Investigation of Damped Oscillators

Izaak van Dongen

November 5, 2018

```r
library(ggplot2)

U_s = 3
U_t = 0.1

s_0s = c(654, 656, 608, 567)
taus = c(30, 19, 17, 10)

cumul_df = data.frame()
s_dfs = c()
per_models = c()

for (i in 1:4) {
    message(paste("Analysing Experiment", i))
    s_df <- read.table(paste0("data/", i, ".csv"), sep=",", header=TRUE)
    s_df$n <- seq_along(s_df$t)
    s_df$i <- i
    s_dfs[[i]] <- s_df
    cumul_df <- rbind(cumul_df, s_df)
    print(s_df)

    t_0 = head(s_df$t, n=1)
    s_0 = s_0s[i]
    A = head(s_df$s, n=1) - s_0
    m_exp <- nls(s ~ s_0 + A * exp(-(t - t_0) / tau),
                 start=list(s_0 = s_0, A = A, tau = taus[i]), data=s_df)
    coef_exp <- coef(summary(m_exp))
    print(coef_exp)

    m_per <- lm(t ~ n, data=s_df)
    coef_per <- coef(summary(m_per))
    print(coef_per)
    per_models[[i]] <- m_per

    message(paste("So, Q = pi * tau / T =",
                  pi * coef_exp["tau", "Estimate"]
                     / coef_per["n", "Estimate"]))

    print(qplot(t, s, data=s_df,
          main=paste0("Local maximum displacement of damped oscillator (",
                      i, ")"),
```

```
            xlab="t / s", ylab="s / mm") +
            geom_errorbar(data=s_df, mapping=aes(ymin=s-U_s, ymax=s+U_s)) +
            geom_errorbarh(data=s_df, mapping=aes(xmin=t-U_t, xmax=t+U_t)) +
            theme(panel.grid.minor = element_line(colour="gray", size=0.4),
                  panel.grid.major = element_line(colour="gray", size=1),
                  panel.background = element_blank()) +
            geom_line(aes(y = predict(m_exp)), color="steelblue", size=1))
}

p_plot <- ggplot(cumul_df, aes(x = n, y = t)) +
    geom_errorbar(aes(ymin=t-U_t, ymax=t+U_t, color=factor(i))) +
    geom_errorbarh(aes(height=2, xmin=n, xmax=n, color=factor(i))) +
    labs(x="n", y="t / s",
         title="Period of oscillation of different damped oscillators") +
    theme(panel.grid.minor = element_line(colour="gray", size=0.4),
          panel.grid.major = element_line(colour="gray", size=1),
          panel.background = element_blank()) +
    scale_color_discrete(name = "Experiment") +
    geom_point(aes(color=factor(i)))

for (i in 1:4) {
    p_plot <- p_plot + geom_line(y = predict(per_models[[i]]), data=s_dfs[[i
    ↪  ]],
                                 aes(color=factor(i)))
}

p_plot
```

Listing 1: Source code of the program `analyse.r`.

```
Analysing Experiment 1
       t   s  n i
1   1.16 700  1 1
2   3.79 698  2 1
3   6.36 697  3 1
4   8.92 694  4 1
5  11.49 691  5 1
6  14.08 687  6 1
7  16.72 683  7 1
8  19.28 682  8 1
9  21.88 681  9 1
10 24.48 677 10 1
11 27.04 675 11 1
12 29.61 673 12 1
13 32.24 671 13 1
14 34.77 670 14 1
15 37.37 670 15 1
16 39.93 668 16 1
17 42.56 669 17 1
18 45.09 669 18 1
19 47.66 670 19 1
```

```
20 50.26 671 20 1
21 52.85 671 21 1
22 55.42 670 22 1
23 57.98 667 23 1
24 60.58 666 24 1
25 63.18 663 25 1
26 65.74 659 26 1
27 68.31 657 27 1
28 70.81 659 28 1
     Estimate Std. Error    t value     Pr(>|t|)
s_0 657.10894   3.030002 216.867506 1.841782e-42
A    44.67239   2.619214  17.055645 2.786917e-15
tau  31.42320   5.368805   5.852923 4.183809e-06
             Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) -1.373810 0.0122735232 -111.9328 1.998742e-36
n            2.581346 0.0007394494 3490.9036 2.941279e-75
So, Q = pi * tau / T = 38.2431831229393
Analysing Experiment 2
       t   s  n i
1   2.17 717  1 2
2   4.80 706  2 2
3   7.43 696  3 2
4  10.06 690  4 2
5  12.69 685  5 2
6  15.33 679  6 2
7  17.92 675  7 2
8  20.56 671  8 2
9  23.19 668  9 2
10 25.78 665 10 2
11 28.42 663 11 2
12 31.01 661 12 2
13 33.64 659 13 2
14 36.28 658 14 2
15 38.87 656 15 2
16 41.47 655 16 2
17 44.10 653 17 2
18 46.70 654 18 2
19 49.30 654 19 2
20 51.96 651 20 2
21 54.49 650 21 2
22 57.16 650 22 2
23 59.76 649 23 2
24 62.35 648 24 2
25 64.95 646 25 2
26 67.58 645 26 2
27 70.15 644 27 2
     Estimate Std. Error   t value     Pr(>|t|)
s_0 645.22575  0.7677368 840.42567 3.816744e-55
A    69.44874  1.0130796  68.55211 4.784322e-29
tau  19.05933  0.7635008  24.96308 1.109966e-18
              Estimate   Std. Error    t value     Pr(>|t|)
```

```
(Intercept) -0.3784615 0.0135172809  -27.99835 2.140169e-20
n            2.6146520 0.0008437322 3098.91234 2.469633e-71
So, Q = pi * tau / T = 22.9004293181007
Analysing Experiment 3
       t   s  n i
1   3.43 644  1 3
2   6.12 641  2 3
3   8.75 638  3 3
4  11.39 635  4 3
5  14.12 631  5 3
6  16.75 626  6 3
7  19.45 622  7 3
8  22.14 618  8 3
9  24.81 614  9 3
10 27.47 611 10 3
11 30.04 609 11 3
12 32.74 607 12 3
13 35.40 606 13 3
14 38.10 605 14 3
15 40.73 604 15 3
16 43.43 604 16 3
17 46.06 604 17 3
18 48.69 604 18 3
19 51.29 605 19 3
20 54.02 604 20 3
21 56.55 605 21 3
22 59.05 604 22 3
23 61.78 605 23 3
24 64.58 604 24 3
25 67.21 604 25 3
26 69.87 603 26 3
27 72.54 602 27 3
     Estimate Std. Error   t value     Pr(>|t|)
s_0 600.92863   1.108944 541.89272 1.430598e-50
A    47.57511   1.654020  28.76332 4.115333e-20
tau  17.66282   1.601854  11.02648 7.053199e-11
            Estimate  Std. Error    t value     Pr(>|t|)
(Intercept) 0.850000 0.028677098   29.64038 5.359208e-21
n           2.654921 0.001789989 1483.20462 2.470542e-63
So, Q = pi * tau / T = 20.9005766245794
Analysing Experiment 4
       t   s  n i
1   4.97 616  1 4
2   7.73 600  2 4
3  10.46 591  3 4
4  13.16 585  4 4
5  15.89 580  5 4
6  18.65 576  6 4
7  21.35 575  7 4
8  24.05 575  8 4
9  26.78 576  9 4
```

```
10 29.55 576 10 4
11 32.31 574 11 4
12 35.04 574 12 4
13 37.71 575 13 4
14 40.50 575 14 4
15 43.13 576 15 4
16 45.87 574 16 4
17 48.53 572 17 4
18 51.23 572 18 4
19 53.93 570 19 4
20 56.66 570 20 4
       Estimate Std. Error    t value     Pr(>|t|)
s_0 573.195923  0.5139199 1115.34105 8.573733e-43
A    42.720434  1.5842461   26.96578 2.160245e-15
tau   6.095009  0.4581154   13.30453 2.043506e-10
              Estimate  Std. Error    t value     Pr(>|t|)
(Intercept) 2.305053 0.023253180   99.12849 4.239933e-26
n           2.720947 0.001941138 1401.72798 8.428421e-47
So, Q = pi * tau / T = 7.0372680956032
```

Listing 2: Output of `analyse.r` (1) when run.

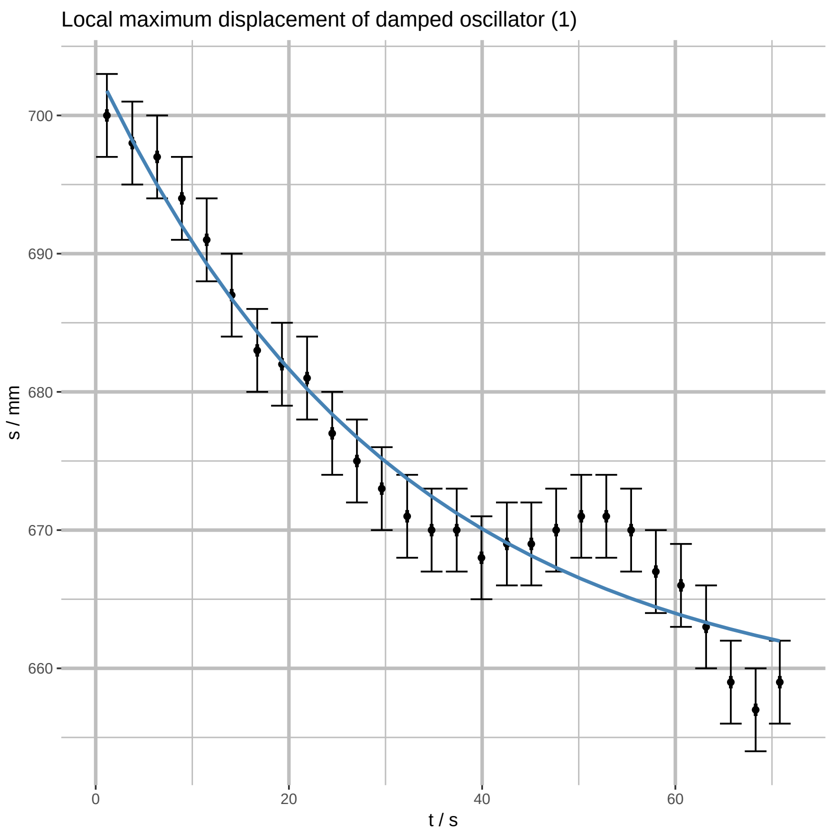

Figure 1: Plot of maximum displacements in experiment 1.

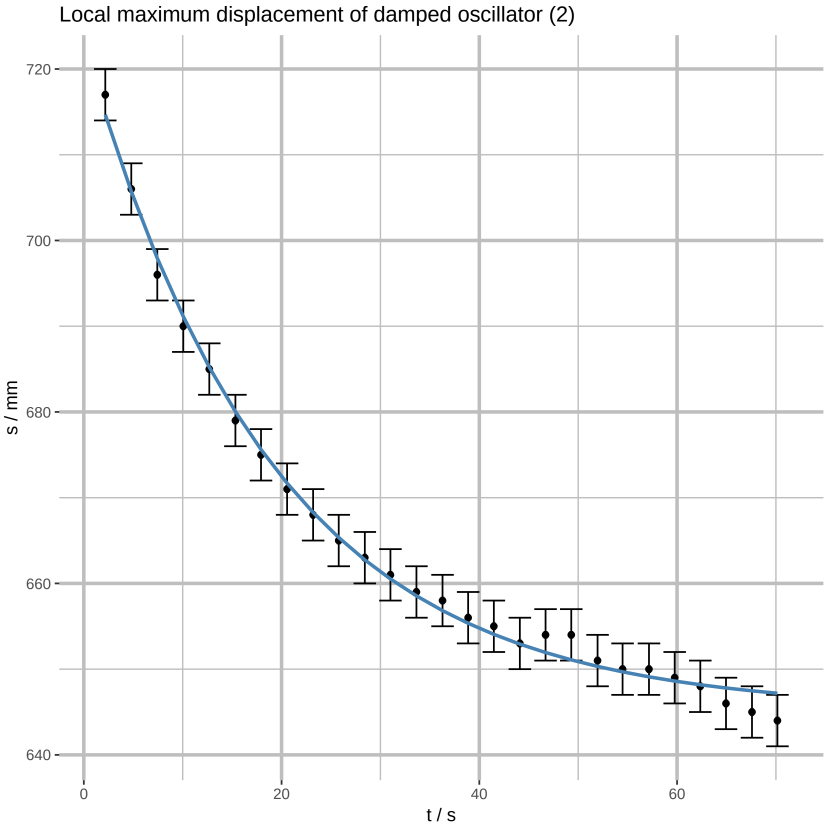

Figure 2: Plot of maximum displacements in experiment 2.

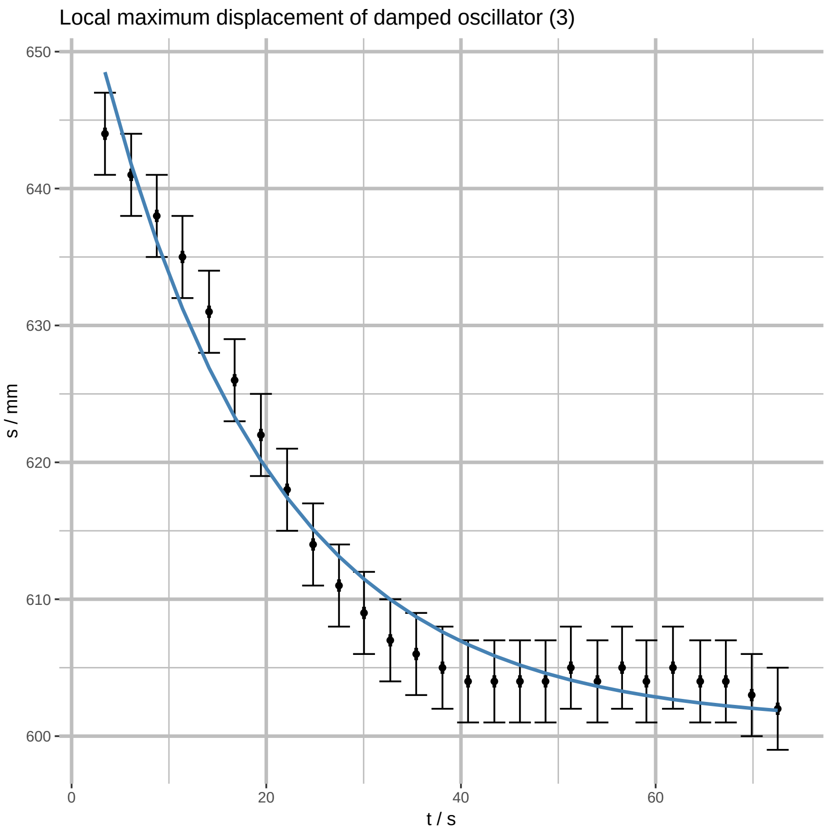

Figure 3: Plot of maximum displacements in experiment 3.

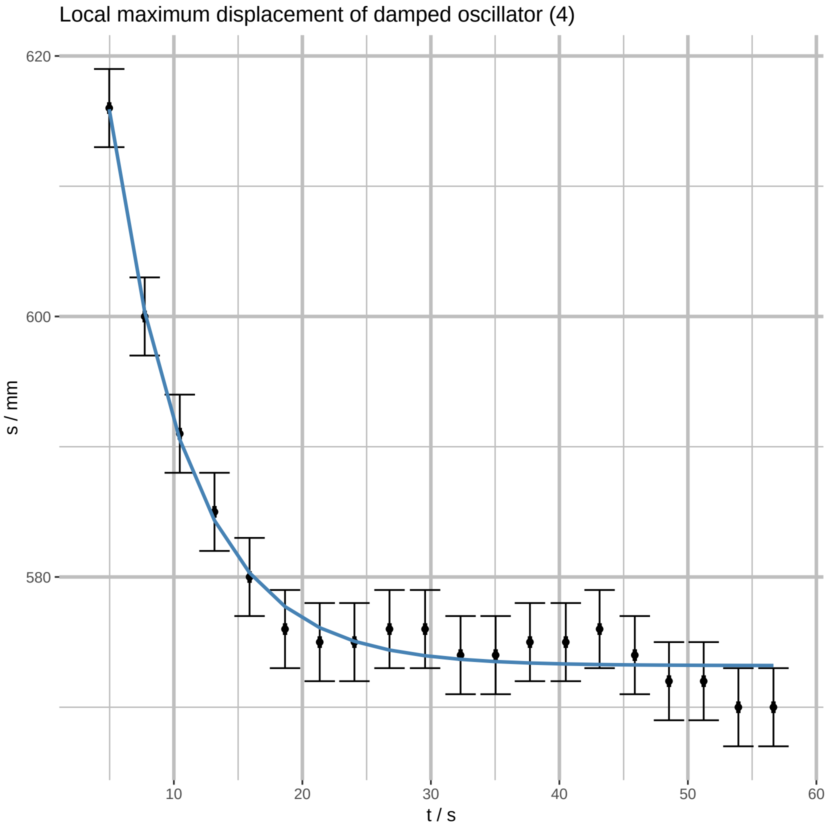

Figure 4: Plot of maximum displacements in experiment 4.

We did four experiments, filming each, and then used the video to find the time and value of displacement at each local point of maximum displacement. In a damped system, $s = Ae^{-\frac{t}{\tau}} \cos \omega t$, so by plotting the maxima we should get a plot of $s = Ae^{-\frac{t}{\tau}}$. However this formula is somewhat oversimplified as it assumes the first peak is when $t = 0\,\mathrm{s}$, and that s is measured from the equilibrium point. Therefore, the fully qualified model is $s = s_0 + Ae^{-\frac{t-t_0}{\tau}}$.

We did directly measure values for s_0, but there would be considerable uncertainties in these as the whole system was somewhat volatile and hard to measure. Because of this uncertainty, A is in a similar boat. t_0, on the other hand, we know with (un)certainty, as this is simply the first t value we decide to plot.

In view of these observations, I decided to use the measurements for s_0 merely as starting values for the model, but to set t_0 as a known constant, in the model in Listing 1. This model was used to find τ.

I also worked out the period of oscillation T for each experiment by finding the gradient of t_n with respect to n, where t_n is the time of the nth observed peak. I took this approach as this would minimise the uncertainty in the final answer, and graphical presentation of each set of t values would be challenging, enjoyable, and allow an inspection of whether or not, for example, T is variable.

Fortunately, as you can see in Figure 5, this approach produces some very straight lines with mildly differing gradients.

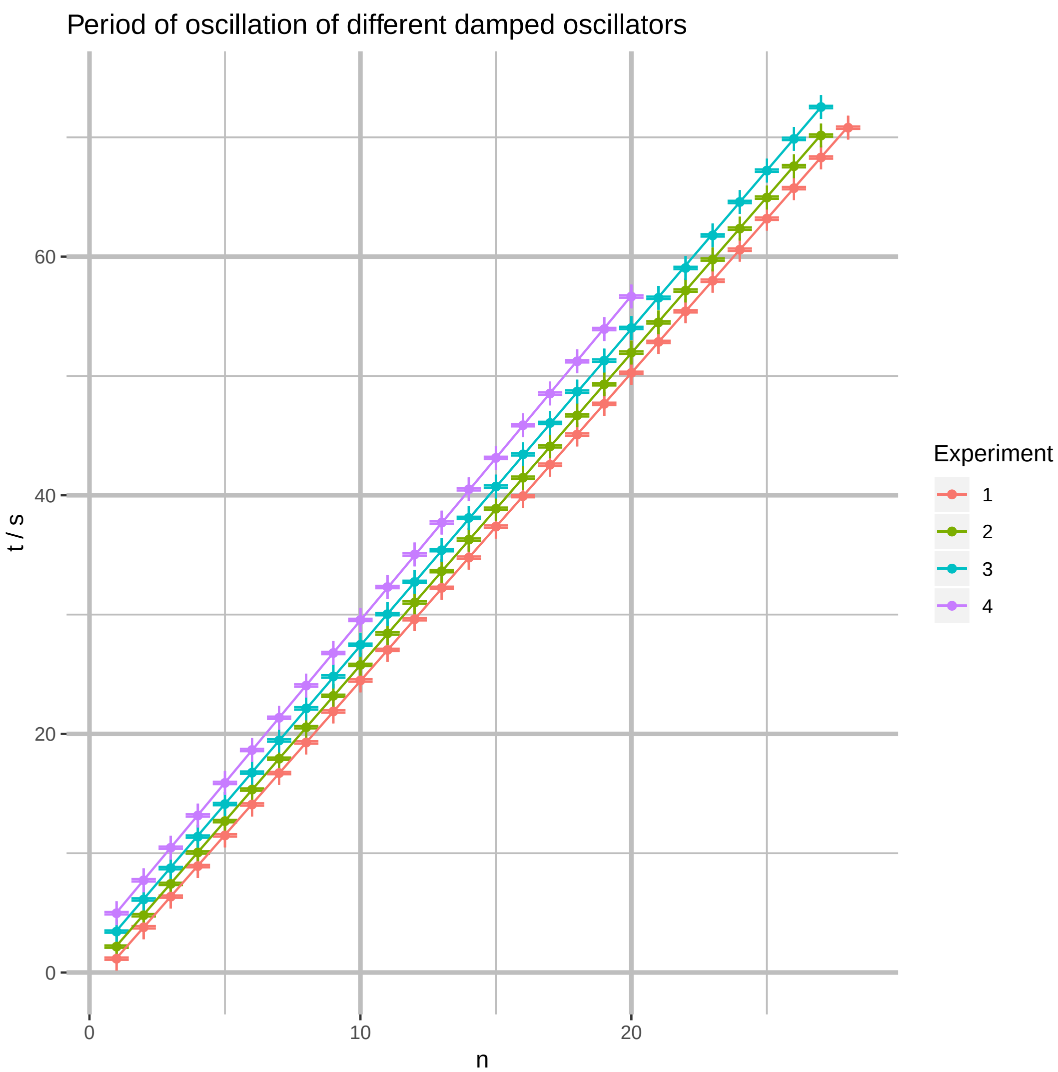

Figure 5: Plot of t_n/s, the time of the nth peak, against n.

From these two results, t and τ, I can now calculate the Q-factor as $Q = \frac{\pi\tau}{T}$. The results are shown in Listing 2, but are presented below for your convenience, together with the diameter of the damper used.

Experiment	$\boldsymbol{d}$/mm	$\boldsymbol{\tau}$/s	$\boldsymbol{T}$/s	$\boldsymbol{Q}$
1	300	31.42	2.58	38.24
2	340	19.06	2.61	22.90
3	405	17.66	2.65	20.90
4	450	6.10	2.72	7.04

This would suggest that Q, and τ all decrease when the area of the damper increases, while T also increases. This makes sense, and fits with the theory I know.

In evaluation, the biggest problem we had was an interfering oscillation from the damping card, which rotated about the point at which it was fixed to the mass. This resulted in a sinusoidy shape being added to some of our decay graphs, which made analysis of our data a bit trickier. The way to fix this would probably be to devise some way to more rigidly attach the damper to the mass. This problem was possible to work around with some of the data, but somewhat jeopardised other parts.