Università degli Studi di Verona

RPC Report

Computer Engineering For Robotics and Smart Industry

Nicola Marchiotto - VR462317

2022

Contents

1 Introduction

The document reports the techniques and results of the estimation of the mechanical part and controllers of the various architecture adopted in the 'A Review of Myoelectric Control Systems for Voluntary Control of Exoskeletons' paper.

List of control architecture and adapters used to collect the data for the estimation

Control Architecture	Adapter
COMP	NONE
FORCE	PLAIN P, MULTICH8
POS V	PLAIN P, MULTICH8
FIX IMP	PLAIN P, MULTICH8
ADM	PLAIN P, MULTICH8
FORCE INT	PLAIN P, MULTICH8

The data used in the scripts are defined inside the file
myo_tools_testing\testing_version1\settings\logs_eval_gains_2019_10_09_10_18_2020_06_26_30

1.1 Data Cleaning

Data cleaning operations were performed before the actual estimation

An experiment was considered an outlier if it was characterized by one of the following aspects

- The last value of the torque signal to be outside a small range.
- Considered an outlier by the *rmoutliers* matlab function, so for each temporal instant, if the mean of all the torque value of that instant, is more than 3 standard deviations computed for the instant

After the cleaning operations, the signals were trimmed at start to remove the non relevant portion of the signal

1.2 Estimation pipeline

To estimate the models, the *tfest* function was exploited. From each experiment contained in the cleaned set, pair of measured input and output, a model was estimated. Then every experiment of the cleaned set was used to validate the computed models.

A score was associated to each tested model. This score considers, for all the experiments, how well the model can produce the experiment output given the input.

The model with the best average score was chosen as best model for the given couple architecture - adapter.

1.3 Other considerations

In the estimation, data with different β adapter parameter were used since no feasible method for filtering was found.

2 Motor model estimation

We wanted to validate the assumption where the mechanical model G has the following form

$$\frac{1}{Js^2 + ds}$$

To estimate G, three approaches were used, the results of these approaches were saved in the *resultStructures* folder

- **Estimation from torque and position:**
  First estimation approach tested, results saved in the *G_resultTable* variable
- **Estimation from torque and position, setting the end tail of the torque signal to zero**
  From the first approach we noticed that sometimes the position output from the estimated mechanical model diverged even if the torque was set constant at the end of the signal for a consistent amount of time. The reason for this behaviour is that the mechanical model contains an integrator, so even if the torque signal is set to constant, the error accumulates making the position diverge, results saved in the *G_resultTable_zero* variable
- **Estimation from torque to velocity**
  Given the considerations discussed in the previous point, we tried to estimate the mechanical model from torque and velocity, so excluding the estimation of integrator in the plant, results saved in the *G_resultTable_vel* variable

The best results were obtained using the second method, the better models for the various architectures usually showed the pole locate in the range $[-1.3, -1.2]$.

For the control estimation, G was assumed as

$$G = \frac{14.706}{s(s+1.25)} = \frac{1}{0.068s^2 + 0.085s}$$

3 Controllers estimation

For each pair of architecture-decoder, controllers were estimated and then tested on every architecture.

Three approaches were used:

- **Estimation from position error and command torque:**
  First technique tried, this method gave the worst results producing always unstable controllers. Plot of these controllers are omitted
- **Whole architecture estimation, then knowing the mechanical plant and the structure of the architecture, the controller was retrieved via algebraic operations:**
  This technique gave relevant results. The whole architecture estimated models W were always stable with an accuracy always greater than 75%.
  The controller from the estimated W, was extracted applying zero-pole cancellations with a tolerance of 0.001
  To be noticed that the extracted controller is not always a PD controller, even if it was the form specified in the W estimation
- **Frequency based approach:**
  Considering the controller obtained using the second technique but without applying any zero-pole cancellations, a sweep reference input with decreasing amplitude was feed to this controller. The sweep frequency goes from 1 to 10 Hz in 10 seconds. The output was recorded and a PD controller was fitted from the frequency response of the pair of input-output. The reason of this technique is to avoid algebraic operations used in the second and so always get a PD controller

Step responses and controllers are now reported for the pairs architecture-adapter. Step amplitude was set to 1, saturators with range $[-3, 3]$ were used to limit unstable models.

In the figures are reported the following signals

- The blue one is the reference, the amplitude is the one of the recorded experiments with the given architecture
- The orange is the architecture response
- The green are the recorded experiment's position output used to estimate the controller.

3.1 COMP - NONE

$$C = \frac{-0.053653(s-68.83)(s+1.25)}{(s^2+7.145s+34.7)}$$

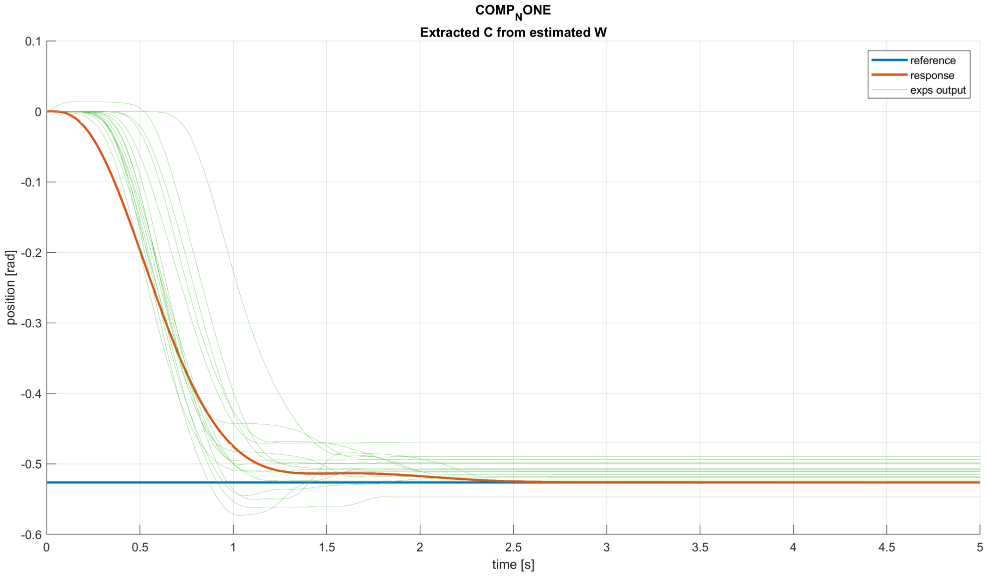

Figure 1: C extracted from estimated W

$$C = \frac{0.51855(s+1.253)}{(s+2.734)}$$

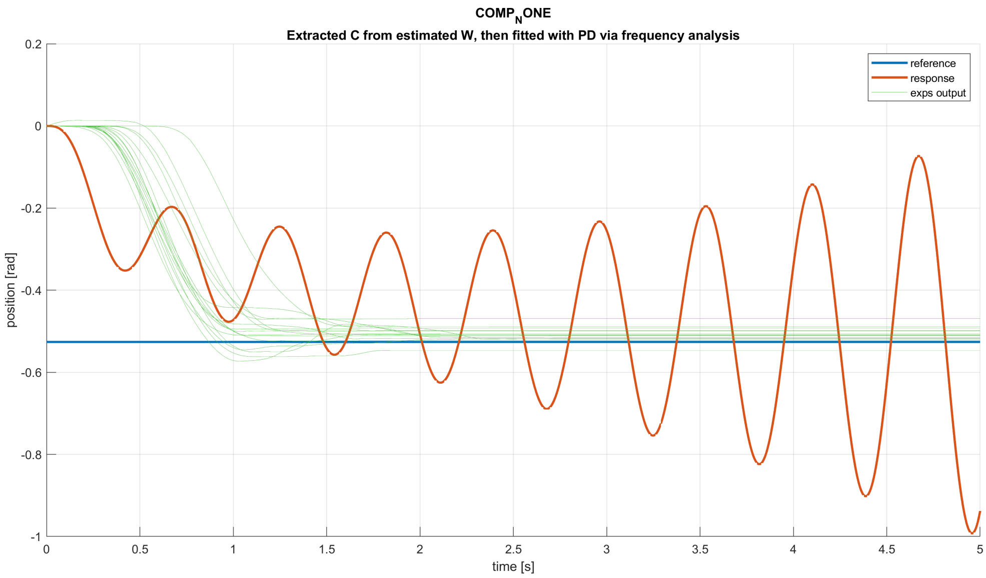

Figure 2: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.2 POS V - PLAIN P

$$C = \frac{3.9537s(s^2 + 4.867s + 28.08)(s^2 + 30.66s + 2059)}{(s + 70)(s - 0.006544)(s^2 + 5.141s + 34.02)(s^2 + 7.188s + 160)}$$

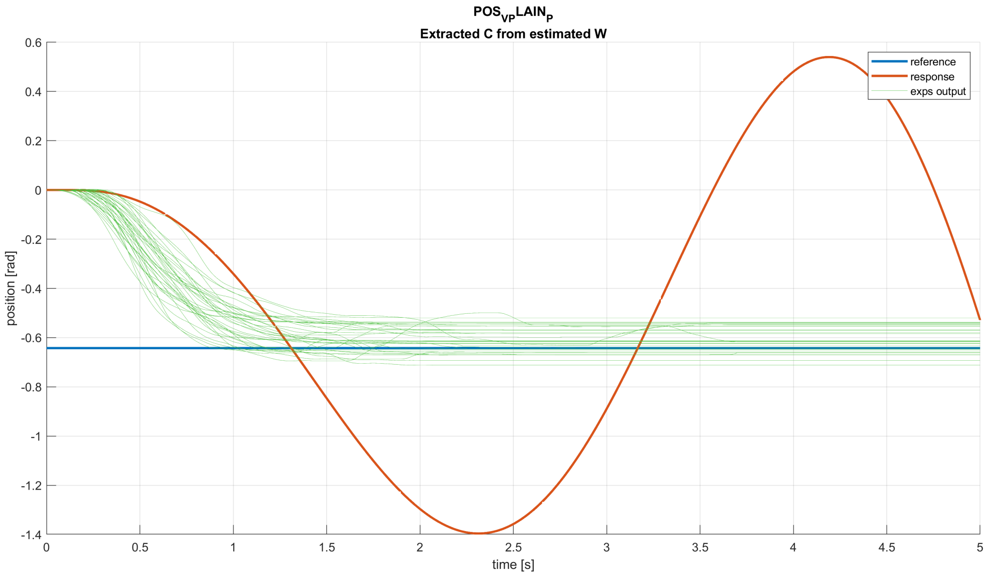

Figure 3: C extracted from estimated W

$$C = \frac{0.68067(s + 4.454)}{(s + 4.476)}$$

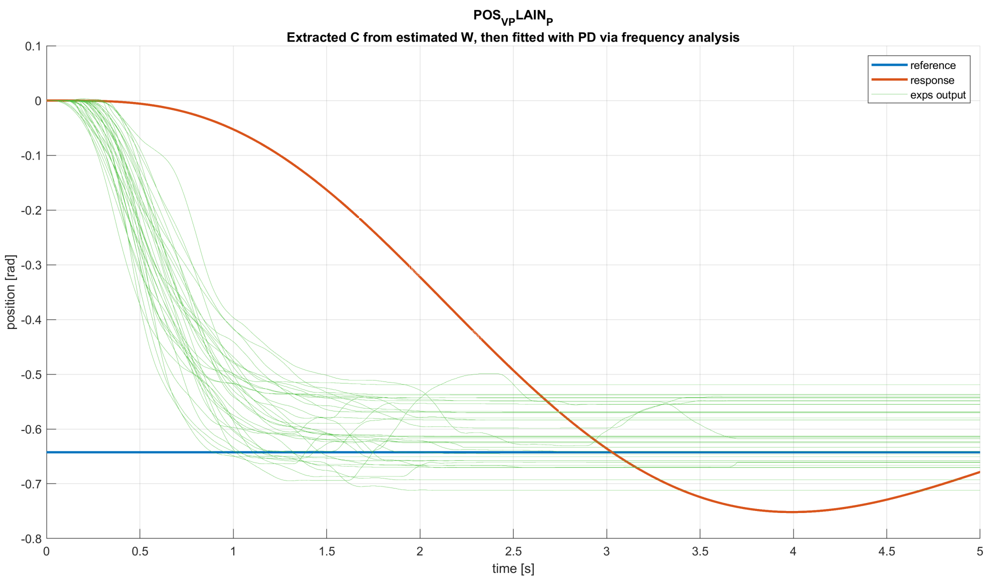

Figure 4: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.3 FORCE - PLAIN P

$$C = \frac{0.040111(s + 18.86)(s + 1.25)}{(s^2 + 5.485s + 19.35)}$$

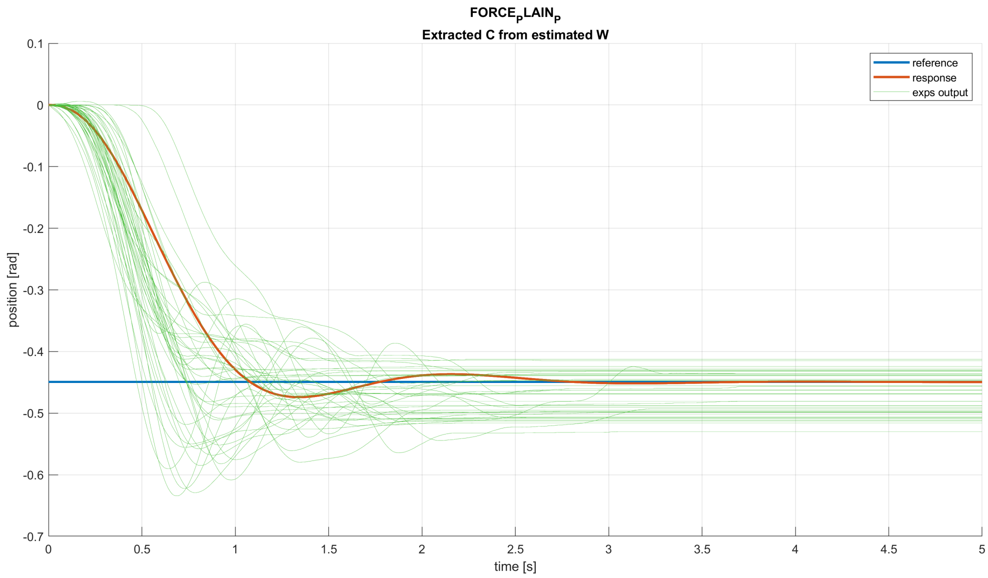

Figure 5: C extracted from estimated W

$$C = \frac{0.15443(s + 1.102)}{(s + 1.976)}$$

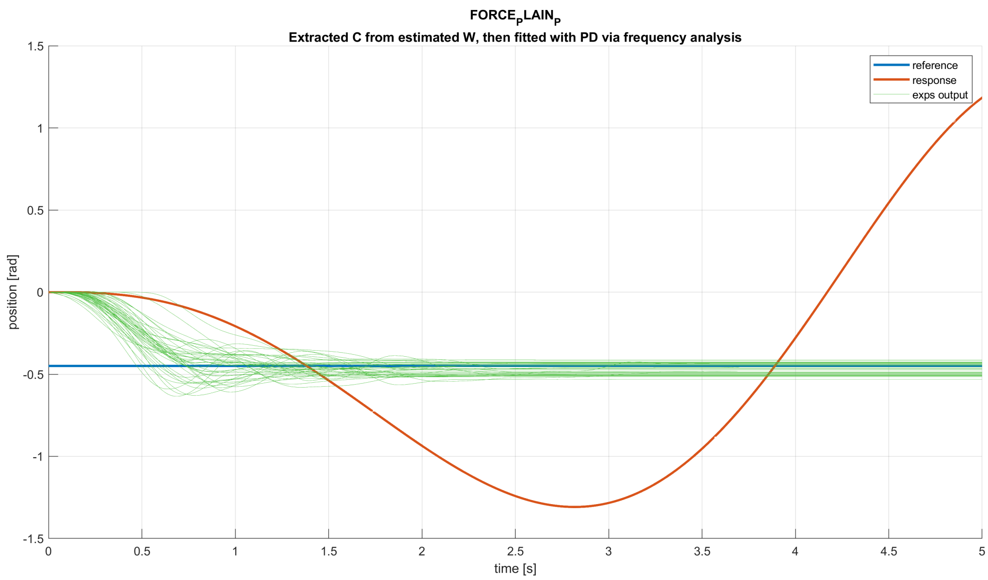

Figure 6: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.4 FORCE INT - PLAIN P

$$C = \frac{-0.12157s(s+1.25)(s-0.0903)}{(s+2.819)(s^2+1.403s+2.697)}$$

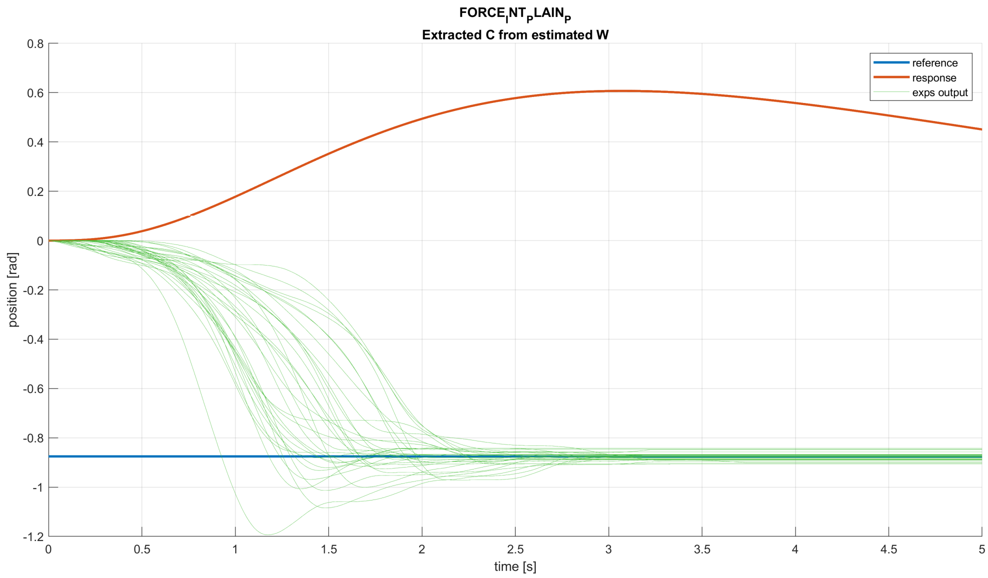

Figure 7: C extracted from estimated W

$$C = \frac{-0.15184(s-0.58)}{(s+3.215)}$$

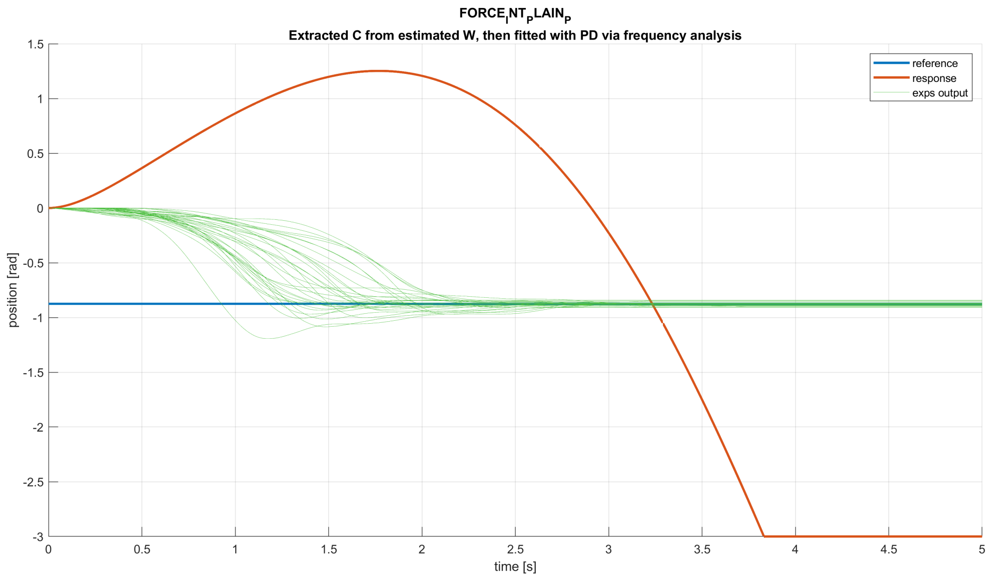

Figure 8: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.5 FIX IMP - PLAIN P

$$C = \frac{0.025939s(s + 0.2187)(s^2 + 1.25s + 117.6)}{(s + 0.1826)(s + 0.01556)(s^2 + 3.888s + 13.34)}$$

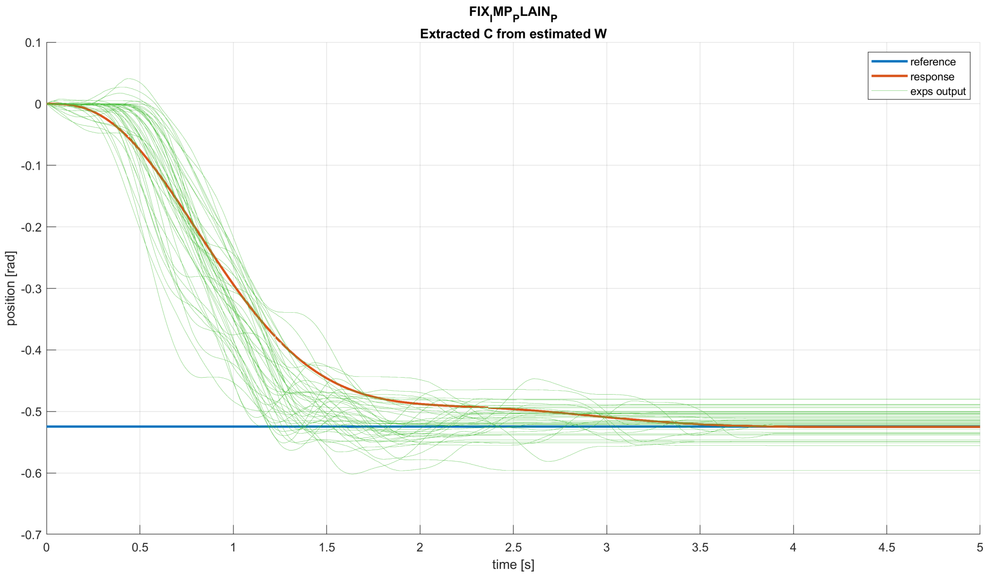

Figure 9: C extracted from estimated W

$$C = \frac{-0.14405(s - 5.278)}{(s + 3.346)}$$

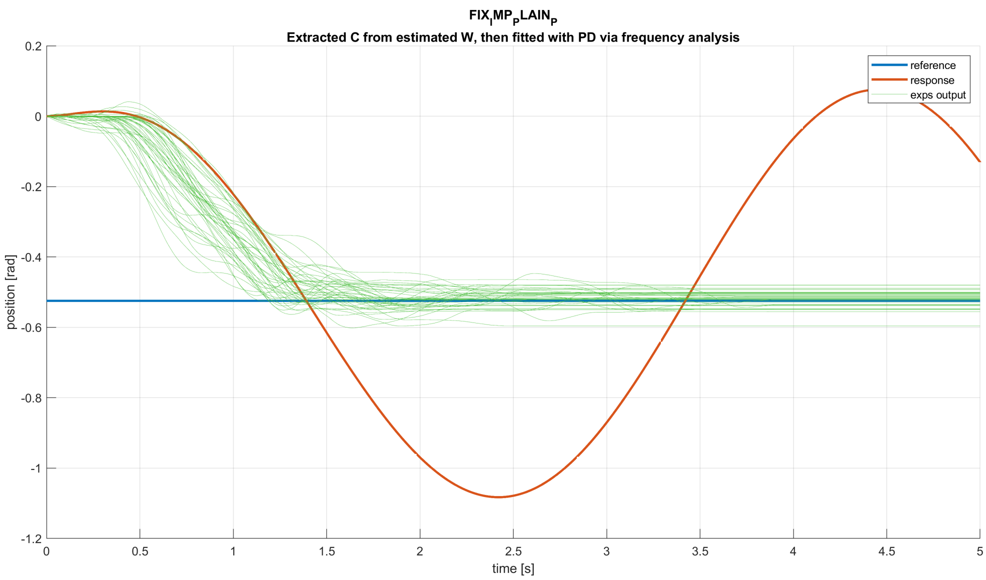

Figure 10: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.6 ADM - PLAIN P

$$C = \frac{-159.03(s+5)(s+2.113)(s-0.9443)(s^2+30.66s+2059)}{(s+70)(s+1.964)(s^2+8.136s+43.75)(s^2+5.317s+957.3)}$$

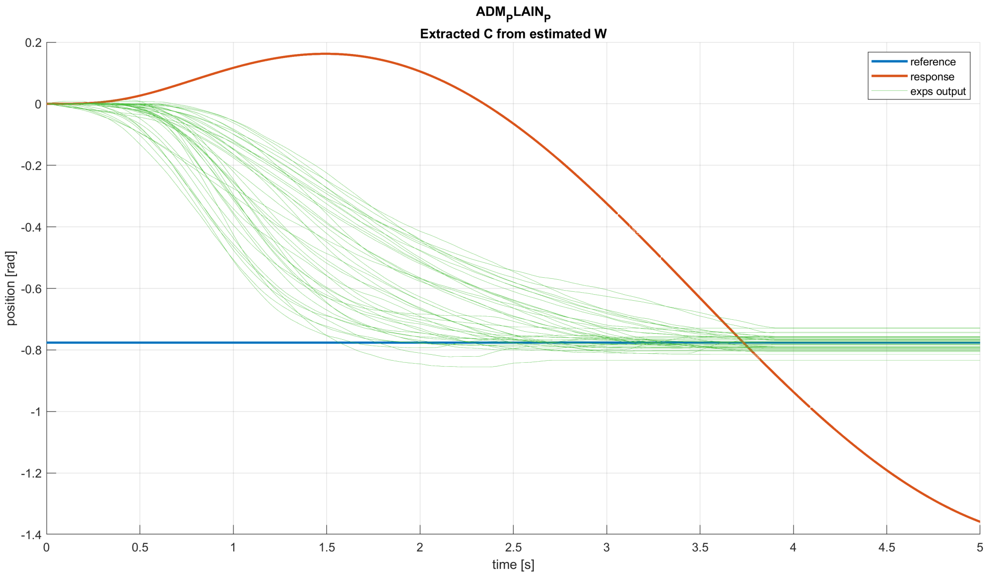

Figure 11: C extracted from estimated W

$$C = \frac{-8.8876(s-1.046)}{(s+12.3)}$$

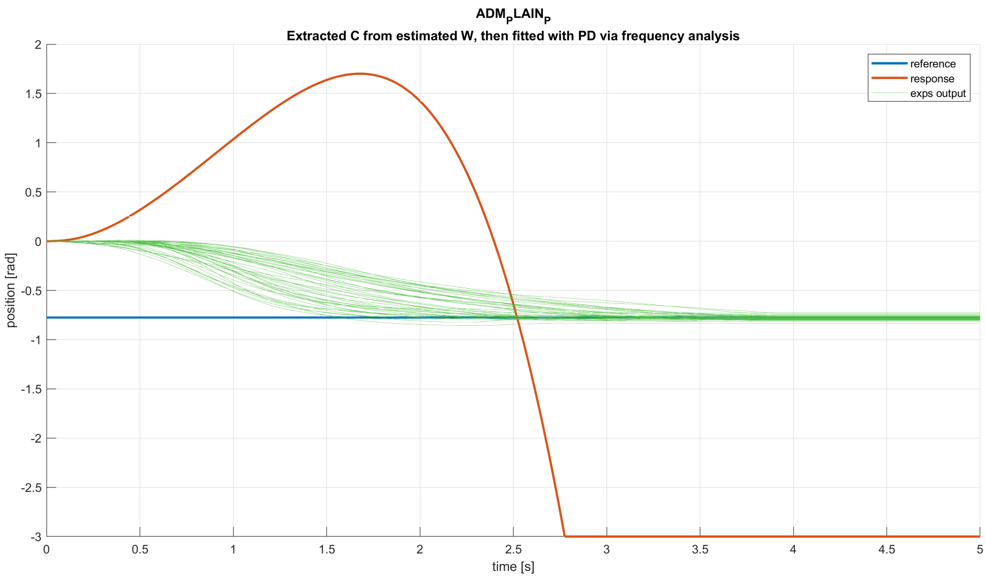

Figure 12: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.7 POS V - MULTICH8

$$C = \frac{15.935(s^2 + 2.677s + 4.882)(s^2 + 30.66s + 2059)}{(s + 70)(s^2 - 0.4969s + 5.428)(s^2 + 25.71s + 312.8)}$$

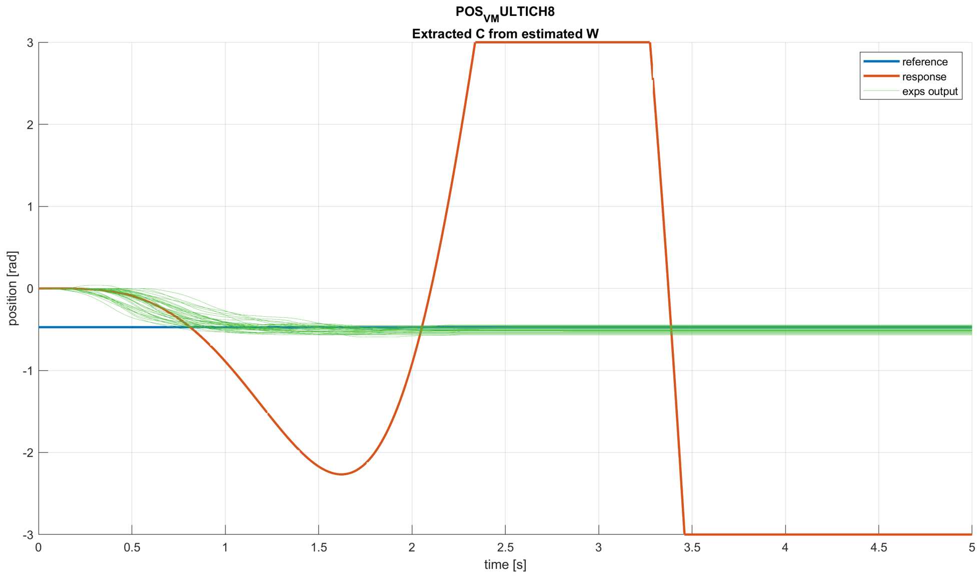

Figure 13: C extracted from estimated W

$$C = \frac{4.8428(s + 0.5479)}{(s - 1.839)}$$

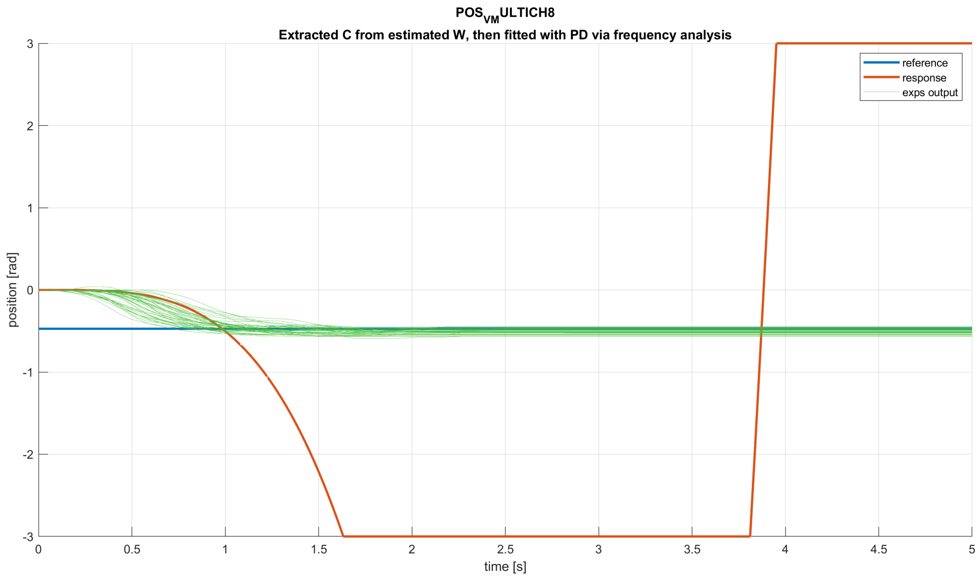

Figure 14: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.8 FORCE - MULTICH8

$$C = \frac{0.16456(s + 12.49)(s + 1.25)}{(s^2 + 9.04s + 39.36)}$$

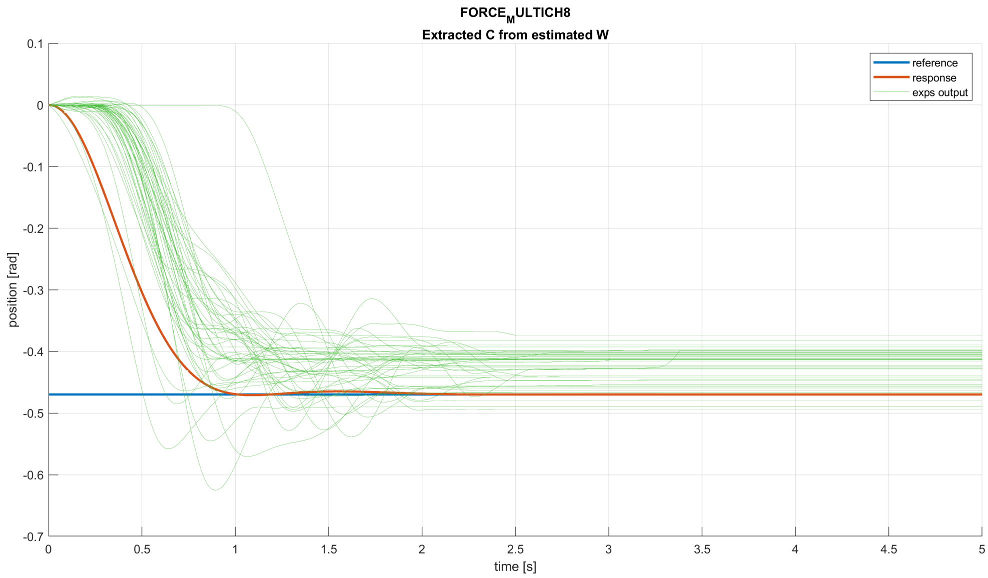

Figure 15: C extracted from estimated W

$$C = \frac{0.30193(s + 1.077)}{(s + 4.554)}$$

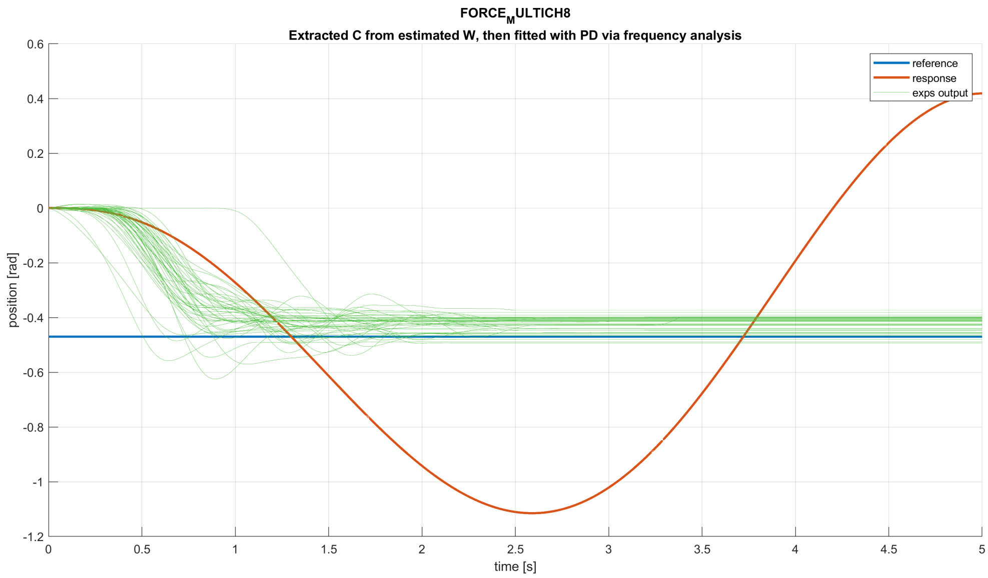

Figure 16: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.9 FORCE INT - MULTICH8

$$C = \frac{0.058527s(s+4.661)(s+1.25)}{(s+2.071)(s^2+3.319s+7.328)}$$

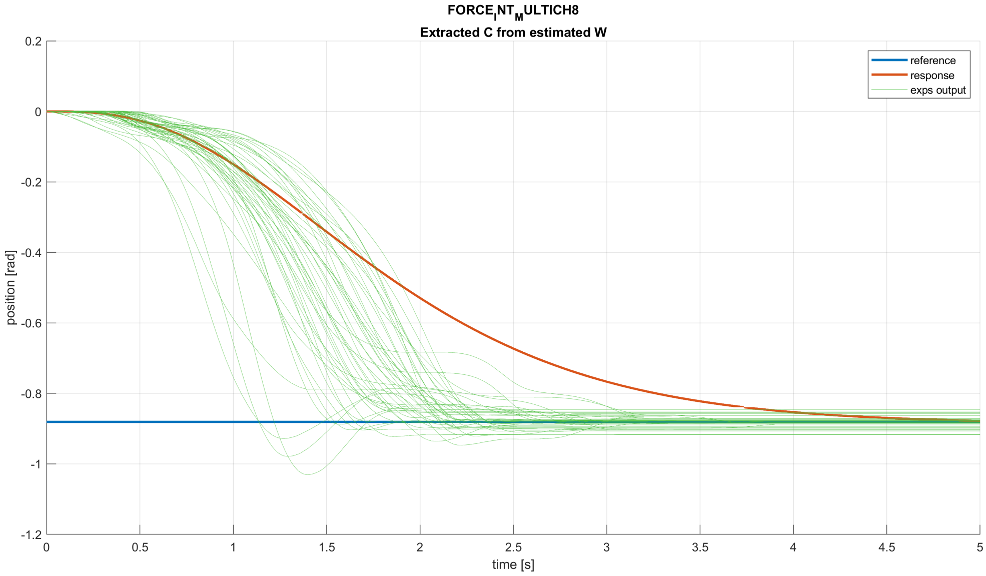

Figure 17: C extracted from estimated W

$$C = \frac{0.099605(s+0.1264)}{(s+2.13)}$$

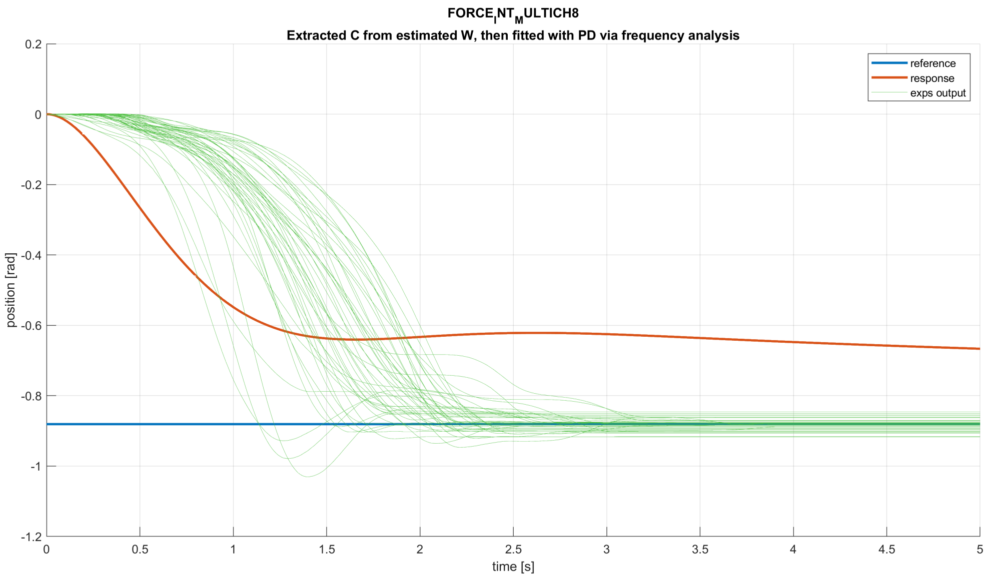

Figure 18: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.10 FIX IMP - MULTICH8

$$C = \frac{0.078224(s+2.23)(s^2+1.25s+117.6)}{(s+2.014)(s^2+5.213s+25.62)}$$

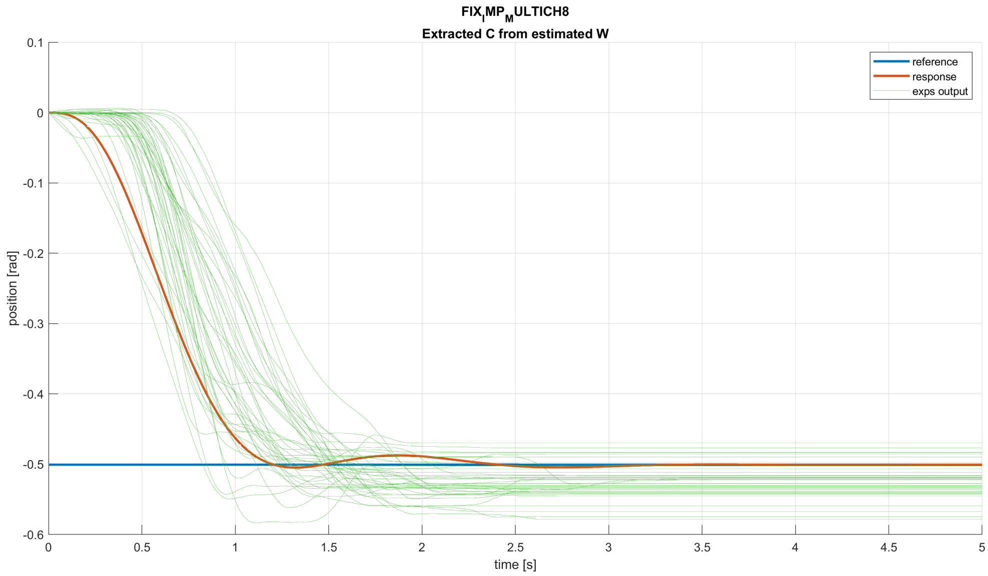

Figure 19: C extracted from estimated W

$$C = \frac{-0.22579(s-9.109)}{(s+5.231)}$$

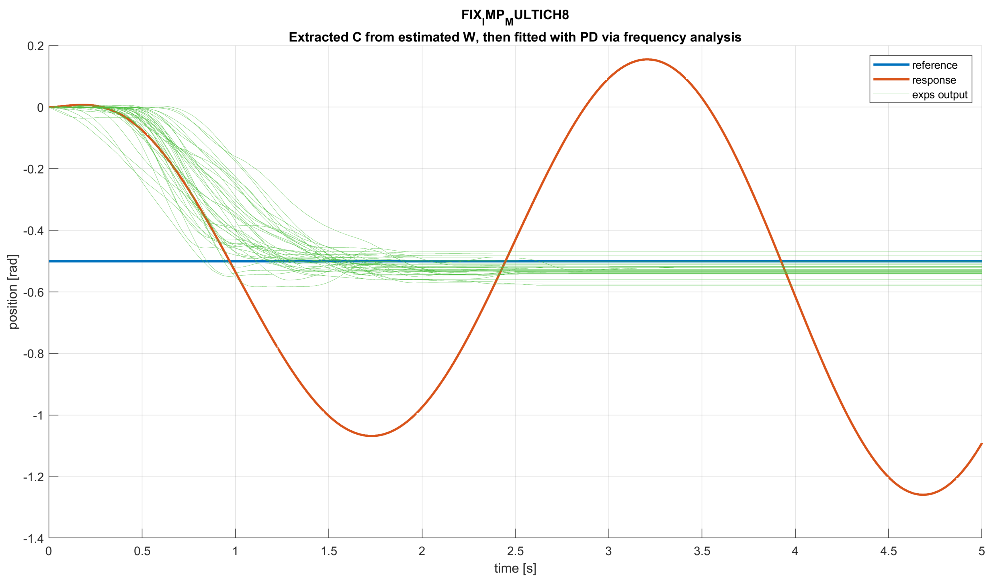

Figure 20: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.11 ADM - MULTICH8

$$C = \frac{2.8557s(s+5)(s^2+1.594s+2.092)(s^2+30.66s+2059)}{(s+70)(s+1.457)(s+0.01544)(s^2+2.157s+8.064)(s^2+2.817s+67.91)}$$

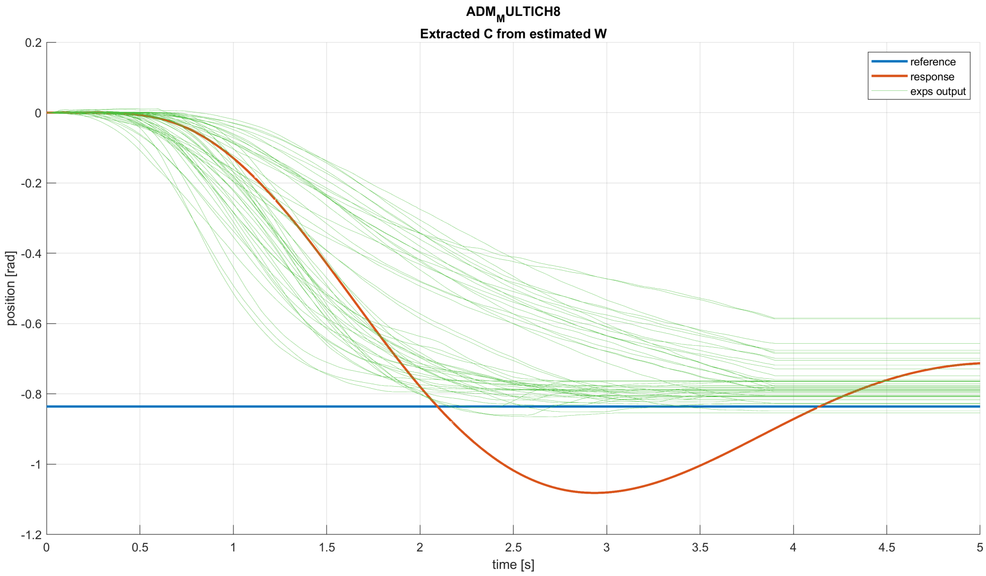

Figure 21: C extracted from estimated W

$$C = \frac{3.1121(s+1.463)}{3.1121(s+1.463)}$$

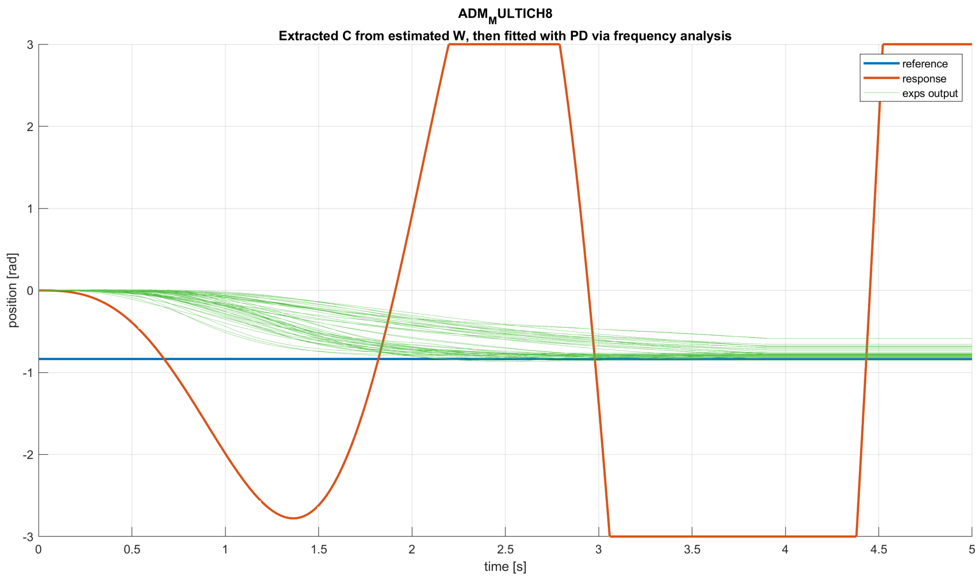

Figure 22: C extracted from estimated W, the fitted the controller with a PD via frequency analysis

3.12 Results

Results are reported in tabular way. Performance of stable behaviours are categorized in *bad, decent, good, perfect*

Technique	Control Architecture	Adapter	#Zeros	#Poles	Behaviour	Performance
C from W	COMP	NONE	2	2	Stable	Perfect
	POS V	PLAIN P	5	6	Unstable	
	FORCE	PLAIN P	2	2	Stable	Perfect
	FORCE INT	PLAIN P	3	3	Stable	Bad
	FIX IMP	PLAIN P	4	4	Stable	Perfect
	ADM	PLAIN P	5	7	Unstable	
	POS V	MULTICH8	4	5	Unstable	
	FORCE	MULTICH8	2	3	Stable	Perfect
	FORCE INT	MULTICH8	3	3	Stable	Perfect
	FIX IMP	MULTICH8	3	3	Stable	Perfect
	ADM	MULTICH8	6	7	Stable	Good
PD frequency fit	COMP	NONE	1	1	Unstable	
	POS V	PLAIN P	1	1	Stable	Good
	FORCE	PLAIN P	1	1	Unstable	
	FORCE INT	PLAIN P	1	1	Unstable	
	FIX IMP	PLAIN P	1	1	Unstable	
	ADM	PLAIN P	1	1	Unstable	
	POS V	MULTICH8	1	1	Unstable	
	FORCE	MULTICH8	1	1	Unstable	
	FORCE INT	MULTICH8	1	1	Stable	Good
	FIX IMP	MULTICH8	1	1	Unstable	
	ADM	MULTICH8	1	1	Unstable	