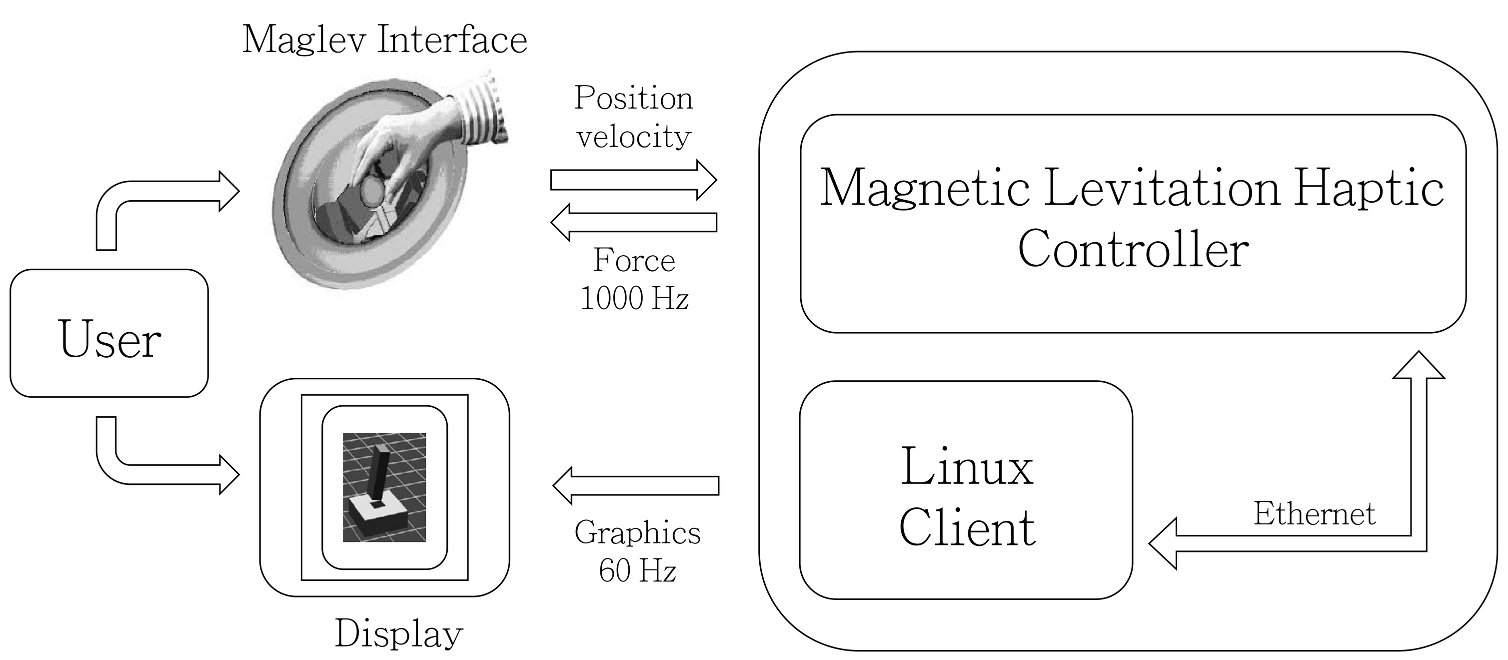
A Square Peg-in-a Square Hole Insertion Task Using Magnetic Levitation Haptic Device

Neel Shihora, Ralph Hollis

Introduction

We present a simulation for a square peg-in-a hole insertion task on a 6-DOF Magnetic levitation haptic device. It allows user to move a peg around along x-y-z directions and put it correctly in the hole in a virtual environment having force feedbacks from a haptic device whenever a peg makes contact with the base having a square hole. This type of demonstrations can be used to train the workers before directly assigning them to work at the assembly line of any particular production unit. The haptic virtual environment can also be used to interact with one or more non existent 3D objects by assembling them together or in some other way.

Approach



Schematic diagram of the system

$$T = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\begin{aligned}
a_{11} &= \cos(\alpha)\cdot\cos(\beta) \\
a_{12} &= \cos(\alpha)\cdot\sin(\beta)\cdot\sin(\gamma) - \sin(\alpha)\cdot\cos(\gamma) \\
a_{13} &= \cos(\alpha)\cdot\sin(\beta)\cdot\cos(\gamma) + \sin(\alpha)\cdot\sin(\gamma) \\
a_{21} &= \sin(\alpha)\cdot\cos(\beta) \\
a_{22} &= \sin(\alpha)\cdot\sin(\beta)\cdot\sin(\gamma) + \cos(\alpha)\cdot\cos(\gamma) \\
a_{23} &= \sin(\alpha)\cdot\sin(\beta)\cdot\cos(\gamma) - \cos(\alpha)\cdot\sin(\gamma) \\
a_{31} &= -\sin(\beta) \\
a_{32} &= \cos(\beta)\cdot\sin(\gamma) \\
a_{33} &= \cos(\beta)\cdot\cos(\gamma)
\end{aligned}$$

where
α = rotation along the x axis of coordinate frame P
β = rotation along the y axis of coordinate frame P
γ = rotation along the z axis of coordinate frame P

These rotation values are set to zero for our recent work.

Transformation Matrix used to transform the vertices of peg from {P}** to {B}***

Define coordinates of all 8 vertices of the peg with respect to P

⇓

$^{B}p_0 = T * {}^{P}p_0$

⇓

Coordinate values of $^{B}p_0 = {}^{B}p_0$ + current actual position of the handle

⇓

Get current location of peg using the coordinates values of all 8 vertices from $^{B}p_{i=0,1,...,7}$

⇓

Constrain the motion of the peg along each of the three axes x, y, and z by choosing the maximum and minimum position values that can be achieved by the peg from that location along respective axis

⇓

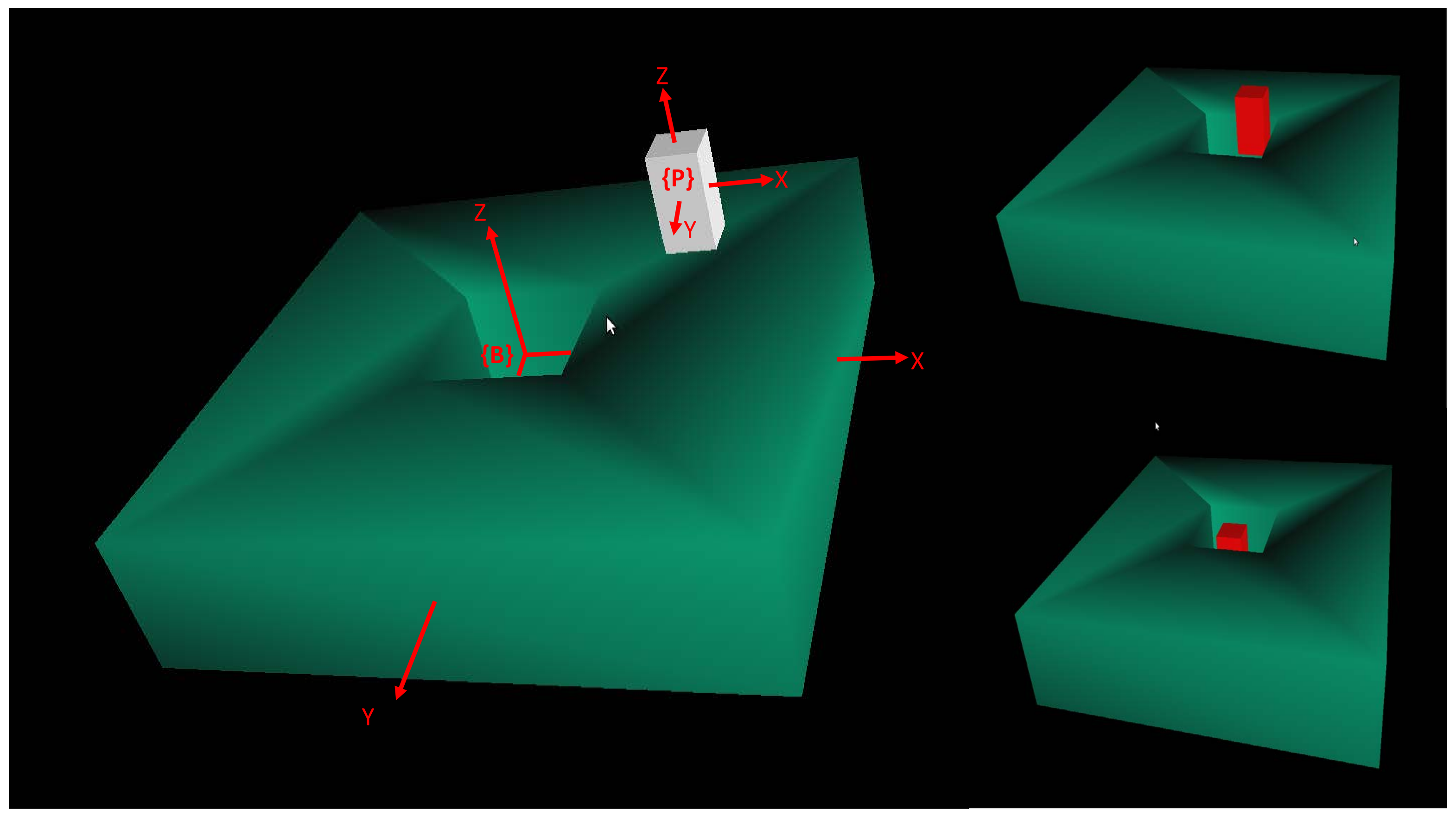
ml_ConstrainAxis()
(This function is provided with the API for the Maglev Haptic Device. It constrains the motion along a particular axis and between the given positions by accepting index of the axis, minimum position, and maximum position as input arguments)

⇓

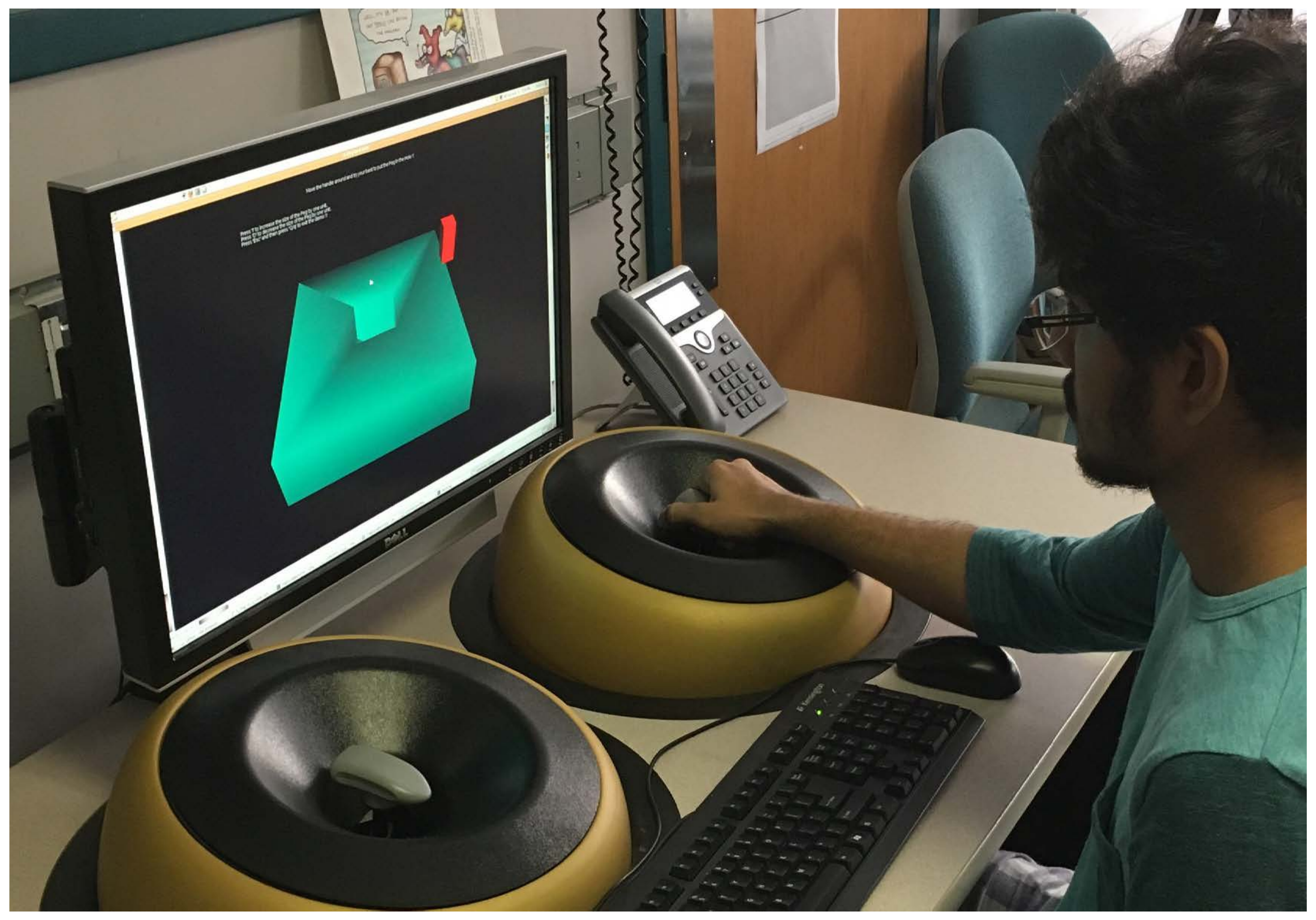
If peg tries to exceed the minimum or maximum position limit then set the values of PIDff gains so that user can feel smooth interaction with virtual environment

New values of PIDff gains ⟹ Magnetic Levitation Haptic Controller which controls the currents flowing through coils of the haptic device ⟹

where
**P = coordinate frame attached with Peg
$^{P}p_0$ = position vector of one vertex of peg w.r.t to P
$^{B}p_0$ = position vector of one vertex of peg w.r.t to B
***B = coordinate frame attached with the Base having a square hole
T = transformation matrix shown on the left side
PIDff = Proportional Integral Derivative feed forward



Left : haptic virtual environment consisting a square peg and a base with a square hole in it and coordinate system attached with them {P} and {B} respectively (we have used coin3d API for graphics)
Right: Contact states



Visual Haptic Interface

Ongoing and Future Work

- In addition to x-y-z motion of the peg use all 6 DOF which also allows user to rotate the peg.
- Add a friction model to generate more realistic interaction with a haptic virtual environment.

Carnegie Mellon
5000 Forbes Avenue, Pittsburgh, PA 15213
Microdynamic Systems Laboratory
RISS Robotics Institute Summer Scholars
THE ROBOTICS INSTITUTE