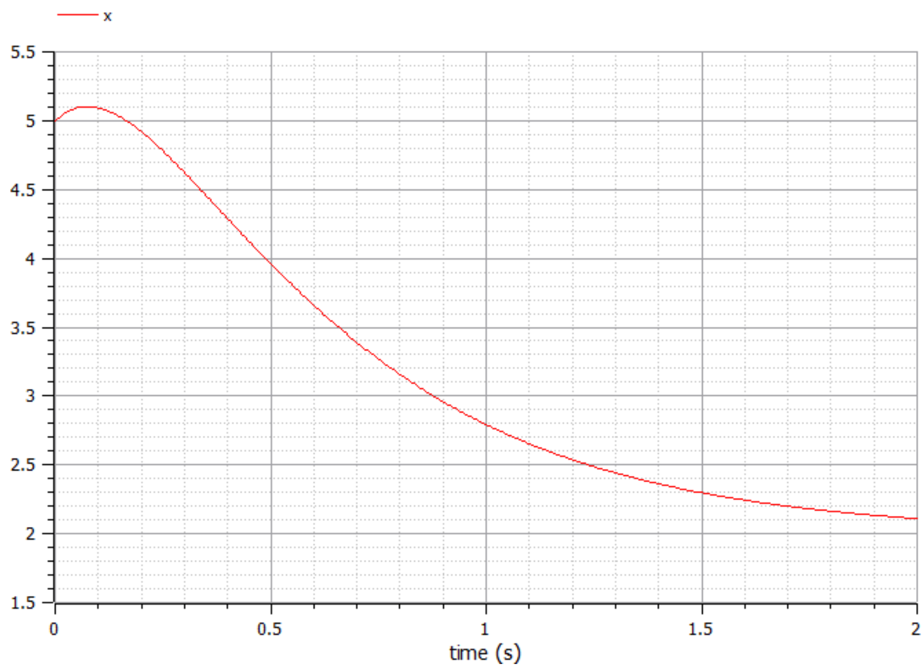
# MODELLING PHYSICAL SYSTEMS




## Objective

The aim of this short paper is to introduce System Modelling and why it can help in the real world.

## Contents



UK: +44-151-329-0643
Ireland: +353-74-971-0878
contact@losiento.biz
www.losiento.biz

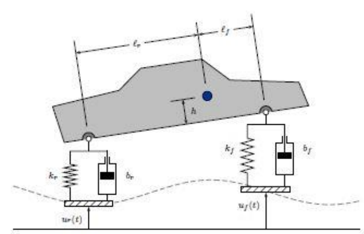
## System Modelling

There is an interesting engineering peculiarity that you might not be aware of; many components in different engineering domains have similar forms of behaviour which are in turn dictated by the fundamental laws of physics.

This means somewhat surprisingly - *and very usefully* - that electrical, mechanical, thermal and fluid systems all look alike mathematically.

| System | Across Variable | Through Variable | Power Equation | Resistor Equation | Capacitor Equation | Inductor Equation |
|---|---|---|---|---|---|---|
| Electrical | Voltage ($V$) | Current ($I$) | $V \times I$ | $V = IR$ | $I = C\frac{dV}{dt}$ | $V = L\frac{di}{dt}$ |
| Mechanical Translation | Velocity ($v$) | Force ($F$) | $v \times F$ | $v = \frac{1}{b}F$ | $F = m\frac{dv}{dt}$ | $v = \frac{1}{k}\frac{dF}{dt}$ |
| Mechanical Rotation | Angular Velocity ($\omega$) | Torque ($T$) | $\omega \times T$ | $\omega = \frac{1}{b}T$ | $T = I\frac{d\omega}{dt}$ | $\omega = \frac{1}{k}T$ |
| Fluid | Pressure ($P$) | Volume Flow Rate ($Q$) | $P \times Q$ | $P = RQ$ | $Q = C_f\frac{dP}{dt}$ | $P = I\frac{dQ}{dt}$ |
| Thermal | Temperature ($T$) | Heat Flow Rate ($Q$) | $Q$ | $T = RQ$ | $Q = C_t\frac{dT}{dt}$ | - |

We can exploit this fact to our advantage by approximating the components of a system with ideal fundamental elements.



We then use these elements to generate differential equations that when solved give us the response of the system with respect to a specific input.

For your reference this approach is commonly known as lumped-parameter modelling or lumped-element modelling.

*The Analog computer used by NASA for their Apollo program*

Now in the not so distant past engineers had only complex Analog computers that had to be programmed with a set of initial conditions to solve for the response of a system.

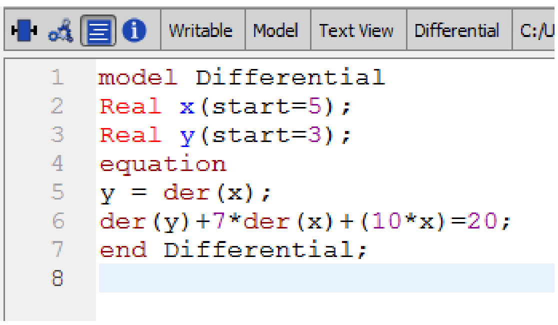
These Analog computers used hundreds if not thousands of vacuum tubes and the programming was carried out by manually wiring connections between the different components of the computer. Today in contrast we are comparatively spoilt…

Powerful software such as Modelica[1] allows us to understand the behaviour of electrical, mechanical, thermal and fluid systems at a far greater speed than could ever have been hoped for in the past.

1. www.openmodelica.org

MODELICA

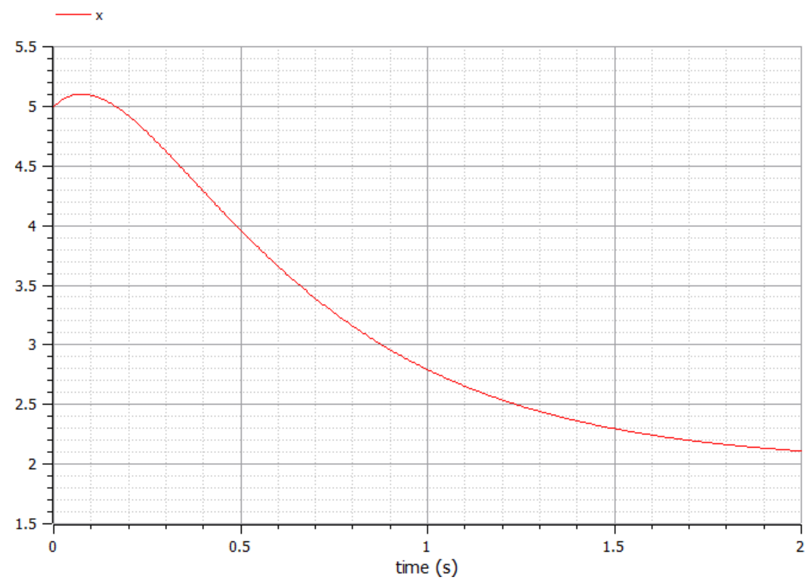
When using Modelica I typically translate equations directly into Modelica Language in order to then plot the result.



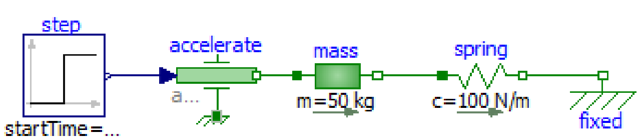

$$\ddot{x} + 7\dot{x} + 10x = 20$$

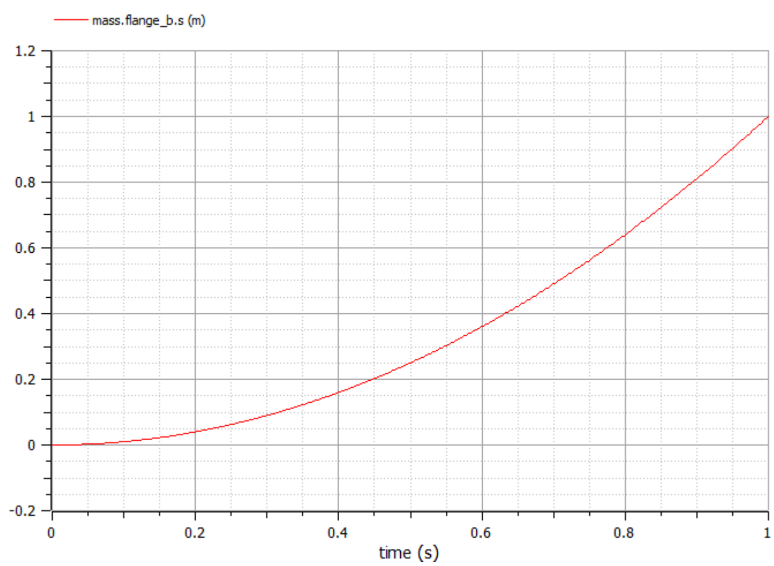
$$x(0) = 5$$

$$\dot{x}(0) = 3$$




I can also drag and drop fundamental elements onto a work sheet and solve for their response if I don't wish to work directly from first principles.




So how can system modelling actually help in the real world…?

## Practical Applications


The traditional way of building a new product and one that a lot of companies still use is to "*build-test-fix*".

We ***build*** a physical prototype of the design, ***test*** it and then ***fix*** any problems we find.

However the danger is that this can quickly lead to the '*Whac-a-mole*' approach to engineering…

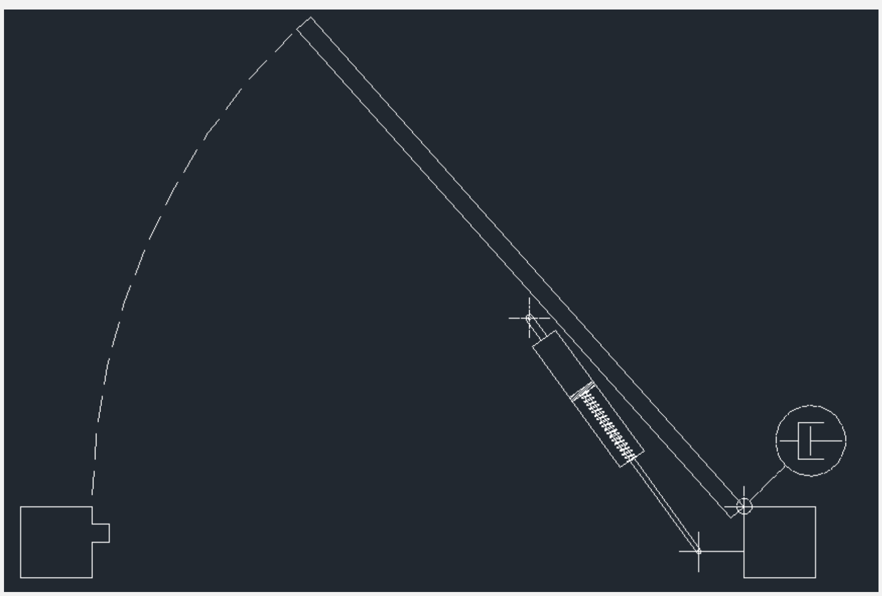
As we correct one problem we find that another one immediately emerges which is a slow and expensive way to work.

A better approach is to investigate key aspects of a design using system modelling techniques so that we can make any mistakes well before we start cutting metal.

In order to help you understand how System Modelling techniques can be applied in the real world I have created two examples…

### 1. Door Closer

The general layout of a door closer assembly has been established. Now we need to get it working.



As the door is opened the spring is compressed and it stores energy. When the door is then released and the spring extends the rate at which it will close is controlled by a rotary damper.

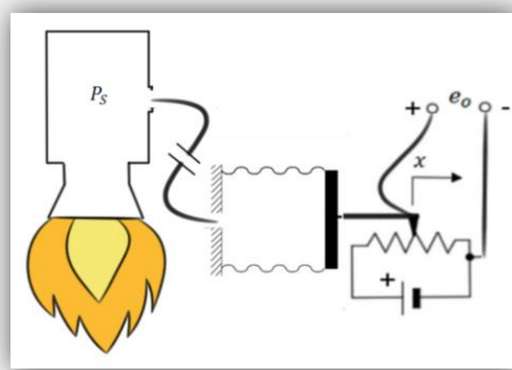
We need to pick the values for the spring and a damper such that the door will close positively without slamming. Once we have this information we then need to use it to generate smaller and bigger versions of the same device.

For the solution see **Appendix A**

## 2. Measuring the pressure of a Rocket Motor

During the operation of a rocket motor the pressure variations occurring within the combustion chamber need to be monitored by some means.

In order to isolate the measuring equipment from the high temperatures a long fluid filled line has been proposed.




The end of the fluid line is to be connected to a piston and bellows assembly which will in turn activate a variable resistor.

The voltage at the variable resistor aims to give us an indication of the pressure occurring within the rocket chamber.

This is an example of a system that involves a mixture of hydraulic, mechanical and electrical components.

For the solution see **Appendix B**

## Conclusion

Creating a simplified model of a system in order to allow us to study its behaviour gives us the confidence that when we do start building a prototype it will work as intended.

Remember that mistakes made on the drawing board cost an awful lot less than those made in metal.

System modelling techniques also help when we are working on a problem that spans multiple engineering domains. The rocket motor example above shows that despite having hydraulic, mechanical and electrical components to contend with we can still develop an understanding of how the system will perform in real life.

*Thanks for taking the time to read this short paper.*
*If you have a design problem that you would like to discuss then get in touch today:*


UK +44 151 329 0643
ROI +353 74 971 0878
contact@losiento.biz

# Appendix A

**Dynamic Analysis:**
**Description:** Door Closer
**Units:** Metric
**Issue:** A

**Overview:**

Values for both the spring (2,200 N/m) and the rotary damper (20 Nms/rad) have been found that achieve a positive closing action with respect to the constraints imposed upon the system.

Dimensional Analysis was then performed to ensure that different versions of the product can quickly be created.

**System Definition:**

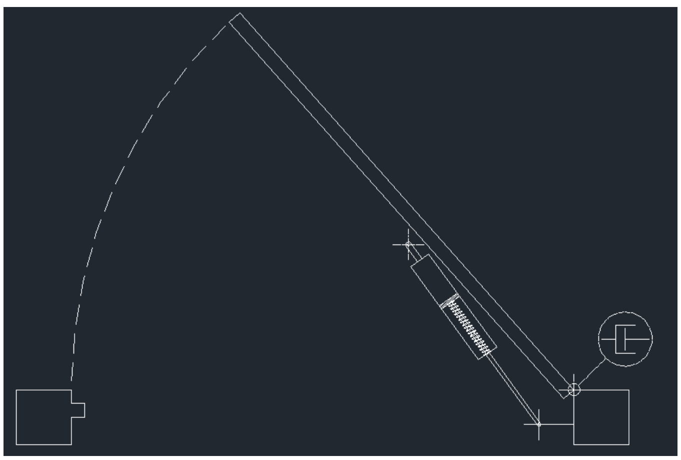


***Figure 1:*** *Door closer assembly*

The following constraints are imposed upon the system:

1. The door can only open 90 degrees
2. The rotary damper lies on the hinge axis
3. As the door is opened the spring is compressed
4. The door must close in less than 5 seconds
5. From practical experimentation it has been determined that the door velocity at 0 degrees must be $< 0.05\, m/s$

The following assumptions have been made:

1. Friction at the door hinges is to be discounted
2. Any form of damping arising from air resistance as the door closes is to be discounted
3. When the door is to be closed it is to be released freely from the 90 degree position, i.e. no additional forces are to be considered

**Nomenclature:**

Note that the following values have been suggested for the concept scheme:

| | Units | Value | Description |
|---|---|---|---|
| $\emptyset$ | Rad | TBD | See diagram |
| $\alpha$ | Rad | 0.12531 | See diagram |
| $\theta$ | Rad | TBD | See diagram |
| $\dot{\theta}$ | Rad/s | TBD | Angular velocity |
| $\ddot{\theta}$ | Rad/s^2 | TBD | Angular acceleration |
| $\theta_d$ | Degrees | TBD | Angle which door is ajar |
| $K$ | N/m | TBD | Spring constant |
| $b$ | Nms/rad | TBD | Damping constant |
| $L$ | mm | TBD | Extended length |
| $L_1$ | mm | 304.8 | Compressed length |
| $x$ | mm | 63.5 | See diagram |
| $y$ | mm | 63.5 | See diagram |
| $z$ | mm | 400 | See diagram |
| $J$ | kg m^2 | TBD | Polar moment of inertia |
| $H$ | mm | 914.4 | Width of door |
| $B$ | mm | 25 | Door thickness |
| $W$ | kg | 35.982 | Mass of door |

Values for both the damping and spring constants need to be determined.

**Mathematical Model:**

The torque that is applied by the door closer is given by:

$$T = sin\,\phi(K(L - L_1))\,z \quad \textbf{[1]}$$

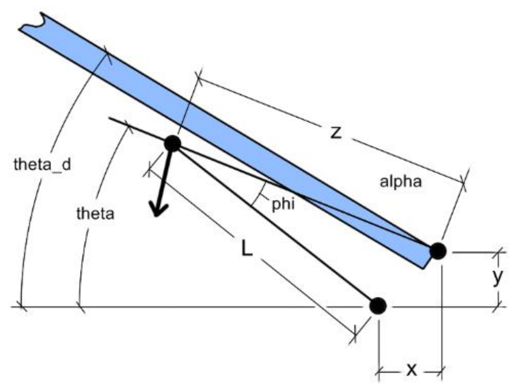



***Figure 2:*** *System nomenclature reference*

To solve for the angle $\emptyset$ with respect to X, Y, L and Z

The cosine rule is given by:

$$c = \sqrt{a^2 + b^2 - 2ab\ \cos\phi} \quad \textbf{[2]}$$

If we the hypotenuse created by $x$ and $y$:

$$\sqrt{c} = \sqrt{z^2 + L^2 - 2zL\ \cos\phi} \quad [3]$$

Rearranging for the angle we get:

$$c = z^2 + L^2 - 2zL\ \cos\phi \quad [4]$$

$$\phi = \cos^{-1}\left(\frac{z^2 + L^2 - (x^2 + y^2)}{2zL}\right) \quad [5]$$

The value of $L$ is derived as:

$$L = \sqrt{x^2 + y^2 + z^2 + 2yz\sin\theta - 2xz\cos\theta} \quad [6]$$

The angle at which the door is ajar is given by:

$$\theta_d = \theta + \alpha \quad [7]$$

Treating the door as a single lumped element then we know that the toque created by the spring will be resisted in turn by effect of the rotary damper.

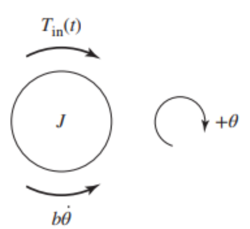



***Figure 3:** Free body diagram*

On this basis:

$$\sum T = J\dot{\omega} \quad [8]$$

Rearranging with [1] gives:

$$sin\ \emptyset(K(L - L_1))\ z = J\ddot{\theta} + b\dot{\theta} \quad [9]$$

Finally we know that the mass moment inertia (Ref: *Figure 4*) of the door about its hinge will be given by:

$$J = \frac{M}{3}(H^2 + B^2) \quad [10]$$

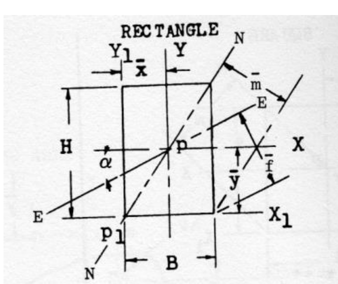



***Figure 4:** Griffel – Handbook of formulas for Stress and Strain*

**Modelica Language:**

The equations in the previous section were coded into Modelica language in order to plot the velocity of the door against time.

Note that the starting condition of the door is at 90 degrees when the spring is under maximum loading.

```
model Door_Closer

parameter Real b=20;          //Damping(Nms/rad)//
parameter Real K=2200;        //Stiffness (N/m)//
constant Real M=35.982;       //Mass = 35.982kg//
constant Real H=914.4;        //Width of door//
constant Real B=25.;          //Depth of door//
constant Real x=0.0635;       //X Hinge Point//
constant Real y=0.0635;       //Y Hinge Point//
constant Real z=0.4;          //See diagram//
constant Real ALPHA=0.12531;  //7.18 angle//

Real L;                       //Stress length//
Real L1=0.3336;               //Unstress length//
Real PHI;                     //See diagram//
Real J;                       //Inertia//
Real w;                       //angular v//
Real wdot;                    //angular a//
Real THETA(start=-1.4454817);//82.82 deg//
Real THETADOOR;               //Door angle rad//
Real THETADEG;                //Door angle deg//
Real DOORVEL;                 //Impact point v//

equation
J=(M/3.)*(H^2+B^2)/1000000.;
L=sqrt(x^2+y^2+z^2+2.*y*z*sin(-THETA)-
2.*x*z*cos(-THETA));
PHI=(acos((z^2+L^2-(x^2+y^2))/(2.*z*L)));
w=der(THETA);
wdot=der(w);
(K*(L-L1)*sin(PHI)*z)=(J*wdot+b*w);
THETADOOR=(THETA)-(ALPHA);
THETADEG=THETADOOR*(180./3.14159);
DOORVEL = w*(H/1000.);

end Door_Closer;
```

**Dynamic Simulation Model:**
In order to be able to verify the hand calculations a simplified CAD model of the door assembly was created and analysed using the dynamic simulation capabilities within Autodesk Inventor.

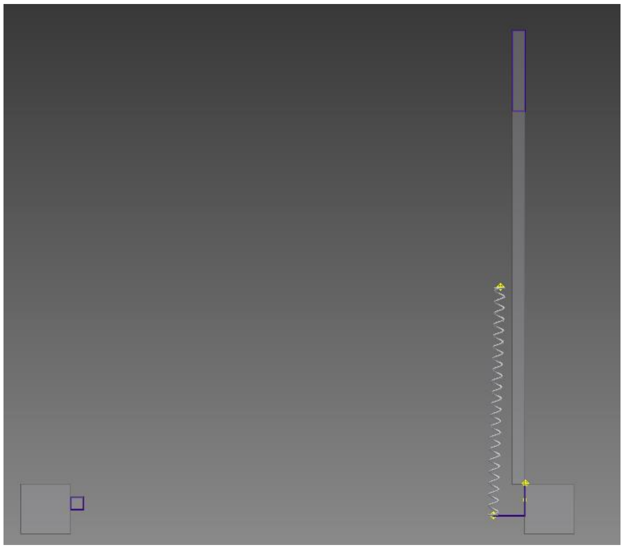


***Figure 5:*** *Door at 90 degree position*

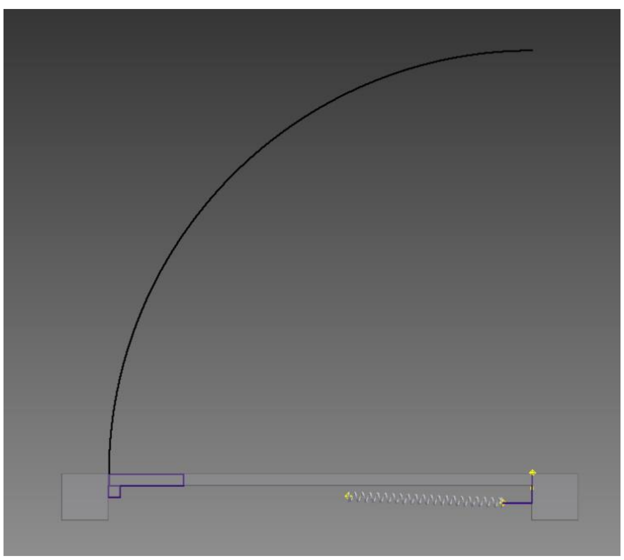


***Figure 6:*** *Door at 0 degree position with impact point trace*

**Results:**
The following design parameters were found to close the door in just shy of 4 seconds and achieve a velocity of 0.026 m/s at the point of impact:

| | | |
|---|---|---|
| DP1 | (Damping Constant) | 20 Nms/rad |
| DP2 | (Stiffness Constant) | 2,200 N/m |

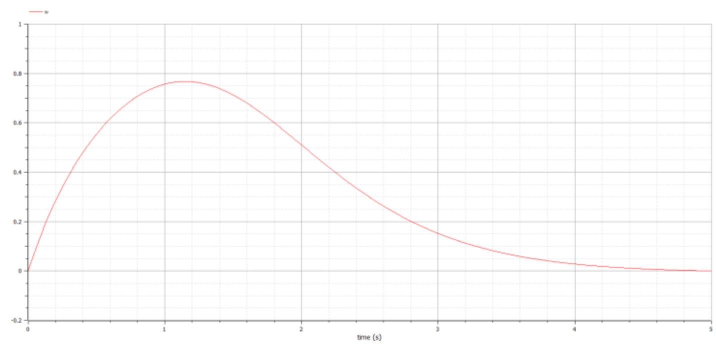


***Figure 7:*** *Modelica response*

When ignoring minor rounding errors the dynamic simulation gives an identical value to the Modelica results.

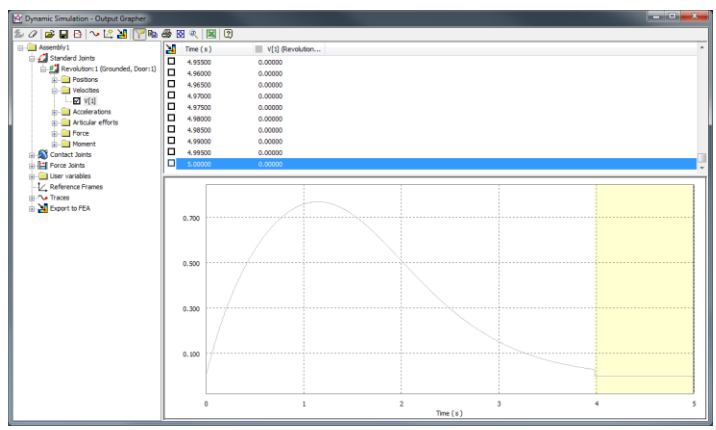


***Figure 8:*** *Inventor response*

**Dimensional Analysis:**

In order to extend a product line to include smaller or larger devices Dimensional Analysis can be used to generate off-scale models.

For the door assembly the following variables were considered to be of interest:

$\omega$ = Angular velocity (radian/second)
$J$ = Inertia (kilogram meter squared/radian squared)
$K$ = Spring constant (newton/meter)
$B$ = Damping constant (N m s/rad)
$t$ = Time (second)
$\alpha$ = Angular acceleration (radian/second squared)
$L$ = Length (m)

On this basis the units required are as follows:

$$\omega = s^{-1} \times rad/sr$$
$$J = kg \times m^2 \times (rad/sr)^{-2}$$
$$K = \frac{F}{x} = \frac{kg\, m\, s^{-2}}{m} = \frac{kg}{s^2} = kg \times s^{-2}$$
$$B = kg \times m^2 \times s^{-1}$$
$$t = s$$
$$\alpha = s^{-2} \times rad/sr$$
$$L = m$$

The $\pi$ terms are thus:

$$\pi = \omega^a J^b K^c B^d t^e \alpha^f L^g \quad \text{[11]}$$

Collecting the powers for each term and rearranging gives:

$$\pi = s^{(-a-2c-d+e-2f)} kg^{(b+c+d)} m^{(2b+2d+g)} rad/sr^{(a-2b+f)} \quad \text{[12]}$$

Remembering that $\pi$ is dimensionless it allows us to write out the exponents we have collected in the bracketed terms and then set each one to zero:

$$-a - 2c - d + e - 2f = 0 \quad \text{[13]}$$

$$b + c + d = 0 \quad \text{[14]}$$

$$2b + 2d + g = 0 \quad \text{[15]}$$

$$a - 2b + f = 0 \quad \text{[16]}$$

We now have a set of simultaneous equations, the variables of which are the exponents of the system variables.

The rank of the dimension matrix is:

$$\mathrm{rank}\left(\begin{bmatrix} -1 & 0 & -2 & -1 & 1 & -2 & 0 \\ 0 & -2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 & 0 & 1 \\ 1 & -2 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}\right) = 4$$

The associated number of pi terms is thus:

$$\#solutions = \#variables - rank$$

$$3 = 7 - 4 \quad \text{[17]}$$

Solving the simultaneous equations gives:

$$d = -b - c \quad \text{[18]}$$
$$e = -a + 3b + c \quad \text{[19]}$$
$$f = 2b - a \quad \text{[20]}$$
$$g = 2c \quad \text{[21]}$$

A solution matrix can be formed on the basis of:

$$\pi = \omega^a J^b K^c B^d t^e \alpha^f L^g \quad \text{[22]}$$

| | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| | $\omega$ | $J$ | $K$ | $B$ | $t$ | $\alpha$ | L |
| $\pi_1$ | **1** | **0** | **0** | 0 | -1 | -1 | 0 |
| $\pi_2$ | **0** | **1** | **0** | -1 | 3 | 2 | 0 |
| $\pi_2$ | **0** | **0** | **1** | -1 | 1 | 0 | 2 |

This lets us write the $\pi$ term such that:

$$\pi_1 = \omega^1 J^0 K^0 B^0 t^{-1} \alpha^{-1} L^0 \quad \text{[23]}$$

$$\pi_2 = \omega^0 J^1 K^0 B^{-1} t^3 \alpha^2 L^0 \quad \text{[24]}$$

$$\pi_3 = \omega^0 J^0 K^1 B^{-1} t^1 \alpha^0 L^2 \quad \text{[25]}$$

Hence:

$$\pi_1 = \omega^1 t^{-1} \alpha^{-1} = \frac{\omega}{t\alpha} \quad \text{[26]}$$

$$\pi_2 = J^1 B^{-1} t^3 \alpha^2 = \frac{Jt^3\alpha^2}{B} \quad [27]$$

$$\pi_3 = K^1 B^{-1} t^1 L^2 = \frac{KtL^2}{B} \quad [28]$$

**Dimensional Scaling:**

If we wish to make a future change to the system but ensure that the behaviour remains the same then we must keep the $\pi$ terms equal.

As an example we wish to change the mass of the door from 35.982 kg to 70.0kg but maintain the same angular velocity response.

On this basis from the original experiment:

$\alpha = 0.45794$
$\omega = 0.46676$
$t = 2$
$J = 10.036$
$K = 2200$
$B = 2200$
$L = 0.9144$

$$\pi_1 = \frac{\omega}{t\alpha} = 0.5096 \quad [29]$$

$$\pi_2 = \frac{Jt^3\alpha^2}{B} = 0.8419 \quad [30]$$

$$\pi_3 = \frac{KtL^2}{B} = 183.948 \quad [31]$$

For the off-scale model we can only alter the spring and damping constants to achieve similarity:

$\alpha = 0.45794$
$\boldsymbol{\omega = TBD}$
$t = 2$
$J = 19.5242$
$\boldsymbol{K = 4279.879}$
$\boldsymbol{B = 38.908}$
$L = 0.9144$

With the new design parameter values for $K$ and $B$ we arrive at:

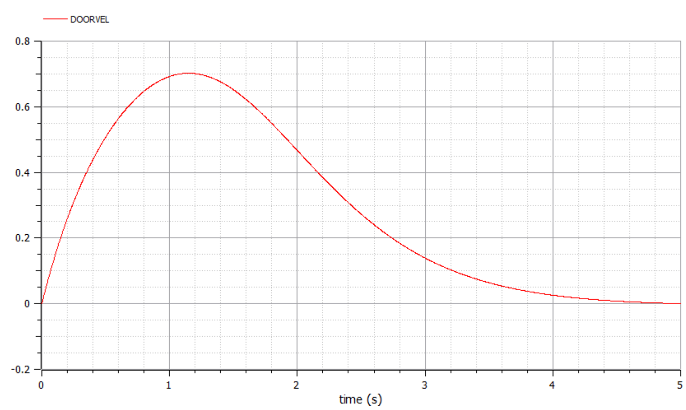
$$\pi_{1\_new} = \frac{\omega}{t\alpha} = 0.5096 \quad [32]$$

$$\pi_{2\_new} = \frac{Jt^3\alpha^2}{B} = 0.8419 \quad [33]$$

$$\pi_{3\_new} = \frac{KtL^2}{B} = 183.948 \quad [34]$$

Performing the revised simulation in Modelica shows an angular velocity of 0.46676 rad/s at 2 seconds which is identical to the desired value.

It can be seen that the response curve for the heavier door mimics the response that was developed within the original study.




**Conclusion:**
As the mathematical model and the dynamic simulation agree with one another we can have confidence in the derived results.

The next step is to look for commercially available OEM solutions that can be purchased as close to the provided target values as possible.

If products are found that are different to the target values then the $\pi$ terms established through the Dimensional Analysis process can be utilised.

**END OF DOCUMENT**

# Appendix B

**Dynamic Analysis:**
**Description:** Rocket Chamber Pressure
**Units:** Imperial
**Issue:** A

## Overview:

A design has been proposed to measure the pressure variations within a rocket engine chamber. Dynamic analysis shows that the design will function as intended though further refinement might well be required depending upon the application.

## System Definition:

During operations of a rocket motor the pressure variations occurring within the combustion chamber need to be monitored.

In order to isolate the measuring equipment from the high temperatures created by the rocket motor the concept of a long fluid filled line has been proposed.

The line is to be connected to a piston and bellows assembly which will activate a potentiometer. Measuring the resultant voltage should therefore give an indication of the pressure within the rocket chamber.

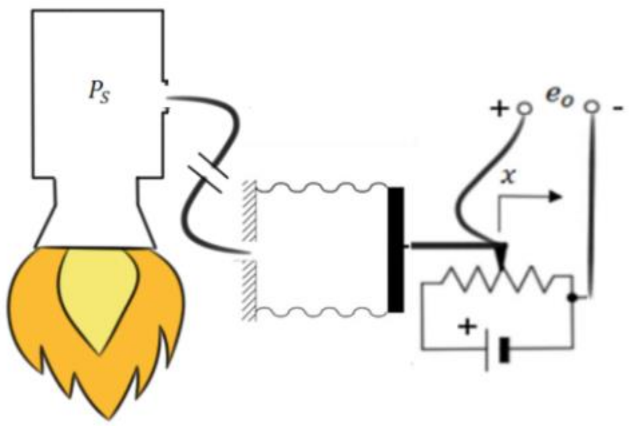



***Figure 1:** Rocket chamber pressure monitoring concept*

By decomposing the design into four sub-systems we can analyse the whole. The four systems are to be classified as:

System 1 = Pressure / Hydraulic
System 2 = Hydraulic / Mechanical
System 3 = Mechanical / Electrical
System 4 = Compensation

## Nomenclature:

Note that the following values have been suggested for the concept scheme:

| | Value | Units | Description |
|---|---|---|---|
| $A$ | 2 | $in^2$ | Area of piston |
| $b$ | 5 | $lb\ s/in$ | Damping |
| $C$ | 0.019 | - | Fluid capacitance |
| $E$ | 10 | $v$ | Voltage |
| $I$ | 110.5243 | - | Fluid inertance |
| $k$ | 5000 | $lb/in$ | Spring stiffness |
| $L$ | 0.6 | $in$ | Bellow length |
| $m$ | 0.1 | $lb$ | Mass |
| $P$ | 0 – 900 | $psi$ | Pressure |
| $P_{atm}$ | 14.6959 | $psi$ | Atmospheric pressure |
| $R$ | 207.7927 | - | Fluid resistance |
| $Q$ | TBD | $ft^3/s$ | Fluid flow rate |
| $x$ | TBD | $in$ | Distance |
| $y$ | TBD | $v$ | Response |
| $\beta$ | TBD | ul | Compensation |

## Subsystem 1: Pressure / Hydraulic

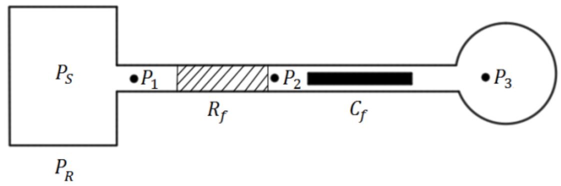



***Figure 2:** System 1*

For the fluid system pressure provided by the rocket assembly is the input to the system whilst the output is the volumetric flow rate of the liquid that enters into the piston assembly.

On this basis using lumped parameter elements:

$$P_{12} = R_f Q_R \quad [1]$$

$$P_{23} = I\dot{Q}_R \quad [2]$$

$$Q_R = C_f \dot{P}_{3R} \quad [3]$$

The fluid resistance[1], inertance[2] and capacitance[3] of the system are defined as:

$$R_f = \frac{128\mu L}{\pi d^4} \quad [4]$$

$$I = \frac{\rho L}{A} \quad [5]$$

$$C_f = \frac{A_p^2}{k_s} \quad [6]$$

1. Circular pipe with laminar flow
2. Frictionless incompressible flow in uniform passage
3. Use of a piston

All pressures will equalise hence:

$P_S = P_{12} + P_{23} + P_{3R}$ [7]

Rearranging to solve for the highest derivative:

$I\dot{Q}_R = P_S - R_f Q_R + P_{3R}$ [8]

$P_S = R_f Q_R + I\dot{Q}_R + P_{3R}$ [9]

$\dot{Q}_R = \frac{1}{I} P_S - \frac{1}{I} R_f Q_R + \frac{1}{I} P_{3R}$ [10]

We can rearrange **[7]** and perform a substitution to arrive at:

$C_f \ddot{P}_{3R} + R_f C_f \dot{P}_{3R} + P_{3R} = P_s$ [11]

**<u>Subsystem 2</u>: Hydraulic / Mechanical**

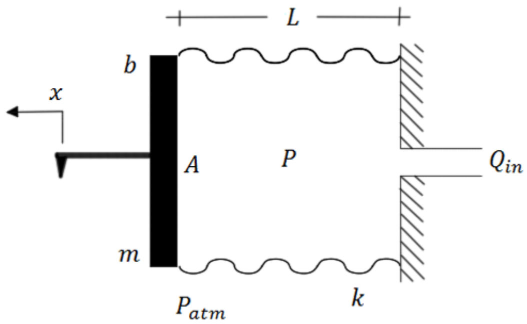



***Figure 3:*** *System 2*

System 2 is a mixed hydraulic and mechanical system.

The input for System 2 is the volume flow rate from System 1.

The output of the System 2 is the resultant linear displacement of mass attached to the bellows.

On this basis:

$\dot{P} = \frac{\beta}{v_o + Ax}(Q_{in} - A\dot{x})$ [12]

$m\ddot{x} + b\dot{x} + kx = (P - P_{atm})$ [13]

Rearranging to solve for the highest derivative gives:

$\ddot{x} = \frac{1}{m}\left((P - P_{atm})A - b\dot{x} - kx\right)$ [14]

**<u>Subsystem 3</u>: Mechanical / Electrical**

System 3 is a mixed mechanical and electrical system.

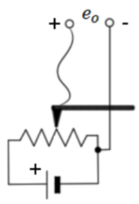



***Figure 4:*** *System 3*

The input is the linear displacement provided by the mass of System 2.

This displacement will change the resistance of the linear variable resistor.

This in turn modifies the output voltage with respect to the known battery voltage, hence:

$e_o = \left(\frac{E}{x_{max}}\right) x(t)$ [15]

**<u>Subsystem 4</u>: Compensation**

As we have an uncertainty to the final value of the electrical output we require an adjustment factor that can be used to tune the system response, hence:

$y = \beta[e_o]$ [16]

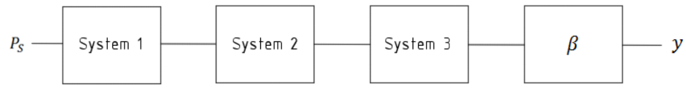



***Figure 5:*** *System 4*

**<u>System Model</u>:**

The equations derived for the four subsystems were translated into Modelica language with a step function used to represent the change of pressure within the rocket chamber.

The coding used for the simulation can be found in Appendix A.

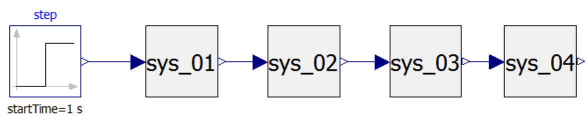



***Figure 6:*** *Modelica System*

**Results:**

After progressively tuning the slope value in subsystem 4 the system response to a step input of 500 psi after 1 second is illustrated below:

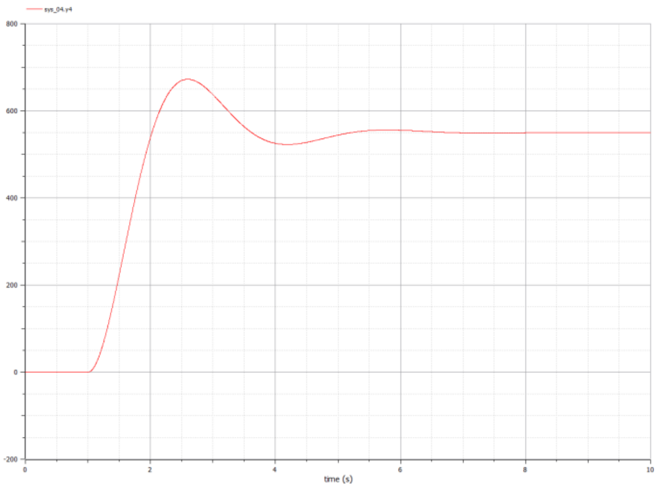



***Figure 7:*** *System response*

We see that we have a small overshoot and then the measurement settles down within 6 seconds to the correct value of 500.00 psi.

Note that when changing the input value to random pressure values (201 psi, 474 psi etc.) we see a direct 1:1 correlation with the response (201.00, 474.00 etc.) after the 6 seconds of settling time has elapsed.

**Conclusion:**

It would appear that the concept of measuring the pressure within the rocket chamber using a long fluid filled line is indeed feasible.

However with the design parameters studied in this paper there is a 6 second settling time that must be allowed for when encountering rapid pressure changes.

If the pressure within the rocket chamber is relatively stable and only needs to be measured at long intervals then the settling time issue would not be problematic.

If continuous monitoring of the rocket motor is required and rapid pressure changes are possible then a different approach to the measurement problem would be recommended.

**END OF DOCUMENT**

## Appendix A

```
block Sys_01
Modelica.Blocks.Interfaces.RealInput u1
annotation(…);
Modelica.Blocks.Interfaces.RealOutput y1
annotation(…);
Real c=0.0019;
Real i=110.5243;
Real r=207.7927;
Real ref=14.6959;
Real p3r_dot_dot;
Real p3r_dot;
Real p3r;
Real p3;
Real q;
equation
q=p3r_dot/(1/c);
p3=p3r-ref;
p3r_dot=der(p3r);
p3r_dot_dot=der(p3r_dot);
p3r_dot_dot=(1./(c*i))*(u1-(p3r)-r*c*p3r_dot);
y1=q;
annotation(…);
end Sys_01;
```

```
block Sys_02
Modelica.Blocks.Interfaces.RealInput y1
annotation(…);
Modelica.Blocks.Interfaces.RealOutput y2
annotation(…);
Real m = 0.1;
Real A = 2;
Real b = 5;
Real k = 5000;
Real x;
Real x_dot;
Real x_dot_dot;
Real P;
Real Pdot;
Real Patm = 15;
Real Beta=320000;
Real vzero = 0.01;
equation
x_dot=der(x);
x_dot_dot=der(x_dot);
x_dot_dot=(1/m)*((P-Patm)*A-b*x_dot-k*x);
Pdot=der(P);
Pdot=Beta/(vzero+A*x)*(y1-A*x_dot);
y2=x;
annotation(…);
end Sys_02;
```

```
block Sys_03
Modelica.Blocks.Interfaces.RealInput y2
annotation(…);
Modelica.Blocks.Interfaces.RealOutput y3
annotation(…);
Real E = 9;
Real xmax = 1;
y3 = (E/xmax)*y2;
annotation(…);
end Sys_03;
```

```
block Sys_04
Modelica.Blocks.Interfaces.RealInput y3
annotation(…);
Modelica.Blocks.Interfaces.RealOutput y4
annotation(…);
parameter Real slope = 117.2;
equation
y4 = y3 * slope;
annotation(…);
end Sys_04;
```