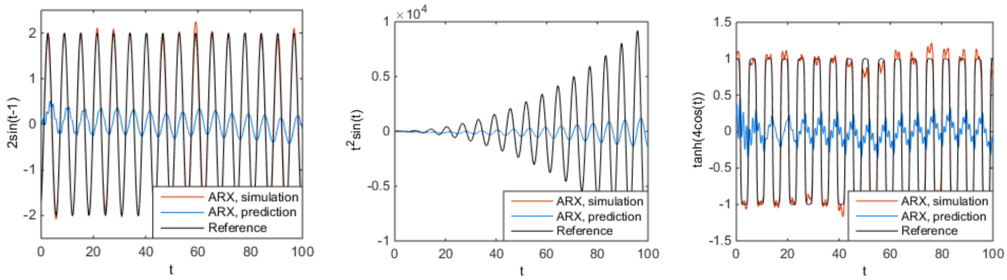
5.2 Results

The results on training set confirm that the system identification is able to find a model fitting noisy data. However, the parameters of the model have to be defined such that it focuses on simulation to have a good fit between the measured and estimated outputs. Indeed, fit values for focus on simulation are over 75% as the values for the same model, except that it focuses on prediction, are equal to 70% for the linear function and under 1% for all trigonometric functions.

The results of the training set are given in Appendix B.

Figure 25 shows the measured and estimated outputs on the validation set. The results are similar to those obtained for the training set. Fit values are over 90% for the model focusing on the simulation and under 3% for the one focusing on prediction.

Despite significant noise added to the inputs, the system identification is able to find a model fitting well the real output. The focus of all models used in next sections are selected on simulation. In fact, the model does not need to be accurate for prediction of future outputs, as the goal is to know the instantaneous velocity of a robot to adapt its behaviour in function of the events perturbing its velocity.



(a) Fit simulation: 94.03
Fit prediction: 2.232

(b) Fit simulation: 99.77
Fit prediction: 2.746

(c) Fit simulation: 90.01
Fit prediction: 1.503

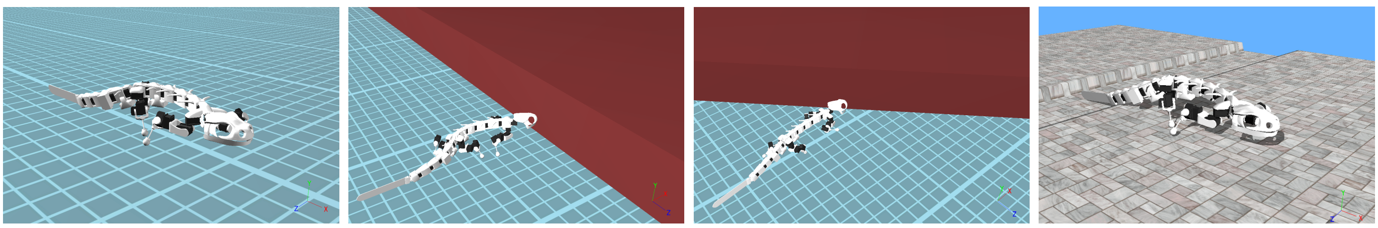
Figure 25: Measured and estimated outputs for the validation set of the estimation on functions with additional random noise

6 System Identification on Pleurobot Simulations

6.1 Acquisition of Data

Pleurobot is estimated in Webots, a development environment used to model, program and make sub-microscopic simulations of mobile robots. As in reality, the robot receives informations about its state from different sensors. Its position is recorded in the simulation by a GPS. As for Oncilla, the acceleration recorded by its sensors is the basis to estimate the velocity of Pleurobot. Other measurements are also given as input to compensate the imperfections and noises of the accelerometer. The evaluation of the models obtained with system identification is also done by comparing the velocity calculated from GPS data and the estimated velocity which is the output of the model.

Four different simulations of Pleurobot are done to acquire data to estimate its velocity. Each experiment contains five different runs. During Experiment 1, Pleurobot walks straight with a variable command speed, which includes stopping and starting again the walk. For Experiment 2, the robot smashes a wall several times per run. The wall is removed after a few seconds so that it starts walking again. Experiment 3 is similar to Experiment 2, but Pleurobot hits the wall at the angle and not perpendicularly as before. During Experiment 4, the robot gets stuck while it is walking over a stair. Figure 26 shows examples of experiments.



(a) Experiment 1

(b) Experiment 2

(c) Experiment 3

(d) Experiment 4

Figure 26: Examples of behaviour of Pleurobot for different experiments

6.2 Processing the Data

6.2.1 Computation of the Reference Speed

The reference speed is computed using the position of the robot provided by the GPS. The numerical gradient of the position is computed using Equation 16 to obtain the speed along each axis of the estimated world. These speeds are then transferred from the world frame of reference to the one of the robot using the rotation matrix and Equation 19.

$$\vec{v}_{robot} = R^{-1}\vec{v} \tag{19}$$

Trying to make the identification of models easier, only one output is considered for system identification. The models are thus expected to fit only one kind of data and the results should be improved. For Pleurobot, the velocity along x-axis is the most interesting and is chosen as the output.

6.2.2 Choice of Inputs

As previous system identification on Oncilla, the accelerations along all axis are given as inputs. Moreover, the gravity vector has been removed of the z-component by subtracting $9.81m/s^2$. Several sets of inputs were tested to obtain good models. The ground reaction forces of each foot of the robot were first added to the acceleration, which gave fifteen inputs. In order to reduce this number, the forces were replaced by the angles between them given by Equation 20 as they give informations about the shape of the friction cone and about the instantaneous slipping of the robot. However, the fit between the estimated and measured outputs was not improved compared to the previous set. The forces along x and y-axis were then replaced by their norm, while the vertical force was kept. The resulting models improved the fit compared to the set of inputs containing the accelerations and all the forces. Adding the frequency reference also improved the results.

$$\begin{aligned} \alpha &= \tan \tfrac{F_y}{F_x} \\ \beta &= \tan \tfrac{F_z}{\sqrt{F_x^2+F_y^2}} \end{aligned} \tag{20}$$

The final set of inputs is thus composed of these twelve elements in the following order:

- x-acceleration;
- y-acceleration;
- z-acceleration;
- norm of horizontal forces on foot 1;
- vertical force on foot 1;
- norm of horizontal forces on foot 2;
- vertical force on foot 2;
- norm of horizontal forces on foot 3;
- vertical force on foot 3;
- norm of horizontal forces on foot 4;
- vertical force on foot 4;
- frequency reference.

6.3 Velocity Estimation

6.3.1 Models Trained on One Experiment

In order to see if training on few experiments is sufficient to obtain a good estimation of the velocity for most cases, the first model is trained only on Experiment 1 and is tested on the others. Furthermore, a BJ model is selected to have a more flexible design. The parameters of the model are the following:

Reference name of the estimation: BJ model on Experiment 1

Model type: BJ, focus on simulation with noise integration and zero initialisation

$$\text{Model parameters: } \begin{cases} nb = \begin{pmatrix} 1 & 1 & 1 & 5 & 3 & 5 & 3 & 5 & 3 & 5 & 3 & 1 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 2 & 2 & 2 & 4 & 1 & 4 & 1 & 4 & 1 & 4 & 1 & 4 \end{pmatrix} \\ nk = \begin{pmatrix} 1 & 1 & 1 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 3 \end{pmatrix} \end{cases}$$

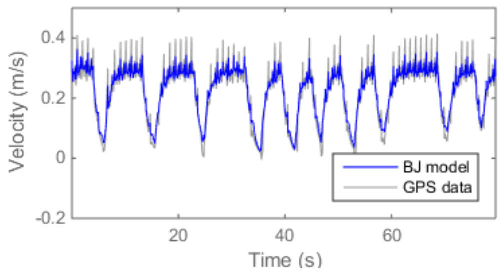
Training set: Experiment 1 (runs 1, 2, 3, 4)

$$\text{Validation set: } \begin{cases} \text{Experiment 1 (run 5)} \\ \text{Experiment 2 (run 5)} \\ \text{Experiment 3 (runs 1, 2, 3, 4, 5)} \\ \text{Experiment 4 (runs 1, 2, 3, 4, 5)} \end{cases}$$

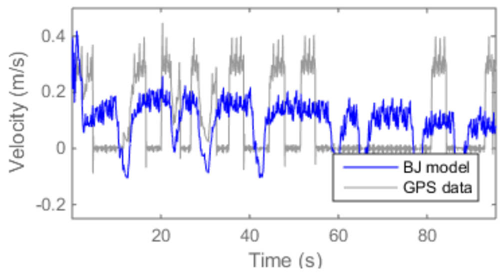
The parameters are hand-tuned to get a good trade-off between the fits of the different runs of the validation set. The parameters concerning the acceleration are selected to encourage an integrating behaviour. The found models are also more accurate with different parameters for horizontal and vertical forces.

Figure 27 shows some of the measured and estimated outputs of the validation set. The remaining results are shown in Appendix C.

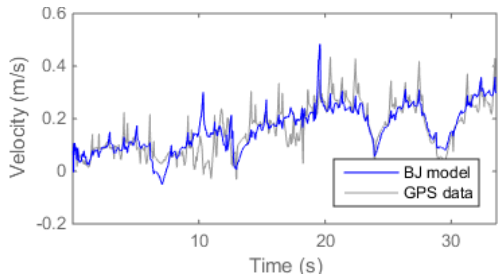
As it was expected due to the training set components, the resulting estimation on the fifth run of Experiment 1 follows well the measured output with a fit equal to 75.75%. The fits obtained on runs of Experiment 3 are respectively equal to 42.41, 23.31%, 30.67%, −2.446% and 37.04% with estimations similar to the reference rather at the end of the runs. However, the model is not able to follow the sudden stop and start of the robot during Experiment 2 with a low fit of −1.204%. Moreover, results on Experiment 4 are also not convincing, as the fit are all negative and equal respectively to −9.672, −11.47%, −24.25%, −13.89% and −9.586%. The model does not reproduce the positive peaks and replaces the negative peaks by positive ones.



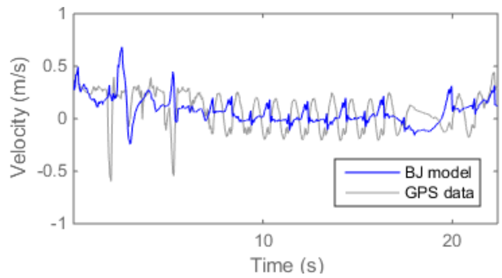
(a) Experiment 1, run 5
Fit: 75.75



(b) Experiment 2, run 5
Fit: −1.204



(c) Experiment 3, run 1
Fit: 42.41



(d) Experiment 4, run 1
Fit: −9.672

Figure 27: Measured and estimated outputs for part of validation set of the estimation BJ model on Experiment 1

Training the model on only one experiment allows thus to correctly estimate different data if they have some similarities. In fact, during Experiments 1 and 3, the velocity of the robot is varied through the command given to the robot or is only slowed by obstacles, as the robot is deviated and restarts walking after the deviation. Changes of speed are thus smooth, contrary to the two other experiments during which variations are sudden and frequent.

To improve the identification of models, the training is then done on Experiments 1 and 2, so that more kind of data are provided.

6.3.2 Models Trained on Two Experiments

In order to check if a simple model can be accurate in fitting the measured outputs, an ARX is first trained on two experiments. The following parameters are used for system identification:

Reference name of the estimation: ARX model on Experiments 1 and 2

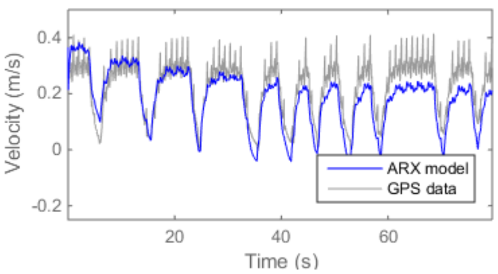
Model type: ARX, focus on simulation with estimated initialisation

$$\text{Model parameters: } \begin{cases} na = 4 \\ nb = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 4 \end{pmatrix} \\ nk = \begin{pmatrix} 1 & 1 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 3 \end{pmatrix} \end{cases}$$

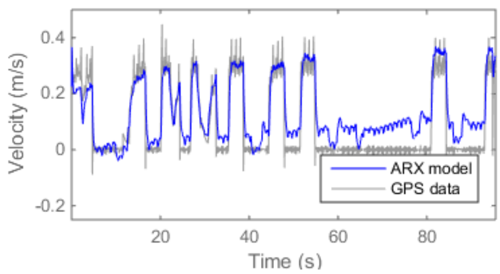
$$\text{Training set: } \begin{cases} \text{Experiment 1 (runs 1, 2, 3, 4)} \\ \text{Experiment 2 (runs 1, 2, 3, 4)} \end{cases}$$

$$\text{Validation set: } \begin{cases} \text{Experiment 1 (run 5)} \\ \text{Experiment 2 (run 5)} \\ \text{Experiment 3 (runs 1, 2, 3, 4, 5)} \\ \text{Experiment 4 (runs 1, 2, 3, 4, 5)} \end{cases}$$

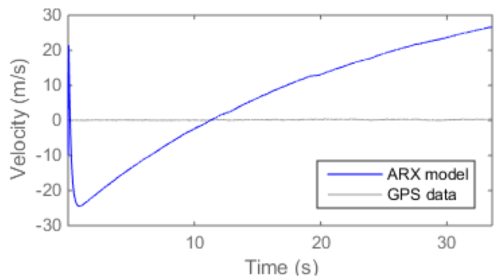
These parameters are also hand-tuned, but changing them allows only to vary the results on the validation runs of the two experiments trained. The results on Experiments 3 and 4 are similar for other parameters.



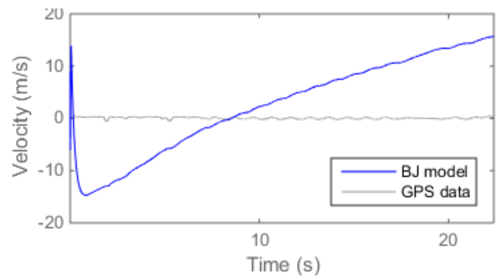
(a) Experiment 1, run 5
Fit: 32.26



(b) Experiment 2, run 5
Fit: 47.86



(c) Experiment 3, run 1
Fit: −17170



(d) Experiment 4, run 1
Fit: −5322

Figure 28: Measured and estimated outputs for part of validation set of the estimation ARX model on Experiments 1 and 2

Figure 28 shows the estimated and measured outputs of one run of each experiment. The estimated outputs follow well the measured one for Experiments 1 and 2. Even if fit values, respectively 32.26% and 47.86% are not very high, the curves are similar. For Experiment 1, the estimation has the same general shape than the real velocity except for short and high peaks that are not reproduced. However, the estimation allows to have a good idea of the longitudinal velocity of the robot. For Experiment 2, the estimated output allows to see when the robot is stopped and restarts even if the estimation is not always equal to zero when the robot cannot go forward because of obstacles.

Despite the similarities between Experiments 1 and 3, respectively 2 and 4, the model is not able to fit accurately any runs of the two last experiments. The estimated velocity always behaves in the same way: the velocity increases, then decreases suddenly and increases again more slowly. Moreover, fit values are all lower than -1000%.

An ARMAX model is then tested in order to see if adding a parameter to modulate the error is sufficient to obtain good estimations on all experiments. The design of the model is done such that the reference frequency component used is the present one multiplied by a constant. Moreover, the actual forces are also used, such that only the acceleration is delayed. Moreover, the number of past errors taken in account does not seem to have some influence to improve the estimation on run of Experiments 3 and 4. It is thus fixed to zero, which gives a model equivalent to an ARX. The parameters used are therefore the following:

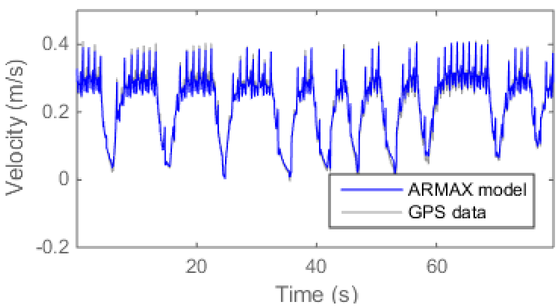
Reference name of the estimation: ARMAX model on Experiments 1 and 2

Model type: ARMAX, focus on simulation

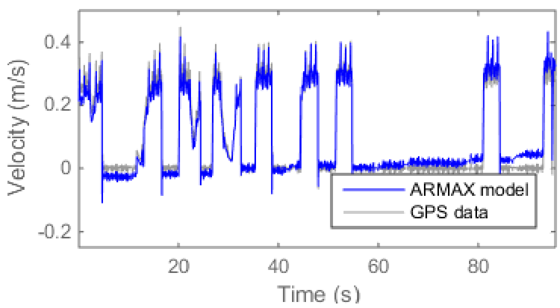
$$\text{Model parameters: } \begin{cases} na = 1 \\ nb = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 \end{pmatrix} \\ nc = 0 \\ nk = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{cases}$$

$$\text{Training set: } \begin{cases} \text{Experiment 1 (runs 1, 2, 3, 4)} \\ \text{Experiment 2 (runs 1, 2, 3, 4)} \end{cases}$$

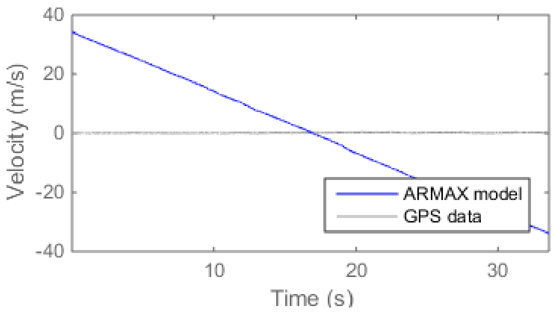
$$\text{Validation set: } \begin{cases} \text{Experiment 1 (run 5)} \\ \text{Experiment 2 (run 5)} \\ \text{Experiment 3 (runs 1, 2, 3, 4, 5)} \\ \text{Experiment 4 (runs 1, 2, 3, 4, 5)} \end{cases}$$



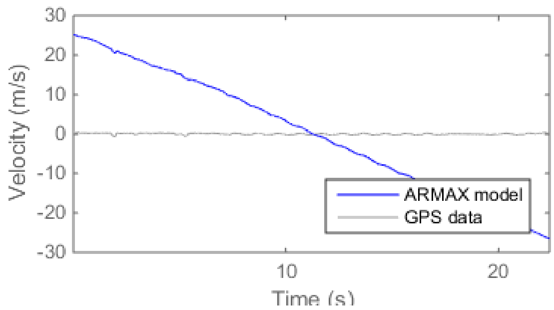
(a) Experiment 1, run 5
Fit: 89.4



(b) Experiment 2, run 5
Fit: 82.77



(c) Experiment 3, run 1
Fit: -21380



(d) Experiment 4, run 1
Fit: -8708

Figure 29: Measured and estimated outputs for part of validation set of the estimation ARMAX model on Experiments 1 and 2

The results of ARMAX model on Experiments 1 and 2 are shown in Figure 29. Estimations are very good for runs 5 of the two first experiments with respectively 89.4% and 82.77%. Visually, the estimated and measured outputs are nearly superposed. However, the model is again not able to estimate correctly Experiments 3 and 4. The estimated velocity is constantly decreasing with fit values as bad as those of ARX model on Experiments 1 and 2.

As BJ model on Experiment 1 gives better results on Experiments 3 and 4 than ARX and ARMAX models of this section, the same BJ model is also trained on two experiments. The parameters are the following:

Reference name of the estimation: BJ model 1 on Experiments 1 and 2

Model type: BJ, focus on simulation with noise integration and zero initialisation

$$\text{Model parameters: } \begin{cases} nb = \begin{pmatrix} 1 & 1 & 1 & 5 & 3 & 5 & 3 & 5 & 3 & 5 & 3 & 1 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 2 & 2 & 2 & 4 & 1 & 4 & 1 & 4 & 1 & 4 & 1 & 4 \end{pmatrix} \\ nk = \begin{pmatrix} 1 & 1 & 1 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 3 \end{pmatrix} \end{cases}$$

$$\text{Training set: } \begin{cases} \text{Experiment 1 (runs 1, 2, 3, 4)} \\ \text{Experiment 2 (runs 1, 2, 3, 4)} \end{cases}$$

$$\text{Validation set: } \begin{cases} \text{Experiment 1 (run 5)} \\ \text{Experiment 2 (run 5)} \\ \text{Experiment 3 (runs 1, 2, 3, 4, 5)} \\ \text{Experiment 4 (runs 1, 2, 3, 4, 5)} \end{cases}$$

Figure 30 shows the estimated and measured velocities for several runs of the validation set. The estimated velocity for the run 5 of Experiment 1 is similar for models trained on one and two experiment. However, the estimation for the run 5 of Experiment 2 is consequently improved with a fit value equal to 47.45%. Even if the estimated velocity goes under zero at the moment when the robot is stopped and tends to be non-zero when the robot stays blocked for a time, the estimation seems sufficient to know if Pleurobot is blocked by an object or not. In contrast, the estimated velocities on Experiment 3 do not correspond any more to the measured ones, as the fit values are respectively −48.67, −44.77%, −144.2%, 39.34% and −61.65% for the different runs. Visually, the curves are not similar to each other except for the fourth run. This run has also a better fit value than the others, which can be explained by the fact that the velocity for this run looks like those of Experiment 3. The estimations for Experiment 4 have also negative values equal respectively to −8.857%, −17.16%, −40.56%, −32.36% and −42.47%. However, the estimated velocities are similar to the measured ones, but they are often shifted up or down, which can explain the negative fit values.

The BJ model is then hand-tuned trying to improve the estimation on Experiment 3. The delay of all forces is chosen equal to two time steps and the denominator of the polynomial function characterising the acceleration has an order equal to one. The new model has the following parameters:

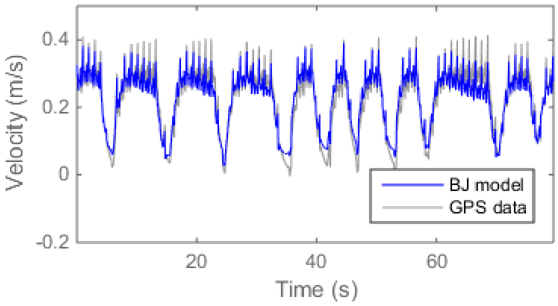
Reference name of the estimation: BJ model 2 on Experiments 1 and 2

Model type: BJ, focus on simulation with noise integration and zero initialisation

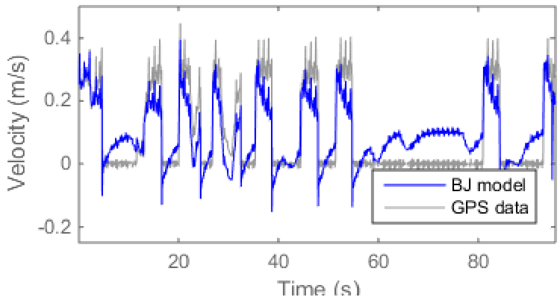
$$\text{Model parameters: } \begin{cases} nb = \begin{pmatrix} 1 & 1 & 1 & 5 & 3 & 5 & 3 & 5 & 3 & 5 & 3 & 1 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 1 & 1 & 1 & 4 & 1 & 4 & 1 & 4 & 1 & 4 & 1 & 4 \end{pmatrix} \\ nk = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 \end{pmatrix} \end{cases}$$

$$\text{Training set: } \begin{cases} \text{Experiment 1 (runs 1, 2, 3, 4)} \\ \text{Experiment 2 (runs 1, 2, 3, 4)} \end{cases}$$

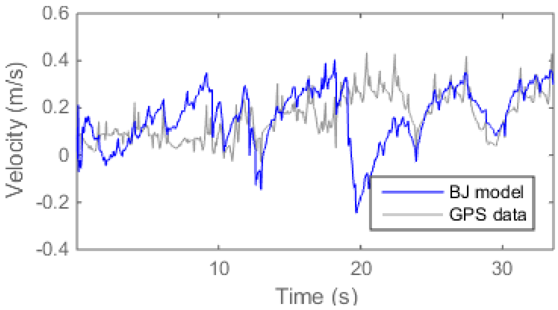
$$\text{Validation set: } \begin{cases} \text{Experiment 1 (run 5)} \\ \text{Experiment 2 (run 5)} \\ \text{Experiment 3 (runs 1, 2, 3, 4, 5)} \\ \text{Experiment 4 (runs 1, 2, 3, 4, 5)} \end{cases}$$



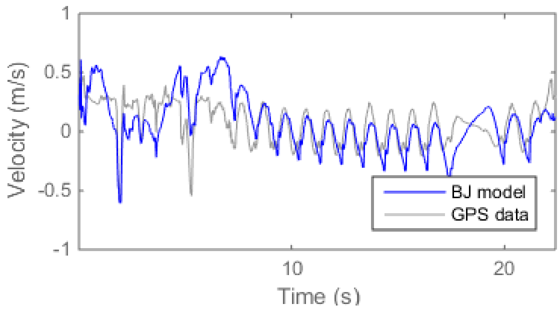
(a) Experiment 1, run 5
Fit: 71.61



(b) Experiment 2, run 5
Fit: 47.45

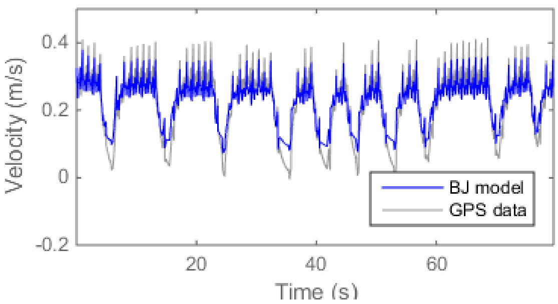


(c) Experiment 3, run 1
Fit: −48.67

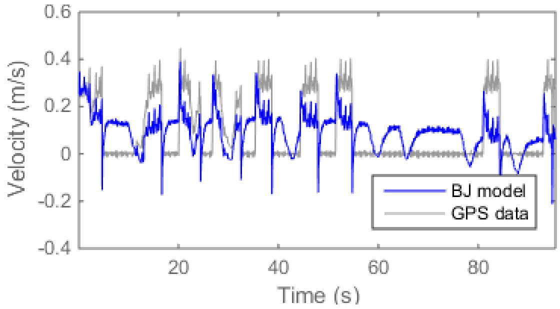


(d) Experiment 4, run 1
Fit: −8.857

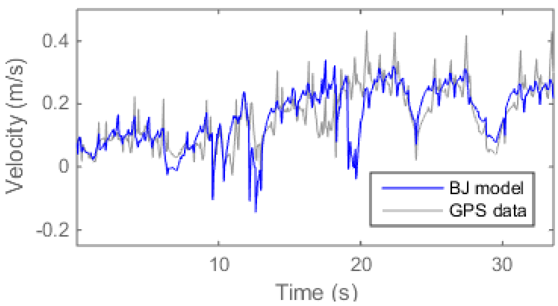
Figure 30: Measured and estimated outputs for part of validation set of the estimation BJ model 1 on Experiments 1 and 2



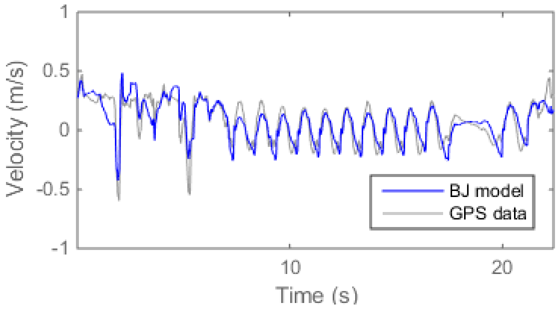
(a) Experiment 1, run 5
Fit: 62.87



(b) Experiment 2, run 5
Fit: 16.13



(c) Experiment 3, run 1
Fit: 28.48



(d) Experiment 4, run 1
Fit: 47.09

Figure 31: Measured and estimated outputs for part of validation set of the estimation BJ model 2 on Experiments 1 and 2

Figure 31 shows the results of this second BJ model on a run of each experiment. Changes made on the model deteriorate the estimations on Experiments 1 and 2 and improve those on Experiments 3 and 4. For the first experiment, the estimated output follows well the peaks of the measured one, but is not reduced as much when the real velocity of the robot is decreased. The fit value is equal 62.87%. The estimation on run 5 of Experiment 2 has a fit value equal to 16.13%. It has positive or negative peaks when the real output increases or decreases suddenly, but stabilises itself at an intermediate speed otherwise. The negative peaks can be useful to determine when the robot is blocked, but the rest of the estimation does not give informations about the velocity of the robot. The estimated outputs of Experiment 3 do not reproduce all variations of the real velocity. Sometimes, a positive sudden rise of measured is replaced by a sudden diminution of the estimated one. The estimations on this experiment are thus not reliable. They have also lower fit values, equal to 28.48%, 20.75%, 21.55%, 16.26% and 29.41%. Results on Experiment 4 are more promising, with fits equal to 47.09%, 44.19%, 29.43%, 46.37% and 34.89%. Visually, the estimated output is similar to the measured one, except that peaks of estimated velocity have often a lower amplitude than real ones.

Next models are trained on three experiments in order to obtain an homogeneous quality of estimation over all runs. Two runs of Experiment 3 are then added to the training set.

The remaining estimations of all models presented above are shown in Appendix C.

6.3.3 Models Trained on Three Experiments

In order to see if training on more kinds of data improves the results obtained with ARMAX model, the identification of model is first computed with an ARMAX trained on Experiments 1, 2 and 3. Its parameters are the following:

Reference name of the estimation: ARMAX model on Experiments 1, 2 and 3

Model type: ARMAX, focus on simulation with regularisation ($\lambda = 10^{-4}$)

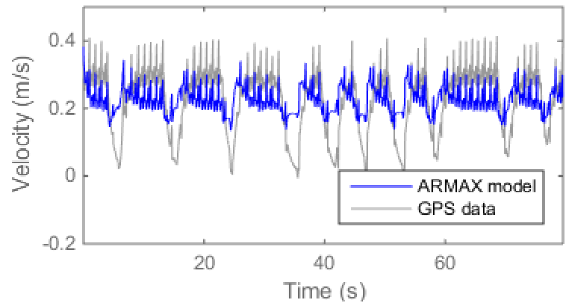
$$\text{Model parameters:} \begin{cases} na = 1 \\ nb = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 \end{pmatrix} \\ nc = 2 \\ nk = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{cases}$$

$$\text{Training set:} \begin{cases} \text{Experiment 1 (runs 1, 2, 3, 4)} \\ \text{Experiment 2 (runs 1, 2, 3, 4)} \\ \text{Experiment 3 (runs 1, 2)} \end{cases}$$

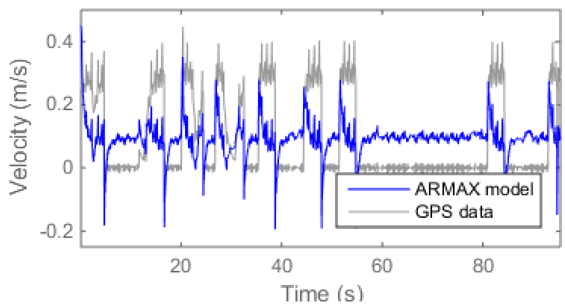
$$\text{Validation set:} \begin{cases} \text{Experiment 1 (run 5)} \\ \text{Experiment 2 (run 5)} \\ \text{Experiment 3 (runs 3, 4, 5)} \\ \text{Experiment 4 (runs 1, 2, 3, 4, 5)} \end{cases}$$

The parameters are hand-tuned to obtain a satisfying trade-off between the quality of the estimations on all runs of the validation set. The delay is null for all parameters except the frequency reference. They are also uniform for horizontal and vertical forces. Moreover, regularisation is added to the model, penalising the model flexibility. It improves the estimations on Experiments 3 and 4, but the quality of those on Experiments 1 and 2 is reduced a lot.

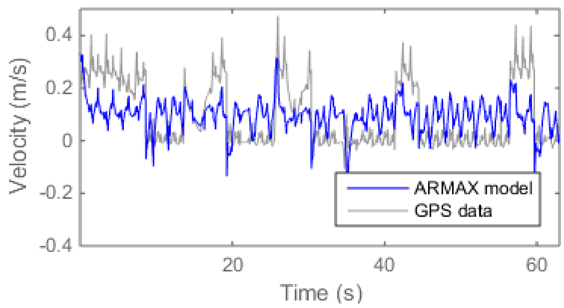
Some of the results of estimations with this model are shown in Figure 32. The estimated output of run 5 of Experiment 1 has a lower fit value than all other estimations computed before, equal to 12.91%. It reproduces the increasing peaks of velocity, but has a small variance compared to the real output. The variance of the estimation on Experiment 2 is also small except at the moments when the robot is suddenly stopped, inducing high negative peaks on the estimated output. The fit is also not satisfying with a value of −7.126%. Compared to ARMAX model on Experiments 1 and 2, the model trained on three experiments is more accurate for Experiments 3 and 4. However, fit values stay relatively low as they are equal to −53.28%, 7.759% and −19.35% for Experiment 3 and 26.63%, 20.47%, 11.15%, 24.68% and 19.3% for Experiment 4. Moreover, estimated outputs of Experiment 3 do not follow the reference. For the last experiment, the estimations have generally the same shape as the reference, but do not reproduce all peaks and small variations of velocity.



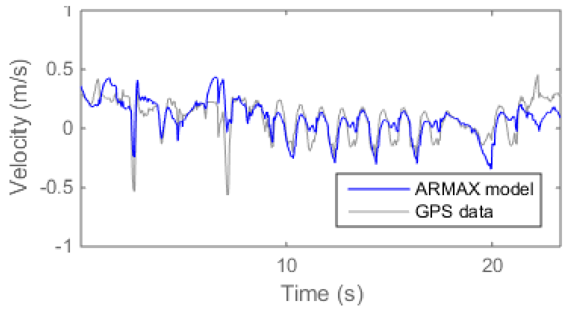
(a) Experiment 1, run 5
Fit: 12.91



(b) Experiment 2, run 5
Fit: −7.126



(c) Experiment 3, run 4
Fit: 7.759



(d) Experiment 4, run 4
Fit: 24.68

Figure 32: Measured and estimated outputs for part of validation set of the estimation ARMAX model on Experiments 1, 2 and 3

A BJ model with the same parameters as BJ model on Experiment 1 and BJ model 1 on Experiments 1 and 2 is tested. The difference with the previous ones is that the integration of noise is not activated when training on three experiments. The parameters of the models are the following:

Reference name of the estimation: BJ model on Experiments 1, 2 and 3

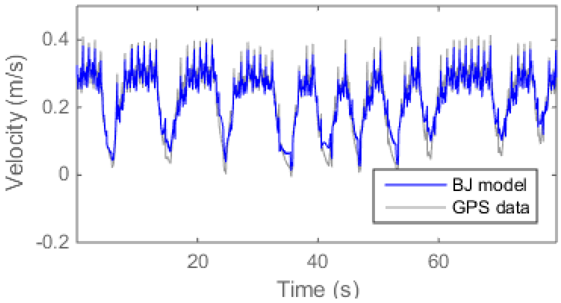
Model type: BJ, focus on simulation with zero initialisation

Model parameters: $\begin{cases} nb = \begin{pmatrix} 1 & 1 & 1 & 5 & 3 & 5 & 3 & 5 & 3 & 5 & 3 & 1 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 1 & 1 & 1 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \end{pmatrix} \\ nk = \begin{pmatrix} 1 & 1 & 1 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 3 \end{pmatrix} \end{cases}$

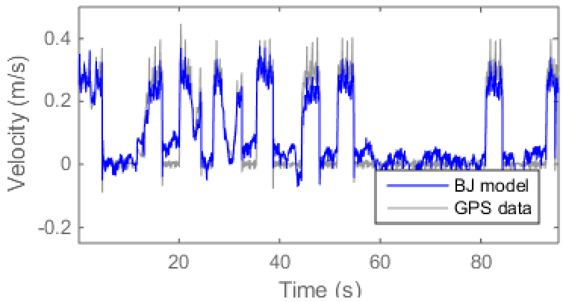
Training set: $\begin{cases} \text{Experiment 1 (runs 1, 2, 3, 4)} \\ \text{Experiment 2 (runs 1, 2, 3, 4)} \\ \text{Experiment 3 (runs 1, 2)} \end{cases}$

Validation set: $\begin{cases} \text{Experiment 1 (run 5)} \\ \text{Experiment 2 (run 5)} \\ \text{Experiment 3 (runs 3, 4, 5)} \\ \text{Experiment 4 (runs 1, 2, 3, 4, 5)} \end{cases}$

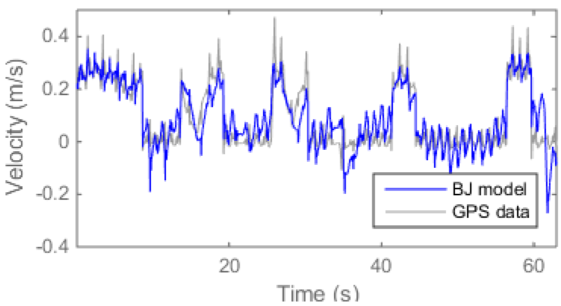
Figure 33 shows the results on some runs. Estimations on two first experiment are very good with fit values respectively equal to 73.88% and 71.57%. The resulting estimated output on the first experiment is similar to those obtained with Bj models trained on less experiments. In contrary, the improvement on Experiment 2 is important, as the remaining difference between estimated and measured outputs is the presence of small oscillations around on the estimation when the real output varies only little. Estimations on Experiment 3 gives better results than the BJ model 1 trained with two experiments as fit values of the validation set are equal to 17.39%, 49.94% and 17.39%. However, those of BJ model 2 fit better the measured velocity, except for run 4. Despite oscillations on the estimated output when the measured one is null, the estimation follows well



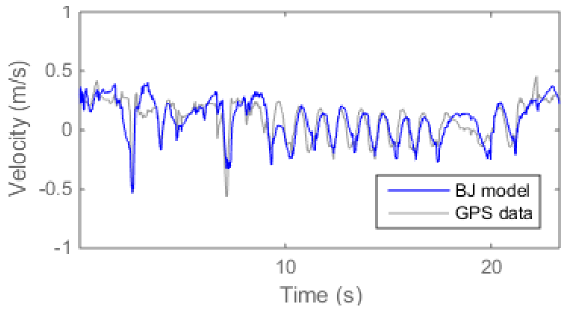
(a) Experiment 1, run 5
Fit: 73.88



(b) Experiment 2, run 5
Fit: 71.57



(c) Experiment 3, run 4
Fit: 49.94



(d) Experiment 4, run 4
Fit: 30.71

Figure 33: Measured and estimated outputs for part of validation set of the estimation BJ model on Experiments 1, 2 and 3

the reference. For the other runs, the general shape of the two curves is the same, but some peaks are removed or added by the estimation, which makes it not reliable. Even if fit values of estimations on Experiment 4 equal to 34.81%, 21.58%, 14.2%, 30.71% and 27.03% are lower than those of BJ model 2 trained on two experiment, the estimated output is similar to the measured one for all runs. In fact, most of variations present in the references are reproduced by the estimations even if they are not perfectly fitted.

The remaining estimations of the two models trained on three experiments are also shown in Appendix C.

6.4 Discussion of the Results

Models obtained with system identification on Pleurobot simulations are more accurate than those found on Oncilla. This improvement can be due to several elements. First, the system identification on Pleurobot is computed with more different experiments, more and longer runs compared to Oncilla. Second, the data collected in simulation can be considered as perfect data. Even if noise is added to the simulation, it is predictable compared to real noise collected with Oncilla data. There is no need to smooth inputs and outputs before using them and the only operation which has to be made before using system identification is the rotation of the velocity given by the GPS. Preprocessing the data does not modify it a lot, so that no or only few imprecisions are added compared to the collected ones. Third, the models are trained to fit only one output which made the identification easier than the one on the velocities along all axis done for Oncilla data. Finally, different sets of parameters are tested for ARX, ARMAX and BJ models. They all focus on simulation, which is fundamental according to the results obtained on noisy functions. Moreover, the integration of noise and the regularisation is added when it is necessary to improve the estimations. Additionally, na, nb, nc, nd, nf and nk matrices are adapted with different values in function of the inputs and different combinations are tested to find a good trade-off for estimations on all experiments.

ARX and ARMAX models trained on one or two experiments used in this section give good results only for experiments on which they were trained. A good example of this case is the ARMAX model on Experiments

1 and 2, which is the model estimating the best runs 5 of the first two experiments. In other cases, even if the fit values are sometimes not really high, the estimations follow visually well the references. However, for other experiments, the estimations give higher and linearly increasing or decreasing velocities, which do not correspond to real ones. The ARMAX model trained on three experiments needs regularisation to obtain satisfying results. However, adding regularisation reduces the variance of the estimation, so that the estimated outputs on the two first experiments do not correspond to the real ones. Moreover, the results obtained on Experiments 3 and 4 are not better than those obtained with BJ models. ARX and ARMAX models are thus not sufficient to obtain good estimations on various experiments.

The system identification with BJ models gives the most promising results for the Pleurobot simulations. The model does not need to be trained on all kinds of experiments collected to give good estimation for each of them. The training set containing only runs of Experiment 1 is sufficient to obtain good fitting between estimated and measured output of two experiments. Moreover, the estimations of Experiment 3 obtained with this model are the best found among all the ones tested in this section. To obtain satisfying fit for Experiments 2 and 4, it is necessary to make the training at least on two different experiment, as the first two ones. The difference of quality between the estimations is not really consequent. Training the model on more experiment is not more costly than training it on two in this case, so that it might be worth doing it to obtain the improvement, even if it is small. Noise Integration is necessary when the model is trained on one or two experiments, but the obtained estimations are better without it when training on three experiments. Additionally, the parameters of the models do not need to be changed to obtain good results when increasing the number of training experiment. However, some changes can improve the estimations on some experiments as it deteriorates the estimations on others. A trade-off has thus to be made.

To improve the identification of models, tests could be made using training sets with different combinations of experiments and runs than those used in this section. More kinds of experiments could also be provided in order to see if the system identification is always accurate.

The BJ models found in this section are then trained on Oncilla data to see if the parameters are good for training on real data and on an other robot. The inputs and output for system identification are selected to be similar to those used for Pleurobot estimations.

7 Improved System Identification on Oncilla

7.1 Objectives of the Improved System Identification

The goal of this section is to exploit results found during system identification on Pleurobot simulations to improve the identification of models on Oncilla. New data is thus acquired, inputs and outputs similar to those of Pleurobot are used for Oncilla and the parameters of the models trained are based on those of the estimations that worked best with Pleurobot.

7.2 Acquisition of Data

As for previous experiments, the acquisition of data is done by recording sensors measurements and the position registered by the Motion Capture System. The ground truth is also the velocity computed from Motion Capture System data and inputs given to system identification are the data collected by Oncilla.

In order to improve the identification of models compared to the previous experiment with Oncilla, only two long runs are done, during which Oncilla walks on the whole area covered by the Motion Capture System. Runs 1 and 2 last respectively 220 and 160 seconds. The commanded speed and direction of the robot are varied during the experiments.

7.3 Processing the Data

7.3.1 Computation of the Reference Speed

The positions recorded by the Motion Capture System have to be derived to obtain the velocity of the robot. It is done as in the previous experiments, using the numerical gradient described by Equation 16.

In order to avoid the presence of singularities mentioned in section 4, the change of coordinate system from Motion Capture System to Oncilla is done without computing any angles, but using scalar products between vectors. This method is described by Equation 21, where $\vec{x_R}$ and $\vec{y_R}$ are computed using the positions of two markers recorded by the Motion Capture System.

$$\begin{aligned} \vec{v} &= v_{xr}\vec{x_R} + v_{yr}\vec{y_R} \\ v_{xr} &= \vec{v} \cdot \frac{\vec{x_R}}{\|\vec{x_R}\|} \\ v_{yr} &= \vec{v} \cdot \frac{\vec{y_R}}{\|\vec{y_R}\|} \end{aligned} \tag{21}$$

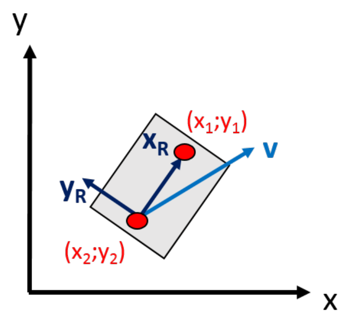


Figure 34: Schema of the robot with necessary vectors to compute the change of coordinate

As for section 6, the most interesting velocity for the robot used is given as output for system identification. For Oncilla, the lateral velocity v_{yr} is chosen as output.

7.3.2 Choice of Inputs

The set of inputs for improved system identification on Oncilla is composed of the same twelve elements as for system identification on Pleurobot simulations. They are the following:

- x-acceleration;
- y-acceleration;
- z-acceleration;
- norm of horizontal forces on foot 1;
- vertical force on foot 1;
- norm of horizontal forces on foot 2;
- vertical force on foot 2;
- norm of horizontal forces on foot 3;
- vertical force on foot 3;
- norm of horizontal forces on foot 4;
- vertical force on foot 4;
- frequency reference.

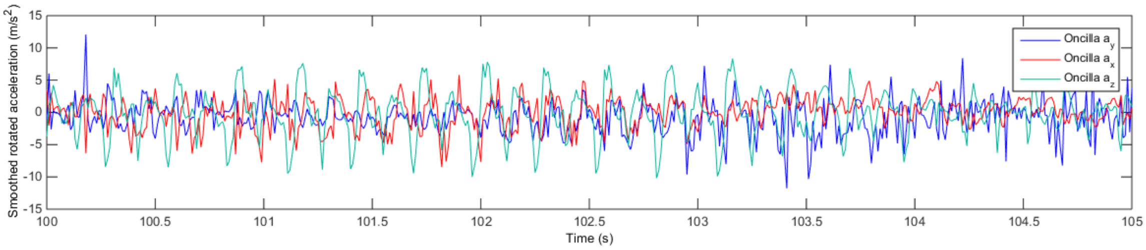
A second set of inputs containing a supplementary element provided by measurement on Oncilla is also tried. The following input is added to the previous set:

- rotation reference.

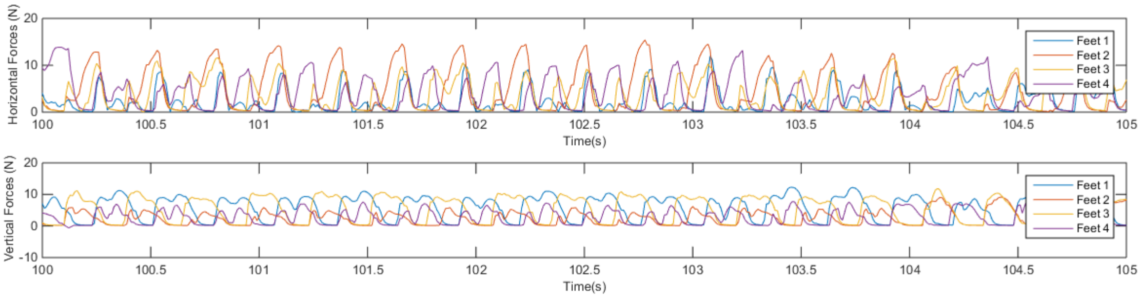
7.3.3 Reduction of Noise

The noise on inputs and outputs of system identification is reduced using the moving average filter described by Equation 18, as for previous experiments with Oncilla.

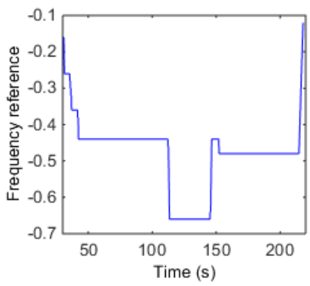
Figure 35 shows a part of inputs and output of Experiment 1 of improved Oncilla data for system identification. Those of Experiment 2 are given in Appendix D.



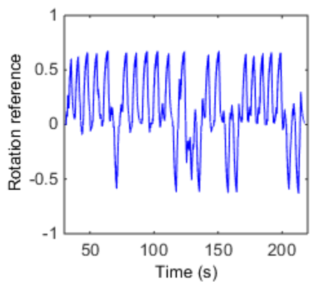
(a) Accelerations



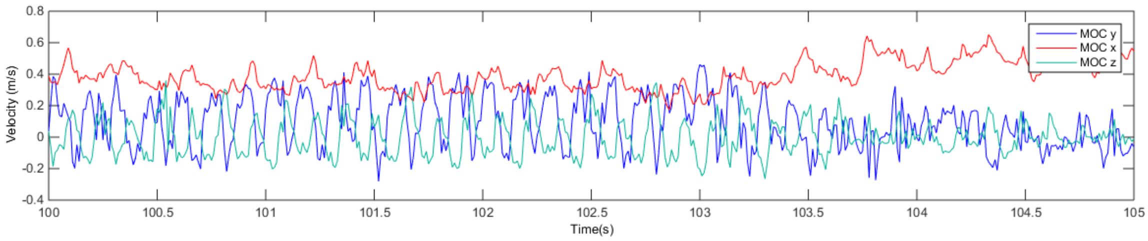
(b) Forces



(c) Frequency Reference



(d) Rotation Reference



(e) Velocity from MOC

Figure 35: Parts of inputs and output of Experiment 1 for Improved System Identification on Oncilla

7.4 Velocity Estimation

The first model trained on new Oncilla data is BJ model on Experiments 1, 2 and 3 from Pleurobot simulations. It is tested with 12 and 13 inputs. The parameters concerning the rotation reference are selected to be the same as those concerning the frequency reference. The parameters of the model are the following:

Reference name of the estimation: BJ model 1 on improved Oncilla data

Model type: BJ, focus on simulation with zero initialisation

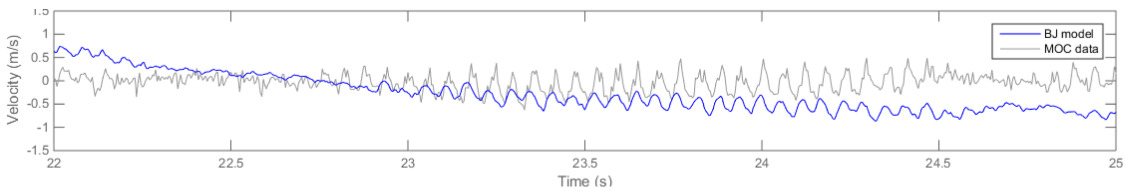
Model parameters (12 inputs): $\begin{cases} nb = \begin{pmatrix} 1 & 1 & 1 & 5 & 3 & 5 & 3 & 5 & 3 & 5 & 3 & 1 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 1 & 1 & 1 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \end{pmatrix} \\ nk = \begin{pmatrix} 1 & 1 & 1 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 3 \end{pmatrix} \end{cases}$

Model parameters (13 inputs): $\begin{cases} nb = \begin{pmatrix} 1 & 1 & 1 & 5 & 3 & 5 & 3 & 5 & 3 & 5 & 3 & 1 & 1 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 1 & 1 & 1 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 3 \end{pmatrix} \\ nk = \begin{pmatrix} 1 & 1 & 1 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 3 & 3 \end{pmatrix} \end{cases}$

Training set: run 1

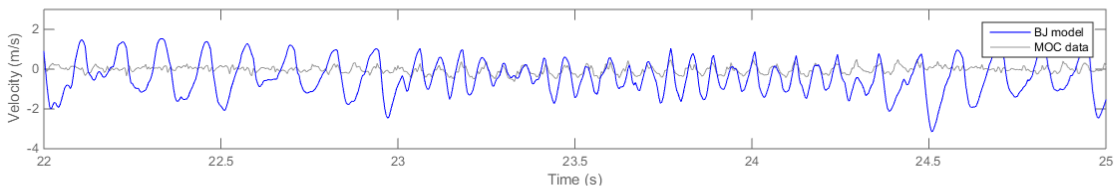
Validation set: run 2

Figures 36a and 36b shows the estimated and measured output for a part of run 2 with 12, respectively 13, inputs. The estimation with 12 inputs has a shape similar to the one of the reference when the variation of velocity are big enough. Otherwise, the curve does not oscillate as the reference. Moreover, the estimated output is drifted compared to the measured one. The estimation with 13 inputs is done without noise integration, as the resulting curve was nearly constant during the run. Without noise integration, the estimated output has a bigger amplitude than the measured one and does not follow its variations.



(a) 12 inputs, with Noise Integration

Fit: -196



(b) 13 inputs, without Noise Integration

Fit: -510

Figure 36: Measured and estimated outputs for the validation run of the estimation BJ model 1 on improved Oncilla data

Results with the BJ model 1 are not convincing as they give negative fit values and do not follow visually the reference. The ARMAX model on Experiments 1, 2 and 3 of Pleurobot simulations is thus tested on Oncilla data with the following parameters:

Reference name of the estimation: ARMAX model on improved Oncilla data

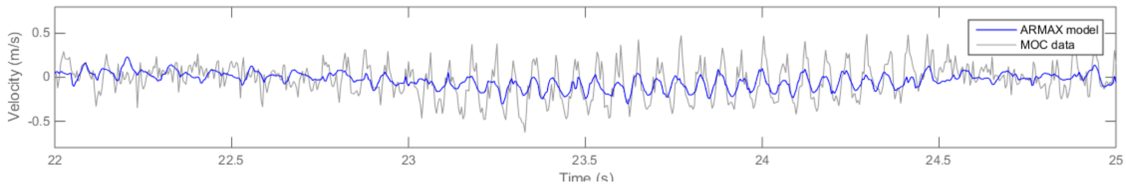
Model type: ARMAX, focus on simulation with regularisation ($\lambda = 10^{-4}$)

Model parameters (12 inputs): $\begin{cases} na = 1 \\ nb = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 \end{pmatrix} \\ nc = 2 \\ nk = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{cases}$

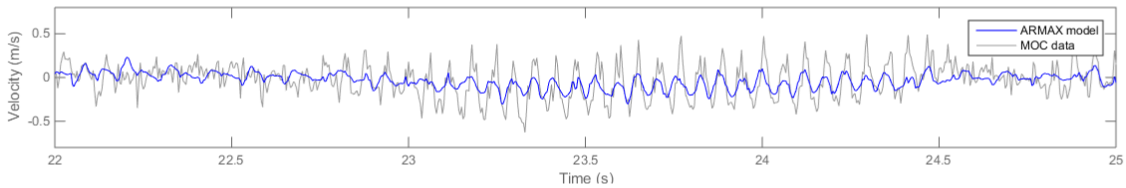
Model parameters (13 inputs): $\begin{cases} na = 1 \\ nb = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 0 \end{pmatrix} \\ nc = 2 \\ nk = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{cases}$

Training set: run 1

Validation set: run 2



(a) 12 inputs
Fit: 14.66



(b) 13 inputs
Fit: 14.66

Figure 37: Measured and estimated outputs for the validation run of the estimation ARMAX model on improved Oncilla data

Figure 37 shows the estimated and measured outputs of this ARMAX model. As for Pleurobot, the regularisation of the estimation is necessary to have outputs that do not diverge, but their variance is consequently reduced and the estimated outputs have a smaller amplitude than the measured ones. Moreover, adding an input does not improve the model in this case, as the obtained estimation seem to be identical and have the same fit values equal to 14.66%.

To avoid problems with regularisation, BJ model 1 is hand-tuned to improve the results of the estimations. The parameters of this second BJ model are designed similarly to those of the previous BJ model: the polynomial orders of the accelerations are the same in nb and nf, horizontal and vertical forces have different parameters contrary to those of frequency and rotation references, which are the same. Delays on the accelerations are tested as null, as others are equal to 1 for forces and to 4 for references. The parameters of the model are the following:

Reference name of the estimation: BJ model 2 on improved Oncilla data

Model type: BJ, focus on simulation with zero initialisation

Model parameters (12 inputs): $\begin{cases} nb = \begin{pmatrix} 2 & 2 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 2 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 2 & 2 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \end{pmatrix} \\ nk = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 \end{pmatrix} \end{cases}$

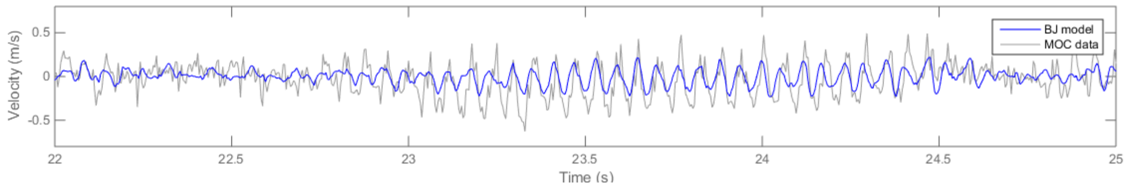
Model parameters (13 inputs): $\begin{cases} nb = \begin{pmatrix} 2 & 2 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 2 & 2 \end{pmatrix} \\ nc = 2 \\ nd = 2 \\ nf = \begin{pmatrix} 2 & 2 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 3 \end{pmatrix} \\ nk = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 4 \end{pmatrix} \end{cases}$

Training set: run 1

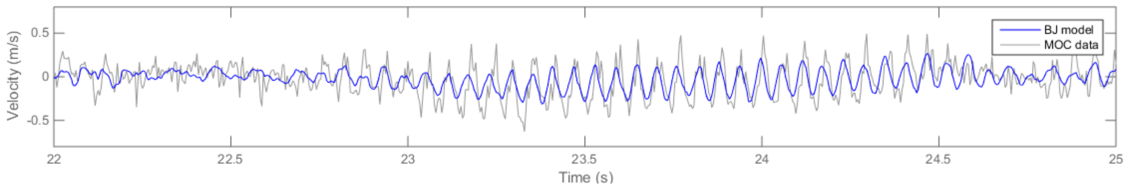
Validation set: run 2

Figure 38 shows the results of the estimation of BJ model 2 with 12 and 13 inputs. They are more promising than those obtained with BJ model 1 as there is no drift between estimated and measured output. Moreover, the estimations follow the reference when the variation of velocity is big enough, even if the amplitude is not exactly the same. Hoever, the small variations of velocity are not reproduced by the estimation, giving low fit values equal to 12.02% and 16.96%.

The same model is also tested taking run 2 as training set and run 1 as validation. Fit values are equal to 7.769% and 9.55%. Visually, the estimations have the same behaviour independently of the repartition of the runs in training and validation sets. They are shown in Appendix D.



(a) 12 inputs
Fit: 12.02



(b) 13 inputs
Fit: 16.96

Figure 38: Measured and estimated outputs for the validation run of the estimation BJ model 2 on improved Oncilla data

A third BJ model is then tested with different parameters for the references, bigger delay for accelerations and different delay for horizontal and vertical forces. The following parameters are used:

Reference name of the estimation: BJ model 3 on improved Oncilla data

Model type: BJ, focus on simulation with zero initialisation

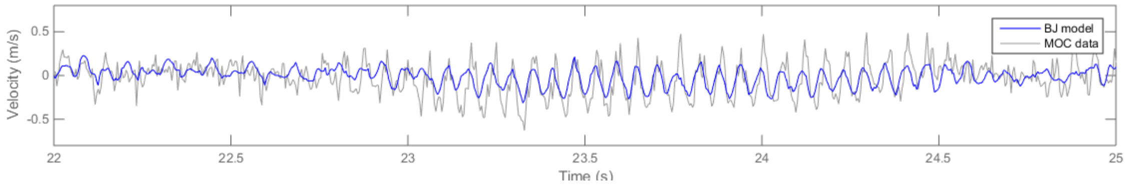
Model parameters (12 inputs): $\begin{cases} nb = \begin{pmatrix} 2 & 2 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 2 \end{pmatrix} \\ nc = 3 \\ nd = 3 \\ nf = \begin{pmatrix} 2 & 2 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \end{pmatrix} \\ nk = \begin{pmatrix} 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 5 \end{pmatrix} \end{cases}$

Model parameters (13 inputs): $\begin{cases} nb = \begin{pmatrix} 2 & 2 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 2 & 1 \end{pmatrix} \\ nc = 3 \\ nd = 3 \\ nf = \begin{pmatrix} 2 & 2 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 1 \end{pmatrix} \\ nk = \begin{pmatrix} 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 5 & 3 \end{pmatrix} \end{cases}$

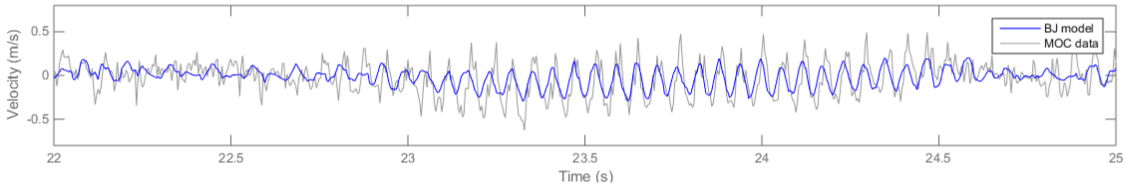
Training set: run 1

Validation set: run 2

Figure 39 shows the estimated and measured outputs of this third BJ model. Fit values are improved compared to the previous one are they are equal to 18.1% and 17.38%. Visually, the estimations follow better



(a) 12 inputs
Fit: 18.1



(b) 13 inputs
Fit: 17.38

Figure 39: Measured and estimated outputs for the validation run of the estimation BJ model 3 on improved Oncilla data

the variation of the references, but the small oscillations of the measured outputs remain not reproduced by the estimated ones.

7.5 Discussion of the Results

The accuracy on Oncilla data of the models designed for Pleurobot simulations is limited. However, BJ estimations are easily improved by hand-tuning some of the parameters, while keeping similar relations between nb and nf corresponding to each input. The necessity of these changes can by explained by the fact that Pleurobot and Oncilla are different robot and the velocity estimated is the longitudinal one for Pleurobot and the lateral one for Oncilla. Their estimations should thus require different models.

Moreover, the estimations on Pleurobot's velocity are trained on several short runs as those for Oncilla's velocity are trained on only one very long run. The influence of the length and quantity of runs provided to estimate models should thus be determined to choose the better method to obtain good fits between estimated and measured outputs.

The influence of a thirteenth output does not seem to be relevant for the estimations. Fit values are similar for models obtained with 12 or 13 inputs. It is possible that the rotated acceleration is perturbed by a drift of the yaw. To avoid this problem, the input for system identification could be composed of the accelerations, the forces, the frequency reference, the pitch and the roll.

The best models found in this section generally estimate well the variations of velocity equal or higher to $0.3m/s$. Smaller ones are ignored by models, which estimate the speed nearly constant. However, as the goal is to know if the robot is perturbed by an external element or falls, small changes of velocity are not really important. These models could thus be sufficient to detect a failure in the behaviour of Oncilla.

8 Detection of Unexpected Pleurobot's Behaviours

8.1 Computation of the Error Between Commanded and Estimated Velocity

The goal of this section is to detect moments when the robot's behaviour is not the expected one. As the estimations used are those obtained on Pleurobot simulations, as they are better than those on Oncilla. The used model is BJ model on Experiments 1, 2 and 3.

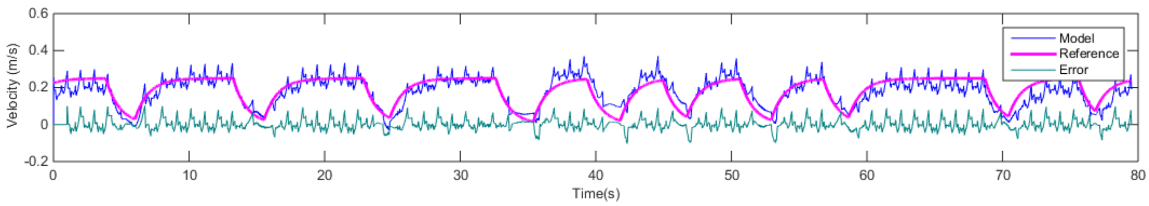
In order to get rid of the System Identification Toolbox, in prevision of future tests on Webots using C++, the function $bjModel$ is implemented in Matlab. This function calculates the output of BJ models with multiple inputs and a single output. The corresponding code and an example of application are given in Appendix E.

To detect unexpected behaviours of the robot, the error between the commanded and the real velocities of Pleurobot needs to be calculated. As the real velocity is not available, the estimated one is used. In this case, the commanded speed is equal to half of the frequency reference. The error is computed numerically using an integration method, so that too sudden variations of the estimation or velocity are not considered. The error between the model output and the frequency, the estimated error, is thus given as the trapezoidal numerical integration over one second of the difference between half the frequency and the estimated velocity. The error between the estimated and measured output, the measured error, is also computed as the trapezoidal numerical integration over one second of the difference between the two velocities. The estimated and measured errors can thus be compared.

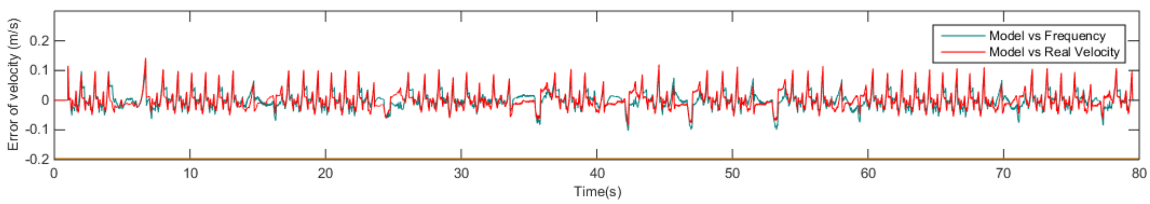
8.2 Detection of Unexpected Behaviours

The detection of unexpected behaviours is based on the estimated error. By observing the frequency reference, the estimated output and the estimated error, it was determined that, if the error is below a threshold, the robot does not behave as it should. The threshold is fixed by hand at $-0.2m/s$. It is chosen negative as, in experiments made, unexpected behaviour happens when the robot is blocked and its velocity decreases.

Figures 40a, 41a, 42a and 43a shows the estimated and commanded velocities and the estimated error of run 5 of Experiments 1 and 2 and runs 4 and 5 of Experiment 3. Figures 40b, 41b, 42b and 43b shows the corresponding estimated and measured errors. On these figures, the threshold is marked by an orange horizontal line. Each time the measured error is below the threshold is signalled by a vertical orange line. This line is full if the measured error goes also under the threshold and dashed or dotted otherwise. Dashed lines show the false positives, which means that a unexpected behaviour is detected when the robot walks normally. Contrariwise, dotted lines show the false negative, so that an unexpected behaviour is not reported when it should be.



(a) Estimated and commanded velocities and their difference



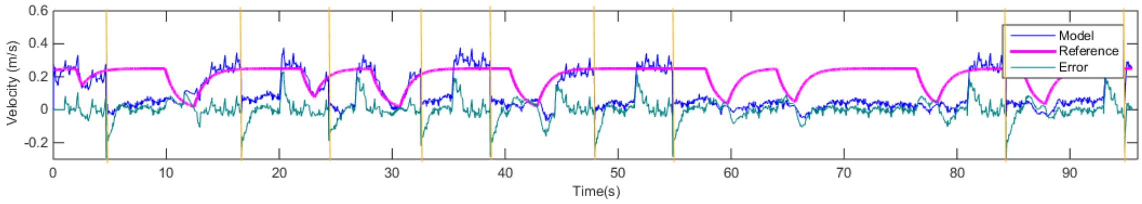
(b) Measured and estimated errors

Figure 40: Detection of non-expected behaviour of Pleurobot during run 5 of Experiment 1

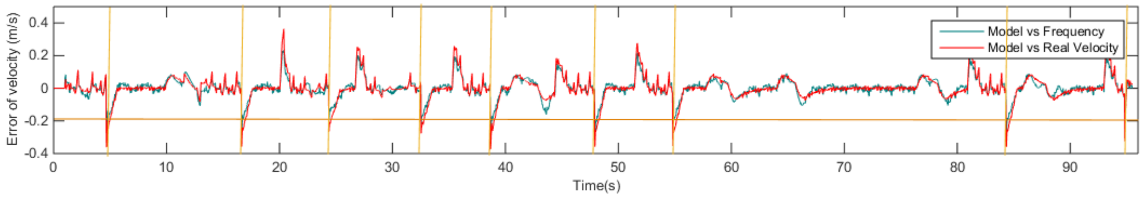
During runs of Experiment 1, the robot is not perturbed by any element in its environment, so that it follows well the commanded velocity. As shown in Figure 40, the estimated velocity is very similar to the frequency reference. Moreover, the difference between estimated and measured errors consists only of small variations of amplitude. None of these errors go under the threshold, so that the robot behaves as expected.

During Experiment 2, the robot is sometimes blocked by an obstacle. As shown in Figure 41, each time it is suddenly stopped, the estimated error goes below the threshold, signalling the unexpected behaviour.

Moreover, the model is good enough so that all reported strange behaviour of the robot corresponds to the measured one, as the measured error goes under the threshold at the same time.



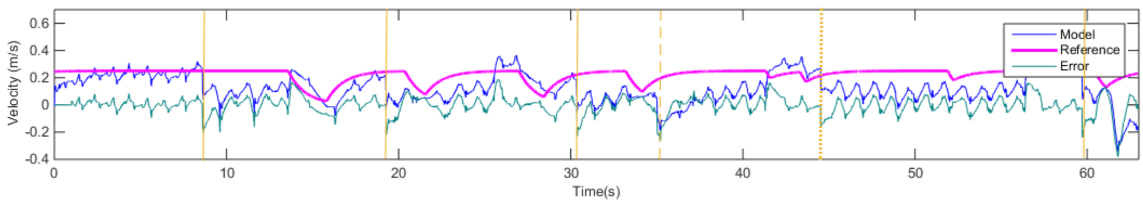
(a) Estimated and commanded velocities and their difference



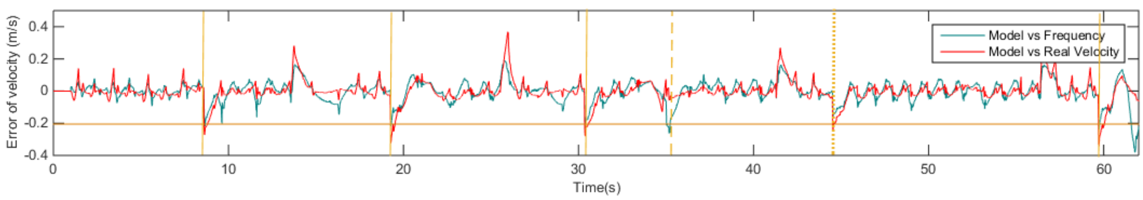
(b) Measured and estimated errors

Figure 41: Detection of non-expected behaviour of Pleurobot during run 5 of Experiment 2

The BJ model chosen in this section does not give totally reliable estimations on Experiment 3, as shown in Section 6.3.3. The better estimation for this Experiment was performed on run 4. As shown in Figure 42, four correct unexpected behaviour are reported. However, the estimated error goes under the threshold as the real one does not at time $t = 35s$. The opposite happens at time $t = 45s$, as the estimated error does not go below the threshold when the measured one does.



(a) Estimated and commanded velocities and their difference



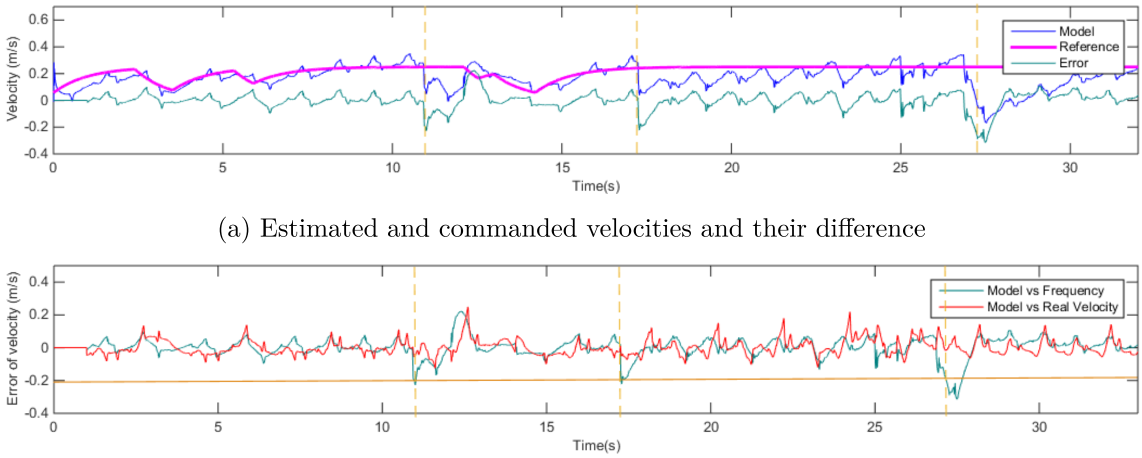
(b) Measured and estimated errors

Figure 42: Detection of non-expected behaviour of Pleurobot during run 4 of Experiment 3

During run 5 of the same experiment, the measured error never goes under the threshold. However, three unexpected behaviour are signalled by the estimated error. Even if the difference between estimated and measured errors is not visually important, the differences between their amplitudes are enough to have false positives.

8.3 Discussion of the Results

Using a simple threshold applied on the difference between a multiple of the frequency reference and the estimated velocity is often sufficient to detect unexpected behaviour of the robot during experiments performed with Pleurobot. The quality of the estimation influences the detection success, but, even if the model does not



(a) Estimated and commanded velocities and their difference

(b) Measured and estimated errors

Figure 43: Detection of non-expected behaviour of Pleurobot during run 5 of Experiment 3

perfectly fit the measured velocity, most of cases of blocked robot are detected. Moreover, false detections are more frequent than missed detections. These false detections can be forgotten if the robot behaves normally again after some time. Similarly, if some unexpected behaviour is not detected immediately but subsists during some time, it will surely be detected after a few seconds.

The unexpected behaviours searched and detected here are cases where the robot is blocked. Different thresholds would surely be necessary to detect other problems, such as cases where the Oncilla falls. Additionally, the choice of threshold value would be improved by testing this method on more different experiments for Pleurobot.

9 Conclusion and Future Work

Models estimating the velocity of robots has been computed using the System Identification Toolbox from Matlab. Estimations has been made on data from a real robot, Oncilla, and from simulations of Pleurobot. Different models has been tested on several experiments with hand-tuned parameters. A function processing the Box-Jenkins model has then be implemented and a method to detect unexpected behaviours of Pleurobot has been tested.

The first observation being made is that multiple or long runs during which the robot walks are necessary to obtain good models. Second, ARX and ARMAX model are often not sufficient to obtain good estimations for several different behaviours of the robot. BJ models were then preferred. Better models are also obtained when only one output has to be estimated. Moreover, the models does not need to be trained on all experiments: training them on two or three experiments over four on Pleurobot simulations was almost always sufficient to obtain good fit between all estimated and measured outputs.

Results on simulations were more promising than those on the real robot, which was expected as simulations provide cleaner data. However, it has been proved that system identification was able to give good estimations on very noisy functions. The way the data are processed is important to avoid falsifying them and adding singularities. The models obtained for Pleurobot simulations has been tested on Oncilla data. More satisfying results were obtained after having tuned some of the parameters. This could be due to the change of robot and to the fact that the velocity estimated is the forward one for Pleurobot and the lateral one for Oncilla.

Unexpected behaviours of Pleurobot has been detected by computing the error between the commanded and the estimated velocities. A simple threshold on the error was sufficient to detect cases where the robot was blocked by obstacles.

To improve the quality of the models, a systematic search or genetic algorithms could be used to find the best parameters. Nonlinear models should also be tested as they are more complex and could give better estimations, especially on Oncilla data. As the computation of these models is very slow with the System Identification Toolbox, some other environment should be used, as the Neural Networks Toolbox, which is more accurate in computing nonlinear models.

The method to detect unexpected behaviour of Pleurobot should be implemented and tested in Webots, so that the detection would be made in real time during simulations. The detection could also be tested on the real Pleurobot and adapted to Oncilla.

References

[1] M. Bloesch, M. Hutter, M.A. Hoepflinger, S. Leutenegger, C. Gehring, C.D. Remy, and R. Siegwart. ”State estimation for legged robots-consistent fusion of leg kinematics and IMU.”, pp. 17-24, in Robotics, 2013.

[2] J. Hermann, ”Sensor Fusion on Quadrupedal Robots.”, Semester project, in EPFL-Biorob, 2015.

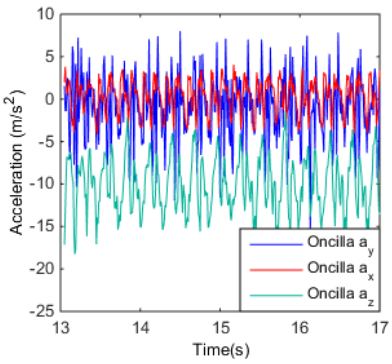
[3] MATLAB and System Identification Toolbox, The MathWorks, Inc., Natick, Massachusetts, United States.

Appendix

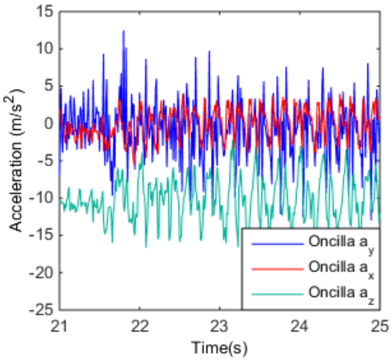
Appendix A: Supplement for System Identification on Oncilla

Inputs for System Identification

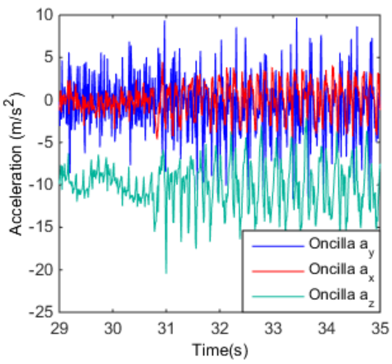
Examples of acceleration and rotated acceleration in function of time are given respectively by Figures 44 and 46 for the Oncilla Moving in Air experiment and by Figure 45 for the Oncilla Walking Straight experiment.



(a) Run 1

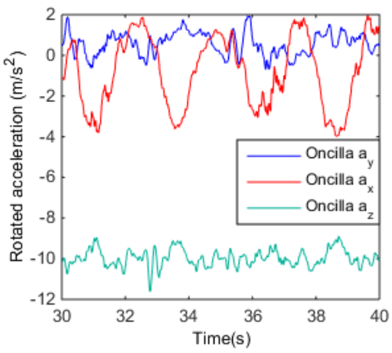


(b) Run 2

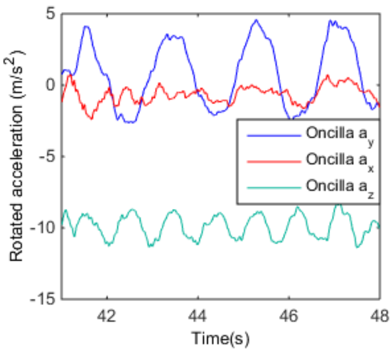


(c) Run 3

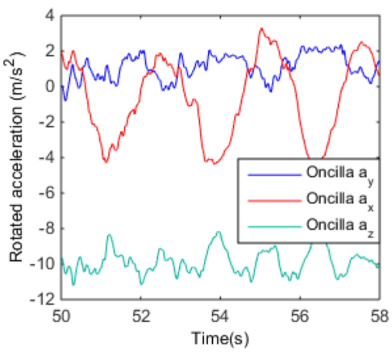
Figure 44: Oncilla acceleration for the Walking Straight experiment



(a) Run 1

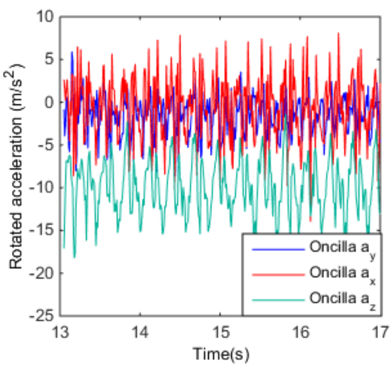


(b) Run 2

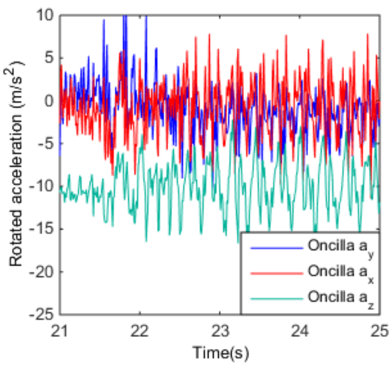


(c) Run 3

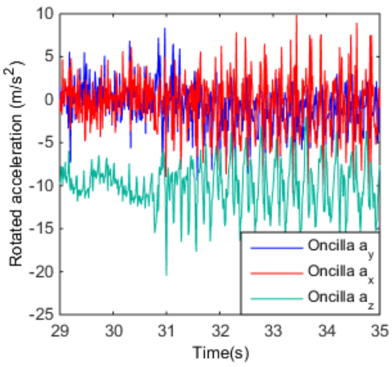
Figure 45: Oncilla rotated acceleration for the Moving in Air experiment



(a) Run 1



(b) Run 2

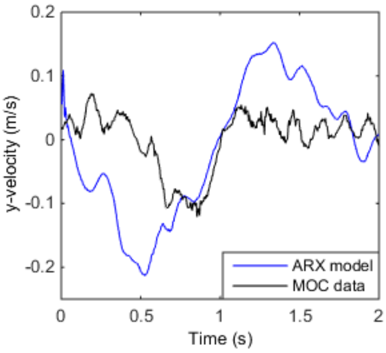


(c) Run 3

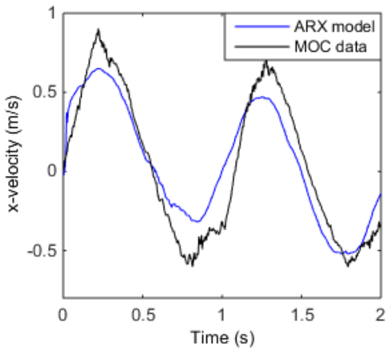
Figure 46: Oncilla rotated acceleration for the Walking Straight experiment

Results of Training Sets

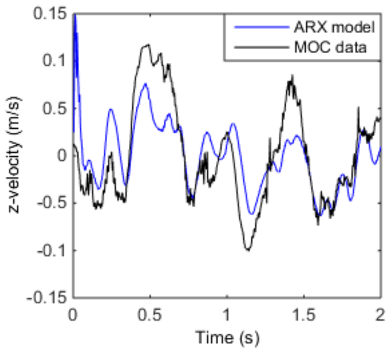
Figure 47 compares the measured and estimated outputs of the training set of the estimation Moving in Air 1.



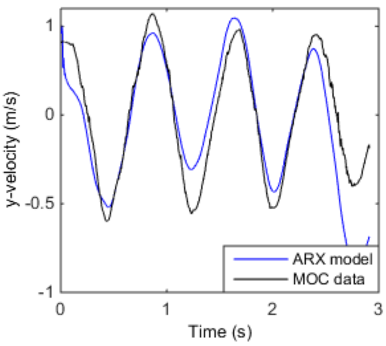
(a) Fit: -106.9



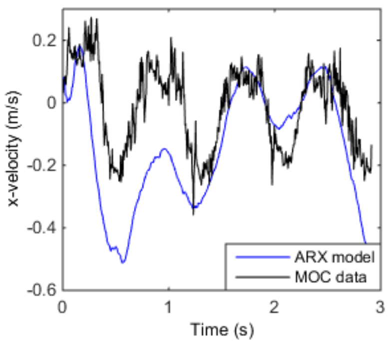
(b) Fit: 63.52



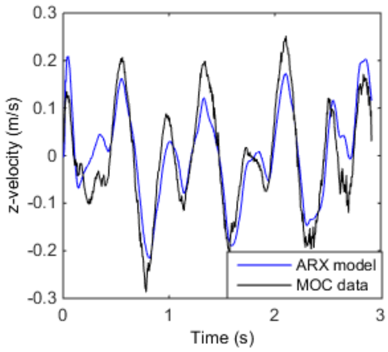
(c) Fit: 30.11



(d) Fit: 44.61



(e) Fit: -48



(f) Fit: 54.67

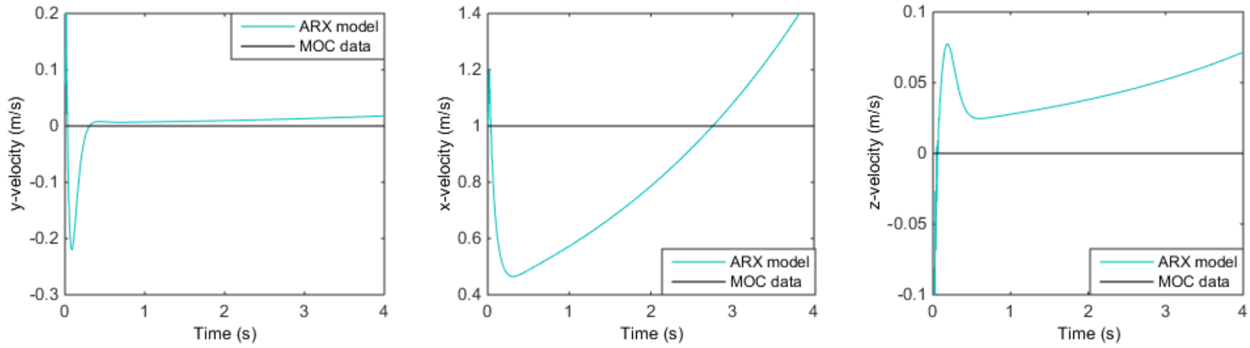
Figure 47: Measured and estimated model outputs for the training set of the estimation Moving in Air 1 : graphs a, b, c are for run 1 and graphs d, e and f are for run 2

Figure 48 compares the measured and estimated outputs of the training set of the estimation Moving in Air 3.

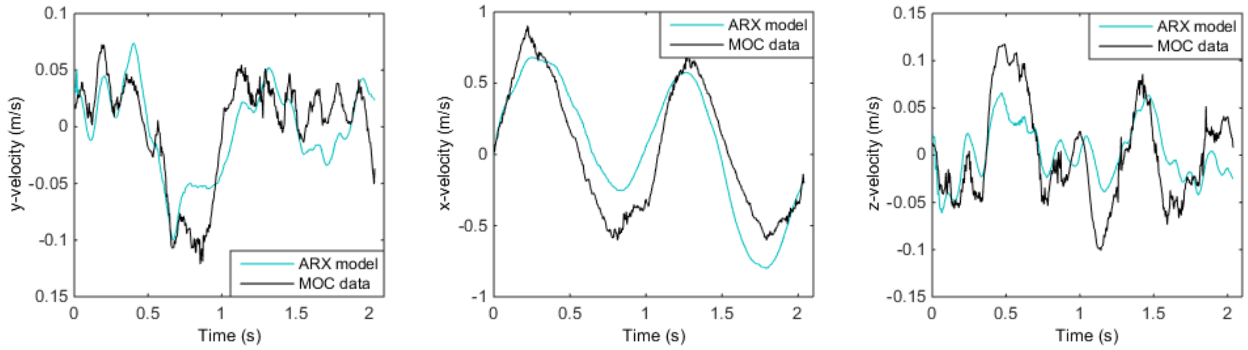
Figure 49 compares the measured and estimated outputs of the training set of the estimation Moving in Air 4.

Figure 50 compares the measured and estimated outputs of the training set of the estimation Walking Straight 1.

Figure 51 compares the measured and estimated outputs of the training set of the estimations Walking Straight 2 and 3.

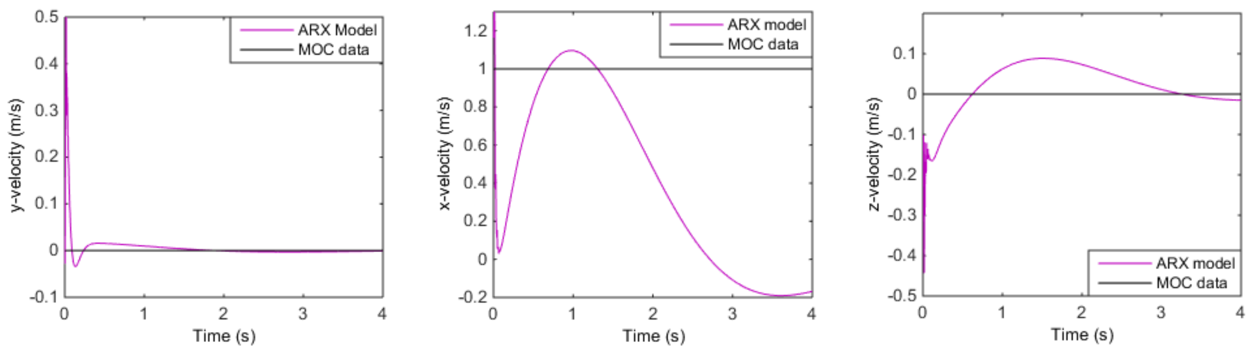


(a) Fit: -∞ (b) Fit: -∞ (c) Fit: -∞

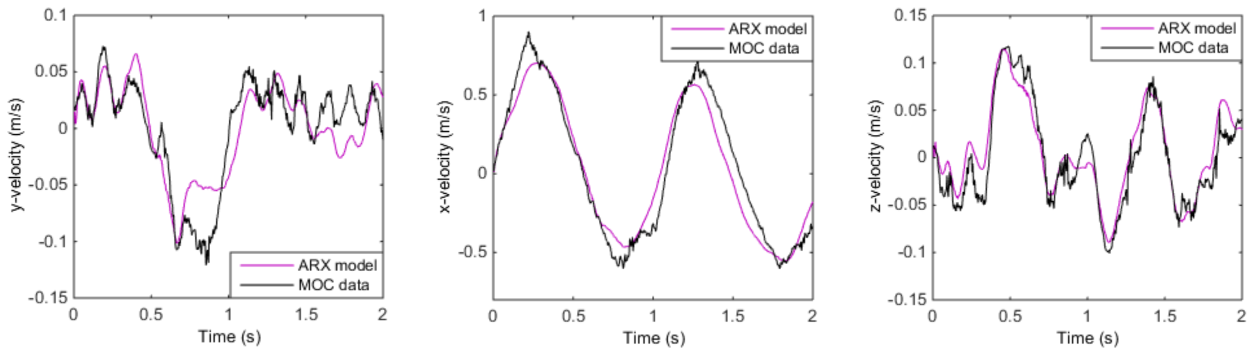


(d) Fit: 38.89 (e) Fit: 50.73 (f) Fit: 31.67

Figure 48: Measured and estimated model outputs for the training set of the estimation Moving in Air 3 : graphs a, b, c are for Zero Acceleration and graphs d, e and f are for run 1 of Oncilla Moving in Air

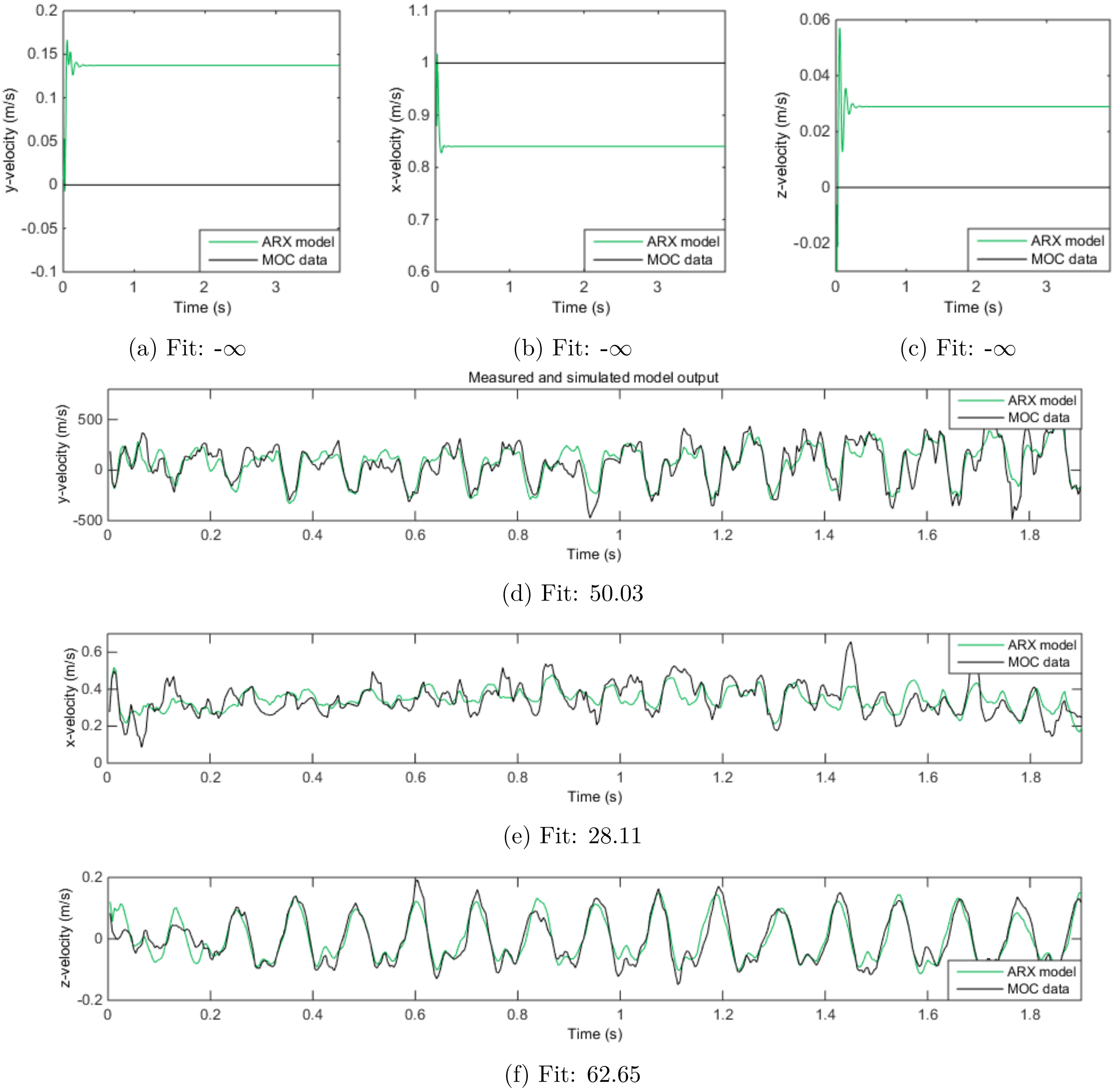


(a) Fit: -∞ (b) Fit: -∞ (c) Fit: -∞



(d) Fit: 45.62 (e) Fit: 74 (f) Fit: 63.61

Figure 49: Measured and estimated model outputs for the training set of the estimation Moving in Air 4 : graphs a, b, c are for Zero Acceleration and graphs d, e and f are for run 1 of Oncilla Moving in Air



(a) Fit: -∞

(b) Fit: -∞

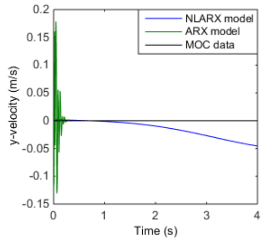
(c) Fit: -∞

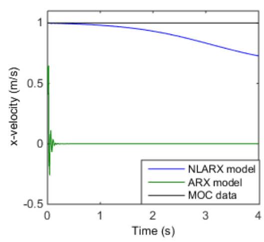
(d) Fit: 50.03

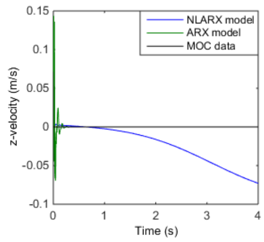
(e) Fit: 28.11

(f) Fit: 62.65

Figure 50: Measured and estimated model outputs for the training set of the estimation Walking Straight 1: graphs a, b, c are for Zero Acceleration and graphs d, e and f are for run 1 of Oncilla Walking Straight



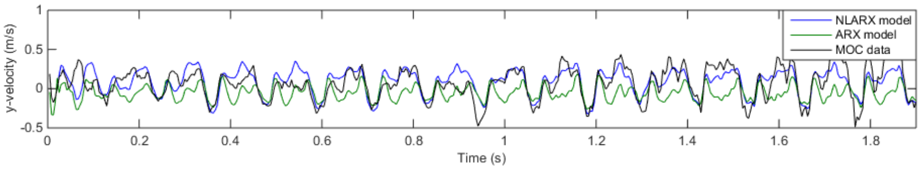




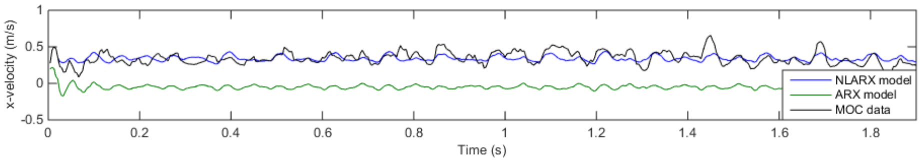
(a) Fit ARX: -∞
Fit NLARX: -∞

(b) Fit ARX: -∞
Fit NLARX: -∞

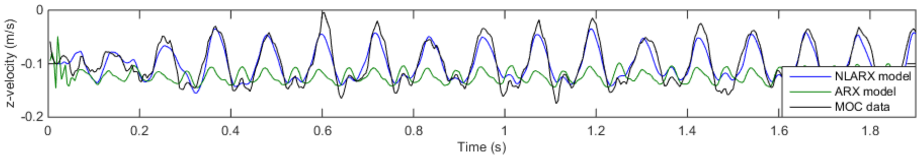
(c) Fit ARX: -∞
Fit NLARX: -∞



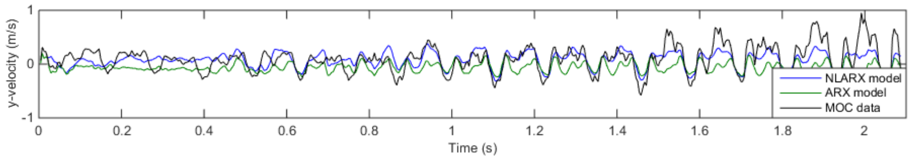
(d) Fit ARX: -7.82
Fit NLARX: 32.77



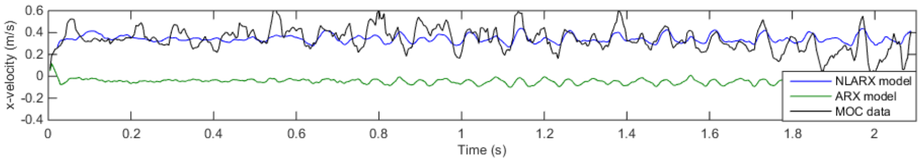
(e) Fit ARX: -359.1
Fit NLARX: 11.07



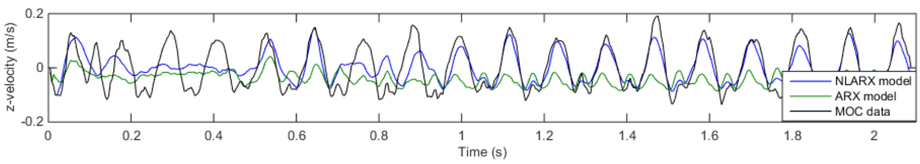
(f) Fit ARX: -12.93
Fit NLARX: 61.03



(g) Fit ARX: -8.774
Fit NLARX: 26.79



(h) Fit ARX: -275.6
Fit NLARX: 10.57



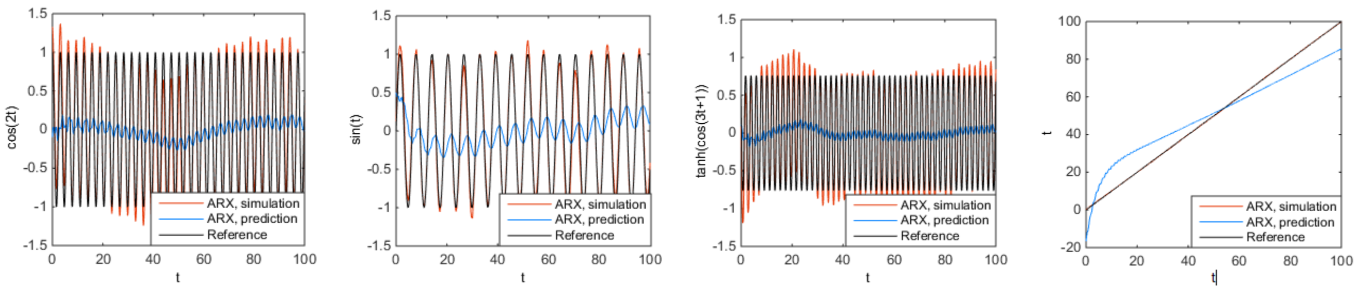
(i) Fit ARX: -8.76
Fit NLARX: 35.31

Figure 51: Measured and estimated model outputs for the training set of the estimation Walking Straight 1: graphs a, b, c are for Zero Acceleration, graphs d, e and f are for run 1 of Oncilla Walking Straight and graphs g, h and i are for run 2 of Oncilla Walking Straight

Appendix B: Supplement for System Identification on Noisy Functions

Results of Training Set

Figure 52 shows the measured and estimated model outputs for the estimation on noisy function. The comparison between an ARX model with focus on prediction and the same model with focus on simulation is done.



(a) Fit simulation: 75.25
Fit prediction: −0.3411

(b) Fit simulation: 86.45
Fit prediction: 0.9363

(c) Fit simulation: 77.29
Fit prediction: −0.3988

(d) Fit simulation: 99.47
Fit prediction: 70.21

Figure 52: Measured and estimated outputs for the training set of the estimation on functions with additional random noise

Appendix C: Supplement for System Identification on Pleurobot Simulations

Inputs and outputs for System Identification

Figures 53, 54, 55 and 56 show the inputs and outputs of each experiment on Pleurobot simulation. The inputs and outputs are respectively the accelerations and forces along each axis and the velocities computed from GPS data.

Models Trained on One Experiment

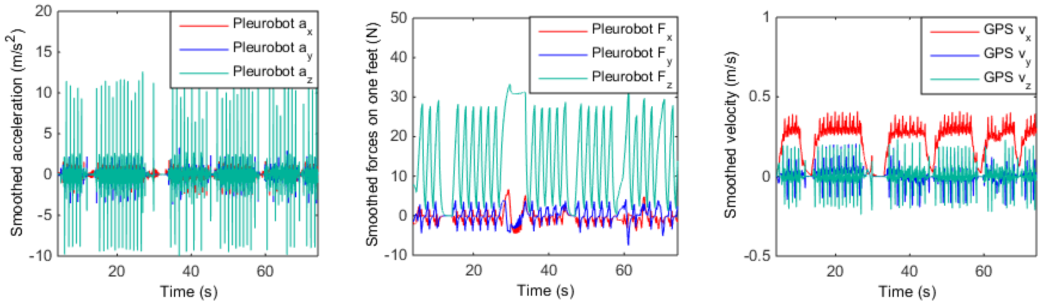
Figure 57 shows the remaining results of the estimation BJ model on Experiment 1.

Models Trained on Two Experiments

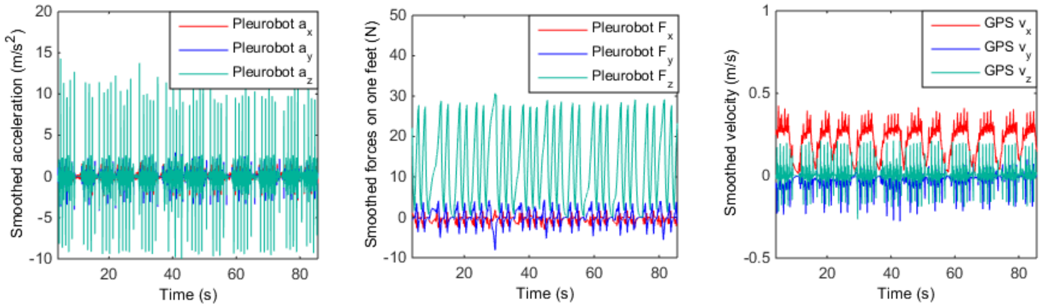
Figure 58 shows the remaining results of the estimation ARX model on Experiments 1 and 2.
Figure 59 shows the remaining results of the estimation ARMAX model on Experiments 1 and 2.
Figure 60 shows the remaining results of the estimation BJ model 1 on Experiments 1 and 2.
Figure 61 shows the remaining results of the estimation BJ model 2 on Experiments 1 and 2.

Models Trained on Three Experiments

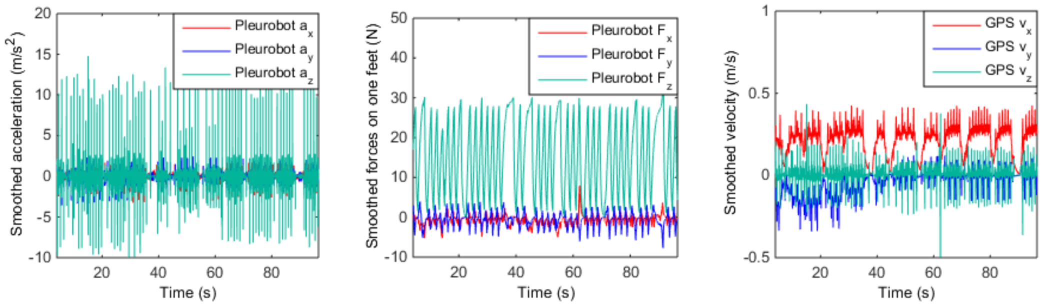
Figure 62 shows the remaining results of the estimation ARMAX model on Experiments 1, 2 and 3.
Figure 63 shows the remaining results of the estimation BJ model on Experiments 1, 2 and 3.



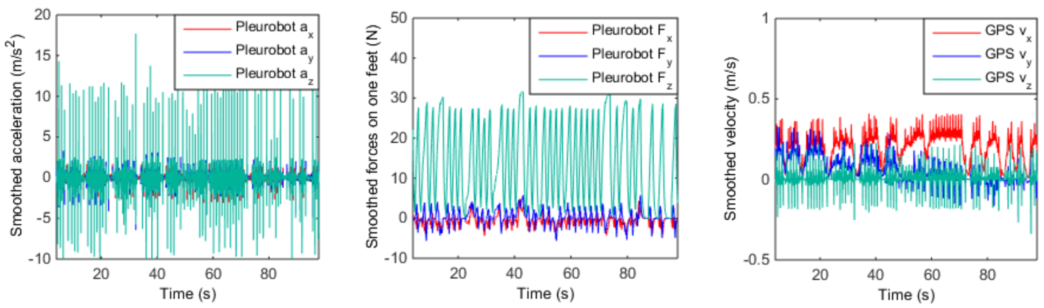
(a) Run 1



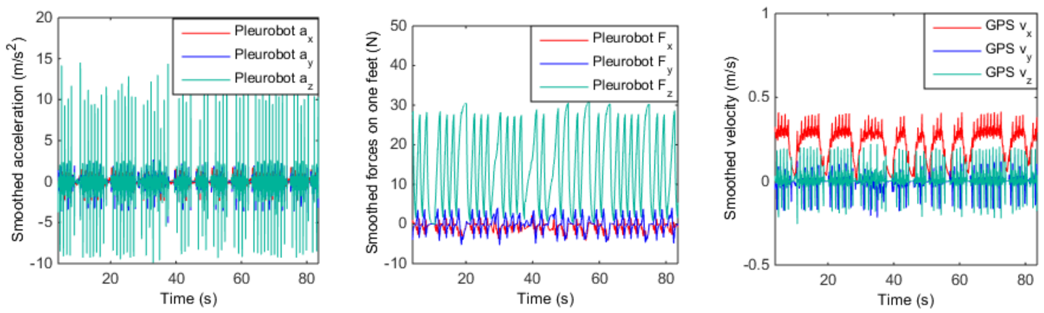
(b) Run 2



(c) Run 3

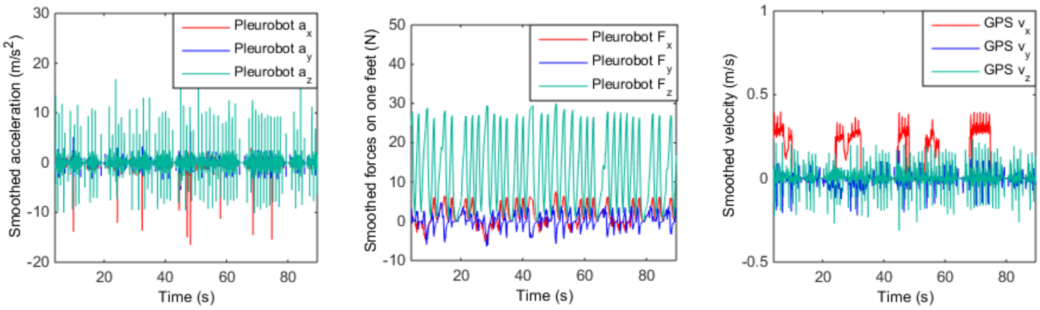


(d) Run 4

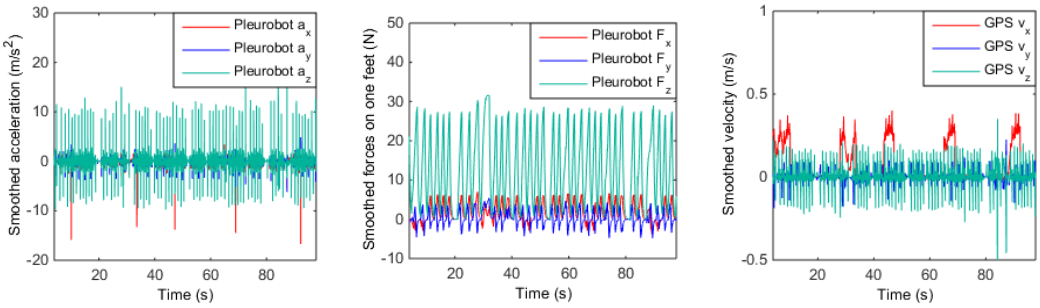


(e) Run 5

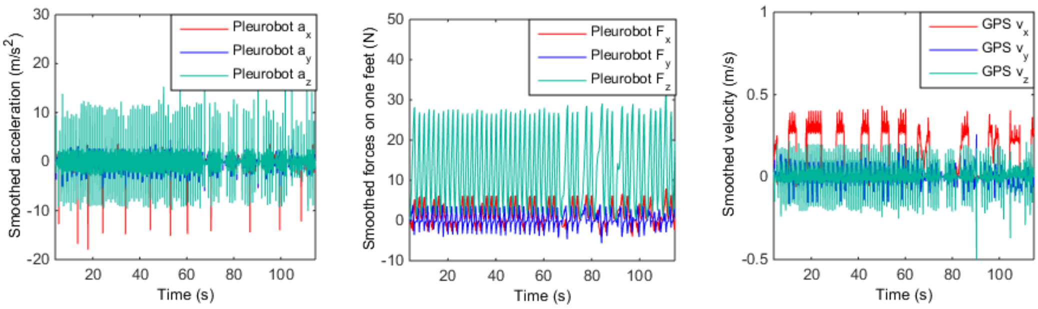
Figure 53: Inputs (acceleration and forces) and outputs (velocity) for system identification on Experiment 1 of Pleurobot simulations



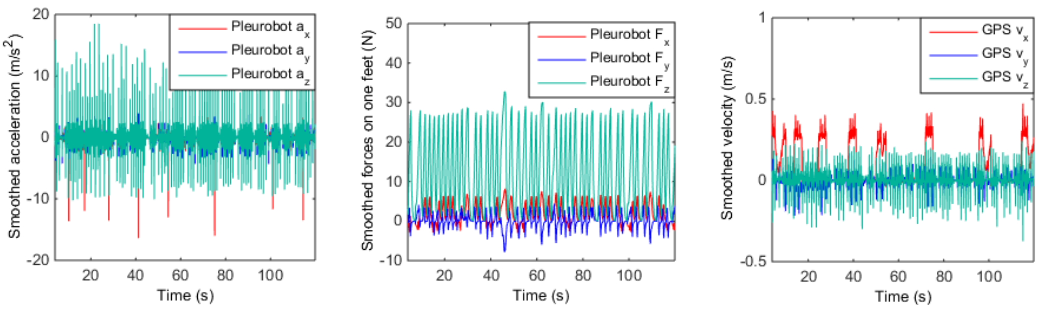
(a) Run 1



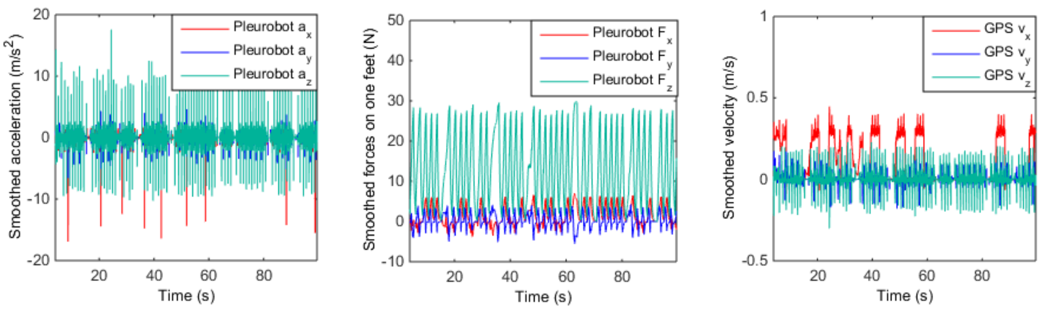
(b) Run 2



(c) Run 3

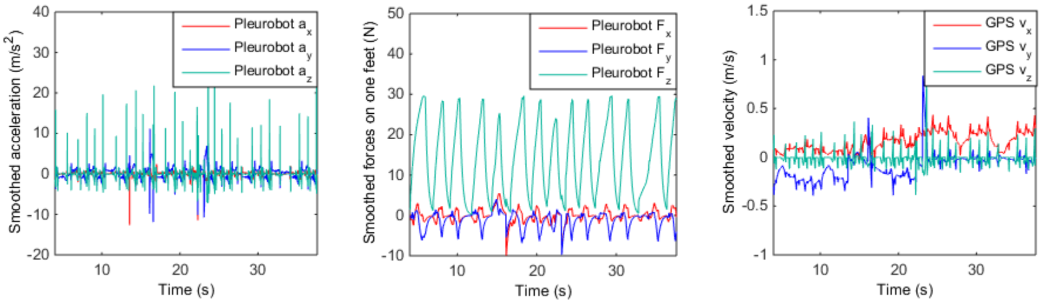


(d) Run 4

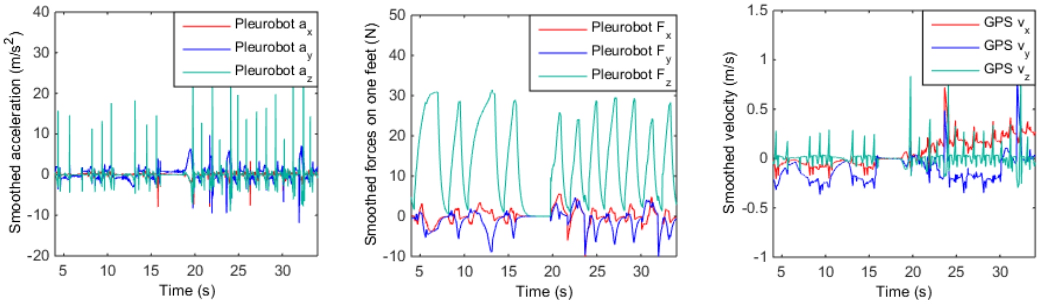


(e) Run 5

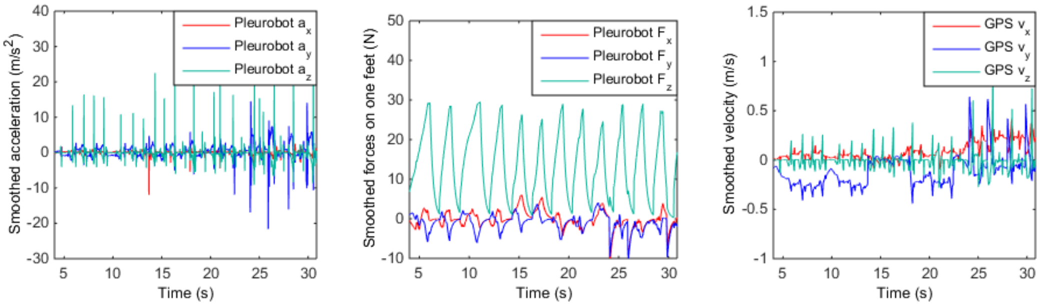
Figure 54: Inputs (acceleration and forces) and outputs (velocity) for system identification on Experiment 2 of Pleurobot simulations



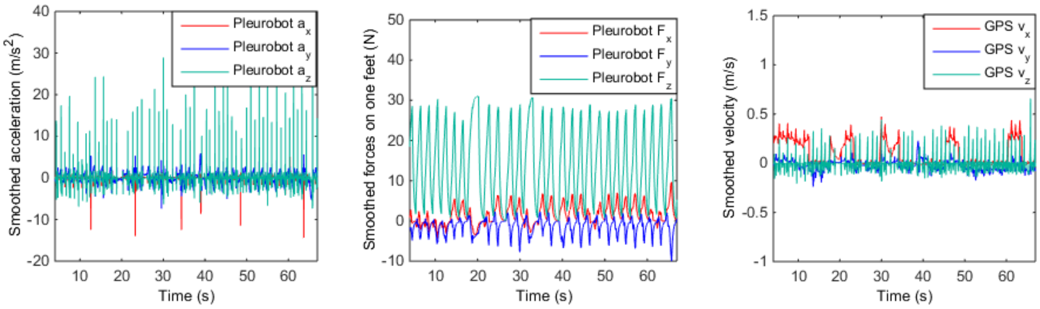
(a) Run 2



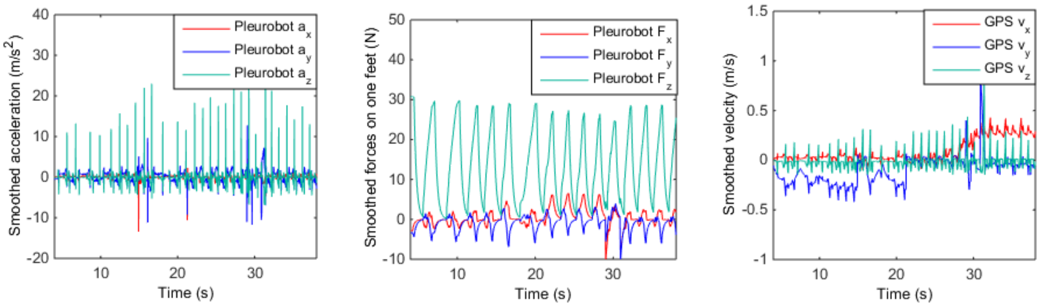
(b) Run 2



(c) Run 3

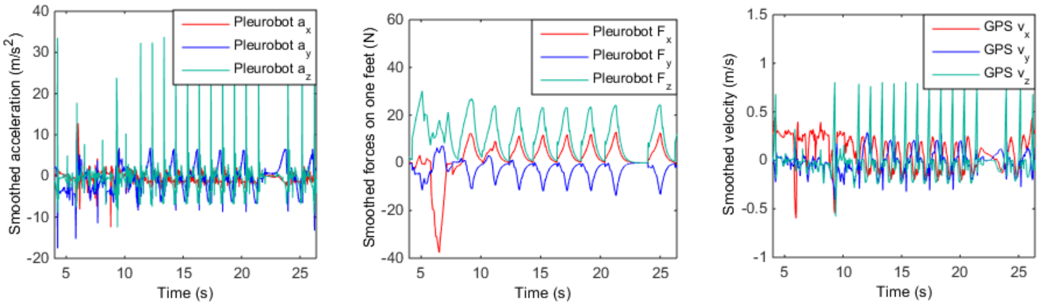


(d) Run 4

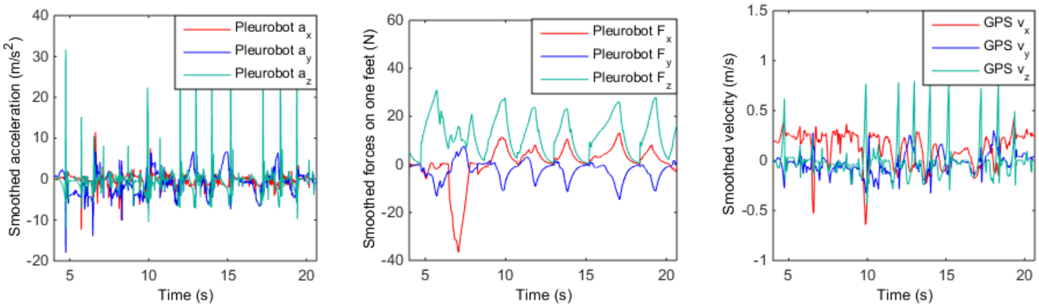


(e) Run 5

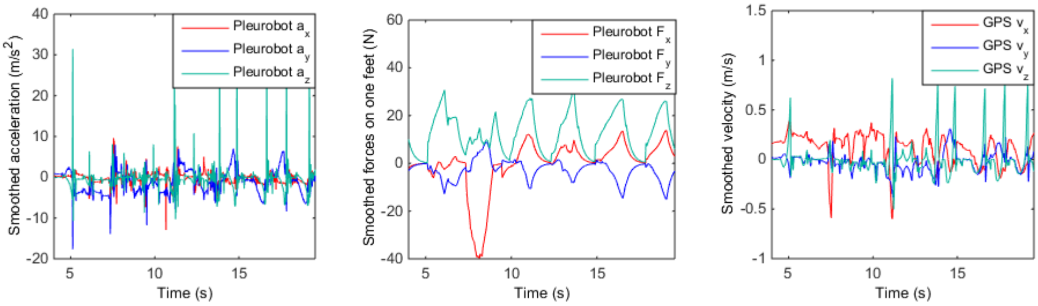
Figure 55: Inputs (acceleration and forces) and outputs (velocity) for system identification on Experiment 3 of Pleurobot simulations



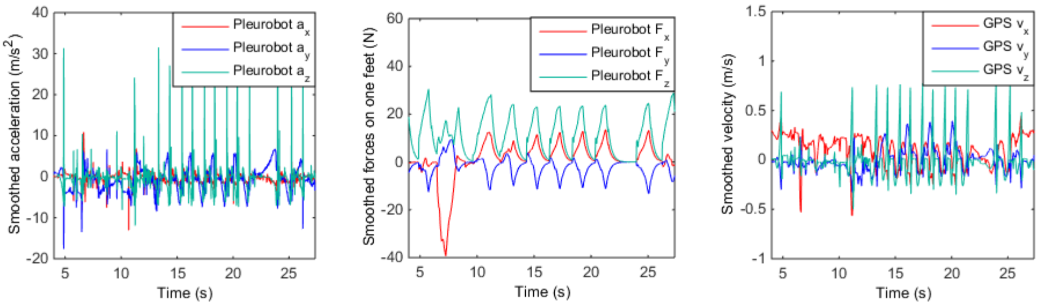
(a) Run 2



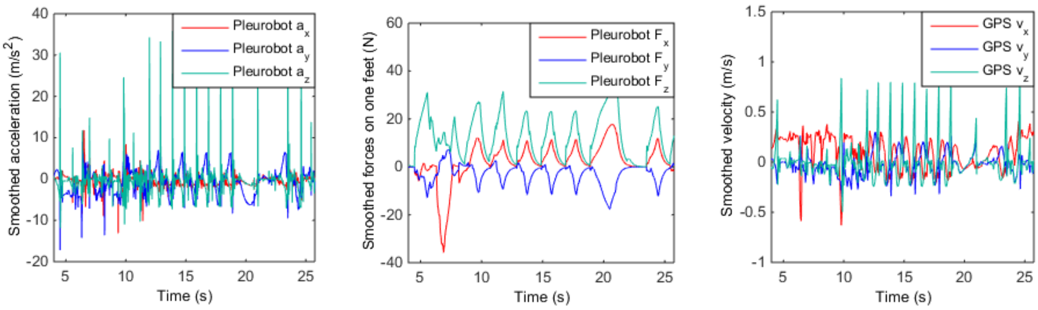
(b) Run 2



(c) Run 3

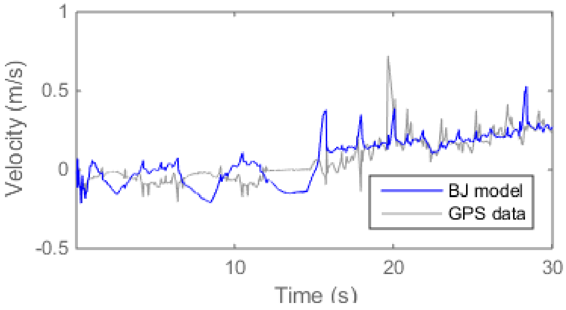


(d) Run 4

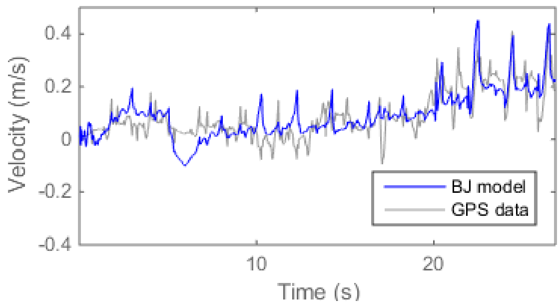


(e) Run 5

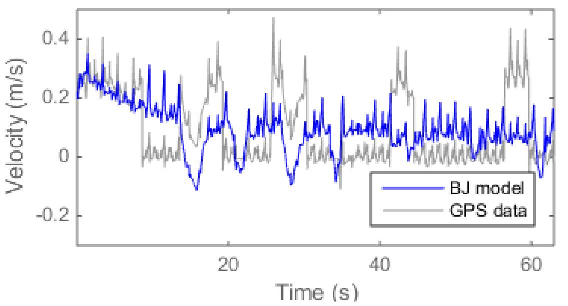
Figure 56: Inputs (acceleration and forces) and outputs (velocity) for system identification on Experiment 4 of Pleurobot simulations



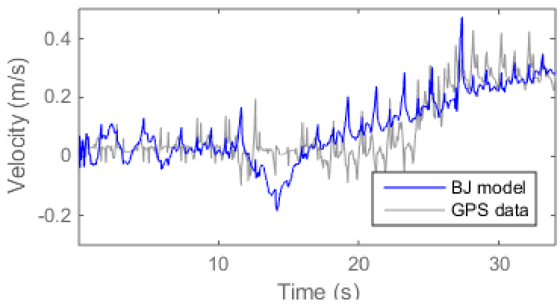
(a) Experiment 3, run 2
Fit: 23.31



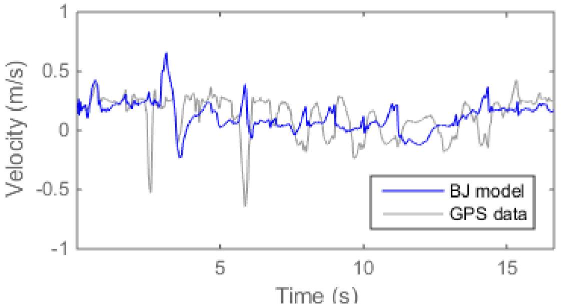
(b) Experiment 3, run 3
Fit: 30.67



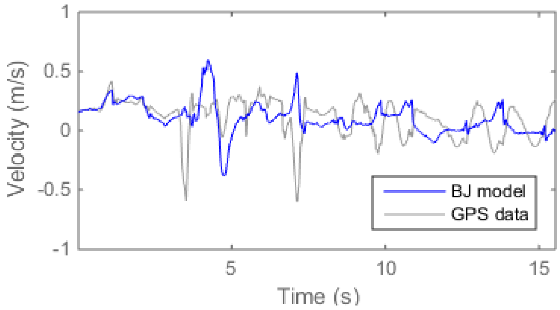
(c) Experiment 3, run 4
Fit: -2.446



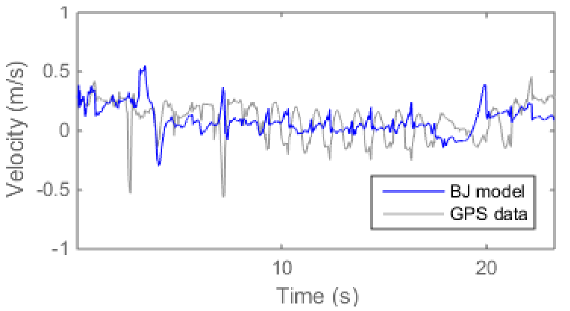
(d) Experiment 3, run 5
Fit: 37.04



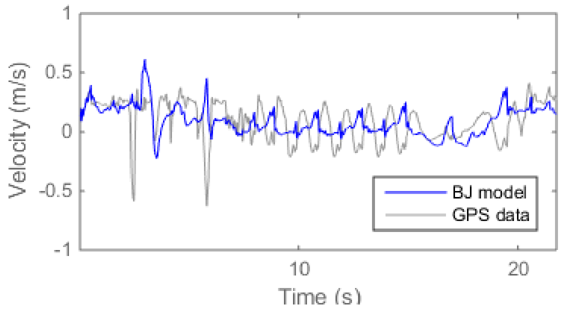
(e) Experiment 4, run 2
Fit: -11.47



(f) Experiment 4, run 3
Fit: -24.25

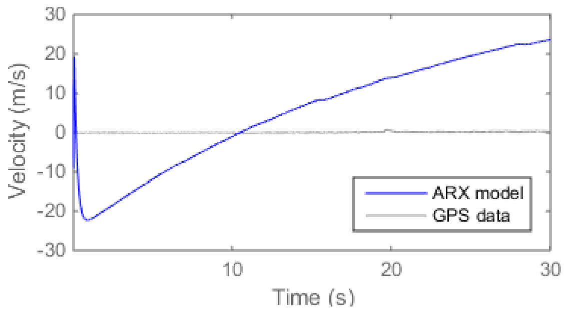


(g) Experiment 4, run 4
Fit: -13.89

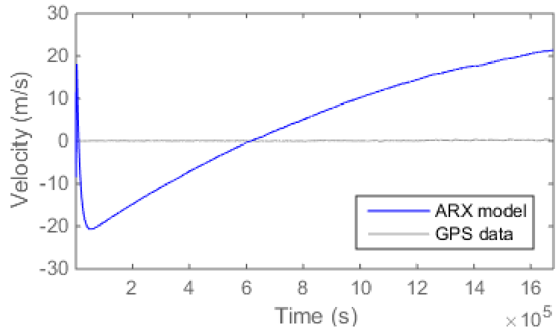


(h) Experiment 4, run 5
Fit: -9.586

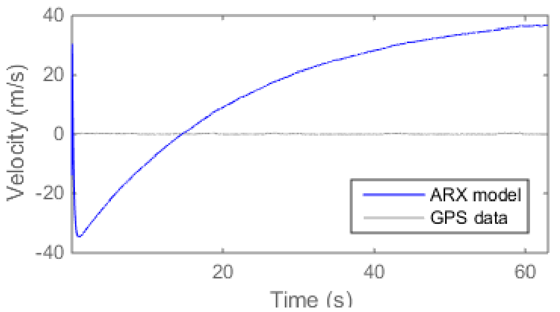
Figure 57: Measured and estimated outputs for remaining part of validation set of the estimation BJ model on Experiment 1



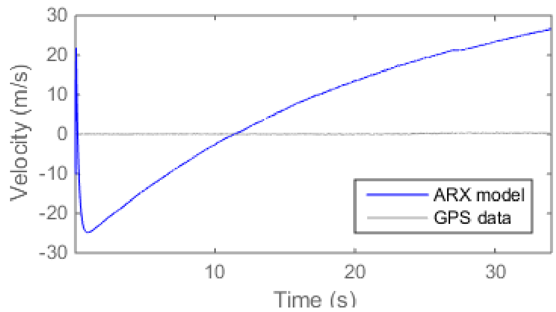
(a) Experiment 3, run 2
Fit: −10040



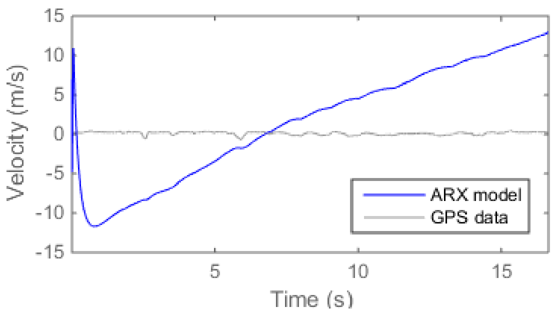
(b) Experiment 3, run 3
Fit: −14760



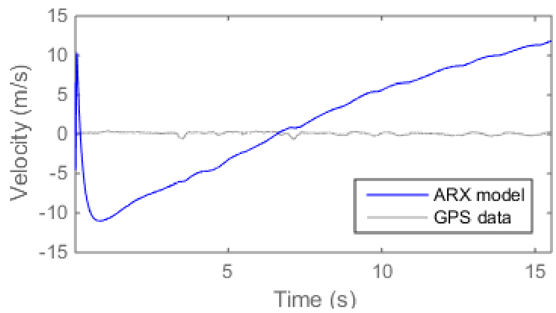
(c) Experiment 3, run 4
Fit: −20810



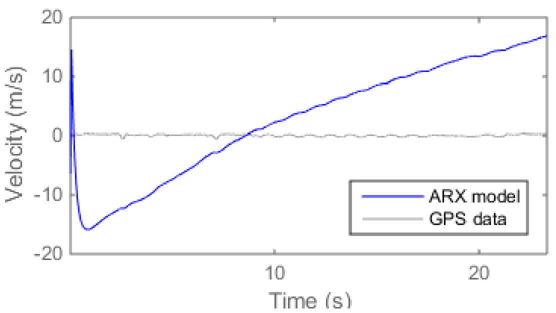
(d) Experiment 3, run 5
Fit: −13850



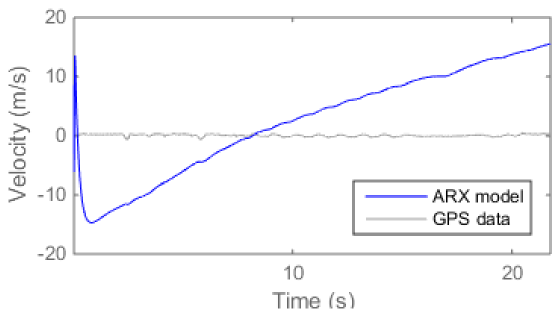
(e) Experiment 4, run 2
Fit: −4285



(f) Experiment 4, run 3
Fit: −4417

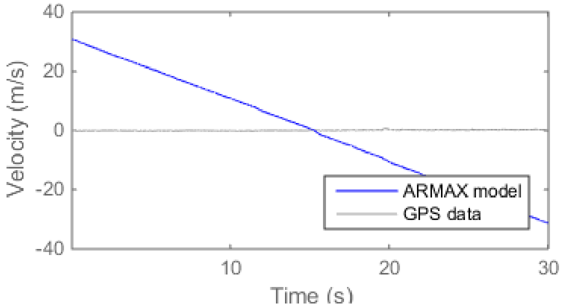


(g) Experiment 4, run 4
Fit: −6293

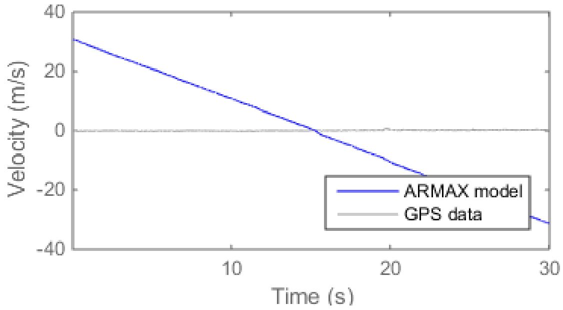


(h) Experiment 4, run 5
Fit: −5390

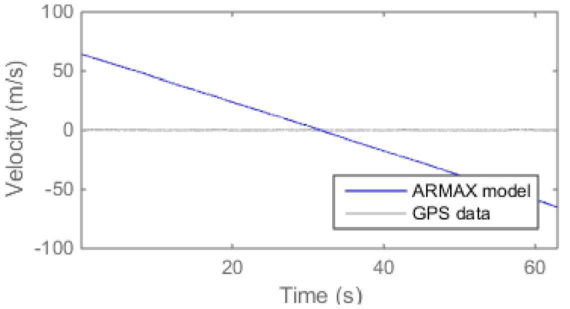
Figure 58: Measured and estimated outputs for remaining part of validation set of the estimation ARX model on Experiments 1 and 2



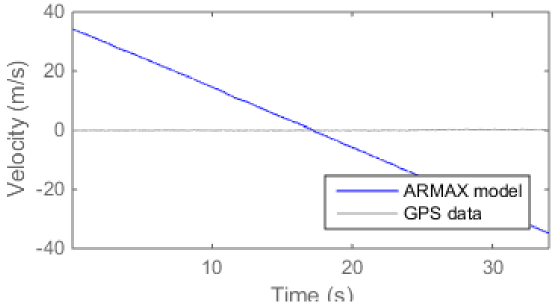
(a) Experiment 3, run 2
Fit: -12750



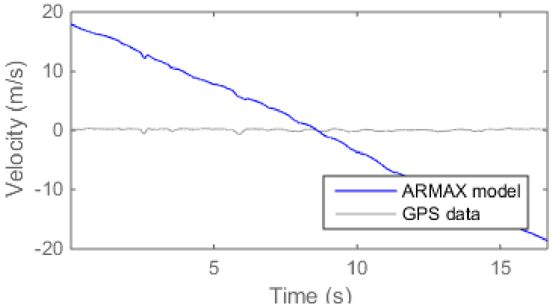
(b) Experiment 3, run 3
Fit: -17970



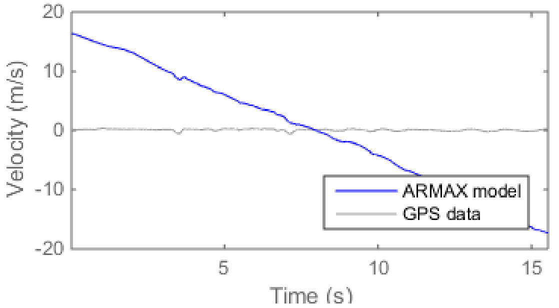
(c) Experiment 3, run 4
Fit: -30860



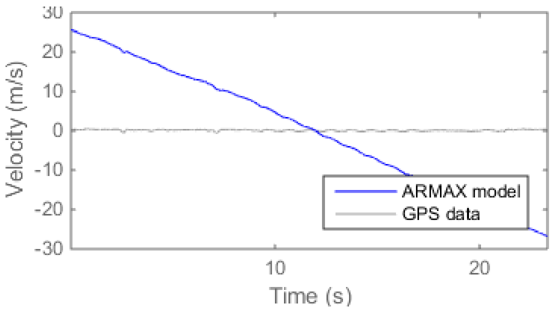
(d) Experiment 3, run 5
Fit: -17210



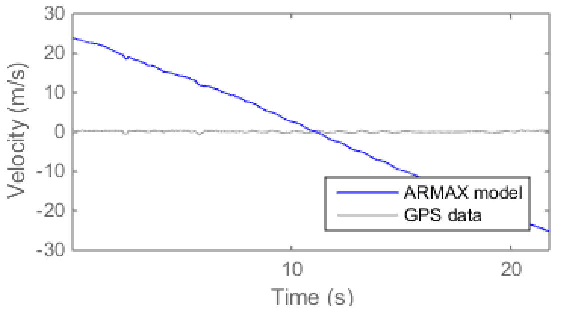
(e) Experiment 4, run 2
Fit: -6301



(f) Experiment 4, run 3
Fit: -6051

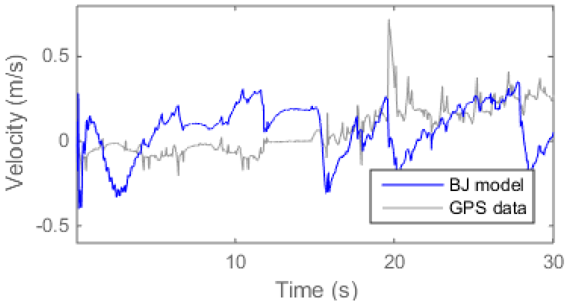


(g) Experiment 4, run 4
Fit: -9694

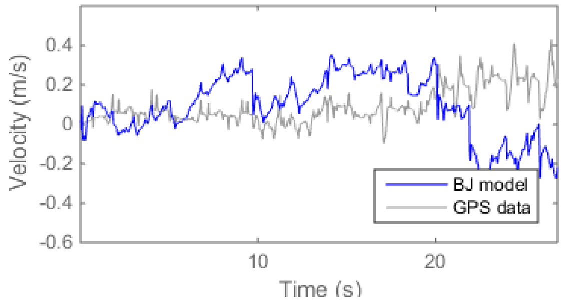


(h) Experiment 4, run 5
Fit: -8661

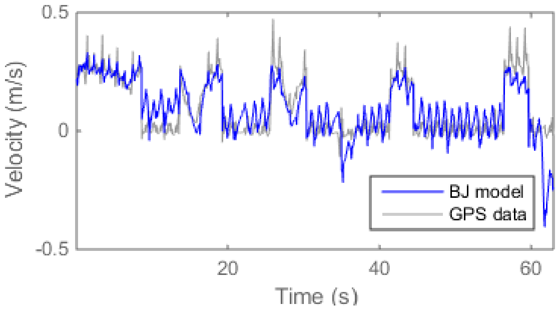
Figure 59: Measured and estimated outputs for remaining part of validation set of the estimation ARMAX model on Experiments 1 and 2



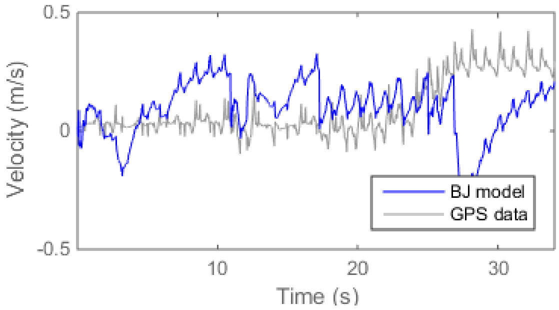
(a) Experiment 3, run 2
Fit: −44.77



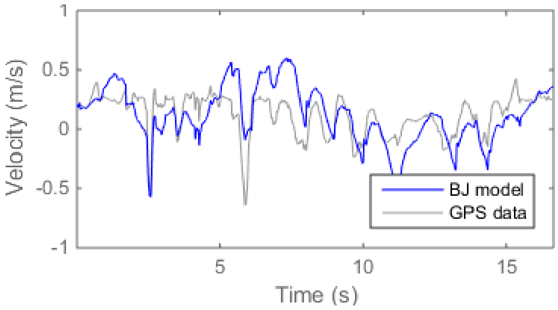
(b) Experiment 3, run 3
Fit: −144.2



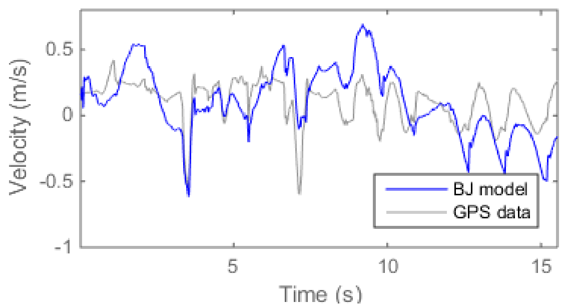
(c) Experiment 3, run 4
Fit: 39.34



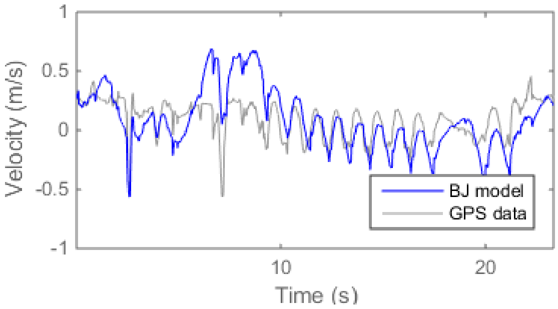
(d) Experiment 3, run 5
Fit: −61.65



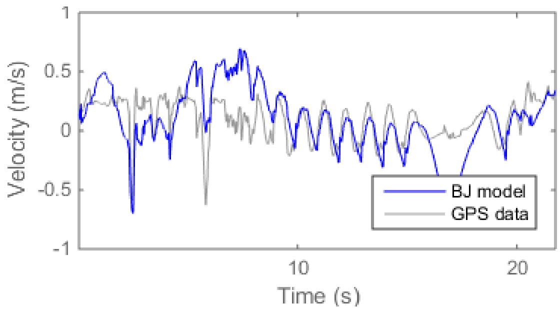
(e) Experiment 4, run 2
Fit: −17.16



(f) Experiment 4, run 3
Fit: −40.56

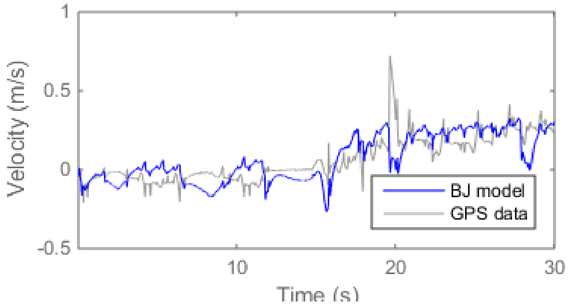


(g) Experiment 4, run 4
Fit: −32.36

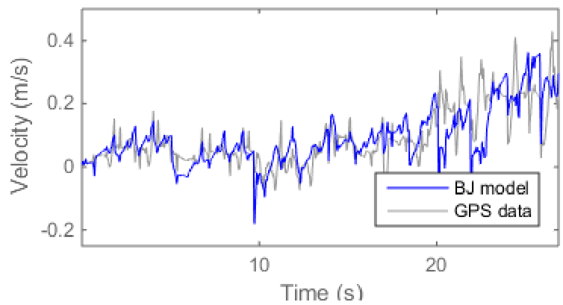


(h) Experiment 4, run 5
Fit: −42.47

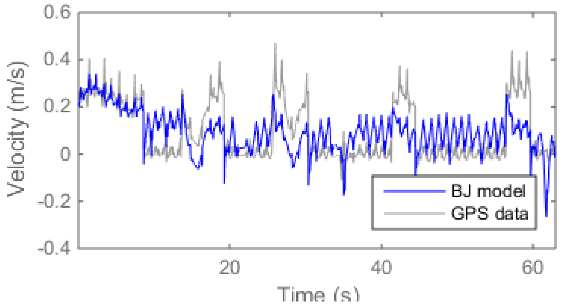
Figure 60: Measured and estimated outputs for remaining part of validation set of the estimation BJ model 1 on Experiments 1 and 2



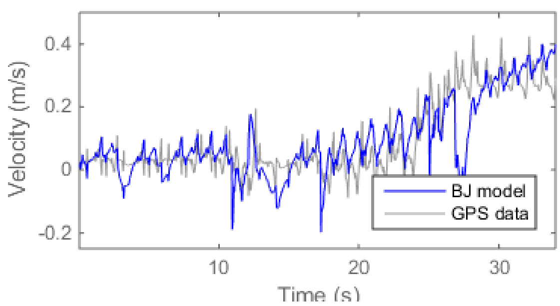
(a) Experiment 3, run 2
Fit: 20.75



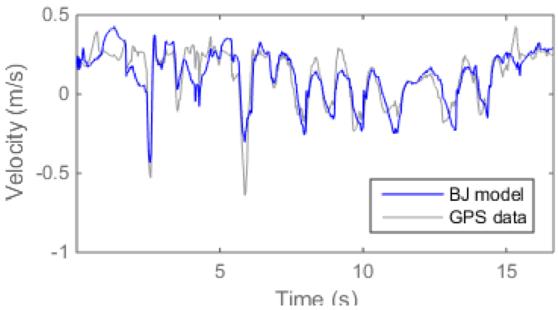
(b) Experiment 3, run 3
Fit: 21.55



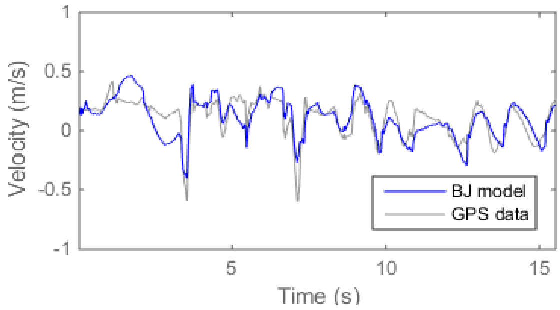
(c) Experiment 3, run 4
Fit: 16.26



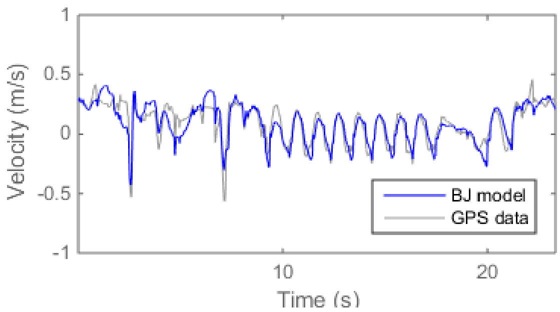
(d) Experiment 3, run 5
Fit: 29.41



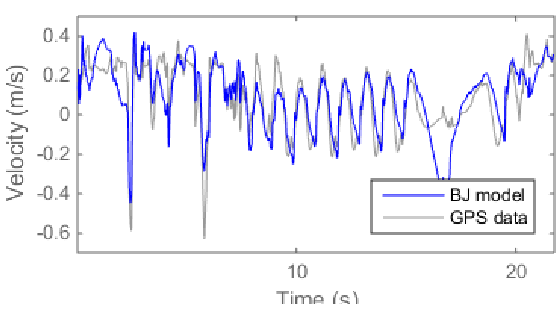
(e) Experiment 4, run 2
Fit: 44.19



(f) Experiment 4, run 3
Fit: 29.43

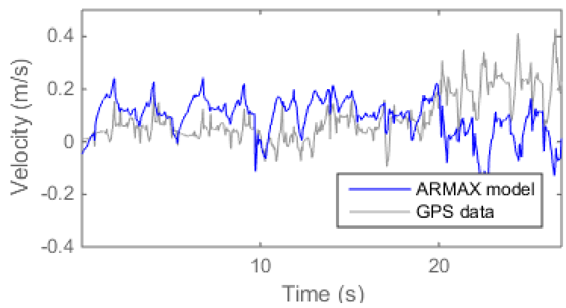


(g) Experiment 4, run 4
Fit: 46.37

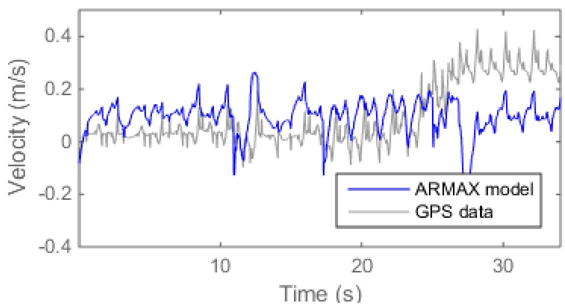


(h) Experiment 4, run 5
Fit: 34.89

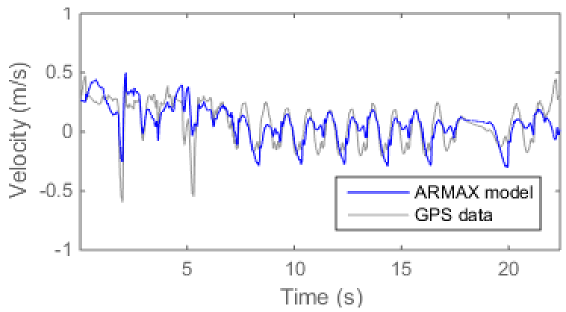
Figure 61: Measured and estimated outputs for remaining part of validation set of the estimation BJ model 2 on Experiments 1 and 2



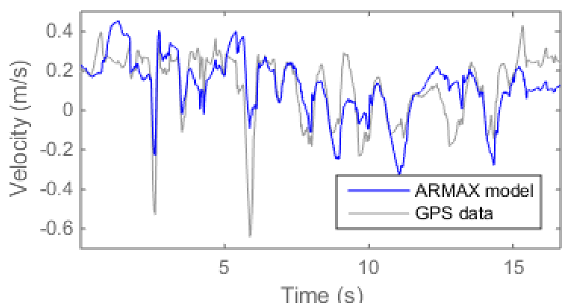
(a) Experiment 3, run 3
Fit: -53.28



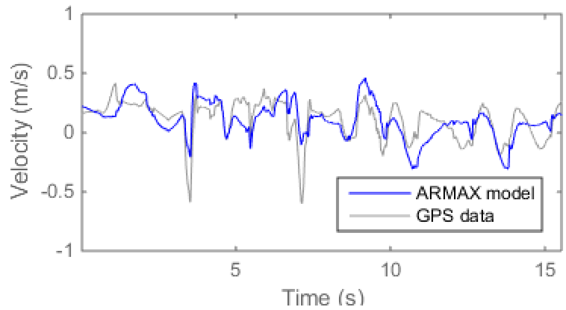
(b) Experiment 3, run 5
Fit: -19.35



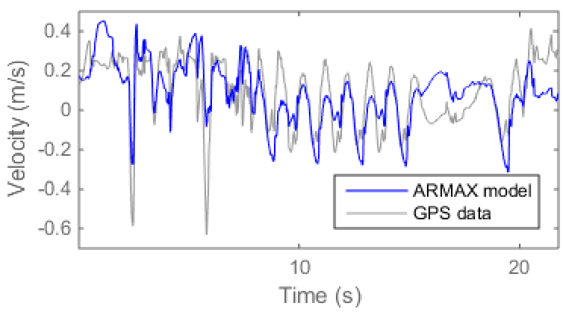
(c) Experiment 4, run 1
Fit: 26.63



(d) Experiment 4, run 2
Fit: 20.47

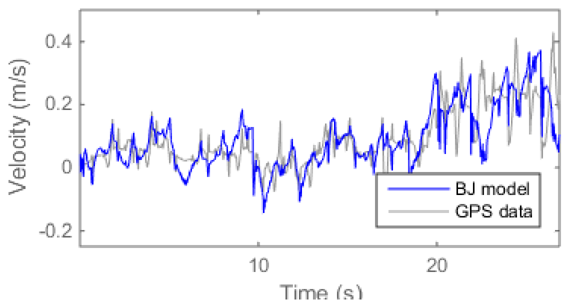


(e) Experiment 4, run 3
Fit: 11.15

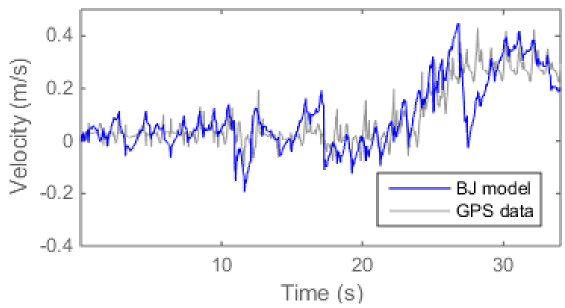


(f) Experiment 4, run 5
Fit: 19.3

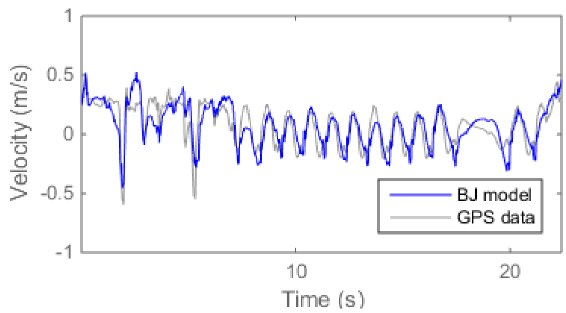
Figure 62: Measured and estimated outputs for remaining part of validation set of the estimation ARMAX model on Experiments 1, 2 and 3



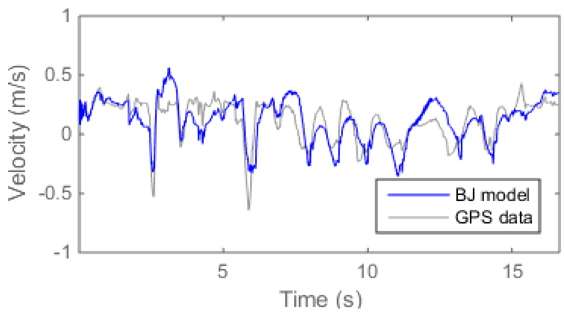
(a) Experiment 3, run 3
Fit: 17.39



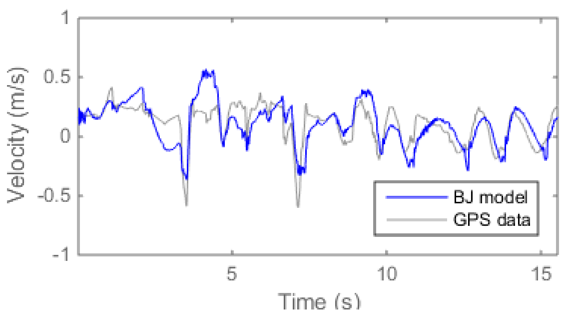
(b) Experiment 3, run 5
Fit: 35.17



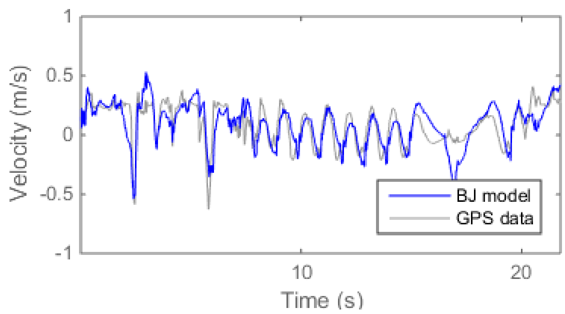
(c) Experiment 4, run 1
Fit: 34.81



(d) Experiment 4, run 2
Fit: 21.58



(e) Experiment 4, run 3
Fit: 14.2



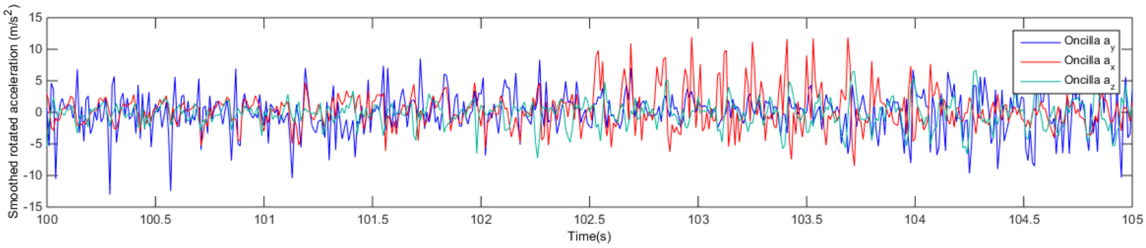
(f) Experiment 4, run 5
Fit: 27.03

Figure 63: Measured and estimated outputs for remaining part of validation set of the estimation BJ model on Experiments 1, 2 and 3

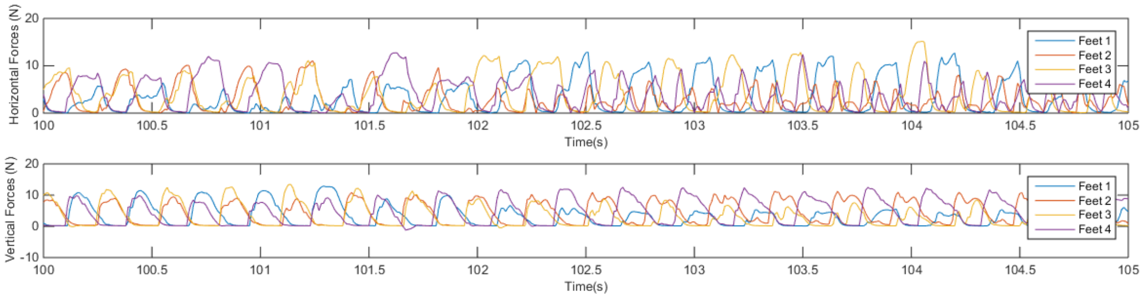
Appendix D : Supplement for Improved System Identification on Oncilla

Inputs and outputs for System Identification

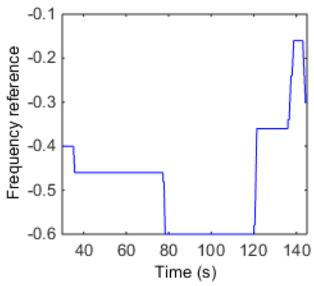
Figure 64 shows a part of inputs and output of Experiment 2 of improved Oncilla data for system identification.



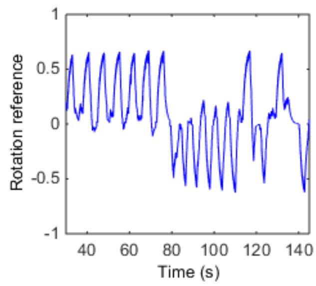
(a) Accelerations



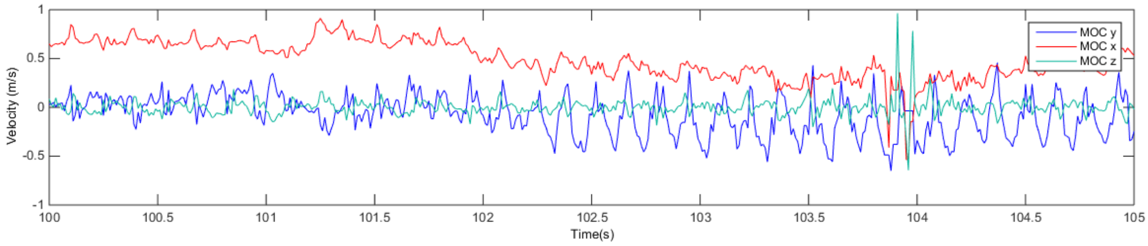
(b) Forces



(c) Frequency Reference



(d) Rotation Reference

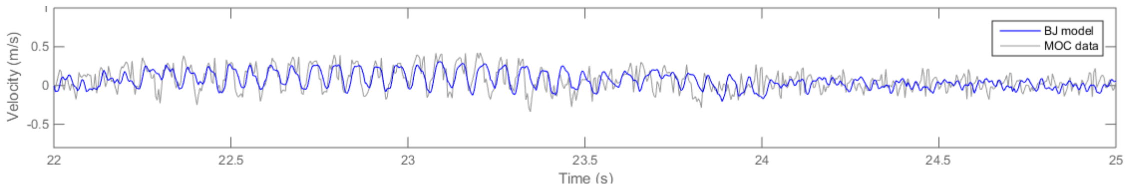


(e) Velocity from MOC

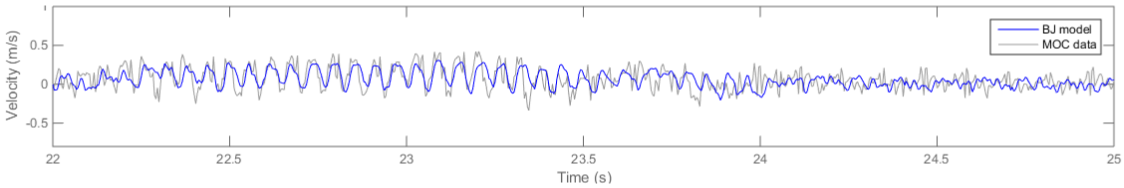
Figure 64: Parts of inputs and output of Experiment 2 for Improved System Identification on Oncilla

Models

Figure 65 shows the results of BJ model 2 trained on the second run and validated on the first one.



(a) 12 inputs
Fit: 7.769



(b) 13 inputs
Fit: 9.55

Figure 65: Measured and estimated outputs for the validation on run 1 of the estimation BJ model 2 on improved Oncilla data

Appendix E : Supplement for Detection of Failures on Pleurobot's Behaviour

Implementation of BJ model

The following Matlab function implements the computation of a single output from BJ models. The inputs of the function are:

- $nIter$, the number of iteration the function performs;
- u, the inputs of the models;
- $y0$, the previous outputs;
- B, C, D, F, nK, the coefficients of the model;
- $NoiseVariance$, $NoiseIntegration$, the parameters for the noise in the model.

```
function y = bjModel(nIter,u,y0,B,C,D,F,nK,NoiseVariance,NoiseIntegration)

        % Required lengths calculation
    [nu,nb] = size(B);
    nc = length(C);
    nd = length(D);
    [nu,nf] = size(F);
    augMat = max(max(max(max(nK)+nb,nc),nd+1),nf);

    % Initialisation zero
    % Augmentation of all matrices to have zeros elements as previous entries
    y = zeros(nIter+augMat,1);
    y(augMat+1-length(y0):augMat) = y0;
    u = [zeros(augMat,nu); u];
    e = zeros(nIter+augMat,1);

    % Initialisation of intermediate matrices
    w = zeros(nIter+augMat,nu);
    v = zeros(nIter+augMat,1);
```

```

    % Computation of the output y
    for n=1+augMat:nIter+augMat
        u_previous = flipud(u(n-nb-nK+1:n-nK,:));
        e_previous = flipud(e(n-nc:n-1));
        w_previous = flipud(w(n-nf+1:n-1,:));
        v_previous = flipud(v(n-nd+1:n-1));

        for i=1:nu
            w(n,i) = B(i,:)*u_previous(:,i) - F(i,2:nf)*w_previous(:,i);
        end

        % e(n) defined here as zero, this can be modified
        e(n) = 0;

        if true(NoiseIntegration)
            v(n) = C*e_previous - D(2:nd)*v_previous + D(1:nd-1)*v_previous;
        else
            v(n) = C*e_previous - D(2:nd)*v_previous;
        end

        y(n) = sum(w(n,:)) + v(n);

    end
end
```

The following Matlab code trains a BJ model on the training set zt, processes the coefficients so that they can be used by the function *bjModel* and uses this function to compute the output y from the input u. The parameters of training are those of BJ model on Experiments 1, 2 and 3 of Pleurobot simulation. The input u is a matrix, which columns are individual inputs of the model from old to new ones.

```
% Parameters for BJ model identification
nb = [1 1 1 5 3 5 3 5 3 5 3 1];
nc = 2;
nd = 2;
nf = [1 1 1 3 2 3 2 3 2 3 2 3];
nk = [1 1 1 4 2 4 2 4 2 4 2 3];

% System identification
bj_model = bj(zt,[nb nc nd  nf nk], Opt);

% Coefficients recording
[A_model,B_model,C_model,D_model,F_model] = polydata(bj_model);

% Transformation of coefficients to have square matrices with additional zeros
b_max = max(nb+nk);
f_max = max(nf+nk);

```

```
B = zeros(length(B_model),b_max);
F = zeros(length(F_model),f_max);

C = C_model{1,1};
D = D_model{1,1};

for i=1:length(B_model)
    for j=1:length(B_model{1,i})
        B(i,j) = B_model{1,i}(1,j);
    end
end

for i=1:length(F_model)
    for j=1:length(F_model{1,i})
        F(i,j) = F_model{1,i}(1,j);
    end
end

% Computation of the output y
y0 = 0;
y = bjModel(length(u),u,y0,B,C,D,F,nk,0,1);
y = y(length(y)-length(u)+1:length(y));
```