Chapter 1
Assistive Optimal Control-on-Request with Application in Standing Balance Therapy and Reinforcement

Anastasia Mavrommati, Alex Ansari and Todd D. Murphey

Abstract This chapter develops and applies a new control-on-request (COR) method to improve the capability of existing shared control interfaces. These COR enhanced interfaces allow users to request on-demand bursts of assistive computer control authority when manual / shared control tasks become too challenging. To enable the approach, we take advantage of the short duration of the desired control responses to derive an algebraic solution for the optimal switching control for differentiable nonlinear systems. Simulation studies show how COR interfaces present an opportunity for human-robot collaboration in standing balance therapy. In particular, we use the Robot Operating System (ROS) to show that optimal control-on-request achieves the therapy objectives of active patient participation and safety. Lastly, we explore the potential of a COR interface as a vibrotactile feedback generator to dynamically reinforce standing balance through sensory augmentation.

1.1 Introduction

This chapter develops and explores the benefits of a specific form of control— referred to as *burst control*—to human-robot collaboration settings. Burst control laws consist of a continuous nominal control mode (NC) that switches to alternate, assistive control modes (AC) lasting for short but finite durations (see Fig. 1.1). One of the simplest way robots can collaborate with humans is to aid in the completion of physical tasks. More often than not, this assistive setting requires that control authority is shared between the collaborators. Where traditional shared control strategies work in conjunction with or filter user input commands (see [13, 15, 25, 48, 63]), burst control strategies can provide for a different form of shared control follow-

Anastasia Mavrommati, Alex Ansari and Todd D. Murphey
Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, USA, e-mail: stacymav@u.northwestern.edu;alexanderansari2011@u.northwestern.edu;t-murphey@northwestern.edu

ing a *control-on-request* (COR) paradigm. Potentially working along with existing shared controllers, burst control signals can be applied on request to implement quick changes, assisting with control goals as human tasks become too challenging to maintain. Due to their short duration, the key advantage of these signals is that they allow computers to take control authority and implement changes before users notice the loss in control. Through interactive simulation studies, we demonstrate how this COR paradigm can be implemented using simple interface components (e.g. buttons, toggles, or switches), providing users the ability to collaborate with robotic systems by quickly requesting bursts of assistance.

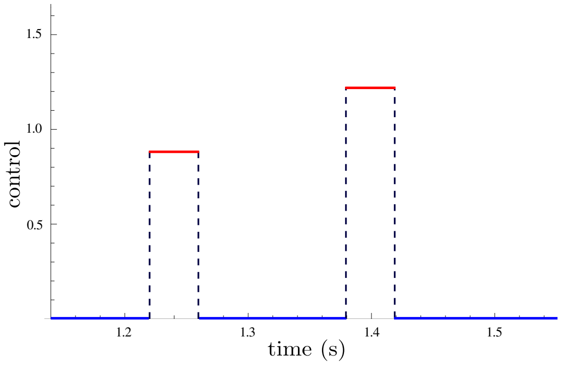
The challenge in implementing these on-demand, quick-assist controllers is that optimal control responses for nonlinear systems require constrained iterative optimization and so are infeasible to compute on-the-fly. To address this concern, the first part of this chapter provides the theoretical foundation that allows for rapid generation of optimal burst control laws. First, the process computes the control action[1] to apply at any time to which a system's tracking objective is optimally sensitive. As Section 1.2.1 proves, this action is guaranteed to exist under standard assumptions. Planning for infinitesimal activation durations, we show optimal actions can be computed extremely rapidly from a simple algebraic expression of state and costate (no iterative optimization). Next, Section 1.2.2 provides a simple line search that is guaranteed to find a finite application duration for any selected activation time, τ_m, that will provide a desired improvement in trajectory. By quickly computing optimal actions and resolving their finite application durations, the approach calculates assistive burst controls on-demand.

The remainder of this chapter explores the potential of the proposed COR interface in enhancing human-robot collaboration in the field of rehabilitation. We present two studies, each of which aims to address different aspects of collaboration between humans and robotic platforms. Our first study focuses on the element of *human-robot interaction* and regards rehabilitative devices as assistant robots. In particular, we demonstrate that COR interfaces embedded in robotic lower-limp exoskeletons can benefit standing balance therapy by promoting real-time collaboration between the therapist and the robotic device. To illustrate our approach, consider the case where a standing patient attempts to maintain an upright posture while wearing a robotic lower-limb exoskeleton. In this scenario, a monitoring physical therapist equipped with a COR interface has the option to either assist the patient manually or activate a corrective exoskeleton response on demand—hence actively collaborating with the robot to improve and facilitate therapy. In the event of COR activation, a burst of optimal assistance is calculated in real time and applied for a very short duration, so that immediately after, full control authority is ceded back to the patient/therapist. Using the Robot Operating System (ROS), Section 1.3 shows that this therapist-robot collaboration paradigm can *ensure patient safety* while promoting *active patient participation*.

However, there is one central aspect of collaboration that is not addressed in this first study, and that is *human-robot communication*. Past results suggest that a robot

[1] In this chapter a control action is a control vector with constant values applied for a (possibly infinitesimal) duration of time.

Fig. 1.1 An example burst control law. Continuous NC mode $u(t) = 0$ (blue curve) switches at $t = 1.22s$ and $t = 1.38s$ to alternate control modes (red) that are applied for short time horizons ($0.04s$).



needs to be able to produce non-verbal communication cues to be an effective collaborative partner [24]. Diverging from commonly employed communication channels, our second—short—study proposes that the robot uses vibrotactile feedback to communicate COR-generated intents. As opposed to the first study where the therapist is the epicenter, Section 1.4 examines how the patient can actively collaborate with the robot in achieving a desired upright posture, by interpreting and following tactile cues.

Following this Introduction, Section 1.2 derives a technique that takes advantage of the structure of burst control signals to compute an algebraic optimal control solution for systems with nontrivial dynamics. Section 1.3 discusses the application of COR interfaces to robot-assisted balance therapy and provides promising simulation results. Finally, Section 1.4 explores the potential of employing COR-generated vibrotactile signals to achieve posture reinforcement.

1.2 Assistive Control Synthesis

This section presents a method to compute burst AC laws that attain specified levels of improvement in trajectory tracking cost functionals. Section 1.2.1 leverages the short intended switching duration of burst AC signals to avoid dynamic constraints and iterative optimization in solving for a schedule of infinitesimal optimal actions to apply at any time. These infinitesimal actions specify the magnitude and direction of the burst control. The following Section 1.2.2 provides a line search process that returns a finite duration to apply each action to produce a burst control law that achieves a desired reduction in tracking cost over a specified horizon.

1.2.1 Calculating a schedule of optimal infinitesimal actions

The type of systems addressed in this chapter are assumed to follow continuous trajectories, $(x(t), u(t))$, such that

$$\dot{x}(t) = f(x(t), u(t)) \,. \tag{1.1}$$

The dynamics vector, $f(x(t), u(t))$, can be nonlinear with respect to the state vector, $x(t) \in \mathbb{R}^n$, but is assumed to be linear (or has been linearized) with respect to control vector, $u(t) \in \mathbb{R}^m$. With these assumptions, (1.1) can be written in control-affine form

$$f(x(t), u(t)) = g(x(t)) + h(x(t))\, u(t) \,. \tag{1.2}$$

The goal of the proposed COR controllers is to predict the trajectory of the system and compute an optimal AC that produces a specified level of improvement in this trajectory when activated. To accomplish this, a cost functional is used to compare the performance of different trajectories. The state tracking cost functional,[2]

$$\begin{aligned} J_1 &= \int_{t_0}^{t_f} l_1(x(t))\, dt + m(x(t_f)) \\ &= \frac{1}{2} \int_{t_0}^{t_f} \|x(t) - x_d(t)\|_Q^2\, dt + \frac{1}{2}\, \|x(t_f) - x_d(t_f)\|_{P_1}^2 \,, \end{aligned} \tag{1.3}$$

serves this purpose. For control derivations, J_1 need only obey general form (1.3). However, the quadratic form is applied in the implementation described in this chapter, i.e. $l_1(x(t)) := \frac{1}{2}\|x(t) - x_d(t)\|_Q^2$ and $m(x(t_f)) := \frac{1}{2}\, \|x(t_f) - x_d(t_f)\|_{P_1}^2$. For this case, $Q = Q^T \geq 0$ defines the metric on incremental state tracking error and $P_1 = P_1^T \geq 0$ defines the metric on state error at terminal time t_f. Initial time, t_0, is assumed to be the activation time of the COR interface to select controls that improve the future system trajectory.

Assume the system can be in one of two dynamic modes at a given time that differ only in control. Under nominal conditions, the system applies NC law, $u(t)$, and dynamics, $f_1 = f(x(t), u(t))$, result. Upon activating the COR interface, the dynamics switch from NC mode f_1 to an alternate AC dynamic mode $f_2 = f(x(t), u_2(t))$ for a short duration, λ^+, before switching back. Consider the extreme case where the system evolves according to f_1, and AC $u_2(t)$ is applied for an infinitesimal duration before switching back to the NC law, $u(t)$. In this case the mode insertion gradient,

$$\frac{dJ_1}{d\lambda^+} = \rho(t)^T [f_2 - f_1] \,, \tag{1.4}$$

measures the resulting change in cost functional (1.3) (see [18, 19, 57]). The term, $\rho(t) \in \mathbb{R}^n$, is the adjoint (co-state) variable calculated from the current system trajec-

[2] The notation, $\|\cdot\|_M^2$, indicates a norm on the argument where matrix, M, provides the metric (i.e. $\|x(t)\|_Q^2 = x(t)^T\, Q\, x(t)$).

tory based on the differential equation[3],

$$\dot{\rho}(t) = -D_x l_1(x(t))^T - D_x f_1^T \rho(t)\,, \tag{1.5}$$

such that at the final time $\rho(t_f) = D_x m(x(t_f))^T$. The control duration, λ^+, is evaluated infinitesimally, as $\lambda^+ \to 0$.

Through the course of the trajectory, any time, τ_m, the mode insertion gradient is negative, the state tracking cost functional can be reduced if $u_2(\tau_m)$ is activated for some duration at that time. The magnitude of the mode insertion gradient provides a first-order model of the change in trajectory cost that would result relative to the duration of application of this control.

To produce a desired degree of improvement in a system trajectory, each applied burst of AC needs to improve cost (1.3) by a specified amount. In other words, control actions, $u_2^*(\tau_m)$, need to be computed that drive the mode insertion gradient to a desired negative value, $\alpha_d \in \mathbb{R}^-$. However, there is generally a cost associated with the application of control authority. As such, a trade-off must be made in tracking the desired value of the mode insertion gradient, α_d, relative to control effort. Following a trajectory optimization approach, these competing goals can be encoded into the cost functional,

$$\begin{aligned} J_2 &= \int_{t_0}^{t_f} l_2(x(t), u(t), u_2(t), \rho(t))\, dt \qquad \text{where} \\ l_2(\cdot) :=& \frac{1}{2}[\frac{dJ_1}{d\lambda^+} - \alpha_d]^2 + \|u_2(t)\|_R^2 := \frac{1}{2}[\rho(t)^T(f_2 - f_1) - \alpha_d]^2 + \|u_2(t)\|_R^2\,. \end{aligned} \tag{1.6}$$

Matrix R allows the designer to encode the cost of control relative to tracking α_d. The continuous schedule[4] of control actions, $u_2^*(t)$, that minimizes (1.6), optimizes this trade-off.

The remainder of the chapter assumes quadratic norms in (1.6) with $R = R^T > 0$. Proved in [9], quadratic functionals and the space of positive semi-definite / definite cones is convex. Because convexity is preserved for non-negative weighted sums and integration, this choice of $R > 0$ provides convexity of (1.6). Additionally, NC is often (indicated by case) assumed to be null, $u(t) = 0$, over the control planning horizon. This choice allows easier interpretation of $u_2^*(\tau_m)$ as the optimal action at τ_m relative to doing nothing (allowing the system to drift for a horizon into the future).

Theorem 1. *Define* $\Lambda \triangleq h(x(t))^T \rho(t) \rho(t)^T h(x(t))$. *The schedule of controls,* $u_2(t)$, *that optimizes* (1.6) *based on dynamics* (1.2) *and state tracking cost functional* (1.3) *is given by the algebraic expression,*

[3] $D_x f(\cdot)$ denotes the partial derivative $\frac{\partial f(\cdot)}{\partial x}$.

[4] At any specified application time, τ_m, of the COR interface, $u_2^*(\tau_m)$ represents the optimal action that balances control authority and drives the mode insertion gradient to α_d if activated around that time. Thus, $u_2^*(t)$ is a schedule of optimal actions that can be switched to from NC, $u(t)$, to produce a desired change in mode insertion gradient at τ_m.

$$u_2^*(t) = (\Lambda + R^T)^{-1} \, [\Lambda \, u(t) + h(x(t))^T \rho(t) \, \alpha_d] \, . \tag{1.7}$$

Proof. Any time $t \in [t_0, t_f]$, action $u_2(t)$ is assumed to be applied infinitesimally and so does not affect state trajectory (i.e. $x = x(u(t), t)$). Optimization of convex cost (1.6) with respect to $u_2(t) \, \forall t \in [t_0, t_f]$ is therefore unconstrained by (1.2) and it is necessary and sufficient for (global) optimality to find a curve $u_2(t)$ that sets its first variation to 0. Using the Gâteaux derivative and the definition of the functional derivative,

$$\begin{aligned}
\delta J_2 =& \int_{t_0}^{t_f} \frac{\delta J_2}{\delta u_2(t)} \delta u_2(t) \, dt \\
=& \frac{d}{d\epsilon} \int_{t_0}^{t_f} l_2(x(t), u(t), u_2 + \epsilon \, \eta(t), \rho(t)) \, dt|_{\epsilon=0} \\
=& \int_{t_0}^{t_f} \frac{d}{d\epsilon} l_2(x(t), u(t), u_2 + \epsilon \, \eta(t), \rho(t))|_{\epsilon=0} \, dt \\
=& \int_{t_0}^{t_f} \frac{\partial l_2(x(t), u(t), u_2(t), \rho(t))}{\partial u_2(t)} \eta(t) \, dt \\
=& 0 \, ,
\end{aligned} \tag{1.8}$$

where ϵ is a scalar and $\epsilon \, \eta(t) = \delta u_2(t)$.

At the optimal value of $u_2(t)$ (i.e. $u_2(t) = u_2^*(t)$), the final equivalence in (1.8) must hold $\forall \, \eta(t)$. By the Fundamental Lemma of Variational Calculus (see [45]), this implies $\frac{\partial l_2(\cdot)}{\partial u_2(t)} = 0$ at the optimizer. The resulting expression,

$$\begin{aligned}
\frac{\partial l_2(\cdot)}{\partial u_2(t)} =& (\rho(t)^T \, h(x(t)) \, [u_2(t) - u(t)] - \alpha_d) \\
& \rho(t)^T \, h(x(t)) + \; u_2(t)^T \, R \\
=& \, h(x(t))^T \rho(t) \, (\rho(t)^T \, h(x(t)) \\
& [u_2(t) - u(t)] \, - \alpha_d) + R^T u_2(t) \\
=& \, [h(x(t))^T \rho(t) \rho(t)^T \, h(x(t))] \, u_2(t) \\
& + R^T u_2(t) - [h(x(t))^T \rho(t) \rho(t)^T \, h(x(t))] \\
& u(t) - h(x(t))^T \rho(t) \, \alpha_d = 0 \, ,
\end{aligned} \tag{1.9}$$

can therefore be solved in terms of $u_2(t)$ to find the value, $u_2^*(t)$, that minimizes (1.6). Algebraic manipulation confirms this optimal value is given by (1.7).

Though the principles applied to derive the schedule of infinitesimal optimal control actions (1.7) are reasonable, they are also non-traditional. To provide intuition regarding the properties of these solutions, the following proposition proves that the optimization posed to derive these controls is equivalent to a well-studied class of Tikhonov regularization problems (see [12], [14], [23], [27]).

Proposition 1. *Assume $u, u_2, \rho, h \in \mathcal{H}$ where $\mathcal{H}$ is an infinite dimensional reproducing kernel Hilbert function space (RKHS).*[5] *With appropriate change of variables, minimization of* (1.6) *obeys the structure of generalized continuous-time linear Tikhonov regularization problem*[6]

$$\min_{z} \|\Gamma z - b\|^2 + \kappa^2 \|L(z - z_0)\|^2 , \tag{1.10}$$

and (1.7) *obeys the structure of associated solution*

$$z^* = (\Gamma^T \Gamma + \kappa^2 L^T L)^{-1} (\Gamma^T b + \kappa^2 L^T L z_0) . \tag{1.11}$$

Above, Γ and L are bounded linear operators on $\mathcal{H}$, vectors z and $z_0 \in \mathcal{H}$, and b and $\kappa \in \mathbb{R}$. See [12], [14], [23], and [27] for more detail on (1.10) *and* (1.11).

Proof. Using the control affine form of dynamics f_1 and f_2, the final equality in (1.6) can be stated as

$$J_2 = \frac{1}{2} \int_{t_0}^{t_f} [\rho(t)^T h(x(t))(u_2(t) - u(t)) - \alpha_d]^2 + \|u_2(t)\|_R^2 \, dt .$$

Performing change in variables $z(t) = u_2(t) - u(t)$, $z_0(t) = -u(t)$, $\Gamma = \rho(t)^T h(x(t))$, and $b = \alpha_d$ yields

$$J_2 = \frac{1}{2} \int_{t_0}^{t_f} [\Gamma z(t) - b]^2 \, dt + \frac{1}{2} \int_{t_0}^{t_f} \|z(t) - z_0(t)\|_R^2 \, dt .$$

Because $R = R^T > 0$, it can be Cholesky factorized as $R = M^T M$. By pulling out a scaling factor κ^2, the factorization can be rewritten $R = M^T M = \kappa^2 (L^T L)$. Applying this factorization and posing the expression in terms of $\mathcal{L}_2$ norms results in

$$J_2 = \frac{1}{2} \|\Gamma z(t) - b\|^2 + \frac{1}{2} \|\kappa L(z(t) - z_0(t))\|^2 .$$

Minimization of (1.6) is thus equivalent to (1.10) up to a constant factor of $\frac{1}{2}$ that can be dropped as it does not affect z^*.

Additionally, equivalence of solutions (1.7) and (1.11) can be proved directly. With the previous change of variables, $u_2^*(t)$ can be written as $z^*(t) - z_0(t)$ and (1.7) as

$$z^*(t) - z_0(t) = (\Gamma^T \Gamma + \kappa^2 L^T L)^{-1} (-\Gamma^T \Gamma z_0(t) + \Gamma^T b) .$$

Algebraic manipulation verifies this is equal to Tikhonov regularization solution (1.11).

[5] Practically, this is not a very stringent requirement. Most spaces of interest are RKHS. Refer to [12] for more detail.

[6] For equivalence, $\|\cdot\|$ refers to the $\mathcal{L}_2$ norm and $\mathcal{H}$ is additionally assumed to be an $\mathcal{L}_2$ space.

As the following corollary indicates, because minimization of (1.6) can be posed as a Tikhonov regularization problem, solutions (1.7) inherit useful properties that regularization solutions obey.

Corollary 1. *With the assumptions stated in Proposition 1, solutions* (1.7) *for minimization of* (1.6) *exist and are unique.*

Proof. The proof follows from Proposition 1, which shows the minimization can be formulated as a Tikhonov regularization problem with convex $\mathcal{L}_2$ error norm, $\|\Gamma z - b\|$. These problems are guaranteed to have solutions that exist and are unique. A proof is provided in [14].

Globally, optimal control actions (1.7) inherit properties of Tikhonov regularization solutions. However, the following corollary shows that near equilibrium points, solutions (1.7) simplify to linear state feedback laws.

Corollary 2. *Assume system* (1.2) *contains equilibrium point* $x = 0$*, the state and co-state are continuous, and tracking cost* (1.3) *is quadratic*[7]*. There exists a neighborhood around the equilibrium and nonzero time horizon for which optimal actions* (1.7) *are equivalent to linear feedback regulators.*

Proof. At final time, $\rho(t_f) = P_1 x(t_f)$. Due to continuity, this linear relationship between the state and co-state must exist for a nonzero neighborhood around t_f such that

$$\rho(t) = P(t)\,x(t)\,. \tag{1.12}$$

Applying this relationship, (1.7) can formulated as

$$\begin{aligned} u_2^*(t) = &(h(x(t))^T P(t)\,x(t)\,x(t)^T P(t)^T h(x(t)) + R^T)^{-1} \\ &[h(x(t))^T P(t)\,x(t)\,x(t)^T P(t)^T h(x(t))\,u(t) \\ &+ h(x(t))^T P(t)\,x(t)\,\alpha_d]\,. \end{aligned}$$

This expression contains terms quadratic in $x(t)$. In the neighborhood of the equilibrium these quadratic terms go to zero faster than the linear terms, and controls converge to

$$u_2^*(t) = R^{-T} h(x(t))^T\,P(t)\,x(t)\,\alpha_d\,. \tag{1.13}$$

By continuity, in a sufficiently small neighborhood of the equilibrium the system dynamics can be approximated as LTV system, $\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t)$, (where $A(t)$ and $B(t)$ are linearizations about the equilibrium). Applying this assumption and differentiating (1.12) produces

$$\begin{aligned} \dot{\rho}(t) &= \dot{P}(t)\,x(t) + P(t)\,\dot{x}(t) \\ &= \dot{P}(t)\,x(t) + P(t)\,(A(t)\,x(t) + B(t)\,u(t))\,. \end{aligned}$$

[7] A quadratic cost is assumed so that resulting equations emphasize the local similarity between burst control and LQR [6].

Inserting relation (1.5) yields

$$-D_x l_1(\cdot)^T - A(t)^T P(t)\, x(t) = \dot{P}(t)\, x(t) + P(t) \\ (A(t)\, x(t) + B(t)\, u(t))\,,$$

which can be re-arranged such that

$$0 = (Q + \dot{P}(t) + A(t)^T P(t) + P(t)A(t))\, x(t) + P(t)\, B(t)\, u(t).$$

When nominal control $u(t) = 0$, this reduces to

$$0 = Q + A(t)^T P(t) + P(t) A(t) + \dot{P}(t). \tag{1.14}$$

Note the similarity to a Lyapunov equation. As mentioned, this relationship must exist for some nonzero neighborhood of t_f. Therefore, by continuity of $\rho(t)$, there must exist a finite time horizon and neighborhood of the equilibrium where (1.7) simplifies to linear feedback regulator (1.13) and $P(t)$ can be computed from (1.14) subject to $P(t_f) = P_1$.

As in model predictive control (MPC) from [4, 5, 11, 26], it is possible to compute open-loop optimal actions (in this case $u_2^*(\tau_m)$) to provide finite-horizon tracking improvements and to sequence these in closed-loop. This would be equivalent to continuously activating a COR interface. In such implementations, one could specify α_d to provide local stability based on (1.13). Alternatively, if (1.3) is quadratic and NC $u(t)$ modeled as applying consecutively computed optimal actions (1.13) near equilibrium, (1.14) becomes a Riccati differential equation for the closed-loop system (see [28]) and actions (1.13) become finite horizon LQR controls [6]. In this case one can prove the existence of a Lyapunov function and guarantee stability using methods from MPC and LQR theory to drive $\dot{P}(t) \to 0$ ([4, 26, 28, 31, 37, 43]). While beyond this scope, we have begun to explore close-loop implementation and stability to leverage the efficient synthesis methods presented.

1.2.2 Computing the control duration

Theorem 1 provides a means to compute a schedule of open-loop optimal AC actions, $u_2^*(t)$. When implemented infinitesimally around any time, τ_m, $u_2^*(\tau_m)$ is the needle variation (see [64]) in $u(\tau_m)$ that optimizes control authority in driving the mode insertion gradient, $\frac{dJ_1}{d\lambda^+}$, to α_d at that time. This value of the mode insertion gradient reflects the achievable sensitivity of cost (1.3) to application of $u_2^*(\tau_m)$ for infinitesimal duration. However, by continuity of the adjoint and mode insertion gradient as $\lambda^+ \to 0$, there exists an open, non-zero neighborhood, V, around $\lambda^+ \to 0$ for which the mode insertion gradient models this sensitivity to first order (see [10, 19]). Hence the mode insertion gradient can be used to model the change in cost achievable by application of $u_2^*(\tau_m)$ over a finite duration $\lambda^+ \in V$ as

Algorithm 1 Burst Control Synthesis

```
Initialize $\alpha_d$, minimum change in cost $\Delta J_{min}$ from (1.15), current time $t_{curr}$, default control duration $\Delta t_{init}$, scale factor $\beta \in (0,1)$, time horizon $T$, and the max line search iterations $i_{max}$.
if COR interface activated then
    $(t_0,t_f) = (t_{curr}, t_{curr} + T)$
    Simulate $(x(t),\rho(t))$ for $t \in [t_0,t_f]$ from $f_1$
    Compute initial cost $J_{1,init}$
    Specify $\alpha_d$
    Compute $u_2^*(t)$ from $(x(t),\rho(t))$ using Theorem 1
    $\tau_m = \frac{\Delta t_{init}}{2} + t_0$
    Initialize $i = 0$, $J_{1,new} = \infty$
    while $J_{1,new} - J_{1,init} > \Delta J_{min}$ and $i \leq i_{max}$ do
        $\lambda^+ = \beta^i \times \Delta t_{init}$
        $(\tau_0,\tau_f) = (\tau_m - \frac{\lambda^+}{2}, \tau_m + \frac{\lambda^+}{2})$
        Re-simulate $x(t)$ applying $f_2$ for $t \in [\tau_0,\tau_f]$
        Compute new cost $J_{1,new}$
        $i = i + 1$
    end while
end if
return $(u_2^*(\tau_m), \tau_0, \tau_f)$
```

Fig. 1.2: The algorithm above includes an optional line search phase which re-simulates the state $x(t)$ and trajectory cost until an appropriate duration for AC application of $u_2^*(\tau_m)$ can be found. This is only required if the approximation provided by the mode insertion gradient is not accurate over the default control interval chosen.

$$\Delta J_1 \approx \frac{dJ_1}{d\lambda^+}\bigg|_{\lambda^+ \to 0} \lambda^+ . \tag{1.15}$$

As $u_2^*(\tau_m)$ regulates $\frac{dJ_1}{d\lambda^+} \approx \alpha_d$, (1.15) becomes $\Delta J_1 \approx \alpha_d \lambda^+$. Thus the choice of λ^+ and α_d allow the control designer to specify the desired degree of change in (1.3) provided by each $u_2^*(\tau_m)$. Also note that for a given $\frac{dJ_1}{d\lambda^+}$ and any choice of λ^+, (1.15) can be applied to test if $\lambda^+ \in V$ or that it at least provides a ΔJ_1 that is known to be achievable for $\lambda^+ \in V$.

Though the neighborhood where (1.15) provides a reasonable approximation varies depending on system, in practice it is fairly straightforward to select a $\lambda^+ \in V$. The easiest approach is to select a single conservative estimate for λ^+. This is analogous to choosing a small, fixed time step in finite differencing or Euler integration. However, to avoid a-priori selection of a $\lambda^+ \in V$ and unnecessarily conservative step sizes, we use a line search with a descent condition to select a $\lambda^+ \in V$ or one that provides a minimum change in cost (1.3). Described in [62], the line search iteratively reduces the duration from a default value. By continuity, the process is guaranteed to find a duration that produces a change in cost within tolerance of that predicted from (1.15). In implementation, we use an iteration limit to bound the algorithm in time. Note that failing to find an acceptable λ^+ within the iteration limit is not usually a concern because (1.7) provides a schedule of control values so a duration can be sought for any nearby time if the system is overly sensitive at the current time.

For a more thorough discussion on the results, see [7]. Next, we perform real-time simulation studies using ROS to show how Algorithm 1 can benefit human-robot collaboration in a rehabilitation setting.

1.3 Human-Robot Interaction in Assisted Balance Therapy

In the previous section, we provided the theoretical foundation for the calculation of optimal burst control signals (i.e. their magnitude and direction u_2^*, and their duration λ^+) when assistance is requested. Here, we consider the application of COR interfaces that use these signals in the field of rehabilitation. Stroke and spinal cord injuries (SCI) are the main causes of motor disability and severely affect the ability to control every-day tasks, such as standing, walking, reaching etc. [47, 22, 50]. During the past decades, there has been a substantial increase in the range, type, and number of robotic trainers for rehabilitation of motor impairments that follow stroke and SCI. Most of the robot-driven trainers currently in use, serve to augment the therapeutic tools available to the physical therapists and physicians, by guiding the patient's limbs through movement [32, 58, 56, 8]. However, the amount of assistance provided by the robot can have a large effect on the patient's recovery. To facilitate motor-neuron recovery, it is essential that patients actively participate in the training task [30]. It is for this reason that assist-as-needed algorithmic paradigms have shown promise for robotic therapy, inherently promoting patient participation by limiting assistance to the minimum the patient requires.

In the particular case of stroke rehabilitation, one of the main objectives is to improve posture stability [59]. In this section, we use the COR interface as an assist-as-requested controller that allows for an alternative robot-assisted training scheme where collaboration between the therapist and the robot is encouraged. In particular, therapist's guidance is integrated in real time with the assistive control, hence ensuring that therapy objectives are attained all while the therapist's work load and interference remains low. See Section 1.3.1 for an analysis of how our approach differs from existing assist-as-needed techniques.

Section 1.3.2 tests the real-time performance of a COR interface acting on a simulated standing figure, using the Robot Operating System (ROS), a standard platform for real robotic applications [1]. COR is shown to be suited for implementation on an embedded platform. In particular, our example verifies that real-time COR activation is feasible and requires only modest computational resources. In addition, it demonstrates how COR interfaces are effective even in the absence of force/pressure sensors which are usually required for assist-as-needed techniques [33, 8]. Note that a modified version of the optimal burst control in (1.7) is employed here, where AC is set to act in conjunction with NC when activated (as shown in Fig. 1.3). For a detailed description of this derivation, see [42].

1.3.1 Related work: assist-as-needed techniques

Most existing assist-as-needed techniques in rehabilitation therapy provide performance based assistance. This is a common solution with promising results that uses task performance metrics as thresholds to initiate robot assistance. Krebs et al. [35, 34] propose an algorithm that uses performance-triggered impedance control to guide the patient through task completion. Speed, time, or EMG thresholds have been tested as performance metrics. Impedance control, systematized by Hogan [29] in 1985, is particularly effective when applied to contact tasks. However, this approach usually requires force feedback and also constitutes a sustained control strategy where the robot is continuously applying support (i.e. after the specified threshold has been exceeded). Banala et al. [8] follow a similar approach to provide support for an active leg exoskeleton (ALEX). In particular, they employ force-field controllers where tangential forces are reduced as the patient's performance improves.

Wolbrecht et al. in [61] establish an assist-as-needed paradigm using a standard model-based adaptive controller and adding a novel force reducing term to the adaptive control law, which decays the force output from the robot when errors in task execution are small. The performance metric used in this technique is the kinematic error between the hand and goal location during reaching tasks.

Emken et al. in [21] formulate the "assist-as-needed" principle as an optimization problem. In particular, the robotic movement trainer is assumed to minimize a cost function that is the weighted sum of robot force and patient movement error. The authors find the controller that minimizes this cost function for the case in which motor recovery is modeled as a process of learning a novel sensory motor transformation. Here, step height achieved during walking is used as performance criterion. Although the approach has the advantage of being optimal, the resulting error-base controller is precalculated based on a fixed training model which does not allow for real-time model updates. For that, and because the controller is constantly active, possible inaccuracies of the selected model can have a negative impact on the long-term controller's performance as far as active patient participation is concerned.

Compared to the aforementioned control approaches that need formal performance metrics to be activated and remain active for a long time period, we follow an alternative approach that promotes real-time collaboration between the therapist and the robotic trainer. We can allow a therapist in the loop because the triggered optimal actions are only applied for a short duration of time and are calculated to assist the system as best as possible in the limited time frame available. This concept of shared control through intermittent bursts of optimal support comprises a unique assist-as-needed approach in that it completely avoids the issues of sustained control strategies where the interaction between two or more controllers acting on the same system simultaneously proves problematic[8]. An interactive simulation study

[8] Interestingly, this scheme of intermittent instead of sustained control has been suggested to be natively used by the central nervous system (CNS) in human posture control in [41, 40].

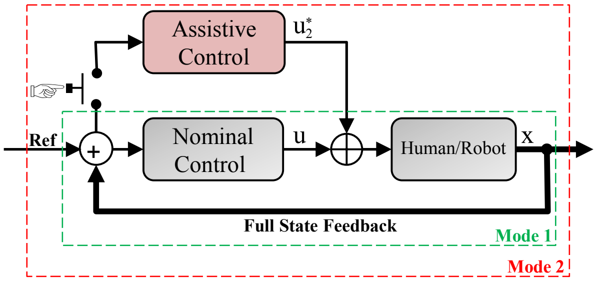


Fig. 1.3: Schematic representation of the COR interface used in balance therapy, consisting of the AC and NC.

provides more information about COR interfaces in the context of balance training in the following section.

1.3.2 *Interactive simulation study*

This section assesses the performance of the COR interface (shown in Fig. 1.3) in shared balance control of a simulated standing, humanoid figure allowed to move in the sagittal plane. Here, we use a double inverted pendulum to model the humanoid. This is a common abstraction used extensively to examine the dynamics of humanoid or human balance [39, 44, 36, 41, 52, 16]. Alternative methods commonly control this system using simplified center of mass (COM) and center of pressure (COP) models [46, 3]. In this work we demonstrate results directly controlling the nonlinear double pendulum model for two main reasons. First, it demonstrates the scalability of the proposed COR approach. In particular, we show that the controller is fast and applies to nonlinear dynamics in a way that directly generalizes to models with different number of links and actuator placement. Secondly, the choice to control to the double pendulum dynamics demonstrates flexibility. While the COR approach can be applied equally well to COP models, these models require specific force/pressure sensing hardware for feedback. Controlling to the double pendulum demonstrates how balance control can be achieved using alternative feedback based on joint angles and velocities.

The system model is illustrated in Fig. 1.4. Specifically, with both feet firmly planted to the ground and the knees locked in place, we allow 2 degrees of freedom, with the two pendulum pivot points corresponding to the hips and ankles respectively. The pendulum may be controlled through direct application of torque to the hip and ankle joints. Thus, the system has $n = 4$ states, i.e. the two joint angles $\boldsymbol{\theta}^{2\times1}$ and their velocities $\dot{\boldsymbol{\theta}}^{2\times1}$, and $m = 2$ inputs, i.e. the torques at the joints. The mass, inertia and length of the model segments were determined from anthropometric data provided by [60]. The pendulum equations of motion take the form in (1.2) and can be derived using Lagrangian Mechanics and the Euler-Lagrange equation. The derivation is straightforward and will not be shown here.

Our performance objective in this application is to drive the system to the upright position, i.e. to keep the center of gravity (COG) at the center of the base of support. As there are no particular constraints on our choice for the tracking cost function, we employ a quadratic performance metric of the form in (1.3), using the vertical body position as an angle reference so that $x_d = \mathbf{0}$.

For performance verification, we consider the example case where a human with impaired balance ability is in a standing position wearing a lower-limb robotic exoskeleton. In this scenario, the main system corresponds to the biomechanical model of a human augmented by a transparent overlying suit, while the NC plays the role of the central nervous system (CNS) attempting to maintain human balance. In literature, the CNS has been implemented both as a PID [44] and an optimal LQR controller [36]. Here, we chose to implement it as a PD controller providing feedback control on each of the joints. The AC is assumed to be embedded in the robotic suit. We emulate a scenario where the PD gains of the NC are insufficiently tuned and as a result, the provided torques fail to hold the figure upright.

The Robot Operating System (ROS), available online [1], lies in the center of the simulation setup, coordinating the real-time communication between the running processes. The humanoid balance model provided by the double inverted pendulum is simulated using the open-source software package called `trep` [2]. The system simulation initiates at time $t = 0$ given an initial state $(\boldsymbol{\theta_0}, \dot{\boldsymbol{\theta}}_\mathbf{0})$. At each time step, the program checks whether assistive control has been requested and enters one of the modes described next and shown in Fig. 1.3.

Mode 1: With no assistance requested, the process simulates the system (1.1) forward according to the NC input $u(t)$, and outputs the next system configuration.

Mode 2: If assistance is requested, the module solves for the state and co-state trajectories and evaluates the control formula (1.7) to compute the assistive controller (AC) output $u_2^*(t)$ (see Algorithm 1). It applies the control for duration λ^+, before switching back to the nominal mode.

The simulation runs indefinitely, allowing the user to activate the AC in real time as needed by pressing a button on the computer screen. For more information on the experimental setup, see [42]. Fig. 1.4 demonstrates a sequence of snapshots where the button is pushed three times sequentially and the figure is driven to the upright position.[9]

1.3.2.1 Results based on therapy objectives

We now present three simulation examples that illuminate the benefits of using a COR interface in robot-assisted balance training therapy. As an informal measure of performance, we specify approximate angle safety constraints at $\theta = \pm 0.2$. It is assumed that if the angles advance beyond these values, the Center of Gravity leaves

[9] As shown in Section 1.2, it is possible to provide parameters that guarantee local stability and convergence in the case when the Assistive Controller is continuously activated.

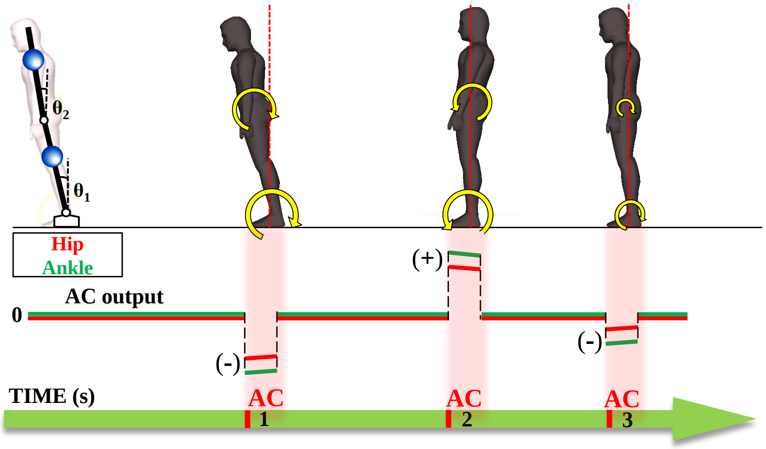


Fig. 1.4: Example of real-time standing balance control using the simulation setup. Assistive control (AC) is applied by pressing the button three times sequentially, eventually driving the system to the upright position.

the base of support and a fall occurs. Furthermore, we quantify the magnitude of nominal control effort as the energy E_{NC} of the signals $u^{(1)}(t)$ and $u^{(2)}(t)$, one for each joint. Specifically,

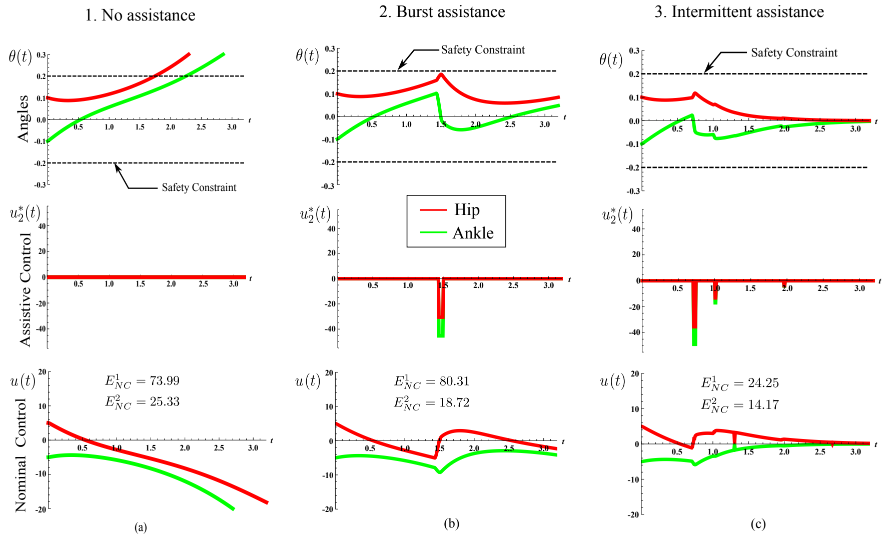
$$E_{NC} = \int_0^{T_f} ||u(t)||^2 dt \tag{1.16}$$

where T_f is either the final time of simulation T in Algorithm 1 or the time when the safety constraints are violated and the figure falls.

The COR parameters in Algorithm 1 were tuned as follows: the time horizon $T = 3.0s$, the default control duration $\Delta\tau_{init} = 0.03s$ and the desired tracking cost reduction $\alpha_d = -200$. The weight matrices in (1.6) and (1.3) were set as

$$Q = \begin{pmatrix} 0.8 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}. \tag{1.17}$$

The higher weights on the velocities ensure a more realizable AC performance where fast motions are avoided. The parameter α_d defines how aggressive the assistance should be, i.e. either a “gentle” or a “strong push” towards the balanced position. Therefore, by setting the absolute value of α_d relatively low, we allow for generation of more realizable and gentle controls for application to humans. Finally, by tuning the time horizon value T, we define how long the assistive control has to affect the change in cost. A value of 3.0s was selected, as it resulted in smooth angle trajectories (see Fig. 1.5).



(a) (b) (c)

Fig. 1.5: Three examples demonstrate the optimal COR benefits in robot-assisted balance therapy. u_2^* indicates robot-applied torques and u corresponds to human-applied torques. E_{NC} indicates total nominal control effort. (a) With no assistance, safety constraints are violated. (b) With burst assistance, balance is maintained with high energy contribution from the NC (preferred implementation). (c) With intermittent assistance, balance is ensured with low energy contribution from the NC.

In all cases described below, once the button was pressed, it took approximately 0.1s for the computation of the optimal assistive control in Python, on a laptop with an Intel Core i7 chipset. This confirms the feasibility of COR activation in real time.

Case 1 - No assistance: Consider the case where a standing patient is allowed to balance without assistance i.e. only the NC is active (Fig. 1.5(a)). With no interference from the robot ($u_2^*(t) = 0$), the joint angles advance beyond the safety constraints indicating a fall. Therefore, although the training goal of active patient's involvement is achieved ($E^1_{NC} = 73.99$, $E^2_{NC} = 25.33$), there is *insufficient automated support* to prevent falls.

Case 2 - Burst assistance: In the second example, a monitoring physical therapist/observer is allowed to apply optimal control-on-request, to provide on-line optimal assistance when deemed necessary. The effect of the assistive controller activation is illustrated in Fig. 1.5(b). In the previous case the unassisted model violated the safety constraints, now the patient is safely "pushed" back towards the upright position and a potential fall is prevented. What's interesting is that the patient is still actively applying torques ($u(t) > 0$) and thus applies almost the same amount of control effort as in the first case ($E^1_{NC} = 80.31$, $E^2_{NC} = 18.72$). The robot interferes with the human effort only for a very short duration, before ceding full control authority back to the NC. Thus, in this case, both training goals are achieved: *the patient is actively involved and balance is maintained*. Notice that this concept is similar to the action taken by a conventional therapist while assisting a person to maintain balance by gently "pushing" them to the right direction as needed. However, here, almost no effort is required from the side of the therapist (automated assistance).

Case 3 - Intermittent assistance: Our third example considers the case where the COR interface is activated multiple times intermittently (Fig. 1.5(c)). It can be observed that not only does the state remain inside the safety constraints but also it successfully reaches the origin. In practice, this means that a fall has been prevented and also that the upright posture has been achieved. However, in comparison to the second case, this approach is not ideal because *the patient is essentially held in the upright position by the robot* without using their own power ($E^1_{NC} = 24.25$, $E^2_{NC} = 14.17$).

Therefore, the second case is the most effective application of the optimal COR interface and indicates how an experienced therapist should take advantage of this control scheme to get the best therapy outcomes out of automated balance training.

1.4 Human-Robot Communication in Posture Reinforcement: a Short Study

In addition to employing COR interfaces to enhance therapist-robot collaboration, in this section we also explore its efficacy as a means for patient-robot communication in posture reinforcement. For this purpose, COR feedback is translated into vibrotactile cues to communicate intents through sensory augmentation. Previous studies by T. Murphey's group have provided evidence that task-based sensory augmentation yields nearly a factor of four improvement in time-to-failure for virtual balance tasks [55]. The idea of combining robotics and sensory substitution ([49, 20]) may be a viable alternative to conventional rehabilitation approaches. Several comparison studies have shown that neurologically intact people provided with tactile feedback can perform better than those acting on similar forms of auditory feedback; there are also cases in which tactile feedback is at least equally effective as vision,

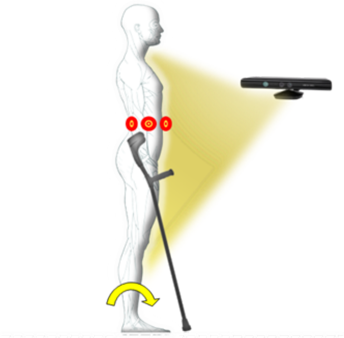


Fig. 1.6 An illustration of the subject's position during trials. The subject wears a vest with eight symmetrically positioned vibration motors around their torso (red circles) and uses crutches to help maintain balance while achieving different target postures.

if not more so ([53, 51]). A number of studies ([17, 38, 54]) conclude that tactile feedback can indeed be effective in promoting desired behaviors.

In this short study, it is hypothesized that optimal COR-generated tactile feedback on the torso can facilitate standing balance and prevent falls by providing cues for the correct placement of torso in real time. A preliminary experiment was designed to assess how a person perceives the optimal vibrotactile feedback generated by the COR interface and whether they are able to interpret it correctly to achieve a desired posture. The study was conducted with four healthy subjects. In the trials, a standing subject is wearing a vest with eight symmetrically positioned vibration motors on their chest and uses crutches to adjust their posture, moving back and forth (along the sagittal plane) and left and right (along the coronal plane). A Microsoft Kinect is used to capture sensory information about the subject's posture in real time. ROS collects and processes data coming in from the Kinect at approximately 30Hz. For an illustration of the subject's position during trials, see Fig. 1.6. The subject is instructed to hold their torso in a constant angle with respect to their hips so as to closely mimic the movement of a person wearing a hip-actuated exoskeleton.

During the main session, the subject is instructed to track eight subsequent random targets as fast as possible. Each target refers to a target ankle angle on the sagittal plane and a target ankle angle on the coronal plane, considering the upright

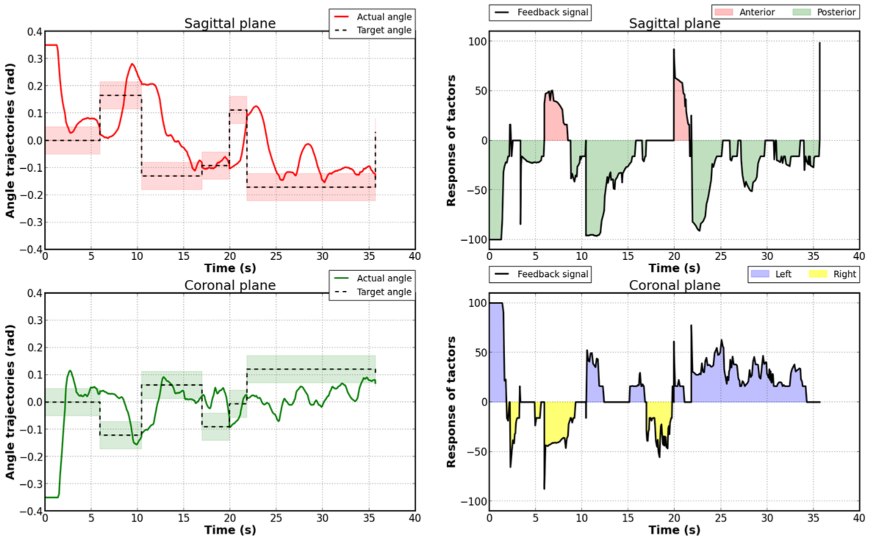


Fig. 1.7: The first and second row of diagrams correspond to the sagittal and coronal plane movement respectively. The first column shows the respective ankle angles with respect to time as they were recorded during the trials. The green/red shaded area around the target angle (black dashed line) illustrates the selected tolerance in angle which is dictated by the accuracy of the sensor used (i.e. Microsoft Kinect). The second column of plots shows the vibration of the tactors with respect to time, as generated by the optimal COR interface.

position as the reference (zero) angle for both planes. The target may also be perceived as a target body Center of Gravity (CoG) with these two definitions used interchangeably. A new target is assigned as soon as the previous target has been reached. There is no other indication that the target has been reached, other than the vibration stopping.

A sample response from a single trial is illustrated in Fig. 1.7. It was observed that the subjects were adequately familiarized to the vibrational signals after a short training session. The metric used for preliminary evaluation of COR-based vibrotactile feedback perception is the mean time-to-target (TTT) normalized over distance between initial CoG and target CoG. Among the four trials, TTT was found to decrease from an initial mean value of $40s/m$ over the first few targets, to approximately $20s/m$ by average for the last target. This preliminary result is an indication that sensory feedback perception based on COR interfaces improved over time for all four subjects resulting in an overall decreased reaction time to stimulus. In addition, optimal COR-generated signals were found to be sufficiently instructive to always drive the subject towards the target posture.

However, a few exceptions were noted. In particular, in instances where reaching the assigned CoG target required diagonal motion from the subject, TTT exploded (up to approximately 120 s/m) to account for the fact that subjects were incapable of interpreting the tactile signals as a cue for diagonal motion (i.e. on the sagittal and coronal plane simultaneously).

Overall, it was concluded that the subjects were able to interpret the COR-generated vibrotactile signals correctly in order to track the targets, while the mean time-to-target (TTT) normalized over distance was found to decrease over the first subsequent targets and remain fairly constant for the remaining targets. This preliminary result is promising in that it promotes an alternative implementation of COR interfaces as a means for human-robot collaboration through sensory augmentation.

1.5 Conclusion

This chapter presented an algebraic expression to compute optimal actions for nonlinear systems at any time. We demonstrated how a line search can resolve short durations to apply these actions to provide long time horizon improvement in tracking. Compared to standard methods where optimal controls are synthesized based on finite switching durations, these methods completely avoid constrained iterative optimization. They facilitate the rapid exchange between human and computer control required to enable a new shared control paradigm where automated assistance can be provided quickly and on-demand.

To demonstrate the performance of these enhanced COR interfaces, we utilized ROS to apply optimal control-on-request to standing balance therapy and posture reinforcement. In particular, the chapter focused on a shared control example where bursts of optimal assistance are applied to a simulated standing figure to prevent it from falling. This approach presents an opportunity for rehabilitation. Robot-assisted balance therapy is challenging as control authority is shared between the robot and the user, rendering their integration problematic. In an ideal rehabilitation setting, the robot should only provide as much assistance as needed to prevent a fall, requiring active patient participation. However, existing assistive control strategies either rely on sustained control or assume the use of force/pressure sensors which are generally not available in current over-ground lower-limb exoskeletons. To address these issues, we considered employing an optimal COR interface to assist therapy. The short duration of activation ensures the feasibility of our design in a shared control scenario, alleviating the need for force feedback. The controller is *easy to implement*, has *low computational demands* and *runs in real time*. Our real-time simulation studies showed that a therapist equipped with a COR interface can *ensure patient safety* while requiring *active participation*.

Finally, we used the proposed COR design to provide instructive vibrotactile feedback for posture reinforcement during standing. Preliminary results from a short human study indicated that the subjects are able to interpret the sensory cues correctly in order to achieve a target posture with a decreasing normalized mean time-to-target (TTT) over time.

References

1. Robot Operating System (2014). URL `http://www.ros.org/`
2. `trep` software package development (2014). URL `trep.googlecode.com`
3. Abdallah, M., Goswami, A.: A biomechanically motivated two-phase strategy for biped upright balance control. In: IEEE International Conference on Robotics and Automation, pp. 1996–2001 (2005)
4. Allgower, F., Findeisen, R., Nagy, Z.K.: Nonlinear model predictive control: From theory to application. Journal of the Chinese Institute of Chemical Engineers **35**(3), 299–316 (2004)
5. Allgöwer, F., Zheng, A.: Nonlinear model predictive control, vol. 26. Birkhäuser Basel (2000)
6. Anderson, B.D.O., Moore, J.B.: Optimal control: Linear quadratic methods. Prentice-Hall, Inc., NJ, USA (1990)
7. Ansari, A., Murphey, T.: Control-on-request: Short-burst assistive control for long time horizon improvement. In: American Control Conference, pp. 1173–1180 (2015)
8. Banala, S.K., Kim, S.H., Agrawal, S.K., Scholz, J.P.: Robot assisted gait training with active leg exoskeleton (ALEX). IEEE Transactions on Neural Systems and Rehabilitation Engineering **17**(1), 2–8 (2009)
9. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
10. Caldwell, T., Murphey, T.: Projection-based optimal mode scheduling. In: IEEE Conference on Decision and Control (CDC) (2013)
11. Camacho, E.F., Alba, C.B.: Model predictive control. Springer (2013)
12. Chen, Z., Haykin, S.: On different facets of regularization theory. Neural Computation **14**(12), 2791–2846 (2002)
13. Cipriani, C., Zaccone, F., Micera, S., Carrozza, M.C.: On the shared control of an EMG-controlled prosthetic hand: Analysis of user–prosthesis interaction. IEEE Transactions on Robotics **24**(1), 170–184 (2008)
14. De Vito, E., Rosasco, L., Caponnetto, A., Piana, M., Verri, A.: Some properties of regularized kernel methods. The Journal of Machine Learning Research **5**, 1363–1390 (2004)
15. Demeester, E., Hüntemann, A., Vanhooydonck, D., Vanacker, G., Van Brussel, H., Nuttin, M.: User-adapted plan recognition and user-adapted shared control: A bayesian approach to semi-autonomous wheelchair driving. Autonomous Robots **24**(2), 193–211 (2008)

16. Deng-Peng, X., Xu, L.: Multiple balance strategies for humanoid standing control. Acta Automatica Sinica **37**(2), 228–233 (2011)
17. Ding, Z.Q., Chen, I.M., Yeo, S.H.: The development of a real-time wearable motion replication platform with spatial sensing and tactile feedback. In: IEEE International Conference on Intelligent Robots and Systems, pp. 3919–3924 (2010)
18. Egerstedt, M., Wardi, Y., Axelsson, H.: Optimal control of switching times in hybrid systems. In: IEEE International Conference on Methods and Models in Automation and Robotics (2003)
19. Egerstedt, M., Wardi, Y., Axelsson, H.: Transition-time optimization for switched-mode dynamical systems. IEEE Transactions on Automatic Control **51**(1), 110–115 (2006)
20. Elbert, T., Sterr, A., Rockstroh, B., Pantev, C., Müller, M.M., Taub, E.: Expansion of the tonotopic area in the auditory cortex of the blind. The Journal of Neuroscience **22**(22), 9941–9944 (2002)
21. Emken, J.L., Bobrow, J.E., Reinkensmeyer, D.J.: Robotic movement training as an optimization problem: Designing a controller that assists only as needed. In: IEEE International Conference on Rehabilitation Robotics, pp. 307–312 (2005)
22. Forster, A., Young, J.: Incidence and consequences offalls due to stroke: A systematic inquiry. Bmj **311**(6997), 83–86 (1995)
23. Franklin, J.N.: Minimum principles for ill-posed problems. SIAM Journal on Mathematical Analysis **9**(4), 638–650 (1978)
24. Greena, Billinghurstb, Chena, Chasea: Human-robot collaboration: A literature review and augmented reality approach in design. International Journal of Advanced Robotic Systems **5**(1) (2008)
25. Griffiths, P., Gillespie, R.B.: Shared control between human and machine: Haptic display of automation during manual control of vehicle heading. In: Proceedings 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 358–366. IEEE (2004)
26. Grüne, L., Pannek, J.: Nonlinear model predictive control. Springer (2011)
27. Hansen, P.C.: The L-curve and its use in the numerical treatment of inverse problems. IMM, Department of Mathematical Modelling, Technical Universityof Denmark (1999)
28. Hespanha, J.P.: Linear systems theory. Princeton university press (2009)
29. Hogan, N.: Impedance control: An approach to manipulation. In: American Control Conference, pp. 304–313 (1984)
30. Hogan, N., Krebs, H.I., Rohrer, B., Palazzolo, J.J., Dipietro, L., Fasoli, S.E., Stein, J., Hughes, R., Frontera, W.R., Lynch, D., et al.: Motions or muscles? Some behavioral factors underlying robotic assistance of motor recovery. Journal of rehabilitation research and development **43**(5), 605 (2006)
31. Jadbabaie, A., Hauser, J.: On the stability of receding horizon control with a general terminal cost. IEEE Transactions on Automatic Control **50**(5), 674–678 (2005)
32. Jezernik, S., Colombo, G., Keller, T., Frueh, H., Morari, M.: Robotic orthosis lokomat: A rehabilitation and research tool. Neuromodulation: Technology at the neural interface **6**(2), 108–115 (2003)
33. Jezernik, S., Colombo, G., Morari, M.: Automatic gait-pattern adaptation algorithms for rehabilitation with a 4-DOF robotic orthosis. IEEE Transactions on Robotics and Automation **20**(3), 574–582 (2004)
34. Krebs, H.I., Hogan, N., Aisen, M.L., Volpe, B.T.: Robot-aided neurorehabilitation. IEEE Transactions on Rehabilitation Engineering **6**(1), 75–87 (1998)
35. Krebs, H.I., Palazzolo, J.J., Dipietro, L., Ferraro, M., Krol, J., Rannekleiv, K., Volpe, B.T., Hogan, N.: Rehabilitation robotics: Performance-based progressive robot-assisted therapy. Autonomous Robots **15**(1), 7–20 (2003)
36. Kuo, A.D.: An optimal control model for analyzing human postural balance. IEEE Transactions on Bio-Medical Engineering **42**(1), 87–101 (1995)
37. Lee, J.H.: Model predictive control: Review of the three decades of development. International Journal of Control, Automation and Systems **9**(3), 415–424 (2011)

38. Lieberman, J., Breazeal, C.: TIKL: Development of a wearable vibrotactile feedback suit for improved human motor learning. IEEE Transactions on Robotics **23**(5), 919–926 (2007)
39. Liu, C., Atkeson, C.G.: Standing balance control using a trajectory library. In: IEEE International Conference on Intelligent Robots and Systems, pp. 3031–3036 (2009)
40. Loram, I.D., Gollee, H., Lakie, M., Gawthrop, P.J.: Human control of an inverted pendulum: Is continuous control necessary? Is intermittent control effective? Is intermittent control physiological? The Journal of Physiology **589**(2), 307–324 (2011)
41. Loram, I.D., Lakie, M.: Human balancing of an inverted pendulum: Position control by small, ballistic-like, throw and catch movements. The Journal of Physiology **540**(3), 1111–1124 (2002)
42. Mavrommati, A., Ansari, A., Murphey, T.: Optimal control-on-request: An application in real-time assistive balance control. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 5928–5934 (2015)
43. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.: Constrained model predictive control: Stability and optimality. Automatica **36**(6), 789–814 (2000)
44. Milton, J., Cabrera, J.L., Ohira, T., Tajima, S., Tonosaki, Y., Eurich, C.W., Campbell, S.A.: The time-delayed inverted pendulum: Implications for human balance control. Chaos: An Interdisciplinary Journal of Nonlinear Science **19**(2), 026,110–026,110 (2009)
45. Naidu, D.S.: Optimal control systems, vol. 2. CRC press (2002)
46. Ott, C., Roa, M.A., Hirzinger, G.: Posture and balance control for biped robots based on contact force optimization. In: IEEE International Conference on Humanoid Robots, pp. 26–33 (2011)
47. Perry, J., Garrett, M., Gronley, J.K., Mulroy, S.J.: Classification of walking handicap in the stroke population. Stroke **26**(6), 982–989 (1995)
48. Philips, J., del R.Millan, J., Vanacker, G., Lew, E., Galan, F., Ferrez, P., Van Brussel, H., Nuttin, M.: Adaptive shared control of a brain-actuated simulated wheelchair. In: IEEE 10th International Conference on Rehabilitation Robotics (ICORR), pp. 408–414 (2007)
49. Rauschecker, J.P.: Compensatory plasticity and sensory substitution in the cerebral cortex. Trends in neurosciences **18**(1), 36–43 (1995)
50. Sekhon, L.H., Fehlings, M.G.: Epidemiology, demographics, and pathophysiology of acute spinal cord injury. Spine **26**(24S), S2–S12 (2001)
51. Sklar, A.E., Sarter, N.B.: Good vibrations: Tactile feedback in support of attention allocation and human-automation coordination in event-driven domains. Human Factors: The Journal of the Human Factors and Ergonomics Society **41**(4), 543–552 (1999)
52. Stephens, B.: Integral control of humanoid balance. In: IEEE International Conference on Intelligent Robots and Systems, pp. 4020–4027 (2007)
53. Sun, M., Ren, X., Cao, X.: Effects of multimodal error feedback on human performance in steering tasks. Journal of Information Processing **18**, 284–292 (2010)
54. Tzorakoleftherakis, E., Bengtson, M.C., Mussa-Ivaldi, F., Scheidt, R.A., Murphey, T.D.: Tactile proprioceptive input in robotic rehabilitation after stroke. In: IEEE International Conference on Robotics and Automation (2015)
55. Tzorakoleftherakis, E., Mussa-Ivaldi, F., Scheidt, R., Murphey, T.D.: Effects of optimal tactile feedback in balancing tasks: A pilot study. In: IEEE American Control Conference, pp. 261 – 270 (2014)
56. Veneman, J.F., Kruidhof, R., Hekman, E.E., Ekkelenkamp, R., Van Asseldonk, E.H., Van Der Kooij, H.: Design and evaluation of the lopes exoskeleton robot for interactive gait rehabilitation. IEEE Transactions on Neural Systems and Rehabilitation Engineering **15**(3), 379–386 (2007)
57. Wardi, Y., Egerstedt, M.: Algorithm for optimal mode scheduling in switched systems. In: American Control Conference, pp. 4546–4551 (2012)
58. West, R.G.: Powered gait orthosis and method of utilizing same (2004). US Patent 6,689,075
59. Winstein, C., Gardner, E., McNeal, D., Barto, P., Nicholson, D.: Standing balance training: Effect on balance and locomotion in hemiparetic adults. Archives of Physical Medicine and Rehabilitation **70**(10), 755–762 (1989)

60. Winter, D.A.: Biomechanics and motor control of human movement. John Wiley & Sons (2009)
61. Wolbrecht, E.T., Chan, V., Reinkensmeyer, D.J., Bobrow, J.E.: Optimizing compliant, model-based robotic assistance to promote neurorehabilitation. IEEE Transactions on Neural Systems and Rehabilitation Engineering **16**(3), 286–297 (2008)
62. Wright, S., Nocedal, J.: Numerical optimization, vol. 2. Springer New York (1999)
63. Yu, H., Spenko, M., Dubowsky, S.: An adaptive shared control system for an intelligent mobility aid for the elderly. Autonomous Robots **15**(1), 53–66 (2003)
64. Zabczyk, J.: Mathematical control theory: An introduction. Springer (2009)