Modeling and System Identification of Pneumatic Robot

Akshit Kaplish

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Mechanical Engineering

University of Washington

2020

Reading Committee:

Emo Todorov, Chair

Sawyer B. Fuller

Steve Brunton

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Modeling and System Identification of Pneumatic Robot

Akshit Kaplish

Chair of the Supervisory Committee:
Professor Emo Todorov
Computer Science and Engineering

Pneumatic actuators can prove to be really useful in robotics as they have natural backdrivability, good strength to weight ratio and lot of other interesting properties. However, they are quite difficult to control owing to their non linearity and high latency due to air's compressability. One of the promising ways to make these actuators usable in advanced robotic tasks is to use model based control. Thus, this work is entirely focused to obtain a model for pneumatic actuators which has good predictability for different kinds of control objectives. In this work, first a physics model is derived through first principles and it's parameters are identified through classical regression techniques and carefully conducted experiments. Then in order to obtain even higher accuracy, neural network based data driven pure and hybrid models are proposed. Finally, all of these models are validated and evaluated by comparing their predictions to the actual hardware data, which is as diversified as possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

STATE: Mathematical form of information that can qualitatively explain system's behaviour for the purpose of control.

ROLLOUT: A trajectory of states and control inputs with respect to time, obtained by forward integrating the model for given horizon or time duration.

ORIFICE: A small opening between air filled volumes, which allows flow.

COMPLIANCE: The ability of a robot to back drive itself under an external force or disturbance. This is governed by the stiffness between the actuator and mechanical joint where the load acts.

ENCODER: Type of sensors, which can detect the change in displacement between two mechanically coupled bodies.

SEED: This effects the initialization of random number generator. If the same seed is chosen every time all the random variables' values will be the same.

COMPRESSABILITY: In reference to compressability of air, it is the property of air or gas that in significant and reasonable temperature and pressure range, the density of molecules can be varied by changing the volume in which the gas is trapped. This property is peculiar to gases compared to fluids or solids.

ACKNOWLEDGMENTS

I am extremely grateful to God for everything in my life. I am thankful to my parents, family and friends for their unconditional support and every effort they have put in for me.

I am grateful to my advisor Dr. Emo Todorov for giving me much awaited opportunity to work in his lab and making me a successful roboticist. He has always inspired me to think about robotics in uniquely brilliant ways. I'd always be thankful to him from the core of my heart for that.

I am thankful to my thesis committee members Dr. Sawyer Fuller and Dr. Steve Brunton for being part of my committee, giving insightful vital suggestions in my presentation and appreciating my work.

I am thankful to Svetoslav Kolev for helping me gain a skill set in electrical engineering and computer science, especially C++. He helped me a lot in conducting experiments, setting up hardware and making nearly-impossible things work. He has played a crucial role in the successful completion of this work.

I am thankful to Dr. Kendall Lowrey and Aravind, for teaching me everything about Reinforcement Learning, AI and helping me gain a huge insight into the world of Robotics. I am also grateful to Ben, Mottoya and Collin for their support.

I am thankful to Kate Gayle, for her unceasing support to help me progress through graduate school. I will forever be in debt of Department of Mechanical Engineering and University of Washington, for instilling the quality in me to achieve great professional success.

Lastly, I would say that I have spent the most wonderful time here in this university and owe so much to it. It has enriched me with a bagful of experience and knowledge. Thanks once again to all who helped me achieve this.

DEDICATION

To my late grandfathers - Mr. Baldev Raj Sharma and Mr. Shubh Kumar Kaplish.

Wish they were here.

Chapter 1

INTRODUCTION

1.1 Motivation

Pneumatic actuators can be highly useful for robotic applications, especially when it comes to dynamic tasks. In such tasks, both precise motion and force control equally play a crucial role in achieving robot's objective. These actuators have high strength to weight ratio, adjustable stiffness, and natural compliance. Moreover, they do not necessarily need complicated transmission drives between the actuator and robot's joint, thus making them immune to issues like back-lashing in gears. Also, the parallels that can be drawn between the working of a biological muscle and a pneumatic actuator, make them really fit in the context of bio-inspired robotics [1]. However, pneumatics are challenging systems to control.

They are highly non linear 3rd order systems, with dynamic timescales varying between 100 to 300 ms especially in the middle control ranges of a proportional valve [2]. So owing to compressability of the air, they're much slower than electric motors and hydraulics. This makes pneumatically actuated robots, perfect candidates for model based control techniques.

In model based control techniques, control commands are generated based upon a mathematical model of the system, which encodes the effects as defined above. Some of the common techniques are feedback linearization [3], MPC - iLQG [4], MPPI [5], etc. While using trajectory optimization methods, such information about system's physics can help make the control quite efficient. [4] In these techniques, the sparse impact of these physical effects on robot's objective can be predicted over a time horizon. Moreover, while doing backward propagation these effects can be incorporated as constraints and thus help achieve the optimal control.

However, a model is required in the first place to use such techniques and this is where things get difficult. Hence, this work is entirely focused on using physics based modeling techniques and

data - driven identification methods to get an accurate model of a pneumatically actuated robot.

1.2 Background

One of the most comprehensive works on pneumatic actuator modeling, [6] involves deriving the physics model from the first principles. This model is based out of ideal gas law, thin plate flow model [7], conservation of mass and energy, etc. Moreover, it also includes the delay in flow and pressure loss induced by the pipe connecting valve and the cylinder. In [8], a second order bi-polynomial model was fit to obtain the mass flow rate as a function of chamber pressure and control voltage. In [1], a simplified version of model in [6] is proposed. In [2], a completely parameterized model has been proposed, where primary objective of this model is to accurately predict the future states and secondary objective is easy integration and differentiation with respect to time. In [9], the model used is similar to one in [2] but MPC - iLQG was used to track desired pressure trajectories on a single degree of freedom pneumatic actuator for validation. In [10], Gaussian-processes based data driven approach was used to obtain a probabilistic forward transition model of a robot driven by pneumatic artificial muscles.

1.3 Main Contribution

In this thesis, we first derive a complete physics model for our pneumatically actuated robot based on first principles' model provided in [6] and robot's rigid body dynamics model given in [1]. After that, we fit different kinds of parametric mass flow rate functions and integrate it to the rest of the model. Then, we evaluate all of these physics based models by comparing their predictions against actual hardware data. Finally, as our primary contribution we propose neural network based black box identification techniques to further improve the predictions of our model. We validate our proposed techniques by comparing the resulting data driven models against one of the physics based models.

Chapter 2

MODELING AND SYSTEM IDENTIFICATION : AN OVERVIEW

Model based control is perhaps the most essential tool in making non linear systems achieve complicated control tasks. However, performance of this control is heavily dependant on how accurately a set of mathematical equations i.e. system model [11], predict the behaviour of the actual system. Usually, there are two steps in obtaining such information about a system :

1. **Modeling** - Use physics to formulate a set of equations and functions which can best explain the real world behaviour of the system.

2. **Identification** - Tune the unknown parameters in these equations such that their prediction or output match the actual hardware data.

2.1 Forward Model

Forward transition model of a dynamical system can be defined as a set of ordinary differential equations which given the current state x_t and control-input u_t predict $\dot{x_t}$ i.e. derivatives of that state.

$$\dot{x_t} = f_{ct}(\theta, x_t, u_t) \tag{2.1}$$

Here θ are set of parameters in above function that control the behaviour of this model.

$$x_{t+1} = f_d(\theta, x_t, u_t) = x_t + \Delta_t f_{ct}(\theta, x_t, u_t) \tag{2.2}$$

Alternatively in discrete time, forward model can be defined as a set of equations which given the current state x_t and control-input u_t predict x_{t+1} i.e. state at the next time step. This is usually

derived by forward integrating equation (2.1) ahead in time by single time step Δ_t. Thus, apart from the system's natural physics, this model is also dependant on the value of this time step.

2.2 Optimization Problem

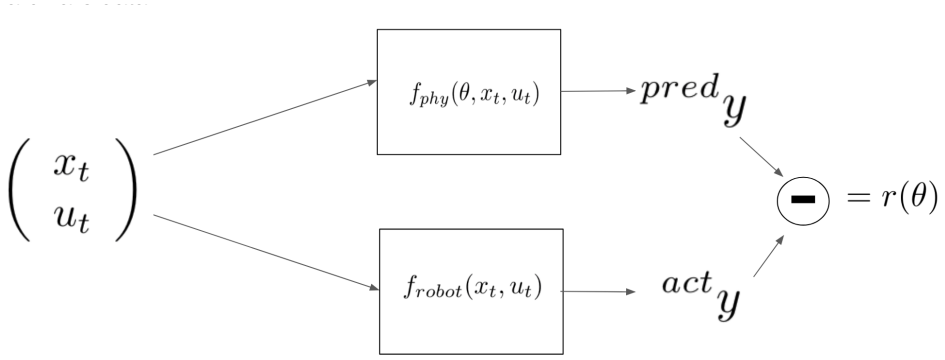

Figure 2.1: System Identification

In order to get the physics model to behave exactly like robot as shown in figure 2.1, the unknown θ parameters need to be tuned such that gap $r(\theta)$ between model's predictions ${}^{pred}y$ and the actual hardware data ${}^{act}y$ is minimum. Hence, search of such parameters can be formulated as the following optimization problem :

$$r_i(\theta) = {}^{pred}\, y_i(\theta) - {}^{act}\, y_i = f_{phy}(\theta, x_i, u_i) - {}^{act}\, y_i \tag{2.3}$$

$$\theta^* = \arg\min_{\theta} \sum r_i^T(\theta) r_i(\theta) \tag{2.4}$$

Here $x_i, u_i, {}^{act}\, y_i$ are current state, control input and derivative or future state respectively sampled on hardware, at a data-point indexed by i. ${}^{pred}y_i(\theta)$ is model's prediction and $r_i(\theta)$ is difference between actual and predicted value i.e. residual. In the parameter space, θ^* is the solution of the least square error fit of above regression problem.

In order to solve (2.4), 4 different kinds of algorithms can be used namely:

1. Solve the system of equations using pseudo inverse of regressor matrix [12], if the model is linearly dependant on the optimization parameters.

2. Use Gauss Newton algorithm [13]

3. Use stochastic gradient descent or its variants (ADAM, etc) [14]

4. Use derivative free methods like simplex methods [13] or CMA - ES[15].

2.3 Evaluation

There are multiple ways to check the predictability and accuracy of this model with optimized parameters θ^* :

1. Sample new test data on hardware and see how predictions compare against actual data.

2. Do trajectory rollout - at a given time t_0, record the complete state information and then forward integrate iteratively for given horizon :

$$x_j = f_d(\theta, u_{j-1}, f_d(\theta, u_{j-2}, ...f_d(\theta, u_0, x_0))) \tag{2.5}$$

$$\tau(t_0) = \{x_{N-1}, x_{N-2}, ..., x_j, ..., x_0\} \tag{2.6}$$

   Here $0 \leq j < N$ is the time index and N is the horizon length of the trajectory τ. The control commands u_j in future steps can be prerecorded from hardware data. This trajectory τ can then be compared against the actual hardware data along time.

3. Use model predictive control to achieve different kinds of control tasks with same model learnt above by changing cost functions for every different task. More the diversified the tasks are, stronger is model's predictability.

Chapter 3

PHYSICS FOR PNEUMATICALLY ACTUATED ROBOT

3.1 Robotic Platform

3.1.1 Pneumatic Actuation Mechanism

Pneumatic actuation works on the principle of force built up by the pressure applied by air molecules over some area. In a regular pneumatic cylinder, there are two chambers separated by a piston connected to the load i.e. entity that needs to be actuated. The difference in pressures in chamber 1 and 2, results in an imbalanced force across the piston, which then pushes or pulls the load. Each chamber is connected to a compressor or exhaust via a proportional valve [2]. This is shown in figure 3.1 (a).

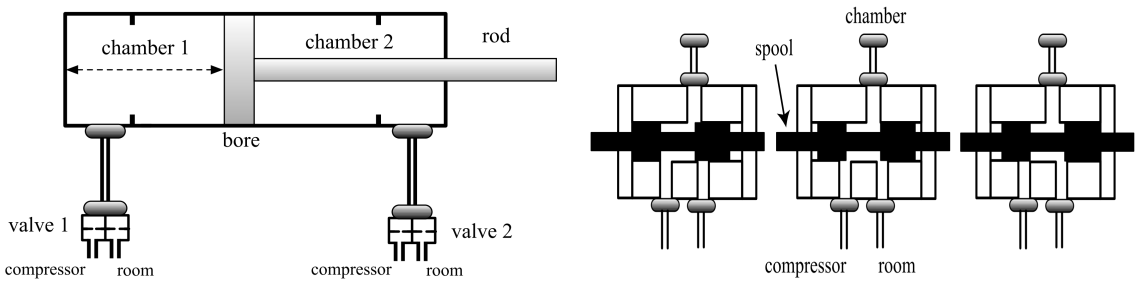

Figure 3.1: (a) Chamber Model (left) & (b) Spool Model (right)

Depending on voltage $0 \leq u_i \leq 10$ supplied to i^{th} valve, it moves a spool as shown in figure 3.1 (b). This varies the areas of two different orifices :

1. $a_{i,c}(u_i)$ - Area of the opening between compressor and i^{th} chamber. This port is open approximately above 5 V.

2. $a_{i,c}(u_i)$ - Area of the opening between exhaust (atmosphere in this case) and i^{th} chamber. This port is open approximately below 5 V.

Here i is the index of a chamber and it's valve. The movement of this spool regulates the amount of high pressure air flowing in or out of a given chamber.

3.1.2 2 DOF Robotic Arm

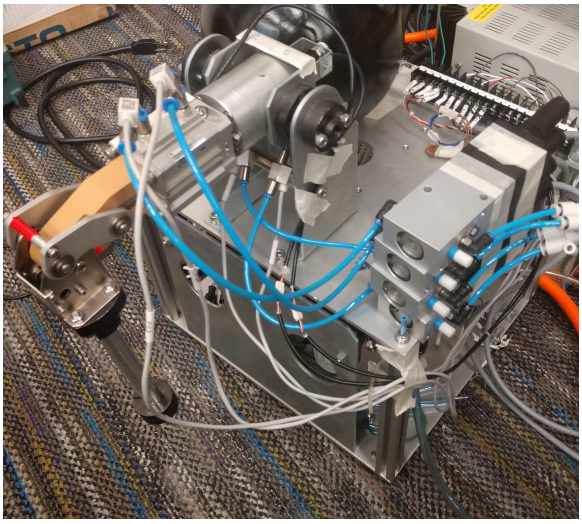

Figure 3.2: 2 DOF Pneumatic Robot Hardware

The robotic platform shown in figure 3.2 and used in this work is manufactured by Kokoro, Japan. It has two orthogonal rotational joints, where joint - 1 has direct drive rotary actuator and joint - 2 is driven by a linear actuator coupled with slider - crank mechanism [16]. Both the cylinders - 1 and 2, are manufactured by SMC and controlled by four 5/3 proportional valves used as 3/2, made by FESTO. As shown in figure 3.3, each chamber of each cylinder is exclusively connected to a single valve, thus the pressure in both chambers can be independently controlled, resulting in tunable compliance. Each chamber has solid state pressure sensors made by SMC and each joint has a position encoder. All the sensors and controllers are of analogue type, thus hardware/software interfacing is done with NIDAQ I/O boards. All the input data is read at a

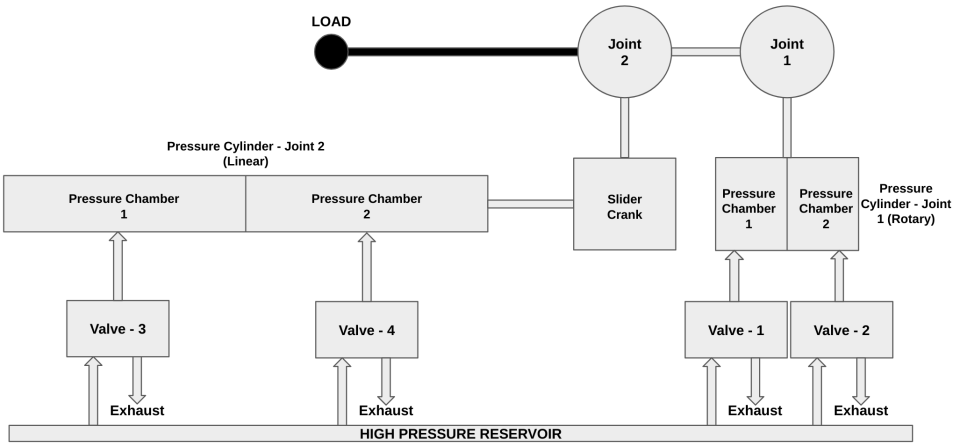

Figure 3.3: 2 DOF Pneumatic Robot Schematic

frequency of about 10 KHz and output data is written at same frequency but with a lag of 2 ms. For the sake of simplicity, the entire modeling and system ID of pneumatics has been restricted to 1 degree of freedom system which only includes the second joint. However, we include both joints while collecting data for training and evaluation, to model the dynamic effects due to coupling such as centrifugal forces [3].

3.2 Physical Model

3.2.1 Thin Plate flow model

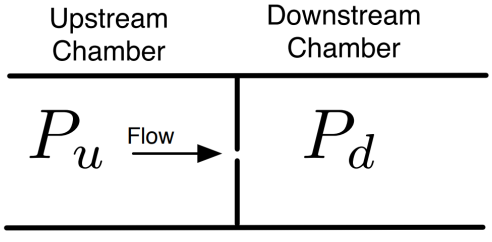

Figure 3.4: Thin plate flow model

This model relates the mass flow rate across an orifice from high or upstream pressure region P_u to low or downstream pressure region P_d, as shown in figure 3.4. Some of the assumptions for this flow function is that orifice plate separating the two regions is thin, gas has near to ideal behaviour and the flow is isentropic (low heat losses and constant entropy) [2]. If the ratio $P_d/P_u \leq P_{cr}$ (P_{cr} is critical ratio) then flow gains high speed i.e. choked flow as described in equation (3.2) and starts varying linearly with respect to P_u.

$$f_1(P_u, P_d) = \alpha P_u \sqrt{(\frac{P_d}{P_u})^{\frac{2}{k}} - (\frac{P_d}{P_u})^{\frac{k+1}{k}}} \tag{3.1}$$

$$f_2(P_u, P_d) = \beta P_u \tag{3.2}$$

Here k, β, α are known constants and their values are given in table 3.2.

3.2.2 Chamber Pressure Model

This model relates the rate of change of pressure $\dot{p_i}$ to mass flow rate $\dot{m_i}$, volume $V_i(q_2)$, pressure p_i and rate of change volume $\dot{V_i}(\dot{q_2}, q_2)$, of the i_{th} chamber :

$$\dot{p_i} = k\frac{\dot{V_i}(\dot{q_2}, q_2)p_i}{V_i(q_2)} + k\frac{RT}{V_i(q_2)}\dot{m_i} \tag{3.3}$$

Here R and T are known constants, their values are given in table 3.2. q_2 and $\dot{q_2}$ are position and velocity of second joint respectively.

3.2.3 Mass Flow Rate Model

This model relates the mass flow rate $\dot{m_i}$ of the i_{th} chamber to the areas $a_{i,c}(u_i)$ and $a_{i,r}(u_i)$ of the orifices connecting to compressor and exhaust, respectively.

$$\dot{m_i} = k\frac{RT}{V_i(q_2)}(a_{i,c}(u_i)f_j(P_c, p_i) - a_{i,r}(u_i)f_j(p_i, P_r)) \tag{3.4}$$

Here P_c is compressor pressure and P_r is exhaust pressure. f_j is as defined in equations (3.1) and (3.2). If $\frac{P_d}{P_u} > 0.528 \approx P_{cr}$, then $j = 1$ else $j = 2$.

3.2.4 2 - Link Arm Dynamics Model

Inverse dynamics for two link robotic arm are derived using Euler-Lagrangian method for multi-body dynamics [3] as in [1]:

$$\tau = M\ddot{\vec{q}} + S(\vec{q},\dot{\vec{q}}) + G(\vec{q}) + B(\dot{\vec{q}}) \tag{3.5}$$

Here M is the inertia matrix, S coriolis and centrifugal force, G is the gravity matrix and B accounts for damping and friction effects. τ are torque values at the joints. $\vec{q}$ is a vector of joint angles.

In order to identify the parameters of this model, the same technique is used as in [1] where the difference between actual and predicted τ act as residuals. The data $\{\vec{q},\dot{\vec{q}},\ddot{\vec{q}}\}$ was collected by tracking multiple frequency varying and phase shifted sine waves using a simple proportional controller that drives the robot with a poor accuracy. The entire data set is about three minutes long, with each minute long desired wave chosen with randomized phase and initial frequency. Also assuming the difference in bore areas is small, the torques ${}^{act}\tau$ were approximated as pressure difference between the two chambers. As this particular model is linearly dependant on it's parameters, this problem was simply solved using pseudo inverse of regression matrix [12]. The test results of this regression problem are presented in table 3.1 and figure 3.5.

Joint ID	Mean Error	Variance	Max Error
2	0.015361	0.000173	0.097441
1	0.026679	0.000325	0.108453

Table 3.1: Error in predictions for pressure difference (MPa)

Once identified, τ was replaced with $\Delta p = p_1 - p_2$, the pressure difference between two adjacent chambers - 1 and 2. Finally, the above inverse dynamics equation (3.5) was inverted back to obtain a forward model :

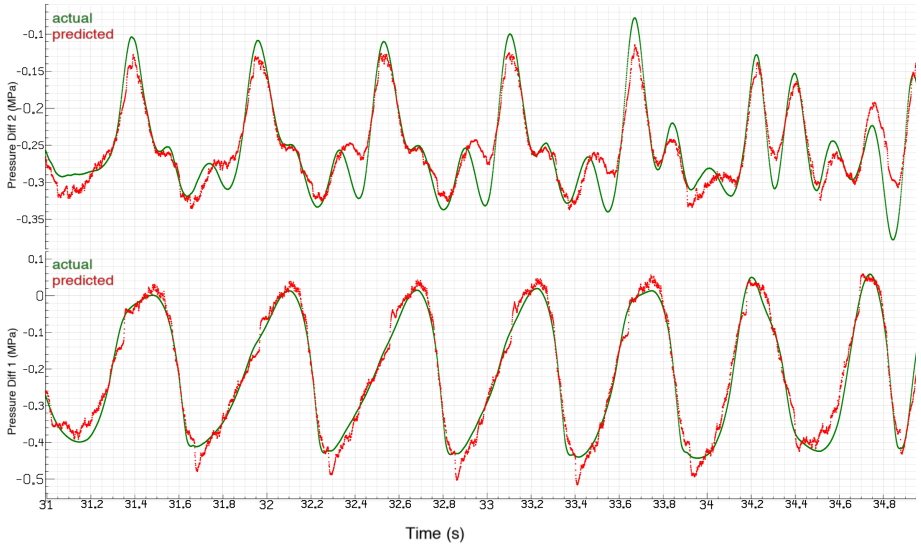

Figure 3.5: Comparison of predicted and actual pressure difference values for joint - 2 (top) and 1 (bottom)

$$\ddot{\vec{q}} = M^{-1}(\vec{q})(\Delta p - S(\vec{q},\dot{\vec{q}}) - G(\vec{q}) - B(\dot{\vec{q}})) \tag{3.6}$$

3.2.5 Coupling Model

This model relates the change in volume of the cylinder to the joint value for the second joint (as stated earlier, modeling for first joint is ignored for now). The slider crank mechanism between this second joint (crank) and piston-rod (slider) of the second pneumatic cylinder can be modelled as follows [16]:

$$\dot{V}_1(\dot{q}_2, q_2) = -A_1(\dot{q}_2 r sin(q_2) + \frac{\dot{q}_2(r^2 sin(q_2)cos(q_2))}{\sqrt{l^2 - r^2 sin^2(q_2)}}) \tag{3.7}$$

$$\dot{V}_2(\dot{q}_2, q_2) = A_2(\dot{q}_2 r sin(q_2) + \frac{\dot{q}_2(r^2 sin(q_2)cos(q_2))}{\sqrt{l^2 - r^2 sin^2(q_2)}}) \tag{3.8}$$

$$V_1(q_2) = A_1(rcos(q_2) + \sqrt{l^2 - r^2 sin^2(q_2)}) + D_1 \tag{3.9}$$

$$V_2(q_2) = A_2(L - rcos(q_2) + \sqrt{l^2 - r^2 sin^2(q_2)}) + D_2 \tag{3.10}$$

Here A_i is bore area, D_i is dead volume of i^{th} chamber, L is the stroke length, r is radius of the connecting rod and l is the length of the crank. The values for these known constants are given in table 3.3. $V_1(q_2)$ and $V_2(q_2)$ are volumes of chambers - 1 and 2 respectively, as functions of q_2 joint position of the second joint.

3.3 Area Model Regression

In order to infer orifice - areas $a_{i,c}(u_i)$ and $a_{i,r}(u_i)$ as functions of control command u_i sent to i^{th} valve, 3 different types of parametric functions were fit. For carrying out regression for each one of these, the data was collected on volume locked assembly (where i^{th} chamber was expanded out to it's max volume). Therefore, equation (3.3) can be reduced to :

$$\dot{p_i} = k\frac{RT}{V_{i,max}}(a_{i,c}(u_i)f_j(P_c, p_i) - a_{i,r}(u_i)f_j(p_i, P_r)) \tag{3.11}$$

In this regression problem, residual is the difference between predicted and actual pressure velocity, $\dot{p_i}$. For collecting the diversified data, P_c, P_r, p_i are recorded through pressure sensors for each voltage u_i (discretized in the steps of 0.2 ranging from 0.2 to 10 Volts). Steps for data collection are as follows:

1. The air in chamber is first completely exhausted (if target $u_i > 5$ Volts) or filled (if target $u_i < 5$ Volts).

2. Step signal from either 0 or 10 to the corresponding target voltage u_i is given. This is kept fixed.

3. After a transient period of about 10 ms, data is recorded until the pressure readings saturate i.e. air completely exhausts or fills the chamber.

As a result, a mapping between u_i and lumped area parameters $k\frac{RT}{V_{i,max}}a_{i,c}$ and $k\frac{RT}{V_{i,max}}a_{i,r}$ is obtained, through regression over dataset defined above.

3.3.1 Area Model - 1

In this case, the information of lumped area for every given voltage (discretized in the steps of 0.2 ranging from 0.2 to 10 Volts) is maintained as a look up table. When a value of voltage is queried, we use the two nearest voltages in the table to linearly interpolate the lumped area. For every single voltage value in this table, a separate linear regression problem is solved. The identified function is plotted with respect to control commands (Volts) in fig (3.6).

3.3.2 Area Model - 2

Here, the lumped area is simply assumed to be a quadratic function of control voltage :

$$a_{i,c}(u_i) =^c \theta_1 +^c \theta_2 u_i +^c \theta_3 u_i^2 \tag{3.12}$$

$$a_{i,r}(u_i) =^r \theta_1 +^r \theta_2 u_i +^r \theta_3 u_i^2 \tag{3.13}$$

This is solved as a combined linear regression problem for all the data points. Here, ${}^c\theta_{i=1,2,3}$ and ${}^r\theta_{i=1,2,3}$ are the parameters. The identified function is plotted with respect to control commands (Volts) in fig (3.7).

3.3.3 Area Model - 3

As mentioned in [2], $smax(x)$ function (smoothed $max(x,0)$) is used to represent the area model. The parameters here L_r, L_c, U_r, U_c, B are optimized by solving a combined non linear regression problem for all the data points :

$$smax(x) = (\sqrt{x^2+1}+x)/2 \tag{3.14}$$

$$a_{i,c}(u_i) = L_c + smax((u_i - U_c)B - L_c) \tag{3.15}$$

$$a_{i,r}(u_i) = L_r + smax((U_r - u_i)B - L_r) \tag{3.16}$$

Here L_r, L_c are minimum leakage areas of the exhaust and compressor ports, respectively. U_r, U_c are respective control voltages beyond which the orifice areas stop changing i.e. areas are open only due to leakage. B is the proportionality constant relating voltage to area. The identified function is plotted with respect to control commands (Volts) in fig (3.8).

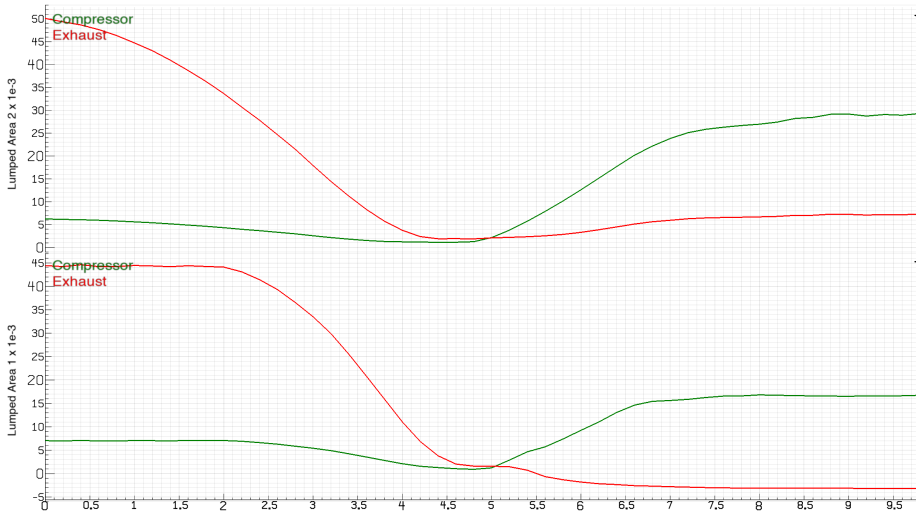

Figure 3.6: Lumped Area for Valve 1 & 2 ($JKg^{-1}m^{-1}$) v/s Control (Volts) for Model I

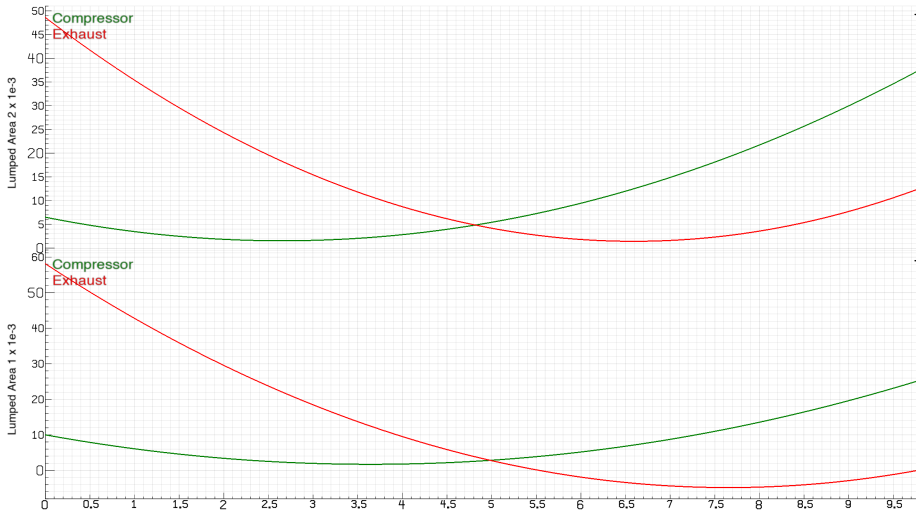

Figure 3.7: Lumped Area for Valve 1 & 2 ($JKg^{-1}m^{-1}$) v/s Control (Volts) for Model II

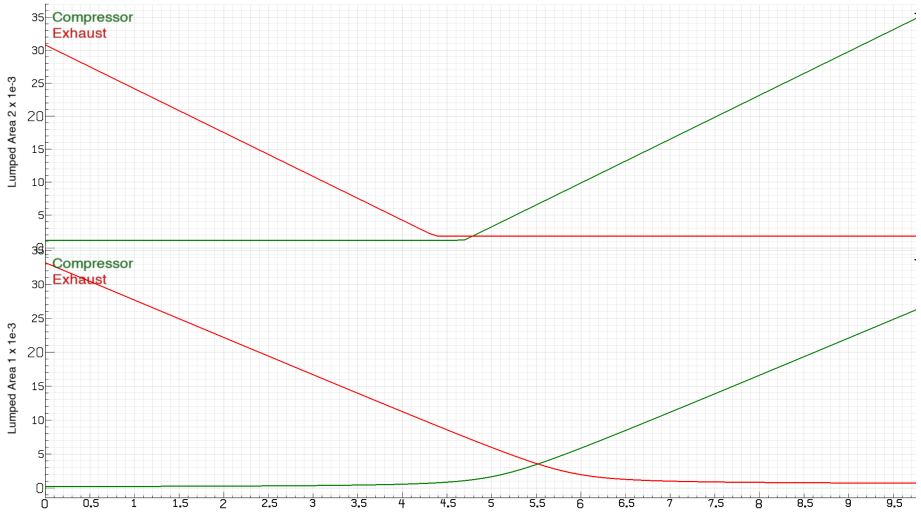

Figure 3.8: Lumped Area for Valve 1 & 2 ($JKg^{-1}m^{-1}$) v/s Control (Volts) for Model III

Constants	α	β	k	R ($JKg^{-1}K^{-1}$)	T (K)
Values	0.1124	0.029	1.4	287.058	298

Table 3.2: Values for known constants used in pressure dynamics

Constants	$A_1(mm^2)$	$A_2(mm^2)$	$D_1(mm^3)$	$D_2(mm^3)$	$l(mm)$	$r(mm)$
Values	804	603	8	6	40	15

Table 3.3: Values for known constants used in coupling model

Chapter 4

EVALUATION OF PHYSICS BASED MODELS

In order to evaluate accuracy of physics based models identified in previous chapter, the first two out of three tests as discussed in section 2.3 are carried out - derivatives and roll outs comparison. **State Space** (x_t) is $\{p_1, p_2, q_2, \dot{q}_2\}$ where p_1, p_2 are pressure in chamber 1 and 2. And q_2, $\dot{q}_2$ are position and velocity of joint 2. **Control Space** (u_t) is $\{u_1, u_2, q_1, \dot{q}_1\}$ where u_1, u_2 are valve commands for chamber 1 and 2 and q_1, $\dot{q}_1$ are joint position and velocity of joint 1. **Predictions** (${}^{pred}y$) include $\{\dot{p}_1, \dot{p}_2, \ddot{q}_2\}$. The system identification and evaluation is only carried out for joint - 2, but q_1, $\dot{q}_1$ are also included in control space to model dynamic effects due to coupling between two joints.

4.1 Hardware data collection

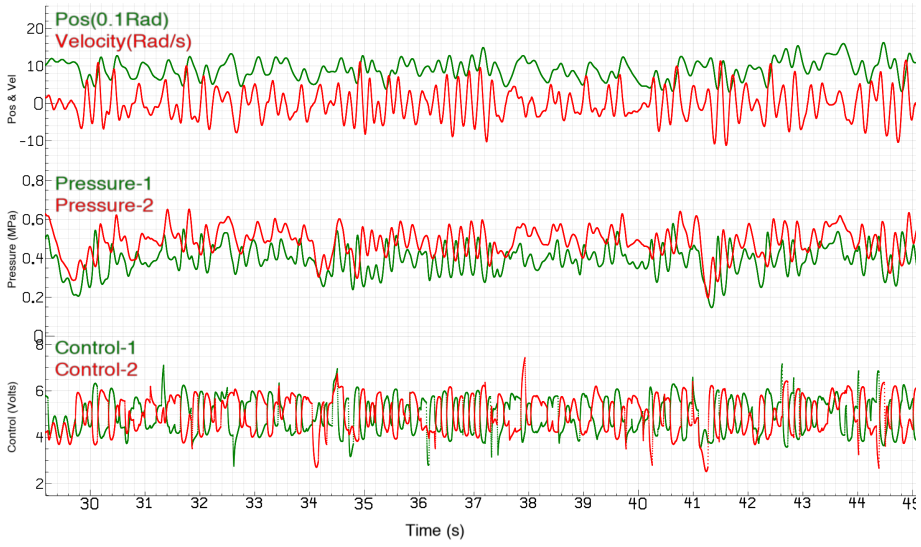

Figure 4.1: Type - I Hardware Data

Figure 4.2: Type - II Hardware Data

In order to strengthen the validity of these testing techniques, data was collected in three different ways to obtain rich and diversified hardware data :

1. **Type - I** : A set of joint positions is randomly sampled out of uniform distribution centered at the middle and spanning approx 60 % of the physical range. This acts as the initial state for the robot. Then for each valve, amplitude U (between 3 and 7 volts) and phase ϕ (between 0 and 2π) are randomly sampled, which define the control signal $u(t) = Usin(\omega t + \phi)$ where ω is approximately around 4π rad/s. This is fed to the robot until it is about to hit a joint limit or achieves a joint velocity over 10 rad/s. The controller is then switched to a simple PD scheme which pulls the robot back to the safe sphere. This whole process is repeated for a given amount of time and a part of it is shown in fig 4.1.

2. **Type - II** : A hand driven trajectory is first recorded by any human and then replayed back by the PD controller. A part of it is shown in fig 4.2.

3. **Type - III** : For each joint, a amplitude X and a phase ϕ (between 0 and 2π) are randomly sampled, which define the desired position signal ${}^{d}x(t) = Xsin(\omega t + \phi)$, here ω is slowly

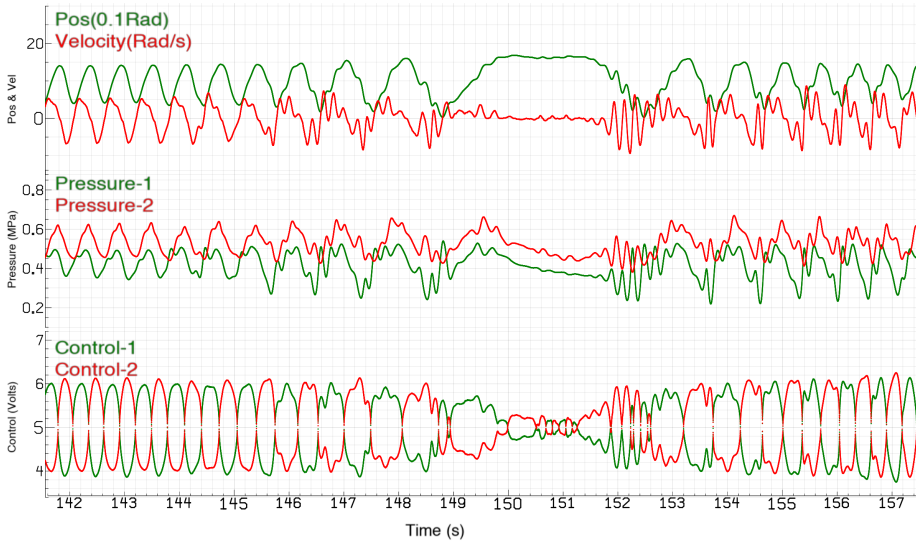

Figure 4.3: Type - III Hardware Data

varied with time. This wave is then tracked by the PD controller, as represented is shown in fig 4.3.

Here, position tracking PD controller is hand tuned and has gains $K_p = 10$, $K_d = 0.1$. This controller is not very accurate and has oscillatory response for some desired signals. However, it is good enough to collect data to validate the models proposed in this work.

4.2 Results

The results here include evaluation of error between predicted and actual hardware values over the entire three minute long test dataset. Test data consists of a minute long type - I & type - III data, randomly sampled with new seed. In addition, it also includes a minute long hand recorded trajectory i.e. type - II data. Table 4.1, represent that model - II does the best job in predicting pressure velocity in chamber - 1, whereas table 4.2 shows that model - I predicts the pressure velocity in chamber - 2 better. Also, table 4.3 shows that all three models have no effect on acceleration prediction as it's purely dependant on current pressure differences in both cylinders, through robot's rigid body dynamics model as described in equation (3.6).

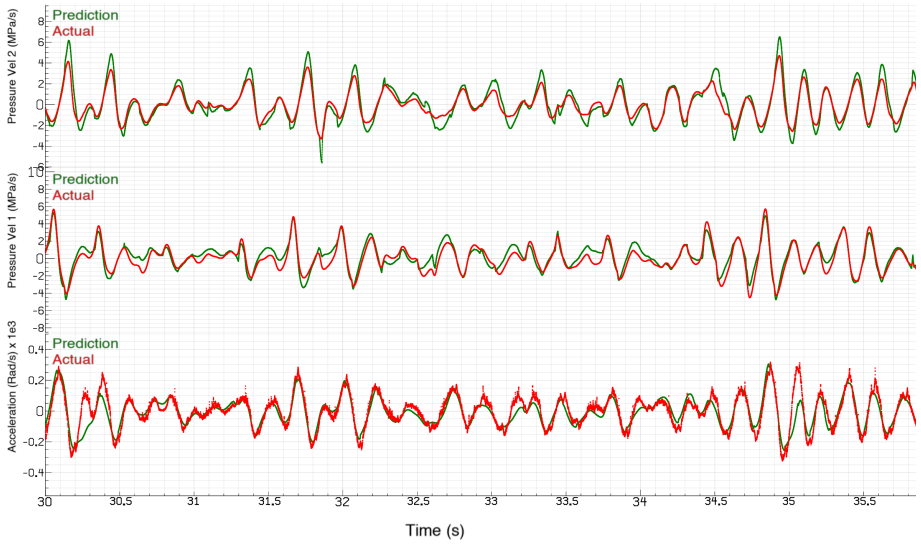

Figure 4.4: Comparison of predicted and actual derivatives for Model I

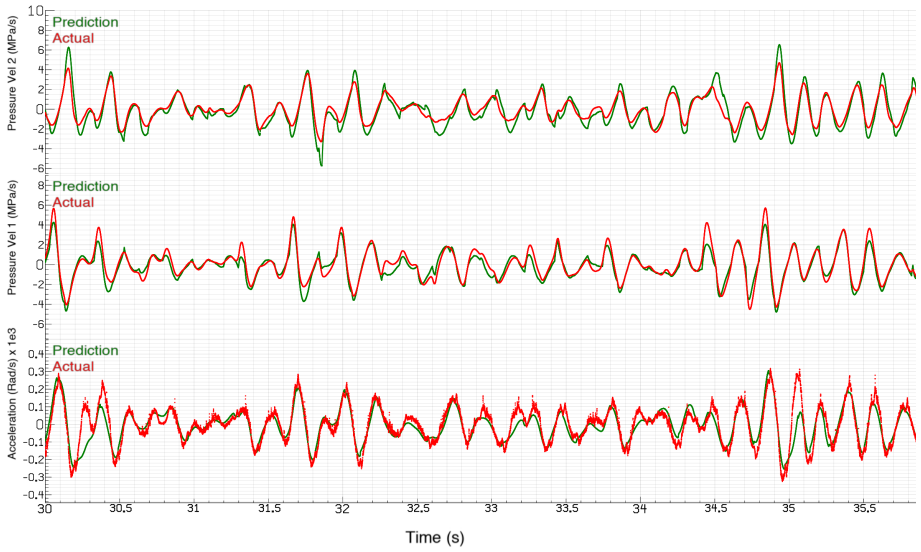

Figure 4.5: Comparison of predicted and actual derivatives for Model II

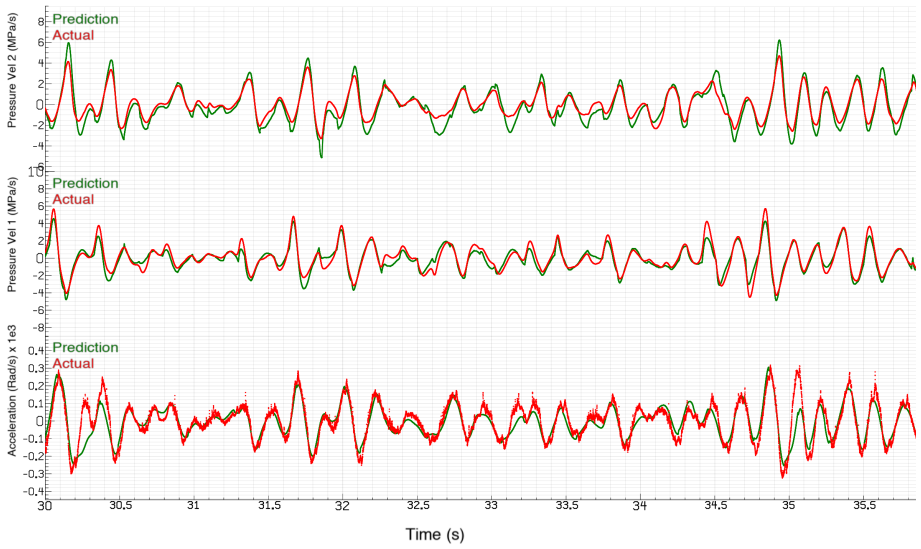

Figure 4.6: Comparison of predicted and actual derivatives for Model III

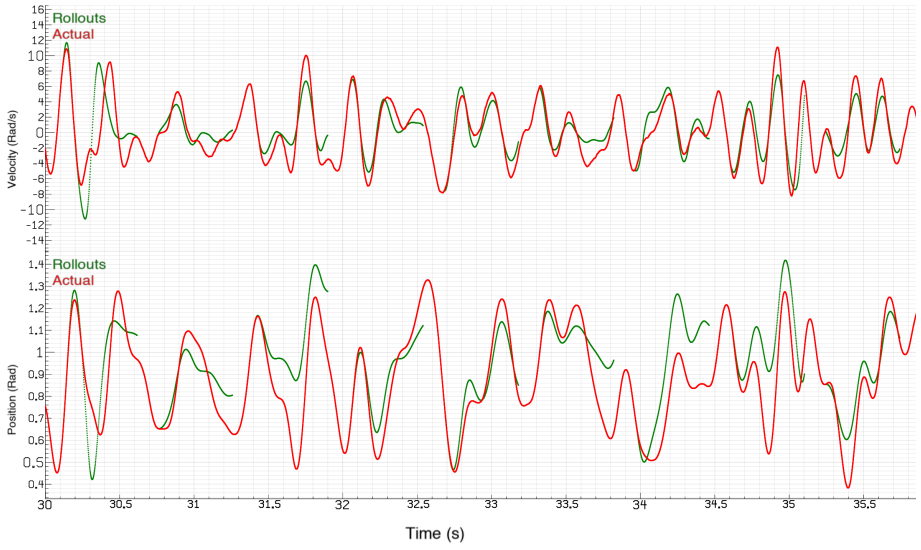

Figure 4.7: Comparison of kinematic-rollouts (position & velocity) and actual data for Model I

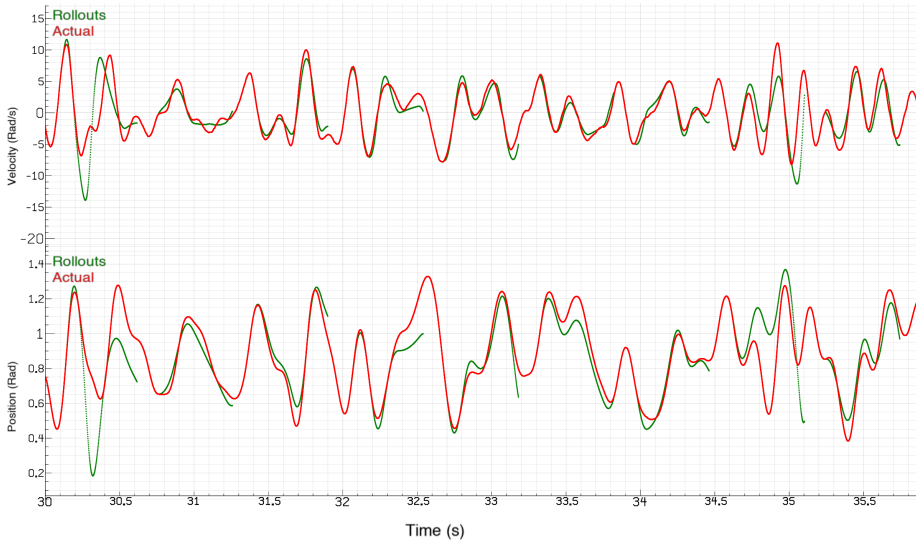

Figure 4.8: Comparison of kinematic-rollouts (position & velocity) and actual data for Model II

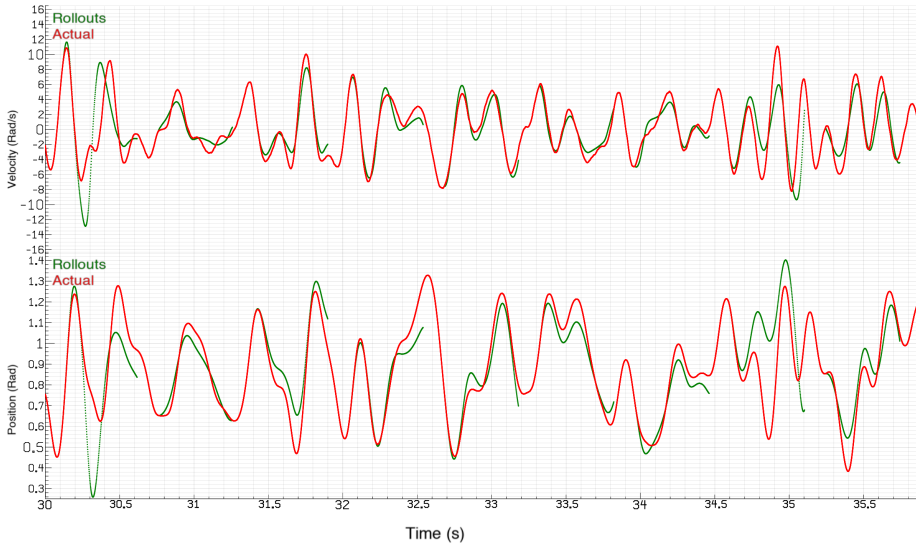

Figure 4.9: Comparison of kinematic-rollouts (position & velocity) and actual data for Model III

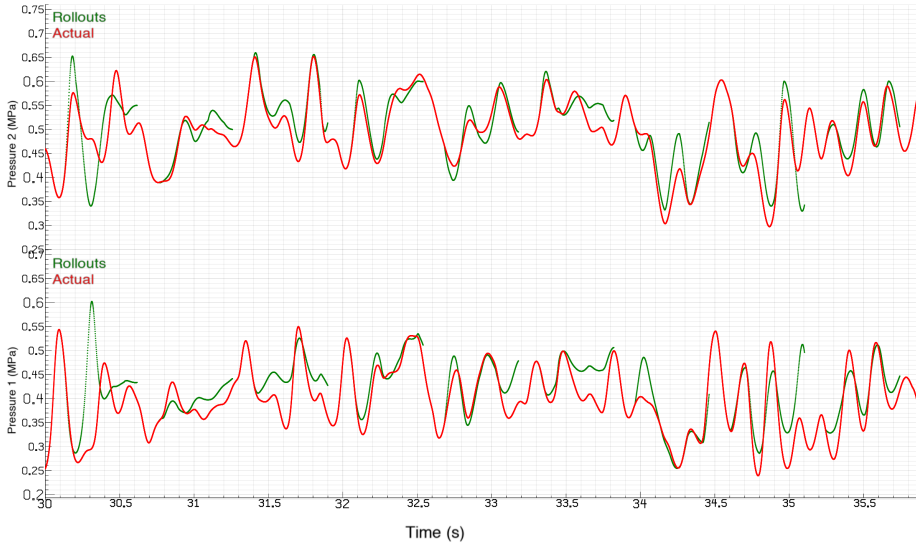

Figure 4.10: Comparison of pressure-rollouts (pressure - 1 & 2) and actual data for Model I

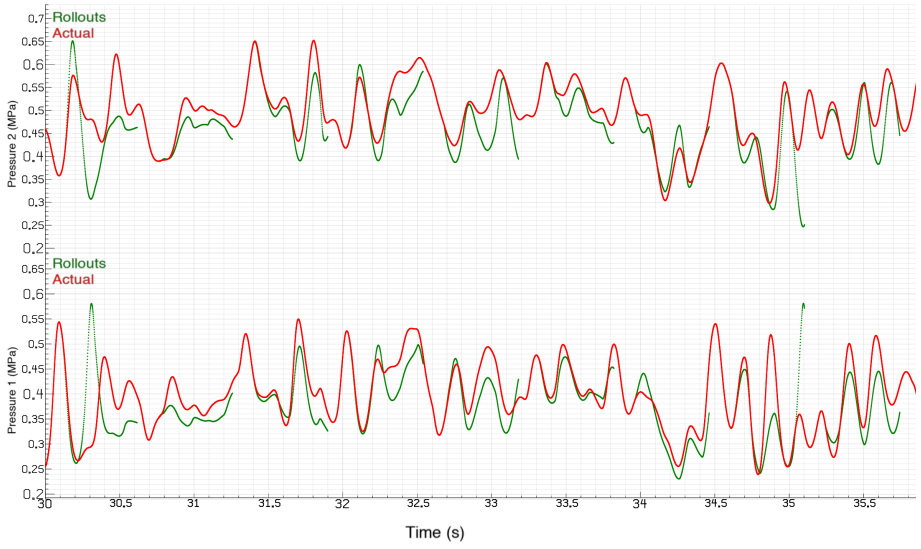

Figure 4.11: Comparison of pressure-rollouts (pressure - 1 & 2) and actual data for Model II

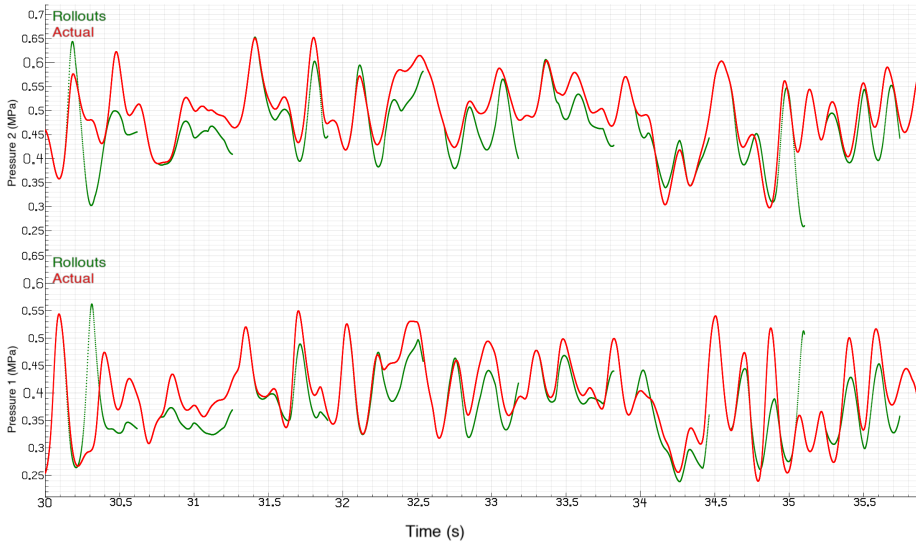

Figure 4.12: Comparison of pressure-rollouts (pressure - 1 & 2) and actual data for Model III

Physics Models	Mean Error	Variance	Max Error
Model - I	0.414935	0.158519	2.588980
Model - II	0.320622	0.118937	3.121290
Model - III	0.325669	0.116033	2.844730

Table 4.1: Error in predictions for pressure velocity - 1 (MPa/s)

Physics Models	Mean Error	Variance	Max Error
Model - I	0.348157	0.158430	3.831720
Model - II	0.589837	0.280684	3.923180
Model - III	0.448592	0.172663	3.237680

Table 4.2: Error in predictions for pressure velocity - 2 (MPa/s)

Physics Models	Mean Error	Variance	Max Error
Model - I	28.793105	789.228292	816.597000
Model - II	28.793105	789.228292	816.597000
Model - III	28.793105	789.228292	816.597000

Table 4.3: Error in predictions for acceleration (Rad/s^2)

Physics Models	Mean Error	Variance	Max Error
Model - I	0.027331	0.001149	0.306210
Model - II	0.029390	0.003371	0.285260
Model - III	0.028906	0.003399	0.265881

Table 4.4: Error in rollouts for pressure in chamber - 1 (MPa)

Physics Models	Mean Error	Variance	Max Error
Model - I	0.026056	0.000951	0.214907
Model - II	0.037778	0.005329	0.241597
Model - III	0.029720	0.004044	0.232425

Table 4.5: Error in rollouts for pressure in chamber - 2 (MPa)

Physics Models	Mean Error	Variance	Max Error
Model - I	0.130483	0.033577	0.964079
Model - II	0.090812	0.023283	0.937466
Model - III	0.100627	0.020564	0.542622

Table 4.6: Error in rollouts for position of joint - 2 (Rad)

Physics Models	Mean Error	Variance	Max Error
Model - I	1.152563	3.574212	11.142500
Model - II	0.932297	2.778308	12.577800
Model - III	0.971925	2.798316	10.802200

Table 4.7: Error in rollouts for velocity of joint - 2 (Rad/s)

Chapter 5

DATA DRIVEN BLACK BOX MODELING

Black box modeling refers to data driven identification of a system's model without explicitly deriving it's physical features through first principles, as done in previous chapters. Neural networks are great function approximaters which can be used to fit and learn a certain distribution of data [14]. Thus, deep learning has been used here to further improve the pneumatic - modeling.

5.1 A Pure Neural Network model

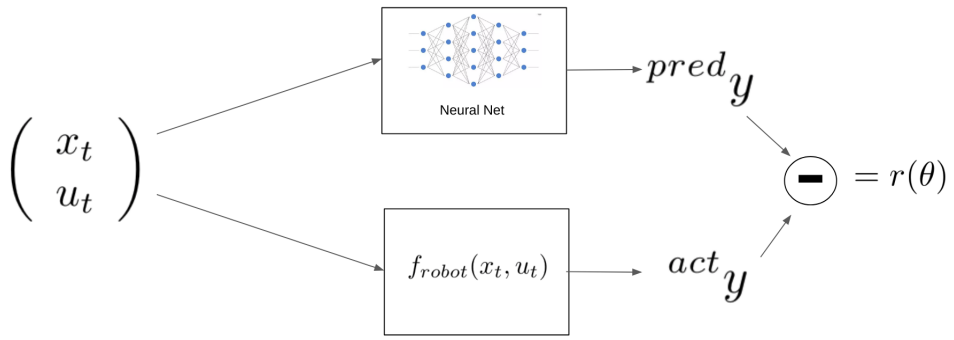

Figure 5.1: Neural Network Model

As shown in figure 5.1, the physics model here has been completely replaced with a neural network, which acts purely as a data driven forward model. The details of neural network [14] are as follows :

1. **Architecture** - It has 9 linear layers interleaved with $relu(x)$ (R) & $tanh(x)$ (T) functions - {(8, 32), R, (32, 64), R, (64, 128), R, (128, 128), R, (128, 128), R, (128, 128), R, (128, 64),

T, (64, 32), T, (32, out)}. Here (a, b) stands for a linear layer shape, where a(b+1) are total number of parameters in that layer.

2. **Hyperparameters** - Batch size used is 32000, optimizer used is ADAM [14], learning rate is 0.005.

3. **Input** - $\{p_1, p_2, q_2, \dot{q_2}\ u_1, u_2, q_1, \dot{q_1}\}$ where p_1, p_2 are pressure in chamber 1 and 2. q_2, $\dot{q_2}$ are position and velocity of joint 2. u_1, u_2 are valve commands for chamber 1 and 2 and q_1, $\dot{q_1}$ are joint position and velocity of joint 1.

4. **Output (targets)** - Derivatives of the state i.e. $\{\dot{p_1}, \dot{p_2}, \ddot{q_2}\}$. Model for predicting $\dot{q_2}$ is identity as it can be obtained directly from input vector.

5. **Dataset** - Training data (total 11 minutes long) consists of 6 minutes of type - I data, 3 different hand recorded minute long trajectories (type - II), 2 minutes of type - III data, as defined in section 4.1.

5.2 Hybrid model

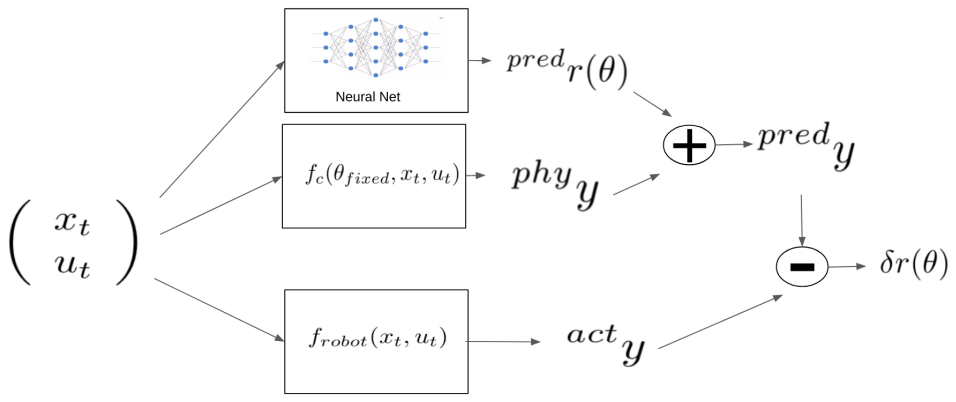

Figure 5.2: Hybrid Model

As shown in figure 5.2, the neural network architecture is same as in section (5.1) but it's used in parallel with previously identified physics model. Model - I has been used here as a candidate for the physics model, as it performed slightly better than other two for this combination.

$$^{act}r_{phy} = ^{act}y - ^{phy}y = f_{robot}(x_t, u_t) - f_c(\theta_{fixed}, x_t, u_t) \tag{5.1}$$

$$\delta r(\theta) = ^{act}r_{phy} - ^{pred}r_{phy}(\theta) = ^{act}y - (^{phy}y + ^{pred}r_{phy}(\theta)) \tag{5.2}$$

Here is $^{pred}r_{phy}$ is output of the network, f_c is the candidate physics model, θ_{fixed} are the parameters identified for it, as discussed in chapter - 3, $\delta r(\theta)$ are the residuals for this regression problem. The goal for this network is to predict the residual error $^{act}r_{phy}$ between actual hardware data and the physics model. Thus, instead of learning the entire dynamics model from the scratch, this network just compensates for the gap between hardware and physics model. Input, dataset and hyperparameters are same as in section (5.1). The only change here is targeted output i.e. $^{act}r_{phy}$.

5.3 Results

Test data (total three Minutes) consists of a minute long type - I & type - III data, randomly sampled with new seed. In addition, it also includes a minute long hand recorded trajectory i.e. type - 2 data. This dataset is same as discussed in section 4.2. It can be clearly seen in tables 5.1 to 5.3 that modeling accuracy has significantly improved with an introduction of a neural network. Moreover for rollouts comparison, tables 5.4 to 5.7 show that hybrid model reduces the variance and maximum absolute test error by a larger factor than a pure neural network. Also in figure 5.6, it can be seen that hybrid network trains considerably faster than a pure neural network.

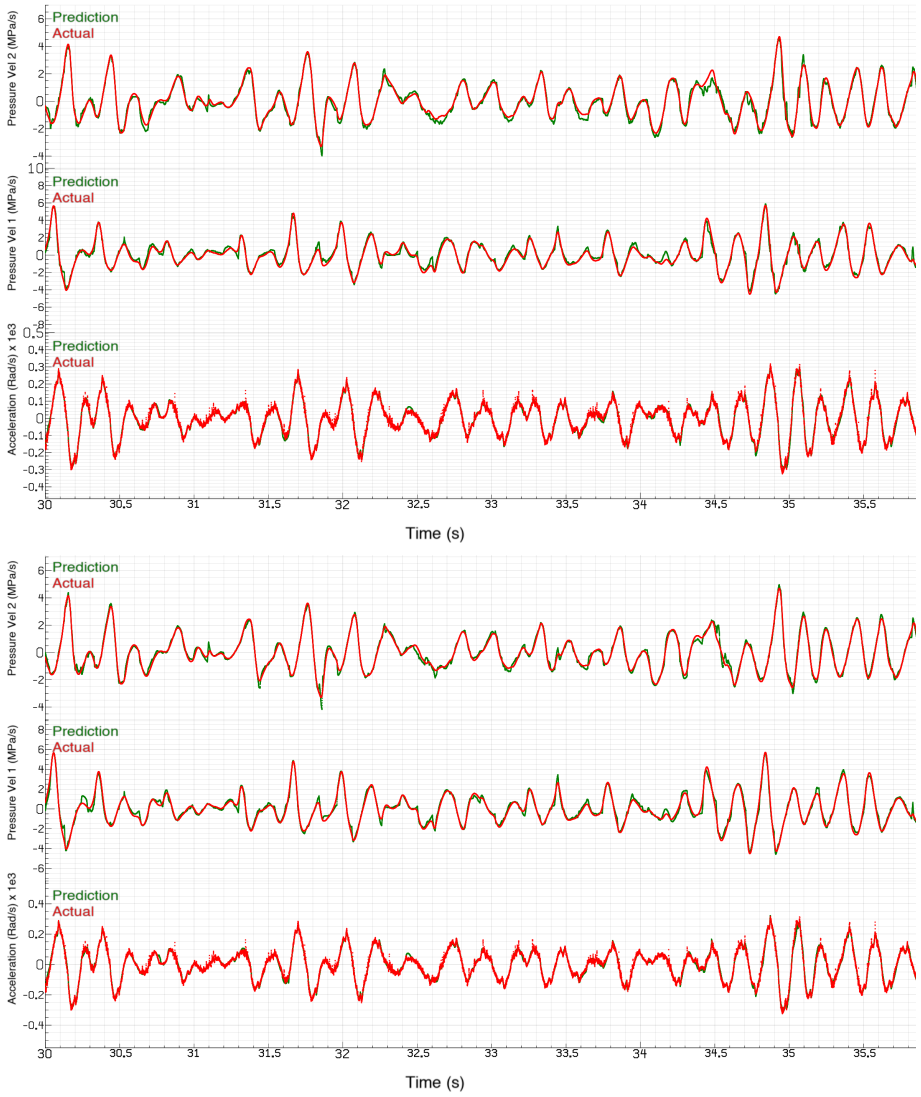

Figure 5.3: Comparison of predicted and actual derivatives for the pure neural network model (top) & hybrid model (bottom)

Figure 5.4: Comparison of predicted and actual pressure roll outs for the pure neural network model (top) & hybrid model (bottom)

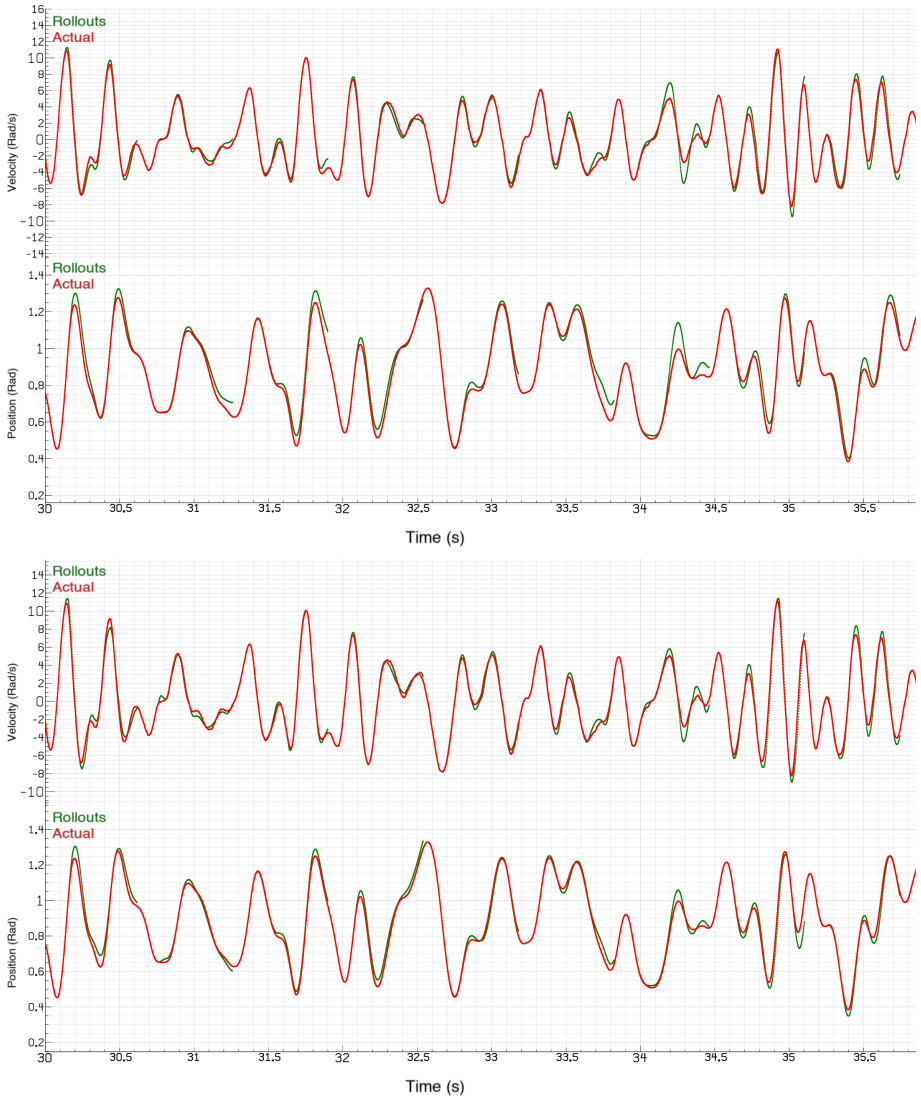

Figure 5.5: Comparison of predicted and actual kinematic roll outs for the pure neural network model (top) & hybrid model (bottom)

Models	Mean Error	Variance	Max Error
Physics Model - I	0.414935	0.158519	2.588980
Pure NN	0.133618	0.023571	2.958950
Hybrid	0.121539	0.017047	2.611395

Table 5.1: Error in predictions for pressure velocity - 1 (MPa/s)

Models	Mean Error	Variance	Max Error
Physics Model - I	0.348157	0.158430	3.831720
Pure NN	0.123270	0.017225	2.684020
Hybrid	0.109308	0.013348	2.347060

Table 5.2: Error in predictions for pressure velocity - 2 (MPa/s)

Models	Mean Error	Variance	Max Error
Physics Model - I	28.793105	789.228292	816.597000
Pure NN	8.156197	111.254949	804.070000
Hybrid	8.259506	107.548734	801.815000

Table 5.3: Error in predictions for acceleration (Rad/s^2)

Models	Mean Error	Variance	Max Error
Physics Model - I	0.027331	0.001149	0.306210
Pure NN	0.007659	0.000121	0.117803
Hybrid	0.006399	0.000077	0.047614

Table 5.4: Error in rollouts for pressure in chamber - 1 (MPa)

Models	Mean Error	Variance	Max Error
Physics Model - I	0.026056	0.000951	0.214907
Pure NN	0.008152	0.000173	0.166643
Hybrid	0.007136	0.000088	0.048070

Table 5.5: Error in rollouts for pressure in chamber - 2 (MPa)

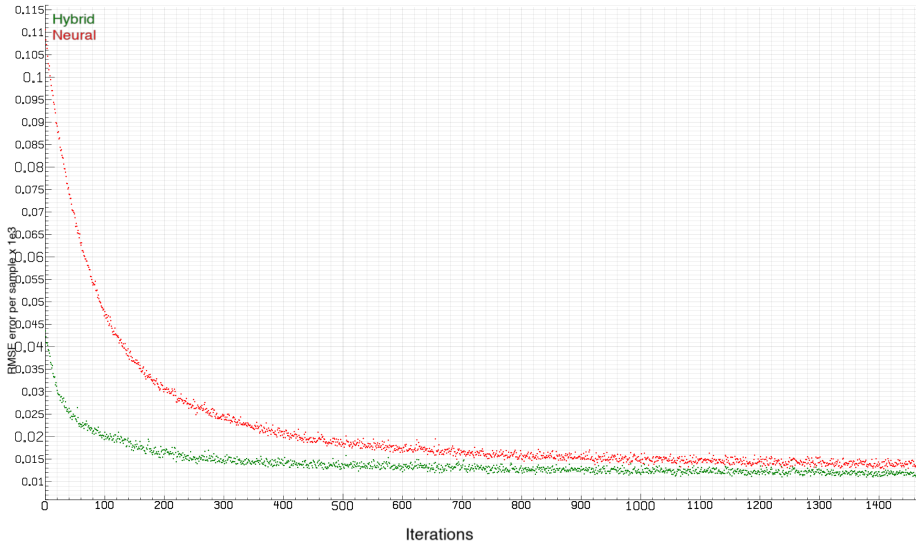

Figure 5.6: Comparison of training time for the pure neural network model (top) & hybrid model (bottom)

Models	Mean Error	Variance	Max Error
Physics Model - I	0.130483	0.033577	0.964079
Pure NN	0.034595	0.002719	0.5981576
Hybrid	0.026008	0.002011	0.227806

Table 5.6: Error in rollouts for position of joint - 2 (Rad)

Models	Mean Error	Variance	Max Error
Physics Model - I	1.152563	3.574212	11.142500
Pure NN	0.353191	0.501714	10.465200
Hybrid	0.282996	0.245991	6.440240

Table 5.7: Error in rollouts for velocity of joint - 2 (Rad/s)

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, multiple physics models for a pneumatically actuated robot were identified using first principles and some classical regression techniques. Despite giving good predictions, these models are limited as in certain segments of test data they can not capture the features that accurately, as shown in figure (6.1) & (6.2). In the motivation to push towards black box data driven modeling, two different deep learning based approaches were tried, as presented in Chapter 5. As shown in table 6.1, on a test dataset a pure neural network representing complete forward dynamics model reduces the L_1 norm error by approx. 70 % whereas a hybrid model reduces the same by 75 %.

Physics Model - I	Pure NN	Hybrid
1.817	0.546	0.444

Table 6.1: L_1 norm of error in rollouts for joint - position (Rad), velocity (Rad/s), pressure in chamber - 1 & 2 (bar)

It can be seen in figure (6.1) & (6.2) that hybrid model is able to learn certain dynamic features much better than a pure neural network. As mentioned before, the test data for computing all of these results consists of both dynamic, high speed, fast varying frequency signals (type - I & III) and static, slow hand recorded trajectories (type - I). The former ensures that all the dynamic and inertial effects are activated, so the network can model those. Whereas relatively slower hand recorded data, ensures that model should generalize well enough for real world expectations and objectives. The ability of the proposed models in this work to predict well in these highly versatile

test scenarios, validates their strength in representing the true robot dynamics. Hope is that our proposed model can be used to solve variety of different control problems like force - control, position control, etc using model predictive control.

6.2 Generalization

Generalization is a crucial concept in data driven modeling. It refers to the ability of a model to be able predict and perform well in sufficiently different kinds of scenarios. For physics-based models as discussed in Chapter - 3, generalization holds quite well as physics would always be consistent as long as hardware isn't changed. When it comes to a black box model, this issue becomes much more important as it can get biased towards one kind of data distribution and perform poorly in other scenarios.

In section 4.1, three different kinds of data regimes were discussed, each of which represented different control scenarios. Type - I data reflects the state and control signals that can be observed in any kind of dynamic control, whereas Type - II and III represent slow and fast position tracking data. One might argue that training and test distributions are not that different to reflect generalization, as Type - I and Type - III data are quite homogeneous. This means even if the test dataset is sampled through a different random seed, it might not be that different from training set. This raises a concern that a black box model might over-fit on them.

However, all three different data regimes kind of capture useful range for most of the control tasks. They have both dynamic and static data, spanning pressure and position ranges, thus providing good information from both force and position control perspective. Type - II test data which consists of hand recorded trajectories are somewhat non-homogeneous as anytime a human is asked to play with the robot and record, he may do totally something else. So, as these model are able to predict well for all these versatile datasets, it has some generalization property. Also the problem for biasing is scaled down while adopting the hybrid model approach instead of a pure neural network, owing to the physics model in parallel, which has stronger generalization property.

6.3 Future Work

Future work can involve :

1. **Stronger evaluation for generalization** : Generalization might be guaranteed better if the black box models were trained on any of the two datasets and then exclusively tested on third.

2. **Model-based control** : One of the important steps discussed in section 2.3 for validation of all these models were not covered in this work. Perhaps, a stronger test of predictability would be to use model predictive control [4] to solve multiple control tasks without modifying the model much.

3. **New ways for black box modeling** : Hybrid model's better predictability over a pure neural network suggests that, using a first principle physics model as a starting point for data driven system identification can provide much better results. One of the interesting directions to explore here would be to identify the unknown blocks in the physics models and use function approximators exclusively for those parts. For an instance, the area model function could have been replaced with a small network, thereby making the physics modeling more compatible with data driven approaches.

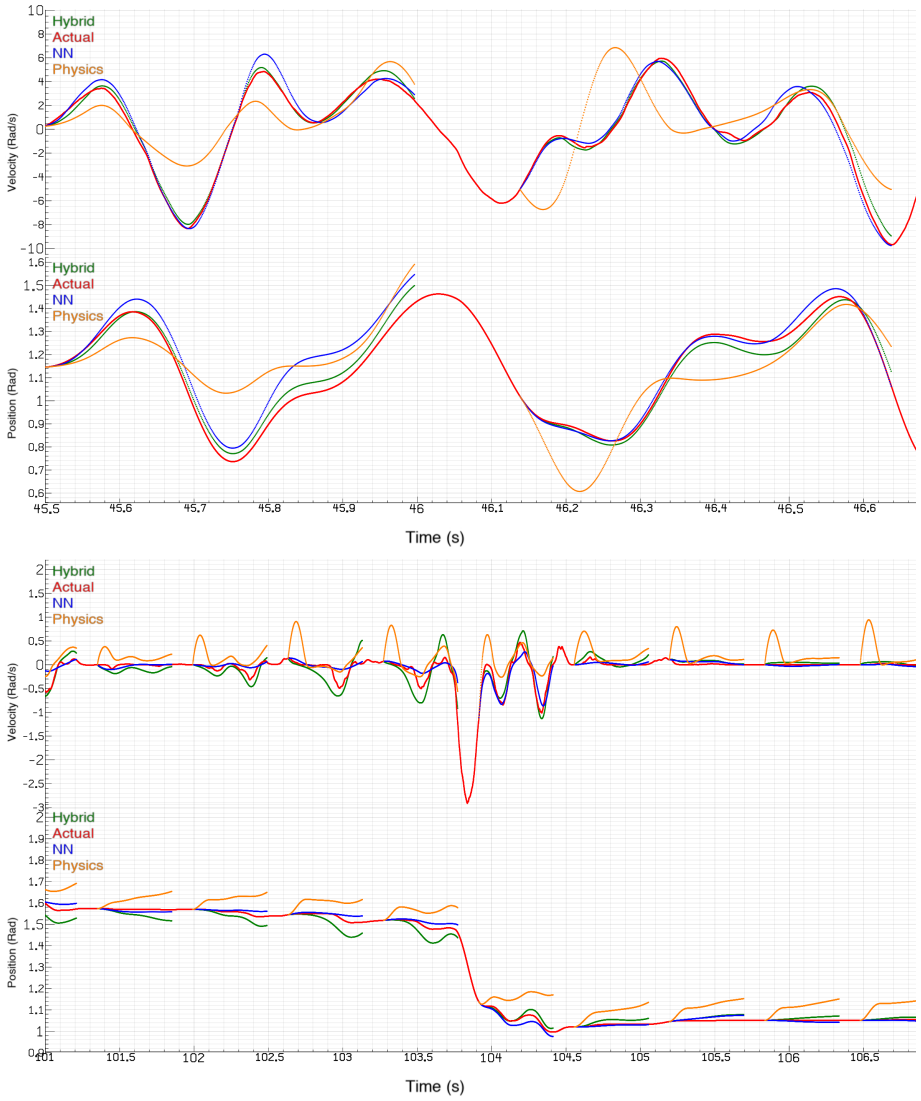

Figure 6.1: Comparison of predicted and actual kinematic roll outs for the physics model, pure neural network model & hybrid model

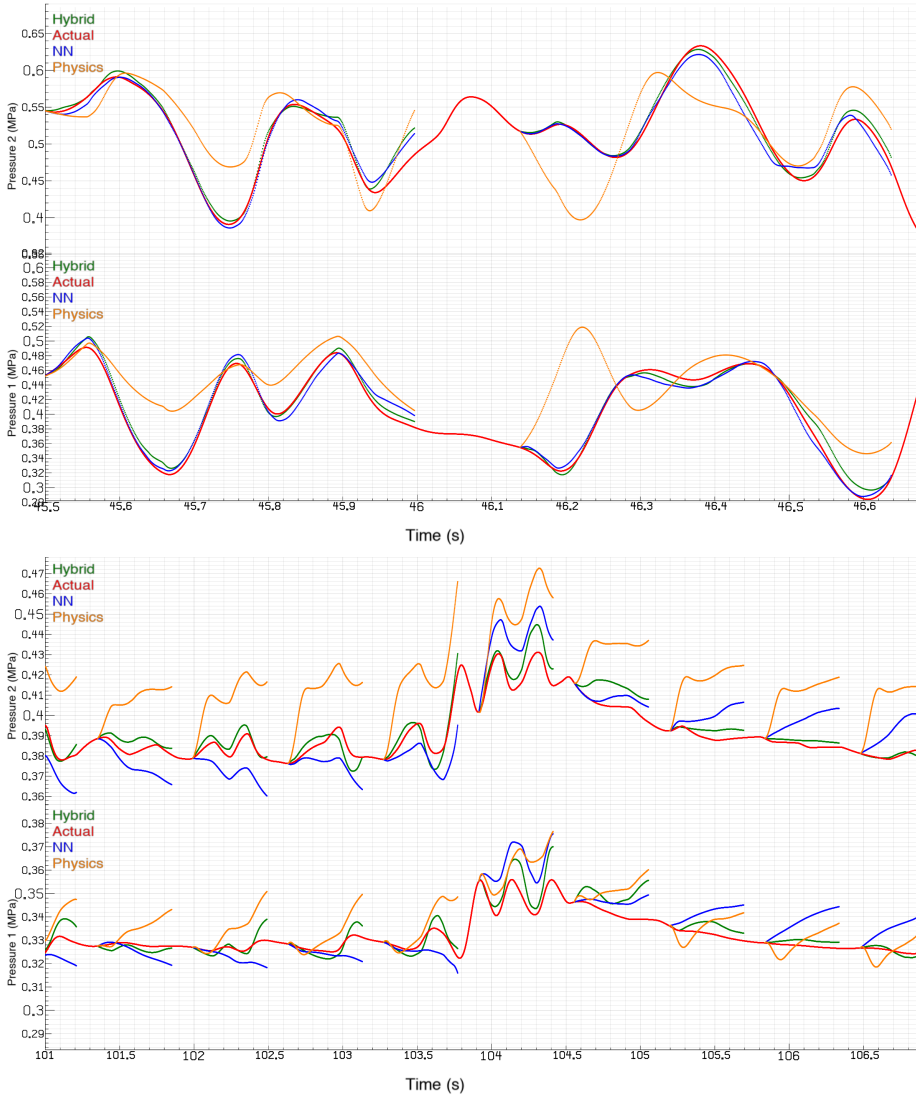

Figure 6.2: Comparison of predicted and actual pressure roll outs for the physics model, pure neural network model & hybrid model

BIBLIOGRAPHY

[1] E. Todorov, C. Hu, A. Simpkins, and J. Movellan, "Identification and control of a pneumatic robot," in *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*. IEEE, 2010, pp. 373–380.

[2] T. Yuval, T. Wu, J. Movellan, and E. Todorov, "Modeling and identification of pneumatic actuators," in *International Conference on Mechatronics anf Automation (ICMA)*, 2013.

[3] M. W. Spong, S. Hutchinson, M. Vidyasagar *et al.*, *Robot modeling and control*. wiley New York, 2006, vol. 3.

[4] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4906–4913.

[5] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1433–1440.

[6] E. Richer and Y. Hurmuzlu, "A high performance pneumatic force actuator system: part ii—nonlinear controller design," *J. Dyn. Sys., Meas., Control*, vol. 122, no. 3, pp. 426–434, 1999.

[7] D. Ben-Dov and S. E. Salcudean, "A force-controlled pneumatic actuator," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 906–911, 1995.

[8] Z. Rao and G. M. Bone, "Nonlinear modeling and control of servo pneumatic actuators," *IEEE transactions on control systems technology*, vol. 16, no. 3, pp. 562–569, 2008.

[9] V. Kumar, C. Visak, and E. Todorov, "High performance pneumatics using model predictive control."

[10] D. Büchler, R. Calandra, B. Schölkopf, and J. Peters, "Control of musculoskeletal systems using learned dynamics models," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3161–3168, Oct 2018.

[11] J. Hespanha, *Linear Systems Theory*. Princeton University Press, 2009. [Online]. Available: https://books.google.com/books?id=uiRymAEACAAJ

[12] W. Khalil and E. Dombre, *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[13] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[15] N. Hansen, "The CMA evolution strategy: A tutorial," *CoRR*, vol. abs/1604.00772, 2016. [Online]. Available: http://arxiv.org/abs/1604.00772

[16] J. M. McCarthy and G. S. Soh, *Geometric design of linkages*. Springer Science & Business Media, 2010, vol. 11.