



Robotic Computer–Aided Tuning of Multi-cavity RF Filters

A thesis submitted in fulfilment of the requirements for the degree of
Master of Engineering

NIMA GOLFOROUSHAN

Bachelor of Engineering – Electronics (Hons.)

**School of Electrical and Computer Engineering
College of Science Engineering and Health
RMIT University**

August, 2015

DECLARATION

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Nima Golfroushan

26/08/2015

ACKNOWLEDGEMENT

I would like to thank the following individuals and organizations for their assistance in my research and in writing this thesis:

- My senior supervisor Prof. Kamran Ghorbani for his continual support, advice and constructive feedback throughout my time at RMIT University.
- Dr. PJ Radcliffe, my second supervisor for his interest in my project, ideas and useful discussions.
- Ernest Fardin, Principal RF Engineer at RFS Pty Ltd. for his technical assistance in calibration and tuning.
- Nail Gibson, quality manager at Radio Frequency Systems (RFS) for his knowledge and expertise in mechanical design, and the constructive discussions that took place.
- Radio Frequency Systems (RFS) for providing financial support and required equipment to finalize this project.

ABSTRACT

Due to the demand for precision RF filter solutions, fully automated cavity filter manufacturing systems are a topic of interest for researchers. Currently, tuning stages for filter production lines are implemented by hand. This stringent process is both expensive and time consuming. Depending on the complexity of the cavity filter, this process may take up to several hours. Therefore, it is not suitable for higher volume production. To overcome this problem, Radio Frequency Systems (RFS Pty Ltd.) company is trying to develop a number of automated filter manufacturing systems that make the leap from conventional ‘trial and error’ manual filter tuning to automatic robotic tuning and set a new standard for filter production. The aim of this project, supported by RFS, is to design and manufacture a Robotic Computer-Aided Tuning (RoboCAT) system.

The first section of this thesis deals with the design and fabrication of an automated robot arm to interface with tuning elements. For this purpose, a customised coaxial screw/nut driver is created to tune and lock the tuning elements simultaneously. Aspects of the automated tuner include: (1) Backlash compensation to increase the tuning resolution and reduce the tuning time. (2) Absolute positioning in order to have a second feedback source for robot codes along with obtained data from Computer-Aided Tuning (CAT) software. (3) Fault recognition ability to detect any potential error in CAT codes in early stage.

The second stage of this project deals with finding a proper tuning instruction. By having a complete tuning instruction, it is feasible to write a standalone tuning code for an automated tuning system. It can be concluded from literature that circuit

model parameter extraction is the only ideal tuning technique to be implemented by automated setup. This technique allows all elements to be tuned simultaneously rather than sequentially. However, in order to tune the filter using this technique, adequate initial optimization variable values are required to prevent the system from running into local minimum or failing to converge to the proper solution. This case arises when the filter is highly detuned. To overcome this problem, a coarse tuning technique based on phase format of input reflection coefficient of the filter is proposed in this thesis. In this method, resonators are tuned by bringing successive resonators to resonance, while the phase passes through the $\pm 180^\circ$ and 0° crossing at the center frequency. At the end of each sequence cross coupling is mapped across the entire range of its motion. Tuned cross coupling is recognized by measuring the return loss of the filter. Written codes based on this technique are able to guide the robot through coarse tuning process in a short time. Then, the circuit model parameter extraction technique is practically implemented by automated setup to complete the tuning.

This proposed tuning system has been validated through experimental results. Results showed that utilizing the backlash compensation solution enables the robot to use accurate '1' arcminute rotational resolution to achieve less than '5' KHz frequency deviation in obtained filter response. This cannot be achieved with manual tuning. Elimination of mapping the tuning elements throughout the lash reduced the overall tuning time by 11 minutes and 23 seconds in an average of twenty tuning attempts. Absolute positioning system of the RoboCAT successfully detected the faulty tuning attempts caused by an error in CAT software. This enabled the robot to be function without supervision. Fabricated comprehensive coaxial/screw nut driver fitted with the designed SCARA, Cartesian, and multi-armed robots in order to tune the different filter types in the company without need for design updates.

Several tuning attempts have been performed by the automated setup utilizing the proposed coarse tuning technique. Obtained filter response at the end of each tuning was very close to the ideal filter response. Therefore perfect initial values for the variables to be optimized were provided for the fine tuning program. This reduced the fine tuning time from 13 minute to less than 32 seconds and prevented the system from running into local minimum. Successful results in all tuning runs showed

effectiveness of the tuning technique. Obtained tuning times via created setup was compared with traditional manual tuning attempts. RoboCAT achieved a tuned filter with an average time of 6 minutes compared to the manual tuning approach which took 42 minutes. These results obtained from 20 tuning attempt on a six-pole cross-coupled filter while the filter was tuned to channel 40 at a center frequency of 613.5 MHz.

TABLE OF CONTENTS

Declaration	i
Acknowledgement.....	iii
Abstract	v
Table of Contents	ix
List of Figures	xii
List of Tables.....	xv
Chapter 1 Introduction	1
1.1 Motivation.....	1
1.2 Objectives and Approach	3
1.3 Summary of Chapters.....	4
1.4 Original Contributions	6
Chapter 2 Literature Review	7
2.1 Introduction.....	7
2.1.1 Mechanical Design and Automation.....	7
2.1.2 Tuning Techniques.....	7
2.2 Mechanical Design and Automation.....	8
2.2.1 Single-armed Tuner.....	8
2.2.2 Multi-armed Tuner	9
2.3 Tuning Techniques.....	10
2.3.1 Sequential Tuning Technique Based on Group Delay Format	10
2.3.2 Time-domain Technique	12
2.3.3 Fuzzy Logic Technique.....	14

2.3.4	Poles and Zeros Technique	16
2.3.5	Circuit Model Parameter Extraction Technique	18
2.4	Conclusion	20
2.4.1	Mechanical Design and Automation	20
2.4.2	Tuning Techniques	21
Chapter 3 Mechanical Design and Automation		25
3.1	Introduction	25
3.2	Design Dimensions	26
3.3	Coaxial Screw/nut Driver	29
3.4	Mechanical Stoppers	34
3.5	Belt and Pulley System	35
3.6	Tuner Head	36
3.7	Multi-armed Robot	38
3.8	Single-armed Cartesian Robot	39
3.9	Single-armed SCARA Robot	40
3.10	Vertical Tuning Elements (Rod)	42
3.11	Mechanical Backlash	44
3.12	Positioning System	46
3.13	Conclusion	48
Chapter 4 Coarse and Fine Tuning		51
4.1	Introduction	51
4.2	Experimental Setup	53
4.3	Coarse Tuning	55
4.3.1	Coarse Tuning of Resonators	55
4.3.2	Coarse Tuning of Inter-resonator Couplings	60
4.3.3	Coarse Tuning of Cross Couplings	62
4.4	Fine Tuning	67
4.5	Conclusion	72
Chapter 5 Conclusion		73
5.1	Automation	73
5.2	Tuning	74
5.3	Future Work	75
References		76
APPENDIX A: Designed and Fabricated Mechanical Parts		81

APPENDIX B: Codes and Algorithm..... 93

LIST OF FIGURES

Figure 1.1	Manual tuning setup.	3
Figure 2.1	Single-armed Cartesian robot made by COM DEV Ltd. [13].	8
Figure 2.2	Multi-armed robot tuner [14].	9
Figure 2.3	Lowpass prototype circuit [18].	11
Figure 2.4	Input reflection coefficient of an eight-pole filter tuned by group delay technique [18].	12
Figure 2.5	The response of a band-pass filter to tuning the resonators [20].	13
Figure 2.6	Effect of increasing first coupling factor [20].	14
Figure 2.7	Output membership function for the Four-pole filter [22].	15
Figure 2.8	Experimental and extracted performance using Fuzzy Logic [22].	16
Figure 2.9	The equivalent circuit of N-cascaded resonators terminated in a short circuit [29].	18
Figure 2.10	A generalized model for coupled resonator filters.	19
Figure 3.1	Locking torque specifications for filter 8PPXX110E made by RFS Company.	27
Figure 3.2	Smallest filter in the company (RFS) with high density of tuning elements.	28
Figure 3.3	Size comparison of 6-pole cross-coupled filter and NEMA 17 actuator.	29
Figure 3.4	Coaxial screw/nut driver 3D drawing.	31
Figure 3.5	Coaxial screw/nut driver 2D drawing.	32
Figure 3.6	Mechanical stopper mounted on inner cylinder.	34

Figure 3.7	Timing belt and pinion pulleys.	35
Figure 3.8	Timing belt and pinion pulley geometry.	36
Figure 3.9	(a) Compatible driving surfaces with automated tuner (b) 3D drawing of replaceable tuner head (c) Transparent view of tuner head.	37
Figure 3.10	Multi-armed robot formation with two aluminium plates.	38
Figure 3.11	One level multi-armed tuner (diametric view).	39
Figure 3.12	Single-armed Cartesian tuner.	40
Figure 3.13	Single-armed SCARA tuner (isometric front view).	41
Figure 3.14	Single-armed SCARA tuner (trimetric back view).	42
Figure 3.15	8-pole RF filter with tuning rods.	43
Figure 3.16	NEMA 17 linear stepper motor with ball screw.	43
Figure 3.17	Mechanical concept for tuning rods.	44
Figure 3.18	Tuning results related to first resonator of a 6-pole filter while anti-backlash code is disabled (blue line), anti-backlash code is enabled (red line).	46
Figure 3.19	Tuning results related to first resonator of a 6-pole filter: using CAT data only (blue line), using CAT data only while motor speed is accelerated (green line), using Prepositioning code plus CAT data (red line).....	47
Figure 4.1	6-pole cross-coupled filter.	53
Figure 4.2	Experimental setup.	54
Figure 4.3	Block diagram of experimental setup (RoboCAT).....	54
Figure 4.4	Phase format of the input reflection coefficient S_{11} while all the resonators are highly detuned.....	57
Figure 4.5	Measured filter response by VNA (Resonator1 is brought to resonance while rest of the resonators are highly detuned).	57
Figure 4.6	Measured filter response by VNA (Resonators 1 and 2 are brought to resonance while resonators 3, 4, 5, 6 are highly detuned).	58
Figure 4.7	Measured filter response by VNA (Resonators 1, 2, 3 are brought to resonance while resonators 4, 5, 6 are highly detuned).	58
Figure 4.8	Measured filter response by VNA (Resonators 1, 2, 3, 4 are brought to resonance while resonators 5, 6 are highly detuned).	59
Figure 4.9	Measured filter response by VNA (Resonators 1, 2, 3, 4, 5 are brought to resonance while resonator 6 is highly detuned).	59

Figure 4.10	Filter response after one run coarse tuning of resonators while cross coupling and couplings are mistuned.....	60
Figure 4.11	Filter response after implementing step 4 of coarse tuning for resonators while coupling 1 is completely withdrawn.	61
Figure 4.12	Filter response after implementing step 4 of coarse tuning for resonators while coupling 1 is fully inserted into the filters frame.....	61
Figure 4.13	Filter response after implementing step 4 of coarse tuning for resonators while coupling 1 is set to the middle of the range that affect the resonator 1.....	62
Figure 4.14	Filter response after implementing coarse tuning for resonators while cross coupling is slightly rotated.....	63
Figure 4.15	Filter response after 4 run sequential tuning while tuning resolution for cross coupling is set to 5 degree.....	64
Figure 4.16	Filter response after 6 run sequential tuning while tuning resolution for cross coupling is set to 4 degree.....	65
Figure 4.17	Filter response after 8 run sequential tuning while tuning resolution for cross coupling is set to 3 degree.....	65
Figure 4.18	Filter response after 10 run sequential tuning while tuning resolution for cross coupling is set to 2 degree.....	66
Figure 4.19	Filter response after 12 run sequential tuning while tuning resolution for cross coupling is set to 1 degree.....	66
Figure 4.20	Circuit model with all possible cross couplings.....	68
Figure 4.21	Measured filter response by VNA after implementing fine tuning.....	69
Figure 4.22	Measured filter response by BCAT while RoboCAT pass the tuning process.....	70
Figure 4.23	Measured filter response by BCAT while higher resolution in tuning is utilized by RoboCAT.....	70
Figure 4.24	Required tuning time for 6- pole multi-cavity filter in 20 run.....	71

LIST OF TABLES

Table 3.1 Description of the parts in coaxial screw/nut driver.	33
--	----

Chapter 1

INTRODUCTION

1.1 Motivation

Introduction of Digital television (DTV) is the biggest evolution in the broadcasting industry since the innovation of the colour television almost 60 years ago. This technology not only increases the sound quality and picture resolution of TV channels but also enables the television stations to broadcast far more channels with distinct simulcast programs. Digital signal conveying brings the ability to reallocate the radio spectrum which allows the government to auction off the freed spectrum to telecommunication industries for new services such as wireless technology and mobile broadband. Therefore, in recent years, governments set new policies to encourage the broadcasters with some funds to launch the digital switchover process. In Australia, switchover from analog to digital television broadcasting began in 2010 and completed on December 2013 [1]. This transmission increases the demand and interest for RF broadcasting equipment. RF broadcasting filters are one of these equipment. To overcome the escalating orders of RF filters, filter companies set the new development plan to fully automate their filter manufacturing systems, from order to delivery.

Due to material and manufacturing tolerances, tuning RF filters become an indispensable postproduction process; therefore, they are constructed with incorporated tuning elements which are generally dielectric screws or lead screws that allow the filter to be tuned. Tuneable functionality of these filters also is

necessary to make them utilizable for a broad range of filtering applications. With this structure, technologists are able to modify the resonance frequency of the resonators and the inter-resonator couplings by changing the penetration depth of tuning elements inside the cavities of the filter to match the final response of the product to desire one. However, adjusting each of these sensitive tuning elements can cause a significant change in the measured filter response; for example in a 6-pole cross-coupled filter (600 MHz center frequency) each micrometre penetration of resonator lead screw inside the cavity translates into a frequency shift close to 20 KHz. Therefore, the tolerance of about 300 micrometres would shift the center frequency over the bandwidth of 6 MHz. Thus, to meet the stringent requirement of the broadcast market, high mapping resolution in tuning is necessary which makes the manual adjustment process a tedious and time consuming job as tuning of complex filters cause disorder in production line.

The complexity in tuning depends on the filter configuration and the employed technology. By increasing the number of cross coupling elements in a filter, interaction between tuning elements increases. Therefore repetitive alignment of a tuning screw is required. High accuracy in tuning and interaction between tuning screws makes the tuning process very slow which in some complex filters may take several hours.

In the established method, these filters are tuned manually by highly skilled personnel, having no instruction throughout the process. These skilled technologists have the expertise to monitor the filter response on a Vector Network Analyser (VNA) and determine which element to tune, what direction and how far in order to get the desired response. As this heuristic knowledge in expert tuners is not structured, it is not teachable to novice personnel; thus, always there is a need for highly specialized staff which increase the final product cost.

In early nineties, several studies have been performed to introduce instructions for manual tuning of different types of filters which could be used by new technicians [2-11]. By rapid augmentation of these instructions, it is now possible to write a software code to guide the operator through the tuning procedure without having any experience. Written Computer Aided Tuning (CAT) software can guide the operator through the tuning process with instruction on manual tuning.

While the CAT system improves the predictability of tuning, it is still a manual process that is not suited to higher volume production. In response to this throughput challenge, Radio Frequency Systems company (RFS Pty Ltd.) is trying to develop a number of automated filter manufacturing systems that make the leap from conventional ‘trial and error’ manual filter tuning to automatic robotic tuning and set a new standard for filter production. The aim of this project, which is supported by RFS, is to design and manufacture RoboCAT (Robotic Computer-aided Tuning) system.

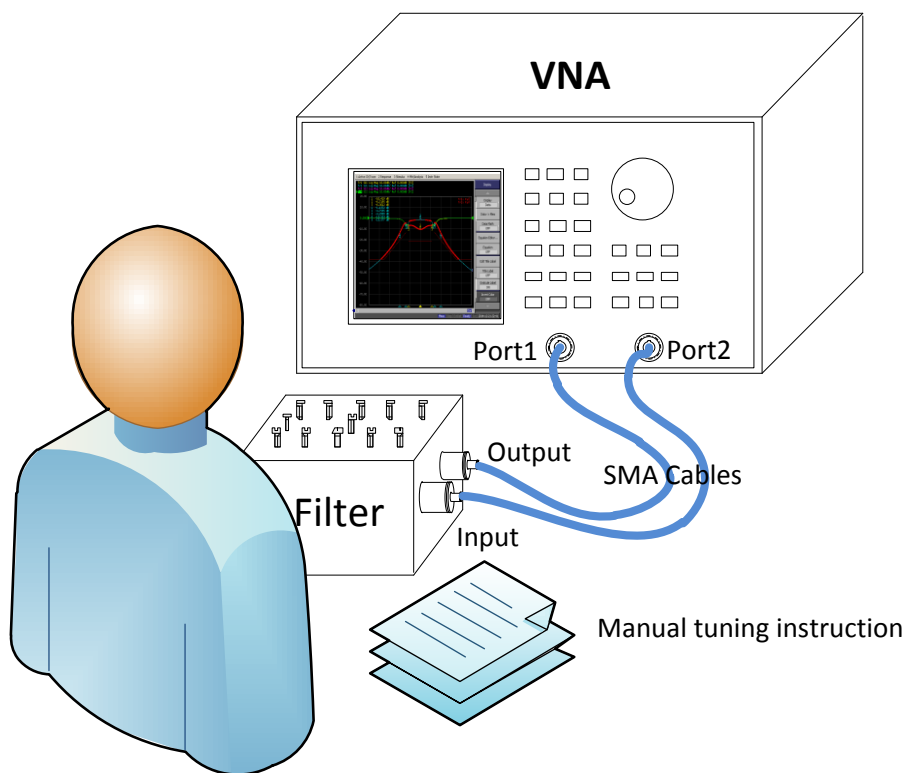


Figure 1.1 Manual tuning setup.

1.2 Objectives and Approach

This study aims to create a fully automated tuning robot which eliminates the intervention of the human in any stage of the process. This would lead to better management of staff and also eliminates the required training stage for novice personnel to learn manual tuning techniques. For this purpose, first, mechanical tuner with incorporated locking mechanism needs to be designed to complete the tuning

and locking. Locking mechanism is needed for locking the lead screws of the tuned filter by high torque. This will increase the product consistency during delivery stage to costumers. Second, suitable tuning instruction for robot needs to be defined. Written codes based on this instruction should be able to guide the robot through the tuning stages independently.

Another aim of this project is to reduce the tuning time as to be able to meet the high order production line requirement. Therefore multi armed tuner design is considered for the robot. As multi armed tuner is capable of adjusting the tuning elements simultaneously, reduction in tuning time is expected. By elimination of dependency to technologists and speeding up the tuning process, final production cost would be significantly reduced.

There are different types of filters in the current production line. By improvement of design technology of filters, new products are introduced rapidly. Manufacturing different tuner arms for each one of these filters is costly. Therefore, comprehensive mechanism for automated tuner needs to be designed to enable the robot to interface with different type of filters in the company with just a small change.

The other objective of this project is to compensate the unwanted mechanical backlash effect. It is unavoidable for nearly all reversing mechanical couplings as there are manufacturing errors. Gaps between two assembled parts cause clearance or lost motion in rotational movement. Accumulation of rotational play between the assembled parts of the automated tuner causes a considerable rotational backlash. Since this amount is way higher than a required tuning resolution, considerable time would be wasted to map the entire backlash in each change of direction during tuning. This is because the movement of a tuner through the backlash won't affect the filter response. By measuring the backlash amount via encoders, unwanted effect of it can be counteracted.

1.3 Summary of Chapters

The layout of the thesis is as follows:

In this chapter (Chapter 1), the motivation factors for taking up this research are discussed. It also focuses on the objective of the research and provides information

about the approach taken towards achieving the final outcome of this particular research.

Chapter 2 focuses on the literature review and consists of two sections. Section one reviews the advantages and disadvantages of different mechanical designs in the automated tuners introduced. Consideration of the advantages of these tuners and augmenting them based on industrial usage, leads to the design stage of this study. The aim of the second section of the literature review is to identify a suitable tuning technique among the proposed techniques. Five existing tuning techniques in the literature are briefly described and evaluated. It can be concluded from the literature that the circuit model parameter extraction technique [12] is the only ideal tuning technique to be implemented by automated setup. This technique allows all the elements to be tuned simultaneously, rather than sequentially. However, in order to tune the filter using this technique, adequate initial optimization variable values are required to prevent the system from running into local minimum or failing to converge to the proper solution. This case arises when the filter is highly detuned. This defect in the chosen technique is addressed to be solved in Chapter 4. It is expected that, by addressing the existing problems associated with the chosen technique, reliable standalone tuning code for the robot will be written.

Chapter 3 introduces a comprehensive customized robot tuner arm. This concludes the creation of the experimental setup in this study. Aspects of the automated tuner include: (1) backlash compensation to increase the tuning resolution and reduce the tuning time; (2) absolute positioning in order to have a second feedback source for robot codes along with obtained data from Computer-Aided Tuning (CAT) software; and (3) fault recognition capability to detect any potential error in CAT codes in the early stage.

Chapter 4 deals with finding a proper tuning instruction. With a complete tuning instruction, it is feasible to write standalone tuning code for an automated tuning system. As mentioned, the circuit model parameter extraction technique is the only ideal tuning technique to be implemented by automated setup. However, in order to tune the filter using this technique, adequate initial optimization variable values are required. To overcome this problem, a coarse tuning technique based on the phase format of the input reflection coefficient of the filter is proposed, to deal with highly

detuned filters. In this method, resonators are tuned by bringing successive resonators to resonance, while the phase passes through the $\pm 180^\circ$ and 0° crossing at the center frequency. At the end of each sequence, cross coupling is mapped across the entire range of its motion. Tuned cross coupling is recognized by measuring the return loss of the filter. Written codes based on this technique are able to guide the robot through the coarse tuning process in a short time. Then, the circuit model parameter extraction technique is practically implemented by automated setup to complete the tuning. Chapter 5 concludes this thesis, together with suggestions for potential future work regarding finding an artificial neural network solution for the fine-tuning process.

1.4 Original Contributions

This work has made the following contributions:

- An automated coaxial screw/nut driver with an absolute positioning system and backlash compensation functionality for filter tuning applications.
- A novel coarse tuning technique as a prerequisite for the circuit model parameter extraction method in the literature, to provide perfect initial values for the variables to be optimised by this technique.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

2.1.1 Mechanical Design and Automation

To adjust the tuning elements and finalize the locking procedure, suitable driver mechanisms attached to the actuator are required. Design of these mechanisms is important to achieve automation goals. Nevertheless, there are only a few quasi-reliable mechanical solutions for industrial usage in the literature presented. Yet, these machines are customized for only a specific filter type. Other studies in this field [13, 14] focus on optimizing tuning software rather than investigating a complete solution for automation. In these studies, simple actuators and tools are utilized only to test the feasibility of the CAT software in the automation process. However, some of the insufficiencies with the tuning software can only be solved by applying mechanical solutions. Furthermore, suitable mechanical design in the tuning procedure of high degree microwave filters with high sensitivity is crucial. The first section of this chapter investigates the automated tuners introduced in the literature.

2.1.2 Tuning Techniques

In the mid-1990s, several studies were undertaken to simulate the empirical tuning method of experienced technologists to provide a theoretical structure that can be used by unskilled tuners to achieve similar results to those of expert tuners. These structures can be categorized into five separate techniques. The second section of this chapter reviews the augmented form of these techniques, together with their

advantages and disadvantages. The contribution of different works, related to each technique, is taken into account. A written tuning algorithm (computer-aided tuning software) based on these techniques can be used to guide the human operator and the automated machine through the tuning procedure.

2.2 Mechanical Design and Automation

2.2.1 Single-armed Tuner

Figure 2.1 illustrates the most complete version of the automated setup found in the literature. This was introduced by COM DEV International Ltd in 2006 [15] and is used for tuning microwave filters and multiplexers for communications satellites. This automated machine uses the Cartesian table to move the single-armed tuner. The coaxial screw/nut driver on this design allows for simultaneous locking and tuning. This setup is still the most augmented design, as it is capable of finalizing the tuning by lockdown procedure.

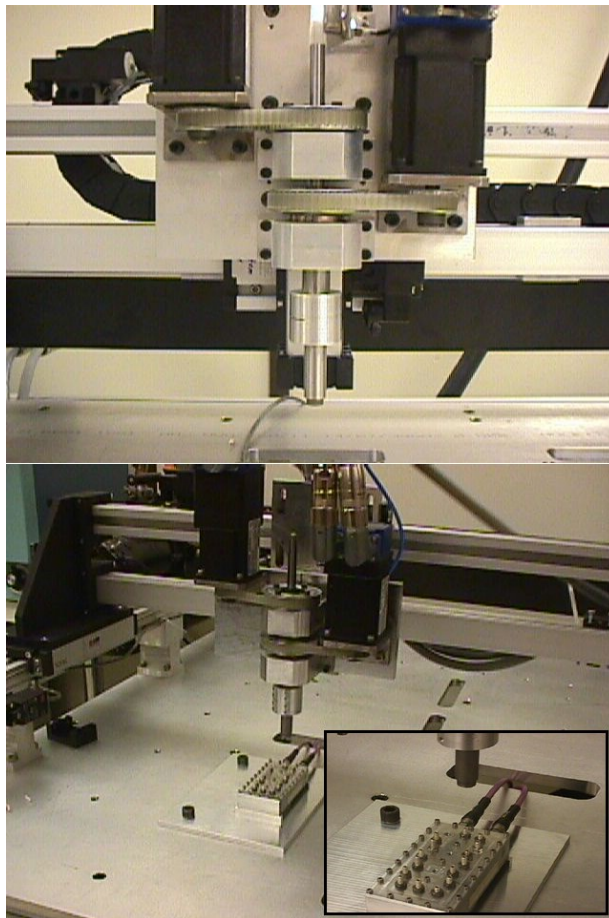


Figure 2.1 Single-armed Cartesian robot made by COM DEV Ltd. [13].

2.2.2 Multi-armed Tuner

The benefits of using a multi-armed design similar to the design shown in Figure 2.2 are also considerable. In this setup, the stepper motors shafts are connected to screws by custom-designed flexible leads. Universal mounting brackets are designed to arbitrarily hold and position the motors on a main aluminium plate. This enables the tuner to have accessibility in three dimensions to target any scattered screws on the particular device. The multi-arm concept of this setup can help in speeding up the tuning process [16].

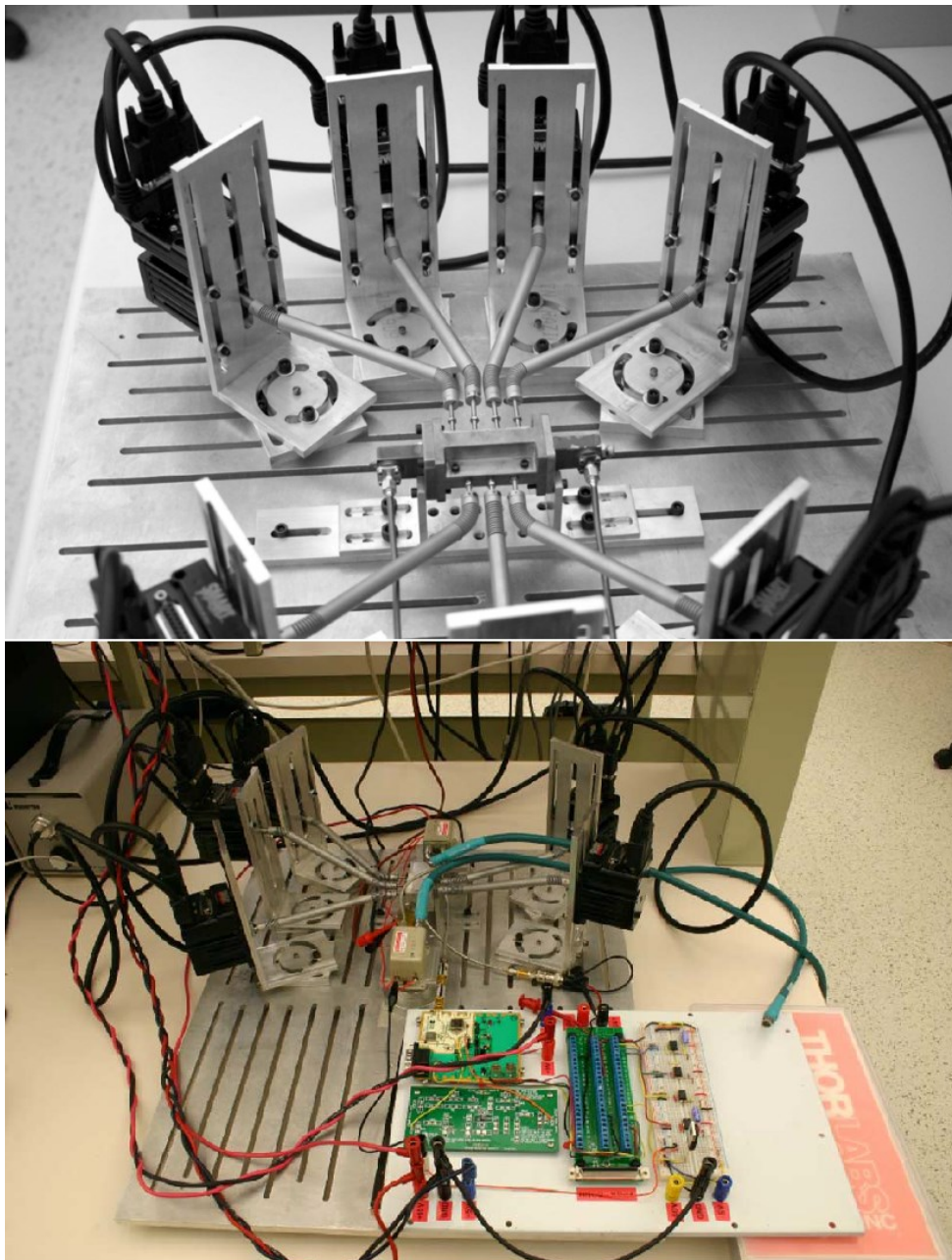


Figure 2.2 Multi-armed robot tuner [14].

2.3 Tuning Techniques

2.3.1 Sequential Tuning Technique Based on Group Delay Format

The alternating short-circuit technique of Dishal [17] was the basis for this approach, which was later modified by Ness[18]. To tune the filter by this technique, in the first step, group delay values $\Gamma_{d1}, \Gamma_{d2}, \dots$ for the given filter specifications need to be calculated. Γ_{d1} is the group delay value of the filter when all the resonators, except the first, are considered to be short-circuited. Γ_{d2} is the group delay value of the filter when all the resonators are considered as short-circuited except the first and the second, and so on. Ness [18] demonstrates that group delay $\Gamma_d(\omega)$ of the reflected signal, S_{11} , can be obtained by using a lowpass prototype circuit Figure 2.3 and implementing lowpass-to-bandpass transformation, given as

$$\omega' \rightarrow \frac{\omega_0}{(\omega_2 - \omega_1)} \left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} \right) \quad (2.1)$$

As a result $\Gamma_d(\omega)$ can be driven as

$$\Gamma_d(\omega) = - \frac{\omega^2 + \omega_0^2}{\omega^2(\omega_2 - \omega_1)} \frac{\partial \phi}{\partial \omega'} \quad (2.2)$$

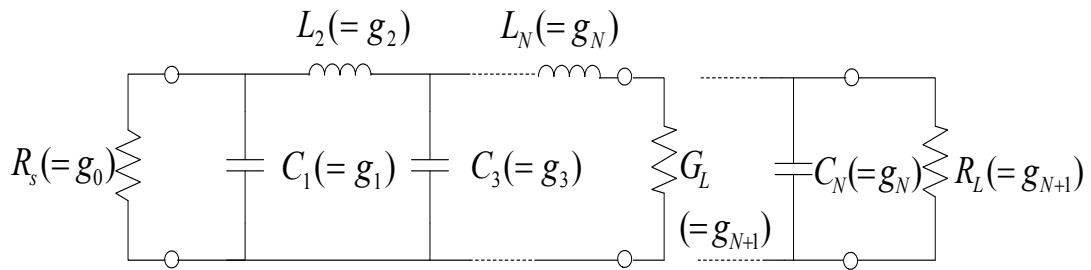
Then, the first theoretical group delay value Γ_{d1} at the desired center frequency ω_0 can be calculated by disconnecting g_2, g_3, \dots in the lowpass prototype circuit and by using Equation (2.2)

$$\Gamma_{d1}(\omega_0) = \frac{4g_0g_1}{(\omega_2 - \omega_1)} \quad (2.3)$$

The second theoretical group delay value Γ_{d2} , which corresponds to resonator 2, can be driven at the same center frequency, while the second element g_2 is short-circuited to the ground.

$$\Gamma_{d2}(\omega_0) = \frac{4g_2}{g_0(\omega_2 - \omega_1)} \quad (2.4)$$

A similar procedure is followed as successive resonators are short-circuited to drive the theoretical group delays in terms of the lowpass g_k values.



2

Figure 2.3 Lowpass prototype circuit [18].

In the second step, resonator 1 and input coupling are practically adjusted to set the measured group delay to the calculated group delay value $\Gamma_{d1}(\omega_0)$, while all resonators except resonator 1 are short-circuited. Then, resonator 2 and the coupling between resonator 1 and 2 are adjusted to set the measured group delay to the obtained group delay value $\Gamma_{d2}(\omega_0)$, while all resonators except resonator 1 and 2 are short-circuited. This sequence continues till the last resonator. The last resonator and output coupling are tuned based on specified return loss.

In this approach, group delay values are calculated while some resonators are considered as completely short-circuited. However, in a real-case scenario, complete short-circuiting of the resonators is not feasible. Consequently, matching the filter response to measured theoretical values does not lead to a precise result at the end of the tuning process. Moreover, if the couplings between resonators (e.g. coupling between Resonator 1 and Resonator 2) are so strong that tuning one resonator (e.g. Resonator 1) detunes the adjacent resonator (e.g. Resonator 2) which is already tuned), slight unsymmetrical results would appear on the measured results. As shown in Figure 2.4, propagation of this error leads to considerable mistuning at the end of the process. To improve the result, one alternative is to observe the filter response and readjust the preceding resonator to make the filter response more symmetrical around the center frequency. In an automated system, this requires the intervention of a human with experience of which resonator to readjust.

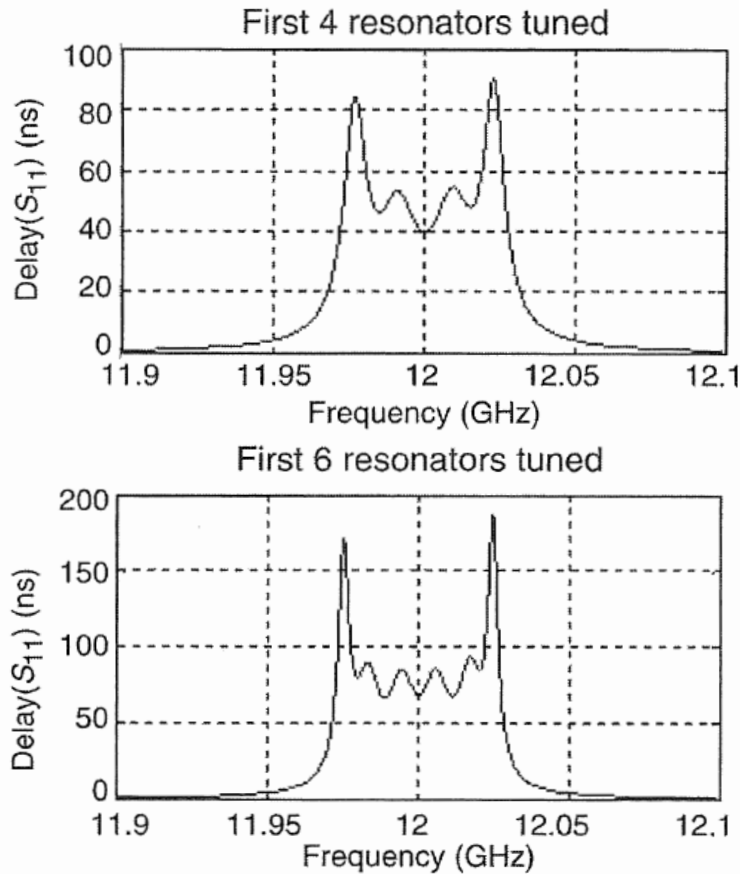


Figure 2.4 Input reflection coefficient of an eight-pole filter tuned by group delay technique [18]

2.3.2 Time-domain Technique

The time-domain response of the input reflection coefficient of the filter is the inverse Fourier transform of the frequency response of S_{11} . Thus, distinct dips and peaks in the time-domain response of the filter for $t > 0$ correspond to the resonators and the inter-resonator couplings respectively. As the time domain response indicates, the reflected energy, dips and peaks for $t < 0$ are meaningless and should be ignored during the tuning process. Therefore, the first dip and the first peak with the shortest time delay (closest to $t = 0$) are related to the first resonator and the first coupling. If the frequency sweep is centered at the desired center frequency of the band-pass filter, the ideal position of the resonators can be found by minimizing the relative dips successively [19–21]. Figure 2.5 shows the time-domain and frequency-domain traces for two conditions. The darker trace shows the filter response, with the second resonator mistuned 2% low in frequency, and the lighter trace shows the ideal filter response.

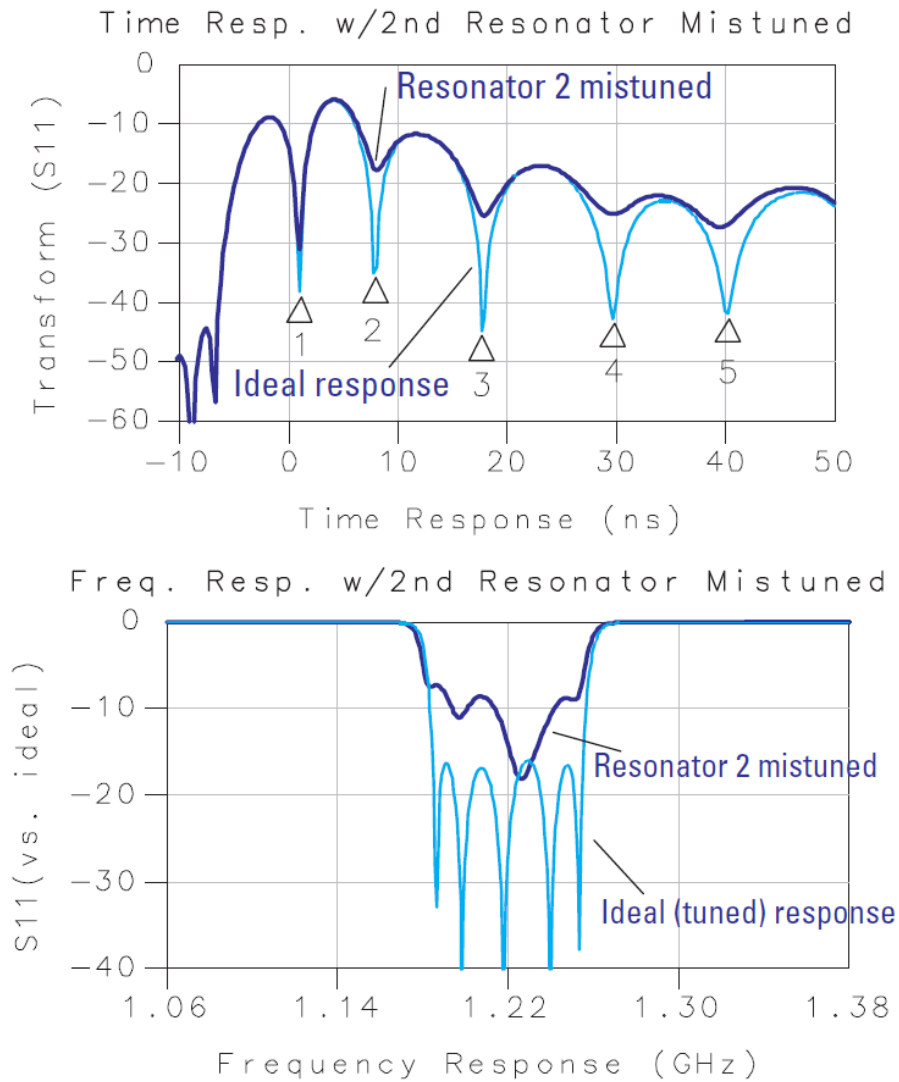


Figure 2.5 The response of a band-pass filter to tuning the resonators [20].

However, tuning the couplings requires the availability of an ideal time-domain response to use as a reference to adjust the peak levels. The target time-domain response can be obtained by measuring the time-domain response of a ‘golden’ filter (tuned filter), or by using Fourier transform of the S_{11} frequency response of an ideal filter and simulating it. Figure 2.6 shows the S_{11} response in both the frequency and time domains, both before and after changing the first coupling factor. The lighter trace shows the ideal response of the filter and the darker trace is related to the same filter, with a 10% mistuned coupling factor.

This method offers a simple tuning procedure to tune the band-pass filter and is generally preferred for manual tunings. However, it does not deal with adjusting the

input/output coupling elements and also is not applicable for tuning elliptic filters that have coupling between nonadjacent resonators.

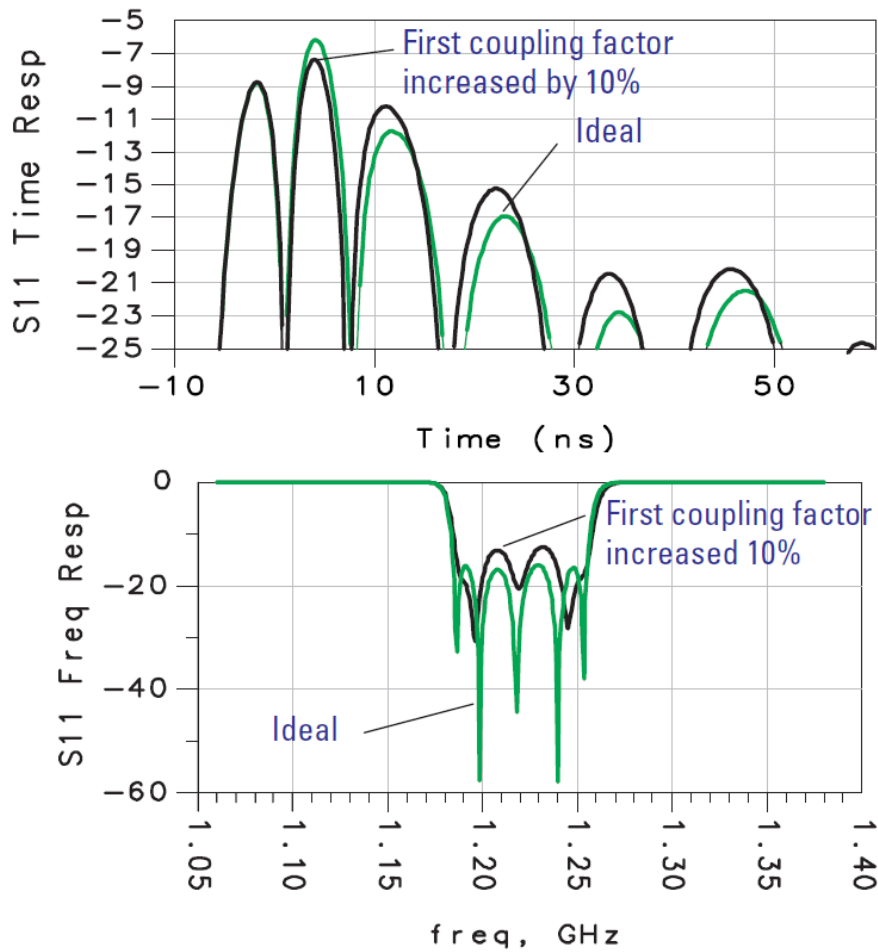


Figure 2.6 Effect of increasing first coupling factor [20].

2.3.3 Fuzzy Logic Technique

The idea of the fuzzy logic technique [16], [22-24] stems from the fact that, within the manual tuning process, technicians use the concept of sets to define the coupling values. These sets for the filter coupling elements can be described as: very small, small, large, and very large. Some technicians have the expertise to observe the measured data and conclude to which of the defined sets a specific coupling element belongs. By assigning the proper set to the right coupling element, they align the relative tuning screws to perfectly tune the filter. This strategy cannot be interpreted by classical Boolean logic. In classical logic, sets are defined in a crisp manner, i.e. an element either belongs to a set or does not belong to it. However, interpreting the technologists' manual tuning method by using fuzzy sets converges to a proper

solution. In Fuzzy Set Theory (FST) [25], a membership value between ‘0’ and ‘1’ is assigned to each element of the set: ‘0’ means the element does not belong to the set at all, whereas ‘1’ means the element belongs wholly to that set. Fuzzy logic interprets the numerical data as linguistic rules. The extracted rules will then be used as a kind of system specification to calculate the output values of the system [26-28].

The main advantage of this technique lies in the capability to integrate human experience, a theoretical model and measured data into one model. However, it is necessary to implement too many membership functions for the elements of the coupling matrix, in order to achieve a fair resolution in the final results. This means greater computational complexity and consequently more computational delays. In fact, this technique is suitable to be used as a rough tuning method by taking into account the complexity of the required algorithm. Figure 2.7 depicts the output membership function of the coupling element y_1 , i.e. m_{12} .

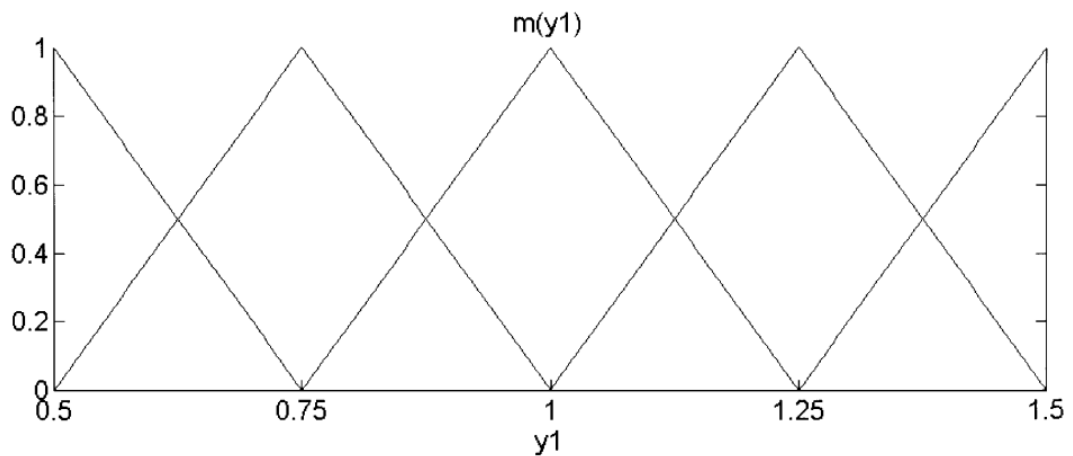


Figure 2.7 Output membership function for the Four-pole filter [22].

Figure 2.8 shows the results for tuning a four-pole band-pass Chebyshev filter by this technique. In this example, m_{12} , m_{23} , m_{34} are the coupling matrix elements that need tuning. Input/output coupling values are assumed to be fixed. As is clear, there is a slight deviation between experimental and extracted performance using this technique, which requires fine tuning by a technician.

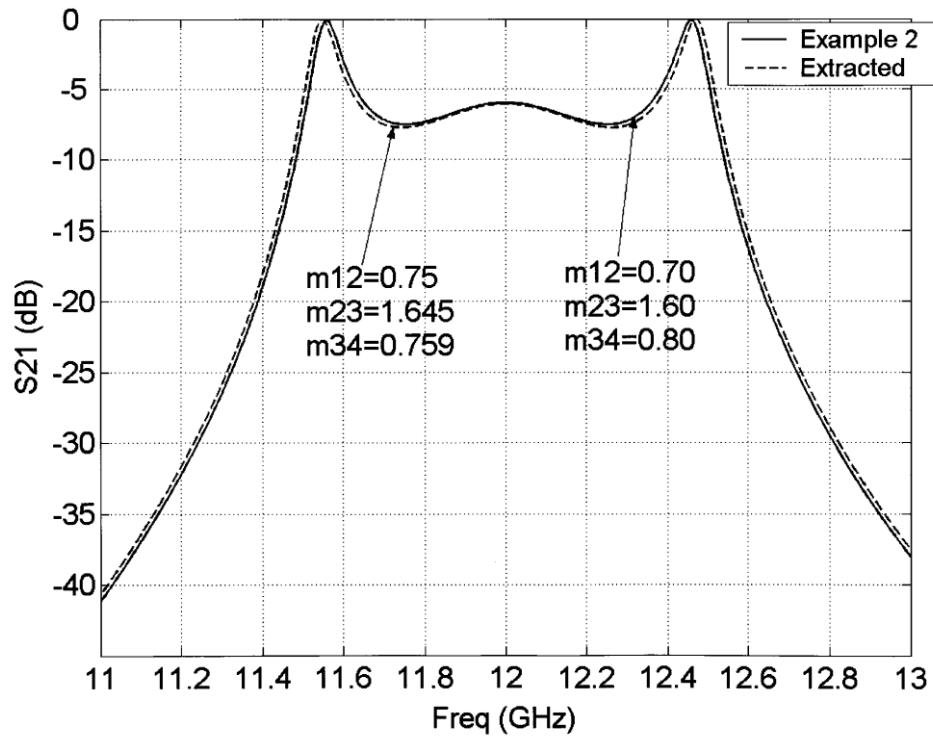


Figure 2.8 Experimental and extracted performance using Fuzzy Logic [22].

2.3.4 Poles and Zeros Technique

The tuning procedure starts with the necessary calibration to determine the proper location of the reference plane for accurate phase measurement. In practice, the procedure is to insert all the filter tuning screws deep into the cavities so that their resonant frequencies are far from the desired resonant frequencies of the individual cavities (the detuned condition). This condition is observed on the polar display of a network analyser when the input reflection coefficient appears as a single spot, while the frequency is being swept several times the bandwidth around the desired center frequency. The phase reference plane is then adjusted to achieve the ‘best’ spot on the sweep, which will be used to correspond to the ‘0’ position of the reflection coefficient [29]. Then, the tuning procedure can be summarized as the following steps:

Step 1- The measurement and adjustment of the input coupling are performed by bringing the first resonator into resonance, while all the remaining resonators in the network are still in the detuned condition. Following the procedures reported in [30], the input coupling resistance can be determined. Possible adjustment of input coupling to the desired value is performed by changing the iris dimension or probe

position, depending on the coupling schemes adopted. The same approach is applied for the measurement and adjustment of the output coupling resistance.

Step 2- All the resonators in the network should be brought into resonance. For this purpose, the output port needs to be terminated with a short circuit. Then, relative tuning screws should be adjusted until all the poles and zeros are shown on the polar display of the vector network analyser (VNA).

Step 3- Measurement of the prototype network in Figure 2.9 is made. All the poles and zeros are recorded from the VNA. The inter-resonator couplings and resonant frequencies of each individual resonator (n resonance frequencies and $n - 1$ couplings) [29] are extracted by using Equations (2.5) and (2.6), which provide a simple and direct correlation between the filter parameters (the inter-resonator couplings and the resonant frequencies of individual resonators) and the poles and zeros of the input impedance of the short-circuited network. Let us consider these extracted values as $\{M'_{12}, M'_{23}, \dots, M'_{n-1,n}, F'_{01}, F'_{02}, \dots, F'_{0n}\}$. The extracted values are different from the desired values at this point.

$$\omega_{0i}^2 = \frac{\prod_{t=1}^{n-i+1} \omega_{zt}^{(i)2}}{\prod_{q=1}^{n-i} \omega_{pq}^{(i)2}} \quad i = 1, 2, \dots, n \quad (2.5)$$

$$m_{i,i+1}^2 = \sum_{t=1}^{n-i+1} \omega_{zt}^{(i)2} - \sum_{q=1}^{n-i} \omega_{pq}^{(i)2} - \omega_{0i}^2 \quad i = 1, 2, \dots, n \quad (2.6)$$

Step 4- By replacing the first extracted value with the desired value M_{12} , a new set would be obtained as $\{M_{12}, M'_{23}, \dots, M'_{n-1,n}, F'_{01}, F'_{02}, \dots, F'_{0n}\}$. This set is used to synthesize the poles and zeros. The new set of poles and zeros is then used as the criterion for the adjustment of the coupling between resonators 1 and 2. The underlying concept here is that by adjusting the tuning screw that corresponds to M_{12} , if the poles and zeros are placed in the recorded positions, the coupling between resonators 1 and 2 would be set to the desired value. The same procedure is followed for all the inter-resonator couplings, until all are set to the desired values. This procedure is individual and sequential for these tuning elements. Thus, $n - 1$ time tuning attempts are required to complete this step.

Then step 3 and step 4 would be repeated until the whole filter is completely tuned. Overall, this is $(2n - 1)$ tuning attempts, after input/output couplings have been set correctly. This iterative tuning procedure is not a good solution for mass production of microwave filters as several tuning attempts are required to finalize the tuning. Also, in order to bring the resonators to resonance, the output port of the filter needs to be terminated with a short circuit, which is not suitable for automated tuning setup.

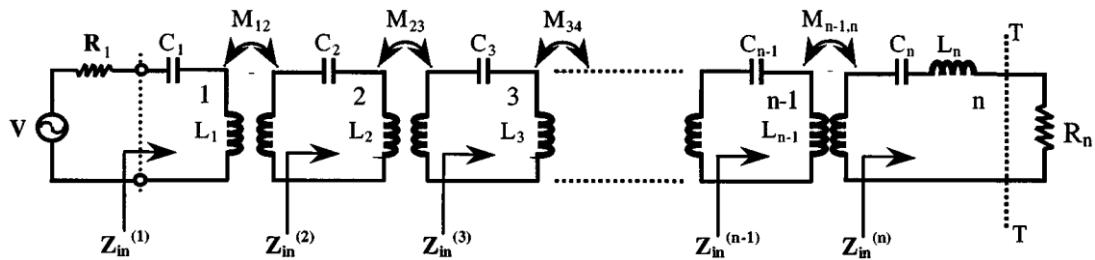


Figure 2.9 The equivalent circuit of N-cascaded resonators terminated in a short circuit [29].

2.3.5 Circuit Model Parameter Extraction Technique

The concept of this approach is to optimize the parameters of the circuit model such that the circuit model response best fits the measured response of the filter. After that, by comparing extracted parameters and the parameters of the ideal filter it will be determined that which tuning element requires tuning [12], [31–37]. Although there are several circuit models that can be applied in this technique, the coupling matrix circuit model has an advantage since its coupling elements are directly related to the depth of the tuning elements. Once one of these coupling elements is calculated and found to deviate from ideal value, it can be easily adjusted back by alignment of the corresponding tuning element. Considering the generalized model circuit depicted in Figure 2.10, the tuning procedure can be summarized as follow:

Step 1- Performance of the filter being tuned needs to be calculated.

Step 2- Based on coupling matrix circuit model; coupling elements should be extracted by optimization for best fit with measured response of the filter. This can be achieved by using Equations (2.7)-(2.11) where I is unity matrix and R is diagonal matrix with all elements zero except $R_{11} = R_1$ and $R_{nn} = R_2$. Objective function to be minimized can be defined as Equation (2.12) while measured performance of the filter should be sampled at several frequencies within the band.

$$M = \begin{bmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nn} \end{bmatrix} \quad (2.7)$$

$$S_{21} = -2j\sqrt{R_1 R_2} [A^{-1}]_{n1} \quad (2.8)$$

$$S_{11} = 1 + 2jR_1 [A^{-1}]_{11} \quad (2.9)$$

$$A = \lambda I - jR + M \quad (2.10)$$

$$\lambda = \frac{f_0}{BW} \left(\frac{f}{f_0} - \frac{f_0}{f} \right) \quad (2.11)$$

$$\varphi = \sum_{freq} \sum_{i=1}^2 \sum_{j=1}^2 \left(abs(S_{ij}^{model}) - abs(S_{ij}^{measured}) \right) \quad (2.12)$$

Step 3- by comparing coupling matrix elements extracted from experimental result with the ideal coupling matrix elements, tuning elements can be adjusted accordingly.

Step 4- Steps 1 till 3 needs to be repeated till the extracted coupling elements from measured data fairly match the ideal coupling elements.

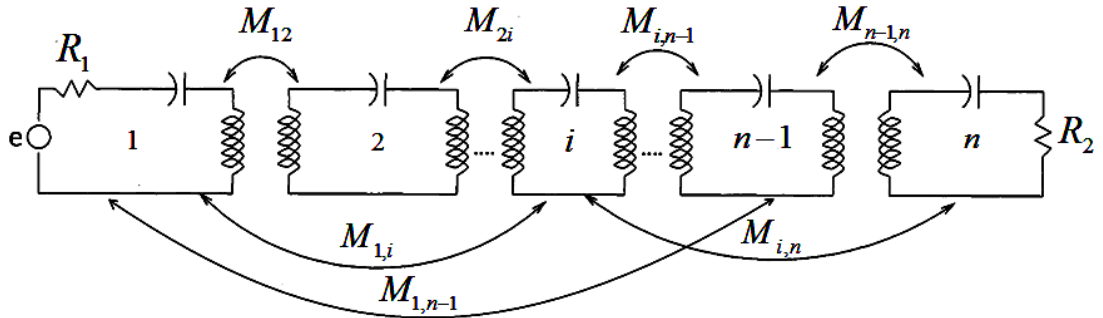


Figure 2.10 A generalized model for coupled resonator filters.

All the tuning elements can be tuned simultaneously and existence of error can be easily compensated in any stage of the tuning. However, in order to tune the filter by this technique, sufficiently close initial values for the variables to be optimized are required to prevent from running into local minimum or failing to converge to the proper solution. This case arises when filter is highly detuned. To overcome this problem, sequential parameter extraction approach [37] has been proposed. In this

method, first, all of the resonators need to be shorted. Then one resonator after the other with adjacent inter-resonator coupling can be tuned by following the steps described for circuit model parameter extraction method. Instead of Equation (2.12), Equation (2.13) needs to be considered as the error function. In each step the filter would be divided to smaller sub filters with different coupling matrixes which the parameters of shorted resonator are considered as a very large value in the relevant coupling matrix.

$$\varepsilon(i) = \sum_j |S_{11}^{meas}(\omega_j) - S_{11}^{mod}(\omega_j)_{(i)}|^2 \quad (2.13)$$

This approach eliminates the convergence problem associated with optimizing all filter parameters at once in highly detuned filters. However, in practice, tuning a resonator in each step of this solution would slightly detune the adjacent inter-resonator coupling which already was tuned. Accumulation of each step's error would lead to a considerable discrepancy between the measured response and the ideal results at the end of the tuning procedure. Therefore, this approach can be considered as a coarse tuning technique rather than a complete tuning technique. Moreover, several optimization steps are required to finalize the procedure via this method which would increase the duration of the tuning procedure. Another problem associated with this solution is that dealing with cross coupling in each step is very difficult.

2.4 Conclusion

2.4.1 Mechanical Design and Automation

In these automated systems actuator movements are based on absolute positioning. Therefore, there are no set limits for rotational movements and the system has to rely wholly on CAT data. As the only feedback for the robot, total reliance on CAT data causes major problems. This software needs too many measurement updates if the tuning elements are far from the ideal frequency. Each measurement can take up to one second. Consequently, waiting for CAT updates after each micro rotation of the actuator makes the progress very slow. In addition, there is the potential for error occurrence in software during tuning. In this case, the software produces ongoing false data and the automated setup has to follow these data blindly. In published

results by COM DEV Ltd. [38], for the purpose of fault recognition, the robot's software tunes the filter for a prescribed maximum tuning time and, if the tuning process lasts longer than that, it treats the CAT response as an error and restarts the software. Although this solution might by-pass the problem, by overturning the actuators during the blind tuning process, the tuning elements could fall into the filter housing or completely withdraw from the filter. Further, errors in software usually occur at the beginning of the procedure. This depends on extracted initial values for the variables to be optimized by the tuning technique. Thus, the system needs to wait for the prescribed maximum time to recognize the existence of an error, which would almost double the duration of the tuning process. Even though the outlined test setups are successful in tuning the filters, they are not designed to perform as standalone setups in an industrial environment.

In this field, there is no study in the literature to investigate the effect of play in assembled parts on the tuning procedure. This drawback can directly impact the tuning speed. COM DEV Ltd. reported that the existence of this lash causes confusion for its robot by any change in tuning direction. As this looseness is unavoidable in assembled joints, part of this study is concerned with finding a solution for it.

Chapter 3 of this thesis deals with the design and fabrication of an automated robot arm to interface with tuning elements, while eliminating the above-mentioned problems in the automated setups.

2.4.2 Tuning Techniques

In tuning techniques based on group delay, time domain and the phase format of the input reflection coefficient, alignment of each pole has to be separate and sequential. Successive alignment of tuning elements in these methods accumulates the possible error from each consecutive stage. This will lead to considerable mistuning at the final results. To match the final response with the desired one, fine tuning by skilled personnel would be necessary.

Further, in the case of deviation of one of the tuning elements at the end of the sequential tuning, the tuning procedure for all elements needs to be repeated, since there are no data in the structure of these methods to find the mistuned element.

Another problem associated with sequential tuning is that dealing with cross coupling in each sequence is difficult or, in some filters such as dual-mode elliptic filters and self-equalized filters, is almost impossible.

In addition to these problems, tuning based on the group delay or poles and zeros techniques requires shorting of the resonators or inter-resonator couplings. However, it is not always convenient to segregate each resonator or coupling element in a filter structure such as dielectric resonator filters. In addition, since complete short circuiting of the tuning elements in a real-case scenario is not possible, the data measured theoretically with these methods does not perfectly match those measured practically.

Among the tuning techniques, the circuit model parameter extraction technique based on the coupling model is a preferred approach in the industry, as it can extract the entire matrix in one run from the measured filter response, and it allows for adjusting the tuning elements simultaneously. This is important, since there is interaction between tuning elements: tuning one element can increase the deviation of another element that has already been tuned. With an entire coupling matrix, the detuned tuning element can be set back to the desired position at any stage of the tuning, without the need to repeat the entire tuning process. However, in order to employ this method, tuning elements should be sufficiently close to the ideal positions. As most manufactured filters are highly detuned in the first place, pre-tuning of these filters is required prior to fine tuning by the circuit model parameter extraction technique.

The sequential parameter extraction instruction introduced to solve this issue has the same problems that are mentioned with regard to successive tuning techniques. Further, due to optimization delay in the tuning of each sub filter, the recorded time for finalizing the tuning process by this approach [37] is high.

Each of the sequential tuning techniques described and the fuzzy logic approach can be used as a coarse tuning method prior to fine tuning by the circuit model parameter extraction approach. However considering the required tuning time and complexity of these techniques, especially in tuning the filters with incorporated cross coupling, it is irrational to use them as a rough tuning method.

In conclusion, part of this study deals with finding a coarse tuning method prior to implementing circuit model parameter extraction that can handle several cross couplings without any difficulty and in a short amount of time.

Chapter 3

MECHANICAL DESIGN AND AUTOMATION

3.1 Introduction

Removing the tuner from the filter causes a slight change in the positions of the tuning elements. This is because the existence of gaps between the external threads (male) of the lead screws and the internal thread (female) of the cavities frame makes these elements very loose. This problem is not addressed in the proposed automated tuning setups in the literature. To overcome this issue, tuning elements need to be locked prior to removing the tuner head from the tuned filter. For this purpose, a customized coaxial screw/nut driver is proposed in this chapter.

In manual tuning attempts, resonance frequency change of the most sensitive resonator to rotational change is measured as 10 kHz per 2 arcminutes. Since mechanical play between mechanical parts of the tuner and tuning elements is higher than this amount, backlash compensation during automatic tuning is essential. This drawback is not considered in the automated tuners in the literature [15, 24, 39]. In this study, the backlash effect on tuning by automated setups is investigated.

Relative positioning is utilized by the proposed automated setups in the literature. This requires total reliance on the tuning codes. In case of any error in the executed codes by the automated tuner, faulty data leads the tuning process to failure. In this chapter, designed mechanical stoppers allow the automated setup to employ the absolute positioning system as a second feedback source for robot codes. This enables the robot to detect the potential error at the early stage of its occurrence.

When employing a regular development plan based on manufacturing, design and integration in the production line of microwave filters, companies that consider the product and relevant processes concurrently can increase the uniformity of the product and relevant tools. Some products have been in production for many years and it is costly to modify them near the end of their life cycle. Therefore, an automated driver mechanism should be designed to interface with these older products, with the capability to take advantage of automation-friendly enhancements to new products as they are introduced. In this study, unlike the previously presented automated tuners, a flexible tuner is designed to match the wide range of filter types used in the company.

All the two- and three-dimensional drawings presented in this chapter are created with SolidWorks design software. A STL (STereoLithography) file format is extracted from each finalized solid model. Then, in order to check the functionality of the parts, a concept model is created. For this purpose, modelled plastic parts are used. These parts are created with a 3D plastic printer. Then, actual metal parts are fabricated by a Computerised Numerical Control machine (CNC) and an Electrical Discharge Machine (EDM).

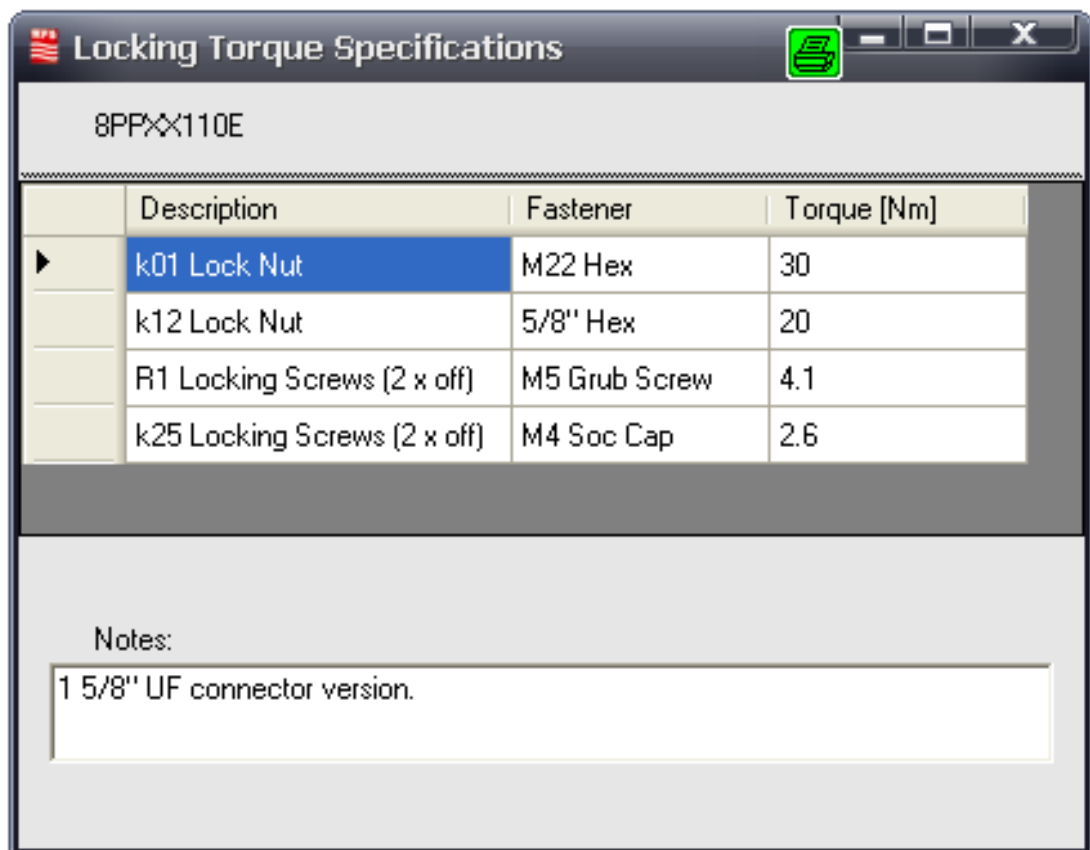
3.2 Design Dimensions

The first step of designing a robot tuner is to consider design limits. These wireless base station filters are designed with packed tuning elements on top of their frames. Therefore, there is not enough space on top of the filters to fit all the mechanical parts of the tuner and locker. This limit also affects the actuator type employed in the automated setup.

Several factors influence the design dimensions. The main factor is the size of the employed tuning actuators. This directly depends on the resolution and torque required to align the tuning element. Since the design needs to be comprehensive for all types of filters in the company, the highest resolution and torque that are enough to tune any filter type in the production line are considered in choosing the size of the actuators. In manual tuning attempts, resonance frequency change of the most sensitive resonator to rotational change is measured as 10 kHz per 2 arcminutes. The highest torque required during these attempts is obtained as 0.38 N-m. This much torque can be found in NEMA 17 with stack size 2; however, microstepping on a

stepper motor decreases its torque. As microstepping is required to obtain the above-mentioned resolution, it is necessary to employ a motor of a greater size and holding torque that, even with microstepping, could provide the required torque. Consequently, a NEMA 17 stepper motor with stack size 3 is utilized to deal with the tuning elements.

Another factor considered in the design dimensions is the size of the locking actuators. These actuators are utilized to lock the mounted lock nuts on the tuning elements. This is to prevent any unwanted change in filter performance during the delivery stage. The amount of required torque varies according to filter size and type of fastener. Figure 3.1 shows the required locking torque for different types of fastener in one filter: the required locking torque can reach 30 N-m. In order to provide this amount of torque, large stepper motors, together with gearboxes, are required.



Description	Fastener	Torque [Nm]
k01 Lock Nut	M22 Hex	30
k12 Lock Nut	5/8" Hex	20
R1 Locking Screws (2 x off)	M5 Grub Screw	4.1
k25 Locking Screws (2 x off)	M4 Soc Cap	2.6

Notes:
1 5/8" UF connector version.

Figure 3.1 Locking torque specifications for filter 8PPXX110E made by RFS Company.

Due to the requirements of the broadcast market, more complex multi-cavity filters in smaller sizes have to be manufactured. Some of these filters include several

cavities in their small framework. Figure 3.2 illustrates one of these filters. Its dimensions are 183 mm (length), 122 mm (width) and 86 mm (depth), and the filter includes six resonators, five couplings, two input/output couplings, and two cross couplings. For simultaneous adjustment of all the tuning elements of this filter, 14 actuators are required to be fitted on top of the filter lead. This filter is the smallest filter in the company, with a high density of tuning elements on its upper lead; therefore, a single adjuster is designed based on the dimensions of this filter to ensure that it can be easily fitted for other filters on the production line.



Figure 3.2 Smallest filter in the company (RFS) with high density of tuning elements.

Comparison of the smallest filter and the chosen stepper motor size is shown in Figure 3.3. In order to fit the multi-adjuster robot on top of this filter, the mechanical parts of the adjuster are miniaturized. Yet, the manufacturing limitations are taken into account during the design process.



Figure 3.3 Size comparison of 6-pole cross-coupled filter and NEMA 17 actuator.

3.3 Coaxial Screw/nut Driver

To make the automated arms capable of adjusting the tuning elements, lead screw drivers are designed on each tuner arm. At the beginning of the tuning process, while the tuner arms are moving downward, the lead screw drivers are rotated by tuner actuators. This will continue until the tuner heads engage with the hex slot of the tuning elements. To keep the tuner heads engaged with the hex slot of the lead screws during tuning, a sliding mechanism is designed for each tuner. This allows for vertical movement of the tuners along the lead screws' axis.

As well as a lead screw driver, the automated tuner requires a locking mechanism to finalize the tuning by locking the lock nut. This will ensure the consistency of the tuned filter until it reaches the customer. The locking torques prescribed by the company for these filters are high. For example, in the filter shown in Figure 3.3, 7 Nm torque is required to lock each resonator. Therefore, the locking mechanism designed needs to be able to tolerate this torque.

Even though separating the lead screw driver from the locking mechanism gives more design space to create mechanical parts, it will not allow simultaneous tuning and locking. This is crucial, as removing the tuner arms from the filter causes a slight change in the positions of the tuning elements. This is because of the existence of gaps between the external threads (male) of the lead screws and the internal thread (female) of the cavities frame, which make these elements very loose. Because of manufacturing tolerances, the existence of these gaps on the product is unavoidable. An amount of change can be ignored for one tuning element; however, in most practical cases, the accumulation of each lead screw's error can significantly distort the final result. To overcome this issue, the coaxial screw/nut driver shown in Figure 3.4 is constructed: this is capable of adjusting the screw and locking the nut simultaneously with independent servo-driven control. The mechanical parts of this design are described in Table 3.1, based on their part numbers as depicted in Figure 3.5. Details of each part are shown in Appendix A. This concept allows locking of the tuning element before removing the tuner head from it. Therefore, removing robot arms from the filter at the end of the tuning procedure will not change the final result.

This capability is also important to eliminate the unwanted effect of lockdown procedure on final filter response. Two factors are causing the change in the position of the lead screws during the locking stage. One is the aforementioned gap between the lead screws' thread (male) and the threads of the cavities frame (female). This gap would let the lead screw move vertically upward when the lock nut is tightened. In this project, by slightly tightening the locknut during the tuning stage (as it does not interrupt the rotation of the tuner head), this gap is lowered. The other fact is rotational change in the tuning element caused by tightening the nut. This change is lowered by engaging the tuner head with the tuning element while the locking procedure is taking place. By minimizing these changes, the effect of the lockdown procedure is calculated. This is only possible by setting the lead screw positions sufficiently close to the ideal position, which will allow the use of a sensitivity matrix [35], to create the relation between the movement of the tuning element and relative frequency deviation. Thus, after completing the tuning process, micro adjustment to the lead screws is made to counteract the associated detuning.

This system is also designed to be compatible with self-locking screws and other nutless screw designs; however, these design elements often have poor passive intermodulation (PIM) performance and are often incompatible with the stringent product specifications of the broadcast market. If design specifications allow the use of self-locking screws, the nut driver can be disabled for faster tuning time without changing the tuning procedure.

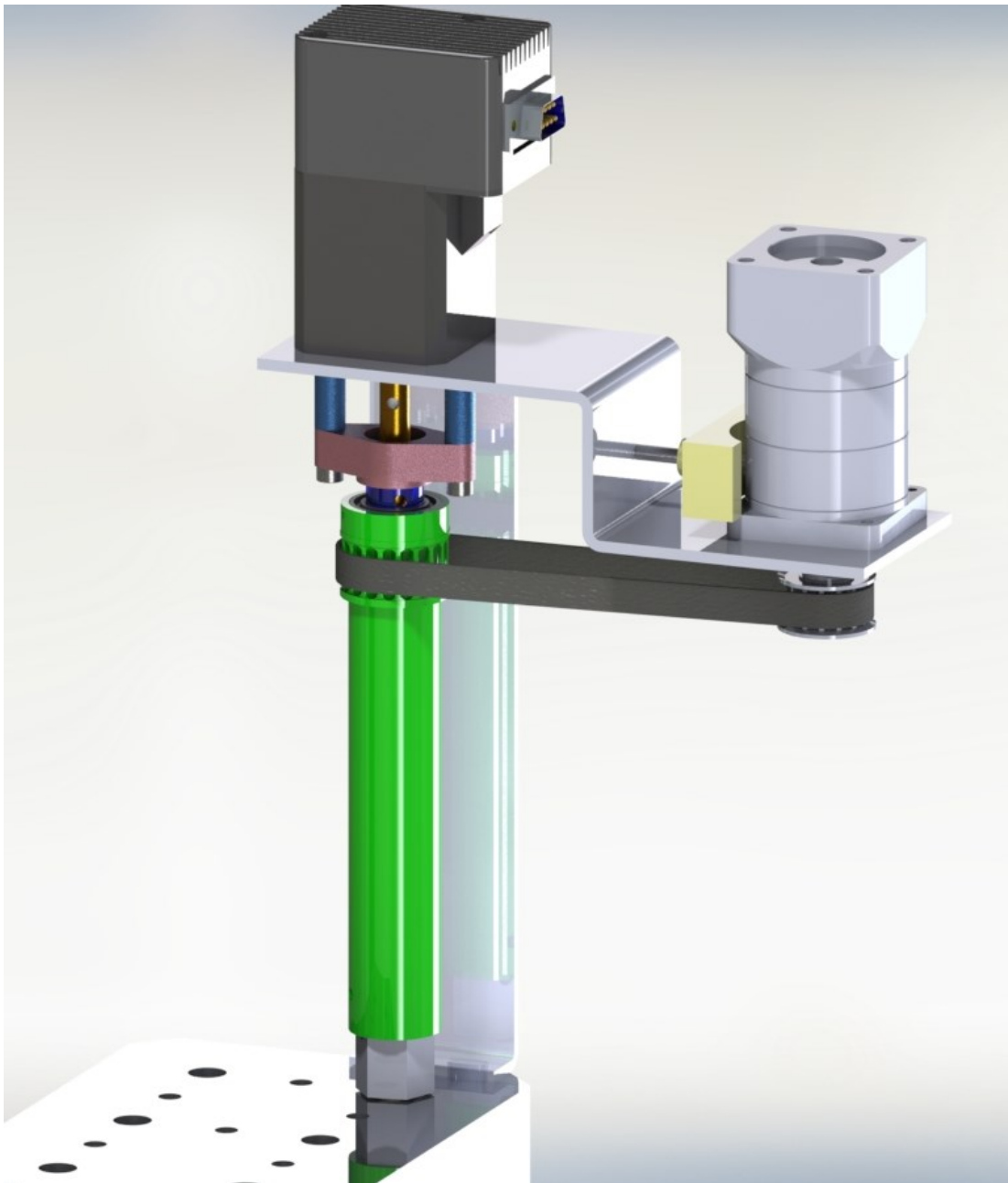


Figure 3.4 Coaxial screw/nut driver 3D drawing.

Table 3.1 Description of the parts in coaxial screw/nut driver.

1	Lock nut housing	2	Sliding housing attached to tuner head
3	Rotational and vertical stopper	4	Interface between the hex of the sliding part and motor shaft
5	Ball bearing between outer cylinder and stopper	6	Radial SKF ball bearing: 21mm
7	Locking tool with hex housing	8	Inner cylinder
9	Inner cylinder spring in order to increase the downward force of the tuner head	10	Outer cylinder spring in order to increase the downward force of the locker head
11	Interface between part 12 and inner cylinder in order to keep the inner cylinder and the stopper stable	12	Interface between part 11 and aluminium plate in order to stabilize the inner cylinder and stopper
13	Two radial SKF ball bearings: 16mm	14	Radial SKF ball bearing: 24mm
15	Pulley belt:100T, length 300mm	16	NEMA 17 stepper motor including rotary 1000 count encoder
17	Interface between sliding part 2 and hex shaft (separation of this part and part 2 is due to difficulties in wire-cutting or sparking with electrical discharge machining (EDM))	18	Hex shaft – transfers the rotational movement of motor shaft to sliding parts
19	Pulley belt tension adjustor	20	Pinion pulley: 18T
21	Gearbox housing with sliding mechanism to adjust the tension of the pulley belt	22	Tuner head
23	Gear box with 10:1 ratio		

3.4 Mechanical Stoppers

The concept of preserving stepper position is in essence flawed, if the start position of the system is unknown. This is true for any motor system in which the closed loop feedback is relative, rather than absolute. As a result, it is important to start counting the steps from a known position. This position is often known as ‘Home’. Since, during the tuning process, the motor can be started at any given position, mechanical stoppers are mounted at the lowest point of the inner cylinders. The positions with the lead screws fully inserted into the filter are set as a ‘Home step’ for the relative positions during tuning. These mechanical limits are crucial for the couplings, since they can fall inside the filter housing by overturning. To prevent the complete removal of tuning screws, another type of stopper is mounted at the highest point of the inner cylinder. These limits are also crucial for prepositioning; fault recognition methods are introduced in the following chapters. Lack of these stoppers in previous automated tuners, such as [38, 40], has led to the use of a relative positioning system. For this reason, these robots have to follow the Computer-Aided Tuning (CAT) software blindly.

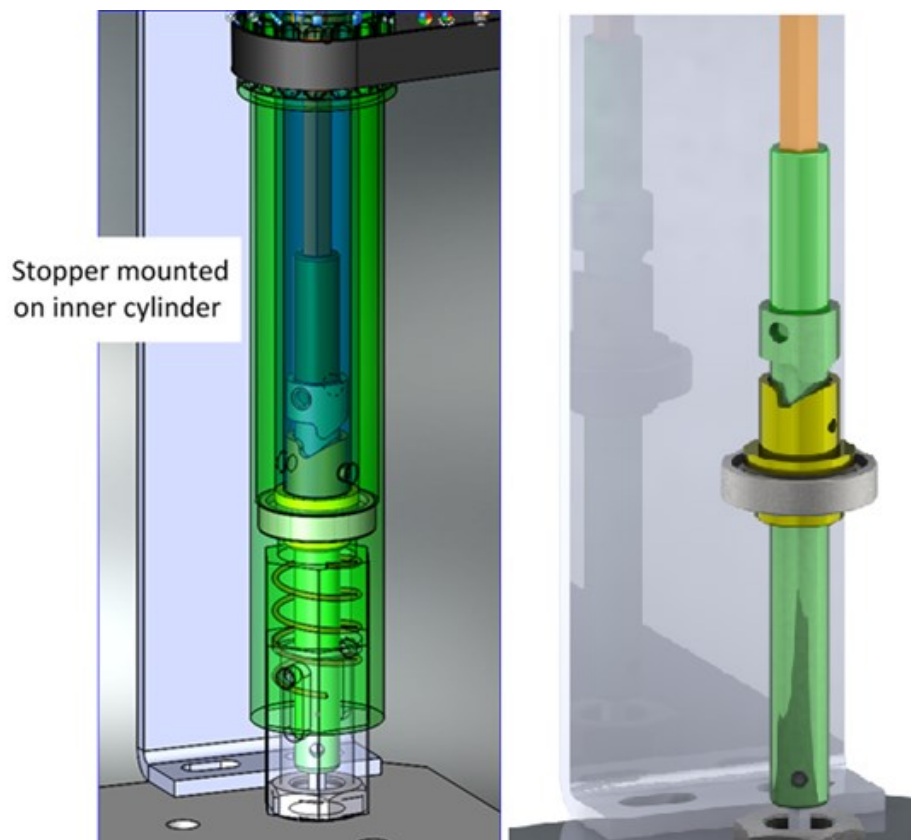


Figure 3.6 Mechanical stopper mounted on inner cylinder.

3.5 Belt and Pulley System

Because of the high density of the tuning actuators on top of the filter, the axes of the lockdown motors need to be set by distance from the relative tuners. Therefore a belt and pulley system is considered for power transmission of unaligned axes. The timing belt and pinion pulleys illustrated in Figure 3.7 are utilized to reduce slippage during transfer of the required high torque.

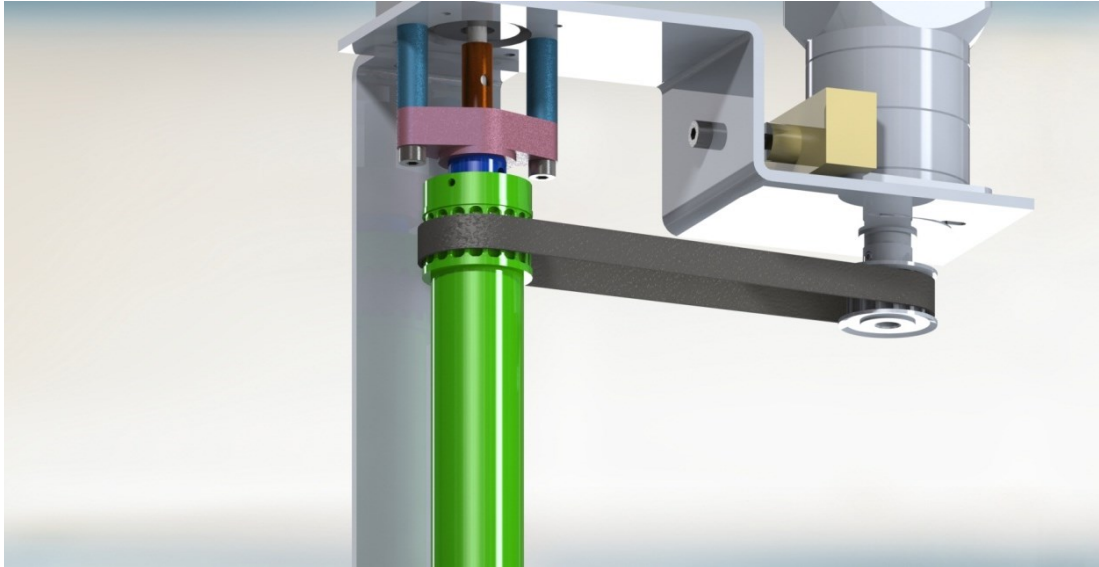


Figure 3.7 Timing belt and pinion pulleys.

A 2D drawing of the geometry of the timing belt and pinion pulleys is shown in Figure 3.8. The relation between the length of the toothed belt and the distance of two axes is calculated from Equations (3.1)-(3.5) [41]. Based on the obtained standard pinion pulley [42], the pitch size and number of teeth for external gears (male) of the outer cylinder are designed. To be able to utilize more of the shelf parts to reduce the cost and manufacturing time, the spur gear shape is considered. To optimize the timing belt tension, a sliding mechanism for the lockdown motor is constructed.

$$2C \sin\phi = L - \pi (R1 + R2) - (\pi - 2\phi)(R1 - R2) \quad (3.1)$$

$$\text{where ; } C \cos\phi = R1 - R2 \quad (3.2)$$

$$C = \left(\frac{1}{2}\right)p [(NB - N1) + k(N1 - N2)] \quad (3.3)$$

$$\text{where ; } k = \left(\frac{1}{\pi}\right) \left[\tan\left(\frac{\pi}{4} - \frac{\phi}{2}\right) + \phi \right] \quad (3.4)$$

$$\left(\frac{1}{\pi}\right) (\tan \phi - \phi) = \frac{(NB - N1)}{(N1 - N2)} \quad (3.5)$$

Where:

C = Center distance

L = Belt length (in) = $p \cdot NB$

P = Pitch of belt

NB = Number of teeth on belt = L/p

$N1$ = Number of teeth (grooves) on larger pulley

$N2$ = Number of teeth (grooves) on smaller pulley

ϕ = One half angle of wrap on smaller pulley

θ = Angle between straight portion of belt and line of centers

$R1$ = Pitch radius of larger pulley

$R2$ = Pitch radius of smaller pulley

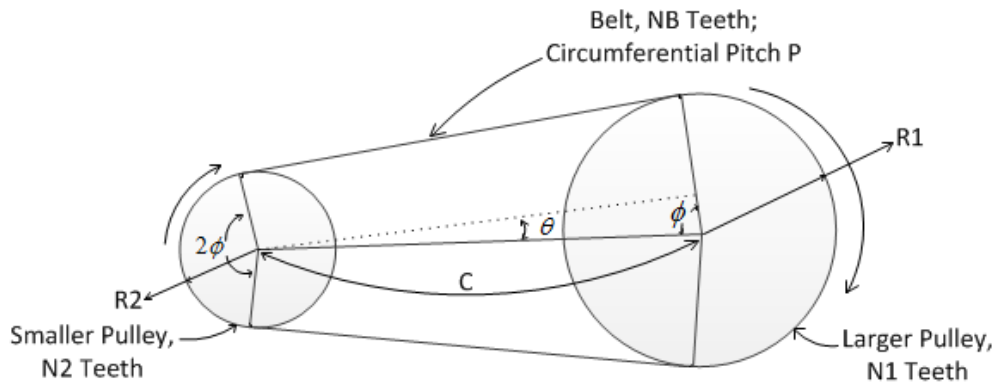


Figure 3.8 Timing belt and pinion pulley geometry.

3.6 Tuner Head

Although precise encoders are used to measure the movements of the adjusters, in some cases the robot is not able to align the tuner head to the center of the hex slot. This is due to inevitable fabrication tolerances and vertical play in the tuning element. Thus, the tuner heads are shaped with curved surfaces. This is shown in Figure 3.9(b). By downward movement of the robot's arm while the tuner actuator is ranging clockwise and anticlockwise along the pre-set angle, these surfaces easily slide into the screw's slot and the tuner engages with the lead screw. To make the tuner tips compatible with driving surfaces on the corresponding different screw

heads, the tuner head is designed to be replaceable. Driving surfaces compatible with the automated tuner created are shown in Figure 3.9(a).

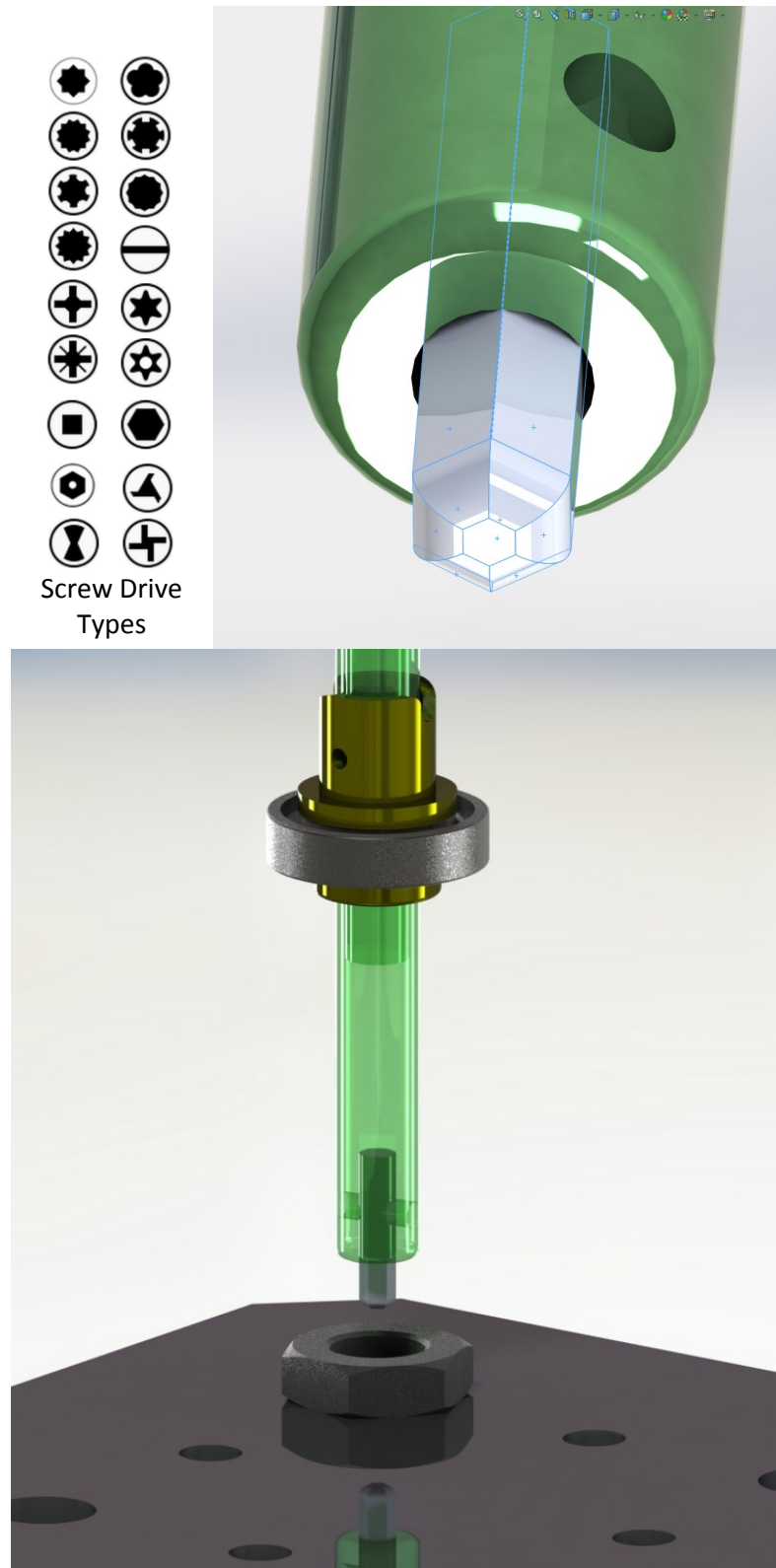


Figure 3.9 (a) Compatible driving surfaces with automated tuner (b) 3D drawing of replaceable tuner head (c) Transparent view of tuner head.

3.7 Multi-armed Robot

As it is shown in Figure 3.9, in a multi-arm robot, two level aluminium plates are utilized to hold the coaxial screw/nut drivers. This makes it possible to assemble sufficient tuners in a small area on top of the filter. Tuners for the resonators are assembled on the first plate and tuners for the couplings and a cross coupling are assembled on the second plate. If there is more than one cross coupling, a third plate is applied to hold the extra tuners. The 3D drawing shown in Figure 3.10 is created based on the 6-pole cross-coupled filter shown in Figure 3.3. However, these plates can be easily fabricated based on filter shape and position of the tuning elements. To move the plates on the Z axis, a linear Cartesian actuator is used (Figure. 3.12). Tuning by this type of tuner significantly reduces the required tuning time. Therefore, in production lines where the relative product is in high demand from customers, using this type of tuner is beneficial.

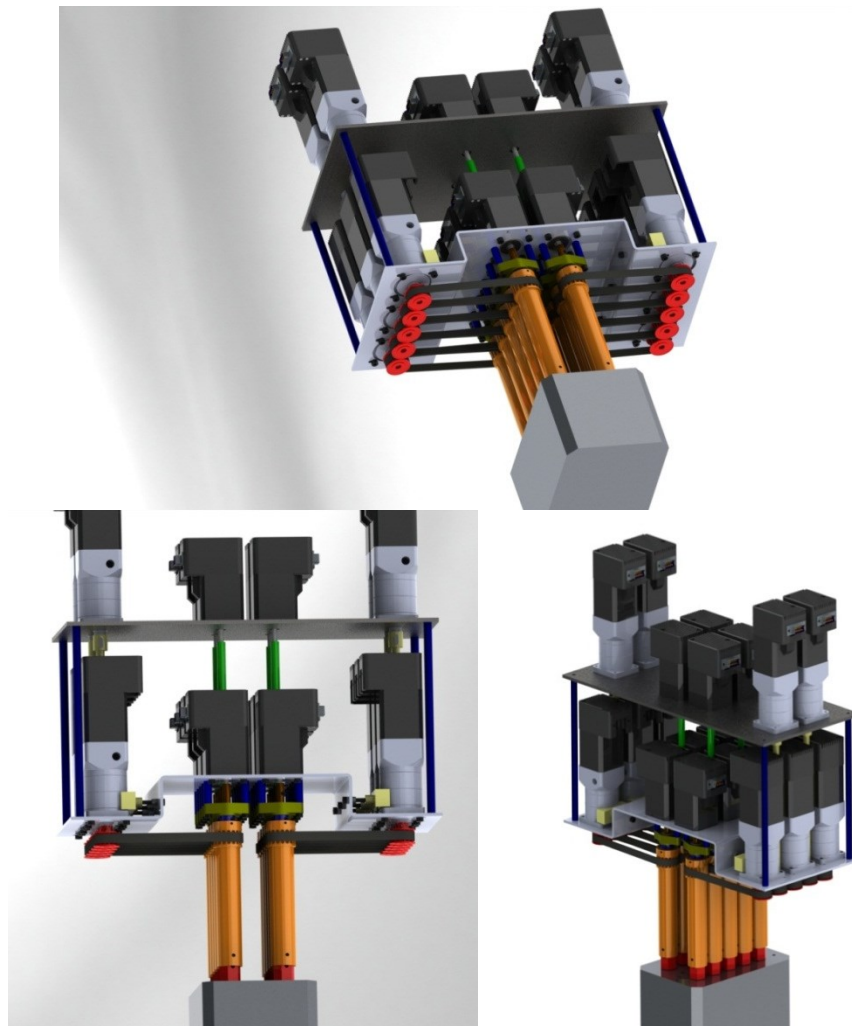


Figure 3.10 Multi-armed robot formation with two aluminium plates.

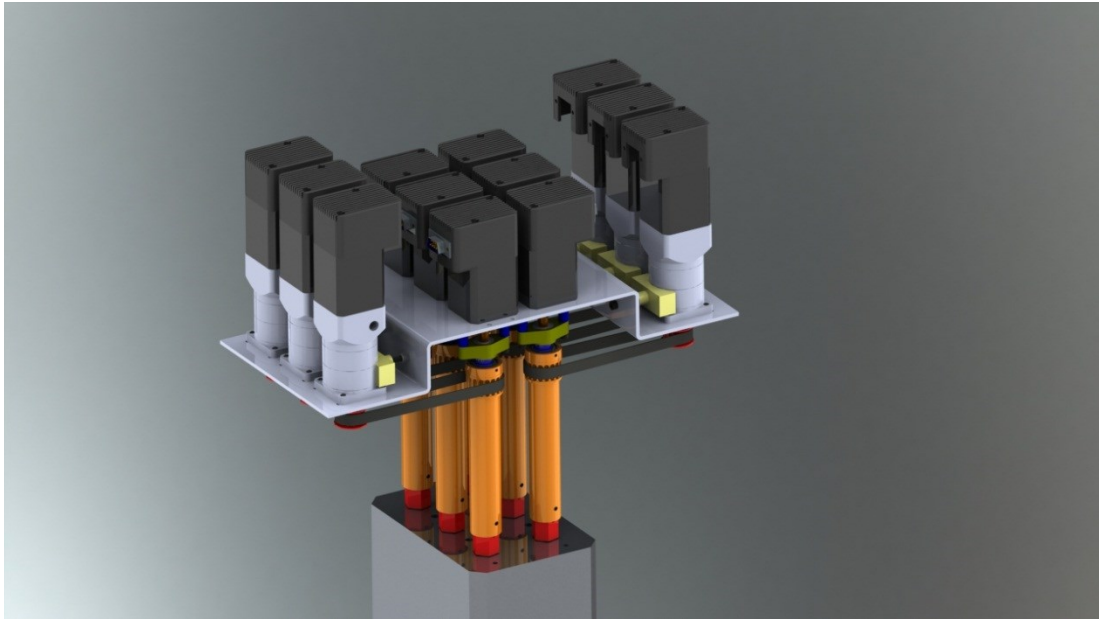


Figure 3.11 One level multi-armed tuner (diametric view).

3.8 Single-armed Cartesian Robot

Since some of the filter types in a production line are produced in lower numbers and a few filter types have a significantly large number of tuning elements, creating a multi-armed concept for each type of these filters is not a good solution, since it requires a large number of actuators and mechanical parts, which increases the cost. Therefore, in designing the process for the tuner's interface, compatibility with the single-arm mechanism is considered. There are two types of single-armed robots, the SCARA and Cartesian robots, which are considered in the design. Both the SCARA and Cartesian styles provide an excellent solution for moving the coaxial screw/nut driver tuners, but each robot style has its own unique features and advantages.

Cartesian robots work from an overhead grid. Also known as gantry robots, this type of robot is an accurate, quick solution for material handling applications. The work envelope (range of movement) of these robots is rectangular, and can be quite large. Therefore this type of robot is considered for larger filter types that occupy more space. The Cartesian robot's overhead grid can take up room overall, but does free up floor space.

Cartesian style robots are designed to be capable of X, Y, Z directional movements. The rigidity of these robots allows for more precision. As they are very easy to

programme and suitable for applications that require straight line insertion movements, Cartesian robots are dependable, strong movers in industry.

These robots are also an excellent choice for larger parts, as their payload is larger and fully supported. This is the reason that this style of robot is considered for moving the aluminium plates of a multi-armed robot. In addition, Cartesian robots provide better accessibility to larger parts, since they work from a grid.

Figure 3.12 illustrates the single-armed Cartesian robot with incorporated coaxial screw/nut driver.

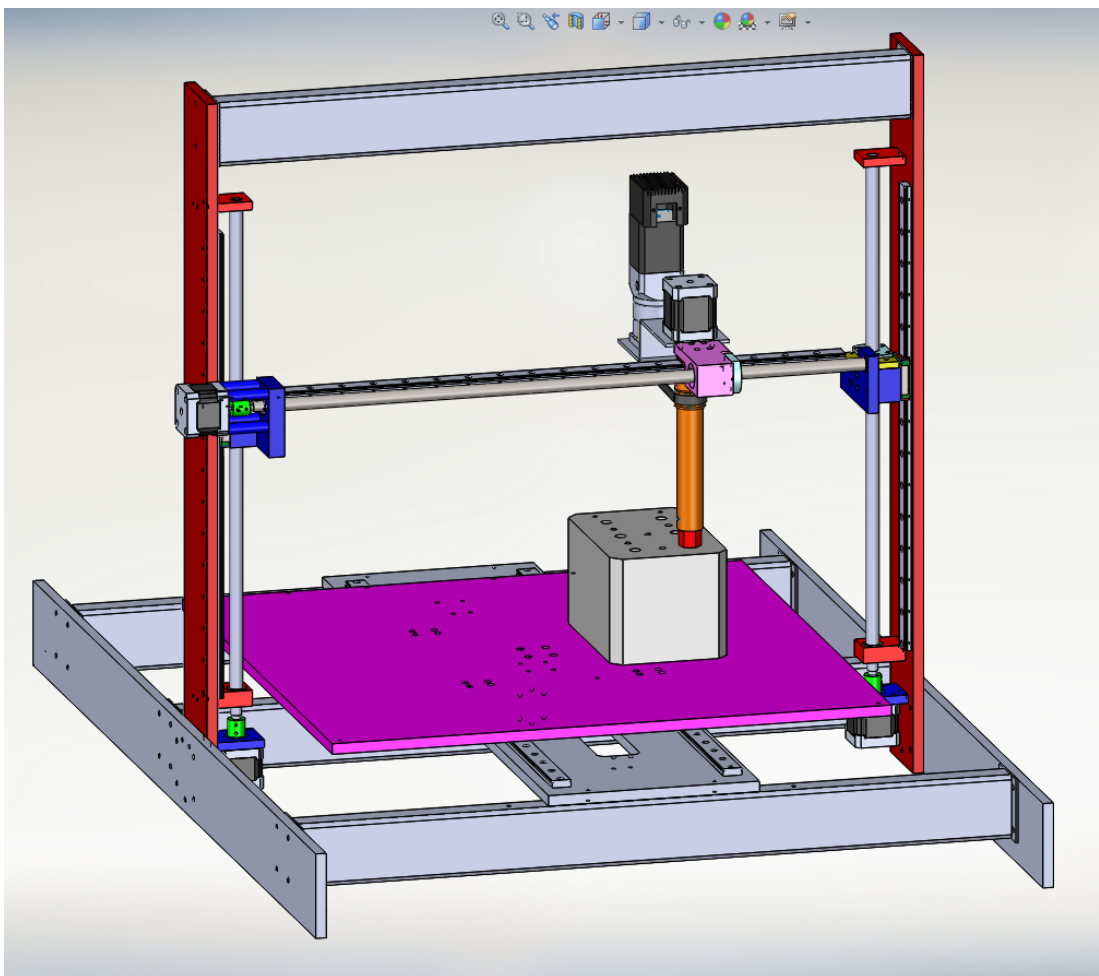


Figure 3.12 Single-armed Cartesian tuner.

3.9 Single-armed SCARA Robot

The SCARA (Selective Compliant Assembly (or Articulated) Robot Arm) robot provides a circular work envelope. This broad movement range allows for added flexibility. SCARA robots have a small footprint and can be built on a smaller scale.

Therefore for low order smaller filters in the company, the SCARA robot style is considered.

SCARA robots are more flexible than the Cartesian robot: 4-axis motions on this robot create the circular work envelope. While rigid, the SCARA robot can move with more flexibility in a horizontal plane. In filter tuning applications, a SCARA robot can perform with more speed than a Cartesian robot. This fact reduces the tuning time.

All the joints on a SCARA robot are located at the end of the arm with limited payload capacity. Thus, SCARA robots are best for smaller sized parts. Figure 3.13 and 3.14 depict the SCARA robot designed with assembled coaxial screw/nut driver.

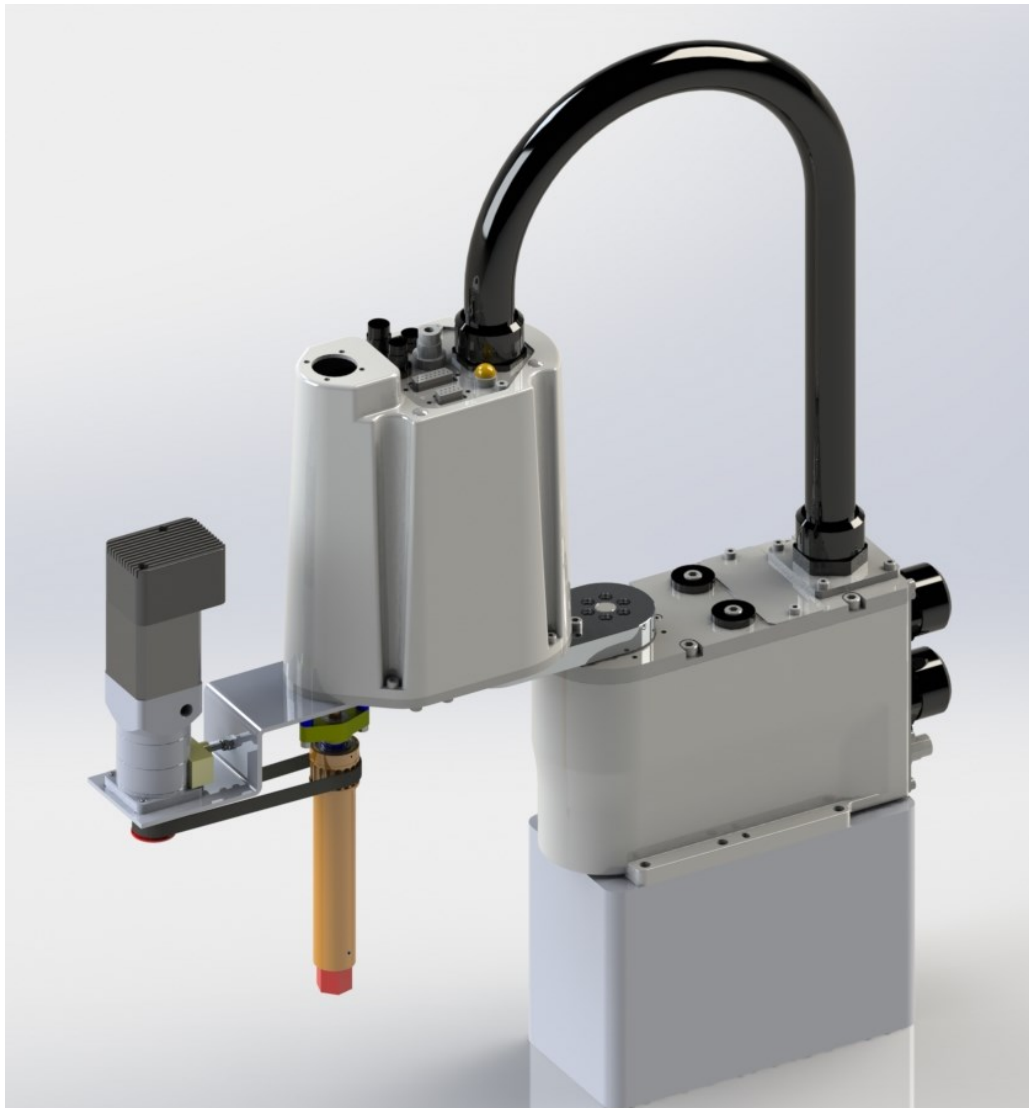


Figure 3.13 Single-armed SCARA tuner (isometric front view).

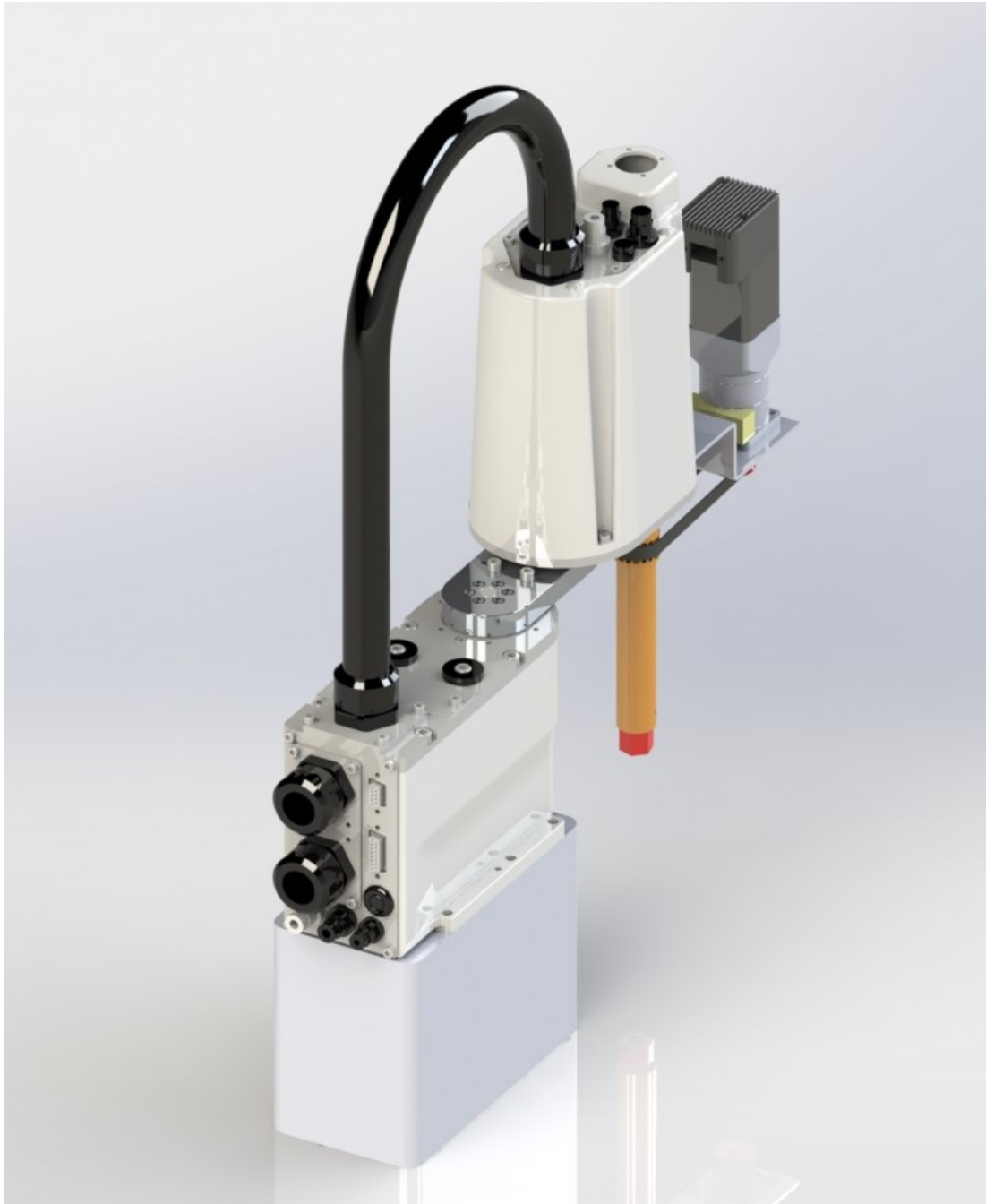


Figure 3.14 Single-armed SCARA tuner (trimetric back view).

3.10 Vertical Tuning Elements (Rod)

In a few older types of filter in production lines (Figure. 3.15), tuning rods are utilized that need vertical adjustment (pushing or pulling) rather than rotational adjustment. As vertical movement requires more precision in adjustment, it requires higher resolution in mapping for ideal positions.

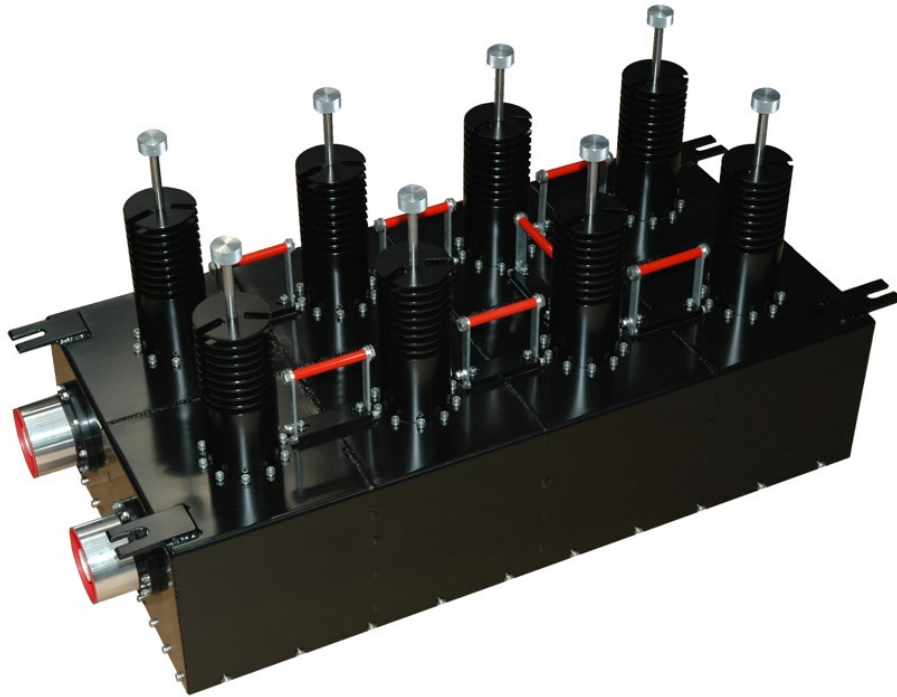


Figure 3.15 8-pole RF filter with tuning rods.

Considering the high torque and precision required for changing the position of these rods, using a linear actuator is not a good solution. Therefore, a stepper motor with ball screw crosses through the actuator is utilized (Figure 3.16). This will allow conversion of the rotational movement of the actuator to a linear movement. As a ball screw with a smaller pitch size is considered, very high resolution with sufficient torque is achieved.

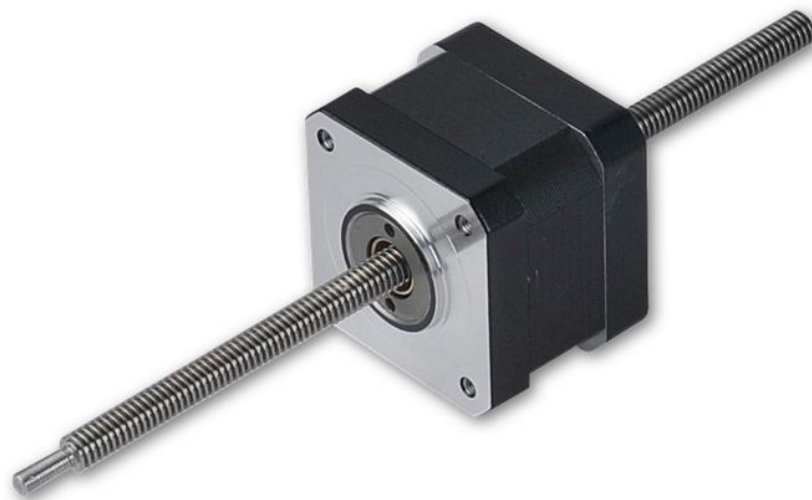


Figure 3.16 NEMA 17 linear stepper motor with ball screw.

Figure 3.17 shows the concept tuner for tuning rods. In this concept design, a NEMA 17 stepper motor with 0.9° resolution per step is used. The pitch size of the ball screw is 1 mm/rev; therefore, each rotational step of the stepper motor translates to 2.5 micrometre vertical movement in the tuning rod, which is sufficient to tune the filters.

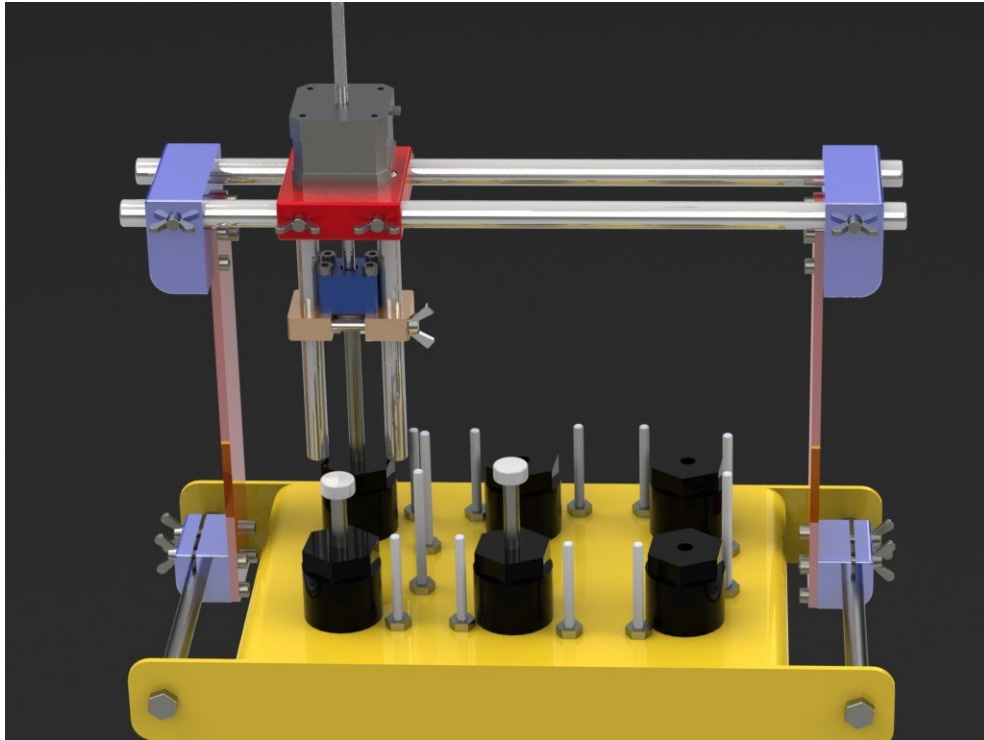


Figure 3.17 Mechanical concept for tuning rods.

3.11 Mechanical Backlash

Backlash, sometimes called ‘lash’ or ‘play’, is clearance or lost motion in a mechanism caused by gaps between the parts. It can be defined as ‘the maximum distance or angle through which any part of a mechanical system may be moved in one direction without applying appreciable force or motion to the next part in mechanical sequence’ [43]. Variables such as manufacturing errors and mounting tolerances often increase the amount of backlash. In the tuner arms created, the overall backlash amount is comprised of clearance between sliding parts, tuner head and slot of the lead screw, and the gearbox of the actuator.

The resonance frequency change of the most sensitive resonator to rotational change is measured as 10 kHz per 2 arcminutes. Since mechanical play between robot parts and tuning elements is higher than this amount, backlash compensation during the

tuning process is essential. However, elimination of this drawback is not investigated in the automated tuners introduced in the literature [38, 40]. The following example, which is based on practical measurements, proves the importance of this fact.

Let us say that the total mechanical looseness of an assembled tuner is about 4.6 degree: this is the average backlash measured in practice. Also, let us consider the required mapping resolution to be 6 arcminutes: this is the average required practical resolution for tuning. The average computational delay for CAT software is about half a second. Therefore, after each 6 arcminute rotation of the tuner, the robot needs to wait about half a second for CAT's data to be updated. Thus, it would take 23 seconds for the robot to map the entire backlash range. Wasting this amount of time at every change of direction in tuning would reduce the tuning speed. Therefore, the following programmatic backlash compensation solution is implemented in the robot's code.

In order to measure backlash, the lock nut is locked. Then, the lead screw is rotated clockwise and anti-clockwise along the lash. This looseness is measured by a 13-bit rotational encoder. Machine code automatically adds that much to the program's Distance-To-Go (DTG) when it changes directions. A simplified pseudocode for the backlash compensation solution follows.

```

IF ((n+α= desired position) && (same direction)) THEN
  {Command = n + α;
  }
ELSE IF ((n+α = desired position) && (change of direction)) THEN
  {Command = n + α + b;
  }
END IF

```

where b is the measured backlash, n is the current position of lead screw and α is distance-to-go. This is called programmatic backlash compensation. Note that there is no necessity for go-past-and-come-back motion, since the start point in tuning is always from the 'Home step', which is determined by the designed mechanical limit.

This solution is applicable with the introduced customized design, as it has mechanical stoppers attached to the sliding mechanism. Figure 3.18 illustrates the

tuning results for two tuning attempts on the first resonator of the 6-pole filter shown in Figure 3.3. The blue line shows the results for tuning the resonator with the anti-backlash code disabled. The red line shows the results for tuning the same resonator while the anti-backlash code is enabled. As is shown, settling time for the tuning attempt with the backlash compensation solution is considerably shorter than for the other attempt.

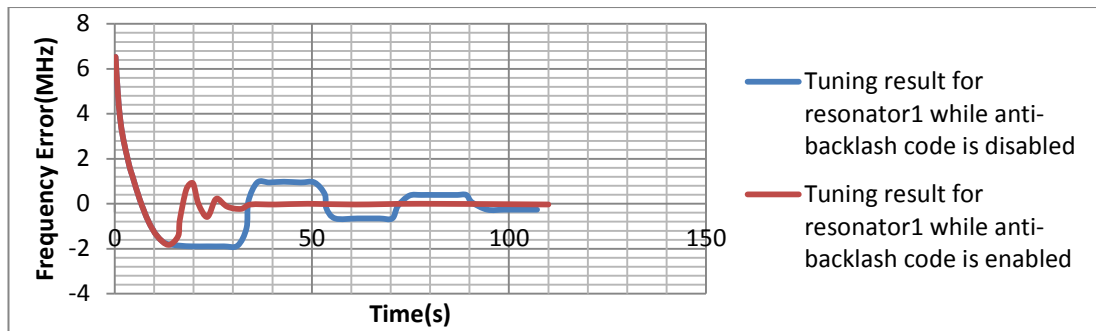


Figure 3.18 Tuning results related to first resonator of a 6-pole filter while anti-backlash code is disabled (blue line), anti-backlash code is enabled (red line).

3.12 Positioning System

Tuning in each center frequency for a filter needs a particular combination of lead screw positions. Consequently, we would expect to get the same filter responses in the same center frequencies from two filters of the same type, if the positions of the lead screws in both filters are matched with each other. This is not applicable in practice because of manufacturing tolerances. However, the filter responses of the above-mentioned filters would be close to each other. Considering this, prepositioning of tuning elements, based on positions extracted from the tuned filter (golden filter) can somewhat tune the filter.

Prepositioning is only achievable if the designed tuner is compatible with absolute positioning. Mounted mechanical stoppers on the designed tuner in this project allow absolute positioning for the robot. Thus, codes are written to save the lead screw positions of the tuned filters in the data base. In the next tuning attempts, the lead screw positions are pre-set based on these data; the filter response then approximates the ideal response. From that point, the tuning process is led by the CAT data. This approach is faster than using CAT data only to tune the filter. If the tuning elements are far from the ideal frequency, CAT software needs too many measurement updates to guide the robot through tuning: this is due to the optimization algorithm in

CAT. An excessive number of measurements in CAT causes computational delay in updating data. Therefore, tuning based on CAT data from the outset of the tuning would be a slow process. But, by pre-setting lead screw positions closer to the ideal points, the number of measurement updates in CAT is reduced. Consequently, computational delay in robot codes is decreased, which speeds up the tuning.

Figure 3.19 demonstrates three tuning attempts on the first resonator of the 6-pole filter shown in Figure 3.3. The blue line is related to the tuning attempt with CAT data only. The overshoot around zero frequency error is because of the delay in updating the CAT data. This delay causes oscillation of frequency error around zero and reduces the tuning speed. The green line shows the tuning results with CAT data while motor speed is accelerated. Although the first time to reach zero frequency error is lower, overshoot frequency is greater than for the first attempt. This increases the oscillation time of the lead screw around the ideal position. The red line shows the tuning results with prepositioning code and CAT data. In this attempt, the position of resonator is pre-set to the ideal position saved in the data base. As mentioned earlier, this reduces the computational delay in CAT. Then, automated setup is guided by CAT data to finalize the tuning. As shown, this approach reduces the settling time of the frequency error and as a result speeds up the tuning.

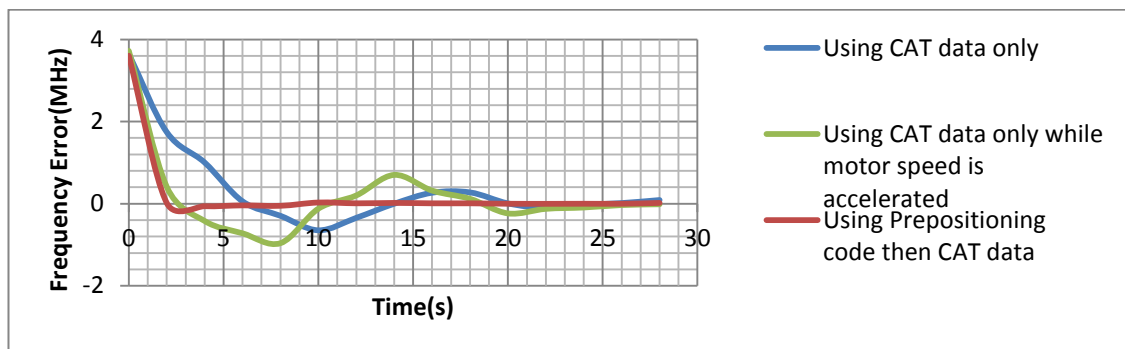


Figure 3.19 Tuning results related to first resonator of a 6-pole filter: using CAT data only (blue line), using CAT data only while motor speed is accelerated (green line), using Prepositioning code plus CAT data (red line).

In case of potential error in tuning, CAT software provides on-going false data. In automated tuners which rely wholly on CAT data [44, 45], machine code would blindly follow these false instructions. This would lead the tuning procedure to failure. In the designed tuner, the measured position of the tuning elements is used as a second feedback source for the machine code. This allows the robot codes to

compare the data of two feedback sources against each other or against the data of previous tunings. Therefore, potential error in any of the feedback sources can be recognized in the early stages of tuning. This capacity for fault recognition by the robot eliminates the need for supervision by an operator. In addition, tuning time is reduced by restarting the procedure in the early stage of the faulty tuning process.

3.13 Conclusion

In this chapter, a customized coaxial screw/nut driver for automated filter tuners has been introduced. Unlike the automated setups in the literature, the simultaneous locking and tuning capability of this tuner allows the robot to lock the tuning elements before removing the robot arms from the filter. This functionality eliminates the unwanted change that occurs by removing the robot arms at the end of the tuning process, and consequently increases the precision of the final product.

The measurements taken show that mechanical play between the mechanical parts of the automated tuner and the tuning elements was much higher than the mapping resolution required for tuning the lead screws. It has been reported that, in previous automated tuners, their setup is more effective for tuning brand-new filters due to lower backlash amount: in some attempts to tune old filters with high backlash, the setup failed the tuning task [38]. It was also reported by COM DEV [38] that discontinuity in tuning direction completely confuses the tuning algorithm of their setup, due to mechanical backlash between mechanical parts. In this chapter, it has been shown that, by using a customized coaxial screw/nut driver, it is feasible to measure the amount of backlash and implement it for the distance-to-go of the robot. The effectiveness of this backlash compensation solution was proven by the results shown. Therefore, the automated tuning setup created in this study is not only able to tune brand-new filters but is also effective for old filters with a higher amount of backlash.

Unlike the few other tuning robots in the literature, an absolute positioning system has been utilized by a created robot instead of a relative positioning system. In this regard, mechanical stoppers were mounted on the inner cylinder of the coaxial screw/nut driver to limit the vertical and rotational movements of the lead screws and set a known ‘Home step’ for the robot. The positions of the lead screws were measured from the pitch size of the tuning elements and the tuner’s angle of rotation,

based on the 'Home step' created. Measured data were used as a second feedback source for the automated setup, together with CAT data. This allows the robot to detect the potential error at an early stage of its occurrence, and consequently prevents wastage of time in the tuning process. In the literature, only the proposed automated setup by COM DEV has implemented a solution to overcome this issue. In this setup, for the purpose of fault recognition, the robot's software tunes the filter for the prescribed maximum tuning time and, if the tuning process lasts longer than that, it treats the CAT response as an error and restarts the software. However, although this solution may by-pass the said problem, overturning the actuators during the blind tuning process could cause the tuning elements to fall into the filter housing or completely withdraw from the filter. Furthermore, errors in software usually occur at the beginning of the procedure. This depends on adequacy of extracted initial values for the variables to be optimized by fine tuning program. Thus, the system needs to wait for the prescribed maximum time to recognize the existence of an error: this would almost double the duration of the tuning process. The outlined test setups in the literature might succeed in tuning the filters at most attempts, but they are not designed to perform as standalone setups in an industrial environment.

Chapter 4

COARSE AND FINE TUNING

4.1 Introduction

In established tuning processes of multi-cavity filters, the coarse and fine tuning stages are conducted manually. Computer-Aided Tuning (CAT) software based on fine tuning techniques guides the human operator through the fine tuning process, while coarse tuning is still performed based on the operator's experience. Consequently, a highly skilled technician is required in order to complete the tuning task. To automate the tuning process, reliable coarse and fine tuning techniques are required.

Five tuning techniques that have been proposed in the literature are investigated. From literature, it can be concluded that the introduced circuit model parameter extraction technique was the most effective fine tuning technique for applications in the automated tuning setup. This technique extracts all of the required parameters for tuning the filter at once. This enables the automated setup to adjust the tuning elements simultaneously rather than sequentially. Therefore, each tuning element can be aligned without needing to wait for other tuning elements to be tuned. Additionally, potential deviation in final result can be moderated by readjusting the relative tuning elements without needing to restart the tuning process from the beginning. The aforementioned advantages make the circuit model parameter extraction tuning instruction a preferable fine tuning technique for filter companies. However, sufficiently close initial values for the variables to be optimised by this technique are required in order to utilise it. This is needed in order to prevent the

optimisation algorithm from running into a local minimum or failing to converge to the proper solution. Therefore, this technique is not applicable for highly detuned filters. As most manufactured filters are in a highly detuned state, finding a compatible coarse tuning instruction with an automated setup is essential. This instruction needs to be implemented prior to performing the circuit model parameter extraction technique.

The proposed sequential parameter extraction approach in [37] eliminates the convergence problem associated with optimising all filter parameters at once in highly detuned filters. However, it converts the circuit model parameter extraction tuning technique into a sequential tuning process. As mentioned in the literature, sequential tuning methods are not suitable for automated setups.

CAT software based on group delay [17, 18], time domain [19–21], fuzzy logic [16], [22–24] or poles and zeros [29] techniques in the literature can be used to perform the coarse tuning on the filter. However, since these techniques are proposed for overall tuning, they are very complex and they require much more time to be completed. In addition, not all of these techniques are applicable for automated setups. Only the fuzzy logic method is utilisable by automated tuners. However, this technique has a complicated algorithm and very slow progress in tuning. Dealing with cross-coupled filters in the time domain and fuzzy logic techniques is impractical.

The next section of this chapter shows the experimental setup used for implementing coarse and fine tuning. The software configuration will also be explained.

In a subsequent section of this chapter, a new approach for coarse tuning of multi-cavity filters is introduced. This approach can deal with cross-coupled filters and does not need any complicated calculation by CAT. Coarse tuning via this technique significantly reduces the overall tuning time. This technique is applicable for both manual and automatic tuning. The validity of this approach is proven by the obtained experimental results which show significant time savings. These results are acquired by implementing the proposed technique in an automated setup that was introduced in chapter 3.

As it is practical to implement the proposed course tuning instruction by automated setup and roughly tune the filter, it is then feasible to implement the circuit model parameter extraction instruction by robot. In the final section of this chapter, the circuit model parameter extraction technique is practically implemented by automated setup to finalize the fine tuning. Achieved results prove the effectiveness of the automated setup created.

4.2 Experimental Setup

Experimental work used a 6-pole cross coupled RF multi-cavity filter as shown in Figure 4.1. This filter has dimensions of $l = 166$ mm, $w = 111$ mm and includes 2 input/output couplings, 6 resonators, 5 couplings and a single cross coupling to meet all the mask requirements. This product is designed for global filtering applications associated with broadcasting digital television (DTV) channels. Test setup shown in Figure 4.2 is used to achieve the experimental results. In the presented results, the filter is tuned to channel 40 with a center frequency of 613.5 MHz.

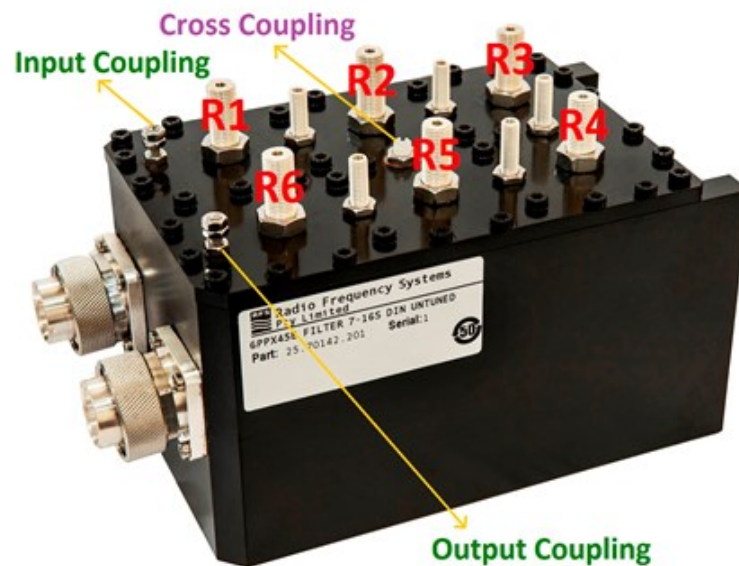


Figure 4.1 6-pole cross-coupled filter.

A block diagram of the automated setup created for Radio Frequency Systems (RFS) Pty Ltd is depicted in Figure 4.3. This automated setup is called the Robotic Computer-Aided Tuning (RoboCAT) setup. The automated tuner developed in chapter 3 is utilized in RoboCAT to interface with the tuning elements. CAT software in this setup runs by a computer which is connected to a Vector Network

Analyser (VNA) through a LAN cable. Programs running in the CAT are able to adjust the VNA and receive the measured data from it by using VNA drivers.

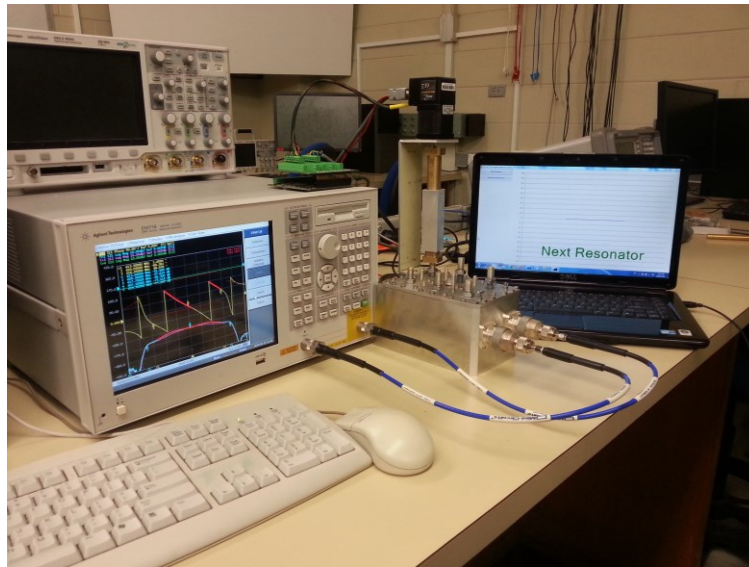


Figure 4.2 Experimental setup.

Since CAT software is licensed, few employees have access to its code to update or troubleshoot. Therefore written code for automated setup is separated from the CAT code. This allows the other staff to troubleshoot possible errors in automated setup without need to access the licensed CAT software. Communication between two programs is done through socket programming. Summary of this code is given in Appendix B. The programs can be run on two different computers or the same computer by changing the communication IP between them. Both CAT and control codes are written in .NET Platform.

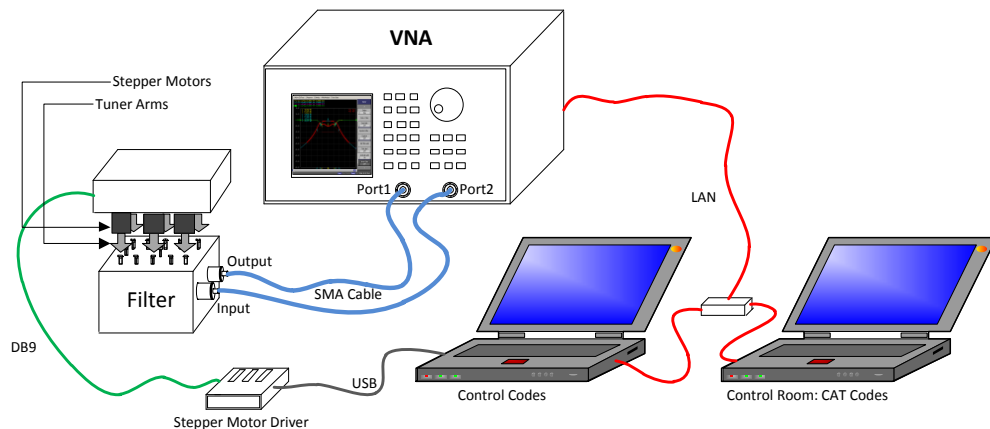


Figure 4.3 Block diagram of experimental setup (RoboCAT).

4.3 Coarse Tuning

Phase format of the input reflection coefficient S_{11} has sufficient information to bring all the resonators to resonance. Therefore, measured filter response in this format is considered to guide the robot through the coarse tuning process. Filter response is measured by the VNA. Only the last resonator is tuned based on log magnitude format of filter response as it needs to be tuned by minimising the filter return loss. Tuning by this technique does not need any complicated calculation. Coarse tuning instructions are independent from calculated data by the fine tuning program. This is important as the high computational delay in fine tuning codes slows down the coarse tuning process. The proposed coarse tuning technique in this section is applicable in both manual and automatic tuning procedures. In an automatic tuning process, VNA drivers are used to adjust the VNA setups and receive the measured data from it. In manual tuning, this can be done by adjusting the VNA setup manually and by observing values at specific frequencies. In the following subsections, tuning instructions for resonators, couplings and cross couplings are described separately.

4.3.1 Coarse Tuning of Resonators

Proposed coarse tuning process for resonators can be summarized in the following steps:

Step 1- To characterize the systematic errors, first the VNA needs to be calibrated by Electronic Calibration (ECal) Module or traditional mechanical calibration kits. Second, the center frequency of the VNA needs to be set to the desired frequency at which the filter needs to be tuned. This depends on the broadcasting channel. Third, the span (start/stop frequencies) of the VNA needs to be set as all the resonating frequencies of the resonators can be seen on the display while it is not too wide to reduce the tuning resolution. The span is set based on specified filter bandwidth. These parameters are fixed automatically by programs in the automated tuner. Manual tuning requires these parameters to be set by the user. In the following results in this subsection, the span is set to 12 MHz at a center frequency of 613.5 MHz for tuning the first 5 resonators. The span was then increased to 30 MHz to tune the last resonator based on filter return loss.

Step 2- Calibration only accounts for the systematic errors from the VNA to the device under the test (DUT). For the distance between input coupling and the first resonators inside the filter, the normalization setting in VNA is used by software. This step can be done manually in the manual tuning process.

Step 3- Trace and markers on the VNA are set by software in the frequencies that phase and magnitude needs to be measured. This is done by using VNA drivers. In manual tuning, this step can be done by using VNA settings. Initially, trace 1 is set to measure S_{11} in phase format for tuning the first five resonators. Later, format of this trace is changed to log magnitude to tune the last resonator based on filter return loss. Trace 2 is set to measure S_{21} in log magnitude format to tune the last resonator in coarse tuning process along with trace 1 in log magnitude format. Trace 3 and 4 are set to measure S_{12} and S_{22} respectively. These traces are set to log magnitude format and are used in the fine tuning process along with trace 1 and 2.

Step 4- All the resonators need to be highly detuned prior to the coarse tuning. This condition is observed on the polar display of the VNA when the input reflection coefficient S_{11} in phase format appears as a horizontal line. However, unlike the aforementioned tuning techniques in literature, complete short circuiting the resonators is not required. Thus, these elements do not need to be fully removed from filters framework or fully inserted deep into the cavities. In this approach, if any frequency in the range of the span has phase of more than $0^\circ \pm \epsilon$, deviation of the relative resonator is increased until measured reflection coefficient S_{11} gets close to a horizontal line. In practical tuning by RoboCAT, there is no need for checking the measured response of the filter by VNA for assurance of highly detuned lead screws. Designed maximum and minimum stoppers in robot arms (Chapter 3- Mechanical stoppers) are fixed in positions which lead screws can be withdrawn to maximum point without falling out of the filter frame or inserted deep into the cavity without dropping inside the filter framework. Turning resonators lead screw to these maximum or minimum points assures the required high deviation states to start the coarse tuning. This capability of designed tuner eliminates the wasted time for searching for these positions. Measured S_{11} in phase format is shown in Figure 4.4 while all the resonators are highly detuned.

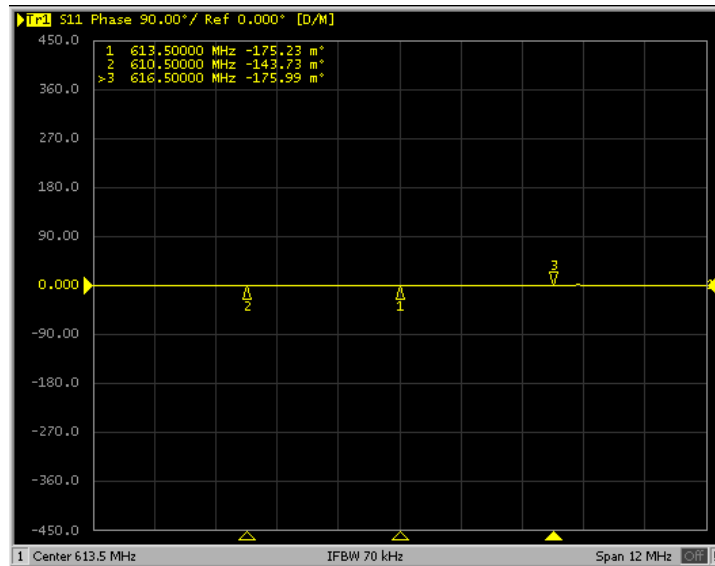


Figure 4.4 Phase format of the input reflection coefficient S_{11} while all the resonators are highly detuned.

Step 5- Considering the phase format of input reflection coefficients S_{11} , resonator 1 is brought to resonance by rotating the relative tuning element until the phase passes through the $\pm 180^\circ$ crossing at the center frequency. This is shown by marker one on the yellow trace in Figure 4.5. The value of this marker is used by the automated setup to tune the first five resonators. As it is depicted in Figure 4.5, there is only one frequency with phase of $\pm 180^\circ$ on the display of the VNA which is related to resonator 1. At this stage resonators 2, 3, 4, 5 and 6 are in a highly detuned state.

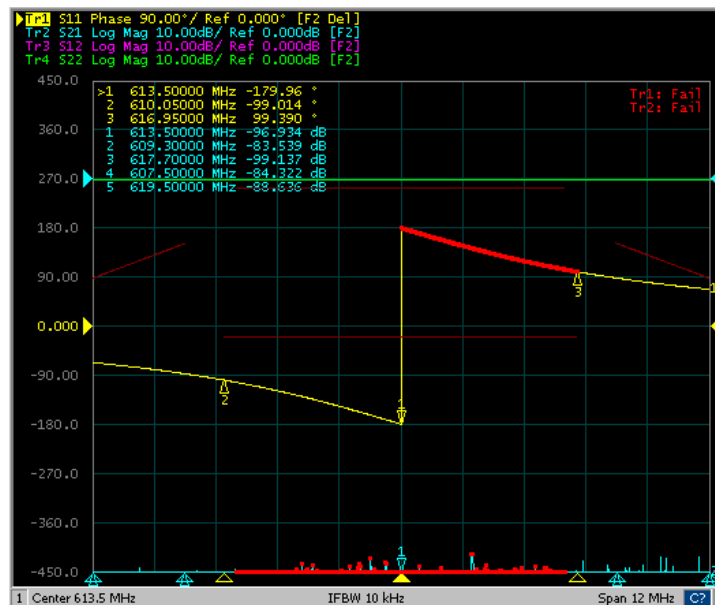


Figure 4.5 Measured filter response by VNA (Resonator1 is brought to resonance while rest of the resonators are highly detuned).

Step 6- Considering the phase format of S_{11} , resonator 2 is brought to resonance by rotating the relative tuning element until the phase passes through the 0° crossing at the center frequency. This is shown by marker one on the yellow trace in Figure 4.6. Resonators 3, 4, 5, 6 are still highly detuned.

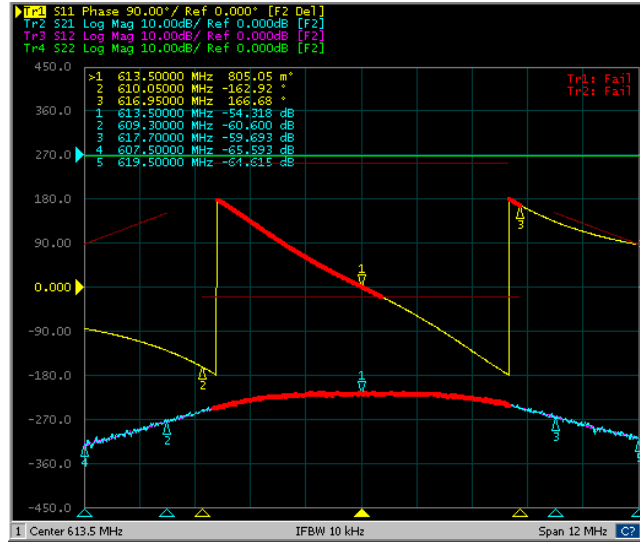


Figure 4.6 Measured filter response by VNA (Resonators 1 and 2 are brought to resonance while resonators 3, 4, 5, 6 are highly detuned).

Step 7- Considering the phase format of S_{11} , resonator 3 is brought to resonance by rotating the relative tuning element until the phase passes through the $\pm 180^\circ$ crossing at the center frequency. This is shown by marker one on the yellow trace in Figure 4.7. At this stage, resonators 4, 5, 6 are in highly detuned state.

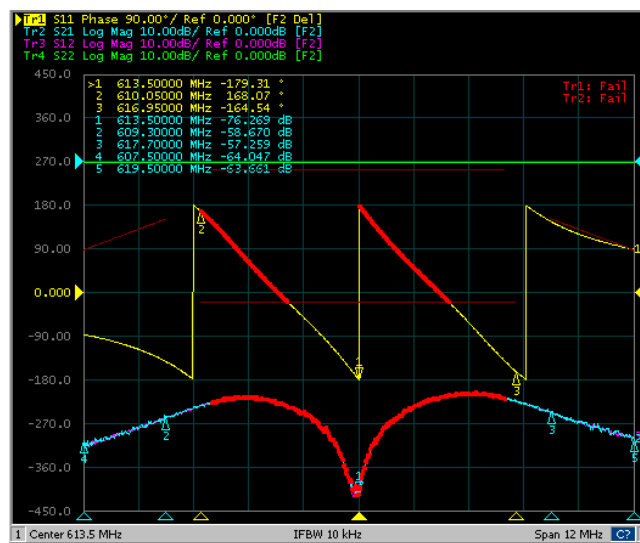


Figure 4.7 Measured filter response by VNA (Resonators 1, 2, 3 are brought to resonance while resonators 4, 5, 6 are highly detuned).

Step 8- As it is depicted in Figure 4.8 resonator 4 is brought to resonance by rotating the relative tuning element until phase passes through the 0° crossing at the center frequency. This is shown by marker one on the yellow trace in Figure 4.7. This trace is set to measure the phase format of S_{11} . Resonators 5 and 6 are highly detuned.

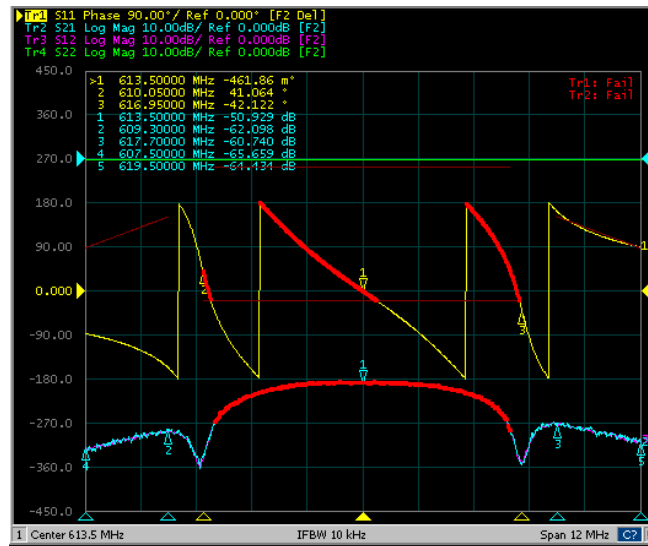


Figure 4.8 Measured filter response by VNA (Resonators 1, 2, 3, 4 are brought to resonance while resonators 5, 6 are highly detuned).

Step 9- Considering phase format of S_{11} , resonator 5 is brought to resonance by rotating the relative tuning element until the phase passes through the $\pm 180^\circ$ crossing at the center frequency. This is illustrated by marker one on the yellow trace in Figure 4.9. Resonator 6 is still highly detuned.

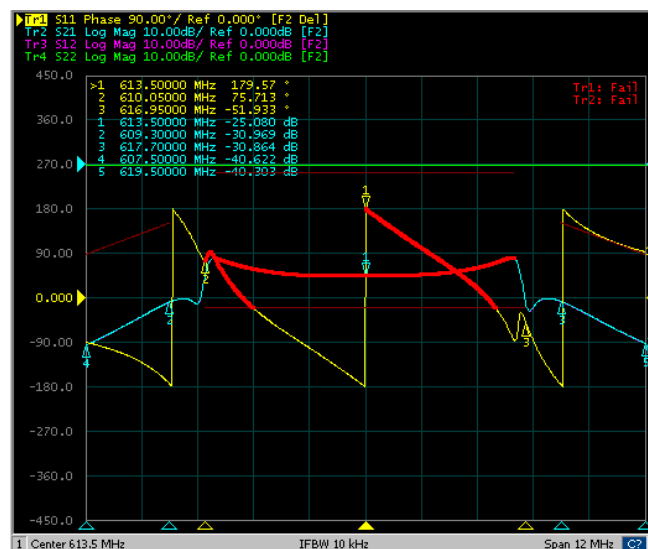


Figure 4.9 Measured filter response by VNA (Resonators 1, 2, 3, 4, 5 are brought to resonance while resonator 6 is highly detuned).

Step 10- In this step, software sets the trace 1 to measure the log magnitude format instead of phase format of the S_{11} . The last resonator is then tuned by minimising the return loss. For this purpose, log magnitude of the input reflection coefficient S_{11} needs to be observed. Span of the VNA is also increased to 30 MHz to fit the filter response on the display. At this stage, return loss cannot be minimised to the ideal return loss. This is because of deviation in cross coupling and couplings. Figure 4.10 demonstrates the final result achieved by the proposed tuning method while cross coupling and couplings are mistuned. Even though the filter return loss is not ideally matched with the tuned filter, yet filter response is closer than before to the ideal filter response. These results demonstrate that the phase of S_{11} passes through $\pm 180^\circ$ and 0° crossing at center frequency ω_0 as the successive resonators are tuned.

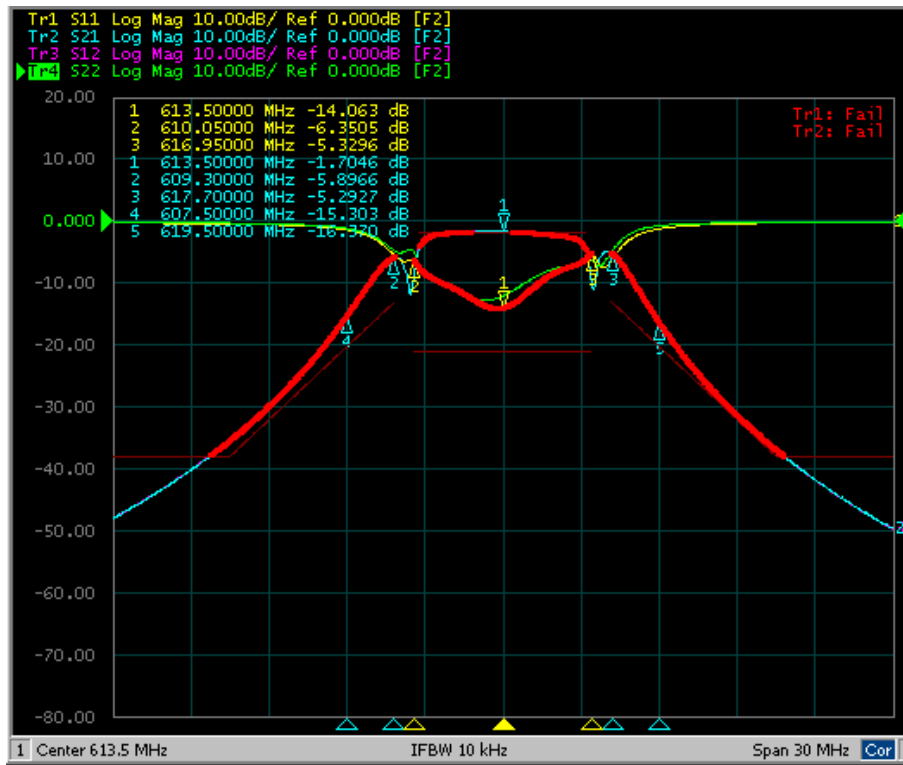


Figure 4.10 Filter response after one run coarse tuning of resonators while cross coupling and couplings are mistuned.

4.3.2 Coarse Tuning of Inter-resonator Couplings

Inter-resonator couplings are not usually strong enough to affect the coarse tuning procedure, however high deviation of these elements can mislead the tuning process. In order to prevent this problem following procedure is implemented.

Step1- After implementing step 4 of coarse tuning for resonators, coupling between resonators 1 and 2 is withdrawn. This shifts the $\pm 180^\circ$ phase from center frequency (613.5 MHz) to the right. The frequency which corresponds to phase of $\pm 180^\circ$ is recorded as ω_1 . As it is shown in Figure 4.11 this frequency is equal to 614.157 MHz.

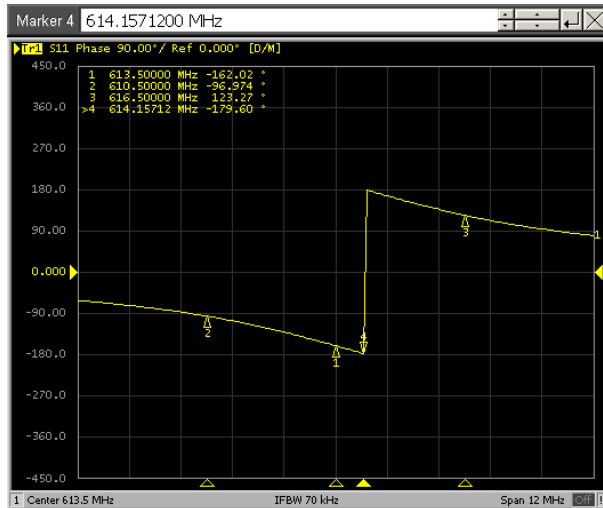


Figure 4.11 Filter response after implementing step 4 of coarse tuning for resonators while coupling 1 is completely withdrawn.

Step 2- Same coupling is inserted into the filters frame. Frequency which corresponds to the phase of $\pm 180^\circ$ is recorded as ω_2 . As it is depicted in Figure 4.12 this frequency is equal to 612.42 MHz.

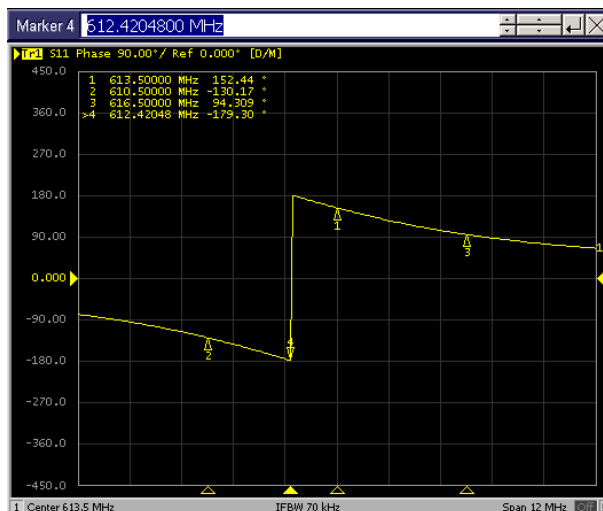


Figure 4.12 Filter response after implementing step 4 of coarse tuning for resonators while coupling 1 is fully inserted into the filters frame.

Step3- Frequency between ω_1 and ω_2 is considered to set the inter-resonator coupling. Let us consider this frequency as ω_3 . Inter-resonator coupling is rotated until phase passes through $\pm 180^\circ$ at ω_3 frequency. As it is demonstrated in Figure 4.13, $\pm 180^\circ$ phase is set to 613.27 MHz frequency.

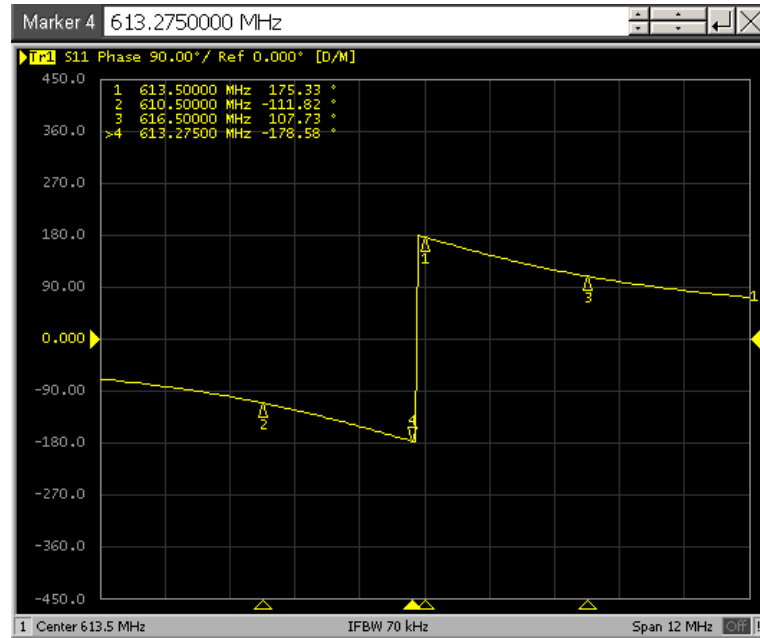


Figure 4.13 Filter response after implementing step 4 of coarse tuning for resonators while coupling 1 is set to the middle of the range that affect the resonator 1.

Step4- Then step 4 of tuning resonators needs to be repeated to bring the resonator 1 to resonance while adjacent coupling is set to the position which is in the middle of the range that can affect the resonance frequency. This assures that coupling is not highly deviated. The same approach is implemented for the rest of the couplings while resonators are tuned sequentially. Similar to the tuning process for resonators, for odd inter-resonators couplings (K1, K3, K5) $\pm 180^\circ$ phase and for even inter-resonators couplings (K2, K4) 0° phase is used.

4.3.3 Coarse Tuning of Cross Couplings

The return loss of the filter cannot be minimized to the desired value if cross coupling deviates from the ideal position. This was experienced when tuning the last resonator (step 9) in the proposed coarse tuning instruction for resonators. In order to overcome this problem the ideal position of the cross coupling needs to be found. For this purpose, automated setup needs to map the entire range of cross coupling. The required mapping resolution needs to be set in automated setup program. The range

of motion for cross coupling in this filter is about 90 degrees. However, all ideal positions observed of cross coupling while the filter is tuned to different channels are located in less than 30 degree range. This range starts from a specific position which can be easily located by creating an absolute positioning system in automated setup (chapter 3). The span of the range is almost the same in different filter types. Since cross coupling can be slightly tuned by tuning the adjacent resonators (R2 and R5), there is no need for high resolution for mapping the 30 degree range. Therefore, initial mapping resolution is considered as 5 degree.

In the next step, automated setup adjusts the cross coupling to the specified starting position of 30 degree range. Filter response in this state is show in Figure (4.14). The filter response is entirely deviated. This is because of high impact of cross coupling on the adjacent resonators (R2, R5).

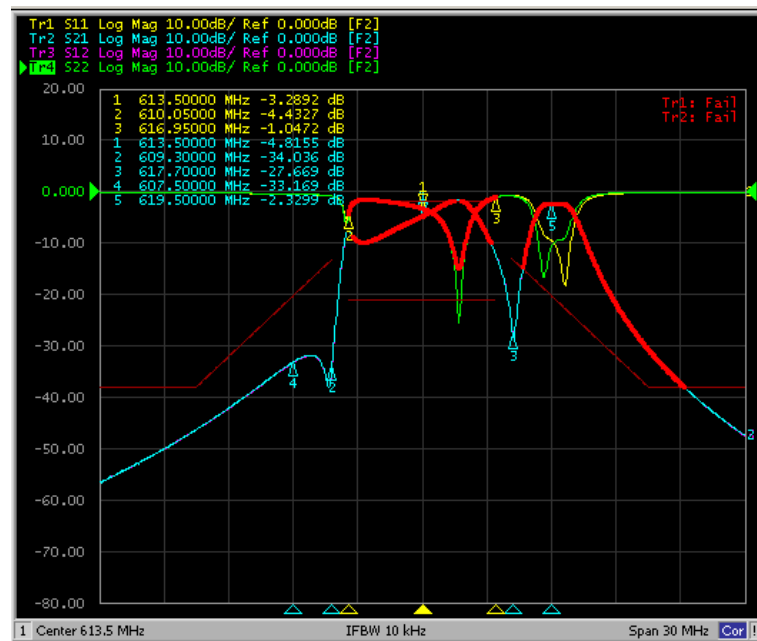


Figure 4.14 Filter response after implementing coarse tuning for resonators while cross coupling is slightly rotated.

In order to bring the resonators back to resonance, coarse tuning steps for resonators needs to be repeated. Since the cross coupling plate is between resonator 2 and 5, only these two resonators needs to be tuned based on measured phase response by VNA. The rest of the resonators can be successively prepositioned by automated setup. As this process needs to be repeated several times, prepositioning of resonators would considerably increase the tuning speed.

After symmetric results are achieved the position of cross coupling is changed. The direction and amount of the rotation is chosen based on amount of return loss measured at the end of the tuning sequence. The amount of rotation is fitted to the set resolution. This process is continued until the achieved return loss at the end of the tuning sequence is lower than the pre-set acceptable return loss in tuning the programs. Since the range of tuning and mapping resolutions are considered as 30 and 5 degree respectively, the required maximum tuning attempts is 6 time. Figure 4.15 demonstrates the filter response after implementing 4 run of sequential coarse tuning method.

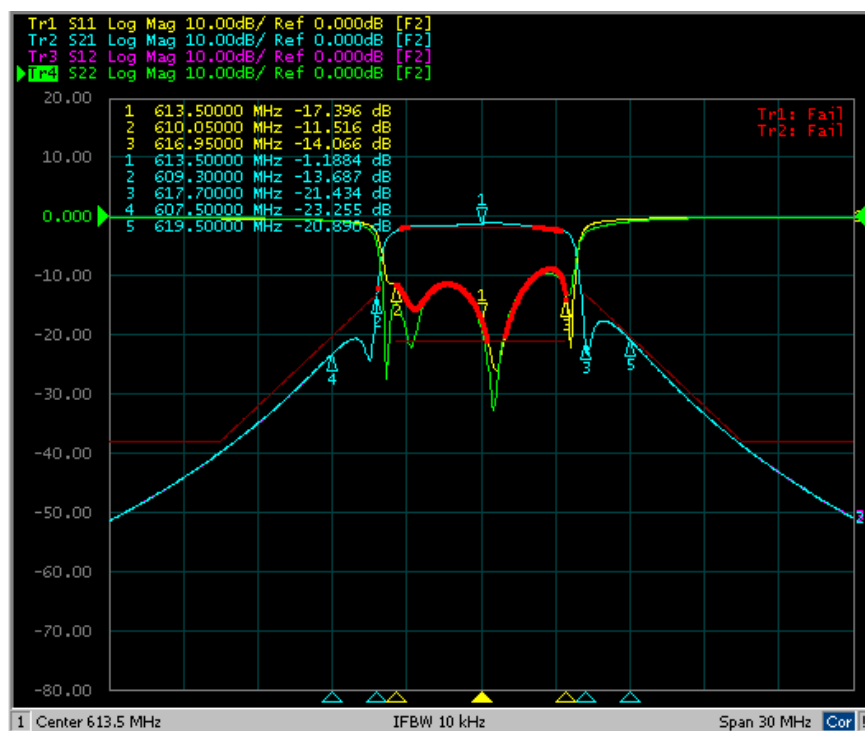


Figure 4.15 Filter response after 4 run sequential tuning while tuning resolution for cross coupling is set to 5 degree.

The response of the filter after coarse tuning is very close to the ideal filter response and a small change in tuning elements can tune the filter to the desired transfer function. The coarse tuning method developed doesn't need any complicated mathematical calculation and it can be done within a short time. Even though this filter response is sufficient to start the fine tuning by circuit model parameter extraction technique the filter response can be improved even more by increasing the mapping resolution. Filter response after tuning by consecutive increase in mapping resolution is shown in Figures 4.16 to 4.19.

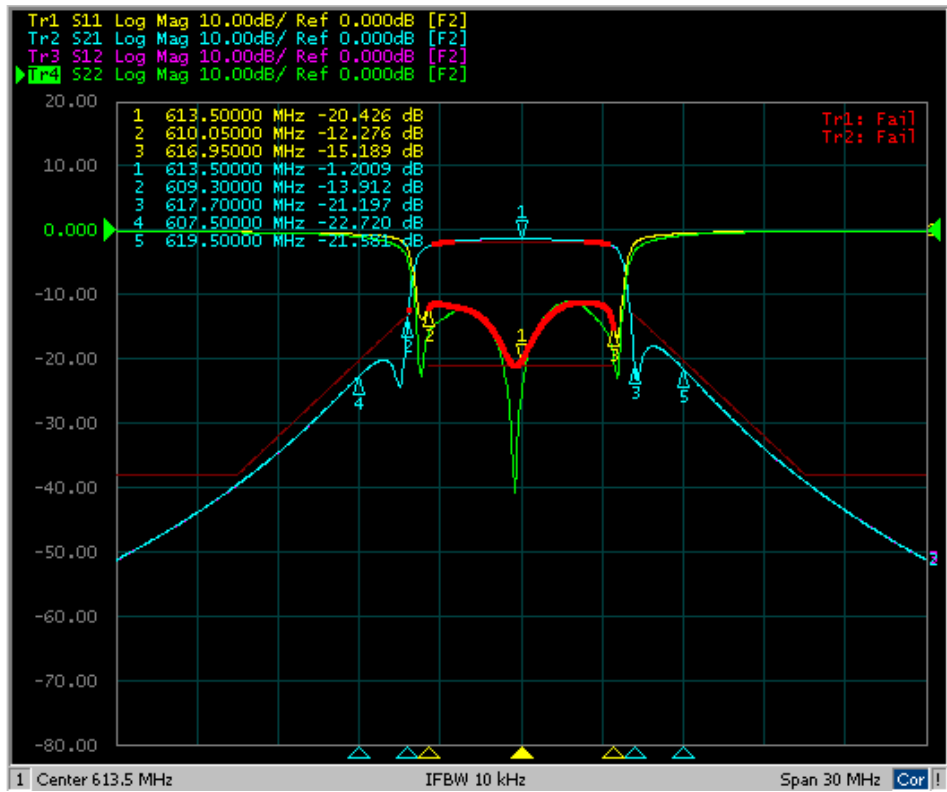


Figure 4.16 Filter response after 6 run sequential tuning while tuning resolution for cross coupling is set to 4 degree.

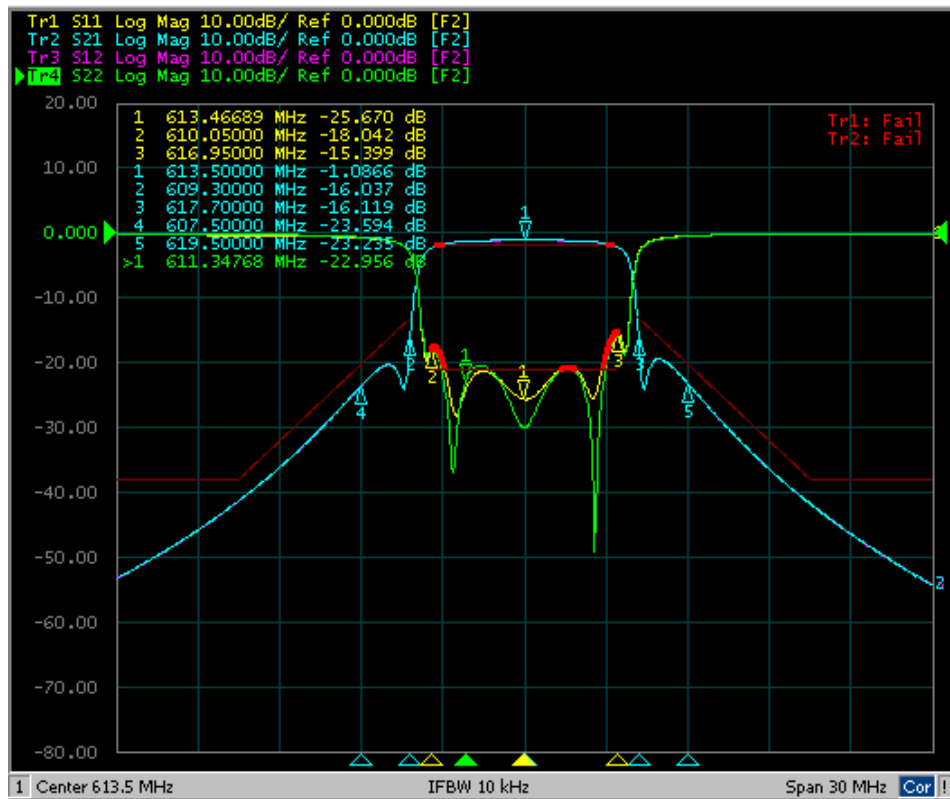


Figure 4.17 Filter response after 8 run sequential tuning while tuning resolution for cross coupling is set to 3 degree.

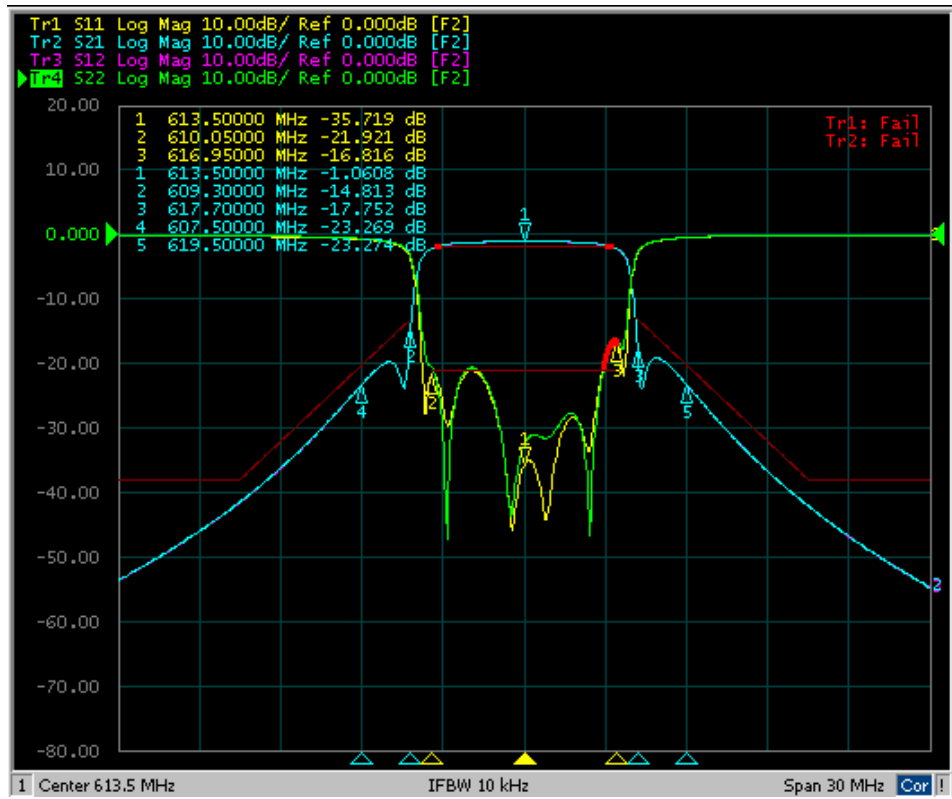


Figure 4.18 Filter response after 10 run sequential tuning while tuning resolution for cross coupling is set to 2 degree.

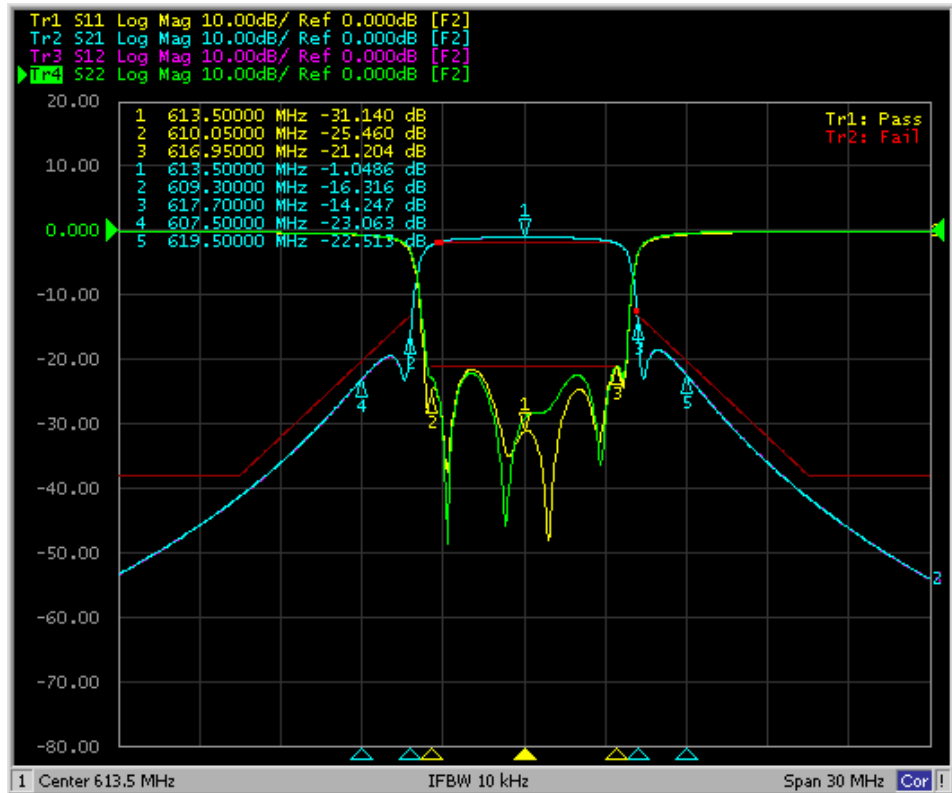


Figure 4.19 Filter response after 12 run sequential tuning while tuning resolution for cross coupling is set to 1 degree.

4.4 Fine Tuning

Circuit model parameter extraction tuning technique which is introduced by Thal et al. [31] allows the lead screws of the filter to be tuned simultaneously as well as allowing the mistuned elements to be aligned in any stage of the tuning without need for restarting the whole tuning process. Because of these advantages, this technique is the preferred fine tuning technique by filter companies in their CAT software. In established tuning processes, CAT software based on this technique is used to guide the technicians through the fine tuning process. In this study, the code based on this technique is used by RoboCAT to perform the fine tuning automatically. Since sufficiently close initial values for the variables to be optimised are required in order to utilise the circuit model parameter extraction technique, first filter needs to be roughly tuned by performing the proposed coarse tuning technique in Section 4.3. As a result of coarse tuning the tuned filter is very close to the ideal filter (tuned filter) response, it is now feasible to implement the circuit model parameter extraction technique to increase the precision of the tuned filter.

There are several circuit models available to be employed in circuit model parameter extraction technique; however, the advantage of using the coupling matrix circuit model is that the coupling parameters are directly related to the physical position of the tuning elements. Figure 4.20 shows the general equivalent circuit of coupled resonator filters. This circuit model is equal to coupling matrix circuit model proposed in [46] with the difference that all the resonators include resistors and the coupling values are considered as complex values [47]. In this circuit model, all the possible cross couplings are shown. The circuit consists of N asynchronously tuned resonators. All the resonators are represented by a LC circuit loop with their losses modelled by the resistance r_i . The coupling between the resonators i and j are modelled by M_{ij} .

Ignoring the losses in each loop, the tuning steps can be briefly summarized as:

Step 1- Measure the performance of the filter being tuned.

Step 2- Considering the coupling matrix circuit model, coupling elements M_{ij} , $R1$ and $R2$ are extracted by optimisation for best fit with measured data. This optimisation process is achieved by using Equations (4.1)-(4.6) where I is unity

matrix and R is a diagonal matrix with all elements set to zero except $R_{11} = R_1$ and $R_{nn} = R_2$. An objective function to be minimized is defined as Equation (2.12) while measured performance of the filter should be sampled at several frequencies within the band.

$$M = \begin{bmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nn} \end{bmatrix} \quad (4.1)$$

$$S_{21} = -2j\sqrt{R_1 R_2} [A^{-1}]_{n1} \quad (4.2)$$

$$S_{11} = 1 + 2jR_1 [A^{-1}]_{11} \quad (4.3)$$

$$A = \lambda I - jR + M \quad (4.4)$$

$$\lambda = \frac{f_0}{BW} \left(\frac{f}{f_0} - \frac{f_0}{f} \right) \quad (4.5)$$

$$\varphi = \sum_{freq} \sum_{i=1}^2 \sum_{j=1}^2 \left(abs(S_{ij}^{model}) - abs(S_{ij}^{measured}) \right) \quad (4.6)$$

Step 3- The coupling matrix elements extracted from the experimental results, are compared with the ideal coupling matrix elements. Relative tuning elements are tuned accordingly.

Step 4) Steps 1-3 are repeated until the extracted coupling elements from measured data match the ideal coupling elements.

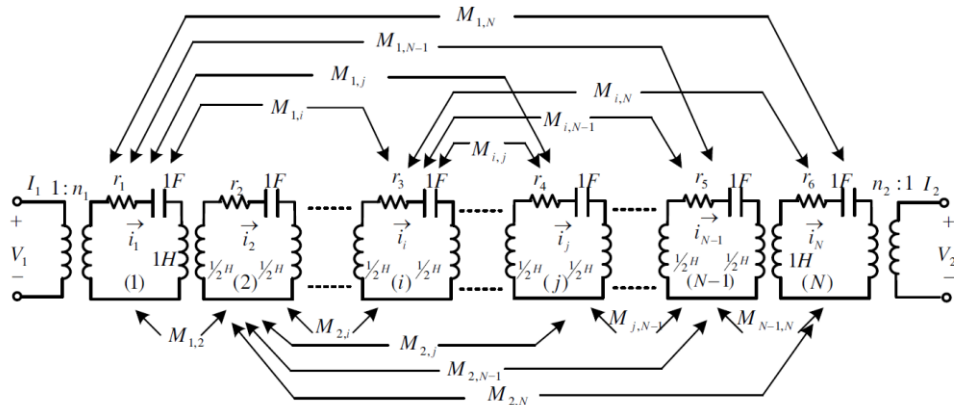


Figure 4.20 Circuit model with all possible cross couplings.

Tuning result shown in Figure 4.21 is achieved by performing this fine tuning technique via RoboCAT setup. This setup uses the proposed coarse tuning technique in Section 4.3 and then circuit model parameter extraction technique from the literature in order to finalize the tuning process.

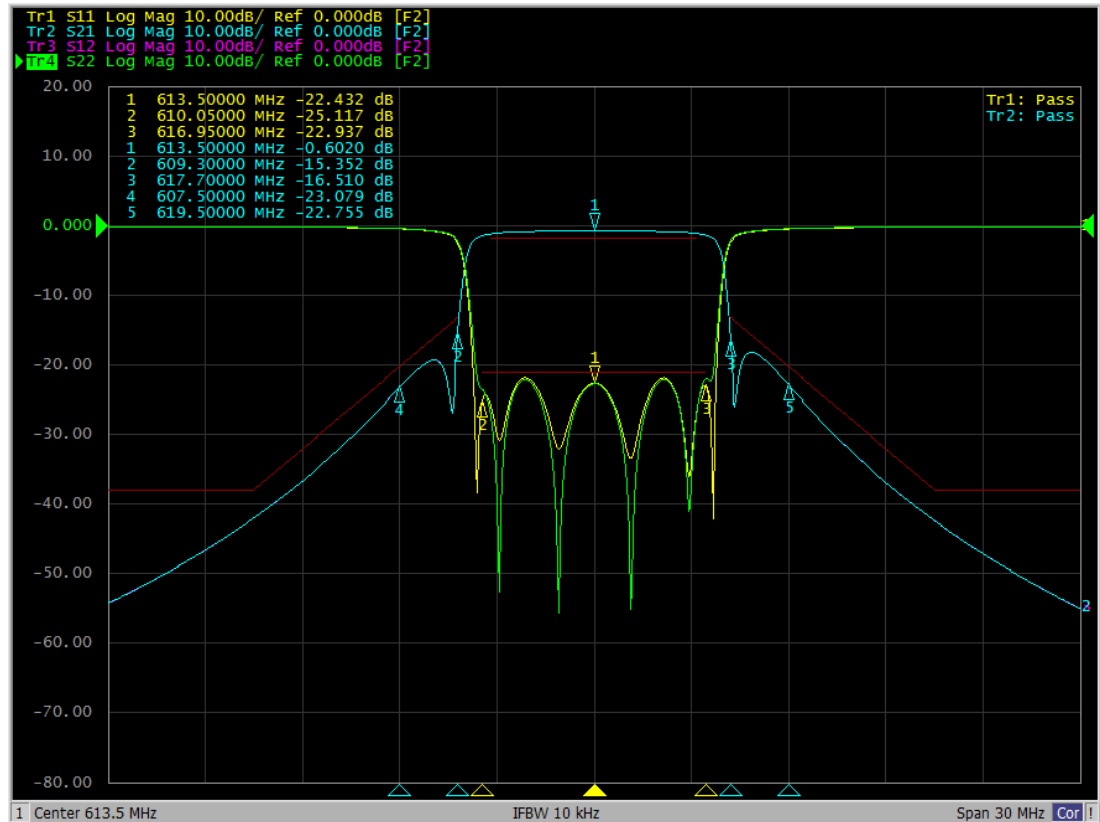


Figure 4.21 Measured filter response by VNA after implementing fine tuning.

Figure 4.22 shows the Graphical User Interface (GUI) made for fine tuning software by RFS Pty Ltd. This software, which is called Broadcast Computer Aided Tuning (BCAT) software, is used to guide the technicians through a manual tuning process. Parameter extraction tuning technique is used in algorithms implemented by RFS in this software. Then bar-graphs on this GUI indicate the error related to each tuning element. As illustrated in Figure 4.22, final filter response after tuning by RoboCAT passes the tuning process and all bar-graphs are reduced to less than 10 KHz deviation.

Even though the filter response in Figure 4.22 passed the tuning process, precision of the tuned filter can be increased by using the higher tuning resolution of the RoboCAT. Accuracy of this automated tuner is illustrated by the achieved results in

Figure 4.23. As it is shown, frequency deviation of all tuning elements is reduced to less than 5 KHz. This much precision is unattainable by manual tuning.

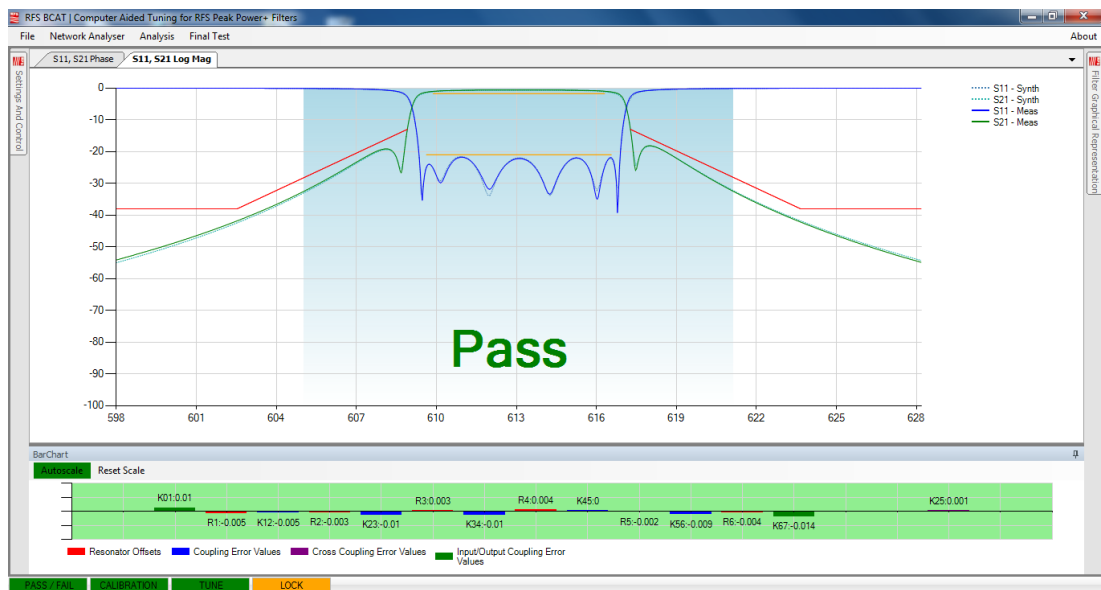


Figure 4.22 Measured filter response by BCAT while RoboCAT pass the tuning process.



Figure 4.23 Measured filter response by BCAT while higher resolution in tuning is utilized by RoboCAT.

The time spent for twenty separate tuning attempts is illustrated in Figure 4.24. The six-pole filter depicted in Figure 4.1 is used in all runs. The top graph corresponds to manual tuning attempts where BCAT was used to guide the human operator through tuning. The bottom graph is related to tuning by RoboCAT. The average tuning time in manual tuning is about 42 minutes while this value is reduced to 6.3 minutes by

the designed automated tuner. The reduced tuning time proves the effectiveness of the proposed setup in production line of the filter companies.

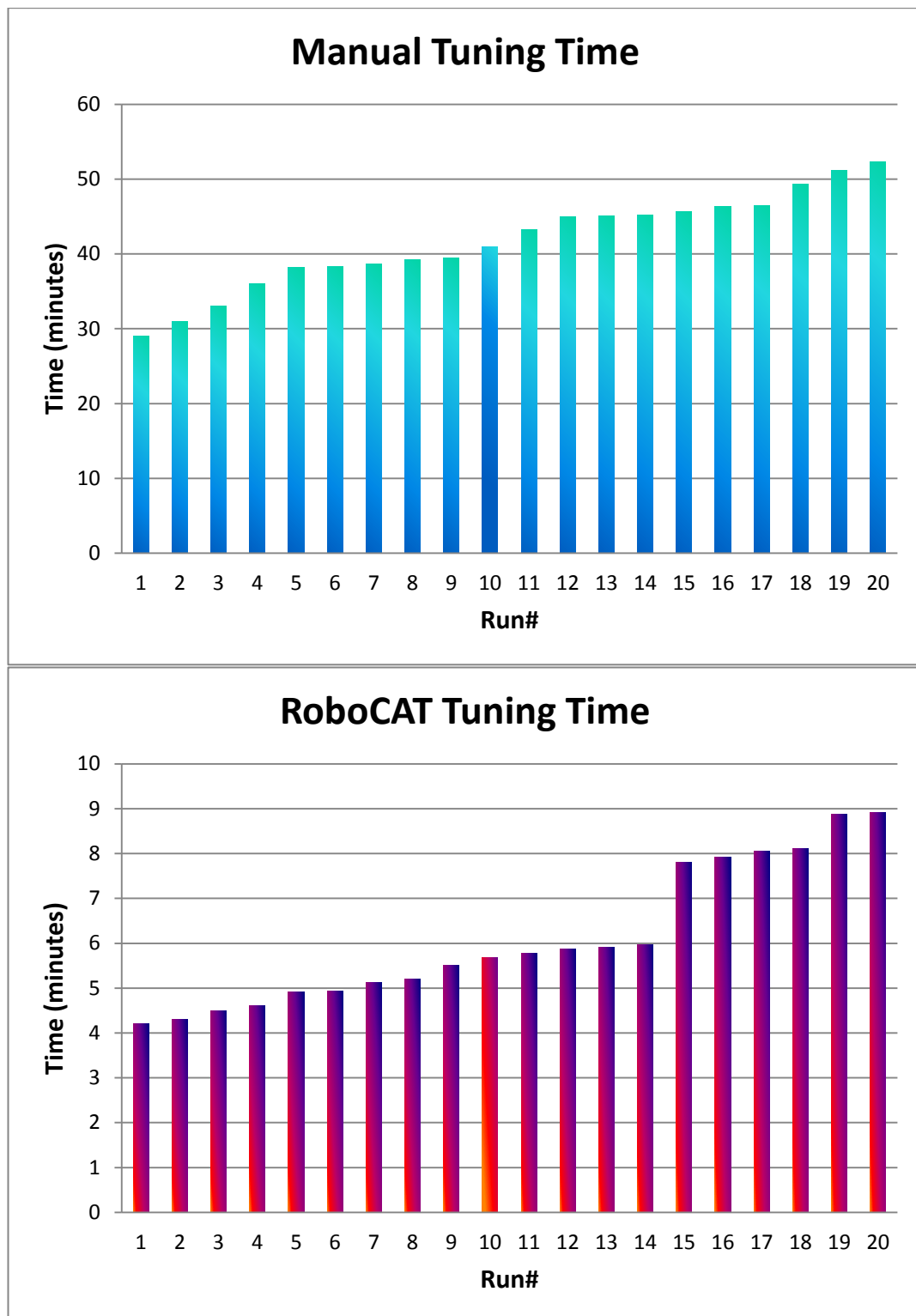


Figure 4.24 Required tuning time for 6-pole multi-cavity filter in 20 run.

4.5 Conclusion

In this chapter a new approach to coarse tuning a cross-coupled multi-cavity filter was introduced. Unlike several methods introduced in literature, sequential tuning of the resonators is based on the phase format of the input reflection coefficient S_{11} which does not have computational delay in extracting tuning data found in group delay [17, 18], time domain [19–21], fuzzy logic [16], [22–24], poles and zeros [29] or circuit model parameter extraction [12], [31–37] techniques in the literature and so reduces the overall tuning time considerably. The sequential tuning solution based on phase format of the filter response is a novel tuning method for complicated cross-coupled filters. The validity of the introduced tuning instruction was affirmed by obtained experimental results. These results were acquired by implementing the proposed technique with an automated setup which was introduced in chapter 3. Tuning outcomes demonstrated that the phase format of input reflection coefficient S_{11} passes through $\pm 180^\circ$ and 0° crossing at center frequency ω_0 as the successive resonators are tuned. Obtained filter response at the end of the tuning was close to the ideal filter response. This provided very close initial values for the variables to be optimised by fine tuning programs and consequently reduced the fine tuning time from 13 minutes to less than 32 seconds.

As it is practical to implement the proposed coarse tuning instruction by automated setup and roughly tune the filter, it is then feasible to implement the circuit model parameter extraction instruction by robot. In the second section of this chapter, circuit model parameter extraction technique was practically implemented by the author developed RoboCAT to finalize the fine tuning. The effectiveness of implementing this technique by RoboCAT was proved by obtained results which achieved a tuned filter in average 6.3 minutes compared to the manual tuning approach which took 42 minutes in average. Frequency deviation of all tuning elements is reduced to less than 5 KHz which is unattainable by manual tuning. These results were obtained from 20 tuning attempts on a six pole cross coupled filter.

Chapter 5

CONCLUSION

5.1 Automation

In this thesis a customised coaxial screw/nut driver for automated filter tuning systems was introduced. The simultaneous locking and tuning capability of the designed driver enables the robot to lock the tuning elements before the tuner was removed, which prevents any unwanted change as a result. This customised tuner allowed the proposed backlash compensation solution to be implemented by automated setup. This solution successfully negated the effect of mechanical backlash within the tuning process. By using this technique, the created automated system was able to eliminate the time the robot needed to map the entire backlash in each change of direction. Results showed that using the backlash compensation solution in twenty tuning attempts reduced 11 minutes and 23 seconds from the overall tuning time.

To further optimize the design mechanical stoppers were mounted on the inner cylinder of the coaxial screw/nut driver which limited the vertical and rotational movements of the lead screws. This allowed the robot to utilise an absolute positioning system which was able to provide exact position of the lead screws based on a known position called 'Home step'. While CAT data was used as a first feedback source for the automated tune, data of the positioning system was used as a second feedback source for the automated setup to enable the robot to detect the potential error in extracted data by CAT software at the early stage of its occurrence. This prevented the automated setup from blindly following the faulty data of CAT

software in case of an error. This characteristic of the automated setup enabled it to perform stand-alone in an industrial environment. In literature, only the proposed automated setup by COM DEV introduced a solution for potential error in CAT data. In this setup, for the purpose of fault recognition, the robot's software tunes the filter for a prescribed maximum tuning time and if the tuning process continues further, then it considers the CAT response as an error and restarts the software. Even though this solution may by-pass the said problem; however, by overturning the actuators during blind tuning process, tuning elements may fall into filters housing or completely withdrawn from the filter. Plus, errors in software usually occur at the beginning of the procedure depending on the extracted initial values for the variables to be optimized. Thus, a system needs to wait for the prescribed maximum time to recognize the existence of an error which doubles the duration of the tuning process. Therefore, outlined test setups in the literature are not designed to perform stand-alone in industrial environment.

This comprehensive coaxial/screw nut driver was designed to fit with SCARA, Cartesian and Multi-armed setups in order to tune different type of filters in the company. All mentioned aspects of the created tuner were successfully utilized in the tuning process presented in chapter 4 to obtain the experimental results. Fault recognition ability and anti-backlash tuners in the created automated tuning system reduced the tuning time considerably during experimental tunings.

5.2 Tuning

In this thesis, a novel coarse tuning technique was introduced as a prerequisite to a circuit model parameter extraction technique in literature. In this method, resonators were tuned by bringing successive resonators to resonance, while the phase passes through the $\pm 180^\circ$ and 0° crossing at the center frequency. At the end of each sequence cross coupling was mapped across the entire range of its motion. Tuned cross coupling was recognized by measuring the return loss of the filter. Thus, unlike the proposed methods in the literature, cross coupling was easy to handle through the tuning process as it didn't need complicated calculations to be tuned. As coarse tuning technique were successfully implemented by the automated setup, it was then feasible to implement the circuit model parameter extraction tuning technique. Proposed coarse tuning technique set adequate initial optimization variable values to

be employed by the circuit model parameter extraction technique. This successfully prevented the system from running into local minimum or failing to converge to the proper solution during the fine tuning process. In the second section of this chapter, a circuit model parameter extraction technique was practically implemented by the RoboCAT, to fine tune the filter.

Since the tuned filter response by the proposed coarse tuning method was very close to an ideal filter response, fine tuning time was reduced from 13 minutes to less than 32 seconds. This reduced the overall tuning time significantly as a slow response due to the fine tuning codes make the tuning process very slow. The coarse tuning time was also reduced as proposed technique does not have a computational delay in extracting tuning data found in group delay, time domain, fuzzy logic, poles and zeros or circuit model parameter extraction techniques in the literature.

Validity of the introduced tuning solution for complicated cross-coupled filters was affirmed through experimental results. These results obtained from 20 tuning attempt on a six-pole cross-coupled filter. The effectiveness of this technique was proved by achieved a tuned filter by RoboCAT in average 6.3 minutes compared to the manual tuning approach which took average 42 minutes.

5.3 Future Work

This research has developed the basis for potential studies in the future which is to find a proper artificial neural network solution for the fine tuning process. Now that RoboCAT is able to tune the filters within 6 minutes, it is feasible to work on a backpropagation algorithm for artificial neural network. This can be done by detuning many tuned filters by RoboCAT in order to find the proper weights between the nodes. Written CAT based on this algorithm can tune any type of filter without the need for any pre-set data about the filters structure.

REFERENCES

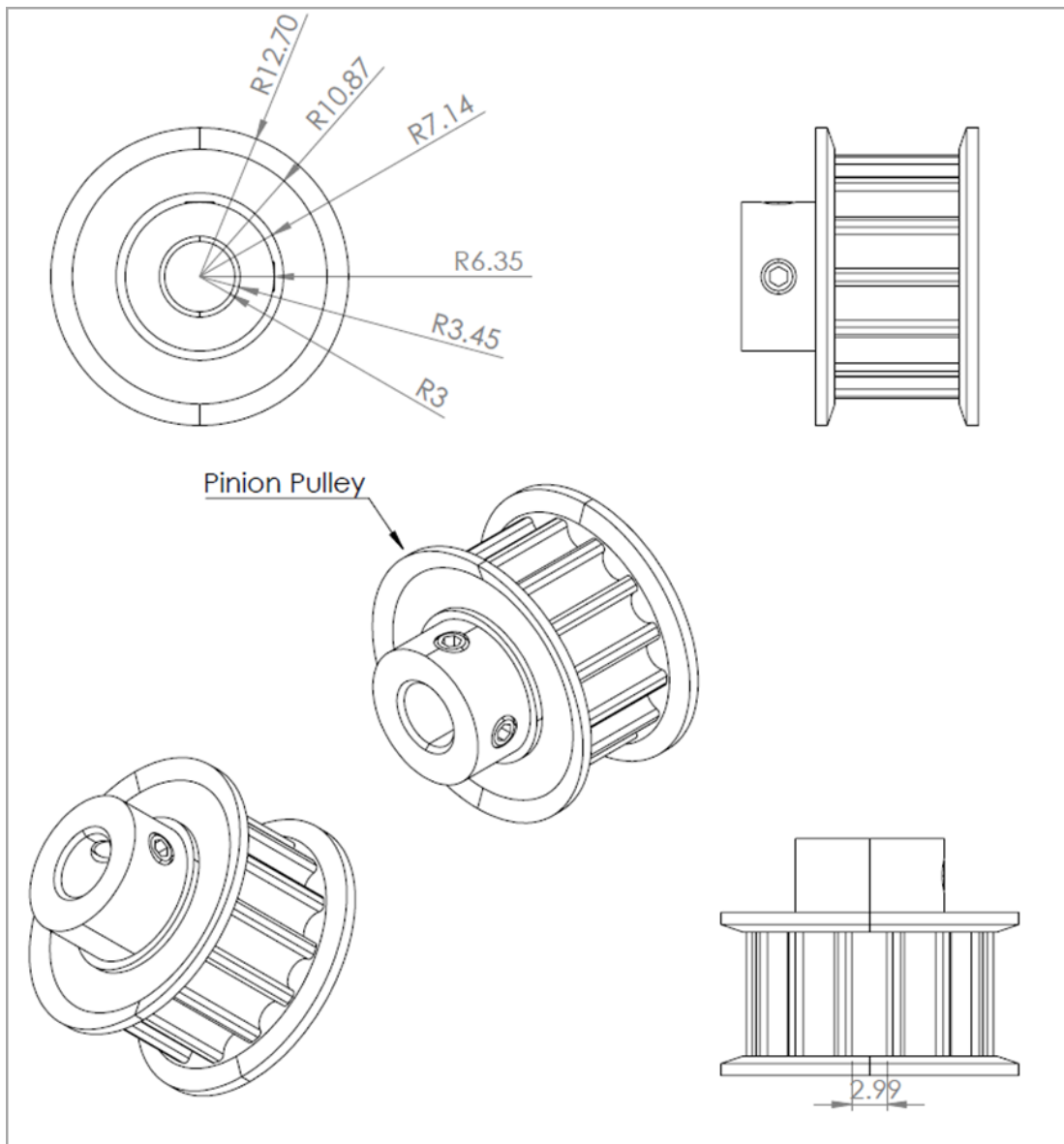
- [1] Australian Government Department of Communications, “Digital TV in Australia.” [Online]. Available: http://www.communications.gov.au/television/digital_tv_in_australia.
- [2] H. A. Wheeler, “Tuning of waveguide filters by perturbing of individual sections,” *Proc. Symp. Modern Advances in Microwave Technology*. pp. 343–353, 1954.
- [3] A. E. Atia and A. E. Williams, “Measurements of Intercavity Couplings (Short Papers),” *Microwave Theory and Techniques, IEEE Transactions on*, vol. 23, no. 6. pp. 519–522, 1975.
- [4] A. E. Williams, R. G. Egri, and R. R. Johnson, “Automatic Measurement of Filter Coupling Parameters,” *Microwave Symposium Digest, 1983 IEEE MTT-S International*. pp. 418–420, 1983.
- [5] T. Ishizaki, H. Ikeda, T. Uwano, M. Hatanaka, and H. Miyake, “A computer aided accurate adjustment of cellular radio RF filters,” *Microwave Symposium Digest, 1990., IEEE MTT-S International*. pp. 139–142 vol.1, 1990.
- [6] A. R. Mirzai, C. F. N. Cowan, and T. M. Crawford, “Intelligent alignment of waveguide filters using a machine learning approach,” *Microwave Theory and Techniques, IEEE Transactions on*, vol. 37, no. 1. pp. 166–173, 1989.
- [7] K. Antreich, E. Gleissner, and G. Muller, “Computer aided tuning of electrical circuits,” *Nachrichtentechnische Zeitschrift* 28, no. 6. pp. 200–206, 1975.
- [8] R. L. Adams and V. K. Manaltala, “An optimization algorithm suitable for computer-assisted network tuning,” *Proc. IEEE Int. Symp. Circuits Systems*. Newton, MA, pp. 210–212, 1975.
- [9] J. W. Bandler and A. E. Salama, “Functional Approach to Microwave Postproduction Tuning,” *Microwave Theory and Techniques, IEEE Transactions on*, vol. 33, no. 4. pp. 302–310, 1985.

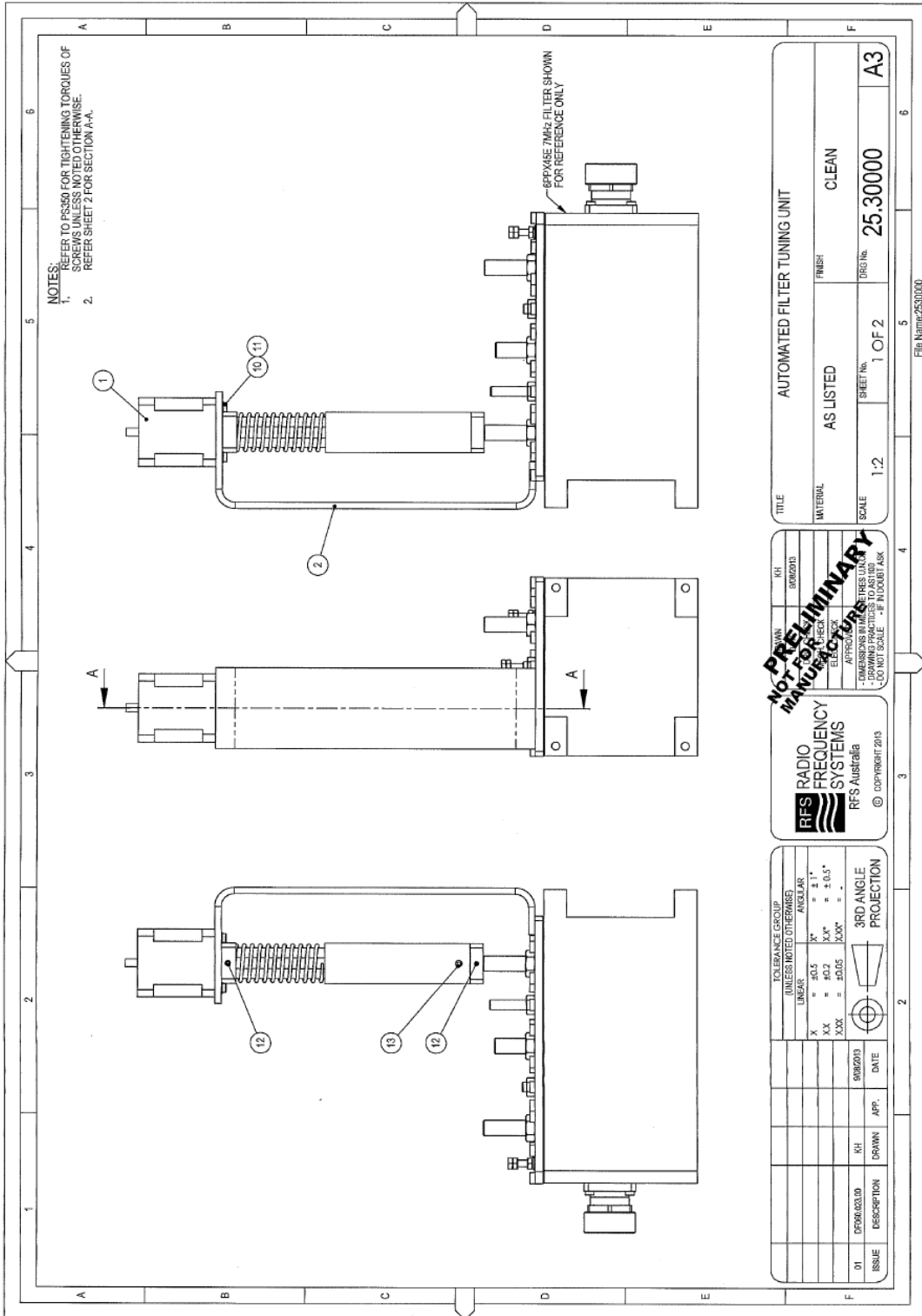
-
- [10] P. V Lopresti, "Optimum design of linear tuning algorithms," *Circuits and Systems, IEEE Transactions on*, vol. 24, no. 3. pp. 144–151, 1977.
- [11] C. J. Alajajian, T. N. Trick, and E. I. El-Masry, "On the design of an efficient tuning algorithm," *Proc. IEEE Int. Symp. Circuits Systems*. Houston, TX, pp. 807–811, 1980.
- [12] P. Harscher, R. Vahldieck, and S. Amari, "Automated filter tuning using generalized low-pass prototype networks and gradient-based parameter extraction," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 49, no. 12. pp. 2532–2538, 2001.
- [13] Y. L. Zhang, P. P. Li, and J. X. Yan, "A Practical Method of Computer-Aided Tuning for Microwave filters," *Communication Problem-Solving (ICCP), 2014 IEEE International Conference on*, pp.39-42, 2014.
- [14] M. M. M. Meng and K.-L. W. K.-L. Wu, "An Analytical Approach to Computer-Aided Diagnosis and Tuning of Lossy Microwave Coupled Resonator Filters," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 57, no. 12, pp. 3188–3195, 2009.
- [15] M. Yu, "Robotic Computer-Aided Tuning," *Microwave Journal*, pp. 136–138, Mar-2006.
- [16] V. Miraftab and R. R. Mansour, "Fully automated RF/microwave filter tuning by extracting human experience using fuzzy controllers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 5. pp. 1357–1367, 2008.
- [17] M. Dishal, "Alignment and Adjustment of Synchronously Tuned Multiple-Resonant-Circuit Filters," *Proceedings of the IRE*, vol. 39, no. 11. pp. 1448–1455, 1951.
- [18] J. B. Ness, "A unified approach to the design, measurement, and tuning of coupled-resonator filters," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 46, no. 4, pp. 343–351, 1998.
- [19] J. Dunsmore, "Tuning band pass filters in the time domain," *1999 IEEE MTT-S International Microwave Symposium Digest (Cat. No.99CH36282)*, vol. 3. pp. 1351–1354 vol.3, 1999.
- [20] Application Note 1287-8: Simplified Filter Tuning Using Time Domain Agilent Technologies Corp., 2001. Palo Alto, CA.
- [21] J. Dunsmore, "Advanced filter tuning using time domain transforms," *Microwave Conference, 1999. 29th European*, vol. 2, pp. 72–75, 1999.
- [22] V. Miraftab and R. R. Mansour, "Computer-aided tuning of microwave filters using fuzzy logic," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 12. pp. 2781–2788, 2002.

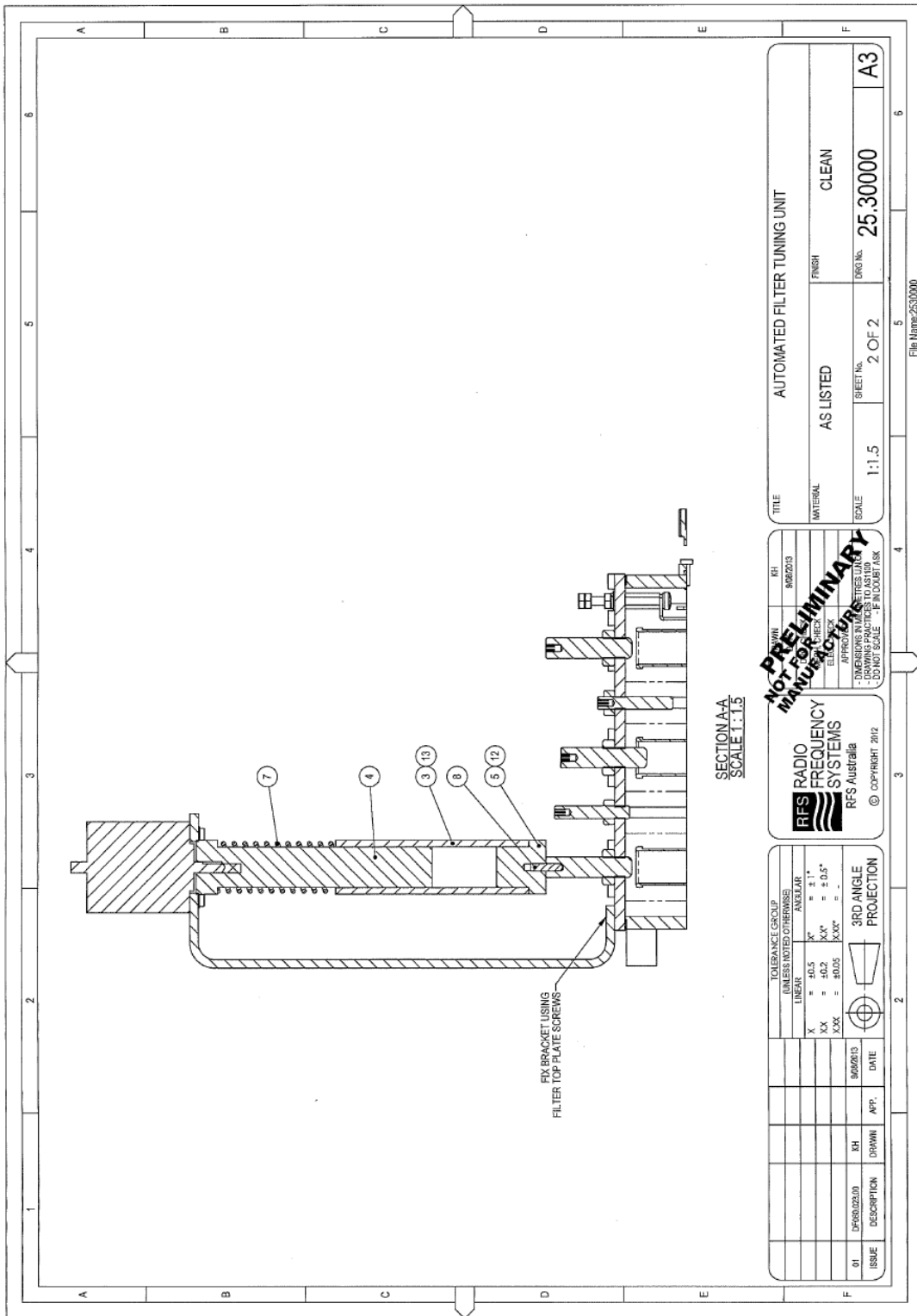
- [23] V. Miraftab and R. R. Mansour, "Computer-aided tuning of microwave filters using fuzzy logic," *Microwave Symposium Digest, 2002 IEEE MTT-S International*, vol.2, pp. 1117–1120 vol.2, 2002.
- [24] V. Miraftab and R. R. Mansour, "A robust fuzzy-logic technique for computer-aided diagnosis of microwave filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 1. pp. 450–456, 2004.
- [25] L. a. Zadeh, "Fuzzy Sets," *Information and control* 8, no. 3. pp. 338–353, 1965.
- [26] M. Sugeno and T. Yasukawa, "Fuzzy-logic-based approach to qualitative modeling," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1. pp. 7–31, 1993.
- [27] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22. no. 6, pp. 1414–1427, 1992.
- [28] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1. pp. 116–132, 1985.
- [29] H. T. Hsu, H. W. Yao, K. a. Zaki, and A. E. Atia, "Computer-aided diagnosis and tuning of cascaded coupled resonators filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 4. pp. 1137–1145, 2002.
- [30] A. E. Atia and A. E. Williams, "Nonminimum-Phase Optimum-Amplitude Bandpass Waveguide Filters," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 22, no. 4. pp. 425–431, 1974.
- [31] H. L. Thal, "Computer-Aided Filter Alignment and Diagnosis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 26. pp. 217–219, 1978.
- [32] L. Accatino, "Computer-Aided Tuning of Microwave Filters," *Microwave Symposium Digest, 1986 IEEE MTT-S International*. pp. 249–252, 1986.
- [33] M. Yu, "Computer Aided Tuning," *COM DEV Internal report*, 1994.
- [34] M. Kahrizi, S. Safavi-Naeini, and S. K. Chaudhuri, "Computer diagnosis and tuning of microwave filters using model-based parameter estimation and multi-level optimization," *Microwave Symposium Digest. 2000 IEEE MTT-S International*, vol. 3. pp. 1641–1644 vol.3, 2000.
- [35] P. Harscher and R. Vahldieck, "Automated computer-controlled tuning of waveguide filters using adaptive network models," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 49, no. 11. pp. 2125–2130, 2001.

-
- [36] M. Yu, "Advanced Simulation / Design Techniques for Microwave Filters - An Engineering Perspective," *Workshop WSA: State-of-the-Art Filter Design using EM and Circuit Simulation Techniques*, International Symposium of IEEE Microwave Theory and Techniques. Phoenix, May-2001.
- [37] G. Pepe, F. Gortz, and H. Chaloupka, "Computer-aided tuning and diagnosis of microwave filters using sequential parameter extraction," *Microwave Symposium Digest, 2004 IEEE MTT-S International*, vol. 3. pp. 1373–1376 Vol.3, 2004.
- [38] M. Yu, "A Fully Automated Filter Tuning Robot for Wireless Base Station Duplexers," *revised version from 2003's IEEE MTT conference IMS*. 2003.
- [39] P. Harscher, J. Hofmann, R. Vahldieck, and B. Ludwig, "Automatic Computer-Controlled Tuning System for Microwave Filters," *Microwave Conference, 2000. 30th European*, pp.1-4, Oct. 2000.
- [40] Telemobile Electronics, "TrimSolutions IAFTT Robot - microwave filter tuning." [Online]. Available: <http://www.telemobile.net.pl/>.
- [41] A. Slocum, *FUN da MENTALS of Design*, Pappalardo. Cambridge-MIT Institute, 2008.
- [42] "TEA-Timing-Belts-and-Pulleys.pdf." T.E.A. Transmissions, Australia.
- [43] V.S Bagad, *Mechatronics*. Technical Publications Pune, 2008.
- [44] K. Ichige, Y. Takeuchi, K. Miyamoto, and Y. Ebine, "Successive optimization of zeros of reflection characteristics for automated microwave filter tuning," *Microwave Symposium Digest (MTT), 2011 IEEE MTT-S International*, pp.1-1, 2011.
- [45] C. Kwak, M. Uhm, I. Yom, H. J. Eom, and S. Member, "Automated Microwave Filter Tuning Using Curve Similarity and Weighted Least Squares," *Microwave and Wireless Components Letters, IEEE*, vol.22, no.10, pp.539-541, 2012
- [46] A. Atia, A. Williams, and R. Newcomb, "Narrow-band multiple-coupled cavity synthesis," *IEEE Transactions on Circuits and Systems*, vol. 21, no. 5. pp. 649–655, 1974.
- [47] V. Miraftab and M. Yu, "Generalized lossy microwave filter coupling matrix synthesis and design using mixed technologies," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 12. pp. 3016–3027, 2008.

APPENDIX A: DESIGNED AND FABRICATED MECHANICAL PARTS







SECTION AA
SCALE 1:1.5

TITLE	AUTOMATED FILTER TUNING UNIT	
MATERIAL	AS LISTED	FINISH CLEAN
SCALE	1:1.5	SHEET No. 2 OF 2 DRG No. 25.30000
		A3

PRELIMINARY
NOT FOR MANUFACTURE
APPROVED FOR CONSTRUCTION
- DIMENSIONS ARE TO ASSUMED UNLESS OTHERWISE SPECIFIED -
- DO NOT SCALE - IF IN DOUBT ASK

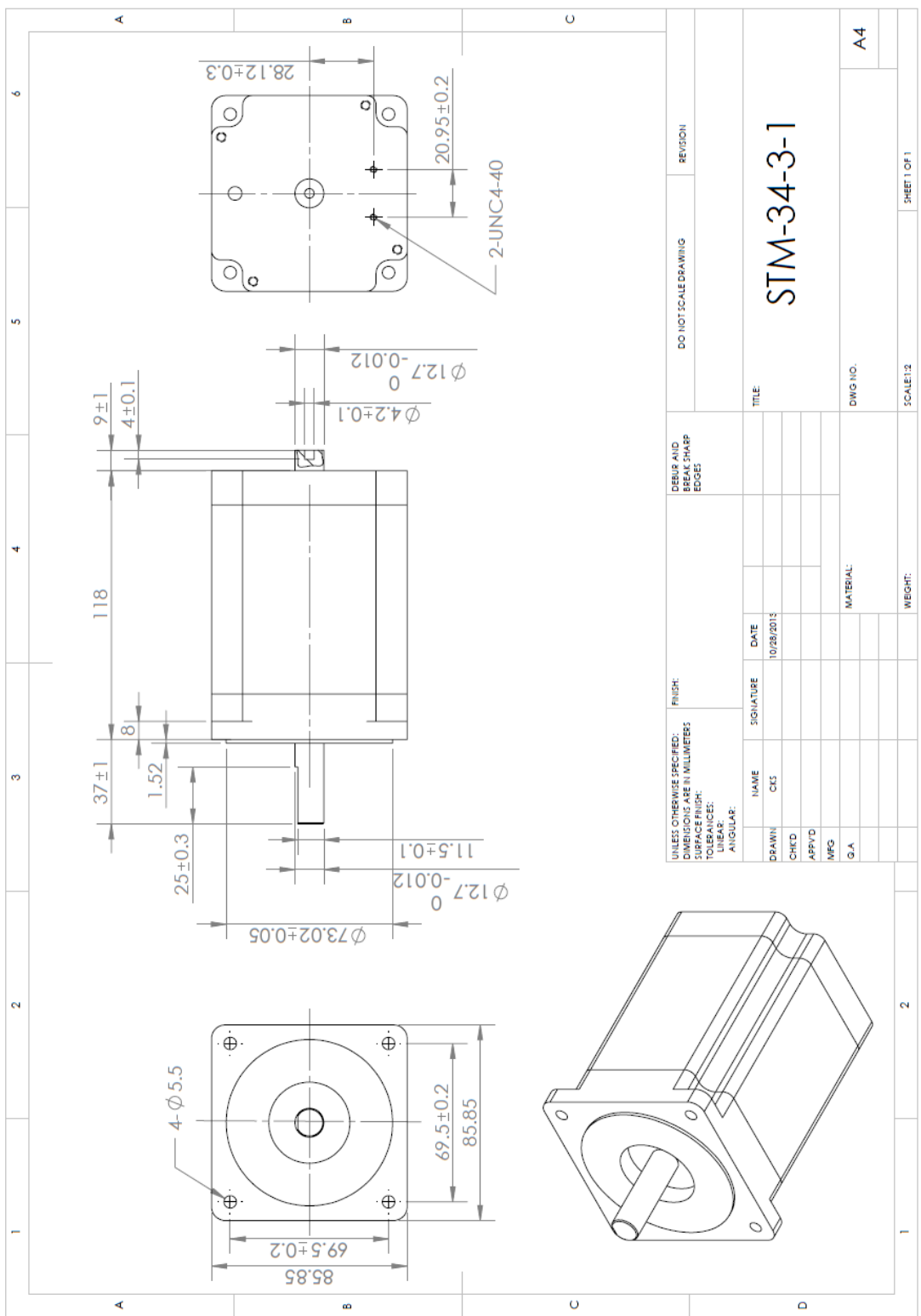
RFS RADIO FREQUENCY SYSTEMS
RFS Australia
© COPYRIGHT 2012

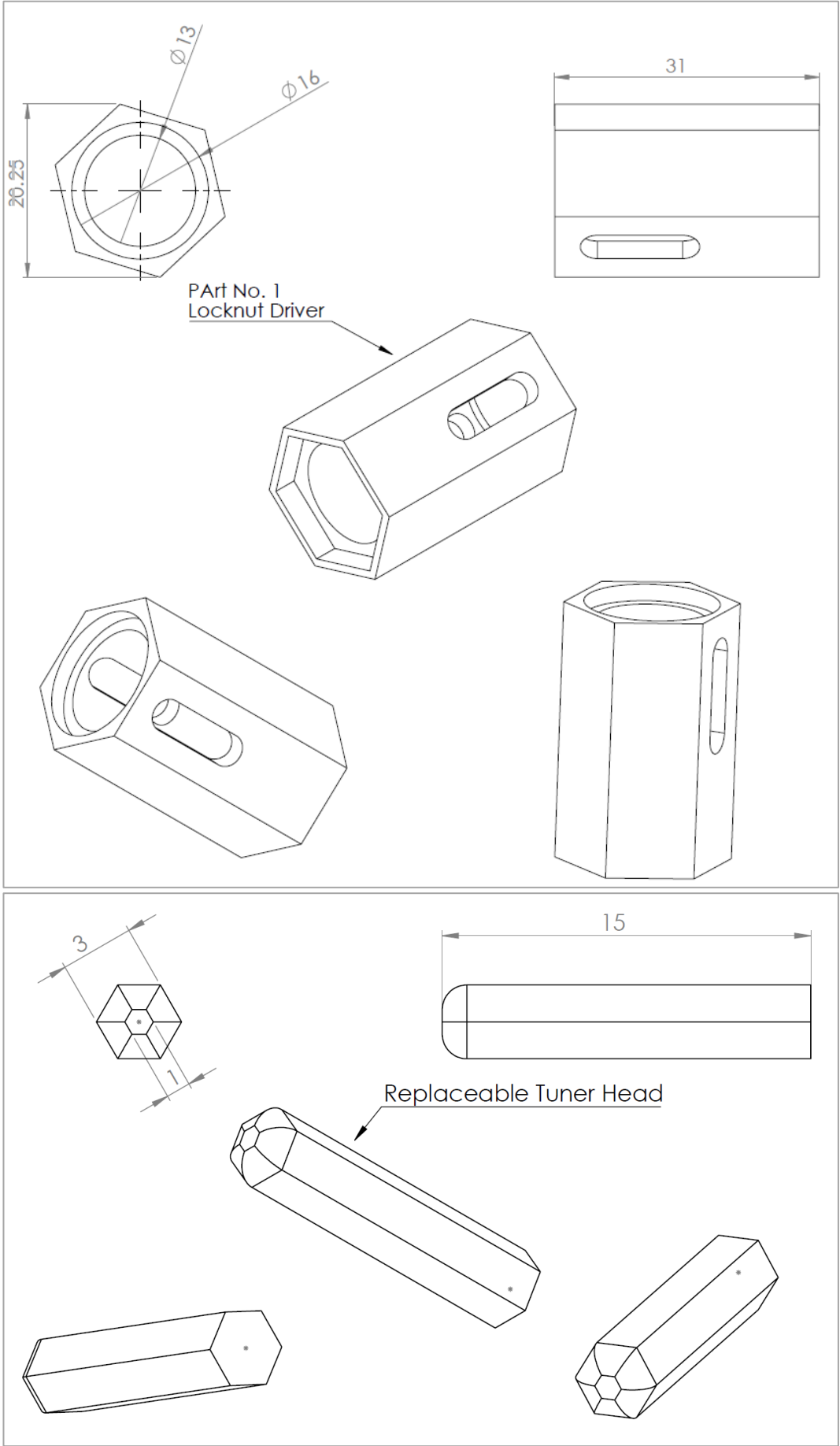
TOLERANCE GROUP		DIMENSIONS NOTED OTHERWISE	
LINEAR		ANGULAR	
X	±0.5	X°	±1°
XX	±0.2	XX°	±0.5°
XXX	±0.05	XXX°	-

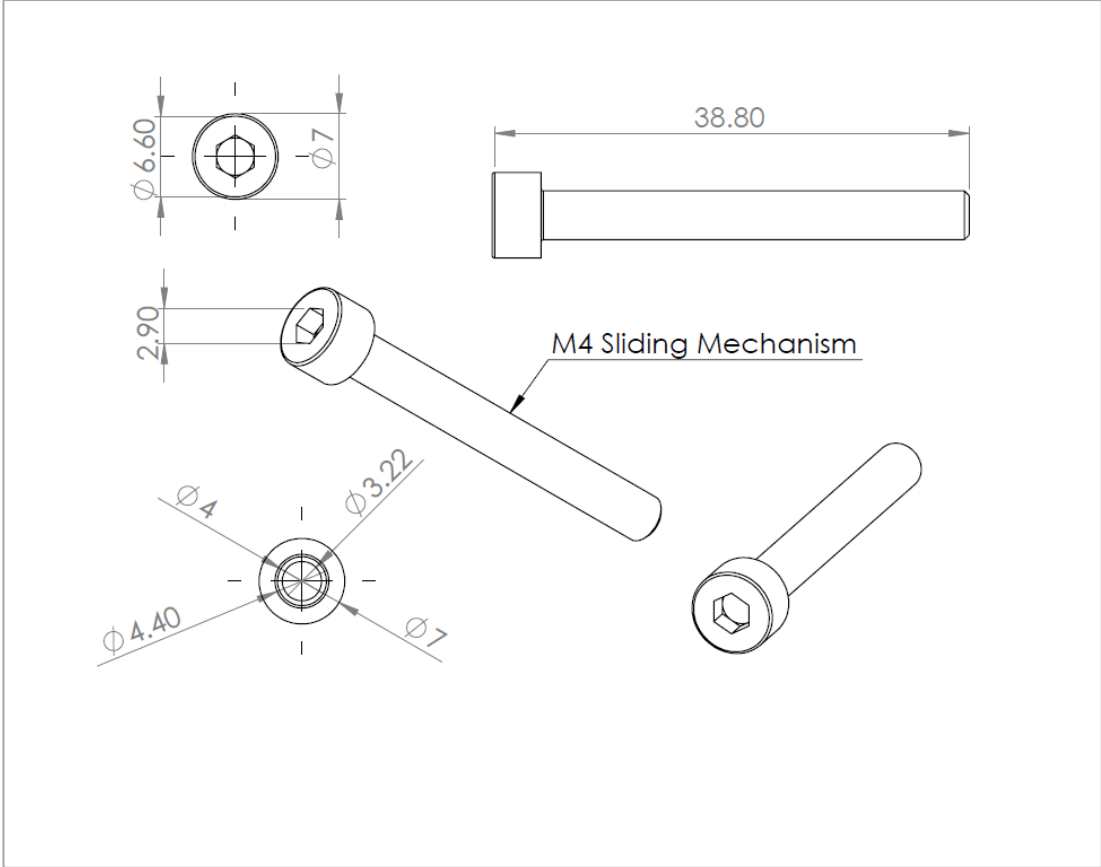
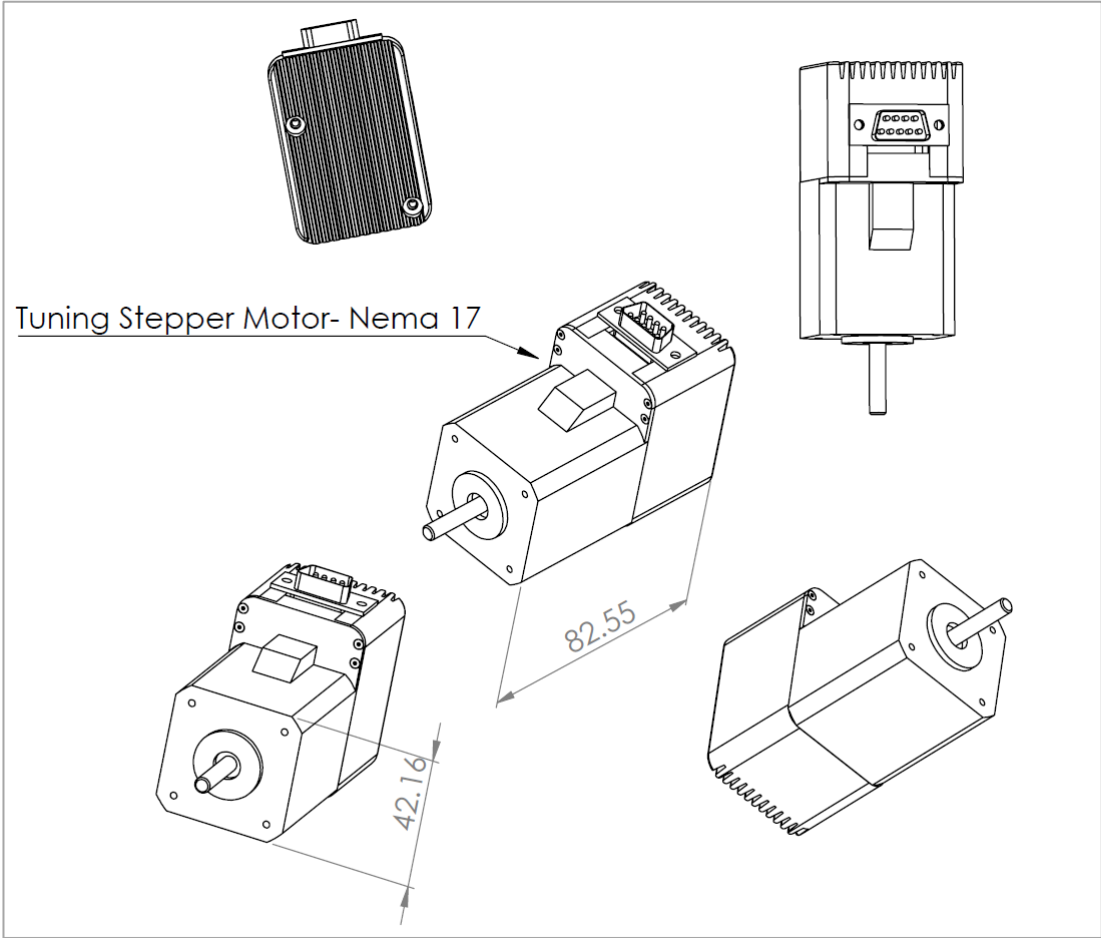
ISSUE	DESCRIPTION	BY	DATE	APP.
01	PRODUCTION	MM	08/06/13	

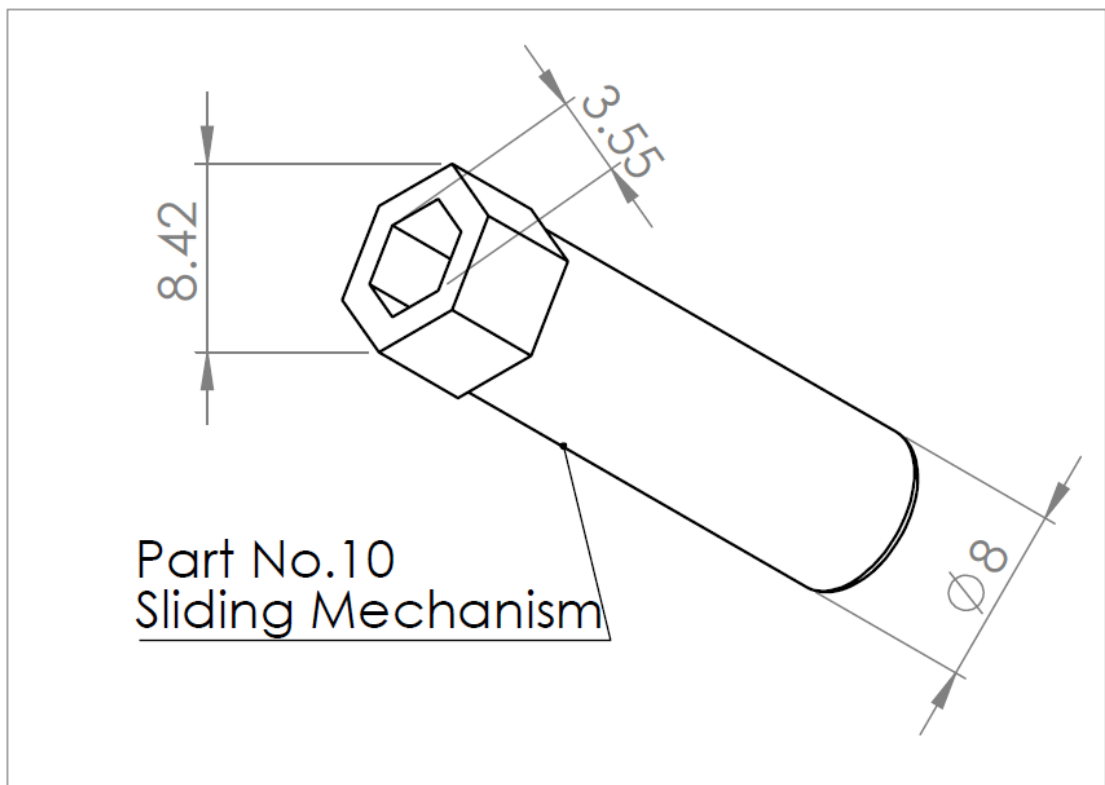
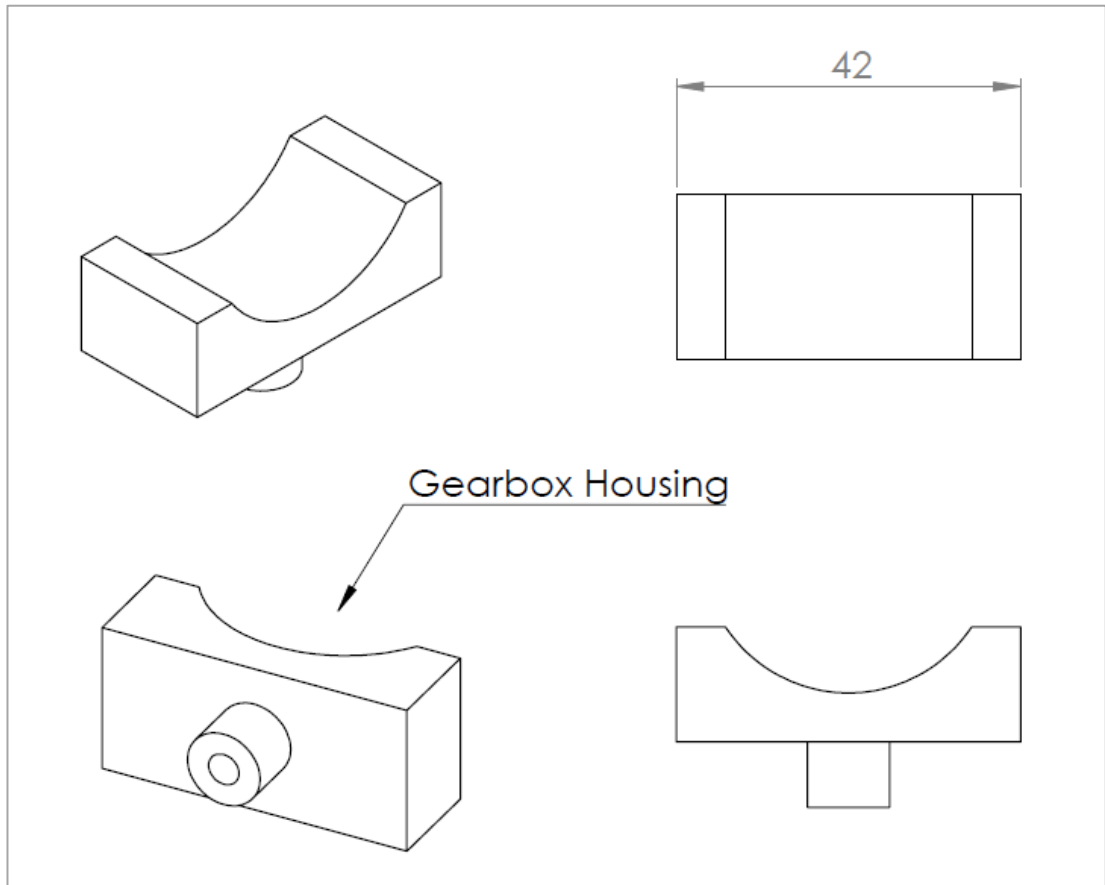


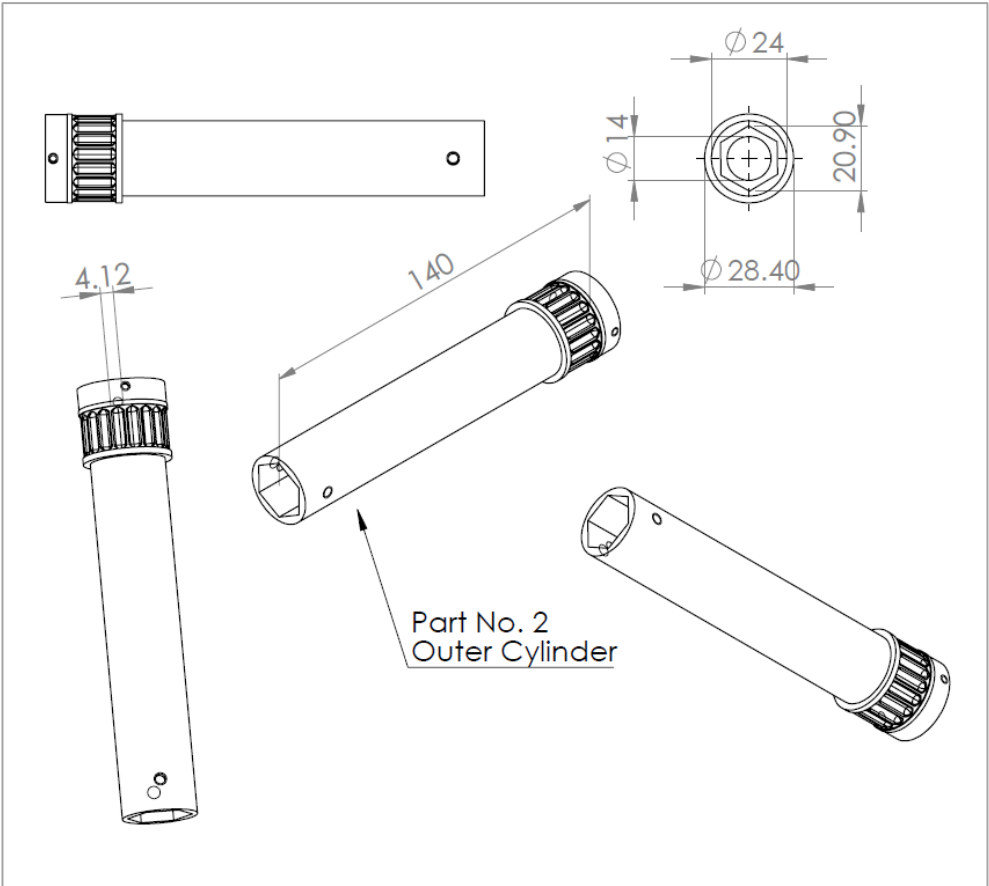
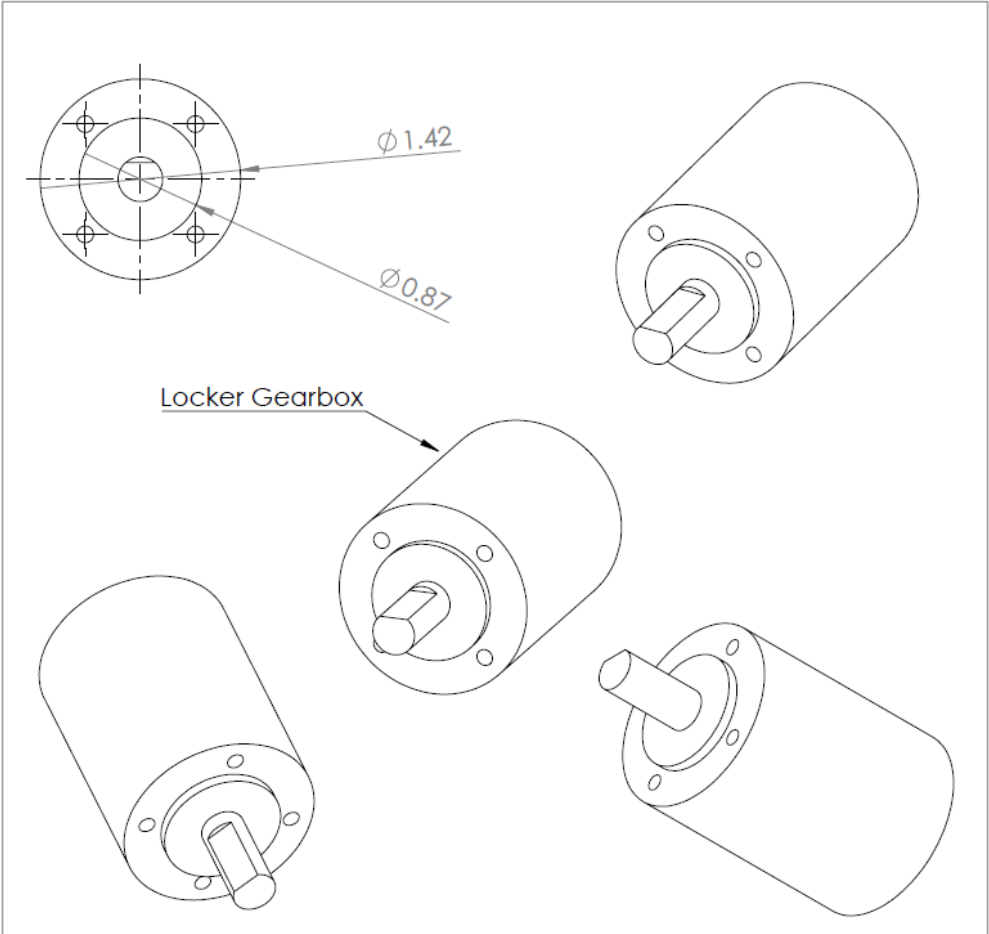
File Name: 2530000

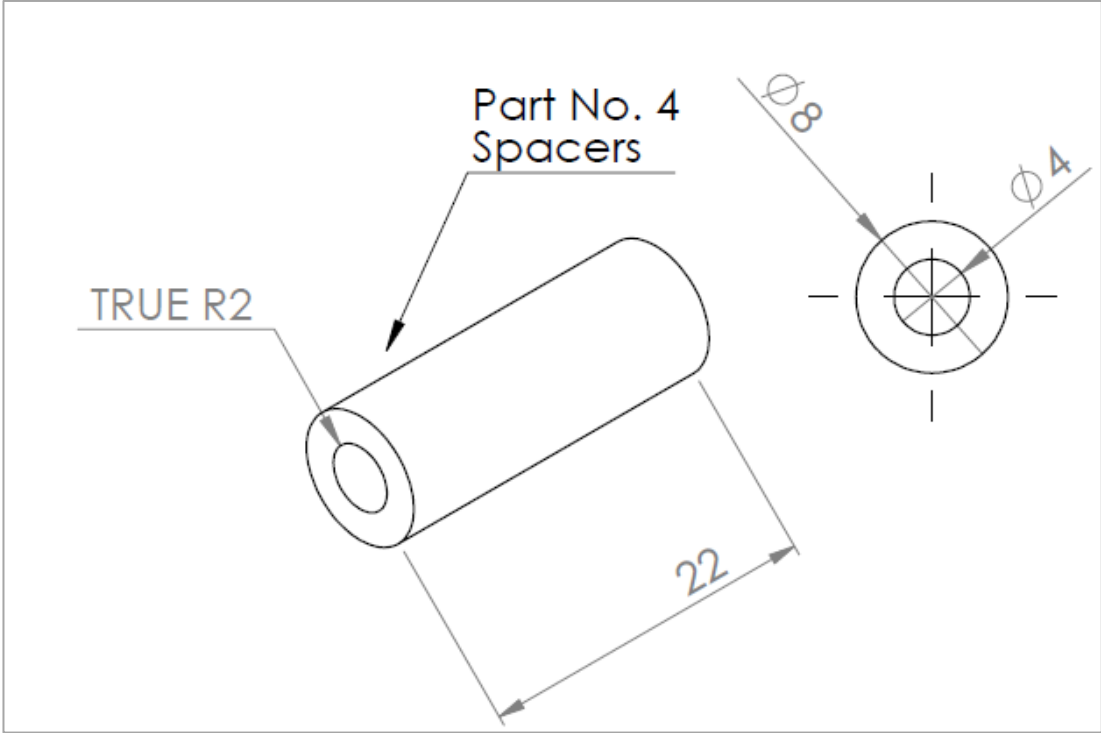
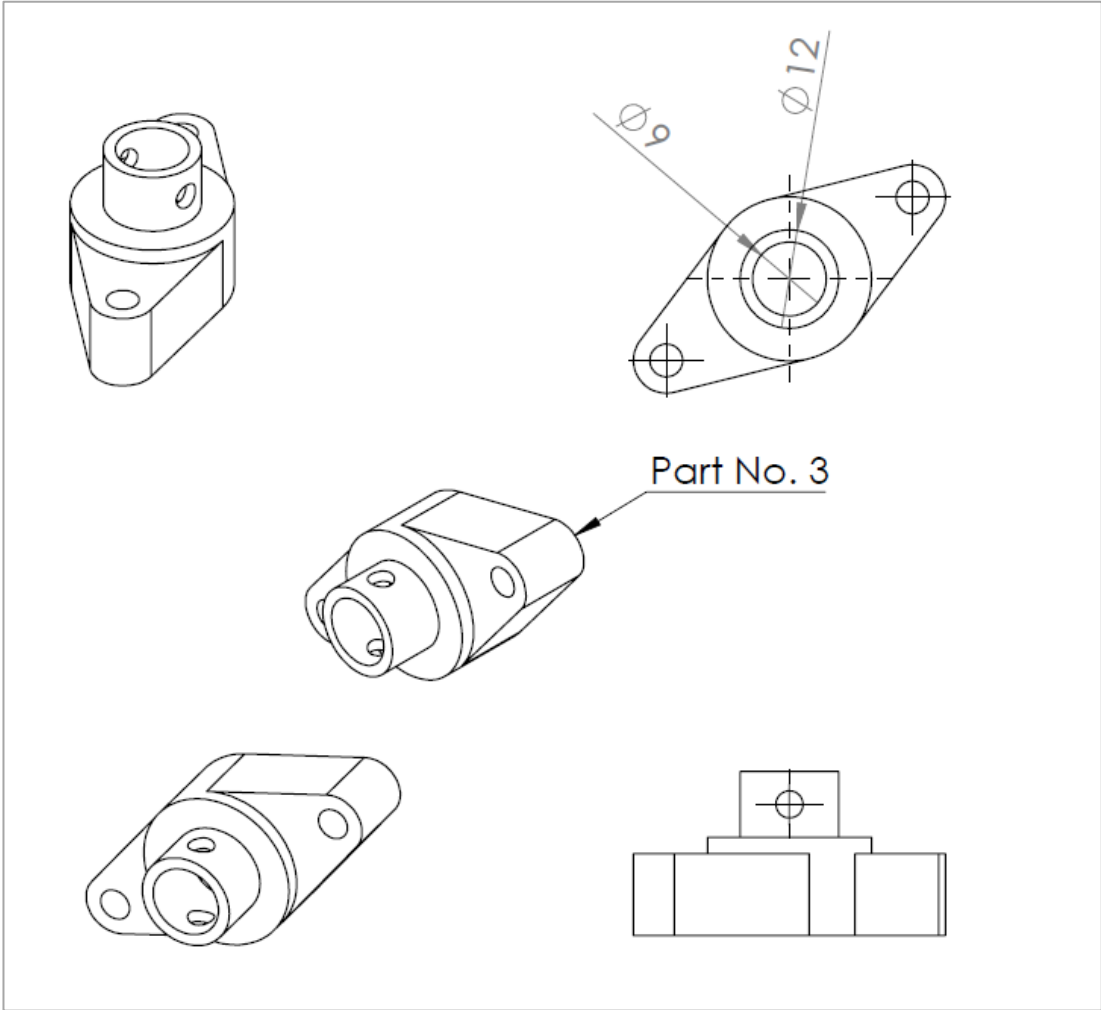


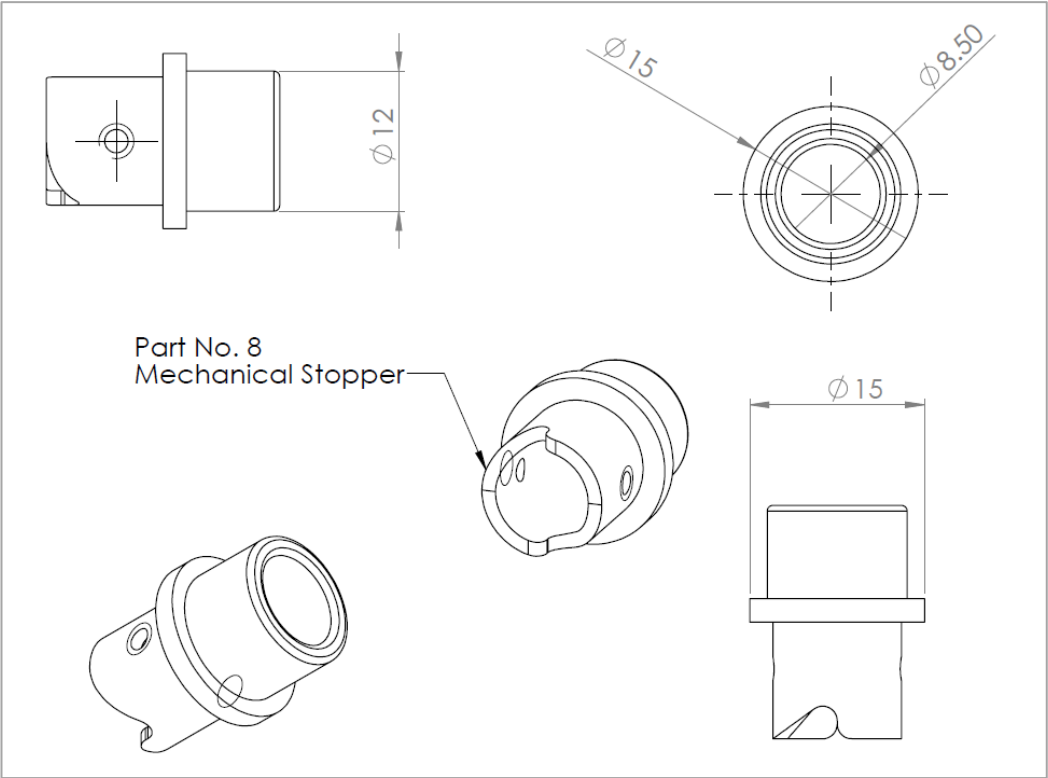
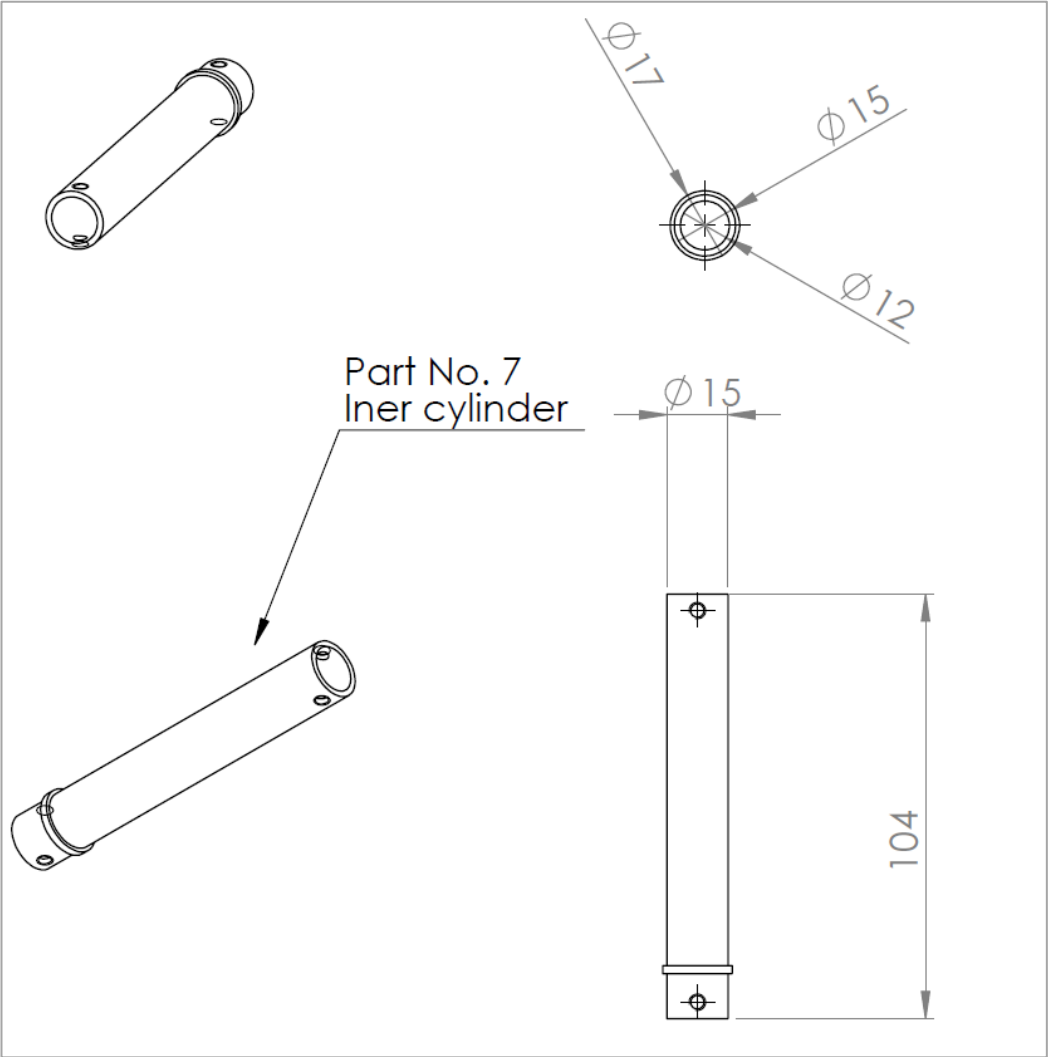












APPENDIX B: CODES AND ALGORITHM

```
////////////////////////////////////Driver Codes////////////////////////////////////

Option Explicit On

Module PerformaxModule
    Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

    Public Declare Function fnPerformaxComGetNumDevices _
        Lib "PerformaxCom" _
        Alias "_fnPerformaxComGetNumDevices@4" _
        (ByRef lpwdNumDevices As Integer) _
    As Integer

    Public Declare Function fnPerformaxComGetProductString _
        Lib "PerformaxCom" _
        Alias "_fnPerformaxComGetProductString@12" _
        (ByVal dwDeviceNum As Integer, _
        ByRef lpvDeviceString As Byte, _
        ByVal dwFlags As Integer) _
    As Integer

    Public Declare Function fnPerformaxComOpen _
        Lib "PerformaxCom" _
        Alias "_fnPerformaxComOpen@8" _
        (ByVal dwDevice As Integer, _
        ByRef cyHandle As Integer) _
    As Integer

    Public Declare Function fnPerformaxComClose _
        Lib "PerformaxCom" _
        Alias "_fnPerformaxComClose@4" _
        (ByVal cyHandle As Integer) _
    As Integer

    Public Declare Function fnPerformaxComSetTimeouts _
        Lib "PerformaxCom" _
        Alias "_fnPerformaxComSetTimeouts@8" _
        (ByVal dwReadTimeout As Integer, _
        ByVal dwWriteTimeout As Integer) _
    As Integer

    Public Declare Function fnPerformaxComSendRecv _
        Lib "PerformaxCom" _
        Alias "_fnPerformaxComSendRecv@20" _
        (ByVal cyHandle As Integer, _
        ByRef lpWBuffer As Byte, _
        ByVal lNumBytesToWrite As Integer, _
        ByVal lNumBytesToRead As Integer, _
```

```

        ByRef lpRBuffer As Byte) _
As Integer

Public Const PERFORMAX_RETURN_SERIAL_NUMBER = &H0
Public Const PERFORMAX_RETURN_DESCRIPTION = &H1
Public Const PERFORMAX_MAX_DEVICE_STRLEN = 256
Public Const INVALID_HANDLE_VALUE = &H1

Public hUSBDevice
Public Status
Public Function PerformaxSendGetReply(ByVal cmdStr As String) As
String
    Dim BytesWriteRequest As Long
    Dim BytesReadRequest As Long
    Dim cc As String
    Dim ci, i, endfound As Integer
    Dim sendStr() As Byte
    Dim ReplyBuffer() As Byte
    Dim replyStr As String

    ReDim sendStr(64)
    ReDim ReplyBuffer(64)

    If (Len(cmdStr) > 64) Then
        PerformaxSendGetReply = -1
        Exit Function
    End If

    For i = 1 To Len(cmdStr)
        cc = Mid(cmdStr, i, 1)
        sendStr(i - 1) = Asc(cc)
    Next i
    For i = Len(cmdStr) To 63
        sendStr(i) = 0
    Next i
    BytesWriteRequest = 64
    BytesReadRequest = 64

    Status = fnPerformaxComSendRecv(hUSBDevice, sendStr(0),
BytesWriteRequest, BytesReadRequest, ReplyBuffer(0))

    replyStr = ""
    endfound = 0
    For i = 1 To 64
        cc = Chr(ReplyBuffer(i - 1))
        ci = Asc(cc)
        If (ci = 0) Then
            endfound = 1
        End If
        If (endfound = 0) Then
            replyStr = replyStr + cc
        End If
    Next i

    If (replyStr = "") Then
        fnPerformaxComClose(hUSBDevice)
        MsgBox("Error communication. Closing application...")
    End
End If

```

```

        If (Len(replyStr) < 0) Then
            MsgBox("No Reply Error. Check communication cable or
power!")
            fnPerformaxComClose(hUSBDevice)
        End If
        PerformaxSendGetReply = replystr
    End Function
    Public Sub MemSet(ByVal Buffer() As Byte, ByVal Value As Byte,
ByVal Amount As Long)
        'this function sets all elements of on array to 0
        Dim i

        For i = 0 To (Amount - 1)
            Buffer(i) = Value
        Next
    End Sub
    Public Function ConvertToVBString(ByVal str)
        Dim NewString As String
        Dim i As Integer

        i = 0
        Do While (i < PERFORMAX_MAX_DEVICE_STRLEN) And (str(i) <> 0)
            NewString = NewString + Chr(str(i))
            i = i + 1
        Loop
        ConvertToVBString = NewString
    End Function
    Public Function BitAND(ByVal tv1 As Long, ByVal tv2 As Long) As
Long
        Dim result, temp1, temp2, v1, v2 As Long
        Dim i As Integer

        result = 0
        v1 = tv1
        v2 = tv2
        For i = 23 To 0 Step -1 '*** 32 bit numbers
            temp1 = v1 \ (2 ^ i)
            v1 = v1 - temp1 * (2 ^ i)
            temp2 = v2 \ (2 ^ i)
            v2 = v2 - temp2 * (2 ^ i)
            If temp1 = 1 And temp2 = 1 Then
                result = result + (2 ^ i)
            End If
        Next i
        BitAND = result
    End Function

    Public Function BitOR(ByVal tv1 As Long, ByVal tv2 As Long) As
Long
        Dim result, temp1, temp2, v1, v2 As Long
        Dim i As Integer

        result = 0
        v1 = tv1
        v2 = tv2
        For i = 23 To 0 Step -1
            temp1 = v1 \ (2 ^ i)
            v1 = v1 - temp1 * (2 ^ i)
            temp2 = v2 \ (2 ^ i)

```

```

        v2 = v2 - temp2 * (2 ^ i)
        If temp1 = 1 Or temp2 = 1 Then
            result = result + (2 ^ i)
        End If
    Next i
    BitOR = result
End Function

Public Function BitShiftLeft(ByVal v1 As Long, ByVal v2 As Long)
As Long
    Dim result As Long

    result = v1 * 2 ^ v2
    BitShiftLeft = result
End Function

Public Function BitShiftRight(ByVal v1 As Long, ByVal v2 As
Long) As Long
    Dim result As Long

    result = v1 \ 2 ^ v2
    BitShiftRight = result
End Function
End Module

////////////////////////////////////Socket Programing////////////////////////////////////

Option Explicit On
Imports System.IO
Imports System.Net.Sockets
Imports System.Threading

Public Class Formo

    Public Message As String
    Public cmdStr As String = ""
    Public replyStr As String = ""
    Dim listener As New TcpListener(500)
    Dim client As TcpClient

    Private Sub _FormClosing() Handles Me.FormClosing
        listener.Stop()
        cmdStr = "EO=0000"
        replyStr = PerformaxSendGetReply(cmdStr)
    End Sub

    Private Sub _Load() Handles MyBase.Load
        cmdStr = "EX=0"
        replyStr = PerformaxSendGetReply(cmdStr)
        cmdStr = "EY=0"
        replyStr = PerformaxSendGetReply(cmdStr)
        Timer1.Start()
        listener.Start()
    End Sub

    Private Sub _Tick() Handles Timer1.Tick

        Dim nStart As Integer
        Dim nLast As Integer

```



```

If Listener.Pending = True Then
    Message = ""
    Client = Listener.AcceptTcpClient()
    Dim Reader As New StreamReader(Client.GetStream())

    While Reader.Peek > -1
        Message &= Convert.ToChar(Reader.Read()).ToString
    End While

    If Message.Contains("</>") Then
        nStart = InStr(Message, "</>") + 4
        nLast = InStr(Message, "<\>")
        Message = Mid(Message, nStart, nLast - nStart)
    End If

    Labell1.Text = Message
    TextBox1.Text = Message
End If
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Timer1.Tick

    Dim cmdStr As String = ""
    Dim replyStr As String = ""
    Dim ret As Integer = 0
    Dim I As Integer = Convert.ToInt32(Me.Message)

    ////////////////////////////////////////////////////////////////////Tuning Codes are written here//////////////////////////////////////////////////////////////////
    ////////////////////////////////////////////////////////////////////Since it is more than 50 pages, it is not included here/////
    ////////////////////////////////////////////////////////////////////Automation codes can be provided upon request//////////////////////////////////////////////////////////////////
    .
    .
    .
    .
    ////////////////////////////////////////////////////////////////////Following codes are simple example of sending
    commands through the form to the driver//////////////////////////////////////////////////////////////////

    If I = 1 Then

        cmdStr = "EO=0001"
        replyStr = PerformaxSendGetReply(cmdStr)
        cmdStr = "JX+"
        replyStr = PerformaxSendGetReply(cmdStr)

        cmdStr = "EX"
        replyStr = PerformaxSendGetReply(cmdStr)
        reply.Text = replyStr

    ElseIf I = 2 Then

        cmdStr = "EO=0010"
        replyStr = PerformaxSendGetReply(cmdStr)
        cmdStr = "JY+"
        replyStr = PerformaxSendGetReply(cmdStr)

    ElseIf I = 3 Then
        cmdStr = "EO=0011"
        replyStr = PerformaxSendGetReply(cmdStr)
        cmdStr = "JX+"

```

```

        replyStr = PerformaxSendGetReply(cmdStr)
        cmdStr = "JY+"
        replyStr = PerformaxSendGetReply(cmdStr)

    Else
        cmdStr = "EO=0000"
        replyStr = PerformaxSendGetReply(cmdStr)

    End If
End Sub

End Class

////////////////////////////////Forms////////////////////////////////

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()>
Partial Class Formo
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
    Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Me.Label1 = New System.Windows.Forms.Label()
        Me.Timer1 = New System.Windows.Forms.Timer(Me.components)
        Me.TextBox1 = New System.Windows.Forms.TextBox()
        Me.reply = New System.Windows.Forms.TextBox()
        Me.SuspendLayout()
        '
        'Label1
        '
        Me.Label1.AutoSize = True
        Me.Label1.Location = New System.Drawing.Point(27, 23)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(39, 13)
        Me.Label1.TabIndex = 0
        Me.Label1.Text = "Label1"
        '
        'Timer1
        '

```

```

Me.Timer1.Interval = 1
'
'TextBox1
'
Me.TextBox1.Location = New System.Drawing.Point(30, 125)
Me.TextBox1.Multiline = True
Me.TextBox1.Name = "TextBox1"
Me.TextBox1.Size = New System.Drawing.Size(217, 92)
Me.TextBox1.TabIndex = 1
'
'reply
'
Me.reply.Location = New System.Drawing.Point(114, 57)
Me.reply.Name = "reply"
Me.reply.Size = New System.Drawing.Size(100, 20)
Me.reply.TabIndex = 2
'
'Formo
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!,
13.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.ClientSize = New System.Drawing.Size(284, 262)
Me.Controls.Add(Me.reply)
Me.Controls.Add(Me.TextBox1)
Me.Controls.Add(Me.Label1)
Me.Name = "Formo"
Me.Text = "Form1"
Me.ResumeLayout(False)
Me.PerformLayout()

End Sub
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents Timer1 As System.Windows.Forms.Timer
Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
Friend WithEvents reply As System.Windows.Forms.TextBox

End Class
////////////////////////////////////

```

//////////simplified flowchart of the tuner only//////////

