Contents

1 Introduction **2**

1.1 Primary goals and definitions 2

1.2 Notations 2

1.3 Machine state 3

1.4 Physical limitations 3

2 Control Model **4**

2.1 Position target : what we want to do 4

2.2 Physical limitations : what we can do 4

2.3 Kinematic constraints : what we choose to do 5

1 Introduction

1.1 Primary goals and definitions

Let a machine, whose behaviour we wish to control;

The machine is defined by a set of n actuators, a control system of size n, a geometric model, and a kinematic model;

The control system is coordinate system that is easier to manipulate, or that has more physical meaning than the actuator's system.
For example, a 6DOF arm's actuator system has no direct spatial meaning, and spacial trajectory are very difficult to express in it.
Target positions and speeds will be expressed in this system;

The geometry is the function that will be used to convert control positions to actuators positions;

1.2 Notations

Throughout the rest of this document, we will use the following notations :

$E = i \in \{0, n-1\}$.

i is the index of an actuator or of an axis of the control system, E.

A_i is an actuator.

- u_{i_A} is its unit;
- D_{i_A}, an interval included in $\mathbb{R}$ is its domain;
- p_{i_A} is its position expressed in u_{i_A};
- v_{i_A} is its speed, expressed in $u_{i_A} \cdot s^{-1}$
- a_{i_A} is its acceleration, expressed in $u_{i_A} \cdot s^{-2}$

C_i is an axis of the control system.

- u_{i_C} is its position;
- D_{i_C}, an interval included in $\mathbb{R}$ is its domain;
- p_{i_C} is its position expressed in u_{i_C};
- v_{i_C} is its speed, expressed in $u_{i_C} \cdot s^{-1}$
- a_{i_C} is its acceleration, expressed in $u_{i_C} \cdot s^{-2}$

$g : \prod_{i \in E} D_{i_C} \to \prod_{i \in E} D_{i_A}, \ (p_{i_C})_{i \in E} \to (p_{i_A})_{i \in E}$

is the geometry function, that translates control positions into actuators positions;

1.3 Machine state

The state of the machine at a given time is defined by the following set of variables

- u_{i_C} its control positions;
- v_{i_C} its control speeds;
- u_{i_C} its actuation positions;
- v_{i_C} its actuation speeds;

There is a direct relation between control and actuation positions, and control and actuation speeds, given by the geometry function, but not between actuation positions and speeds, or control positions and speeds;

To change the state of the machine, we execute a movement.

A movement is defined by :

- d_i its actuation distances. Distances could also be expressed in the control system, but it makes more sense to consider them in the actuation system, as actuators do the final movement;
- t its duration;

1.4 Physical limitations

The machine is controlled by actuators, that can be positioned at any point of their domain, without restrictions.

However, precautions must be taken when attempting to move them. Each type of actuator has its own kind of speed limitations, often correlated with the geometry;

For a given actuator, part of a machine, variation of the speed must be carefully monitored. If speed constraints are not met, actuators may halt, or be damaged, which in both case, would compromise the machine's integrity;

2 Control Model

2.1 Position target : what we want to do

Controlling the behaviour of the machine, in the less restrictive meaning, means determining successive actuation distances, and time intervals between them.

Ideally, we would like to determine consecutive positions (in any of the two coordinate systems), then determine actuation distances, and determine time intervals so that speed of a group of axis (in any of the two coordinate systems) matches a target one;

2.2 Physical limitations : what we can do

However, this is not so simple. Indeed, the fact of determining target distances for each actuator, at a given state, will by itself impose constraints on the time;

Suppose the machine in the state S. We wish to execute a movement, whose actuation distances have been computed.

Actuators are at a determined speed, and still have physical limitations. It is important to mention here that actuators are all moved simultaneously. Therefore, the time we choose for this movement will determine actuators speeds. But as we said previously, the variation of actuators speeds can't be set randomly without risking to damage actuators and the machine;

More formally, given a machine state S and a set of actuation distances $(d_i)_i$, each actuator will provide a duration interval $I_i = [t_{i_{\min}}, t_{i_{\max}}]$ that is acceptable for it;

Given this set of intervals, we must choose a duration t that respects all time constraints, ie $d \in \bigcap\limits_{i \in E} I_i$

What now if $\bigcap\limits_{i \in E} I_i = \emptyset$?

This means that the movement we are trying to execute is simply non realisable. This is caused by two time intervals that do not intersect, ie two actuators, that have incompatible speed variations : there is no compatible duration for one to decelerate slowly enough, and for the other to accelerate slowly enough; [1]

There are two potential solutions to this problem.

The naive (but simple !)solution, is to select an arbitrary time, given an absolute criteria. We could for example select a time that would limit acceleration, but break deceleration rules. The two great advantages of this solution are to respect required movement distances, and to require few calculation; The disavantage is that it breaks actuations limitations

The second solution complements the first one by modifying movement distances to match actuators limitations. Now that we have determined our movement time, we will modify movement distance for actuators that see their limitations broken; This solution makes the assumption that the movement distance that matched speed limitations for a given time can actually be determined. If so, this method will give a movement that can be safely executed by all actuators in its duration.

For now, we have determined movement distances, and a time interval in which we can safely choose the movement duration. This interval can be of length 0, and then we have no choice on the duration;

[1] This sentence can lead to assume erroneously that speed limitations only reside in their derivative, which might be true in simple machines, but in more complex ones (robotic arms for ex), may not be always the case.

2.3 Kinematic constraints : what we choose to do

Now that we have a determined set of movement distances, and a variable duration, we can select the most relevant duration.

We could, for example, restrict again the duration interval, to values that match acceleration bounds on a control axis. It could also be possible to verify that a deceleration is required to satisfy eventual jerk bounds on a control axis.

More generally, the selection of the duration comes by consecutive restrictions of the duration interval by ordered optional constraints; These constraints are functions that take the current state of the machine and provide a duration bound. They are qualified optionnal because the final duration will necessarily be in the interval that satisfies actuators constraints; They are qualified ordered because a constraint will evaluated only if the interval is not of length 0;

There can be as many constraints as we want. They are evaluated in order until interval is of length 0, or until all have been evaluated; If all constraints have been evaluated and the duration interval is not of length 0, the upper duration is selected;