Elevator Timing Calculation Methodology

Project Iteration #0 Deliverable

SYSC3303 – L1G7

Wilson Amoussougbo – 101119293

Kenny Deng – 101122713

Cory Helm – 101171699

Colin Marsh – 101112765

Chase Scott – 101092194

2022-02-05

Summary

This report summarizes the reasoning and steps we took to calculate the estimated rate of acceleration/deceleration, estimated maximum speed, and estimated time to load/unload an elevator car. The calculated estimated values are provided in the following table:

Work Products for Iteration #0:

```
Elevator Acceleration (Calculated): 1.181834708105293 metres/second^2
Max Elevator Speed (Calculated, all-floors): 1.5132408575031522 metres/second
Average Elevator Load/Unload: 9.344642857142858 seconds
```

*** NOTE: The source code that was used to create these values are included in the Appendix (Appendix A and Appendix B) ***

Introduction

The following Elevator Timing calculations are made using the provided elevator timing data on Brightspace (*Canal Elevator Measurements.xlsx*). A copy is provided in *Appendix A*.

We are in no way expert statisticians nor do we claim so, but we have made what we believe are reasonable assumptions regarding the elevator timing data and the nature/function of the observed elevator. There are quite a few assumptions that are made during the calculations and all assumptions made will be stated alongside the calculations (in addition to explanations for each assumption).

With that said, we believe the calculated/estimated values are a good representation of the observed elevator. The values seem to be in-line with elevator values found from online searches on the internet.

Calculation Methodology & Reasoning

The purpose of this document is to explain the though process the steps we went through to reach the elevator timing calculations.

Work Products for Iteration #0:

- Your estimated values for the rate of acceleration and deceleration of an elevator car

- The maximum speed of the car
- Your estimated time it takes to load and unload a car

Summary of Steps Taken:

Steps to calculate time to load/unload:

- Simple average, sum all load/unload times and divide by total number of occurrences

Steps to calculate max speed:

- Use Floor1->Floor7 and Floor7->Floor1 observed data to use for calculated velocity (only 1 start and 1 stop event (Figure 1), compared to 6 start and 6 stop event for floor-to-floor observed data (Figure 2))
    - Every time you travel from Floor X to Floor Y, there is 1 stop and start event
- Model velocity (speed) as function of time (for simplicity's sake considering velocity as a piecewise function (simple tapered/ramped square wave, Figure 1), a more accurate model would be to gradually taper to a max speed instead of linearly increase (to smooth out peaks) but given the many unknowns (elevator speed profile, sample size bias, etc) we came to the conclusion that a simple taper would suffice)
- Integrate function over time to find average (avg, Figure 3)
- Vmax = VavgObserved/avg (Figure 3)

Steps to calculate average acceleration:

- Derivate avg velocity function with respect to time to find piecewise acceleration equations (Figure 4)
- Assume rate of acceleration = deceleration (same absolute value, different sign/direction)

For the source code to calculate the Work product values, refer to Appendix B.

> *(There is additional documentation in the code that explains further how we arrived to the final values. Fair warning, the legibility and format of the source code is slightly messy and hard on the eyes.)*

For a summary of the calculated values we arrived to, refer to Appendix C.

Figures:

Figure 1 & 2. #start-stop events bottom->top vs floor->floor

Figure 3. Vmax calculations

Figure 4. Acceleration/deceleration calculation

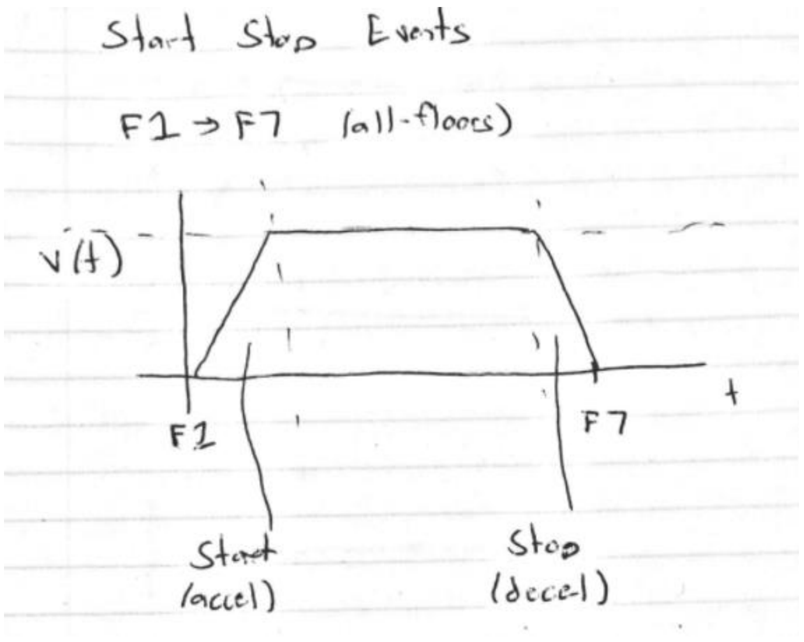


Figure 1. Floor1->Floor7 Velocity Plot & Start/Stop Events

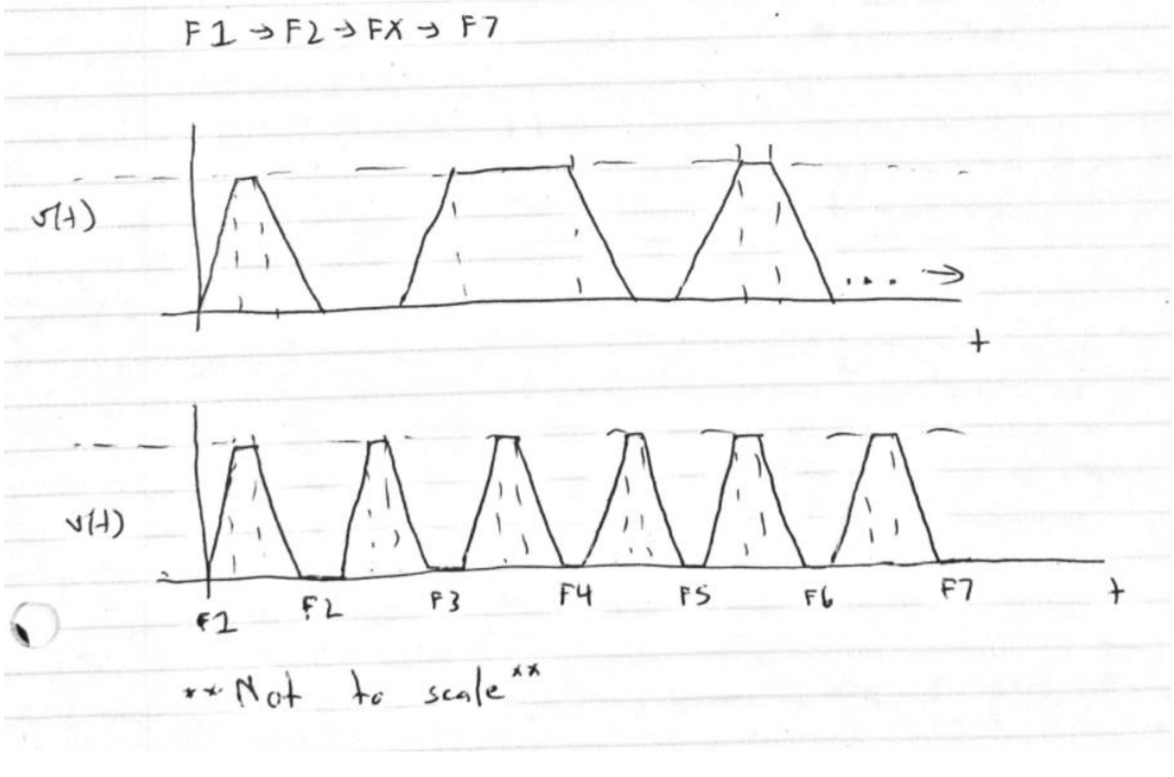
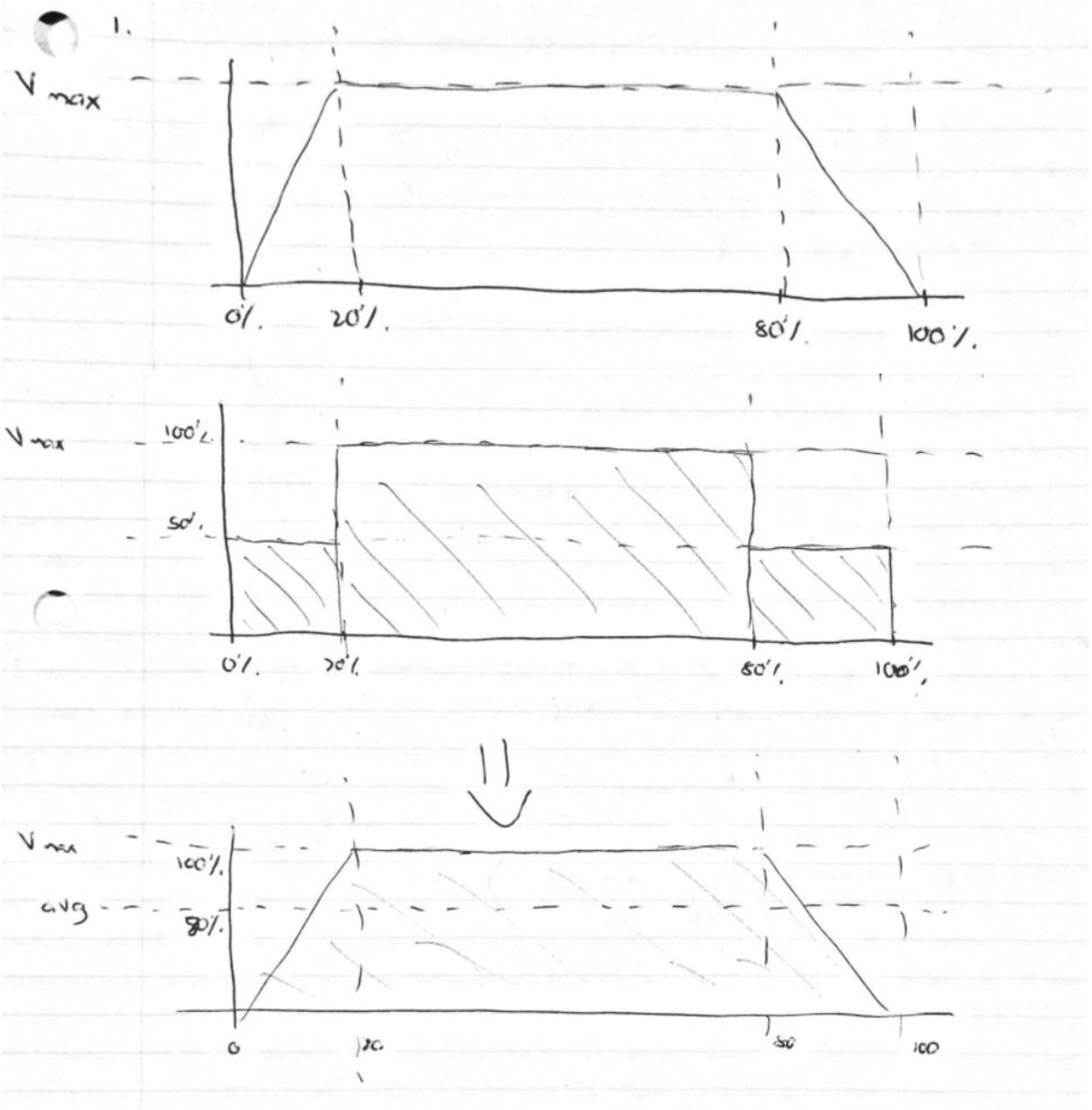


Figure 2. Floor1->FloorX->Floor7 Velocity Plot & Start/Stop Events

1.

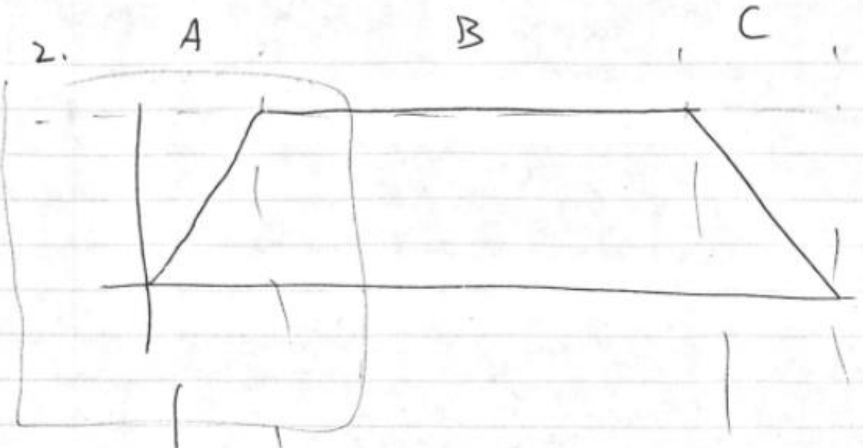


$$0.8\,(V_{max}) = avg$$

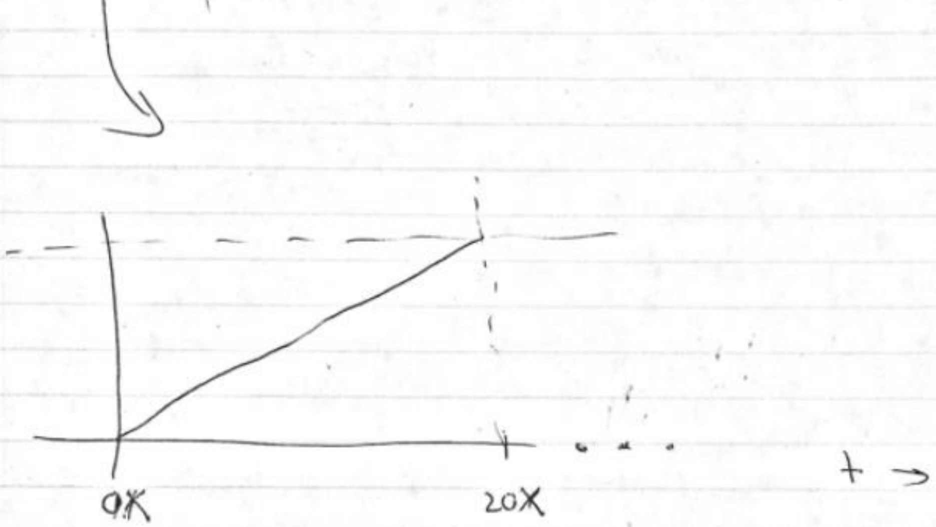
$$\Rightarrow V_{max} = \frac{avg}{0.8} \longleftarrow \underline{\text{avg is } 80\% \text{ of max}}$$

Figure 3. Estimating Max Velocity Given Elevator Velocity Profile and Vavg of Observed Elevator

2.



$$V = \begin{cases} \dfrac{V_{max}}{0.2X} x, & 0 \le x \le 2 \\ V_{max}, & 20 \le x \le 80 \\ \dfrac{-V_{max}}{0.2X} x, & 80 \le x \le 100 \end{cases}$$



$$\left.\begin{aligned} y' &= \frac{V_{max}}{0.2X} t \\ y'' &= \frac{V_{max}}{0.2X} \end{aligned}\right\}$$

where X is observed speed (calculated) from data set

Figure 4. Deriving Acceleration Value of Elevator Given Velocity Plot

Appendix

Appendix A: Sample Elevator Data (Provided from Brightspace)

all-floors

Laps	Time	Cumulative Time
1	0:08.0	0:08.0
2	0:17.6	0:25.6
3	0:09.9	0:35.5
4	0:19.6	0:55.1
5	0:11.0	1:06.1
6	0:19.6	1:25.7
7	0:07.8	1:33.5
8	0:22.5	1:56.0
Summary	1:56.0	1:56.0

per-floor

Laps	Time	Cumulative Time
1	0:06.2	0:06.2
2	0:06.5	0:12.8
3	0:09.7	0:22.5
4	0:08.5	0:31.0
5	0:10.6	0:41.6
6	0:07.7	0:49.2
7	0:10.7	1:00.0
8	0:12.7	1:12.7
9	0:10.2	1:22.9
10	0:08.4	1:31.3
11	0:11.0	1:42.3
12	0:13.2	1:55.5
13	0:08.2	2:03.7

Appendix B: *ElevatorTimingFixed.java* (Script to Calculate Elevator Timing)

```
/*
 *
 * @author: Kenny Deng
 *
 * WARNINGS:
 *          - avgRideTime is average of floor to floor ride times AND normalized
floor1->floor7 (and f7->f1 times),
 *                 issue with using f1->f7 (and f7->f1) is there is only one accel
and one decel event for all floors
 *                 vs 6 accel and 6 decel if travelling floor to floor.
 *
 *                 Will continue to use both in calculation as in the real life
elevators often skip floors (not always
 *                 sequential travelling).
 */


public class ElevatorTimingFixed {
```

```
public static void main(String[] args) {

              /*
               * Your estimated values for the rate of acceleration and
deceleration of an elevator car, and the maximum
               * speed of the car. To estimate the distance between floors, count
the number of steps.  A typical riser is
               * seven inches in height.  Show him you estimate how you come up
with acceleration.  You can assume that
               * the elevator decelerates at the same rate.
               */

              /*
               * Your estimated time it takes to load and unload a car.  You should
probably take note whether there is a
               * significant difference in the loading time of a car if there are a
lot of people getting in and out or not (Hint:
               * the Dunton elevators may be your best bet here!).  This may not be
an average, but a function of the
               * number of people moving in and out of the elevator.   You can
include this information in your excel
               * spreadsheet, or submit it as a separate PDF file.
               */

              /*
               * Finally, answer the three questions in the quiz for Iteration 0.
You will need to submit the maximum speed
               * for the elevator, the rate of acceleration for the elevator, and
the average loading/unloading time.  Only one
               * person per group needs to do this (assuming I have figured out how
to correctly configure this in CU Learn).
               */

              // import all-floors data (floor to floor time)
              double[] loadUnloadTimeAllFloors = {8.0, 9.9, 11.0, 7.8};
              double[] rideTimeAllFloors = {17.6, 19.6, 19.6, 22.5};

              // import per-floor data
              double[] loadUnloadTimePerFloor = {6.2, 9.7, 10.6, 10.7, 10.2, 11.0,
8.2};
              double[] rideTimePerFloor = {6.5, 8.5, 7.7, 12.7, 8.4, 13.2};


              // average loading/unloading time of "All Floors" and "Per Floor"
              // 0.5 * (avgAllFloors + avgPerFloor)
              double avgLoadUnloadTime =
averageAllFloorsPerFloor(simpleAverage(loadUnloadTimeAllFloors),
simpleAverage(loadUnloadTimePerFloor));
              System.out.println("Average Elevator Load/Unload: " +
avgLoadUnloadTime + " seconds");

              // average ride time of "All Floors" and "Per Floor"
              // 0.5 * (avgAllFloors + avgPerFloor)
              // BIG thing to watch our for, AllFloors doesnt have to slow down for
each floor (only 1 accel and 1 decel)
```

```
		double avgRideTime = 
averageAllFloorsPerFloor(normalizeAverageAllFloors(rideTimeAllFloors, 6), 
simpleAverage(rideTimePerFloor));
		System.out.println("Average Elevator Ride Time Between Floors: " + 
avgRideTime + " seconds");

		// average elevator speed
		// velocity = distance/time
		double avgElevatorSpeed = averageElevatorSpeed(avgRideTime, 4);   // 4 
metres between each floor as shown on "Project" file Version 3.0,

						// 2022/01/10, pg. 7 of 12, paragraph 1
		System.out.println("Average Elevator Speed (all-floors + per-floor 
data): " + avgElevatorSpeed + " metres/second");

		// average all-floors elevator speed of f1->f7 and f7->f1
		// using these values as there are only 1 accel and 1 decel events in 
each measurement
		// velocity = distance/time
		double avgElevatorSpeedF1toF7 = 
(6*4)/simpleAverage(rideTimeAllFloors);
		System.out.println("Average Elevator Speed (all-floors): " + 
avgElevatorSpeedF1toF7 + " metres/second");

		/*
		// max velocity = average/0.90
		// ask Kenny for the math, he will explain (I drew it out on paper, 
will transfer to electronic later)
		// jist of it is 10% of time accel to max speed, 80% cruising at max 
speed, 10% of time decel to stop
		// a lot of assumptions are being made
		// assuming ramped square velocity characteristic
		double maxSpeedCalculated = avgElevatorSpeed/0.9;
		System.out.println("Max Elevator Speed (Calculated): " + 
maxSpeedCalculated + " metres/second");

		// acceleration = d/dt (velocity)
		// velocity = [ (y1 - y0)/(x1 - x0) ]x + b
		// vel = [(Vmax-0)/((0.1*time) - 0)]x + 0
		// vel = (Vmax/0.1time)x
		// accel = vel' = Vmax/0.1time
		double accelCalculated = maxSpeedCalculated/(0.1*avgRideTime);
		System.out.println("Elevator Acceleration (Calculated): " + 
accelCalculated + " metres/second^2");
		System.out.println("Time Elevator Spent Accelerating/Decelerating 
(Calculated): " + 0.1*avgRideTime + " seconds");
		*/
		double maxSpeedCalculatedAllFloors = avgElevatorSpeedF1toF7/0.8;
		System.out.println("Max Elevator Speed (Calculated, all-floors): " + 
maxSpeedCalculatedAllFloors + " metres/second");
		double accelCalculatedAllFloors = 
maxSpeedCalculatedAllFloors/(0.2*avgRideTime);
		System.out.println("Elevator Acceleration (Calculated): " + 
accelCalculatedAllFloors + " metres/second^2");
```

```
		System.out.println("Time Elevator Spent Accelerating/Decelerating 
(Calculated): " + 0.2*avgRideTime + " seconds");

	}

	/*
	 * Calculate simple average of array of double data. No extrapolation or 
manipulating/curving.
	 * 
	 * Parameter double[] with elevator data
	 * 
	 * Return average with type double
	 */
	public static double simpleAverage(double[] arr) {
		double sum = 0;
		for (int i = 0; i < arr.length; i++) {
			sum += arr[i];
		}

		return sum/arr.length;
	}

	/*
	 * Convert floor1->floor7 time into average of floor to floor times
	 */
	public static double normalizeAverageAllFloors(double[] arr, int 
floorsTravelled) {
		double normSum = 0;
		for (int i = 0; i < arr.length; i++) {
			normSum += arr[i]/floorsTravelled;
		}

		return normSum/arr.length;
	}

	/*
	 * Calculate average of "AllFloors" and "PerFloor" from return value of 
simpleAverage()
	 * 
	 * Average = 0.5 * (sum of all data)
	 */
	public static double averageAllFloorsPerFloor(double allFloors, double 
perFloor) {
		return 0.5 * (allFloors + perFloor);
	}

	/*
	 * Returns average elevator speed.
	 * 
	 * Parameters: average (AllFloors & PerFloor) and floor to floor distance 
(metres)
	 */
	public static double averageElevatorSpeed(double averageTime, double 
floorToFloorDistance) {
		return floorToFloorDistance/averageTime;
```

```
       }

}
```

Appendix C: *ElevatorTimingFixed.java* Script Output (Appendix B)

```
Average Elevator Load/Unload: 9.344642857142858 seconds
Average Elevator Ride Time Between Floors: 6.402083333333334 seconds
Average Elevator Speed (all-floors + per-floor data): 0.6247966156849983
metres/second
Average Elevator Speed (all-floors): 1.2105926860025218 metres/second
Max Elevator Speed (Calculated, all-floors): 1.5132408575031522 metres/second
Elevator Acceleration (Calculated): 1.181834708105293 metres/second^2
Time Elevator Spent Accelerating/Decelerating (Calculated): 1.2804166666666668
seconds
```