Avoiding Saturation By Trajectory Reparameterization*

George J. Pappas
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720
gpappas@eecs.berkeley.edu
35th CDC

Abstract

The problem of trajectory tracking in the presence of input constraints is considered. The desired trajectory is reparameterized on a slower time scale in order to avoid input saturation. Necessary conditions that the reparameterizing function must satisfy are derived. The deviation from the nominal trajectory is minimized by formulating the problem as an optimal control problem.

Keywords: Input saturation, trajectory reparameterization, feedback linearization

1 Introduction

Input saturation is inherent in all physical systems. In particular, bounds on control inputs affect the stabilization of systems and their regions of attraction. Furthermore, the set of reference trajectories which can be feasibly tracked under input constraints is restricted.

In this paper, the problem of trajectory tracking in the presence of input constraints is considered. The main idea of this paper is to reparameterize infeasible trajectories on a different time scale in order to satisfy the input constraints. The system will traverse the same trajectory in output space but on a slower time scale. This idea will be illustrated on feedback linearizable systems although its applicability could be much wider.

The motivation for this work is the successful application of feedback linearization in flight control systems. Very frequently, Air Traffic Controllers order aircraft to be at a specific place at a specific time. However, such an order may be infeasible due to actuator limitations. In that scenario, on-line replanning of the reference trajectory is needed in order to follow the same trajectory but arrive at the destination point at a later time. The automation of the on-line trajectory replanning is an essential element of an automated *Air Traffic Management System* [1].

The problem of stabilization in the presence of input constraints goes at least as far back as [2]. In [3] it is shown that linear systems with poles in the right half plane can not be globally stabilized using bounded controls. In [2, 4], it is shown that for chains of integrators with more than two states, one must use nonlinear feedback in order to obtain global stabilization. For systems with poles on the

*Research supported by the Army Research Office under grant DAAH 04-95-1-0588

closed left half plane, under the additional assumption of null controllability, one can obtain either global results using nonlinear control laws ([5, 6]) or semi-global results using linear feedback ([7]). Similar results for discrete-time systems can be found in [8, 9]. Semi-global output regulation for linear, null controllable systems can be found in [10] and is also considered in [6].

Most of the previous results hold for linear systems. Stabilization results for a certain class of nonlinear systems (called *feed forward systems*) can be found in [6] while in [11] stabilization is considered for input-output linearizable systems. Most of the nonlinear control techniques for linear and nonlinear systems under saturation, utilize the concept of nested saturation feedback, introduced in [6].

An interesting idea which converts trajectory tracking to maneuver regulation was presented in [12]. The main idea of maneuver regulation is to obtain a manifold which is transverse to the desired trajectory and then stabilize the transverse dynamics. Although the formulation in [12] does not explicitly consider input constraints, it essentially reparameterizes the trajectory.

For trajectory planning and tracking problems, the hierarchical control structure displayed in Figure 1 is envisioned. At the trajectory planning level, trajectory generation is performed having higher level objectives in mind. For example, in ATMS, higher level objectives could be collision avoidance, optimality etc. *Input constraints are not considered at this level in order to reduce the complexity of the planning process.*

At the trajectory reparameterization level, the desired trajectory designed by the planner is reparameterized in order to become feasible given the input bounds. The reparameterized trajectory may then be checked by the planning layer to verify that it still satisfies the desired higher level objectives. In ATMS for example, if the initial trajectory is collision free it is desirable that the reparameterized trajectory is collision free as well. If the reparameterization algorithm fails, it will inform the trajectory planning layer to replan trajectories by explicitly considering input constraints.

Once a feasible trajectory has been designed, it is passed to the trajectory tracking layer which ensures that the system follows the reparameterized trajectory. If, due to disturbances, the tracking errors are sufficiently large, then saturation may still occur. The tracking level will then inform the reparameterization level to further rescale the desired trajectory in order to distribute more control authority to converging to the desired trajectory. If reparameterization fails in this case, then the planner is informed of the large tracking error and replans a new trajectory which considers the large deviation from the nominal performance.

The above control structure clearly reduces the workload of the trajectory planning layer which can focus on meeting high level objectives and considers saturation only when solutions at the trajectory reparameterization layer do not exist or when reparameterized trajectories do not meet the higher level objectives.

This paper focuses on the reparameterization module shown in Figure 1. In Section 2, after a quick review of feedback linearization of SISO systems, a necessary differential equation that the reparameterizing function must satisfy is obtained. Minimization between the actual and the reparameterized trajectories is posed as an optimal control problem. The effects of tracking errors and inverse dynamics are also considered in this section. Section 3, generalizes the concept to MIMO systems where a set of conditions that the reparameterizing function must satisfy are derived. Conclusions and issues for further research are discussed in Section 4.

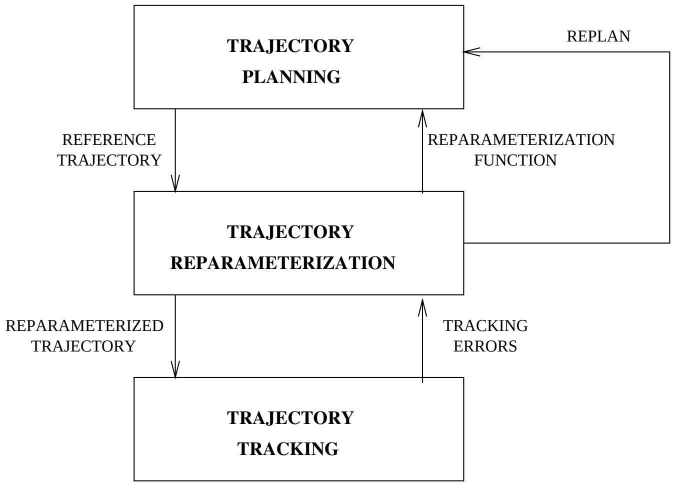

Figure 1: Trajectory Planning, Reparameterization and Tracking Structure

2 Single-Input Single-Output Systems

We start with a brief review of feedback linearization of single-input single-output nonlinear systems at the level of [13].

2.1 Feedback Linearization

Consider the single-input, single-output nonlinear system,

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{1}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}$, output $y \in \mathbb{R}$, vector fields f, g on $\mathbb{R}^n$ and output function h.[1] Without loss of generality, we assume that the origin is an equilibrium point of the autonomous system and that $h(0) = 0$. Let X be an open set containing the origin. The nonlinear system (1) has a well defined relative degree $\gamma \leq n$ at $0 \in X$ iff

$$\begin{aligned} L_g L_f^i h(x) &\equiv 0 \quad \forall x \in X, \;\; i = 0, \ldots, \gamma - 2 \\ L_g L_f^{\gamma-1} h(0) &\neq 0 \end{aligned} \tag{2}$$

Under conditions (2) there exists a local diffeomorphism, $\Phi : X \longrightarrow \mathbb{R}^n$, with $\Phi(0) = 0$,

$$\Phi : x \longrightarrow \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_\gamma \\ \eta \end{pmatrix} = \begin{pmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{\gamma-1} h(x) \\ \eta \end{pmatrix} \tag{3}$$

[1] Vector fields f,g and function h are assumed to be as smooth as needed.

such that system (1) is in the following normal form,

$$\begin{array}{rcl}\dot{z}_1 & = & z_2 \\ \dot{z}_2 & = & z_3 \\ & \vdots & \\ \dot{z}_\gamma & = & b(z,\eta) + a(z,\eta)u \\ \dot{\eta} & = & q(z,\eta) \\ y & = & z_1\end{array} \tag{4}$$

where $z = [z_1, \ldots, z_\gamma]^T$, $\eta \in \mathbb{R}^{n-\gamma}$ are the zero dynamics variables and

$$\begin{array}{rclcl}a(z,\eta) & = & L_g L_f^{\gamma-1} h(x) & = & L_g L_f^{\gamma-1} h(\Phi^{-1}(z,\eta)) \\ b(z,\eta) & = & L_f^\gamma h(x) & = & L_f^\gamma h(\Phi^{-1}(z,\eta))\end{array}$$

Therefore, if one chooses the following state feedback law

$$u = \frac{1}{a(z,\eta)}(-b(z,\eta) + v) = \frac{1}{L_g L_f^{\gamma-1} h(x)}(-L_f^\gamma h(x) + v) \tag{5}$$

then equations (4) take the following normal form,

$$\begin{array}{rcl}\dot{z}_1 & = & z_2 \\ \dot{z}_2 & = & z_3 \\ & \vdots & \\ \dot{z}_\gamma & = & v \\ \dot{\eta} & = & q(z,\eta) \\ y & = & z_1\end{array} \tag{6}$$

The system dynamics from the new external input v to output y is now linear (chain of γ integrators). The η dynamics have been made unobservable by the feedback law (5). The external input v is then designed to achieve stabilization and reference trajectory tracking.

2.2 Exact Tracking Without Inverse Dynamics

We first consider the case where the relative degree of system (1) is n and therefore the system is full state, feedback linearizable. In this case, the local diffeomorphism (3) is,

$$\Phi : x \longrightarrow \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} = \begin{pmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{pmatrix} \tag{7}$$

Let $Z = \Phi(X)$. Then $Z \subset \mathbb{R}^n$ is an open set and it contains the origin. The state feedback law (5) becomes,

$$u = \frac{1}{L_g L_f^{n-1} h(x)}(-L_f^n h(x) + v) = \frac{1}{a(z)}(-b(z) + v) \tag{8}$$

In exact output tracking we wish that the output $y(t)$ is exactly equal to some desired trajectory $r(t) \in C^n([0,t_f])$ where one may also choose $t_f = +\infty$. Let $z_d(t) = [r(t)\ \frac{dr}{dt}(t)\ \ldots\ \frac{d^{n-1}r}{dt^{n-1}}(t)]^T$. From now on, we will assume that the desired trajectory does not exit the neighborhood X where the assumptions of relative degree hold, that is for all time $t \in [0,t_f]$ we have $z_d(t) \in Z = \Phi(X)$. Exact output tracking requires that the initial conditions satisfy $z(0) = z_d(0)$ and that the control input (5) is

$$u = \frac{1}{a(z)}(-b(z) + v) = \frac{1}{a(z_d)}(-b(z_d) + \frac{d^n r}{dt^n}) \tag{9}$$

However, since $-M \leq u \leq M$ we must have [2]

$$-M \leq \frac{1}{a(z_d)}\ (-b(z_d) + \frac{d^n r}{dt^n}) \leq M \tag{10}$$

Without loss of generality we may assume that $a(z_d) > 0$. Then equation (10) implies that,

$$-M \cdot a(z_d) + b(z_d) \leq \frac{d^n r}{dt^n} \leq M \cdot a(z_d) + b(z_d) \tag{11}$$

or otherwise

$$\frac{d^n r}{dt^n} \in [-M \cdot a(z_d) + b(z_d)\ ,\ M \cdot a(z_d) + b(z_d)] \tag{12}$$

In order for a desired trajectory to be feasible it must satisfy necessary condition (12). A more geometric description of condition (12) may be found in [14]. Since many reference trajectories may not satisfy this necessary condition, we ask whether we can satisfy necessary condition (12) by reparameterizing the desired trajectory $r(t)$ to $r(s(t))$. Note that the system still traverses the same trajectory in output space but on a different time scale.

The reparameterizing function $s(t)$ will be meaningful only if we place certain constraints on $s(t)$. First, since the desired trajectories $r(t)$ are functions belonging in $C^n([0,t_f])$ it is clear that we must have $0 \leq s(t) \leq t_f$. Second, in order to avoid going backwards in time, $s(t)$ must satisfy $\frac{ds}{dt} \geq 0$ for all t. Therefore, valid reparameterizing functions will be assumed to satisfy the following conditions:

$$0 \leq s(t) \leq t_f \tag{13}$$

$$\frac{ds}{dt} \geq 0 \tag{14}$$

Note, that $s(t)$ is allowed to be greater than t. This may result in acausal control laws where the system looks ahead at the trajectory and tries to exploit situations where input saturation does not occur. If acausal control laws are not desired, then $s(t)$ must satisfy $0 \leq s(t) \leq t$.

The key idea in this work is that the external input $v = \frac{d^n r}{dt^n}$ can, in some sense, *help* the input u not to saturate. We can control the external input $\frac{d^n r}{dt^n}$ by reparameterizing the desired output trajectory to $r(s(t))$. The strategy is then to choose the reparameterizing function $s(t)$ which satisfies conditions (13,14) so that equation (12) is satisfied. If the reparameterizing function $s(t)$ does not satisfy conditions (13,14), then the problem of avoiding saturation by trajectory reparameterization is not solvable by a valid reparameterization function.

[2] In this paper we will assume symmetric input constraints. This can obviously be generalized to nonsymmetric constraints by replacing $-M$ with M_1 and M with M_2, assuming however that $0 \in [M_1, M_2]$

We now derive necessary conditions that the reparameterizing function $s(t)$ must satisfy. Consider the reparameterized trajectory $r(s(t))$. Simple calculations show that,

$$\begin{aligned}
\frac{dr}{dt} &= r^{(1)}\frac{ds}{dt} \\
\frac{d^2r}{dt^2} &= r^{(2)}(\frac{ds}{dt})^2 + r^{(1)}\frac{d^2s}{dt^2} \\
\frac{d^3r}{dt^3} &= r^{(3)}(\frac{ds}{dt})^3 + 3r^{(2)}\frac{d^2s}{dt^2}\frac{ds}{dt} + r^{(1)}\frac{d^3s}{dt^3}
\end{aligned} \tag{15}$$

where $(\cdot)^{(i)} = \frac{d^i(\cdot)}{ds^i}$. In general we can see that

$$\frac{d^nr}{dt^n} = T(s, \frac{ds}{dt}, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) + r^{(1)}\frac{d^ns}{dt^n} \tag{16}$$

where the nature of the nonlinear function T depends on the relative degree of the system as well as the desired trajectory. Plugging equation (16) into equation (12) we obtain,

$$T(s, \frac{ds}{dt}, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) + r^{(1)}\frac{d^ns}{dt^n} \in [-M \cdot a(z_d) + b(z_d)\ ,\ M \cdot a(z_d) + b(z_d)] \tag{17}$$

or otherwise

$$r^{(1)}\frac{d^ns}{dt^n} \in [-M \cdot a(z_d) + b(z_d) - T, M \cdot a(z_d) + b(z_d) - T] \tag{18}$$

Recall that $z_d(t) = [r(t)\ \frac{dr}{dt}(t)\ \ldots\ \frac{d^{n-1}r}{dt^{n-1}}(t)]^T$. From equation (16) it is clear that $z_d(s(t))$ is a function of $s, \frac{ds}{dt}, \ldots, \frac{d^{n-1}s}{dt^{n-1}}$. Defining

$$\begin{aligned}
A(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) &= a(z_d(s(t))) \\
B(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) &= b(z_d(s(t))) \\
S_1(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) &= -M \cdot A(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) + B(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) - T(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) \qquad (19) \\
S_2(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) &= M \cdot A(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) + B(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) - T(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) \qquad (20)
\end{aligned}$$

results in the following necessary condition,

$$r^{(1)}\frac{d^ns}{dt^n} \in [S_1(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}})\ ,\ S_2(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}})] \tag{21}$$

Therefore given a feedback linearizable system, input bounds and a desired output trajectory $r(t)$, one chooses the reparameterizing function $s(t)$ so that necessary condition (21) is satisfied. We now consider two cases:

Case I : $r^{(1)} \neq 0$ Without loss of generality we can assume that $r^{(1)} > 0$. In this case, the necessary condition (21) can be written as

$$\frac{d^ns}{dt^n} \in [\frac{1}{r^{(1)}}S_1(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}})\ ,\ \frac{1}{r^{(1)}}S_2(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}})] \tag{22}$$

If we define

$$\begin{aligned}
s_1 &= s \\
s_2 &= \frac{ds}{dt} \\
&\vdots \\
s_{n-1} &= \frac{d^{n-2}s}{dt^{n-2}} \\
s_n &= \frac{d^{n-1}s}{dt^{n-1}}
\end{aligned}$$

then necessary condition (22) requires that the reparameterizing function $s(t)$ satisfies the following differential inclusion

$$\begin{aligned}
\dot{s}_1 &= s_2 \\
\dot{s}_2 &= s_3 \\
&\vdots \\
\dot{s}_{n-1} &= s_n \\
\dot{s}_n &\in [\frac{1}{r^{(1)}}S_1(s_1,\ldots,s_n)\ ,\ \frac{1}{r^{(1)}}S_2(s_1,\ldots,s_n)]
\end{aligned} \tag{23}$$

or equivalently the following controlled differential equation

$$\begin{aligned}
\dot{s}_1 &= s_2 \\
\dot{s}_2 &= s_3 \\
&\vdots \\
\dot{s}_{n-1} &= s_n \\
\dot{s}_n &= w
\end{aligned} \tag{24}$$

where the control w must take values in the interval W defined as

$$W = [\frac{1}{r^{(1)}}S_1(s_1,\ldots,s_n)\ ,\ \frac{1}{r^{(1)}}S_2(s_1,\ldots,s_n)] \tag{25}$$

In order to ensure that reparameterization does not affect the differentiability of the trajectory, the initial conditions are chosen so that $z_d(0) = z_d(s(0))$. From equation (15) we obtain

$$\begin{aligned}
s_1(0) &= s(0) = 0 \\
s_2(0) &= \frac{ds}{dt}(0) = 1 \\
s_3(0) &= \frac{d^2s}{dt^2}(0) = 0 \\
&\vdots \\
s_n(0) &= \frac{d^{n-1}s}{dt^{n-1}}(0) = 0
\end{aligned} \tag{26}$$

which clearly satisfy conditions (13,14). If $z_d(0) = z_d(s(0))$ is not required, then one could choose any initial conditions which satisfy (13,14).

Note that $s(t)$ satisfies a controlled differential equation which implies that there may be many solutions for $s(t)$ which avoid input saturation. Any $s(t)$ which satisfies equation (23) will in fact avoid saturation. However, the question now becomes whether the system will reach its final goal, namely whether we can steer s_1 to the desired value t_f. Furthermore, any valid solution to the above problem must also satisfy conditions (13,14). But condition (14) along with $s_1(0) = 0$ imply that condition (13) will be satisfied since once $s_1 = t_f$, the problem is solved. Therefore the only real constraint is (14). Hence, existence of a valid reparameterization function can be shown by solving the following problem.

Problem 1 *Given the controlled differential equation (24), initial conditions (26) and the control interval W defined in (25), choose the control $w \in W$ so as to steer s_1 to t_f while satisfying the constraint $s_2 \geq 0$.*

In case there are many solutions which satisfy the above constraints, we choose the reparameterizing function $s(t)$ which steers s_1 to t_f in minimum time. Other optimization costs could be considered as well. The optimal reparameterization function is obtained by solving the following optimal control problem.

Problem 2 *Given the controlled differential equation (24), initial conditions (26) and the control interval W defined in (25), choose the control $w \in W$ so as to steer s_1 to t_f in minimum time while satisfying the constraint $s_2 \geq 0$.*

Note that in the above optimal control problem, the system is linear, the cost is convex and the search space is convex. The final time is free, but the final state is constrained. The above control problem is difficult because of the state dependent control interval W as well as the state constraint. The nonlinearities of the system (1) and the desired trajectory $r(t)$ are all hidden in the control interval W. Because of the choice of optimization cost and since W is a bounded and state dependent interval, solutions to the above optimal control problem will be of bang-bang nature. Furthermore, solutions will be numerical except for simple examples like the following one.

Example: Consider the system

$$\begin{aligned} \dot{x} &= u \\ y &= x \end{aligned}$$

where $|u| \leq 1$ and consider the desired trajectory $r(t) = 5t$ with $t \in [0, 2]$. By reparameterizing $r(t)$ to $r(s(t))$ we obtain that

$$u = r^{(1)} \frac{ds}{dt} = 5 \frac{ds}{dt}$$

and therefore the reparameterizing function must satisfy the necessary condition

$$\frac{ds}{dt} \in [-\frac{1}{5}, \frac{1}{5}]$$

Clearly, any $s(t)$ with $\frac{ds}{dt} \in (0, \frac{1}{5}]$ will avoid saturation, eventually reach the desired value $s = 2$, and satisfy the state constraint (14). The redundancy is now resolved by choosing the function $s(t)$ which reaches $s = 2$ in minimum time. It is clear that in this example, the minimum value is achieved by choosing $\frac{ds}{dt} = \frac{1}{5}$ in which case $s(10) = 2$. This results in the reparameterizing function $s(t) = \frac{1}{5}t$ and thus the reparameterized trajectory is $r(s(t)) = 5\frac{1}{5}t = t$ for $t \in [0, 10]$. $\square$

Case II: $r^{(1)} = 0$ In the degenerate case where the first derivative of the reference trajectory vanishes, necessary condition (21) becomes

$$0 \in [S_1(s_1, \ldots, s_n) \ , \ S_2(s_1, \ldots, s_n)] \tag{27}$$

which implies that $\frac{d^n s}{dt^n}$ is unconstrained as long as condition (27) is satisfied. If condition (27) is not satisfied when $r^{(1)} = 0$ then saturation cannot be avoided by reparameterization. Notice that in this case the idea of reparameterization is meaningless since the desired trajectory freezes and thus reparameterization serves no purpose. Two cases must be considered. The first one occurs when $r^{(1)}(s) = 0$ at some particular point s, and the second when $r^{(1)}(s) \equiv 0$ for all s in some interval S which has nonzero measure.

In the first case where $r^{(1)}(s) = 0$ at some particular point s it is assumed that condition (27) is satisfied for $s_1 = s$. But then, by smoothness of solutions, we obtain that (27) is satisfied for $s_1 = s_-$ as well as for $s_1 = s_+$. Furthermore $r^{(1)}(s_+) \neq 0$ and we can therefore proceed by solving again the optimal control problem 2 after s_+. We can therefore jump over points where the reference trajectory freezes momentarily as long as condition (27) is satisfied. If condition (27) is satisfied then the control input w in the differential equation (24) is essentially unconstrained and we could choose $w = w_-$ in order to preserve continuity of the control input w.

We now consider the case where $r^{(1)}(s) \equiv 0$ for all $s(t) \in [\sigma_1, \sigma_2]$ or equivalently for all $t \in [t_1, t_2]$. It is clear in this case that all higher derivatives $r^{(2)}, r^{(3)}, \ldots, r^{(n)}$ are equal to zero on the same interval. From equations (15,16), we have

$$\frac{d^n r}{dt^n} = T(s, \frac{ds}{dt}, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) + r^{(1)} \frac{d^n s}{dt^n} = 0 \tag{28}$$

In this case, condition (27) simplifies in the following manner

$$\begin{aligned}
0 &\in [S_1(s_1, \ldots, s_n) \ , \ S_2(s_1, \ldots, s_n)] \Rightarrow \\
-B + T &\in [-M \cdot A \ , \ M \cdot A] \Rightarrow \\
\frac{1}{A}(-B + T) &\in [-M \ , \ M] \Rightarrow \\
\frac{1}{A}(-B) &\in [-M \ , \ M] \Rightarrow \\
\frac{1}{a(z_d)}(-b(z_d)) &\in [-M \ , \ M]
\end{aligned} \tag{29}$$

Condition (29) can also be obtained from the control law (9) after substituting $\frac{d^n r}{dt^n} = 0$. The meaning of equation (29) is that the trajectory $r(t)$ should not freeze at a place where the cost of feedback linearization is big enough to saturate the control.

If $r^{(1)}(s) = 0$ for all $s(t) \in [\sigma_1, \sigma_2]$, then by equations (16,15) it is clear that

$$\begin{aligned}
r(s(t)) &= r(t) \ \text{ for } \ t \in [t_1, t_2] \\
\frac{dr}{dt} &= r^{(1)} \frac{ds}{dt} = 0 \\
&\vdots \\
\frac{d^n r}{dt^n} &= T(s, \frac{ds}{dt}, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) + r^{(1)} \frac{d^n s}{dt^n} = 0
\end{aligned}$$

and therefore the state will be at the point $[r(s), \overbrace{0, 0, \ldots, 0}^{n-1}]^T$ until $s = \sigma_2$ or equivalently $t = t_2$. The nominal trajectory $r(t)$ may be also stationary (if $t < t_2$) or may be nonstationary (if $t \geq t_2$). Given that condition (29) is satisfied, we can take advantage of the fact that the trajectory is frozen for some time by reseting the reparameterizing function $s(t)$ in the following way

$$s(t) = \begin{cases} t & \text{if } t < t_2 \\ t_2 & \text{if } t \geq t_2 \end{cases} \tag{30}$$

Therefore if $t < t_2$, then the nominal and the reparameterized trajectories are both frozen at the same location in state space. Then we can simply set $s(t)$ equal to t in order to catch up to the nominal time parameterization. Notice that although this may result in $s(t)$ being discontinuous, the state as well as the output trajectories still remain smooth since $[r(s(t)), 0, 0, \ldots, 0]^T = [r(t), 0, 0, \ldots, 0]^T$. Furthermore, this minimizes deviation from the original plan.

If, however, $t \geq t_2$ then the nominal trajectory $r(t)$ is no longer stationary and therefore the best the reparameterized trajectory can do is catch up to the last point where $r(t)$ was stationary. Therefore, in this case we set $s(t) = t_2$. In other words, the reparameterized trajectory will not remain stationary and will skip the entire time interval where the nominal trajectory was stationary.

When $t > t_2$, then $r^{(1)} \neq 0$ and we can proceed by solving again the optimal control problem 2. The above strategy is shown in the following example.

Example: Consider the system,

$$\begin{aligned} \dot{x} &= u \\ y &= x \end{aligned}$$

where $-1 \leq u \leq 1$ and consider the desired trajectory

$$r(t) = \begin{cases} 2t & \text{if } 0 \leq t \leq 1 \\ 2 & \text{if } 1 < t \leq 4 \\ t-2 & \text{if } t > 4 \end{cases} \tag{31}$$

Although the trajectory $r(t)$ is nonsmooth, it is used to illustrate the idea. Clearly, any smooth approximation to $r(t)$ will do the job as well. By reparameterizing $r(t)$ to $r(s(t))$ we obtain that

$$u = r^{(1)} \frac{ds}{dt} = 2\frac{ds}{dt}$$

and therefore the reparameterizing function must satisfy the necessary condition

$$\frac{ds}{dt} \in [-\frac{1}{2}, \frac{1}{2}]$$

and therefore the reparameterizing function $s(t) = \frac{t}{2}$ will minimize deviation from the nominal plan. This will hold as long as $0 \leq s(t) \leq 1$ or equivalently $0 \leq t \leq 2$ When $t = 2$ (or $s(t) = 1$) then $r^{(1)}(s) = 0$. Notice that in this case condition (29) becomes

$$0 \in [-1, 1]$$

which is trivially satisfied. Since the nominal trajectory $r(t)$ will be stationary for all $t \in [1, 4]$ we can set $s(t) = t$ from $t = 2$ until $t = 4$. At $t = 4$ we return back to the previous optimal control

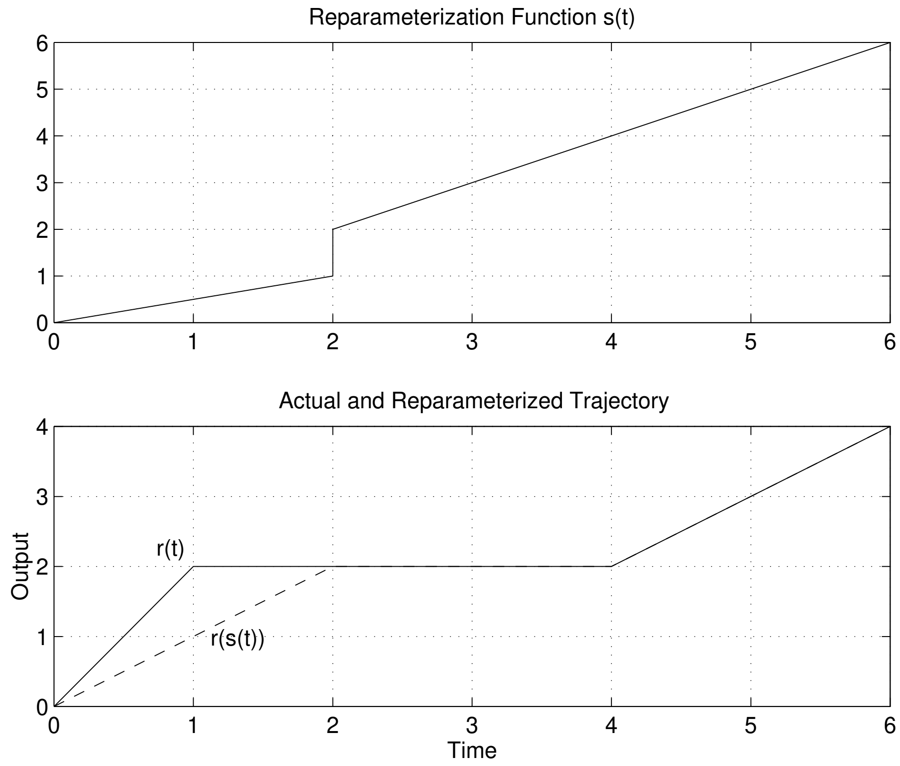

Figure 2: Reparameterization of a Stationary Trajectory

calculations to show that the trajectory does not need reparameterization since it is feasible. The overall reparameterizing function is

$$s(t) = \begin{cases} \frac{t}{2} & \text{if } 0 \leq t \leq 2 \\ t & \text{if } t > 2 \end{cases} \tag{32}$$

and the reparameterized trajectory is

$$r(s(t)) = \begin{cases} t & \text{if } 0 \leq t \leq 2 \\ 2 & \text{if } 2 < t \leq 4 \\ t-2 & \text{if } t > 4 \end{cases} \tag{33}$$

As can be seen from above as well as Figure 2, the reparameterized trajectory stayed at the stationary point $r(t) = 2$ one time unit less than the nominal one in order to catch up with it.

2.3 Exact Tracking With Inverse Dynamics

In the case where the relative degree of system (1) is $\gamma < n$ much of the story remains the same. The only difference is that the inverse dynamics must now be appended in our previous formulation.

Exact output tracking requires that the initial conditions satisfy $z(0) = z_d(0)$ and that the control input (5) is

$$u = \frac{1}{a(z,\eta)}(-b(z,\eta) + v) = \frac{1}{a(z_d,\eta)}(-b(z_d,\eta) + \frac{d^n r}{dt^n}) \tag{34}$$

However, since $-M \leq u \leq M$ we must have

$$-M \leq \frac{1}{a(z_d, \eta)} \left(-b(z_d, \eta) + \frac{d^n r}{dt^n}\right) \leq M \tag{35}$$

Without loss of generality we may assume that $a(z_d) > 0$. Then,

$$-M \cdot a(z_d, \eta) + b(z_d, \eta) \leq \frac{d^n r}{dt^n} \leq M \cdot a(z_d, \eta) + b(z_d, \eta) \tag{36}$$

or otherwise

$$\frac{d^n r}{dt^n} \in [-M \cdot a(z_d, \eta) + b(z_d, \eta)\ ,\ M \cdot a(z_d, \eta) + b(z_d, \eta)] \tag{37}$$

Recall that after reparameterization

$$\frac{d^n r}{dt^n} \quad = \quad T(s, \frac{ds}{dt}, \ldots, \frac{d^{n-1} s}{dt^{n-1}}) + r^{(1)} \frac{d^n s}{dt^n} \tag{38}$$

and therefore we obtain,

$$T(s, \frac{ds}{dt}, \ldots, \frac{d^{n-1} s}{dt^{n-1}}) + r^{(1)} \frac{d^n s}{dt^n} \in [-M \cdot a(z_d, \eta) + b(z_d, \eta)\ ,\ M \cdot a(z_d, \eta) + b(z_d, \eta)] \tag{39}$$

or otherwise

$$r^{(1)} \frac{d^n s}{dt^n} \in [-M \cdot a(z_d, \eta) + b(z_d, \eta) - T, M \cdot a(z_d, \eta) + b(z_d, \eta) - T] \tag{40}$$

We now define

$$\begin{aligned}
A(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) &= a(z_d(s(t)), \eta) \\
B(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) &= b(z_d(s(t)), \eta) \\
S_1(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) &= -M \cdot A(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) + B(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) - T(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}) \qquad (41) \\
S_2(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) &= M \cdot A(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) + B(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) - T(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}})] \qquad (42)
\end{aligned}$$

in order to obtain the following necessary condition,

$$r^{(1)} \frac{d^n s}{dt^n} \in [S_1(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta)\ ,\ S_2(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta) \tag{43}$$

We now consider the case where $r^{(1)} \neq 0$. The case where $r^{(1)} = 0$ can be considered in a similar manner as before. Without loss of generality we can assume that $r^{(1)} > 0$. In this case, the necessary condition (43) can be written as

$$\frac{d^n s}{dt^n} \in [\frac{1}{r^{(1)}} S_1(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta)\ ,\ \frac{1}{r^{(1)}} S_2(s, \ldots, \frac{d^{n-1} s}{dt^{n-1}}, \eta)] \tag{44}$$

Therefore necessary condition (44) requires that the reparameterizing function $s(t)$ to satisfy the following controlled differential equation

$$\begin{aligned}
\dot{s}_1 &= s_2 \\
\dot{s}_2 &= s_3 \\
&\vdots \\
\dot{s}_{n-1} &= s_n \\
\dot{s}_n &= w \\
\dot{\eta} &= q(\eta, z_d) = q(\eta, s_1, \ldots, s_n)
\end{aligned} \tag{45}$$

where the control w must take values in the interval W defined as

$$W = [\frac{1}{r^{(1)}} S_1(s_1, \ldots, s_n, \eta) \ , \ \frac{1}{r^{(1)}} S_2(s_1, \ldots, s_n, \eta)] \tag{46}$$

From this point on, the formulation of the problem is exactly as it would be in the absence of inverse dynamics. We can again formulate the steering problem whose solution will show existence of a reparameterizing function as well as the optimal control problem in order to minimize deviation from the nominal plan. We can also have similar strategies in cases where the reference trajectory becomes stationary.

The only added difficulty in this problem is the appending of the inverse dynamics in our formulation. Note that the inverse dynamics must be well behaved. One has to ensure that z_d, which is essentially an input to the inverse dynamics, does not steer the zero dynamics and the state out of X, the region where feedback linearization is valid. For sufficient conditions which ensure this problem does not happen, see [14].

2.4 Asymptotic Tracking Without Inverse Dynamics

We consider using reparameterization to avoid saturation in asymptotically tracking a reference trajectory. In case the tracking error is large, reparameterization will slow down the desired trajectory and distribute more control authority to stabilizing the tracking error. Note, however, that since the feedback law for trajectory tracking using feedback linearization contains a linear function of the error, sufficiently large tracking errors will saturate the control and, in that case, the trajectory needs to be replaned.

In asymptotic tracking we have $z(0) \neq z_d(0)$. Therefore, the following change of coordinates

$$\begin{aligned}
e_1 &= z_1 - r \\
e_2 &= z_2 - \frac{dr}{dt} \\
&\vdots \\
e_n &= z_n - \frac{d^n r}{dt^n}
\end{aligned}$$

or more compactly $e = z - z_d$ results in

$$\begin{aligned}
\dot{e}_1 &= e_2 \\
\dot{e}_2 &= e_3
\end{aligned}$$

$$\begin{aligned} &\vdots \\ \dot{e}_{n-1} &= e_n \\ \dot{e}_n &= b(z) + a(z)u - \frac{d^n r}{dt^n} \end{aligned} \tag{47}$$

If $s^n + c_{n-1}s^{n-1} + \ldots + c_1 s + c_0$ is a Hurwitz polynomial then the following control law

$$u = \frac{1}{a(z)}(-b(z) - c_0 e_1 - c_1 e_2 - \ldots - c_{n-1} e_n + \frac{d^n r}{dt^n}) = \frac{1}{a(z)}(-b(z) - c^T e + \frac{d^n r}{dt^n}) \tag{48}$$

will ensure that the error e converges to zero exponentially. However, since $-M \leq u \leq M$ we must ensure that the input bounds are not violated. Therefore we must have

$$\begin{aligned} -M \leq \frac{1}{a(z)}(-b(z) - c^T e + \frac{d^n r}{dt^n}) \leq M \Rightarrow \\ -M \leq \frac{1}{a(z_d + e)}(-b(z_d + e) - c^T e + \frac{d^n r}{dt^n}) \leq M \end{aligned}$$

Notice that we must ensure that $z \in \Phi(X)$. This may potentially place restrictions on how large the initial error e can be. For sufficient conditions on the initial error which guarantee that $z \in \Phi(X)$ see [14].

We will again use a reperameterization algorithm in order to avoid saturation. The only difference in this case is the addition of the tracking error. Using equation (16),

$$\begin{aligned} A(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) &= a(z_d + e) \\ B(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) &= b(z_d + e) \\ S_1(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) &= -M \cdot A(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) + B(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) + c^T e - T(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) \\ S_2(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) &= M \cdot A(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) + B(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) + c^T e - T(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}) \end{aligned}$$

and assuming without loss of generality that $a(z_d + e) > 0$, results in the following necessary condition,

$$r^{(1)} \frac{d^n s}{dt^n} \in [S_1(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e) \ , \ S_2(s, \ldots, \frac{d^{n-1}s}{dt^{n-1}}, e)] \tag{49}$$

As in the exact tracking case, we can consider cases where the trajectory becomes stationary or not. Problems similar to Problems 1 and 2 can be formulated by simply appending the error dynamics to the formulation just like we appended the inverse dynamics in the previous section. Although the problems are more difficult in this case, they are conceptually the same. The main difference is the dependence of the reparameterization function on the tracking error. This allows the system to reparameterize the trajectory depending on how large the tracking error is. Therefore, if the tracking error is large then the system distributes more control authority towards trajectory tracking by slowing down the trajectory. Once the tracking error converges to zero, the above results reduce to the exact tracking results. The system will then be able to distribute more control authority towards speeding up the trajectory in order to catch up with the nominal one.

The case of asymptotic tracking in the presence of inverse dynamics can be handled in a similar way. In this more general scenario, one needs to append both the error and the inverse dynamics in formulating problems similar to Problems 1 and 2.

3 Extension to Multi-Input Multi-Output Systems

We would like to extend the reparameterization story obtained in the last section to MIMO nonlinear systems. Although we will only consider performing exact tracking without inverse dynamics, it is clear from the SISO derivations how to incorporate both error and inverse dynamics in our formulation.

3.1 Feedback Linearization for MIMO Systems

We first review feedback linearization for MIMO systems ([13]). Consider the square (i.e. number of inputs is equal to the number of outputs) Multi-Input Multi-Output (MIMO) nonlinear control system described by

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x) u_i \\ y_j &= h_j(x) \quad j = 1, \cdots, m \end{aligned} \tag{50}$$

where $x \in \mathbb{R}^n$, $f(x), g_1(x), \cdots, g_m(x)$ are smooth vector fields on $\mathbb{R}^n$ and $h_1(x), \cdots, h_m(x)$ smooth functions on $\mathbb{R}^n$. For convenience, these equations are written in vector notation as

$$\begin{aligned} \dot{x} &= f(x) + g(x) u \\ y &= h(x) \end{aligned}$$

We will assume that x_0 is an equilibrium point of the autonomous system, that is $f(x_0) = 0$, and (without loss of generality) that $h(x_0) = 0$.

We *assume* in what follows that each output y_j has a well defined *relative degree* γ_j, in an open neighborhood X of x_0, i.e. there exists an integer γ_j such that

$$L_{g_i} L_f^l h_j(x) \equiv 0 \quad \forall \ l < \gamma_j - 1, \ \ \forall \ 1 \leq i \leq m, \ \ \forall \, x \in U$$

Collecting these calculations, we have

$$\begin{array}{ccccccc} \begin{bmatrix} y_1^{\gamma_1} \\ y_2^{\gamma_2} \\ \vdots \\ y_m^{\gamma_m} \end{bmatrix} & = & \begin{bmatrix} L_f^{\gamma_1} h_1(x) \\ L_f^{\gamma_2} h_2(x) \\ \vdots \\ L_f^{\gamma_m} h_m(x) \end{bmatrix} & + & \begin{bmatrix} L_{g_1} L_f^{\gamma_1 - 1} h_1(x) & \cdots & L_{g_m} L_f^{\gamma_1 - 1} h_1(x) \\ L_{g_1} L_f^{\gamma_2 - 1} h_2(x) & \cdots & L_{g_m} L_f^{\gamma_2 - 1} h_2(x) \\ \cdots & \cdots & \cdots \\ L_{g_1} L_f^{\gamma_m - 1} h_m(x) & \cdots & L_{g_m} L_f^{\gamma_m - 1} h_m(x) \end{bmatrix} u \\ & =: & d(x) & + & D(x)\, u \end{array} \tag{51}$$

Matrix $D(x)$ is called the decoupling matrix. If $D(x)$ is invertible at every point in X, then the *static state feedback* given by

$$u = (D(x))^{-1} \left[\, -d(x) + v \, \right] \tag{52}$$

will result in a closed loop system which is decoupled from input v to output y. This decoupled and input-output linearized system is given by

$$\begin{bmatrix} y_1^{\gamma_1} \\ y_2^{\gamma_2} \\ \vdots \\ y_m^{\gamma_m} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \tag{53}$$

In the case the nonlinear system has vector relative degree $[\gamma_1, \ldots, \gamma_m]^T$ with $\sum_{i=1}^m \gamma_i \leq n$ then by choosing the following coordinates

$$\begin{aligned} z_1^i &= h_i(x) \\ z_2^i &= L_f h_i(x) \\ &\vdots \\ z_{\gamma_i}^i &= L_f^{\gamma_i - 1} h_i(x) \end{aligned}$$

for $1 \leq i \leq m$ we get the following normal form

$$\begin{aligned} \dot{z}_1^i &= z_2^i \\ &\vdots \\ \dot{z}_{\gamma_i}^i &= v_i \end{aligned}$$

for $1 \leq i \leq m$. In addition to the m chains of integrators, we obtain the $(n - \sum_{i=1}^m \gamma_i)$-dimensional unobservable dynamics

$$\dot{\eta} = Q(\eta, z) + P(\eta, z)u \tag{54}$$

where z stands for the states of the m integrator chains

$$z = \begin{bmatrix} z_1^1 \\ \vdots \\ z_{\gamma_1}^1 \\ \vdots \\ z_1^m \\ \vdots \\ z_{\gamma_m}^m \end{bmatrix} \tag{55}$$

The local diffeomorphism, $\Phi : X \longrightarrow \mathbb{R}^n$, in this case is

$$\Phi : x \longrightarrow \begin{bmatrix} z \\ \eta \end{bmatrix} = \begin{bmatrix} z_1^1 \\ \vdots \\ z_{\gamma_1}^1 \\ \vdots \\ z_1^m \\ \vdots \\ z_{\gamma_m}^m \\ \eta \end{bmatrix} = \begin{bmatrix} h_1(x) \\ \vdots \\ L_f^{\gamma_1 - 1} h_1(x) \\ \vdots \\ h_m(x) \\ \vdots \\ L_f^{\gamma_m - 1} h_m(x) \\ \eta \end{bmatrix} \tag{56}$$

If the matrix $D(x)$ is *singular*, we cannot use a static state feedback to decouple the nonlinear system ([13]), and we have to look for dynamic state feedback to achieve input-output decoupling.

3.2 MIMO Exact Tracking

In case of input saturation, we add the following constraints to the system equations (50):

$$|u_i| \leq M_i \qquad M_i > 0, \quad i = 1, \cdots, m \tag{57}$$

If we define $\Delta(x) = \delta_{ij}(x) = (A(x))^{-1}$ as the inverse of the decoupling matrix and the vector $\beta(x)$ as

$$\begin{bmatrix} \beta_1(x) \\ \vdots \\ \beta_m(x) \end{bmatrix} = \Delta(x) \cdot d(x) = \Delta(x) \cdot \begin{bmatrix} d_1(x) \\ \vdots \\ d_m(x) \end{bmatrix} \tag{58}$$

then the feedback linearizing control law (52) has to satisfy

$$|\beta_i(x) + \Delta(x) v_i| \leq M_i \tag{59}$$

for $1 \leq i \leq m$. In order for the outputs y_i to exactly track the reference trajectories $r_i(t)$, the external inputs v_i are chosen as

$$\begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} \frac{d^{\gamma_1} r_1}{dt^{\gamma_1}} \\ \vdots \\ \frac{d^{\gamma_m} r_m}{dt^{\gamma_m}} \end{bmatrix} \tag{60}$$

Notice that we have transformed the MIMO problem into m decoupled SISO problems. This was done by implicitly assuming that $\Delta(x)$ is known either analytically or numerically. Combining the last two equations results in the following set of necessary conditions for the reference trajectories,

$$-M_i \leq \beta_i(x) + \sum_{j=1}^{m} \delta_{ij}(x) \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} \leq M_i \tag{61}$$

or otherwise,

$$\sum_{j=1}^{m} \delta_{ij}(x) \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} \in [-M_i - \beta_i(x), M_i - \beta_i(x)] \tag{62}$$

Condition (62) is a necessary condition that must be satisfied by all output reference trajectories r_i. If it is not satisfied for any i, we again ask ourselves whether we could satisfy the above condition by reparameterizing time to $s(t)$. Notice that in order to traverse the same trajectory in output space, the reparameterizing function $s(t)$ must be the *same* for all outputs r_i. This will obviously increase the number of conditions that the reparameterization function must satisfy as well as the difficulty of solving the problem.

Recall that by reparameterizing $r_i(t)$ to $r_i(s(t))$, the time derivatives of the outputs are of the form,

$$\frac{d^k r_i}{dt^k} = T_i(s, \frac{ds}{dt}, \ldots, \frac{d^{k-1}s}{dt^{k-1}}) + r_i^{(1)} \cdot \frac{d^k s}{dt^k} \tag{63}$$

In the case where the system is full state feedback linearizable, we obtain from equation (56) that $x = \Phi^{-1}(z)$. Since the bundled state z is a function of the reference trajectories and their derivatives, by equations (56, 63) it is clear that the state x depends on $s, \frac{ds}{dt}, \ldots, \frac{d^{\gamma_i} s}{dt^{\gamma_i}}$ for $1 \leq i \leq m$. Thus, necessary conditions (62) become

$$\sum_{j=1}^{m} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} \in [-M_i - \beta_i, M_i - \beta_i] \tag{64}$$

where now every term in condition (64) is a function of $s, \frac{ds}{dt}, \ldots, \frac{d^{\gamma_i} s}{dt^{\gamma_i}}$ for $1 \leq i \leq m$. In the SISO case, the equivalent condition is condition (12). Notice that in the SISO case, the highest derivative of the reference trajectory must satisfy condition (12). For MIMO systems, because of the decoupling matrix, it is no longer clear how to extract the highest derivative from conditions (64) since many of the δ_{ij} may be zero and furthermore many of the outputs may have the same relative degree.

Consider condition (64) for a fixed index $i \in \{1, \ldots, m\}$. Let γ_i^* be the largest derivative degree appearing in (64). Define the set index K_i as

$$K_i = \{1 \leq j \leq m \mid \gamma_j = \gamma_i^* \text{ and } \delta_{ij} \neq 0\} \tag{65}$$

We can then break up the following sum

$$\begin{aligned}
\sum_{j=1}^{m} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} &= \sum_{j \in K_i} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} + \sum_{j \notin K_i} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} \\
&= \sum_{j \in K_i} \delta_{ij} \frac{d^{\gamma_i^*} r_j}{dt^{\gamma_i^*}} + \sum_{j \notin K_i} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} \\
&= \left(\sum_{j \in K_i} \delta_{ij} \right) \frac{d^{\gamma_i^*} r_j}{dt^{\gamma_i^*}} + \sum_{j \notin K_i} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}}
\end{aligned} \tag{66}$$

Using equation (63) we can furthermore obtain

$$\begin{aligned}
\sum_{j=1}^{m} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} &= \left(\sum_{j \in K_i} \delta_{ij} \right) \frac{d^{\gamma_i^*} r_j}{dt^{\gamma_i^*}} + \sum_{j \notin K_i} \delta_{ij} \frac{d^{\gamma_j} r_j}{dt^{\gamma_j}} \\
&= \left(\sum_{j \in K_i} \delta_{ij} \right) \left(T_j(s, \ldots, \frac{d^{\gamma_i^*-1} s}{dt^{\gamma_i^*-1}}) + r_j^{(1)} \frac{d^{\gamma_i^*} s}{dt^{\gamma_i^*}} \right) \\
&\quad + \sum_{j \notin K_i} \delta_{ij} \left(T_j(s, \ldots, \frac{d^{\gamma_j-1} s}{dt^{\gamma_j-1}}) + r_j^{(1)} \frac{d^{\gamma_j} s}{dt^{\gamma_j}} \right) \\
&= \left(\sum_{j \in K_i} \delta_{ij} r_j^{(1)} \right) \frac{d^{\gamma_i^*} s}{dt^{\gamma_i^*}} + \sum_{j \notin K_i} \delta_{ij} r_j^{(1)} \frac{d^{\gamma_j} s}{dt^{\gamma_j}} + \sum_{j=1}^{m} \delta_{ij} T_j(s, \ldots, \frac{d^{\gamma_j-1} s}{dt^{\gamma_j-1}})
\end{aligned} \tag{67}$$

If we define the function Γ_i as

$$\Gamma_i = \sum_{j \notin K_i} \delta_{ij} r_j^{(1)} \frac{d^{\gamma_j} s}{dt^{\gamma_j}} + \sum_{j=1}^{m} \delta_{ij} T_j(s, \ldots, \frac{d^{\gamma_j-1} s}{dt^{\gamma_j-1}}) \tag{68}$$

then necessary condition (64) can be written as

$$\left(\sum_{j \in K_i} \delta_{ij} r_j^{(1)} \right) \frac{d^{\gamma_i^*} s}{dt^{\gamma_i^*}} \in [-M_i - \beta_i - \Gamma_i, M_i - \beta_i - \Gamma_i] \tag{69}$$

Notice that necessary condition (69) must be satisfied for each $1 \leq i \leq m$. If $\left(\sum_{j\in K_i} \delta_{ij} r_j^{(1)}\right) \neq 0$ then we can assume without loss of generality that

$$\left(\sum_{j\in K_i} \delta_{ij} r_j^{(1)}\right) > 0$$

and therefore condition (69) can be equivalently expressed as

$$\frac{d^{\gamma_i^*} s}{dt^{\gamma_i^*}} \in S_i \tag{70}$$

where the interval S_i is defined as

$$S_i = [\frac{1}{\left(\sum_{j\in K_i} \delta_{ij} r_j^{(1)}\right)}(-M_i - \beta_i - \Gamma_i), \frac{1}{\left(\sum_{j\in K_i} \delta_{ij} r_j^{(1)}\right)}(M_i - \beta_i - \Gamma_i)] \tag{71}$$

If, however, $\left(\sum_{j\in K_i} \delta_{ij} r_j^{(1)}\right) = 0$, then it is clear from condition (69) that in this case the trajectory must satisfy the condition

$$0 \in [-M_i - \beta_i - \Gamma_i, M_i - \beta_i - \Gamma_i] \tag{72}$$

Therefore, if condition (72) is satisfied then $\frac{d^{\gamma^*} s}{dt^{\gamma^*}}$ is unconstrained and thus

$$\frac{d^{\gamma_i^*} s}{dt^{\gamma_i^*}} \in \mathbb{R} \tag{73}$$

If condition (72) is not satisfied then the necessary condition is not satisfied and thus

$$\frac{d^{\gamma_i^*} s}{dt^{\gamma_i^*}} \in \emptyset \tag{74}$$

Grouping the above we can see that for each $1 \leq i \leq m$ we have that

$$\frac{d^{\gamma_i^*} s}{dt^{\gamma_i^*}} \in W_i = \begin{cases} S_i & \text{if } \sum_{j\in K_i} \delta_{ij} r_j^{(1)} \neq 0 \\ \mathbb{R} & \text{if } \sum_{j\in K_i} \delta_{ij} r_j^{(1)} = 0 \text{ and condition (72) is satisfied} \\ \emptyset & \text{otherwise} \end{cases} \tag{75}$$

It might be the case, however, that for $i_1 \neq i_2$ we have $\gamma_{i_1}^* = \gamma_{i_2}^*$. In that case condition (75) results in two constraints on $\gamma_{i_1}^*$-th derivative of $s(t)$. In that case we must have

$$\frac{d^{\gamma_{i_1}^*} s}{dt^{\gamma_{i_1}^*}} = \frac{d^{\gamma_{i_2}^*} s}{dt^{\gamma_{i_2}^*}} \in W_{i_1} \bigcap W_{i_2}$$

Motivated by the above we define the index sets

$$N_k = \{1 \leq j \leq m \mid \gamma_j^* = k\} \tag{76}$$

The index sets N_k are essentially a partition of the index set $\{1, \ldots, m\}$ under the following equivalence relation: indices i_1 and i_2 are equivalent if $\gamma_{i_1}^* = \gamma_{i_2}^*$. We can then group conditions (75) and thus the k-th derivative of $s(t)$ must satisfy the following set of constraints

$$\frac{d^k s}{dt^k} \in V_k = \bigcap_{i\in N_k} W_i \tag{77}$$

Conditions (77) are the conditions that the reparameterizing function must satisfy. Notice that the problem has no solution if the set V_k for any k is the empty set. This may happen in two ways. First, any of the W_i may be empty. This means that reparameterization cannot avoid saturation in the input-output channel i. Second, the intersection of the W_i may be empty which means that it is impossible to avoid saturation in all input channels simultaneously by reparameterization.

Let k^* be the highest number k for which N_k is nonempty. In other words, k^* is the degree of the highest derivative of $s(t)$ that is constrained in conditions (77). If we define,

$$\begin{aligned} s_1 &= s \\ &\vdots \\ s_{k^*} &= \frac{d^{k^*-1}s}{dt^{k^*-1}} \end{aligned}$$

then the reparameterizing function $s(t)$ must satisfy the following controlled differential equation

$$\begin{aligned} \dot{s}_1 &= s_2 \\ \dot{s}_2 &= s_3 \\ &\vdots \\ \dot{s}_{k^*} &= w \end{aligned} \tag{78}$$

as well as the state constraints

$$\begin{aligned} s_2 &\in V_1 \\ s_3 &\in V_2 \\ &\vdots \\ s_{k^*} &\in V_{k-1} \end{aligned} \tag{79}$$

in addition to the previous state constraint (14)

$$s_2 \geq 0 \tag{80}$$

and the control must be in $w \in V_k$. Existence of a reparameterizing solution can thus be shown by solving the following problem.

Problem 3 *Given the controlled differential equation (78), choose the control $w \in V_k$ in order to steer s_1 to t_f while satisfying state constraints (79,80).*

In case of nonunique solution, redundancy is resolved by solving the following optimal control problem.

Problem 4 *Given the controlled differential equation (78), choose the control $w \in V_k$ in order to steer s_1 to t_f in minimum time while satisfying state constraints (79,80).*

Note that the initial conditions must also satisfy state constraints (79,80). This may result in initial conditions different from (26), which will result in loss of smoothness. In that case, however, the initial conditions can be treated as parameters in which case the optimization can be performed over the initial conditions as well. The above optimal control problem may be very difficult to solve, even numerically, since we are trying to satisfy many constraints by varying only one parameter. The complexity of the above formulation should be expected since we want to avoid saturation in many input output channels by altering one parameter. Furthermore, solutions may not exist. The concepts of the above theory is shown in the following simple example.

Example: Consider the system

$$\begin{aligned}\ddot{x}_1 &= u_1\\ \dot{x}_2 &= u_2\\ y_1 &= x_1\\ y_2 &= x_2\end{aligned}$$

where $|u_1| \le 1$, $|u_2| \le 1$ and consider the desired trajectories $r_1(t) = 4t$, $r_2(t) = 2t$ for $t \in [0, 2]$. By reparameterizing $r_1(t)$ and $r_2(t)$ respectively to $r_1(s(t))$ and $r_2(s(t))$ respectively we obtain that

$$\begin{aligned}u_1 &= r_1^{(1)}\frac{d^2 s}{dt^2} + r_1^{(2)}\left(\frac{ds}{dt}\right)^2 = 4\frac{d^2 s}{dt^2}\\ u_2 &= r_2^{(1)}\frac{ds}{dt} = 2\frac{ds}{dt}\end{aligned}$$

and therefore the reparameterizing function must satisfy the necessary conditions,

$$\begin{aligned}\frac{d^2 s}{dt^2} &\in W_1 = [-\frac{1}{4}, \frac{1}{4}]\\ \frac{ds}{dt} &\in W_2 = [-\frac{1}{2}, \frac{1}{2}]\end{aligned}$$

The optimal control problem can be now formulated. Given

$$\begin{aligned}\dot{s}_1 &= s_2\\ \dot{s}_2 &= w\end{aligned}$$

with $w \in W_1 = [-\frac{1}{4}, \frac{1}{4}]$, steer to $s_1 = 2$ in minimum time subject to the state constraint $s_2 \in (0, \frac{1}{2}]$. Note that $s_2(0)$ cannot be equal to 1 in this case but, on the other hand, one could choose any $s_2(0) \in (0, \frac{1}{2}]$. □

The complexity of Problems 3 and 4 reduces significantly in the special case where all the outputs have the same relative degree γ^*. Then the set $\{1, \dots, m\}$ is essentially partitioned in only one equivalence class, namely N_{γ^*}, and as a result all of necessary conditions (77) constrain $s(t)$ at the γ^*-th derivative. In that case, state constraints (79) reduce to the following single constraint

$$\frac{d^{\gamma^*} s}{dt^{\gamma^*}} \in V = \bigcap_{i=1}^{m} W_i \tag{81}$$

and Problems 3 and 4 simplify to the following.

Problem 5 *Given the controlled differential equation,*

$$\begin{aligned}\dot{s}_1 &= s_2\\ \dot{s}_2 &= s_3\\ &\vdots\\ \dot{s}_{\gamma^*-1} &= w\end{aligned}$$

choose the control

$$w \in V = \bigcap_{i=1}^{m} W_i$$

in order to steer s_1 to t_f and satisfy state constraint $s_2 \ge 0$.

In case of nonunique solution, redundancy is resolved by solving the following optimal control problem.

Problem 6 *Given the controlled differential equation,*

$$\begin{aligned} \dot{s}_1 &= s_2 \\ \dot{s}_2 &= s_3 \\ &\vdots \\ \dot{s}_{\gamma^*-1} &= w \end{aligned}$$

choose the control

$$w \in V = \bigcap_{i=1}^{m} W_i$$

in order to steer s_1 *to* t_f *in minimum time and satisfy state constraint* $s_2 \geq 0$.

The above scenario is illustrated in the following example.

Example: Consider the system

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ y_1 &= x_1 \\ y_2 &= x_2 \end{aligned}$$

where $|u_1| \leq 1$, $|u_2| \leq 1$ and consider the desired trajectories $r_1(t) = 4t$, $r_2(t) = 2t$ for $t \in [0, 2]$. By reparameterizing $r_1(t)$ and $r_2(t)$ respectively to $r_1(s(t))$ and $r_2(s(t))$ respectively we obtain that

$$\begin{aligned} u_1 = r_1^{(1)} \frac{ds}{dt} = 4\frac{ds}{dt} \\ u_2 = r_2^{(1)} \frac{ds}{dt} = 2\frac{ds}{dt} \end{aligned}$$

and therefore the reparameterizing function must satisfy the necessary conditions,

$$\begin{aligned} \frac{ds}{dt} \in W_1 = [-\frac{1}{4}, \frac{1}{4}] \\ \frac{ds}{dt} \in W_2 = [-\frac{1}{2}, \frac{1}{2}] \end{aligned}$$

and therefore

$$\frac{ds}{dt} \in W_1 \bigcap W_2 = [-\frac{1}{4}, \frac{1}{4}]$$

It is clear, that in this case the optimal reparametization function is $s(t) = \frac{1}{4}t$ and $s(8) = 2$. Thus the reparameterized trajectories are $r_1(s(t)) = 4\frac{1}{4}t = t$ and $r_2(s(t)) = 2\frac{1}{4}t = \frac{1}{2}t$ for $t \in [0, 8]$. $\square$

4 Conclusions

The problem of reparameterizing desired output trajectories in order to avoid input saturation was considered. Necessary conditions that the reparameterizing function must satisfy were derived for SISO and MIMO feedback linearizable systems. The deviation from the nominal trajectory was minimized by formulating an optimal control problem.

The formulation of the optimal control problem raises several issues which will be looked upon in the near future such as conditions under which existence of solutions is guaranteed, structure of optimal control laws etc. Furthermore, an interesting issue is to examine the effect of inverse dynamics and tracking errors on the existence of solutions. Necessary conditions for other classes of systems and feedback control laws will be investigated in order to show the generailty of the main idea.

Acknowledgment: The author would like to thank John Lygeros, Datta Godbole and John-Morten Godhavn for many interesting discussions on this subject as well as his advisor, Prof. S.S. Sastry, for being a continuous source of inspiration.

References

[1] S. Sastry, G. Meyer, C. Tomlin, J. Lygeros, D. Godbole, and G. Pappas. Hybrid control in air traffic management systems. In *Proceedings of the 1995 IEEE Conference in Decision and Control*, pages 1478–83, New Orleans, LA, December 1995.

[2] Fuller A.T. In-the-large stability of relay and saturating control systems with linear controller. *International Journal of Control*, 10:457–480, 1969.

[3] Sontag E.D. An algebraic approach to bounded controllability of linear systems. *International Journal of Control*, 39(1):181–188, 1984.

[4] Sontag E.D. and Yang Y. On the stabilizability of multiple integrators by means of bounded feedback controls. In *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 70–72, 1991.

[5] Sontag E.D. and H.J. Sussmann. Nonlinear output feedback design for linear systems with saturating controls. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 3414–3416, 1990.

[6] A. R. Teel. *Feedback Stabilization: Nonlinear Solutions to Inherently Nonlinear Problems.* PhD thesis, Department of Electrical Engineering, University of California, Berkeley, California, 1992.

[7] Lin Z. and Saberi A. Semi-global exponential stabilization of linear systems subject to input saturation via linear feedbacks. *Systems & Control Letters*, 21(3):225–239, 1993.

[8] Y. Yang. *Global stabilization of linear systems with bounded feedback.* PhD thesis, Rutgers University, New Brunswick, New Jersey, 1993.

[9] Lin Z. and Saberi A. Semi-global exponential stabilization of linear discrete-time systems subject to input saturation via linear feedbacks. *Systems & Control Letters*, 24:125–132, 1995.

[10] Stoorvogel A.A. Lin Z. and Saberi A. Output regulation for linear systems subject to input saturation. In *Proceedings of the 33th IEEE Conference on Decision and Control*, pages 2362–2367, 1994.

[11] Teel A.R. and Praly L. Tools for semi-global stabilization by partial state and output feedback. *SIAM Journal of Control and Optimization*, 33(5):1443–88, September 1995.

[12] John Hauser and Rick Hindman. Maneuver regulation from trajectory tracking: Feedback linearizable systems. In *Proceedings of NOLCOS '95*, Lake Tahoe, CA, June 1995.

[13] Alberto Isidori. *Nonlinear Control Systems*. Communications and Control Engineering Series. Springer Verlag, 1989.

[14] George J. Pappas, John Lygeros, and Datta N. Godbole. Stabilization and tracking of feedback linearizable systems under input constraints. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 596–601, New Orleans, LA, December 1995.