# SG90 Servo Characterization.

Guido di Pasquo[1]

*Abstract* - This paper investigates the transient and steady state response of a SG90 Servo and two methods to measure the angle of its shaft using an Inertial Measurement Unit (IMU). It will also show its transfer function and some instabilities and accuracy problems found during testing. It also analyzes the response obtained using the IMU as a feedback sensor for a closed loop system.

## I. Introduction

Since programable micro controllers are now inexpensive and widely available, they have become the first contact with electronics for many students, including University Students. One of the most used servos for homemade projects is the SG90. Its low cost has made it very attractive to all kind of students. University projects usually start like low-cost projects, but with an engineering approach. Reason that makes them affordable but complex.

## II. SG90 Characteristics and Applications

The SG90 (Figure 1) is a light weight, low-cost hobby servo. It is controlled by Pulse Width Modulation (PWM), with a duty cycle of 600µs to 2400µs (measured), and a total period of 20ms (50Hz). With the specifications provided by the manufacturer (Table 1), this completes all the data easily available in Internet.

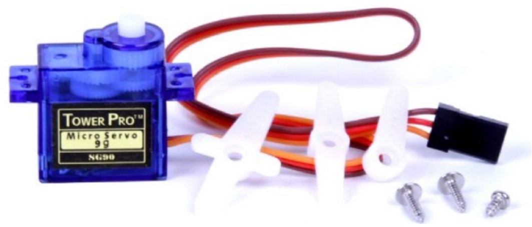


*Figure 1 – SG90 with the three horns that comes with it [1].*

As shown in Table 1, there is no information about the response of the servo to standard inputs. It is also important to know, that in many cases, the servo is controlled with libraries. These libraries translate the input of the programmer (in degrees) to the adequate PWM.
The PWM period then is transformed, for the user, into a sample time. This sample time means that commands with a lifespan of less than 20ms might not be read, and it acts as a delay for longer ones.
As for applications, there are several in multiple disciplines, but since this Paper is made by Aeronautical Engineering students, it will focus on small angle inputs with small loads, to resemble the movement of control fins. Inputs of 45 and 90 degrees will be covered, but not analyzed in depth.

**TowerPro SG90 Servo**

**Specifications**

| | |
|---|---|
| Modulation: | Analog |
| Torque: | **4.8V:** 25.00 oz-in (1.80 kg-cm) |
| Speed: | **4.8V:** 0.12 sec/60° |
| Weight: | 0.32 oz (9.0 g) |
| Dimensions: | **Length:** 0.91 in (23.0 mm)<br>**Width:** 0.48 in (12.2 mm)<br>**Height:** 1.14 in (29.0 mm) |
| Motor Type: | 3-pole |
| Gear Type: | Plastic |
| Rotation/Support: | Bushing |

*Table 1– Datasheet provided by the manufacturer [2].*

## III. Testing

An MPU6050 is used to measure angle. This Inertial Measurement Unit (IMU) has a 3 Degrees of Freedom (DOF) accelerometer and 3DOF gyroscope. The angle is measured with both sensors. When using the accelerometer, the gravity vector is decomposed in two components. The angle is found using inverse trigonometric functions. For the gyroscope, the angle is calculated by integrating the measured angular velocity. Since the accelerometer is perturbated by the accelerations produced by the movement of the IMU, but it measures angle directly (from an external reference vector), it does not have drift nor integrating error, therefore, its measurement is used to calculate steady state error. The gyroscope data gives a more precise transient response but is prone to drift and integrating error.
To collect the data, an Arduino Nano is used. The sample time is 6ms.The accelerometer data is filtered with a Kalman filter to smooth it. This causes it to behave 'slower' than the raw input.
The Servo is connected directly to the 5V Arduino output. Higher actuation speeds are achievable with higher voltages, therefore the results shown are only valid for a 5V operating voltage. The test stand used is shown in Figure 2.
To measure the angular position of the servo shaft, it was centered at 90º, and then the IMU was fixed to the horn.
The input consists of 5, 10, 20, 45- and 90-degree steps, followed by a 60 and 300 degree/second ramp, and finish with an impulse.
The raw data obtained by the IMU is the shown in Figure 4.

[1] Aeronautical Engineering Student at Universidad Tecnológica Nacional – Regional Haedo – Argentina (gdipasquo387@alumnos.frh.utn.edu.ar)

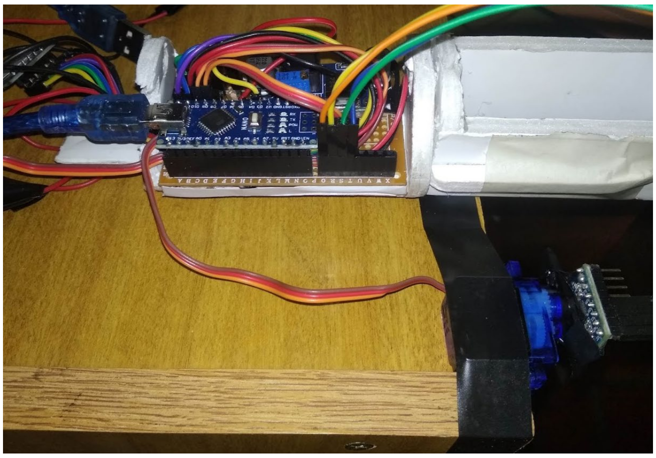

Figure 2 – Test stand.

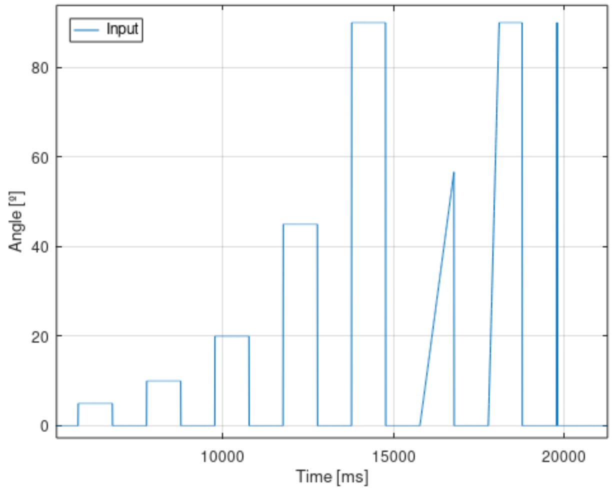


Figure 3 – Inputs.

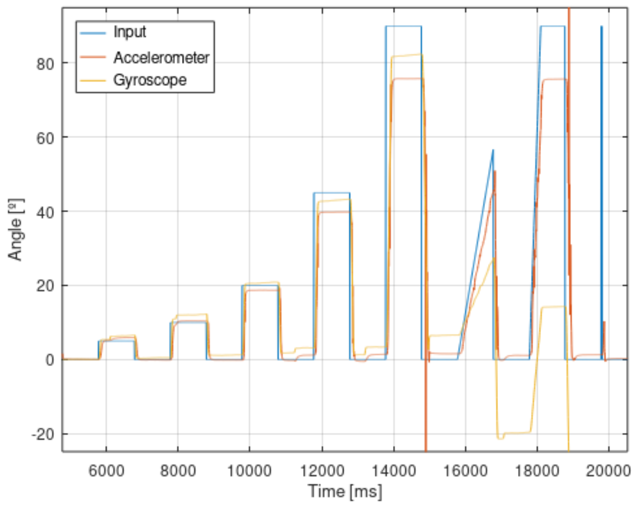


Figure 4 – Data obtained with the accelerometer and gyroscope.

It is easy to see the gyroscope integrating error going out of control after the ramp input. Besides this, it is possible to see the accelerometer noise and steady state error in the servo position. The last one is going the be analyzed in detail.

## IV. Results

The system proposed is an open loop system. Even though the servo has a microcontroller with feedback in it, the user does not have access to it. So, an external sensor (IMU in this case) is required to measure the position of the servo's shaft. In practice, this method is cumbersome and not applicable for aerospace projects. Therefore, the servo is, for the user, an open loop system. Ways of using the internal potentiometer to close the feedback loop with the Arduino are not covered in this paper.

STEP RESPONSE:

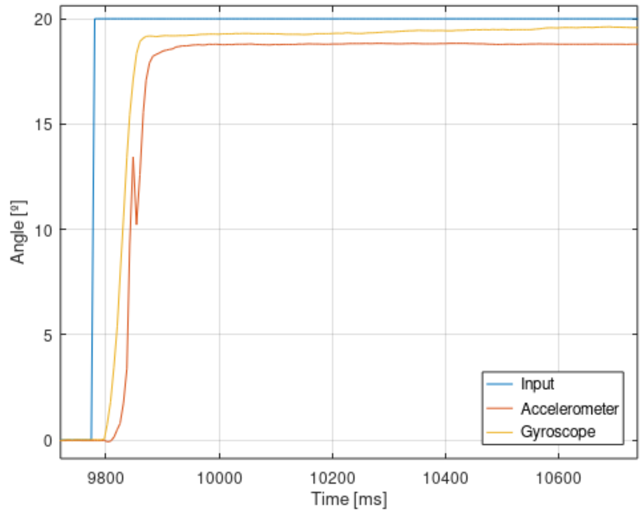


*Figure 5 – Detail of the 20-degree step. It is possible to see both the noise of the accelerometer and the drift of the gyroscope. Note the delay between the input and the response, caused by the servo's sample rate.*

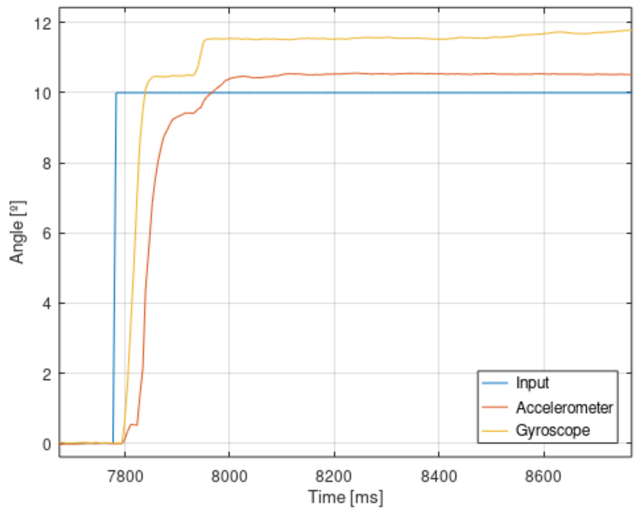


*Figure 6 – Detail of the 10-degree step, a small jump in angle can be seen. This jump appears semi-randomly (0 degrees (center point) is also affected).*

Analyzing the step responses:

| Input [°] | RT [s] | $E_{ss}$ [%] | Average Speed [°/s] |
|---|---|---|---|
| 5 | 0.059 | -9.133 | 92.98 |
| 10 | 0.059 | 0.225 | 196.22 |
| 20 | 0.069 | 5.75 | 278.26 |
| 45 | 0.104 | 14.44 | 375.47 |
| 60 | 0.116 | 15.43 | 458.1 |
| 90 | 0.15 | 17.33[2] | 496.83 |

*Table 2 – Results, where RT is the rise time and $E_{ss}$ the steady state error. Results from the five- and ten-degree inputs are the average of four runs.*

The average speed was calculated dividing the real angle turned (instead of the intended input), by the rise time.
The 60-degree input was added just to compare its rise time with the advertised speed in the datasheet. The real speed is 0.116s/60º, confirming the datasheet value. It is important to remember that the servo only moved 50º.
Because the servo is ideally a second order system[3], a second order transfer function is proposed. To find the transfer functions, the `arx` and `d2c` functions of Octave's control package are used.

| Input | Transfer Function | |
|---|---|---|
| 5º | $\dfrac{3.006\,s + 4006}{s^2 + 91.31\,s + 3761}$ | (1) |
| 10º | $\dfrac{4.419\,s + 5118}{s^2 + 103.6\,s + 4903}$ | (2) |
| 20º | $\dfrac{-0.5396\,s + 2054}{s^2 + 59.67\,s + 2159}$ | (3) |
| 45º | $\dfrac{1.562\,s + 589.2}{s^2 + 30.42\,s + 691.9}$ | (4) |
| 90º | $\dfrac{0.9876\,s + 207.5}{s^2 + 16\,s + 259.9}$ | (5) |

The step responses of the obtained transfer functions are compared with the data of Table 2 in Table 3.

| Input [°] | $RT_{TF}$ [s] | RT Error [%] | $Overshoot_{TF}$ [%] |
|---|---|---|---|
| 5 | 0.057 | 3.5 | 3.8 |
| 10 | 0.0506 | 16.6 | 3.16 |
| 20 | 0.0632 | 9.17 | 7.14 |
| 45 | 0.0972 | 6.99 | 11.76 |
| 90 | 0.144 | 4.16 | 16.73 |

*Table 3 – Results of the transfer functions, where RT is the rise time.*

The transfer functions can be simplified by eliminating the higher order term from the numerator, which removes all zeros. This was not done to maintain the output of the `arx` function unaltered. As an example, (2) becomes:

| | | |
|---|---|---|
| 10° | $\dfrac{5118}{s^2 + 103.6\,s + 4903}$ | (2b) |

And if one wished to eliminate the $E_{ss}$, the following transfer function should be used:

$$\frac{K}{s^2 + J\,s + K}$$

Where the numerator is set to the K value of the denominator.

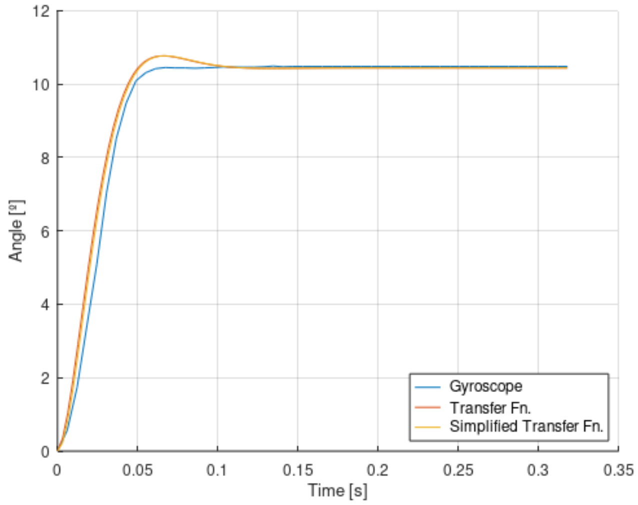


*Figure 7 – Transfer function calculated with Eq. 2 and Eq.2b, and real data of the 10º input. It is seen that the response is similar. However, the simulated transfer function has a faster acceleration rate, lower rise time, and 3.8% of overshoot.*

Another way of simulating the servo is using a Rate Limiter set at the average speed of the servo. It is important to note that the average speed varies with the input, so, a transfer function is preferred.
In both cases, it is highly recommended the use of a zero-order hold to simulate the sample time of the servo.

RAMP RESPONSE:

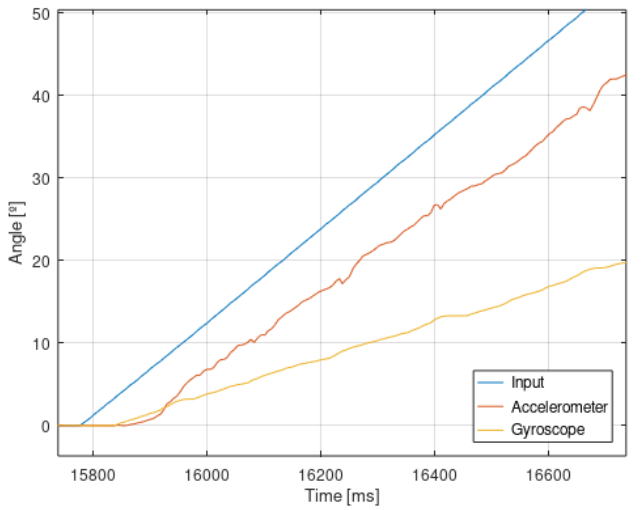


*Figure 8 – Transient response to a ramp input. At low angular speeds, the accelerometer is much more precise.*

Looking at the response, it is seen how the servo lags behind the input. Note that, if the Ess of the step input is negative,

[2] This Ess value cannot be trusted since the servo reached its maximum deflection angle.

[3] Simple rotating mechanism.

the servo amplifies the input, so it should overtake the ramp. This does not happen, indicating that the transfer function used for the step response (Eq. 1) does not match the ramp response of the servo. Therefore, a new pair of Transfer Functions are calculated.

| Input | Transfer Function |
|---|---|
| 60º/s | $$\frac{-19.17\,s^2 + 1.23e4\,s + 3.967e05}{s^3 + 295.3\,s^2 + 1.984e04\,s + 9.88e05} \quad (6)$$ |
| 300º/s | $$\frac{103\,s^2 + 706.2\,s + 8.747e05}{s^3 + 482.3\,s^2 + 4.879e04\,s + 2.329e06} \quad (7)$$ |

IMPULSE RESPONSE:

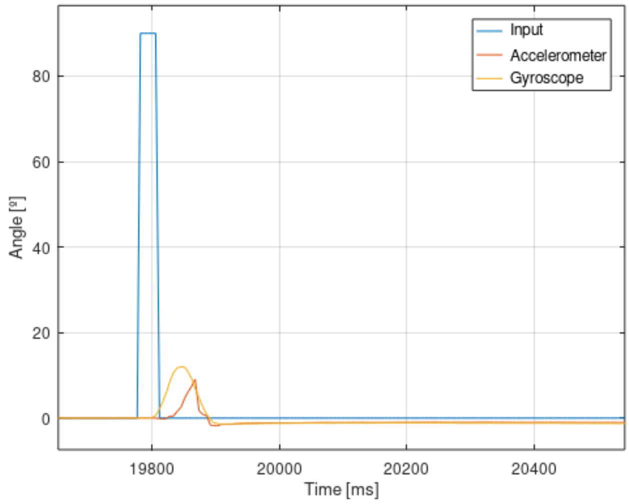


*Figure 9 – Response to a 25ms impulse. Note that the servo doesn't go back to cero after the impulse is gone.*

To simulate the impulse response, Eq.3 can be used.
In Figure 9, it is seen the non-repeatability nature of the servo, where the same 0º input (before and after the impulse) produces two different outputs (0º and -1.15º).

## V. ISSUES

As seen in Figure 6, there is a small jump in angle after the output stabilizes, this jump was present in two of the six runs for the 10-degree input, and in only one for the 5- degree input. The same jump is seen when the servo returns to the center point after inputs greater than 20º.
This behavior changed if the input was inverted, as shown in Figure 10.
Note the off-cero angle after the 5º input, and the jump at 20º and 90º. A smaller steady state error in seen for the 5- and 10-degree inputs. This behavior was present in both runs of the test.
Afterwards, the center point was switched to 10º instead of 90º, which gave the results shown in Figure 11.
With the center point at 10º, a bigger 'negative' error is seen for the first three inputs, note the off-center angle of the servo when it tries to return to cero. The error for the 45- and 90-degree input is lower.
A method to predict these jumps could not be found. Therefore, these jumps are thought to be random. It is possible to see the big differences in the steady state response using different center points or inputs. This makes for non-predictable-non linearities, that are usually disregarded due to the servo's price.

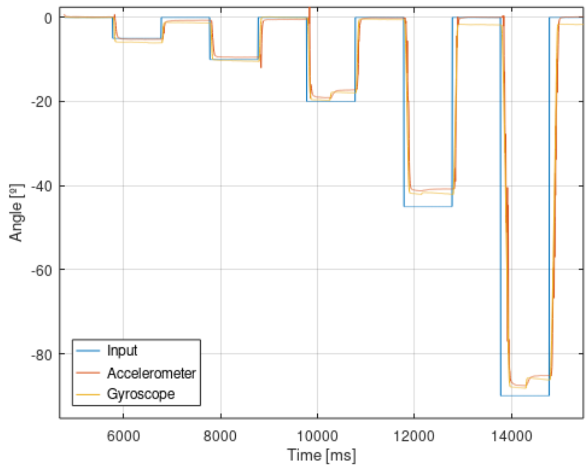


*Figure 10 – Negative inputs.*

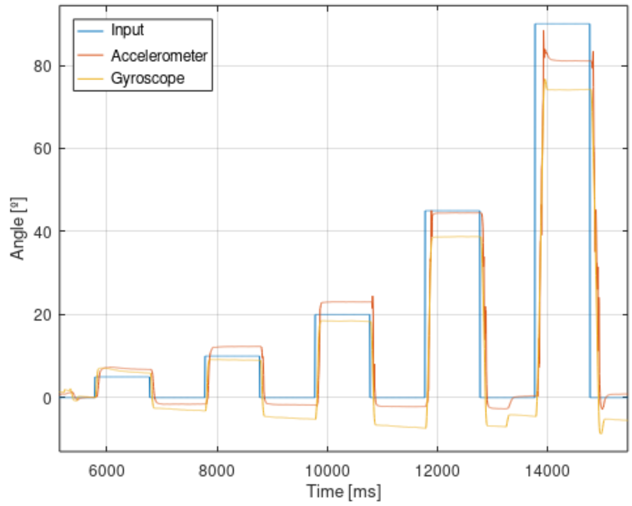


*Figure 11 – Center point at 10º.*

The transient response was always similar, overshoot was seen only for the 0 to 90º and 90º to 0º inputs with a 10º center point.
Repeatability of the output of the servos is achievable if the jumps are disregarded. Still, the jumps usually appear for the same inputs at similar times. That allows some level of predictability in characterized servos.
Because these servos are made by multiple manufacturers, the quality is variable, and the response can be different from the one shown in this paper.

## VI. FEEDBACK

A closed loop system is proposed to compensate the error and improve the response of the servo.
The two PID (parallel TF) tested are:

| PID | P | I | D |
|---|---|---|---|
| A | 0.5 | 10 | 0 |
| B | 1 | 10 | 0 |

*Table 4 – PID coefficients.*

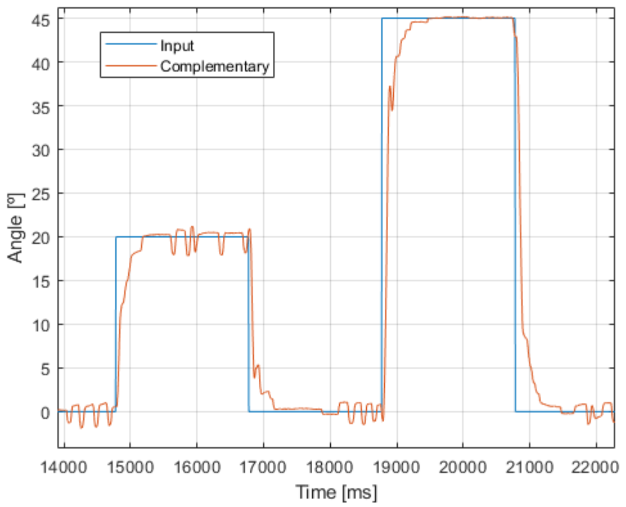

*Figure 12 – Response to a 20º and 45º step (PID A).*

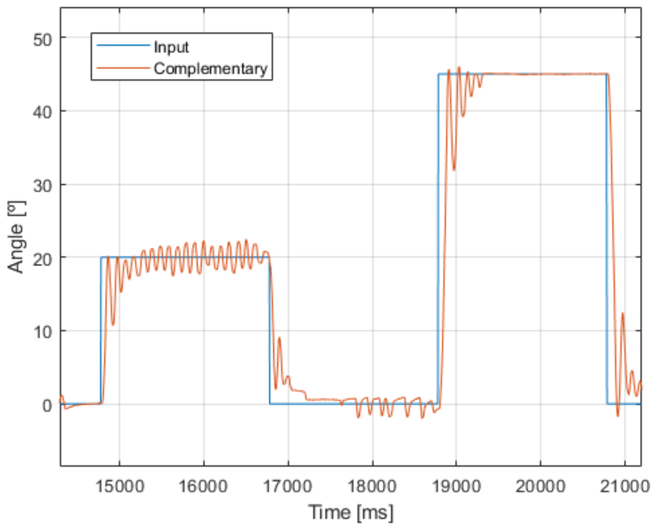

*Figure 13 – Response to a 20º and 45º step (PID B).*

It is seen that the *Ess* is now cero, but the servo oscillates around the setpoint[4]. The reason for this behavior is not known, but it is probably related to the servo's definition and precision (same case as the jumps). An increase in the P term increased the amplitude of the oscillations, in contrast, a decrease in the P term reduced their frequency. The response time is slower (0.55s vs 0.1s) and a new transient response can be seen, composed of, at least, a real pole and two imaginary ones. Faster response times could not be achieved without uncontrollable oscillations, so they are not shown here.
In the other hand, the ramp response improved and the Ess now has a finite value. The response was the same for both PIDs.
Even though a stable closed loop system has been achieved, the simulation of the Closed Loop Transfer Function did not give the same response. This means that the transfer function calculated in Eq. 1 cannot be used to simulate this servo in a closed loop configuration.

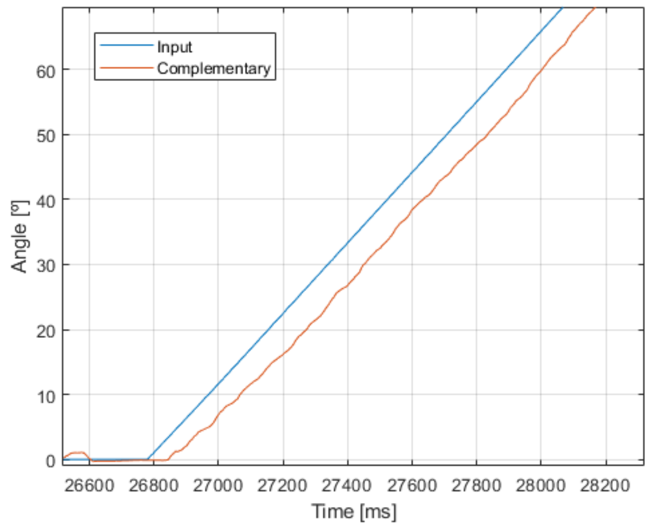

*Figure 14. Response to a 60º/s ramp.*

## VII. CONCLUSIONS

The transient and steady state response have been measured and a transfer function obtained. The limits of the inputs that this transfer function can simulate, and the error expected have been considered. Issues with the servo response have been analyzed. It has been determined that the inexpensive nature of the servo must be considered, since that produces great variation in quality. On the other hand, it has been shown that the response might be adequate for the applications intended, since the transient response is a strong point of the SG90 tested.
A closed loop system has been analyzed and the results have been shown.
The results obtained here, since only one servo was tested, are not a statistically valid representation of all SG90, therefore characterization of the actual servos used is encouraged.
In conclusion, this papers results allow for a transfer function to be introduced into the design and simulation phase of a project to obtain an approximation of the influence of the servo's dynamics on the system.

REFERENCES:

[1] http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[2] https://servodatabase.com/servo/towerpro/sg90

[3] Katsuhiko Ogata - Modern Control Engineering (5th Edition)

[4]https://www.electronics-lab.com/project/using-sg90-servo-motor-arduino/

[5] EEEnthusiast - Ep. 57 Arduino Accelerometer & Gyroscope Tutorial MPU-6050 6DOF Module.

[4] Measurements were done with a complementary filter ($\alpha = 0.96$) to avoid drift and still have a good representation of the transient response.