Model Predictive Task-Priority Control using Control Lyapunov Functions

Eirik L. Foseid [†], Erlend A. Basso [†], Henrik M. Schmidt-Didlaukies [†],
Kristin Y. Pettersen [†], Jan Tommy Gravdahl [†]

Abstract— **Redundant robotic systems allow for the simultaneous execution of multiple tasks. It is often desirable to have priority between these tasks, such that lower priority tasks do not interfere with higher priority tasks. Many existing methods for task-priority control, while providing stability and strict priority between tasks, do not take into account the overall performance of the closed-loop system. This paper presents an optimization-based framework for dynamic task-priority control of redundant robotic systems, providing strict priority between tasks while improving the performance of the closed-loop system relative to a user-defined performance metric. The method is based on using a nominal dynamic task-priority control law together with a hierarchical control Lyapunov function based model-predictive control method. The proposed method is validated in simulation on a redundant robotic system, where it is shown to provide improved performance over existing dynamic task-priority control methods.**

I. Introduction

Many robotic systems, including all vehicle-manipulator systems, are redundant systems. Robotic systems are said to be *redundant* if they have more degrees of freedom than what is minimally required to achieve an objective. Redundancy can be exploited in order to satisfy multiple objectives simultaneously. Objectives, or tasks, are often divided into groups according to their priority, such as safety-critical, mission-related or optimization tasks. Clearly, an optimization task should not interfere with the execution of a higher-priority mission-related task. As a consequence, the redundancy resolution algorithm must ensure strict prioritization between different sets of tasks.

Redundancy resolution algorithms ensuring strict-priority between tasks are known as *task-priority controllers*. Introduced in [1], improved in [2] and generalized to an arbitrary number of tasks in [3], task-priority control has been widely studied. While these task-priority methods decouple the system, generating velocity or acceleration references to be tracked by a dynamic controller, the operational space formulation, introduced in [4], and further developed and extended in [5]–[7], directly assigns generalized joint torques solving the redundancy resolution. Through the use of null-space operators [8], [9], these methods achieve prioritization between tasks such that lower-priority tasks do not interfere with higher-priority tasks. At the kinematic level, this ensures that lower-priority tasks do not generate task-velocities in higher-priority tasks. And at the dynamic level, this ensures that lower-priority tasks do not generate task-accelerations, or generalized task-forces, in the task-space of higher-priority tasks. This property is of particular importance in cases where some tasks are safety-critical while others are not.

Developed in works such as [10], [11], the *Control Lyapunov Function* (CLF) has become a successful generalization of the more than a century-old method of Lyapunov. While a systematic way of finding CLFs does not exist, the knowledge of one enables the design stabilizing controllers for nonlinear systems by choosing controllers that satisfy certain conditions on the time derivative of the CLF. One such controller, the pointwise min-norm controller, can be found by pointwise selecting the value that minimizes the norm of the control input [12]. The design of this controller can be seen as an optimization problem where the cost is the norm of the control input together with a constraint on the time derivative of the CLF. A closed-form solution for this optimization problem exists [12]. There also exist closed-form solutions of extensions of this problem where the cost is also a function of the state, satisfying certain properties [13].

When constructed as an optimization problem, constraints on the time derivative of multiple CLFs can be added to solve multiple tasks simultaneously [14], but without any priority between tasks. If strict priority between tasks is required, the method proposed in [15] achieves strict priority between tasks by solving a hierarchy of quadratic programs where the solution for the optimal control input is incrementally refined to solve multiple tasks while constraining the solutions to not interfere with higher-priority tasks.

The CLF-based methods can choose between a large number of feasible control inputs, but typically one is chosen by minimizing a cost function pointwise in time. However, this choice of control input might not be optimal when we consider the same cost function integrated over a time horizon. Receding horizon control, or model predictive control, employs a model of the system to predict the future behaviour of the closed-loop system, both to achieve stability, and improve performance. The method proposed by [13] combines model predictive control, to yield improved performance with respect to a cost function over a time horizon, with CLFs to guarantee stability.

In this paper we present a framework for dynamic task-priority control of redundant robotic systems based on CLFs and model predictive control. This framework allows for an arbitrary number of tasks spread over an arbitrary number of

* This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, through the ERC Advanced Grant 101017697-CRÈME. The work is also supported by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 – NTNU AMOS and project No. 304667.

[†]Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: `eirik.l.foseid@ntnu.no`

priority levels, with strict priority between tasks at different priority levels. Moreover, it uses ideas from model predictive control to improve the performance of the closed-loop system.

This paper is organized as follows. Section II presents relevant background material. Section III presents the main contributions of the paper. Section IV presents the design of a nominal task-priority control law required in the method proposed in Section III. Section V presents our results from a simulation study where we apply the method to a redundant robotic system. Finally, Section VI presents our conclusions.

II. Background Material

This section presents background material relevant to the remaining parts of the paper. We start by presenting control Lyapunov functions (CLFs) and corresponding pointwise min-norm control laws. We then give a brief overview of the receding horizon CLF-based method from [13], that combines CLFs with ideas from optimal control to improve the performance of a nominal control law. Finally, we present the hierarchical dynamic task-prioritization method using CLFs from [15] that solves redundancy resolution, dynamic control and control allocation simultaneously.

A. Control Lyapunov Functions

Consider a nonlinear control affine system

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where f, g are locally Lipschitz, $x \in D \subset \mathbb{R}^n$, $u \in U \subset \mathbb{R}^p$ is the set of admissable control inputs, and D and U are closed. We define a control Lyapunov function similarly to [16] as a continously differentiable candidate Lyapunov function $V(x)$ satisfying

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \tag{2a}$$

$$\inf_{u\in U} \left[L_f V(x) + L_g V(x)u + \sigma(x)\right] < 0,\ \forall x \neq 0 \tag{2b}$$

where α_1 and α_2 are $\mathcal{K}_\infty$ functions, and σ is a continuous positive definite function. That is, a CLF is a candidate Lyapunov function for which the derivative can be made negative pointwise by a choice of control input. Assuming we have a CLF, solving the optimization problem

$$\begin{aligned} \min_{u\in U} \quad & u^T u \\ \text{s.t.}\ & L_f V(x) + L_g V(x)u + \sigma(x) \leq 0 \end{aligned} \tag{3}$$

yields a pointwise min-norm stabilizing controller. By defining

$$\psi_0(x) := L_f V(x) + \sigma(x) \tag{4}$$

$$\psi_1(x) := \left[L_g V(x)\right]^{\mathrm{T}} \tag{5}$$

the optimization problem (3) can be solved [12], yielding the min-norm controller

$$u(x) = \begin{cases} -\frac{\psi_0(x)\psi_1(x)}{\psi_1^{\mathrm{T}}(x)\psi_1(x)} & \text{when } \psi_0(x) > 0 \\ 0 & \text{when } \psi_0(x) \leq 0 \end{cases} \tag{6}$$

B. Receding Horizon Control using Control Lyapunov Functions

Since, by definition, the min-norm control law (6) only minimizes the cost pointwise, we might want to improve the solution by solving an optimization problem over a time horizon. One method building on this idea is proposed in [13]. This method starts by simulating the system following the min-norm control law forward in time over a time horizon from t to $t+T$. Then, a receding horizon control problem is solved to improve this solution. The receding horizon control problem proposed by [13] is given by

$$\begin{aligned} \inf_{u(\cdot)} \quad & \int_t^{t+T} \left(q(x(\tau)) + u(\tau)^T u(\tau)\right) d\tau \\ \text{s.t.} \quad & \dot{x}(\tau) = f(x(\tau)) + g(x(\tau))u(\tau), \forall \tau \in [t, t+T] \\ & u(\tau) \in U, \forall \tau \in [t, t+T] \\ & L_f V(x(t)) + L_g V(x(t))u(t) + \sigma(x) \leq 0 \\ & V(x(t+T)) \leq V\left(x^*(t+T)\right) \end{aligned} \tag{7}$$

where $q(x)$ is a continuously differentiable and positive semidefinite cost function, $V(x(t+T)) \leq V\left(x^*(t+T)\right)$ is a terminal constraint on the value of the CLF at the end of the time horizon, and $x^*(\cdot)$ is the state along the trajectory generated by following the min-norm controller.

C. Task Prioritization through Control Lyapunov Functions

For the nonlinear control affine system in (1) we define a *task* as a function

$$y = h(x) \in \mathbb{R}^m \tag{8}$$

that we want to drive to zero. The input-output dynamics for such a task becomes

$$y^{(\rho)} = L_f^\rho h(x) + L_g L_f^{\rho-1} h(x) u \tag{9}$$

under the assumption that

$$\begin{aligned} L_g L_f^k h(x) = 0, \quad & 0 \leq k \leq \rho - 2 \\ L_g L_f^{\rho-1} h(x) \neq 0 & \end{aligned} \tag{10}$$

Furthermore, by defining the transverse states $\eta = \operatorname{col}\left(y, \dot{y}, \ldots, y^{(\rho-1)}\right) \in X \subset \mathbb{R}^{\rho m}$, and the internal dynamic states $z \in Z \subset \mathbb{R}^{n-\rho m}$ the dynamics of the system (1) can be expressed as

$$\begin{aligned} \dot{\eta} &= \bar{f}(\eta, z) + \bar{g}(\eta, z)u \\ \dot{z} &= f_z(\eta, z) \end{aligned} \tag{11}$$

Assuming we have a CLF $V(\eta)$ that satisfies

$$\inf_{u\in U} \left[L_f V(\eta, z) + L_g V(\eta, z)u + \sigma(\eta)\right] < 0 \tag{12}$$

for all (η, z) such that $\eta \neq 0$, we can solve a similar optimization problem to (3)

$$\begin{aligned} \min_{u\in U} \quad & u^T u \\ \text{s.t.} \quad & L_{\bar{f}} V(\eta, z) + L_{\bar{g}} V(\eta, z)u + \sigma(\eta) \leq 0 \end{aligned} \tag{13}$$

to get a min-norm stabilizing controller for the system (11). A method for task-priority control based on CLFs is found in [15]. It considers the case where we have a system on the form (1) and $Z = \sum_{i=1}^{k} Z_i$ total tasks spread over k priority levels with Z_i tasks at each priority level $i = 1, 2, \ldots, k$. Each task, defined as an output $y_i = h_i(x) \in \mathbb{R}^{m_i}$ for $i = 1, 2, \ldots, Z$ that we want to regulate to zero, has a corresponding CLF V_i. We define the priority index $p \in \{1, 2, \ldots, k\}$, and the number of tasks up to and including the priority level p as $\bar{Z}_p = Z_1 + Z_2 + \ldots + Z_p$. Given a priority level p, we define a set of indices of all the tasks on all priority levels up to, and including, the current priority level p as $\mathbb{I}_p = \{1, 2, \ldots, \bar{Z}_p\}$. We also define the set of indices for the tasks at the next priority level $p+1$ as $\mathbb{J}_p = \{\bar{Z}_p + 1, \bar{Z}_p + 2, \ldots, \bar{Z}_{p+1}\}$. The method starts by first solving the optimization problem

$$\begin{aligned} \min_{u \in U} \quad & u^T u + \sum_{j \in \mathbb{J}_1} \nu_j \delta_j^2 \\ \text{s.t.} \quad & L_{\bar{f}_i} V_i(\eta_i, z_i) + L_{\bar{g}_i} V_i(\eta_i, z_i) u + \sigma_i(\eta_i) \leq 0 \quad \forall i \in \mathbb{I}_1 \\ & L_{\bar{f}_j} V_j(\eta_j, z_j) + L_{\bar{g}_j} V_j(\eta_j, z_j) u + \sigma_j(\eta_j) \leq \delta_j \quad \forall j \in \mathbb{J}_1 \end{aligned} \tag{14}$$

where $\delta_j > 0$ are slack variables for the CLF constraints for tasks at priority level $p = 2$, and $\nu_j > 0$ are constant weights for the slack variables. This optimization problem solves for a min-norm stabilizing input that satisfies (12) for the Z_1 tasks at priority level $p = 1$, while attempting to solve the tasks at the next priority level by including a slacked version of the constraint. Assuming the system is redundant with respect to the Z_p tasks at priority level p, we can then use the ideas from [15] to further refine the solution to also consider the set of tasks at the next priority level. To do this we, iteratively, for $p = 2, 3, \ldots, k-1$, solve the optimization problem

$$\begin{aligned} \min_{u \in U} \quad & u^T u + \sum_{j \in \mathbb{J}_p} \nu_j \delta_j^2 \\ \text{s.t.} \quad & L_{\bar{g}_i} V_i u \leq L_{\bar{g}_i} V_i u_{p-1}^* \quad \forall i \in \mathbb{I}_p \\ & L_{\bar{f}_j} V_j + L_{\bar{g}_i} V_j u + \sigma_j(\eta_j) \leq \delta_j \quad \forall j \in \mathbb{J}_p \end{aligned} \tag{15}$$

where u_{p-1}^* is the optimal control input given by the optimization problem for the previous priority level $p-1$. This effectively allows for strict prioritization between tasks at different priority levels due to the constraint

$$L_{\bar{g}_i} V_i u \leq L_{\bar{g}_i} V_i u_{p-1}^* \quad \forall i \in \mathbb{I}_p \tag{16}$$

only allowing the tasks at priority level p to be solved if it is possible without interfering with tasks at priority level $p-1$. It also allows for soft prioritization between tasks at the same priority level through the weights $\{\nu_j\}_{j \in \mathbb{J}_p}$ on the slack variables. See the original article [15] for more details.

III. Model Predictive Dynamic Task-Priority Control

This section presents the main results of the paper, a model predictive control-based method for dynamic task-priority control, building on the ideas behind the method for receding horizon control using CLFs found in [13] and the CLF-based task prioritization framework found in [15].

Assume we have a system model on the form (1) and a set of Z tasks with corresponding CLFs distributed over k priority levels as described in Section II-C. Additionally we define the set of indices for all the tasks as $\mathbb{T} = \{1, 2, \ldots, Z\}$. Assume the existence of a (potentially suboptimal) nominal task-priority control law $u = \kappa(x)$. At time t we use the system model and the nominal control law to simulate the system forward in time over the interval $[t, t+T]$, where $T > 0$ is the prediction horizon. We then solve the optimization problem

$$\begin{aligned} & \min_{u(\cdot)} \quad \int_t^{t+T} \left(c_1\left(x(\tau), u(\tau)\right) + \sum_{j \in \mathbb{J}_p} \nu_j \delta_j^2(\tau) \right) d\tau \\ & \text{s.t.} \\ & \dot{x}(\tau) = f(x(\tau)) + g(x(\tau)) u(\tau), \forall \tau \in [t, t+T] \\ & u(\tau) \in U, \forall \tau \in [t, t+T] \\ & L_{\bar{f}_i} V_i\left(\eta_i(t), z_i(t)\right) + L_{\bar{g}_i} V_i\left(\eta_i(t), z_i(t)\right) u(t) + \sigma_i \leq 0, \forall i \in \mathbb{I}_1 \\ & L_{\bar{f}_j} V_j\left(\eta_j(t), z_j(t)\right) + L_{\bar{g}_j} V_j\left(\eta_j(t), z_j(t)\right) u(t) + \sigma_j \leq \delta_j, \forall j \in \mathbb{J}_1 \\ & V_l\left(\eta_l(t+T), z_l(t+T)\right) \leq V_l\left(\eta_{l,0}^*(t+T), z_{l,0}^*(t+T)\right), \forall l \in \mathbb{T} \end{aligned} \tag{17}$$

where $c_1(x, u)$ is a continuously differentiable and positive semi-definite cost function, $\eta_{l,0}^*(t+T)$ and $z_{l,0}^*(t+T)$ are the values of transverse and internal dynamic states for the task with index l at time $t+T$ following the nominal control law $u = \kappa(x)$ over the interval $[t, t+T]$. Now, let $u_1^*(\cdot)$ and $x_1^*(\cdot)$ be the control input and the state along the optimal trajectory found by solving (17). We then, similarly to the method from [15] presented in section II-C, try to further refine the solution to also consider the set of tasks at the next priority level. This is done by iteratively, for $p = 2, 3, \ldots, k-1$, solving the optimization problem

$$\begin{aligned} & \min_{u(\cdot)} \quad \int_t^{t+T} \left(c_p\left(x(\tau), u(\tau)\right) + \sum_{j \in \mathbb{J}_p} \nu_j \delta_j^2(\tau) \right) d\tau \\ & \text{s.t.} \\ & \dot{x}(\tau) = f(x(\tau)) + g(x(\tau)) u(\tau), \forall \tau \in [t, t+T] \\ & u(\tau) \in U, \forall \tau \in [t, t+T] \\ & L_{\bar{g}_i} V_i\left(\eta_i(t), z_i(t)\right) u(t) \leq L_{\bar{g}_i} V_i\left(\eta_i(t), z_i(t)\right) u_{p-1}^*(t) \quad \forall i \in \mathbb{I}_p \\ & L_{\bar{f}_j} V_j\left(\eta_j(t), z_j(t)\right) + L_{\bar{g}_j} V_j\left(\eta_j(t), z_j(t)\right) u(t) + \sigma_j \leq \delta_j \quad \forall j \in \mathbb{J}_p \\ & V_l(\eta_l(t+T), z_l(t+T)) \leq V_l(\eta_{l,p-1}^*(t+T), z_{l,p-1}^*(t+T)) \quad \forall l \in \mathbb{T} \end{aligned} \tag{18}$$

where $c_p(x, u)$ is a continuously differentiable and positive semi-definite cost function, $u_{p-1}^*(\cdot)$, $\eta_{l,p-1}^*(\cdot)$ and $z_{l,p-1}^*(\cdot)$ now are the optimal control input, transverse and internal dynamic states along the optimal trajectory found by solving the previous optimization problem for priority level $p-1$. This scheme can be seen as a receding horizon variation of the pointwise optimization problems (14)-(15), but with some modifications: We are constraining the solution to satisfy the CLF condition, but only for the first time step. Additionally, a terminal constraint on the Lyapunov function value for each task is added. This constrains the solutions to those where the value of the Lyapunov function for each task at time $t+T$ is smaller what is would be following the nominal control law $u = \kappa(x)$. The proposed method is summarized in Algorithm 1.

Algorithm 1 Hierarchical Model Predictive Dynamic Task-Priority Control using Control Lyapunov Functions

```
INPUT: $t$, $T$, $x(t)$, $\kappa(x)$, $\{c_p(x,u)\}_{p\in\{1,\dots,k\}}$, $\{\nu_i\}_{i\in\mathbb{T}}$, $\{V_i\}_{i\in\mathbb{T}}$
OUTPUT: $u^*_{k-1}$
  Simulate the system forward in time over $[t, t+T]$ using the nominal control law $\kappa(x)$ to obtain $\{\eta^*_{l,0}(t+T)\}_{l\in\mathbb{T}}$ and $\{z^*_{l,0}(t+T)\}_{l\in\mathbb{T}}$
  Solve (17) to obtain $u^*_1$, $\{\eta^*_{l,1}(t+T)\}_{l\in\mathbb{T}}$ and $\{z^*_{l,1}(t+T)\}_{l\in\mathbb{T}}$
  for $p = 2, 3, \dots, k-1$ do
      Solve (18) to obtain $u^*_p$, $\{\eta^*_{l,p}(t+T)\}_{l\in\mathbb{T}}$ and $\{z^*_{l,p}(t+T)\}_{l\in\mathbb{T}}$
  end for
  return $u^*_{k-1}$
```

IV. Constructing a Nominal Task-Priority Control Law

In this section we construct a dynamic task-priority control law, based on input-output feedback linearization and min-norm controllers, for use as the nominal controller required in the method proposed in Section III.

A. Dynamic Task-Priority Control

Many robotic systems can be modeled as a second-order system on the form

$$\dot{x}_1 = F(x_1)x_2 \tag{19a}$$

$$\dot{x}_2 = M(x_1)^{-1}\left(u - C(x_1,x_2)x_2 - G(x_1)\right) \tag{19b}$$

with state variables $x_1 \in \mathbb{R}^{n_1}$, $x_2 \in \mathbb{R}^{n_2}$ and control input $u \in \mathbb{R}^p$, where $M(x_1)$ is the inertia matrix, $C(x_1,x_2)$ is the Coriolis-centripetal matrix and $G(x_1)$ is a vector of gravitational forces. For a task

$$y_i = h_i(x_1) \in \mathbb{R}^{m_i} \tag{20}$$

we define the task error as

$$\tilde{y}_i = h_i(x_1) - h_{i,d} \in \mathbb{R}^{m_i} \tag{21}$$

where $h_{i,d} \in \mathbb{R}^{m_i}$ is the desired value for task y_i. The task-error derivative is

$$\dot{\tilde{y}}_i = \frac{\partial h(x_1)}{\partial x_1}\dot{x}_1 \tag{22}$$

$$= \underbrace{\frac{\partial h(x_1)}{\partial x_1}F(x_1)}_{J_i(x_1)}x_2 \tag{23}$$

and the task-error dynamics are given by

$$\ddot{\tilde{y}}_i = J_i(x_1)\dot{x}_2 + \dot{J}_i(x_1,x_2)x_2 \tag{24}$$

For readability we omit function dependencies for the rest of this Section. In the case where we have one task, y_1, it is clear that if we define $u = J_1^T F_1$, where F_1 is a generalized task force, we can rewrite the task-error dynamics as

$$\Lambda_1\ddot{\tilde{y}}_1 + \Lambda_1\left(J_1M^{-1}\left(Cx_2+G\right) - \dot{J}_1x_2\right) = F_1 \tag{25}$$

where

$$\Lambda_1 = \left(J_1M^{-1}J_1^T\right)^{-1} \in \mathbb{R}^{m_1\times m_1} \tag{26}$$

If the system (19) is redundant with respect to the task y_1, we can define a null-space operator

$$N = I - J_1^T\bar{J}_1^T \tag{27}$$

where $\bar{J}_1$ is a weighted pseudoinverse of J_1 given by

$$\bar{J}_1 = M^{-1}J_1^T\left(J_1M^{-1}J_1^T\right)^{-1} \in \mathbb{R}^{n_2\times m_1} \tag{28}$$

Importantly, it can be shown that N satisfies

$$J_1M^{-1}N = 0 \tag{29}$$

making it possible to solve an auxiliary task in the null-space of y_1 by using the input [4]

$$u = J_1^T F_1 + N\tau_0 \tag{30}$$

where τ_0 is some input solving the auxiliary task. This achieves strict priority between the tasks in the sense that the auxiliary task never generates an acceleration in the task-space of the first task, y_1, due to the property (29).

B. Extension to k Tasks

The scheme presented above in Section IV-A can be extended to a set of k tasks [5] by defining null-space operators N_i recursively as

$$\begin{aligned} N_1 &= I \\ N_{k+1} &= \left(I - N_kJ_k^T\bar{J}_k^T\right)N_k \end{aligned} \tag{31}$$

where $\bar{J}_k$ is a dynamically consistent pseudoinverse defined as the weighted pseudoinverse

$$\bar{J}_i = M^{-1}J_i^T\Lambda_i \in \mathbb{R}^{n\times m} \tag{32}$$

and Λ_i is the task inertia matrix given by

$$\begin{aligned} \Lambda_i &= \left(J_iN_i^TM^{-1}N_iJ_i^T\right)^{-1} \\ &= \left(J_iM^{-1}N_iJ_i^T\right)^{-1} \end{aligned} \tag{33}$$

This null-space operator can be shown to satisfy

$$J_iM^{-1}N_j = 0 \quad \forall i < j \tag{34}$$

Thus, for a set of k tasks, using the control law

$$\begin{aligned} u &= \sum_{j=1}^{k} N_jJ_j^T\tau_j \\ &= \underbrace{\begin{bmatrix} J_1^T & N_2J_2^T & \dots & N_kJ_k^T \end{bmatrix}}_{T(x_1)} \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_k \end{bmatrix}}_{\tau} \\ &= T(x_1)\tau \end{aligned} \tag{35}$$

where $\tau_i \in \mathbb{R}^{m_i}$ is a generalized force in the task space of task i, achieves priority between tasks in the sense that generalized task forces do not generate accelerations in the task-space of higher-priority tasks, due to property (34).

C. Input-output Feedback Linearization for Task-Priority Control

The dynamic task-priority method presented above is used in [17] to design an input-output feedback linearizing control law. Using the control law (35) the input-output dynamics of the system are given by

$$\underbrace{\begin{bmatrix} \ddot{\tilde{y}}_1 \\ \ddot{\tilde{y}}_2 \\ \vdots \\ \ddot{\tilde{y}}_k \end{bmatrix}}_{\ddot{\tilde{y}}} = \underbrace{\begin{bmatrix} J_1 M^{-1} T(x_1) \\ J_2 M^{-1} T(x_1) \\ \vdots \\ J_k M^{-1} T(x_1) \end{bmatrix}}_{A(x_1)} \tau + \underbrace{\begin{bmatrix} b_1(x_1, x_2) \\ b_2(x_1, x_2) \\ \vdots \\ b_k(x_1, x_2) \end{bmatrix}}_{b(x_1, x_2)} \tag{36}$$

where

$$b_i(x_1, x_2) = -J_i M^{-1} \left(C x_2 + G \right) + \dot{J}_i x_2 \tag{37}$$

and

$$A(x_1) = \begin{bmatrix} J_1 M^{-1} J_1^T & 0_{m_1 \times m_2} & 0_{m_1 \times m_3} & \cdots & 0_{m_1 \times m_k} \\ J_2 M^{-1} J_1^T & J_2 M^{-1} N_2 J_2^T & 0_{m_2 \times m_3} & \cdots & 0_{m_2 \times m_k} \\ J_3 M^{-1} J_1^T & J_3 M^{-1} N_2 J_2^T & J_3 M^{-1} N_3 J_3^T & \ddots & 0_{m_3 \times m_k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ J_k M^{-1} J_1^T & J_k M^{-1} N_2 J_2^T & J_k M^{-1} N_3 J_3^T & \cdots & J_k M^{-1} N_k J_k^T \end{bmatrix} \tag{38}$$

due to the property of the null-space operator (31) that $J_i M^{-1} N_k = 0$ for $i < k$. Assuming [17, Theorem 1] is satisfied, a feedback-linearizing control law given by

$$\tau = A^{-1}(x) \left(\mu - b(x) \right) \tag{39}$$

can then be used to transform the system to $m = \sum_{i=1}^{k} m_i$ fully linearized and independent single-input single-output systems

$$\begin{bmatrix} \ddot{\tilde{y}}_1 \\ \ddot{\tilde{y}}_2 \\ \vdots \\ \ddot{\tilde{y}}_k \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{bmatrix} \tag{40}$$

D. Nominal Min-Norm Control Law

Since all tasks are relative degree two, we define $\eta_i = \operatorname{col}\left(\tilde{y}_i, \dot{\tilde{y}}_i \right)$. Using the input-output feedback linearizing control law (39) to transform the system to the form (40), the dynamics of η_i become

$$\dot{\eta}_i = F_i \eta_i + G_i \mu_i \tag{41}$$

with F_i and G_i given by

$$F_i = \begin{bmatrix} 0_{m_i \times m_i} & I_{m_i \times m_i} \\ 0_{m_i \times m_i} & 0_{m_i \times m_i} \end{bmatrix}, \quad G_i = \begin{bmatrix} 0_{m_i \times m_i} \\ I_{m_i \times m_i} \end{bmatrix} \tag{42}$$

We can then construct rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs) following [18] to stabilize each task at a rate ϵ_i by using the candidate control Lyapunov function

$$V_{\epsilon_i}(\eta_i) = \eta_i^T P_{\epsilon_i} \eta_i \tag{43}$$

where

$$P_{\epsilon_i} = \begin{bmatrix} \frac{1}{\epsilon_i} I & 0 \\ 0 & I \end{bmatrix} P_i \begin{bmatrix} \frac{1}{\epsilon_i} I & 0 \\ 0 & I \end{bmatrix} \tag{44}$$

and $P_i = P_i^T > 0$ is the solution to the continuous time algebraic Riccati equation

$$F_i^T P_i + P_i F_i - P_i G_i G_i^T P_i + Q_i = 0 \tag{45}$$

for some positive definite matrix Q_i. We then choose the function $\sigma_i(\eta_i)$ for each task as

$$\sigma_i(\eta_i) = \frac{\gamma_i}{\epsilon_i} V_i(\eta_i) \tag{46}$$

where $\gamma_i = \frac{\lambda_{\min}(Q_i)}{\lambda_{\max}(P_i)} > 0$. Using this, a min-norm controller $\mu_i = \kappa_i(x)$ can be designed as in (6) for each task, where $\kappa_i(x)$ now is given by

$$\kappa_i(x) = \begin{cases} -\frac{\psi_{i,0}(x) \psi_{i,1}(x)}{\psi_{i,1}^{\mathrm{T}}(x) \psi_{i,1}(x)} & \text{when } \psi_{i,0}(x) > 0 \\ 0 & \text{when } \psi_{i,0}(x) \leq 0 \end{cases} \tag{47}$$

with

$$\begin{aligned} \psi_{i,0}(x) &:= L_{\bar{f}_i} V_i \left(\eta_i \right) + \sigma_i(\eta_i) \\ \psi_{i,1}(x) &:= \left[L_{\bar{g}_i} V_i \left(\eta_i \right) \right]^{\mathrm{T}} \end{aligned} \tag{48}$$

where

$$L_{\bar{f}_i} V_i = \eta_i^T \left(F_i^T P_{\epsilon_i} + P_{\epsilon_i} F_i \right) \eta_i \tag{49}$$

$$L_{\bar{g}_i} V_i = 2 \eta_i^T P_{\epsilon_i} G_i \tag{50}$$

The full nominal task-priority control law $u = \kappa(x)$ is given by

$$\kappa(x) = T\left(x_1 \right) A^{-1}(x) (\mu - b(x)) \tag{51}$$

which solves the task-priority control problem with strict priority between tasks at different priority levels.

V. Case Study: Two-Link Manipulator on a Cart

We will now consider a two-link manipulator on a cart, where m_0, m_1 and m_2 are the masses of the cart, first and second link of the manipulator. Let I_1, I_2 be the moments of inertia around the mass-center of each link, and l_1, l_2 be the distance from the start of each link to the center of mass. Finally, L_1, L_2 are the total lengths of each link. We assume the system is fully actuated such that the we can directly control the force on the cart and the torques applied to the joints. Let $x_1 = \operatorname{col}(p, \theta_1, \theta_2)$, where p is the horizontal position of the cart and θ_1, θ_2 are the angles from vertical of the first and second link, respectively. Letting $x_2 = \dot{x}_1$, the dynamics of the system are described by the nonlinear second-order differential equation on the form (19) with $F(x_1) = I_{3 \times 3}$ and

$$M(x_1) = \begin{bmatrix} m_0 + m_1 + m_2 & (m_1 l_1 + m_2 L_1) \cos\theta_1 & m_2 l_2 \cos\theta_2 \\ (m_1 l_1 + m_2 L_1) \cos\theta_1 & m_1 l_1^2 + m_2 L_1^2 + I_1 & m_2 L_1 l_2 \cos(\theta_1 - \theta_2) \\ m_2 l_2 \cos\theta_2 & m_2 L_1 l_2 \cos(\theta_1 - \theta_2) & m_2 l_2^2 + I_2 \end{bmatrix}$$

$$C(x_1, x_2) = \begin{bmatrix} 0 & -\left(m_1 l_1 + m_2 L_1 \right) \sin\theta_1 \dot{\theta}_1 & -m_2 l_2 \sin\theta_2 \dot{\theta}_2 \\ 0 & 0 & m_2 L_1 l_2 \sin\left(\theta_1 - \theta_2 \right) \dot{\theta}_2 \\ 0 & -m_2 L_1 l_2 \sin\left(\theta_1 - \theta_2 \right) \dot{\theta}_1 & 0 \end{bmatrix}$$

$$G(x_1) = \begin{bmatrix} 0 \\ -\left(m_1 l_1 + m_2 L_1 \right) g \sin\theta_1 \\ -m_2 g l_2 \sin\theta_2 \end{bmatrix}$$

It can also be expressed on the form (1) by defining $x = \mathrm{col}(x_1, x_2)$ and

$$f(x) = \begin{bmatrix} x_2 \\ -M^{-1}(x_1)\left(C(x_1, x_2)x_2 + G(x_1)\right) \end{bmatrix} \quad (52)$$

$$g(x) = \begin{bmatrix} 0_{3\times 3} \\ M^{-1}(x_1) \end{bmatrix} \quad (53)$$

A. Tasks

The first task we will consider is the cart position defined as

$$y_1 = p \quad (54)$$

The second task we will consider is the position of the end of the second link, which we will call the end-effector, defined by

$$y_2 = \begin{bmatrix} x_{\mathrm{ee}} \\ y_{\mathrm{ee}} \end{bmatrix} = \begin{bmatrix} p + L_1 \sin(\theta_1) + L_2 \sin(\theta_2) \\ h_c + L_1 \cos(\theta_1) + L_2 \cos(\theta_2) \end{bmatrix} \quad (55)$$

where h_c is the vertical position of the center of the cart. The last task we consider is the angle of the first joint

$$y_3 = \theta_1 \quad (56)$$

B. CLF Design

We now want to design the CLFs required in the optimization problems (17)-(18). Since all tasks have relative degree two, we can define $\eta_i = \mathrm{col}\left(\tilde{y}_i, \dot{\tilde{y}}_i\right)$ which is governed by the differential equation

$$\dot{\eta}_i = F_i \eta_i + G_i \left(J_i M^{-1} \left(u - Cx_2 - G\right) + \dot{J}_i x_2 \right) \quad (57)$$

where F_i and G_i are defined as in (42). We can then construct RES-CLFs identically to (43), to stabilize each task at a rate ϵ_i. Thus, for these CLFs we have

$$\begin{aligned} L_{\bar{f}_i} V_i &= \eta_i^T \left(F_i^T P_{\epsilon_i} + P_{\epsilon_i} F_i\right) \eta_i \\ &\quad + 2\eta_i^T P_{\epsilon_i} G_i \left(-J_i M^{-1} \left(Cx_2 + G\right) + \dot{J}_i x_2 \right) \quad (58) \\ L_{\bar{g}_i} V_i &= 2\eta_i^T P_{\epsilon_i} G_i J_i M^{-1} u \end{aligned}$$

To solve the continuous time algebraic Riccati equation (45) for P_i we chose $Q_i = I_{2m_i \times 2m_i}$ where m_i is the dimension of each task. We use the same functions $\sigma_i(\eta_i)$ as in (46). We choose the values

$$\epsilon_1 = 3, \quad \epsilon_2 = 4, \quad \epsilon_3 = 4 \quad (59)$$

for the tasks y_1, y_2, y_3 defined in Section V-A. We use the same values ϵ_i for the CLFs defined here for the CLFs used in the nominal controller designed in Section IV-D.

C. Model Predictive Task-Priority Control Law

We spread the three defined tasks on three priority levels $p = \{1, 2, 3\}$ such that

$$\mathbb{I}_1 = \{1\}, \quad \mathbb{I}_2 = \{1, 2\} \quad (60)$$

and

$$\mathbb{J}_1 = \{2\}, \quad \mathbb{J}_2 = \{3\} \quad (61)$$

in the optimization problems (17)-(18). As the nominal control law for Algorithm 1, we use the input-output feedback linearizing task-priority control law (51). The cost functions in (17)-(18) are chosen to be

$$c_p(x, u) = \varphi u^T u + \sum_{i \in \mathbb{I}_p \cup \mathbb{J}_p} \lambda_i \eta_i(x)^T \eta_i(x) \quad (62)$$

for all priority levels $p \in \{1, 2, 3\}$, where $\varphi > 0$ and $\lambda_i > 0$ are constant weights. These quantities effectively penalize tracking-errors and large control inputs, respectively, over the prediction horizon. We choose $\varphi = 0.1$ and $\lambda_i = 50$ for all $i \in \mathbb{I}_p \cup \mathbb{J}_p$ and all $p \in \{1, 2, 3\}$. We use a relatively short prediction horizon of $T = 0.5\,\mathrm{s}$

D. Comparison with the Nominal Min-Norm Control Law

Here we compare the proposed method, using the controller designed in Section IV-D as a nominal controller, with only using the nominal min-norm controller from Section IV-D. We set the desired value for each task over the simulation as a series of steps $\Delta h_{i,d}$ from the start value $y(x(0))$ for each task. These steps are chosen to be

t	0	50	100
$\Delta h_{1,d}$	1.1	0	-1
$\Delta h_{2,d}$	$[1, 0]^T$	$[0, 0.1]^T$	$[-1, -0.1]^T$
$\Delta h_{3,d}$	0	0	0

(63)

and the initial state $x(0)$ of the system is initialized to

$$x(0) = \begin{bmatrix} 0\,\mathrm{m} \\ \frac{\pi}{6}\,\mathrm{rad} \\ \frac{\pi}{3}\,\mathrm{rad} \end{bmatrix} \quad (64)$$

Figures 1 - 4 show results in simulation comparing the proposed method with the nominal min-norm task-priority controller. In Figures 1 and 2 we can clearly see that the proposed method achieves faster convergence to the desired value for the first two tasks y_1 and y_2, with only a small overshoot for the second task. As seen in Figure 4 there are small spikes in the control input due to the fact that the desired value changes discontinuously, but from $t = 25$ s the control input is in fact similar to, or smaller than, using the nominal controller.

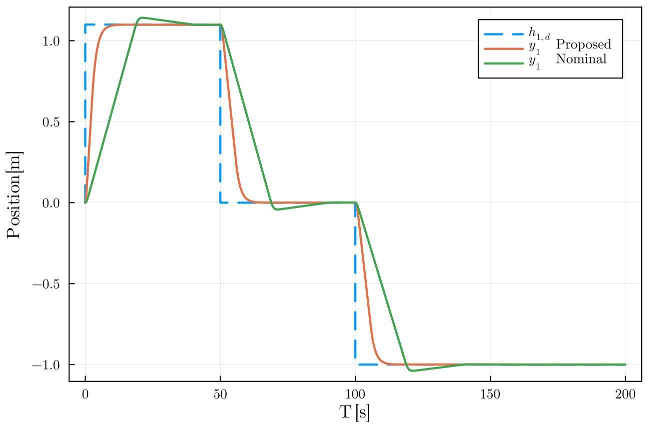

Fig. 1. Cart position

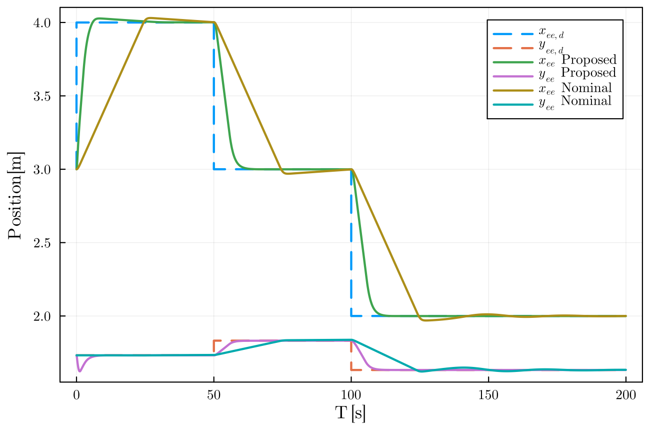

Fig. 2. End-effector position

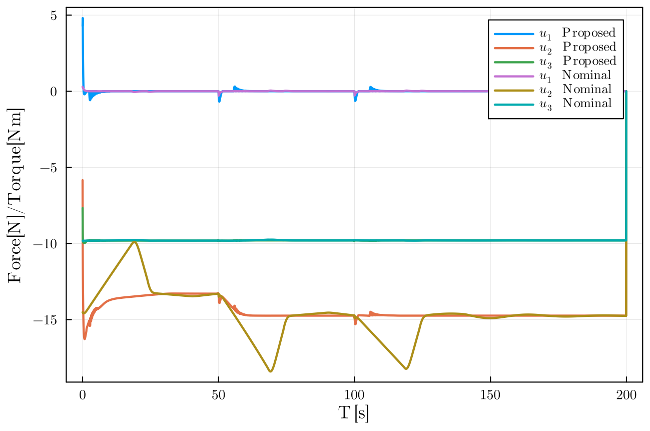

Fig. 4. Control inputs

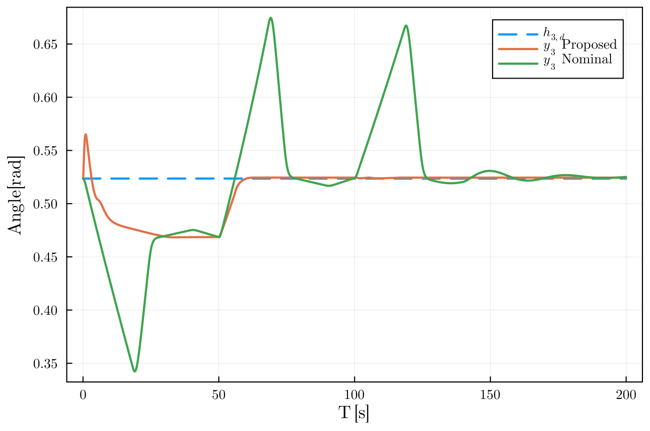

Fig. 3. Angle of the first joint

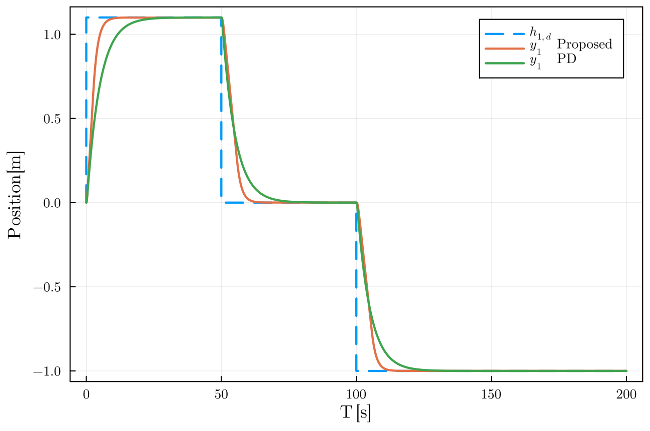

Fig. 5. Cart position

E. Comparison with a PD Control Law

We now compare the proposed method, using the controller designed in Section IV-D as a nominal controller, with a control law using the input-output feedback linearizing control law (39) together with a PD control law for each task. The PD control law is given by

$$\mu_i = -K_{p,i}\tilde{y}_i - K_{d,i}\dot{\tilde{y}}_i \tag{65}$$

We let $K_{p,i} = 1$ and $K_{d,i} = 5$ for all tasks. We use the same initial value for $x(0)$ and desired set-point values as in Section V-D. As we can see from Figures 5 - 7, the proposed method has slightly faster convergence to the desired value than the PD control law. The spike in control inputs at $t = 0\ s$ is slightly larger, but the spikes at subsequent set-point changes are smaller.

VI. Conclusions

This paper has presented a hierarchical model predictive dynamic task-priority control method using control Lyapunov functions, allowing for an arbitrary number of tasks spread over an arbitrary number of priority levels with strict priority between tasks at different priority levels. The method uses model predictive control to improve the closed-loop performance of the system relative to a user specified cost function. The proposed method is verified in simulation on a redundant robotic system where it shows good performance compared with existing dynamic task-priority control methods, without requiring higher control-efforts.

References

[1] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," *IFAC Proceedings Volumes*, vol. 14, no. 2, pp. 1927–1932, Aug. 1981.

[2] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, Jun. 1987.

[3] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, Pisa, Italy: IEEE, 1991, 1211–1216 vol.2.

[4] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

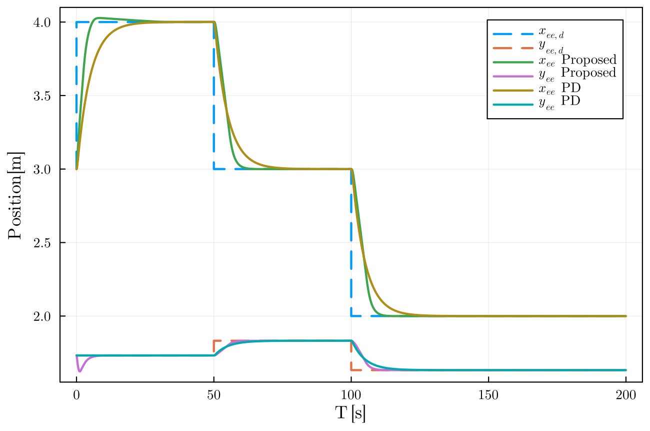

Fig. 6. End-effector position

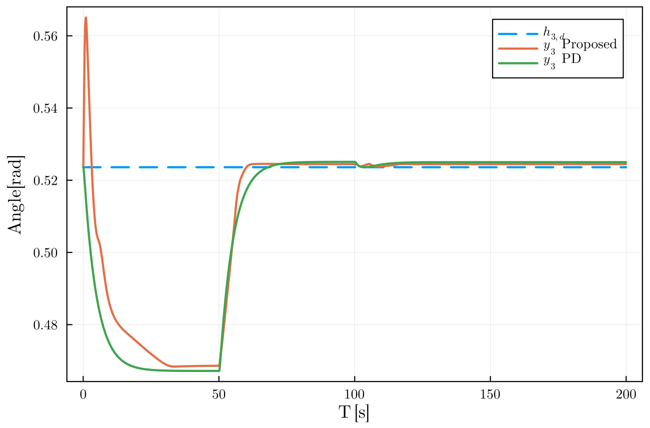

Fig. 7. Angle of the first joint

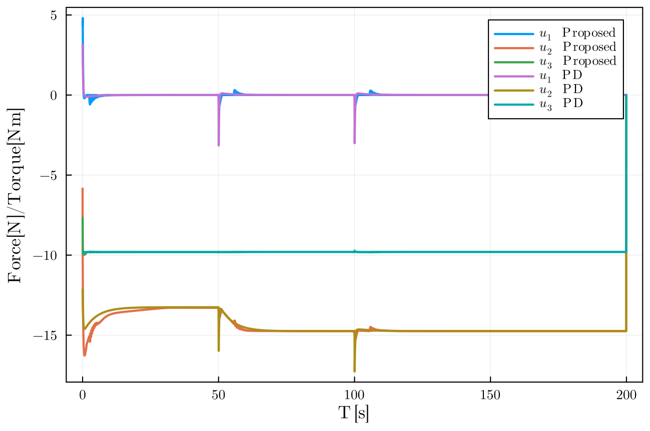

Fig. 8. Control inputs

[5] L. Sentis and O. Khatib, “Prioritized multi-objective dynamics and control of robots in human environments,” in *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, vol. 2, Santa Monica, CA, USA: IEEE, 2004, pp. 764–780.

[6] L. Sentis and O. Khatib, “A whole-body control framework for humanoids operating in human environments,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, FL, USA: IEEE, 2006, pp. 2641–2648.

[7] L. Sentis, “Synthesis and control of whole-body behaviors in humanoid systems,” Ph.D. dissertation, Standford University, Stanford, CA, USA, 2007.

[8] P. Baerlocher and R. Boulic, “Task-priority formulations for the kinematic control of highly redundant articulated structures,” in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 1, Victoria, BC, Canada: IEEE, 1998, pp. 323–329.

[9] A. Dietrich, C. Ott, and A. Albu-Schäffer, “An overview of null space projections for redundant, torque-controlled robots,” *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1385–1400, Sep. 2015.

[10] Z. Artstein, “Stabilization with relaxed controls,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, Jan. 1983.

[11] E. D. Sontag, “A ‘universal’ construction of artstein’s theorem on nonlinear stabilization,” *Systems & Control Letters*, vol. 13, no. 2, pp. 117–123, Aug. 1989.

[12] R. A. Freeman and P. V. Kokotovic, “Inverse optimality in robust stabilization,” *SIAM Journal on Control and Optimization*, vol. 34, no. 4, pp. 1365–1391, Jul. 1996.

[13] J. Primbs, V. Nevistic, and J. Doyle, “A receding horizon generalization of pointwise min-norm controllers,” *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 898–909, May 2000.

[14] A. D. Ames and M. Powell, “Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs,” in *Control of Cyber-Physical Systems*, ser. Lecture Notes in Control and Information Sciences, D. C. Tarraf, Ed., vol. 449, Heidelberg: Springer International Publishing, 2013, pp. 219–240.

[15] E. A. Basso and K. Y. Pettersen, “Task-priority control of redundant robotic systems using control lyapunov and control barrier function based quadratic programs,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9037–9044, 2020.

[16] R. A. Freeman and P. Kokotović, *Robust Nonlinear Control Design*. Boston, MA: Birkhäuser Boston, 1996.

[17] E. A. Basso and K. Y. Pettersen, “MIMO feedback linearization of redundant robotic systems using task-priority operational space control,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5459–5466, 2020.

[18] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, “Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, Apr. 2014.