ReadMe

1.)

The file submitted has the fully functioning Puma Robot, the object, wine glass isn't completely smooth & would not be programmed to work for variable k & t (The Puma Robot works for variable k & t).

(a.) The code written to generate robot.ang (Robot_code.m)
object.traj (Object_code.m)

(b) making the program work for multiple k & t.

It's relatively easier to generate exact 't' no. of frames mathematically,

$$\begin{array}{l} k\,)\,t\,(\,Q \\ \quad\underline{\;-\;} \\ \quad R \end{array} \qquad t = Q \times (k-1) + R. \qquad \left(\begin{array}{l} Q = \text{Quotinet} \\ R = \text{Reminder} \end{array} \right)$$

`R is always less than 'k-1'.

for the given k & t (5 & 67 respectively

we first compute k-1 (4) ∵ those are the no. of times we perform interpolate

$67/4$ gives $Q = 16$ & $R = 3$.

so, $17 + 17 + 17 + 16 = 67$

What was done is,

we add the Quotient (Q) k-1 times,

if $R = 0$

then, if Q is added k-1 times we get 't'.

if $R \neq 0$

then,

we add +1 to 'R' no. of terms,

from given k & t,

$$\begin{pmatrix} 16 \\ +1 \end{pmatrix} + \begin{pmatrix} 16 \\ +1 \end{pmatrix} + \begin{pmatrix} 16 \\ +1 \end{pmatrix} + 16 = 67$$

This was easier to do for the Robot. ang but unfortunately I couldn't realize it for the obj. traj.

Ⓒ If we continue to use MATLAB then, Aerospace Toolbox would simplify the process in a great detail.

else,

We can use other scripting languages like Python or R, which would help us to perform operations on the Matrices & arrays in a quicker & short way.