Approximation of a functional curve in a digital image

Adapted from a Mathcad Advisor file by Marco Asmad. Also see https://community.ptc.com/t5/PTC-Mathcad/Extract-data-from-chart/m-p/165695 by RichardJ.

We start with the curve image in gray scale:

Yellow boxes are the places where you can modify the values for scan your own curve.

```
M := Mean(image2rgb("imgPerformance.jpg"))
```

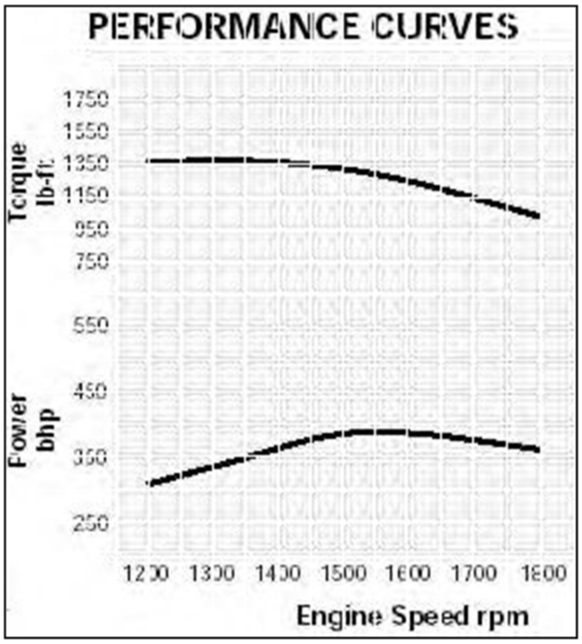
and make a validation of the data computing the average value of each row and column, then compares this value with kline which has represents the gray value selected to define the maximum "black" pixel.

```
clean(M#, k11, k12) := | [M := M#  Mc := M  R := 0  C := 0]
                       | [m := [1..rows(M)]  n := [1..cols(M)]]
                       | [R_m := Mean(row(M, m))  C_n := Mean(col(M, n))]
                       | [R C] := eval([ vec(k11 ≤ R ≤ k12)  vec(k11 ≤ C ≤ k12) ])
                       | for m ∈ [1..rows(M)]
                       |   for n ∈ [1..cols(M)]
                       |     if R_m ∧ C_n
                       |       continue
                       |     else
                       |       Mc_(m n) := 255
                       | Mc
```

```
k1 := 70        Mc := clean(M, k1, 255)
```

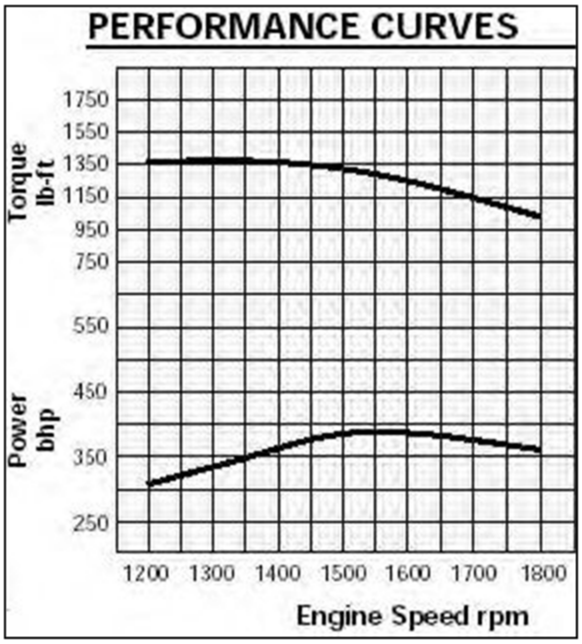
Try for different kl values until find a good one.

Cleaned data



Mc

Original data



M

Shorthand for boxes:

```
box(x, y) := | [a b c d] := [x_1  x_length(x)  y_1  y_length(y)]
             | stack(augment(a, y), augment(b, y), augment(x, c), augment(x, d))
```

With the matrix cleaned we can look for pixels from the function curve. With the following function we can scan the cleaned image matrix in the vertical and horizontal directions to find pixels that belong to the curve; the maximum opaque color is contained in the variable kcurve.

$$scan(M, k_{c1}, k_{c2}) := \left| \begin{array}{l} [\,nc := \mathrm{cols}(M)\;\; n := [1..\mathrm{length}(M)]\;\; A := 0\,] \\ v := \mathrm{col}\left(\mathrm{findrows}\left(\mathrm{eval}\left(\mathrm{augment}\left(\overrightarrow{k_{c1} \le M \le k_{c2}}_{n}, n\right)\right), 1, 1\right), 2\right) \\ [\,r\;\; c\,] := \mathrm{eval}\left(\left[\left[\,\overrightarrow{1 + \mathrm{trunc}\left(\frac{v-1}{nc}\right)}\;\; \overrightarrow{1 + \mathrm{mod}(v-1, nc)}\,\right]\right]\right) \\ \mathrm{eval}(\mathrm{csort}(\mathrm{augment}(c, \mathrm{rows}(M) - r + 1), 1)) \end{array} \right.$$

$\boxed{k_c := 140}$ $\quad Ms := scan(Mc, 0, k_c)$ Try for different kc values until find a good one.

Also isolate the curves selectring adequate ranges for torque, power and speed

Read range values here

$\boxed{\omega r := [72..266]}$

$\boxed{Tr := [192..256]}$

$\boxed{Pr := [64..128]}$

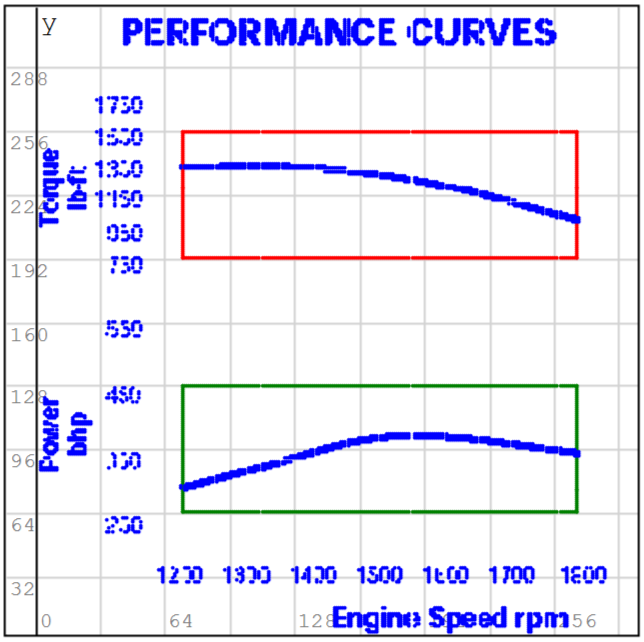
and the same true values for the boxes here

$\boxed{\omega v := [\,1200\;\; 1800\,]}$

$\boxed{Tv := [\,750\;\; 1550\,]}$

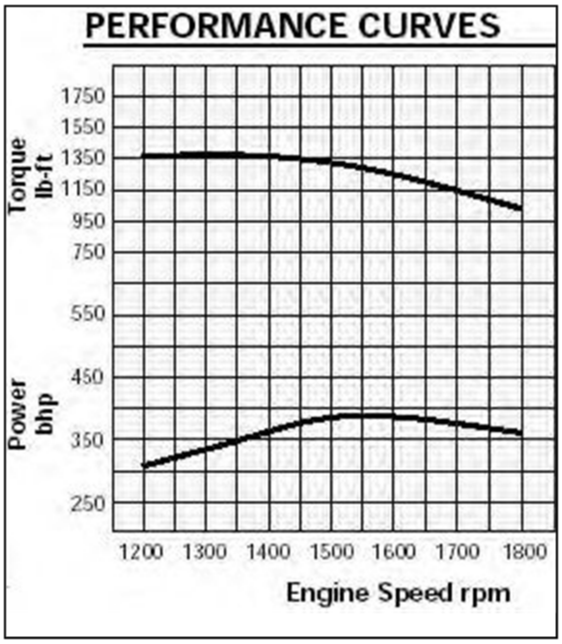
$\boxed{Pv := [\,250\;\; 450\,]}$

Scanned data



Ms
$box(\omega r, Tr)$
$box(\omega r, Pr)$

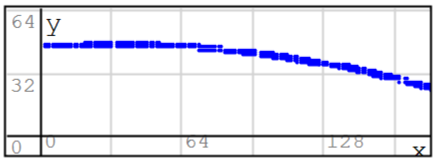
Original data



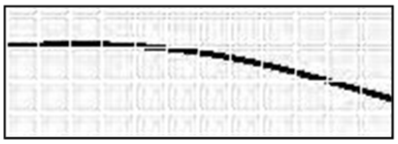
M

Now we convert into img coordinates:

$$\left| \begin{array}{l} McT := Mc_{\mathrm{reverse}(\mathrm{rows}(Mc) - Tr)\;\omega r} \\ \omega Ts := scan(McT, 0, k_c) \end{array} \right.$$

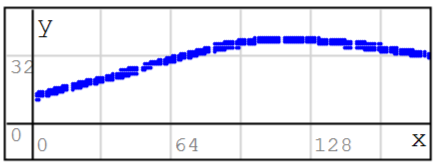


ωTs

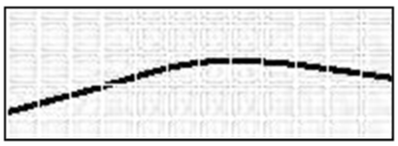


McT

$$\left| \begin{array}{l} McP := Mc_{\mathrm{reverse}(\mathrm{rows}(Mc) - Pr)\;\omega r} \\ \omega Ps := scan(McP, 0, k_c) \end{array} \right.$$



ωPs



McP

Shorthand for interpolation:

$$CInterp(XY, x) := \left| \begin{array}{l} [X\ Y] := [\mathrm{col}(XY,1)\ \ \mathrm{col}(XY,2)] \\ \mathrm{cinterp}(X, Y, x) \cdot \min(X) \le x \le \max(X) \end{array} \right.$$

Shorthand for plots:

$$plot(f, b, \sigma) := \begin{cases} f \\ \mathrm{augment}(b, \text{"o"}) \\ \mathrm{augment}(\sigma, \text{"."}, 1, \text{"\#550000FF"}) \end{cases}$$

This scale the scanned values

$$scale(xy, M, xv, yv) := \left| \begin{array}{l} X := xv_1 + \dfrac{xv_2 - xv_1}{\mathrm{cols}(M)} \cdot \mathrm{col}(xy, 1) \\ Y := yv_1 + \dfrac{yv_2 - yv_1}{\mathrm{rows}(M)} \cdot \mathrm{col}(xy, 2) \\ \mathrm{eval}(\mathrm{augment}(X, Y)) \end{array} \right.$$

Now average the set of Y values at each X value to define the final data points, then to fit a curve with only N points.

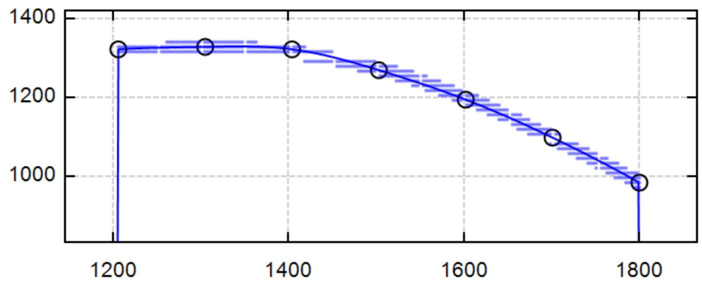
$$XYBar(XY, N) := \left| \begin{array}{l} [X\ Y] := [\mathrm{col}(XY,1)\ \ \mathrm{col}(XY,2)] \\ [k := 0\ \ n := 0\ \ \xi := 0\ \ \psi := 0] \\ \text{while } n < \mathrm{length}(X) \\ \quad \left| \begin{array}{l} [n := n+1\ \ h := n\ \ k := k+1] \\ \text{while } (n < \mathrm{length}(X)) \land (X_n = X_{n+1}) \\ \quad n := n+1 \\ [\psi_k := \mathrm{Mean}(Y_{[h..n]})\ \ \xi_k := X_h] \end{array} \right. \\ [n := [1..N]\ \ x := 0\ \ y := 0] \\ x := \min(X) + \dfrac{\max(X) - \min(X)}{N-1} \cdot (n-1) \\ y_n := \mathrm{cinterp}(\xi, \psi, x_n) \\ \mathrm{eval}(\mathrm{augment}(x, y)) \end{array} \right.$$

Choosing some few points we get

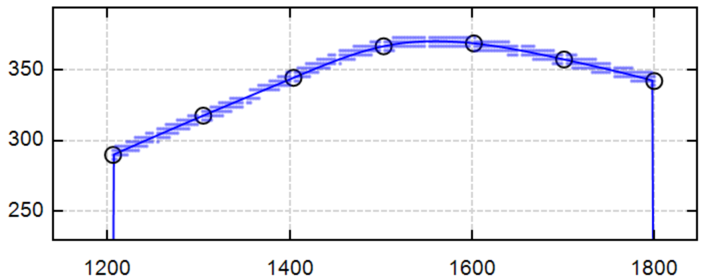
$N := 7$

$\omega T\sigma := scale(\omega Ts, McT, \omega v, Tv)$ $\qquad \omega T := XYBar(\omega T\sigma, N)$ $\qquad T(\omega) := CInterp(\omega T, \omega)$

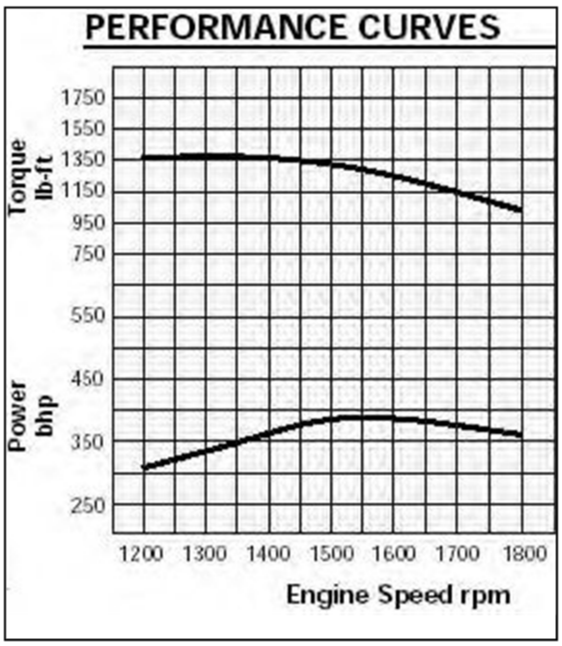
$\omega P\sigma := scale(\omega Ps, McP, \omega v, Pv)$ $\qquad \omega P := XYBar(\omega P\sigma, N)$ $\qquad P(\omega) := CInterp(\omega P, \omega)$



$plot(T(x), \omega T, \omega T\sigma)$



$plot(P(x), \omega P, \omega P\sigma)$



M

So, this two matrices it's all we need for reconstruct both curves

$$\omega T = \begin{bmatrix} 1206.1538 & 1322.3077 \\ 1305.1282 & 1328.4615 \\ 1404.1026 & 1322.266 \\ 1503.0769 & 1269.3798 \\ 1602.0513 & 1194.4763 \\ 1701.0256 & 1098.0378 \\ 1800 & 983.8462 \end{bmatrix} \qquad \omega P = \begin{bmatrix} 1206.1538 & 290 \\ 1305.1282 & 317.7898 \\ 1404.1026 & 344.5654 \\ 1503.0769 & 366.9128 \\ 1602.0513 & 368.9486 \\ 1701.0256 & 357.5773 \\ 1800 & 342.3077 \end{bmatrix}$$

Alvaro