AMORO: Advanced and Robot Modelling
Kinematics and dynamics of a biglide

Date: Wed 23:59, 09.11.16

Prof. S. Caro, Prof. S. Briot, MSc D. Six

Rabbia Asghar, BEng, Ernest Skrzypczyk, BSc

Contents

List of Figures

List of Tables

Experimental setup

The biglide planar parallel robot shown in figure (1).

Table 1: Parametrization of simulated biglide parallel robot

d	l_{A_1C}	l_{A_2C}	m_{f1}	m_{f2}	m_p
0.4 $[m]$	0.3606 $[m]$	0.3606 $[m]$	1.0 $[kg]$	1.0 $[kg]$	3.0 $[kg]$

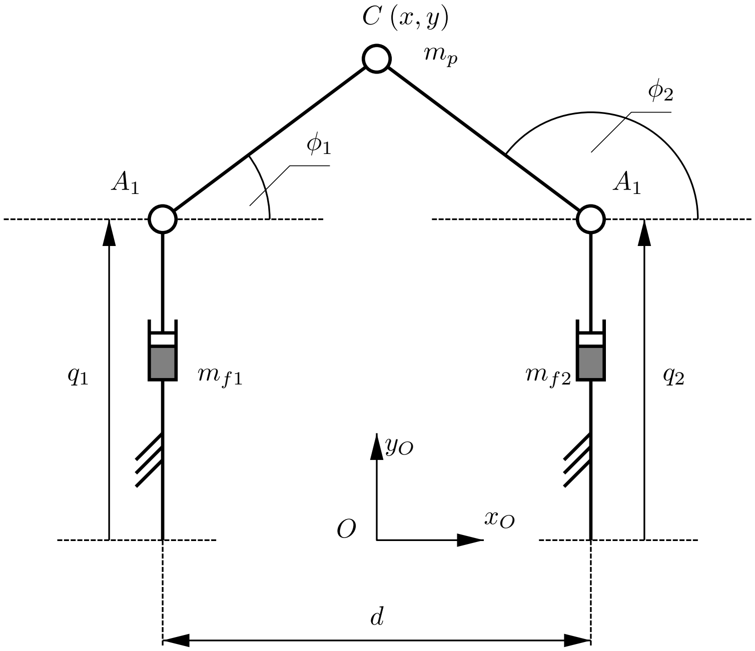


Figure 1: Kinematic model of a biglide parallel robot used for simulation.

The system variables are defined as follows in (1).

$$\underline{\mathbf{x}} = \begin{bmatrix} x \\ y \end{bmatrix},\ \underline{\mathbf{q_a}} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix},\ \underline{\mathbf{q_d}} = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \tag{1}$$

Chapter 1

Kinematic analysis

Geometric analysis

Direct geometric model

Starting from the constraint equations (1.1) and (1.2) the geometric model is developed through (1.3) and (1.4) by rearrangements and squaring of (1.1) and (1.2). The final direct geometric model (1.5) is derived from rearranged (1.3) and by substituting $\underline{\mathbf{x}}$.

$$\underline{\mathbf{x}} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -{}^{d}\!/\!{}_{2} \\ q_1 \end{bmatrix} + l_{A_1C} \begin{bmatrix} cos(\phi_1) \\ sin(\phi_1) \end{bmatrix} \tag{1.1}$$

$$\underline{\mathbf{x}} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} {}^{d}\!/\!{}_{2} \\ q_2 \end{bmatrix} + l_{A_2C} \begin{bmatrix} cos(\phi_2) \\ sin(\phi_2) \end{bmatrix} \tag{1.2}$$

$$\begin{array}{rcl} (x + {}^{d}\!/\!{}_{2})^2 + (y - q_1)^2 & = & l^2_{A_1C} \\ (x - {}^{d}\!/\!{}_{2})^2 + (y - q_2)^2 & = & l^2_{A_2C} \end{array} \tag{1.3}$$

$$\begin{array}{rcl} (x + {}^{d}\!/\!{}_{2})^2 + y^2 & = & l^2_{A_1C} cos(\phi_1) + (l^2_{A_1C} sin(\phi_1) + q_1)^2 \\ (x - {}^{d}\!/\!{}_{2})^2 + y^2 & = & l^2_{A_2C} cos(\phi_2) + (l^2_{A_2C} sin(\phi_2) + q_2)^2 \end{array} \tag{1.4}$$

$$\underline{\mathbf{x}} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{l^2_{A_1C} - l^2_{A_2C} + q_1^2 - q_2^2 + 2q_1 l_{A_1C} sin(\phi_1) - 2q_2 l_{A_2C} sin(\phi 2)}{2d} \\ \pm\sqrt{l^2_{A_1C} - (x + {}^{d}\!/\!{}_{2})^2} + q1 = \pm\sqrt{l^2_{A_2C} - (x - {}^{d}\!/\!{}_{2})^2} + q2 \end{bmatrix} \tag{1.5}$$

Because there are 2 solutions available for the direct geometric model, for the derivation of point C coordinates the distance between the joints A_1 and A_2 with the vector $\overrightarrow{\mathbf{MC}}$, normal to $l_{A_1A_2}$, was used as shown in (1.1). Depending on the value of γ the assembly mode, ergo the solution for the direct geometric model, can be chosen between C and C'.

The vector $\overrightarrow{\mathbf{A_2A_1}}$ defined in (1.6) is used for specifying $\overrightarrow{\mathbf{A_2M}}$ in (1.7), which then using normal vectors in (1.8) allows for definition of $\overrightarrow{\mathbf{MC}}$ in (1.9), where γ is an assembly mode parameter.

$$\overrightarrow{\mathbf{A_2A_1}} = \overrightarrow{\mathbf{OA_2}} - \overrightarrow{\mathbf{OA_1}} = \begin{bmatrix} -{}^{d}\!/\!{}_{2} \\ q_1 \end{bmatrix} - \begin{bmatrix} {}^{d}\!/\!{}_{2} \\ q_2 \end{bmatrix} = \begin{bmatrix} -d \\ q_1 - q_2 \end{bmatrix} \tag{1.6}$$

$$\overrightarrow{\mathbf{A_2M}} = \frac{1}{2}\overrightarrow{\mathbf{A_2A_1}} = \frac{1}{2} \begin{bmatrix} -d \\ q_1 - q_2 \end{bmatrix} \tag{1.7}$$

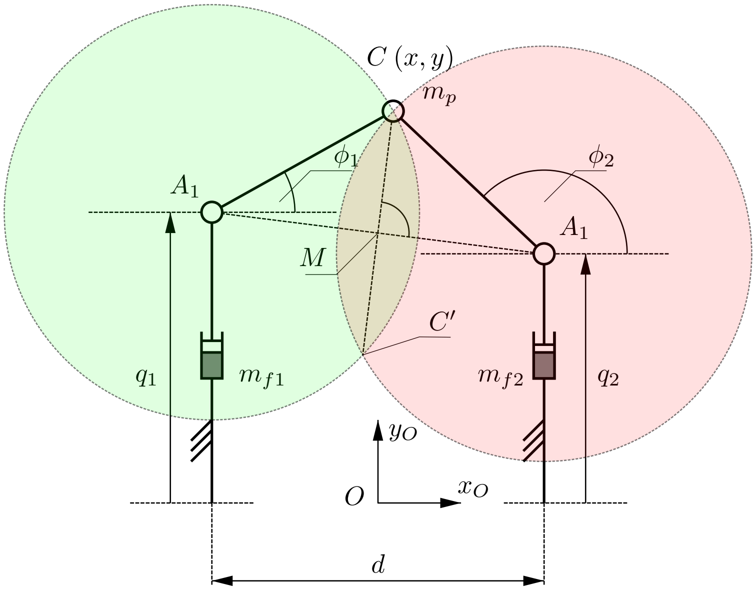


Figure 1.1: Kinematic model of a biglide parallel robot with configuration parametrization.

Movement constraint of the 🟩 – left upper arm, 🟥 – right upper arm

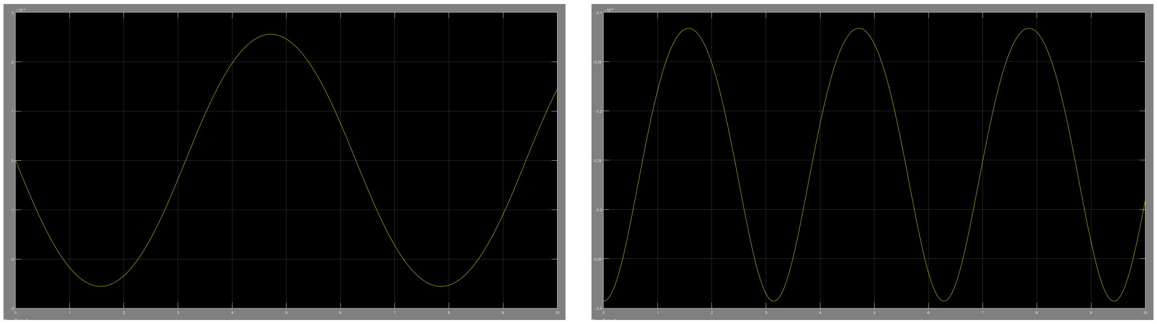
$$\mathbf{n}_{\overrightarrow{\mathbf{MC}}} = \sqrt{\mathbf{n}^2_{\overrightarrow{\mathbf{A_2C}}} - \mathbf{n}^2_{\overrightarrow{\mathbf{A_2M}}}} = \sqrt{l^2_{A_2C} - \mathbf{n}^2_{\overrightarrow{\mathbf{A_2M}}}} \tag{1.8}$$

$$\overrightarrow{\mathbf{MC}} = \gamma \frac{\mathbf{n}_{\overrightarrow{\mathbf{MC}}}}{\mathbf{n}_{\overrightarrow{\mathbf{A_2M}}}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \overrightarrow{\mathbf{A_2M}} \tag{1.9}$$

The direct geometric model is then described with accordance to the above derivation in (1.10).

$$\underline{\mathbf{x}} = \begin{bmatrix} x \\ y \end{bmatrix} = \overrightarrow{\mathbf{OC}} = \overrightarrow{\mathbf{OA_2}} + \overrightarrow{\mathbf{A_2M}} + \overrightarrow{\mathbf{MC}} = \begin{bmatrix} d/2 \\ q_2 \end{bmatrix} + \begin{bmatrix} -d \\ q_1 - q_2 \end{bmatrix} + \overrightarrow{\mathbf{MC}} = \begin{bmatrix} -d/2 \\ q_1 \end{bmatrix} + \overrightarrow{\mathbf{MC}} \tag{1.10}$$

Script file dgm.m describes the direct geometric model according to (1.10). The direct geometric model was then simulated in Simulink and compared against the outputs obtained from Adams. Same sinusoidal input was given to active joints in the Simulink and in Adams. Errors of order 10^{-5} were observed that confirmed that the model was correctly defined. The plots are shown in the figure (1.3).



(a) Error in x.

(b) Error in y.

Figure 1.3: Error in x and y between Simulink direct geometric model and Adams plant for same input.

Inverse geometric model

From the constraint equations (1.1) and (1.2) the inverse geometric model is defined in (1.11), which is described in file igm.m where γ_1 and γ_2 are used to select respective working mode. Inverse geometric model was used in kinematic and dynamic control for trajectory tracking.

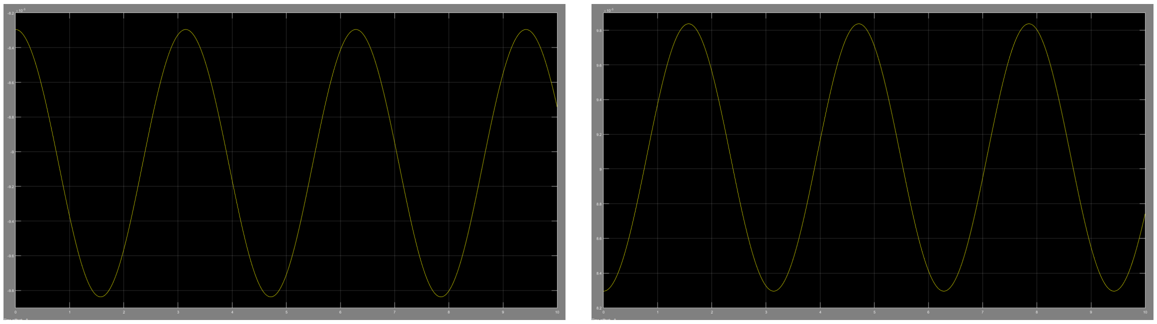
$$\underline{\mathbf{q_a}} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} y - \gamma_l\sqrt{(l_{A1C}^2 - (x + d/2)^2)} \\ y - \gamma_2\sqrt{(l_{A2C}^2 - (x - d/2)^2)} \end{bmatrix} \tag{1.11}$$

Passive angles position

Similarly from the constraint equations (1.1) and (1.2) the passive angles positions are defined in (1.12), which file pa.m defines.

$$\underline{\mathbf{q_d}} = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} tan^{-1}(\frac{y-q_1}{x+{}^{d}/_{2}})^2 \\ tan^{-1}(\frac{y-q_2}{x-{}^{d}/_{2}})^2 \end{bmatrix} \tag{1.12}$$

The passive angles model was then simulated in Simulink and compared against the outputs obtained from Adams. Same sinusoidal input was given to active joints in the Simulink and in Adams. Errors of order 10^{-5} confirmed that the model was correctly defined. The plots are shown in the figure (1.4).



(a) Error in ϕ_1.

(b) Error in ϕ_2.

Figure 1.4: Error in ϕ_1 and ϕ_2 between Simulink direct geometric model and Adams plant for same input.

Kinematic analysis

Direct kinematic model

The direct kinematic model is derived from the equilibrium (1.13), which using (1.3), thus (1.14), leads to (1.15) and further (1.16), which is stored in dkm.m.

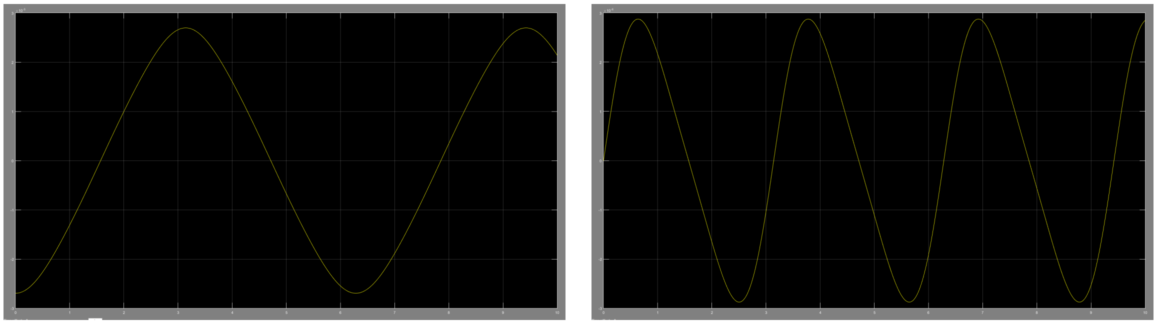
$$\begin{aligned} A\dot{\underline{\mathbf{x}}} + B\dot{\underline{\mathbf{q_a}}} &= 0 \\ \left[\frac{\delta h(\underline{\mathbf{x}}, \underline{\mathbf{q_a}})}{\delta \underline{\mathbf{x}}}\right] \dot{\underline{\mathbf{x}}} + \left[\frac{\delta h(\underline{\mathbf{x}}, \underline{\mathbf{q_a}})}{\delta \underline{\mathbf{q_a}}}\right] \dot{\underline{\mathbf{q_a}}} &= 0 \end{aligned} \tag{1.13}$$

$$\begin{aligned} h(\underline{\mathbf{x}}, \underline{\mathbf{q_a}}) &= 0 \\ h_1(\underline{\mathbf{x}}, \underline{\mathbf{q_a}}) &= (x + {}^{d}/_{2})^2 + (y - q_1)^2 - l_{A_1C}^2 = 0 \\ h_2(\underline{\mathbf{x}}, \underline{\mathbf{q_a}}) &= (x - {}^{d}/_{2})^2 + (y - q_2)^2 - l_{A_2C}^2 = 0 \end{aligned} \tag{1.14}$$

$$\begin{bmatrix} x + {}^{d}/_{2} & y - q_1 \\ x - {}^{d}/_{2} & y - q_2 \end{bmatrix} \dot{\underline{\mathbf{x}}} + \begin{bmatrix} q_1 - y & 0 \\ 0 & q_2 - y \end{bmatrix} \dot{\underline{\mathbf{q_a}}} = 0 \tag{1.15}$$

$$\underline{\dot{\mathbf{x}}} = -A^{-1}B\underline{\dot{\mathbf{q}}_{\mathbf{a}}} = -\begin{bmatrix} x + {}^{d}\!/_{2} & y - q_1 \\ x - {}^{d}\!/_{2} & y - q_2 \end{bmatrix}^{-1} \begin{bmatrix} q_1 - y & 0 \\ 0 & q_2 - y \end{bmatrix} \underline{\dot{\mathbf{q}}_{\mathbf{a}}} \tag{1.16}$$

The direct kinematic model was then simulated in Simulink and compared against the outputs obtained from Adams. Same sinusoidal input was given to active joints in the Simulink and in Adams. Errors of order 10^{-5} confirmed that the model was correctly derived. The plots are shown in the figure (1.5).



(a) Error in $\dot{x}$.

(b) Error in $\dot{y}$.

Figure 1.5: Error in $\dot{x}$ and $\dot{y}$ between Simulink direct kinematic model and Adams plant for same input.

Passive angles velocities model

The premise for the direct kinematic model (1.13) can be expanded by passive angles constraints into (1.17), which using (1.4) and therefore (1.18), leads to (1.19) and further (1.20) as defined in file pa2.m. Since the angle or orientation of the end effector is neglected there is no need for a further constraint specification.

$$\begin{aligned} A\underline{\dot{\mathbf{x}}} + B\underline{\dot{\mathbf{q}}_{\mathbf{a}}} + C\underline{\dot{\mathbf{q}}_{\mathbf{d}}} &= 0 \\ \left[\frac{\delta h(\underline{\mathbf{x}},\underline{\mathbf{q_a}},\underline{\mathbf{q_d}})}{\delta \underline{\mathbf{x}}}\right]\underline{\dot{\mathbf{x}}} + \left[\frac{\delta h(\underline{\mathbf{x}},\underline{\mathbf{q_a}},\underline{\mathbf{q_d}})}{\delta \underline{\mathbf{q_a}}}\right]\underline{\dot{\mathbf{q}}_{\mathbf{a}}} + \left[\frac{\delta h(\underline{\mathbf{x}},\underline{\mathbf{q_a}},\underline{\mathbf{q_d}})}{\delta \underline{\mathbf{q_d}}}\right]\underline{\dot{\mathbf{q}}_{\mathbf{d}}} &= 0 \end{aligned} \tag{1.17}$$

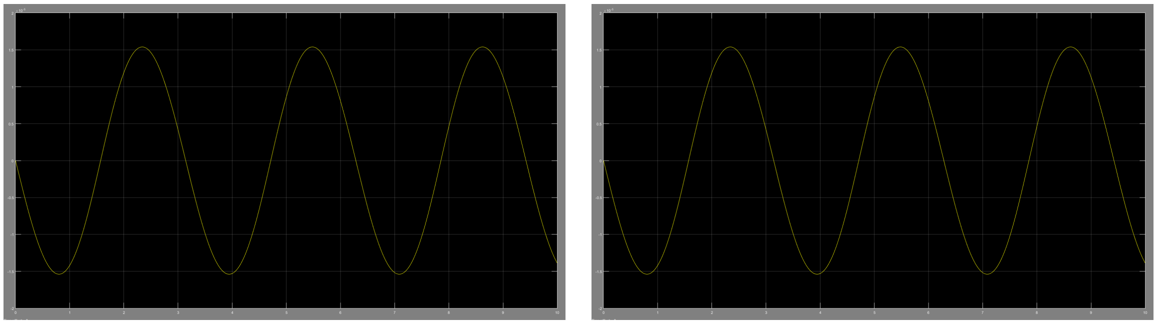
$$\begin{aligned} h(\underline{\mathbf{x}},\underline{\mathbf{q_a}},\underline{\mathbf{q_d}}) &= 0 \\ h_1(\underline{\mathbf{x}},\underline{\mathbf{q_a}},\underline{\mathbf{q_d}}) &= (x + {}^{d}\!/_{2})^2 + y^2 - l_{A_1C}^2 - 2l_{A_1C}q_1 sin(\phi_1) - q_1^2 = 0 \\ h_2(\underline{\mathbf{x}},\underline{\mathbf{q_a}},\underline{\mathbf{q_d}}) &= (x - {}^{d}\!/_{2})^2 + y^2 - l_{A_2C}^2 - 2l_{A_2C}q_2 sin(\phi_2) - q_2^2 = 0 \end{aligned} \tag{1.18}$$

$$\begin{bmatrix} x + {}^{d}\!/_{2} & y \\ x - {}^{d}\!/_{2} & y \end{bmatrix}\underline{\dot{\mathbf{x}}} + \begin{bmatrix} -l_{A_1C}sin(\phi_1) - q_1 & 0 \\ 0 & -l_{A_2C}sin(\phi_2) - q_2 \end{bmatrix}\underline{\dot{\mathbf{q}}_{\mathbf{a}}} + \begin{bmatrix} -l_{A_1C}q_1cos(\phi_1) & 0 \\ 0 & -l_{A_2C}q_2cos(\phi_2) \end{bmatrix}\underline{\dot{\mathbf{q}}_{\mathbf{d}}} = 0 \tag{1.19}$$

$$\underline{\dot{\mathbf{q}}_{\mathbf{d}}} = -C^{-1}(A\underline{\dot{\mathbf{x}}} + B\underline{\dot{\mathbf{q}}_{\mathbf{a}}})$$

$$\underline{\dot{\mathbf{q}}_{\mathbf{d}}} = -\begin{bmatrix} -l_{A_1C}q_1cos(\phi_1) & 0 \\ 0 & -l_{A_2C}q_2cos(\phi_2) \end{bmatrix}^{-1}\left(\begin{bmatrix} x + {}^{d}\!/_{2} & y \\ x - {}^{d}\!/_{2} & y \end{bmatrix}\underline{\dot{\mathbf{x}}} + \begin{bmatrix} -l_{A_1C}sin(\phi_1) - q_1 & 0 \\ 0 & -l_{A_2C}sin(\phi_2) - q_2 \end{bmatrix}\underline{\dot{\mathbf{q}}_{\mathbf{a}}}\right) \tag{1.20}$$

The passive angles velocity model was then simulated in Simulink and compared against the outputs obtained from Adams. Same sinusoidal input was given to active joints in the Simulink and in Adams. Errors of order 10^{-5} confirmed that the model was correctly defined. The plots are shown in the figure (1.6).



(a) Error in $\dot{\phi_1}$.

(b) Error in $\dot{\phi_2}$.

Figure 1.6: Error in $\dot{\phi_1}$ and $\dot{\phi_2}$ between Simulink passive angle velocities model and Adams plant for same input.

Acceleration analysis

Second order direct kinematic model

The second order direct kinematic model is derived from the direct kinematic model (1.16) by differentiating in time starting from (1.21), leading to (1.22) and further (1.23) found in file script dkm2.m.

$$\begin{aligned} \tfrac{d}{dt}(A\underline{\dot{\mathbf{x}}} + B\underline{\dot{\mathbf{q}}_{\mathbf{a}}}) &= \\ \dot{A}\underline{\dot{\mathbf{x}}} + A\underline{\ddot{\mathbf{x}}} + \dot{B}\underline{\dot{\mathbf{q}}_{\mathbf{a}}} + B\underline{\ddot{\mathbf{q}}_{\mathbf{a}}} &= 0 \end{aligned} \tag{1.21}$$

$$\begin{bmatrix} \dot{x} & \dot{y} - \dot{q_1} \\ \dot{x} & \dot{y} - \dot{q_2} \end{bmatrix} \underline{\dot{\mathbf{x}}} + \begin{bmatrix} x + {}^{d}\!/_{2} & y - q_1 \\ x - {}^{d}\!/_{2} & y - q_2 \end{bmatrix} \underline{\ddot{\mathbf{x}}} + \begin{bmatrix} \dot{q_1} - \dot{y} & 0 \\ 0 & \dot{q_2} - \dot{y} \end{bmatrix} \underline{\dot{\mathbf{q}}_{\mathbf{a}}} + \begin{bmatrix} q_1 - y & 0 \\ 0 & q_2 - y \end{bmatrix} \underline{\ddot{\mathbf{q}}_{\mathbf{a}}} = 0 \tag{1.22}$$

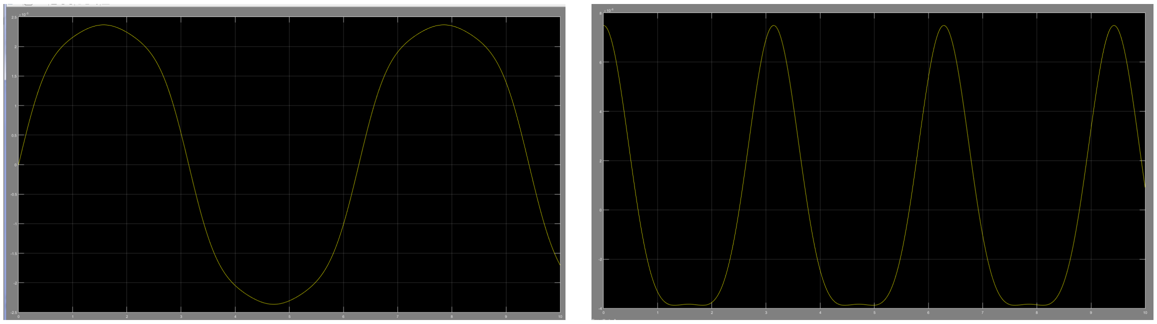
$$\begin{gathered} \underline{\ddot{\mathbf{x}}} = -A^{-1}(\dot{A}\underline{\dot{\mathbf{x}}} + \dot{B}\underline{\dot{\mathbf{q}}_{\mathbf{a}}} + B\underline{\ddot{\mathbf{q}}_{\mathbf{a}}}) = \\ -\begin{bmatrix} x + {}^{d}\!/_{2} & y - q_1 \\ x - {}^{d}\!/_{2} & y - q_2 \end{bmatrix}^{-1} \left(\begin{bmatrix} \dot{x} & \dot{y} - \dot{q_1} \\ \dot{x} & \dot{y} - \dot{q_2} \end{bmatrix} \underline{\dot{\mathbf{x}}} + \begin{bmatrix} \dot{q_1} - \dot{y} & 0 \\ 0 & \dot{q_2} - \dot{y} \end{bmatrix} \underline{\dot{\mathbf{q}}_{\mathbf{a}}} + \begin{bmatrix} q_1 - y & 0 \\ 0 & q_2 - y \end{bmatrix} \underline{\ddot{\mathbf{q}}_{\mathbf{a}}} \right) \end{gathered} \tag{1.23}$$

The direct second order kinematic model was then simulated in Simulink and compared against the outputs obtained from Adams. Same sinusoidal input was given to active joints in the Simulink and in Adams. Errors of order 10^{-5} confirmed that the model was correctly defined. The plots are shown in the figure (1.7).

Passive angles acceleration model

As with velocities analysis, the second order direct kinematic model can be expanded by passive angles constraints into (1.24), which leads to (1.25) and further (1.26) as defined in file pa3.m. Since the angle or orientation of the end effector is neglected there is no need for a further constraint specification.

$$\begin{aligned} \tfrac{d}{dt}\left(A\underline{\dot{\mathbf{x}}} + B\underline{\dot{\mathbf{q}}_{\mathbf{a}}} + C\underline{\dot{\mathbf{q}}_{\mathbf{d}}}\right) &= \\ \dot{A}\underline{\dot{\mathbf{x}}} + A\underline{\ddot{\mathbf{x}}} + \dot{B}\underline{\dot{\mathbf{q}}_{\mathbf{a}}} + B\underline{\ddot{\mathbf{q}}_{\mathbf{a}}} + \dot{C}\underline{\dot{\mathbf{q}}_{\mathbf{d}}} + C\underline{\ddot{\mathbf{q}}_{\mathbf{d}}} &= 0 \end{aligned} \tag{1.24}$$



(a) Error in $\ddot{x}$.

(b) Error in $\ddot{y}$.

Figure 1.7: Error in $\ddot{x}$ and $\ddot{y}$ between Simulink direct second order kinematic model and Adams plant for same input.

$$\begin{bmatrix}\dot{x} & \dot{y}\\ \dot{x} & \dot{y}\end{bmatrix}\underline{\dot{\mathbf{x}}}+\begin{bmatrix}x+{}^{d}\!/_{2} & y\\ x-{}^{d}\!/_{2} & y\end{bmatrix}\underline{\ddot{\mathbf{x}}}+$$
$$\begin{bmatrix}-l_{A_1C}cos(\phi_1)\dot{\phi_1}-\dot{q_1} & 0\\ 0 & -l_{A_2C}cos(\phi_2)\dot{\phi_2}-\dot{q_2}\end{bmatrix}\underline{\dot{\mathbf{q}}_{\mathbf{a}}}+\begin{bmatrix}-l_{A_1C}sin(\phi_1)-q_1 & 0\\ 0 & -l_{A_2C}sin(\phi_2)-q_2\end{bmatrix}\underline{\ddot{\mathbf{q}}_{\mathbf{a}}}+$$
$$\begin{bmatrix}l_{A_1C}sin(\phi_1)\dot{\phi_1} & 0\\ 0 & -l_{A_2C}cos(\phi_2)\end{bmatrix}\underline{\dot{\mathbf{q}}_{\mathbf{d}}}+\begin{bmatrix}-l_{A_1C}cos(\phi_1) & 0\\ 0 & -l_{A_2C}cos(\phi_2)\end{bmatrix}\underline{\ddot{\mathbf{q}}_{\mathbf{d}}}=0 \tag{1.25}$$

$$\underline{\ddot{\mathbf{q}}_{\mathbf{d}}}=-C^{-1}\left(\dot{A}\underline{\dot{\mathbf{x}}}+A\underline{\ddot{\mathbf{x}}}+\dot{B}\underline{\dot{\mathbf{q}}_{\mathbf{a}}}+B\underline{\ddot{\mathbf{q}}_{\mathbf{a}}}+\dot{C}\underline{\dot{\mathbf{q}}_{\mathbf{d}}}\right)$$

$$\underline{\ddot{\mathbf{q}}_{\mathbf{d}}}=-\begin{bmatrix}-l_{A_1C}cos(\phi_1) & 0\\ 0 & -l_{A_2C}cos(\phi_2)\end{bmatrix}^{-1}\left(\begin{bmatrix}\dot{x} & \dot{y}\\ \dot{x} & \dot{y}\end{bmatrix}\underline{\dot{\mathbf{x}}}+\begin{bmatrix}x+{}^{d}\!/_{2} & y\\ x-{}^{d}\!/_{2} & y\end{bmatrix}\underline{\ddot{\mathbf{x}}}+\right.$$
$$\begin{bmatrix}-l_{A_1C}cos(\phi_1)\dot{\phi_1}-\dot{q_1} & 0\\ 0 & -l_{A_2C}cos(\phi_2)\dot{\phi_2}-\dot{q_2}\end{bmatrix}\underline{\dot{\mathbf{q}}_{\mathbf{a}}}+\begin{bmatrix}-l_{A_1C}sin(\phi_1)-q_1 & 0\\ 0 & -l_{A_2C}sin(\phi_2)-q_2\end{bmatrix}\underline{\ddot{\mathbf{q}}_{\mathbf{a}}}+$$
$$\left.\begin{bmatrix}l_{A_1C}sin(\phi_1)\dot{\phi_1} & 0\\ 0 & -l_{A_2C}cos(\phi_2)\end{bmatrix}\underline{\dot{\mathbf{q}}_{\mathbf{d}}}\right) \tag{1.26}$$

The passive angles acceleration model was then simulated in Simulink and compared against the outputs obtained from Adams. Same sinusoidal input was given to active joints in the Simulink and in Adams. Errors of order 10^{-5} confirmed that the model was correctly defined. The plots are shown in the figure (1.6).

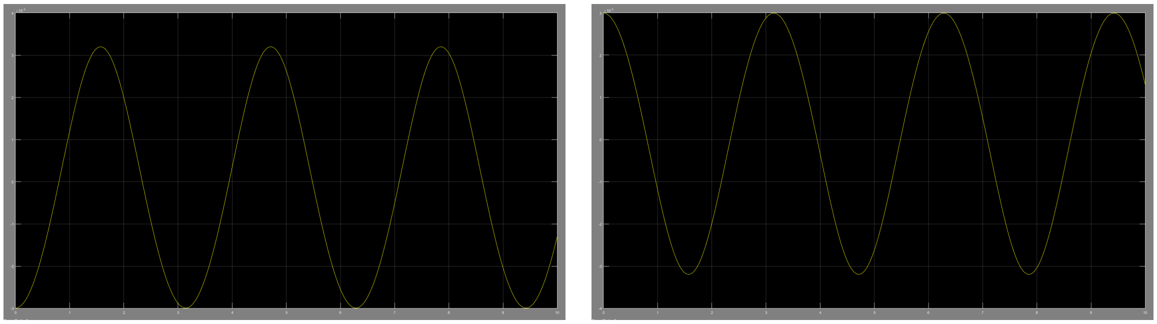
Kinematic control simulation

In accordance with the above derivation a Simulink kinematic model of the biglide was developed. For the kinematic control part two scenarios were developed using:

- velocities of prismatic joints $\dot{q_1}$ and $\dot{q_2}$,
- accelerations of prismatic joints $\ddot{q_1}$ and $\ddot{q_2}$.

Kinematic velocity control

Kinematic velocity control is used here to track desired trajectory of the end effector by velocity control of the prismatic joints. A simple straight line trajectory is defined for time interval 1 second without any



(a) Error in $\ddot{\phi}_1$.

(b) Error in $\ddot{\phi}_2$.

Figure 1.8: Error in $\ddot{\phi}_1$ and $\ddot{\phi}_2$ between Simulink passive angle velocities model and Adams plant for same input.

singularity crossing. Figure (1.9) shows the desired trajectory defined in Matlab.

As it can be observed in the plot (1.10) for error in trajectory, error converges to zero. K_p was tuned to value 10.

Next communication interval time with Adams simulation was varied. This is similar to varying the update rate of a velocity controller. The default communication interval time was set to 5 ms. Simulation was run again with two different time intervals: 0.5 ms and 50 ms. The plots for trajectory error are shown in figure (1.11).

With 50 ms communication interval, it is apparent that the update rate is very slow. Instead of a smooth curve, steps are recorded as the position feedback from Adams is received at a lower frequency. Also the error is unable to converge to zero. With 0.5 ms, the plot for error in trajectory is quite similar to the one recorded with default communication interval. However, the simulation carried out very slowly due to high communication rate. Thus the default value offers a good processing speed as well as allows favorable implementation of control.

Kinematic acceleration control

Kinematic acceleration control was used to track desired trajectory of the end effector by acceleration control of the active joints. Same trajectory was used for simulation as shown in previously in figure (1.9) . The controller was designed as follows.

Error, e is defined by the difference in current position and the desired position of the end effector.

$$\underline{\mathbf{e}} = \underline{\mathbf{x}} - \underline{\mathbf{x_t}} \tag{1.27}$$

For asymptotic stability, error should converge to zero. Following equation is defined (1.28).

$$\ddot{\underline{\mathbf{e}}} + \lambda_2 \dot{\underline{\mathbf{e}}} + \lambda_1 \underline{\mathbf{e}} = 0 \tag{1.28}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$.

Substituting equation (1.28) in (1.27), and rearranging it, expression in (1.29) is obtained.

$$\ddot{\underline{\mathbf{x}}} = \ddot{\underline{\mathbf{x}}}_t - \lambda_2(\dot{\underline{\mathbf{x}}} - \dot{\underline{\mathbf{x}}}_t) - \lambda_1(\underline{\mathbf{x}} - \underline{\mathbf{x_t}}) \tag{1.29}$$

It can be seen that λ_1 is the proportional gain K_p and λ_2 is the derivative gain K_d.

The desired control input $\ddot{\underline{\mathbf{q_a}}}$, can then simply be computed using second order kinematic equation derived in (1.22).

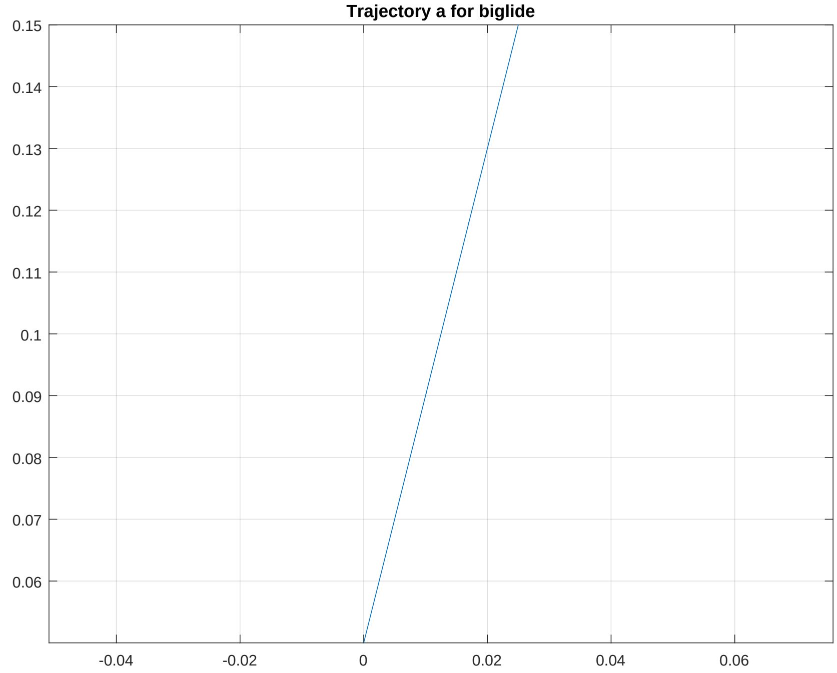


Figure 1.9: Desired trajectory used for kinematic control simulation scheme in Simulink.

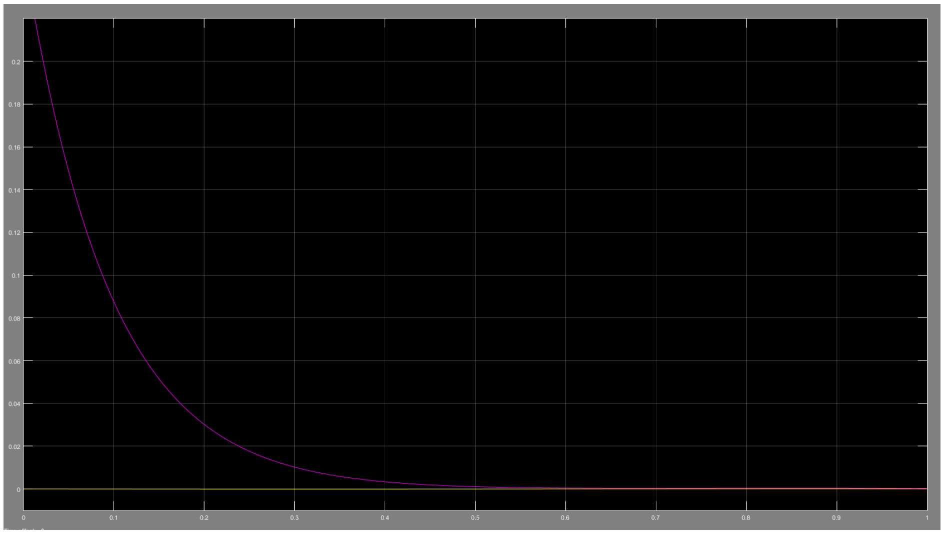
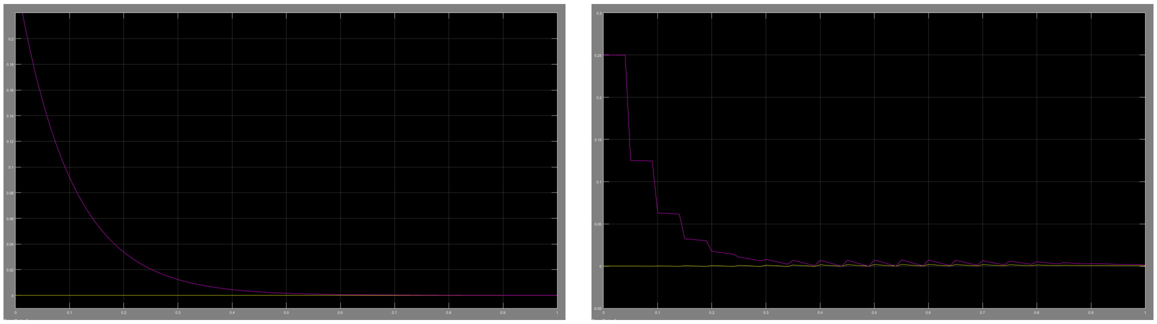


Figure 1.10: Error in trajectory using kinematic velocity control, $K_p = 10$.



(a) Communication interval time = 0.5ms .

(b) Communication interval time = 50ms .

Figure 1.11: Error in trajectory plotted for different communication interval time with Adams.

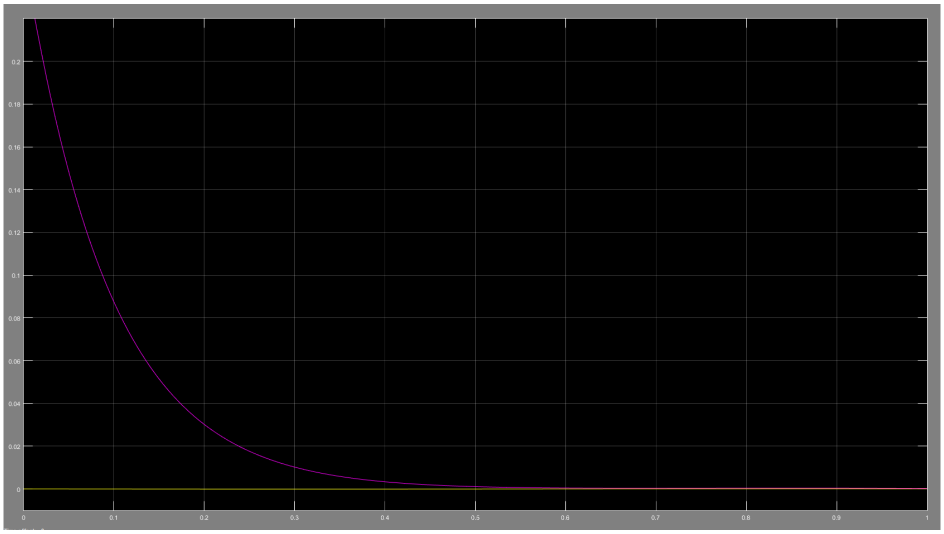


Figure 1.12: Error in trajectory using kinematic acceleration control, $K_p = 75$, $K_d = 15$.

Figure (1.12) shows the plot for error in trajectory converging to zero. Proportional gain was tuned to $K_p = 75$ and derivative gain was tuned to $K_d = 15$.

Chapter 2

Dynamic analysis

Dynamic model

The dynamic model is based upon analysis of the system with respect to kinematic E and potential energy U using the general Lagrangian equation (2.1) for body B_{ij}.

$$L_{ij} = E_{ij} - U_{ij} \tag{2.1}$$

Since the gravity is neglected, all potential energy for all bodies is $U = 0$. For the kinematic energy E the general approach using twist describing velocities of a body B_{ij} in translation and angular components can be expressed as (2.2).

$$E_{ij} = \frac{1}{2}\begin{bmatrix} v_{ij} & \omega_{ij} \end{bmatrix}\begin{bmatrix} m_{ij}\mathbf{1}_3 & 0 \\ 0 & I_{ij} \end{bmatrix}\begin{bmatrix} v_{ij} \\ \omega_{ij} \end{bmatrix} = \frac{1}{2}m_{ij}v_{ij}^T v_{ij} + \frac{1}{2}I_{ij}\omega_{ij}^T\omega_{ij} \tag{2.2}$$

First the tree structure of the parallel robot is divided into limbs with a free floating mass for the end effector according to figure (2.1).

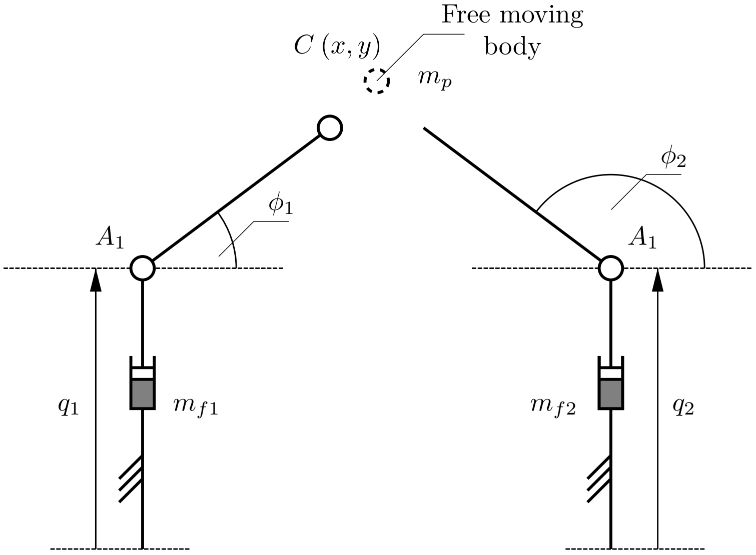


Figure 2.1: Kinematic model of a biglide parallel robot with configuration parametrization.

Since only the mass of each foot m_{f1}, m_{f2} and the end effector m_p are considered, where the rest of parameters and processes like gravity and friction are neglected, the dynamic model can be expressed as Lagrangian (2.3) consisting as a sum of left and right limbs.

$$L = f(\dot{q_1}, \dot{q_2}) = L_{11} + L_{12} \tag{2.3}$$

$$L_{11} = E_{11} = \frac{1}{2} m_{f1} \dot{q_1}^2 \tag{2.4}$$

$$L_{21} = E_{21} = \frac{1}{2} m_{f2} \dot{q_2}^2 \tag{2.5}$$

$$\tau = \begin{bmatrix} \tau_a \\ \tau_d \end{bmatrix} = \begin{bmatrix} \tau_{11} \\ \tau_{12} \\ \tau_{21} \\ \tau_{22} \end{bmatrix} = \begin{bmatrix} \tau_{11} \\ \tau_{21} \\ 0 \\ 0 \end{bmatrix} \tag{2.6}$$

$$\tau_a = \begin{bmatrix} \tau_{11} \\ \tau_{12} \end{bmatrix}, \ \tau_d = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.7}$$

$$\tau_{11} = \frac{d}{dt}\left(\frac{\delta L}{\delta \dot{q_1}}\right)^T - \left(\frac{\delta L}{\delta q_1}\right)^T = \frac{d}{dt}\left(m_{f1}\dot{q_1}\right) - 0 = m_{f1}\ddot{q_1} \tag{2.8}$$

$$\tau_{12} = \frac{d}{dt}\left(\frac{\delta L}{\delta \dot{q_2}}\right)^T - \left(\frac{\delta L}{\delta q_2}\right)^T = \frac{d}{dt}\left(m_{f2}\dot{q_2}\right) - 0 = m_{f2}\ddot{q_2} \tag{2.9}$$

$$\underline{\mathbf{x_p}} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.10}$$

$$\underline{\dot{\mathbf{x_p}}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{2.11}$$

$$L_p = \frac{1}{2} m_p \underline{\dot{\mathbf{x_p}}}^T \underline{\dot{\mathbf{x_p}}} \tag{2.12}$$

$$w_r = \frac{d}{dt}\left(\frac{\delta L_p}{\delta \dot{\underline{\mathbf{x}}}}\right)^T - \left(\frac{\delta L_p}{\delta \underline{\mathbf{x}}}\right)^T = \frac{1}{2} m_p \frac{d}{dt}\begin{pmatrix} \dot{x}^2 \\ \dot{y}^2 \end{pmatrix} - 0 = m_p \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} \tag{2.13}$$

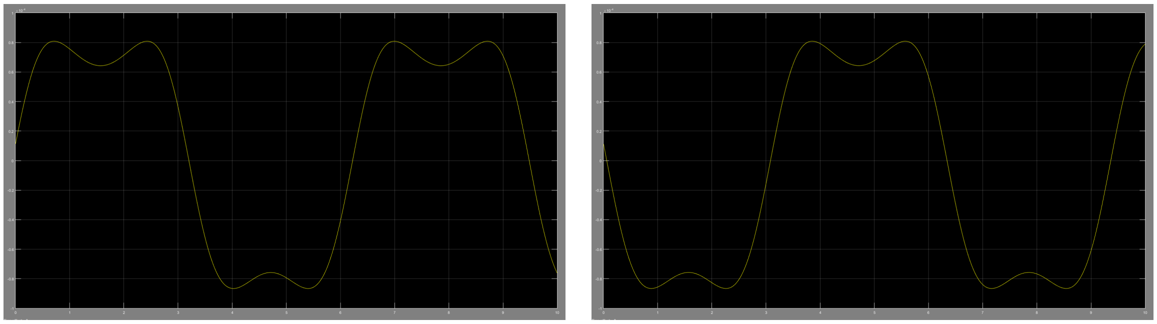
Finally, the inverse dynamic model can be expressed as shown in equation (2.14).

$$\tau = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} m_f \ddot{q_1} \\ m_f \ddot{q_2} \end{bmatrix} - (A^{-1}B)^T \begin{bmatrix} m_p \ddot{x} \\ m_p \ddot{y} \end{bmatrix} \tag{2.14}$$

The inverse dynamic model was then simulated in Simulink and compared against the outputs obtained from Adams. Errors of order 10^{-4} confirmed that the model was correctly defined. The plots are shown in the figure (2.2).

Control co-simulation

Computed torque control was used for the dynamic control simulation. For this control, feedback linearization was implemented by using inverse dynamic model defined in the previous section. The dynamic model of a parallel robot can be represented in the generic form as shown in (2.15) where M is the inertia matrix and c is the vector of Coriolis and centrifugal effects.



(a) Error in f_1.

(b) Error in f_2.

Figure 2.2: Error in f_1 and f_2 between Simulink inverse dynamic model and Adams plant for same input.

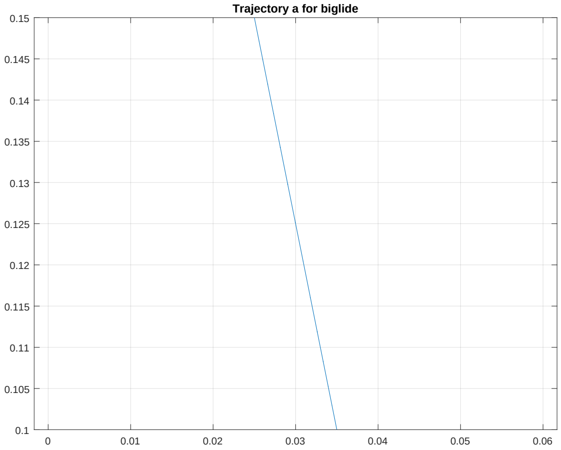


Figure 2.3: Desired trajectory used for Dynamic control simulation without crossing type 2 singularity.

$$\tau = M\ddot{q_a} + c \tag{2.15}$$

Matrix M and vector c can be determined by using inverse dynamic model equation defined in (2.14) and second order kinematic model (1.23).

For simulation, a short straight line trajectory was defined of time interval $1s$. Figure (1.9) shows the desired trajectory defined in Matlab. This trajectory did not cross singularity.

Figure (2.4) shows the plot for error in trajectory for the desired trajectory. It can be observed that the error converges to zero in $1s$. The proportional and derivative gain were tuned to $K_p = 100$ and $K_d = 20$.

Type 2 singularity

Type 2 singularity (a parallel singularity) of a parallel manipulator occurs when the matrix A, defined in direct kinematic model (1.13) is not invertible. That happens when A is rank deficient and $det(A) = 0$.

Using the relations from constraint equations (1.1) and (1.2), matrix A can also be represented as shown in equation (2.16).

$$A = \begin{bmatrix} l_{A_1C}cos(\phi_1) & l_{A_1C}sin(\phi_1) \\ l_{A_2C}cos(\phi_2) & l_{A_2C}sin(\phi_2) \end{bmatrix} \tag{2.16}$$

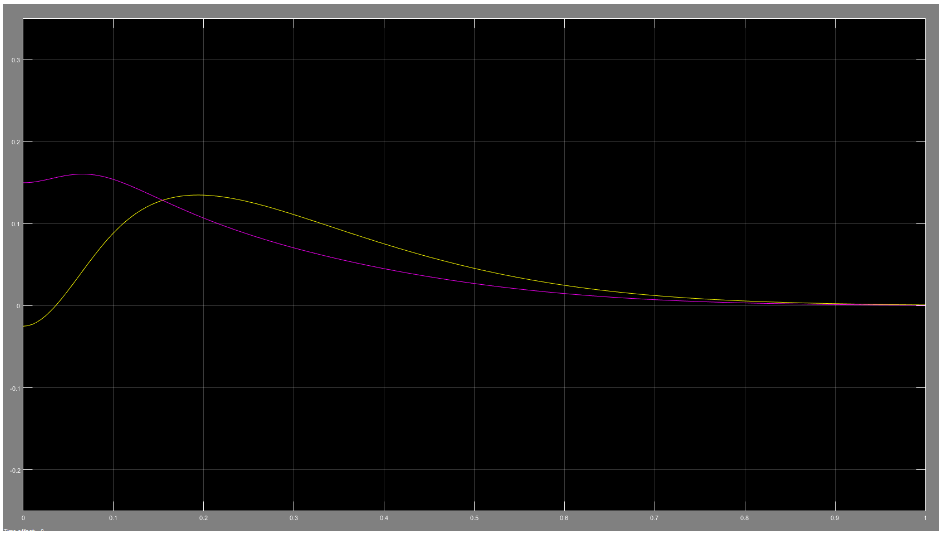


Figure 2.4: Error in trajectory using computed torque control, no singularity defined in trajectory, $K_p = 100$, $K_d = 20$.

For the condition when determinant of matrix A, the following relation (2.17) is achieved using trigonometric identities.

$$det(A) = l_{A_1C} l_{A_2C} sin(\phi_2 - \phi_1) \tag{2.17}$$

Thus, for the type 2 singularity, the following condition can be concluded where $k = 0, \pm 1, ...$

$$\phi_2 = \phi_1 + k\pi \tag{2.18}$$

The figure 2.5 shows one possible configuration of the system under type 2 singularity, which was tested in simulation.

Next, trajectory for simulation crossing was defined. Since the length of links $A1C$ and $A2C$ are same, $l_{A_1C} = l_{A_2C}$, and the absolute coordinate frame is defined in the center of biglide feet, the singularity condition occurs when $x = 0$ and $y = (q_1 + q_2)/2$ for the end effector. Equation (2.19) shows the other relation when the robot is in type 2 singularity.

$$(q_1 - q_2)^2 + d^2 = (l_{A_1C} + l_{A_2C})^2 \tag{2.19}$$

Therefore in order to simulate trajectory for the crossing of type 2 singularity, a straight line was taken that included point when $x = 0$. Also, in order for the manipulator to be at singularity, the working mode must be appropriately defined. In computation of inverse geometric model, γ_1 and γ_2 must be selected such that one is set to value -1 and the other $+1$.

In the first simulation with singularity crossing, the time interval for the defined trajectory was taken to be 1 second. Figure (2.6) shows the desired trajectory defined in Matlab. This trajectory is further referred to as $b1$.

The simulation of the control generated interesting results. Figure (2.7) shows plot for error in x and active joints position $\underline{\mathbf{q_a}}$ for the desired trajectory $b1$. It can be observed that the computed torque controller was unable to fully converge until in the very end. Thus the end effector never reached the position $x = 0$. Even though $\underline{\mathbf{q_a}}$ converged to zero in the end of simulation (2.7b) but the trajectory error never fully converges to zero (2.8) and there is a significant error in trajectory tracking. This is because the Adams model remained in the same assembly mode and did not cross singularity. Instead it retreated close to singularity condition.

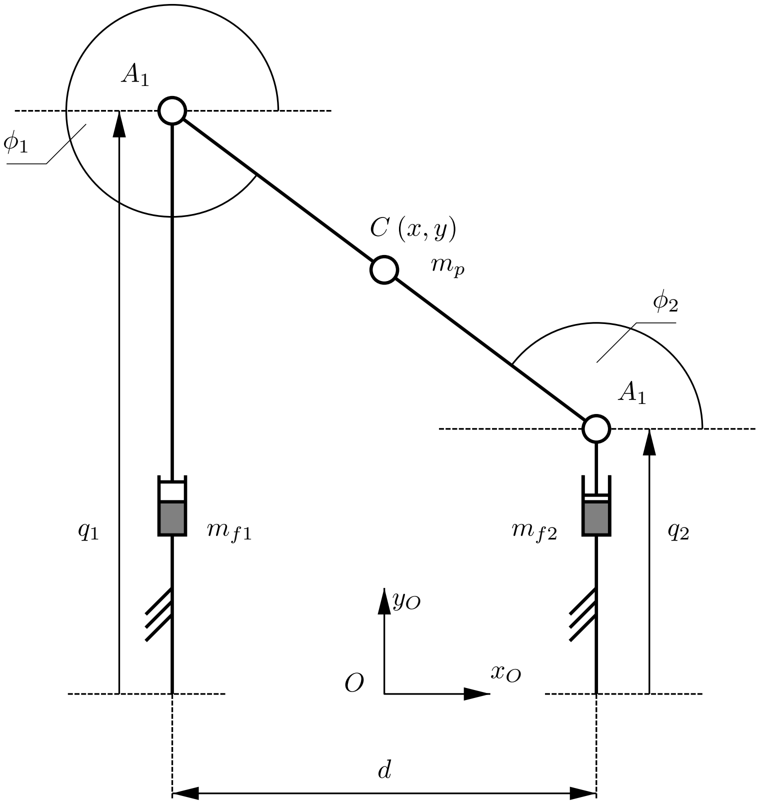


Figure 2.5: Kinematic model of a biglide parallel robot in a singularity type 2 configuration.

This is because for the manipulator to cross singularity, the assembly mode changes. But the assembly mode in the Adams remained the same.

In the second simulation with singularity crossing, the same path was selected as in (2.6) but the time interval for the defined trajectory was taken to be $3s$. This trajectory is further referred to as $b2$. This was done to ensure that the controller converged before reaching singularity condition.

Figures (2.9) and (2.10) show the plots for the error in x position of end effector, active joints position $\underline{\mathbf{q_a}}$ and error in trajectory for the desired trajectory $b2$. Now, the controller is able to fully converge before reaching singularity point. Thus, at singularity crossing, matrix A becomes singular, the input efforts very large and the computed torque controller is unable to compute control torque. Therefore the simulink simulation stops with error. It can be seen that the plots end before $t = 3s$.

Trajectory tracking

Straight trajectory is defined that crosses type 2 singularity for simulation in Adams alone as shown in figure (2.11) defined in Matlab.

No constraints were defined for singularity crossing at first. During simulation in Adams, it was observed that the links A1C and A2C vibrate and shake a lot when crossing singularity. This is because the robot input efforts were very large at singularity. The results are shown in figure (2.12). In practice, this singularity can break down the system or damage parts.

As portrayed in figure (2.12), computed input efforts at singularity become very large. However, it is still possible to cross type 2 singularity without degeneracy of the input efforts. This is done by optimal trajectory planning. When matrix A becomes rank deficient, a non-null vector t_s can be computed that belongs to the

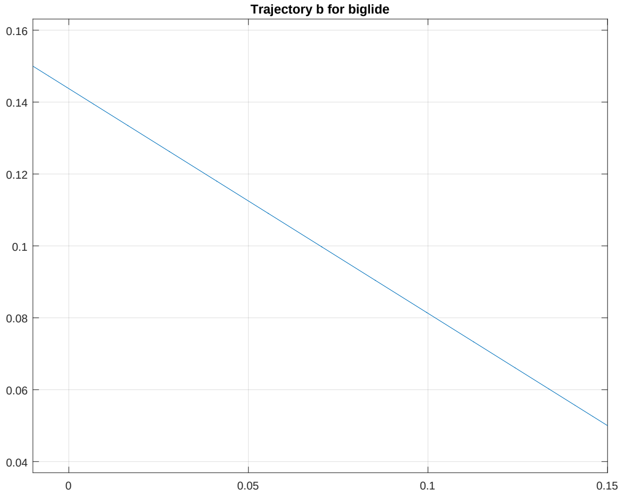
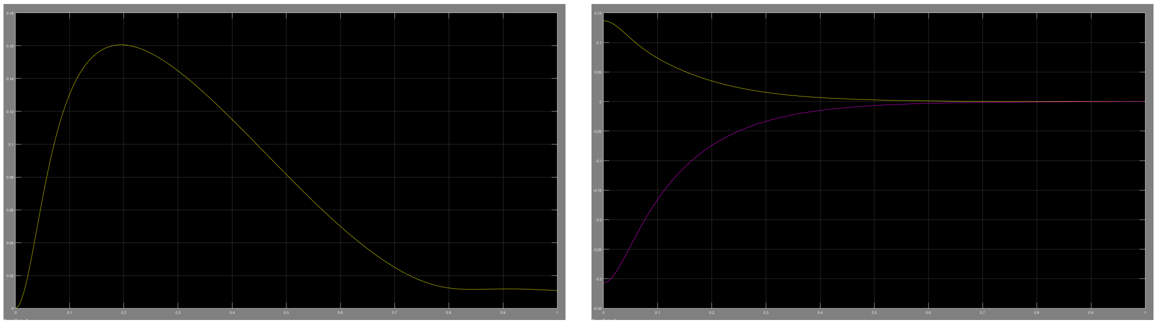


Figure 2.6: Desired trajectory $b1$ used for dynamic control simulation for crossing type 2 singularity, time interval $1s$.



(a) Error in x position only plotted for desired trajectory $b1$.

(b) Error in active joints position plotted for desired trajectory $b1$.

Figure 2.7: Error in trajectory and active joints position plotted for desired trajectory $b1$.

kernel of matrix A as shown in equation (2.20).

$$At_s = 0 \tag{2.20}$$

Using the singularity condition defined in (2.18), following t_s can be selected as shown in (2.21) where $\Psi = \phi_1$ or ϕ_2.

$$t_s = \begin{bmatrix} -sin(\Psi) \\ cos(\Psi) \end{bmatrix} \tag{2.21}$$

The criterion to cross the singularity is given by equation (2.22). It can be further simplified to be expressed as (2.23) This criterion ensures that the sum of wrenches applied on the platform is reciprocal to the uncontrollable motion of platform inside the singularity.

$$\begin{bmatrix} -sin(\Psi) & cos(\Psi) \end{bmatrix} \begin{bmatrix} m_p\ddot{x} \\ m_p\ddot{y} \end{bmatrix} = 0 \tag{2.22}$$

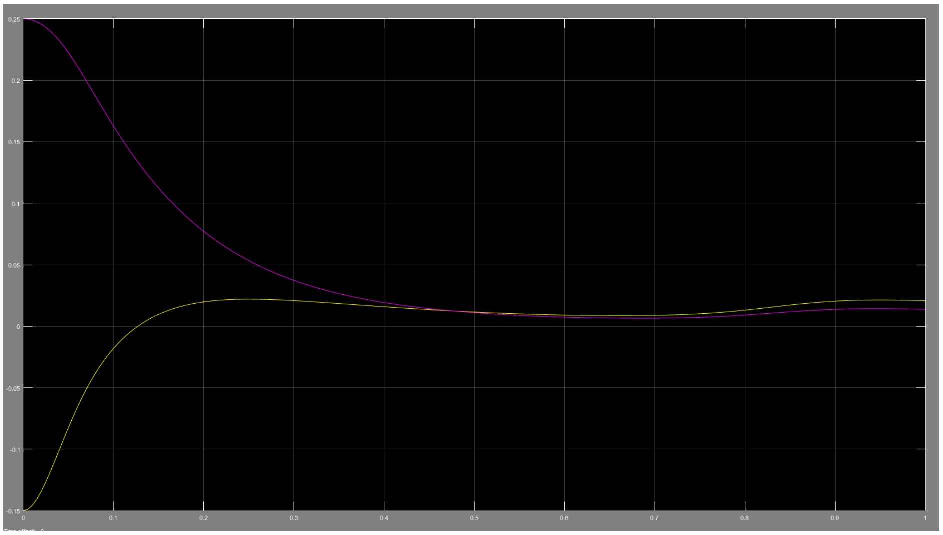
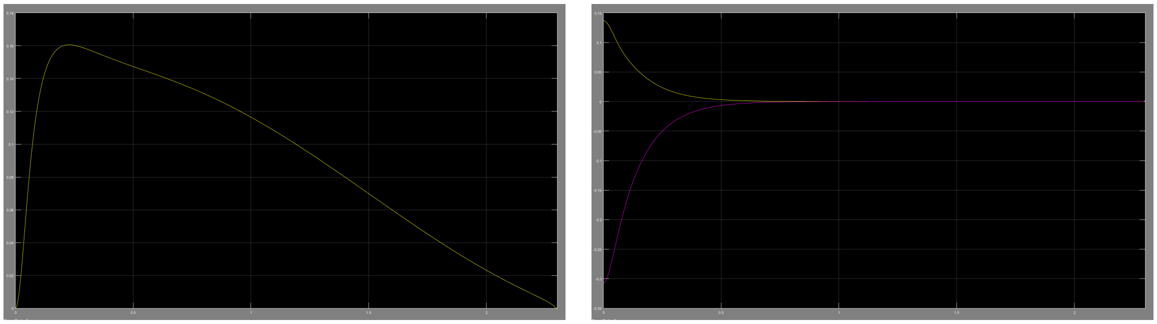


Figure 2.8: Error in trajectory using computed torque control for desired trajectory $b1$, $K_p = 265$, $K_d = 40$.



(a) Error in x position only plotted for desired trajectory $b2$.

(b) Error in active joints position plotted for desired trajectory $b2$.

Figure 2.9: Error in trajectory and active joints position plotted for desired trajectory $b2$.

$$\ddot{y} = tan(\Psi)\ddot{x} \tag{2.23}$$

Thus, a simple solution to cross type 2 singularity in biglide is by ensuring that the acceleration in both x and y direction are zero at singularity. Using this constraint, the trajectory was modified and redefined using 7th order polynomial. Singularity was defined a point $(0, 0.1)$ when $t = 0.5s$. The initial and final positions were kept same but acceleration at singularity crossing was kept zero, as shown in figure (2.13).

This trajectory was then simulated in Adams. It was observed that the input efforts again became very large near type 2 singularity irregardless of the defined constraint. This was because acceleration was defined zero only at singularity instant. When the system is approaching singularity and is very close to it, the forces become very large, approaching infinite value.

To improve the results, another trajectory was defined of 8th order polynomial. The initial, final positions and singularity point were kept same. The acceleration as well as the jerk at point of singularity was set to zero, as shown in figure (2.14). This was done to ensure that the acceleration was zero in a small region around singularity.

The simulation was repeated in Adams. This did not fix the problem. Figure (2.15) shows the plots for force recorded on biglide feet. However, it can be observed that the input effort at instant $0.5s$ at the point of singularity is zero. But as soon as the acceleration near the singularity point becomes different from zero,

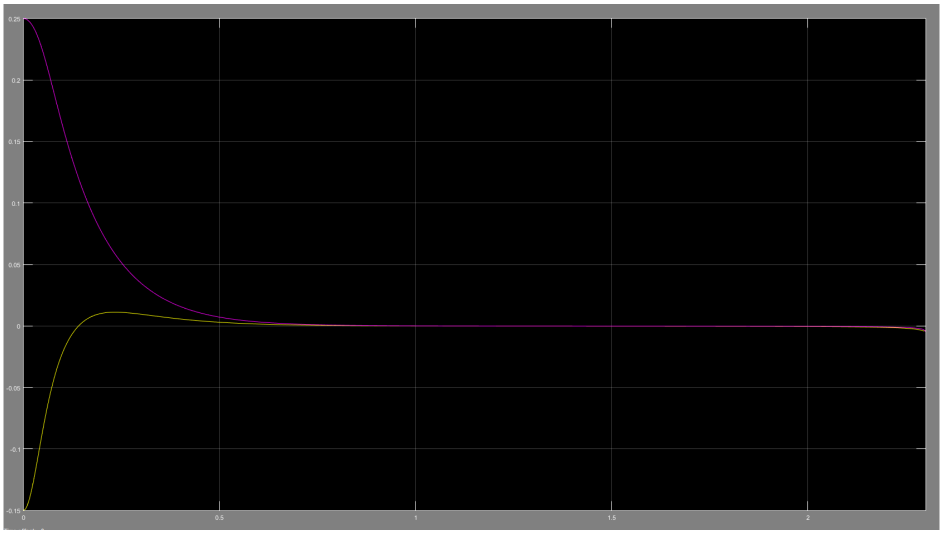


Figure 2.10: Error in trajectory using computed torque control for desired trajectory $b2$, $K_p = 265$, $K_d = 40$.

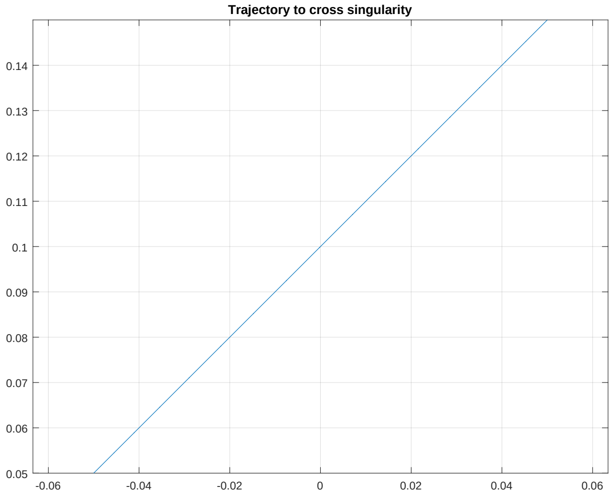


Figure 2.11: Trajectory definition for simualtion in Adams alone.

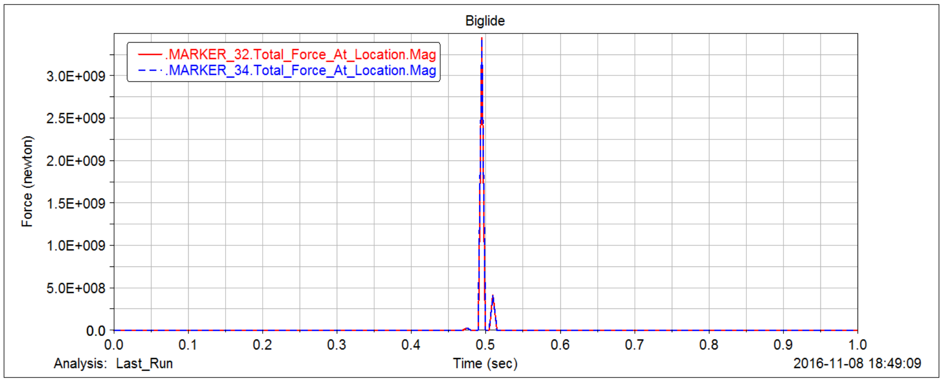
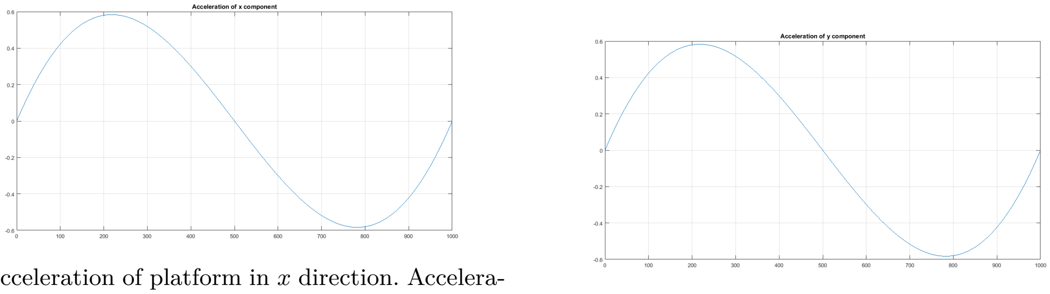


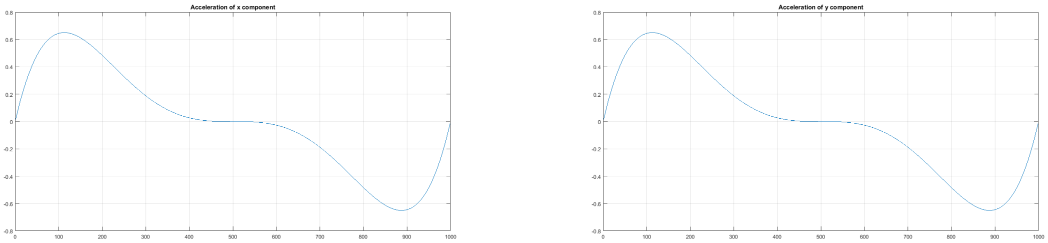
Figure 2.12: Total force on biglide feet going semi-infinite at singularity crossing.



(a) Acceleration of platform in x direction. Acceleration is defined 0 for singularity crossing at $t = 500ms$.

(b) Acceleration of platform in y direction. Acceleration is defined 0 for singularity crossing at $t = 500ms$.

Figure 2.13: Acceleration of platform in x and y direction, as defined by the trajectory.



(a) Acceleration of platform in x direction. Jerk is defined 0 for singularity crossing at $t = 500ms$.

(b) Acceleration of platform in y direction.Jerk is defined 0 for singularity crossing at $t = 500ms$.

Figure 2.14: Acceleration of platform in x and y direction, as defined by the trajectory.

the torque becomes very large again. This means that the acceleration around the singularity point needs to be kept zero for wider interval.

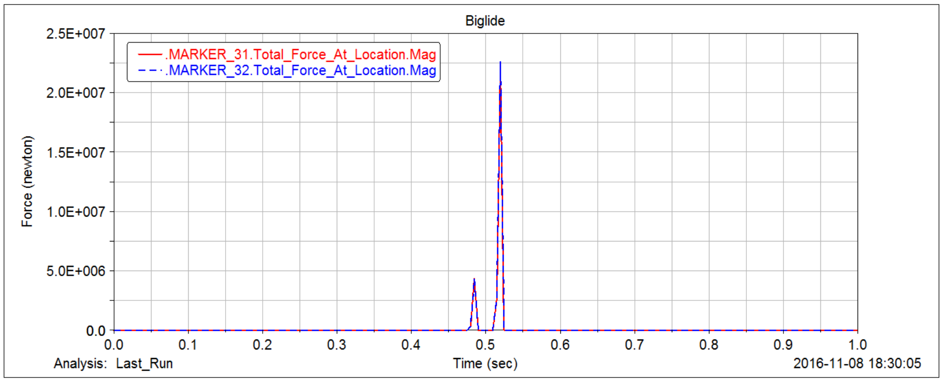


Figure 2.15: Total force on biglide feet going semi-infinite near singularity crossing but not at singularity. This is because jerk was defined 0 for type 2 singularity.

Conclusions

The geometric, kinematic and dynamic models of biglide mechanism were first mathematically derived and then simulated and compared in Matlab and in Adams. Once the models were validated, kinematic velocity and acceleration control as well as dynamic control was implemented to follow desired trajectories. Condition for type 2 singularity was studied in co-simulation with Adams and Simulink. The criterion to cross singularity type 2 was examined in simulations using Adams.

The results from the simulation confirm the validity of the models and allow for use this approach in various areas of robotics.