UNIVERSITY COLLEGE DUBLIN

SCHOOL OF ELECTRICAL, ELECTRONIC & COMMUNICATION ENG.

EEEN30150 MODELLING AND SIMULATION

Minor Project 1

STATIC EQUATIONS

Solve **one** of the following problems or one of your own design (subject to approval by module coordinator). Two of the problems involve relatively extensive visualisation. You should consult the module notes SIM2 and experiment MS0.2 for useful information. The first three problems are *primary problems*, meaning that they pose a reasonably good challenge. The next three problems are *secondary problems*, meaning that they pose a manifestly lesser challenge. You may submit a report on any *one* of the problems, but if you opt to undertake a secondary problem then the highest grade which you can earn for this minor project is C+. Note, the module is substantially concerned with methods for solving equations numerically, accordingly a good portion of the grade steps are earned by writing and reporting good Matlab code to solve the equations arising by means of one (or more) of the methods outlined in the module notes. You will not earn these grade steps if you employ an in-built Matlab function (such as **inv**) or some other package to solve the equations, nor will you earn them if you solve the equations analytically, in the event that this is possible.

On the issue of reporting consult experiment MS1 for a description of requirements. Recall also the comments in introductory laboratory MS0.2 concerning the development of function M-files. A large component of your results, regardless of what project you elect to do, is comprised of Matlab function M-files. If you wish to obtain all of the grade steps assigned to this portion of your work then you must adhere to the requirements that I have previously imposed: (i) help files must be written, (ii) version information must be present, (iii) error checking code should be included to ensure greater robustness and reliability, (iv) there should be no “magic numbers” embedded in the code, (v) names of files and variables should be well chosen, being descriptive of the function or meaning, (vi) comments should be extensive and M-files should have greater rather than lesser functionality, meaning for example that rather than making explicit assumptions concerning parameter values within the code these might instead be input arguments. If you wish to truly impress you may write several different versions of your M-file, one having relatively few inputs where various parameters are assigned to default values at the outset and other versions where some or all of these parameters are passed to the function M-file as additional input arguments. This of course is how Matlab commands are generally written. Work of this kind, if appropriate, will possibly earn bonus grade steps. A certain number of grade steps will be set aside for the writing of function M-files. You may expect to earn at most half of these assigned grade steps should you ignore entirely the six requirements listed above.

Electronic versions of your report are neither required nor accepted for the reasons outlined in MS1. However, the assessor will, in a number of cases, need to execute your code in order to ascertain that it is in fact functioning correctly. In some cases the code in question may be extensive and inputting for every student may become an intolerable chore. Accordingly *after* the submission deadline you may be contacted and asked to submit an electronic version of your code to a specified teaching assistant. You will be contacted at your ucdconnect e-mail address, so make sure that you are monitoring this address. During the laboratory sessions after the submission deadline you may well also be asked to demonstrate your results to the assessor where that is appropriate. So for example students who have solved primary problem 1 where a portion of the outcome is comprised of a collection of audio signals will be asked to play these signals. Students who have solved primary problems 2 or 3 where a portion of the outcome is comprised of an animation or .avi file will be asked to play this animation/file.

On the issue of plagiarism the report and code which you present and claim as your own work must, in fact, be your own work. In this module we will rely mainly on the honour system, however, if two reports or pieces of code are identified by the assessors as being unreasonably similar the module coordinator will investigate. Random spot checks will also be undertaken. Accordingly the module coordinator will approach some of you during the laboratories after the submission deadline and require that you talk through your code and your report, clearly establishing it to be your own work.

Primary Problem 1: Simple Push-Pull Output Stage for Audio Amplifier.

The purpose of a power amplifier, unsurprisingly, is to amplify power. Given a relatively low power source the amplifier must essentially replicate the source but at a much higher power level. A typical application, and the one which we consider here, is in creating an audio amplifier. Audio sources (such as antennas and microphones) generally provide a very low power, low voltage copy of the audio signal. A preamplifier stage must be employed to significantly boost the voltage level and it will be our assumption that this has already been achieved. We are left with a relatively high voltage but very low power copy of the audio signal. The power amplifier must produce another copy of the audio signal but with much higher power so that it can be sent to a speaker with sufficient power to be audible. Although some reduction in the voltage level can be tolerated, this should not be excessive.

To increase the power delivered to the load (i.e. the speaker) we will employ a *complementary symmetry*, *push-pull output stage* fabricated from discrete components (namely bipolar junction transistors). The circuit diagram for the amplifier is shown in Figure 1.1. The diodes (both IN4148 diodes) are included for the purposes of eliminating *crossover distortion* which occurs if both transistors are off for some period of time. If you already have sufficient knowledge you may wish to quickly employ a software package such as LTspice to get a feel for the performance of the circuit before you undertake this project.

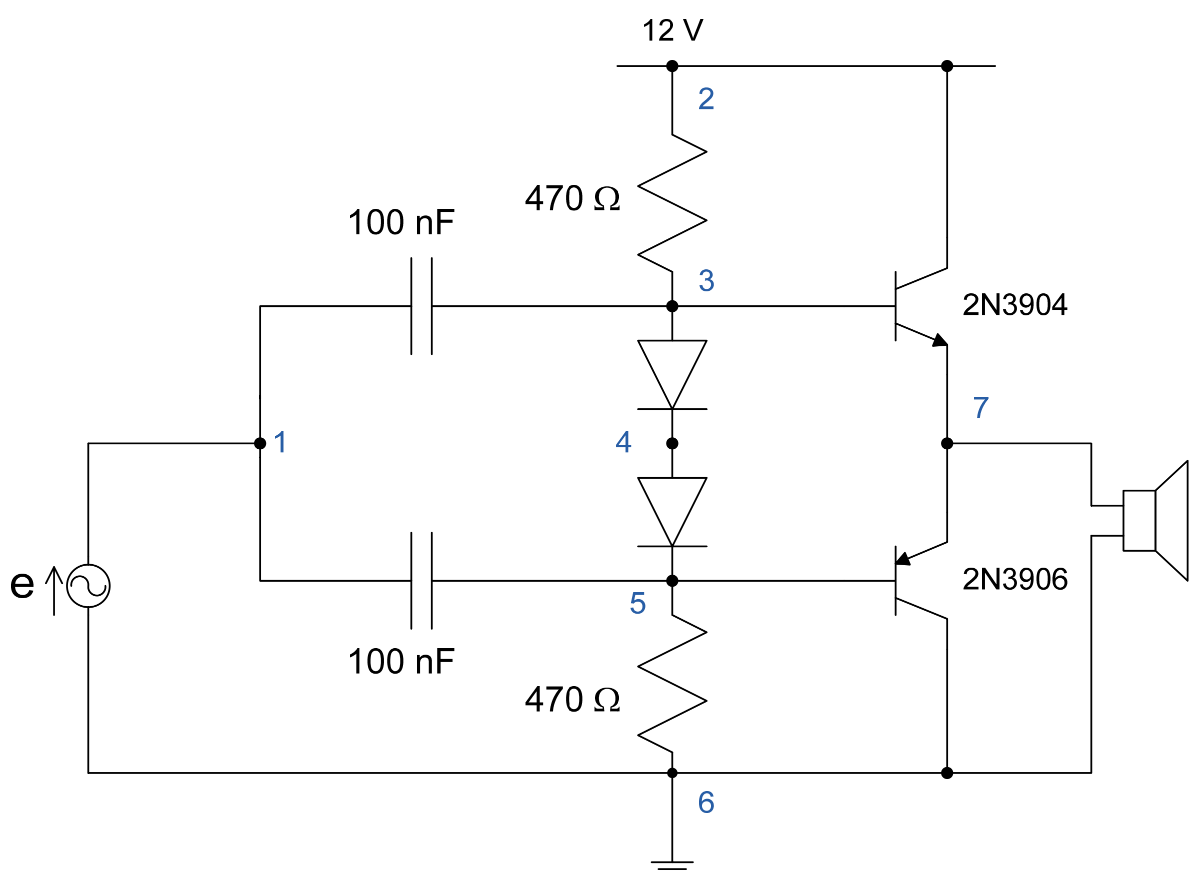

Figure 1.1: Circuit diagram for complementary symmetry, push-pull, output stage of power amplifier. Nodes are numbered in blue.

The assumed effect of the capacitors in this case is to block DC. This means that the voltage at node 3 is the quiescent value (the value attained when the audio input signal is switched off) with the audio signal added to it. Likewise the voltage at node 5 is the quiescent value with the audio signal added to it.

You may model the speaker by its Thévenin equivalent circuit, namely an input resistance. An appropriate value to take for this resistance would lie between 8 and 15 Ω. You should model the diodes by fitting for example the ideal diode equation to the given data.

Manufacturer's datasheets for the transistors are available to download from Fairchild Semiconductor, however they do not explicitly show the typical DC output characteristics. Such characteristics are shown in Figures 1.2 and 1.3. I have cheated here, generating the characteristics by employing circuit simulation software. Accordingly these characteristics are not actual experimental characteristics and are therefore only as good as the models incorporated into the software. However, it is my understanding that these models are rather accurate. The devices can handle moderately large continuous collector currents up to about 200 mA and can safely handle power up to a few watts. You must choose for yourself how to model these devices. You may, for example, model them using the modified Ebers-Moll equations presented in laboratory MS1. In this case you will need to choose values for the parameters of the model which give good agreement with the data given and in this regard you should use the methods of curve fitting presented in the module notes. The relevant Matlab code to achieve this should comprise part of your report.

For given input voltage $e(t)$ at time t and subject to the hypothesised effect of the capacitors discussed above, write down the circuit equations. You *may* need to employ a somewhat systematic procedure such as modified nodal analaysis but I do not insist upon this. Write a function M-file to solve these equations for given input voltage $e(t)$. Note that, although the capacitors imply that the circuit is really described by differential equations, i.e. by dynamic equations, the hypothesised simple effect of the capacitors permits description by static (or perhaps more accurately *pseudo-static*) equations. For a number of input voltage values between -1 V and 1 V determine the power (i.e. instantaneous value of voltage multiplied by current) being drawn from the audio source and the power being delivered to the speaker. Confirm that the circuit has indeed affected a significant amplification of power. Determine also the power being drawn from the DC voltage supply. Confirm that this is much greater than the power being drawn from the audio source. By considering the ratio of the power delivered to the speaker to the power drawn from the DC supply (which is essentially equal to the power input) estimate the *efficiency* of the amplifier.

Consider also the simpler circuit where the two diodes have been omitted. Again write down circuit equations for given input $e(t)$ and create a second function M-file to solve.

Assume that the input signal is a 2 kHz, 0.5 V (ampl.) sinusoid. Create a short burst of such a signal in Matlab and use the **sound** command to listen to it. Find the output signal achieved for this choice of input signal for each of the circuits (the circuit with the diodes and the one without). Again using the **sound** command listen to the output in each case. Can you hear the crossover distortion? A sinusoidal signal is not very interesting. If you wish to earn a bonus grade step find an audio file and repeat this last exercise with that file (or a portion of it). The .wav format is one of the audio formats which are easiest to work with in Matlab. The command which imports such a file into Matlab is the **wavread** command. The command which writes a Matlab sound file to this format is the **wavwrite** command.

DC characteristic data for IN4148 diode. Clearly you are given a large amount of data, but you do not have to use it all.

V (mV)	I (pA)
-1000	-15.2
-600	-14.79
-400	-14.57
-200	-14.2

-150	-13.9
-100	-12.8
-50	-9.68
-40	-8.27
-30	-6.93
-20	-5.07
-10	-2.67
0	0
10	3.73
20	8.27
30	14
50	30
100	123
150	421
V (mV)	I (nA)
200	1.323
250	4.16
300	13.2
350	40.4
380	80
400	128
420	197
440	315
460	496
480	784
V (mV)	I (μA)
500	1.24
520	1.94
540	3.0
560	4.8
580	7.56
600	12
620	18.88
640	29.65
660	46.4
680	73.6
700	116
720	180
740	284
760	440
780	672
V (mV)	I (mA)
800	1.024
820	1.547
840	2.240
860	3.200
880	4.480
900	6.133
920	8.053
940	10.29
960	12.8
980	15.52

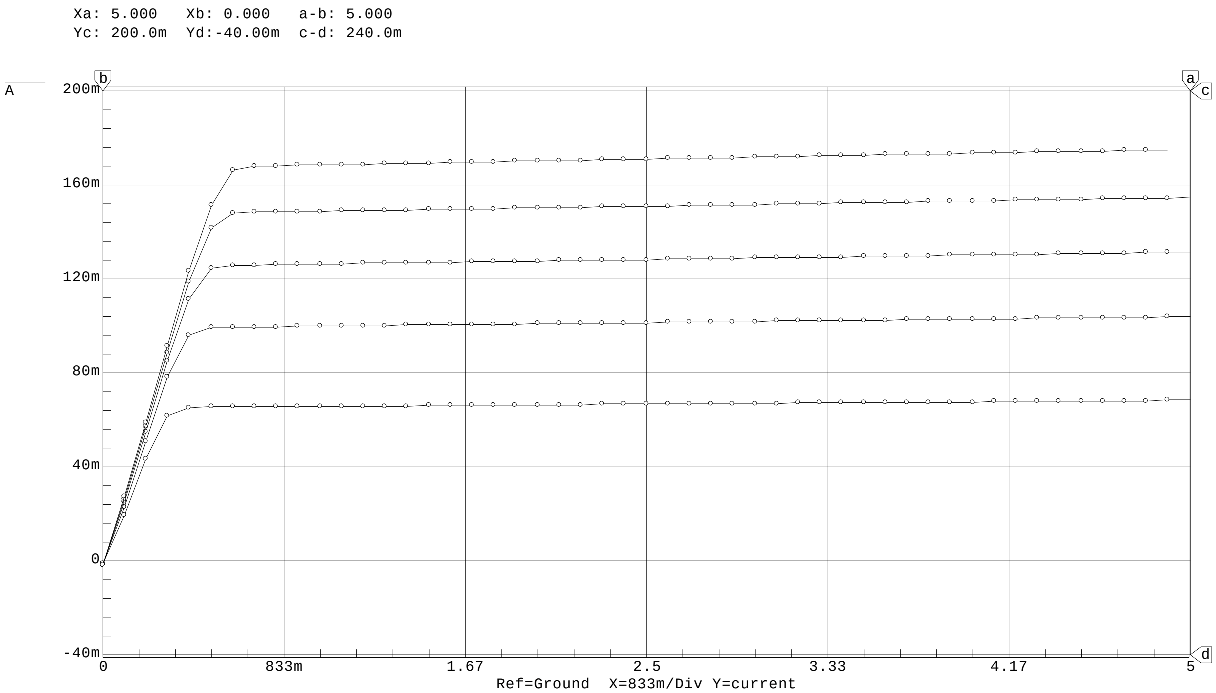

Figure 1.2: DC output characteristics (i.e. I_C vs V_{CE}) of 2N3904 npn BJT.
I_B varying from 1 mA (lower characteristic) to 5 mA (upper characteristic) in steps of 1 mA.

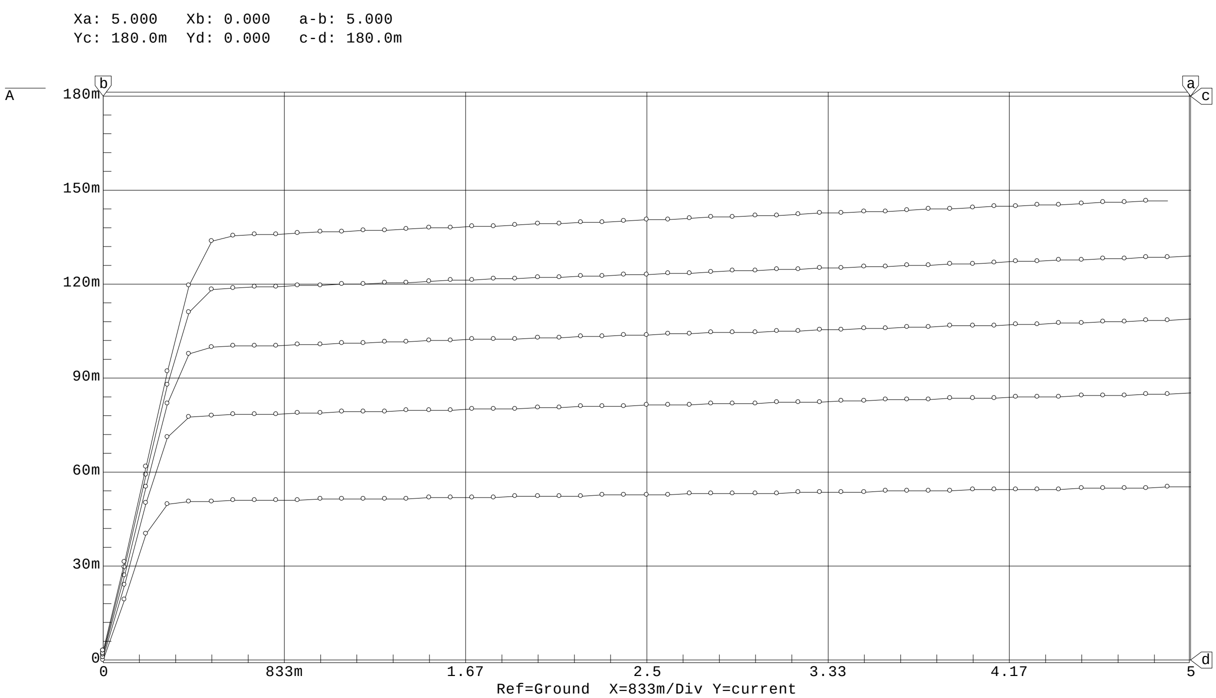

Figure 1.3: DC output characteristics (i.e. I_C vs V_{CE}) of 2N3906 pnp BJT.
I_B varying from 1 mA (lower characteristic) to 5 mA (upper characteristic) in steps of 1 mA.

Primary Problem 2: Mechanical linkage for storage and deployment of roof of a 1991 Ford Mustang convertible.

The mechanical linkage of Figure 2.1 is a *ten bar planar linkage*. The precise positions of three major links are controlled by a number of minor links. All joints are *rotary* (also known as *revolute* or *pin*). In the brief discussion below the *angle* of a link refers to the angle subtended by the line through the link and the positive horizontal axis. The joints are referred to be their number from 0 to 12 as shown in Figure 2.1. The lengths of the various links are as follows:

$$l_{0\to1}=1,\quad l_{1\to2}=1,\quad l_{1\to3}=2,\quad l_{3\to4}=2,\quad l_{0\to5}=5,\quad l_{2\to4}=2.5,\quad l_{4\to6}=2.5,\quad l_{6\to7}=4/3,$$
$$l_{5\to6}=4/3,\quad l_{6\to8}=4,\quad l_{8\to9}=1,\quad l_{7\to10}=4.5,\quad l_{10\to8}=4/3,\quad l_{10\to11}=4/3,\quad l_{9\to11}=1,\quad l_{11\to12}=4/3.$$

where the notation $l_{i\to j}$ refers to the length from joint *i* to joint *j* with the units of length chosen such that the distance between joints 0 and 1 is one unit. Figure 2.1 is approximately drawn for an angle of link 2-4 (the link between joints 2 and 4) equal to 30°.

The linkage is driven by a crank at joint 2. For a given angle of link 2-4 basic kinematics can be employed to determine all of the angles of all of the links and subsequently all of the positions of all of the joints. The numerical problem can be reduced to a series of nonlinear systems of equations which can readily be solved by means of the Newton-Raphson method. So for example I may note that it is possible to express the vector from joint 1 to joint 4 in two ways: as the sum of the known vectors from joints 1 to 2 and from joints 2 to 4, or as the sum of the as yet unknown vectors from joints 1 to 3 and from joints 3 to 4. This yields a *closure equation* from which it is possible to solve for the angle of links 1-3 and 3-4 and therefore to find the position of joint 3. Note that the closure equation permits the determination of two quantities because it is a vector closure equation involving two dimensional vectors. A systematic analysis of the linkage after this fashion yields all of the required joint positions for a given angle of link 2-4.

Write a function M-file which employs the Newton-Raphson method to solve for all of the positions of all of the joints in the mechanism of Figure 2.1 for any given angle of link 2-4. Note the stipulation concerning which numerical procedure is to be employed. Note moreover, the stipulation that a *numerical* procedure, not an analytic procedure be employed.

Employing a number of **patch** graphics objects (see laboratory MS0.2 and module notes SIM2) and employing the animation facilities within Matlab (see module notes SIM2) create either a Matlab movie or a .avi file which animates the performance of the linkage as the angle of link 2-4 sweeps back and forth over a suitable range.

It will probably help to see a reasonably good animation of this mechanism before you start. The animation upon which I have based this problem can be viewed at:

http://synthetica.eng.uci.edu/~mccarthy/animations.html

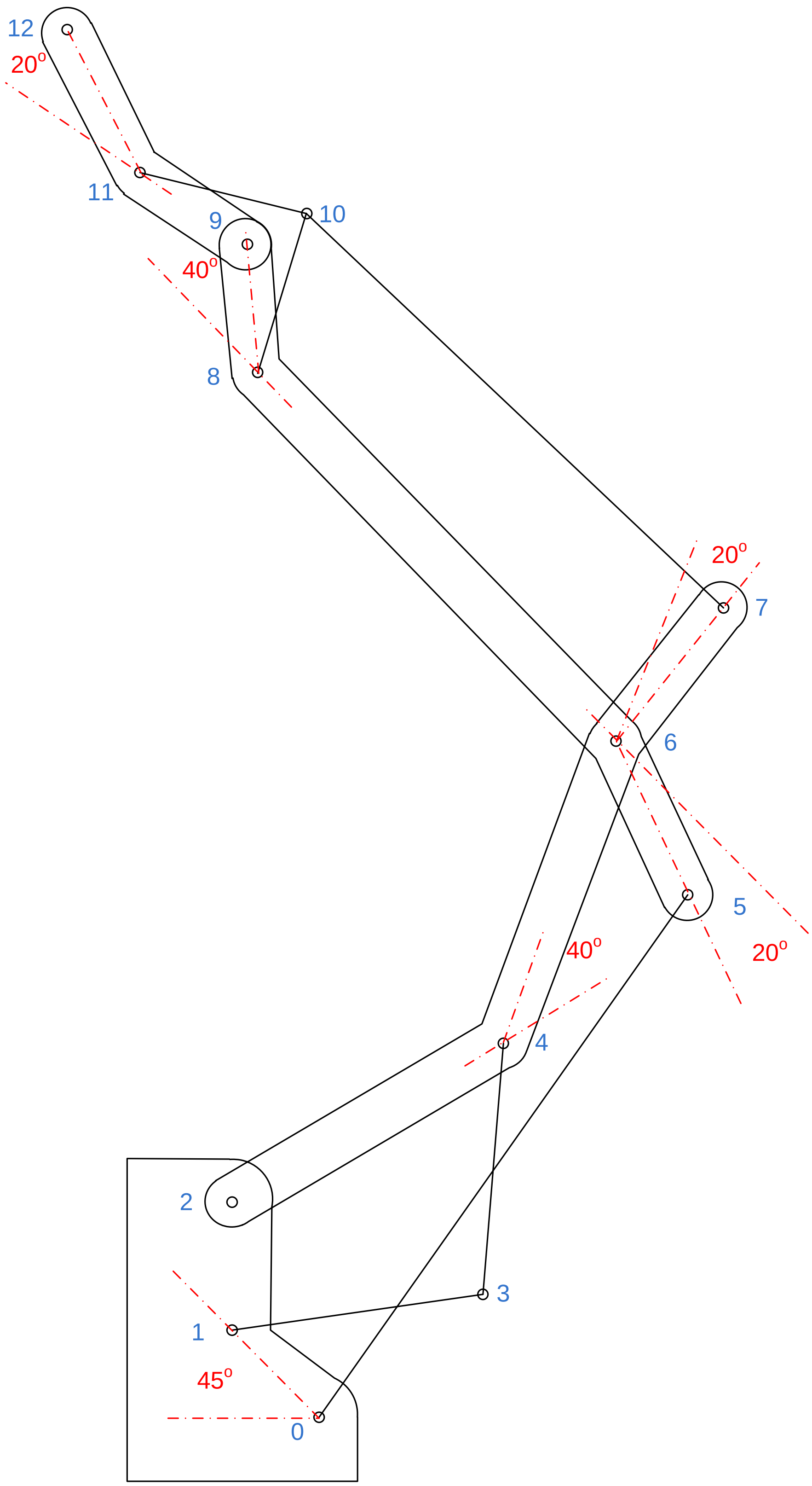

Figure 2.1: Linkage for 1991 Ford Mustang convertible.

Primary Problem 3: Pratt truss bridge.

A truss bridge has a load-bearing superstructure composed of triangular elements. The Pratt truss appears to be one of the oldest truss designs still in use. In the analysis of trusses in general we assume that all loads are applied at joints only. If a weight of a member (or link) is significant then half of this weight may be considered to be applied to the joint at each end of the member. It is assumed that every member is in pure compression or pure tension. Shear, bending moments, *etc* are all considered to be negligible.The forces acting upon the bridge shown in Figure 3.1 can be decomposed into forces acting in the x-y plane and forces acting in the x-z plane (i.e. the forces in the lateral bracing members). Show how this observation permits a very simple analysis of the structure whereby the cross-bracing members can be simply ignored and where the truss to be analysed is a *planar truss*, i.e. a truss in which all members lie in one plane.

A planar truss is *statically determinate* if the number of joints j and the number of members m satisfy the relation $2j = m+3$. Show that the planar truss which adequately models the bridge of Figure 3.1 is statically determinate. The analysis of a statically determinate truss is relatively straightforward entailing only a balance of forces at each joint.

On each side of the bridge of Figure 3.1 each horizontal member and each vertical member is 4 m in length. All of the heavier members of the bridge are assumed to carbon steel S275 bars having a square cross-section of side 120 mm. The lighter cross-bracing members have negligible weight in comparison. The bridge deck accommodates just a single lane of traffic. It is 4 m wide and made of reinforced concrete (of thickness 0.15 m) with an asphalt overlay (of thickness 0.05 m) supported by transverse floor panels also of reinforced concrete. The floor panels are square, of side 4 m and have a thickness of 0.025 m. You should have no difficulty consulting either websites or texts to find the density of carbon steel S275, reinforced concrete and asphalt.

A 2-axle lorry has a wheelbase of 4 m. The front axle load is 10 kN and the rear axle load is 36 kN. Given the speed limit on the bridge the movement of the lorry at this limit has the dynamical effect of increasing the axle loads by 45%

For a given position of the lorry on the bridge determine all of the forces in the various members. Write a function M-file to calculate the forces in all of the members (excluding the cross-bracing members) for a given position of the lorry. Write Matlab code to create a graphical representation of the bridge. Let the colour of a member be chosen to reflect the force in it, with blue indicating a member in which the force is very low and red a member in which the force is very high and colour smoothly transitioning from blue to red as forces increase. Employing a number of **patch** graphics objects (see laboratory MS0.2 and module notes SIM2) and employing the animation facilities within Matlab (see module notes SIM2) create either a Matlab movie or a .avi file which animates the movement of the lorry across the bridge, with changing member colours indicating the changing forces within the various members.

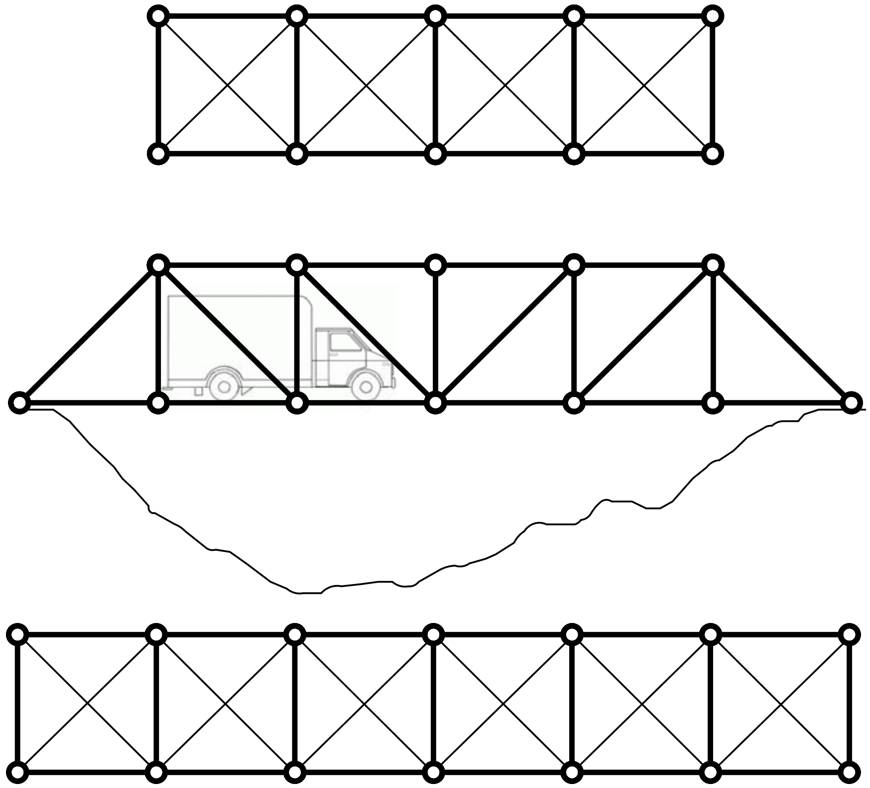

Figure 3.1: 2-axle lorry crossing a Pratt truss bridge.

Secondary Problem 1: Adiabatic Flame Temperature of products after ideal combustion of liquid Octane

Note that you cannot earn higher than a C+ grade for solving this problem.

Combustion is the process of oxidation of all of the constituents of a fuel which can be oxidised. Many, if not all fuels are *hydrocarbons*, meaning that the constituents are hydrogen, which will generally oxidise to water, H_2O, and carbon which will generally oxidise to carbon dioxide, CO_2 or, more rarely, to carbon monoxide, CO. If there is a deficiency of oxygen the combustion is said to be *incomplete*. If the deficiency is slight then CO usually occurs as a product. If the deficiency is significant then uncombusted hydrocarbons appear as products. Otherwise, if there is no deficiency and combustion is complete the oxides appearing as products are H_2O and CO_2 only, since this outcome minimizes the *free energy*. For example octane, C_8H_{18}, permits the following equation:

assuming complete combustion, i.e. an adequate supply of oxygen. The hydrocarbon fuel provides only the hydrogen and the carbon. The oxygen involved is most often provided by the air. Of course air is not made up of dioxygen only. On a molal basis, air is about 21% O_2, 78% N_2 and 1% Ar. We generally assume that the nitrogen and the argon do not undergo any chemical reaction. This does not mean that they can be neglected however, since they will leave the combustion chamber at the same temperature as the other products and this is almost certainly different from the temperature at which they entered. A simplifying assumption is commonly made concerning the air. We assume that, on a molal basis, it is 21% O_2 and 79% N_2. As 79/21 is approximately 3.76, we infer that in this idealised air for every mole of dioxygen there are about 3.76 moles of diatomic nitrogen. Accordingly a better combustion equation, assuming complete combustion, for octane in the presence of sufficient idealised air is:

We see from this equation that for every mole of octane we require 12.5 moles of dioxygen and 12.5 x 3.76 = 47 moles of diatomic nitrogen to ensure complete combustion. This amount of air is called *100% theoretical air*. In practice complete combustion is usually not achieved unless the amount of air somewhat exceeds this theoretical minimum. For example if the amount of air is twice this minimum we say that we have *200% theoretical air*.

If a combustion process takes place adiabatically (i.e. with no heat exchanged), with no work done and with no change in the kinetic or potential energy involved then the temperature of the products reaches a value called the *adiabatic flame temperature*, T_{AFT}. The standard method for determining this temperature amounts to enthalpy balance, requiring that the total enthalpy of the reactants is equal to the total enthalpy of the products. After this brief background we may state the problem to be solved:

Liquid octane at 25^{o}C is burned with 400% theoretical air at 25^{o}C in a steady-flow process at a constant pressure of 0.1 MPa. Determine the adiabatic flame temperature.

You should solve using enthalpy balance. As in the module notes you may define enthalpy, up to a constant of integration, in terms of the specific heat capacity at the constant pressure of 0.1 MPa, the formula being:

$$\overline{h} = \overline{h}_0 + \int_{T_0}^{T} \overline{C}_p(T)\,dT$$

where the overbar indicates that the enthaply is on a molal basis. You should take the reference temperature T_0 to be 25°C.

In performing the enthalpy balance not all of these constants of integration will cancel. However, it will be possible to rearrange these unknown constants of integration so that only the following differences of these quantities appear:

$$\bar{h}_{0,C_8H_{18}} - 8\bar{h}_{0,C} - 9\bar{h}_{0,H_2}$$
$$\bar{h}_{0,CO_2} - \bar{h}_{0,C} - \bar{h}_{0,O_2}$$
$$\bar{h}_{0,H_2O} - \bar{h}_{0,H_2} - 0.5\bar{h}_{0,O_2}$$

As in the module notes we refer to these differences as the enthalpies of formation of liquid n-Octane, Carbon Dioxide and water vapour respectively. The measured values for these quantities are:

Substance	Enthalpy of formation at 25°C, 0.1 MPa Pressure kJ/kmol K
Liquid n-Octane, C_8H_{18}	-250,105
Water, H_2O, gas	-241,826
Carbon Dioxide, CO_2	-393,522

You should find that the value for the enthalpy of formation of the liquid n-Octane is the only value which you require with regard to this substance. On the other hand you will need to find expressions either for the specific heat capacities at constant pressure of 0.1 MPa on a molal basis or for the change in enthalpy at constant pressure of 0.1 MPa on a molal basis from its value at the reference temperature for Dioxygen, Diatomic Nitrogen, Carbon Dioxide and water. In this regard you should write a function M-file which permits you to curve fit using the following information:

O_2		N_2	
T (K)	$\bar{h} - \bar{h}_{298}$ (kJ/kmol)	T (K)	$\bar{h} - \bar{h}_{298}$ (kJ/kmol)
0	-8683	0	-8670
100	-5777	100	-5768
200	-2868	200	-2857
298	0	298	0
300	54	300	54
400	3027	400	2971
500	6086	500	5911
600	9245	600	8894
700	12499	700	11937
800	15836	800	15046
900	19241	900	18223
1000	22703	1000	21463
1100	26212	1100	24760
1200	29761	1200	28109
1300	33345	1300	31503
1400	36958	1400	34936

1500	40600	1500	38405
1600	44267	1600	41904
1700	47959	1700	45430
1800	51674	1800	48979
1900	55414	1900	52549
2000	59176	2000	56137
2200	66770	2200	63362
2400	74453	2400	70640
2600	82225	2600	77963
2800	90080	2800	85323
3000	98013	3000	92715
3200	106022	3200	100134
3400	114101	3400	107577
3600	122245	3600	115042
3800	130447	3800	122526
4000	138705	4000	130027
4400	155374	4400	145078
4800	172240	4800	160188
5200	189312	5200	175352
5600	206618	5600	190572
6000	224210	6000	205848

CO_2		H_20	
T (K)	$\bar{h} - \bar{h}_{298}$ (kJ/kmol)	T (K)	$\bar{h} - \bar{h}_{298}$ (kJ/kmol)
0	-9364	0	-9904
100	-6457	100	-6617
200	-3413	200	-3282
298	0	298	0
300	69	300	62
400	4003	400	3450
500	8305	500	6922
600	12906	600	10499
700	17754	700	14190
800	22806	800	18002
900	28030	900	21937
1000	33397	1000	26000
1100	38885	1100	30190
1200	44473	1200	34506
1300	50148	1300	38941
1400	55895	1400	43491
1500	61705	1500	48149
1600	67569	1600	52907
1700	73480	1700	57757
1800	79432	1800	62693
1900	85420	1900	67706
2000	91439	2000	72788
2200	103652	2200	83153
2400	115779	2400	93741
2600	128074	2600	104520
2800	140435	2800	115463

3000	152853	3000	126548
3200	165321	3200	137756
3400	177836	3400	149073
3600	190394	3600	160484
3800	202990	3800	171981
4000	215624	4000	183552
4400	240992	4400	206892
4800	266488	4800	230456
5200	292112	5200	254216
5600	317870	5600	278161
6000	343782	6000	302295

Finally you should write a function M-file which permits you to solve the resulting enthalpy balance equation to determine the adiabatic flame temperature.

Secondary Problem 2: Toy bridge

Note that you cannot earn higher than a C+ grade for solving this problem.

A bridge is made of sticks as shown in Figure S2.1. The sticks are very light. The (maximal) length of the bridge at is 60 cm. The height is 10 cm. Two books are balanced on top of the bridge. The total mass of the books is 5 kg. Write a function M-file in Matlab to find the forces in all of the members and determine whether they are in tension or in compression.

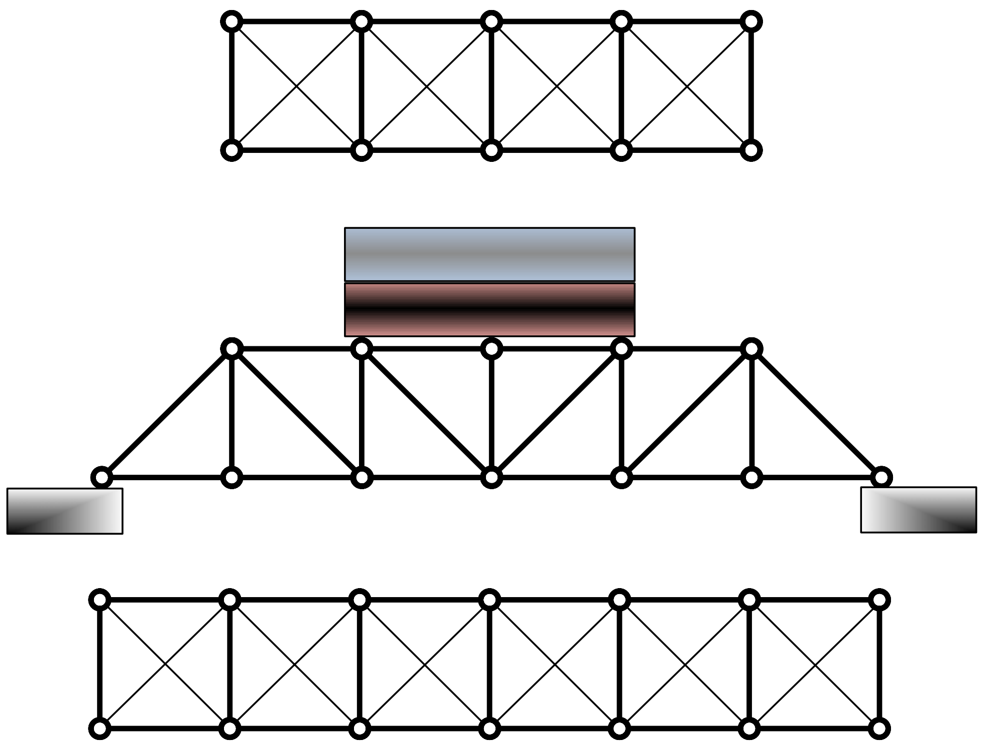

Figure S2.1: Toy Pratt Truss Bridge

In a second scenario the books are hung by employing a net having a hook at each of the four corners which can be connected to a joint. The books are suspended below the bridge again symmetrically placed about the mid joint. Making your own decision as to which joints the hooks are attached, recalculate the forces in all of the members and determine in which scenario the maximal force is greater.

In a third scenario the books are balanced on top of the bridge but moved to the left by the distance between two successive joints. Again recalculate the forces and determine what affect this new placement of the books has on the maximal force.

Secondary Problem 3: Push-Pull Output Stage of Power Amplifier

Note that you cannot earn higher than a C+ grade for solving this problem.

The circuit of Figure S3.1 shows the output stage of a push-pull power amplifier employing two enhancement-mode power MOSFETs, the IRF630 (an N-Channel device) and the IRF9630 (a P-Channel device). The manufacturer of each of these devices is Fairchild Semiconductor. The manufacturer's datasheets for the two devices are available to download, or alternatively they are included as documents IRF630.pdf and IRF9630.pdf on Blackboard. Analyse the biasing circuitry by showing that when no input signal is present the Gate-Source voltage across the N-Channel IRF630 is 3.5745 V and the Gate-Source voltage across the P-Channel IRF9630 is also 3.5745 V. Confirm from the manufacturer's datasheets that the IRF630 has a threshold voltage of approximately 3.5745 V and that the IRF9630 has a threshold voltage of approximately -3.5745 V. The two devices indeed are matched so that their threshold voltages are very nearly equal in magnitude, although opposite in sign of course. You should note from the datasheets that the devices are capable of handling a relatively large current (of the order of several amps) and relatively large drain-source voltages (of the order of hundreds of volts). Hence they can handle rather significant amounts of power (of the order of several tens of watts). High power applications until relatively recently were the sole preserve of bipolar junction transistors, but this is no longer the case.

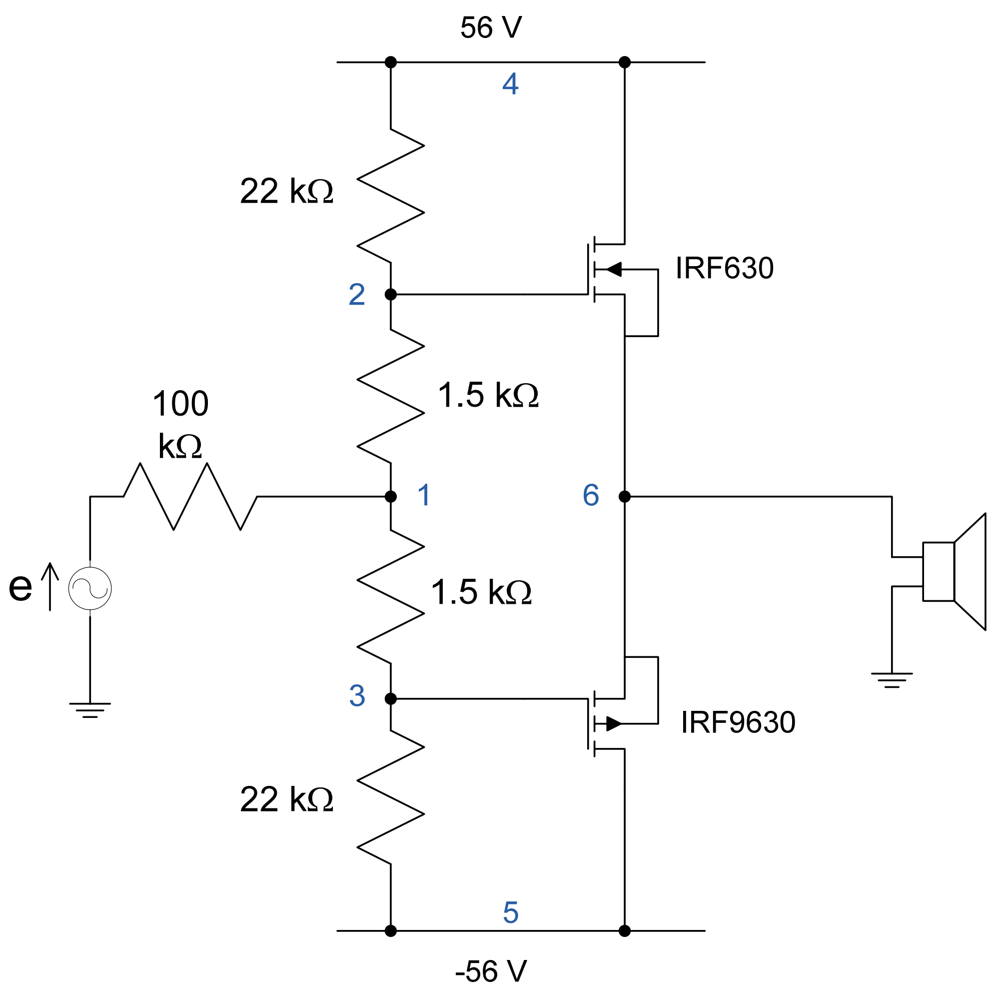

Figure S3.1: Push-Pull Output Stage for Power Amplifier

In the circuit of Figure S3.1 the 100 kΩ resistor models the output resistance of the voltage source which is assumed to be rather high, hence the need for the power amplifier. You may model the speaker by its input resistance which you may take to be relatively low, for example 8 Ω. In the case of each transistor the source is the terminal connected to node 6 and connected to the *body*. To avoid confusion I have employed the standard symbol for a four terminal enhancement-mode MOSFET, although as you will see it is not the more specialised symbol in the manufacturer's datasheets.

By employing the information in the manufacturer's datasheets, in particular the *transfer characteristics* (Figure 7 in each of the datasheets) determine an appropriate model for each of the devices. You may curve fit using the following approximate equations:

IRF630 $$I_D = K\left(V_{GS} - V_{TH}\right)^2\left(1 + \left(\frac{V_{DS} - V_{DS,sat}}{R_{DS}}\right)\right) \quad \text{if } V_{GS} \geq V_{TH}\ , V_{DS} > V_{GS} - V_{TH}$$

$$I_D = 0 \quad \text{if } V_{GS} < V_{TH}$$

where I_D denotes the drain current, i.e. the current flowing into the drain and out of the source, V_{GS} denotes the gate-source voltage (also called the gate to source voltage) and V_{DS} denotes the drain-source voltage (also called the drain to source voltage). Parameters K, V_{TH}, $V_{DS,sat}$ and R_{DS} should be chosen to yield a good fit.

IRF9630 $$I_D = -K\left(V_{SG} - V_{TH}\right)^2\left(1 + \left(\frac{V_{SD} - V_{DS,sat}}{R_{DS}}\right)\right) \quad \text{if } V_{SG} \geq V_{TH}\ , V_{SD} > V_{SG} - V_{TH}$$

$$I_D = 0 \quad \text{if } V_{SG} < V_{TH}$$

where I_D denotes the drain current, i.e. the current flowing into the drain and out of the source, V_{SG} denotes the source-gate voltage (also called the source to gate voltage) and V_{SD} denotes the source-drain voltage (also called the source to drain voltage). Parameters K, V_{TH}, $V_{DS,sat}$ and R_{DS} should again be chosen to yield a good fit.

You will not find that these equations give a perfect fit to the manufacturer's data. Nevertheless the fit can be made acceptable. By assuming zero current below threshold the proposed equations ignore leakage. It is reasonable therefore to also ignore gate current, i.e. to assume that each of the gate currents is zero. This greatly simplifies the analysis of the circuit.

Write a function M-file or M-files in Matlab to "solve the circuit", i.e. to find all of the currents and voltages for a given input voltage *e*. You may assume that this input voltage is reasonably small, less than 1 V in magnitude for example. You may solve the entire circuit by employing the Newton-Raphson algorithm or you may note that the circuit equations divide into two completely independent sets of equations, one of which is a set of linear equations which may be solved by Gaussian elimination. I reiterate, if you solve a system of linear equations as part of your solution to this problem you must write Matlab code to solve these equations using the methods discussed in the module.

Finally calculate the electrical power delivered to the load (i.e. to the speaker) for a number of input voltages between -1 V and 1 V for example. Confirm that rather significant power is being delivered to the load.