Training stimulus is a number between 0 and 1

Samuel Leblanc, 2021

```
## Resistance training can be understood with Newtonian physics,
therefore making it possible to compute its main stimulus.
```

There are different approaches on how to divide training stimuli. The one that's most often used in the general public is:

- Strength or Neural;

- Hypertrophy or Mechanical Stress;

- Endurance or Metabolic stress.*

Hence creating a dichotomy between three separated classes. The obvious problem in that approach is limiting yourself to one stimulus, even if it is possible to work strength and hypertrophy at the same time, for example. We must apply nuance since there's a difference between doing a 1 RM (repetition maximum) and 6 RM, the first being almost exclusively for strength purposes and the latter being a potential median between strength and hypertrophy, depending on many other variables.

**I will use these terms as if they are synonyms.*

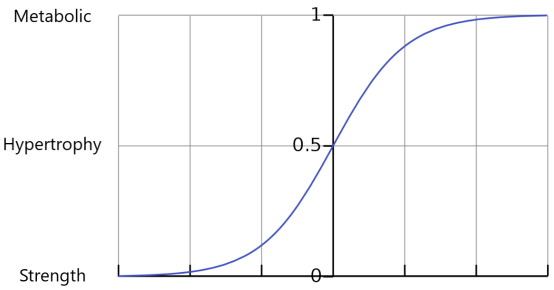
Variables used

The number of repetitions (reps) is the evident first answer. Doing more reps, independently of other metrics, will slowly tilt the stimulus into the metabolic stress category. The main reason being an increase in time under tension (TUT), that will most likely lead to an increase in energy (glucose) demand, a reduction in oxygen availability, an increase in lactate concentration and nitric oxide production. All of which are closely linked to metabolic work. The resistance profile is also an extremely important metric.† Like explained in the article linked below, if you have an exercise which makes it more difficult to execute (due to an increase in power) in the part of the movement where the muscle is at its weakest, the metabolic demand will rise and vise versa. The tempo also needs to be taken into account. The first and third number are already included in the resistance profile, but not the second and last. They manifest a clear impact in the training stimulus, since spending more time in the most metabolically demanding part of the movement (shortened muscle) will increase the metabolic demand, and spending time in the lengthened position will reduce the "stretch reflex" which is an often important component of strength training. The fact that the concentric part of the movement generates more power than the eccentric part (or if the opposite is true) must be included in the process of finding the main stimulus. Since the

concentric portion demands more energy than the eccentric, prioritizing the concentric, for instance, will lead to a slightly more metabolically demanding workout.

† To read more on resistance profiles, click here.

The equation and the program



A sigmoid function (*right*) was chosen, due to its property of nuance in the middle and none in the extremes. Doing 30 reps per set or 100 reps are both in the endurance world, but 8 reps could be in any category, depending on every variable.

The current equation is most likely far from being the best possible one. It only works with single joint movement, with free weights. This version also only functions with muscles that shorten as the joint angle increases, but that would be extremely simple to adjust. It also needs a lot more accuracy. The current equation is almost built on a guess of how much impact a single variable has on determining the stimulus. It is therefore necessary to come up with new ways to test this. Furthermore, changing the way we get data is essential. Going from multiple input functions where the user enters variables to a computer vision program that automatically inputs all of them would help resolve many of the current issues.

```
## This is the python program used for every example showed below

import math
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
from matplotlib.text import Text

## Input from the user and other variables

name = input("Name of the exercise: ")
if "preacher" in name:
     preacher = input("Angle of the preacher bench: ")
else:
     preacher = 0

tempo = input("Tempo: ")
rom = input("Range of motion: ")
reps = input("Number of reps: ")
weight = 10
radius = 0.5

## Resistance profile equation
start = int(rom[:2])
end = int(rom[5:])
contr = int(tempo[3])
stre = int(tempo[1])
```

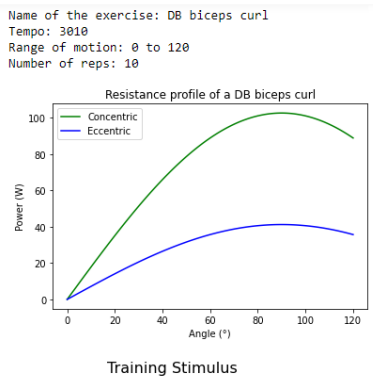
```
maxAng = (math.pi / 180) * end
romRad = (math.pi / 180) * (end - start)
romDeg = list(range(1,((end+int(preacher))-(start+int(preacher))+
2)))
conVel = romRad / int(tempo[2])
eccVel = romRad / int(tempo[0])
simPen = (2 * math.pi) * math.sqrt(radius / 9.8)
pendulum = simPen*(1+((maxAng**2)/16)+((11/3074)*(maxAng**4)))
usedAmp = 0.25 - (start / 360)
gravitVel = romRad / (pendulum * usedAmp)
usedAcc = 1 - (eccVel / gravitVel)
angle = np.arange((start+int(preacher)),(end+1+int(preacher)),1)
concentric = weight*radius*np.sin(np.deg2rad(angle))*9.8*conVel

eccentric = weight*radius*np.sin(np.deg2rad(angle))*9.8*usedAcc

## Plot the resistance profile
x = list(range(start, (end + 1)))
p1 = plt.plot(x, concentric, "g", label="Concentric")
p2 = plt.plot(x, eccentric, 'b', label="Eccentric")
plt.legend(loc="upper left")
plt.xlabel("Angle (°)")
plt.ylabel("Power (W)")
plt.title(f"Resistance profile of a {name}")
plt.show()

## Make sure that there is tension before allocating value to the
time spent in a position
if stre > 0:
     if (weight*radius*np.sin(np.deg2rad(start+int(preacher)))*9.8
     *conVel) == 0:
          stre = 0
     else:
          pass
if contr > 0:
     if (weight*radius*np.sin(np.deg2rad(start+int(preacher)))*9.8
     *conVel) == 0:
          contr = 0
     else:
          pass

## Stimulus equation

if max(concentric) > max(eccentric):
     conType = 0
else:
     conType = -4
maxValue =
math.asin(max(concentric)/(weight*radius*9.8*conVel))*(180/math.pi
)
maxIndex = (romDeg.index(round(maxValue))+1)-(start+int(preacher))
maxTen = maxIndex / len(x)
point =
((int(reps)/2)-5)+(-0.5*(stre+int(tempo[0])))+((contr+(conVel/2))/
2)+conType+(maxTen*2)
stimulus = 1 / (1 + (math.e ** (-1 * point)))
```

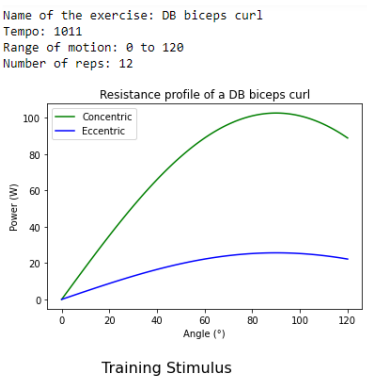
```
## Results
fig1 = plt.figure(facecolor='white')
ax1 = plt.axes(frameon=False)
ax1.get_axis().tick_bottom()
ax1.axes.get_axis().set_visible(False)
x = np.arange(0, 2, 1)
ax1.scatter(stimulus, 1, s=150, marker=7, color="b",
label=f"Stimulus = {round(stimulus, 4)}")
ax1.plot(np.full(x.shape, 1), color="b", linewidth=2)
plt.legend(loc="lower right")
plt.title("Training Stimulus", fontsize=16)
xmin, xmax = ax1.get_xaxis().get_view_interval()
ymin, ymax = ax1.get_yaxis().get_view_interval()
ax1.add_artist(Line2D((xmin,xmax),(ymin,ymin),color='black',
linewidth=2))
plt.show()
```

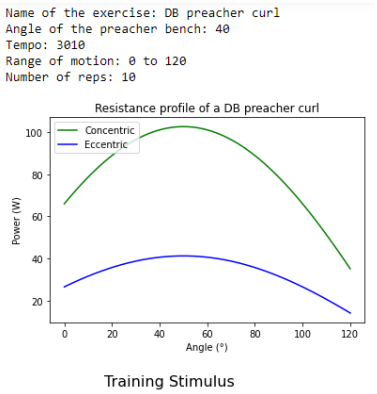
Results

Instead of naming training stimuli "strength", "hypertrophy", "glycolytic" or any other traditional way that we think about it. I chose to go down a level of abstraction since we now have the tools to do so and call a training stimulus by a number between 0 and 1. The lower the number, the closer it is to absolute strength work and the higher, the closer to absolute endurance. Muscle hypertrophy can happen at any number, but is predominant the closer it is to 0.5.

Comparing three different biceps curl

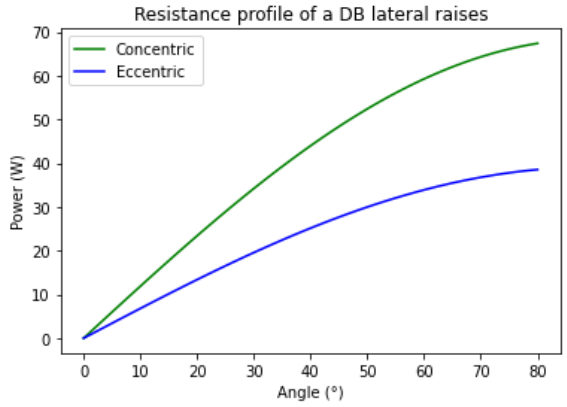




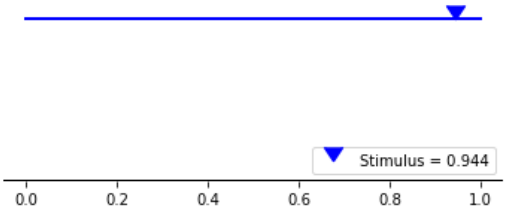


Comparing two different lateral (or deltoids) raises

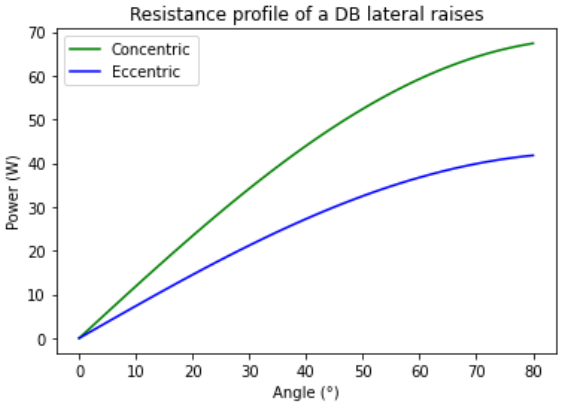
```
Name of the exercise: DB lateral raises
Tempo: 2011
Range of motion: 0 to 80
Number of reps: 12
```



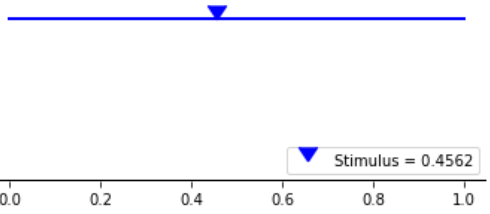
Training Stimulus




```
Name of the exercise: DB lateral raises
Tempo: 3010
Range of motion: 0 to 80
Number of reps: 8
```



Training Stimulus



Conclusion

Like said above, finding a way to prove how much influence a variable has over the outcome is necessary to provide us with the actual stimulus. A statistical analysis of resistance training studies and being able to link the mechanics of the exercise to the muscle architecture's is the starting point in achieving that.

Being able to accurately find the stimulus of an exercise is the first step toward finding the outcome of a training, and then translating that into a full periodization based on the main goal, their strength and their weakness.