Faculty of Power and Aeronautical Engineering
Department of Robotics

Robot Programming Methods

Jafar Jafar

supervised by
prof. dr hab. inż. Cezary Zieliński

Contents

List of Figures

1 Introduction

Design a controller for a system defined as follows:

- The hardware of the system consists of a 6dof manipulator and a two-state gripper. The manipulator is actuated by 6 electric motors with encoders attached to their rotors. Moreover, the system is equipped with a Kinect sensor.
- The task of the system is to detect red cubes moving on a conveyor having a white tape, at a velocity not greater than 0.05 m/s and does not vary a lot. The cubes having dimensions of 4cm $\pm$ 0.5cm have to be grasped and placed on a palette 50cm by 50cm in a 10 by10 arrangement. Other cubes should be ignored. The operation of the conveyor is not controlled. The cubes are spread far apart on the conveyor. The conveyor is not stopped when the robot grasps a cube, thus the motion of the cube has to be taken into account during the grasping operation.
- The system starts its work when an empty pallet is available and terminates it when the pallet is full. The START command is issued when an empty pallet is made available and the STOP command when it is full. The START and STOP commands informing about the state of the pallet are issued by a separate agent that need not be specified here.

Highlights:

- The system is composed of a 6dof manipulator, a two-state gripper and a Kinect sensor.
- The manipulator is driven by 6 electric motors with an encoder mounted on the rotor.
- The cubes are spread far apart on the conveyor.
- The red cubes of 4cm $\pm$ 0.5cm should be accommodated in a 10 by 10 arrangement on a pallet, the rest of the cubes should be ignored.
- The conveyor is not stopping when the robot performs grasp operation, more over the conveyor is not controlled and is it behavior is totally independent to our agent.
- The START and STOP commands informing about the state of the pallet are delivered by a separate agent which does not have to be specified.

2 Internal structure of the agent

2.1 Hardware setup

The definition of the agent's internal structure with the required hardware and the roles of this hardware is presented below:

- Manipulator – to approach and translocate the object (6 dof) and it is provided with effector control and state flow of data.
- Gripper – it is a two-state gripper and responsible for grasping and placing of the cubes. It is defined to be effector and effector state flow of data environment.
- Kinect sensor – detects the red cube with size of 4cm ± 0.5cm on the conveyor. And its application constrained to only sending reading of the environment to the control subsystem.

In general, Hardware Role looks like:

- Manipulator (6-dof) – effector.
- Gripper - effector.
- Kinect sensor - receptor (exteroceptor).

2.2 Coordinate frames

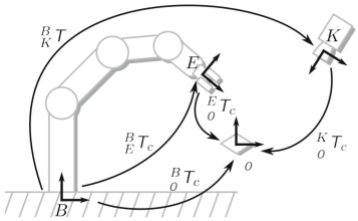

Figure 1: Coordinate frames

- B – robot base reference frame.
- E – end-effector frame.
- K – Kinect frame.
- O – object frame.
- c – current pose.
- T – transformation.
- ${}^{\mathrm{B}}_{\mathrm{K}}\mathrm{T}$ - Kinect w.r.t. base.
- ${}^{\mathrm{B}}_{\mathrm{E}}\mathrm{T}$ - Effector w.r.t. base.
- ${}^{\mathrm{K}}_{\mathrm{O}}\mathrm{T}$ - Object w.r.t. Kinect.
- ${}^{\mathrm{B}}_{\mathrm{O}}\mathrm{T}$ - Object w.r.t. base.
- ${}^{\mathrm{E}}_{\mathrm{O}}\mathrm{T}$ - Object w.r.t. effector.

2.3 General system structure

The structure of the system is analogous to that used in vision-guided machines. The control circle moves the robot to its initial position and makes Kinect obtain RGB-D data from the environment, which is processed to detect a red cube with a white string on it. These objects are far from each other and it is assumed that only one at a time will be visible by Kinect sensors. The pose will be estimated and fed to the controller, which will generate a motion command. It controls the movement of the manipulator as well as the jaw of the gripper, which is assumed to give status whether they are open or closed.

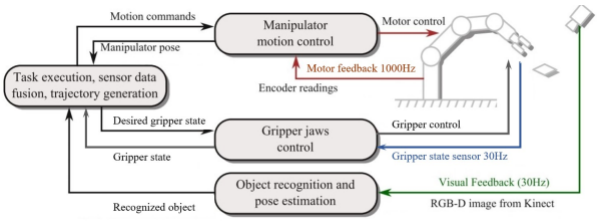

Figure 2: General system structure

2.4 System structure a_j

The system's structure is divided into two real effectors referring to the 6 DOF manipulator and gripper, and one real receptor, the Kinect sensor. With their corresponding virtual effectors and receptor. Figure 3 shows a summarized representation of the structure. Figure 4 illustrates the detailed agent structure including the input and output buffers, as well as the memory.

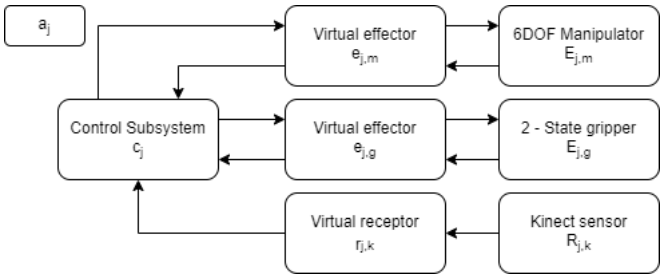

Figure 3: General agent a_j structure

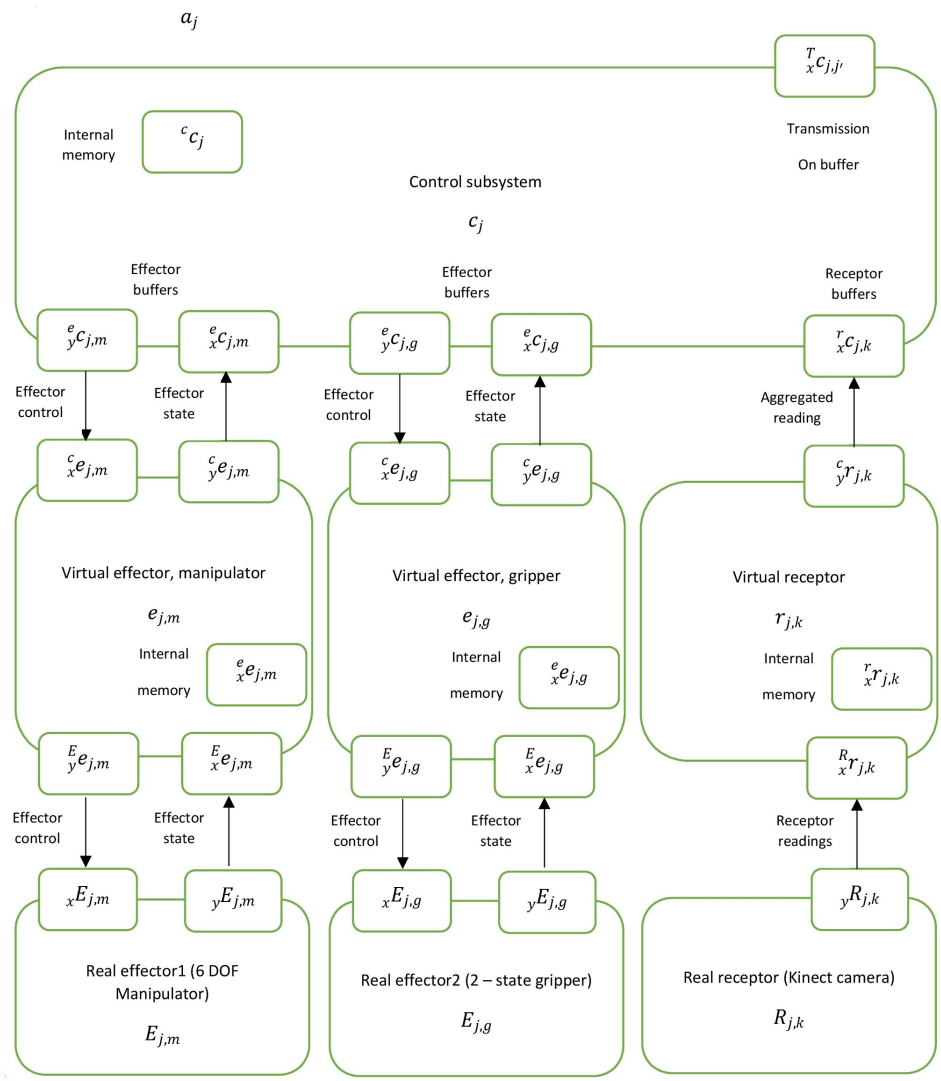

Figure 4: Detailed agent a_j structure

3 Sampling rate of agent subsystem

The sampling rates are defined as follows:

- Virtual receptor Kinect: The sampling speed of the Kinect sensor is determined by the manufacturer. This can take up to 30 frames per seconds, resulting in a sampling period of approximately 30 ms ${}^{r}T = 30\text{ms}$.

- Virtual effector, manipulator: The test for the robot in the ABB data sheet is carried out for a speed of 1.6 m/s, so that it is safe to assume that the maximum linear velocity of the manipulator can be 1m/s ${}^{e,m}T = 1\text{ms}$. The conveyor moving with a speed of 0.05 m/s so that each manipulator sample of an object on the conveyor moves with a velocity the maximum speed will move 0.05mm,

this range is acceptable to use.

- Virtual effector, gripper: We consider effectors with same sampling rate as the manipulator, 1000hz ${}^{e,g}\mathrm{T} = 1\mathrm{ms}$.the conveyor is moving at a speed of 0.05 m/s so for every manipulator sample, an object on the conveyor moving at maximum speed will moving 0.05mm, this range is acceptable for use.

- Control subsystem: The sampling rate of the control subsystem is determined for highest sampling rate frequency, ${}^{c}\mathrm{T} = 1\mathrm{ms}$ delivers control commands to the virtual effectors, manipulator and gripper.

4 General behavior

4.1 Manipulator virtual effector

The virtual manipulator effector corresponds to the interface between the real 6dof manipulator and controller. It will take as input a motion command containing homogeneous the transformation matrix of the desired position, ${}^{\mathrm{B}}_{\mathrm{E}}\mathrm{T}_{\mathrm{d}}$, produces a path between the current pose and aim, and find the desired motor value, which will be output as current. It is assumed that the speed of the manipulator is 1m/s for the joint space, for the length movement, and 0.1 m/s in the task room, for precise short movements. In addition, the position will be read from the encoder and will be fed to the controller after completing the DKP as the current transformation matrix ${}^{\mathrm{B}}_{\mathrm{E}}\mathrm{T}_{\mathrm{c}}$.

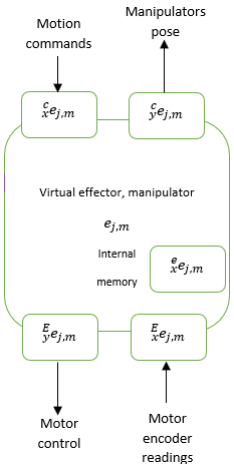

Figure 5: Virtual effector for the manipulator

Input buffers of the manipulator virtual effector $\mathrm{e}_{\mathrm{j,m}}$ from the control subsystem c_{j}. For PTP (Point to Point) trajectory generation:

- ${}^{\mathrm{c}}_{\mathrm{x}}\mathrm{e}_{\mathrm{j,m}}$ $[{}^{\mathrm{B}}_{\mathrm{E}}\mathrm{T}_{\mathrm{d}}]$, Desired cartesian pose of end-effector (E) w.r.t. robot base (B).

- ${}^{\mathrm{c}}_{\mathrm{x}}\mathrm{e}_{\mathrm{j,m}}$ [space], Boolean variable indicates the space in which the manipulator will compute the trajectory. Corresponds to 1 for Task space, and 0 for Joint space.

Output buffers of the manipulator virtual effector $e_{j,m}$ to the control subsystem c_j.

- ${}^{c}_{y}e_{j,m}$ $[{}^{B}_{E}T_c]$, Current cartesian pose of end-effector (E) w.r.t. robot base (B).

Input buffers of the manipulator real effector $E_{j,m}$ to the virtual effector $e_{j,m}$.

- ${}^{E}_{x}e_{j,m}$ $[m_m]$,Currently measured motor positions.

Output buffers to the manipulator real effector $E_{j,m}$ from the virtual effector $e_{j,m}$.

- ${}^{E}_{y}e_{j,m}$ $[c_d]$, Desired values of current driving manipulator.

Internal memory

- ${}^{e}e_{j,m}$ $[m_d]$, Desired motor absolute positions for gravity compensation.
- ${}^{e}e_{j,m}$ $[q_d]$, The previous connection configuration, is used to select the correct solution for the inverse kinematics problem.

4.2 Gripper virtual effector

The gripper virtual effector corresponds to the interface between the real 2-state gripper and the controller. It will receive as an input the desired value of the gripper g_{state} and output the corresponding current to the real device. The jaw is assumed to measure its state with a boolean switch that will be sent back to the control subsystem.

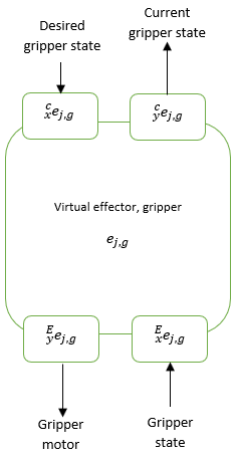

Figure 6: Virtual effector for the gripper

Input buffers of the gripper virtual effector $e_{j,g}$ from the control subsystem c_j.

- ${}^{c}_{x}e_{j,g}$ $[g_{state_d}]$, Boolean variable indicating desired gripper position. Open (0) or closed (1).

Output buffers of the gripper virtual effector $e_{j,g}$ to the control subsystem c_j.

- ${}^{c}_{y}e_{j,g}$ [g_{state_c}], Boolean variable indicating current gripper position. Open (0) or closed (1).

Input buffers of the gripper real effector $E_{j,g}$ from the virtual effector $e_{j,g}$.

- ${}^{E}_{x}e_{j,g}$ [g_{state_m}], Measured gripper state.

Output buffers to the gripperr real effector $E_{j,g}$ from the virtual effector $e_{j,g}$.

- ${}^{E}_{y}e_{j,g}$ [c_g], Desired gripper state current supply.

4.3 Kinect virtual receptor

Kinect virtual receptors correspond to the interface between the camera and the real Kinect controller. It will accept the image read from the device, and compare it to the pre-configured red memory cube model with white bands and their specific dimensions. The output will be a verified hypothesis with the cube coordinates at the scene. The result will also contain the cube velocity obtained from the knowledge of the position difference in space and the sensor frame rate.

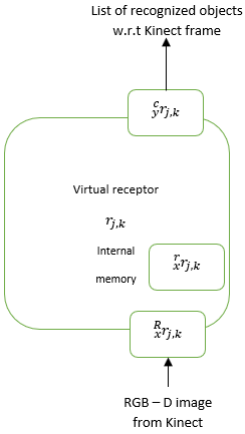

Figure 7: Virtual receptor for the kinect

Input buffers of the Kinect virtual receptor $r_{j,k}$ from the real receptor $R_{j,k}$

- ${}^{R}_{x}r_{j,k}$ [I_c^{RGB}], RGB image with three channels.
- ${}^{R}_{x}r_{j,k}$ [I_c^{D}], Depth image with a single channel.

Output buffers of the Kinect virtual receptor $r_{j,k}$ to the control subsystem c_j.

- ${}^{c}_{y}r_{j,k}$ [${}^{K}O_c$], Verified hypothesis of identified red cube with white tape w.r.t Kinect frame.

Internal memory

- ${}^{r}r_{j,k}$ [P_O], Parameters of the object that needs to be picked. In this case, red cubes, with a white tape, and dimensions of 4cm $\pm$ 0.5cm.

- ${}^{r}r_{j,k}$ [P_K], Intrinsic parameters of the Kinect sensor (camera matrix, distortion coefficient, frame rate).

5 Data structures

The data structure used in this task corresponds to a virtual receptor. It is assumed that because the blocks are far apart on the conveyor belt, the Kinect sensors will only detect one at a time. The red block with the white band is represented by an object model containing the RGB colour point cloud information of the cube, $C = C^{RGB}$. When recognized by Kinect, the verified hypothesis was ${}^{K}O_c = ({}^{K}C_c^{RGB}, {}^{K}T_c^{D}, c_c, v_c)$. Where:

- ${}^{K}C_c^{RGB}$: A point cloud RGB object model is projected onto the scene in relation to the Kinect frame.
- ${}^{K}T_c^{D}$: The position of the object on the conveyor belt w.r.t. Kinect Frame.
- c_c: Predictive confidence scores were obtained from image processing techniques to verify the hypothesis of a red cube with a white tape.
- v_c: Identified cube speed in conveyor.

5.1 Buffers

The buffers from the control subsystem are given by:

- ${}^{e}_{y}c_{j,m} = {}^{c}_{x}e_{j,m}$
- ${}^{e}_{x}c_{j,m} = {}^{c}_{y}e_{j,m}$
- ${}^{e}_{y}c_{j,g} = {}^{c}_{x}e_{j,g}$
- ${}^{e}_{x}c_{j,g} = {}^{c}_{y}e_{j,g}$
- ${}^{e}_{x}c_{j,k} = {}^{c}_{y}r_{j,k}$
- ${}^{T}_{x}c_{j,j}$,: A transmission buffer that accepts START, and STOP commands from external agents.

5.2 Memory

5.2.1 System calibration

System parameters:

- ${}^{c}c_{j}[\,{}^{B}_{K}T]$: Position of the Kinect camera with respect to the robot base.

- ${}^{c}c_{j}[\ {}^{B}_{Es}T]$: Constant homogeneous transformation matrix of the start position.

Grasping:

- ${}^{c}c_{j}[\ {}^{Pg}_{G}T]$: Constant homogeneous transformation matrix between the grasp and pre-grasp poses.

Placing:

- ${}^{c}c_{j}[\ {}^{Pp}_{p}T]$: Constant homogeneous transformation matrix between the place and pre-place poses.

5.2.2 Place parameters for the palette

- ${}^{c}c_{j}[\ {}^{B}_{F}T]$: First position in the palette with respect to the robot base.
- ${}^{c}c_{j}[\ {}^{B}_{c}T]$: Current position in the palette with respect to the robot base. Set to ${}^{B}_{F}T$ every time the memory buffer cmd switches from STOP to START.
- ${}^{c}c_{j}[\ [d_{cubes}]$: Constant distance between the cubes in the palette, where $d_{cubes}[x] = [5cm]$ and $d_{cubes}[x,y] = [5cm, -45cm]$.
- ${}^{c}c_{j}[\ [s_{cubes}]$: Calculates addition for the cubes placed in the pack. Must be reset every time cmd buffer memory goes from STOP to START.

5.2.3 Perceived model of the scene

- ${}^{c}c_{j}[\ {}^{B}O]$: Verified hypothesis w.r.t. robot base.

5.2.4 Desired end-effector and gripper poses

- ${}^{c}c_{j}[\ {}^{B}_{Eg}T_{d}]$: Desired pose of the manipulator w.r.t. robot base reference frame in which the object is going to be picked up.
- ${}^{c}c_{j}[\ {}^{B}_{Epg}T_{d}]$: Desired pose of the manipulator w.r.t. robot base reference frame for the pregrasp position.
- ${}^{c}c_{j}[\ {}^{B}_{Ep}T_{d}]$: Desired pose of the manipulator w.r.t. robot base reference frame in which the object is going to be placed.
- ${}^{c}c_{j}[\ {}^{B}_{Epp}T_{d}]$: Desired pose of the manipulator w.r.t. robot base reference frame for the preplace position.
- ${}^{c}c_{j}[\ g_{state_{d}}]$: Desired state of the gripper. Can be open (1) or closed (0).

5.2.5 Others

- ${}^{c}c_{j}[\ {}^{E}_{TCP}T_{d}]$: Constant transformation between the the pose of the center of the gripper, also known

as tool control point (TCP) w.r.t. the end-effector (E).

- ${}^{c}c_{j}$[cmd]: Value of the last command issued by the external agent. Can only be START or STOP (should be initialized to STOP to keep the agent in the idle state).

6 Finite state machine

Three graphs of the agent finite-state machine were plotted. The first one, Figure 8, shows the different stages clearly indicated with their names, joint space has been reduced to j.s. and task space to t.s. for briefness. The second graph, Figure 9 is corresponds to the previous diagram but with the variable corresponding to the states. Finally, Figure 10 displays all the states plotted as a cycle with the initial conditions, state variables and corresponding behaviours. Only the names of the initial conditions were shown for a better overall view of the graph.

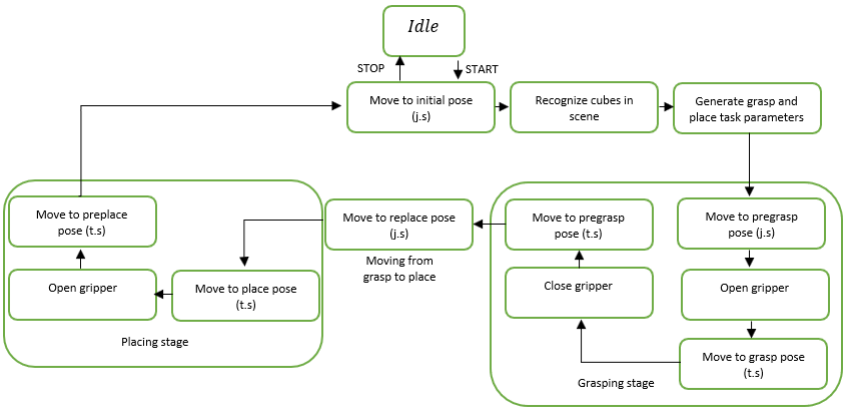

Figure 8: Finite State Machine flow diagram with names

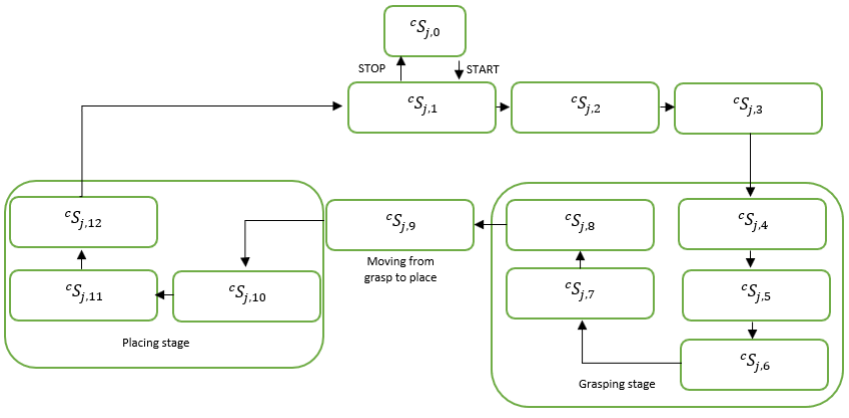

Figure 9: Finite State Machine flow diagram with states

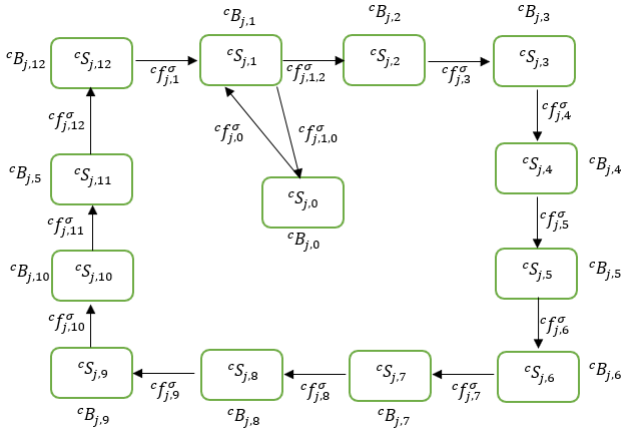

Figure 10: Finite State Machine flow diagram with variables

6.1 Mapping of the control subsystem states to its behaviors

State	Behaviour	Description
${}^{c}S_{j,0}$	${}^{c}B_{j,0}$	Idle
${}^{c}S_{j,1}$	${}^{c}B_{j,1}$	Move to initial pose (joint space).
${}^{c}S_{j,2}$	${}^{c}B_{j,2}$	Recognize cubes in scene.
${}^{c}S_{j,3}$	${}^{c}B_{j,3}$	Generate grasp and place task parameters.
${}^{c}S_{j,4}$	${}^{c}B_{j,4}$	Move to pregrasp pose (joint space).
${}^{c}S_{j,5}$	${}^{c}B_{j,5}$	Open gripper.
${}^{c}S_{j,6}$	${}^{c}B_{j,6}$	Move to grasp pose (task space).
${}^{c}S_{j,7}$	${}^{c}B_{j,7}$	Close gripper.
${}^{c}S_{j,8}$	${}^{c}B_{j,8}$	Move to pregrasp pose (task space).
${}^{c}S_{j,9}$	${}^{c}B_{j,9}$	Move to preplace pose (joint space).
${}^{c}S_{j,10}$	${}^{c}B_{j,10}$	Move to place pose (task space).
${}^{c}S_{j,11}$	${}^{c}B_{j,5}$	Open gripper.
${}^{c}S_{j,12}$	${}^{c}B_{j,12}$	Move to preplace (task space).

6.2 Behavior terminal conditions

$$
\begin{aligned}
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},0} \triangleq (CMD = START)\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},1} \triangleq ({}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{c}} = {}^{\mathrm{B}}_{\mathrm{Es}}T_{\mathrm{d}})\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},2} \triangleq (\exists^{\mathrm{B}}O)\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},3} \triangleq (TRUE)\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},4} \triangleq ({}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{c}} = {}^{\mathrm{B}}_{\mathrm{Epg}}T_{\mathrm{d}})\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},5} \triangleq (g_{\mathrm{state_c}} = 1)\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},6} \triangleq ({}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{c}} = {}^{\mathrm{B}}_{\mathrm{Eg}}T_{\mathrm{d}})\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},7} \triangleq (g_{\mathrm{state_c}} = 0)\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},8} \triangleq ({}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{c}} = {}^{\mathrm{B}}_{\mathrm{Eg}}T_{\mathrm{d}}) \triangleq {}^{\mathrm{c}}f^{\tau}_{\mathrm{j},4}\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},9} \triangleq ({}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{c}} = {}^{\mathrm{B}}_{\mathrm{Epp}}T_{\mathrm{d}})\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},10} \triangleq ({}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{c}} = {}^{\mathrm{B}}_{\mathrm{Ep}}T_{\mathrm{d}})\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},11} \triangleq (g_{\mathrm{state_c}} = 1) \triangleq {}^{\mathrm{c}}f^{\tau}_{\mathrm{j},5}\\
&{}^{\mathrm{c}}f^{\tau}_{\mathrm{j},12} \triangleq ({}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{c}} = {}^{\mathrm{B}}_{\mathrm{Epp}}T_{\mathrm{c}}) \triangleq {}^{\mathrm{c}}f^{\tau}_{\mathrm{j},9}
\end{aligned}
$$

6.3 Behavior error conditions

For all $l \in \{0,1,2,..,12\}$, the error condition is given by:

$$
{}^{\mathrm{c}}f^{\varepsilon}_{\mathrm{j,l}} = FALSE
$$

6.4 Behavior initial conditions

The initial conditions from state 1 to states 0 and 2 are given by:

$$
{}^{\mathrm{c}}f^{\sigma}_{\mathrm{j,l},0} \triangleq (cmd = STOP)
$$

$$
{}^{\mathrm{c}}f^{\sigma}_{\mathrm{j,l},2} \triangleq (cmd = START)
$$

Knowing that a) ${}^{\mathrm{c}}\mathrm{B}_{\mathrm{j},1}$ finishes when terminal condition of ${}^{\mathrm{c}}\mathrm{S}_{\mathrm{j},1}$ is false and b) the controller memory buffer cmd can only hold one value (START or STOP), the separability and integrality conditions can be observed from:

$$
{}^{\mathrm{c}}f^{\sigma}_{\mathrm{j,l},0} \wedge {}^{\mathrm{c}}f^{\sigma}_{\mathrm{j,l},2} = START \wedge STOP = FALSE
$$

$$^{c}f^{\sigma}_{j,l,0} \vee {}^{c}f^{\sigma}_{j,l,2} = START \vee STOP = TRUE$$

It can be proven that that the rest of the states following a sequential form fulfill the separability and integrality conditions. For the other states, $l \in \{0,1,2,..,12\}$, the initial conditions are:

$$^{c}f^{\sigma}_{j,l} = TRUE$$

6.5 State transition table of the control subsystem FSM

Current State		Next State	
State	**Terminal/Error condition**	**Initial condition**	**State**
$^{c}S_{j,0}$	$^{c}f^{\tau}_{j,0}$=TRUE	$^{c}f^{\tau}_{j,1}$=TRUE	$^{c}S_{j,1}$
$^{c}S_{j,1}$	$^{c}f^{\tau}_{j,1}$=TRUE	$^{c}f^{\tau}_{j,1,2}$=TRUE	$^{c}S_{j,2}$
$^{c}S_{j,1}$	$^{c}f^{\tau}_{j,1}$=TRUE	$^{c}f^{\tau}_{j,1,0}$=TRUE	$^{c}S_{j,0}$
$^{c}S_{j,2}$	$^{c}f^{\tau}_{j,2}$=TRUE	$^{c}f^{\tau}_{j,3}$=TRUE	$^{c}S_{j,3}$
$^{c}S_{j,3}$	$^{c}f^{\tau}_{j,3}$=TRUE	$^{c}f^{\tau}_{j,4}$=TRUE	$^{c}S_{j,4}$
$^{c}S_{j,4}$	$^{c}f^{\tau}_{j,4}$=TRUE	$^{c}f^{\tau}_{j,5}$=TRUE	$^{c}S_{j,5}$
$^{c}S_{j,5}$	$^{c}f^{\tau}_{j,5}$=TRUE	$^{c}f^{\tau}_{j,6}$=TRUE	$^{c}S_{j,6}$
$^{c}S_{j,6}$	$^{c}f^{\tau}_{j,6}$=TRUE	$^{c}f^{\tau}_{j,7}$=TRUE	$^{c}S_{j,7}$
$^{c}S_{j,7}$	$^{c}f^{\tau}_{j,7}$=TRUE	$^{c}f^{\tau}_{j,8}$=TRUE	$^{c}S_{j,8}$
$^{c}S_{j,8}$	$^{c}f^{\tau}_{j,8}$=TRUE	$^{c}f^{\tau}_{j,9}$=TRUE	$^{c}S_{j,9}$
$^{c}S_{j,9}$	$^{c}f^{\tau}_{j,9}$=TRUE	$^{c}f^{\tau}_{j,10}$=TRUE	$^{c}S_{j,10}$
$^{c}S_{j,10}$	$^{c}f^{\tau}_{j,10}$=TRUE	$^{c}f^{\tau}_{j,11}$=TRUE	$^{c}S_{j,11}$
$^{c}S_{j,11}$	$^{c}f^{\tau}_{j,11}$=TRUE	$^{c}f^{\tau}_{j,12}$=TRUE	$^{c}S_{j,12}$
$^{c}S_{j,12}$	$^{c}f^{\tau}_{j,12}$=TRUE	$^{c}f^{\tau}_{j,1}$=TRUE	$^{c}S_{j,1}$

6.6 Behavior definitions

- Behavior $^{c}B_{j,0}$, Corresponding to idle state behavior. It is defined as:

$$^{c}B_{j,0} \coloneqq {}^{c}B_{j,0}({}^{c,c}f_{j,0}, {}^{c}f^{\tau}_{j,0}, {}^{c}f^{\varepsilon}_{j,0})$$

  Its transition function $^{c,c}f_{j,0}$ is:

$$^{c,c}f_{j,0}(c_j) \triangleq Wait()$$

- Behavior $^{c}B_{j,1}$, Corresponding to move to initial pose in the joint space state behavior. It is defined as:

$$^{c}B_{j,1} \coloneqq {}^{c}B_{j,1}({}^{c,e}f_{j,1}, {}^{c}f^{\tau}_{j,1}, {}^{c}f^{\varepsilon}_{j,1})$$

Its transition function ${}^{c,e}\mathrm{f}_{j,1}$ is separated into two sub-functions:

$$ {}^{e}_{y}c^{i+1}_{j,m}[{}^{B}_{E}T_{d}] = {}^{c,e}f_{j,1,1}({}^{e}_{y}c^{i}_{j,m}) \triangleq {}^{B}_{Es}T $$

$$ {}^{e}_{y}c^{i+1}_{j,m}[space] = {}^{c,e}f_{j,1,2}({}^{e}_{y}c^{i}_{j,m}) \triangleq 0(Jointspace) $$

- Behavior ${}^{c}\mathrm{B}_{j,2}$, Corresponding to scene recognition state behavior. Where the pose of the blocks w.r.t. robot base B are computed. It is defined as:

$$ {}^{c}B_{j,2} \coloneqq {}^{c}B_{j,2}({}^{c,c}f_{j,2}, {}^{c}f^{\tau}_{j,2}, {}^{c}f^{\varepsilon}_{j,2}) $$

Its transition function ${}^{c,c}\mathrm{c}_{j,m}$ is:

$$ {}^{e}_{y}c^{i+1}_{j,m}[{}^{B}O] \coloneqq {}^{c,c}f_{j,2}({}^{e}_{y}c^{i}_{j,m}) \triangleq GetCubePose({}^{K}O, {}^{B}_{K}T) $$

Where GetCubePose() defines the transformation${}^{B}\mathrm{O}= {}^{K}\mathrm{O}, {}^{B}_{K}\mathrm{T}$

- Behavior ${}^{c}\mathrm{B}_{j,3}$, Corresponding to generate grasp and place task parameters inside the controller. It is defined as:

$$ {}^{c}B_{j,3} \coloneqq {}^{c}B_{j,3}({}^{c,c}f_{j,3}, {}^{c}f^{\tau}_{j,3}, {}^{c}f^{\varepsilon}_{j,3}) $$

Its transition function ${}^{c,c}\mathrm{f}_{j,3}$ can be divided into four transition sub-functions:

$$ Pre-grasppose : {}^{e}_{y}c^{i+1}_{j,m}[{}^{B}_{Epg}T_{d}] \coloneqq {}^{c,c}f_{j,3,1}({}^{e}_{y}c^{i}_{j,m}) \triangleq StaticGraspCompute({}^{B}O, {}^{E}_{TCP}T){}^{P}_{G}T^{-1} $$

$$ Grasppose : {}^{e}_{y}c^{i+1}_{j,m}[{}^{B}_{Eg}T_{d}] \coloneqq {}^{c,c}f_{j,3,2}({}^{e}_{y}c^{i}_{j,m}) \triangleq DynamicGraspCompute({}^{B}O, {}^{E}_{TCP}T) $$

$$ Pre{-}placepose : {}^{e}_{y}c^{i+1}_{j,m}[{}^{B}_{Epp}T_{d}] \coloneqq {}^{c,c}f_{j,3,3}({}^{e}_{y}c^{i}_{j,m}) \triangleq PlaceCompute({}^{B}_{F}T, d_{cubes}, c_{cubes}, {}^{E}_{TCP}T){}^{P}_{G}T^{-1} $$

$$ Placepose : {}^{e}_{y}c^{i+1}_{j,m}[{}^{B}_{Ep}T_{d}] \coloneqq {}^{c,c}f_{j,3,4}({}^{e}_{y}c^{i}_{j,m}) \triangleq PlaceCompute({}^{B}_{F}T, d_{cubes}, c_{cubes}, {}^{E}_{TCP}T){}^{P}_{G}T $$

- StaticGraspCompute() function definition

  This auxiliary function will allow for the computation of the pregrasp pose of the robot, so that it positions and orients correctly on top of the cube. The returning value of this function will only be used to find the pregrasp pose. The real grasp position for the moving cube will be found out by the function DynamicGraspCompute() called after it. First it is necessary to consider the desired result, and the available data:

  - Output: ${}^{B}_{EG}\mathrm{T}$. The grasping pose coincides with the center of the object.
  - Inputs: ${}^{B}\mathrm{O}$, ${}^{E}_{TCP}\mathrm{T}$

Given that the TCP should coincide with the object's position ($^{\mathrm{B}}_{\mathrm{O}}\mathrm{T} = {}^{\mathrm{B}}_{\mathrm{TCP}}\mathrm{T}$), the transformation needed to achieve the result is given by:

$$^{\mathrm{B}}_{\mathrm{EG}}T_{\mathrm{d}} = {}^{\mathrm{B}}_{\mathrm{O}}T\,{}^{\mathrm{E}}_{\mathrm{TCP}}T^{-1}$$

Where, the homogeneous transformation between the object and base frame can be obtained from the translation, and the Euler rotation matrix:

$$^{\mathrm{B}}_{\mathrm{O}}T = \begin{bmatrix} c\alpha c\beta & (c\alpha s\beta s\gamma) - (s\alpha c\gamma) & (c\alpha s\beta c\gamma) + (s\alpha s\gamma) & {}^{\mathrm{B}}p_{\mathrm{x}} \\ s\alpha s\beta & (s\alpha s\beta s\gamma) + (c\alpha c\gamma) & (s\alpha s\beta c\gamma) - (c\alpha s\gamma) & {}^{\mathrm{B}}p_{\mathrm{y}} \\ -s\beta & c\beta s\gamma & c\beta c\gamma & {}^{\mathrm{B}}p_{\mathrm{z}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Angles α, β, γ correspond to the rotation of the cube w.r.t. the base XYZ axis respectively. Note: Notation has been defined for simplicity of representation: $cos(\alpha_{\mathrm{i}}) = c\alpha_{\mathrm{i}}$, $sin(\alpha_{\mathrm{i}}) = s\alpha_{\mathrm{i}}$, and the position of the object in the base frame ${}^{\mathrm{B}}\mathrm{P} = [{}^{\mathrm{B}}\mathrm{p}_{\mathrm{x}}, {}^{\mathrm{B}}\mathrm{p}_{\mathrm{y}}, {}^{\mathrm{B}}\mathrm{p}_{\mathrm{z}}]_{\mathrm{T}}$

- DynamicGraspCompute() function definition

  After calculating the pregrasp pose, this function will calculate the actual grasping pose consider moving objects on conveyor belts. Figure 11 will be a reference. d is the distance between the pregrasp pose and grip, v_{R} is the speed determined from manipulator for this motion (0.1 m/s in the task space), v_{c} is the speed of the cube inconveyor, and $\mathrm{x}_{\mathrm{reach}}$ is the distance required for the robot to reach the gripping position of the pre-hold pose. Speed of conveyor belt and cube, v_{c}, obtained from virtual Kinect receptor, and the robot will make an approach movement with a speed of 0.1 m/s until gripper and object parallel. This can be done because the pose of the cube is always known by knowing the speed of the conveyor. Since the calculation is done only once, and movement will be carried out in the task room, setting intermediate waypoints is not important.

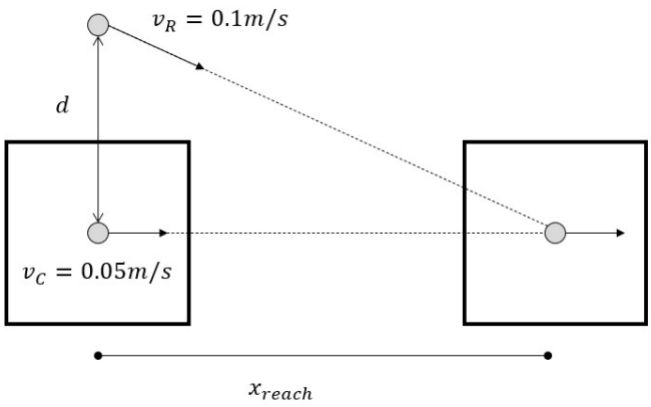

Figure 11: Moving cube gripper pose computation

The total distance, assuming the base X axis is aligned with the direction of the movement of the conveyor, and the time taken to perform this, are given by:

$$x_{\text{reach}} = d\sqrt{\frac{v_{\text{C}}^2}{v_{\text{R}}^2 - v_{\text{C}}^2}}$$

$$t_{\text{reach}} = \frac{\sqrt{x^2 + d^2}}{v_{\text{R}}}$$

The necessary transformation to be performed is:

$$\overset{\text{B}}{\text{Eg}}T_{\text{d}} = \overset{\text{B}}{\text{Epg}}T + \overset{\text{B}}{\text{x}}T$$

Where in the case that there is no movement in the Y axis, keeping the end-effector orientation, the matrix $\overset{\text{B}}{\text{x}}\text{T}$ is given as:

$$\overset{\text{B}}{\text{x}}T = \begin{bmatrix} 0 & 0 & 0 & x_{\text{reach}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- PlaceCompute() function definition

  In order to obtain the desired transformation $\overset{\text{B}}{\text{Ep}}\text{T}_{\text{d}}$, the next steps are similar to the ones followed in GraspCompute(). Again it is necessary to consider the desired result, and the available data:

  - Output: $\overset{\text{B}}{\text{Ep}}\text{T}_{\text{d}}$. The placing pose of the end-effector coincides with the center of the object
  - Input: $\overset{\text{B}}{\text{F}}\text{T}$, d_{cubes}, c_{cubes}, $\overset{\text{E}}{\text{TCP}}\text{T}$

Assuming the palette has the XY axis aligned with the base of the robot, the current XYZ place position is defined by:

if modulus(c_{cubes},10) $\neq 0$
${}^{B}_{c}T_{(1,5)}={}^{B}_{c}T_{(1,5)}+d_{cubes}[x]$
else
${}^{B}_{c}T_{(1,5),(2,5)}={}^{B}_{c}T_{(1,5),(2,5)}+d_{cubes}[x,y]^{T}$
end.

This condition indicates that if the number of items from the cube counter is multiple of 10, a row will be filled, and the manipulator will move to the next row. Additionally, orientation should be perpendicular to the place position of the object, and aligned with the XY axis, leading to a variable with similar characteristics as ${}^{B}O$ from GraspCompute(). The transformation matrix computation is analogous using the Euler rotation.

- Behavior ${}^{c}B_{j,4}$, Corresponding to move to pregrasp pose in the joint space state behavior. It is defined as:

$$ {}^{c}B_{j,4} \coloneqq {}^{c}B_{j,4}({}^{c,e}f_{j,4}, {}^{c}f^{\tau}_{j,4}, {}^{c}f^{\varepsilon}_{j,4}) $$

Its transition function ${}^{c,e}f_{j,4}$ is:

$$ {}^{e}_{y}c^{i+1}_{j,m}[{}^{B}_{E}T_{d}] = {}^{c,e}f_{j,4,1}({}^{e}_{y}c^{i}_{j,m}) \triangleq {}^{B}_{Epg}T_{d} $$

$$ {}^{e}_{y}c^{i+1}_{j,m}[space] = {}^{c,e}f_{j,4,2}({}^{e}_{y}c^{i}_{j,m}) \triangleq 0(Jointspace) $$

- Behavior ${}^{c}B_{j,5}$, Corresponding to open gripper state behavior. It is defined as:

$$ {}^{c}B_{j,5} \coloneqq {}^{c}B_{j,5}({}^{c,e}f_{j,5}, {}^{c}f^{\tau}_{j,5}, {}^{c}f^{\varepsilon}_{j,5}) $$

Its transition function ${}^{c,e}f_{j,5}$ is:

$$ {}^{e}_{y}c^{i+1}_{j,m}[g_{state_d}] = {}^{c,e}f_{j,5}({}^{e}_{y}c^{i}_{j,m}) \triangleq (1 = OPEN) $$

- Behavior ${}^{c}B_{j,6}$, Corresponding to move to grasp pose in the task space state behavior. It is defined as:

$$ {}^{c}B_{j,6} \coloneqq {}^{c}B_{j,6}({}^{c,e}f_{j,6}, {}^{c}f^{\tau}_{j,6}, {}^{c}f^{\varepsilon}_{j,6}) $$

Its transition function ${}^{c,e}\mathrm{f}_{\mathrm{j},6}$ is:

$$ {}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}+1}_{\mathrm{j,m}}[{}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{d}}] = {}^{c,e}f_{\mathrm{j},6,1}({}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}}_{\mathrm{j,m}}) \triangleq {}^{\mathrm{B}}_{\mathrm{EG}}T_{\mathrm{d}} $$

$$ {}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}+1}_{\mathrm{j,m}}[space] = {}^{c,e}f_{\mathrm{j},6,2}({}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}}_{\mathrm{j,m}}) \triangleq 1(Taskspace) $$

- Behavior ${}^{\mathrm{c}}\mathrm{B}_{\mathrm{j},7}$, Corresponding to close gripper state behavior. It is defined as:

$$ {}^{\mathrm{c}}B_{\mathrm{j},7} \coloneqq {}^{\mathrm{c}}B_{\mathrm{j},7}({}^{c,e}f_{\mathrm{j},7}, {}^{\mathrm{c}}f^{\tau}_{\mathrm{j},7}, {}^{\mathrm{c}}f^{\varepsilon}_{\mathrm{j},7}) $$

Its transition function ${}^{c,e}\mathrm{f}_{\mathrm{j},7}$ is:

$$ {}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}+1}_{\mathrm{j,m}}[g_{\mathrm{state_d}}] = {}^{c,e}f_{\mathrm{j},7}({}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}}_{\mathrm{j,m}}) \triangleq (0 = CLOSED) $$

- Behavior ${}^{\mathrm{c}}\mathrm{B}_{\mathrm{j},8}$, Corresponding to move to pregrasp pose in the task space state behavior. It is defined as:

$$ {}^{\mathrm{c}}B_{\mathrm{j},8} \coloneqq {}^{\mathrm{c}}B_{\mathrm{j},8}({}^{c,e}f_{\mathrm{j},8}, {}^{\mathrm{c}}f^{\tau}_{\mathrm{j},8}, {}^{\mathrm{c}}f^{\varepsilon}_{\mathrm{j},8}) $$

Its transition function ${}^{c,e}\mathrm{f}_{\mathrm{j},8}$ is:

$$ {}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}+1}_{\mathrm{j,m}}[{}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{d}}] = {}^{c,e}f_{\mathrm{j},8,1}({}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}}_{\mathrm{j,m}}) \triangleq {}^{\mathrm{B}}_{\mathrm{Epg}}T_{\mathrm{d}} $$

$$ {}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}+1}_{\mathrm{j,m}}[space] = {}^{c,e}f_{\mathrm{j},8,2}({}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}}_{\mathrm{j,m}}) \triangleq 1(Taskspace) $$

- Behavior ${}^{\mathrm{c}}\mathrm{B}_{\mathrm{j},9}$, Corresponding to move to preplace pose in the joint space state behavior. It is defined as:

$$ {}^{\mathrm{c}}B_{\mathrm{j},9} \coloneqq {}^{\mathrm{c}}B_{\mathrm{j},9}({}^{c,e}f_{\mathrm{j},9}, {}^{\mathrm{c}}f^{\tau}_{\mathrm{j},9}, {}^{\mathrm{c}}f^{\varepsilon}_{\mathrm{j},9}) $$

Its transition function ${}^{c,e}\mathrm{f}_{\mathrm{j},9}$ is:

$$ {}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}+1}_{\mathrm{j,m}}[{}^{\mathrm{B}}_{\mathrm{E}}T_{\mathrm{d}}] = {}^{c,e}f_{\mathrm{j},9,1}({}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}}_{\mathrm{j,m}}) \triangleq {}^{\mathrm{B}}_{\mathrm{Epp}}T_{\mathrm{d}} $$

$$ {}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}+1}_{\mathrm{j,m}}[space] = {}^{c,e}f_{\mathrm{j},9,2}({}^{\mathrm{e}}_{\mathrm{y}}c^{\mathrm{i}}_{\mathrm{j,m}}) \triangleq 0(Jointspace) $$

- Behavior ${}^{\mathrm{c}}\mathrm{B}_{\mathrm{j},10}$, Corresponding to move to place pose in the task space state behavior. It is defined as:

$$ {}^{\mathrm{c}}B_{\mathrm{j},10} \coloneqq {}^{\mathrm{c}}B_{\mathrm{j},10}({}^{c,e}f_{\mathrm{j},10}, {}^{\mathrm{c}}f^{\tau}_{\mathrm{j},10}, {}^{\mathrm{c}}f^{\varepsilon}_{\mathrm{j},10}) $$

Its transition function ${}^{\mathrm{c,e}}\mathrm{f}_{\mathrm{j},10}$ is:

$$\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}+1}[\underset{\mathrm{E}}{\mathrm{B}}T_{\mathrm{d}}] = {}^{\mathrm{c,e}}f_{\mathrm{j},10,1}(\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}}) \triangleq \underset{\mathrm{Ep}}{\mathrm{B}}T_{\mathrm{d}}$$

$$\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}+1}[space] = {}^{\mathrm{c,e}}f_{\mathrm{j},10,2}(\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}}) \triangleq 1(Taskspace)$$

- Behavior ${}^{\mathrm{c}}\mathrm{B}_{\mathrm{j},12}$, Corresponding to move to preplace pose in the task space state behavior. It is defined as:

$${}^{\mathrm{c}}B_{\mathrm{j},12} \coloneqq {}^{\mathrm{c}}B_{\mathrm{j},12}({}^{\mathrm{c,e}}f_{\mathrm{j},12}, {}^{\mathrm{c}}f_{\mathrm{j},12}^{\tau}, {}^{\mathrm{c}}f_{\mathrm{j},12}^{\varepsilon})$$

Its transition function ${}^{\mathrm{c,e}}\mathrm{f}_{\mathrm{j},12}$ is:

$$\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}+1}[\underset{\mathrm{E}}{\mathrm{B}}T_{\mathrm{d}}] = {}^{\mathrm{c,e}}f_{\mathrm{j},12,1}(\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}}) \triangleq \underset{\mathrm{Epp}}{\mathrm{B}}T_{\mathrm{d}}$$

$$\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}+1}[space] = {}^{\mathrm{c,e}}f_{\mathrm{j},12,2}(\underset{\mathrm{y}}{\mathrm{e}}c_{\mathrm{j,m}}^{\mathrm{i}}) \triangleq 1(Taskspace)$$