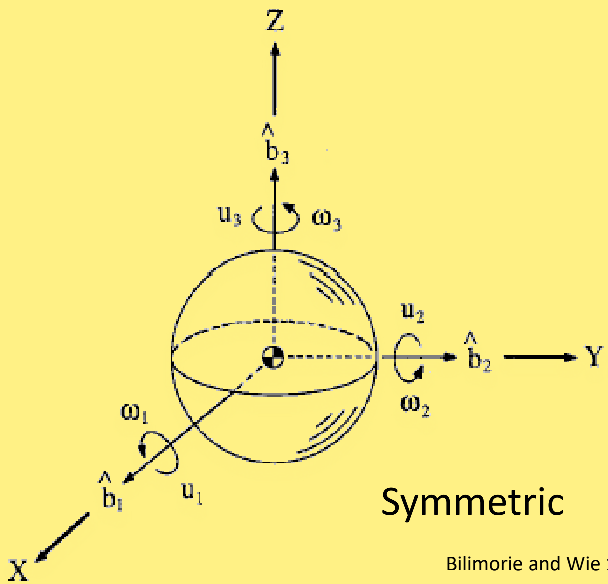
Time-Optimal Reorientation using Neural Network and Particle Swarm Formulation

AIAA/AAS Astrodynamics Specialist Conference
August 20-24, 2017, Stevenson Washington

Ko Basu, Robert Melton and Sarah Aguasvivas Manzano

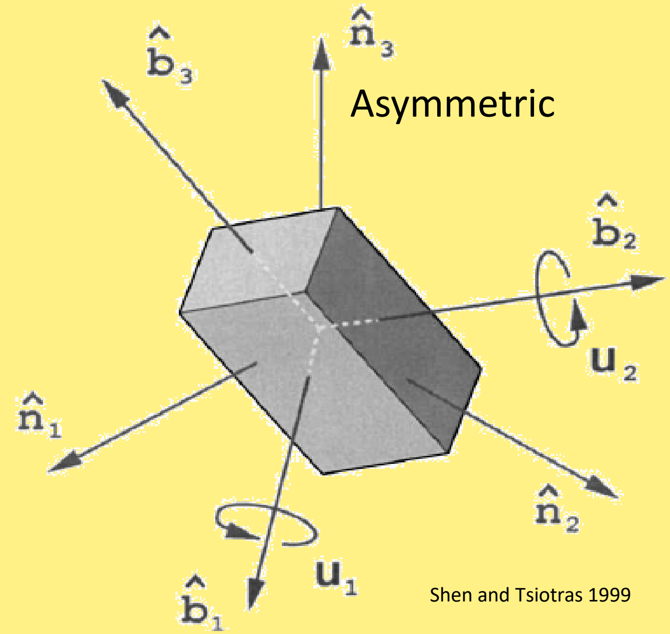
PennState
College of Engineering

1954 2017
American Astronautical Society



Symmetric

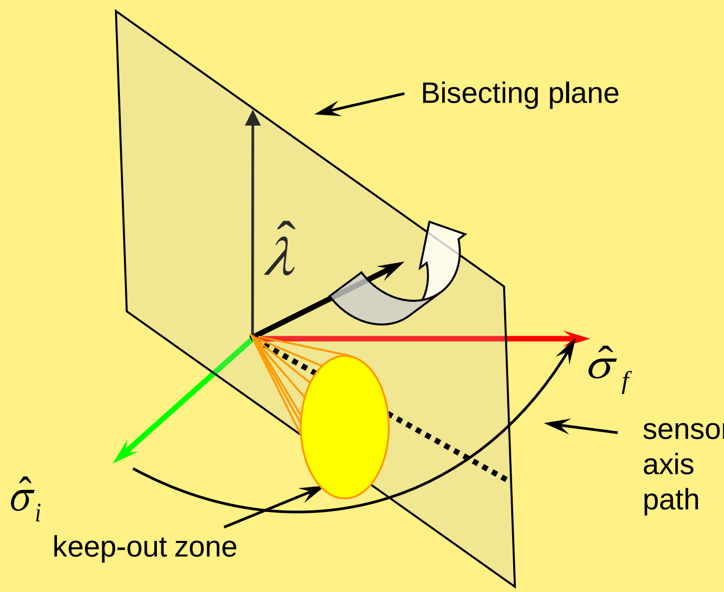
Bilimorie and Wie 1993



Asymmetric

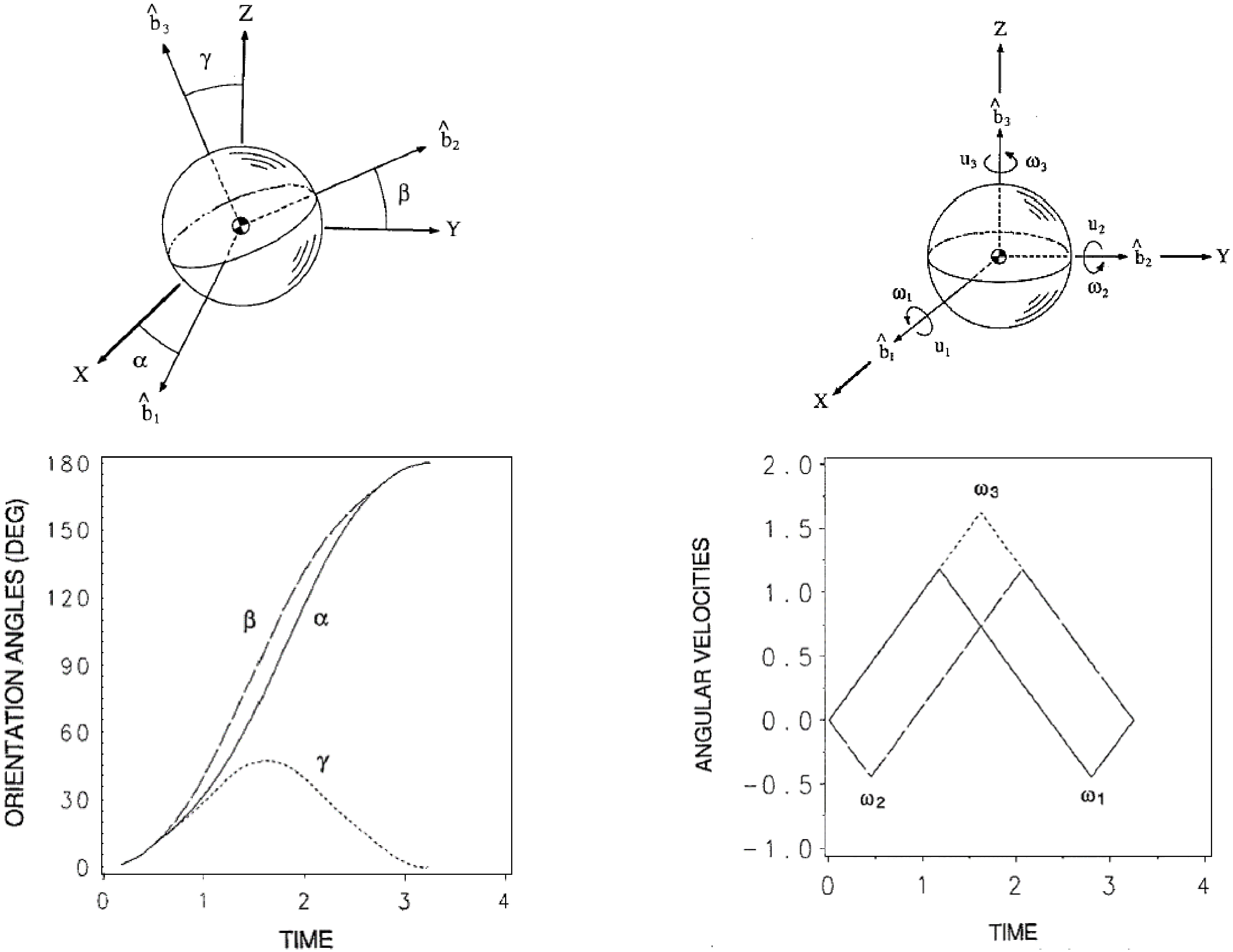
Shen and Tsiotras 1999

Constrained



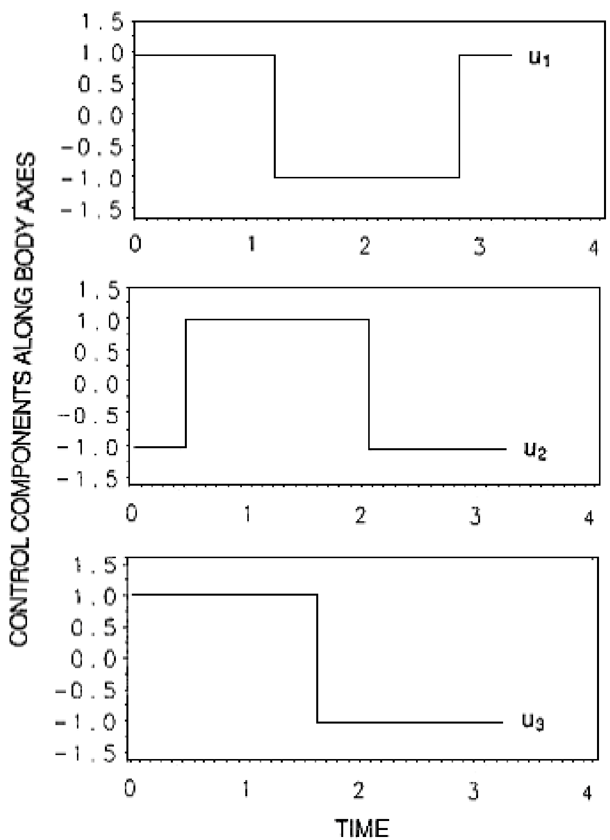
Melton 2009

Constrained Time-Optimal Slewing Maneuvers for Rigid Spacecraft

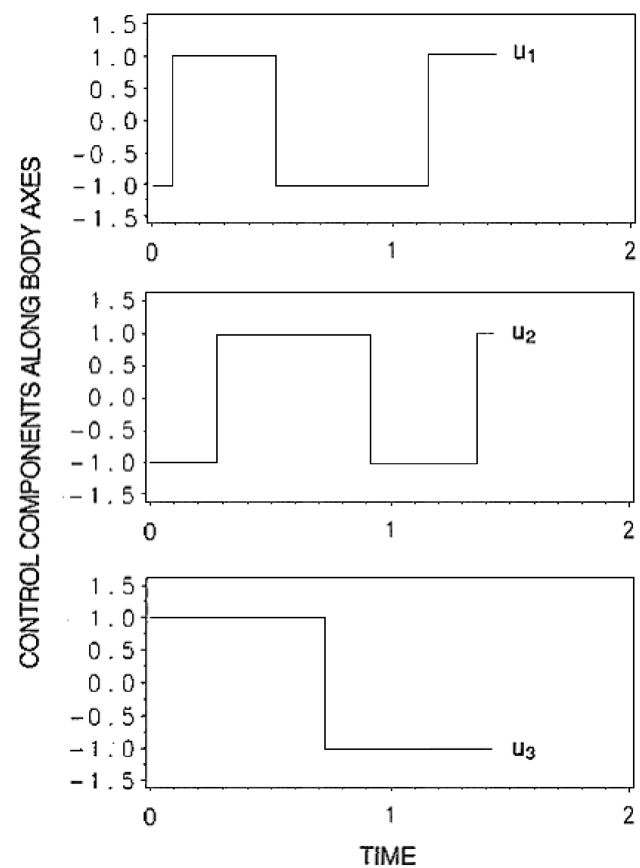


Bilimorie and Wie 1993

Time-Optimal Three-Axis Reorientation of a Rigid Spacecraft



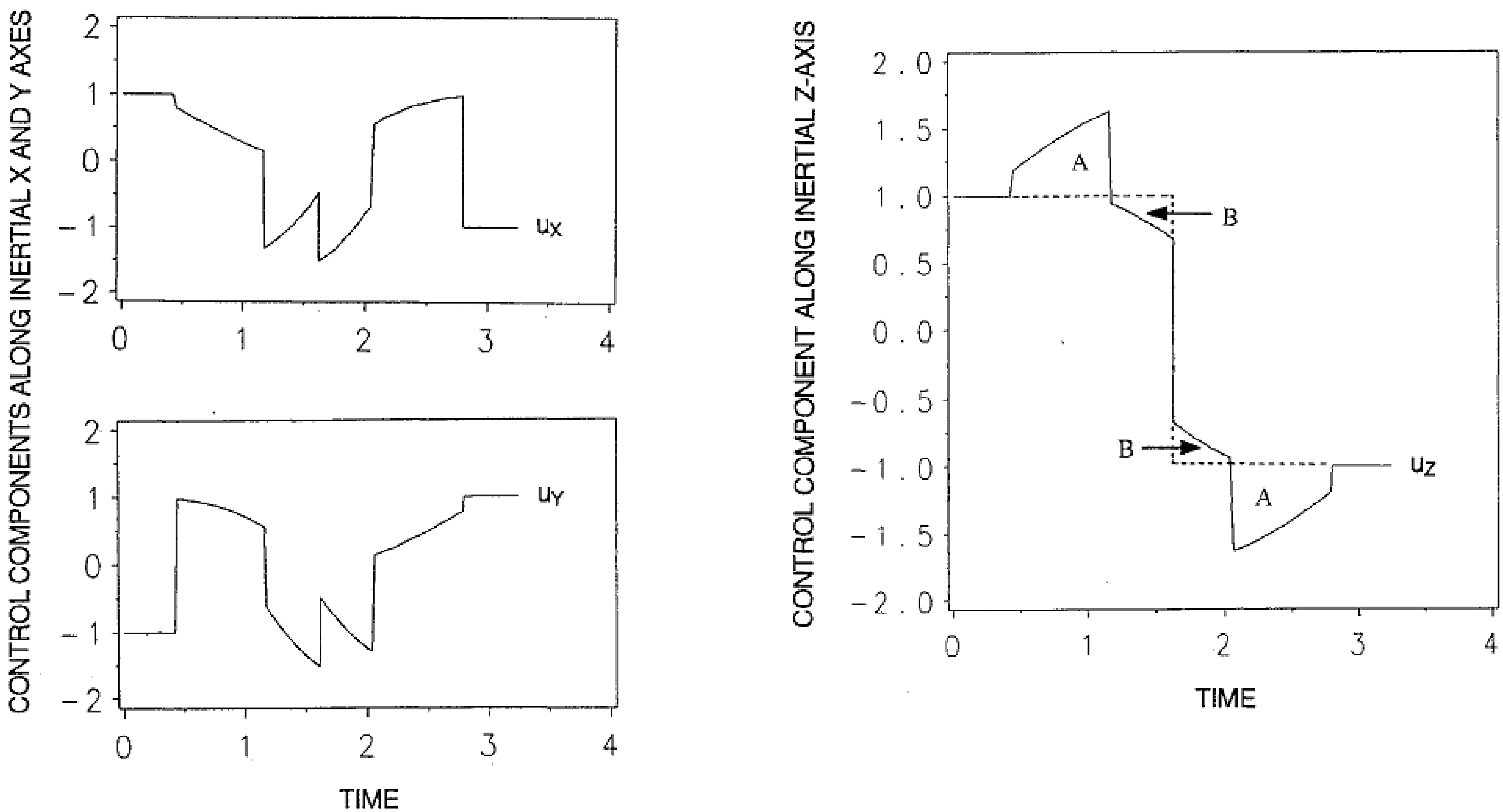
Control Input for ø = 180˚ (5 switches)



Control Input for ø = 30˚(7 switches)

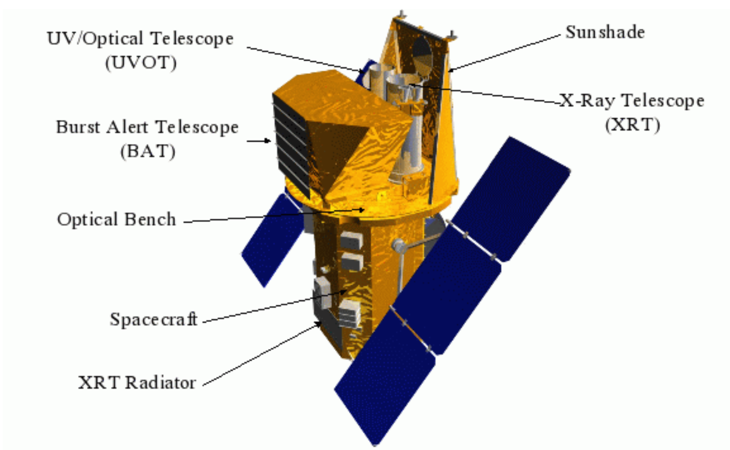
Bilimorie and Wie 1993

Time-Optimal Three-Axis Reorientation of a Rigid Spacecraft



Bilimorie and Wie 1993

Swift Telescope and Keep Out Cones



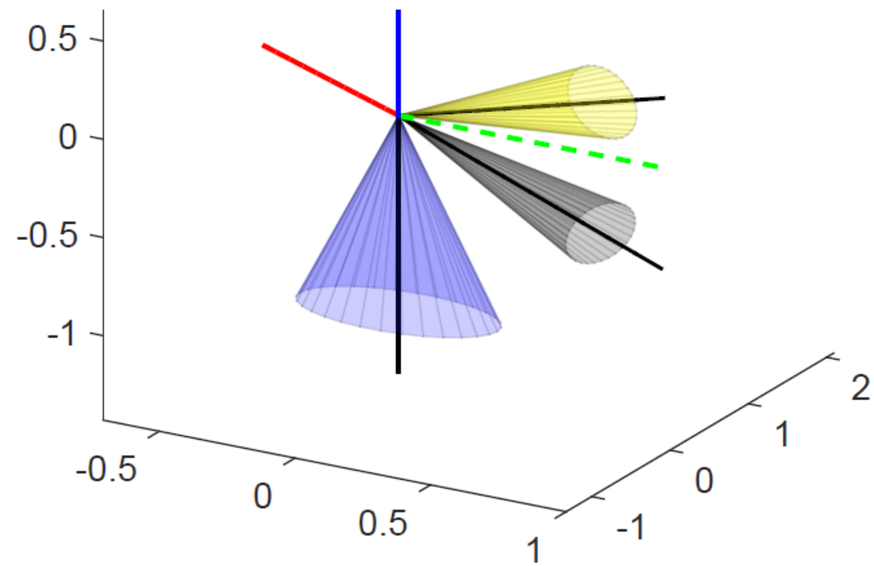
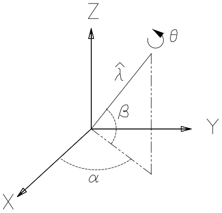


Figure 1.2. Path Constraint Cones

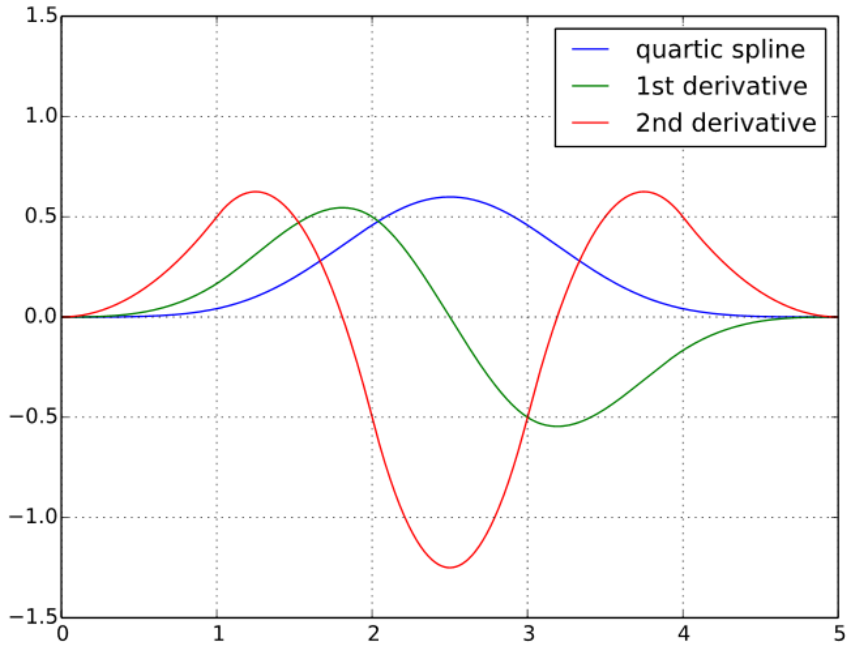
Variables of the Particle Swarm



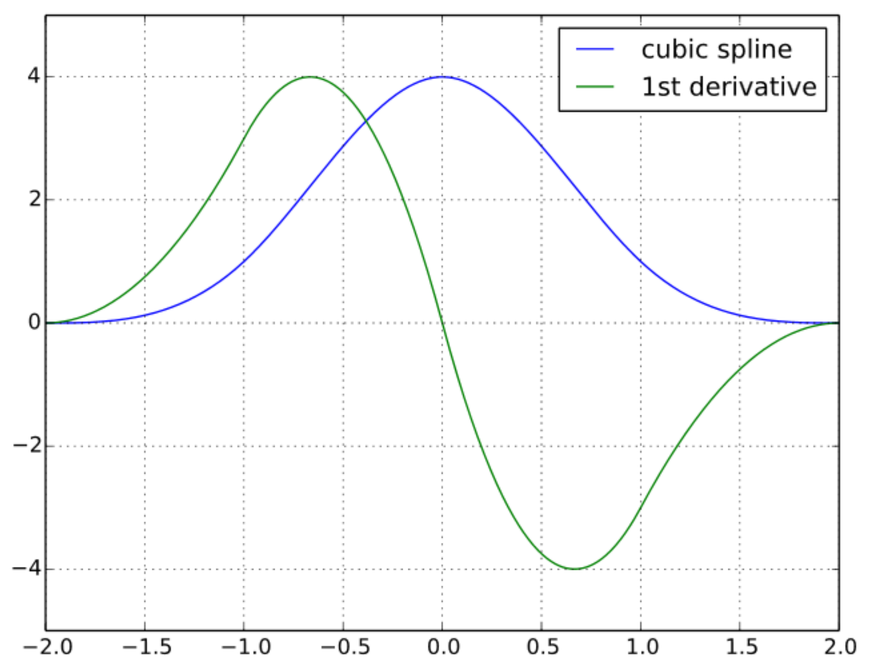
$$
\begin{aligned}
\lambda_1 &= \cos(\beta)\cos(\alpha) \\
\lambda_2 &= \cos(\beta)\sin(\alpha) \\
\lambda_3 &= \sin(\beta)
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
\dot{\lambda}_1 &= -\dot{\beta}\sin(\beta) cos(\alpha) - \dot{\alpha}\cos(\beta)\sin(\alpha) \\
\dot{\lambda}_2 &= -\dot{\beta}\sin(\beta) sin(\alpha) + \dot{\alpha}\cos(\alpha)\cos(\beta) \\
\dot{\lambda}_3 &= \dot{\beta}\cos(\beta)
\end{aligned}
\tag{6}
$$

B-Spline Interpolation



Quartic B-spline with knot vector (0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5, 5) and control points (0, 0, 0, 0, 1, 0, 0, 0, 0), and its first and second derivatives



Cubic B-spline with knot vector (−2, −2, −2, −2, −1, 0, 1, 2, 2, 2, 2) and control points (0, 0, 0, 6, 0, 0, 0), and its first derivative

0.965006 -1.30342 0.30792 -0.0740249 -0.145152 -0.140172 -0.630729 0.669519 0.426373 0.00263573 0.00635012 -0.0260977 0.902781 1.5708 0.99433 -1.22665 -0.870726 -0.308493 -0.127239 0.0477669 0.144741 3.49653

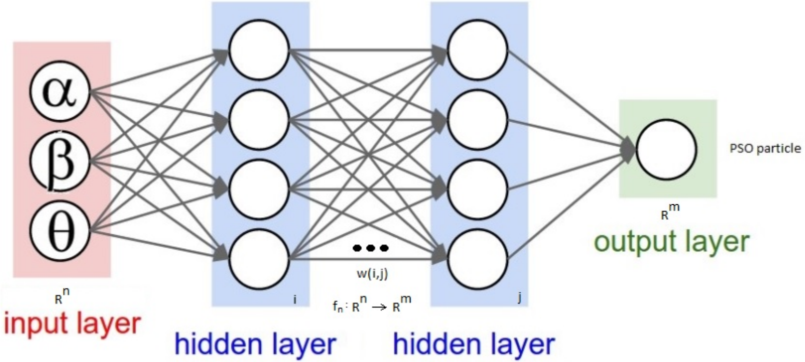
0.951935 -1.18388 1.21771 0.0752473 0.133309 0.00888675 -0.314355 0.16688 0.033942 -0.0113174 0.201051 0.465341 0.839104 1.12897 0.984569 -1.10822 -0.772198 -0.23889 0.168443 0.650257 1.10836 3.36989

1.61873 0.716839 1.40345 -0.127834 -0.162759 0.289203 -0.123212 -0.203722 -0.447669 0.0651735 0.406652 0.337175 0.652673 1.11825 1.49734 0.611742 0.618711 0.44124 0.596403 0.86567 1.27369 4.08297

2.88387 1.20304 0.0408532 0.00526357 0.0385806 0.217128 0.291822 0.182548 0.343912 0.0106673 0.0998913 0.136166 0.143576 -0.0940833 1.45298 1.15623 0.94852 0.597429 0.250286 0.0588486 0.0324745 2.67959

2.30851 -1.05954 0.479355 -0.195105 0.0487739 0.395276 -0.476468 0.489897 0.0590391 -0.0310422 0.0939462 -0.321852 0.478763 1.56167 1.57079 -0.905581 -0.3118 0.19842 0.16382 0.548128 0.614144 4.82329

Neural Network



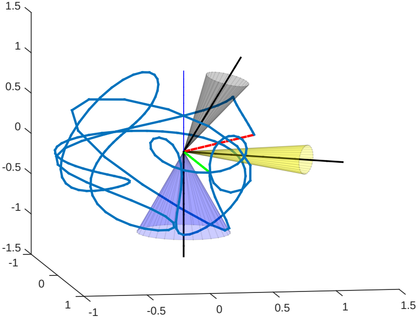
Neural Network

Hyper-Parameters

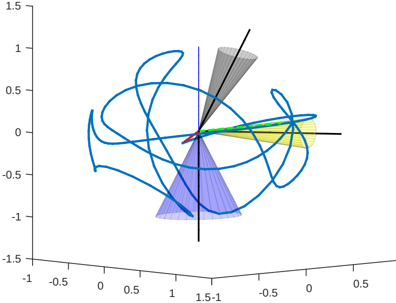
Neural Network Parameters		
Parameter	**Value in Time Problem**	**Value in Path Problem**
η	1×10^{-3}	1×10^{-6}
a	5.0	3.15
Tol	5×10^{-3}	5×10^{-3}
Architecture	3-10-10-1	3-100-100-100-18

Table: Problem-specific selection of hyper-parameters for neural network.

Initial PSO Solution vs NN Solution

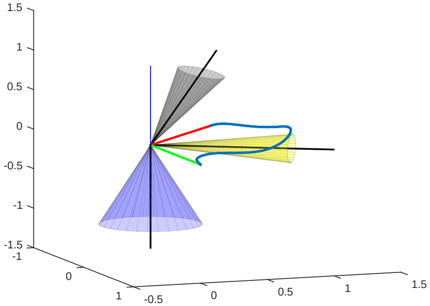


Initial PSO Solution

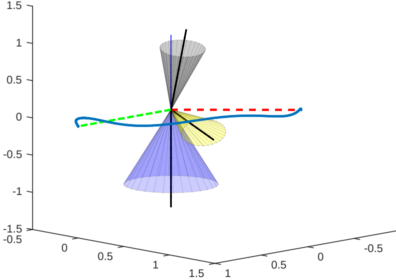


NN Solution

PSO Solution

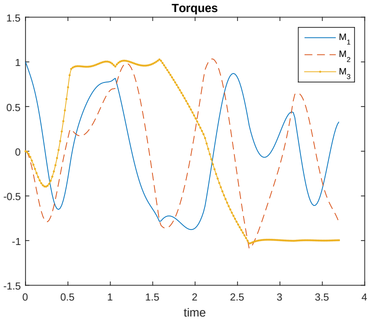


Side View

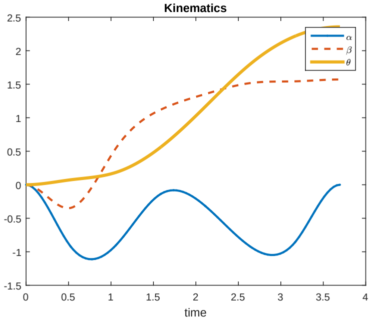


Front View

PSO Solution

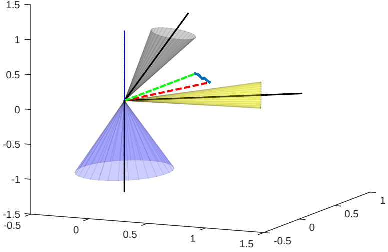


DU

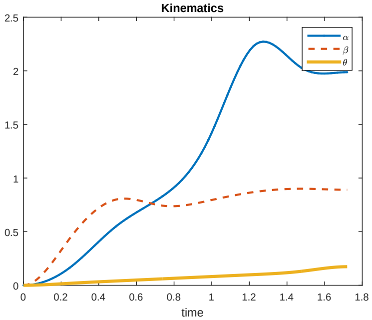


DU

Reduced Solution Space NN Solution



Path Maneuver



Kinematic Solution

Conclusion

- Neural Network Predictions have the possibility of providing good initial guess to PSO
- Initial guess may improve the quality of the solution of the PSO
- Good Initial guess may reduce the number of iterations required
- Neural Network utilizes previous maneuvers
- Future goal would be the Neural Network to provide a standalone feasible solution

Thank you

Constraints

Path Constraint

$$\hat{\sigma} \cdot \hat{\sigma}_x \leq \cos(\alpha_x) \tag{1}$$

Dynamic Constraints

$$\begin{aligned} M_1 &= I_1\dot{\omega}_1 + \omega_2\omega_3\left(I_3 - I_2\right) \\ M_2 &= I_2\dot{\omega}_2 + \omega_3\omega_1\left(I_1 - I_3\right) \\ M_3 &= I_3\dot{\omega}_3 + \omega_1\omega_2\left(I_2 - I_1\right) \end{aligned} \tag{2}$$

Boundary Constraints

$$\begin{gathered} \omega_1(0) = \omega_2(0) = \omega_3(0) = 0 \\ \epsilon_1(0) = \epsilon_2(0) = \epsilon_3(0) = 0, \quad \epsilon_4(0) = 1 \\ \omega_1(t_f) = \omega_2(t_f) = \omega_3(t_f) = 0 \\ \epsilon_1(t_f) = \epsilon_2(t_f) = 0, \quad \epsilon_3(t_f) = \epsilon_{3f} \quad \epsilon_4(t_f) = \epsilon_{4f} \end{gathered} \tag{3}$$

Torque Constraints

$$M_i \leq M_{max} \tag{4}$$

Overview of Inverse Dynamics

- Traditional methods integrate for body rates.
- Inversion method solves the kinematics variables.
- The kinematic variables lead to the control torques.

$$
\begin{aligned}
\epsilon_1 &= \lambda_1 \sin\tfrac{\theta}{2} \\
\epsilon_2 &= \lambda_2 \sin\tfrac{\theta}{2} \\
\epsilon_3 &= \lambda_3 \sin\tfrac{\theta}{2} \\
\epsilon_4 &= \cos\tfrac{\theta}{2}
\end{aligned} \tag{7}
$$

$$
\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \epsilon_4^2 = 1 \tag{8}
$$

$$
\begin{aligned}
\dot{\epsilon}_i &= \dot{\lambda}_i \sin\tfrac{\theta}{2} + \tfrac{\dot{\theta}}{2}\lambda_i \cos\tfrac{\theta}{2} \\
\dot{\epsilon}_4 &= -\tfrac{\dot{\theta}}{2} \sin\tfrac{\theta}{2}
\end{aligned} \tag{9}
$$

$$
\ddot{\epsilon}_i = \ddot{\lambda}_i \sin\tfrac{\theta}{2} + 2\tfrac{\dot{\theta}}{2}\dot{\lambda}_i \cos\tfrac{\theta}{2} + \tfrac{\ddot{\theta}}{2}\lambda_i \cos\tfrac{\theta}{2} - \tfrac{\dot{\theta}^2}{4}\lambda_i \sin\tfrac{\theta}{2}
$$

$$
\ddot{\epsilon}_4 = -\tfrac{\ddot{\theta}}{2} \sin\tfrac{\theta}{2} - \tfrac{\dot{\theta}^2}{4} \cos\tfrac{\theta}{2} \tag{10}
$$

where i = (1,2,3)

Pseudocode

Algorithm 1 Particle Swarm Optimization

Initialize a population of particles with random values positions and velocities from D dimensions in the search space
for Number of Iterations i **do**
  for Each particle i **do**
    Adapt velocity of the particle using Equation 1
    Update the position of the particle using Equation 2
    Evaluate the fitness $J(\overrightarrow{X}_i)$
    Update pBest using Equation 3
    Update gBest using Equation4
  end for
end for

$$V_k^{(j+1)}(i) = c_I V_k^{(j)}(i) + c_C \left[pBest_k^{(j)}(i) - P_k^{(j)}(i) \right] + c_S \left[gBest_k^{(j)} - P_k^{(j)}(i) \right] \tag{1}$$

$$P_k^{(j+1)} = P_k^{(j)} + V_k^{(j)} \qquad (j = 1, \ldots, N_{IT}) \tag{2}$$

$$\mathbf{pBest}^{(j)}(i) = \mathbf{P}^{(l)}(i) \qquad \left(l = arg\ min_{p=1,\ldots,j}\ J^{(p)}(i) \right) \tag{3}$$

$$\mathbf{gBest}^{(j)} = \mathbf{pBest}^{(j)}(q) \qquad \left(q = arg\ min_{i=1,\ldots,N}\ J_{Best}^{(j)}(i) \right) \tag{4}$$

Penalty Function

Equality constraints of the form

$$g_i(\mathbf{x}) = 0 \tag{1}$$

Penalty function is

$$G_i(\mathbf{x}) = k_{eq,i} \left| g_i(\mathbf{x}) \right| \tag{2}$$

For inequality constraints of the form

$$h_j(\mathbf{x}) \leq 0 \tag{3}$$

The penalty function is

$$H_j(\mathbf{x}) = k_{ineq,j} \max \left\{ 0, h_j(\mathbf{x}) \right\} \tag{4}$$

The resulting modified cost function is then

$$J' = t_f + \sum_{i=1}^{N_{eq}} G_i + \sum_{j=1}^{N_{ineq}} H_j \tag{5}$$

where N_{eq} and N_{ineq} are the number of equality and inequality constraints, respectively.

Direct PSO

$$\dot{\omega}_1 = \left[M_1 - \omega_2\omega_3\left(I_3 - I_2\right)\right]/I_1$$
$$\dot{\omega}_2 = \left[M_2 - \omega_3\omega_1\left(I_1 - I_3\right)\right]/I_2$$
$$\dot{\omega}_3 = \left[M_3 - \omega_1\omega_2\left(I_2 - I_1\right)\right]/I_3$$

$$\begin{bmatrix} \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \\ \dot{\epsilon}_4 \end{bmatrix} = \begin{bmatrix} \epsilon_4 & -\epsilon_3 & \epsilon_2 & \epsilon_1 \\ \epsilon_3 & \epsilon_4 & -\epsilon_1 & \epsilon_2 \\ -\epsilon_2 & \epsilon_1 & \epsilon_4 & \epsilon_3 \\ -\epsilon_1 & -\epsilon_2 & -\epsilon_3 & \epsilon_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix}$$

Inverse Dynamics PSO

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} = 2\begin{bmatrix} \epsilon_4 & \epsilon_3 & -\epsilon_2 & -\epsilon_1 \\ -\epsilon_3 & \epsilon_4 & \epsilon_1 & -\epsilon_2 \\ \epsilon_2 & -\epsilon_1 & \epsilon_4 & -\epsilon_3 \\ \epsilon_1 & \epsilon_2 & \epsilon_3 & \epsilon_4 \end{bmatrix} \begin{bmatrix} \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \\ \dot{\epsilon}_4 \end{bmatrix}$$

$$\omega = 2\mathbf{E}\,\dot{\epsilon}$$

$$\dot{\omega} = 2\dot{\mathbf{E}}\dot{\epsilon} + 2\mathbf{E}\ddot{\epsilon}$$

$$M_1 = I_1\dot{\omega}_1 + \omega_2\omega_3\left(I_3 - I_2\right)$$
$$M_2 = I_2\dot{\omega}_2 + \omega_3\omega_1\left(I_1 - I_3\right)$$
$$M_3 = I_3\dot{\omega}_3 + \omega_1\omega_2\left(I_2 - I_1\right)$$

Parallelization Results

Table 5.2. arcaca.aero.psu cluster

# of proc.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ART	131.04	74.91	50.43	37.28	30.21	24.89	21.65	19.21	17.03	15.69	14.26	13.44	12.46	11.53	10.90