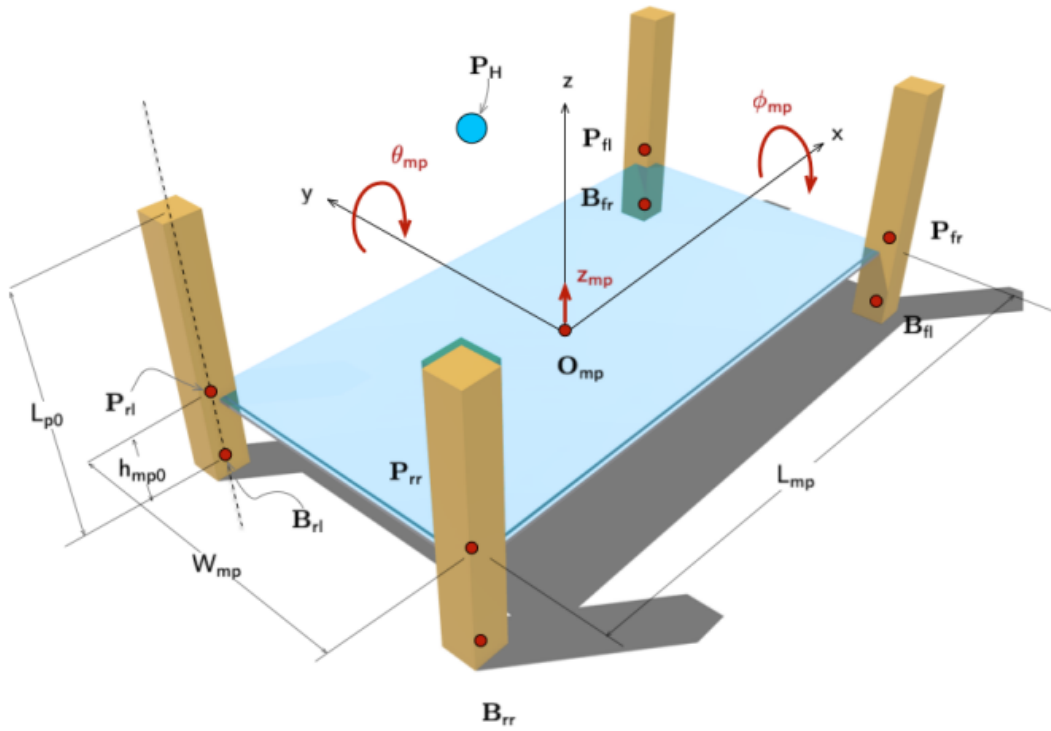


# Study the kinematics and dynamics of a driving simulator

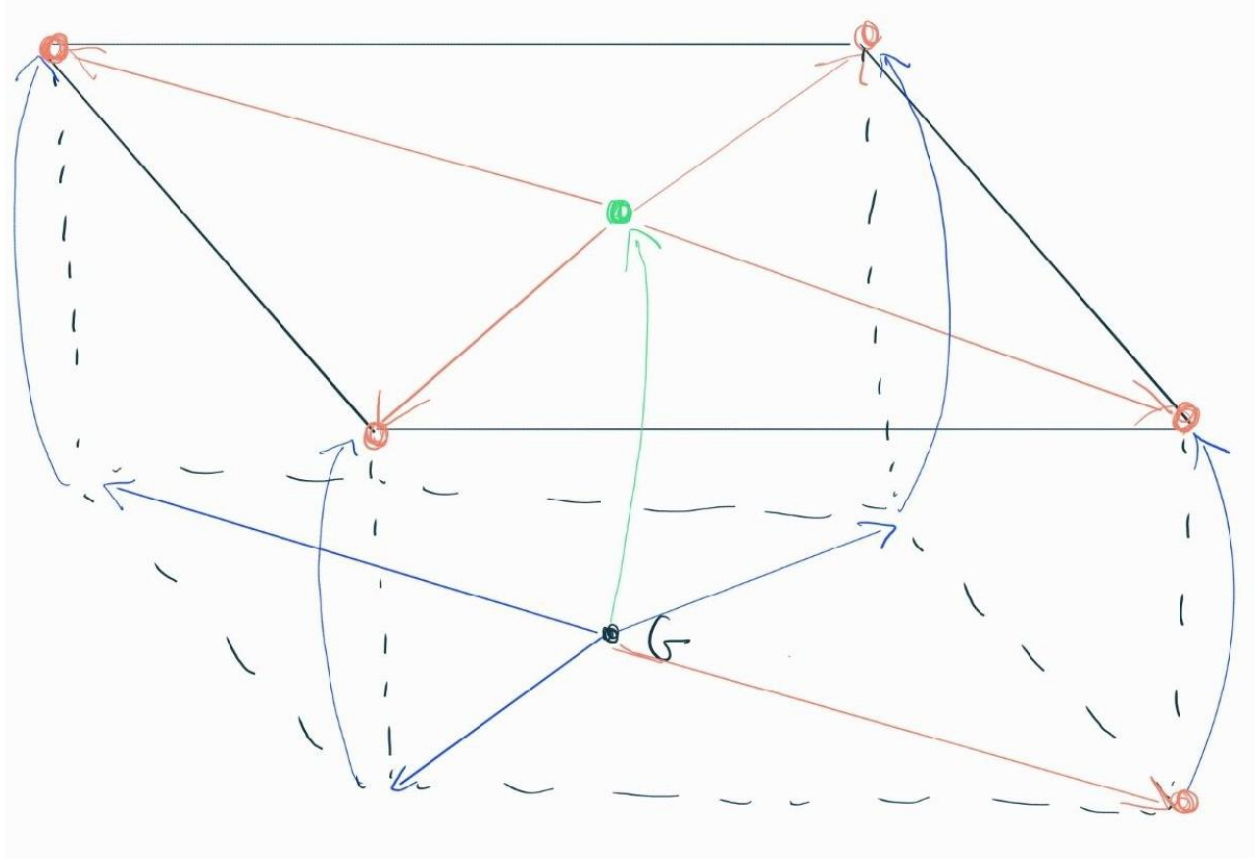
Arnau Bayer Mena  
Giacomo Montagna  
Jacopo Veronese

June 2025



## Kinematic Analysis

The kinematic and dynamic analysis of the multibody system was performed using the following linear graph and a global approach.



To solve the kinematic problem, one of the platform's feet is considered fixed, while the other three are treated as joints free to move in the  $x$ - $y$  plane.

fixed foot translation matrix:

$$\begin{pmatrix} 1.0000 & 0 & 0 & -0.8000 \\ 0 & 1.0000 & 0 & -0.4100 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$$

Translation Matrix for a Moving Foot:

$$\begin{pmatrix} 1 & 0 & 0 & x_{rl} \\ 0 & 1 & 0 & y_{rl} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A roto-translation matrix is defined to describe the transformation from the ground to the center of mass of the platform, where rotations occur only around the  $x$  and  $y$  axes.

$$\begin{pmatrix} \cos(\theta_{mp}) & 0 & \sin(\theta_{mp}) & x_{mp} \\ \sin(\phi_{mp}) \sin(\theta_{mp}) & \cos(\phi_{mp}) & -\cos(\theta_{mp}) \sin(\phi_{mp}) & y_{mp} \\ -\cos(\phi_{mp}) \sin(\theta_{mp}) & \sin(\phi_{mp}) & \cos(\phi_{mp}) \cos(\theta_{mp}) & z_{mp} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Under the assumption of small angles, this matrix can be simplified.

$$\begin{pmatrix} 1 & 0 & \theta_{mp} & x_{mp} \\ \phi_{mp} \theta_{mp} & 1 & -\phi_{mp} & y_{mp} \\ -\theta_{mp} & \phi_{mp} & 1 & z_{mp} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Fixed transformation matrices are also defined from the platform's center of mass to its edges in local coordinates.

The transformations from the base to the top of the actuator are modeled as a prismatic joint, which move vertically on an axis that shares the same rotation as the platform.

$$\begin{pmatrix} \cos(\theta_{mp}) & 0 & \sin(\theta_{mp}) & hmp_{fr} \sin(\theta_{mp}) \\ \sin(\phi_{mp}) \sin(\theta_{mp}) & \cos(\phi_{mp}) & -\cos(\theta_{mp}) \sin(\phi_{mp}) & -hmp_{fr} \cos(\theta_{mp}) \sin(\phi_{mp}) \\ -\cos(\phi_{mp}) \sin(\theta_{mp}) & \sin(\phi_{mp}) & \cos(\phi_{mp}) \cos(\theta_{mp}) & hmp_{fr} \cos(\phi_{mp}) \cos(\theta_{mp}) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

These matrices are then simplified, again under the assumption of small angles.

$$\begin{pmatrix} 1 & 0 & \theta_{mp} & hmp_{fr} \theta_{mp} \\ \phi_{mp} \theta_{mp} & 1 & -\phi_{mp} & -hmp_{fr} \phi_{mp} \\ -\theta_{mp} & \phi_{mp} & 1 & hmp_{fr} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## Direct kynematics

Equations are created to find the coordinates for the feet, the platform, and a single actuator's length, all based on the lengths of the three other actuators. This is accomplished using the T-graph. It's worth noting that the R-graph equations do not need to be solved, as they are inherently satisfied because the joints undergo the same rotations as the plane itself.

T-graph equations:

$$\begin{pmatrix} hmp_{rr} \theta_{mp} - x_{mp} = 0 \\ \sigma_1 - hmp_{rr} \phi_{mp} - y_{mp} = 0 \\ hmp_{rr} + \frac{41 \phi_{mp}}{100} - \frac{4 \theta_{mp}}{5} - z_{mp} = 0 \\ x_{fl} - x_{mp} + hmp_{fl} \theta_{mp} - \frac{4}{5} = 0 \\ y_{fl} - y_{mp} - hmp_{fl} \phi_{mp} - \sigma_1 - \frac{41}{100} = 0 \\ hmp_{fl} - \frac{41 \phi_{mp}}{100} + \frac{4 \theta_{mp}}{5} - z_{mp} = 0 \\ x_{fr} - x_{mp} + hmp_{fr} \theta_{mp} - \frac{4}{5} = 0 \\ y_{fr} - y_{mp} - hmp_{fr} \phi_{mp} - \sigma_1 + \frac{41}{100} = 0 \\ hmp_{fr} + \frac{41 \phi_{mp}}{100} + \frac{4 \theta_{mp}}{5} - z_{mp} = 0 \\ x_{rl} - x_{mp} + hmp_{rl} \theta_{mp} + \frac{4}{5} = 0 \\ y_{rl} - y_{mp} - hmp_{rl} \phi_{mp} + \sigma_1 - \frac{41}{100} = 0 \\ hmp_{rl} - \frac{41 \phi_{mp}}{100} - \frac{4 \theta_{mp}}{5} - z_{mp} = 0 \end{pmatrix}$$

where

$$\sigma_1 = \frac{4 \phi_{mp} \theta_{mp}}{5}$$

Direct kynematics solution:

```

hmp_rr: hmp_fr - hmp_fl + hmp_rl
x_mp: (5*hmp_fl^2)/8 - (5*hmp_fl*hmp_rl)/4 - (5*hmp_fr*hmp_fl)/8 +
      (5*hmp_rl^2)/8 + (5*hmp_fr*hmp_rl)/8
y_mp: (25*hmp_fl^2)/41 - (75*hmp_fl*hmp_fr)/41 - (25*hmp_rl*hmp_fl)/41 + (50*hmp_fr^2)/41 +
      (25*hmp_rl*hmp_fr)/41
z_mp: hmp_fr/2 + hmp_rl/2
phi_mp: (50*hmp_fl)/41 - (50*hmp_fr)/41
theta_mp: (5*hmp_rl)/8 - (5*hmp_fl)/8
x_fr: (5*hmp_fl^2)/8 - (5*hmp_fl*hmp_rl)/4 + (5*hmp_rl^2)/8 + 4/5
y_fr: -41/100
x_fl: (5*hmp_fl^2)/4 - (15*hmp_fl*hmp_rl)/8 - (5*hmp_fr*hmp_fl)/8 + (5*hmp_rl^2)/8 +
      (5*hmp_fr*hmp_rl)/8 + 4/5
y_fl: (50*hmp_fl^2)/41 - (100*hmp_fl*hmp_fr)/41 + (50*hmp_fr^2)/41 + 41/100
x_rl: (5*hmp_fr*hmp_rl)/8 - (5*hmp_fl*hmp_rl)/8 - (5*hmp_fl*hmp_fr)/8 + (5*hmp_fl^2)/8 - 4/5
y_rl: (50*hmp_fl^2)/41 - (100*hmp_fl*hmp_fr)/41 + (50*hmp_fr^2)/41 + 41/100

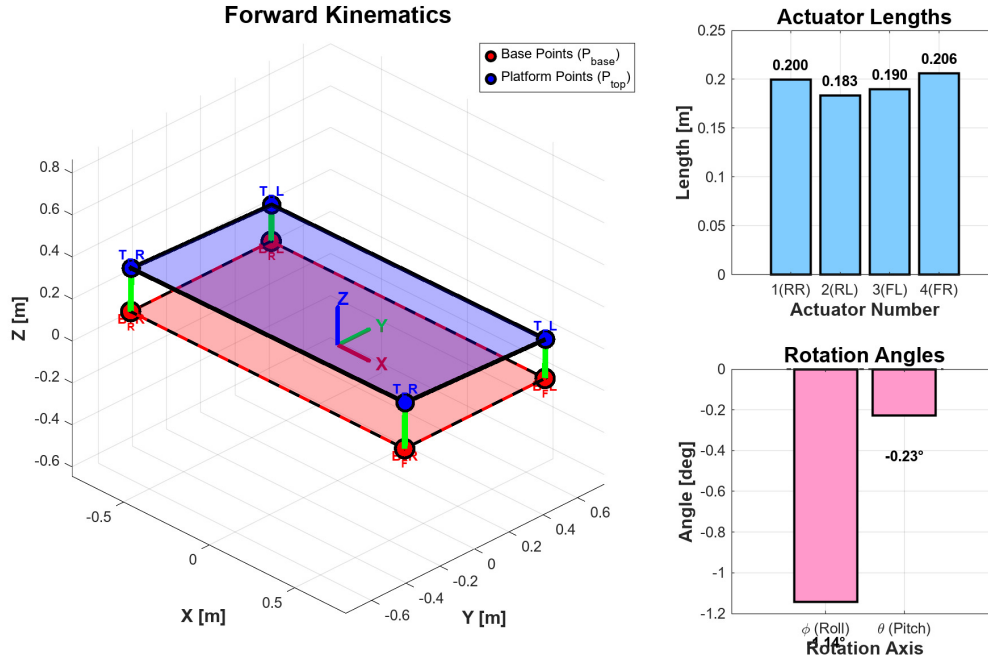
```

Next, the results are plotted for three specified actuator heights:

```

hrr = 0.19;
hrl = 0.189;
hfl = 0.1896;

```



## Inverse kynematics

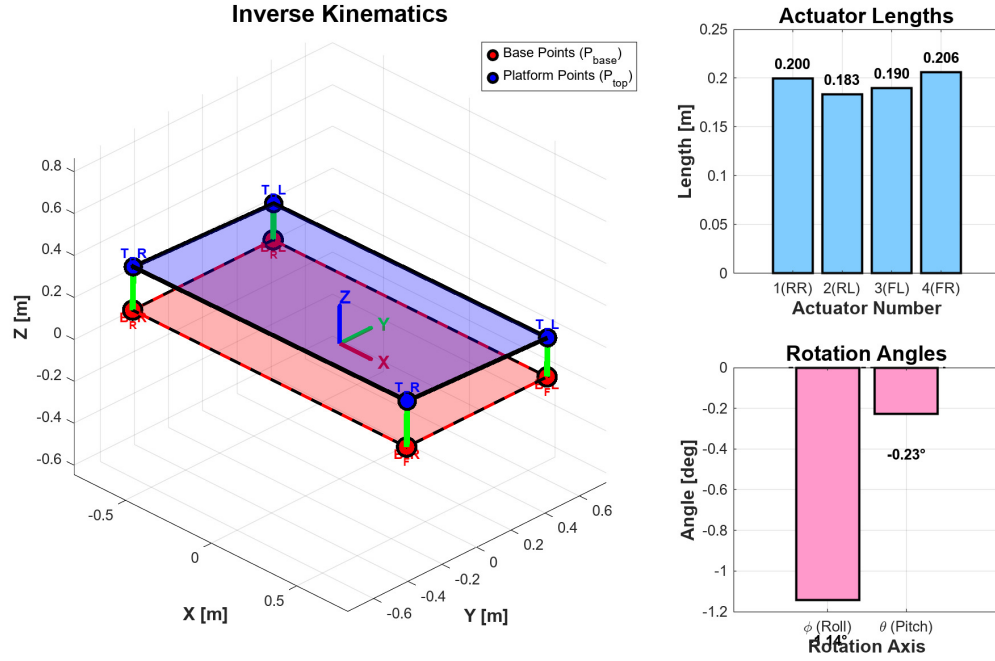
The inverse kinematics can be derived from the same set of equations obtained from the T-Graph. This allows the required actuator lengths to be computed for a given platform pose (position and orientation).

```

hmp_fr: z_mp - (4*theta_mp)/5 - (41*phi_mp)/100
hmp_rr: (4*theta_mp)/5 - (41*phi_mp)/100 + z_mp
hmp_rl: (41*phi_mp)/100 + (4*theta_mp)/5 + z_mp
hmp_fl: (41*phi_mp)/100 - (4*theta_mp)/5 + z_mp
x_mp: (theta_mp*(80*theta_mp - 41*phi_mp + 100*z_mp))/100
y_mp: (41*phi_mp^2)/100 - z_mp*phi_mp
x_fr: (8*theta_mp^2)/5 + 4/5
y_fr: -41/100
x_fl: (8*theta_mp^2)/5 - (41*phi_mp*theta_mp)/50 + 4/5
y_fl: (41*phi_mp^2)/50 + 41/100
x_rl: - (41*phi_mp*theta_mp)/50 - 4/5
y_rl: (41*phi_mp^2)/50 + 41/100

```

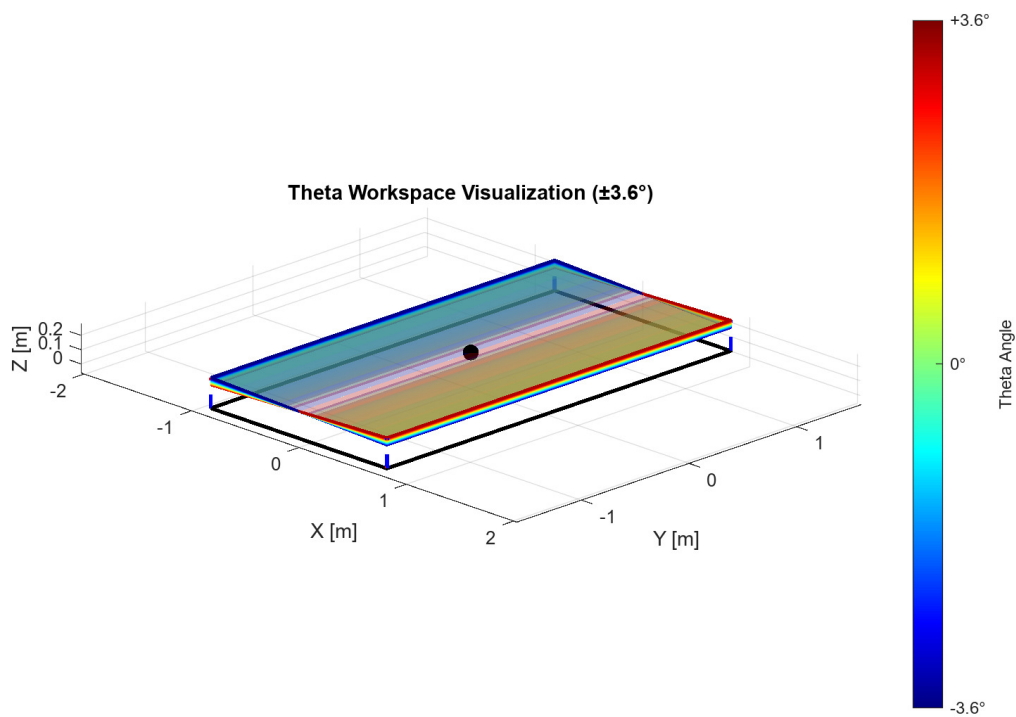
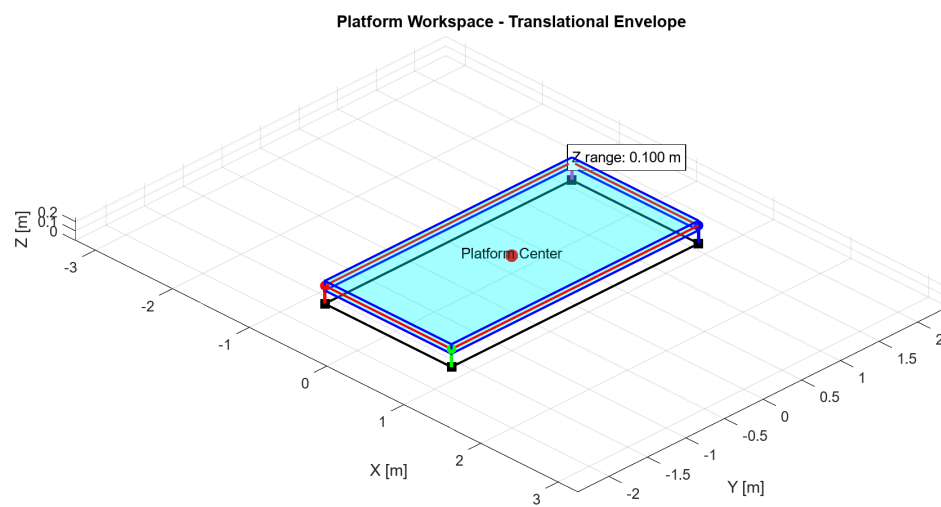
To verify the self-consistency of the models, the  $z$ ,  $\phi$ , and  $\theta$  values obtained from the direct dynamics simulation were used as inputs for the inverse dynamics model. This test confirmed that the resulting system configuration is identical to the original one, thus validating the approach

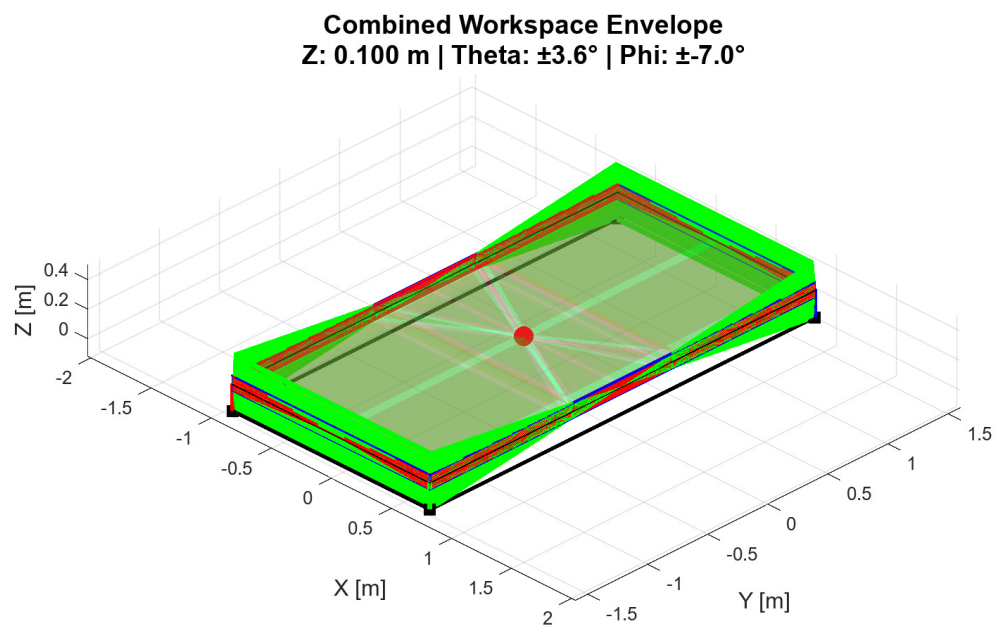
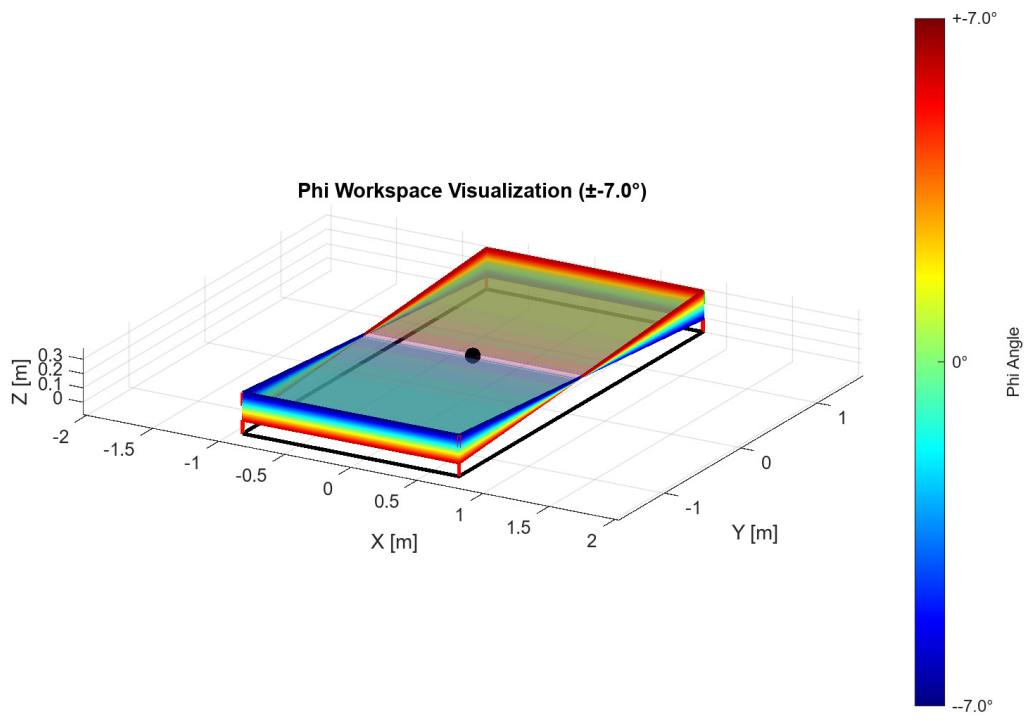


## Motion ranges

The system's workspace is evaluated based on the assumption that each actuator can travel  $\pm 50\%$  of its full stroke from a central position.

The motion ranges are determined by evaluating the system's behavior at maximum and minimum actuator lengths. For small angles, the equations for each actuator's length are linear and independent. Therefore, the principle of superposition is applied: the system's total displacement is calculated as the sum of individual behaviors. This is done by varying one degree of freedom at a time while keeping the other two constant to isolate their effects.





The resulting theoretical motion ranges are summarized as follows:

- $z$  (heave):  $\pm 50mm$
- $\Theta$  (pitch):  $\pm 3.581$  degrees
- $\Phi$  (roll):  $\pm 6.9873$  degrees

This confirms that the small-angle assumptions are valid. The following ranges represent the maximum and minimum for each degree of freedom when the others are inactive. Naturally, if other degrees of freedom have non-zero values, the available range for any given degree of freedom will be more restricted.

A discrepancy was noted between the calculated ranges for pitch and roll and those specified in the datasheet. This difference is resolved by repeating the calculation using the plate dimensions from the datasheet, which are slightly smaller than those provided in the assignment. Using the datasheet dimensions yields results that perfectly match the specified ranges for both angles.

### Pilot

To compute the velocity and acceleration perceived by the driver, a point is defined to represent the position of their head. The driver is modeled as a rigid body (xp,yp,zp), whose movement is not autonomous but rather an indirect effect of the motion of the plate on which it rests. The resulting velocity and acceleration are then calculated at that specific point within the body-fixed frame.

pilot position:

$$\begin{pmatrix} xp + x_{mp}(t) + zp \theta_{mp}(t) \\ yp + y_{mp}(t) - zp \phi_{mp}(t) + xp \phi_{mp}(t) \theta_{mp}(t) \\ zp + z_{mp}(t) + yp \phi_{mp}(t) - xp \theta_{mp}(t) \end{pmatrix}$$

pilot linear velocity

$$\begin{pmatrix} zp \frac{\partial}{\partial t} \theta_{mp}(t) + \frac{\partial}{\partial t} x_{mp}(t) \\ \frac{\partial}{\partial t} y_{mp}(t) - zp \frac{\partial}{\partial t} \phi_{mp}(t) + xp \phi_{mp}(t) \frac{\partial}{\partial t} \theta_{mp}(t) + xp \theta_{mp}(t) \frac{\partial}{\partial t} \phi_{mp}(t) \\ yp \frac{\partial}{\partial t} \phi_{mp}(t) - xp \frac{\partial}{\partial t} \theta_{mp}(t) + \frac{\partial}{\partial t} z_{mp}(t) \end{pmatrix}$$

pilot linear acceleration

$$\begin{pmatrix} zp \sigma_1 + \frac{\partial^2}{\partial t^2} x_{mp}(t) \\ \frac{\partial^2}{\partial t^2} y_{mp}(t) - zp \sigma_2 + 2xp \frac{\partial}{\partial t} \theta_{mp}(t) \frac{\partial}{\partial t} \phi_{mp}(t) + xp \phi_{mp}(t) \sigma_1 + xp \theta_{mp}(t) \sigma_2 \\ yp \sigma_2 - xp \sigma_1 + \frac{\partial^2}{\partial t^2} z_{mp}(t) - \frac{981}{100} \end{pmatrix}$$

where

$$\sigma_1 = \frac{\partial^2}{\partial t^2} \theta_{mp}(t)$$

$$\sigma_2 = \frac{\partial^2}{\partial t^2} \phi_{mp}(t)$$

pilot angular velocity

$$\begin{pmatrix} \frac{\partial}{\partial t} \phi_{mp}(t) \\ \frac{\partial}{\partial t} \theta_{mp}(t) - \phi_{mp}(t) \theta_{mp}(t) \frac{\partial}{\partial t} \phi_{mp}(t) \\ \theta_{mp}(t) \frac{\partial}{\partial t} \phi_{mp}(t) \end{pmatrix}$$



pilot angular acceleration

$$\begin{pmatrix} \theta_{\text{mp}}(t)^2 \sigma_1 + \sigma_1 + 2\theta_{\text{mp}}(t) \frac{\partial}{\partial t} \theta_{\text{mp}}(t) \frac{\partial}{\partial t} \phi_{\text{mp}}(t) \\ \sigma_3 - \theta_{\text{mp}}(t) \sigma_2 - \phi_{\text{mp}}(t) \theta_{\text{mp}}(t) \sigma_1 - \phi_{\text{mp}}(t) \frac{\partial}{\partial t} \theta_{\text{mp}}(t) \frac{\partial}{\partial t} \phi_{\text{mp}}(t) \\ -\theta_{\text{mp}}(t) \phi_{\text{mp}}(t)^2 \sigma_1 + \phi_{\text{mp}}(t) \sigma_3 - 2\theta_{\text{mp}}(t) \phi_{\text{mp}}(t) \sigma_2 - \frac{\partial}{\partial t} \theta_{\text{mp}}(t) \phi_{\text{mp}}(t)^2 \frac{\partial}{\partial t} \phi_{\text{mp}}(t) + \frac{\partial}{\partial t} \theta_{\text{mp}}(t) \frac{\partial}{\partial t} \phi_{\text{mp}}(t) \end{pmatrix}$$

where

$$\sigma_1 = \frac{\partial^2}{\partial t^2} \phi_{\text{mp}}(t)$$

$$\sigma_2 = \left( \frac{\partial}{\partial t} \phi_{\text{mp}}(t) \right)^2$$

$$\sigma_3 = \frac{\partial^2}{\partial t^2} \theta_{\text{mp}}(t)$$

## Dynamic Modeling

To ensure efficient computation time for the dynamic analysis, a reduction in the number of coordinates and constraints is necessary. The methodology is as follows:

1. Selection of Coordinates: A set of generalized coordinates,  $q$ , is chosen to include the platform's vertical position ( $z_{mp}$ ), its pitch ( $\theta_{mp}$ ) and roll ( $\phi_{mp}$ ) angles, and the four actuator heights ( $hmp$ ).
2. Imposing Constraints: The relevant constraint for the generalized coordinates are selected from the T-graph equations. The motion of three actuators is defined as a prescribed function of time. This allows their lengths to be treated as direct inputs for the system analysis.
3. Resulting System: By defining the motion of three actuators, the platform's three degrees of freedom are effectively constrained, reducing the system's final degrees of freedom to zero.

Constraints equations:

$$\begin{pmatrix} hmp_{rr}(t) + \sigma_2 - \sigma_1 - z_{mp}(t) = 0 \\ hmp_{fl}(t) - \sigma_2 + \sigma_1 - z_{mp}(t) = 0 \\ hmp_{fr}(t) + \sigma_2 + \sigma_1 - z_{mp}(t) = 0 \\ hmp_{rl}(t) - \sigma_2 - \sigma_1 - z_{mp}(t) = 0 \\ hmp_{fr}(t) - f_1(t) = 0 \\ hmp_{rl}(t) - f_2(t) = 0 \\ hmp_{fl}(t) - f_3(t) = 0 \end{pmatrix}$$

where

$$\sigma_1 = \frac{4\theta_{mp}(t)}{5}$$

$$\sigma_2 = \frac{41\phi_{mp}(t)}{100}$$

Rank of Jacobian: 7

Number of constraints (M): 7

Number of generalized coordinates (N): 7

The constraints are linearly independent in this configuration.

Number of DOF: 0

As expected, using three additional constraints to control the actuator positions as a function of time has reduced the system's degrees of freedom (DOF) to zero. Otherwise, if the actuators had been controlled by force instead of by adding a constraint, the system would have had three degrees of freedom.

Subsequently, the potential and kinetic energy are expressed as a function of the generalized coordinates in order to use the energy in the Lagrangian formulation:

$$\frac{d}{dt} \left( \frac{dL}{dq_i'} \right) - \frac{dL}{dq_i} - \lambda \frac{d\Phi}{dq_i} = Q_i$$

kinetic energy:

$$\begin{aligned}
& 15 \left( \frac{\partial}{\partial t} \text{hmp}_{\text{fl}}(t) \right)^2 + 15 \left( \frac{\partial}{\partial t} \text{hmp}_{\text{fr}}(t) \right)^2 + 15 \left( \frac{\partial}{\partial t} \text{hmp}_{\text{rl}}(t) \right)^2 + 15 \left( \frac{\partial}{\partial t} \text{hmp}_{\text{rr}}(t) \right)^2 + \\
& \frac{4156625437653019 \left( \frac{\partial}{\partial t} \phi_{\text{mp}}(t) \right)^2}{281474976710656} + \frac{1371470290886227 \left( \frac{\partial}{\partial t} \theta_{\text{mp}}(t) \right)^2}{70368744177664} + 75 \left( \frac{\partial}{\partial t} z_{\text{mp}}(t) \right)^2 + \\
& \frac{192 \left( \phi_{\text{mp}}(t) \frac{\partial}{\partial t} \theta_{\text{mp}}(t) + \theta_{\text{mp}}(t) \frac{\partial}{\partial t} \phi_{\text{mp}}(t) \right)^2}{5}
\end{aligned}$$

potential energy:

$$\frac{2943 \text{hmp}_{\text{fl}}(t)}{10} + \frac{2943 \text{hmp}_{\text{fr}}(t)}{10} + \frac{2943 \text{hmp}_{\text{rl}}(t)}{10} + \frac{2943 \text{hmp}_{\text{rr}}(t)}{10} + \frac{2943 z_{\text{mp}}(t)}{2}$$

Equations of Motion for the Generalized Variables:

```

sol_d = struct with fields:
  a1: lambda1/150 + lambda2/150 + lambda3/150 + lambda4/150 - 981/100
  a2: -(35184372088832*(281151409631676535*lambda1 - 281151409631676535*lambda2 + ...
  a3: -(140737488355328*(103915635941325475*lambda2 - 103915635941325475*lambda1 + ...
  a4: - lambda1/30 - 981/100
  a5: - lambda4/30 - lambda6/30 - 981/100
  a6: - lambda2/30 - lambda7/30 - 981/100
  a7: - lambda3/30 - lambda5/30 - 981/100

```

The resulting system is a set of Differential-Algebraic Equations (DAE), composed of both a differential part (the equations of motion) and an algebraic part (the constraint equations).

### Index Reduction and converting the model for numerical solution

To determine the index of the DAE, we compute the Jacobian of the constraint equations with respect to the Lagrange multipliers. If the Jacobian is not full-rank, the constraints cannot yet be solved for the multipliers, indicating an index greater than 1.

```

max_rank: 7 | Rank: 0 | Determinant is non-zero: false

```

As expected, the Jacobian is singular, confirming that the system is not of index-1. We proceed to differentiate the constraint equations once with respect to time, and substitute the known expressions for the derivatives.

### First derivation

These are the velocity-level (first derivative) constraints. However, their Jacobian is still not full-rank:

```

max_rank: 7 | Rank: 0 | Determinant is non-zero: false

```

### Second derivation

```

max_rank: 7 | Rank: 7 | Determinant is non-zero: true

```

This time, the Jacobian is full-rank and nonsingular. This confirms that the system has been reduced to index-1, and is now in a form suitable for numerical integration. The system is formally of index-3, as two differentiations were required.

Finally, we solve the second-derivative constraint system to obtain explicit expressions for the Lagrange multipliers.

Equation of the lagrange multiplier

```
lambda1: 30*diff(f_3(t), t, t) - 30*diff(f_2(t), t, t) - 30*diff(f_1(t), t, t) - 2943/10
lambda2: 30*phi_mp(t)^2*diff(f_3(t), t, t) - 30*phi_mp(t)^2*diff(f_2(t), t, t) - ...
lambda3: (192000*theta_mp(t)^2*diff(f_1(t), t, t))/1681 - ...
lambda4: 30*phi_mp(t)^2*diff(f_2(t), t, t) - 30*phi_mp(t)^2*diff(f_3(t), t, t) + ...
lambda5: (192000*theta_mp(t)^2*diff(f_3(t), t, t))/1681 - (192000*theta_mp(t)^2* ...
lambda6: 30*phi_mp(t)^2*diff(f_3(t), t, t) - 30*phi_mp(t)^2*diff(f_2(t), t, t) - ...
lambda7: 30*phi_mp(t)^2*diff(f_2(t), t, t) - 30*phi_mp(t)^2*diff(f_3(t), t, t) + ...
```

## System Response Analysis

Once the system has been reduced to index-1, we can numerically integrate it using a standard ODE solver. In this case, we use ode45. Before doing so, it is essential to ensure that the initial state satisfies both the position-level and velocity-level constraints, otherwise numerical errors will grow rapidly over time.

### Sinusoidal

For the system analysis, three functions  $f_1$ ,  $f_2$ , and  $f_3$  are defined. These functions serve as the time-varying length inputs for the actuators.

$$f_1: \frac{\sin(t)}{40} + \frac{3549}{10000}$$

$$f_2: \frac{\sin\left(\frac{11t}{10}\right)}{1000} + \frac{3549}{10000}$$

$$f_3: \frac{3 \sin(t)}{100} + \frac{3549}{10000}$$

The solution to the Ordinary Differential Equation (ODE) is initiated by defining the initial conditions, which are derived from the constraints functions

Initial state of the generalized coordinates and their first-order derivatives:

$$\left( \frac{3549}{10000} \quad 0 \quad 0 \quad \frac{3549}{10000} \quad \frac{3549}{10000} \quad \frac{3549}{10000} \quad \frac{3549}{10000} \quad \frac{261}{20000} \quad \frac{1}{164} \quad -\frac{289}{16000} \quad -\frac{39}{10000} \quad \frac{11}{10000} \quad \frac{3}{100} \quad \frac{1}{40} \right)$$

This check ensures that both the holonomic and hidden (velocity-level) constraints are satisfied at time zero. Failing to meet these would invalidate the results of any integration process.

### Integration

Next, the system of differential equations is defined. This is achieved by substituting the previously solved expressions for the Lagrange multipliers into the DAE, effectively converting the system into a set of Ordinary Differential Equations (ODEs).

The results obtained for the sinusoidal inputs are now presented below.

### Impulse

For the system analysis, three functions  $f_1$ ,  $f_2$ , and  $f_3$  are defined. These functions serve as the time-varying length inputs for the actuators.

Since a true impulse cannot be generated numerically, the sigmoid function was chosen as a smooth and stable approximation that avoids numerical errors.

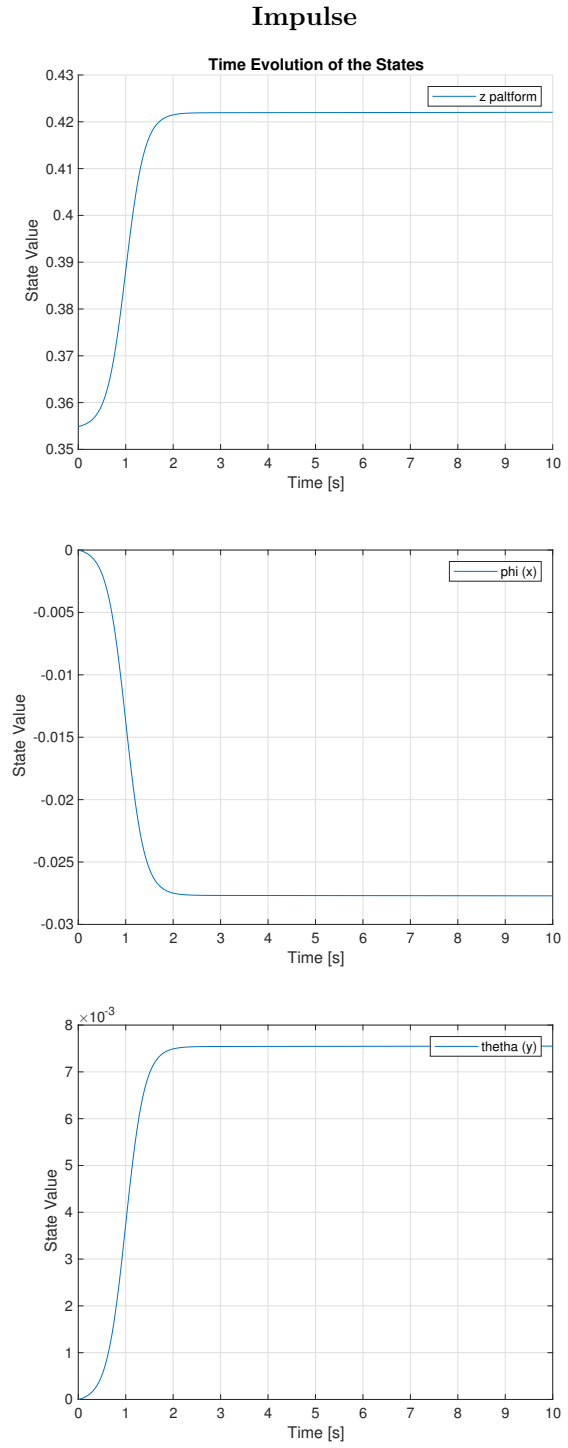
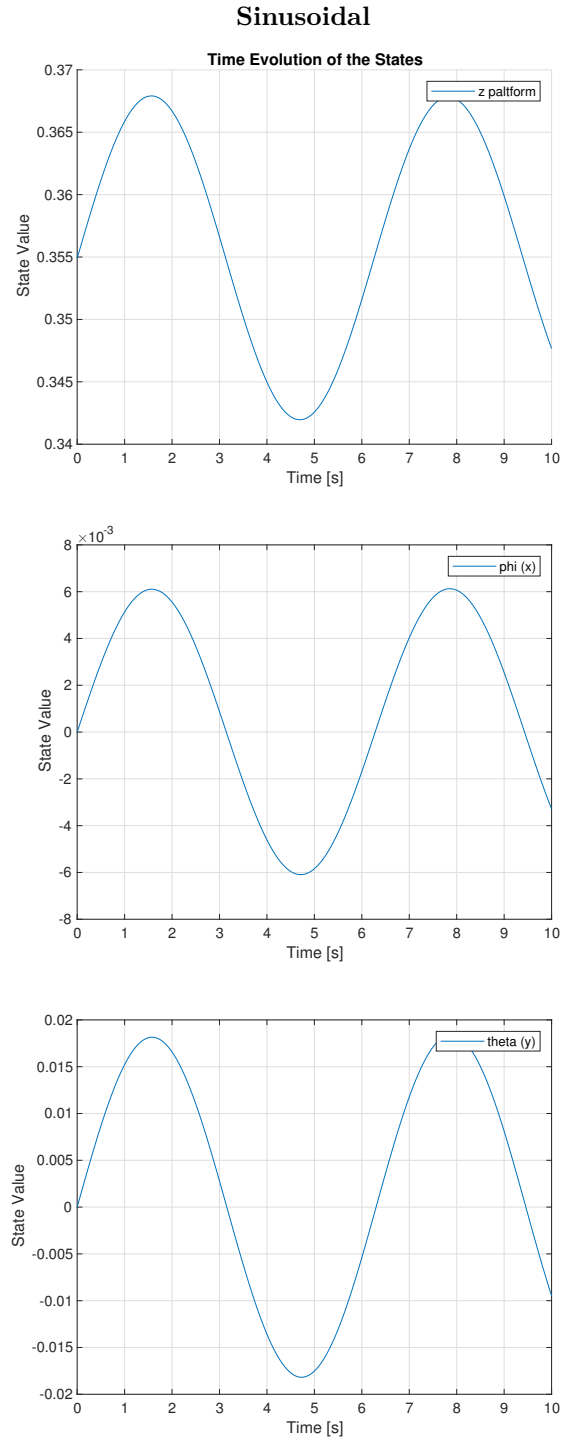
$$f_1: \frac{17}{200(e^{5-5t} + 1)}$$

$$f_2: \frac{1}{20(e^{5-5t} + 1)}$$

$$f_3: \frac{1243}{20000(e^{5-5t} + 1)}$$

## Integration

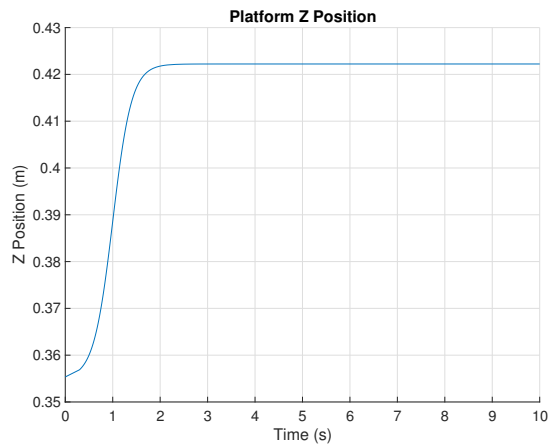
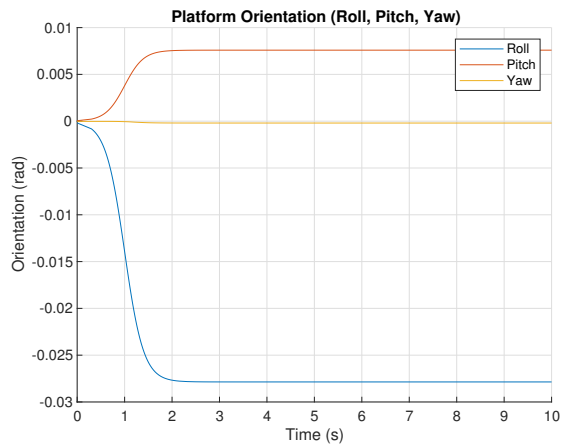
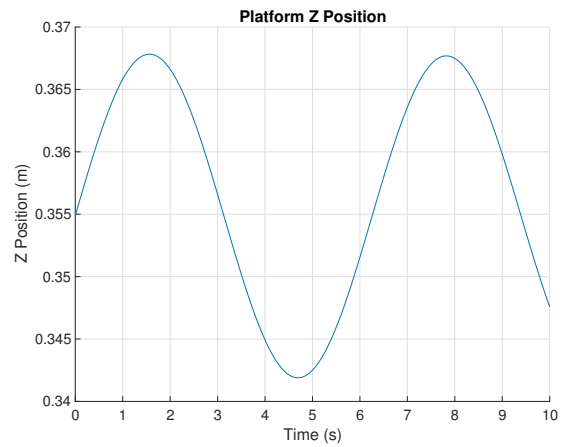
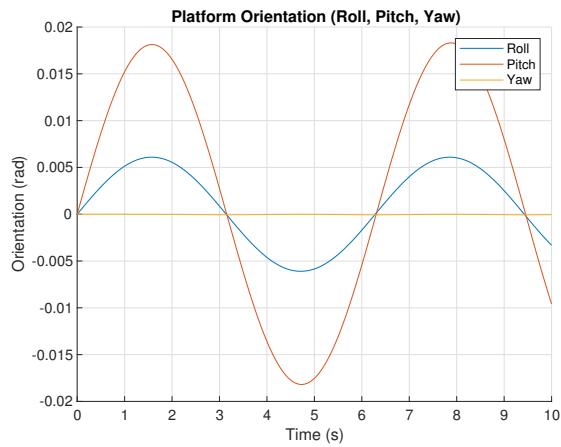
The results obtained for the impulse inputs are now presented below.



## Simscape Model

To illustrate its dynamic response, the attached video shows the platform's behavior when three sinusoidal movements are applied to the actuators.

To validate the mathematical model, the same sinusoidal and impulse inputs were applied to the Simscape model. The resulting platform movement was observed to be identical to the behavior predicted by the previously developed model, confirming its accuracy.



The Simscape model was created following the same considerations as the mathematical model, in order to validate its results. As can be seen from the images, the architecture of the physical model mirrors the matlab one:

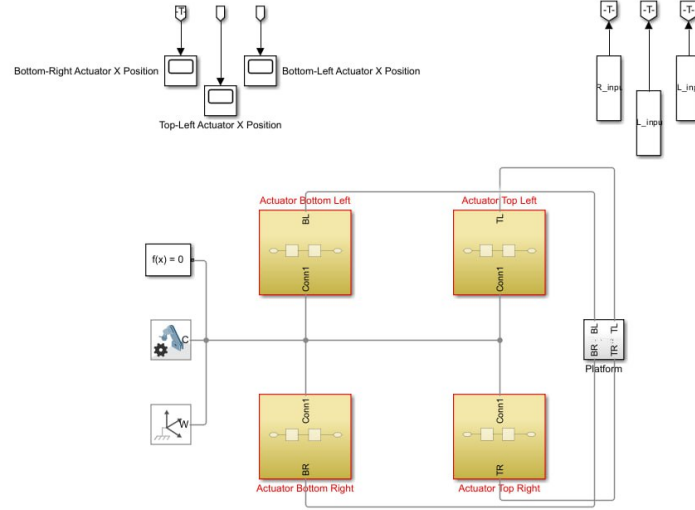


Figure 1: General scheme of the model

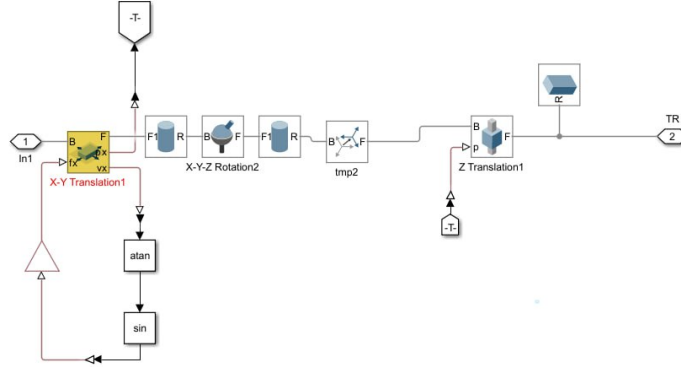


Figure 2: Non fixed actuator model

- For each moving foot, a planar joint was inserted that allows its free movement along the  $x$  and  $y$  axes of the base plane.
- In succession, a revolute joint and a prismatic joint were applied to model the actuator.
- Unlike the mathematical model, a friction force was also applied to the foot, implemented via a feedback loop based on its velocity.

The fact that the results of the two models are identical further confirms the validity of the key assumptions made during the analytical derivation, in particular:

- The small-angle approximation is correct.
- The assumption that the rotation of the actuators is the same as the platform's is valid for the system's kinematics.



# Conclusion

Despite its utility, the developed model has inherent limitations that distinguish it from the behavior of a real system. These limitations stem from both the physical configuration of the platform and the assumptions made during the modeling phase.

## 1. Degrees of Freedom (3-DOF vs. 6-DOF)

The most significant limitation of the system lies in its 3 degrees of freedom (heave, pitch, roll), compared to the 6-DOF of a real vehicle. The absence of lateral (sway), longitudinal (surge), and vertical axis (yaw) rotation prevents the direct reproduction of fundamental driving sensations, such as:

- Sustained acceleration or braking (related to surge).
- Lateral slip or centrifugal forces in a turn (related to sway).
- Vehicle rotation or yaw (related to yaw).

## 2. Physical Constraints and Motion Cueing Algorithms

The platform operates within defined physical limits, which prevent it from replicating the movements of a real vehicle on a 1:1 scale. Added to this are the intrinsic limits of the actuators, whose maximum velocity ( $800\text{mm/s}$ ) and acceleration ( $0.8g$ ) are specified in the datasheet.

These physical constraints necessitate the use of motion cueing algorithms. Such algorithms are tasked with "translating" the vehicle's dynamics and accelerations into platform movements that, while remaining within the limited workspace, manage to provide the driver with a plausible sense of motion. However, this introduces two compromises:

- High-Frequency Events: The platform's ability to reproduce road vibrations or rapid impacts is limited by the maximum acceleration of the actuators.
- Command Saturation: If a simulated maneuver requires an acceleration exceeding  $0.8\text{ g}$ , the control algorithm must "clip" or filter the command, delivering a movement that is compatible with the physical limits but loses realism for the driver.

## 3. Modeling Assumptions

The model is based on several key assumptions that, although justified to simplify the analysis, introduce discrepancies with real-world behavior:

- Small-Angle Approximation: Although verified a posteriori, this remains a simplification compared to the real, nonlinear kinematics, especially during large maneuvers.
- Platform as a Rigid Body: This assumption ignores possible structural flexibilities or deformations of the platform, which could introduce secondary dynamics.
- Friction Neglected: Is possible that the friction (viscous in the actuators or sliding in the joints) plays a non-negligible role. The absence of friction in this model leads to an unrealistic estimation of energy consumption and long-term positioning accuracy.

Every assumption, however small, introduces a discrepancy between the model and reality. However, the choice of the level of approximation represents a fundamental engineering trade-off between model fidelity and its computational complexity. The definition of this balance is up to the modeler, in accordance with the specific goals and requirements of the project.