# 2.5 Checking of results

When writing the equations there are places where errors can easily sneak in (e.g. notation errors). To check whether the generated results are correct, control calculations have been made. This part of the text will be about checking the kinematics.

# 2.5.1 Checking the position

To check the position, we calculated the position of the point F in 2 different ways. Because F is a stationary point it is first calculated via the measured/given distance and angle from point A:

F1 = r1b\*exp(1i\*phiAF)

The second way is by following a path of consecutive points in the mechanism:

F2 = r2c\*exp(1i\*(phi2 + phiA)) + r3\*exp(1i\*phi3) + r4\*exp(1i\*phi4)

The difference between F1 and F2 is plotted in figure [?]. The x and y components match well, there is nothing special to report there. The relative error 1 – F1/F2 takes on values from the order of 10^-12 and the absolute error from the order of 10^-10 cm. This is not yet near machine precision, so the error is probably a result of using fsolve. The error is small enough to consider this check successful

We can also visually inspect the positionby viewing an animation of the mechanism. In our animation we can see that point F (and A and E) doesn’t move. You can find the animation in the Matlab project

Figuur [?]: Position errors as functions of time.

# 2.5.2 Checking the speed and acceleration

Checking the speed and acceleration is done in the same way. Because F is a stationary point, it’s speed and acceleration have to be equal to 0 at all times. Then, by plotting the speed or acceleration of F calculated via a path of consecutive points from A to F, we will plot the error.

The speed of point F (via A) can be calculated as follows:

V\_F = cross(omega2,AC\_vec)+ cross(omega3,CD\_vec) + cross(omega4,DF\_vec)

Figure [?] presents the error in velocity of point F. The x and y components match well, there is nothing special to report there. Except for the 2 deviating values, this error seems to stay perfectly at 0. After zooming in we can see that this error doesn’t perfectly equal 0 but is instead in the order of 10^-14 cm/s. This is starting to approach machine precision but is still not as precise. Again, the error is a result of using fsolve. The error is small enough to consider this check successful

Figuur [?]: Speed errors as functions of time.

We can calculate the accleration of point F (via A) as follows:

A\_F = cross(omega2,cross(omega2,AC\_vec)) + cross(alpha2,AC\_vec) + cross(omega3,cross(omega3,CD\_vec)) + cross(alpha3,CD\_vec) + cross(omega4,cross(omega4,DF\_vec)) + cross(alpha4,DF\_vec)

Figure [?] shows the acceleration error. Once again the x and y components match so there is nothing special to report. The error here is from the order of 10^-14 cm/s^2. Like with the speed, this isn’t machine precision yet but a deviation introduced by fsolve. It’s small enough to consider this check successful

Figuur [?]: Acceleration errors as functions of time

# 3.3 Checking of results

For checking the dynamics we make use of the method of shaking forces, in which we determine the reaction forces that act on the ground. These reaction forces are the opposites of the internal forces FA, FF and FE . The shaking forces are equal to these reaction forces on the ground

but are also calculated as follows (given for the x direction, y direction is similar):

Fshake\_x\_1 = FAx + FFx + Fex

Fshake\_x\_2 = acc2x\*m2 + acc3x\*m3 + acc4x\*m4 + acc5x\*m5 + acc6x\*m6 + acc7x\*m7 + acc8x\*m8 + acc9x\*m9 + acc10x\*m10 + acc11x\*m11 + acc12x\*m12

We take the sum over all linkages except for the ground. The error of the shaking forces is plotted in figure [?]. The x and y components match well enough, even though one is one order of magnitude greater than the other. The error is from the order of 10^-9 N for the x component to 10^-10 N for the y component. This error is greater than machine precision is a result of the used math algorithm. The error is small enough to consider this check successful