A Dynamical System Approach for Adaptive Grasping, Navigation and Co-Manipulation with Humanoid Robots

Nadia Figueroa*†, Salman Faraji*, Mikhail Koptev and Aude Billard

***Abstract*— In this paper, we present an integrated approach that provides compliant control of an iCub humanoid robot and adaptive reaching, grasping, navigating and co-manipulating capabilities. We use state-dependent dynamical systems (DS) to (i) coordinate and drive the robots hands (in both position and orientation) to grasp an object using an intermediate virtual object, and (ii) drive the robot's base while walking/navigating. The use of DS as motion generators allows us to adapt smoothly as the object moves and to re-plan on-line motion of the arms and body to reach the object's new location. The desired motion generated by the DS are used in combination with a whole-body compliant control strategy that absorbs perturbations while walking and offers compliant behaviors for grasping and manipulation tasks. Further, the desired dynamics for the arm and body can be learned from demonstrations. By integrating these components, we achieve unprecedented adaptive behaviors for whole body manipulation. We showcase this in simulations and real-world experiments where iCub robots (i) walk-to-grasp objects, (ii) follow a human (or another iCub) through interaction and (iii) learn to navigate or co-manipulate an object from human guided demonstrations; whilst being robust to changing targets and perturbations.**

I. Introduction

One of the current goals in robotics is to deploy humanoid robots that can interact and aid humans in our everyday lives. A plethora of applications can be envisioned, be it in homes, the work-place and smart-factories that would benefit from such robotic assistants. Examples include reaching for and picking-and-placing objects, carrying objects with a human (or with another robot) and navigating through constrained environments – while carrying or pushing objects (Fig.1). To perform these tasks in dynamic environments, the robots should be: (i) *adaptive* during task/motion planning and (ii) *compliant* during both manipulation and locomotion.

Many humanoid robots are controllable solely in position. Position-control is often seen as leading to stiff behavior. In this work we show that we can achieve real-time adaptive and compliant behavior with a position-controlled iCub robot, by tackling the following challenges in an integrated approach:

1) *Adaptive* whole-body motion planning.
2) *Balancing* during reaching, grasping and manipulation.
3) *Compliant* arm control for manipulation tasks.
4) *Compliant* walking – robust to perturbations.

Adaptive motion planning for robots with many degrees of freedom (DoF), such as humanoids, is still an open problem [1]. The underlying issue is that most of the approaches, for both manipulation and locomotion, rely on offline motion planning and trajectory optimization algorithms [2], [3], [4]. In this work, instead of using a motion planning approach to optimize for a desired trajectory, we adopt a dynamical system (DS)-based motion generation approach. DS have become the go-to method in robotics to provide *adaptability* during motion planning, thanks to their ability to generate on-line motion plans inherently robust to uncertainties and changes in dynamic environments [5], [6], [7].

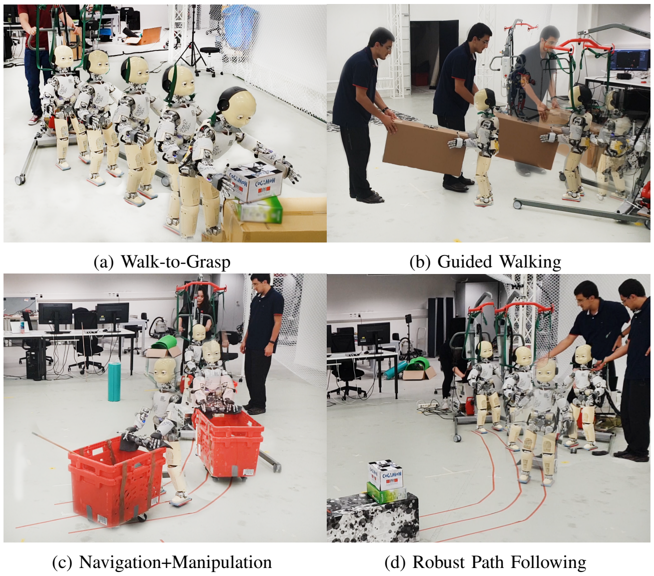
(a) Walk-to-Grasp
(b) Guided Walking
(c) Navigation+Manipulation
(d) Robust Path Following

Fig. 1: Tasks achieved with our proposed approach.

DS-based Motion Planning: In a DS-based motion planning problem, we assume that the motion of a robotic system is defined in state space $\boldsymbol{\xi} \in \mathbb{R}^M$ and constrained by a system of ODEs (Ordinary Differential Equation). Let $\mathbf{f}(\boldsymbol{\xi})$ be a first-order, autonomous DS describing a nominal motion plan,

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi}) \quad \left\{ \lim_{t\to\infty} \|\boldsymbol{\xi} - \boldsymbol{\xi}^*\| = 0 \right. \tag{1}$$

where $\mathbf{f}(\cdot) : \mathbb{R}^M \to \mathbb{R}^M$ is a continuous differentiable vector-valued function representing a DS that converges to a single stable equilibrium point $\boldsymbol{\xi}^*$; i.e. target or attractor. DS-based motion generators are more often used in simpler robotic systems; i.e. generating motion in Cartesian task-space of an end-effector or of a mobile platform. Notably, the use of DS-motion generation in humanoids is limited as using a single DS to control the entire state-space of a multi-DoF humanoid can become infeasible. Hence, we take a divide-and-conquer approach, and instead generate the whole-body motion of the robot with two decoupled DS (**Challenge 1**):

- *grasp-DS*: Generates coordinated velocities of both hands to grasp an object in synchrony. (Section II-A).

All authors are with Learning Algorithm and Systems Laboratory, EPFL.
†Corresponding author, email: nadia.figueroafernandez@epfl.ch
*Nadia Figueroa and Salman Faraji hold equal contribution in this work.

- *walk-DS*: Generates velocities for the base of the robot to reach a target or follow a path. (Section IV-.1, IV).

When the robot should grasp or manipulate an object, the *grasp-DS* is activated. To achieve the desired motion from the *grasp-DS* while *balancing*, we use an inverse kinematics (IK) method which directly solves for joint positions given a set of desired Cartesian tasks (**Challenge 2**). This method is implemented as a non-linear optimization which also exploits the sparsity of Jacobians to solve the whole-body problem in less than $2ms$ [8]. If the task requires the robot to walk towards a target or follow a path, the *walk-DS* is activated. Transitioning between these two states is achieved by running feasibility/reachability check routines.

To exploit the robustness and adaptability of DS-based motion generators, the robot must be compliant [9], [10]. *Compliant* behavior in robotic systems can be achieved either passively (due to mechanical design) or actively (due to the control design). Recent efforts in humanoid robot design have achieved compliant torque-controlled humanoids with flexible (or passively compliant) joints (COMAN [11], M2V2 [12] and ATLAS). Recent works have provided walking and/or balance/fall recovery schemes for such torque-controlled robots by exploiting their natural dynamics [13], [14], [15]. These strategies, however, are difficult to transfer to position-controlled robots. To provide compliance when manipulating an object, we extend the IK-balancing method described above by adding compliant control laws to the Cartesian tasks in order to track grasping forces and be compliant to perturbations (**Challenge 3**). More recent whole-body methods calculate accelerations through inverse-dynamics to produce the desired tasks [16], [17], [18]. These formulations incorporate physical limitations of the robot as well, i.e. torque limits and contact friction cones. While with these constraints one can push the robot to the extremes, we assume slower motions and only include positions in our IK. The method in [18] directly applies the resulting joint torques on the body which can bring whole-body compliance. However, due to missing torque sensors in iCub, we adopt and simplify the methods in [16], [17] and use the 6D force/torque sensors to achieve compliance in the arms.

Compliant Position-Controlled Locomotion: In this work, we adopt a compliant walking algorithm developed in [19] which stabilizes a position-controlled robot dynamically by adjusting footstep positions. This method solves inverse kinematics and keeps the Center of Mass (CoM) on top of the ankle joints. In presence of asymmetries, e.g. external forces or internal shift of CoM, the robot starts falling as it minimally relies on the foot size and ankle torques. By predicting future steps using a simplified model of the robot called 3LP [20], our method calculates plausible next footstep locations to capture the fall and restore balance. The 3LP model takes both stance and swing dynamics into account and can produce faster motions [21] compared to other push-recovery methods in the literature [22], [23]. With this algorithm, external forces can move the robot in any direction while foot-placement resists moderately. This makes the robot compliant, behaving like a global damper in the environment (**Challenge 4**). Unlike recent works in position-controlled walking [24], our controller does not need any reference trajectory, since walking naturally emerges from an interaction between foot-stepping and asymmetries.

TABLE I: Notation

Variable	**Domain**	**Description**
$\boldsymbol{\xi}_l, \boldsymbol{\xi}_r$	$\in \mathbb{R}^3 \times \mathbb{S}^3$	Left/right hand poses, $\boldsymbol{\xi}_{l/r} = [\boldsymbol{x}_{l/r}, \boldsymbol{q}_{l/r}]$
$\boldsymbol{x}_l, \boldsymbol{x}_r$	$\in \mathbb{R}^3$	Left/right hand positions.
$\boldsymbol{q}_l, \boldsymbol{q}_r$	$\in \mathbb{S}^3 \subset \mathbb{R}^4$	Left/right hand orientations.
$\dot{\boldsymbol{\xi}}_l, \dot{\boldsymbol{\xi}}_r$	$\in \mathbb{R}^6$	Hand velocities $\dot{\boldsymbol{\xi}}_{l/r} = [\dot{\boldsymbol{x}}_{l/r}, \boldsymbol{\omega}_{l/r}]$
$\dot{\boldsymbol{x}}_{l/r}, \boldsymbol{\omega}_{l/r}$	$\in \mathbb{R}^3, \in \mathbb{R}^3$	Hand linear and angular velocities.
$\boldsymbol{x}$	$\in \mathbb{R}^2$	2D position on the xy-plane of the CoM.
θ_x	$[-2\pi, 2\pi]$	Heading of the CoM wrt. x-axis of the world.
$\dot{\boldsymbol{x}}, \omega_x$	$\in \mathbb{R}^2, \in \Re$	Linear and angular velocity on the xy-plane of the CoM of the robot's base.
$\mathbf{q}$	$\in \mathbb{R}^M$	Joint angles of a robot with M-DoF.

Novelty and Paper Contributions: The novelty of this work lies in the integration and extension of three sets of techniques developed previously by the co-authors, namely DS-based coordinated planning of reach to grasp motion [25], whole-body balancing [8] and compliant walking [19]. We extended the methods as follows:

– *grasp-DS (Section II):* In [25] we introduced a DS-based coordination strategy that allows multiple robot arms to coordinate with each other, with a moving object or target. The approach smoothly transitions between (i) a *synchronous* behavior, where the robot arm's task is to coordinate with each other to simultaneously reach for a moving object and, (ii) an *asynchronous* behavior, where each robot arm has its own target or desired motion. In this work, we adopt this motion planning strategy to transition between (i) generating reaching-to-grasp motions for a humanoid robot and (ii) an ideal standing posture when the object is not reachable. We simplify [25] and add quaternion orientation dynamics.

– *Compliant walking (Section III):* We showcase how our compliant position-controlled walking algorithm [19] can make a first generation iCub (10 years old) walk. We extend the algorithm to detect the intention of a collaborator through interaction, by initiating walking in any direction given displacements and interaction forces at the arm level.

– We use the compliant walking approach to collect trajectories and learn desired complex navigation behaviors with a state-of-the-art DS-based learning scheme [26] *(Section IV)*.

Project webpage, including videos and code found here: `https://epfl-lasa.github.io/iCub-Assistant/`.

II. Reaching, Grasping and Manipulating Objects while Balancing

A. Coordinated Multi-Arm DS-based Motion Planner

If the object is reachable and feasible to grasp, the *grasp-DS* generates the desired motion of the left/right hands as,

$$\dot{\boldsymbol{\xi}}_{l/r} = \begin{bmatrix} \dot{\boldsymbol{x}}_{l/r} \\ \boldsymbol{\omega}_{l/r} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{x}_{l/r}(\boldsymbol{x}_{l/r}, \boldsymbol{x}^{*}_{l/r}, \boldsymbol{x}^{v}_{l/r}) \\ \mathbf{f}^{q}_{l/r}(\boldsymbol{q}_{l/r}, \boldsymbol{q}^{*}_{l/r}, \boldsymbol{q}^{v}_{l/r}). \end{bmatrix} \tag{2}$$

Each DS is composed of both *synchronous* and *asynchronous* behaviors for position and orientation, and coupled to the pose of the object ($\boldsymbol{\xi}^{o} = [\boldsymbol{x}^{o}, \boldsymbol{q}^{o}] \in \mathbb{R}^3 \times \mathbb{S}^3$) through a virtual object ($\boldsymbol{\xi}^{v} = [\boldsymbol{x}^{v}, \boldsymbol{q}^{v}] \in \mathbb{R}^3 \times \mathbb{S}^3$) that tracks the real object. Each DS in (2) either tracks the grasping pose on the virtual

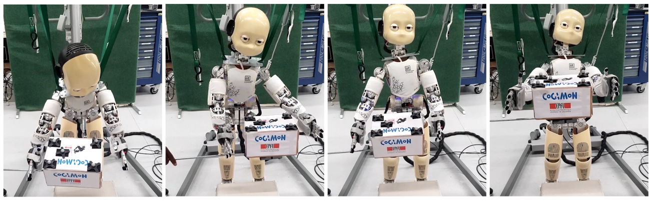

Fig. 2: iCub following a moving object with the *virtual-object* DS.

object, defined by $\boldsymbol{\xi}^v_{l/r} = [\boldsymbol{x}^v_{l/r}, \boldsymbol{q}^v_{l/r}] \in \mathbb{R}^3 \times \mathbb{S}^3$, or converges to an ideal standing posture $\boldsymbol{\xi}^*_{l/r} = [\boldsymbol{x}^*_{l/r}, \boldsymbol{q}^*_{l/r}] \in \mathbb{R}^3 \times \mathbb{S}^3$.

1) Position Grasp DS: The position *grasp-DS*, $\mathbf{f}^x_{l/r}(\cdot)$, modulates these two behaviors by a synchronization parameter $0 \le \tau_x(t) \le 1$ as shown below,

$$\begin{aligned}\dot{\boldsymbol{x}}_{l/r} = \;& \tau_x(t)\dot{\boldsymbol{x}}^v_{l/r} + \dot{\tau}_x(t)(\boldsymbol{x}^v_{l/r} - \boldsymbol{x}^*_{l/r}) \\ & + \mathbf{A}^x_{l/r}\Big(\boldsymbol{x}_{l/r} - \tau_x(t)\boldsymbol{x}^v_{l/r} + (\tau_x(t) - 1)\boldsymbol{x}^*_{l/r}\Big)\end{aligned} \tag{3}$$

with $\mathbf{A}^x_{l/r} \in \mathbb{R}^{3\times3}$ being the linear system matrix that defines the convergence rate for each hand's desired Cartesian position. $\tau_x(t) \in \Re$ is a continuously differentiable function in the range of $[0, 1]$ as in [25]. $\boldsymbol{x}^v_{l/r}, \dot{\boldsymbol{x}}^v_{l/r} \in \mathbb{R}^3$ are the position/velocity of the grasping points on the virtual object.

Proposition 1: The position *grasp-DS* (3) globally asymptotically converges to the attractor $\tau_x(t)\boldsymbol{x}^v_{l/r} + (1 - \tau_x(t))\boldsymbol{x}^*_{l/r}$, i.e.

$$\lim_{t\to\infty} ||\boldsymbol{x}_{l/r} - \tau_x(t)\boldsymbol{x}^v_{l/r} + (\tau_x(t) - 1)\boldsymbol{x}^*_{l/r}|| = 0 \tag{4}$$

if the following condition hold,

$$\Big\{\mathbf{A}^x_{l/r} = (\mathbf{A}^x_{l/r})^T \prec 0 \tag{5}$$

Proof: Follows immediately from the proof of Theoreom 1 in [25] and using $V(x) = \frac{1}{2}(\boldsymbol{x}_{l/r} - \tau_x\boldsymbol{x}^v_{l/r} + (\tau_x - 1)\boldsymbol{x}^*_{l/r})^T(\boldsymbol{x}_{l/r} - \tau_x\boldsymbol{x}^v_{l/r} + (\tau_x - 1)\boldsymbol{x}^*_{l/r})$ as its candidate Lyapunov function. ■

2) Orientation Grasp DS: To generate the desired rotational motion of the robot's hands, the orientation *grasp-DS* (2) transitions between tracking the orientation of the virtual object ($\boldsymbol{q}^v_{l/r}(t) \in \mathbb{S}^3$) or of the ideal posture ($\boldsymbol{q}^*_{l/r} \in \mathbb{S}^3$). This transition is controlled by the switching variable $\tau_q = [0, 1] \in \mathbb{Z}$ which yields $\tau_q = 1$ when the object is reachable and $\tau_q = 0$ otherwise. Unlike (3), τ_q is not a continuous time-varying function. Due to hardware limitations on the wrist joints, rather than interpolating between the rotational behaviors we transition only when the object is reachable or not. Thus, the orientation *grasp-DS*, $\mathbf{f}^q_{l/r}(\cdot)$ is[1],

$$\begin{aligned}\boldsymbol{\omega}_{l/r} = \tau_q\boldsymbol{\omega}^v_{l/r} + \mathbf{A}^q_{l/r}\Big(&\tau_q k^v_q(\cdot)\log(\boldsymbol{q}_{l/r} \otimes \bar{\boldsymbol{q}}^v_{l/r}) \\ &+ (1 - \tau_q)k^*_q(\cdot)\log(\boldsymbol{q}_{l/r} \otimes \bar{\boldsymbol{q}}^*_{l/r})\Big)\end{aligned} \tag{6}$$

with $\mathbf{A}^q_{l/r} \in \mathbb{R}^{3\times3}$ being the linear system matrix. $\boldsymbol{q}^v_{l/r}, \boldsymbol{\omega}^v_{l/r}$ are the orientation and angular velocity of the grasping points defined on the virtual object. $k^{v/*}_q(\cdot)$ are defined in (8).

Theorem 1: The orientation *grasp-DS* (6) globally asymptotically converges to the attractor $\tau_q\boldsymbol{q}^v_{l/r} + (1 - \tau_q)\boldsymbol{q}^*_{l/r}$, i.e.

$$\lim_{t\to\infty} ||\tau_q \log(\boldsymbol{q}^v_{l/r} \otimes \bar{\boldsymbol{q}}_{l/r}) + (1 - \tau_q)\log(\boldsymbol{q}^*_{l/r}, \otimes\bar{\boldsymbol{q}}_{l/r})|| = 0 \tag{7}$$

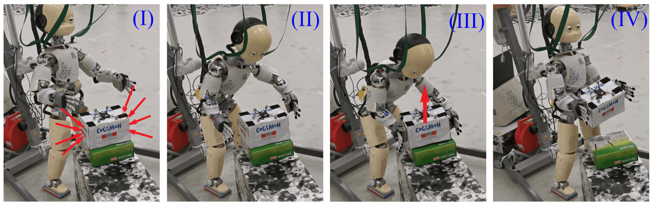

Fig. 3: An iCub grasping an object and standing upwards.

if the following conditions hold,

$$\begin{cases}\mathbf{A}^q_{l/r} = (\mathbf{A}^q_{l/r})^T \prec 0 \\ k^{v/*}_q(\boldsymbol{q}_{l/r}, \boldsymbol{q}^{v/*}_{l/r}) = \frac{||\text{vec}(\boldsymbol{q}_{l/r}\otimes\bar{\boldsymbol{q}}^{v/*}_{l/r})||}{\arccos(\text{scalar}(\boldsymbol{q}_{l/r}\otimes\bar{\boldsymbol{q}}^{v/*}_{l/r}))}\end{cases} \tag{8}$$

Proof: See Appendix B. ■

3) Grasp DS Behavior: When $\tau_x(t) = \tau_q = 1$, the object is within the robot's reachable workspace. If (5),(8) are ensured, (3),(6) perfectly track the *virtual object*-DS (Fig.2),

$$\dot{\boldsymbol{x}}_{l/r} = \dot{\boldsymbol{x}}^v_{l/r} + \mathbf{A}^x_{l/r}(\boldsymbol{x}_{l/r} - \boldsymbol{x}^v_{l/r}) \tag{9}$$

$$\boldsymbol{\omega}_{l/r} = \boldsymbol{\omega}^v_{l/r} + \mathbf{A}^q_{l/r}k^v_q(\cdot)\log(\boldsymbol{q}_{l/r} \otimes \bar{\boldsymbol{q}}^v_{l/r}) \tag{10}$$

When the object is no longer reachable $\tau_x(t) = \tau_q = 0$, (3) and (6) converge towards the ideal standing posture pose,

$$\dot{\boldsymbol{x}}_{l/r} = \mathbf{A}^x_{l/r}(\boldsymbol{x}_{l/r} - \boldsymbol{x}^*_{l/r}) \tag{11}$$

$$\boldsymbol{\omega}_{l/r} = \mathbf{A}^q_{l/r}k^*_q(\cdot)\log(\boldsymbol{q}_{l/r} \otimes \bar{\boldsymbol{q}}^*_{l/r}) \tag{12}$$

4) Virtual Object Dynamics: (3) and (6) rely on a DS describing the motion of the virtual object, which tracks the motion of the real object. Formulations of the *virtual object-DS* proposed in [27], [25] modulate this tracking behavior by a coordination parameter $\gamma(t) \in \Re_{>0}$. This is a continuously differentiable function that controls the level of coordination between the virtual object ($\boldsymbol{\xi}^v$) and the real object ($\boldsymbol{\xi}^o$). It used to consider the feasibility of grasping the object at the predicted intercept point. In this work, we do not predict a feasible intercept point, as we trigger the *grasp-DS* solely when the object is reachable and feasible to grasp. Hence, we simplify the formulations from [27], [25] and extend it by adding an orientation dynamics as follows,

$$\dot{\boldsymbol{\xi}}^v = \begin{bmatrix}\dot{\boldsymbol{x}}^v \\ \boldsymbol{\omega}^v\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{x}}^o + \mathbf{A}^x_v(\boldsymbol{x}^v - \boldsymbol{x}^o) \\ \boldsymbol{\omega}^o + \mathbf{A}^q_v k^o_v(\cdot)\log(\boldsymbol{q}^v \otimes \bar{\boldsymbol{q}}^o)\end{bmatrix}. \tag{13}$$

(13) describes the tracking dynamics of the virtual object wrt. the real object. The virtual object is a geometric replica of the real object centered at the mid-point between the left and right hands. $\mathbf{A}^x_o, \mathbf{A}^q_o \in \mathbb{R}^{3\times3}$ are the linear system matrices.

Theorem 2: The *virtual object-DS* defined in (13) globally asymptotically converge to the the attractors $\boldsymbol{x}^o$ and $\boldsymbol{\omega}^o$, i.e.

$$\begin{cases}\lim_{t\to\infty} ||\boldsymbol{x}^v - \boldsymbol{x}^o|| = 0, & \lim_{t\to\infty} ||\dot{\boldsymbol{x}}^v - \dot{\boldsymbol{x}}^o|| = 0 \\ \lim_{t\to\infty} ||\log(\boldsymbol{q}^v \otimes \bar{\boldsymbol{q}}^o)|| = 0, & \lim_{t\to\infty} ||\boldsymbol{\omega}^v - \boldsymbol{\omega}^o|| = 0\end{cases} \tag{14}$$

if the following conditions hold,

$$\begin{cases}\mathbf{A}^x_v = (\mathbf{A}^x_v)^T \prec 0, \quad \mathbf{A}^x_q = (\mathbf{A}^x_q)^T \prec 0 \\ k^o_v(\boldsymbol{q}^o, \boldsymbol{q}^v) = \frac{||\text{vec}(\boldsymbol{q}^o\otimes\bar{\boldsymbol{q}}^v)||}{arccos(\text{scalar}(\boldsymbol{q}^o\otimes\bar{\boldsymbol{q}}^v))}\end{cases} \tag{15}$$

Proof: The position DS proof is trivially derived by following Theorem 2 in [25] and using $V(x) = \frac{1}{2}(\boldsymbol{x}^o + \boldsymbol{x}^v)^T(\boldsymbol{x}^o + \boldsymbol{x}^v)$. The orientation DS proof follows the steps of proof of Theorem 1, when using $V(\boldsymbol{q}) = (s^o - s^v)^2 + ||\boldsymbol{u}^o - \boldsymbol{u}^v||^2$. ■

[1] See Appendix A for preliminaries on quaternion operations.

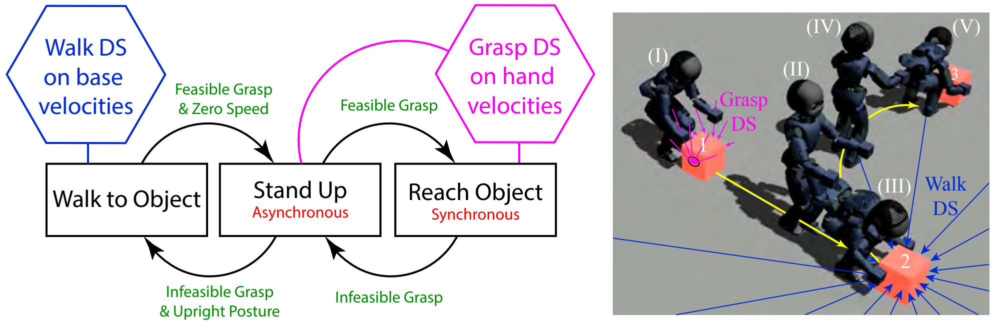

Fig. 4: iCub reaching an object with changing targets. At (I) the iCub follows a *grasp-DS* towards the object at location 1. When the object moves to 2, the iCub stands up (II) and follows a *walk-DS* towards it (III), etc.

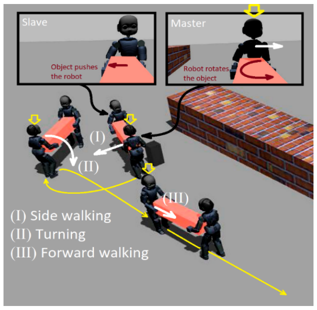

Fig. 5: Two iCubs carrying an object as a master-slave.

B. Inverse Kinematics and Compliant Arm Control

To follow the desired hand motion generated by (3) and (6) while keeping balance and guaranteeing kinematic limits, we propose an IK solver for position control which extends previous work on singularity-tolerant IK [8]. This method solves a nonlinear optimization problem online to satisfy a set of desired Cartesian tasks. Such tasks are defined in a vector $\Xi \in \mathbb{R}^{N<M}$, and include an ideal pose of the CoM ($\boldsymbol{x}^d_b \in \mathbb{R}^3 \times \mathbb{S}^3$), ideal positions of the left/right foot ($\boldsymbol{x}^d_{lf}, \boldsymbol{x}^d_{rf} \in \mathbb{R}^3$) and the desired hand poses ($\boldsymbol{\xi}^d_{l/r} \in \mathbb{R}^3 \times \mathbb{S}^3$); i.e. $\Xi = [\boldsymbol{\xi}^d_b, \boldsymbol{x}^d_{lf}, \boldsymbol{x}^d_{rf}, \boldsymbol{\xi}^d_l, \boldsymbol{\xi}^d_r]$. $\boldsymbol{\xi}^d_b, \boldsymbol{x}^d_{lf}, \boldsymbol{x}^d_{rf}$ are set to predefined ideal postures, the desired hand poses are computed by integrating the *grasp-DS* (3) and (6) as follows,

$$\boldsymbol{\xi}^d_{l/r} = \begin{bmatrix} \boldsymbol{x}^d_{l/r} \\ \boldsymbol{q}^d_{l/r} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{l/r} + \dot{\boldsymbol{x}}_{l/r}\Delta t \\ \exp\left(\boldsymbol{\omega}_{l/r}\Delta t/2\right) \otimes \boldsymbol{q}_{l/r} \end{bmatrix} \tag{16}$$

Given $\Xi \in \mathbb{R}^{N<M}$, we search for the desired joint angles, $\mathbf{q} \in \mathbb{R}^M$, by solving the quadratic optimization problem:

$$\begin{aligned} &\min_{\Delta\mathbf{q},\boldsymbol{\delta}} \boldsymbol{\delta}^T\mathbf{Q}\boldsymbol{\delta} + \Delta\mathbf{q}^T\mathbf{R}\Delta\mathbf{q} \\ &h(\mathbf{q}^-) + \frac{\partial h(\mathbf{q}^-)}{\partial \mathbf{q}^-}\Delta\mathbf{q} = \Xi + \begin{bmatrix} \boldsymbol{\delta} \\ 0 \end{bmatrix} \\ &\mathbf{q}^l \leq (\mathbf{q}^- + \Delta\mathbf{q}) \leq \mathbf{q}^u \end{aligned} \tag{17}$$

where $\mathbf{q}^- \in \mathbb{R}^M$ are the previous joint angles and $\Delta\mathbf{q} \in \mathbb{R}^M$ is the motion adjustment to be found. The matrix $\mathbf{R} \in \mathbb{R}^{M\times M}$ is equivalent to the well-known damping term in least-squares IK methods and $\mathbf{Q} \in \mathbb{R}^{N\times N}$ is a diagonal matrix that sets weights for the Cartesian tasks. $h(\mathbf{q})$ and $J(\mathbf{q}) = \frac{\partial h(\mathbf{q})}{\partial \mathbf{q}}$ denote the forward kinematics and Jacobian functions. $\mathbf{q}^l, \mathbf{q}^u \in \mathbb{R}^M$ are lower/upper joint limits and $\boldsymbol{\delta} \in \mathbb{R}^N$ are slack variables to be minimized. After solving (17) the desired joint angles are computed as $\mathbf{q} = \mathbf{q}^- + \Delta\mathbf{q}$. To grasp (or hold) objects with a desired force we modify the Cartesian tasks defined in (16), to track a desired force:

$$\tilde{\boldsymbol{x}}^d_{l/r} = \boldsymbol{x}^d_{l/r} + k_F(\boldsymbol{F}^d_{l/r} - \boldsymbol{F}^m_{l/r})\Delta t \tag{18}$$

where $\boldsymbol{F}^m_{l/r}, \boldsymbol{F}^d_{l/r} \in \mathbb{R}^3$ are the measured and desired Cartesian forces with the latter being computed as, $\boldsymbol{F}^d_{l,r} = \frac{\bar{\boldsymbol{x}}^d_{l,r} - \boldsymbol{x}^d_{l,r}}{||\bar{\boldsymbol{x}}^d_{l,r} - \boldsymbol{x}^d_{l,r}||}(1 - g(t))F_{\max}$ with $\bar{\boldsymbol{x}}^d_{l,r} = 1/2(\bar{\boldsymbol{x}}^d_r + \bar{\boldsymbol{x}}^d_l)$ representing the mid-point between the desired left and right hand positions. $k_F \in \Re_+$ in (18) is a force tracking gain, $F_{\max} \in \Re_+$ is the max. desired grasping force and $g(t) = g(t-1)(1 - k_G\Delta t)$ is a decaying function with $g(0) = 1$.

C. Feasible Grasping/Manipulating while Balancing

To achieve grasping while balancing (Fig. 2 and 3), we must verify two important feasibility issues before commanding velocities to the robot.

1) Reachability/Feasibility: We ensure that the object is reachable and a feasible whole-body grasping posture exists by solving (17) with the current pose of the real object.

2) Self-Collision Detection: To avoid collided configurations we perform a self-collision test every time we compute a new feasible posture from the IK. To detect collisions we use a mesh model of the robot derived from the official `.sdf` files and the open-source library FCL [28]. Full self-collision assessment of a given posture takes $< 0.2ms$. The entire routine is executed at every control tick and takes $6-7ms$. We validate this approach in real-world experiments where the iCub tracks a moving object, grasps and manipulates it while maintaining balance, see *accompanying video*.

III. Compliant and Adaptive Walking-to-Grasp and Guided Walking

The approach presented in Section II provides adaptive manipulation behaviors thanks to the merging of DS-based motion planning and compliant control. We follow the same approach to achieve such behavior during locomotion.

A. Walking DS-based Motion Planner

The simplest *walk-DS*, $\mathbf{f}_w(\boldsymbol{x})$, can take the form of $\dot{\boldsymbol{x}} = \mathbf{A}_w(\boldsymbol{x} - \boldsymbol{x}^*)$, where $\boldsymbol{x}^* \in \mathbb{R}^2$ is the desired 2D position of the object or robot on the xy-plane and $\mathbf{A}_w \prec 0 \in \mathbb{R}^{2\times 2}$ is a linear system matrix. $\mathbf{f}_w(\boldsymbol{x})$ generates linear trajectories towards the object from anywhere in the state-space as depicted in Fig. 4. In Section IV, we introduce means to learn non-linear DS to generate complex navigation behaviors. The desired angular velocity of the base, $\omega_x \in \Re$, is computed by aligning the heading angle of the robot to the direction of motion given by a *walk-DS*. Let $\theta_x \in [-2\pi, 2\pi]$ be the current heading angle wrt. the x-axis of the world coordinate frame, then ω_x is computed as $\omega_x = -k_b\Big(\theta_x - \frac{\mathbf{f}_w(\boldsymbol{x})\cdot\mathbf{I}_x}{||\mathbf{f}_w(\boldsymbol{x})||}\Big)$ where $k_b \in \Re$ is a gain and $\mathbf{I}_x$ is the world x-axis.

B. Compliant Walking Controller

To provide compliant walking for a position-controlled robot we use the time-projection foot-stepping controller [19], [21]. In this method, the robot naturally steps in-place by regulating the CoM on top of the ankle joints (to minimize the ankle torques). The feet are commanded to follow arc

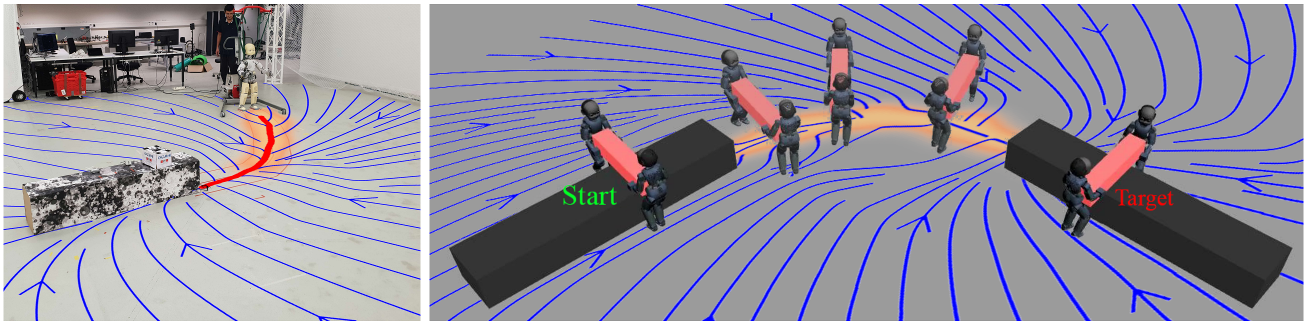

Fig. 6: Tasks Learned from Demonstrations: **(left)** Scenario 1: single iCub navigation towards target **(right)** Scenario 2: co-manipulation with 2 iCubs. The orange shaded regions denote trajectories used for learning/shaping the DS, which are depicted as the super-imposed blue vector fields.

trajectories via IK, and the joints track desired joint positions by PID controllers. Locomotion is achieved when the robot is being pulled/pushed externally, or by shifting its CoM horizontally. With these asymmetric disturbances, the controller in the stance hip joint makes the robot fall naturally, while the foot-stepping method suggests new footsteps to recover from falling. Hence, given the desired CoM velocity from the *walk-DS*, this time-projection controller is capable of naturally generating sagittal, lateral and turning motions to achieve it. Further, the robot behaves like a damper in the environment, not resisting against external forces, but stepping along their direction until they disappear.

Guided Walking and Co-Manipulation: Since both the walking controller and the arm control are compliant (Section II-B), we can make multiple agents (robot-robot or human-robot) interact with each other while carrying objects. We showcase this with a pair of humanoid robots picking up an object on the floor, lifting it up and then carrying the object while avoiding obstacles (see Fig. 5). Each robot separately and tasked to grasp one side of the object and lift it compliantly (following the approach presented in Section II). When both robots start walking, we can command desired walking velocities to one of them or both. In the master-slave combination, one robot is commanded and the other behaves as a damper and slows the motion down. To improve performance, we ask the passive slave robot to measure hand displacement with respect to the default hand posture and use it as an indication of motion intention in the other robot. This displacement is inversely proportional to the task-space impedance of the arms. When multiplied by positive-feedback gains, it produces desired velocities for the slave robot, decreasing resistance and improving walking speed. We showcase this in simulation with two iCubs and in a real-world experiment with an iCub and a human, see *video*.

C. Adaptive Walking-to-Grasp Strategy

We now extend the feasible grasping/manipulation approach presented in Section II-C to include walking. Specifically, when the object is not reachable, the robot walks towards it following a *walk-DS*. When the object becomes reachable, the robot stops and follows the *grasp-DS* (16). If the object suddenly moves away, the *grasp-DS* (16) moves the robot back to a default standing posture. Once reaching the upright posture, the robot starts walking again with the *walk-DS* towards the new target. Note that, in order to transition from walking to reaching, the robot must stop and stand in an upright posture, as depicted in the state-machine of Fig. 4. We showcase this capability in real-world experiments in which an iCub grasps an object while being perturbed by a human and while dynamically changing the location of the object, see *accompanying video*.

IV. Learning to Navigate with DS

With our compliant walking controller, when following a desired velocity, the robot will be compliant to external forces that will deviate it from a desired path. Hence, in scenarios as the ones depicted in Fig. 6 where a single (or team) of robots should follow a specific path, a linear DS for walking (as in Section IV-.1) is no longer suitable. To alleviate this, in this work, we propose to learn a non-linear *walk-DS* from demonstrations gathered via guided walking.

1) Learning DS-based Walking Motion Planners: Works in DS-based learning from demonstration encode a non-linear DS as a Linear Parameter Varying (LPV)-DS [29], [30], [31], [26], [32] that converges to an attractor $\boldsymbol{x}^*$ as,

$$\dot{\boldsymbol{x}} = \mathbf{f}_w(\boldsymbol{x}) = \sum\nolimits_{k=1}^{K} \gamma_k(\boldsymbol{x})(\mathbf{A}_w^k \boldsymbol{x} + \boldsymbol{b}_w^k) \qquad (19)$$

where $\gamma_k(\boldsymbol{x})$ is a state-dependent mixing function that must be $0 < \gamma_k(\boldsymbol{x}) \leq 1$ and $\sum_{k=1}^{K} \gamma_k(\boldsymbol{x}) = 1$. Further, $\boldsymbol{b}_w^k = -\mathbf{A}_w^k \boldsymbol{x}^* \; \forall k = 1, \dots, K$.

Proposition 2: The nonlinear walking DS (19) globally asymptotically converges to the attractor $\boldsymbol{x}^*$, if,

$$\Big\{ (\mathbf{A}_w^k)^T \mathbf{P} + \mathbf{P}\mathbf{A}_w^k \prec \mathbf{Q}_k, \quad \mathbf{Q}_k = \mathbf{Q}_k^T \prec 0 \quad \forall k = 1, \dots, K \qquad (20)$$

Proof: See proof of Proposition 2 in [26]. ■

We follow the approach introduced in [26] to learn the parameters of (19) from the demonstrated walking trajectories. Demonstrations are collected by (i) performing the task with a human (using the compliant walking controller presented in Section III-B an shown in Fig. 1b) for Fig. 6 (left) or (ii) by performing the task in Gazebo simulation via joystick commands for Fig. 6 (right). Given the demonstrated walking trajectories, $\gamma(\boldsymbol{x})$ is learned as a Gaussian Mixture Model (GMM) that is automatically fitted with a physically-consistent Bayesian non-parametric approach that discovers the optimal K number of Gaussians/DS. The set of linear DS parameters $\{\mathbf{A}_w^k, \boldsymbol{b}_w^k, \}_{k=1}^{K}$ is then learned by solving a constrained semi-definite programming problem ensuring (20). The learned DS are depicted as vector fields in Fig. 6.

2) Scenario 1: Real-World Experiments: As shown in the *accompanying video*, and Fig. 1c and 1d, the same DS learned from guided demonstrations is used in two applications. First, we use the compliant arm control strategy from Section II-B to grasp a cart (Fig. 1c) and track a desired force. The learned DS is then used to drive the CoM of the robot, resulting in an iCub pushing a cart while following a desired path. Then, we use the same DS to perform the walk-to-grasp strategy from Section III-C. If we want the robot to follow precisely the path, one can add additional shaping to the LPV-DS to make it stiff around the desired trajectory, as in the stiff-DS shaping approach introduced in [32]. As shown in Fig. 1d, thanks to the shaping of the DS with reference trajectories the robot is able to recover, follow the path while ultimately reaching/grasping the object.

3) Scenario 2: Simulated Experiment: Here we use the learned DS to generate the motion of the object, which is grasped by both robots following the approach from Section II. In this case, the desired velocity of the robot is transformed to desired linear/angular velocities of the CoM as we do in the guided walking approach (Fig. 5). As shown in Fig. 6 and in the *video*, the robots are capable of picking the object from one conveyor belt and sliding it onto a second conveyor belt by following the learned DS.

V. Discussion and Future Work

We proposed an integrated approach that provides adaptive and compliant behaviors for position-controlled humanoids. The approach relies on leveraging DS-based motion planning [25], [26] with IK-based whole-body balancing [8] and time-projection walking control [19]. While the approach was validated in a variety of adaptive and collaborative tasks, there is still room for improvement. Firstly, the feasibility-check routine does not actively avoid self-collisions. Instead, if the computed posture from the IK leads to a collision we simply stop the motion. We are currently developing an approach to learn a self-collision avoidance function, as in [25], that can be used as a constraint in the IK solver. Further, in the case of co-manipulation, the set of coupled robotics system is not proven to be stable. Current work is focused on such proof to find principles for tuning the positive-feedback gains in case of having more agents involved in the task.

Appendix

A. Preliminaries on Quaternion Math and Rotational Motion

The unit quaternion $\boldsymbol{q} \in \mathbb{S}^3 \subset \mathbb{R}^4$ is defined as,

$$\boldsymbol{q} = \begin{bmatrix} s \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \text{scalar}(\boldsymbol{q}) \\ \text{vec}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\boldsymbol{n} \end{bmatrix} \tag{21}$$

where $\boldsymbol{u} = [u_x, u_y, u_z]^T \in \mathbb{R}^3$, $s \in \Re$ and θ,$\boldsymbol{n}$ are the angle and normalized axis of rotations in the axis-angle representation. $\otimes$ is the symbol for a quaternion product, computed as follows,

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} s_1 s_2 - \boldsymbol{u}_1^T \boldsymbol{u}_2 \\ s_1 \boldsymbol{u}_2 + s_2 \boldsymbol{u}_1 + \mathbf{S}(\boldsymbol{u}_1)\boldsymbol{u}_2 \end{bmatrix} \tag{22}$$

with $\mathbf{S}(\boldsymbol{u}) \in \mathbb{R}^{3\times 3}$ being a skew-symmetric matrix with the following known properties: (i) $\mathbf{S}(\boldsymbol{u})^T = -\mathbf{S}(\boldsymbol{u})$, (ii) $\mathbf{S}(\boldsymbol{u})\boldsymbol{u} = 0$ and (iii) $\mathbf{S}(\boldsymbol{u}_1)\boldsymbol{u}_2 = -\mathbf{S}(\boldsymbol{u}_2)\boldsymbol{u}_1$. Even though the quaternion product is non-commutative, i.e. $\boldsymbol{q}_1 \otimes \bar{\boldsymbol{q}}_2 \neq \boldsymbol{q}_2 \otimes \bar{\boldsymbol{q}}_1$ thanks to the properties of skew-symmetric matrices the following equivalences hold:

$$\begin{aligned} \text{scalar}(\boldsymbol{q}_1 \otimes \bar{\boldsymbol{q}}_2) &= \text{scalar}(\boldsymbol{q}_2 \otimes \bar{\boldsymbol{q}}_1) \\ \text{vec}(\boldsymbol{q}_1 \otimes \bar{\boldsymbol{q}}_2) &= -\text{vec}(\boldsymbol{q}_2 \otimes \bar{\boldsymbol{q}}_1) \end{aligned}. \tag{23}$$

Finally, $\bar{\boldsymbol{q}} = [s, -\boldsymbol{u}^T]^T$ is the quaternion conjugate. The angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ required to rotate $\boldsymbol{q}_2$ onto $\boldsymbol{q}_1$ is computed as $\boldsymbol{\omega} = 2\log(\Delta\boldsymbol{q})/\Delta t$, where $\Delta\boldsymbol{q} = \boldsymbol{q}_1 \otimes \bar{\boldsymbol{q}}_2$ and $\log(\cdot) : \mathbb{S}^3 \to \mathbb{R}^3$ is computed as follows,

$$\log(\Delta\boldsymbol{q}) = \log(\begin{bmatrix} s \\ \boldsymbol{u} \end{bmatrix}) = \begin{cases} \arccos(s)\frac{\boldsymbol{u}}{||\boldsymbol{u}||} & \text{if } ||\boldsymbol{u}|| > 0 \\ [0,0,0]^T, & \text{otherwise} \end{cases} \tag{24}$$

which is the geometric logarithm equation that takes into account the singularity at quaternion $\boldsymbol{q} = [-1,0,0,0]^T$ [33]. $\Delta\boldsymbol{q}$ is the rotation difference from $\boldsymbol{q}_2 \to \boldsymbol{q}_1$, computed as $\Delta\boldsymbol{q} = \boldsymbol{q}_1 \otimes \bar{\boldsymbol{q}}_2$. The evolution of a time varying unit quaternion with angular velocity, $\boldsymbol{\omega}$, is defined by,

$$\dot{\boldsymbol{q}} = \frac{1}{2}\tilde{\boldsymbol{\omega}} \otimes \boldsymbol{q} \to \dot{\boldsymbol{q}} = \begin{bmatrix} \dot{s} \\ \dot{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\boldsymbol{u}^T\boldsymbol{\omega} \\ \frac{1}{2}(s\mathbf{I} - \mathbf{S}(\boldsymbol{u}))\boldsymbol{\omega} \end{bmatrix} \tag{25}$$

with $\tilde{\boldsymbol{\omega}} = [0, \boldsymbol{\omega}^T]^T$ [34]. The next desired quaternion is computed as $\boldsymbol{q}(t+\Delta t) = \exp(\boldsymbol{\omega}(t)\Delta t/2) \otimes \boldsymbol{q}(t)$, with $\exp(\cdot) : \mathbb{R}^3 \to \mathbb{S}^3$,

$$\exp(\boldsymbol{\omega}\Delta t/2) = \begin{cases} \begin{bmatrix} \cos(||\boldsymbol{\omega}\Delta t/2||) \\ \frac{\boldsymbol{\omega}}{||\boldsymbol{\omega}||}\sin(||\boldsymbol{\omega}\Delta t/2||) \end{bmatrix} & \text{if } ||\boldsymbol{\omega}\Delta t/2|| > 0 \\ [1,0,0,0]^T, & \text{otherwise} \end{cases} \tag{26}$$

if we limit the domain of (26) to $0 \leq ||\boldsymbol{\omega}|| \leq \pi$, it becomes a one-to-one mapping and (24) can be considered its inverse as described in [33], [35].

B. Proof of Orientation Grasp DS

We propose the following Lyapunov function:

$$V(\boldsymbol{q}) = (\tau_q s^v + \bar{\tau}_q s^* - s)^2 + ||\tau_q \boldsymbol{u}^v + \bar{\tau}_q \boldsymbol{u}^* - \boldsymbol{u}||^2 \tag{27}$$

where $\bar{\tau}_q = 1 - \tau_q$. (27) is positive definite, radially unbounded and continuously differentiable. (27) holds for both hands; i.e. $\boldsymbol{q} = \boldsymbol{q}_{l/r}$, to improve readability, we drop the l/r indices for all variables. Hence, to prove global asymptotic stability of (6) towards $\tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*$ the following conditions must hold: (i) $V(\tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*) = 0$, (ii) $V(\boldsymbol{q}) > 0 \; \forall \boldsymbol{q} \in \mathbb{S}^3 \setminus \boldsymbol{q} = \tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*$, (iii) $\dot{V}(\tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*) = 0$ and (iv) $\dot{V}(\boldsymbol{q}) < 0 \; \forall \boldsymbol{q} \in \mathbb{S}^3 \setminus \boldsymbol{q} = \tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*$. From (27) it is straightforward to see that conditions (i) and (ii) hold. To prove conditions (iii) and (iv) we compute the time derivative of (27) as below,

$$\begin{aligned} \dot{V}(\boldsymbol{q}) &= 2(\tau_q s^v + \bar{\tau}_q s^* - s)(\tau_q \dot{s}^v - \dot{s}) + 2(\tau_q \boldsymbol{u}^v + \bar{\tau}_q \boldsymbol{u}^* - \boldsymbol{u}^T(\tau_q \dot{\boldsymbol{u}}^v - \dot{u}) \\ &= 2(\tau_q s^v + \bar{\tau}_q s^* - s)(\tau_q \dot{s}^v) + 2(\tau_q \boldsymbol{u}^v + \bar{\tau}_q \boldsymbol{u}^* - \boldsymbol{u})^T(\tau_q \dot{\boldsymbol{u}}^v) \\ &\quad - 2(\tau_q s^v + \bar{\tau}_q s^* - s)(\dot{s}) - 2(\tau_q \boldsymbol{u}^v + \bar{\tau}_q \boldsymbol{u}^* - \boldsymbol{u}^T)\dot{\boldsymbol{u}} \\ &= \tau_q\Big(-(\tau_q s^v + \bar{\tau}_q s^* - s)((\dot{\boldsymbol{u}}^v)^T\boldsymbol{\omega}) + (\tau_q \boldsymbol{u}^v + \bar{\tau}_q \boldsymbol{u}^* - \boldsymbol{u})^T(s^v\mathbf{I} - \mathbf{S}(\boldsymbol{u}^v))\boldsymbol{\omega}^v\Big) \\ &\quad - (\tau_q s^v + \bar{\tau}_q s^* - s)(\dot{\boldsymbol{u}}^T\boldsymbol{\omega}) - (\tau_q \boldsymbol{u}^v + \bar{\tau}_q \boldsymbol{u}^* - \boldsymbol{u})^T(s\mathbf{I} - \mathbf{S}(\boldsymbol{u}))\boldsymbol{\omega} \\ &= -\tau_q(\boldsymbol{\omega}^v)^T\Big(\underbrace{(-s\boldsymbol{u}^v + s\boldsymbol{u} - \mathbf{S}(\boldsymbol{u})\boldsymbol{u}^v)}_{\text{vec}(\boldsymbol{q}\otimes\bar{\boldsymbol{q}}^v)} + \bar{\tau}_q\underbrace{(s^*\boldsymbol{u}^v - s^v\boldsymbol{u}^* - \mathbf{S}(\boldsymbol{u}^v)\boldsymbol{u}^*)}_{\text{vec}(\boldsymbol{q}^v\otimes\bar{\boldsymbol{q}}^*)}\Big) \\ &\quad + \boldsymbol{\omega}^T\Big(\tau_q\underbrace{(s^v\boldsymbol{u} - s\boldsymbol{u}^v - \mathbf{S}(\boldsymbol{u})\boldsymbol{u}^v)}_{\text{vec}(\boldsymbol{q}\otimes\bar{\boldsymbol{q}}^v)} + \bar{\tau}_q\underbrace{(s^*\boldsymbol{u} - s\boldsymbol{u}^* - \mathbf{S}(\boldsymbol{u})\boldsymbol{u}^*)}_{\text{vec}(\boldsymbol{q}\otimes\bar{\boldsymbol{q}}^*)}\Big) \end{aligned} \tag{28}$$

Given that (via equivalences (23)): $\tau_q\text{vec}(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^v) + \bar{\tau}_q\text{vec}(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^*) = \text{vec}(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^*) - \tau_q\Big(\text{vec}(\boldsymbol{q}^v \otimes \bar{\boldsymbol{q}}^*)\Big)$, (28) becomes:

$$\dot{V}(\boldsymbol{q}) = \Big(\tau_q\text{vec}(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^v) + \bar{\tau}_q\text{vec}(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^*)\Big)^T\Big(\boldsymbol{\omega} - \tau_q\boldsymbol{\omega}^v\Big) \tag{29}$$

Now, substituting (6) into (29)

$$\begin{aligned} \dot{V}(\boldsymbol{q}) &= \Big(\tau_q\text{vec}(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^v) + \bar{\tau}_q\text{vec}(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^*)\Big)^T\Big(\tau_q\boldsymbol{\omega}^v - \tau_q\boldsymbol{\omega}^v \\ &\qquad + \mathbf{A}^q\Big(\tau_q\underbrace{k_q^v(\boldsymbol{q},\boldsymbol{q}^v)}_{(8)}\log(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^v) + \bar{\tau}_q\underbrace{k_q^*(\boldsymbol{q},\boldsymbol{q}^v)}_{(8)}\log(\boldsymbol{q} \otimes \bar{\boldsymbol{q}}^*)\Big) \\ &= \Big(\tau_q\text{vec}(\boldsymbol{q}^v \otimes \bar{\boldsymbol{q}}) + \bar{\tau}_q\text{vec}(\boldsymbol{q}^* \otimes \bar{\boldsymbol{q}})\Big)^T\underbrace{\mathbf{A}^q}_{\prec 0\ (8)}\Big(\tau_q\text{vec}(\boldsymbol{q}^v \otimes \bar{\boldsymbol{q}}) + \bar{\tau}_q\text{vec}(\boldsymbol{q}^* \otimes \bar{\boldsymbol{q}})\Big) \leq 0 \end{aligned} \tag{30}$$

From (30) we see that $\dot{V}(\tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*) = 0$ and $\dot{V}(\boldsymbol{q}) < 0 \; \forall \boldsymbol{q} \in \mathbb{S}^3 \setminus \boldsymbol{q} = \tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*$. Hence, if (8) are met, (6) is globally asymptotically stable at the attractor $\tau_q\boldsymbol{q}^v + \bar{\tau}_q\boldsymbol{q}^*$ i.e., $\lim_{t\to\infty} ||\tau_q\text{vec}(\boldsymbol{q}^v \otimes \bar{\boldsymbol{q}}) + \bar{\tau}_q\text{vec}(\boldsymbol{q}^* \otimes \bar{\boldsymbol{q}})|| = 0$ which implies that $\lim_{t\to\infty} ||\tau_q\log(\boldsymbol{q}^v \otimes \bar{\boldsymbol{q}}) + \bar{\tau}_q\log(\boldsymbol{q}^* \otimes \bar{\boldsymbol{q}})|| = 0.$■

References

[1] N. Vahrenkamp, C. Scheurer, T. Asfour, J. Kuffner, and R. Dillmann, “Adaptive motion planning for humanoid robots,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2008, pp. 2127–2132.

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

[3] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *2014 IEEE-RAS International Conference on Humanoid Robots*, Nov 2014, pp. 295–302.

[4] C. Sung, T. Kagawa, and Y. Uno, “Whole-body motion planning for humanoid robots by specifying via-points,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 7, p. 300, 2013.

[5] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, Oct 1992.

[6] H. Jacob, S. Feder, and J. jacques E. Slotine, “Real-time path planning using harmonic potentials in dynamic environment,” in *in Proceedings of the IEEE International Conference on Robotics and Automation*, 1997, pp. 874–881.

[7] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, “Provably safe and robust learning-based model predictive control,” *Automatica*, vol. 49, no. 5, pp. 1216 – 1226, 2013.

[8] S. Faraji and A. J. Ijspeert, “Singularity-tolerant inverse kinematics for bipedal robots: An efficient use of computational power to reduce energy consumption,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1132–1139, April 2017.

[9] K. Kronander and A. Billard, “Passive interaction control with dynamical systems,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 106–113, 2016.

[10] M. Khoramshahi and A. Billard, “A dynamical system approach to task-adaptation in physical human-robot interaction,” *Autonomous Robots*, 2018.

[11] N. G. Tsagarakis, S. Morfey, G. Medrano Cerda, L. Zhibin, and D. G. Caldwell, “Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 673–678.

[12] J. Pratt and B. Krupp, “Design of a bipedal walking robot,” in *Unmanned Systems Technology X*, G. R. Gerhart, D. W. Gage, and C. M. Shoemaker, Eds., vol. 6962, International Society for Optics and Photonics. SPIE, 2008, pp. 480 – 492. [Online]. Available: https://doi.org/10.1117/12.777973

[13] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, “Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models,” *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[14] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, “Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid,” *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, 2012.

[15] M. Shafiee-Ashtiani, A. Yousefi-Koma, M. Shariat-Panahi, and M. Khadiv, “Push recovery of a humanoid robot based on model predictive control and capture point,” in *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, Oct 2016, pp. 433–438.

[16] S. Dafarra, F. Romano, and F. Nori, “Torque-controlled stepping-strategy push recovery: Design and implementation on the icub humanoid robot,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 152–157.

[17] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, “Quadratic programming for multirobot and task-space force control,” *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 64–77, Feb 2019.

[18] L. Sentis, J. Park, and O. Khatib, “Compliant control of multicontact and center-of-mass behaviors in humanoid robots,” *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, June 2010.

[19] S. Faraji, P. Müllhaupt, and A. Ijspeert, “Imprecise dynamic walking with time-projection control,” *arxiv*, 2018. [Online]. Available: http://infoscience.epfl.ch/record/260430

[20] S. Faraji and A. J. Ijspeert, “3lp: A linear 3d-walking model including torso and swing dynamics,” *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 4, pp. 436–455, 2017.

[21] S. Faraji, H. Razavi, and A. J. Ijspeert, “Bipedal walking and push recovery with a stepping strategy based on time-projection control,” *The International Journal of Robotics Research (IJRR)*, vol. 38, no. 5, pp. 587–611, 2019.

[22] P. Kryczka, P. Kormushev, N. G. Tsagarakis, and D. G. Caldwell, “Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 3352–3357.

[23] C. Zhou, Z. Li, X. Wang, N. Tsagarakis, and D. Caldwell, “Stabilization of bipedal walking based on compliance control,” *Autonomous Robots*, vol. 40, no. 6, pp. 1041–1057, Aug 2016.

[24] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, “A versatile and efficient pattern generator for generalized legged locomotion,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3555–3561.

[25] S. Mirrazavi, N. Figueroa, and A. Billard, “A unified framework for coordinated multi-arm motion planning,” *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1205–1232, 2018.

[26] N. Figueroa and A. Billard, “A physically-consistent bayesian non-parametric mixture model for dynamical system learning,” in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 87. PMLR, 29–31 Oct 2018, pp. 927–946.

[27] S. S. M. Salehian, N. Figueroa, and A. Billard, “Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty,” in *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[28] J. Pan, S. Chitta, and D. Manocha, “Fcl: A general purpose library for collision and proximity queries,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 3859–3866.

[29] S. Khansari-Zadeh and A. Billard, “Learning stable nonlinear dynamical systems with gaussian mixture models,” *Robotics, IEEE Transactions on*, vol. 27, no. 5, pp. 943–957, Oct 2011.

[30] H. Ravichandar, I. Salehi, and A. Dani, “Learning partially contracting dynamical systems from demonstrations,” in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 369–378.

[31] C. Blocher, M. Saveriano, and D. Lee, “Learning stable dynamical systems using contraction theory,” in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, June 2017, pp. 124–129.

[32] N. Figueroa, “From high-level to low-level robot learning of complex tasks: Leveraging priors, metrics and dynamical systems,” Ph.D. dissertation, Lausanne, Switzerland, 2019. [Online]. Available: http://infoscience.epfl.ch/record/270640

[33] A. Ude, B. Nemec, T. Petri, and J. Morimoto, “Orientation in cartesian space dynamic movement primitives,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2997–3004.

[34] M. Saveriano, F. Franzel, and D. Lee, “Merging position and orientation motion primitives,” in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 7041–7047.

[35] A. Ude, “Filtering in a unit quaternion space for model-based object tracking,” *Robotics and Autonomous Systems*, vol. 28, no. 2, pp. 163 – 172, 1999, intelligent Autonomous Systems (IAS-5).