3. PATH AND TRAJECTORY PLANNING

3.1. General problems of path and trajectory planning

Industrial robots must have high flexibility to execute different technological operations and work together with human workers. Compared with human flexibility, a robot has a multitude of problems to realize even simple motions in a working space. To move between two space points, different tasks must be solved. The best trajectory must be found, obstacles and collisions must be avoided, other limitations must be considered, and the high efficiency and work productivity must be achieved. To control the robot’s motion normally the previous motion planning is used. The **path planning** is the planning of the whole way from point A to point B, including stopping in defined path points. The path includes several continuous motion trajectories that need the **trajectory planning**. If a path can not be previously planned because of limited previous information, the motion task is named as **path finding**.

Tasks of robot control can be classified in different ways. For example, different path planning strategies can be used in the case of different situations. There are two types of constraints that must be considered in path planning. First, the motion of a robot can be restricted by obstacles and **obstacle constraints** have to be used. On the other hand there can be some kind of constraints for path selection. These constraints are known as **path constraints**.

In the case of obstacle constraints it will be assumed that some points in the robot motion space are occupied and the robot cannot plan the path through these areas. In the case of path constraints, there can be some referred points that must be necessarily passed through.

The following path planning strategies exist:

- path constrained (signed path) off-line or on-line path planning with collision avoidance
- position controlled motion with on-line obstacle identification and collision avoidance (without path constraints, i.e. path signs)
- path constrained off-line path planning or on-line pass through the signed path (collisions are possible)
- position controlled motion without obstacle identification (collisions are possible)

These strategies can be used to solve path planning tasks in robotics in most of the cases. The **position controlled motion** is the motion along interpolated trajectories between signed and referred path points. The **signed path** is the path having regular defined points that must be unconditionally passed through.

The main path planning tasks for a robot are as follows:

- grasping and releasing objects
- moving from place to place
- following previously specified paths
- following moving objects
- working with other manipulators
- exerting forces (i.e. pushing, pulling and holding)
- exerting torques
- collecting data
- using tools

Robots are subject to all of the constraints of mechanics. In the case of manipulators with many joints (prismatic or revolute), the physical limits of motion become evident. For the best solution, the limits of joint and actuator positions, velocity, acceleration, and jerk must be considered. The physical nature of the device also means that there are dimensions which must be considered, thus kinematics and collision avoidance come into play. When a robot makes any move, it expends energy to accelerate, hold and brake. This also means that the energy efficiency of the manipulator should be optimized by reducing unnecessary expenditures of energy. Most importantly, if robots are to be cost effective, then their speed is of concern. In a high production situation, a cycle time that is 10% faster could save millions of dollars. Thus, the time of path traversal can most often be the most important path planning factor.

Essential performances of a robot:

- time for path traversal
- velocity of manipulator links or joints
- stored energy
- actuator forces
- proximity to obstacles

Mechanical constraints of a manipulator:

- joint positions, velocities, accelerations and jerks
- actuator forces and motion dynamics
- kinematics (including singularities)
- collisions with obstacles
- time when moving obstacles are involved

General requirements, evaluation criteria:

- dimensions of space (2D, 2.5D, 3D)
- collision avoidance (none, contact detection, proximity calculation)
- multilink manipulators
- rotations of payload or mobile robot
- moving workspace obstacles
- multi robot coordination
- degree of automation (automated or manual path planning)

Evaluation criteria of information setup:

- information source (knowledge based, sensor based)
- world modeling (world model)

Evaluation criteria of the control method:

- path planning strategies for information passing (e.g. hierarchical)
- path planning methods (algorithms used for path planning)
- internal representations of path or trajectory
- minimization (which costs are minimized?)
- limits (which limits are considered?)
- solution type (robot, joint space, Cartesian space, straight line, via points with rotations, using of splines, etc.)

Evaluation criteria of implementation:

- execution time, machine type, programming language
- testing (what are the experimental results?)

The optimization of path planners

To aid the description of the path planning problem, a generalized statement of the optimization criteria will be given. This will be presented for both the measure of performance and constraints. The first most important measure of performance is time for the path. To find this and other factors, a number of relations will be derived. First assume that the path is made up of a number of discrete segments (trajectories). These segments are linked together to form the path of motion. Motion along the path will then have a few characteristics, and these provide the basis for some equations.

More information about path planners can be found on website:
http://claymore.engineer.gvsu.edu/~jackh/eod/mechtron/mechtron-453.html

3.2. Obstacles and collision detection

A 2D problem is relatively simple and good solutions already exist for finding paths in this representation. This 2.5D problem is also within the grasp of current problem solving routines.

Normally in the working space of a robot other machines, different constructions or devices exist. These can be considered as obstacles that have different dimensionality. The obstacles make programming of a robot more complicated. If the robot has planar motion and two dimension obstacles exist, the two-dimensional path planning is used. Generally, the obstacles can be classified as follows (see Fig. 4.1).

2D –	Planar motion around obstacles is a relatively simple task, and good solutions already exist for finding paths in this case.
2,5D –	Planar motion of a robot considering the height of obstacles is a 2.5D problem for path planners.
3D –	Motion through the three dimensional openings is the 3D problem for path planners (considering openings and robot dimensions is needed)

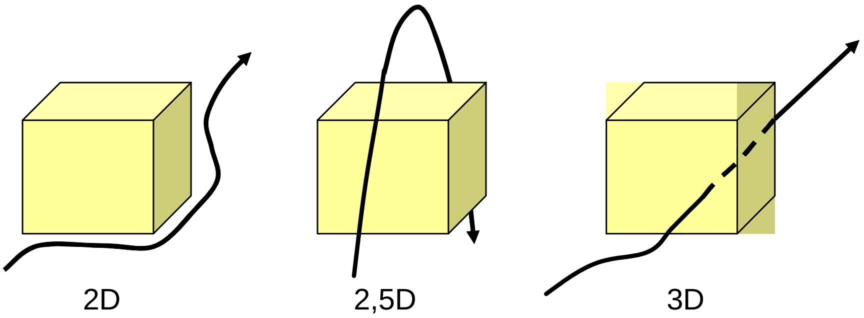


Figure 3.1 Dimensionality of obstacles

Obstacles and collision avoidance in 2D and 2.5D space

A wellknown problem of collision avoidance is moving high dimensional objects in rubbish environment. This is known as the piano mover's problem in a little apartment if the piano cannot be lifted up.

In this case the obstacles are considered as infinitely high. The method is useful if the manipulator of the robot is operating in clear workspace and fulfils picking and placing type operations. Path finding problem in that kind of **2D** space can be easily solved.

If the height of the obstacle is known, then the robot operates in 2.5D space. The problem of avoiding collision with obstacles can be solved in different ways. First, all pick and place type operations may be organized above obstacles and all moving manipulator's links of the robot must be also above the obstacles. Secondly, different algorithms may be found for passing of obstacles. This first method is normally used by people if something is needed to be transferred in a classroom with many tables. It is easier to transfer an object above the tables not to find the way how to pass them.

Obstacles and collision avoidance in 3D space

Sometimes a 3D image can be substituted with a 2D image and the 2D method of path finding may be used. If a possibility to find a path in a 2D space exists, then it may exist also in a 3D space. Using of 2D images simplifies the solution of the path finding problem.

The image of moving objects can also be simplified and simple boxes or cylinders may be used instead of complex form objects. To avoid collisions with moving objects, additional free space must be considered around these objects.

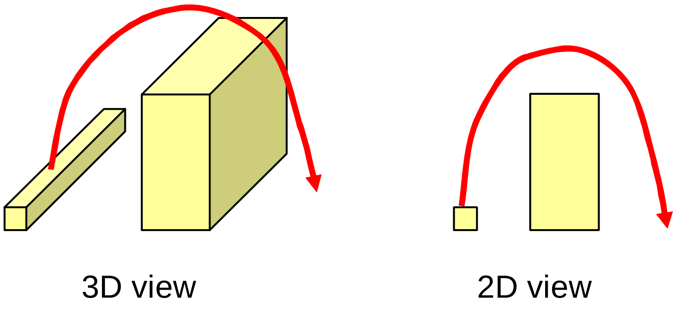
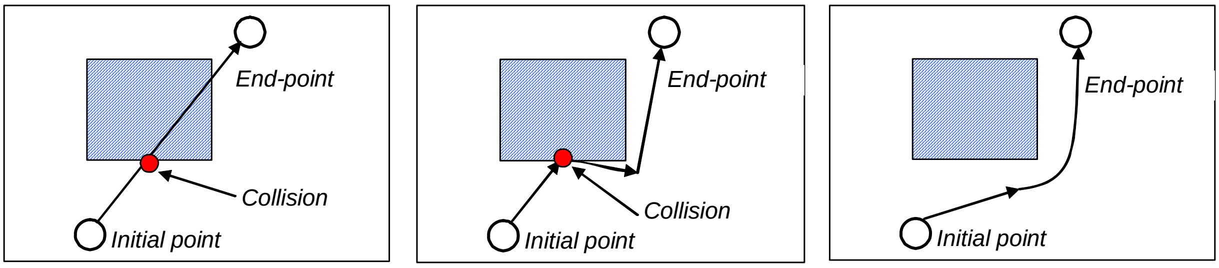


Figure 3.2 Collision avoidance in 3D space or and using 2D view

Collision detection and collision avoidance
Collision detection is the most important factor of Path Planning. Without automatic collision avoidance, the robotic work cell must be engineered to be collision free, or sub-optimal paths must be chosen by a human programmer.

Local collision detection is important when moving through an unknown or uncertain environment. These allow for feedback to the planner, for halting paths which contain collisions. Global Collision Avoidance may be done for planning paths which should avoid objects by a certain margin of safety.



No collision avoidance

Local collision avoidance

Global collision avoidance

Figure 3.3 Collision avoidance

The **number of degrees of freedom** is also important in robot applications. If a manipulator has 6 degrees of freedom, then it can obtain any position or orientation in space. Some specific cases of problems require only 3 or 4 degrees of freedom. This can be a great time saver.

When an **environment is cluttered,** then it may be desirable to have a higher number of degrees of freedom than six, so that the redundancy of the robot can move through the environment. The complexity of most routines increases exponentially with the number of degrees of freedom, thus it is best to match the manipulator degrees of freedom to the complexity of the task to be done.

One assumption that helps reduce the problem complexity is the approximation of **motion in a single plane**. The net result of this effort is that the robot is reduced to 2 or 3 degrees of freedom. The gripper, tool or payload may also be neglected, or fixed, and thus the degrees of freedom are reduced.

A second approach is to approximate the volume of the links swept out over a small volume in space. This volume is then checked against obstacles for collisions. A payload on a manipulator may sometimes be approximated as part of the robot if it is small, or it is symmetrical. This means that the number of degrees of freedom for a manipulator may be reduced, and thus the problem is simplified in some cases.

Multilink manipulators have a variety of configurations. For different configurations different coordinate systems exist in which the best path planning solutions could be achieved. For example

- cartesian (i.e. x, y, z motions)
- cylindrical

- spherical (Stanford manipulator)
- vertically articulated or Revolute (like human arm)
- horizontally articulated (SCARA)

The various robot configurations are fundamentally different. Many approaches have tried to create general solutions for all configurations, or alternate solutions for different specific manipulators. The fastest solutions are the ones which have been made manipulator specific. With a manipulator it is also possible to describe motions in both Joint Space (Manipulator Space) and Cartesian Space (Task Space). There are approaches which use one or both of these.

Rotation problems

Rotations can be also a problem for some path planners. It can be difficult to rotate during motion, thus some will not rotate, some will rotate only at certain points, and some will rotate along a complete path (Fig. 3.4).

- no rotation
- rotation at discrete points
- continuous rotation

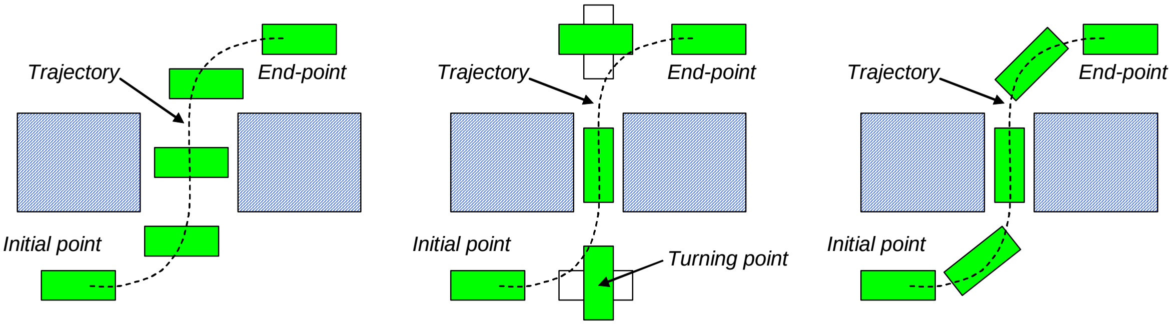
The best scenario is when rotations may be performed to avoid collisions.

Motion of obstacles

Motion of obstacles can cause significant path planning problems. Motion occurs in the form of rotation and translation. In most cases an obstacle in the environment may be categorized into motion categories:

- static (un-moving),
- deterministic (has predictable occurrence and positions), and
- random (Freely moving, with no regular occurrence).

All of these are of interest because most parts fixed in a workcell are static, workpieces from feeders and conveyors are deterministic, and human intruders are random. Random obstacle motion usually occurs so quickly that path planning may only be able to escape the path of the obstacle, not compensate it. In the case of random moving obstacles a robot must have sensors for the detection of obstacles to avoid collisions.



Parallel motion   Turning in definite points   Continuous turning

Figure 3.4 Problems of robot motion if turning is needed to avoid collision

Coordination of two or more robots

If working envelopes of robots intersect, then these robots have common working space and measures must be applied to avoid collisions between different manipulators. The solution of this problem may be a common operation strategy of two or more robots. To realize this common strategy, the common memory of status vector or status link is needed.

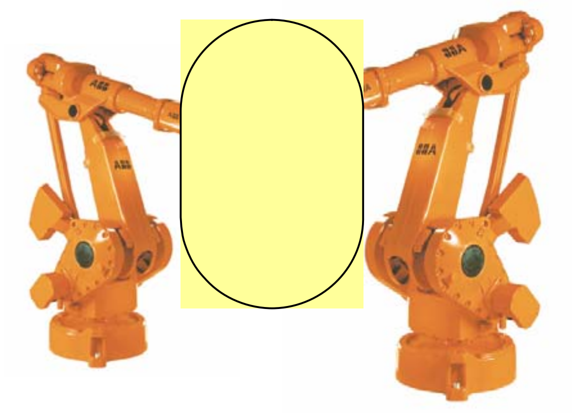


Figure 3.5 The common working envelope of two robots

To work together with another robot, the first robot must have information about its path planning strategy in the common working envelope. In the case of common path planning, first the trajectory of the second robot that seems as the moving obstacle for the first robot will be realized. Then the robot plans its own motion. If the working space of the robot is organized and fully determined (no occasional events), then more simplified path planning algorithms can be used. Generally, if two robots have a common working envelope, then the following strategies may be used:

- coordination of different motions
- common trajectory memory
- common path planning strategy considering both robots and trajectories

Common work organizion for two or more robots is especially complicated if robots have different or opposite destination functions, e.g. in the case of robots competition.

3.3. Environment identification and modelling

Environment models
To solve path planning tasks, the surroundings of the robot environment must be adequately described. The description of the environment surrounding a robot is named also the world model of a robot. This model includes obstacles that must be considered when planning the path for robot motion. Normally all obstacles are considered to be solid and rigid. The conclusion is that no resilient deformations exist.

The environment and objects can be described by the use of different images (Fig 3.12), e.g. matrixes, multi matrixes, rectangles, polygons or the special language named **Constructive Solid Geometry** (CSG). The language of constructive geometry is based on the logic description of different images and their mutual position in the environment. In the case of 3D space the polyhedras, analytically described surfaces, matrixes, oct-trees etc. are used. Each method of description defines limits for image selection. The selected method can be used together with CAD systems.

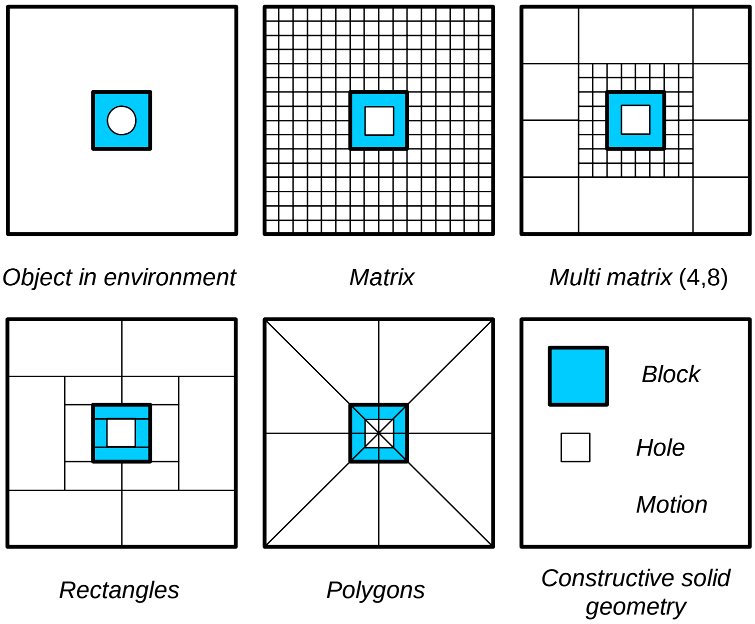


Figure 3.12 Description of the environment surrounding a robot

The most common method of representing objects (in all dimensions) is with convex polygons. These are ideal when working with flat surfaces in the real world. Curved surfaces use flat polygons to approximate their surfaces. One factor that makes the polygons an excellent representation is that if a point is found to lie outside one wall of a polygon, then it may be declared to be outside the entire polygon. Most methods do not allow for concave polygons, because they are much more difficult to deal with in computation. The way to overcome this is to use overlapping convex polygons, to represent a concave polygon. These types of representations can typically be derived from most CAD systems. This form allows easy use of existing facilities.

For the **modeling of obstacles,** the imaginary potential field near the bounders of obstacles can be used. For the description of a potential field, the two dimensional Laplacian of Gaussean (*2D Laplacian of Gaussian*) can be used. The 3D image of Laplacian of Gaussean is shown in Fig. 3.13. The form of functions image is similar to the form of a Mexican cap. In this case we can say that the robot's navigation in the environment having obstacles can be organized according to a Mexican cap. This function (Laplacian of Gaussean) becomes popular for robots that have artificial sight. Generally, also other functions are used to describe obstacles in the robot environment.

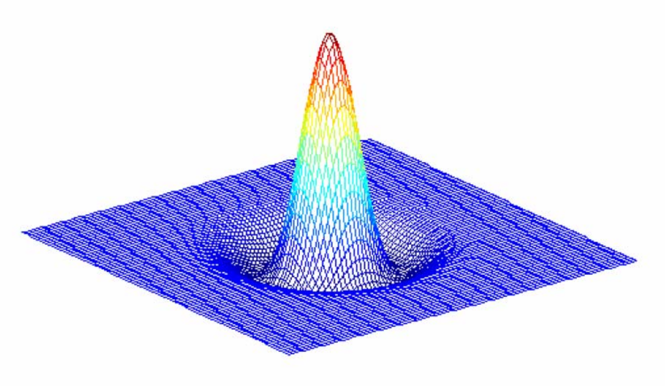


Figure 3.13 A 3D image described by the Laplacian of Gaussian

In the case of several obstacles near each other the composite emergency function as the superposition of obstacles emergency fields can be composed. The form of the emergency field considers the dimensions of the robot and conditions for passing an obstacle (rotations). If many obstacles are near each other in a row, then they form a canyon for robot navigation.

The planned path near the obstacles must consider the robot and its load dimensions and also conditions for turning. Figure 4.14 shows the path planning using the Laplacian of Gaussean or the Mexican cap. The experiments were made by scientist James Mentez from the University of Virginia and the photos are shown in the Internet.

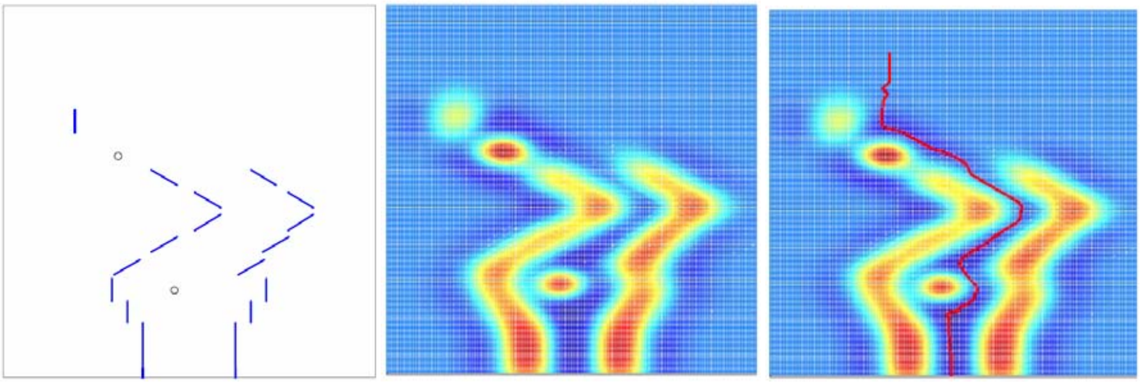


Figure 3.14 An example of robot navigation if obstacles are described using *The Laplacian of Gaussian*

3.4. Automated path planning

Path and trajectory

A path will refer to the complete route traced from the start to the goal end point. The path is made up of a number of segments and each of these path segments is continuous (no stop points). Another name for a path segment could be a trajectory. This is significant, when considering a trajectory planner, which basically chooses a locally optimal direction, as opposed to a complete path. Only some path planners use trajectory based planning, which is easier and faster to compute, but generally produces sub-optimal paths.

A path or a trajectory of robot can be planned by an operator using special software for a robot (e.g. virtual robotics software) or by using programming by teaching. The path or trajectory needed will be programmed during a special teaching process when several paths or trajectory points will be stored in robot memory.

These methods are easily used if industrial robots will be programmed. Normally these robots are working in the same mode for several weeks or months. The problem happens if it is needed to reprogram a robot every day and a lot of time must be spent on programming. In this case the best solution is **automated path planning**.

The information source that will be used is the most important to select a method of path planning.

The environment can be identified previously before the start to the path by environment mapping or during the path through going by obstacle detection. Consequently, two general ways for path planning exist:

- Collision Detection and Local or Trajectory Path Planners that are using information about collision detection
- Obstacle Information Global Path Planners that are using information about previously detected obstacles

Path planners listed in website
http://claymore.engineer.gvsu.edu/~jackh/eod/mechtron/mechtron-453.html
are as follows:

- knowledge-based simple path planner
- knowledge-based hybrid path planner
- sensor-based path planner
- static knowledge- and sensor-based hierarchical path planner
- dynamic knowledge- and sensor-based path planner
- path planner based on *off-line programming*
- path planner based on *on-line programming*

Knowledge based path planning

It is much easier to solve a problem if all the information is available at the beginning of the solution. For a robot the paths can be planned before their execution if some knowledge of the environment is known. This is strictly a **"blind"** strategy that trusts the knowledge of the environment provided. Planning paths before execution allow efforts to get a shorter path time, more efficient dynamics, and absolute collision avoidance. When working in this mode knowledge known before is used. Different techniques are available to solve a variety of problems, when given the a priori information. Some of the knowledge which we use for a priori path planning may come from vision systems, engineering specifications, or CAD programs.

Prior knowledge may be applicable to moving objects if they have a predictable frequency or motion. This may not be used for unpredictable or **random motion** if there is no detection method allowed. Prior knowledge may be derived from the results of modelling or with the help of high level sensors. These sensors are like laser scanners or video systems. These sensors are slow and typically drive a World modeler in an off-line programmer. Video system is the most desired information collector for robotics in the future. Some of these sensors require knowledge from the world modeler for object recognition purposes. In general, these sensors are slower because of their need to interpret low level data, before providing high level descriptions.

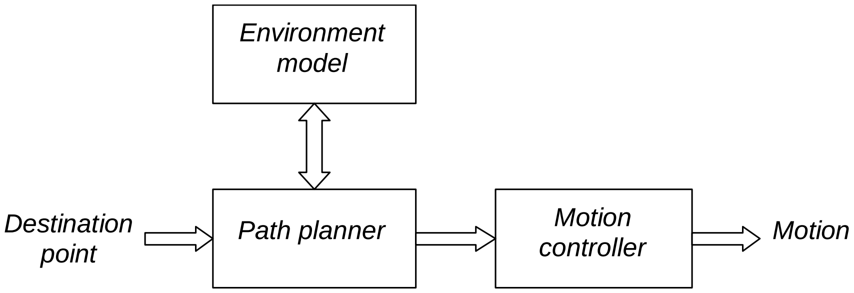


Figure 4.6 The control of robot's motion using knowledge-based path planning

Knowledge-based simple path planner

A simple path planner uses environmental information (incl. position coordinate values) for motion start and end points. Then using an algorithmic process the trajectory via points between the start and end point will be calculated, segments of trajectory will be determined by mathematical description in the defined coordinate system. For example, in the base coordinates of a manipulator or in the world coordinates of a robot system.

Knowledge-based hybrid path planner

If knowledge-based hybrid path planner is used, first a number of possible path variants will be determined. Then the optimal path variant from the possible variants will be selected. This method is more complex than other methods, but gives more chances to find the best path for a robot.

Sensor-based path planning

In this case (Fig. 3.7) information is not available when we begin to solve a problem of path planning. Thus we must solve the problem in stages as the information from the sensors (known after knowledge) becomes available. Sensor-based planning is an indispensable function when environments change with time, are unknown, or there are inaccuracies in the robotic equipment. Subsequent knowledge may be used to find the next trajectory in a path (by collecting information about the outcome of the previous trajectory) or even be used strictly to guide the robot in a random sense when exploring an environment. Sensors that will be used may be very different: from simple contact switches and tactile sensors up to complicated video systems.

These sensors will typically detect various expected conditions. The sensors can give a signal when contact is made with obstacles, or measure a force being applied. When sensors are used in a feedback loop, they may provide actual joint position for a position control algorithm. High level sensors also have the ability to provide low level data and may be used to detect events.

The amount of knowledge which a path planner has may be very limited. If the robot has no previous knowledge of the environment, then information must be gathered while the robot is in motion. Trajectory planners rely on feedback for finding new trajectories and detecting poor results. Contact or distance sensors are used to detect an obstacle and the manipulator trajectory is altered to avoid collision. This method will typically guarantee a solution (if it exists or if it does not encounter a blind alley), but at a much higher time cost, and a longer path. The collection of current data becomes critical when dealing with moving obstacles that do not have a periodic cycle.

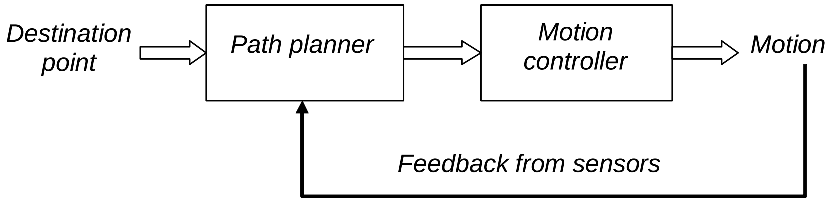


Figure 3.7 The control of robot’s motion using sensor-based path planning

Combined path planning

Advanced robots are using combined path planning methods based on the use of knowledge- and sensor-based information. Part of information is gathered before the path planning. During the path through passing they check this information using sensor signals. Combining of two path planning principles gives the best result (Fig. 4.8).

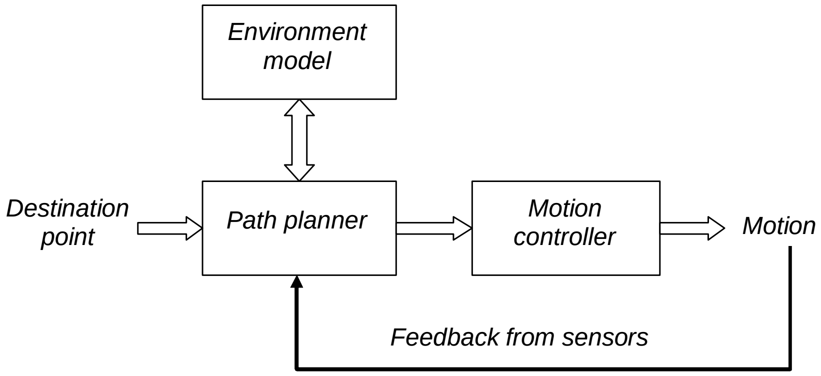


Figure 3.8 The control of robot’s motion using combined knowledge-based and sensor-based path planning

Hierarchical path planning

If the best of both controllers is desired in a single system, it is possible to use a high level prior knowledge planner to produce rough paths, and then use a low level sensor-based planner when executing the path. This would make the planner able to deal with a complex unexpected situation. This also has the ability to do rough path planning in the prior knowledge level, and let the subsequent level to smooth the corners.

Dynamic path planning

Dynamical path planners are a combination from knowledge- and sensor-based path planners. The knowledge-based path planner could plan a path with limited or inaccurate information. If during the execution of this path, a sensor detects a collision, the knowledge-based path planner is informed, and it updates its world or **environment model**, and then finds a new path.

The dynamic Planner is characterized by separate **path planning** and **path execution** modules, in which the execution module may give feedback to the planning module (Fig. 3.10). This is definitely a preferred path planner for truly intelligent robotic systems. Some dynamic planners have been suggested which would allow motion on a path, while the path is still being planned, to overcome the path planning bottle neck of computation time.

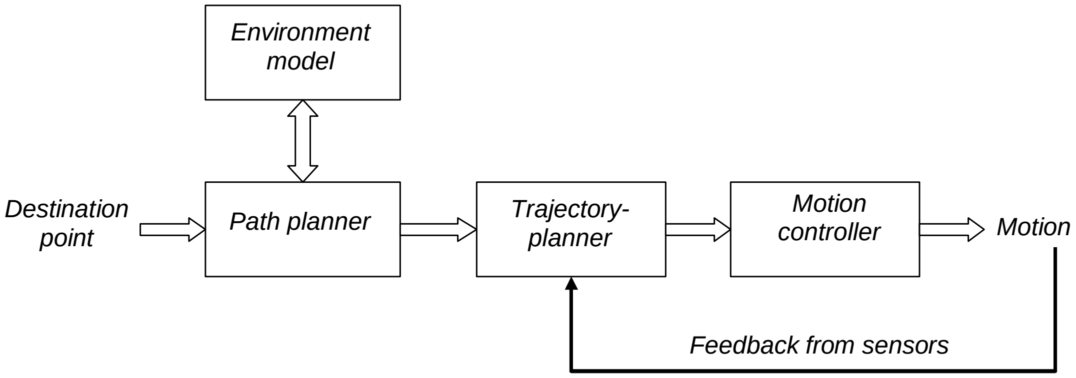


Figure 4.9 Hierarchical path planning

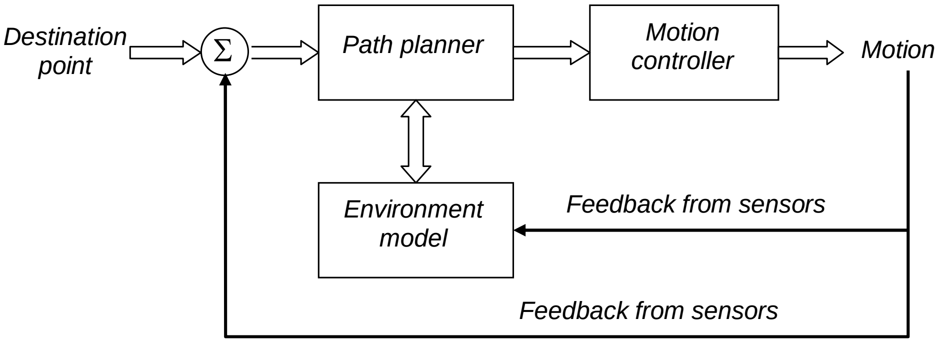


Figure 3.10 Dynamical path planning

The comparison of knowledge- and sensor-based path planning methods is illustrated in Fig. 3.11. Using knowledge-based path planning, the characteristic points of the path that must be passed through during path finding are determined. The sensor information is used to correct the motion if an obstacle appears near the robot and to avoid the collision with the obstacle.

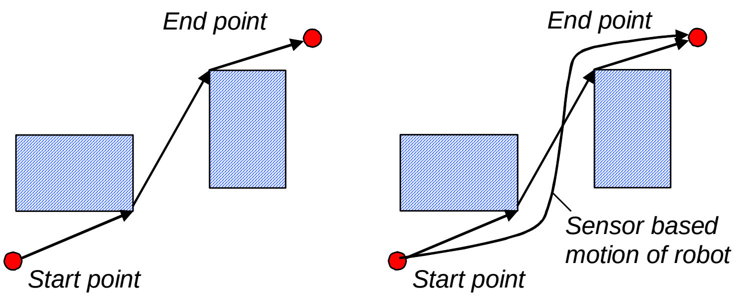


Figure 3.11 Comparison of knowledge- and sensor-based path planning methods

On-line programming methods
These methods for optimal path planning include: methods of mapped space, field methods, gradient methods, etc. Path optimization can be done in a global or local space of a robot. The local path planning methods try to avoid collisions. Global path planning methods try to avoid extra long path and non-effective passing of obstacles on the path. An optimal solution will be calculated using mathematical methods. For example, an optimal solution can be found by using the calculus of variations and dynamic programming.

Method of mapped space
First, the room map will be composed, then optimal path can be found.

Field methods
Field methods are based on the evaluation of collision danger, calculating potential value of danger by special mathematical functions (like the Mexican cap). When path planning occurs, the potential value will be given also to the start and end point of motion.

Gradient method
The gradient method is very similar to field methods. The direction of motion will be calculated using information about the maximum value of the field gradient.

3.5. Methods of path evaluation

The selected path planning method in some cases may have very good properties, but in other cases it can be useless. As the result of theoretical investigations, numerous path planning strategies have been found, but many of them may not have been practically tested. Consequently, the standardization of testing methods of path planning algorithms will be needed.

The criteria most essential for path planning as follows:

- purpose of the manipulator (e.g. 2D mobile robots or 3D manipulators)
- realization of the trajectory
- time of computation (solution calculation)
- time for passing the path

- path longitude
- maximum forces or maximum torques
- energy consumption

Methods of evaluation of path planning

The ultimate path planning test could be a **needle through a hole**. In this case, a box with a hole in it could be held at an odd angle in space (Fig. 3.12). The cylinder could be picked up off the ground and inserted into the hole. The manipulator could then force the cylinder through the hole as far as possible, and then remove the cylinder from the other side and return it to its original position. This could also be approached as a single path or as many as twelve separate paths.

1. Move the arm near the cylinder.
2. Move to and grasp the cylinder.
3. Move near the hole.
4. Insert the peg in the hole and release.
5. Move the arm away from the cylinder.
6. Move the arm near the other side of the cylinder.
7. Grasp the cylinder.
8. Remove the cylinder from the hole.
9. Move the peg the near original position.
10. Place the cylinder on the ground and release.
11. Move away from the cylinder.
12. Move the arm to start position.

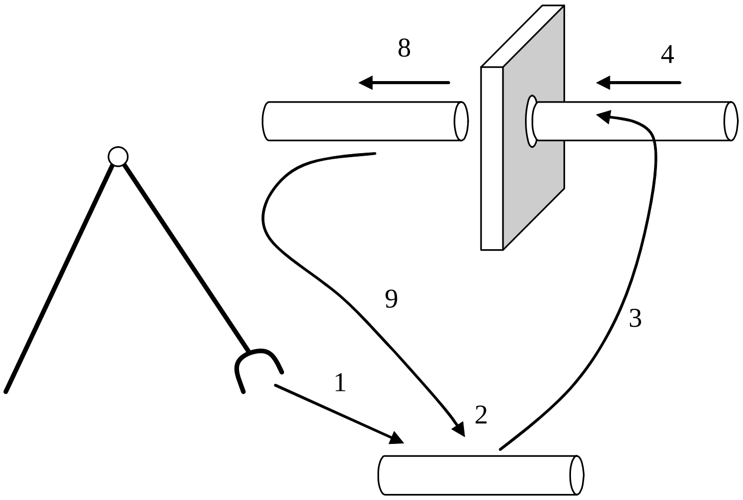


Figure 3.12 Evaluation of robot’s path planning using a test needle through the hole

3.6. Planning of manipulator motion

When planning and programming the motion of the manipulator, several **limits of motion** can exist. In the case of linear motion and straight line trajectory, the programmed trajectory can not go out of the working envelope (Fig 3.13, *a*), because that motion is not possible. In another case (Fig. 3.13, *b*), the change of the pose may be needed to reach the end point of the trajectory. It is not possible to realize the motion because the change of the pose during continuous motion along a straight line trajectory is not possible. In some cases inverse kinematics tasks may be complicated to solve, because more than one solution exists. It means that the manipulator gripper or tool can reach the destination point with the help of different configurations (poses) of the manipulator. When planning the motion of the manipulator it must be considered that a mathematically described motion in some cases may not be realized without setting of additional conditions. That kind of situations are named manipulator singularities.

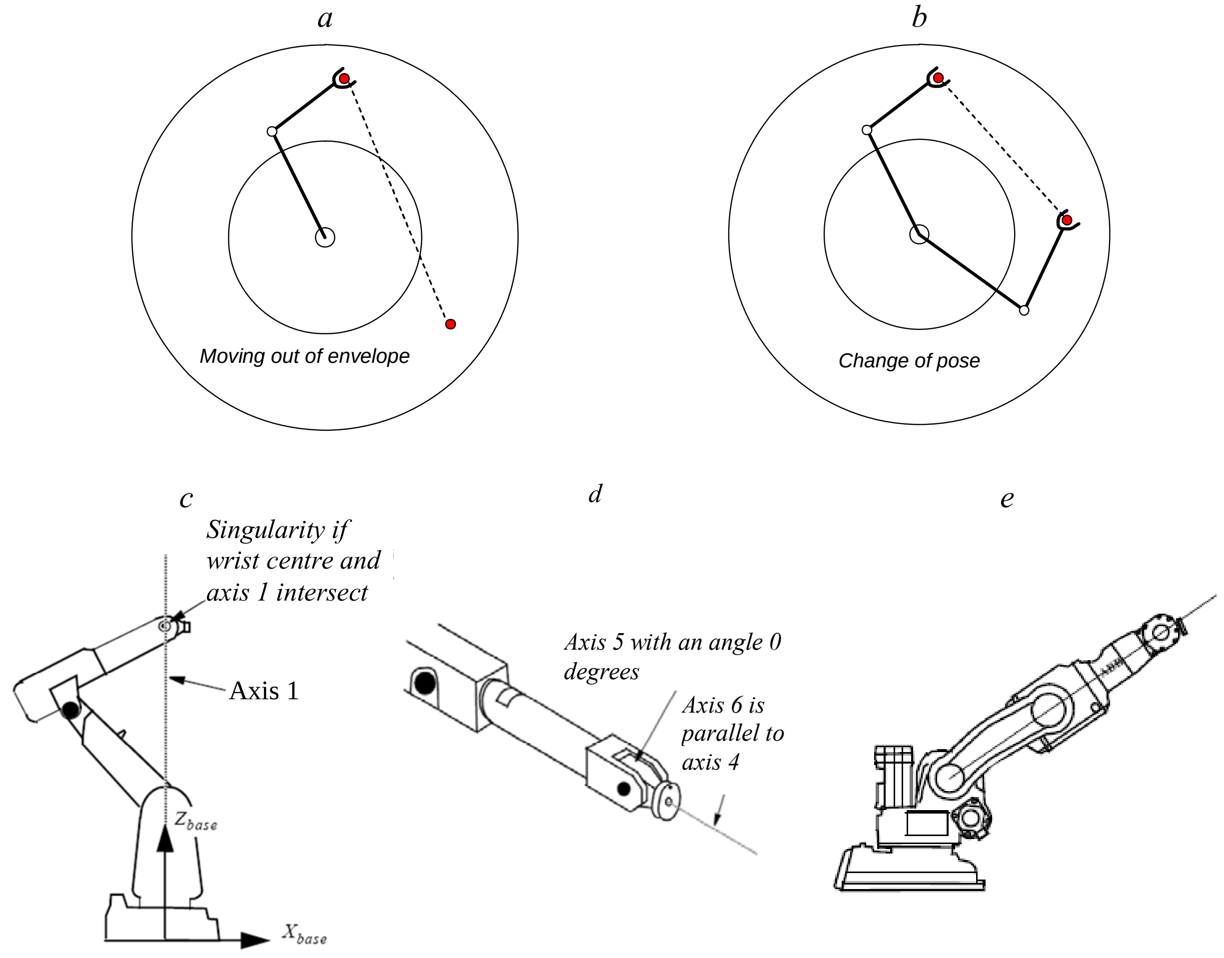


Figure 3.13 Limitations for manipulator motion and singularities

Singularities

Some points in a robot's working envelope are very special, because manipulator links and the gripper or tool can reach these points having an indefinite number of different poses. These points are named singularity points or singularities.

Generally, a robot has two types of singularities - **arm and wrist singularities**. For example, robot IRB 140 from ABB has arm singularities in the case of configurations if the wrist centre point (intersection point of axes 4, 5 and 6) is located on the rotation axis of manipulator's first link (Fig. 4.13, *c*).

Wrist singularities happen in all wrist poses when axes 4 and 6 are on the same line, i.e. rotation angle around axis 5 is equal to zero (Fig. 4.13, *d*).

Robot IRB140 has also the third type singularity when the wrist centre point and arm links (2 and 3) centre points are located on one straight line (Fig. 4.13, *e*).

Because of singularities there are additional problems when solving the inverse kinematics task of the manipulator. In some cases there may be an indefinite number of solutions. In other cases the calculated velocities and forces may have indefinite values and that kind of motion cannot be realized.

3.7. Motion diagrams

Motion diagram represents graphically described time depending functions of changing position, velocity, acceleration and jerk of manipulator link, gripper or tool. Theoretically these functions may have different forms. The best way to describe motion diagrams using mathematical functions is using splines, i.e. curves composed from several polynomials. manipulator motion may be described in different coordinates, e.g in joint coordinates, base coordinates or world coordinates. Base and world coordinates are normally Cartesian coordinates. In some cases the base coordinates may be also polar (cylindrical or spherical) or angular coordinates. Motion will be planned in the world Cartesian coordinates (where also the robot's work is described), but the manipulator will realize the motion in joint coordinates.

It is known that velocity is the derivative of the position function, acceleration is the derivative of the velocity function and jerk is the derivative of the acceleration function. Because of this, to describe motion along the trajectory, the following equations (3.1…3.4) consisting of polynomial are used. Polynomials are the best functions because their derivatives can be very easily found.

$$s(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \tag{3.1}$$

$$v(t) = \frac{ds}{dt} = c_1 + 2c_2 t + 3c_3 t^2 \tag{3.2}$$

$$a(t) = \frac{dv}{dt} = 2c_2 + 6c_3 t \tag{3.3}$$

$$j(t) = \frac{da}{dt} = 6c_3 \tag{3.4}$$

where coefficients $c_0 \ldots c_3$ define the character of motion.

The graphical form of motion (position, velocity, acceleration and jerk) description is named the motion diagram (Fig. 4.14). Motion is needed to be planned for the robot after the program instruction is read from program memory and before the drives of manipulator start motion. Multiple limits are considered on trajectory planning (travel time, maximal values of velocity and acceleration and jerk. The maximal velocity limit can depend on technology or emergency conditions. Higher velocity is more dangerous for people working in the same room with a robot. Higher acceleration and deceleration values need drive motors with higher output power and manipulator construction standing higher forces and torques. Jerk value can be limited by smooth acceleration and deceleration. The simple motion diagram described by one third order polynomial of position, second order polynomials of velocity and linear acceleration (deceleration) function is shown in Fig. 3.14.

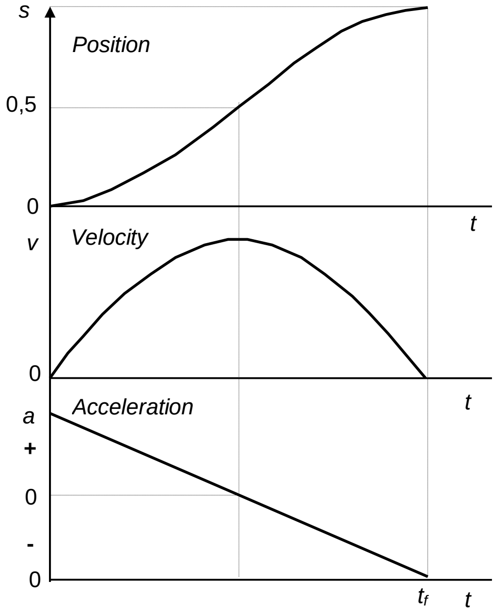


Figure 3.14 Motion diagram if position is described by the third order polynomial

The derivative of acceleration is a jerk that can be sensed by people if the lifting machine is used. A jerk can be sensed by all objects that have non-stable mass centre (depending on the moving of liquid or component masses). The jerk limits are obligatory in the case of machines that will be used for people transport or transportation of jerk sensitive objects (e.g. filled with liquid vessels).

In simple cases the motion diagram of the trajectory is described by simple functions. The velocity diagram has the form of triangle or trapezoid (Fig. 3.15). The position function can be composed from the second order polynomials. Generally, the function composed piece by piece from polynomials is named a **spline**. The motion diagram described by splines is shown in Fig. 3.15.

The motion diagram shown in Fig. 3.15 left side describes the fastest motion between two points when the velocity diagram has the form of a triangle. Motion starts with maximal acceleration and stops after maximal deceleration. Maximal values of acceleration and deceleration are defined by the maximal power of drive. In this case there is no constant speed motion. On the right side diagram, the velocity curve has the form of a trapezium. After initial acceleration, the manipulator moves with a constant speed. Because the jerk is not limited (has theoretically indefinite value) the position function can be described by the spline consisting of the second order polynomials.

Reciprocal motion is the cyclic forward and back-forward motion between points A and B. During one cycle of motion based on the velocity and acceleration diagrams of triangular, trapezoidal or rectangular form, jerks of indefinite value occur. That kind of diagram is typical of most of industrial machines such as robots, cranes, machine tools etc.

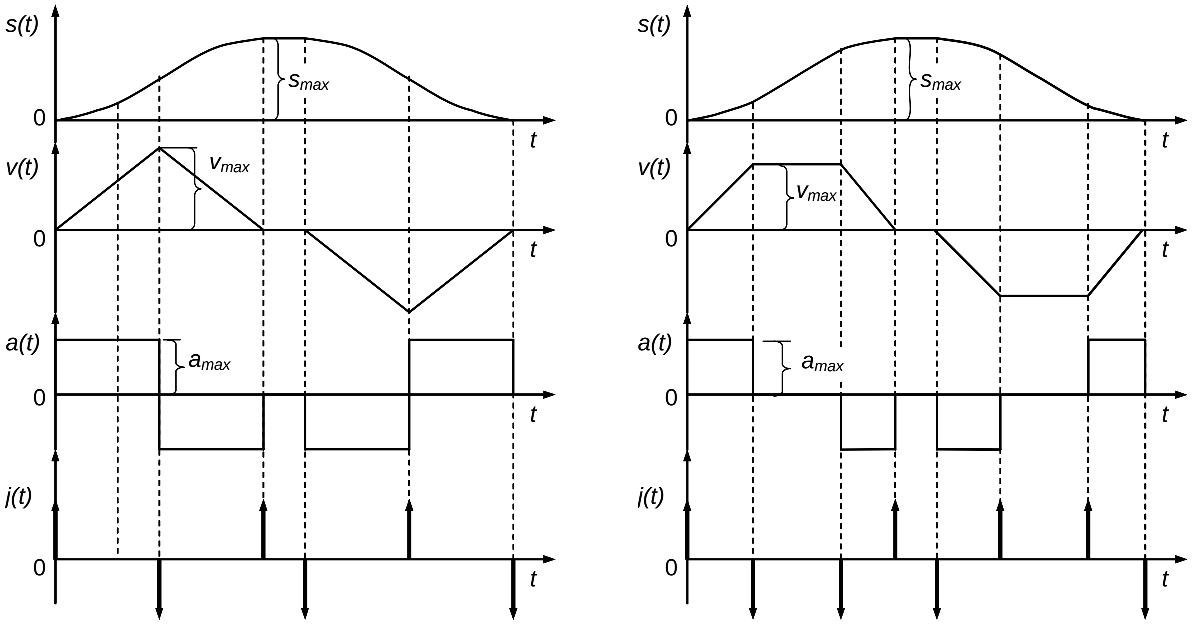


Figure 3.15 Planning of trajectory for triangular and trapezoidal velocity diagrams

The distance s_{max} between two space points A and B will be calculated as:

$$s_{\max} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}, \tag{3.5}$$

where points A and B are defined as:

$$A = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix}, \qquad B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}. \tag{3.6}$$

Equations (4.1…4.6) can be used for trajectory planning but we can also solve the problems of optimal motion considering the limits of maximal acceleration if the force or torque of drive is constrained or limit of maximal energy consumption if mobile robots have accumulators with limited stored energy or an autonomous robot must pass through long distance without storage batteries.

Continuity of motion

Position and velocity of a manipulator can not change instantaneously and consequently the position and velocity functions must be continuous. The trajectory segments must be connected so that the end position and velocity of the first segment must be equal to the initial position and the velocity of second segment. For smooth motion also acceleration must be continuous.

The motion of the manipulator can be described also by **sine-cosine functions** the derivatives of which can also be easily found. In this case the motion is smooth and jerk has a definite value (Fig. 4.16). The motion according to sine-cosine functions is the most energy efficient motion.

$$s(t)=\frac{s_{\max}}{2}-\frac{s_{\max}}{2}\cos(\Omega\, t) \tag{3.7}$$

$$v(t)=\frac{ds}{dt}=\frac{s_{\max}}{2}\cdot\Omega\cdot\sin(\Omega\, t) \tag{3.8}$$

$$a(t)=\frac{dv}{dt}=\frac{s_{\max}}{2}\cdot\Omega^2\cos(\Omega\, t) \tag{3.9}$$

$$j(t)=\frac{da}{dt}=-\frac{s_{\max}}{2}\Omega^3\sin(\Omega\, t) \tag{3.10}$$

$$\Omega=\frac{2\pi}{T}, \tag{3.11}$$

where T is the period of reciprocal motion.

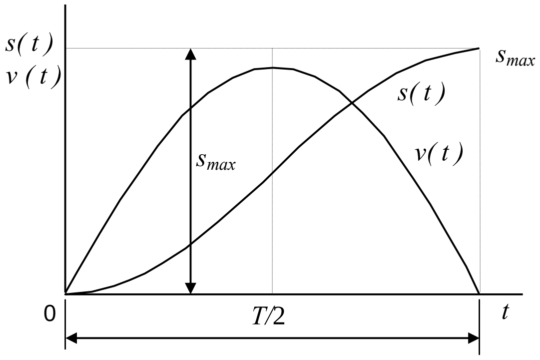


Figure 3.16 The motion time diagram described by sine-cosine functions of position and velocity

S-form diagram

For smooth and jerk limited motion of a manipulator tool or gripper, the drive speed starting and stopping processes are referred with the help of S-form diagrams (Fig. 4.17). S-form diagrams are included and in preprogrammed software of most modern drive control devices, e.g. frequency converters. During drive installation the parameters, giving the right form to the S-form diagram are needed to be stored to the control device. In this case the S-curve of full or partial form can be used. If the S-curve of the partial form is used (Fig. 4.17, *b*), the velocity curve between velocities v_1 and v_2 is linear and acceleration is constant.

The motion diagrams using S-curves for acceleration and deceleration are shown in Fig. 4.18. S-curve consists of two polynomials S1 and S2. If the full S-curve is used, then two polynomials are connected in the middle point of acceleration. If a partial S-curve is used, then two polynomials S1 and S2 are connected with a linear line. If the velocity diagram has S-curves for acceleration and deceleration, then acceleration and deceleration functions are linear and jerk has a definite constant value. The position of the manipulator is changing according to the position diagram that can be described by the function composed by the third order polynomials named the third order spline.

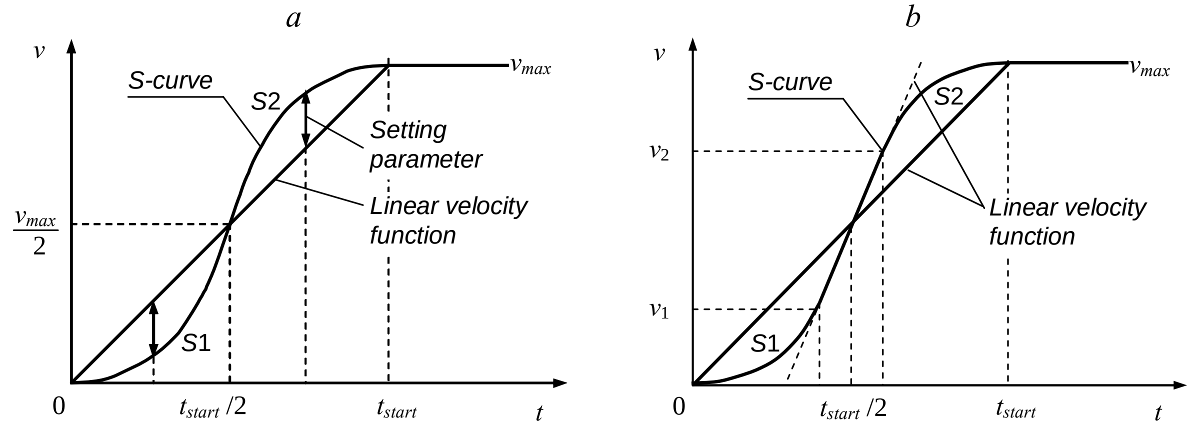


Figure 3.17 S-form acceleration diagram:
a – fully S-form acceleration, *b* – partially S-form acceleration

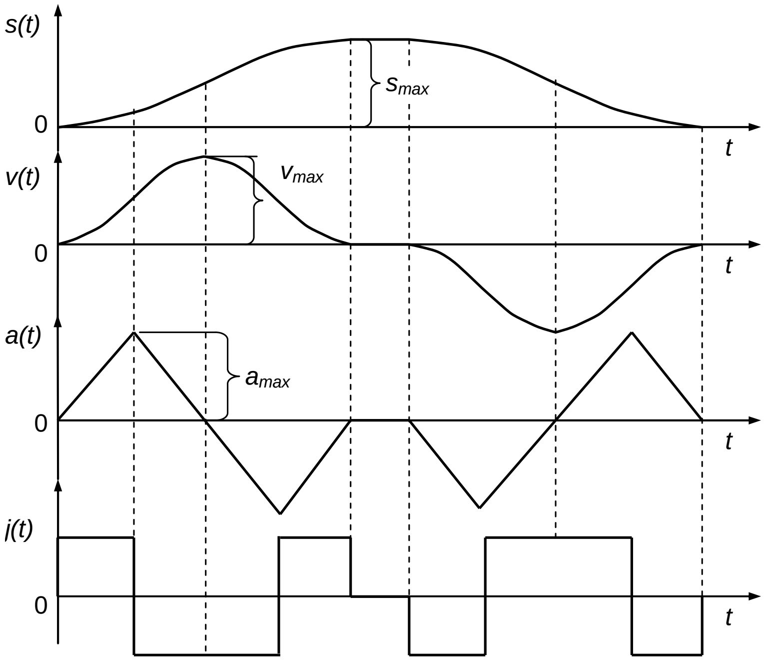


Figure 3.18 Using S-form acceleration and deceleration for limiting jerk

The motion time diagrams described above show the linear spatial motion from starting point A to final point B in the direction determined by the needs of the technological process. The planar projection of the spatial motion of manipulator joints is shown in Fig. 4.19 as an example. If the gripper or tool of the robot is moving with constant speed linearly, the joint motion may be very different in speed.

The planned motion will be realized by robot's drives. To get the reference signals for drive control, the planned motion must be described firstly in Cartesian coordinates and then transformed to the joint coordinate system. It means that we must describe the motion vector by projections x, y, z and then solve the inverse kinematics task of the manipulator and determine the joint rotation angles in relation to the position of each trajectory reference point.

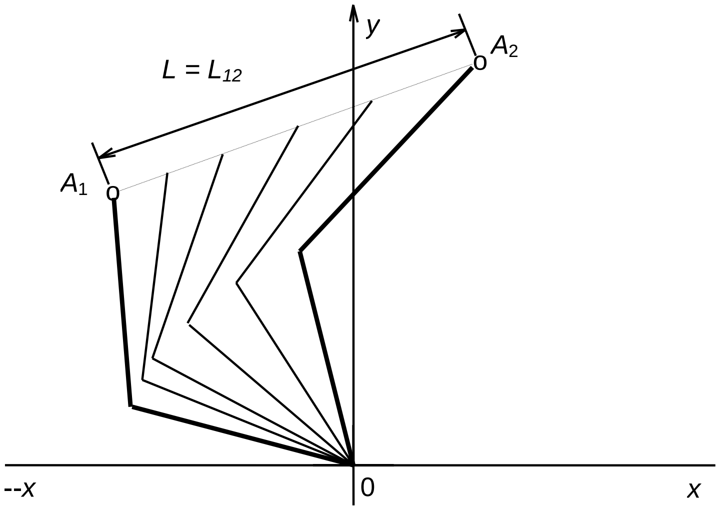


Figure 3.19 Motion of manipulator joints if the gripper or tool moves linearly

To calculate of projection points to the coordinate axes of the position such as time functions, in the case of linear motion, the distance between the starting and final motion points must be determined (see 4.5)

$$L = L_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

The trajectory direction coefficients

$$\begin{aligned}
k_{yx} &= (y_2 - y_1)/(x_2 - x_1) = 1/k_{xy};\\
k_{zx} &= (z_2 - z_1)/(x_2 - x_1) = 1/k_{xz};\\
k_{xy} &= (x_2 - x_1)/(y_2 - y_1) = 1/k_{yx};\\
k_{zy} &= (z_2 - z_1)/(y_2 - y_1) = 1/k_{yz};\\
k_{xz} &= (x_2 - x_1)/(z_2 - z_1) = 1/k_{zx};\\
k_{yz} &= (y_2 - y_1)/(z_2 - z_1) = 1/k_{zy}.
\end{aligned} \tag{3.12}$$

The coordinates of trajectory projection points on Cartesian axes can be calculated from

$$x_e = x_1 + L \cdot sign(dx)\sqrt{\frac{1}{(1 + k_{yx}^2 + k_{zx}^2)}}; \tag{3.13}$$

$$y_e = y_1 + L \cdot sign(dy)\sqrt{\frac{1}{(k_{xy}^2 + 1 + k_{zy}^2)}}; \tag{3.14}$$

$$z_e = z_1 + L \cdot sign(dz)\sqrt{\frac{1}{(k_{xz}^2 + k_{yz}^2 + 1)}}, \tag{3.15}$$

where L is the distance passed, x_1, y_1, z_1 are coordinates of initial position, dx, dy, dz are differentials of position during the movement of the gripper or tool. The time dependence functions of the distance passed, velocity and acceleration are named as motion diagrams. Therefore,

$$
\begin{aligned}
&L = s(t) \\
&dL/dt = ds/dt = v(t) \\
&d^2L/dt^2 = d^2s/dt^2 = dv/dt = a(t)
\end{aligned}
\tag{3.16}
$$

During the planning of robot motion, the motion diagrams of different forms can be selected. The triangular velocity time diagram guarantees the fastest motion between two points. The trapezoidal velocity time diagram is more efficient considering energy use and the sine form diagram gives maximal smoothness to motion.

The total motion time depends on the distance, maximal velocity and acceleration. In the case of motion triangular time diagram, the final time can be easily calculated from the geometrical data of the triangle:

$$t_f = 2 \cdot \sqrt{s_{\max} / a_{\max}}, \tag{3.17}$$

where a_{max} is the maximal acceleration. If acceleration and deceleration time is much less than the constant speed motion time, then the velocity time diagram is approximately simple rectangle and final motion time

$$t_f = s_{\max} / v_{\max}, \tag{3.18}$$

where v_{max} is maximal speed in this case of constant travel speed. In the case of sine form diagram the time interval to travel from the starting point to the final point is equal to half period of the sine and can be calculated on the given travel distance s_{max} by the formula:

$$t_f = \left(\pi / \sqrt{2}\right) \cdot \sqrt{s_{\max} / a_{\max}}. \tag{3.19}$$

To describe the motion diagram the distance passed must be given as the time function. In the case of triangular velocity diagram the time functions are as follows:

$$
\begin{aligned}
&s(t) = a_{\max} \cdot t^2 / 2, && \text{if} \quad t < t_f / 2 \\
&s(t) = (a_{\max} / 4) \cdot [t_f^2 - 2(t - t_f)^2], && \text{if} \quad t > t_f / 2
\end{aligned}
\tag{3.20}
$$

In the case of velocity sine form time diagram (Fig. 4.16) the passed distanceis calculated from the equation

$$s(t) = \frac{s_{\max}}{2} - \frac{s_{\max}}{2} \cos(\Omega\, t) \tag{4.21}$$

where $\Omega = \sqrt{2a_{\max} / s_{\max}}$.

Planned motion diagrams are used to control the robot motion. Equations (3.13…3.15) can be solved if the distance passed L is substituted by the position time function $s(t)$ given by equations (3.20) and (3.21). After that the manipulator inverse kinematics task must be solved and reference values for drives α_{1s}, α_{2s} ... α_{ns} found.

In the general form, the **task of trajectory planning** can be solved by the use of polynomials for trajectory interpolation. This procedure can be done in the joint coordinate system of the manipulator as well as in the Cartesian coordinate system (base or world coordinates). The motion from the starting position point s_0 to the final point s_f can be executed using different trajectories (Fig. 3.20), but all of these can be described with polynomials.

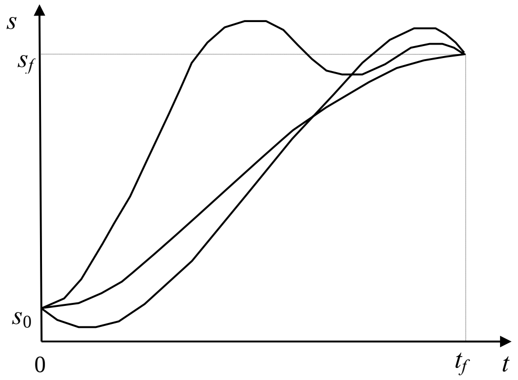


Figure 3.20 The motion from the starting position point to the final point and different trajectories

In the case of manipulator **position controlled motion the** following restrictions exist:

$s(0) = s_0;$ (the initial position)
$s(t_f) = s_f;$ (the final position)
$\dot{s}(0) = 0;$ (the initial velocity is zero)
$\dot{s}(t_f) = 0$ (the final velocity is zero)

These restrictions can be considered by using of the third order polynomial:

$$s(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \tag{3.22}$$

The equations for the calculation of velocity and acceleration:

$$v(t) = \dot{s}(t) = c_1 + 2c_2 t + 3c_3 t^2, \tag{3.23}$$

$$a(t) = \dot{v}(t) = \ddot{s}(t) = 2c_2 + 6c_3 t \tag{3.24}$$

Considering the restriction and equations (3.22), (3.23) and (3.24) formulas for the calculation of coefficients c_0...c_3 can be found:

$$\begin{aligned} s_0 &= c_0; \\ s_f &= c_0 + c_1 t_f + c_2 t_f^2 + c_3 t_f^3; \\ 0 &= c_1; \\ 0 &= c_1 + 2c_2 t_f + 3c_3 t_f^2. \end{aligned} \tag{3.25}$$

Solving of equations (3.25) gives the next values for coefficients c_i:

$$\begin{aligned}
c_0 &= s_0; \\
c_1 &= 0; \\
c_2 &= \frac{3}{t_f^2}(s_f - s_0); \\
c_3 &= -\frac{2}{t_f^3}(s_f - s_0).
\end{aligned} \tag{3.26}$$

If the position time diagram is described by the third order polynomial, then the velocity time diagram will be described by the second order polynomial and the acceleration time diagram is the straight line (Fig. 3.14).

In the case of **continuous path control** robot the motion between two path points must be continuous and the velocity values calculated from different polynomials must be equal (if the velocity is not equal to zero), i.e.

$$\begin{aligned}
\dot{s}(0) &= v(0) = v_0 \\
\dot{s}(t_f) &= v(t_f) = v_f
\end{aligned} \tag{4.27}$$

The motion is described by equations:

$$\begin{aligned}
s_0 &= a_0; \\
s_f &= c_0 + c_1 t_f + c_2 t_f^2 + c_3 t_f^3; \\
v_0 &= c_1; \\
v_f &= c_1 + 2c_2 t_f + 3c_3 t_f^2
\end{aligned} \tag{4.28}$$

From equations (3.28) the following can be found:

$$\begin{aligned}
c_0 &= s_0; \\
c_1 &= v_0; \\
c_2 &= \frac{3}{t_f^2}(s_f - s_0) - \frac{2}{t_f} v_0 - \frac{1}{t_f} v_f; \\
c_3 &= -\frac{2}{t_f^3}(s_f - s_0) + \frac{1}{t_f^2}(v_f - v_0).
\end{aligned} \tag{4.29}$$

In the case of continuous path control of the robot, the referred to path is given by discrete values of path coordinates $s_i(t)$, and the path (or trajectory) can be planned as smoothly connected together polynomials known as the interpolation by the use of a spline. The reference velocity in trajectory points can be determined as follows:

1) reference velocity is given by the robot user;
2) reference velocity is calculated by the use of given heuristic rules;
3) reference velocity is automatically calculated considering the continuity of acceleration

To calculate velocity in path points automatically the rules easily applied in computer must be defined. For example, the velocity is zero in the case of the sign change of the position value. Because the derivation of trajectory time function is equal to the instant value of velocity, the reference value for velocity in path points can be calculated as the average value of the derivative of the trajectory time function. Certainly, other heuristic rules for automated calculation of the reference velocity can be used.

The third method of calculation of the reference velocity is the most useful. It guarantees the continuity of acceleration on referred path points and also the continuous derivative of the motion time function (Fig. 3.21).

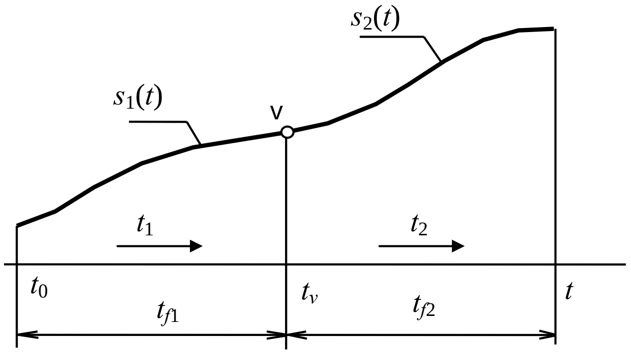


Figure 3.21 Connection of trajectory sectors in the case of acceleration continuity

If two trajectory sectors described with the third order polynomials

$$\begin{aligned} s_1(t) &= c_{10} + c_{11}t + c_{12}t^2 + c_{13}t^3 \\ s_2(t) &= c_{20} + c_{21}t + c_{22}t^2 + c_{23}t^3 \end{aligned} \tag{3.30}$$

are given and these trajectory sectors must be connected in the defined position point considered the condition of acceleration continuity, the following equations will be found:

$$\begin{aligned} s_0 &= c_{10}, && \text{because } t_1 = 0; \\ s_v &= c_{10} + c_{11}t_{f1} + c_{12}t_{f1}^2 + c_{13}t_{f1}^3, && \text{because } t = t_{fI}; \\ s_v &= a_{20}, && \text{because if } t_1 = t_V\text{, then } t_2 = 0; \\ s_v &= c_{20} + c_{21}t_{f2} + c_{22}t_{f2}^2 + c_{23}t_{f2}^3, && \text{because } t_2 = t_{f2}; \\ 0 &= c_{11}, && \text{because } v_1(0) = 0; \\ 0 &= c_{21} + 2c_{22}t_{f2} + 3c_{23}t_{f2}^2, && \text{because } v_2(t_{f2}) = 0. \end{aligned} \tag{3.31}$$

The conditions for the continuity of acceleration

$$\begin{aligned} c_{21} &= c_{11} + 2c_{12}t_{f1} + 3c_{13}t_{f1}^2; \\ 2c_{22} &= 2c_{12} + 6c_{13}t_{f1} \end{aligned} \tag{3.32}$$

The solution of equations (3.31) and considering the conditions (3.32), the coefficients of the third order polynomials can be found and the motion of continuous acceleration is defined. If $t_f = t_{f1} = t_{f2}$, i.e. the reference position points of the trajectory are given for equal time intervals:

$$
\begin{aligned}
c_{10} &= s_0; \\
c_{11} &= 0; \\
c_{12} &= \frac{12s_v - 3s_f - 9s_0}{4t_f^2}; \\
c_{13} &= \frac{-8s_v + 3s_f + 5s_0}{4t_f^3}; \\
c_{20} &= s_v; \\
c_{21} &= \frac{3s_f - 3s_0}{4t_f}; \\
c_{22} &= \frac{-12s_v + 6s_f + 6s_0}{4t_f^2}; \\
c_{23} &= \frac{8s_v - 5s_f - 3s_0}{4t_f^3}.
\end{aligned}
\tag{3.33}
$$

For better smoothness of the trajectory not only velocity but also acceleration and jerk must be referred to. In this case the order of polynomials describing the trajectory must be higher. Instead of the third order polynomials the fourth or fifth order polynomials could be used.

Sometimes trajectory sections described with polynomials have to be connected with sections described with linear lines (Fig. 3.22).

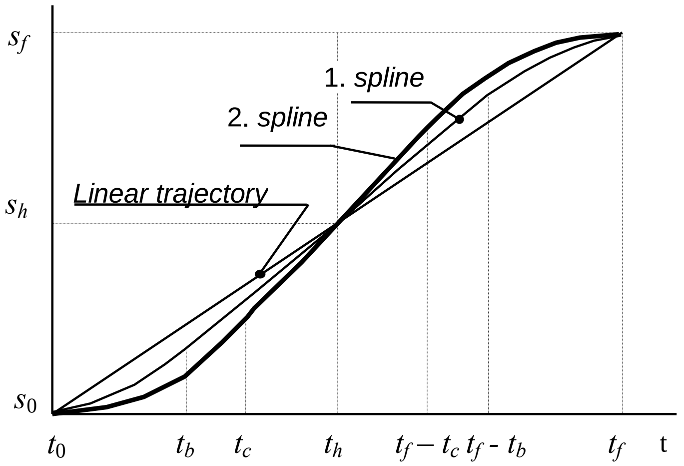


Figure 3.22 Description of the trajectory by linear lines and second order polynomials

In the case of linear position time diagram, the velocity is described by the rectangle form diagram and infinite acceleration and deceleration values could exist that, however, is practically impossible. In some cases, if simplified, the very slow motion with constant velocity can be described with the rectangle form diagram. If the position time diagram is described with two second order polynomials (parabolas), then the velocity time diagram has the triangular form and minimal execution time for motion can be achieved. All the intermediate forms of time diagrams are close the trapezoidal form. If equal values for acceleration and deceleration are planned, then motion time diagrams are symmetrical and have the central travel and time point s_h, t_h.

To connect the polynomials with the linear section of the trajectory (spline 1), the acceleration for both trajectory sectors in the connection point b must be equal and can be written:

$$at_b = \frac{s_h - s_b}{t_h - t_b}, \tag{3.34}$$

where

$$s_b = s_0 + \frac{1}{2} a t_b^2 . \tag{3.35}$$

If we consider that $t_f = 2t_h,$ the equation can be transformed

$$at^2 - at_f t_b + (s_f - s_0) = 0, \tag{3.36}$$

where t_f is the desired final motion time and t_b time of acceleration. The desired motion time can be found if acceleration is known.

$$t_f = [at_b^2 + (s_f - s_0)] / at_b \tag{3.37}$$

Starting or acceleration time t_v can be found if acceleration a and desired motion time t_f are given from the equation:

$$t_v = \frac{t_f}{2} - \frac{\sqrt{a^2 t_f^2 - 4a(s_f - s_0)}}{2a} \tag{3.38}$$

In the case of given desired motion time, the minimal value of acceleration can be calculated from the following:

$$a \geq \frac{4(s_f - s_0)}{t_f^2}. \tag{3.39}$$

If acceleration is below the calculated value, the motion from the starting point to the final point during reference time is not possible.