The Hill-Keller Model

Kurt Bryan and SIMIODE

A Mathematica notebook to fit the Hill-Keller model v'(t) = P - k*v(t) to the data from Usain Bolt's 2008 Beijing gold medal Olympic race. This is a minor extension of the worksheet "hill_keller_model1.nb" Here we fit both P and k.

The Data: Here is the data for Usain Bolt's 2008 Beijing race. The data consists of times for 0, 10, 20, ..., 100 meters and t is the corresponding split time.

```
In[1]:= bolttimes = {0.165, 1.85, 2.87, 3.78, 4.65, 5.50, 6.32, 7.14, 7.96, 8.79, 9.69}
```

A quick plot, after putting the data into (time, distance) pairs:

```
In[2]:= bolt = Table[{bolttimes[[i]], 10 * (i - 1)}, {i, 1, 11}]

In[3]:= plt1 = ListPlot[bolt]
```

Solving The ODE: Solve the Hill-Keller ODE, both "P and "k" as unknown. Also use initial condition v(0.165) = 0.

```
In[7]:= ode = v'[t] == P - k * v[t]
        t0 = 0.165;
        sol = DSolve[{ode, v[t0] == 0}, v, t]
```

Define this as a function of t

```
In[10]:= vsol = v /. sol[[1]]
```

Integrate to find position

```
In[11]:= dis = Integrate[vsol[tau], {tau, t0, t}]
```

Make this into a function of t

```
In[12]:= X[t_] = dis
```

Let's guess a value of P and k and plot X(t) with the data.

```
In[13]:= plt2 = Plot[X[t] /. {k → 1, P → 11}, {t, 0, 9.69}, PlotStyle → {Red}];

In[14]:= Show[plt1, plt2]
```

The Optimal Choice for k and P: Not bad, but we can do better by forming a sum of squares SS and minimizing

```
In[15]:= SS = Sum[(X[bolt[[i, 1]]] - bolt[[i, 2]])^2, {i, 1, 11}]
```

We already know that k somewhere around 1 and P around 11 yields a reasonable fit. Plot SS as a function of P and k to get a better estimate of where SS is minimized. Although plotting ln(SS) is even better:

In[17]:=
```
Plot3D[Log[SS], {k, 0.7, 1.1}, {P, 8, 12}]
```

Rotating the graph around shows k around 0.85 and P around 10.3 looks promising. So set d(SS)/dk = 0 and d(SS)/dP = 0 and use Mathematica's FindRoot command to find a good solution

In[25]:=
```
dSSdk = D[SS, k];
dSSdP = D[SS, P];
sol = FindRoot[{dSSdk == 0, dSSdP == 0}, {k, 1}, {P, 11}]
```

The residual is

In[28]:=
```
SS /. sol
```

Use the optimal k and P values to plot and compare to the data

In[29]:=
```
plt2 = Plot[X[t] /. sol, {t, 0, 9.69}, PlotStyle → {Red}];
Show[plt1, plt2]
```

Alternatively, we can use Mathematica's built-in routines to perform the minimization and achieve the same result.

In[31]:=
```
NMinimize[SS, {k, P}]
```