Sequential Composition of Robust Controller Specifications

Jerome Le Ny and George J. Pappas

Abstract— **We present a general notion of robust motion specification and a mechanism for sequentially composing them. These motion specifications form tubular abstractions of the trajectories of a robot in different control modes, and are inspired by the techniques available for the design and analysis of low-level controllers. In particular, characterizations of input to output stability with respect to the disturbance inputs and trajectory tracking error output provide a suitable parametrization of the tubes. We also describe a randomized planner that composes different motion specifications from a given database to guarantee that any corresponding sequence of control modes steers a robot to a given region while avoiding obstacles. The main features of this compositional framework are that: i) at the planning level, it does not require the integration of the robot dynamics and the knowledge of how the controllers operate, but only the specification of the tracking performance achieved by these controllers. This enables a rigorous separation of concerns between establishing the feasibility of the high-level planning task and refining the low-level controller designs; ii) it can account quantitatively for robustness to unmodeled dynamics and various sources of disturbance and sensor noise. In particular it can help evaluate the impact of the sensing and actuation quality on the overall feasibility of a task.**

I. Introduction

Formalisms available for the analysis of cyber-physical systems such as robotic systems, based for example on hybrid automata mixing discrete events with differential equations, are arguably unwieldy to use for design purposes. System descriptions based on such general formalisms often accidentally include undesired behaviors complicating the analysis, such as Zeno phenomena, and lead to ubiquitous state-space explosion problems. To address the latter problem in particular, increasing emphasis is being placed on compositional design frameworks that allow one to build such systems from components and derive system properties from the separate analysis of the individual components.

In robotics and in particular motion planning, several such compositional frameworks have been proposed, including motion description languages (MDLs) [1], [2] and the maneuver automaton [3], as well as the sequential composition of funnels [4]–[6] based on preimage backchaining [7]. In these examples, a set of controllers is available to execute specific atomic behaviors, and one wishes to build more complex behaviors from these atoms.

A drawback of the MDL framework however is that it relies on the explicit knowledge of how the individual controllers work, i.e., of what input they generate in different part of the state space and as time evolves. As a result, it is difficult to separate the task of composing behaviors from the task of designing the individual controllers, and analyzing high-level composed behaviors still requires the complex integration of the dynamics of hybrid automata. In contrast, the approach in [4] to sequential composition of behaviors does not require the knowledge of the specific controller designs, but only of what the controllers achieve, namely local regulation of the trajectories to certain points in the state space. The framework is limited to the class of regulating controllers however, and does not address quantitatively certain issues, such as robustness to noise, disturbance, and unmodeled dynamics.

In Section II, we introduce an abstract notion of motion or controller specification and a mechanism for sequentially composing such specifications. Essentially, a specification defines a tube around a reference trajectory describing how accurately a control mode tracks this reference. Disturbances prevent perfect tracking, but the role of a controller is to eventually bring the tracking errors sufficiently close to zero. The funnels of [4] correspond to the case with no external disturbance and all reference trajectories consisting of fixed points. We also specialize the definition to the important case of input to output stability specifications, for which the robot in each control mode is required to be input to output stable [8] with respect to the disturbance inputs and the tracking error output. This notion provides a suitable concrete way of parametrizing the tubes, using standard Lyapunov analysis techniques. Finally, the mechanism for composing specifications describes how the tracking errors change (typically increase) as we switch between control modes.

Having defined a notion of control mode specification and of sequential composition of such specifications, we can then consider the general problem of searching

The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA `jeromel, pappasg@seas.upenn.edu`.

in the space of sequences of specifications for a plan achieving a desired behavior. Section III focuses on one fundamental such problem, where the goal is to steer the robot from one region to another while avoiding obstacles. This in fact a building block for more complex specifications based on temporal logic for example. We propose a planner to solve this problem, based on the Rapidly-exploring Random Tree (RRT) algorithm [9]. Finally, in Section IV we discuss a detailed example involving a curvature constrained vehicle moving among obstacles using simple line tracking controllers.

II. Robust Motion Specifications

A. Notation

For $s \in \mathbb{R}$ and a map $t \in \mathbb{R} \mapsto F(t)$, we define the time shifted map F^s by $F^s(t) := F(t+s)$. $\mathcal{P}(X)$ denotes the set of subsets of a set X. We denote the Euclidean norm of a vector x by $|x|$. $\mathcal{B}_\rho(x_0)$ denotes the Euclidean ball with center x_0 and radius $\rho \geq 0$. Finally, we denote the sup norm of a signal $t \to w(t)$ over the interval $[0, t)$ by

$$\|w\|_t := \sup_{s \in [0,t)} |w(t)|,$$

including $\|w\|_\infty$ when $t = \infty$ in the definition.

B. Motion Specifications

Consider a robot with dynamics described in local coordinates by

$$\dot{x} = f(x, u, w), \tag{1}$$

$$y = Hx, \tag{2}$$

where $x(t) \in \mathsf{X} \cong \mathbb{R}^n$ is the robot state, u is a control input, and w is a disturbance input, which can also account for modeling errors. We wish to steer the output $y \in \mathsf{Y} \cong \mathbb{R}^p$ (a linear transformation of the state) in the obstacle free subset $\mathsf{Y}_{\text{free}} \subset \mathsf{Y}$. For example, y can be a subset of the coordinates of x that are subject to collision avoidance constraints. We assume that H is full row rank.

We use a set of predesigned tracking controllers to steer the robot. A *control mode* m consists of a reference trajectory $t \mapsto r_m(t) \in \mathsf{X}$, together with a controller K_m specifying the input $u(\cdot)$ as long as the mode is engaged. The controller is designed so that the state x tracks r_m and hence the output y tracks the signal Hr_m. In some cases, only an output reference trajectory $y_{ref,m}$ could be specified and then we can typically define $r_m = H^\dagger y_{ref,m}$, where $H^\dagger$ is the Moore-Penrose pseudo-inverse of H, see [10, chap. 4]. In addition, a particular control setup can introduce additional disturbances denoted ν_m, e.g., measurement noise. Hence a control mode is also associated to a particular disturbance signal $w_m = [w^T, \nu_m^T]$, possibly of larger dimension than the open-loop disturbance w.

For planning purposes, we only rely on certain performance specifications for a controller and not on the knowledge of the actual controller design or implementation. More precisely, we need the following data for a given control mode.

Definition 1: A *motion specification* for a control mode m engaged on the interval $[t_m, t'_m)$ is a tuple $\sigma_m = (r_m, \mathsf{C}_m, \mathcal{E}_m, \mathcal{Z}_m, \mathsf{D}_m)$ where

1) $r_m : \mathbb{R}_+ \to \mathsf{X}$ is called the *reference trajectory* for the motion. We then define the *tracking error* $e_m : \mathbb{R}_+ \to \mathsf{X}$ by $e_m(t) = x^{t_m}(t) - r_m(t)$, $0 \leq t \leq t'_m - t_m$.
2) $\mathsf{C}_m \subset \mathsf{X}$ is called the *enabling condition* and one must have $e_m(0) \in \mathsf{C}_m$.
3) $\mathcal{E}_m : \mathbb{R}_+ \to \mathcal{P}(\mathsf{X})$ is a set-valued function of time, satisfying $\mathsf{C}_m \subset \mathcal{E}_m(0)$ and $e_m(t) \in \mathcal{E}_m(t)$ for all $0 \leq t \leq t'_m - t_m$.
4) $\mathcal{Z}_m : \mathbb{R}_+ \to \mathcal{P}(\mathsf{Y})$ is a set-valued function of time, satisfying $He_m(t) \in \mathcal{Z}_m(t)$ for all $0 \leq t \leq t'_m - t_m$.
5) D_m is a set of of admissible disturbance signals for the mode, equipped with a norm $\|\cdot\|_{\mathsf{D}_m}$.

In other words, a motion specification represents the tracking performance of a controller for a specific reference trajectory. There is an enabling condition that must be met by the robot state in order to start the motion. The sets $\mathcal{E}_m(t)$ and $\mathcal{Z}_m(t)$ represent the tracking errors in the state and output space respectively. Note that we can always choose $\mathcal{Z}_m(t) = H\mathcal{E}_m(t)$ in Definition 1-4). The reason for a separate definition is that one can sometime get a better estimate for the output tracking error than for the state tracking error. The next subsection provides more concrete representations of these tracking error sets. Next, we introduce a mechanism for sequentially composing motion specifications and specifying the propagation of the tracking error.

Definition 2: Two motion specifications $\sigma_{m_1}, \sigma_{m_2}$ for two modes m_1 and m_2 can be *sequentially composed* after a time d_1 in mode m_1, denoted

$$\sigma_{m_1} \rhd_{d_1} \sigma_{m_2},$$

if they satisfy

$$r_{m_1}(d_1) + \mathcal{E}_{m_1}(d_1) \subset r_{m_2}(0) + \mathsf{C}_{m_2}.$$

In this case, we define the *tracking error transition map* $\tau^{d_1}_{\sigma_{m_1},\sigma_{m_2}} : \mathcal{P}(\mathsf{X}) \times \mathcal{P}(\mathsf{Y}) \to \mathcal{P}(\mathsf{X}) \times \mathcal{P}(\mathsf{Y})$ by

$$\tau^{d_1}_{\sigma_{m_1},\sigma_{m_2}}(\mathcal{E}_{m_1}(d_1), \mathcal{Z}_{m_1}(d_1)) = (\mathcal{E}_{m_2}(0), \mathcal{Z}_{m_2}(0)).$$

Hence two motion specifications can be sequentially composed after duration d_1 in mode m_1 if we know with certainty that mode m_1 steered the state close enough to $r_2(0)$ in order to engage mode m_2. In this case, the transition map transforms the set of possible tracking errors for mode m_1 at the time of switching to an initial set of possible tracking errors for mode m_2. Finally, the planning problem that we address in this paper is the following.

Problem 1: Given a set $S \subset \mathsf{X}$ of possible initial states $x(0)$ for the robot, and a set $G \subset \mathsf{Y}_{\text{free}}$ of desired outputs to reach, given a family $\mathcal{F}$ of motion specifications, does there exist $K \in \mathbb{N}$ and sequences $\sigma_{m_0}, \sigma_{m_1}, \ldots \sigma_{m_K} \in \mathcal{F}$ and $d_0, \ldots, d_{K-1} \geq 0$ such that

1) $S \subset \mathsf{C}_{m_0}$,
2) $\sigma_{m_i} \rhd_{d_i} \sigma_{m_{i+1}}$, $0 \leq i \leq K-1$,
3) $Hr_{m_i}(t) + \mathcal{Z}_{m_i}(t) \in \mathsf{Y}_{\text{free}}$, for all $0 \leq i \leq K-1, 0 \leq t \leq d_i$ (i.e., collisions are avoided),
4) $Hr_{m_K}(d_K) + \mathcal{Z}_{m_K}(d_K) \subset G$.

That is, we are looking at the planning level for a sequence of specifications which is guaranteed to lead from a starting region to a goal region while keeping the output in the obstacle free space Y_{free}.

C. Input to Output Stability Specifications

The previous subsection defines motions specifications in abstract set-theoretical terms. In practice, we need to instantiate Definition 1 to versions that are computationally more convenient and motivated by the tools available for the analysis and design of controllers.

If some control mode m with a controller K_m tracking a known continuously differentiable trajectory $t \to r_m(t) \in \mathsf{X}$ is engaged from time t_m onward, the closed-loop system follows the dynamics, for $t \geq 0$,

$$\dot{\xi}_m = f_m(t, \xi_m, w_m), \quad e_m(0) \in \mathsf{C}_m, \tag{3}$$

$$z_m := y - Hr_m = He_m, \quad w_m \in \mathsf{D}_m \tag{4}$$

where e_m denotes the tracking error as in Definition 1 and the state of the closed-loop system $\xi_m = [e_m^T, \zeta_m^T]^T$ includes the state ζ_m of the controller K_m. The disturbance signal is $w_m = [w^T, \nu_m^T]$ as explained above.

Example 2.1: Consider a dynamic controller with error feedback,

$$\begin{aligned} \dot{\zeta}_m &= \eta(\zeta_m, h(e_m) + \nu_m) \\ u &= \theta(\zeta_m), \end{aligned}$$

where ν_m is a noise contaminating a tracking error measurement $h(e_m)$. Then we have for the closed loop system

$$\begin{aligned} \dot{e}_m &= f(e_m + r_m, \theta(\zeta_m), w) - \dot{r}_m \\ \dot{\zeta}_m &= \eta(\zeta_m, h(e_m) + \nu_m) \\ z_m &:= He_m, \end{aligned}$$

which is of the form (3), (4), since since $r_m, \dot{r}_m$ are known functions of time.

A convenient requirement for the control modes is that the closed-loop dynamics (3) in each mode be input to state stable (ISS) or input to output stable (IOS) [8], [11]. We use the following terminology. A function $\gamma : [0, \infty) \to [0, \infty)$ is of class $\mathcal{G}$ if it is continuous, non-decreasing and satisfies $\gamma(0) = 0$. It is of class $\mathcal{K}$ if it is of class $\mathcal{G}$ and strictly increasing. $\mathcal{GL}$ (resp. $\mathcal{KL}$) is the class of functions $[0, \infty)^2 \to [0, \infty)$ that are of class $\mathcal{G}$ (resp. $\mathcal{K}$) on their first argument and decrease to zero on their second argument.

For a given mode m with dynamics (3), let us assume available a set of J inequalities of the form

$$|g^i(e_m(t))| \leq \max\{\beta_m^i(|\xi_m(0)|, t), \gamma_m^i(\|w_m\|_t)\}, \tag{5}$$

for all $t \geq 0$, $e_m(0) \in \mathsf{C}_m$, $w_m \in \mathsf{D}_m^\alpha = \{w | \|w\|_\infty \leq \alpha\}$, and $i = 1, \ldots, K$, where β_m^i are $\mathcal{GL}$ functions and γ_m^i are of class $\mathcal{G}$. In other words, the closed-loop system in mode m is locally input to output stable [8] with respect to the input disturbance w_m and the outputs $g^i(e_m)$, $i = 1, \ldots, J$, with the minor variation that we use functions of class $\mathcal{G}$ and $\mathcal{GL}$ instead of $\mathcal{K}$ and $\mathcal{KL}$ in the definition. These inequalities can be used to give a representation of the set-valued maps $\mathcal{E}_m$, $\mathcal{Z}_m$ of Definition 1.

The ISS and IOS notions [8], [11] are typically used to study asymptotic stability properties of nonlinear systems. The function β_m characterizes the transient regime of the mode, and the quantity $\gamma_m(\|w_m\|_\infty)$ the steady-state tracking error. Here however, we use the functions β_m, γ_m to abstract the dynamics of the robot over finite time intervals. Hence special cases of the general definition are useful for computations. Of particular interest are functions β_m of the exponentially decreasing form

$$\beta_m(\xi, t) = k_m(\xi) e^{-\lambda_m t},$$

where k_m is a function of class $\mathcal{G}$, for example $k_m(\xi) = k_{0,m} |\xi|$. Assuming that the functions k_m and γ_m admit finite dimensional parametrizations, they can be stored in memory together with the decay rate λ_m. This provides a finite-dimensional abstraction of the closed-loop dynamics of mode m, including the effect of perturbations. The main advantage of using characterizations of the form (5) for computations is that Lyapunov analysis techniques are available to derive such bounds, see Section IV for an example.

Example 2.2: Suppose that we only have the following two such inequalities for each motion specification. The first bounds the Euclidean norm of the state tracking error

$$|e_m(t)| \leq \max\{\beta_m^1(|\xi_m(0)|, t), \gamma_m^1(\|w_m\|_t)\}. \quad (6)$$

The second bounds the Euclidean norm of the output tracking error

$$|z_m(t)| \leq \max\{\beta_m^2(|\xi_m(0)|, t), \gamma_m^2(\|w_m\|_t)\}, \quad (7)$$

assuming it is less conservative than using $|He_m| \leq \|H\||e_m|$ in (6). In this case, $r_m(t) + \mathcal{E}_m(t)$ and $Hr_m(t) + \mathcal{Z}_m(t)$ are Euclidean balls around the references $r_m(t)$ and $Hr_m(t)$ respectively.

Consider now the transition between two motion specifications σ_{m_1} and σ_{m_2}, where σ_{m_1} is followed for a duration d_1. From the first inequality (6), we get a bound of the form $|x(t_1 + d_1) - r_{m_1}(d_1)| \leq \rho_1$, where t_1 is the time at which mode m_1 is engaged. If $\mathcal{B}_{\rho_1}(r_{m_1}(d_1)) \subset r_{m_2}(0) + \mathsf{C}_{m_2}$, we can conclude that the modes can be sequentially composed. Let $t_2 = t_1 + d_1$. Since $t \mapsto x(t)$ is continuous, we have

$$\begin{aligned} |e_{m_2}(0)| &= |x(t_1 + d_1) - r_{m_2}(0)| \\ &\leq \rho_1 + |r_{m_1}(0) - r_{m_2}(0)| := \rho_2. \end{aligned}$$

Hence we can define $\mathcal{E}_2(0)$ as $\mathcal{B}_{\rho_2}(r_{m_2}(0))$. Moreover, we have

$$|\xi_{m_2}(0)| \leq \sqrt{\rho_2 + |\zeta_{m_2}(0)|^2},$$

where $\zeta_{m_2}(0)$ depends on the initialization of the controller for mode m_2. This last bound is used to replace $|\xi_{m_2}(0)|$ in the inequalities (6), (7) for σ_{m_2}, and to obtain representations of the error tracking maps $\mathcal{E}_{m_2}(t), \mathcal{Z}_{m_2}(t)$.

III. Searching for a Motion Composition

In this section, we describe a randomized algorithm to solve Problem 1, based on the RRT planner of [9], and exploiting the notion of sequential composition of robust motion specifications presented in Section II-B. The pseudo-code for the algorithm is given as Algorithms 1-3.

The RRT algorithm builds a graph that is eventually used to steer the vehicle to the goal region by sequentially composing modes. A node n in the graph records

1) the specification σ_{m_n} of the mode m_n used to reach the node.
2) the duration d_n since the mode m_n was last engaged.
3) The index $\texttt{pred}(n)$ of the predecessor node in the tree.

Intuitively, a node n can be associated to the point $r_{m_n}(d_n) \in \mathsf{X}$. Moreover, mode m_n was last engaged at some ancestor of node n, although there was perhaps no mode switch at the direct predecessor $\texttt{pred}(n)$. We initialize the tree with the root node with index 0, recording $\hat{x}_0$, an estimate of the initial state of the robot, and $\hat{\mathcal{E}}_0$, so that $S \subset \hat{x}_0 + \hat{\mathcal{E}}_0$, with S as in Problem 1.

Given a partially constructed tree, to create a new node, we first generate a new sample point s in the free output space $\mathsf{Y}_{\texttt{free}}$. Then, we find a node in the RRT, say node n, close to this sample point according to some heuristic notion of distance $\mu : \mathcal{N} \times \mathsf{Y} \to \mathbb{R}_+$, where $\mathcal{N}$ is the set of nodes in the RRT. Generally μ involves the distance from $Hr_{m_n}(d_n)$ to s and a measure of the size of the output or state tracking error $\mathcal{E}_{m_n}(d_n)$ or $\mathcal{Z}_{m_n}(d_n)$ at the node. In the example of Section IV, we take

$$\mu(n, s) = |Hr_{m_n}(d_n) - s| + \alpha \, \text{diam}(\mathcal{Z}_{m_n}(d_n)),$$

for some constant $\alpha \geq 0$.

Once the sample s is produced and a node n in the tree is selected "close" to s, we create one or more new nodes from node n. The function generating new nodes is described in Algorithm 3. First, we consider a number of possible motion specifications that can be sequentially composed with σ_{m_n} after d_n (see Section II-B), with reference trajectories starting from or close to $r_{m_n}(d_n)$, and preferably steering the output toward s. We also select deterministically or at random a duration T for which we consider following each of these modes.

Among the set of motions considered, we first include the possibility of continuing to follow mode m_n, for an additional time T. We also select some other compatible motion specifications to potentially switch to and to follow at least for some time T from node n. For example, we could switch to tracking a straight line towards s, if such a motion exists. Collision avoidance is checked at this stage, to verify that the tube around the reference trajectory does not intersect any obstacle, and to remove the motion from consideration if it does. In case of mode switching, we use the transition map τ to evaluate the change (typically an increase in size) in the tracking error sets due to switching. For each of the motions considered, we have a specification, in particular a final position on the corresponding reference trajectory at time T, together with tracking error sets. We finally select among these motions the one that steers the vehicle closest to s, again based on the "distance" function μ. We add a corresponding node to the tree with predecessor n, recording the motion specification picked, as well as a motion duration equal to $d_n + T$ if the mode m_n was extended, and T otherwise. Note that it is also possible to add several nodes at this step, if several motions give satisfactory performance. However, there is a tradeoff since the search for the closest node to the generated sample takes longer as the tree gets bigger.

Allowing to continue a given motion is important in practice to obtain tighter bounds. Indeed, tracking error sets can grow as we switch from one motion specification to another. For example a robot performing a maneuver can see its localization performance decrease for typical sensor packages.

Algorithm 1 Robust Forward RRT Planner. $\hat{x}_0$ is the initial state estimate, with $\hat{x}_0 \in \hat{\mathcal{E}}_0$. G is the desired goal set. $\mathcal{T}$ is the constructed tree.

Require: $\hat{x}_0, \hat{\mathcal{E}}_0, \hat{\mathcal{Z}}_0; G$
  $\sigma \leftarrow (r \leftarrow \hat{x}_0, \mathsf{C} \leftarrow \perp, \mathcal{E} \leftarrow \hat{\mathcal{E}}_0, \mathcal{Z} \leftarrow \hat{\mathcal{Z}}_0, \mathsf{D} \leftarrow \perp)$
  $\mathcal{T}$.addNode$(\sigma, d \leftarrow 0, \texttt{pred} \leftarrow \perp)$
  while 1 **do**
    $s \leftarrow$ randomOutput$(\mathsf{Y}_{\texttt{free}})$
    $n \leftarrow \mathcal{T}$.extend(s)
    if $r_{m_n}(d_n) + \mathcal{E}_{m_n}(d_n) \subset G$ **then**
      return $\mathcal{T}, n$
    end if
  end while

Algorithm 2 $\mathcal{T}$.extend(s). $\mathcal{T}$.nearestNeighbor(x) returns the node "closest" to x in the tree according to μ, see the main text.

  $\text{n}_{near} \leftarrow \mathcal{T}$.nearestNeighbor$(s, \mu)$
  $\text{n}_{new} \leftarrow$ generateChildNode(n_{near}, s)
  $\mathcal{T}$.addNode(n_{new})

IV. Example

A. Problem Formulation

The purpose of this section is to illustrate the concepts outlined above for a specific example, involving robust motion planning for a Dubins vehicle. Much of the recent work on planning robust paths assumes a probabilistic description of the disturbances, see e.g. [12], [13]. These planners either make more restrictive assumptions on the environment and the motion library [12], or aim at higher precision but require much more computations because they integrate the system dynamics and perform many simulations [13].

Algorithm 3 generateChildNode(n, s). Creates a new node with predecessor node n and a motion specification and duration steering the robot from n towards s. m_n is the mode used to reach n. closestNode returns a node "closest" to the sample $s \in \mathsf{Y}$ from a list of candidate nodes, with distance measured according to μ.

  modeList $\leftarrow$ generateCandidateMotions(n, s)
  T = randomTime() {- duration of motion ; random or fixed -}
  for m in modeList **do**
    if $\text{m} = m_\text{n}$ **then**
      {- we continue the same motion -}
      $d \leftarrow d_\text{n} + T$
      $\sigma \leftarrow \sigma_{m_\text{n}}$
      Check for collision of the tube $r_{m_\text{n}}(t) + \mathcal{Z}_{m_\text{n}}(t), t \in [d_n, d_n + T]$.
    else
      {- we switch mode -}
      Check $r_{m_\text{n}}(d_\text{n}) + \mathcal{E}_{m_\text{n}}(d_\text{n}) \subset r_m(0) + \mathsf{C}_m$
      $\mathcal{E}_0, \mathcal{Z}_0 \leftarrow \tau^{d_\text{n}}_{m_\text{n},\text{m}}$
      $\sigma \leftarrow \sigma_\text{m}$ s.t. $\mathcal{E}_\text{m}(0) \supset \mathcal{E}_0$, $\mathcal{Z}_\text{m}(0) \supset \mathcal{Z}_0$
      $d \leftarrow T$
      Check for collision of the tube $r_\text{m}(t) + \mathcal{Z}_\text{m}(t), t \in [0, d]$.
    end if
    if All checks passed **then**
      Add to candidateNodeList the node $(\sigma, d, \texttt{pred} \leftarrow \text{n})$
    end if
  end for
  return closestNode(candidateNodeList$, s, \mu)$

The dynamics of the vehicle with configuration $(x, y, \theta) \in \mathbb{R}^2 \times \mathcal{S}^1$, fixed constant velocity v and minimum turning radius $1/2$ are described by the equations

$$\begin{aligned} \dot{x} &= v \cos\theta + w_x, \\ \dot{y} &= v \sin\theta + w_y, \\ \dot{\theta} &= 2\, v \,\text{sat}\left(\frac{1}{2}(u + w_\theta)\right), \end{aligned}$$

where sat is the saturation nonlinearity restricting a scalar to $[-1, 1]$, i.e., $\text{sat}(x) = \max\{-1, \min\{1, x\}\}$. The signal $w(t) = [w_x(t), w_y(t), w_\theta(t)]^T$ represents a bounded perturbation. The only available input u controls the angular rate $\dot{\theta}$. In particular, the vehicle can only move forward at fixed velocity v, and we can set $v = 1$ in the following without loss of generality. We consider a family of controllers tracking motions along half-lines with arbitrary starting point and arbitrary direction.

B. Line tracking controllers

Consider a directed half-line with orientation θ_l starting from $[r_x(0), r_y(0)]^T$. By a change to a new set of coordinates denoted (χ, δ, ϕ), the half-line becomes the χ-axis, oriented toward increasing χ-coordinates, δ is the vehicle distance to the line, and ϕ its orientation with respect to the line. First, in this subsection, we design a controller regulating δ and ϕ to zero. That is, it steers the vehicle to asymptotically move along the half-line. However, it is not designed to track a specific open-loop reference trajectory $[\chi(t), 0, 0]$ along that line. Subsection IV-C details how to produce a control mode specification as in Definition 1 from this design.

We propose a controller that works under the following assumption on the initial state

$$\mathsf{C} = \left\{ [\chi, \delta, \phi]^T \middle| |\delta| \leq 0.9, |\phi| \leq \pi/3 \right\}.$$

In particular, the vehicle is initially oriented in the desired direction. The dynamics of interest are

$$\dot{\delta} = \sin\phi + w_\delta, \quad \dot{\phi} = 2\,\mathrm{sat}\left(\frac{1}{2}(u + w_\phi)\right),$$

where w_δ, w_ϕ are disturbance signals, and we keep the notation u for the control input after the coordinate change for simplicity. Consider the simple proportional controller

$$u = -(k_1\delta + k_2 \sin\phi), \quad k_1, k_2 > 0. \tag{8}$$

To study the asymptotic stability of the point $\delta = \phi = 0$ for the closed loop system, and hence regulation to the line, we introduce the Lyapunov function

$$\begin{aligned} V(\delta,\phi) &= \frac{\alpha}{2}\delta^2 + \beta\delta\phi + \gamma\int_0^\phi \sin\psi d\psi \\ &= \frac{\alpha}{2}\delta^2 + \beta\delta\phi + \gamma(1-\cos\phi), \end{aligned}$$

where $\alpha, \gamma > 0$. Assuming a priori that ϕ remains in $[-\pi/3, \pi/3]$, we have

$$\frac{\alpha}{2}\delta^2 + \beta\delta\phi + \frac{1}{2}\frac{\gamma}{1.1}\phi^2 \leq V(\delta,\phi) \leq \frac{\alpha}{2}\delta^2 + \beta\delta\phi + \frac{\gamma}{2}\phi^2. \tag{9}$$

In particular, a sufficient condition for V to be positive definite is $\frac{\alpha\gamma}{1.1} - \beta^2 > 0$.

Let us also assume temporarily that the dynamics do not saturate, i.e., $|u(t) + w_\phi(t)| \leq 2$ for all t along the trajectory, and verify that this is the case once we have chosen the parameters k_1, k_2 below. Denote $\dot{V} := \frac{d}{dt}V(\delta,\phi)$ the derivative of V along the trajectories. We have, in the absence of saturation,

$$\begin{aligned} \dot{V} =& \alpha\delta(\sin\phi + w_1) + \beta\phi(\sin\phi + w_1) \\ &+ \beta\delta(-k_1\delta - k_2\sin\phi + w_2) \\ &+ \gamma\sin\phi(-k_1\delta - k_2\sin\phi + w_2) \\ =& (\alpha - \beta k_2 - \gamma k_1)\delta\sin\phi - \gamma k_2 \sin^2\phi \\ &+ \beta\phi\sin\phi - \beta k_1\delta^2 \\ &+ (\alpha w_1 + \beta w_2)\delta + \beta w_1\phi + \gamma w_2 \sin\phi \\ \leq& -(\gamma k_2 - 1.25\beta)\sin^2\phi - \beta k_1\delta^2 \\ &+ (\alpha|w_1| + |\beta||w_2|)\delta + |\beta||w_1||\phi| + \gamma|w_2||\sin\phi|, \end{aligned} \tag{10}$$

by taking $\alpha = \beta k_2 + \gamma k_1$, and using the fact that $\phi\sin\phi \leq \sin^2\phi$ for $\phi \in [-\pi/3, \pi/3]$. We impose $\beta > 0$ to get a negative quadratic term in (10).

The two components of the IOS bound (5) are now obtained by considering two regions in the state space. First, introduce $\theta \in (0,1)$ and rewrite (10) as

$$\begin{aligned} \dot{V} \leq& -(1-\theta)[(\gamma k_2 - 1.25\beta)\sin^2\phi + \beta k_1\delta^2] \\ &- \theta[(\gamma k_2 - 1.25\beta)\sin^2\phi + \beta k_1\delta^2] \\ &+ (\alpha|w_1| + \beta|w_2|)|\delta| + (1.25\beta|w_1| + \gamma|w_2|)|\sin\phi|, \end{aligned}$$

using $|\phi| \leq 1.25|\sin\phi|$ on $[-\pi/3, \pi/3]$. Now let b_i be an upper bound on $|w_i|, i \in \{\delta, \phi\}$, and define

$$\begin{aligned} p(\delta,\phi) =& |\delta|(\theta\beta k_1|\delta| - \alpha b_\delta - \beta b_\phi) \\ &+ |\sin\phi|(\theta(\gamma k_2 - 1.25\beta)|\sin\phi| - (1.25\beta b_\delta + \gamma b_\phi)). \end{aligned}$$

Consider the region of the state space

$$R := \{(\delta,\phi) | p(\delta,\phi) \geq 0\}.$$

In that region, and for $\phi \in [-\pi/3, \pi/3]$, we have

$$\begin{aligned} \dot{V} &\leq -(1-\theta)[\beta k_1\delta^2 + (\gamma k_2 - 1.25\beta)\sin^2\phi] - p(\delta,\phi) \\ &\leq -(1-\theta)[\beta k_1\delta^2 + 0.68(\gamma k_2 - 1.25\beta)\phi^2]. \end{aligned} \tag{11}$$

We can compare $\dot{V}$ to V in this case to find an exponentially converging upper bound on V. Indeed, comparing the quadratic functions on the right-hand sides of (11) and (9), we can obtain an inequality of the form

$$\dot{V} \leq -\lambda V, \tag{12}$$

from which we conclude

$$V(\delta(t),\phi(t)) \leq V(\delta(0),\phi(0))e^{-\lambda t}, \tag{13}$$

for all t. Inequality (13) provides an upper bound on the norm of the vector $[\delta,\phi]^T$ by using the left-hand side of (9). In fact, we can bound the two coordinates independently, since

$$V(\delta,\phi) \geq \min_\phi V(\delta,\phi) \geq \frac{1}{2}(\alpha - 1.1\beta^2/\gamma)\delta^2,$$

by the Schur complement formula. Hence

$$\frac{1}{2}\left(\alpha - 1.1\frac{\beta^2}{\gamma}\right)\delta^2 \leq V(\delta(0),\phi(0))e^{-\lambda t}. \qquad (14)$$

Similarly, we have

$$\frac{1}{2}\left(\frac{\gamma}{1.1} - \frac{\beta^2}{\alpha}\right)\phi^2 \leq V(\delta(0),\phi(0))e^{-\lambda t}. \qquad (15)$$

To obtain the second component of the maximum in (5), i.e, the ultimate bound, we consider the region outside of R, where we cannot conclude that V is decreasing from the argument above. Let

$$K_1 = \beta\theta k_1,\ K_2 = \alpha b_\delta + \beta b_\phi,$$
$$L_1 = \theta(\gamma k_2 - 1.25\beta),\ L_2 = 1.25\beta b_\delta + \gamma b_\phi,$$

so that

$$p(\delta,\phi) = K_1\delta^2 - K_2|\delta| + L_1|\sin\phi|^2 - L_2|\sin\phi|.$$

Now $p(\delta,\phi) < 0$ leads to an upper bound on the norm of $[\delta,\phi]^T$, whose value depends on b_δ, b_ϕ. Again, we can also bound each coordinate independently. We must have

$$K_1\delta^2 - K_2|\delta| < \max_{\phi}\{-L_1|\sin\phi|^2 + L_2|\sin\phi|\}, \quad (16)$$

which provides a bound on $|\delta|$. Similarly, we obtain a bound on $|\sin\phi|$ from

$$L_1\sin^2\phi - L_2|\sin\phi| < \max_{\delta}\{-K_1\delta^2 + K_2|\delta|\}. \quad (17)$$

Finally, from the bounds on $|\delta|$, $|\sin\phi|$ and the norm of $[\delta,\phi]^T$ in the transient and ultimate regime, one can also verify for specific values of k_1, k_2 and bounds $\epsilon_\delta \leq 0.9, \epsilon_\phi \leq \pi/3$ such that

$$|\delta(0)| \leq \epsilon_\delta,\quad |\psi(0)| \leq \epsilon_\phi,$$

that the controller does not saturate, and that ϕ remains in $[-\pi/3, \pi/3]$, as assumed in the previous calculations.

Numerical values: Let us take $k_1 = 1.3, k_2 = 0.9$, and assume $b_\delta = 0.02, b_\phi = 0.05$. We then choose $\theta = 0.5, \beta = 0.3, \gamma = 0.75$, and $\alpha = \gamma k_1 + \beta k_2$. Calculations then show that $\lambda = 0.0775$ works in (12). Inequality (16) leads to $|\delta| \leq \delta_\infty := 0.27$ and (17) to $|\sin\phi| \leq 0.34$ or $|\phi| \leq \phi_\infty := 0.3469$ after the transient regime.

C. Line tracking mode specification

From the analysis in the previous subsection, we can now derive a control mode specification as in Definition 1 for a controller tracking a half-line with origin $[r_x(0), r_y(0)]^T$ and orientation θ_l. For the parametrization of the error tracking tubes $\mathcal{E}(t)$ and $\mathcal{Z}(t)$, we can as before work in the rotated coordinate system $[\chi, \delta, \phi]$. The output coordinates are χ, δ. Assuming for simplicity that the coordinate χ can be measured exactly, the reference trajectory tracked is

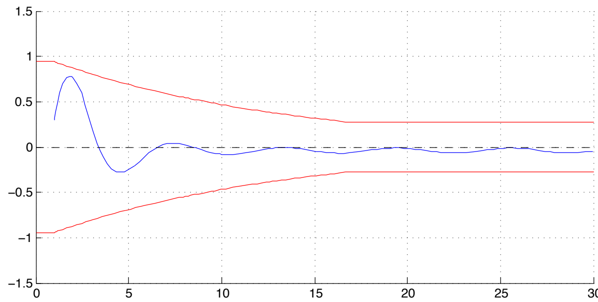

Fig. 1. Illustration of the tubular approximation for a straight line motion tracked by the vehicle, with $\phi(0) = \pi/3, \delta(0) = 0.3, \chi(0) = 1$ and perturbation $w_\chi = -0.02, w_\delta = 0.02\cos(t), w_\phi = -0.05$. The line tracked is the χ-axis, dashed.

$$r(0) = 0,\ r(t) = [\chi(t), 0]^T,\ t > 0.$$

That is, we do not give an open-loop specification of the χ coordinate, the vehicle tracks its own projection on the half-line. This simplification is possible for our purpose here since the speed at which the vehicle progresses along the line is irrelevant.

Note also that because the vehicle speed is 1, we have clearly $\chi(t) \leq \chi(0) + t$. Assume an initial tracking error of the form

$$\mathcal{E}(0) = \left\{[\chi,\delta,\phi]^T \middle| |\chi| \leq \epsilon_\chi, |\delta| \leq \epsilon_\delta, |\psi| \leq \epsilon_\phi\right\},$$

with $\epsilon_\delta \leq 0.9, \epsilon_\phi \leq \pi/3$. Then for $t > 0$, we can take

$$\mathcal{E}(t) = \Big\{[\chi,\delta,\phi]^T \Big| |\chi| = 0,\ |\delta| \leq \max\{k_\delta(\epsilon_\delta,\epsilon_\phi)e^{-\lambda t}, \delta_\infty\},\ |\phi| \leq \max\{k_\phi(\epsilon_\delta,\epsilon_\phi)e^{-\lambda t}, \phi_\infty\}\Big\},$$

for some functions k_δ, k_ϕ derived from (14), (15). Fig. 1 illustrates the tubes $r(t) + \mathcal{E}(t), t \geq 0$, containing all the line tracking trajectories with initial error in $\mathcal{E}(0)$.

Finally, we consider the composition of two successive motion specifications. After following mode m for the duration d, the robot state is $[\chi(d), \delta(d), \psi(d)]^T \in [\chi(d), 0, 0]^T + \mathcal{E}(d)$ in the local coordinate system. We consider switching to a mode tracking a line with orientation ϕ_l with respect to the current direction, and origin $[\chi(d), 0]$ in the coordinate system of the specification σ_m. Let $\hat{\sigma}$ be the specification of this second mode. Then one can see that the composition is valid if

$$|\delta(d)| \leq 0.9 \text{ and } |\phi(d)| + |\phi_l| \leq \pi/3.$$

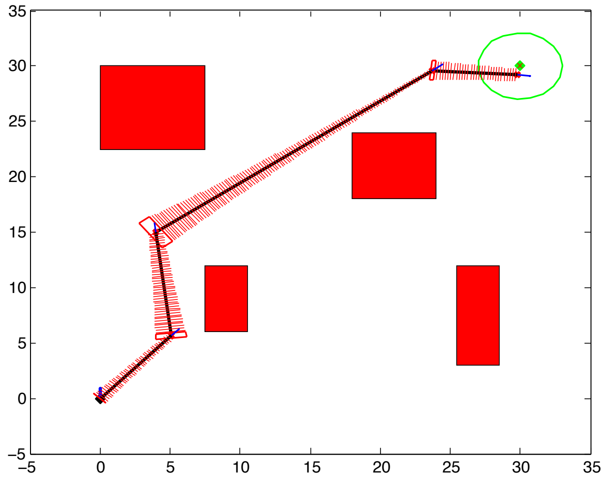

Fig. 2. Example of robust path with error tracking tubes. The vehicle starts from the lower left corner and must reach the encircled green zone with certainty.

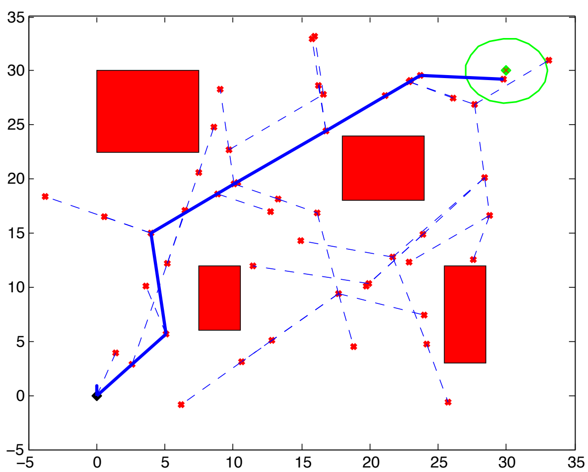

Fig. 3. RRT tree built to obtain the solution of Fig. 2.

Moreover, we can set

$$\hat{\mathcal{E}}(0) = \Big\{[\hat{\chi}, \hat{\delta}, \hat{\phi}] \Big| |\hat{\chi}(0)| \leq |\delta(d)|, |\hat{\delta}(0)| \leq |\delta(d)|, |\hat{\phi}(0)| \leq |\phi(d)| + |\phi_l| \Big\}$$

and use these values as $\epsilon_{\hat{\chi}}, \epsilon_\delta, \epsilon_\psi$. Fig. 2 and 3 show an example of composition of motion specifications reaching a desired goal set, obtained using the RRT algorithm of Section III and the specifications of this section.

V. Conclusion

This paper introduces a notion of robust motion specification, and details how to sequentially compose such specifications to establish the feasibility of a motion planning task, taking into account various sources of disturbance. In contrast to previously proposed motion description languages, the atoms do not specify explicit control inputs, i.e., how the controllers operate, but focus instead on what the controllers can achieve, in terms of robust tracking of reference trajectories. Many techniques from control theory, essentially relying on Lyapunov analysis, are available to prove that specific controllers satisfy the proposed notion of specification. The main directions for our future work are to establish a form of completeness guarantee for the planner described, and to integrate the approach with higher-level planning techniques based on temporal logics.

References

[1] R. W. Brockett, "Formal languages for motion description and map making," in *Robotics*, R. W. Brockett, Ed., vol. 41. Providence, RI: American Mathematical Society, 1990, pp. 181–193.

[2] V. Manikonda, P. S. Krishnaprasad, and J. Hendler, "Languages, behaviors, hybrid architectures and motion control," in *Mathematical Control Theory*, J. Bailleul and J. C. Willems, Eds. New-York: Springer, 1998, pp. 199–226.

[3] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transaction on Robotics*, vol. 21, no. 6, pp. 1077–1091, December 2005.

[4] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[5] L. Yang and S. M. LaValle, "The sampling-based neighborhood graph: A framework for planning and executing feedback motion strategies," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 419–432, June 2004.

[6] R. Tedrake, "LQR-trees: Feedback motion planning on sparse randomized trees," *Proceedings of Robotics: Science and Systems (RSS)*, 2009.

[7] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.

[8] E. D. Sontag and Y. Wang, "Notions of input to output stability," *Systems and Control Letters*, vol. 38, pp. 235–248, 1999.

[9] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[10] B. D. O. Anderson and J. Moore, *Optimal Control: Linear Quadratic Methods*. Dover, 2007, republication of the 1990 edition.

[11] E. D. Sontag, "Smooth stabilization implies coprime factorization," *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 435–443, April 1989.

[12] T. Schouwenaars, B. Mettler, E. Feron, and J. P. How, "Robust motion planning using a maneuver automaton with built-in uncertainties," in *Proceedings of the American Control Conference*, 2003, pp. 2211–2216.

[13] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance constrained stochastic predictive control," *IEEE Transaction on Robotics*, vol. 26, no. 3, pp. 502–517, June 2010.