Ballistic Trajectory with Drag

Ken Harmon

2020 May 27

https://iopscience.iop.org/article/10.1088/1742-6596/1287/1/012032

$$V(\theta) = \frac{V_0 \cos\theta_0}{\cos\theta\sqrt{1+kV_0^2\cos^2\theta_0(f(\theta_0)-f(\theta))}},$$
$$f(\theta) = \frac{\sin\theta}{\cos^2\theta} + \ln\tan\left(\frac{\theta}{2}+\frac{\pi}{4}\right),$$

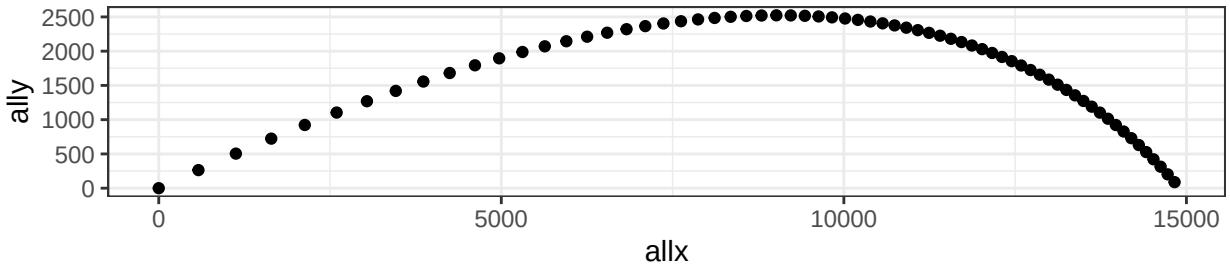
(2)

$$x = x_0 - \frac{1}{g}\int_{\theta_0}^{\theta} V^2 d\theta, y = y_0 - \frac{1}{g}\int_{\theta_0}^{\theta} V^2\tan\theta d\theta,$$
$$t = t_0 - \frac{1}{g}\int_{\theta_0}^{\theta} \frac{V}{\cos\theta} d\theta.$$

Figure 1: Motion Formulas

```
V0 <- 671 # initial velocity in m/s for M795 with M232A1 3H
am0 <- 442.1 # QE in mils for a level 9000 m shot
th0 <- am0 * pi / 3200 # initial angle in radians
amt <- -697 # terminal angle in mils
tht <- amt * pi / 3200 #Terminal angle in radians
x0 <- 0 #Initial x
y0 <- 0 # initial y
t0 <- 0 # initial time
g <- 9.80665 # gravitational force in m/s/s
# All functions are in terms of angle of the trajectory

k <- .0000019 # is the drag constant

#Lets build a table along the trajectory

TOF <- 43 #time of flight
```

```
ths <- seq(th0, tht*.9, by = ((tht - th0)/(TOF*2)))

f0 = sin(th0)/(cos(th0))^2 + log(tan(th0/2+pi/4))
allf <- NA
allV <- NA
allx <- NA
ally <- NA
allt <- NA


for (th in ths) {
  f = sin(th)/(cos(th))^2 + log(tan(th/2+pi/4))
  allf <- c(allf,f)
  V = V0*cos(th0)/(cos(th)*sqrt(1+k*(V0*cos(th0))^2*(f0-f)))
  allV <- c(allV,V)
  xint <- integrate(function(x)
    {(V0*cos(th0)/(cos(x)*sqrt(1+k*(V0*cos(th0))^2*(f0-f))))^2},th0,th)
  x <- x0 - 1/g*as.numeric(xint[1])
  allx <- c(allx,x)
  yint <- integrate(function(x)
    {tan(x)*(V0*cos(th0)/(cos(x)*sqrt(1+k*(V0*cos(th0))^2*(f0-f))))^2},th0,th)
  y <- y0 - 1/g*as.numeric(yint[1])
  ally <- c(ally,y)
  tint <- integrate(function(x)
    {V0*cos(th0)/((cos(x))^2*sqrt(1+k*(V0*cos(th0))^2*(f0-f)))},th0,th)
  t <- t0 - 1/g*as.numeric(tint[1])
  allt <- c(allt,t)
}
traj <- data.frame(cbind(allf,allV,allt,allx,ally))
trajp <- traj[2:nrow(traj),]%>%filter(ally>=0)
trajp$theta <- as.numeric(0)
trajp[1,6] <- am0
for (i in 2:(nrow(trajp)-1)) {
  trajp[i,6] <- atan((trajp[i+1,5]-trajp[i-1,5])/(trajp[i+1,4]-trajp[i-1,4]))*3200/pi
}
i <- nrow(trajp)
trajp[i,6] <- atan((trajp[i,5]-trajp[i-1,5])/(trajp[i,4]-trajp[i-1,4]))*3200/pi
trajp %>% ggplot(aes(allx,ally)) + geom_point() +
    coord_fixed(ratio = 1)
```




```
colnames(trajp) <- c("f","V m/s","t s","x m","y m","mils")
trajp
```

```
##              f    V m/s       t s        x m        y m        mils
## 1   0.959215209 671.0000  0.0000000     0.0000    0.00000  442.100000
## 2   0.924701171 659.0751  0.9632186   579.3932  264.00445  428.931788
## 3   0.890784881 647.8060  1.8933382  1125.8274  504.25350  415.202251
## 4   0.857435797 637.1415  2.7928599  1642.4097  723.02462  401.159180
## 5   0.824624791 627.0359  3.6640275  2131.8628  922.30895  386.807905
## 6   0.792324053 617.4485  4.5088618  2596.5828 1103.85446  372.153581
## 7   0.760507004 608.3428  5.3291890  3038.6868 1269.20156  357.201217
## 8   0.729148208 599.6859  6.1266653  3460.0528 1419.71264  341.955698
## 9   0.698223294 591.4479  6.9027968  3862.3524 1556.59669  326.421809
## 10  0.667708891 583.6021  7.6589582  4247.0787 1680.92990  310.604253
## 11  0.637582550 576.1239  8.3964068  4615.5694 1793.67308  294.507680
## 12  0.607822689 568.9912  9.1162964  4969.0269 1895.68639  278.136702
## 13  0.578408529 562.1836  9.8196882  5308.5349 1987.74181  261.495914
## 14  0.549320038 555.6826 10.5075605  5635.0726 2070.53385  244.589915
## 15  0.520537882 549.4712 11.1808176  5949.5275 2144.68874  227.423323
## 16  0.492043372 543.5337 11.8402969  6252.7055 2210.77227  210.000796
## 17  0.463818419 537.8559 12.4867758  6545.3401 2269.29662  192.327046
## 18  0.435845491 532.4245 13.1209775  6828.1009 2320.72629  174.406857
## 19  0.408107568 527.2273 13.7435758  7101.5997 2365.48318  156.245096
## 20  0.380588106 522.2529 14.3551999  7366.3973 2403.95110  137.846730
## 21  0.353270997 517.4911 14.9564384  7623.0080 2436.47969  119.216835
```

```
## 22  0.326140536 512.9319 15.5478429  7871.9051 2463.38783  100.360614
## 23  0.299181382 508.5665 16.1299313  8113.5243 2484.96668   81.283397
## 24  0.272378530 504.3865 16.7031902  8348.2675 2501.48234   61.990659
## 25  0.245717278 500.3841 17.2680784  8576.5061 2513.17821   42.488025
## 26  0.219183195 496.5521 17.8250284  8798.5836 2520.27706   22.781273
## 27  0.192762093 492.8836 18.3744487  9014.8186 2522.98291    2.876343
## 28  0.166440000 489.3724 18.9167261  9225.5063 2521.48263  -17.220660
## 29  0.140203130 486.0126 19.4522269  9430.9211 2515.94746  -37.503466
## 30  0.114037860 482.7987 19.9812988  9631.3182 2506.53431  -57.965638
## 31  0.087930699 479.7255 20.5042721  9826.9350 2493.38689  -78.600573
## 32  0.061868264 476.7881 21.0214612 10017.9929 2476.63683  -99.401504
## 33  0.035837260 473.9822 21.5331657 10204.6983 2456.40456 -120.361507
## 34  0.009824445 471.3033 22.0396713 10387.2440 2432.80019 -141.473500
## 35 -0.016183385 468.7475 22.5412510 10565.8098 2405.92430 -162.730259
## 36 -0.042199426 466.3112 23.0381662 10740.5642 2375.86854 -184.124418
## 37 -0.068236891 463.9907 23.5306668 10911.6646 2342.71635 -205.648485
## 38 -0.094309029 461.7829 24.0189930 11079.2581 2306.54345 -227.294851
## 39 -0.120429159 459.6846 24.5033751 11243.4829 2267.41838 -249.055803
## 40 -0.146610684 457.6929 24.9840347 11404.4680 2225.40293 -270.923538
## 41 -0.172867125 455.8051 25.4611849 11562.3343 2180.55258 -292.890176
## 42 -0.199212145 454.0186 25.9350312 11717.1953 2132.91684 -314.947782
## 43 -0.225659572 452.3311 26.4057719 11869.1571 2082.53963 -337.088376
## 44 -0.252223428 450.7404 26.8735987 12018.3190 2029.45956 -359.303957
## 45 -0.278917957 449.2442 27.3386967 12164.7742 1973.71018 -381.586515
## 46 -0.305757654 447.8406 27.8012454 12308.6096 1915.32029 -403.928056
## 47 -0.332757293 446.5279 28.2614186 12449.9067 1854.31410 -426.320620
## 48 -0.359931958 445.3041 28.7193851 12588.7416 1790.71151 -448.756299
## 49 -0.387297075 444.1679 29.1753089 12725.1851 1724.52819 -471.227256
## 50 -0.414868445 443.1175 29.6293495 12859.3034 1655.77585 -493.725747
## 51 -0.442662279 442.1516 30.0816621 12991.1579 1584.46230 -516.244143
## 52 -0.470695235 441.2690 30.5323981 13120.8056 1510.59165 -538.774940
## 53 -0.498984452 440.4683 30.9817053 13248.2992 1434.16438 -561.310788
## 54 -0.527547596 439.7485 31.4297278 13373.6874 1355.17745 -583.844504
## 55 -0.556402898 439.1085 31.8766068 13497.0150 1273.62443 -606.369087
## 56 -0.585569201 438.5474 32.3224803 13618.3226 1189.49552 -628.877740
## 57 -0.615066004 438.0643 32.7674835 13737.6476 1102.77765 -651.363881
## 58 -0.644913517 437.6583 33.2117491 13855.0233 1013.45456 -673.821162
## 59 -0.675132710 437.3288 33.6554072 13970.4798  921.50680 -696.243475
## 60 -0.705745372 437.0751 34.0985857 14084.0435  826.91182 -718.624974
## 61 -0.736774171 436.8966 34.5414105 14195.7374  729.64395 -740.960076
## 62 -0.768242716 436.7928 34.9840050 14305.5812  629.67448 -763.243480
## 63 -0.800175631 436.7632 35.4264913 14413.5912  526.97163 -785.470168
## 64 -0.832598624 436.8075 35.8689894 14519.7805  421.50060 -807.635419
## 65 -0.865538568 436.9252 36.3116177 14624.1586  313.22353 -829.734810
## 66 -0.899023587 437.1161 36.7544930 14726.7319  202.09956 -851.764223
## 67 -0.933083144 437.3800 37.1977308 14827.5033   88.08475 -862.725215
```