# Unit IV Navigation, Path Planning and Control Architecture

## 4.6 Cartesian Control Architecture

Parallel manipulators are robotic devices that differ from the more traditional serial robotic manipulators by their kinematic structure. Parallel manipulators are composed of multiple closed kinematic loops.

Typically, these kinematic loops are formed by two or more kinematic chains that connect a moving platform to a base, where one joint in the chain is actuated and the other joints are passive. This kinematic structure allows parallel manipulators to be driven by actuators positioned on or near the base of the manipulator. In contrast, serial manipulators do not have closed kinematic loops and are usually actuated at each joint along the serial linkage. Accordingly, the actuators that are located at each joint along the serial linkage can account for a significant portion of the loading experienced by the manipulator, whereas the links of a parallel manipulator generally need not carry the load of the actuators. This allows the parallel manipulator links to be made lighter than the links of an analogous serial manipulator. The most noticeable interesting features of parallel mechanisms being:

High payload capacity, High throughput movements (high accelerations), High mechanical rigidity, Low moving mass, simple mechanical construction. Actuators can be located on the base.

However, the most noticeable disadvantages being:

- They have smaller workspaces than serial manipulators of similar size.
- Singularities within working volume.
- High coupling between the moving kinematic chains.

***Cartesian Control in robot manipulators:***

In order to understand application of cartesian control in robot manipulators a case of study will be used, which all the concepts were evaluated. In this section we will obtain the forward kinematics of a three degrees of freedom cartesian robot, Figure 4.4; and we will use this information in the following sections.

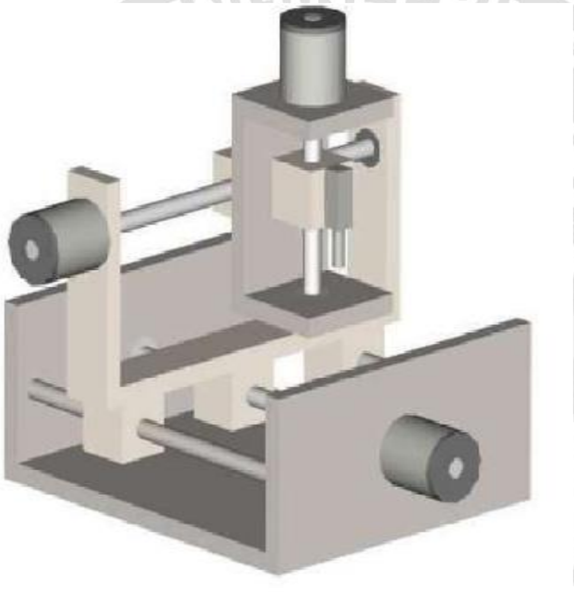

**Fig. 4.4 3 –DOF Cartesian Robot**

As it is observed, translation is the unique movement that realizes this kind of robots, then the forward kinematics are defined as:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} q_1 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ 0 \end{bmatrix} \quad ; \quad \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} . \qquad \text{---(1)}$$

where q1 , q2, q3 are join displacements; and m1 , m2, m3 represent the masses of each link.

We can observe, that in the first vector is contemplated only by the first displacement of value q1 , in the second one considers the movement of translation in q1 and q2 respecting the axis x and y, and finally the complete displacement in third axis described in the last vector, being this representation the robot forward kinematics.

***Jacobian matrix:***

The Jacobian matrix J(q) is a multidimensional form of the derivative. This matrix is used to relate the joint velocity $\dot{q}$ with the cartesian velocity $\dot{x}$, based on this reason we are able to think about Jacobian matrix as mapping velocities in q to those in x:

$$\dot{x} = J(q)\,\dot{q}.$$

***------ (2)***

where $\dot{x}$ is the velocity on cartesian space; $\dot{q}$ is the velocity in joint space; and J(q) is the Jacobian matrix of the system.

One of the most important quantities (for the purpose of analysis) in (2), is the Jacobian matrix J(q). It reveals many properties of a system and can be used for the formulation of motion equations, analysis of special system configurations, static analysis, motion planning, etc. The robot manipulator's Jacobian matrix J(q) is defined as follow:

$$J(q) = \frac{\partial f(q)}{\partial q} = \begin{bmatrix} \frac{\partial f_1(q)}{\partial q_1} & \frac{\partial f_1(q)}{\partial q_2} & \cdots & \frac{\partial f_1(q)}{\partial q_n} \\ \frac{\partial f_2(q)}{\partial q_1} & \frac{\partial f_2(q)}{\partial q_2} & \cdots & \frac{\partial f_2(q)}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(q)}{\partial q_1} & \frac{\partial f_m(q)}{\partial q_2} & \cdots & \frac{\partial f_m(q)}{\partial q_n} \end{bmatrix}$$

----(3)

where f(q) is the relationship of forward kinematics, equation (3); n is the dimension of q; and m is the dimension of x. We are interested about finding what joint velocities $\dot{q}$ result in given (desired) v. Hence, we need to solve a system equations.

***Jacobian matrix of the cartesian robot***

In order to obtain the Jacobian matrix of the three degrees of freedom cartesian robot it is necessary to use the forward kinematics which is defined as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Now, doing the partial derivation of $x$ in reference to $q_1, q_2, q_3$ we have:

----(4)

$$\frac{\partial x}{\partial q_1} = \frac{\partial q_1}{\partial q_1} = \dot{q}_1$$

$$\frac{\partial x}{\partial q_2} = \frac{\partial q_1}{\partial q_2} = 0 \qquad (5)$$

$$\frac{\partial x}{\partial q_3} = \frac{\partial q_1}{\partial q_3} = 0$$

The partial derivation of $y$ in reference to $q_1, q_2, q_3$ are:

$$\frac{\partial y}{\partial q_1} = \frac{\partial q_2}{\partial q_1} = 0$$

$$\frac{\partial y}{\partial q_2} = \frac{\partial q_2}{\partial q_2} = \dot{q}_2 \qquad (6)$$

$$\frac{\partial y}{\partial q_3} = \frac{\partial q_2}{\partial q_3} = 0$$

The partial derivation of $z$ in reference to $q_1, q_2, q_3$, we have:

$$\frac{\partial z}{\partial q_1} = \frac{\partial q_3}{\partial q_1} = 0$$

$$\frac{\partial z}{\partial q_2} = \frac{\partial q_3}{\partial q_2} = 0 \qquad (7)$$

$$\frac{\partial z}{\partial q_3} = \frac{\partial q_3}{\partial q_3} = \dot{q}_3$$

The system $\dot{x} = J(q)\dot{q}$ is described by following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \qquad (8)$$

where the Jacobian matrix elements are defined using the equations (8), (9) and (10): (9)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

## 4.7 Force and Hybrid control Architecture

The basic hybrid control idea is an architectural concept that links the constraints of a task requiring force feedback to the controller design. The transformation form (C) to the joints of the manipulator is such that, for the general case, control of one manipulator joint involves every dimension in (C):

Qi =Qi(Xi,X2 XN) ------- (1)

where: q, = position of/th joint
of manipulator fi, = inverse kinematic function
Xj = position of /'th degree of freedom in {C}

Therefore in hybrid control the actuator drive signal for each joint represents that particular joint's instantaneous contribution to satisfying each positional and each force constraint. The actuator control signal for the /'th joint has N components - one for each force controlled degree of freedom in [C], and one for each position controlled degree of freedom:

$$\tau_i = \sum_{j=1}^{N} \{ \Gamma_{ij}[s_j \Delta f_j] + \psi_{ij}[(1 - s_j)\Delta x_j] \} \qquad (2)$$

where:

$\tau_i$ = torque applied by the $i$th actuator
$\Delta f_j$ = force error in $j$th DOF of $\{C\}$
$\Delta x_j$ = position error in $j$th DOF of $\{C\}$
$\Gamma_{ij}$ and $\psi_{ij}$ = force and position compensation functions, respectively, for the $j$th input and this $i$th output
$s_j$ = component of compliance selection vector.

The compliance selection vector, S, is a binary iV-tuple that specifies which degrees of freedom in [C\are under force control (indicated by Sj = 1), and which are under position control (Sj = 0). (In this paper it is assumed that the number of manipulator joints equals N < 6.)

For example: if S = [0,0, 1,0, l,l]$^{T}$

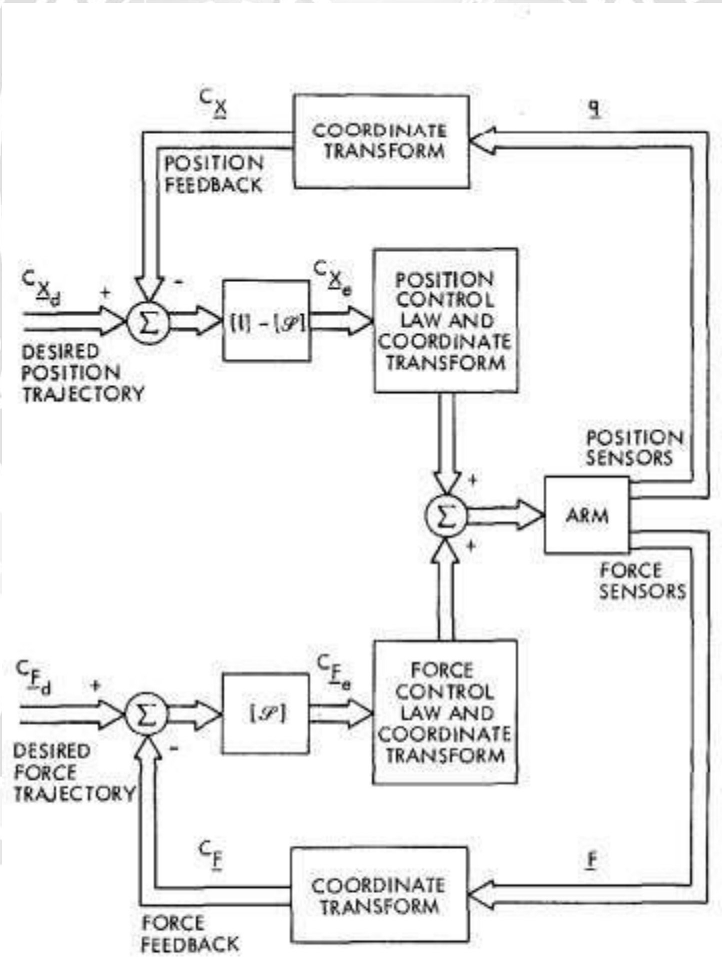


**Fig. 4.5 Conceptual Organisation of Hybrid Cylinder**

then

$$\tau_i = \psi_{i1}(\Delta x_1) + \psi_{i2}(\Delta x_2) + \Gamma_{i3}(\Delta f_3) + \psi_{i4}(\Delta x_4) + \Gamma_{i5}(\Delta f_5) + \Gamma_{i6}(\Delta f_6)$$

Though the total number of active control loops is always N, the type mix will vary as the task geometry and natural constraints change.

constraints change.

We also define the compliance selection matrix [S]:

$$[S] = \text{diag}(S) = \begin{bmatrix} S_1 & & & & 0 \\ & S_2 & & & \\ & & S_3 & & \\ & & & \cdots & \\ 0 & & & & S_N \end{bmatrix}$$

Figure illustrates a hybrid control system that incorporates these ideas. The two complementary sets of feedback loops (upper-position, lower-force), each with its own sensory system and control law, are shown here controlling a common plant, the manipulator. Notice that sensory signals must be transformed from the coordinate system of the transducer, [q] for position and [H] for force, into (C) before errors are found and the compliance selection vector is applied:

$$^{C}\mathbf{X} = \Lambda(\mathbf{q}) \qquad (3a)$$

$$^{C}\mathbf{F} = \left[ {}^{C}_{H}T \right] {}^{H}\mathbf{F} \qquad (3b)$$

where:

$\mathbf{X}$ = position of manipulator hand

$\Lambda$ = kinematic transform from $\{q\}$ to $\{C\}$

$\mathbf{F}$ = force on manipulator hand

$$\left[ {}^{C}_{H}T \right] = \left[ \begin{array}{c:c} \left[ {}^{C}_{H}R \right] & 0 \\ \hdashline \left[ \mathbf{V} \times \right] \left[ {}^{C}_{H}R \right] & \left[ {}^{C}_{H}R \right] \end{array} \right]$$

force transformation matrix from $\{H\}$ to $\{C\}$

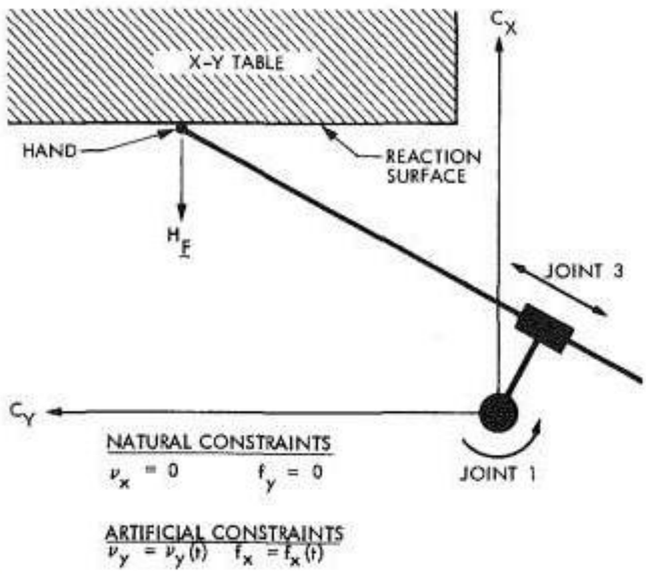


**Fig. 4.6 Physical layout of manipulator and reaction surface used in experiments. Joints 1 and 3 of the Scheinman arm were used to provide motion in plane normal to gravity vector. Hand is in contact with numerically controlled X-Y table.**

In addition to these error-driven control signals, an ideal manipulator trajectory controller, whether controlling position or force, can include feed-forward compensation for the nonlinear dynamics that characterize the manipulator control problem [8], Such signals take into account the configuration dependent inertia and gravity forces, state dependent Coriolis forces, velocity dependent friction forces, and externally generated hand contact forces. Ideally, when a wrist- mounted force sensor is used there should also be adjustments for accelerations of the hand mass present between a wrist sensor and contact surfaces of interest (including mass of hand held objects or tools).

While each degree of freedom in (C) is controlled by only one loop, both sets of loops act cooperatively to control each manipulator joint. This is the central idea of hybrid control. As is usually the case when sensing, control, and actuation each take place in different coordinate systems, the same sensors and actuators participate in each "separate" control loop.

In certain respects, hybrid control is a modification and extension of Paul and Shimano's compliant control. Both approaches employ a task-related coordinate system C, both partition [C] into position controlled and force controlled subspaces, and both given freedom in specifying position and force trajectories. However, because they pair individual force controlled joints with individual force constraints in [Cj on each servo cycle, position and force errors result. These errors are corrected on subsequent cycles by adjusting the position set points differentially. These adjustments are not necessary for the hybrid approach because each joint always contributes to control of force and position.

## 4.8 BEHAVIOUR BASED CONTROL

Behavioural Mapping

### Introduction

For any system we can define a space that includes all the system's states. We call this space the state space. Each point of the space represents a different state of the system. The configuration space is an example of such a space that covers all the geometric position of an n - degrees-of-freedom system in an Euclidean space .

Another example is the phase space in which each point represents the kinematics state of the system. A topological space was used to describe the possibilities of a living system. A subspace of the states' space is used to describe the "permitted" domain of the system, the free space in the case of the configuration space or the viability subspace.

We name this subspace the subspace of existence or the Viability subspace for a system. One can regard the system behaviour as the results of a function (or functions) that the system uses to change its state and the state of the world. By analysing the behaviour of the latter in the existence subspace one can obtain an interesting result. The existence subspace can be divided into zones in which the function(s) is continuous and to zones in which the function is not continuous. Those zones of non continuity can be regarded as zones of decision where the system or its operator should select one of the possibilities, if there are any, or activate another function. When these zones are retracted to points and the continuous zones in between are retracted to lines connecting these points, a graph representation of the existence subspace is obtained.

The advantage of such a representation lies in the control of the system, in continuous zone it is not necessary to consider or calculate any option, only following that zone by the right behaviour and a local optimisation are required. In the next paragraph the mapping of single behaviour is discussed. This theory of behaviour mapping can be generalised to a multi behaviour system. In this case we define the term behaviours' equivalence. Behaviours are defined equivalent if at a given state their effects are topologicaly equal, that is if they change the system situation to an equal topological state.

***Single Behaviour Mapping***

Let us consider a system that exercises only one behaviour that conforms to the existence constraints. Hence from any place in the existence subspace the behaviour will conduct the system to a place inside that subspace. We define as trivial the case when that behaviour is continuous all over the subspace of existence. A more interesting case is when there are points or zones of non continuity and bifurcation of the behaviour in the existence subspace. We can use these zones and points of non continuity as landmarks and describe the existence subspace of a system as graph in which these zones are the nodes. Whenever the behaviour connects one zone to another their representations in the graph are connected. A formal definition is given below.

Let :

W be the topological space that represents the world.

V be a connected subspace of W that represents the subspace of existence for the system. $f:V \rightarrow V$ an application defined over V that represents the system behaviour.

f is continuous at $v \in V$ iff:

(1) f(v) = u, u ∈V

(2) for any neighbourhood of u, Nu, we can find a neighbourhood of v, Nv, such as Nu⊇ f(Nv)

Let d be an open set of non continuous points such as

(1) v ∈d if f (v ) is not continuous

(2) d is a connected set The closure of d cl(d) is composed of continuous points such as that at any neighbourhood of v ∈ cl(d) there is at least one point of d.

Let D = {d1...dn} be the collection of the non continuous sets of V under f. Let n f (v) = f ° f ° f °...° f(v) describes a sequence of n iteration of the behaviour f. The set dj is said to be directly connected to di by f n iff

1. there exists x ∈cl(di) such as n f ∈cl(d j).

2. f (x)... n−1 f (x) are continuous under f We define a path from di to dj as l x x ∈di , n ( f ∈d j) = f(x)... n { f } Let ij p = l(x1)U...Ul(xk) where l(x1)...l(xk) are all the paths from di to dj.

We call p a passage from di to dj . Any d from which there are two or more passages is a bifurcation zone.

V now can be described as a directed graph. The non continuous sets {d} are the nodes of the graph which are connected according to the passages defined above. Now we can plan the future of the system based on the graph. The nodes play quite a significant role, they are the decision points for the system. On a passage the behaviour will lead the system to the same neighbourhood independent of perturbations or deviations on the way.

In the non continuous zones the situation is quite different, any deviation or little perturbation can lead the system to a totally different situation. We call these zones decision zones because a decision can be taken here, a decision that will have quite an influence on the future of the system

***An example of uni-behaviour***

We bring here as an example go-toward behaviour. The behaviour was developed by a team of the Institute for Micro technology of the University of Neuchâtel (IMT). A vision system tracks a reflecting target and the robot moves toward that target. When the target is too close the robot stops. The angle of vision is 90 deg.

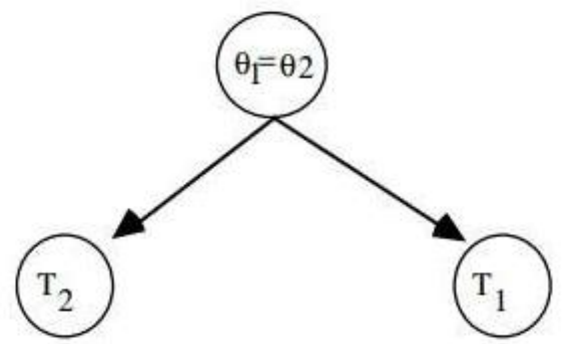


Figure 4.7 The graph of go-toward behaviour with two targets

When there is only one target the non continuity zones are: 1. when the distance from the robot to the target is less than the minimum and the robot stops 2. When the angle between the centre line of the camera and the target is out of the ± 45 deg. range the robot looses the target and stops For a distance bigger than the minimum and within the ± 45 deg. the behaviour is continuous. When a second target is introduced into the scene the situation changes. The robot is programmed to follow the target which is closer to the centre line. The line defined by θ1 = θ 2, where θ1 and θ2 are the angles to the first and the second target respectively, becomes a bifurcation zone. The graph describing the subspace of existence for the robot under the go-toward behaviour is given in figure 4.7

***Multi Behaviours Mappings***

The analysis of the subspace V can be generalized to multi behaviours' mapping. Here we deal not only with bifurcation of a single behaviour. To define bifurcation points for several behaviours first we have to define the term b-equivalence that stands for behaviours' equivalence. We use here the term b-equivalence when different behaviours give the same results when exercised at the same situation.

Two behaviours are considered to be b-equivalent at a certain part of the existence subspace if when exercise at any point of this part they bring the system to the same topological state. Hence any behaviour of the system that follows gives the same results. The points of bifurcation are the points where the behaviours' results separate. These points are used to create a graph description in which the nodes represent the bifurcation points and the branches are defined according to their connection by the behaviours.

in which either at least one of the behaviours is not continuous or where the behaviours are not b- equivalent. The set dj is said to be directly connected to di by f and g iff 1. there exists n, m and x ∈cl(di) such as n f (x) ∈cl(d j) and . mg (x) ∈cl(d j) 2. g and f are continuous and b-equivalent along the path. We can now describe the existence space of the multi behaviours system by a graph. The nodes of the graph represent the bifurcation zones. The branches are according to the direct connectivity defined above.

An example for such a system is a corridors' navigator mobile robot that possesses the three following behaviours:

1 follow-the-centre of the corridor

2. follow-the-right-wall

3. follow- the-left-wall.

As long as the robot is in a corridor the three behaviours are b-equivalent. When the robot arrives to a corridors' intersection the follow-the-centre behaviour becomes non continuous, the robot can follow anyone of the corridors that exit the intersection. The other two behaviours are not equivalent. The follow-the-left-wall will follow the most left corridor while the follow-the-right- wall will follow the most right corridor. Any selection of a behaviour will give different results, as a consequence the corridors' intersection becomes a decision zone.

***Meta Behaviour Mapping***

In this section we generalise the notion of a behaviour by defining the Meta Behaviour, which is the result of looking at the system's actions as being one behaviour even if it results from the composition of several behaviours. In the following paragraph we lay the mathematical foundation for such a definition.

Let $F: V \rightarrow V$ $F : V \rightarrow V$ be an ensemble of behaviours $\{f\}$.

Let $D = \{d\}$ be the finite ensemble of the sets $d$ in $V$ such as:

$x \in d$ iff

(1) there exists at least one $f \in F$ that is not continuous at $x$.

or

(2) there exist at least $f_i$ and $f_j$, $i \neq j$ which are not b-equivalent at $x$.

Let $P = \{p\}$ be the ensemble of passages defined by one or more of the behaviours $\{f\}$.

Let $G_F(V)$ be a graph in which the nodes represent the closed sets $\{d\}$ and the branches represent the passages $\{p\}$ that connect them. Such a graph is a unique representation of $V$ under $F$.

We call $F$ a *meta-behaviour*, a behaviour that consists of all the behaviours $\{f\}$. $F$ is not continuous in $D$ but continuous elsewhere in $V$.

### *4.8.2. Meta-Behaviour and Control*

We can look now at a system that possesses an ensemble of behaviours B that it executes in a working space U. Let assume that U is decomposable to decision zones D and passages P under the behaviours B. Looking at the behaviour of the system within the frame of the graph description of the working space an interesting phenomenon emerges. The system demonstrates a behaviour of following the passages from one decision point to another. This behaviour which we denote meta-behaviour is independent of the behaviour actually executed by the system as long as it follows the passages.

A definition of a trajectory can be given as a sequence of passages to be followed. Following a given path is reduced to selecting at each decision point the behaviour that will follow the next passage. Each of the equivalent behaviours will be adequate.

To navigate, the system should be able to recognise the decision points and to select the appropriate behaviour while in the decision zone. The selection of the behaviour can be made based on local parameters or using a plan. In the latter case the plan is not the classic plan as a program but more like a general frame that gives the system the criteria by which it makes the selection.