

Digitally Programmable Analog Computer

EE 344 - Report (Group **BT02)**

Puneet Nemade, 16D070003

Harsh Deshpande, 16D070011

Viraj Nadkarni - 16D070013

Faculty Advisor: Prof. Mukul Chandorkar

April 2019

Contents

1	Introduction	3
1.1	Project Objectives	3
1.2	Project deliverables	3
1.3	Background and motivation	3
1.4	Block diagram	4
2	System overview	5
2.1	Design alternatives	5
2.2	Design approach	6
2.3	Required Subsystems	7
2.3.1	Components used	7
2.3.2	Circuit subsystems	8
3	Project implementation	9
3.1	Power Management Circuit	9
3.2	Digital subsystem	10
3.3	Analog subsystem	11
4	Performance evaluation	14
4.1	First prototype	14
4.1.1	Design	14
4.1.2	Experiments and results	15
4.2	The final PCB	16
4.2.1	Features and Specifications	16
4.2.2	Testing and Evaluation of the Final Board	20
4.2.3	Experiments and results	21
5	Conclusion	32
5.1	Conclusion and Future Work	32
5.2	References	32
6	Mistakes	32

1 Introduction

1.1 Project Objectives

- To design a digitally programmable analog computer capable of solving non-linear differential equations with maximum 8 variables and 6 forcing functions of the following form :

$$\dot{x}_1 = f_1(x_1, x_2, \dots, x_n)$$

$$\dot{x}_2 = f_2(x_1, x_2, \dots, x_n)$$

⋮

$$\dot{x}_n = f_n(x_1, x_2, \dots, x_n)$$

- The user can program a differential equation digitally whose result will be available as an analog signal
- To solve a 5-variable non-linear differential system for an Induction motor

1.2 Project deliverables

- We will be using the TMS320F28069 micro-controller with 12 inbuilt ADC channels which will be able to be programmed by both JTAG and UART.
- The outputs will be from the externally interfaced "8-channel TLV5610" DAC present on the board and will go to 8 op-amp (TL084) based integrators as input
- Apart from this we have included an analog linear computation unit that can calculate linear part upto 2 variables to provide flexibility. The coefficients of analog part are decided using switches implemented using daisy chaining. This part can either be used or bypassed.

1.3 Background and motivation

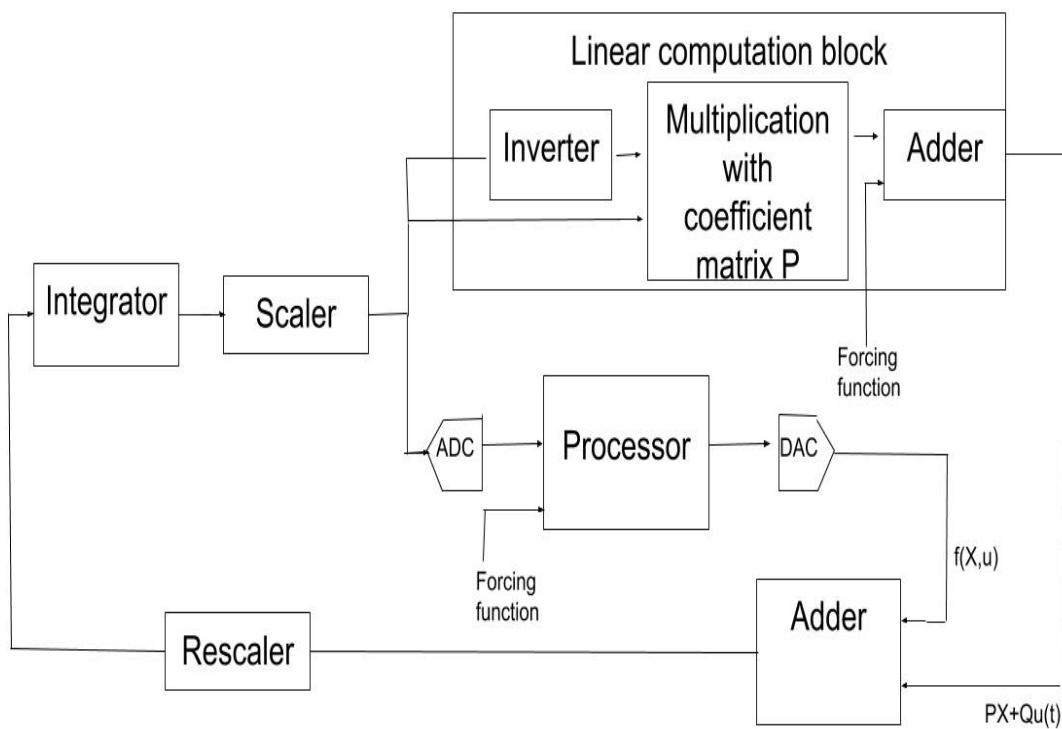
In today's world, computers have made their way into every aspect of our lives. It can be said that in any large or small corporation, be it research or business, bulk of the processing is carried out by computers. This leads to us wanting faster and faster computers.

A lot of real-world applications require the computation and solving of differential equations. Examples include control systems and power electronics. Models of an induction motor with 5 variables, navigation systems for vehicles and ships and many other applications need fast solving of 1st or 2nd order non-linear differential equations to be

solved. The main motivation for this project is to develop a product that would solve this type of problems efficiently and quickly.

1.4 Block diagram

Figure 1: Block diagram of the DPAC



2 System overview

2.1 Design alternatives

The general form of the non-linear equations which we wish to solve is as follows:

$$\dot{x} = f(X, u) + PX1 + Qu1$$

Where P is a 2×2 matrix while Q is a 2×1 matrix . The state variable vector X can have maximum 8 different variables. The variable X1 contains the first two state variables from the vector X. The forcing function vector u(t) can have maximum 4 different variables. The forcing function matrix (1×2) is given by u1(t). f(X,u) is the non-linear function formed using the vectors X and u(t).

The most important part of the analog computer used for solving non-linear differential equations is a multiplier which is used for computing the function f(X,u). We can either use an analog multiplier or multiply signals using microprocessor. As we want to solve a 5 variable system, there can be many multiplications which need to be done. This will require many analog multipliers which is an expensive option and puts on restriction on complexity the non-linear function.

For the multiplication operations, it is better to use a processor. There are two ways we can implement the analog computer while using processor for non-linear part

1. **Analog grid :** In this method, the linear computation is done outside the processor using gain blocks, summers, inverters and switches. The detailed working is explained in the system-level overview. Here the output of integrator is given to processor (non-linear part) and the linear computation block and the final outputs of both are added and given to input of the integrator to complete the loop.
2. **All inside processor approach:** Here we do both linear and non-linear computation inside the processor. So, only the integration operation is done outside the processor.

In the analog grid way, the major advantage is that it is fast as compared to latter since analog part is faster and it saves the time of computing linear part inside the processor. But the other problem in the analog grid is that the delay of the processor and the linear block should match for proper working.

We will be using the initial approach of analog grid.

2.2 Design approach

The method of solving differential equations on hardware using operational amplifiers is fairly well documented. The following are the key steps involved that we followed while designing our circuit:

- Design an integrator block. The resistor-capacitor values used in this block should be such that the prime frequency at which we wish to solve differential equations should have a reasonable output in the magnitude response of the integrator.
- Providing an array of summers for providing forcing functions to our system of differential equations.
- Splitting the further problem into two parts : a nonlinear computation block and a linear computation block.
- In the nonlinear computation block , we planned to have microprocessor which would read in the values of the various state variable and external forcing functions.
- The processor would then compute the values of $f_1, f_2 \dots f_n$ alluded to in the equations above and output the values through a DAC.
- These values computed would then go to the input of the integrator.
- The output of the integrator block would feed back into the input of the nonlinear and linear computation blocks, thus completing the loop.

2.3 Required Subsystems

2.3.1 Components used

The following is a list of the particular components used in building the final circuit:

- **Power circuit :** We used ICs 7805 and 7905 to derive VDD and VCC DC voltages. A low-dropout (LDO) voltage regulator TPS7333 is used to obtain a DC supply for the microcontroller.
- **The microprocessor :** We used the TMS320F28069 processor for managing all nonlinear computations and switch coordination via SPI. We directly used the on-chip ADC's for taking in the values of the state variables.
- **Programming :** The above processor was programmed via a USB-to-UART converter FT232. We also brought out a JTAG header from the processor as an alternative way to program.
- **Analog computations :** We used the IC TL084 (operational amplifier) for computations like integration, summing and scaling in the analog circuit.
- **DAC :** As a DAC, we used the IC TLV5610, a 12-bit DAC with 8 channels.
- **Switching :** We used the IC MAX395 as SPI-controllable switches in the linear computation part of our circuit.

2.3.2 Circuit subsystems

Our final circuit was divided into the following subsystems:

1. Power supply

In our design, the microprocessor has a voltage level of 3.3V. We have used analog switches and op-amps which have a bipolar supply of +5V and -5V. We derive our power supply from 12V and -12V DC respectively. Thus, we require a circuit to supply 3.3V, +5V and -5V from the ± 12 V DC voltage.

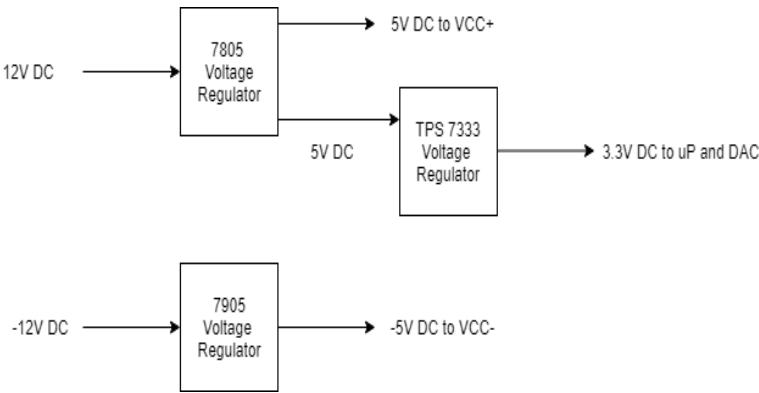


Figure 2: Power Supply Block Diagram

2. Digital subsystem

The processor , DAC and the all the bootloading circuit surrounding it formed the digital subsystem of our circuit.The schematic of this circuit was designed by taking 28069 launchpad schematic as a reference.

3. The scaling circuit

This part of the analog circuit did the function of mapping the signals in the analog domain (with values ranging from -5 V to +5 V) to the ADC readable domain (0 to 3.3 V).For this we used opamps for scaling and shifting. The detailed circuitry is given in the section 3.2.

4. The rescaling circuit

This part of the analog circuit did the function of mapping the signals in the DAC domain (with values ranging from 0 V to +3.3 V) to the opamp's analog domain (-5 to +5V).For this we used opamps for scaling and shifting. The detailed circuitry is given in the section 3.2.

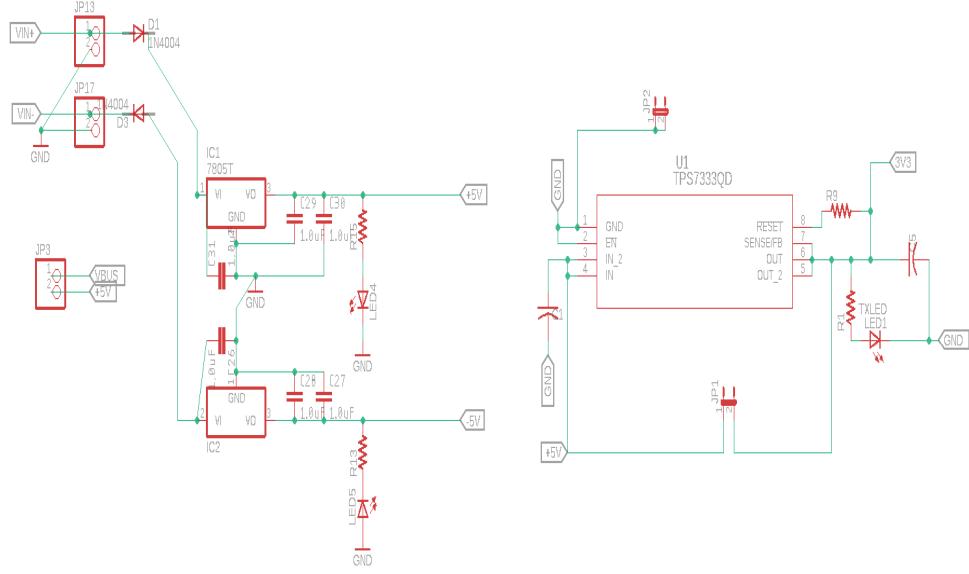


Figure 3: Power Circuit Schematic

5. Linear computation circuit

This part of the circuit linearly combined the analog values of various state variables (x_1, x_2, \dots, x_n) and produced a desired linear combination of them as an output. We used switches for getting some amount of choice over the coefficients in the linear combination. The circuit for this part is given in section 3.2.

3 Project implementation

3.1 Power Management Circuit

The input to the power supply circuit is DC voltage of 12V and -12V. ICs 7805 and 7905 have been used to derive +5V and -5V DC voltages. The op-amps and analog switches are powered by +5V and -5V. A low-dropout (LDO) voltage regulator TPS7333 is used to obtain a 3.3V supply for the microprocessor.

Decoupling capacitors of 0.1uF are placed at power supply pins of every IC. Larger decoupling capacitors of 10uF and 1000uF are placed around the 7805 and 7905 ICs. The operating range of 7805 and 7905 requires their input voltage to be at least 8V. Reverse-biased diodes are added to prevent any back-voltage.

3.2 Digital subsystem

- For programming of the 80-pin package TMS320F28069 processor, we have provided two available options - namely through USB or through JTAG.
- The FT232RL chip is an interface between the USB port on the PC and the UART serial port on the processor.
- The boot mode selection pins (that lets one choose in what way one wants to program the processor) have been brought on header pins. We are planning on performing the SCI boot.

- **ADC**

We are using the inbuilt 12-Bit ADC for taking the inputs from the integrator to compute the non-linear part in the differential equation. There are total 12 channels available in the ADC. So, 8 out of these channels can be used to input the state variables coming from the integrator through scaler circuit. Other 4 channels can be used to input forcing functions.

- **DAC**

This processor has two SPI modules. One is used to communicate with DAC(TLV5610) while the other is used to program MAX395 switches via SPI daisy chaining. The 8 processed outputs from the ADCs are given to the 8 channels of the DAC which goes to input of integrator through rescaler circuit thus completing the hardware-in-the-loop configuration

- **MAX395**

The switches used are MAX395 ICs, each IC having 8 analog SPST switches. The switches are programmed by sending 8 bits (1 Byte) using SPI. The MAX395 switches are controlled in cascade by SPI communication using daisy chaining. The input data appearing at the DIN port of the register is clocked in at rising edge of SCLK and clocked out at the DOUT at the falling edge of SCLK . The data at DOUT is simply DIN delayed by 8 clock cycles. As can be seen in the figure below, the DOUT pin of a switch may be connected to the DIN of another to transfer the register data. The daisy chained switches and the microprocessor communicate with each other using SPI communication.

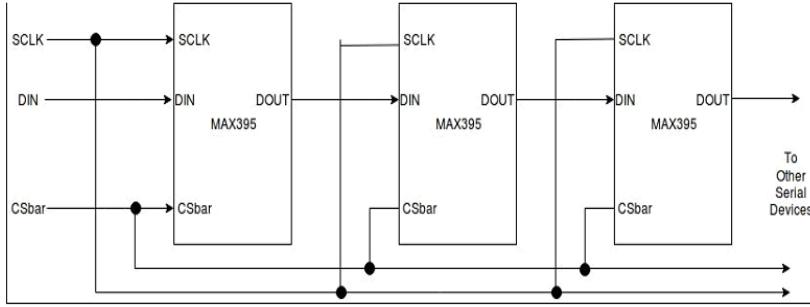


Figure 4: Block Diagram for daisy chaining of switches

3.3 Analog subsystem

As discussed earlier this block evaluated the linear part of the RHS of the general equation we intend to solve. This block has three main stages - inverter, adder and RC-integrator. All three of these are implemented using op-amps and passive elements. For our 2×2 system, the first section has 2 analog inverters to invert the signals x_1 and x_2 . This is followed by the adder stage. We have 5 adders here. The first adder adds x_1, x_2 and u_1 (forcing function) scaled by different coefficients, and so on.

- **Integrator**

The gain of the integrator depends on the frequency of the signals used. When a signal $A\sin(2\pi ft)$ is given, we get $\frac{-A*2\pi f \cos(2\pi ft)}{RC}$. In order to have a reasonable gain for a large range of frequencies, we have made the possibility of changing the resistance at the input of the integrator using a MAX395 switch. This is shown in figure 5.

- The coefficients for scaling of input are chosen by varying the input resistors of each signal which is done using a network of switches. The circuit is made for 2 variables. This is shown in figure 6.
- We have 8 possible values $\pm(0,1,2,3)$ for each coefficient by having 2 resistors, their parallel combination and an open circuit and same for negative input obtained using inverter.
- Since we need 3 switches and have only one SPI interface available, we have used daisy chaining of switches.
- The circuit for the scaler alluded to in the preceding sections is shown in figure 7.

- The circuit for the rescaler alluded to in the preceding sections is shown in figure 8.

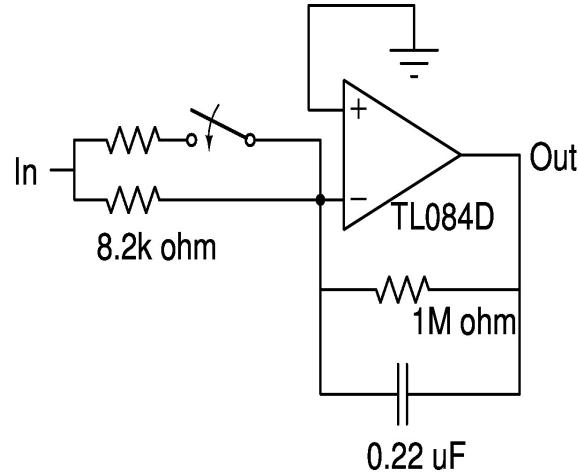
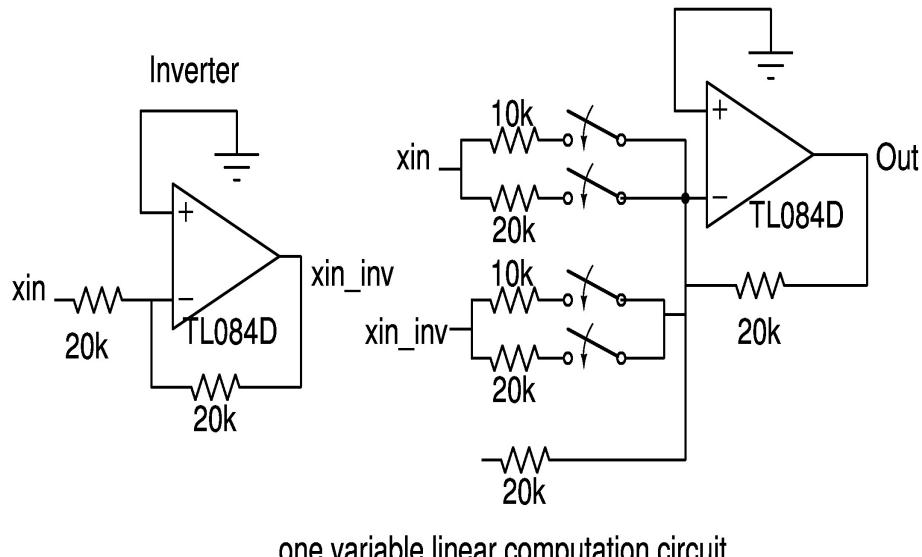


Figure 5: The integrator



one variable linear computation circuit

Figure 6: The linear computation block

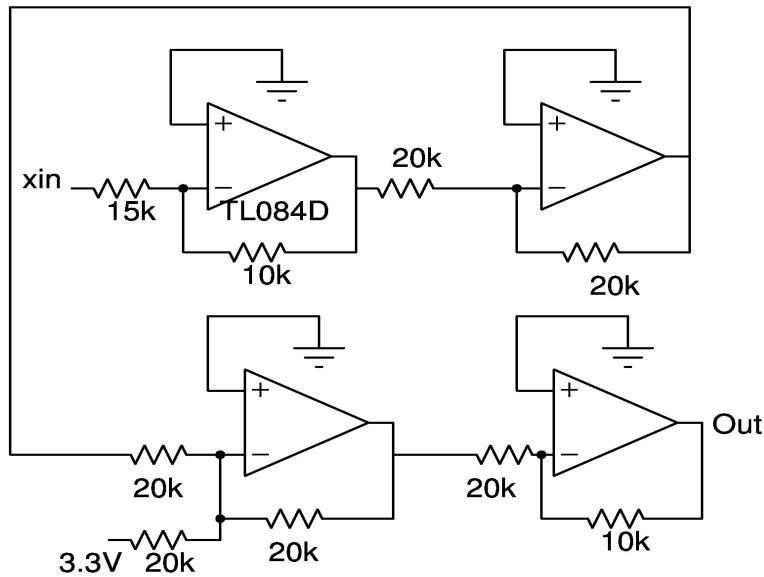


Figure 7: The scaler block

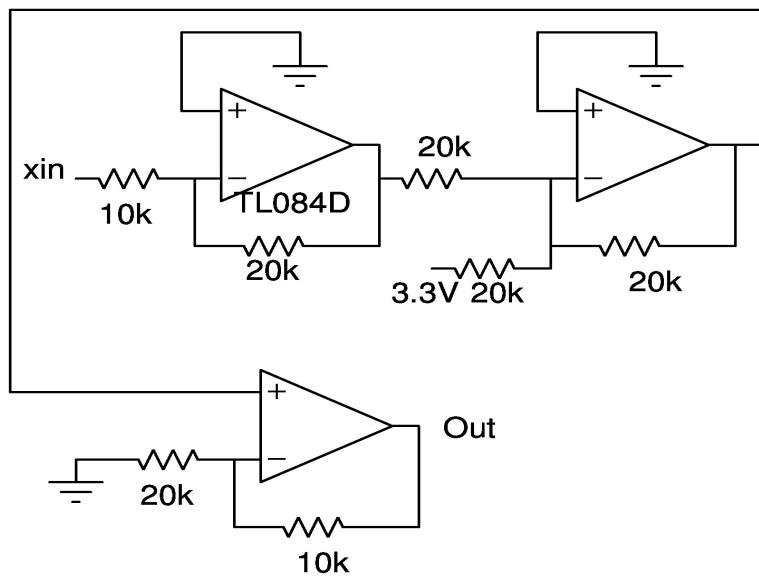


Figure 8: The rescaler block

4 Performance evaluation

4.1 First prototype

Given the complexity of the desired system and the large lead times involved in the planning, layout, design and fabrication of the main board, a preliminary prototype was designed. The specifications of are as following :

- Can solve non-linear (with launchpad) dynamical systems of upto second order
 - Consists of break-out board for DAC and MAX395 switch for individual testing

4.1.1 Design

The motivation behind creating the prototype was to understand the behaviour and functioning of a hardware-in-loop configuration using the launchpad and printed circuit board before the final board arrives. The prototype is powered by stable DC voltage sources, and the power management circuit is omitted. The system consists of two state variables x_1 , x_2 .

The board layout of prototype which comprises of a dual-layer PCB, fabricated at the PCB Printing Laboratory within the Department of Electrical Engineering at IIT Bombay. The op-amps used were operated at a V_{cc} of $\pm 12V$.

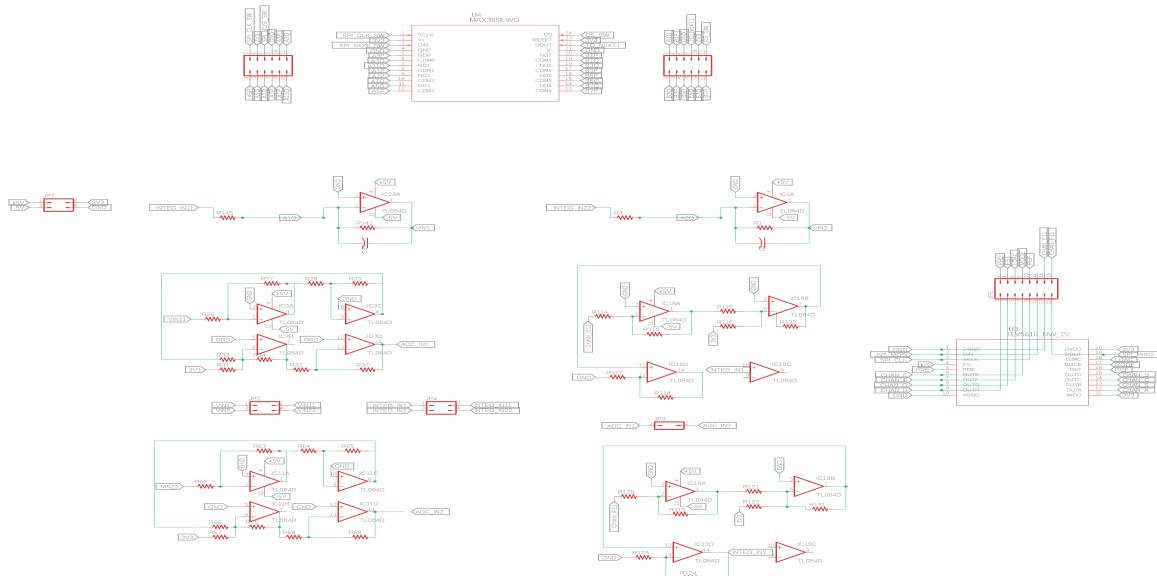


Figure 9: Schematic of the prototype board

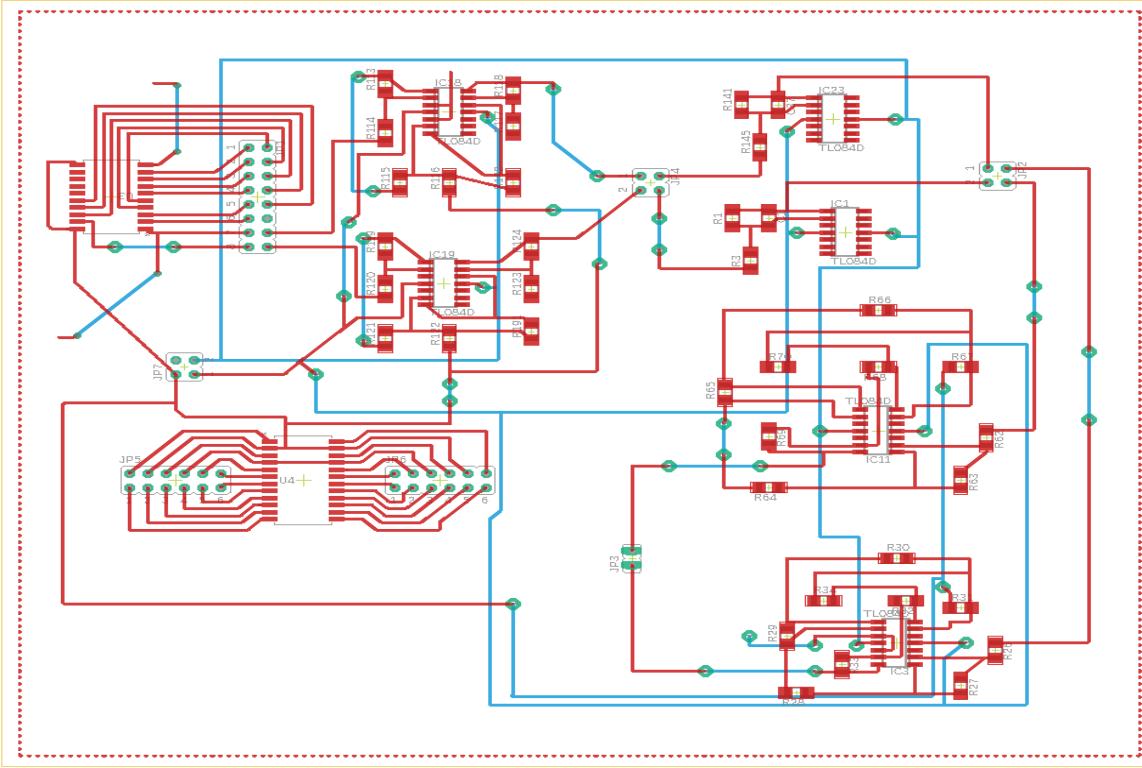


Figure 10: Layout of the prototype board

4.1.2 Experiments and results

We tested a single order equation of type $\frac{dy}{dt} = -y + \sin(t)$ in closed loop with processor(launchpad) and the prototype analog circuit. Since there was a problem of clearance in this board, we couldn't test the switch and DAC on this board. So, we designed two other small boards where one had breakout board for DAC while other had reakout board for two switches (testing daisy chaining).

4.2 The final PCB

4.2.1 Features and Specifications

The final board DPACv2.0 comprises of a full-fledged implementation of a digitally programmable non-linear dynamical system solver, with capability to solve systems upto order 8. It consists of an onboard microprocessor - the Texas Instruments TMS320F28069- which is programmed via USB. The board also supports communication via JTAG. The board features its own power management circuitry, and can be given $\pm 12V$ DC voltage. The board was fabricated at PCB Power Market, Ahmedabad. The pictures of the final schematic, final layout and final soldered PCB are given in in the Figure 11,12 and 13 respectively.



Figure 11: The schematic of PCB

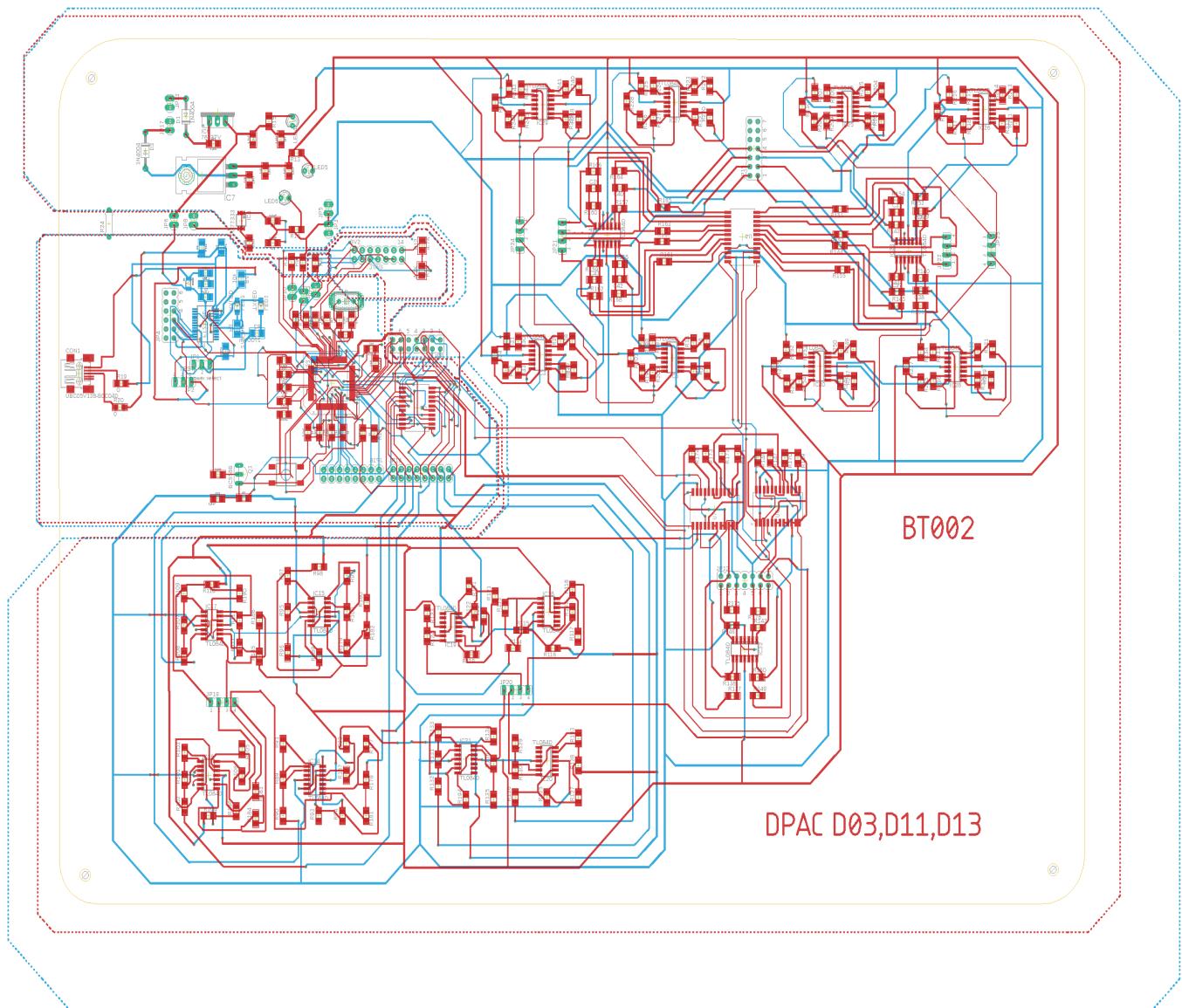


Figure 12: The layout of PCB

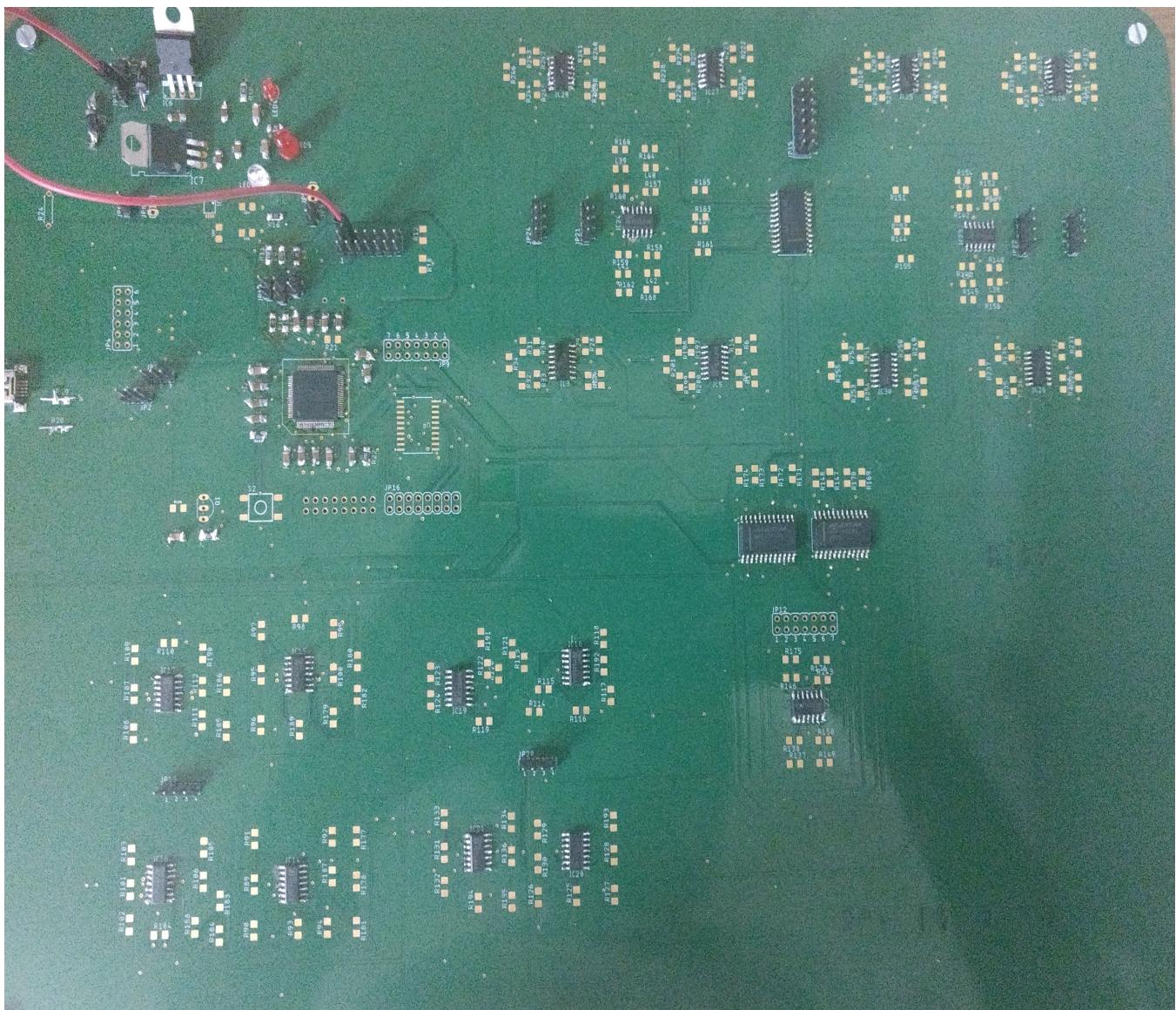


Figure 13: The soldered final PCB

4.2.2 Testing and Evaluation of the Final Board

All the different stages of the board were individually tested. This allowed us to isolate the cause of any discrepancy that could be present on the board and hence take the necessary correcting actions. The steps included:

1. The first step was to solder and test the power circuitry. The 7805 and 7905 LDO were soldered , tested and verified. The 3.3V LDO packaging that we ordered didn't arrive. To tackle this, we made a breakout board using the available packaging of the IC and its outputs were then interfaced to the original board.
2. The USB to UART (FT232) IC was soldered first alongwith the microprocessor and its remaining passive components. Due to some error , we were unable to boot the on-board micro-processor.
3. This was followed by soldering of the integrator and rescaler circuitry. All 8 channels were soldered and their working was verified by integrating waveforms like a sine-wave or square-wave
4. This was followed by the soldering of the Scaler circuitry and its working was tested and verified as above
5. The power circuit and the analog computation part of the circuit was functioning well ,which is why we decided to interface the board with the 28069 launchpad and external DAC for which we had breakout board made.

4.2.3 Experiments and results

Once the working of all the individual components was verified , we proceeded to solve a set of differential equations on the DPACv2.0.

We had tuned our integrator circuit to give a gain of 1 around 160 Hz. This was done as we required a gain of 1 around our range of operation which was a few hundred Hz to a few KHz:

$$R = 10k$$

$$C = 100nF$$

$$1/(2\pi RC) = 160Hz$$

The set of equations that were solved were:

1. Linear first order equation

The general form of equation we implemented is as follows:

$$\dot{x} = -2\pi f_0 x + u(t)$$

The first case is :

$$\dot{x} = -1000x + \sin(1000t).$$

Expected stable output :

$$0.7 * \cos(1000t + 45^\circ)$$

Observed stable output

$$0.76 * \cos(1000t + 44.64^\circ)$$

The transient plot for the above solved equation is given in figure 15

In figure 14, the yellow coloured one is the output.

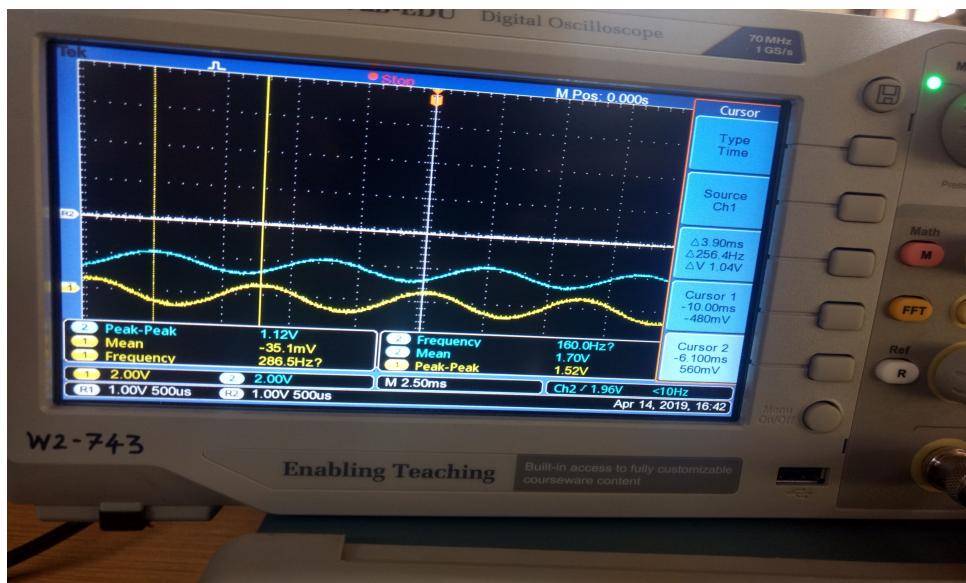


Figure 14: Stable Output for First Order Linear DIfferential Equation

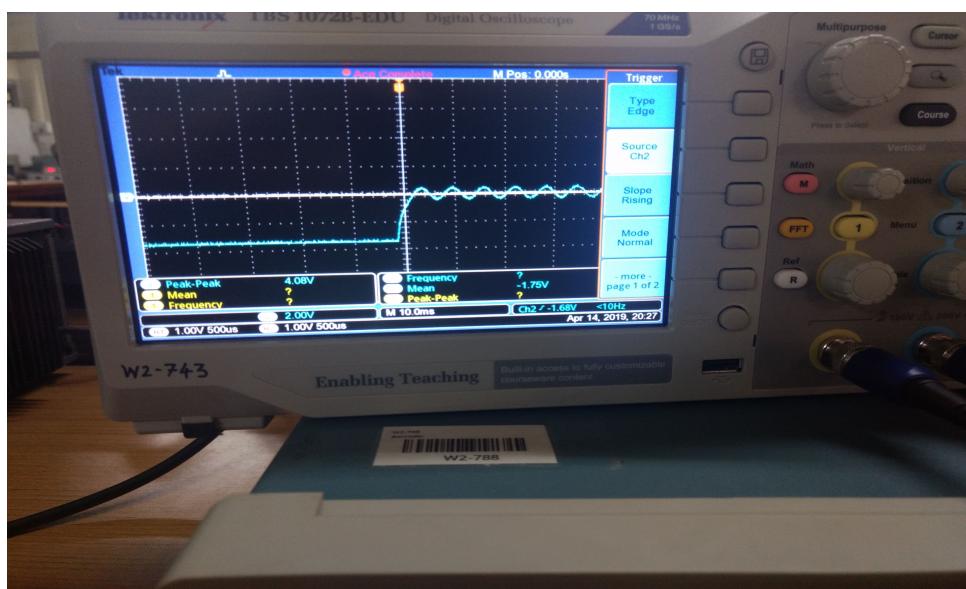


Figure 15: Transient obtained for First Order Linear DIfferential Equation

The second case is :

$$\dot{x} = -1000x + 0.6 * \sin(628t).$$

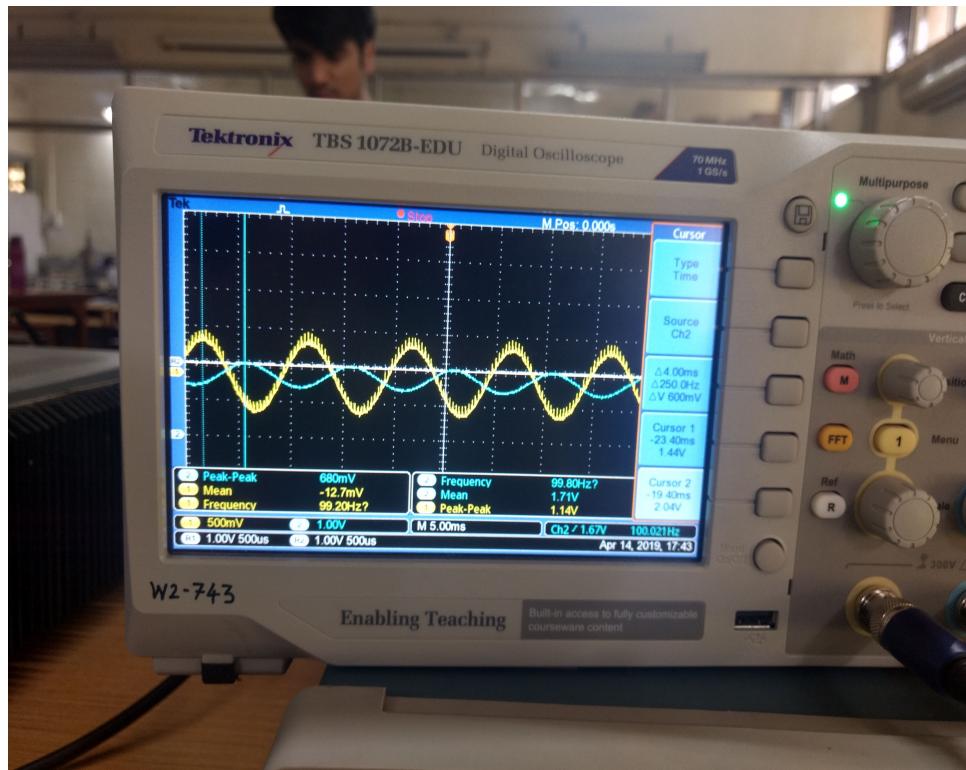
Expected stable output :

$$0.5064 * \cos(628t + 58^\circ)$$

Observed stable output

$$0.57 * \cos(628t + 54.4^\circ)$$

In the figure below, the yellow coloured one is the output.



2. Linear second order equation

The equations for simulation of sinusoidal excitation of a series R-L-C circuit are as follows :

$$u(t) = i_L R + L \frac{di_L}{dt}$$

$$i = C \frac{dV_c}{dt}$$

$$u(t) = f(t) - V_c$$

where :

$$R = 2, L = 1, C = 1$$

$f(t)$ is the forcing function used

The following set of equations were then converted to the corresponding state space equations as follows:

$$X = \begin{bmatrix} i & v \end{bmatrix}^T$$

$$\dot{X} = 2\pi f_0(AX + u(t))$$

$$A = \begin{bmatrix} -2 & -1 \\ 14 & 0 \end{bmatrix}$$

$$x_1 = V$$

$$x_2 = i$$

The equations programmed into the processor were as follows :

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = 0.45 \sin(1000t) - 2x_2 - x_1$$

Expected stable output :

$$x_1 = 0.45 * \sin(1000t - 90^\circ)$$

$$x_2 = 0.45 * \sin(1000t)$$

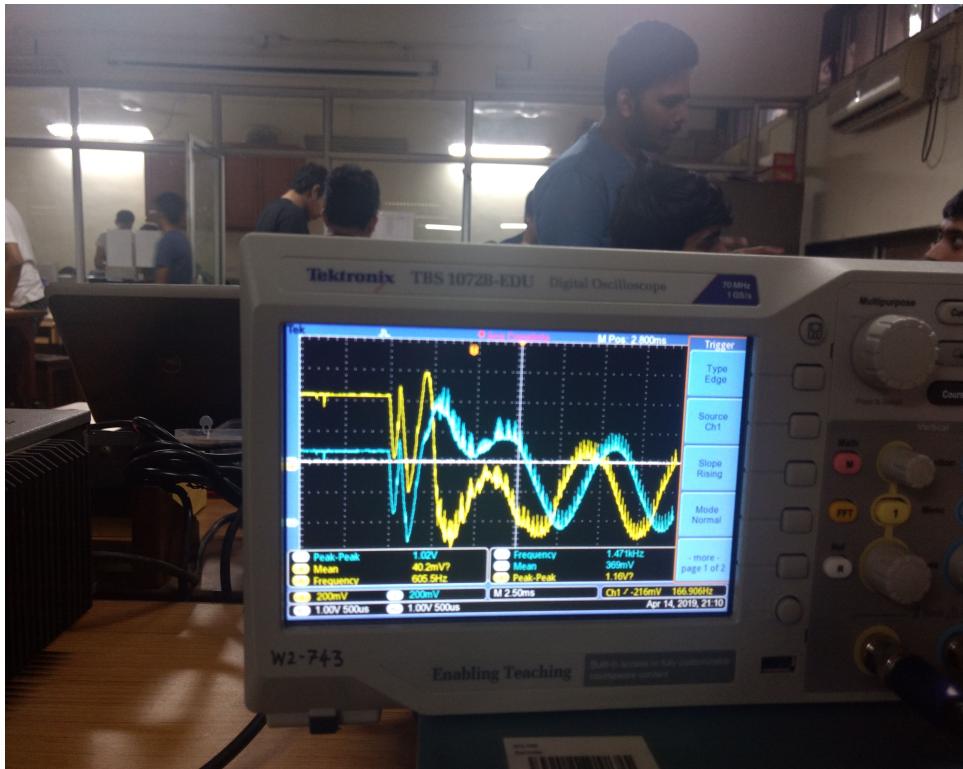
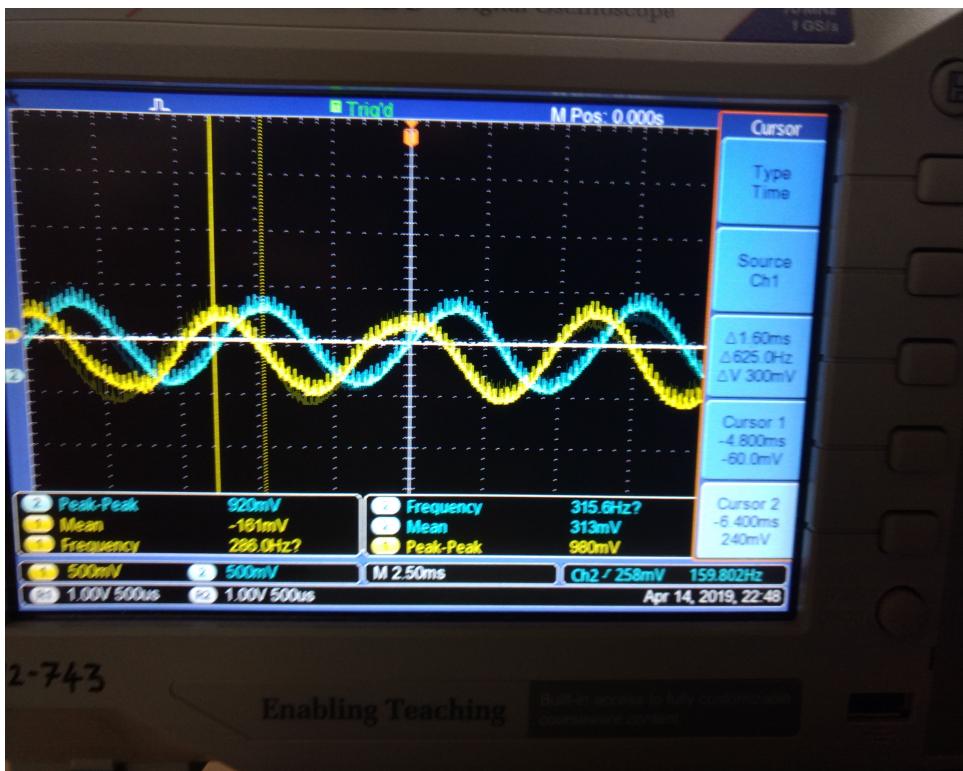
Observed stable output

$$x_1 = 0.49 * \sin(1000t - 92.16^\circ)$$

$$x_2 = 0.46 * \sin(1000t)$$

The transient plot for the above solved equation is given in figure 15

The yellow output is current while the blue output is voltage across capacitor



3. Nonlinear second order equation

The next step would be to test and simulate a second order non-linear differential equation that would give a stable and bounded output. For this, we chose the Van der pol oscillator equation. The equation is of the form:

$$\ddot{X}(t) - \mu(1 - X(t)^2)\dot{X}(t) + X(t) + f(t) = 0$$

where:

μ is a scalar parameter that indicating the nonlinearity and the strength of the damping

$f(t)$ is just any exciting function

The state space representation of the equation is:

$$\begin{aligned} X &= \begin{bmatrix} X_1 & X_2 \end{bmatrix}^T \\ \dot{X}_2 &= X_1 \\ \dot{X}_1 &= \mu(1 - X_2^2)X_1 - X_2 - f(t) \end{aligned}$$

MATLAB simulations were carried out for values of μ equal to 8 and 16. The plots are attached below:

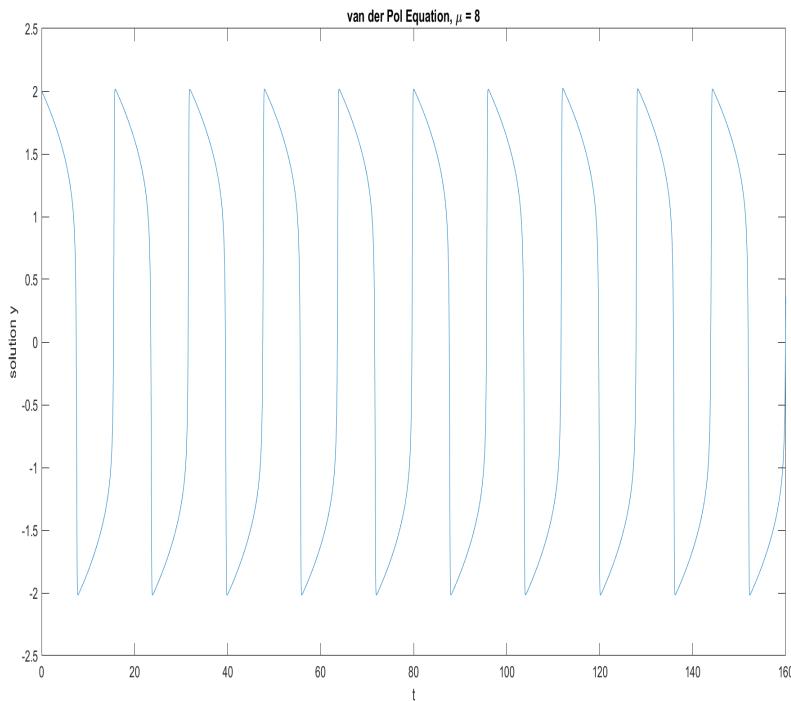


Figure 16: Van der Pol simulations for $\mu = 8$

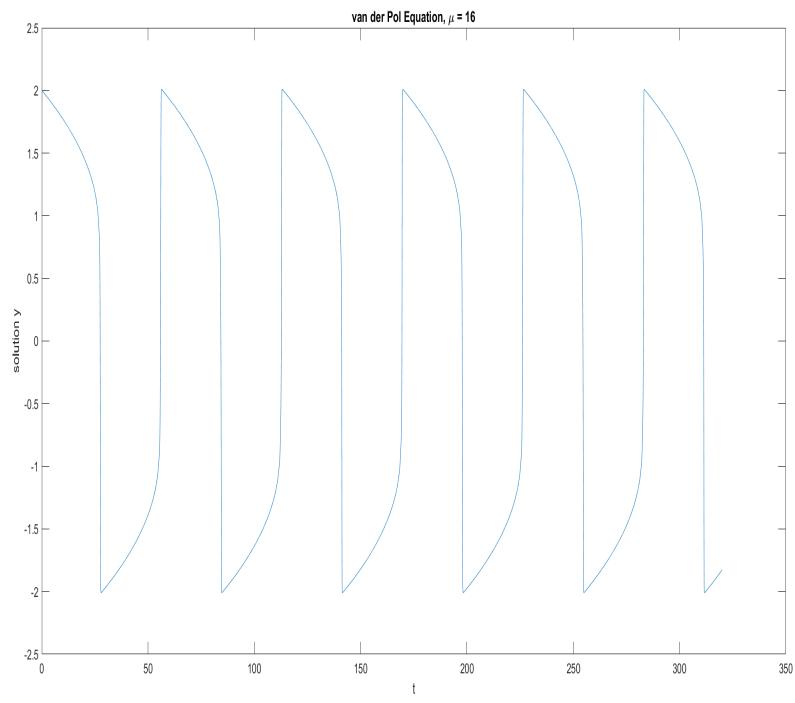


Figure 17: Van der Pol simulations for $\mu = 16$

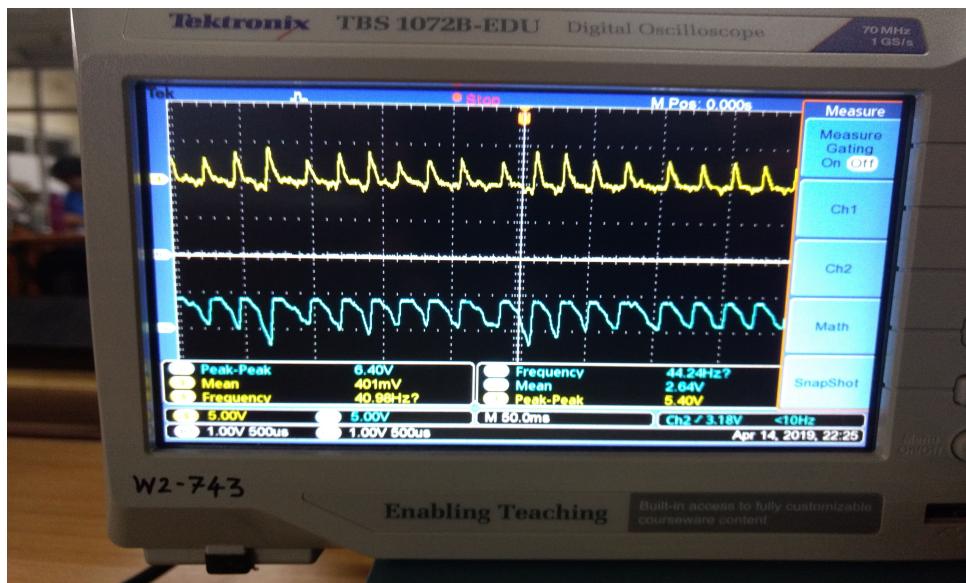


Figure 18: Van der Pol results for $\mu = 8$

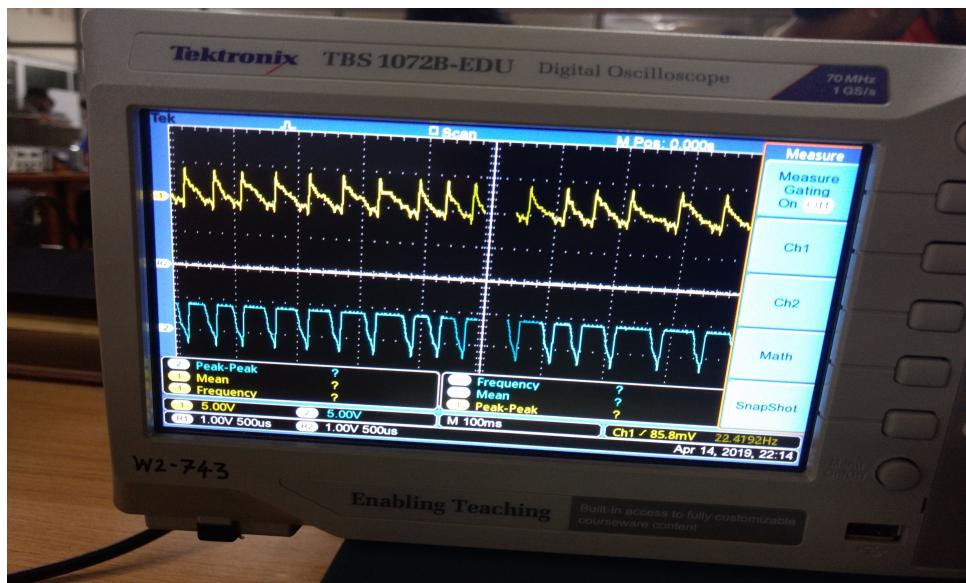


Figure 19: Van der Pol results for $\mu = 16$

4. Nonlinear fifth order equation

After solving a second order non-linear differential equation, we tried to solve the fifth order induction motor equations which are as follows :

A Model for an Induction Motor

State equations	Derived quantities
$\frac{d}{dt}\lambda_{ds} = v_{ds} - r_s i_{ds}$	$i_{ds} = \frac{1}{l_e}(\frac{l_r}{l_m}\lambda_{ds} - \lambda_{dr})$
$\frac{d}{dt}\lambda_{qs} = v_{qs} - r_s i_{qs}$	$i_{qs} = \frac{1}{l_e}(\frac{l_r}{l_m}\lambda_{qs} - \lambda_{qr})$
$\frac{d}{dt}\lambda_{dr} = -\omega_r \lambda_{qr} - r_r i_{dr}$	$i_{dr} = \frac{1}{l_m}(\lambda_{ds} - l_{ls}i_{ds}) - i_{ds}$
$\frac{d}{dt}\lambda_{qr} = +\omega_r \lambda_{dr} - r_r i_{qr}$	$i_{qr} = \frac{1}{l_m}(\lambda_{qs} - l_{ls}i_{qs}) - i_{qs}$
$\frac{d}{dt}\omega_{rm} = \frac{T_{em} - T_L}{J}$	$T_{em} = \frac{3}{4}Pl_m(i_{qs}i_{dr} - i_{ds}i_{qr})$
	$\omega_r = \frac{P}{2}\omega_{rm}$

Figure 20: Equations for induction motor

The values of the constants used are given below :

$$\begin{aligned}
 l_{ls} &= 0.0032; \\
 l_{lr} &= 0.0032; \\
 l_m &= 0.0955; \\
 l_s &= l_{ls} + l_m = 0.0987; \\
 l_r &= l_{lr} + l_m = 0.0987; \\
 l_e &= \frac{l_s * l_r}{l_m} - l_m = 0.0065; \\
 r_s &= 0.01; \\
 rr &= 0.01; \\
 T_l &= 1; \\
 J &= 1; \\
 P &= 4
 \end{aligned}$$

These values were taken from the reference document provided by sir except for the constants r_s , rr and T_l . The value of load torque (T_l) was reduced since the values given in the document were for higher rated voltage. The rotor and stator resistances were reduced since the OP-AMPS were saturating if their values were

kept high (0.2-0.3 ohm).

The input voltages direct-axis stator voltage (v_{ds}) and quadrature-axis stator voltage (v_{qs}) are as follows :

$$v_{ds} = \sin(1000t) \text{ V}$$

$$v_{qs} = \sin(1000t + 90^\circ) \text{ V}$$

The final output signal corresponding to λ_{ds} and λ_{qs} are given in figure 21. The final output signal corresponding to λ_{dr} and λ_{qr} are given in figure 22. The transient plot of ω_{rm} and λ_{ds} is given in figure 23

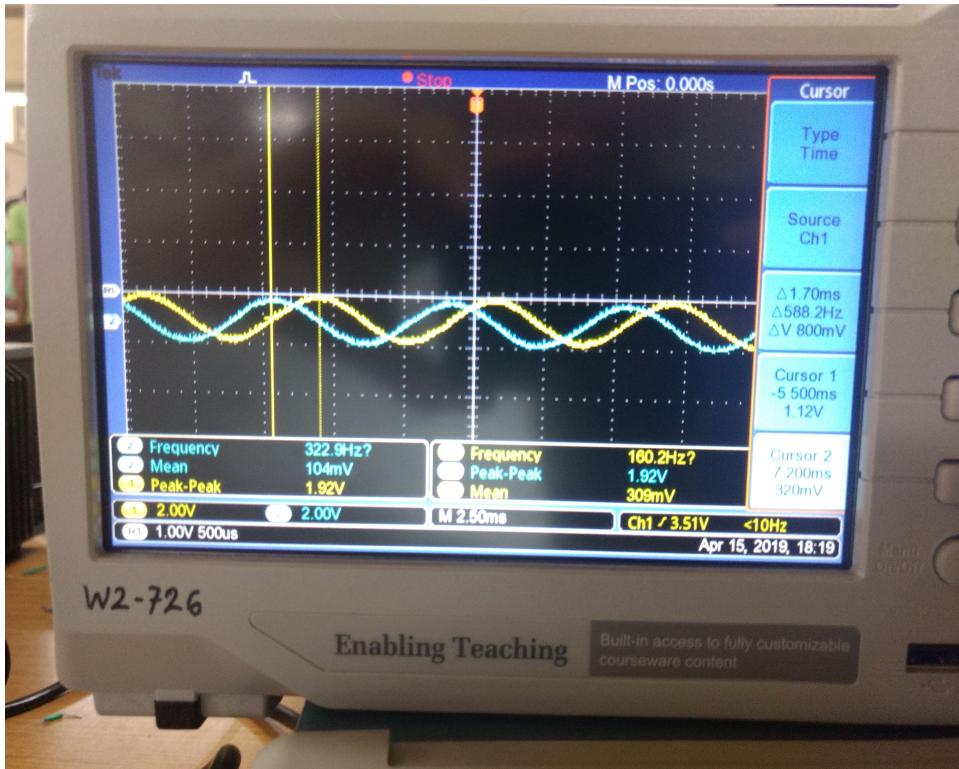


Figure 21: Yellow - λ_{ds} , Blue - λ_{qs}

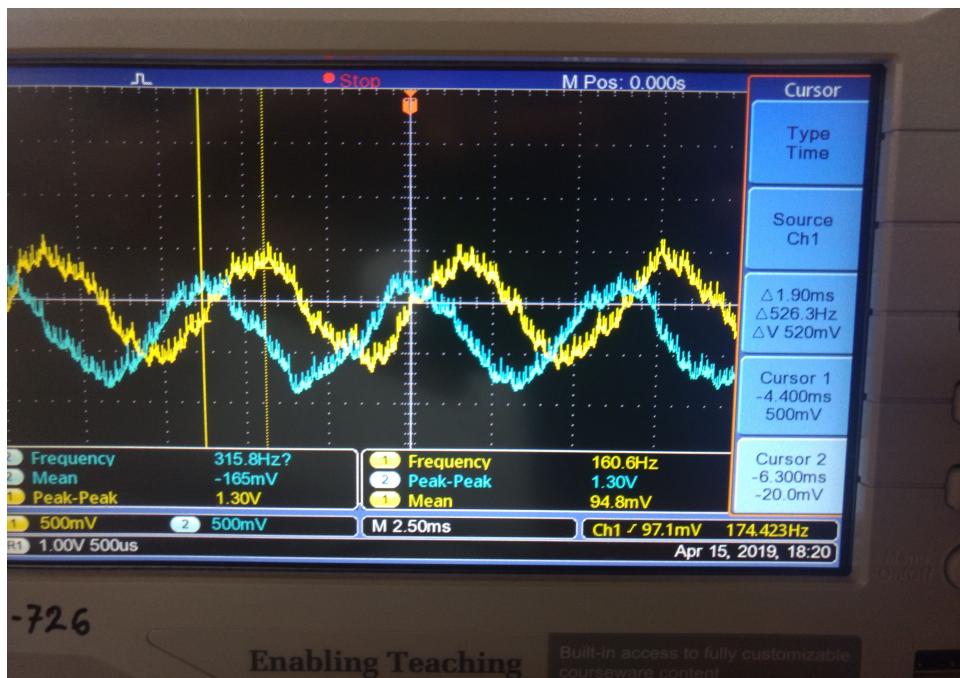


Figure 22: Yellow - λ_{dr} , Blue - λ_{qr}

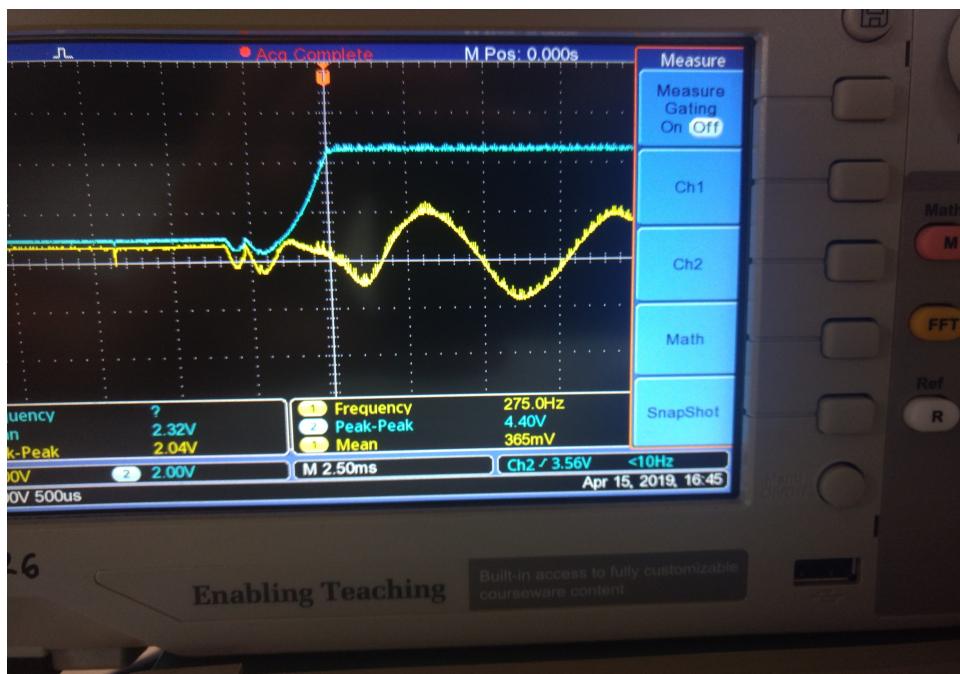


Figure 23: Yellow - λ_{ds} , Blue - ω_{rm} (Transient plot)

5 Conclusion

5.1 Conclusion and Future Work

Thus, the designed DPACv2.0 is capable of solving upto 8th order non-linear differential equations by converting them to a state-space representation and solving it. As seen in the first solution the transient times depend on the RC values of the integrator and hence can be reduced to much smaller times than that required for digital simulations to converge.

Future Work would include the following things

- Booting up the microprocessor using FT232. Another option would be to boot through JTAG for better debugging.
- The final goal could be to solve the 5th order Induction Motor equation with given initial conditions for all values of resistances.
- Integrating the linear computation block to evaluate the linear part in the differential equations

5.2 References

- The entire code, datasheets and schematics can be found at the following open source git repository for future reference:

Git Repository for DPACv2.0

6 Mistakes

- In the scaler circuit, 2 OP-AMPS each block could have been avoided while in the case of rescaler circuit there was 1 redundant OP-AMP per block. If this wasn't the case, around 5-6 TL084 IC's could have been avoided
- During the layout, we left too much space between different blocks because of which the board was bigger than it should have been
- We forgot to provide a rectifier circuit on the board because of which we have to power the board using a DC voltage source and not using the 230V AC voltage passed through a transformer